

To our customers,

Old Company Name in Catalogs and Other Documents

On April 1st, 2010, NEC Electronics Corporation merged with Renesas Technology Corporation, and Renesas Electronics Corporation took over all the business of both companies. Therefore, although the old company name remains in this document, it is a valid Renesas Electronics document. We appreciate your understanding.

Renesas Electronics website: <http://www.renesas.com>

April 1st, 2010
Renesas Electronics Corporation

Issued by: Renesas Electronics Corporation (<http://www.renesas.com>)

Send any inquiries to <http://www.renesas.com/inquiry>.

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: "Standard", "High Quality", and "Specific". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as "Specific" without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as "Specific" or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is "Standard" unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.

"Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots.

"High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.

"Specific": Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.

8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

The revision list can be viewed directly by clicking the title page. The revision list summarizes the locations of revisions and additions. Details should always be checked by referring to the relevant text.

H8/3008

Hardware Manual

Renesas 16-Bit Single-Chip Microcomputer H8 Family/H8/300H Series

| | |
|---------|--------------|
| H8/3008 | HD6413008F |
| | HD6413008TE |
| | HD6413008VF |
| | HD6413008VTE |

Notes regarding these materials

1. This document is provided for reference purposes only so that Renesas customers may select the appropriate Renesas products for their use. Renesas neither makes warranties or representations with respect to the accuracy or completeness of the information contained in this document nor grants any license to any intellectual property rights or any other rights of Renesas or any third party with respect to the information in this document.
2. Renesas shall have no liability for damages or infringement of any intellectual property or other rights arising out of the use of any information in this document, including, but not limited to, product data, diagrams, charts, programs, algorithms, and application circuit examples.
3. You should not use the products or the technology described in this document for the purpose of military applications such as the development of weapons of mass destruction or for the purpose of any other military use. When exporting the products or technology described herein, you should follow the applicable export control laws and regulations, and procedures required by such laws and regulations.
4. All information included in this document such as product data, diagrams, charts, programs, algorithms, and application circuit examples, is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas products listed in this document, please confirm the latest product information with a Renesas sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas such as that disclosed through our website. (<http://www.renesas.com>)
5. Renesas has used reasonable care in compiling the information included in this document, but Renesas assumes no liability whatsoever for any damages incurred as a result of errors or omissions in the information included in this document.
6. When using or otherwise relying on the information in this document, you should evaluate the information in light of the total system before deciding about the applicability of such information to the intended application. Renesas makes no representations, warranties or guaranties regarding the suitability of its products for any particular application and specifically disclaims any liability arising out of the application and use of the information in this document or Renesas products.
7. With the exception of products specified by Renesas as suitable for automobile applications, Renesas products are not designed, manufactured or tested for applications or otherwise in systems the failure or malfunction of which may cause a direct threat to human life or create a risk of human injury or which require especially high quality and reliability such as safety systems, or equipment or systems for transportation and traffic, healthcare, combustion control, aerospace and aeronautics, nuclear power, or undersea communication transmission. If you are considering the use of our products for such purposes, please contact a Renesas sales office beforehand. Renesas shall have no liability for damages arising out of the uses set forth above.
8. Notwithstanding the preceding paragraph, you should not use Renesas products for the purposes listed below:
 - (1) artificial life support devices or systems
 - (2) surgical implantations
 - (3) healthcare intervention (e.g., excision, administration of medication, etc.)
 - (4) any other purposes that pose a direct threat to human lifeRenesas shall have no liability for damages arising out of the uses set forth in the above and purchasers who elect to use Renesas products in any of the foregoing applications shall indemnify and hold harmless Renesas Technology Corp., its affiliated companies and their officers, directors, and employees against any and all damages arising out of such applications.
9. You should use the products described herein within the range specified by Renesas, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas shall have no liability for malfunctions or damages arising out of the use of Renesas products beyond such specified ranges.
10. Although Renesas endeavors to improve the quality and reliability of its products, IC products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Please be sure to implement safety measures to guard against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other applicable measures. Among others, since the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
11. In case Renesas products listed in this document are detached from the products to which the Renesas products are attached or affixed, the risk of accident such as swallowing by infants and small children is very high. You should implement safety measures so that Renesas products may not be easily detached from your products. Renesas shall have no liability for damages arising out of such detachment.
12. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written approval from Renesas.
13. Please contact a Renesas sales office if you have any questions regarding the information contained in this document, Renesas semiconductor products, or if you have any other inquiries.

General Precautions in the Handling of MPU/MCU Products

The following usage notes are applicable to all MPU/MCU products from Renesas. For detailed usage notes on the products covered by this manual, refer to the relevant sections of the manual. If the descriptions under General Precautions in the Handling of MPU/MCU Products and in the body of the manual differ from each other, the description in the body of the manual takes precedence.

1. Handling of Unused Pins

Handle unused pins in accord with the directions given under Handling of Unused Pins in the manual.

- The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of LSI, an associated shoot-through current flows internally, and malfunctions may occur due to the false recognition of the pin state as an input signal. Unused pins should be handled as described under Handling of Unused Pins in the manual.

2. Processing at Power-on

The state of the product is undefined at the moment when power is supplied.

- The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the moment when power is supplied.

In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the moment when power is supplied until the reset process is completed.

In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the moment when power is supplied until the power reaches the level at which resetting has been specified.

3. Prohibition of Access to Reserved Addresses

Access to reserved addresses is prohibited.

- The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.

4. Clock Signals

After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has stabilized.

- When the clock signal is generated with an external resonator (or from an external oscillator) during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Moreover, when switching to a clock signal produced with an external resonator (or by an external oscillator) while program execution is in progress, wait until the target clock signal is stable.

5. Differences between Products

Before changing from one product to another, i.e. to one with a different type number, confirm that the change will not lead to problems.

- The characteristics of MPU/MCU in the same group but having different type numbers may differ because of the differences in internal memory capacity and layout pattern. When changing to products of different type numbers, implement a system-evaluation test for each of the products.

Preface

The H8/3008 is a high-performance single-chip microcomputer that incorporates the internal 32-bit H8/300H CPU and is also equipped with peripheral functions necessary for configuring a user system.

The H8/3008 is built in with a variety of peripheral functions such as ROM, RAM, 16-bit timer, 8-bit timer, programmable timing pattern controller (TPC), watchdog timer (WDT), serial communication interface (SCI), D/A converter, A/D converter and I/O ports.

Target Readers: This manual is designed for use by people who design application systems using the H8/3008.

To use this manual, basic knowledge of electric circuits, logic circuits and microcomputers is required.

Purpose: This manual provides the information of the hardware functions and electrical characteristics of the H8/3008.

The H8/300H Series Programming Manual contains detailed information of executable instructions. Please read the Programming Manual together with this manual.

How to Use the Book:

- To understand general functions
 - Read the manual from the beginning.
 - The manual explains the CPU, system control functions, peripheral functions and electrical characteristics in that order.
- To understanding CPU functions
 - Refer to the separate H8/300H Series Programming Manual.
- To see the detailed functions of registers with known names
 - Refer to Appendix B “Internal I/O Registers” for the summary of addresses, bit description and initialization.

Explanatory Note: Bit sequence: upper bit at left, and lower bit at right

List of Related Documents: The latest documents are available on our Web site. Please make sure that you have the latest version.

<http://www.renesas.com/>

User manual for H8/3008

| Document Title | Document No. |
|--------------------------------|---------------------|
| H8/3008 Hardware Manual | This manual |
| H8/300H Series Software Manual | REJ09B0213-0300 |

User manual for development tools

| Document Title | Document No. |
|--|---------------------|
| C/C++ Compiler, Assembler, Optimizing Linkage Editor User's Manual | REJ10B0058-0100H |
| H8S, H8/300 Series Simulator/Debugger User's Manual | REJ10B0211-0300 |
| High-performance Embedded Workshop User's Manual | REJ10J1554-0100 |
| H8S, H8/300 Series High-performance Embedded Workshop, High-performance Debugging Interface User's Manual | ADE-702-231 |

Application note

| Document Title | Document No. |
|---|---------------------|
| H8/300H for CPU Application Note | ADE-502-033 |
| H8/300H On-Chip Supporting Modules Application Note | REJ05B0522-0300 |
| H8/300H Technical Q&A | REJ05B0521-0200 |

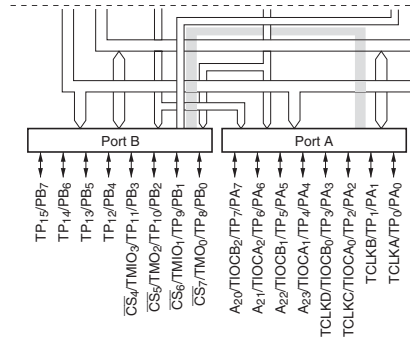
Main Revisions for This Edition

| Item | Page | Revision (See Manual for Details) |
|------|------|---|
| All | — | Company name and brand names amended (Before) Hitachi, Ltd. → (After) Renesas Technology Corp. |

| | | |
|--------------|---|---|
| 1.1 Overview | 1 | Description amended Four MCU operating modes offer a choice of bus width and address space size. The modes (modes 1 to 4) include four expanded modes. |
|--------------|---|---|

| | | |
|-------------------|---|----------------|
| 1.2 Block Diagram | 5 | Figure amended |
|-------------------|---|----------------|

Figure 1.1 Block Diagram



| | |
|-------------------------|---------------|
| 1.3.1 Pin Arrangement 6 | Table amended |
|-------------------------|---------------|

| | |
|--|--|
| Table 1.2 Comparison of H8/3008 Pin Arrangements | H8/3062 F-ZTAT B-Mask Version |
|--|--|

| | | |
|--|-----|---------------|
| 8.2.10 Timer I/O Control Register (TIOR) | 203 | Table amended |
|--|-----|---------------|

Bits 6 to 4—I/O Control B2 to B0 (IOB2 to IOB0)

| Bit 6 IOB2 | Bit 5 IOB1 | Bit 4 IOB0 | Function |
|---------------|---------------|---------------|----------------------------------|
| 1 | 0 | 0 | GRB is an input capture register |
| | | 1 | |
| | 1 | 0 | |
| | | 1 | |

Bits 2 to 0—I/O Control A2 to A0 (IOA2 to IOA0)

Table amended

| Bit 2 IOA2 | Bit 1 IOA1 | Bit 0 IOA0 | Function |
|---------------|---------------|---------------|----------------------------------|
| 1 | 0 | 0 | GRA is an input capture register |
| | | 1 | |
| | 1 | 0 | |
| | | 1 | |

9.4.1 8TCNT Count Timing

Figure 9.8 Count Timing for Internal Clock Input

Figure amended

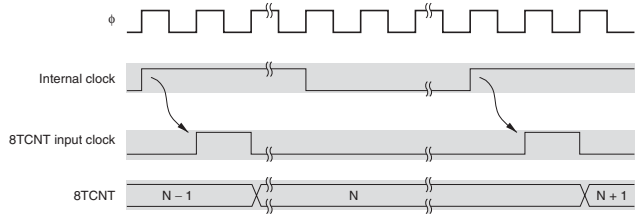
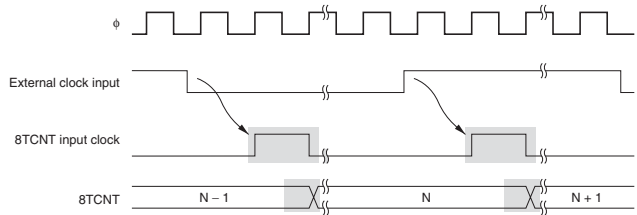


Figure 9.9 Count Timing for External Clock Input (Both-Edge Detection)

Figure amended



9.4.2 Compare Match Timing

Figure 9.11 Timing of Clear by Compare Match

Figure amended

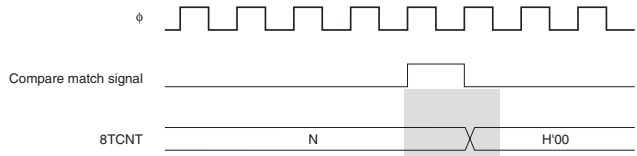
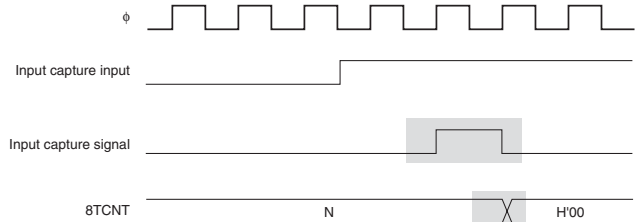


Figure 9.12 Timing of Clear by Input Capture

Figure amended

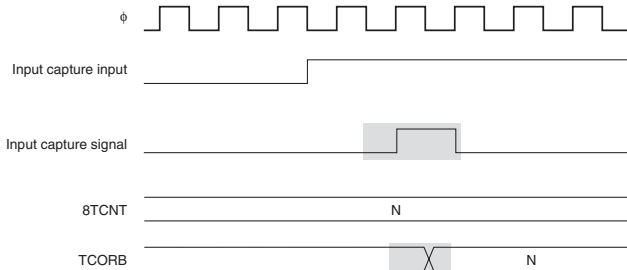


9.4.3 Input Capture Signal Timing

264

Figure amended

Figure 9.13 Timing of Input Capture Input Signal



9.4.4 Timing of Status Flag Setting

Figure 9.14 CMF Flag Setting Timing when Compare Match Occurs

Figure amended

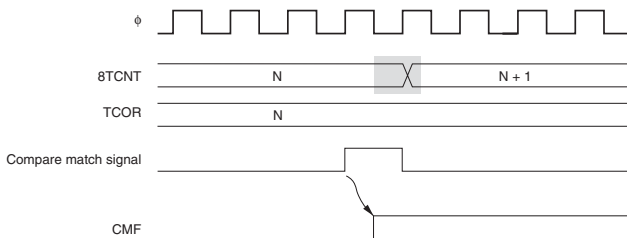


Figure 9.15 CMFB Flag Setting Timing when Input Capture Occurs

265

Figure amended

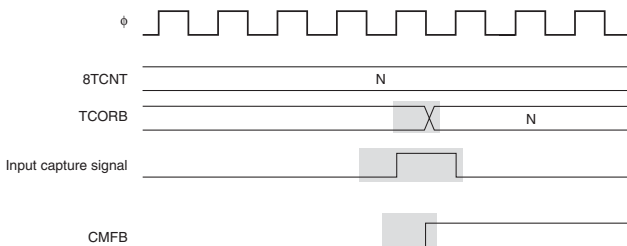
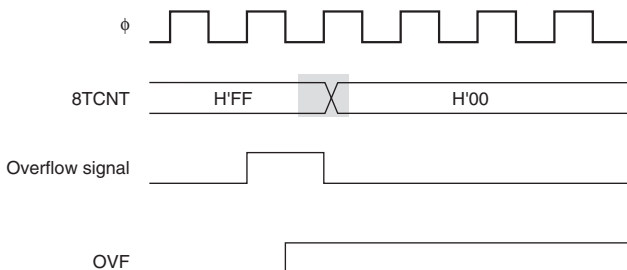


Figure 9.16 Timing of OVF Setting

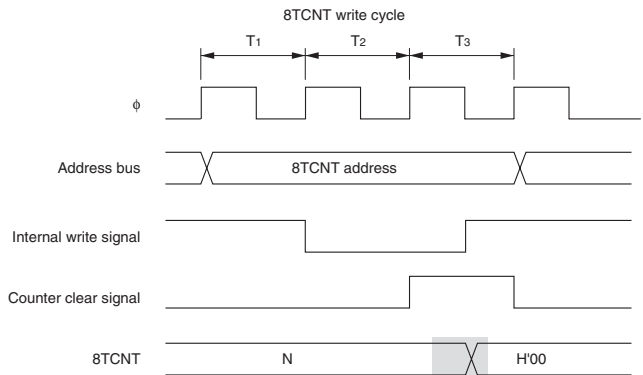
Figure amended



| Item | Page | Revision (See Manual for Details) |
|--|------|---|
| 9.4.5 Operation with Cascaded Connection Compare Match Count Mode | 267 | <p>Description amended</p> <ul style="list-style-type: none"> Channels 0 and 1: When bits CKS2 to CKS0 are set to (100) in 8TCR1, 8TCNT1 counts channel 0 compare match A events. Channels 0 and 1 are controlled independently. CMF flag setting, interrupt generation, TMO pin output, counter clearing, and so on, is in accordance with the settings for each channel. Note: When bit ICE = 1 in 8TCSR1, the compare match register function of TCORB0 in channel 0 cannot be used. Channels 2 and 3: When bits CKS2 to CKS0 are set to (100) in 8TCR3, 8TCNT3 counts channel 2 compare match A events. Channels 2 and 3 are controlled independently. CMF flag setting, interrupt generation, TMO pin output, counter clearing, and so on, is in accordance with the settings for each channel. Note: When bit ICE = 1 in 8TCSR3, the compare match register function of TCORB2 in channel 2 cannot be used. |

9.7.1 Contention between 8TCNT Write and Clear 272 Figure amended

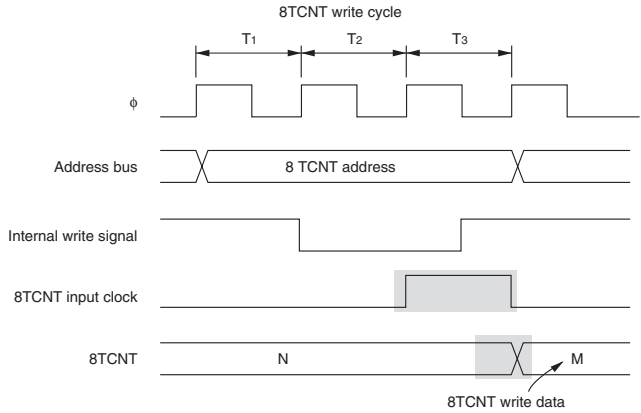
Figure 9.18 Contention between 8TCNT Write and Clear



9.7.2 Contention 273 Figure amended

between 8TCNT Write and Increment

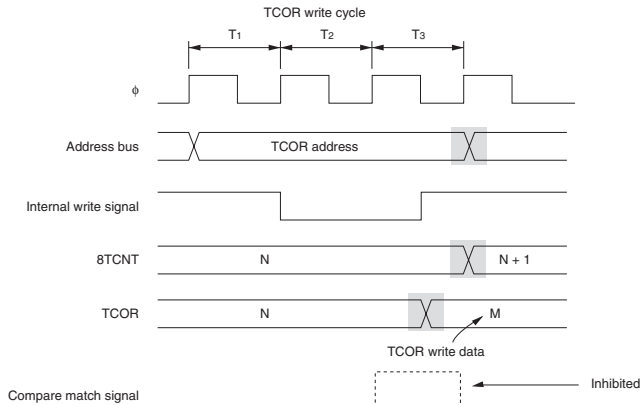
Figure 9.19 Contention between 8TCNT Write and Increment



9.7.3 Contention 274 Figure amended

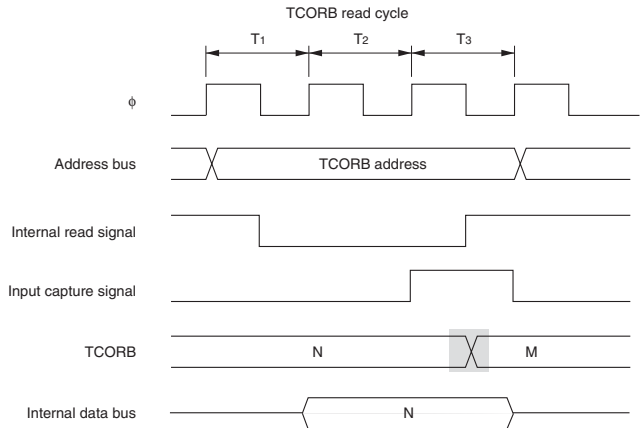
between TCOR Write and Compare Match

Figure 9.20 Contention between TCOR Write and Compare Match



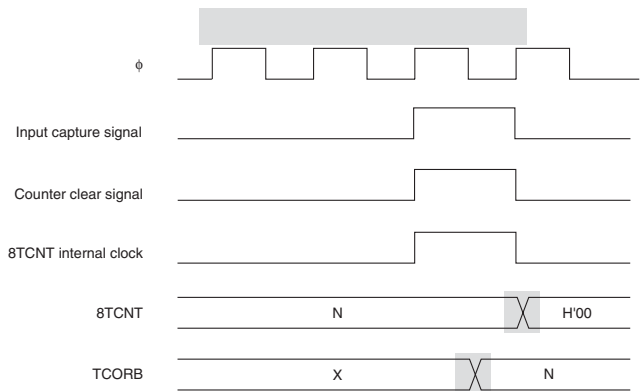
9.7.4 Contention between TCOR Read and Input Capture 275 Figure amended

Figure 9.21 Contention between TCOR Read and Input Capture



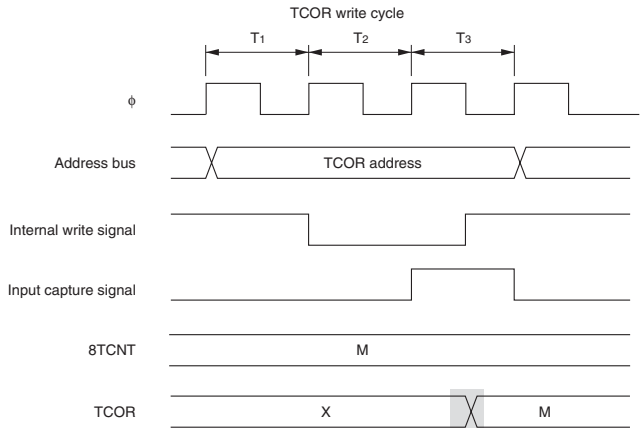
9.7.5 Contention between Counter Clearing by Input Capture and Counter Increment 276 Figure amended

Figure 9.22 Contention between Counter Clearing by Input Capture and Counter Increment



9.7.6 Contention 277 Figure amended
 between TCOR Write
 and Input Capture

Figure 9.23 Contention
 between TCOR Write
 and Input Capture



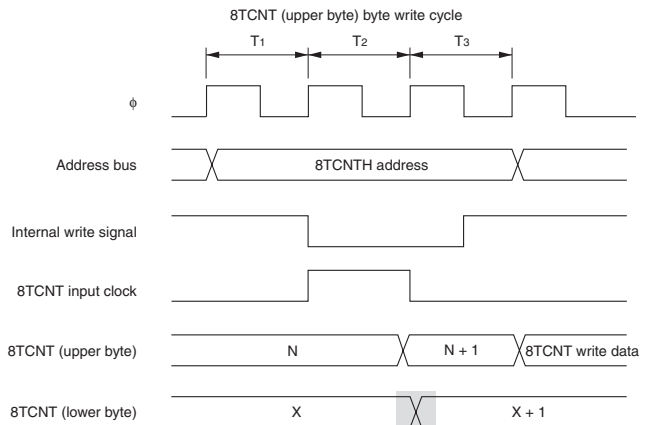
9.7.7 Contention 278
 between 8TCNT Byte
 Write and Increment in
 16-Bit Count Mode
 (Cascaded Connection)

Description amended

If an increment pulse occurs in the T₂ or T₃ state of an 8TCNT byte write cycle in 16-bit count mode, the counter write takes priority and the byte data for which the write was performed is not incremented. The byte data for which a write was not performed is incremented. Figure 9.24 shows the timing when an increment pulse occurs in the T₂ state of a byte write to 8TCNT (upper byte). If an increment pulse occurs in the T₂ state, on the other hand, the increment takes priority.

Figure 9.24 Contention
 between 8TCNT Byte
 Write and Increment in
 16-Bit Count Mode

Figure amended



| Item | Page | Revision (See Manual for Details) |
|--|------|---|
| 12.3.2 Operation in Asynchronous Mode Figure 12.4 Sample Flowchart for SCI Initialization | 356 | Figure amended and note added (4) Wait for at least the interval required to transmit or receive one bit, then set the TE or RE bit to 1 in SCR*. Set the RIE, TIE, TEIE, and MPIE bits as necessary. Setting the TE or RE bit enables the SCI to use the TxD or RxD pin. Note: * In simultaneous transmitting and receiving, the TE and RE bits should be cleared to 0 or set to 1 simultaneously. |

| 13.3.5 Clock | 396 | Table amended | | | | | | | | | | | | | | | | | | | |
|---|--------------|---|---------|--------------|--|--|-------|-------|-------|---|---------|---------|---------|---|---------|---------|---------|---|--------|--------|---------|
| Table 13.5 Bit Rates (bits/s) for Various BRR Settings (When n = 0) | | <table border="1"> <thead> <tr> <th rowspan="2">N</th> <th colspan="3">ϕ (MHz)</th> </tr> <tr> <th>18.00</th> <th>20.00</th> <th>25.00</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>24193.5</td> <td>26881.7</td> <td>33602.2</td> </tr> <tr> <td>1</td> <td>12096.8</td> <td>13440.9</td> <td>16801.1</td> </tr> <tr> <td>2</td> <td>8064.5</td> <td>8960.6</td> <td>11200.7</td> </tr> </tbody> </table> | N | ϕ (MHz) | | | 18.00 | 20.00 | 25.00 | 0 | 24193.5 | 26881.7 | 33602.2 | 1 | 12096.8 | 13440.9 | 16801.1 | 2 | 8064.5 | 8960.6 | 11200.7 |
| N | ϕ (MHz) | | | | | | | | | | | | | | | | | | | | |
| | 18.00 | 20.00 | 25.00 | | | | | | | | | | | | | | | | | | |
| 0 | 24193.5 | 26881.7 | 33602.2 | | | | | | | | | | | | | | | | | | |
| 1 | 12096.8 | 13440.9 | 16801.1 | | | | | | | | | | | | | | | | | | |
| 2 | 8064.5 | 8960.6 | 11200.7 | | | | | | | | | | | | | | | | | | |

| 18.4.3 Selection of Waiting Time for Exit from Software Standby Mode | 397 | Table amended | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--|--------------|---|-------|--------------|---|-------|--|--|--|-------|--|-------|--|-------|--|---|-------|---|-------|---|-------|------|---|-------|---|------|---|-------|
| Table 13.6 BRR Settings for Typical Bit Rates (bits/s) (When n = 0) | | <table border="1"> <thead> <tr> <th rowspan="3">bit/s</th> <th colspan="6">ϕ (MHz)</th> </tr> <tr> <th colspan="2">18.00</th> <th colspan="2">20.00</th> <th colspan="2">25.00</th> </tr> <tr> <th>N</th> <th>Error</th> <th>N</th> <th>Error</th> <th>N</th> <th>Error</th> </tr> </thead> <tbody> <tr> <td>9600</td> <td>2</td> <td>15.99</td> <td>2</td> <td>6.66</td> <td>3</td> <td>12.49</td> </tr> </tbody> </table> | bit/s | ϕ (MHz) | | | | | | 18.00 | | 20.00 | | 25.00 | | N | Error | N | Error | N | Error | 9600 | 2 | 15.99 | 2 | 6.66 | 3 | 12.49 |
| bit/s | ϕ (MHz) | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 18.00 | | | 20.00 | | 25.00 | | | | | | | | | | | | | | | | | | | | | | |
| | N | Error | N | Error | N | Error | | | | | | | | | | | | | | | | | | | | | | |
| 9600 | 2 | 15.99 | 2 | 6.66 | 3 | 12.49 | | | | | | | | | | | | | | | | | | | | | | |

| 18.4.3 Selection of Waiting Time for Exit from Software Standby Mode | 466 | Table amended | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--|------|---|------|------|---------------|-------|-------|--------------|-------|-----------------|-------|------|------|---|---|---|---|---|-------------|-----|-----|------|-------|----|--|--|---|---|---|--------------|-----|------|------|------|--|--|--|---|---|---|--------------|-------|------|------|------|--|--|--|---|---|---|--------------|------|------|------|-------|--|--|--|---|---|---|---------------|------|------|-------|-------|--|--|--|---|---|---|---------------|------|-------|-------|-------|--|--|--|---|---|---|-------------|------|------|-----|-----|--|--|--|---|---|---|--|--|--|--|--|-----------------|
| Table 18.3 Clock Frequency and Waiting Time for Clock to Settle | | <table border="1"> <thead> <tr> <th>DIV1</th> <th>DIV0</th> <th>STS2</th> <th>STS1</th> <th>STS0</th> <th>Waiting Time</th> <th>6 MHz</th> <th>4 MHz</th> <th>2 MHz</th> <th>1MHz</th> <th>Unit</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>1</td> <td>0</td> <td>0</td> <td>0</td> <td>8192 states</td> <td>2.7</td> <td>4.1</td> <td>8.2*</td> <td>16.4*</td> <td>ms</td> </tr> <tr> <td></td> <td></td> <td>0</td> <td>0</td> <td>1</td> <td>16384 states</td> <td>5.5</td> <td>8.2*</td> <td>16.4</td> <td>32.8</td> <td></td> </tr> <tr> <td></td> <td></td> <td>0</td> <td>1</td> <td>0</td> <td>32768 states</td> <td>10.9*</td> <td>16.4</td> <td>32.8</td> <td>65.5</td> <td></td> </tr> <tr> <td></td> <td></td> <td>0</td> <td>1</td> <td>1</td> <td>65536 states</td> <td>21.8</td> <td>32.8</td> <td>65.5</td> <td>131.1</td> <td></td> </tr> <tr> <td></td> <td></td> <td>1</td> <td>0</td> <td>0</td> <td>131072 states</td> <td>43.7</td> <td>65.5</td> <td>131.1</td> <td>262.1</td> <td></td> </tr> <tr> <td></td> <td></td> <td>1</td> <td>0</td> <td>1</td> <td>262144 states</td> <td>87.4</td> <td>131.1</td> <td>262.1</td> <td>524.3</td> <td></td> </tr> <tr> <td></td> <td></td> <td>1</td> <td>1</td> <td>0</td> <td>1024 states</td> <td>0.34</td> <td>0.51</td> <td>1.0</td> <td>2.0</td> <td></td> </tr> <tr> <td></td> <td></td> <td>1</td> <td>1</td> <td>1</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td>Illegal setting</td> </tr> </tbody> </table> | DIV1 | DIV0 | STS2 | STS1 | STS0 | Waiting Time | 6 MHz | 4 MHz | 2 MHz | 1MHz | Unit | 0 | 1 | 0 | 0 | 0 | 8192 states | 2.7 | 4.1 | 8.2* | 16.4* | ms | | | 0 | 0 | 1 | 16384 states | 5.5 | 8.2* | 16.4 | 32.8 | | | | 0 | 1 | 0 | 32768 states | 10.9* | 16.4 | 32.8 | 65.5 | | | | 0 | 1 | 1 | 65536 states | 21.8 | 32.8 | 65.5 | 131.1 | | | | 1 | 0 | 0 | 131072 states | 43.7 | 65.5 | 131.1 | 262.1 | | | | 1 | 0 | 1 | 262144 states | 87.4 | 131.1 | 262.1 | 524.3 | | | | 1 | 1 | 0 | 1024 states | 0.34 | 0.51 | 1.0 | 2.0 | | | | 1 | 1 | 1 | | | | | | Illegal setting |
| DIV1 | DIV0 | STS2 | STS1 | STS0 | Waiting Time | 6 MHz | 4 MHz | 2 MHz | 1MHz | Unit | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | 0 | 0 | 0 | 8192 states | 2.7 | 4.1 | 8.2* | 16.4* | ms | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | 0 | 0 | 1 | 16384 states | 5.5 | 8.2* | 16.4 | 32.8 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | 0 | 1 | 0 | 32768 states | 10.9* | 16.4 | 32.8 | 65.5 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | 0 | 1 | 1 | 65536 states | 21.8 | 32.8 | 65.5 | 131.1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | 1 | 0 | 0 | 131072 states | 43.7 | 65.5 | 131.1 | 262.1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | 1 | 0 | 1 | 262144 states | 87.4 | 131.1 | 262.1 | 524.3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | 1 | 1 | 0 | 1024 states | 0.34 | 0.51 | 1.0 | 2.0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | 1 | 1 | 1 | | | | | | Illegal setting | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| 19.2 DC Characteristics | 476 | Table amended | | | | | | | | | | | | | | | | | | | | | | | | | | |
|-----------------------------------|-------------------------------|---|------|--------|-----|-----|-----|-----------------------|----------------------------|---|-----|----|---|---|----|-----------------------------|-----------------------|---|-----|-----|-------------------------------|---|-----|-----|------|---|------|-----|
| Table 19.2 DC Characteristics (2) | | <table border="1"> <thead> <tr> <th>Item</th> <th>Symbol</th> <th>Min</th> <th>Typ</th> <th>Max</th> </tr> </thead> <tbody> <tr> <td rowspan="2">Current dissipation*2</td> <td rowspan="2">Standby mode I_{cc}^{*3}</td> <td>—</td> <td>1.0</td> <td>10</td> </tr> <tr> <td>—</td> <td>—</td> <td>80</td> </tr> <tr> <td rowspan="3">Analog power supply current</td> <td>During A/D conversion</td> <td>—</td> <td>0.6</td> <td>1.5</td> </tr> <tr> <td>During A/D and D/A conversion</td> <td>—</td> <td>0.6</td> <td>1.5</td> </tr> <tr> <td>Idle</td> <td>—</td> <td>0.01</td> <td>5.0</td> </tr> </tbody> </table> | Item | Symbol | Min | Typ | Max | Current dissipation*2 | Standby mode I_{cc}^{*3} | — | 1.0 | 10 | — | — | 80 | Analog power supply current | During A/D conversion | — | 0.6 | 1.5 | During A/D and D/A conversion | — | 0.6 | 1.5 | Idle | — | 0.01 | 5.0 |
| Item | Symbol | Min | Typ | Max | | | | | | | | | | | | | | | | | | | | | | | | |
| Current dissipation*2 | Standby mode I_{cc}^{*3} | — | 1.0 | 10 | | | | | | | | | | | | | | | | | | | | | | | | |
| | | — | — | 80 | | | | | | | | | | | | | | | | | | | | | | | | |
| Analog power supply current | During A/D conversion | — | 0.6 | 1.5 | | | | | | | | | | | | | | | | | | | | | | | | |
| | During A/D and D/A conversion | — | 0.6 | 1.5 | | | | | | | | | | | | | | | | | | | | | | | | |
| | Idle | — | 0.01 | 5.0 | | | | | | | | | | | | | | | | | | | | | | | | |

19.3 AC 481, 482 Table amended
 Characteristics

Table 19.6 Bus Timing

| Item | Symbol | Condition | | | | Unit | Test Conditions |
|-------------------------|------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|------|--------------------------|
| | | A | | B and C | | | |
| | | Min | Max | Min | Max | | |
| Read data setup time | t_{RDS} | 25 | — | 25 | — | ns | Figure 19.7, figure 19.8 |
| Read data hold time | t_{RDH} | 0 | — | 0 | — | ns | |
| Write data delay time | t_{WDD} | — | 35 | — | 35 | ns | |
| Write data setup time 1 | t_{WDS1} | $1.0 t_{\text{cyc}}$ -30 | — | $1.0 t_{\text{cyc}}$ -30 | — | ns | |
| Write data setup time 2 | t_{WDS2} | $2.0 t_{\text{cyc}}$ -30 | — | $2.0 t_{\text{cyc}}$ -30 | — | ns | |
| Write data hold time | t_{WDH} | $0.5 t_{\text{cyc}}$ -15 | — | $0.5 t_{\text{cyc}}$ -15 | — | ns | Figure 19.7, figure 19.8 |
| Read data access time 1 | t_{ACC1} | — | $2.0 t_{\text{cyc}}$ -45 | — | $2.0 t_{\text{cyc}}$ -45 | ns | |
| Read data access time 2 | t_{ACC2} | — | $3.0 t_{\text{cyc}}$ -45 | — | $3.0 t_{\text{cyc}}$ -45 | ns | |
| Read data access time 3 | t_{ACC3} | — | $1.5 t_{\text{cyc}}$ -45 | — | $1.5 t_{\text{cyc}}$ -45 | ns | |
| Read data access time 4 | t_{ACC4} | — | $2.5 t_{\text{cyc}}$ -45 | — | $2.5 t_{\text{cyc}}$ -45 | ns | |
| Precharge time 1 | t_{PCH1} | $1.0 t_{\text{cyc}}$ -20 | — | $1.0 t_{\text{cyc}}$ -20 | — | ns | |
| Precharge time 2 | t_{PCH2} | $0.5 t_{\text{cyc}}$ -20 | — | $0.5 t_{\text{cyc}}$ -20 | — | ns | |

19.4 A/D Conversion 486 Note added
 Characteristics

Table 19.8 A/D
 Conversion
 Characteristics

C.7 Port B Block 616 Figure amended
 Diagrams

Figure C.7 (a) Port B
 Block Diagram (Pins
 PB₀ and PB₂)

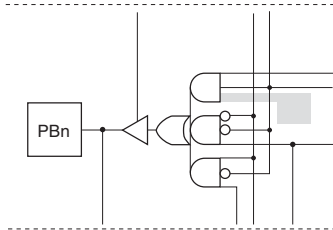
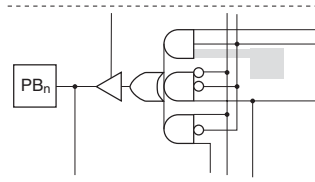


Figure C.7 (b) Port B 617 Figure amended
 Block Diagram (Pins
 PB₁ and PB₃)



All trademarks and registered trademarks are the property of their respective owners.

Contents

| | | |
|-----------|--|----|
| Section 1 | Overview | 1 |
| 1.1 | Overview | 1 |
| 1.2 | Block Diagram | 5 |
| 1.3 | Pin Description..... | 6 |
| 1.3.1 | Pin Arrangement..... | 6 |
| 1.3.2 | Pin Functions | 8 |
| 1.3.3 | Pin Assignments in Each Mode | 12 |
| Section 2 | CPU | 17 |
| 2.1 | Overview..... | 17 |
| 2.1.1 | Features..... | 17 |
| 2.1.2 | Differences from H8/300 CPU..... | 18 |
| 2.2 | CPU Operating Modes | 18 |
| 2.3 | Address Space | 19 |
| 2.4 | Register Configuration..... | 20 |
| 2.4.1 | Overview..... | 20 |
| 2.4.2 | General Registers | 21 |
| 2.4.3 | Control Registers | 22 |
| 2.4.4 | Initial CPU Register Values | 23 |
| 2.5 | Data Formats..... | 24 |
| 2.5.1 | General Register Data Formats | 24 |
| 2.5.2 | Memory Data Formats | 25 |
| 2.6 | Instruction Set | 27 |
| 2.6.1 | Instruction Set Overview | 27 |
| 2.6.2 | Instructions and Addressing Modes | 28 |
| 2.6.3 | Tables of Instructions Classified by Function..... | 29 |
| 2.6.4 | Basic Instruction Formats | 38 |
| 2.6.5 | Notes on Use of Bit Manipulation Instructions..... | 39 |
| 2.7 | Addressing Modes and Effective Address Calculation | 41 |
| 2.7.1 | Addressing Modes | 41 |
| 2.7.2 | Effective Address Calculation..... | 43 |
| 2.8 | Processing States..... | 47 |
| 2.8.1 | Overview..... | 47 |
| 2.8.2 | Program Execution State..... | 47 |
| 2.8.3 | Exception-Handling State | 48 |
| 2.8.4 | Exception Handling Operation..... | 49 |
| 2.8.5 | Bus-Released State..... | 50 |
| 2.8.6 | Reset State..... | 50 |

| | | |
|------------------|---|-----------|
| 2.8.7 | Power-Down State | 51 |
| 2.9 | Basic Operational Timing | 51 |
| 2.9.1 | Overview..... | 51 |
| 2.9.2 | On-Chip Memory Access Timing..... | 51 |
| 2.9.3 | On-Chip Supporting Module Access Timing | 52 |
| 2.9.4 | Access to External Address Space | 53 |
| Section 3 | MCU Operating Modes | 55 |
| 3.1 | Overview..... | 55 |
| 3.1.1 | Operating Mode Selection | 55 |
| 3.1.2 | Register Configuration..... | 56 |
| 3.2 | Mode Control Register (MDCR) | 56 |
| 3.3 | System Control Register (SYSCR) | 57 |
| 3.4 | Operating Mode Descriptions | 60 |
| 3.4.1 | Mode 1 | 60 |
| 3.4.2 | Mode 2..... | 60 |
| 3.4.3 | Mode 3..... | 60 |
| 3.4.4 | Mode 4..... | 60 |
| 3.4.5 | Modes 5 to 7 | 60 |
| 3.5 | Pin Functions in Each Operating Mode | 61 |
| 3.6 | Memory Map in Each Operating Mode | 62 |
| 3.6.1 | Reserved Areas | 62 |
| Section 4 | Exception Handling | 65 |
| 4.1 | Overview..... | 65 |
| 4.1.1 | Exception Handling Types and Priority..... | 65 |
| 4.1.2 | Exception Handling Operation..... | 65 |
| 4.1.3 | Exception Vector Table | 66 |
| 4.2 | Reset..... | 68 |
| 4.2.1 | Overview..... | 68 |
| 4.2.2 | Reset Sequence | 68 |
| 4.2.3 | Interrupts after Reset..... | 70 |
| 4.3 | Interrupts..... | 71 |
| 4.4 | Trap Instruction..... | 71 |
| 4.5 | Stack Status after Exception Handling..... | 72 |
| 4.6 | Notes on Stack Usage | 73 |
| Section 5 | Interrupt Controller | 75 |
| 5.1 | Overview..... | 75 |
| 5.1.1 | Features..... | 75 |

| | | |
|---------------------------------------|--|------------|
| 5.1.2 | Block Diagram | 76 |
| 5.1.3 | Pin Configuration | 77 |
| 5.1.4 | Register Configuration | 77 |
| 5.2 | Register Descriptions | 78 |
| 5.2.1 | System Control Register (SYSCR) | 78 |
| 5.2.2 | Interrupt Priority Registers A and B (IPRA, IPRB) | 79 |
| 5.2.3 | IRQ Status Register (ISR) | 84 |
| 5.2.4 | IRQ Enable Register (IER) | 85 |
| 5.2.5 | IRQ Sense Control Register (ISCR) | 86 |
| 5.3 | Interrupt Sources | 87 |
| 5.3.1 | External Interrupts | 87 |
| 5.3.2 | Internal Interrupts | 88 |
| 5.3.3 | Interrupt Exception Handling Vector Table | 88 |
| 5.4 | Interrupt Operation | 92 |
| 5.4.1 | Interrupt Handling Process | 92 |
| 5.4.2 | Interrupt Exception Handling Sequence | 97 |
| 5.4.3 | Interrupt Response Time | 98 |
| 5.5 | Usage Notes | 99 |
| 5.5.1 | Contention between Interrupt and Interrupt-Disabling Instruction | 99 |
| 5.5.2 | Instructions that Inhibit Interrupts | 100 |
| 5.5.3 | Interrupts during EEPMOV Instruction Execution | 100 |
| Section 6 Bus Controller | | 101 |
| 6.1 | Overview | 101 |
| 6.1.1 | Features | 101 |
| 6.1.2 | Block Diagram | 102 |
| 6.1.3 | Pin Configuration | 103 |
| 6.1.4 | Register Configuration | 104 |
| 6.2 | Register Descriptions | 104 |
| 6.2.1 | Bus Width Control Register (ABWCR) | 104 |
| 6.2.2 | Access State Control Register (ASTCR) | 105 |
| 6.2.3 | Wait Control Registers H and L (WCRH, WCRL) | 106 |
| 6.2.4 | Bus Release Control Register (BRCR) | 110 |
| 6.2.5 | Bus Control Register (BCR) | 111 |
| 6.2.6 | Chip Select Control Register (CSCR) | 114 |
| 6.2.7 | Address Control Register (ADRCR) | 115 |
| 6.3 | Operation | 116 |
| 6.3.1 | Area Division | 116 |
| 6.3.2 | Bus Specifications | 118 |
| 6.3.3 | Memory Interfaces | 119 |

| | | |
|---------------------------|---|-----|
| 6.3.4 | Chip Select Signals | 119 |
| 6.3.5 | Address Output Method..... | 120 |
| 6.4 | Basic Bus Interface | 122 |
| 6.4.1 | Overview..... | 122 |
| 6.4.2 | Data Size and Data Alignment..... | 122 |
| 6.4.3 | Valid Strobes..... | 123 |
| 6.4.4 | Memory Areas | 124 |
| 6.4.5 | Basic Bus Control Signal Timing | 125 |
| 6.4.6 | Wait Control | 132 |
| 6.5 | Idle Cycle..... | 134 |
| 6.5.1 | Operation | 134 |
| 6.5.2 | Pin States in Idle Cycle | 136 |
| 6.6 | Bus Arbiter..... | 137 |
| 6.6.1 | Operation | 137 |
| 6.7 | Register and Pin Input Timing..... | 139 |
| 6.7.1 | Register Write Timing | 139 |
| 6.7.2 | $\overline{\text{BREQ}}$ Pin Input Timing | 140 |
| Section 7 I/O Ports | | 141 |
| 7.1 | Overview..... | 141 |
| 7.2 | Port 4..... | 144 |
| 7.2.1 | Overview..... | 144 |
| 7.2.2 | Register Descriptions | 145 |
| 7.3 | Port 6..... | 148 |
| 7.3.1 | Overview..... | 148 |
| 7.3.2 | Register Descriptions | 148 |
| 7.4 | Port 7..... | 151 |
| 7.4.1 | Overview..... | 151 |
| 7.4.2 | Register Description..... | 151 |
| 7.5 | Port 8..... | 152 |
| 7.5.1 | Overview..... | 152 |
| 7.5.2 | Register Descriptions | 153 |
| 7.6 | Port 9..... | 156 |
| 7.6.1 | Overview..... | 156 |
| 7.6.2 | Register Descriptions | 157 |
| 7.7 | Port A..... | 161 |
| 7.7.1 | Overview..... | 161 |
| 7.7.2 | Register Descriptions | 162 |
| 7.8 | Port B..... | 172 |
| 7.8.1 | Overview..... | 172 |

| | | |
|--|--|-----|
| 7.8.2 | Register Descriptions | 173 |
| Section 8 16-Bit Timer..... 177 | | |
| 8.1 | Overview..... | 177 |
| 8.1.1 | Features..... | 177 |
| 8.1.2 | Block Diagrams | 179 |
| 8.1.3 | Pin Configuration..... | 182 |
| 8.1.4 | Register Configuration..... | 183 |
| 8.2 | Register Descriptions | 184 |
| 8.2.1 | Timer Start Register (TSTR)..... | 184 |
| 8.2.2 | Timer Synchro Register (TSNC) | 185 |
| 8.2.3 | Timer Mode Register (TMDR)..... | 187 |
| 8.2.4 | Timer Interrupt Status Register A (TISRA)..... | 189 |
| 8.2.5 | Timer Interrupt Status Register B (TISRB) | 192 |
| 8.2.6 | Timer Interrupt Status Register C (TISRC) | 195 |
| 8.2.7 | Timer Counters (16TCNT) | 197 |
| 8.2.8 | General Registers (GRA, GRB)..... | 198 |
| 8.2.9 | Timer Control Registers (16TCR) | 199 |
| 8.2.10 | Timer I/O Control Register (TIOR) | 202 |
| 8.2.11 | Timer Output Level Setting Register C (TOLR) | 204 |
| 8.3 | CPU Interface..... | 206 |
| 8.3.1 | 16-Bit Accessible Registers | 206 |
| 8.3.2 | 8-Bit Accessible Registers | 208 |
| 8.4 | Operation..... | 209 |
| 8.4.1 | Overview..... | 209 |
| 8.4.2 | Basic Functions..... | 209 |
| 8.4.3 | Synchronization | 217 |
| 8.4.4 | PWM Mode..... | 219 |
| 8.4.5 | Phase Counting Mode | 223 |
| 8.4.6 | 16-Bit Timer Output Timing..... | 225 |
| 8.5 | Interrupts..... | 226 |
| 8.5.1 | Setting of Status Flags..... | 226 |
| 8.5.2 | Timing of Clearing of Status Flags | 228 |
| 8.5.3 | Interrupt Sources..... | 229 |
| 8.6 | Usage Notes | 230 |
| Section 9 8-Bit Timers | | |
| 9.1 | Overview..... | 243 |
| 9.1.1 | Features..... | 243 |
| 9.1.2 | Block Diagram..... | 245 |

| | | |
|---|--|-----|
| 9.1.3 | Pin Configuration..... | 246 |
| 9.1.4 | Register Configuration..... | 247 |
| 9.2 | Register Descriptions | 248 |
| 9.2.1 | Timer Counters (8TCNT)..... | 248 |
| 9.2.2 | Time Constant Registers A (TCORA)..... | 249 |
| 9.2.3 | Time Constant Registers B (TCORB)..... | 250 |
| 9.2.4 | Timer Control Register (8TCR)..... | 251 |
| 9.2.5 | Timer Control/Status Registers (8TCSR)..... | 254 |
| 9.3 | CPU Interface..... | 259 |
| 9.3.1 | 8-Bit Registers | 259 |
| 9.4 | Operation | 261 |
| 9.4.1 | 8TCNT Count Timing..... | 261 |
| 9.4.2 | Compare Match Timing..... | 262 |
| 9.4.3 | Input Capture Signal Timing | 263 |
| 9.4.4 | Timing of Status Flag Setting | 264 |
| 9.4.5 | Operation with Cascaded Connection..... | 265 |
| 9.4.6 | Input Capture Setting..... | 268 |
| 9.5 | Interrupt | 269 |
| 9.5.1 | Interrupt Sources..... | 269 |
| 9.5.2 | A/D Converter Activation..... | 270 |
| 9.6 | 8-Bit Timer Application Example..... | 271 |
| 9.7 | Usage Notes | 272 |
| 9.7.1 | Contention between 8TCNT Write and Clear..... | 272 |
| 9.7.2 | Contention between 8TCNT Write and Increment | 273 |
| 9.7.3 | Contention between TCOR Write and Compare Match | 274 |
| 9.7.4 | Contention between TCOR Read and Input Capture | 275 |
| 9.7.5 | Contention between Counter Clearing by Input Capture and Counter Increment | 276 |
| 9.7.6 | Contention between TCOR Write and Input Capture..... | 277 |
| 9.7.7 | Contention between 8TCNT Byte Write and Increment in 16-Bit Count Mode (Cascaded Connection)..... | 278 |
| 9.7.8 | Contention between Compare Matches A and B | 279 |
| 9.7.9 | 8TCNT Operation and Internal Clock Source Switchover | 279 |
| Section 10 Programmable Timing Pattern Controller (TPC) | | 283 |
| 10.1 | Overview..... | 283 |
| 10.1.1 | Features..... | 283 |
| 10.1.2 | Block Diagram..... | 284 |
| 10.1.3 | Pin Configuration..... | 285 |
| 10.1.4 | Register Configuration..... | 286 |

| | | |
|------------------------------------|--|---------|
| 10.2 | Register Descriptions | 287 |
| 10.2.1 | Port A Data Direction Register (PADDR) | 287 |
| 10.2.2 | Port A Data Register (PADR) | 287 |
| 10.2.3 | Port B Data Direction Register (PBDDR)..... | 288 |
| 10.2.4 | Port B Data Register (PBDR) | 288 |
| 10.2.5 | Next Data Register A (NDRA) | 289 |
| 10.2.6 | Next Data Register B (NDRB)..... | 291 |
| 10.2.7 | Next Data Enable Register A (NDERA)..... | 293 |
| 10.2.8 | Next Data Enable Register B (NDERB) | 294 |
| 10.2.9 | TPC Output Control Register (TPCR) | 295 |
| 10.2.10 | TPC Output Mode Register (TPMR) | 298 |
| 10.3 | Operation..... | 300 |
| 10.3.1 | Overview..... | 300 |
| 10.3.2 | Output Timing..... | 301 |
| 10.3.3 | Normal TPC Output..... | 302 |
| 10.3.4 | Non-Overlapping TPC Output | 304 |
| 10.3.5 | TPC Output Triggering by Input Capture | 306 |
| 10.4 | Usage Notes | 307 |
| 10.4.1 | Operation of TPC Output Pins | 307 |
| 10.4.2 | Note on Non-Overlapping Output..... | 307 |
| Section 11 Watchdog Timer..... | | 309 |
| 11.1 | Overview..... | 309 |
| 11.1.1 | Features | 309 |
| 11.1.2 | Block Diagram..... | 310 |
| 11.1.3 | Pin Configuration..... | 310 |
| 11.1.4 | Register Configuration..... | 311 |
| 11.2 | Register Descriptions | 311 |
| 11.2.1 | Timer Counter (TCNT)..... | 311 |
| 11.2.2 | Timer Control/Status Register (TCSR)..... | 312 |
| 11.2.3 | Reset Control/Status Register (RSTCSR)..... | 314 |
| 11.2.4 | Notes on Register Access..... | 315 |
| 11.3 | Operation..... | 317 |
| 11.3.1 | Watchdog Timer Operation | 317 |
| 11.3.2 | Interval Timer Operation | 318 |
| 11.3.3 | Timing of Setting of Overflow Flag (OVF)..... | 318 |
| 11.3.4 | Timing of Setting of Watchdog Timer Reset Bit (WRST) | 319 |
| 11.4 | Interrupts | 320 |
| 11.5 | Usage Notes | 320 |

| | | |
|------------|--------------------------------------|-----|
| Section 12 | Serial Communication Interface | 321 |
| 12.1 | Overview..... | 321 |
| 12.1.1 | Features..... | 321 |
| 12.1.2 | Block Diagram..... | 323 |
| 12.1.3 | Pin Configuration..... | 324 |
| 12.1.4 | Register Configuration..... | 325 |
| 12.2 | Register Descriptions | 326 |
| 12.2.1 | Receive Shift Register (RSR) | 326 |
| 12.2.2 | Receive Data Register (RDR) | 326 |
| 12.2.3 | Transmit Shift Register (TSR) | 327 |
| 12.2.4 | Transmit Data Register (TDR)..... | 327 |
| 12.2.5 | Serial Mode Register (SMR) | 328 |
| 12.2.6 | Serial Control Register (SCR)..... | 332 |
| 12.2.7 | Serial Status Register (SSR) | 337 |
| 12.2.8 | Bit Rate Register (BRR) | 342 |
| 12.3 | Operation | 350 |
| 12.3.1 | Overview..... | 350 |
| 12.3.2 | Operation in Asynchronous Mode | 353 |
| 12.3.3 | Multiprocessor Communication..... | 362 |
| 12.3.4 | Synchronous Operation..... | 369 |
| 12.4 | SCI Interrupts..... | 377 |
| 12.5 | Usage Notes | 378 |
| 12.5.1 | Notes on Use of SCI | 378 |
| Section 13 | Smart Card Interface | 383 |
| 13.1 | Overview..... | 383 |
| 13.1.1 | Features..... | 383 |
| 13.1.2 | Block Diagram..... | 384 |
| 13.1.3 | Pin Configuration..... | 385 |
| 13.1.4 | Register Configuration..... | 385 |
| 13.2 | Register Descriptions | 386 |
| 13.2.1 | Smart Card Mode Register (SCMR)..... | 386 |
| 13.2.2 | Serial Status Register (SSR) | 388 |
| 13.2.3 | Serial Mode Register (SMR) | 389 |
| 13.2.4 | Serial Control Register (SCR)..... | 390 |
| 13.3 | Operation | 391 |
| 13.3.1 | Overview..... | 391 |
| 13.3.2 | Pin Connections | 391 |
| 13.3.3 | Data Format | 392 |
| 13.3.4 | Register Settings | 394 |

| | | |
|------------|--|-----|
| 13.3.5 | Clock..... | 396 |
| 13.3.6 | Transmitting and Receiving Data..... | 398 |
| 13.4 | Usage Notes | 406 |
| | | |
| Section 14 | A/D Converter | 411 |
| 14.1 | Overview..... | 411 |
| 14.1.1 | Features..... | 411 |
| 14.1.2 | Block Diagram..... | 412 |
| 14.1.3 | Pin Configuration..... | 413 |
| 14.1.4 | Register Configuration..... | 414 |
| 14.2 | Register Descriptions | 415 |
| 14.2.1 | A/D Data Registers A to D (ADDRA to ADDR D)..... | 415 |
| 14.2.2 | A/D Control/Status Register (ADCSR) | 416 |
| 14.2.3 | A/D Control Register (ADCR) | 418 |
| 14.3 | CPU Interface..... | 419 |
| 14.4 | Operation..... | 421 |
| 14.4.1 | Single Mode (SCAN = 0)..... | 421 |
| 14.4.2 | Scan Mode (SCAN = 1)..... | 423 |
| 14.4.3 | Input Sampling and A/D Conversion Time..... | 425 |
| 14.4.4 | External Trigger Input Timing | 427 |
| 14.5 | Interrupts | 427 |
| 14.6 | Usage Notes | 428 |
| | | |
| Section 15 | D/A Converter | 435 |
| 15.1 | Overview..... | 435 |
| 15.1.1 | Features..... | 435 |
| 15.1.2 | Block Diagram..... | 436 |
| 15.1.3 | Pin Configuration..... | 437 |
| 15.1.4 | Register Configuration..... | 437 |
| 15.2 | Register Descriptions | 438 |
| 15.2.1 | D/A Data Registers 0 and 1 (DADR0, DADR1)..... | 438 |
| 15.2.2 | D/A Control Register (DACR) | 438 |
| 15.2.3 | D/A Standby Control Register (DASTCR)..... | 440 |
| 15.3 | Operation..... | 440 |
| 15.4 | D/A Output Control | 442 |
| | | |
| Section 16 | RAM | 443 |
| 16.1 | Overview..... | 443 |
| 16.1.1 | Block Diagram..... | 444 |
| 16.1.2 | Register Configuration..... | 444 |

| | | |
|---|--|------------|
| 16.2 | System Control Register (SYSCR) | 445 |
| 16.3 | Operation | 446 |
| Section 17 Clock Pulse Generator | | 447 |
| 17.1 | Overview..... | 447 |
| 17.1.1 | Block Diagram..... | 448 |
| 17.2 | Oscillator Circuit..... | 449 |
| 17.2.1 | Connecting a Crystal Resonator..... | 449 |
| 17.2.2 | External Clock Input..... | 451 |
| 17.3 | Duty Adjustment Circuit..... | 454 |
| 17.4 | Prescalers | 454 |
| 17.5 | Frequency Divider | 454 |
| 17.5.1 | Register Configuration..... | 454 |
| 17.5.2 | Division Control Register (DIVCR) | 455 |
| 17.5.3 | Usage Notes | 456 |
| Section 18 Power-Down State | | 457 |
| 18.1 | Overview..... | 457 |
| 18.2 | Register Configuration..... | 459 |
| 18.2.1 | System Control Register (SYSCR)..... | 459 |
| 18.2.2 | Module Standby Control Register H (MSTCRH)..... | 461 |
| 18.2.3 | Module Standby Control Register L (MSTCRL)..... | 462 |
| 18.3 | Sleep Mode | 464 |
| 18.3.1 | Transition to Sleep Mode..... | 464 |
| 18.3.2 | Exit from Sleep Mode..... | 464 |
| 18.4 | Software Standby Mode..... | 464 |
| 18.4.1 | Transition to Software Standby Mode | 464 |
| 18.4.2 | Exit from Software Standby Mode | 465 |
| 18.4.3 | Selection of Waiting Time for Exit from Software Standby Mode..... | 465 |
| 18.4.4 | Sample Application of Software Standby Mode..... | 467 |
| 18.4.5 | Note..... | 467 |
| 18.5 | Hardware Standby Mode | 468 |
| 18.5.1 | Transition to Hardware Standby Mode | 468 |
| 18.5.2 | Exit from Hardware Standby Mode | 468 |
| 18.5.3 | Timing for Hardware Standby Mode | 468 |
| 18.6 | Module Standby Function..... | 469 |
| 18.6.1 | Module Standby Timing | 469 |
| 18.6.2 | Read/Write in Module Standby..... | 469 |
| 18.6.3 | Usage Notes | 469 |
| 18.7 | System Clock Output Disabling Function..... | 470 |

| | |
|---|-------------|
| Section 19 Electrical Characteristics..... | 471 |
| 19.1 Absolute Maximum Ratings | 471 |
| 19.2 DC Characteristics | 472 |
| 19.3 AC Characteristics | 479 |
| 19.4 A/D Conversion Characteristics..... | 485 |
| 19.5 D/A Conversion Characteristics..... | 487 |
| 19.6 Operational Timing | 488 |
| 19.6.1 Clock Timing | 488 |
| 19.6.2 Control Signal Timing | 489 |
| 19.6.3 Bus Timing | 491 |
| 19.6.4 TPC and I/O Port Timing..... | 495 |
| 19.6.5 Timer Input/Output Timing | 495 |
| 19.6.6 SCI Input/Output Timing..... | 496 |
| Appendix A Instruction Set | 497 |
| A.1 Instruction List..... | 497 |
| A.2 Operation Code Maps | 512 |
| A.3 Number of States Required for Execution | 515 |
| Appendix B Internal I/O Registers | 525 |
| B.1 Address List | 525 |
| B.2 Functions..... | 540 |
| Appendix C I/O Port Block Diagrams..... | 598 |
| C.1 Port 4 Block Diagram | 598 |
| C.2 Port 6 Block Diagrams | 599 |
| C.3 Port 7 Block Diagrams | 603 |
| C.4 Port 8 Block Diagrams | 604 |
| C.5 Port 9 Block Diagrams | 608 |
| C.6 Port A Block Diagrams | 614 |
| C.7 Port B Block Diagrams | 617 |
| Appendix D Pin States | 623 |
| D.1 Port States in Each Mode | 623 |
| D.2 Pin States at Reset | 626 |
| Appendix E Timing of Transition to and Recovery from Hardware Standby Mode | 628 |
| Appendix F Product Code Lineup | 629 |

| | | |
|------------|---|-----|
| Appendix G | Package Dimensions | 630 |
| Appendix H | Comparison of H8/300H Series Product Specifications..... | 632 |
| H.1 | Differences between H8/3067 and H8/3062 Group, H8/3048 Group, H8/3006 and H8/3007, and H8/3008 | 632 |
| H.2 | Comparison of Pin Functions of 100-Pin Package Products (FP-100B, TFP-100B) | 635 |

Figures

Section 1 Overview

| | | |
|------------|--|---|
| Figure 1.1 | Block Diagram | 5 |
| Figure 1.2 | Pin Arrangement of H8/3008 (FP-100B or TFP-100B Package, Top View) | 7 |

Section 2 CPU

| | | |
|-------------|---|----|
| Figure 2.1 | CPU Operating Modes | 18 |
| Figure 2.2 | Memory Map | 19 |
| Figure 2.3 | CPU Registers | 20 |
| Figure 2.4 | Usage of General Registers | 21 |
| Figure 2.5 | Stack | 22 |
| Figure 2.6 | General Register Data Formats (1) | 24 |
| Figure 2.7 | General Register Data Formats (2) | 25 |
| Figure 2.8 | Memory Data Formats | 26 |
| Figure 2.9 | Instruction Formats | 39 |
| Figure 2.10 | Memory-Indirect Branch Address Specification | 43 |
| Figure 2.11 | Processing States | 47 |
| Figure 2.12 | Classification of Exception Sources | 48 |
| Figure 2.13 | State Transitions | 49 |
| Figure 2.14 | Stack Structure after Exception Handling | 50 |
| Figure 2.15 | On-Chip Memory Access Cycle | 52 |
| Figure 2.16 | Pin States during On-Chip Memory Access (Address Update Mode 1) | 52 |
| Figure 2.17 | Access Cycle for On-Chip Supporting Modules | 53 |
| Figure 2.18 | Pin States during Access to On-Chip Supporting Modules | 53 |

Section 3 MCU Operating Modes

| | | |
|------------|--|----|
| Figure 3.1 | Memory Map of H8/3008 in Each Operating Mode | 63 |
|------------|--|----|

Section 4 Exception Handling

| | | |
|------------|--|----|
| Figure 4.1 | Exception Sources | 66 |
| Figure 4.2 | Reset Sequence (Modes 1 and 3) | 69 |
| Figure 4.3 | Reset Sequence (Modes 2 and 4) | 70 |
| Figure 4.4 | Interrupt Sources and Number of Interrupts | 71 |
| Figure 4.5 | Stack after Completion of Exception Handling | 72 |
| Figure 4.6 | Operation when SP Value is Odd | 74 |

Section 5 Interrupt Controller

| | | |
|------------|--|----|
| Figure 5.1 | Interrupt Controller Block Diagram | 76 |
| Figure 5.2 | Block Diagram of Interrupts IRQ_5 to IRQ_0 | 87 |

| | | |
|------------|--|----|
| Figure 5.3 | Timing of Setting of IRQnF | 88 |
| Figure 5.4 | Process Up to Interrupt Acceptance when UE = 1 | 93 |
| Figure 5.5 | Interrupt Masking State Transitions (Example) | 95 |
| Figure 5.6 | Process Up to Interrupt Acceptance when UE = 0 | 96 |
| Figure 5.7 | Interrupt Exception Handling Sequence..... | 97 |
| Figure 5.8 | Contention between Interrupt and Interrupt-Disabling Instruction | 99 |

Section 6 Bus Controller

| | | |
|-------------|--|-----|
| Figure 6.1 | Block Diagram of Bus Controller..... | 102 |
| Figure 6.2 | Access Area Map for Each Operating Mode..... | 116 |
| Figure 6.3 | Memory Map in 16-Mbyte Mode..... | 117 |
| Figure 6.4 | \overline{CS}_n Signal Output Timing (n = 0 to 7) | 120 |
| Figure 6.5 | Sample Address Output in Each Address Update Mode (Basic Bus Interface, 3-State Space) | 120 |
| Figure 6.6 | Example of Consecutive External Space Accesses in Address Update Mode 2 | 121 |
| Figure 6.7 | Access Sizes and Data Alignment Control (8-Bit Access Area) | 122 |
| Figure 6.8 | Access Sizes and Data Alignment Control (16-Bit Access Area) | 123 |
| Figure 6.9 | Bus Control Signal Timing for 8-Bit, Three-State-Access Area | 125 |
| Figure 6.10 | Bus Control Signal Timing for 8-Bit, Two-State-Access Area | 126 |
| Figure 6.11 | Bus Control Signal Timing for 16-Bit, Three-State-Access Area (1) (Byte Access to Even Address) | 127 |
| Figure 6.12 | Bus Control Signal Timing for 16-Bit, Three-State-Access Area (2) (Byte Access to Odd Address) | 128 |
| Figure 6.13 | Bus Control Signal Timing for 16-Bit, Three-State-Access Area (3) (Word Access)..... | 129 |
| Figure 6.14 | Bus Control Signal Timing for 16-Bit, Two-State-Access Area (1) (Byte Access to Even Address) | 130 |
| Figure 6.15 | Bus Control Signal Timing for 16-Bit, Two-State-Access Area (2) (Byte Access to Odd Address) | 131 |
| Figure 6.16 | Bus Control Signal Timing for 16-Bit, Two-State-Access Area (3) (Word Access)..... | 132 |
| Figure 6.17 | Example of Wait State Insertion Timing..... | 133 |
| Figure 6.18 | Example of Idle Cycle Operation (ICIS1 = 1)..... | 134 |
| Figure 6.19 | Example of Idle Cycle Operation (ICIS0 = 1)..... | 135 |
| Figure 6.20 | Example of Idle Cycle Operation | 136 |
| Figure 6.21 | Example of External Bus Master Operation..... | 138 |
| Figure 6.22 | ASTCR Write Timing | 139 |
| Figure 6.23 | DDR Write Timing..... | 139 |
| Figure 6.24 | BRCR Write Timing | 140 |

Section 7 I/O Ports

| | | |
|------------|-------------------------------|-----|
| Figure 7.1 | Port 4 Pin Configuration..... | 144 |
| Figure 7.2 | Port 6 Pin Configuration..... | 148 |
| Figure 7.3 | Port 7 Pin Configuration..... | 151 |
| Figure 7.4 | Port 8 Pin Configuration..... | 153 |
| Figure 7.5 | Port 9 Pin Configuration..... | 156 |
| Figure 7.6 | Port A Pin Configuration..... | 162 |
| Figure 7.7 | Port B Pin Configuration..... | 172 |

Section 8 16-Bit Timer

| | | |
|-------------|---|-----|
| Figure 8.1 | 16-bit timer Block Diagram (Overall) | 179 |
| Figure 8.2 | Block Diagram of Channels 0 and 1..... | 180 |
| Figure 8.3 | Block Diagram of Channel 2..... | 181 |
| Figure 8.4 | 16TCNT Access Operation [CPU → 16TCNT (Word)] | 206 |
| Figure 8.5 | Access to Timer Counter (CPU Reads 16TCNT, Word)..... | 206 |
| Figure 8.6 | Access to Timer Counter H (CPU Writes to 16TCNTH, Upper Byte)..... | 207 |
| Figure 8.7 | Access to Timer Counter L (CPU Writes to 16TCNTL, Lower Byte) | 207 |
| Figure 8.8 | Access to Timer Counter H (CPU Reads 16TCNTH, Upper Byte)..... | 207 |
| Figure 8.9 | Access to Timer Counter L (CPU Reads 16TCNTL, Lower Byte)..... | 208 |
| Figure 8.10 | 16TCR Access (CPU Writes to 16TCR) | 208 |
| Figure 8.11 | 16TCR Access (CPU Reads 16TCR) | 209 |
| Figure 8.12 | Counter Setup Procedure (Example) | 210 |
| Figure 8.13 | Free-Running Counter Operation | 211 |
| Figure 8.14 | Periodic Counter Operation | 212 |
| Figure 8.15 | Count Timing for Internal Clock Sources | 212 |
| Figure 8.16 | Count Timing for External Clock Sources (when Both Edges are Detected)..... | 213 |
| Figure 8.17 | Setup Procedure for Waveform Output by Compare Match (Example)..... | 213 |
| Figure 8.18 | 0 and 1 Output (TOA = 1, TOB = 0) | 214 |
| Figure 8.19 | Toggle Output (TOA = 1, TOB = 0) | 214 |
| Figure 8.20 | Output Compare Output Timing..... | 215 |
| Figure 8.21 | Setup Procedure for Input Capture (Example) | 216 |
| Figure 8.22 | Input Capture (Example) | 216 |
| Figure 8.23 | Input Capture Signal Timing | 217 |
| Figure 8.24 | Setup Procedure for Synchronization (Example) | 218 |
| Figure 8.25 | Synchronization (Example) | 219 |
| Figure 8.26 | Setup Procedure for PWM Mode (Example)..... | 220 |
| Figure 8.27 | PWM Mode (Example 1) | 221 |
| Figure 8.28 | PWM Mode (Example 2) | 222 |
| Figure 8.29 | Setup Procedure for Phase Counting Mode (Example) | 223 |
| Figure 8.30 | Operation in Phase Counting Mode (Example)..... | 224 |

| | | |
|-------------|--|-----|
| Figure 8.31 | Phase Difference, Overlap, and Pulse Width in Phase Counting Mode | 224 |
| Figure 8.32 | Timing for Setting 16-Bit Timer Output Level by Writing to TOLR..... | 225 |
| Figure 8.33 | Timing of Setting of IMFA and IMFB by Compare Match | 226 |
| Figure 8.34 | Timing of Setting of IMFA and IMFB by Input Capture..... | 227 |
| Figure 8.35 | Timing of Setting of OVF | 228 |
| Figure 8.36 | Timing of Clearing of Status Flags | 228 |
| Figure 8.37 | Contention between 16TCNT Write and Clear | 230 |
| Figure 8.38 | Contention between 16TCNT Word Write and Increment..... | 231 |
| Figure 8.39 | Contention between 16TCNT Byte Write and Increment | 232 |
| Figure 8.40 | Contention between General Register Write and Compare Match..... | 233 |
| Figure 8.41 | Contention between 16TCNT Write and Overflow..... | 234 |
| Figure 8.42 | Contention between General Register Read and Input Capture | 235 |
| Figure 8.43 | Contention between Counter Clearing by Input Capture and Counter Increment..... | 236 |
| Figure 8.44 | Contention between General Register Write and Input Capture | 237 |

Section 9 8-Bit Timers

| | | |
|-------------|--|-----|
| Figure 9.1 | Block Diagram of 8-Bit Timer Unit (Two Channels: Group 0) | 245 |
| Figure 9.2 | 8TCNT Access Operation (CPU Writes to 8TCNT, Word)..... | 259 |
| Figure 9.3 | 8TCNT Access Operation (CPU Reads 8TCNT, Word)..... | 259 |
| Figure 9.4 | 8TCNT0 Access Operation (CPU Writes to 8TCNT0, Upper Byte)..... | 259 |
| Figure 9.5 | 8TCNT1 Access Operation (CPU Writes to 8TCNT1, Lower Byte) | 260 |
| Figure 9.6 | 8TCNT0 Access Operation (CPU Reads 8TCNT0, Upper Byte)..... | 260 |
| Figure 9.7 | 8TCNT1 Access Operation (CPU Reads 8TCNT1, Lower Byte) | 260 |
| Figure 9.8 | Count Timing for Internal Clock Input..... | 261 |
| Figure 9.9 | Count Timing for External Clock Input (Both-Edge Detection) | 262 |
| Figure 9.10 | Timing of Timer Output | 262 |
| Figure 9.11 | Timing of Clear by Compare Match..... | 263 |
| Figure 9.12 | Timing of Clear by Input Capture | 263 |
| Figure 9.13 | Timing of Input Capture Input Signal | 264 |
| Figure 9.14 | CMF Flag Setting Timing when Compare Match Occurs..... | 264 |
| Figure 9.15 | CMFB Flag Setting Timing when Input Capture Occurs..... | 265 |
| Figure 9.16 | Timing of OVF Setting..... | 265 |
| Figure 9.17 | Example of Pulse Output..... | 271 |
| Figure 9.18 | Contention between 8TCNT Write and Clear | 272 |
| Figure 9.19 | Contention between 8TCNT Write and Increment..... | 273 |
| Figure 9.20 | Contention between TCOR Write and Compare Match..... | 274 |
| Figure 9.21 | Contention between TCOR Read and Input Capture..... | 275 |
| Figure 9.22 | Contention between Counter Clearing by Input Capture and Counter Increment..... | 276 |

| | | |
|-------------------|--|-----|
| Figure 9.23 | Contention between TCOR Write and Input Capture..... | 277 |
| Figure 9.24 | Contention between 8TCNT Byte Write and Increment in 16-Bit Count Mode..... | 278 |
| | | |
| Section 10 | Programmable Timing Pattern Controller (TPC) | |
| Figure 10.1 | TPC Block Diagram | 284 |
| Figure 10.2 | TPC Output Operation..... | 300 |
| Figure 10.3 | Timing of Transfer of Next Data Register Contents and Output (Example)..... | 301 |
| Figure 10.4 | Setup Procedure for Normal TPC Output (Example)..... | 302 |
| Figure 10.5 | Normal TPC Output Example (Five-Phase Pulse Output) | 303 |
| Figure 10.6 | Setup Procedure for Non-Overlapping TPC Output (Example)..... | 304 |
| Figure 10.7 | Non-Overlapping TPC Output Example (Four-Phase Complementary Non-Overlapping Pulse Output)..... | 305 |
| Figure 10.8 | TPC Output Triggering by Input Capture (Example)..... | 306 |
| Figure 10.9 | Non-Overlapping TPC Output..... | 307 |
| Figure 10.10 | Non-Overlapping Operation and NDR Write Timing | 308 |
| | | |
| Section 11 | Watchdog Timer | |
| Figure 11.1 | WDT Block Diagram | 310 |
| Figure 11.2 | Format of Data Written to TCNT and TCSR | 315 |
| Figure 11.3 | Format of Data Written to RSTCSR..... | 316 |
| Figure 11.4 | Operation in Watchdog Timer Mode..... | 317 |
| Figure 11.5 | Interval Timer Operation..... | 318 |
| Figure 11.6 | Timing of Setting of OVF | 318 |
| Figure 11.7 | Timing of Setting of WRST Bit and Internal Reset..... | 319 |
| Figure 11.8 | Contention between TCNT Write and Count up | 320 |
| | | |
| Section 12 | Serial Communication Interface | |
| Figure 12.1 | SCI Block Diagram | 323 |
| Figure 12.2 | Data Format in Asynchronous Communication (Example: 8-Bit Data with Parity and 2 Stop Bits) | 353 |
| Figure 12.3 | Phase Relationship between Output Clock and Serial Data (Asynchronous Mode)..... | 355 |
| Figure 12.4 | Sample Flowchart for SCI Initialization..... | 356 |
| Figure 12.5 | Sample Flowchart for Transmitting Serial Data..... | 357 |
| Figure 12.6 | Example of SCI Transmit Operation in Asynchronous Mode (8-Bit Data with Parity and One Stop Bit)..... | 358 |
| Figure 12.7 | Sample Flowchart for Receiving Serial Data | 359 |
| Figure 12.8 | Example of SCI Receive Operation (8-Bit Data with Parity and One Stop Bit)..... | 362 |

| | | |
|--------------|---|-----|
| Figure 12.9 | Example of Communication among Processors using Multiprocessor Format (Sending Data H'AA to Receiving Processor A)..... | 363 |
| Figure 12.10 | Sample Flowchart for Transmitting Multiprocessor Serial Data..... | 364 |
| Figure 12.11 | Example of SCI Transmit Operation (8-Bit Data with Multiprocessor Bit and One Stop Bit)..... | 365 |
| Figure 12.12 | Sample Flowchart for Receiving Multiprocessor Serial Data..... | 366 |
| Figure 12.13 | Example of SCI Receive Operation (8-Bit Data with Multiprocessor Bit and One Stop Bit)..... | 368 |
| Figure 12.14 | Data Format in Synchronous Communication..... | 369 |
| Figure 12.15 | Sample Flowchart for SCI Initialization..... | 370 |
| Figure 12.16 | Sample Flowchart for Serial Transmitting..... | 371 |
| Figure 12.17 | Example of SCI Transmit Operation..... | 372 |
| Figure 12.18 | Sample Flowchart for Serial Receiving..... | 373 |
| Figure 12.19 | Example of SCI Receive Operation..... | 375 |
| Figure 12.20 | Sample Flowchart for Simultaneous Serial Transmitting and Receiving..... | 376 |
| Figure 12.21 | Receive Data Sampling Timing in Asynchronous Mode..... | 379 |
| Figure 12.22 | Example of Synchronous Transmission..... | 380 |
| Figure 12.23 | Operation when Switching from SCK Pin Function to Port Pin Function..... | 381 |
| Figure 12.24 | Operation when Switching from SCK Pin Function to Port Pin Function (Example of Preventing Low-Level Output)..... | 382 |

Section 13 Smart Card Interface

| | | |
|--------------|---|-----|
| Figure 13.1 | Block Diagram of Smart Card Interface..... | 384 |
| Figure 13.2 | Smart Card Interface Connection Diagram..... | 392 |
| Figure 13.3 | Smart Card Interface Data Format..... | 393 |
| Figure 13.4 | Timing of TEND Flag Setting..... | 399 |
| Figure 13.5 | Sample Transmission Processing Flowchart..... | 400 |
| Figure 13.6 | Relation Between Transmit Operation and Internal Registers..... | 401 |
| Figure 13.7 | Timing of TEND Flag Setting..... | 401 |
| Figure 13.8 | Sample Reception Processing Flowchart..... | 402 |
| Figure 13.9 | Timing for Fixing Cock Output..... | 403 |
| Figure 13.10 | Procedure for Stopping and Restarting the Clock..... | 405 |
| Figure 13.11 | Receive Data Sampling Timing in Smart Card Interface Mode..... | 406 |
| Figure 13.12 | Retransmission in SCI Receive Mode..... | 408 |
| Figure 13.13 | Retransmission in SCI Transmit Mode..... | 408 |

Section 14 A/D Converter

| | | |
|-------------|---|-----|
| Figure 14.1 | A/D Converter Block Diagram..... | 412 |
| Figure 14.2 | A/D Data Register Access Operation (Reading H'AA40)..... | 420 |
| Figure 14.3 | Example of A/D Converter Operation (Single Mode, Channel 1 Selected)..... | 422 |

| | | |
|--|---|-----|
| Figure 14.4 | Example of A/D Converter Operation (Scan Mode, Channels AN ₀ to AN ₂ Selected) | 424 |
| Figure 14.5 | A/D Conversion Timing | 426 |
| Figure 14.6 | External Trigger Input Timing | 427 |
| Figure 14.7 | Example of Analog Input Protection Circuit | 429 |
| Figure 14.8 | Analog Input Pin Equivalent Circuit | 429 |
| Figure 14.9 | A/D Converter Accuracy Definitions (1) | 431 |
| Figure 14.10 | A/D Converter Accuracy Definitions (2) | 432 |
| Figure 14.11 | Analog Input Circuit (Example) | 433 |
| Section 15 D/A Converter | | |
| Figure 15.1 | D/A Converter Block Diagram..... | 436 |
| Figure 15.2 | Example of D/A Converter Operation | 441 |
| Section 16 RAM | | |
| Figure 16.1 | RAM Block Diagram | 444 |
| Section 17 Clock Pulse Generator | | |
| Figure 17.1 | Block Diagram of Clock Pulse Generator | 448 |
| Figure 17.2 | Connection of Crystal Resonator (Example) | 449 |
| Figure 17.3 | Crystal Resonator Equivalent Circuit | 450 |
| Figure 17.4 | Oscillator Circuit Block Board Design Precautions | 451 |
| Figure 17.5 | External Clock Input (Examples) | 451 |
| Figure 17.6 | External Clock Input Timing | 453 |
| Figure 17.7 | External Clock Output Settling Delay Timing | 453 |
| Section 18 Power-Down State | | |
| Figure 18.1 | NMI Timing for Software Standby Mode (Example) | 467 |
| Figure 18.2 | Hardware Standby Mode Timing | 469 |
| Figure 18.3 | Starting and Stopping of System Clock Output..... | 470 |
| Section 19 Electrical Characteristics | | |
| Figure 19.1 | Darlington Pair Drive Circuit (Example) | 478 |
| Figure 19.2 | Output Load Circuit..... | 484 |
| Figure 19.3 | Oscillator Settling Timing | 488 |
| Figure 19.4 | Reset Input Timing | 489 |
| Figure 19.5 | Reset Output Timing | 489 |
| Figure 19.6 | Interrupt Input Timing | 490 |
| Figure 19.7 | Basic Bus Cycle: Two-State Access..... | 492 |
| Figure 19.8 | Basic Bus Cycle: Three-State Access..... | 493 |

| | | |
|--------------|---|-----|
| Figure 19.9 | Basic Bus Cycle: Three-State Access with One Wait State | 494 |
| Figure 19.10 | Bus-Release Mode Timing | 494 |
| Figure 19.11 | TPC and I/O Port Input/Output Timing..... | 495 |
| Figure 19.12 | Timer Input/Output Timing..... | 495 |
| Figure 19.13 | Timer External Clock Input Timing | 496 |
| Figure 19.14 | SCI Input Clock Timing | 496 |
| Figure 19.15 | SCI Input/Output Timing in Synchronous Mode | 496 |

Appendix C I/O Port Block Diagrams

| | | |
|----------------|--|-----|
| Figure C.1 | Port 4 Block Diagram..... | 598 |
| Figure C.2 (a) | Port 6 Block Diagram (Pin P6 ₀)..... | 599 |
| Figure C.2 (b) | Port 6 Block Diagram (Pin P6 ₁)..... | 600 |
| Figure C.2 (c) | Port 6 Block Diagram (Pin P6 ₂)..... | 601 |
| Figure C.2 (d) | Port 6 Block Diagram (Pin P6 ₇)..... | 602 |
| Figure C.3 (a) | Port 7 Block Diagram (Pins P7 ₀ to P7 ₅)..... | 603 |
| Figure C.3 (b) | Port 7 Block Diagram (Pins P7 ₆ and P7 ₇) | 603 |
| Figure C.4 (a) | Port 8 Block Diagram (Pin P8 ₀)..... | 604 |
| Figure C.4 (b) | Port 8 Block Diagram (Pins P8 ₁ and P8 ₂) | 605 |
| Figure C.4 (c) | Port 8 Block Diagram (Pin P8 ₃)..... | 606 |
| Figure C.4 (d) | Port 8 Block Diagram (Pin P8 ₄)..... | 607 |
| Figure C.5 (a) | Port 9 Block Diagram (Pin P9 ₀)..... | 608 |
| Figure C.5 (b) | Port 9 Block Diagram (Pin P9 ₁)..... | 609 |
| Figure C.5 (c) | Port 9 Block Diagram (Pin P9 ₂)..... | 610 |
| Figure C.5 (d) | Port 9 Block Diagram (Pin P9 ₃)..... | 611 |
| Figure C.5 (e) | Port 9 Block Diagram (Pin P9 ₄)..... | 612 |
| Figure C.5 (f) | Port 9 Block Diagram (Pin P9 ₅)..... | 613 |
| Figure C.6 (a) | Port A Block Diagram (Pins PA ₀ and PA ₁)..... | 614 |
| Figure C.6 (b) | Port A Block Diagram (Pins PA ₂ and PA ₃)..... | 615 |
| Figure C.6 (c) | Port A Block Diagram (Pins PA ₄ to PA ₇) | 616 |
| Figure C.7 (a) | Port B Block Diagram (Pins PB ₀ and PB ₂) | 617 |
| Figure C.7 (b) | Port B Block Diagram (Pins PB ₁ and PB ₃) | 618 |
| Figure C.7 (c) | Port B Block Diagram (Pin PB ₄)..... | 619 |
| Figure C.7 (d) | Port B Block Diagram (Pin PB ₅)..... | 620 |
| Figure C.7 (e) | Port B Block Diagram (Pin PB ₆)..... | 621 |
| Figure C.7 (f) | Port B Block Diagram (Pin PB ₇)..... | 622 |

Appendix D Pin States

| | | |
|------------|---|-----|
| Figure D.1 | Reset during Memory Access (Modes 1 and 2)..... | 626 |
| Figure D.2 | Reset during Memory Access (Modes 3 and 4)..... | 627 |

Appendix G Package Dimensions

| | | |
|------------|------------------------------------|-----|
| Figure G.1 | Package Dimensions (FP-100B)..... | 630 |
| Figure G.2 | Package Dimensions (TFP-100B)..... | 631 |

Tables

Section 1 Overview

| | | |
|-----------|---|----|
| Table 1.1 | Features | 2 |
| Table 1.2 | Comparison of H8/3008 Pin Arrangements | 6 |
| Table 1.3 | Pin Functions..... | 8 |
| Table 1.4 | Pin Assignments in Each Mode (FP-100B, TFP-100B)..... | 12 |

Section 2 CPU

| | | |
|------------|---|----|
| Table 2.1 | Instruction Classification..... | 27 |
| Table 2.2 | Instructions and Addressing Modes | 28 |
| Table 2.3 | Data Transfer Instructions | 30 |
| Table 2.4 | Arithmetic Operation Instructions..... | 31 |
| Table 2.5 | Logic Operation Instructions | 33 |
| Table 2.6 | Shift Instructions | 33 |
| Table 2.7 | Bit Manipulation Instructions..... | 34 |
| Table 2.8 | Branching Instructions..... | 36 |
| Table 2.9 | System Control Instructions | 37 |
| Table 2.10 | Block Transfer Instruction..... | 38 |
| Table 2.11 | Addressing Modes..... | 41 |
| Table 2.12 | Absolute Address Access Ranges..... | 42 |
| Table 2.13 | Effective Address Calculation | 44 |
| Table 2.14 | Exception Handling Types and Priority | 48 |

Section 3 MCU Operating Modes

| | | |
|-----------|---------------------------------|----|
| Table 3.1 | Operating Mode Selection..... | 55 |
| Table 3.2 | Registers | 56 |
| Table 3.3 | Pin Functions in Each Mode..... | 61 |

Section 4 Exception Handling

| | | |
|-----------|------------------------------------|----|
| Table 4.1 | Exception Types and Priority | 65 |
| Table 4.2 | Exception Vector Table..... | 67 |

Section 5 Interrupt Controller

| | | |
|-----------|---|----|
| Table 5.1 | Interrupt Pins | 77 |
| Table 5.2 | Interrupt Controller Registers..... | 77 |
| Table 5.3 | Interrupt Sources, Vector Addresses, and Priority | 89 |
| Table 5.4 | UE, I, and UI Bit Settings and Interrupt Handling | 92 |
| Table 5.5 | Interrupt Response Time | 98 |

Section 6 Bus Controller

| | | |
|-----------|--|-----|
| Table 6.1 | Bus Controller Pins | 103 |
| Table 6.2 | Bus Controller Registers | 104 |
| Table 6.3 | Bus Specifications for Each Area (Basic Bus Interface) | 119 |
| Table 6.4 | Data Buses Used and Valid Strobes | 124 |
| Table 6.5 | Pin States in Idle Cycle | 136 |

Section 7 I/O Ports

| | | |
|------------|--|-----|
| Table 7.1 | Port Functions | 141 |
| Table 7.2 | Port 4 Registers | 145 |
| Table 7.3 | Input Pull-Up MOS Transistor States (Port 4) | 147 |
| Table 7.4 | Port 6 Registers | 148 |
| Table 7.5 | Port 6 Pin Functions in Modes 1 to 4 | 150 |
| Table 7.6 | Port 7 Data Register | 151 |
| Table 7.7 | Port 8 Registers | 153 |
| Table 7.8 | Port 8 Pin Functions in Modes 1 to 4 | 155 |
| Table 7.9 | Port 9 Registers | 157 |
| Table 7.10 | Port 9 Pin Functions | 159 |
| Table 7.11 | Port A Registers | 162 |
| Table 7.12 | Port A Pin Functions (Modes 1 and 2) | 165 |
| Table 7.13 | Port A Pin Functions (Modes 3 and 4) | 167 |
| Table 7.14 | Port A Pin Functions (Modes 1 to 4)..... | 169 |
| Table 7.15 | Port B Registers..... | 173 |
| Table 7.16 | Port B Pin Functions (Modes 1 to 4)..... | 175 |

Section 8 16-Bit Timer

| | | |
|---------------|--|-----|
| Table 8.1 | 16-bit timer Functions | 178 |
| Table 8.2 | 16-bit timer Pins | 182 |
| Table 8.3 | 16-bit timer Registers | 183 |
| Table 8.4 | PWM Output Pins and Registers | 220 |
| Table 8.5 | Up/Down Counting Conditions..... | 224 |
| Table 8.6 | 16-bit timer Interrupt Sources | 229 |
| Table 8.7 (a) | 16-bit timer Operating Modes (Channel 0) | 239 |
| Table 8.7 (b) | 16-bit timer Operating Modes (Channel 1) | 240 |
| Table 8.7 (c) | 16-bit timer Operating Modes (Channel 2) | 241 |

Section 9 8-Bit Timers

| | | |
|-----------|--|-----|
| Table 9.1 | 8-Bit Timer Pins | 246 |
| Table 9.2 | 8-Bit Timer Registers | 247 |
| Table 9.3 | Operation of Channels 0 and 1 when Bit ICE is Set to 1 in 8TCSR1 Register .. | 257 |

| | | |
|-----------|--|-----|
| Table 9.4 | Operation of Channels 2 and 3 when Bit ICE is Set to 1 in 8TCSR3 Register .. | 257 |
| Table 9.5 | Types of 8-Bit Timer Interrupt Sources and Priority Order | 269 |
| Table 9.6 | 8-Bit Timer Interrupt Sources..... | 270 |
| Table 9.7 | Timer Output Priority Order..... | 279 |
| Table 9.8 | Internal Clock Switchover and 8TCNT Operation | 280 |

Section 10 Programmable Timing Pattern Controller (TPC)

| | | |
|------------|--------------------------------|-----|
| Table 10.1 | TPC Pins..... | 285 |
| Table 10.2 | TPC Registers..... | 286 |
| Table 10.3 | TPC Operating Conditions | 300 |

Section 11 Watchdog Timer

| | | |
|------------|--|-----|
| Table 11.1 | WDT Pin | 310 |
| Table 11.2 | WDT Registers | 311 |
| Table 11.3 | Read Addresses of TCNT, TCSR, and RSTCSR | 316 |

Section 12 Serial Communication Interface

| | | |
|-------------|---|-----|
| Table 12.1 | SCI Pins..... | 324 |
| Table 12.2 | SCI Registers..... | 325 |
| Table 12.3 | Examples of Bit Rates and BRR Settings in Asynchronous Mode..... | 343 |
| Table 12.4 | Examples of Bit Rates and BRR Settings in Synchronous Mode..... | 346 |
| Table 12.5 | Maximum Bit Rates for Various Frequencies (Asynchronous Mode) | 348 |
| Table 12.6 | Maximum Bit Rates with External Clock Input (Asynchronous Mode) | 349 |
| Table 12.7 | Maximum Bit Rates with External Clock Input (Synchronous Mode)..... | 350 |
| Table 12.8 | SMR Settings and Serial Communication Formats | 352 |
| Table 12.9 | SMR and SCR Settings and SCI Clock Source Selection | 352 |
| Table 12.10 | Serial Communication Formats (Asynchronous Mode)..... | 354 |
| Table 12.11 | Receive Error Conditions | 361 |
| Table 12.12 | SCI Interrupt Sources | 377 |
| Table 12.13 | SSR Status Flags and Transfer of Receive Data..... | 378 |

Section 13 Smart Card Interface

| | | |
|------------|---|-----|
| Table 13.1 | Smart Card Interface Pins..... | 385 |
| Table 13.2 | Smart Card Interface Registers..... | 385 |
| Table 13.3 | Smart Card Interface Register Settings..... | 394 |
| Table 13.4 | n-Values of CKS1 and CKS0 Settings | 396 |
| Table 13.5 | Bit Rates (bits/s) for Various BRR Settings (When n = 0)..... | 396 |
| Table 13.6 | BRR Settings for Typical Bit Rates (bits/s) (When n = 0) | 397 |
| Table 13.7 | Maximum Bit Rates for Various Frequencies (Smart Card Interface Mode) | 397 |
| Table 13.8 | Smart Card Interface Mode Operating States and Interrupt Sources..... | 404 |

Section 14 A/D Converter

| | | |
|------------|--|-----|
| Table 14.1 | A/D Converter Pins | 413 |
| Table 14.2 | A/D Converter Registers | 414 |
| Table 14.3 | Analog Input Channels and A/D Data Registers (ADDRA to ADDR D) | 415 |
| Table 14.4 | A/D Conversion Time (Single Mode) | 426 |
| Table 14.5 | Analog Input Pin Ratings | 429 |

Section 15 D/A Converter

| | | |
|------------|-------------------------------|-----|
| Table 15.1 | D/A Converter Pins | 437 |
| Table 15.2 | D/A Converter Registers | 437 |

Section 16 RAM

| | | |
|------------|--|-----|
| Table 16.1 | H8/3008 On-Chip RAM Specifications | 443 |
| Table 16.2 | System Control Register | 444 |

Section 17 Clock Pulse Generator

| | | |
|----------------|------------------------------------|-----|
| Table 17.1 (1) | Damping Resistance Value | 449 |
| Table 17.1 (2) | External Capacitance Values | 450 |
| Table 17.2 | Crystal Resonator Parameters | 450 |
| Table 17.3 | Clock Timing (Preliminary) | 452 |
| Table 17.4 | Frequency Division Register | 454 |

Section 18 Power-Down State

| | | |
|------------|--|-----|
| Table 18.1 | Power-Down State and Module Standby Function | 458 |
| Table 18.2 | Control Register | 459 |
| Table 18.3 | Clock Frequency and Waiting Time for Clock to Settle | 466 |
| Table 18.4 | ϕ Pin State in Various Operating States | 470 |

Section 19 Electrical Characteristics

| | | |
|------------|--|-----|
| Table 19.1 | Absolute Maximum Ratings | 471 |
| Table 19.2 | DC Characteristics (1) | 472 |
| Table 19.2 | DC Characteristics (2) | 475 |
| Table 19.3 | Permissible Output Currents | 477 |
| Table 19.4 | Clock Timing | 479 |
| Table 19.5 | Control Signal Timing | 480 |
| Table 19.6 | Bus Timing | 481 |
| Table 19.7 | Timing of On-Chip Supporting Modules | 483 |
| Table 19.8 | A/D Conversion Characteristics | 485 |
| Table 19.9 | D/A Conversion Characteristics | 487 |

Appendix A Instruction Set

| | | |
|-----------|--|-----|
| Table A.1 | Instruction Set..... | 499 |
| Table A.2 | Operation Code Map (1)..... | 512 |
| Table A.2 | Operation Code Map (2)..... | 513 |
| Table A.2 | Operation Code Map (3)..... | 514 |
| Table A.3 | Number of States per Cycle..... | 516 |
| Table A.4 | Number of Cycles per Instruction | 517 |

Appendix D Pin States

| | | |
|-----------|------------------|-----|
| Table D.1 | Port States..... | 623 |
|-----------|------------------|-----|

Appendix F Product Code Lineup

| | | |
|-----------|----------------------------------|-----|
| Table F.1 | H8/3008 Product Code Lineup..... | 629 |
|-----------|----------------------------------|-----|

Appendix H Comparison of H8/300H Series Product Specifications

| | | |
|-----------|---|-----|
| Table H.1 | Pin Arrangement of Each Product (FP-100B, TFP-100B) | 635 |
|-----------|---|-----|

Section 1 Overview

1.1 Overview

The H8/3008 is a microcontroller (MCU) that integrates system supporting functions together with an H8/300H CPU core having an original Renesas Technology architecture.

The H8/300H CPU has a 32-bit internal architecture with sixteen 16-bit general registers, and a concise, optimized instruction set designed for speed. It can address a 16-Mbyte linear address space. Its instruction set is upward-compatible at the object-code level with the H8/300 CPU, enabling easy porting of software from the H8/300 Series.

The on-chip system supporting functions include RAM, a 16-bit timer, an 8-bit timer, a programmable timing pattern controller (TPC), a watchdog timer (WDT), a serial communication interface (SCI), an A/D converter, a D/A converter, I/O ports, and other facilities.

Four MCU operating modes offer a choice of bus width and address space size. The modes (modes 1 to 4) include four expanded modes.

Table 1.1 summarizes the features of the H8/3008.

Table 1.1 Features

| Feature | Description |
|----------------------|---|
| CPU | <p>Upward-compatible with the H8/300 CPU at the object-code level General-register machine</p> <ul style="list-style-type: none"> Sixteen 16-bit general registers (also usable as sixteen 8-bit registers plus eight 16-bit registers, or as eight 32-bit registers) <p>High-speed operation</p> <ul style="list-style-type: none"> Maximum clock rate: 25 MHz Add/subtract: 80 ns Multiply/divide: 560 ns <p>16-Mbyte address space</p> <p>Instruction features</p> <ul style="list-style-type: none"> 8/16/32-bit data transfer, arithmetic, and logic instructions Signed and unsigned multiply instructions (8 bits × 8 bits, 16 bits × 16 bits) Signed and unsigned divide instructions (16 bits ÷ 8 bits, 32 bits ÷ 16 bits) Bit accumulator function <p>Bit manipulation instructions with register-indirect specification of bit positions</p> |
| Memory | <p>H8/3008</p> <ul style="list-style-type: none"> RAM: 4 kbytes |
| Interrupt controller | <ul style="list-style-type: none"> Seven external interrupt pins: NMI, \overline{IRQ}_0 to \overline{IRQ}_5 27 internal interrupts Three selectable interrupt priority levels |
| Bus controller | <ul style="list-style-type: none"> Address space can be partitioned into eight areas, with independent bus specifications in each area Chip select output available for areas 0 to 7 8-bit access or 16-bit access selectable for each area Two-state or three-state access selectable for each area Selection of two wait modes Number of program wait states selectable for each area Bus arbitration function Two address update modes |

| Feature | Description |
|---|---|
| 16-bit timer, 3 channels | <ul style="list-style-type: none"> • Three 16-bit timer channels, capable of processing up to six pulse outputs or six pulse inputs • 16-bit timer counter (channels 0 to 2) • Two multiplexed output compare/input capture pins (channels 0 to 2) • Operation can be synchronized (channels 0 to 2) • PWM mode available (channels 0 to 2) • Phase counting mode available (channel 2) |
| 8-bit timer, 4 channels | <ul style="list-style-type: none"> • 8-bit up-counter (external event count capability) • Two time constant registers • Two channels can be connected |
| Programmable timing pattern controller (TPC) | <ul style="list-style-type: none"> • Maximum 16-bit pulse output, using 16-bit timer as time base • Up to four 4-bit pulse output groups (or one 16-bit group, or two 8-bit groups) • Non-overlap mode available |
| Watchdog timer (WDT), 1 channel | <ul style="list-style-type: none"> • Internal reset signal can be generated by overflow • Reset signal can be output externally • Usable as an interval timer |
| Serial communication interface (SCI), 2 channels | <ul style="list-style-type: none"> • Selection of asynchronous or synchronous mode • Full duplex: can transmit and receive simultaneously • On-chip baud-rate generator • Smart card interface functions added |
| A/D converter | <ul style="list-style-type: none"> • Resolution: 10 bits • Eight channels, with selection of single or scan mode • Variable analog conversion voltage range • Sample-and-hold function • A/D conversion can be started by an external trigger or 8-bit timer compare-match |
| D/A converter | <ul style="list-style-type: none"> • Resolution: 8 bits • Two channels • D/A outputs can be sustained in software standby mode |
| I/O ports | <ul style="list-style-type: none"> • 35 input/output pins • 12 input-only pins |

| Feature | Description | | | | |
|------------------|---|---------------|-----------------------------------|-------------------------|----------------|
| Operating modes | Four MCU operating modes | | | | |
| | Mode | Address Space | Address Pins | Initial Bus Width | Max. Bus Width |
| | Mode 1 | 1 Mbyte | A ₁₉ to A ₀ | 8 bits | 16 bits |
| | Mode 2 | 1 Mbyte | A ₁₉ to A ₀ | 16 bits | 16 bits |
| | Mode 3 | 16 Mbytes | A ₂₃ to A ₀ | 8 bits | 16 bits |
| | Mode 4 | 16 Mbytes | A ₂₃ to A ₀ | 16 bits | 16 bits |
| | <ul style="list-style-type: none"> On-chip ROM is disabled in modes 1 to 4 | | | | |
| Power-down state | <ul style="list-style-type: none"> Sleep mode Software standby mode Hardware standby mode Module standby function Programmable system clock frequency division | | | | |
| Other features | <ul style="list-style-type: none"> On-chip clock pulse generator | | | | |
| Product lineup | Product Type | | Model | Package (Package Code) | |
| | H8/3008 | 5 V operation | HD6413008F | 100-pin QFP (FP-100B) | |
| | | | HD6413008TE | 100-pin TQFP (TFP-100B) | |
| | | 3 V operation | HD6413008VF | 100-pin QFP (FP-100B) | |
| | | | HD6413008VTE | 100-pin TQFP (TFP-100B) | |

1.2 Block Diagram

Figure 1.1 shows an internal block diagram.

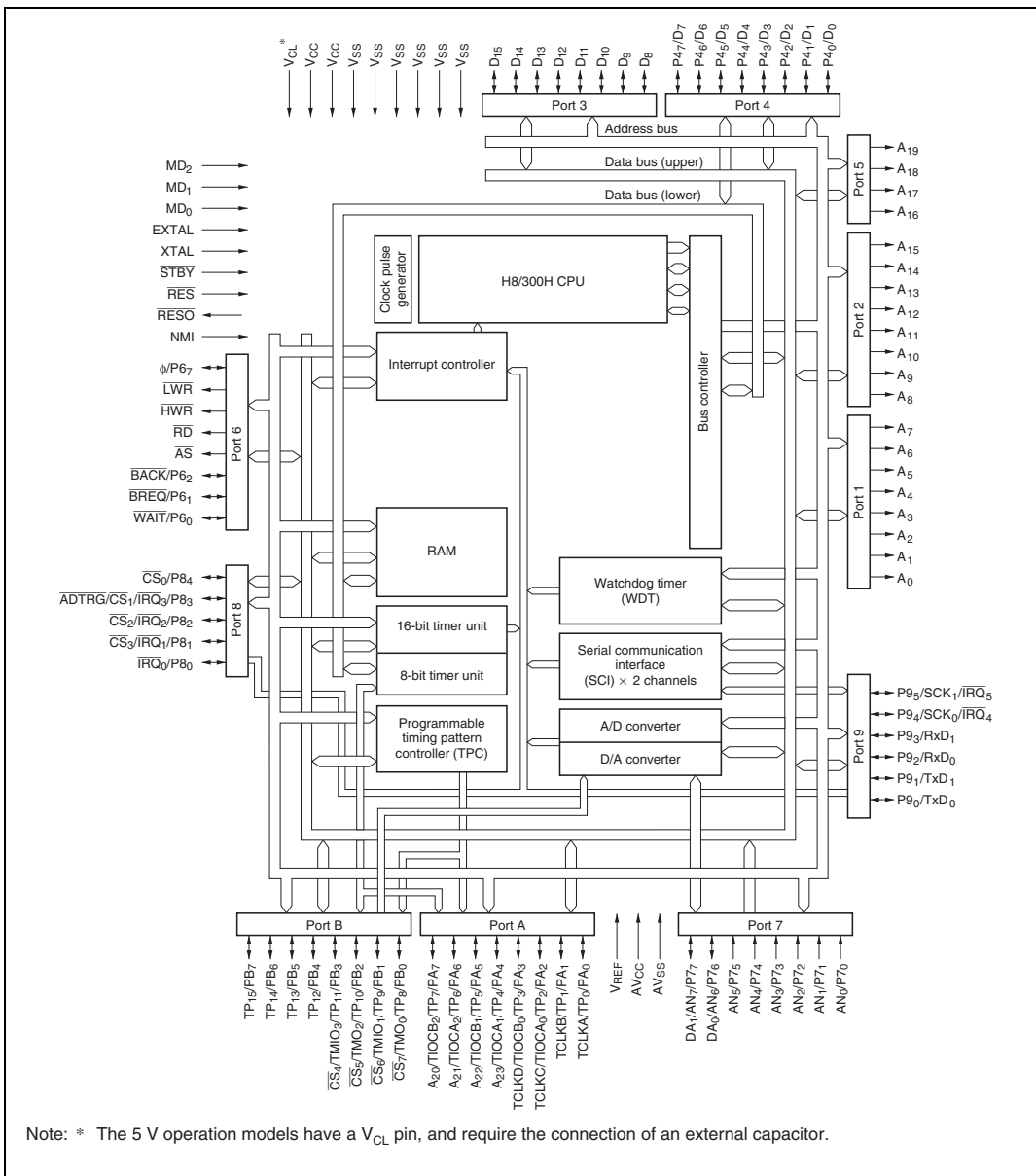


Figure 1.1 Block Diagram

1.3 Pin Description

1.3.1 Pin Arrangement

The pin arrangement of the H8/3008 is shown in figures 1.2 and 1.3. Differences in the H8/3008 pin arrangements are shown in table 1.2. Except for the differences shown in table 1.2, the pin arrangements are the same.

Table 1.2 Comparison of H8/3008 Pin Arrangements

| Package | Pin Number | H8/3064 F-ZTAT B-Mask Version | | H8/3062 F-ZTAT B-Mask Version | | H8/3024 F-ZTAT H8/3008 ROMless | |
|------------|------------|--|----------|--|----------|--------------------------------------|----------|
| | | Operation Model | | | | | |
| | | 5 V | 3 V | 5 V | 3 V | 5 V | 3 V |
| FP-100B | 1 | V_{CL} | V_{CC} | V_{CL} | V_{CC} | V_{CL} | V_{CC} |
| (TFP-100B) | 10 | FWE | FWE | FWE | FWE | RESO | RESO |

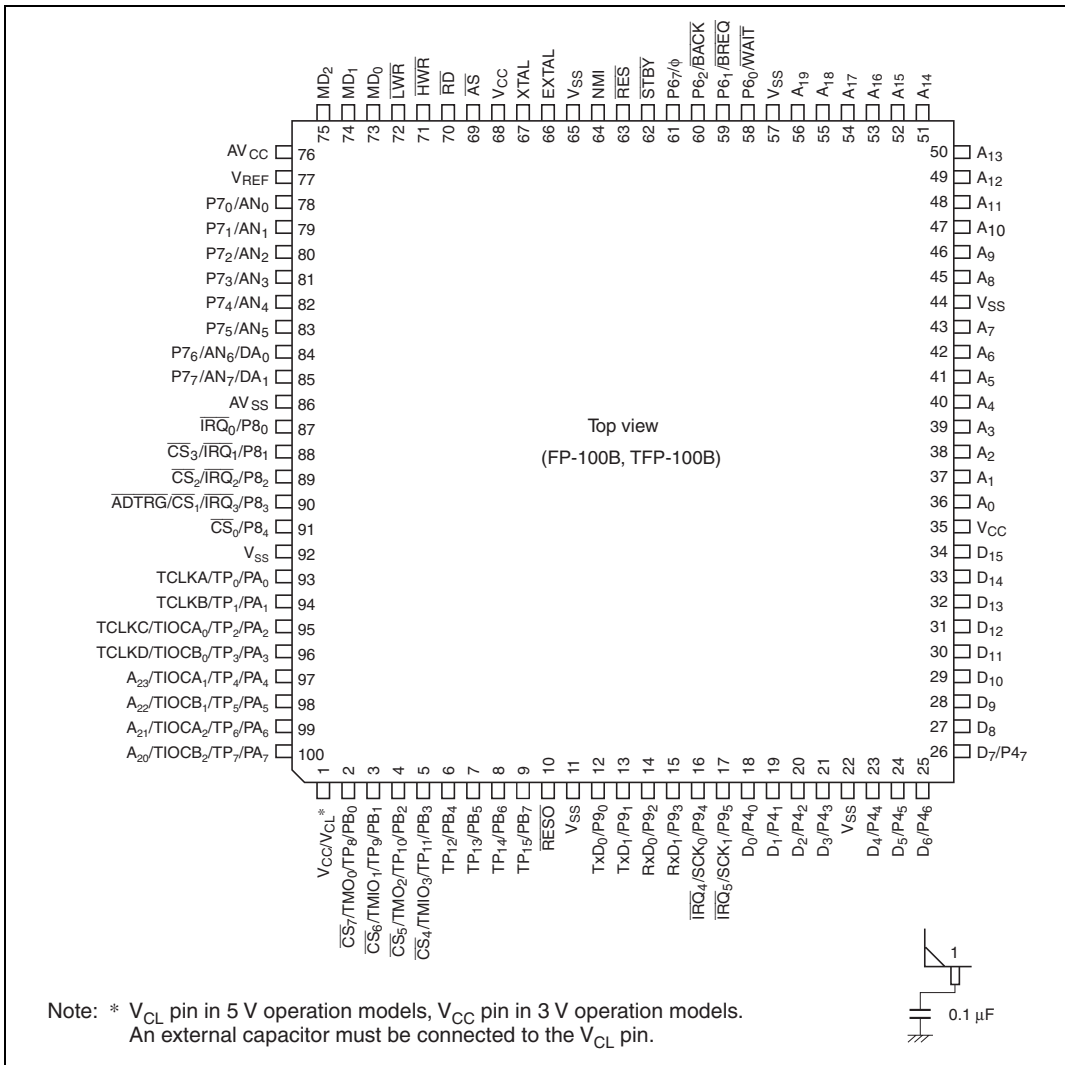
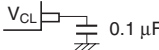


Figure 1.2 Pin Arrangement of H8/3008 (FP-100B or TFP-100B Package, Top View)

1.3.2 Pin Functions

Table 1.3 summarizes the pin functions. The 5 V operation models have a V_{CL} pin, and require the connection of an external capacitor.

Table 1.3 Pin Functions

| Type | Symbol | Pin No. | | Name and Function | | | | |
|------------------------|------------------|---------------------------|----------|---|--------|--------|--------|--------------------|
| | | FP-100B | TFP-100B | | | | | |
| Power | V_{CC} | 1* | 35, 68 | Power: For connection to the power supply. Connect all V_{CC} pins to the system power supply. | | | | |
| | V_{SS} | 11, 22, 44, 57, 65, 92 | | Ground: For connection to ground (0 V). Connect all V_{SS} pins to the 0-V system power supply. | | | | |
| Internal step-down pin | V_{CL} | 1* | | Output Connect an external capacitor between this pin and GND (0 V). Do not connect to V_{CC} .  | | | | |
| Clock | XTAL | 67 | | Input For connection to a crystal resonator. For examples of crystal resonator and external clock input, see section 20, Clock Pulse Generator. | | | | |
| | EXTAL | 66 | | Input For connection to a crystal resonator or input of an external clock signal. For examples of crystal resonator and external clock input, see section 20, Clock Pulse Generator. | | | | |
| | ϕ | 61 | | Output System clock: Supplies the system clock to external devices. | | | | |
| Operating mode control | MD_2 to MD_0 | 75 to 73 | | Input Mode 2 to mode 0: For setting the operating mode, as follows. Inputs at these pins must not be changed during operation. | | | | |
| | | | | | MD_2 | MD_1 | MD_0 | Operating Mode |
| | | | | | 0 | 0 | 0 | Setting prohibited |
| | | | | | 0 | 0 | 1 | Mode 1 |
| | | | | | 0 | 1 | 0 | Mode 2 |
| | | | | | 0 | 1 | 1 | Mode 3 |
| | | | | | 1 | 0 | 0 | Mode 4 |
| | | | | | 1 | 0 | 1 | Setting prohibited |
| | | | | | 1 | 1 | 0 | Setting prohibited |
| | | | | | 1 | 1 | 1 | Setting prohibited |

| Type | Symbol | Pin No. | | Name and Function |
|----------------|--|-------------------------------|--------------|--|
| | | FP-100B | TFP-100B | |
| System control | $\overline{\text{RES}}$ | 63 | Input | Reset input: When driven low, this pin resets the chip. This pin must be driven low at power-up. |
| | $\overline{\text{RESO}}$ | 10 | Output | Reset output: Outputs the reset signal generated by the watchdog timer to external devices |
| | $\overline{\text{STBY}}$ | 62 | Input | Standby: When driven low, this pin forces a transition to hardware standby mode |
| | $\overline{\text{BREQ}}$ | 59 | Input | Bus request: Used by an external bus master to request the bus right |
| | $\overline{\text{BACK}}$ | 60 | Output | Bus request acknowledge: Indicates that the bus has been granted to an external bus master |
| Interrupts | NMI | 64 | Input | Nonmaskable interrupt: Requests a nonmaskable interrupt |
| | $\overline{\text{IRQ}}_5$ to $\overline{\text{IRQ}}_0$ | 17, 16, 90 to 87 | Input | Interrupt request 5 to 0: Maskable interrupt request pins |
| Address bus | A_{23} to A_0 | 97 to 100, 56 to 45, 43 to 36 | Output | Address bus: Outputs address signals |
| Data bus | D_{15} to D_0 | 34 to 23, 21 to 18 | Input/output | Data bus: Bidirectional data bus |
| Bus control | $\overline{\text{CS}}_7$ to $\overline{\text{CS}}_0$ | 2 to 5, 88 to 91 | Output | Chip select: Select signals for areas 7 to 0 |
| | $\overline{\text{AS}}$ | 69 | Output | Address strobe: Goes low to indicate valid address output on the address bus |
| | $\overline{\text{RD}}$ | 70 | Output | Read: Goes low to indicate reading from the external address space |
| | $\overline{\text{HWR}}$ | 71 | Output | High write: Goes low to indicate writing to the external address space; indicates valid data on the upper data bus (D_{15} to D_8). |
| | $\overline{\text{LWR}}$ | 72 | Output | Low write: Goes low to indicate writing to the external address space; indicates valid data on the lower data bus (D_7 to D_0). |
| | $\overline{\text{WAIT}}$ | 58 | Input | Wait: Requests insertion of wait states in bus cycles during access to the external address space |

| Type | Symbol | Pin No. | | Name and Function |
|--|---|----------------------|------------------|---|
| | | FP-100B | TFP-100B | |
| 16-bit timer | TCLKD to TCLKA | 96 to 93 | Input | Clock input D to A: External clock inputs |
| | TIOCA ₂ to TIOCA ₀ | 99, 97, 95 | Input/ output | Input capture/output compare A2 to A0: GRA2 to GRA0 output compare or input capture, or PWM output |
| | TIOCB ₂ to TIOCB ₀ | 100, 98, 96 | Input/ output | Input capture/output compare B2 to B0: GRB2 to GRB0 output compare or input capture |
| 8-bit timer | TMO ₀ , TMO ₂ | 2, 4 | Output | Compare match output: Compare match output pins |
| | TMIO ₁ , TMIO ₃ | 3, 5 | Input/ output | Input capture input/compare match output: Input capture input or compare match output pins |
| | TCLKD to TCLKA | 96 to 93 | Input | Counter external clock input: These pins input an external clock to the counters. |
| Program- mable timing pattern controller (TPC) | TP ₁₅ to TP ₀ | 9 to 2, 100 to 93 | Output | TPC output 15 to 0: Pulse output |
| Serial com- munication interface (SCI) | TxD ₁ , TxD ₀ | 13, 12 | Output | Transmit data (channels 0, 1): SCI data output |
| | RxD ₁ , RxD ₀ | 15, 14 | Input | Receive data (channels 0, 1): SCI data input |
| | SCK ₁ , SCK ₀ | 17, 16 | Input/ output | Serial clock (channels 0, 1): SCI clock input/output |
| A/D converter | AN ₇ to AN ₀ | 85 to 78 | Input | Analog 7 to 0: Analog input pins |
| | ADTRG | 90 | Input | A/D conversion external trigger input: External trigger input for starting A/D conversion |
| D/A converter | DA ₁ , DA ₀ | 85, 84 | Output | Analog output: Analog output from the D/A converter |

| Type | Symbol | Pin No. | | Name and Function |
|---------------------|------------------------------|-----------------------|------------------|---|
| | | FP-100B | TFP-100B | |
| Analog power supply | AV_{CC} | 76 | Input | Power supply pin for the A/D and D/A converters. Connect to the system power supply when not using the A/D and D/A converters. |
| | AV_{SS} | 86 | Input | Ground pin for the A/D and D/A converters. Connect to system ground (0 V). |
| | V_{REF} | 77 | Input | Reference voltage input pin for the A/D and D/A converters. Connect to the system power supply when not using the A/D and D/A converters. |
| I/O ports | $P4_7$ to $P4_0$ | 26 to 23, 21 to 18 | Input/ output | Port 4: Eight input/output pins. The direction of each pin can be selected in the port 4 data direction register (P4DDR). |
| | $P6_7$, $P6_5$ to $P6_0$ | 61, 60 to 58 | Input/ output | Port 6: Eight input/output pins. The direction of each pin can be selected in the port 6 data direction register (P6DDR). |
| | $P7_7$ to $P7_0$ | 85 to 78 | Input | Port 7: Eight input pins |
| | $P8_4$ to $P8_0$ | 91 to 87 | Input/ output | Port 8: Five input/output pins. The direction of each pin can be selected in the port 8 data direction register (P8DDR). |
| | $P9_5$ to $P9_0$ | 17 to 12 | Input/ output | Port 9: Six input/output pins. The direction of each pin can be selected in the port 9 data direction register (P9DDR). |
| | PA_7 to PA_0 | 100 to 93 | Input/ output | Port A: Eight input/output pins. The direction of each pin can be selected in the port A data direction register (PADDR). |
| | PB_7 to PB_0 | 9 to 2 | Input/ output | Port B: Eight input/output pins. The direction of each pin can be selected in the port B data direction register (PBDDR). |

Note: * In 5 V operation models. This is a V_{CC} pin in 3 V operation models.

1.3.3 Pin Assignments in Each Mode

Table 1.4 lists the pin assignments in each mode.

Table 1.4 Pin Assignments in Each Mode (FP-100B, TFP-100B)

| Pin No. | Pin Name | | | |
|---------------------|---------------------------------------|---------------------------------------|---------------------------------------|---------------------------------------|
| FP-100B TFP-100B | Mode 1 | Mode 2 | Mode 3 | Mode 4 |
| 1 | $V_{CC}(V_{CL})^{*3}$ | $V_{CC}(V_{CL})^{*3}$ | $V_{CC}(V_{CL})^{*3}$ | $V_{CC}(V_{CL})^{*3}$ |
| 2 | $PB_0/TP_8/TMO_0/\overline{CS}_7$ | $PB_0/TP_8/TMO_0/\overline{CS}_7$ | $PB_0/TP_8/TMO_0/\overline{CS}_7$ | $PB_0/TP_8/TMO_0/\overline{CS}_7$ |
| 3 | $PB_1/TP_9/TMIO_1/\overline{CS}_6$ | $PB_1/TP_9/TMIO_1/\overline{CS}_6$ | $PB_1/TP_9/TMIO_1/\overline{CS}_6$ | $PB_1/TP_9/TMIO_1/\overline{CS}_6$ |
| 4 | $PB_2/TP_{10}/TMO_2/\overline{CS}_5$ | $PB_2/TP_{10}/TMO_2/\overline{CS}_5$ | $PB_2/TP_{10}/TMO_2/\overline{CS}_5$ | $PB_2/TP_{10}/TMO_2/\overline{CS}_5$ |
| 5 | $PB_3/TP_{11}/TMIO_3/\overline{CS}_4$ | $PB_3/TP_{11}/TMIO_3/\overline{CS}_4$ | $PB_3/TP_{11}/TMIO_3/\overline{CS}_4$ | $PB_3/TP_{11}/TMIO_3/\overline{CS}_4$ |
| 6 | PB_4/TP_{12} | PB_4/TP_{12} | PB_4/TP_{12} | PB_4/TP_{12} |
| 7 | PB_5/TP_{13} | PB_5/TP_{13} | PB_5/TP_{13} | PB_5/TP_{13} |
| 8 | PB_6/TP_{14} | PB_6/TP_{14} | PB_6/TP_{14} | PB_6/TP_{14} |
| 9 | PB_7/TP_{15} | PB_7/TP_{15} | PB_7/TP_{15} | PB_7/TP_{15} |
| 10 | \overline{RESO} | \overline{RESO} | \overline{RESO} | \overline{RESO} |
| 11 | V_{SS} | V_{SS} | V_{SS} | V_{SS} |
| 12 | $P9_0/TxD_0$ | $P9_0/TxD_0$ | $P9_0/TxD_0$ | $P9_0/TxD_0$ |
| 13 | $P9_1/TxD_1$ | $P9_1/TxD_1$ | $P9_1/TxD_1$ | $P9_1/TxD_1$ |
| 14 | $P9_2/RxD_0$ | $P9_2/RxD_0$ | $P9_2/RxD_0$ | $P9_2/RxD_0$ |
| 15 | $P9_3/RxD_1$ | $P9_3/RxD_1$ | $P9_3/RxD_1$ | $P9_3/RxD_1$ |
| 16 | $P9_4/SCK_0/\overline{IRQ}_4$ | $P9_4/SCK_0/\overline{IRQ}_4$ | $P9_4/SCK_0/\overline{IRQ}_4$ | $P9_4/SCK_0/\overline{IRQ}_4$ |
| 17 | $P9_5/SCK_1/\overline{IRQ}_5$ | $P9_5/SCK_1/\overline{IRQ}_5$ | $P9_5/SCK_1/\overline{IRQ}_5$ | $P9_5/SCK_1/\overline{IRQ}_5$ |
| 18 | $P4_0/D_0^{*1}$ | $P4_0/D_0^{*2}$ | $P4_0/D_0^{*1}$ | $P4_0/D_0^{*2}$ |
| 19 | $P4_1/D_1^{*1}$ | $P4_1/D_1^{*2}$ | $P4_1/D_1^{*1}$ | $P4_1/D_1^{*2}$ |
| 20 | $P4_2/D_2^{*1}$ | $P4_2/D_2^{*2}$ | $P4_2/D_2^{*1}$ | $P4_2/D_2^{*2}$ |
| 21 | $P4_3/D_3^{*1}$ | $P4_3/D_3^{*2}$ | $P4_3/D_3^{*1}$ | $P4_3/D_3^{*2}$ |
| 22 | V_{SS} | V_{SS} | V_{SS} | V_{SS} |
| 23 | $P4_4/D_4^{*1}$ | $P4_4/D_4^{*2}$ | $P4_4/D_4^{*1}$ | $P4_4/D_4^{*2}$ |
| 24 | $P4_5/D_5^{*1}$ | $P4_5/D_5^{*2}$ | $P4_5/D_5^{*1}$ | $P4_5/D_5^{*2}$ |
| 25 | $P4_6/D_6^{*1}$ | $P4_6/D_6^{*2}$ | $P4_6/D_6^{*1}$ | $P4_6/D_6^{*2}$ |
| 26 | $P4_7/D_7^{*1}$ | $P4_7/D_7^{*2}$ | $P4_7/D_7^{*1}$ | $P4_7/D_7^{*2}$ |
| 27 | D_8 | D_8 | D_8 | D_8 |

| Pin No. | Pin Name | | | |
|---------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| FP-100B TFP-100B | Mode 1 | Mode 2 | Mode 3 | Mode 4 |
| 28 | D ₉ | D ₉ | D ₉ | D ₉ |
| 29 | D ₁₀ | D ₁₀ | D ₁₀ | D ₁₀ |
| 30 | D ₁₁ | D ₁₁ | D ₁₁ | D ₁₁ |
| 31 | D ₁₂ | D ₁₂ | D ₁₂ | D ₁₂ |
| 32 | D ₁₃ | D ₁₃ | D ₁₃ | D ₁₃ |
| 33 | D ₁₄ | D ₁₄ | D ₁₄ | D ₁₄ |
| 34 | D ₁₅ | D ₁₅ | D ₁₅ | D ₁₅ |
| 35 | V _{CC} | V _{CC} | V _{CC} | V _{CC} |
| 36 | A ₀ | A ₀ | A ₀ | A ₀ |
| 37 | A ₁ | A ₁ | A ₁ | A ₁ |
| 38 | A ₂ | A ₂ | A ₂ | A ₂ |
| 39 | A ₃ | A ₃ | A ₃ | A ₃ |
| 40 | A ₄ | A ₄ | A ₄ | A ₄ |
| 41 | A ₅ | A ₅ | A ₅ | A ₅ |
| 42 | A ₆ | A ₆ | A ₆ | A ₆ |
| 43 | A ₇ | A ₇ | A ₇ | A ₇ |
| 44 | V _{SS} | V _{SS} | V _{SS} | V _{SS} |
| 45 | A ₈ | A ₈ | A ₈ | A ₈ |
| 46 | A ₉ | A ₉ | A ₉ | A ₉ |
| 47 | A ₁₀ | A ₁₀ | A ₁₀ | A ₁₀ |
| 48 | A ₁₁ | A ₁₁ | A ₁₁ | A ₁₁ |
| 49 | A ₁₂ | A ₁₂ | A ₁₂ | A ₁₂ |
| 50 | A ₁₃ | A ₁₃ | A ₁₃ | A ₁₃ |
| 51 | A ₁₄ | A ₁₄ | A ₁₄ | A ₁₄ |
| 52 | A ₁₅ | A ₁₅ | A ₁₅ | A ₁₅ |
| 53 | A ₁₆ | A ₁₆ | A ₁₆ | A ₁₆ |
| 54 | A ₁₇ | A ₁₇ | A ₁₇ | A ₁₇ |
| 55 | A ₁₈ | A ₁₈ | A ₁₈ | A ₁₈ |
| 56 | A ₁₉ | A ₁₉ | A ₁₉ | A ₁₉ |
| 57 | V _{SS} | V _{SS} | V _{SS} | V _{SS} |
| 58 | P6 _Q /WAIT | P6 _Q /WAIT | P6 _Q /WAIT | P6 _Q /WAIT |

1. Overview

| Pin No. | Pin Name | | | |
|---------------------|---|---|---|---|
| FP-100B TFP-100B | Mode 1 | Mode 2 | Mode 3 | Mode 4 |
| 59 | $P6_1/\overline{BREQ}$ | $P6_1/\overline{BREQ}$ | $P6_1/\overline{BREQ}$ | $P6_1/\overline{BREQ}$ |
| 60 | $P6_2/\overline{BACK}$ | $P6_2/\overline{BACK}$ | $P6_2/\overline{BACK}$ | $P6_2/\overline{BACK}$ |
| 61 | ϕ | ϕ | ϕ | ϕ |
| 62 | \overline{STBY} | \overline{STBY} | \overline{STBY} | \overline{STBY} |
| 63 | \overline{RES} | \overline{RES} | \overline{RES} | \overline{RES} |
| 64 | NMI | NMI | NMI | NMI |
| 65 | V_{SS} | V_{SS} | V_{SS} | V_{SS} |
| 66 | EXTAL | EXTAL | EXTAL | EXTAL |
| 67 | XTAL | XTAL | XTAL | XTAL |
| 68 | V_{CC} | V_{CC} | V_{CC} | V_{CC} |
| 69 | \overline{AS} | \overline{AS} | \overline{AS} | \overline{AS} |
| 70 | \overline{RD} | \overline{RD} | \overline{RD} | \overline{RD} |
| 71 | \overline{HWR} | \overline{HWR} | \overline{HWR} | \overline{HWR} |
| 72 | \overline{LWR} | \overline{LWR} | \overline{LWR} | \overline{LWR} |
| 73 | MD_0 | MD_0 | MD_0 | MD_0 |
| 74 | MD_1 | MD_1 | MD_1 | MD_1 |
| 75 | MD_2 | MD_2 | MD_2 | MD_2 |
| 76 | AV_{CC} | AV_{CC} | AV_{CC} | AV_{CC} |
| 77 | V_{REF} | V_{REF} | V_{REF} | V_{REF} |
| 78 | $P7_0/AN_0$ | $P7_0/AN_0$ | $P7_0/AN_0$ | $P7_0/AN_0$ |
| 79 | $P7_1/AN_1$ | $P7_1/AN_1$ | $P7_1/AN_1$ | $P7_1/AN_1$ |
| 80 | $P7_2/AN_2$ | $P7_2/AN_2$ | $P7_2/AN_2$ | $P7_2/AN_2$ |
| 81 | $P7_3/AN_3$ | $P7_3/AN_3$ | $P7_3/AN_3$ | $P7_3/AN_3$ |
| 82 | $P7_4/AN_4$ | $P7_4/AN_4$ | $P7_4/AN_4$ | $P7_4/AN_4$ |
| 83 | $P7_5/AN_5$ | $P7_5/AN_5$ | $P7_5/AN_5$ | $P7_5/AN_5$ |
| 84 | $P7_6/AN_6/DA_0$ | $P7_6/AN_6/DA_0$ | $P7_6/AN_6/DA_0$ | $P7_6/AN_6/DA_0$ |
| 85 | $P7_7/AN_7/DA_1$ | $P7_7/AN_7/DA_1$ | $P7_7/AN_7/DA_1$ | $P7_7/AN_7/DA_1$ |
| 86 | AV_{SS} | AV_{SS} | AV_{SS} | AV_{SS} |
| 87 | $P8_0/\overline{IRQ_0}$ | $P8_0/\overline{IRQ_0}$ | $P8_0/\overline{IRQ_0}$ | $P8_0/\overline{IRQ_0}$ |
| 88 | $P8_1/\overline{IRQ_1}/\overline{CS_3}$ | $P8_1/\overline{IRQ_1}/\overline{CS_3}$ | $P8_1/\overline{IRQ_1}/\overline{CS_3}$ | $P8_1/\overline{IRQ_1}/\overline{CS_3}$ |
| 89 | $P8_2/\overline{IRQ_2}/\overline{CS_2}$ | $P8_2/\overline{IRQ_2}/\overline{CS_2}$ | $P8_2/\overline{IRQ_2}/\overline{CS_2}$ | $P8_2/\overline{IRQ_2}/\overline{CS_2}$ |

| Pin No. | Pin Name | | | |
|---------------------|--|--|--|--|
| FP-100B TFP-100B | Mode 1 | Mode 2 | Mode 3 | Mode 4 |
| 90 | $P8_3/\overline{IRQ}_3/\overline{CS}_1/\overline{ADTRG}$ | $P8_3/\overline{IRQ}_3/\overline{CS}_1/\overline{ADTRG}$ | $P8_3/\overline{IRQ}_3/\overline{CS}_1/\overline{ADTRG}$ | $P8_3/\overline{IRQ}_3/\overline{CS}_1/\overline{ADTRG}$ |
| 91 | $P8_4/\overline{CS}_0$ | $P8_4/\overline{CS}_0$ | $P8_4/\overline{CS}_0$ | $P8_4/\overline{CS}_0$ |
| 92 | V_{SS} | V_{SS} | V_{SS} | V_{SS} |
| 93 | $PA_0/TP_0/TCLKA$ | $PA_0/TP_0/TCLKA$ | $PA_0/TP_0/TCLKA$ | $PA_0/TP_0/TCLKA$ |
| 94 | $PA_1/TP_1/TCLKB$ | $PA_1/TP_1/TCLKB$ | $PA_1/TP_1/TCLKB$ | $PA_1/TP_1/TCLKB$ |
| 95 | $PA_2/TP_2/TIOCA_0/\overline{TCLKC}$ | $PA_2/TP_2/TIOCA_0/\overline{TCLKC}$ | $PA_2/TP_2/TIOCA_0/\overline{TCLKC}$ | $PA_2/TP_2/TIOCA_0/\overline{TCLKC}$ |
| 96 | $PA_3/TP_3/TIOCB_0/\overline{TCLKD}$ | $PA_3/TP_3/TIOCB_0/\overline{TCLKD}$ | $PA_3/TP_3/TIOCB_0/\overline{TCLKD}$ | $PA_3/TP_3/TIOCB_0/\overline{TCLKD}$ |
| 97 | $PA_4/TP_4/TIOCA_1$ | $PA_4/TP_4/TIOCA_1$ | $PA_4/TP_4/TIOCA_1/A_{23}$ | $PA_4/TP_4/TIOCA_1/A_{23}$ |
| 98 | $PA_5/TP_5/TIOCB_1$ | $PA_5/TP_5/TIOCB_1$ | $PA_5/TP_5/TIOCB_1/A_{22}$ | $PA_5/TP_5/TIOCB_1/A_{22}$ |
| 99 | $PA_6/TP_6/TIOCA_2$ | $PA_6/TP_6/TIOCA_2$ | $PA_6/TP_6/TIOCA_2/A_{21}$ | $PA_6/TP_6/TIOCA_2/A_{21}$ |
| 100 | $PA_7/TP_7/TIOCB_2$ | $PA_7/TP_7/TIOCB_2$ | A_{20} | A_{20} |

- Notes:
1. In modes 1 and 3 the $P4_0$ to $P4_7$ functions of pins $P4_0/D_0$ to $P4_7/D_7$ are selected after a reset, but they can be changed by software.
 2. In modes 2 and 4 the D_0 to D_7 functions of pins $P4_0/D_0$ to $P4_7/D_7$ are selected after a reset, but they can be changed by software.
 3. This pin functions as V_{CL} in 5 V operation models, and as V_{CC} in 3 V operation models.

Section 2 CPU

2.1 Overview

The H8/300H CPU is a high-speed central processing unit with an internal 32-bit architecture that is upward-compatible with the H8/300 CPU. The H8/300H CPU has sixteen 16-bit general registers, can address a 16-Mbyte linear address space, and is ideal for realtime control.

2.1.1 Features

The H8/300H CPU has the following features.

- Upward compatibility with H8/300 CPU
Can execute H8/300 Series object programs
- General-register architecture
Sixteen 16-bit general registers (also usable as sixteen 8-bit registers or eight 32-bit registers)
- 64 basic instructions
 - 8/16/32-bit arithmetic and logic instructions
 - Multiply and divide instructions
 - Powerful bit-manipulation instructions
- Eight addressing modes
 - Register direct [Rn]
 - Register indirect [@ERn]
 - Register indirect with displacement [@(d:16, ERn) or @(d:24, ERn)]
 - Register indirect with post-increment or pre-decrement [@ERn+ or @-ERn]
 - Absolute address [@aa:8, @aa:16, or @aa:24]
 - Immediate [#xx:8, #xx:16, or #xx:32]
 - Program-counter relative [@(d:8, PC) or @(d:16, PC)]
 - Memory indirect [@@aa:8]
- 16-Mbyte linear address space
- High-speed operation
 - All frequently-used instructions execute in two to four states
 - Maximum clock frequency: 25 MHz
 - 8/16/32-bit register-register add/subtract: 80 ns@25 MHz
 - 8 × 8-bit register-register multiply: 560 ns@25 MHz
 - 16 ÷ 8-bit register-register divide: 560 ns@25 MHz
 - 16 × 16-bit register-register multiply: 880 ns@25 MHz

— $32 \div 16$ -bit register-register divide: 880 ns@25 MHz

- Two CPU operating modes
 - Normal mode
 - Advanced mode
- Low-power mode
 - Transition to power-down state by SLEEP instruction

2.1.2 Differences from H8/300 CPU

In comparison to the H8/300 CPU, the H8/300H CPU has the following enhancements.

- More general registers
 - Eight 16-bit registers have been added.
- Expanded address space
 - Advanced mode supports a maximum 16-Mbyte address space.
 - Normal mode supports the same 64-kbyte address space as the H8/300 CPU.
- Enhanced addressing
 - The addressing modes have been enhanced to make effective use of the 16-Mbyte address space.
- Enhanced instructions
 - Data transfer, arithmetic, and logic instructions can operate on 32-bit data.
 - Signed multiply/divide instructions and other instructions have been added.

2.2 CPU Operating Modes

The H8/300H CPU has two operating modes: normal and advanced. Normal mode supports a maximum 64-kbyte address space. Advanced mode supports up to 16 Mbytes.

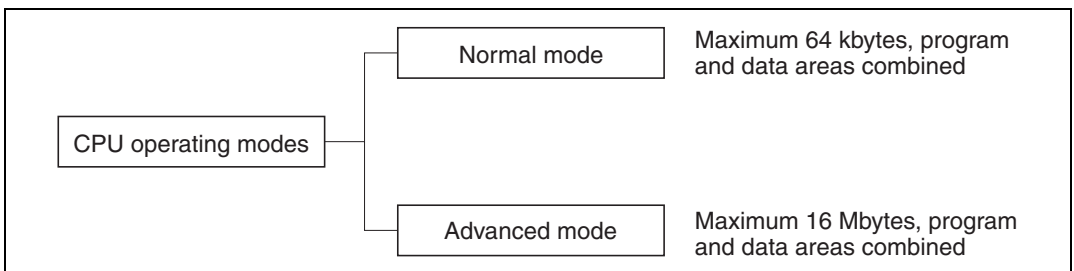


Figure 2.1 CPU Operating Modes

2.3 Address Space

Figure 2.2 shows a simple memory map for the H8/3008. The H8/300H CPU can address a linear address space with a maximum size of 64 kbytes in normal mode, and 16 Mbytes in advanced mode. For further details see section 3.6, Memory Map in Each Operating Mode.

The 1-Mbyte operating modes use 20-bit addressing. The upper 4 bits of effective addresses are ignored.

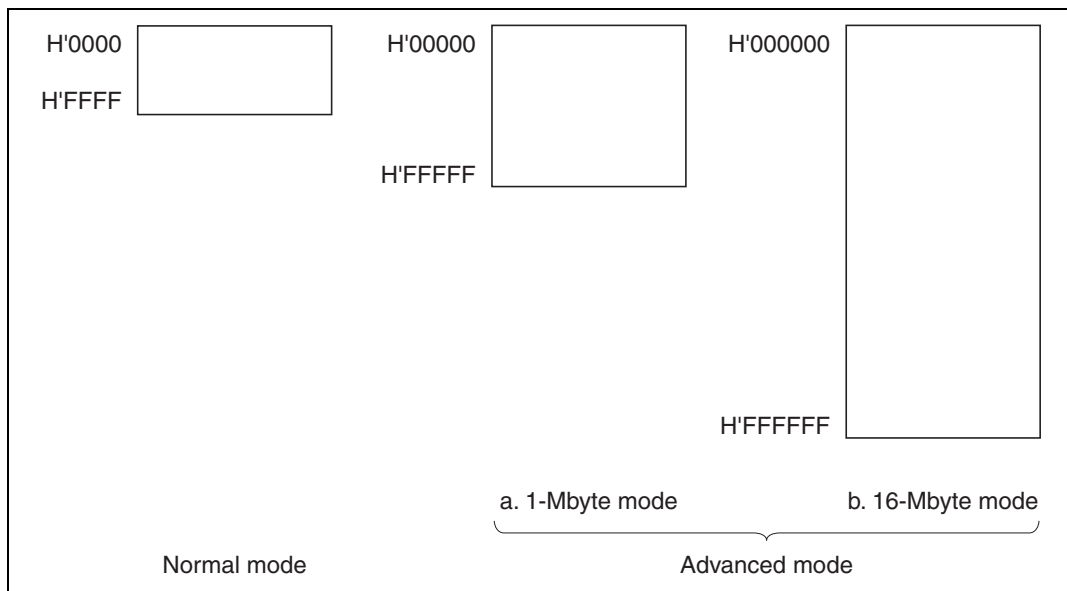


Figure 2.2 Memory Map

2.4 Register Configuration

2.4.1 Overview

The H8/300H CPU has the internal registers shown in figure 2.3. There are two types of registers: general registers and control registers.

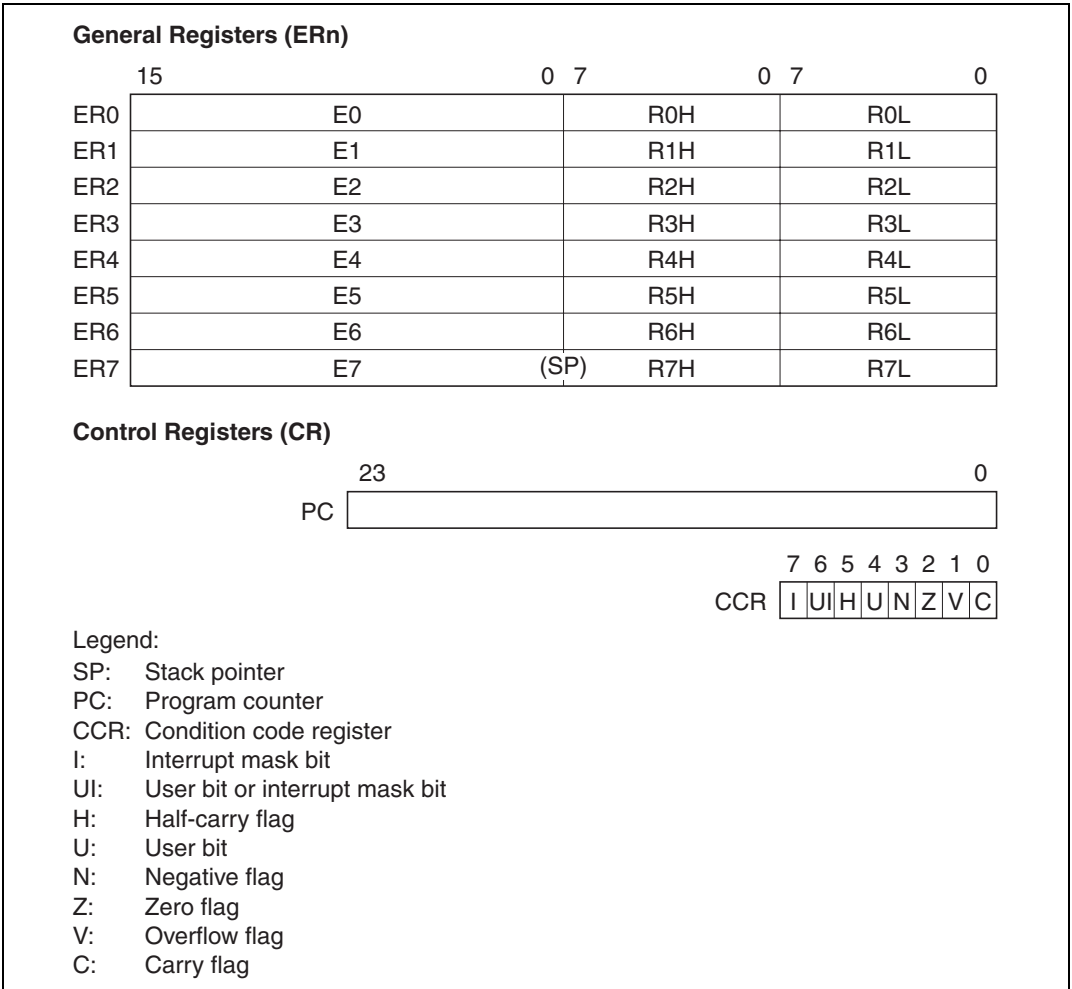


Figure 2.3 CPU Registers

2.4.2 General Registers

The H8/300H CPU has eight 32-bit general registers. These general registers are all functionally alike and can be used without distinction between data registers and address registers. When a general register is used as a data register, it can be accessed as a 32-bit, 16-bit, or 8-bit register. When the general registers are used as 32-bit registers or as address registers, they are designated by the letters ER (ER0 to ER7).

The ER registers divide into 16-bit general registers designated by the letters E (E0 to E7) and R (R0 to R7). These registers are functionally equivalent, providing a maximum sixteen 16-bit registers. The E registers (E0 to E7) are also referred to as extended registers.

The R registers divide into 8-bit general registers designated by the letters RH (R0H to R7H) and RL (R0L to R7L). These registers are functionally equivalent, providing a maximum sixteen 8-bit registers.

Figure 2.4 illustrates the usage of the general registers. The usage of each register can be selected independently.

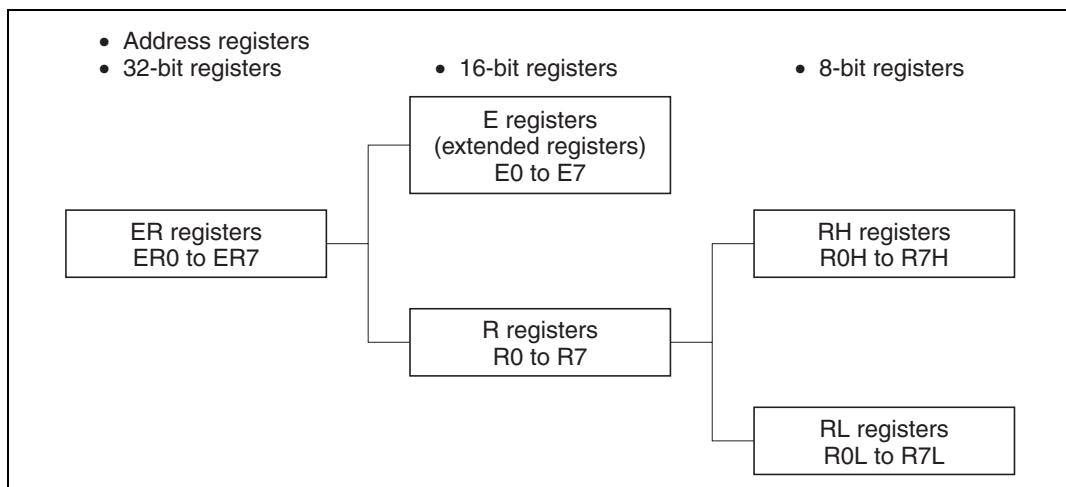


Figure 2.4 Usage of General Registers

General register ER7 has the function of stack pointer (SP) in addition to its general-register function, and is used implicitly in exception handling and subroutine calls. Figure 2.5 shows the stack.

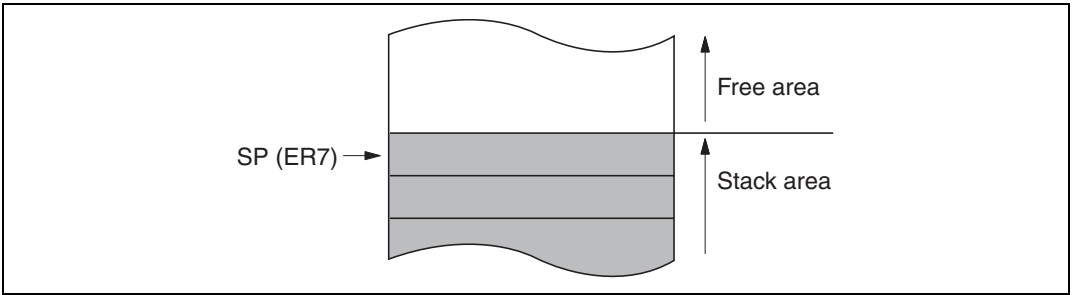


Figure 2.5 Stack

2.4.3 Control Registers

The control registers are the 24-bit program counter (PC) and the 8-bit condition code register (CCR).

Program Counter (PC): This 24-bit counter indicates the address of the next instruction the CPU will execute. The length of all CPU instructions is 2 bytes (one word), so the least significant PC bit is ignored. When an instruction is fetched, the least significant PC bit is regarded as 0.

Condition Code Register (CCR): This 8-bit register contains internal CPU status information, including the interrupt mask bit (I) and half-carry (H), negative (N), zero (Z), overflow (V), and carry (C) flags.

Bit 7—Interrupt Mask Bit (I): Masks interrupts other than NMI when set to 1. NMI is accepted regardless of the I bit setting. The I bit is set to 1 at the start of an exception-handling sequence.

Bit 6—User Bit or Interrupt Mask Bit (UI): Can be written and read by software using the LDC, STC, ANDC, ORC, and XORC instructions. This bit can also be used as an interrupt mask bit. For details see section 5, Interrupt Controller.

Bit 5—Half-Carry Flag (H): When the ADD.B, ADDX.B, SUB.B, SUBX.B, CMP.B, or NEG.B instruction is executed, this flag is set to 1 if there is a carry or borrow at bit 3, and cleared to 0 otherwise. When the ADD.W, SUB.W, CMP.W, or NEG.W instruction is executed, the H flag is set to 1 if there is a carry or borrow at bit 11, and cleared to 0 otherwise. When the ADD.L, SUB.L, CMP.L, or NEG.L instruction is executed, the H flag is set to 1 if there is a carry or borrow at bit 27, and cleared to 0 otherwise.

Bit 4—User Bit (U): Can be written and read by software using the LDC, STC, ANDC, ORC, and XORC instructions.

Bit 3—Negative Flag (N): Stores the value of the most significant bit of data, regarded as the sign bit.

Bit 2—Zero Flag (Z): Set to 1 to indicate zero data, and cleared to 0 to indicate non-zero data.

Bit 1—Overflow Flag (V): Set to 1 when an arithmetic overflow occurs, and cleared to 0 at other times.

Bit 0—Carry Flag (C): Set to 1 when a carry is generated by execution of an operation, and cleared to 0 otherwise. Used by:

- Add instructions, to indicate a carry
- Subtract instructions, to indicate a borrow
- Shift and rotate instructions

The carry flag is also used as a bit accumulator by bit manipulation instructions.

Some instructions leave flag bits unchanged. Operations can be performed on CCR by the LDC, STC, ANDC, ORC, and XORC instructions. The N, Z, V, and C flags are used by conditional branch (Bcc) instructions.

For the action of each instruction on the flag bits, see appendix A.1, Instruction List. For the I and UI bits, see section 5, Interrupt Controller.

2.4.4 Initial CPU Register Values

In reset exception handling, PC is initialized to a value loaded from the vector table, and the I bit in CCR is set to 1. The other CCR bits and the general registers are not initialized. In particular, the initial value of the stack pointer (ER7) is also undefined. The stack pointer (ER7) must therefore be initialized by an MOV.L instruction executed immediately after a reset.

2.5 Data Formats

The H8/300H CPU can process 1-bit, 4-bit (BCD), 8-bit (byte), 16-bit (word), and 32-bit (longword) data. Bit-manipulation instructions operate on 1-bit data by accessing bit n ($n = 0, 1, 2, \dots, 7$) of byte operand data. The DAA and DAS decimal-adjust instructions treat byte data as two digits of 4-bit BCD data.

2.5.1 General Register Data Formats

Figures 2.6 and 2.7 show the data formats in general registers.

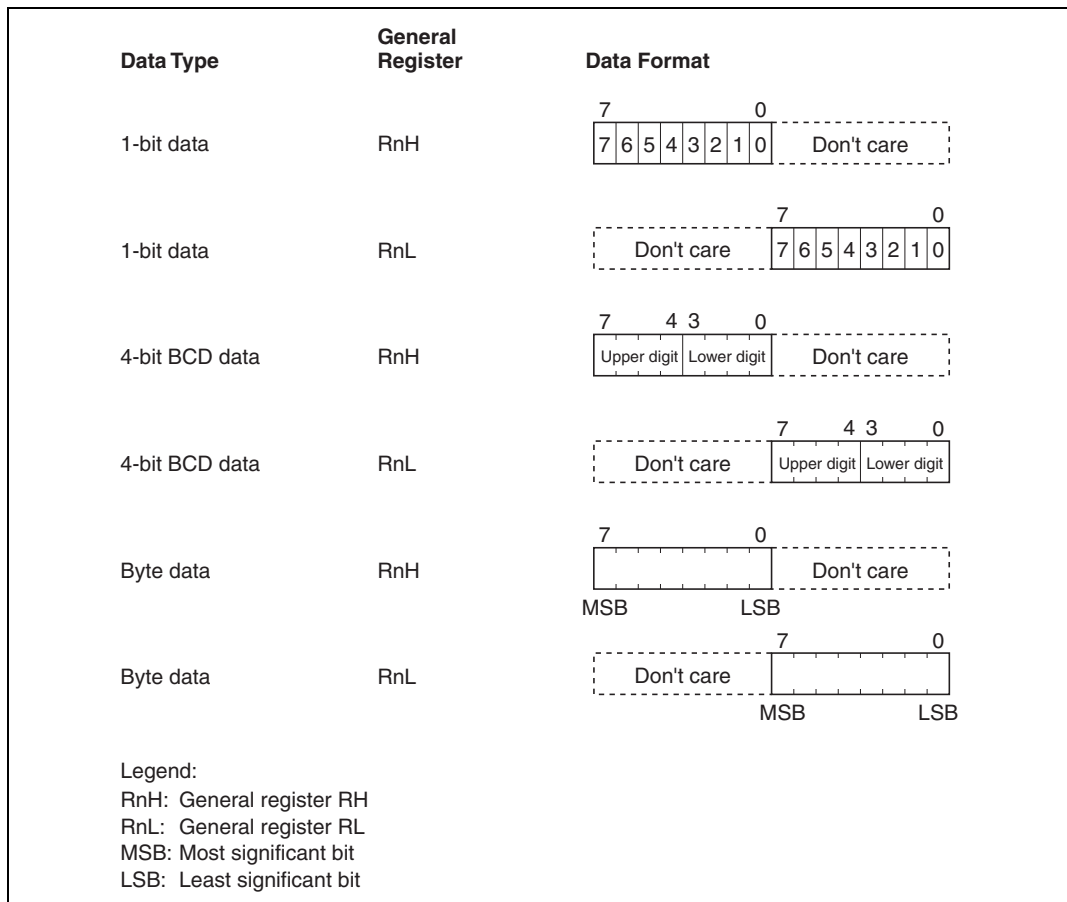


Figure 2.6 General Register Data Formats (1)

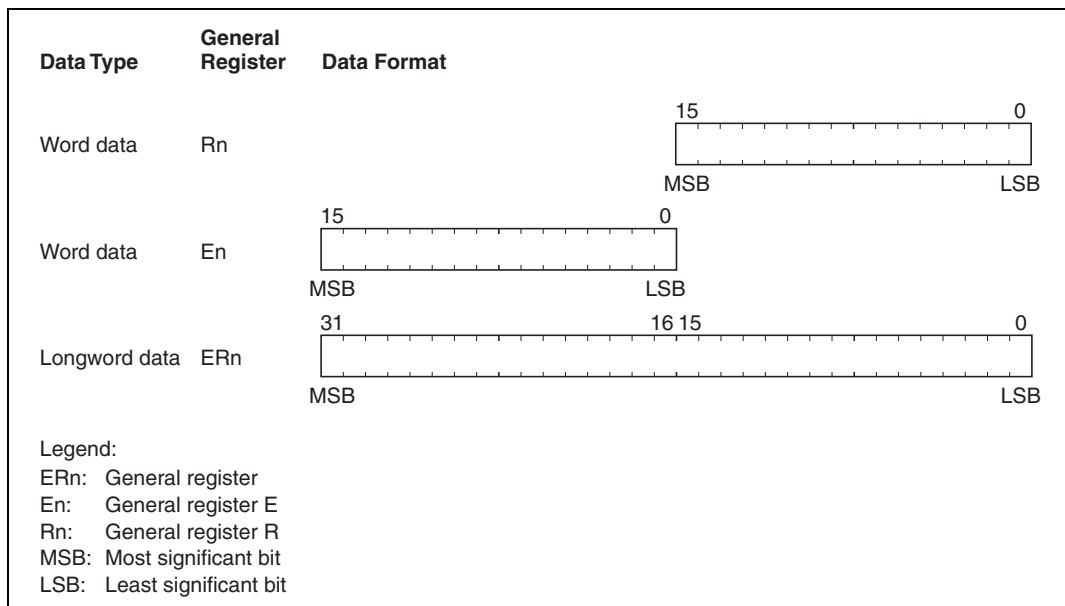


Figure 2.7 General Register Data Formats (2)

2.5.2 Memory Data Formats

Figure 2.8 shows the data formats on memory. The H8/300H CPU can access word data and longword data on memory, but word or longword data must begin at an even address. If an attempt is made to access word or longword data at an odd address, no address error occurs but the least significant bit of the address is regarded as 0, so the access starts at the preceding address. This also applies to instruction fetches.

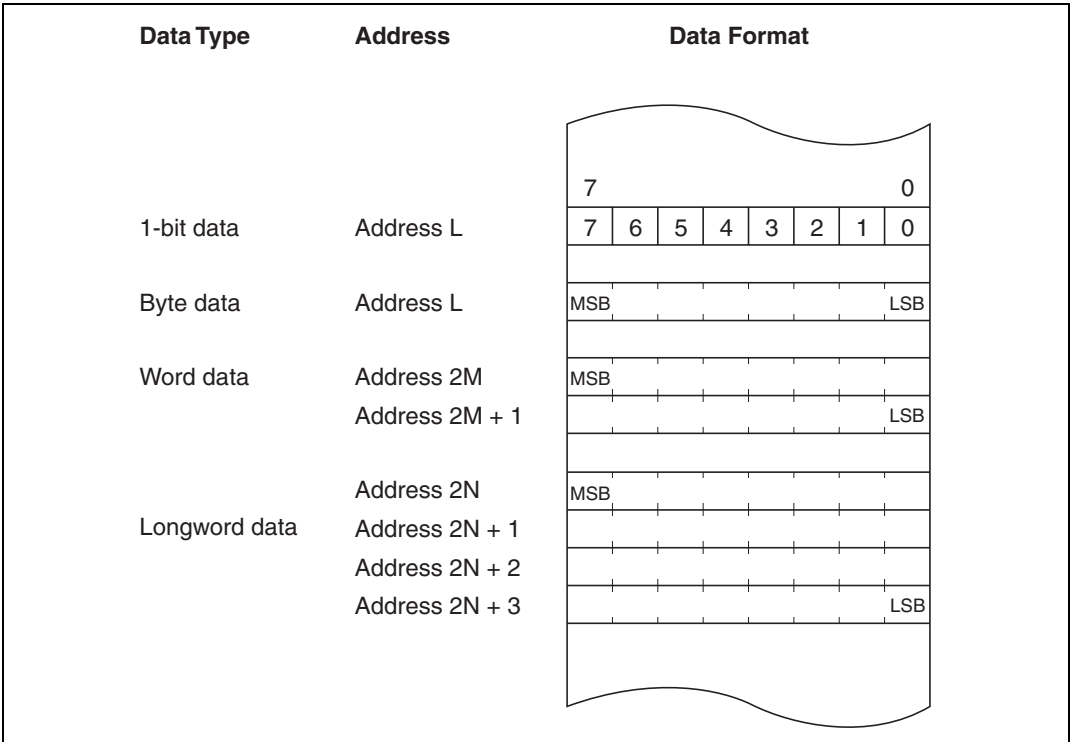


Figure 2.8 Memory Data Formats

When ER7 (SP) is used as an address register to access the stack, the operand size should be word size or longword size.

2.6 Instruction Set

2.6.1 Instruction Set Overview

The H8/300H CPU has 64 types of instructions, which are classified in table 2.1.

Table 2.1 Instruction Classification

| Function | Instruction | Types |
|-----------------------|--|-------|
| Data transfer | MOV, PUSH* ¹ , POP* ¹ , MOVTPE* ² , MOVFPE* ² | 5 |
| Arithmetic operations | ADD, SUB, ADDX, SUBX, INC, DEC, ADDS, SUBS, DAA, DAS, MULXU, MULXS, DIVXU, DIVXS, CMP, NEG, EXTS, EXTU | 18 |
| Logic operations | AND, OR, XOR, NOT | 4 |
| Shift operations | SHAL, SHAR, SHLL, SHLR, ROTL, ROTR, ROTXL, ROTXR | 8 |
| Bit manipulation | BSET, BCLR, BNOT, BTST, BAND, BAND, BOR, BIOR, BXOR, BIXOR, BLD, BILD, BST, BIST | 14 |
| Branch | Bcc* ³ , JMP, BSR, JSR, RTS | 5 |
| System control | TRAPA, RTE, SLEEP, LDC, STC, ANDC, ORC, XORC, NOP | 9 |
| Block data transfer | EEPMOV | 1 |

Total 64 types

- Notes: 1. POP.W Rn is identical to MOV.W @SP+, Rn.
 PUSH.W Rn is identical to MOV.W Rn, @-SP.
 POP.L ERn is identical to MOV.L @SP+, Rn.
 PUSH.L ERn is identical to MOV.L Rn, @-SP.
2. Not available in the H8/3008.
3. Bcc is a generic branching instruction.

2.6.2 Instructions and Addressing Modes

Table 2.2 indicates the instructions available in the H8/300H CPU.

Table 2.2 Instructions and Addressing Modes

| Function | Instruction | #xx | Rn | Addressing Modes | | | | | | | | | | |
|-----------------------|-------------------------------------|-----|-----|------------------|--------------|--------------|-------------|-------|--------|--------|------------|-------------|--------|----|
| | | | | @ERn | @(d:16, ERn) | @(d:24, ERn) | @ERn+/@-ERn | @aa:8 | @aa:16 | @aa:24 | @(d:8, PC) | @(d:16, PC) | @@aa:8 | — |
| Data transfer | MOV | BWL | BWL | BWL | BWL | BWL | BWL | B | BWL | BWL | — | — | — | — |
| | POP, PUSH | — | — | — | — | — | — | — | — | — | — | — | — | WL |
| | MOVFPE, MOVTPE | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Arithmetic operations | ADD, CMP | BWL | BWL | — | — | — | — | — | — | — | — | — | — | — |
| | SUB | WL | BWL | — | — | — | — | — | — | — | — | — | — | — |
| | ADDX, SUBX | B | B | — | — | — | — | — | — | — | — | — | — | — |
| | ADDS, SUBS | — | L | — | — | — | — | — | — | — | — | — | — | — |
| | INC, DEC | — | BWL | — | — | — | — | — | — | — | — | — | — | — |
| | DAA, DAS | — | B | — | — | — | — | — | — | — | — | — | — | — |
| | MULXU, MULXS, DIVXU, DIVXS | — | BW | — | — | — | — | — | — | — | — | — | — | — |
| | NEG | — | BWL | — | — | — | — | — | — | — | — | — | — | — |
| | EXTU, EXTS | — | WL | — | — | — | — | — | — | — | — | — | — | — |
| Logic operations | AND, OR, XOR | — | BWL | — | — | — | — | — | — | — | — | — | — | — |
| | NOT | — | BWL | — | — | — | — | — | — | — | — | — | — | — |
| Shift instructions | — | BWL | — | — | — | — | — | — | — | — | — | — | — | |
| Bit manipulation | — | B | B | — | — | — | B | — | — | — | — | — | — | |
| Branch | Bcc, BSR | — | — | — | — | — | — | — | — | — | — | — | — | — |
| | JMP, JSR | — | — | ○ | — | — | — | — | — | — | ○ | ○ | — | — |
| | RTS | — | — | — | — | — | — | — | — | ○ | — | — | ○ | — |
| System control | TRAPA | — | — | — | — | — | — | — | — | — | — | — | — | ○ |
| | RTE | — | — | — | — | — | — | — | — | — | — | — | — | ○ |
| | SLEEP | — | — | — | — | — | — | — | — | — | — | — | — | ○ |
| | LDC | B | B | W | W | W | W | — | W | W | — | — | — | ○ |
| | STC | — | B | W | W | W | W | — | W | W | — | — | — | — |
| | ANDC, ORC, XORC | B | — | — | — | — | — | — | — | — | — | — | — | — |
| | NOP | — | — | — | — | — | — | — | — | — | — | — | — | ○ |
| Block data transfer | — | — | — | — | — | — | — | — | — | — | — | — | — | BW |

2.6.3 Tables of Instructions Classified by Function

Tables 2.3 to 2.10 summarize the instructions in each functional category. The operation notation used in these tables is defined next.

Operation Notation

| | |
|---------------|---|
| Rd | General register (destination)* |
| Rs | General register (source)* |
| Rn | General register* |
| ERn | General register (32-bit register or address register)* |
| (EAd) | Destination operand |
| (EAs) | Source operand |
| CCR | Condition code register |
| N | N (negative) flag of CCR |
| Z | Z (zero) flag of CCR |
| V | V (overflow) flag of CCR |
| C | C (carry) flag of CCR |
| PC | Program counter |
| SP | Stack pointer |
| #IMM | Immediate data |
| disp | Displacement |
| + | Addition |
| − | Subtraction |
| × | Multiplication |
| ÷ | Division |
| ^ | AND logical |
| ∨ | OR logical |
| ⊕ | Exclusive OR logical |
| → | Move |
| ¬ | NOT (logical complement) |
| :3/:8/:16/:24 | 3-, 8-, 16-, or 24-bit length |

Note: * General registers include 8-bit registers (R0H to R7H, R0L to R7L), 16-bit registers (R0 to R7, E0 to E7), and 32-bit data or address registers (ER0 to ER7).

Table 2.3 Data Transfer Instructions

| Instruction | Size* | Function |
|--------------------|--------------|---|
| MOV | B/W/L | (EAs) → Rd, Rs → (EAd) Moves data between two general registers or between a general register and memory, or moves immediate data to a general register. |
| MOVFP | B | (EAs) → Rd Cannot be used in the H8/3008. |
| MOVTPE | B | Rs → (EAs) Cannot be used in the H8/3008. |
| POP | W/L | @SP+ → Rn Pops a general register from the stack. POP.W Rn is identical to MOV.W @SP+, Rn. Similarly, POP.L ERn is identical to MOV.L @SP+, ERn. |
| PUSH | W/L | Rn → @-SP Pushes a general register onto the stack. PUSH.W Rn is identical to MOV.W Rn, @-SP. Similarly, PUSH.L ERn is identical to MOV.L ERn, @-SP. |

Note: * Size refers to the operand size.

B: Byte

W: Word

L: Longword

Table 2.4 Arithmetic Operation Instructions

| Instruction | Size* | Function |
|--------------------|--------------|--|
| ADD,SUB | B/W/L | $Rd \pm Rs \rightarrow Rd$, $Rd \pm \#IMM \rightarrow Rd$ Performs addition or subtraction on data in two general registers, or on immediate data and data in a general register. (Immediate byte data cannot be subtracted from data in a general register. Use the SUBX or ADD instruction.) |
| ADDX, SUBX | B | $Rd \pm Rs \pm C \rightarrow Rd$, $Rd \pm \#IMM \pm C \rightarrow Rd$ Performs addition or subtraction with carry or borrow on data in two general registers, or on immediate data and data in a general register. |
| INC, DEC | B/W/L | $Rd \pm 1 \rightarrow Rd$, $Rd \pm 2 \rightarrow Rd$ Increments or decrements a general register by 1 or 2. (Byte operands can be incremented or decremented by 1 only.) |
| ADDS, SUBS | L | $Rd \pm 1 \rightarrow Rd$, $Rd \pm 2 \rightarrow Rd$, $Rd \pm 4 \rightarrow Rd$ Adds or subtracts the value 1, 2, or 4 to or from data in a 32-bit register. |
| DAA, DAS | B | Rd decimal adjust $\rightarrow Rd$ Decimal-adjusts an addition or subtraction result in a general register by referring to CCR to produce 4-bit BCD data. |
| MULXU | B/W | $Rd \times Rs \rightarrow Rd$ Performs unsigned multiplication on data in two general registers: either 8 bits \times 8 bits \rightarrow 16 bits or 16 bits \times 16 bits \rightarrow 32 bits. |
| MULXS | B/W | $Rd \times Rs \rightarrow Rd$ Performs signed multiplication on data in two general registers: either 8 bits \times 8 bits \rightarrow 16 bits or 16 bits \times 16 bits \rightarrow 32 bits. |

| Instruction | Size* | Function |
|-------------|-------|--|
| DIVXU | B/W | $Rd \div Rs \rightarrow Rd$ Performs unsigned division on data in two general registers: either 16 bits \div 8 bits \rightarrow 8-bit quotient and 8-bit remainder or 32 bits \div 16 bits \rightarrow 16-bit quotient and 16-bit remainder |
| DIVXS | B/W | $Rd \div Rs \rightarrow Rd$ Performs signed division on data in two general registers: either 16 bits \div 8 bits \rightarrow 8-bit quotient and 8-bit remainder, or 32 bits \div 16 bits \rightarrow 16-bit quotient and 16-bit remainder |
| CMP | B/W/L | $Rd - Rs, Rd - \#IMM$ Compares data in a general register with data in another general register or with immediate data, and sets CCR according to the result. |
| NEG | B/W/L | $0 - Rd \rightarrow Rd$ Takes the two's complement (arithmetic complement) of data in a general register. |
| EXTS | W/L | Rd (sign extension) $\rightarrow Rd$ Extends byte data in the lower 8 bits of a 16-bit register to word data, or extends word data in the lower 16 bits of a 32-bit register to longword data, by extending the sign bit. |
| EXTU | W/L | Rd (zero extension) $\rightarrow Rd$ Extends byte data in the lower 8 bits of a 16-bit register to word data, or extends word data in the lower 16 bits of a 32-bit register to longword data, by padding with zeros. |

Note: * Size refers to the operand size.

B: Byte

W: Word

L: Longword

Table 2.5 Logic Operation Instructions

| Instruction | Size* | Function |
|--------------------|--------------|---|
| AND | B/W/L | $Rd \wedge Rs \rightarrow Rd$, $Rd \wedge \#IMM \rightarrow Rd$ Performs a logical AND operation on a general register and another general register or immediate data. |
| OR | B/W/L | $Rd \vee Rs \rightarrow Rd$, $Rd \vee \#IMM \rightarrow Rd$ Performs a logical OR operation on a general register and another general register or immediate data. |
| XOR | B/W/L | $Rd \oplus Rs \rightarrow Rd$, $Rd \oplus \#IMM \rightarrow Rd$ Performs a logical exclusive OR operation on a general register and another general register or immediate data. |
| NOT | B/W/L | $\neg Rd \rightarrow Rd$ Takes the one's complement (logical complement) of general register contents. |

Note: * Size refers to the operand size.

B: Byte

W: Word

L: Longword

Table 2.6 Shift Instructions

| Instruction | Size* | Function |
|--------------------|--------------|---|
| SHAL, SHAR | B/W/L | $Rd \text{ (shift)} \rightarrow Rd$ Performs an arithmetic shift on general register contents. |
| SHLL, SHLR | B/W/L | $Rd \text{ (shift)} \rightarrow Rd$ Performs a logical shift on general register contents. |
| ROTL, ROTR | B/W/L | $Rd \text{ (rotate)} \rightarrow Rd$ Rotates general register contents. |
| ROTXL, ROTXR | B/W/L | $Rd \text{ (rotate)} \rightarrow Rd$ Rotates general register contents, including the carry bit. |

Note: * Size refers to the operand size.

B: Byte

W: Word

L: Longword

Table 2.7 Bit Manipulation Instructions

| Instruction | Size* | Function |
|--------------------|--------------|---|
| BSET | B | $1 \rightarrow (\text{<bit-No.> of <EAd>})$ Sets a specified bit in a general register or memory operand to 1. The bit number is specified by 3-bit immediate data or the lower 3 bits of a general register. |
| BCLR | B | $0 \rightarrow (\text{<bit-No.> of <EAd>})$ Clears a specified bit in a general register or memory operand to 0. The bit number is specified by 3-bit immediate data or the lower 3 bits of a general register. |
| BNOT | B | $\neg (\text{<bit-No.> of <EAd>}) \rightarrow (\text{<bit-No.> of <EAd>})$ Inverts a specified bit in a general register or memory operand. The bit number is specified by 3-bit immediate data or the lower 3 bits of a general register. |
| BTST | B | $\neg (\text{<bit-No.> of <EAd>}) \rightarrow Z$ Tests a specified bit in a general register or memory operand and sets or clears the Z flag accordingly. The bit number is specified by 3-bit immediate data or the lower 3 bits of a general register. |
| BAND | B | $C \wedge (\text{<bit-No.> of <EAd>}) \rightarrow C$ ANDs the carry flag with a specified bit in a general register or memory operand and stores the result in the carry flag. The bit number is specified by 3-bit immediate data. |
| BIAND | B | $C \wedge [\neg (\text{<bit-No.> of <EAd>})] \rightarrow C$ ANDs the carry flag with the inverse of a specified bit in a general register or memory operand and stores the result in the carry flag. The bit number is specified by 3-bit immediate data. |

| Instruction | Size* | Function |
|-------------|-------|--|
| BOR | B | $C \vee (\text{<bit-No.> of <EAd>}) \rightarrow C$ ORs the carry flag with a specified bit in a general register or memory operand and stores the result in the carry flag. The bit number is specified by 3-bit immediate data. |
| BIOR | B | $C \vee [\neg (\text{<bit-No.> of <EAd>})] \rightarrow C$ ORs the carry flag with the inverse of a specified bit in a general register or memory operand and stores the result in the carry flag. The bit number is specified by 3-bit immediate data. |
| BXOR | B | $C \oplus (\text{<bit-No.> of <EAd>}) \rightarrow C$ Exclusive-ORs the carry flag with a specified bit in a general register or memory operand and stores the result in the carry flag. The bit number is specified by 3-bit immediate data. |
| BIXOR | B | $C \oplus [\neg (\text{<bit-No.> of <EAd>})] \rightarrow C$ Exclusive-ORs the carry flag with the inverse of a specified bit in a general register or memory operand and stores the result in the carry flag. The bit number is specified by 3-bit immediate data. |
| BLD | B | $(\text{<bit-No.> of <EAd>}) \rightarrow C$ Transfers a specified bit in a general register or memory operand to the carry flag. The bit number is specified by 3-bit immediate data. |
| BILD | B | $\neg (\text{<bit-No.> of <EAd>}) \rightarrow C$ Transfers the inverse of a specified bit in a general register or memory operand to the carry flag. The bit number is specified by 3-bit immediate data. |
| BST | B | $C \rightarrow (\text{<bit-No.> of <EAd>})$ Transfers the carry flag value to a specified bit in a general register or memory operand. The bit number is specified by 3-bit immediate data. |
| BIST | B | $C \rightarrow \neg (\text{<bit-No.> of <EAd>})$ Transfers the inverse of the carry flag value to a specified bit in a general register or memory operand. The bit number is specified by 3-bit immediate data. |

Note: * Size refers to the operand size.

B: Byte

Table 2.8 Branching Instructions

| Instruction | Size | Function | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|-------------|----------------------------|--|----------|-------------|-----------|----------|---------------|--------|----------|---------------|-------|-----|------|----------------|-----|-------------|----------------|-----------|----------------------------|---------|-----------|-----------------|---------|-----|-----------|---------|-----|-------|---------|-----|----------------|---------|-----|--------------|---------|-----|------|---------|-----|-------|---------|-----|------------------|------------------|-----|-----------|------------------|-----|--------------|---------------------------|-----|---------------|---------------------------|
| Bcc | — | Branches to a specified address if address specified condition is met. The branching conditions are listed below. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | <table border="1"> <thead> <tr> <th>Mnemonic</th> <th>Description</th> <th>Condition</th> </tr> </thead> <tbody> <tr> <td>BRA (BT)</td> <td>Always (true)</td> <td>Always</td> </tr> <tr> <td>BRN (BF)</td> <td>Never (false)</td> <td>Never</td> </tr> <tr> <td>BHI</td> <td>High</td> <td>$C \vee Z = 0$</td> </tr> <tr> <td>BLS</td> <td>Low or same</td> <td>$C \vee Z = 1$</td> </tr> <tr> <td>Bcc (BHS)</td> <td>Carry clear (high or same)</td> <td>$C = 0$</td> </tr> <tr> <td>BCS (BLO)</td> <td>Carry set (low)</td> <td>$C = 1$</td> </tr> <tr> <td>BNE</td> <td>Not equal</td> <td>$Z = 0$</td> </tr> <tr> <td>BEQ</td> <td>Equal</td> <td>$Z = 1$</td> </tr> <tr> <td>BVC</td> <td>Overflow clear</td> <td>$V = 0$</td> </tr> <tr> <td>BVS</td> <td>Overflow set</td> <td>$V = 1$</td> </tr> <tr> <td>BPL</td> <td>Plus</td> <td>$N = 0$</td> </tr> <tr> <td>BMI</td> <td>Minus</td> <td>$N = 1$</td> </tr> <tr> <td>BGE</td> <td>Greater or equal</td> <td>$N \oplus V = 0$</td> </tr> <tr> <td>BLT</td> <td>Less than</td> <td>$N \oplus V = 1$</td> </tr> <tr> <td>BGT</td> <td>Greater than</td> <td>$Z \vee (N \oplus V) = 0$</td> </tr> <tr> <td>BLE</td> <td>Less or equal</td> <td>$Z \vee (N \oplus V) = 1$</td> </tr> </tbody> </table> | Mnemonic | Description | Condition | BRA (BT) | Always (true) | Always | BRN (BF) | Never (false) | Never | BHI | High | $C \vee Z = 0$ | BLS | Low or same | $C \vee Z = 1$ | Bcc (BHS) | Carry clear (high or same) | $C = 0$ | BCS (BLO) | Carry set (low) | $C = 1$ | BNE | Not equal | $Z = 0$ | BEQ | Equal | $Z = 1$ | BVC | Overflow clear | $V = 0$ | BVS | Overflow set | $V = 1$ | BPL | Plus | $N = 0$ | BMI | Minus | $N = 1$ | BGE | Greater or equal | $N \oplus V = 0$ | BLT | Less than | $N \oplus V = 1$ | BGT | Greater than | $Z \vee (N \oplus V) = 0$ | BLE | Less or equal | $Z \vee (N \oplus V) = 1$ |
| Mnemonic | Description | Condition | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| BRA (BT) | Always (true) | Always | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| BRN (BF) | Never (false) | Never | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| BHI | High | $C \vee Z = 0$ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| BLS | Low or same | $C \vee Z = 1$ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Bcc (BHS) | Carry clear (high or same) | $C = 0$ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| BCS (BLO) | Carry set (low) | $C = 1$ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| BNE | Not equal | $Z = 0$ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| BEQ | Equal | $Z = 1$ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| BVC | Overflow clear | $V = 0$ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| BVS | Overflow set | $V = 1$ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| BPL | Plus | $N = 0$ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| BMI | Minus | $N = 1$ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| BGE | Greater or equal | $N \oplus V = 0$ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| BLT | Less than | $N \oplus V = 1$ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| BGT | Greater than | $Z \vee (N \oplus V) = 0$ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| BLE | Less or equal | $Z \vee (N \oplus V) = 1$ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| JMP | — | Branches unconditionally to a specified address | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| BSR | — | Branches to a subroutine at a specified address | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| JSR | — | Branches to a subroutine at a specified address | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| RTS | — | Returns from a subroutine | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Table 2.9 System Control Instructions

| Instruction | Size* | Function |
|--------------------|--------------|--|
| TRAPA | — | Starts trap-instruction exception handling |
| RTE | — | Returns from an exception-handling routine |
| SLEEP | — | Causes a transition to the power-down state |
| LDC | B/W | (EAs) → CCR Moves the source operand contents to the condition code register. The condition code register size is one byte, but in transfer from memory, data is read by word access. |
| STC | B/W | CCR → (EAd) Transfers the CCR contents to a destination location. The condition code register size is one byte, but in transfer to memory, data is written by word access. |
| ANDC | B | CCR ∧ #IMM → CCR Logically ANDs the condition code register with immediate data. |
| ORC | B | CCR ∨ #IMM → CCR Logically ORs the condition code register with immediate data. |
| XORC | B | CCR ⊕ #IMM → CCR Logically exclusive-ORs the condition code register with immediate data. |
| NOP | — | PC + 2 → PC Only increments the program counter. |

Note: * Size refers to the operand size.

B: Byte

W: Word

Table 2.10 Block Transfer Instruction

| Instruction | Size | Function |
|--------------------|-------------|---|
| EEPMOV.B | — | if R4L \neq 0 then repeat @ER5+ \rightarrow @ER6+, R4L - 1 \rightarrow R4L until R4L = 0 else next; |
| EEPMOV.W | — | if R4 \neq 0 then repeat @ER5+ \rightarrow @ER6+, R4 - 1 \rightarrow R4 until R4 = 0 else next; Block transfer instruction. This instruction transfers the number of data bytes specified by R4L or R4, starting from the address indicated by ER5, to the location starting at the address indicated by ER6. At the end of the transfer, the next instruction is executed. |

2.6.4 Basic Instruction Formats

The H8/300H instructions consist of 2-byte (word) units. An instruction consists of an operation field (OP field), a register field (r field), an effective address extension (EA field), and a condition field (cc).

Operation Field: Indicates the function of the instruction, the addressing mode, and the operation to be carried out on the operand. The operation field always includes the first 4 bits of the instruction. Some instructions have two operation fields.

Register Field: Specifies a general register. Address registers are specified by 3 bits, data registers by 3 bits or 4 bits. Some instructions have two register fields. Some have no register field.

Effective Address Extension: Eight, 16, or 32 bits specifying immediate data, an absolute address, or a displacement. A 24-bit address or displacement is treated as 32-bit data in which the first 8 bits are 0 (H'00).

Condition Field: Specifies the branching condition of Bcc instructions.

Figure 2.9 shows examples of instruction formats.

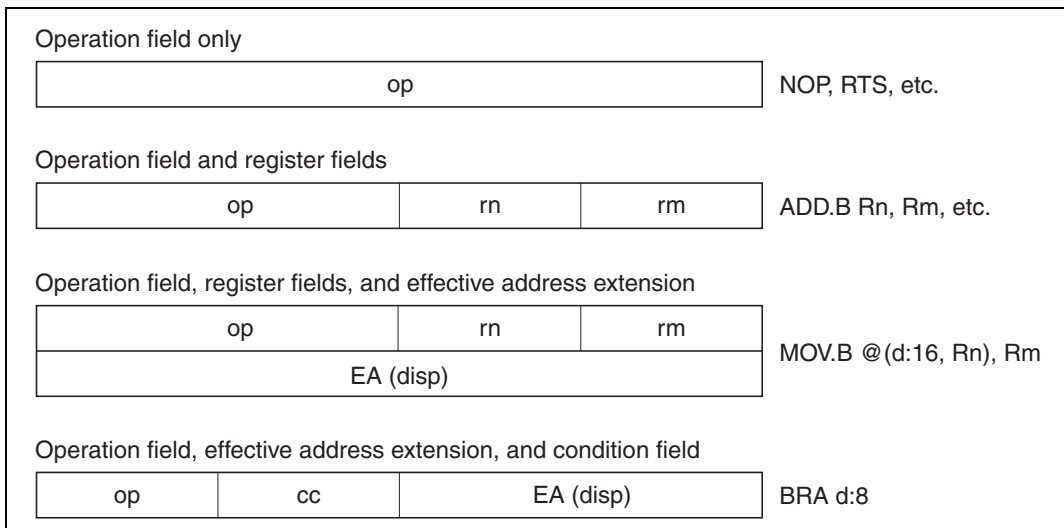


Figure 2.9 Instruction Formats

2.6.5 Notes on Use of Bit Manipulation Instructions

The BSET, BCLR, BNOT, BST, and BIST instructions read a byte of data, modify a bit in the byte, then write the byte back. Care is required when these instructions are used to access registers with write-only bits, or to access ports.

| Step | | Description |
|------|--------|--|
| 1 | Read | Read one data byte at the specified address |
| 2 | Modify | Modify one bit in the data byte |
| 3 | Write | Write the modified data byte back to the specified address |

Example 1: BCLR is executed to clear bit 0 in the port 4 data direction register (P4DDR) under the following conditions.

P4₇, P4₆: Input pins
P4₅ – P4₀: Output pins

The intended purpose of this BCLR instruction is to switch P4₀ from output to input.

Before Execution of BCLR Instruction

| | P4 ₇ | P4 ₆ | P4 ₅ | P4 ₄ | P4 ₃ | P4 ₂ | P4 ₁ | P4 ₀ |
|--------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| Input/output | Input | Input | Output | Output | Output | Output | Output | Output |
| DDR | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |

Execution of BCLR Instruction

BCLR #0, @P4DDR ; Execute BCLR instruction on DDR

After Execution of BCLR Instruction

| | P4 ₇ | P4 ₆ | P4 ₅ | P4 ₄ | P4 ₃ | P4 ₂ | P4 ₁ | P4 ₀ |
|--------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| Input/output | Output | Output | Output | Output | Output | Output | Output | Input |
| DDR | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |

Explanation: To execute the BCLR instruction, the CPU begins by reading P4DDR. Since P4DDR is a write-only register, it is read as H'FF, even though its true value is H'3F.

Next the CPU clears bit 0 of the read data, changing the value to H'FE.

Finally, the CPU writes this value (H'FE) back to P4DDR to complete the BCLR instruction.

As a result, P4₀DDR is cleared to 0, making P4₀ an input pin. In addition, P4₇DDR and P4₆DDR are set to 1, making P4₇ and P4₆ output pins.

The BCLR instruction can be used to clear flags in the on-chip registers to 0. In the case of the IRQ status register (ISR), for example, a flag must be read as a condition for clearing it, but when using the BCLR instruction, if it is known that a flag has been set to 1 in an interrupt-handling routine, for instance, it is not necessary to read the flag ahead of time.

2.7 Addressing Modes and Effective Address Calculation

2.7.1 Addressing Modes

The H8/300H CPU supports the eight addressing modes listed in table 2.11. Each instruction uses a subset of these addressing modes. Arithmetic and logic instructions can use the register direct and immediate modes. Data transfer instructions can use all addressing modes except program-counter relative and memory indirect. Bit manipulation instructions use register direct, register indirect, or absolute (@aa:8) addressing mode to specify an operand, and register direct (BSET, BCLR, BNOT, and BTST instructions) or immediate (3-bit) addressing mode to specify a bit number in the operand.

Table 2.11 Addressing Modes

| No. | Addressing Mode | Symbol |
|-----|---|---------------------------|
| 1 | Register direct | Rn |
| 2 | Register indirect | @ERn |
| 3 | Register indirect with displacement | @(d:16, ERn)/@(d:24, ERn) |
| 4 | Register indirect with post-increment Register indirect with pre-decrement | @ERn+ @-ERn |
| 5 | Absolute address | @aa:8/@aa:16/@aa:24 |
| 6 | Immediate | #xx:8/#xx:16/#xx:32 |
| 7 | Program-counter relative | @(d:8, PC)/@(d:16, PC) |
| 8 | Memory indirect | @@aa:8 |

Register Direct—Rn: The register field of the instruction code specifies an 8-, 16-, or 32-bit register containing the operand. R0H to R7H and R0L to R7L can be specified as 8-bit registers. R0 to R7 and E0 to E7 can be specified as 16-bit registers. ER0 to ER7 can be specified as 32-bit registers.

Register Indirect—@ERn: The register field of the instruction code specifies an address register (ERn), the lower 24 bits of which contain the address of the operand.

Register Indirect with Displacement—@(d:16, ERn) or @(d:24, ERn): A 16-bit or 24-bit displacement contained in the instruction code is added to the contents of an address register (ERn) specified by the register field of the instruction, and the lower 24 bits of the sum specify the address of a memory operand. A 16-bit displacement is sign-extended when added.

Register Indirect with Post-Increment or Pre-Decrement—@ERn+ or @-ERn:

- Register indirect with post-increment—@ERn+
The register field of the instruction code specifies an address register (ERn) the lower 24 bits of which contain the address of a memory operand. After the operand is accessed, 1, 2, or 4 is added to the address register contents (32 bits) and the sum is stored in the address register. The value added is 1 for byte access, 2 for word access, or 4 for longword access. For word or longword access, the register value should be even.
- Register indirect with pre-decrement—@-ERn
The value 1, 2, or 4 is subtracted from an address register (ERn) specified by the register field in the instruction code, and the lower 24 bits of the result become the address of a memory operand. The result is also stored in the address register. The value subtracted is 1 for byte access, 2 for word access, or 4 for longword access. For word or longword access, the resulting register value should be even.

Absolute Address—@aa:8, @aa:16, or @aa:24: The instruction code contains the absolute address of a memory operand. The absolute address may be 8 bits long (@aa:8), 16 bits long (@aa:16), or 24 bits long (@aa:24). For an 8-bit absolute address, the upper 16 bits are all assumed to be 1 (H'FFFF). For a 16-bit absolute address the upper 8 bits are a sign extension. A 24-bit absolute address can access the entire address space. Table 2.12 indicates the accessible address ranges.

Table 2.12 Absolute Address Access Ranges

| Absolute Address | 1-Mbyte Modes | 16-Mbyte Modes |
|-------------------------|--|---|
| 8 bits (@aa:8) | H'FFF00 to H'FFFFF (1048320 to 1048575) | H'FFFF00 to H'FFFFFF (16776960 to 16777215) |
| 16 bits (@aa:16) | H'00000 to H'07FFF, H'F8000 to H'FFFFFF (0 to 32767, 1015808 to 1048575) | H'000000 to H'007FFF, H'FF8000 to H'FFFFFF (0 to 32767, 16744448 to 16777215) |
| 24 bits (@aa:24) | H'00000 to H'FFFFFF (0 to 1048575) | H'000000 to H'FFFFFF (0 to 16777215) |

Immediate—#xx:8, #xx:16, or #xx:32: The instruction code contains 8-bit (#xx:8), 16-bit (#xx:16), or 32-bit (#xx:32) immediate data as an operand.

The instruction codes of the ADDS, SUBS, INC, and DEC instructions contain immediate data implicitly. The instruction codes of some bit manipulation instructions contain 3-bit immediate data specifying a bit number. The TRAPA instruction code contains 2-bit immediate data specifying a vector address.

Program-Counter Relative—@(d:8**, PC) or @(**d:16**, PC):** This mode is used in the Bcc and BSR instructions. An 8-bit or 16-bit displacement contained in the instruction code is sign-extended to 24 bits and added to the 24-bit PC contents to generate a 24-bit branch address. The PC value to which the displacement is added is the address of the first byte of the next instruction, so the possible branching range is -126 to $+128$ bytes (-63 to $+64$ words) or -32766 to $+32768$ bytes (-16383 to $+16384$ words) from the branch instruction. The resulting value should be an even number.

Memory Indirect—@aa:8**:** This mode can be used by the JMP and JSR instructions. The instruction code contains an 8-bit absolute address specifying a memory operand. This memory operand contains a branch address. The memory operand is accessed by longword access. The first byte of the memory operand is ignored, generating a 24-bit branch address. See figure 2.10. The upper bits of the 8-bit absolute address are assumed to be 0 (H'0000), so the address range is 0 to 255 (H'000000 to H'0000FF). Note that the first part of this range is also the exception vector area. For further details see section 5, Interrupt Controller.

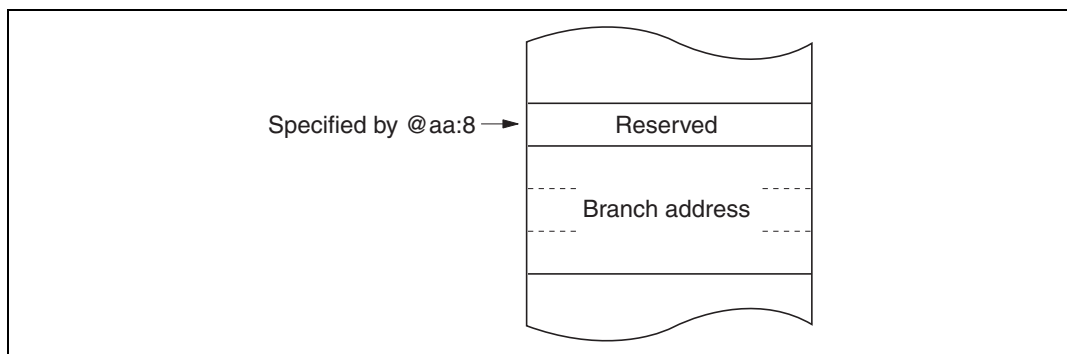


Figure 2.10 Memory-Indirect Branch Address Specification

When a word-size or longword-size memory operand is specified, or when a branch address is specified, if the specified memory address is odd, the least significant bit is regarded as 0. The accessed data or instruction code therefore begins at the preceding address. See section 2.5.2, Memory Data Formats.

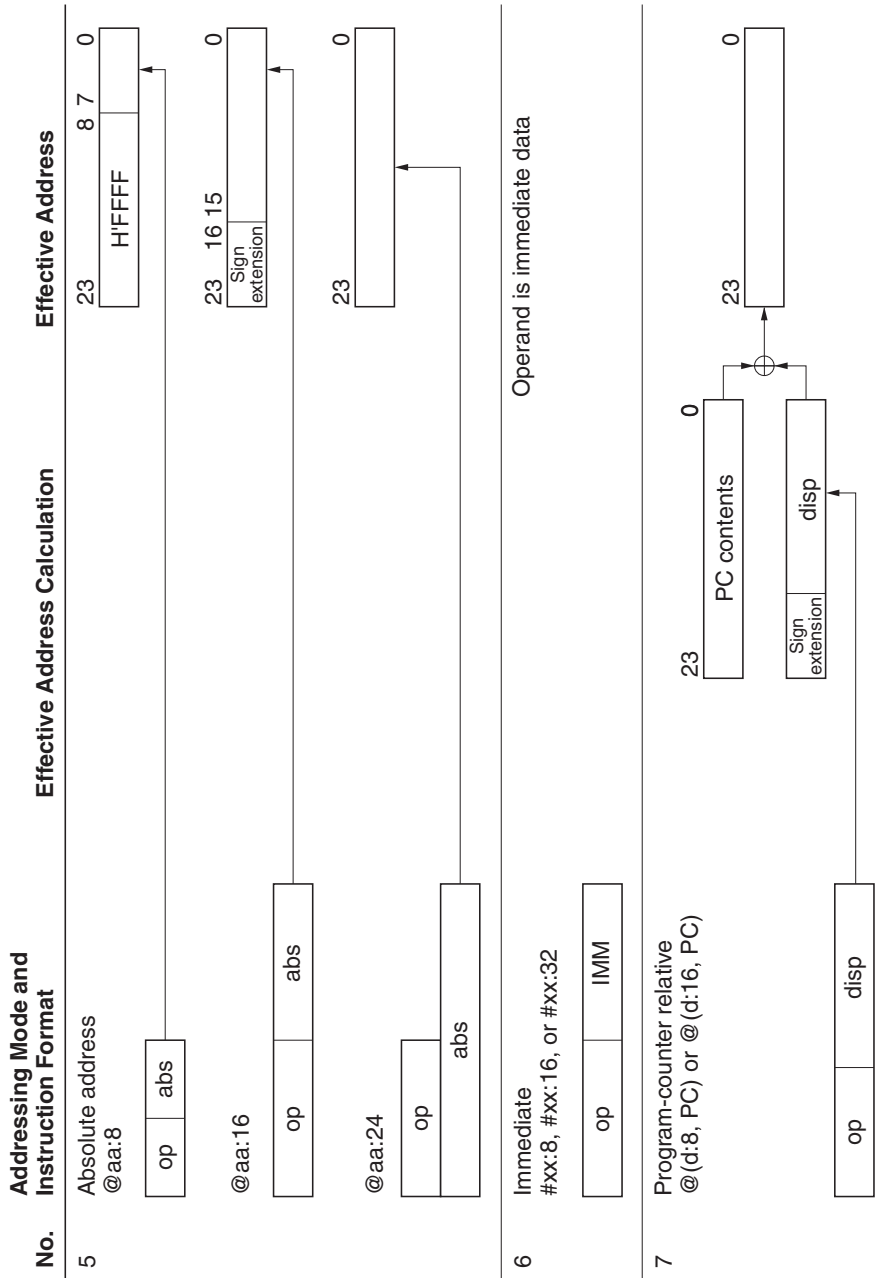
2.7.2 Effective Address Calculation

Table 2.13 explains how an effective address is calculated in each addressing mode. In the 1-Mbyte operating modes the upper 4 bits of the calculated address are ignored in order to generate a 20-bit effective address.

Table 2.13 Effective Address Calculation

| No. | Addressing Mode and Instruction Format | Effective Address Calculation | Effective Address |
|-----|---|-------------------------------|--------------------------------------|
| 1 | Register direct (Rn) op rm rn | | Operand is general register contents |
| 2 | Register indirect (@ERn) op r | | |
| 3 | Register indirect with displacement @ (d:16, ERn)/@ (d:24, ERn) op r disp | | |
| 4 | Register indirect with post-increment or pre-decrement Register indirect with post-increment @ERn+ Register indirect with pre-decrement @-ERn | | |

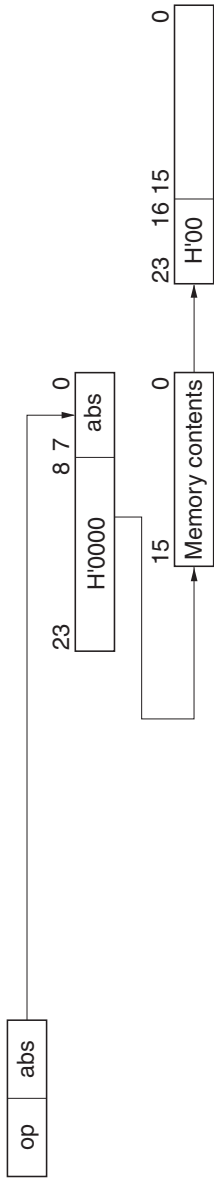
1 for a byte operand,
2 for a word operand,
4 for a longword operand



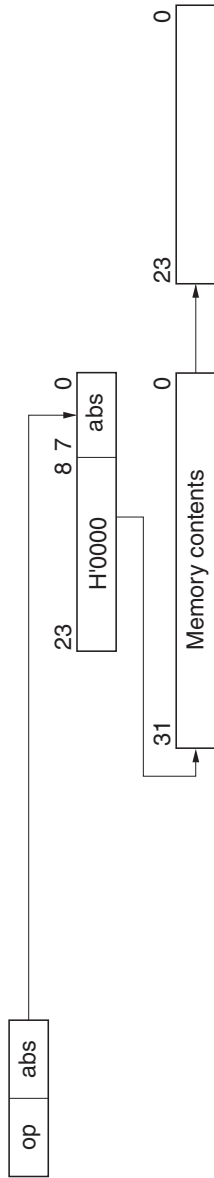
Addressing Mode and Instruction Format **Effective Address Calculation** **Effective Address**

8 Memory indirect @aa:8

Normal mode



Advanced mode



Legend:

- r, rm, rn: Register field
- op: Operation field
- disp: Displacement
- IMM: Immediate data
- abs: Absolute address

2.8 Processing States

2.8.1 Overview

The H8/300H CPU has five processing states: the program execution state, exception-handling state, power-down state, reset state, and bus-released state. The power-down state includes sleep mode, software standby mode, and hardware standby mode. Figure 2.11 classifies the processing states. Figure 2.13 indicates the state transitions.

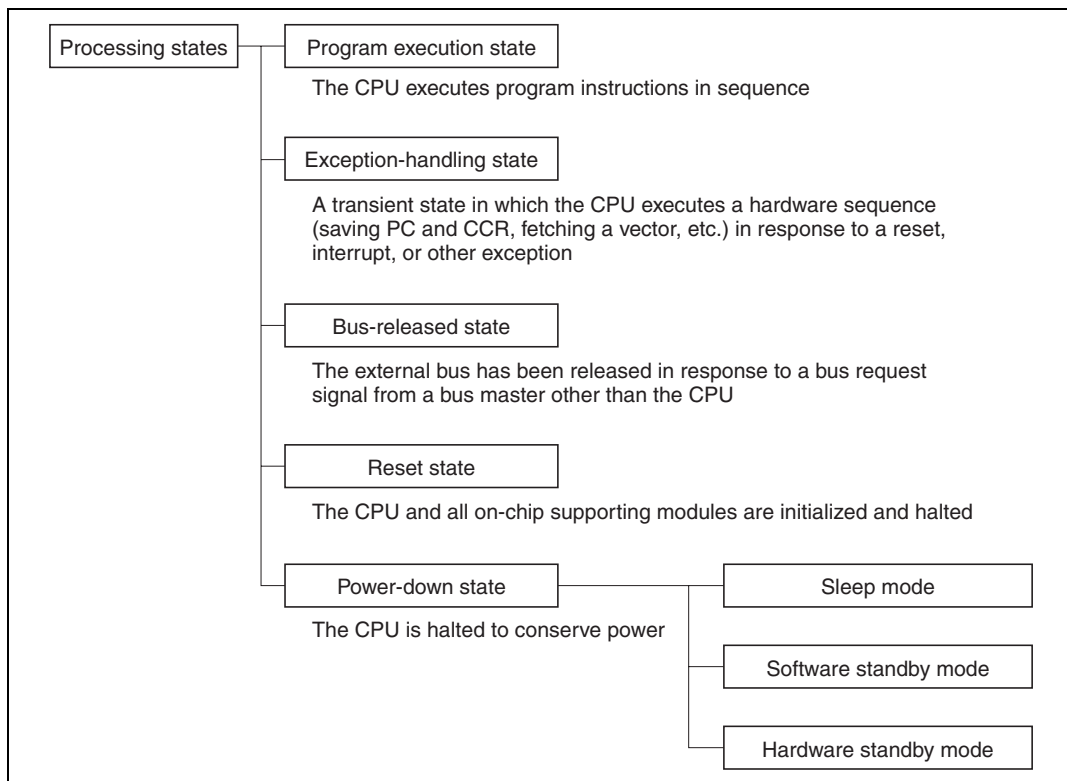


Figure 2.11 Processing States

2.8.2 Program Execution State

In this state the CPU executes program instructions in normal sequence.

2.8.3 Exception-Handling State

The exception-handling state is a transient state that occurs when the CPU alters the normal program flow due to a reset, interrupt, or trap instruction. The CPU fetches a starting address from the exception vector table and branches to that address. In interrupt and trap exception handling the CPU references the stack pointer (ER7) and saves the program counter and condition code register.

Types of Exception Handling and Their Priority: Exception handling is performed for resets, interrupts, and trap instructions. Table 2.14 indicates the types of exception handling and their priority. Trap instruction exceptions are accepted at all times in the program execution state.

Table 2.14 Exception Handling Types and Priority

| Priority | Type of Exception | Detection Timing | Start of Exception Handling |
|------------------|-------------------|--|--|
| High ↑ Low | Reset | Synchronized with clock | Exception handling starts immediately when $\overline{\text{RES}}$ changes from low to high |
| | Interrupt | End of instruction execution or end of exception handling* | When an interrupt is requested, exception handling starts at the end of the current instruction or current exception-handling sequence |
| | Trap instruction | When TRAPA instruction is executed | Exception handling starts when a trap (TRAPA) instruction is executed |

Note: * Interrupts are not detected at the end of the ANDC, ORC, XORC, and LDC instructions, or immediately after reset exception handling.

Figure 2.12 classifies the exception sources. For further details about exception sources, vector numbers, and vector addresses, see section 4, Exception Handling, and section 5, Interrupt Controller.

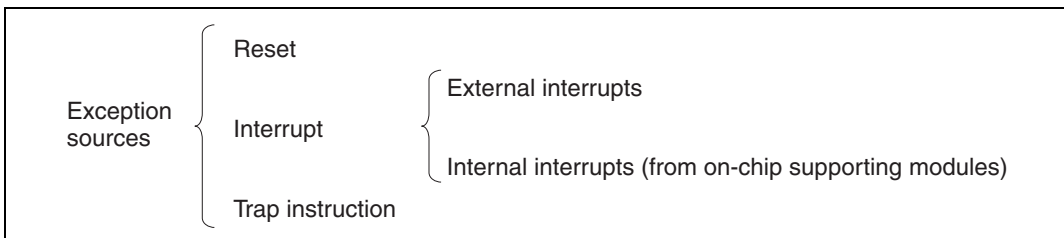


Figure 2.12 Classification of Exception Sources

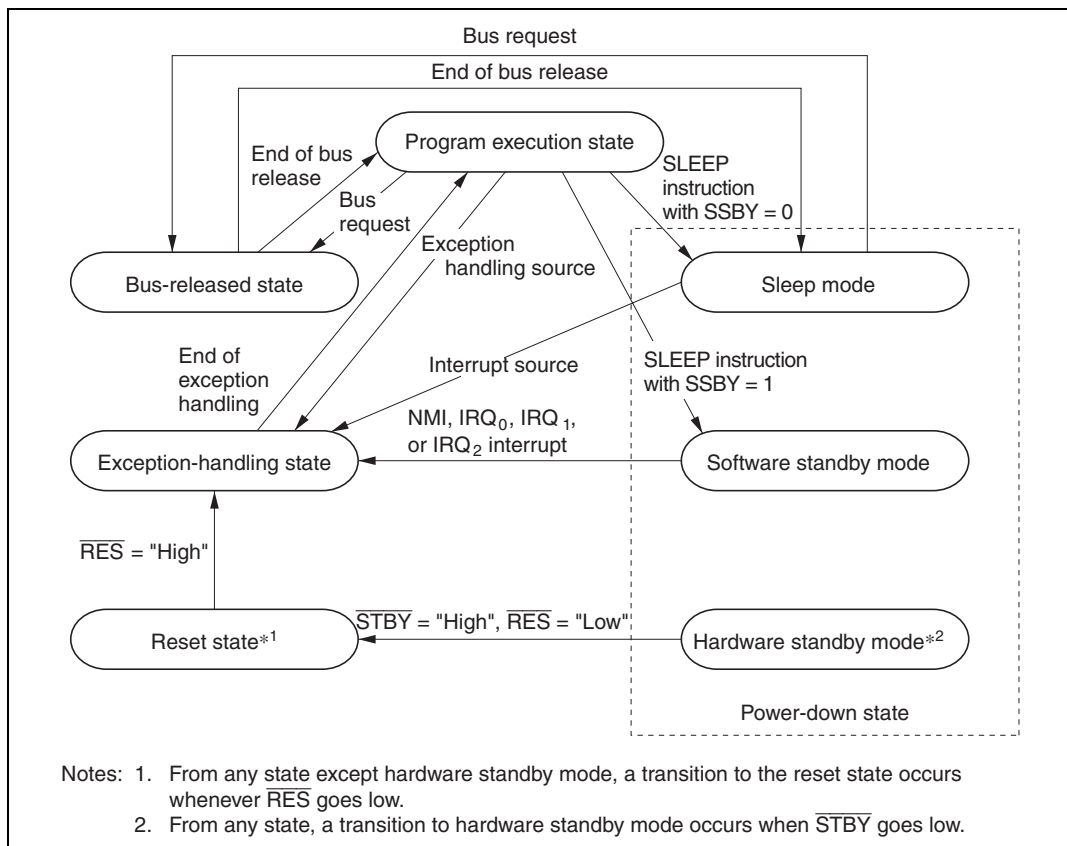


Figure 2.13 State Transitions

2.8.4 Exception Handling Operation

Reset Exception Handling: Reset exception handling has the highest priority. The reset state is entered when the \overline{RES} signal goes low. Reset exception handling starts after that, when \overline{RES} changes from low to high. When reset exception handling starts the CPU fetches a start address from the exception vector table and starts program execution from that address. All interrupts, including NMI, are disabled during the reset exception-handling sequence and immediately after it ends.

Interrupt Exception Handling and Trap Instruction Exception Handling: When these exception-handling sequences begin, the CPU references the stack pointer (ER7) and pushes the program counter and condition code register on the stack. Next, if the UE bit in the system control register (SYSCR) is set to 1, the CPU sets the I bit in the condition code register to 1. If the UE bit is cleared to 0, the CPU sets both the I bit and the UI bit in the condition code register to 1. Then

the CPU fetches a start address from the exception vector table and execution branches to that address.

Figure 2.14 shows the stack after the exception-handling sequence.

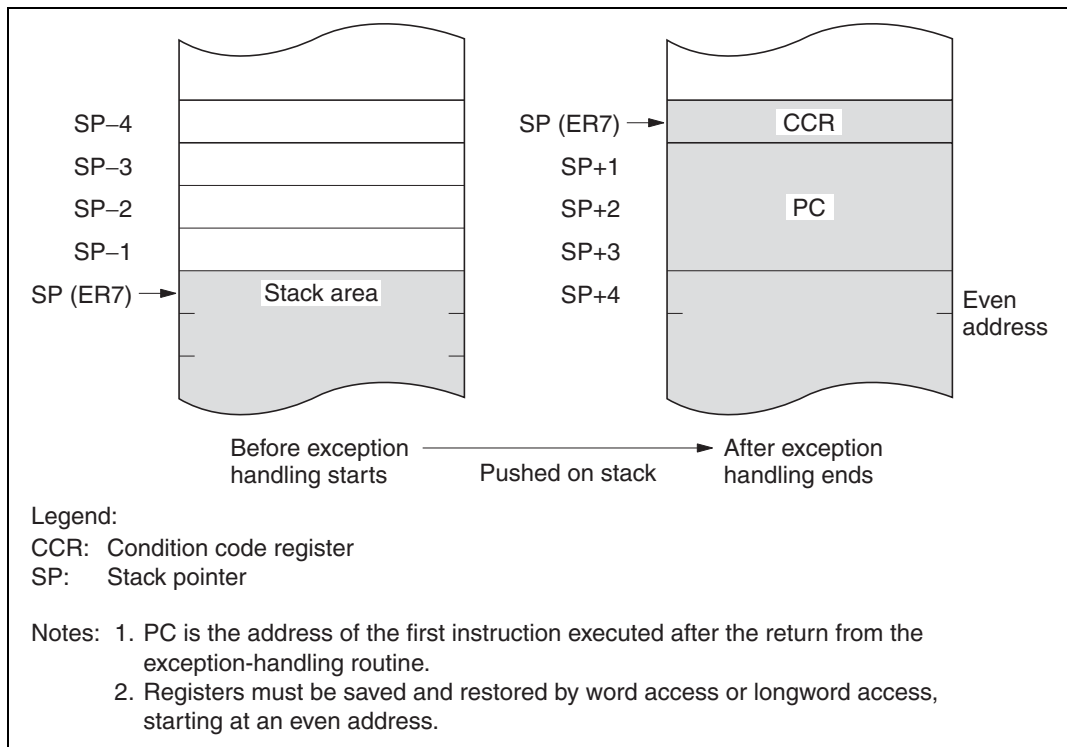


Figure 2.14 Stack Structure after Exception Handling

2.8.5 Bus-Released State

In this state the bus is released to a bus master other than the CPU, in response to a bus request. The bus masters other than the CPU is an external bus master. While the bus is released, the CPU halts except for internal operations. Interrupt requests are not accepted. For details see section 6.6, Bus Arbiter.

2.8.6 Reset State

When the $\overline{\text{RES}}$ input goes low all current processing stops and the CPU enters the reset state. The I bit in the condition code register is set to 1 by a reset. All interrupts are masked in the reset state. Reset exception handling starts when the $\overline{\text{RES}}$ signal changes from low to high.

The reset state can also be entered by a watchdog timer overflow. For details see section 11, Watchdog Timer.

2.8.7 Power-Down State

In the power-down state the CPU stops operating to conserve power. There are three modes: sleep mode, software standby mode, and hardware standby mode.

Sleep Mode: A transition to sleep mode is made if the SLEEP instruction is executed while the SSBY bit is cleared to 0 in the system control register (SYSCR). CPU operations stop immediately after execution of the SLEEP instruction, but the contents of CPU registers are retained.

Software Standby Mode: A transition to software standby mode is made if the SLEEP instruction is executed while the SSBY bit is set to 1 in SYSCR. The CPU and clock halt and all on-chip supporting modules stop operating. The on-chip supporting modules are reset, but as long as a specified voltage is supplied the contents of CPU registers and on-chip RAM are retained. The I/O ports also remain in their existing states.

Hardware Standby Mode: A transition to hardware standby mode is made when the $\overline{\text{STBY}}$ input goes low. As in software standby mode, the CPU and all clocks halt and the on-chip supporting modules are reset, but as long as a specified voltage is supplied, on-chip RAM contents are retained.

For further information see section 18, Power-Down State.

2.9 Basic Operational Timing

2.9.1 Overview

The H8/300H CPU operates according to the system clock (ϕ). The interval from one rise of the system clock to the next rise is referred to as a “state.” A memory cycle or bus cycle consists of two or three states. The CPU uses different methods to access on-chip memory, the on-chip supporting modules, and the external address space. Access to the external address space can be controlled by the bus controller.

2.9.2 On-Chip Memory Access Timing

On-chip memory is accessed in two states. The data bus is 16 bits wide, permitting both byte and word access. Figure 2.15 shows the on-chip memory access cycle. Figure 2.16 indicates the pin states. The H8/3008 has a function for changing the method of outputting addresses from the address pins. For details see section 6.3.5, Address Output Method.

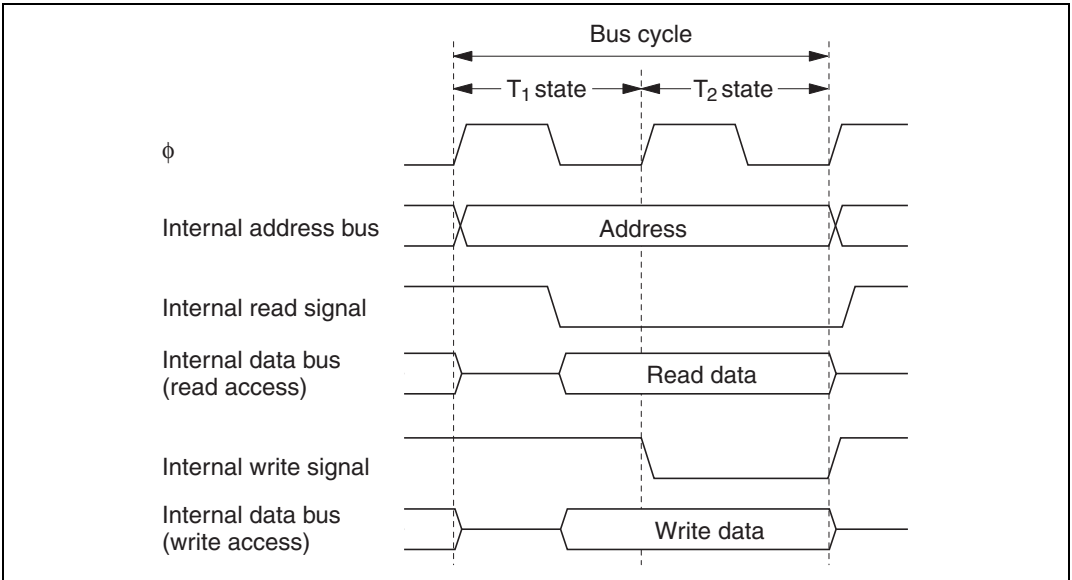


Figure 2.15 On-Chip Memory Access Cycle

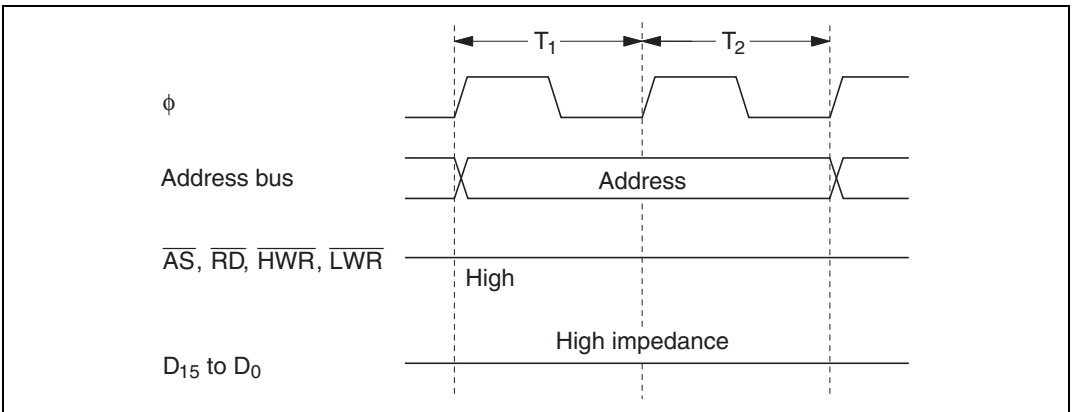


Figure 2.16 Pin States during On-Chip Memory Access (Address Update Mode 1)

2.9.3 On-Chip Supporting Module Access Timing

The on-chip supporting modules are accessed in three states. The data bus is 8 or 16 bits wide, depending on the internal I/O register being accessed. Figure 2.17 shows the on-chip supporting module access timing. Figure 2.18 indicates the pin states.

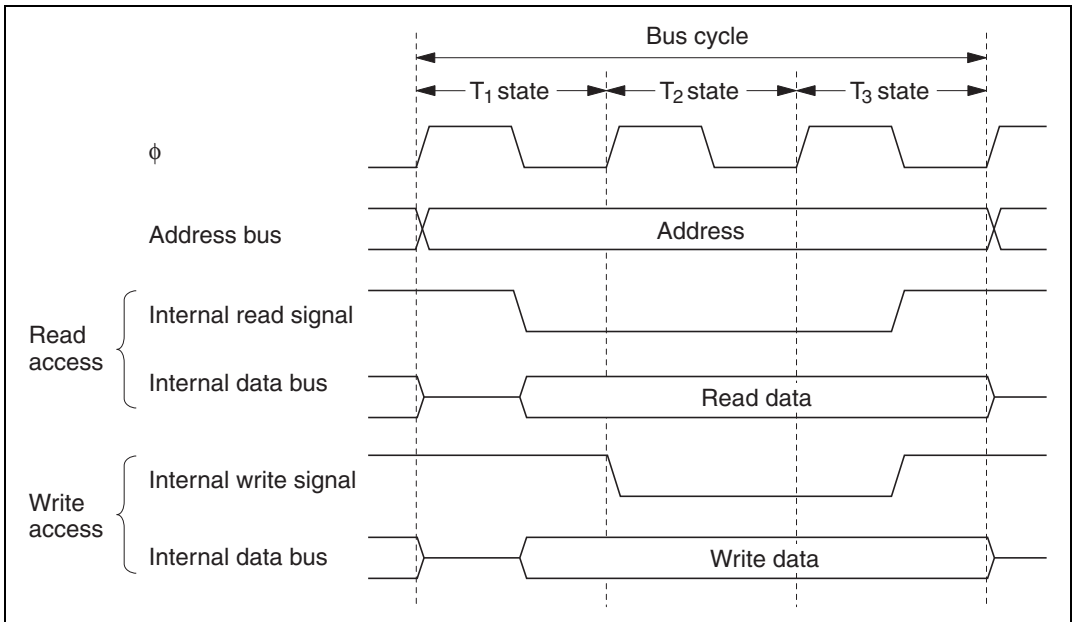


Figure 2.17 Access Cycle for On-Chip Supporting Modules

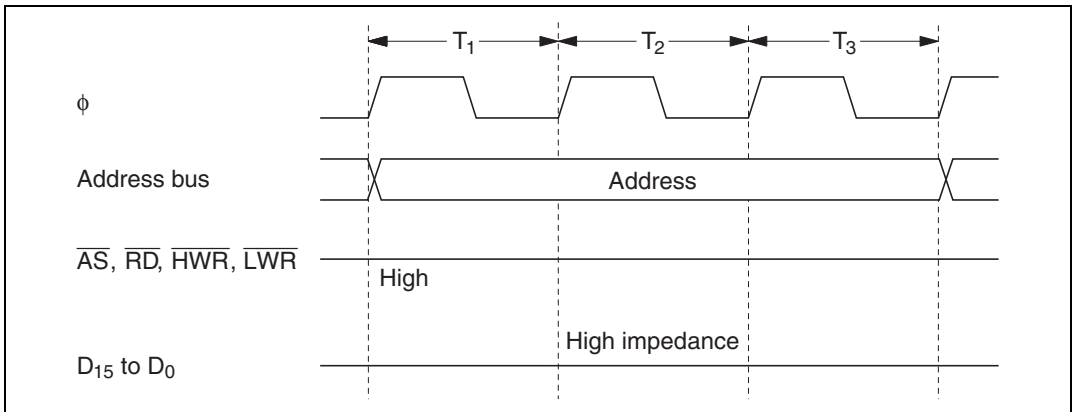


Figure 2.18 Pin States during Access to On-Chip Supporting Modules

2.9.4 Access to External Address Space

The external address space is divided into eight areas (areas 0 to 7). Bus-controller settings determine whether each area is accessed via an 8-bit or 16-bit data bus, and whether it is accessed in two or three states. For details see section 6, Bus Controller.

Section 3 MCU Operating Modes

3.1 Overview

3.1.1 Operating Mode Selection

The H8/3008 has four operating modes (modes 1 to 4) that are selected by the mode pins (MD_2 to MD_0) as indicated in table 3.1. The input at these pins determines the size of the address space and the initial bus mode.

Table 3.1 Operating Mode Selection

| Operating Mode | Mode Pins | | | Description | | | |
|----------------|-----------|--------|--------|--------------------|--------------------------------|--------------------|-----------------------|
| | MD_2 | MD_1 | MD_0 | Address Space | Initial Bus Mode* ¹ | On-Chip ROM | On-Chip RAM |
| — | 0 | 0 | 0 | Setting prohibited | Setting prohibited | Setting prohibited | Setting prohibited |
| Mode 1 | 0 | 0 | 1 | Expanded mode | 8 bits | Disabled | Enabled* ² |
| Mode 2 | 0 | 1 | 0 | Expanded mode | 16 bits | Disabled | Enabled* ² |
| Mode 3 | 0 | 1 | 1 | Expanded mode | 8 bits | Disabled | Enabled* ² |
| Mode 4 | 1 | 0 | 0 | Expanded mode | 16 bits | Disabled | Enabled* ² |
| — | 1 | 0 | 1 | Setting prohibited | Setting prohibited | Setting prohibited | Setting prohibited |
| — | 1 | 1 | 0 | Setting prohibited | Setting prohibited | Setting prohibited | Setting prohibited |
| — | 1 | 1 | 1 | Setting prohibited | Setting prohibited | Setting prohibited | Setting prohibited |

Notes: 1. In modes 1 to 4, an 8-bit or 16-bit data bus can be selected on a per-area basis by settings made in the area bus width control register (ABWCR). For details see section 6, Bus Controller.

2. If the RAME bit in SYSCR is cleared to 0, these addresses become external addresses.

For the address space size there are two choices: 1 Mbyte or 16 Mbyte. The external data bus is either 8 or 16 bits wide depending on ABWCR settings. 8-bit bus mode is used only if 8-bit access is selected for all areas. For details see section 6, Bus Controller.

Modes 1 to 4 are externally expanded modes that enable access to external memory and peripheral devices and disable access to the on-chip ROM. Modes 1 and 2 support a maximum address space of 1 Mbyte. Modes 3 and 4 support a maximum address space of 16 Mbytes.

The H8/3008 can be used only in modes 1 to 4. The inputs at the mode pins must select one of these four modes. The inputs at the mode pins must not be changed during operation. Set the reset state before changing the inputs at these pins.

3.1.2 Register Configuration

The H8/3008 has a mode control register (MDCR) that indicates the inputs at the mode pins (MD₂ to MD₀), and a system control register (SYSCR). Table 3.2 summarizes these registers.

Table 3.2 Registers

| Address* | Name | Abbreviation | R/W | Initial Value |
|----------|-------------------------|--------------|-----|---------------|
| H'EE011 | Mode control register | MDCR | R | Undetermined |
| H'EE012 | System control register | SYSCR | R/W | H'09 |

Note: * Lower 20 bits of the address in advanced mode.

3.2 Mode Control Register (MDCR)

MDCR is an 8-bit read-only register that indicates the current operating mode of the H8/3008.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|---|---|---|---|---|------|------|------|
| | — | — | — | — | — | MDS2 | MDS1 | MDS0 |
| Initial value | 1 | 1 | 0 | 0 | 0 | —* | —* | —* |
| Read/Write | — | — | — | — | — | R | R | R |

Reserved bits
Mode select 2 to 0
Bits indicating the current operating mode

Note: * Determined by pins MD₂ to MD₀.

Bits 7 and 6—Reserved: These bits can not be modified and are always read as 1.

Bits 5 to 3—Reserved: These bits can not be modified and are always read as 0.

Bits 2 to 0—Mode Select 2 to 0 (MDS2 to MDS0): These bits indicate the logic levels at pins MD₂ to MD₀ (the current operating mode). MDS2 to MDS0 correspond to MD₂ to MD₀. MDS2 to

MDS0 are read-only bits. The mode pin (MD₂ to MD₀) levels are latched into these bits when MDCR is read.

3.3 System Control Register (SYSCR)

SYSCR is an 8-bit register that controls the operation of the H8/3008.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|------|------|------|------|-----|-------|------|------|
| | SSBY | STS2 | STS1 | STS0 | UE | NMIEG | SSOE | RAME |
| Initial value | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

RAM enable
Enables or disables on-chip RAM

Software standby output port enable
Selects the output state of the address bus and bus control signals in software standby mode

NMI edge select
Selects the valid edge of the NMI input

User bit enable
Selects whether to use the UI bit in CCR as a user bit or an interrupt mask bit

Standby timer select 2 to 0
These bits select the waiting time at recovery from software standby mode

Software standby
Enables transition to software standby mode

Bit 7—Software Standby (SSBY): Enables transition to software standby mode. (For further information about software standby mode see section 18, Power-Down State.)

When software standby mode is exited by an external interrupt, and a transition is made to normal operation, this bit remains set to 1. To clear this bit, write 0.

Bit 7

| SSBY | Description | |
|-------------|--|-----------------|
| 0 | SLEEP instruction causes transition to sleep mode | (Initial value) |
| 1 | SLEEP instruction causes transition to software standby mode | |

Bits 6 to 4—Standby Timer Select 2 to 0 (STS2 to STS0): These bits select the length of time the CPU and on-chip supporting modules wait for the internal clock oscillator to settle when software standby mode is exited by an external interrupt.

When using a crystal oscillator, set these bits so that the waiting time will be at least 7 ms at the system clock rate.

For further information about waiting time selection, see section 18.4.3, Selection of Waiting Time for Exit from Software Standby Mode.

| Bit 6 STS2 | Bit 5 STS1 | Bit 4 STS0 | Description | |
|-----------------------|-----------------------|-----------------------|-------------------------------|-----------------|
| 0 | 0 | 0 | Waiting time = 8,192 states | (Initial value) |
| 0 | 0 | 1 | Waiting time = 16,384 states | |
| 0 | 1 | 0 | Waiting time = 32,768 states | |
| 0 | 1 | 1 | Waiting time = 65,536 states | |
| 1 | 0 | 0 | Waiting time = 131,072 states | |
| 1 | 0 | 1 | Waiting time = 262,144 states | |
| 1 | 1 | 0 | Waiting time = 1,024 states | |
| 1 | 1 | 1 | Illegal setting | |

Bit 3—User Bit Enable (UE): Selects whether to use the UI bit in the condition code register as a user bit or an interrupt mask bit.

| Bit 3 UE | Description |
|-------------|---|
| 0 | UI bit in CCR is used as an interrupt mask bit |
| 1 | UI bit in CCR is used as a user bit (Initial value) |

Bit 2—NMI Edge Select (NMIEG): Selects the valid edge of the NMI input.

| Bit 2 NMIEG | Description |
|----------------|--|
| 0 | An interrupt is requested at the falling edge of NMI (Initial value) |
| 1 | An interrupt is requested at the rising edge of NMI |

Bit 1—Software Standby Output Port Enable (SSOE): Specifies whether the address bus and bus control signals (\overline{CS}_0 to \overline{CS}_7 , \overline{AS} , \overline{RD} , \overline{HWR} , \overline{LWR}) are kept as outputs or fixed high, or placed in the high-impedance state in software standby mode.

| Bit 1 SSOE | Description |
|---------------|---|
| 0 | In software standby mode, the address bus and bus control signals are all high-impedance (Initial value) |
| 1 | In software standby mode, the address bus retains its output state and bus control signals are fixed high |

Bit 0—RAM Enable (RAME): Enables or disables the on-chip RAM. The RAME bit is initialized by the rising edge of the \overline{RES} signal. It is not initialized in software standby mode.

| Bit 0 RAME | Description |
|---------------|--|
| 0 | On-chip RAM is disabled |
| 1 | On-chip RAM is enabled (Initial value) |

3.4 Operating Mode Descriptions

3.4.1 Mode 1

Ports 1, 2, and 5 function as address pins A_{19} to A_0 , permitting access to a maximum 1-Mbyte address space. The initial bus mode after a reset is 8 bits, with 8-bit access to all areas. If at least one area is designated for 16-bit access in ABWCR, the bus mode switches to 16 bits.

3.4.2 Mode 2

Ports 1, 2, and 5 function as address pins A_{19} to A_0 , permitting access to a maximum 1-Mbyte address space. The initial bus mode after a reset is 16 bits, with 16-bit access to all areas. If all areas are designated for 8-bit access in ABWCR, the bus mode switches to 8 bits.

3.4.3 Mode 3

Ports 1, 2, and 5 and part of port A function as address pins A_{23} to A_0 , permitting access to a maximum 16-Mbyte address space. The initial bus mode after a reset is 8 bits, with 8-bit access to all areas. If at least one area is designated for 16-bit access in ABWCR, the bus mode switches to 16 bits. A_{23} to A_{21} are valid when 0 is written in bits 7 to 5 of the bus release control register (BRCCR). (In this mode A_{20} is always used for address output.)

3.4.4 Mode 4

Ports 1, 2, and 5 and part of port A function as address pins A_{23} to A_0 , permitting access to a maximum 16-Mbyte address space. The initial bus mode after a reset is 16 bits, with 16-bit access to all areas. If all areas are designated for 8-bit access in ABWCR, the bus mode switches to 8 bits. A_{23} to A_{21} are valid when 0 is written in bits 7 to 5 of BRCCR. (In this mode A_{20} is always used for address output.)

3.4.5 Modes 5 to 7

These modes cannot be used in the H8/3008. Pin settings must not be made for these modes.

3.5 Pin Functions in Each Operating Mode

The pin functions of ports 1 to 5 and port A vary depending on the operating mode. Table 3.3 indicates their functions in each operating mode.

Table 3.3 Pin Functions in Each Mode

| Port | Mode 1 | Mode 2 | Mode 3 | Mode 4 |
|--------|---------------------------------|-------------------------------|--|--|
| Port 1 | A_7 to A_0 | A_7 to A_0 | A_7 to A_0 | A_7 to A_0 |
| Port 2 | A_{15} to A_8 | A_{15} to A_8 | A_{15} to A_8 | A_{15} to A_8 |
| Port 3 | D_{15} to D_8 | D_{15} to D_8 | D_{15} to D_8 | D_{15} to D_8 |
| Port 4 | $P4_7$ to $P4_0$ * ¹ | D_7 to D_0 * ¹ | $P4_7$ to $P4_0$ * ¹ | D_7 to D_0 * ¹ |
| Port 5 | A_{19} to A_{16} | A_{19} to A_{16} | A_{19} to A_{16} | A_{19} to A_{16} |
| Port A | PA_7 to PA_4 | PA_7 to PA_4 | PA_6 to PA_4 , A_{20} * ² | PA_6 to PA_4 , A_{20} * ² |

- Notes:
1. Initial state. The bus mode can be switched by settings in ABWCR. These pins function as $P4_7$ to $P4_0$ in 8-bit bus mode, and as D_7 to D_0 in 16-bit bus mode.
 2. Initial state. A_{20} is always an address output pin. PA_6 to PA_4 are switched over to A_{23} to A_{21} output by writing 0 in bits 7 to 5 of BRCCR.

3.6 Memory Map in Each Operating Mode

Figure 3.1 shows memory map of the H8/3008. In the expanded modes, the address space is divided into eight areas.

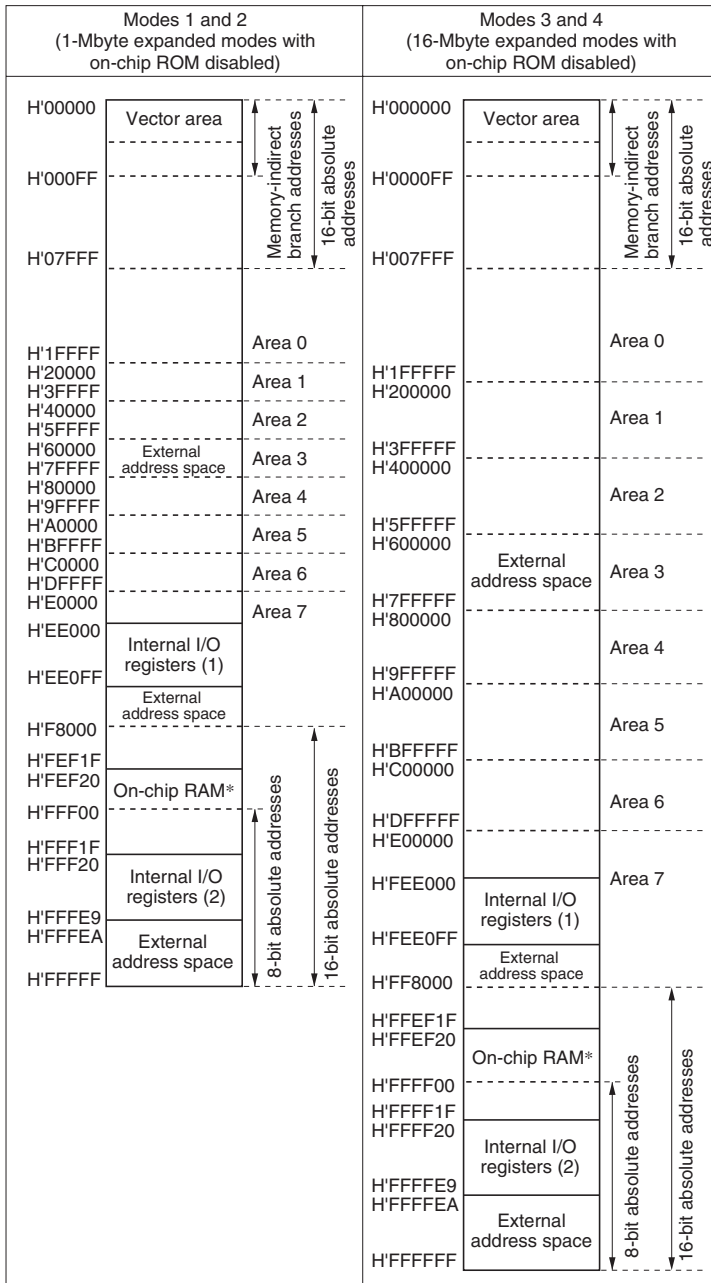
The initial bus mode differs between modes 1 and 2, and also between modes 3 and 4.

The address locations of the on-chip RAM and on-chip registers differ between the 1-Mbyte modes (modes 1 and 2) and the 16-Mbyte modes (modes 3 and 4). The address range specifiable by the CPU in the 8- and 16-bit absolute addressing modes (@aa:8 and @aa:16) also differs.

3.6.1 Reserved Areas

The H8/3008 memory map includes reserved areas to which access (reading or writing) is prohibited. Normal operation cannot be guaranteed if the following reserved areas are accessed.

Reserved Area in Internal I/O Register Space: The H8/3008 internal I/O register space includes a reserved area to which access is prohibited. For details see Appendix B, Internal I/O Registers.



Note: * External addresses can be accessed by disabling on-chip RAM.

Figure 3.1 Memory Map of H8/3008 in Each Operating Mode

Section 4 Exception Handling

4.1 Overview

4.1.1 Exception Handling Types and Priority

As table 4.1 indicates, exception handling may be caused by a reset, interrupt, or trap instruction. Exception handling is prioritized as shown in table 4.1. If two or more exceptions occur simultaneously, they are accepted and processed in priority order. Trap instruction exceptions are accepted at all times in the program execution state.

Table 4.1 Exception Types and Priority

| Priority | Exception Type | Start of Exception Handling |
|----------|--------------------------|--|
| High | Reset | Starts immediately after a low-to-high transition at the $\overline{\text{RES}}$ pin |
| ↑ | Interrupt | Interrupt requests are handled when execution of the current instruction or handling of the current exception is completed |
| | Trap instruction (TRAPA) | Started by execution of a trap instruction (TRAPA) |
| Low | | |

4.1.2 Exception Handling Operation

Exceptions originate from various sources. Trap instructions and interrupts are handled as follows.

1. The program counter (PC) and condition code register (CCR) are pushed onto the stack.
2. The CCR interrupt mask bit is set to 1.
3. A vector address corresponding to the exception source is generated, and program execution starts from that address.

Note: For a reset exception, steps 2 and 3 above are carried out.

4.1.3 Exception Vector Table

The exception sources are classified as shown in figure 4.1. Different vectors are assigned to different exception sources. Table 4.2 lists the exception sources and their vector addresses.

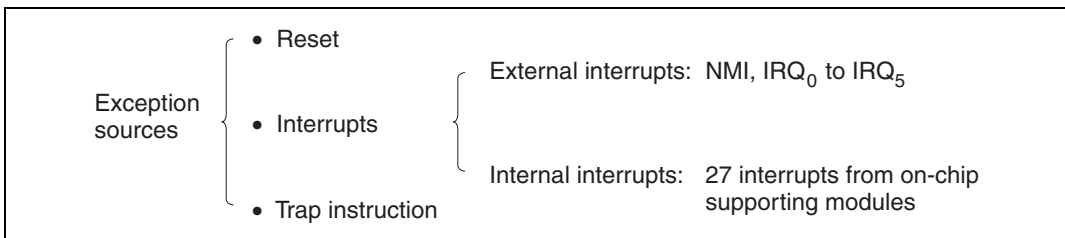


Figure 4.1 Exception Sources

Table 4.2 Exception Vector Table

| Exception Source | Vector Number | Vector Address* ¹ | |
|-------------------------------------|--------------------------|------------------------------|------------------|
| | | Advanced Mode | Normal Mode |
| Reset | 0 | H'0000 to H'0003 | H'0000 to H'0001 |
| Reserved for system use | 1 | H'0004 to H'0007 | H'0002 to H'0003 |
| | 2 | H'0008 to H'000B | H'0004 to H'0005 |
| | 3 | H'000C to H'000F | H'0006 to H'0007 |
| | 4 | H'0010 to H'0013 | H'0008 to H'0009 |
| | 5 | H'0014 to H'0017 | H'000A to H'000B |
| | 6 | H'0018 to H'001B | H'000C to H'000D |
| | External interrupt (NMI) | 7 | H'001C to H'001F |
| Trap instruction (4 sources) | 8 | H'0020 to H'0023 | H'0010 to H'0011 |
| | 9 | H'0024 to H'0027 | H'0012 to H'0013 |
| | 10 | H'0028 to H'002B | H'0014 to H'0015 |
| | 11 | H'002C to H'002F | H'0016 to H'0017 |
| External interrupt IRQ ₀ | 12 | H'0030 to H'0033 | H'0018 to H'0019 |
| External interrupt IRQ ₁ | 13 | H'0034 to H'0037 | H'001A to H'001B |
| External interrupt IRQ ₂ | 14 | H'0038 to H'003B | H'001C to H'001D |
| External interrupt IRQ ₃ | 15 | H'003C to H'003F | H'001E to H'001F |
| External interrupt IRQ ₄ | 16 | H'0040 to H'0043 | H'0020 to H'0021 |
| External interrupt IRQ ₅ | 17 | H'0044 to H'0047 | H'0022 to H'0023 |
| Reserved for system use | 18 | H'0048 to H'004B | H'0024 to H'0025 |
| | 19 | H'004C to H'004F | H'0026 to H'0027 |
| Internal interrupts* ² | 20 | H'0050 to H'0053 | H'0028 to H'0029 |
| | to | to | to |
| | 63 | H'00FC to H'00FF | H'007E to H'007F |

Notes: 1. Lower 16 bits of the address.

2. For the internal interrupt vectors, see section 5.3.3, Interrupt Vector Table.

4.2 Reset

4.2.1 Overview

A reset is the highest-priority exception. When the $\overline{\text{RES}}$ pin goes low, all processing halts and the chip enters the reset state. A reset initializes the internal state of the CPU and the registers of the on-chip supporting modules. Reset exception handling begins when the $\overline{\text{RES}}$ pin changes from low to high.

The chip can also be reset by overflow of the watchdog timer. For details see section 11, Watchdog Timer.

4.2.2 Reset Sequence

The chip enters the reset state when the $\overline{\text{RES}}$ pin goes low.

To ensure that the chip is reset, hold the $\overline{\text{RES}}$ pin low for at least 20 ms at power-up. To reset the chip during operation, hold the $\overline{\text{RES}}$ pin low for at least 10 system clock (ϕ) cycles. In the versions with on-chip flash memory, the $\overline{\text{RES}}$ pin must be held low for at least 20 system clock cycles. See appendix D.2, Pin States at Reset, for the states of the pins in the reset state.

When the $\overline{\text{RES}}$ pin goes high after being held low for the necessary time, the chip starts reset exception handling as follows.

- The internal state of the CPU and the registers of the on-chip supporting modules are initialized, and the I bit is set to 1 in CCR.
- The contents of the reset vector address (H'0000 to H'0003 in advanced mode, H'0000 to H'0001 in normal mode) are read, and program execution starts from the address indicated in the vector address.

Figure 4.2 shows the reset sequence in modes 1 and 3. Figure 4.3 shows the reset sequence in modes 2 and 4.

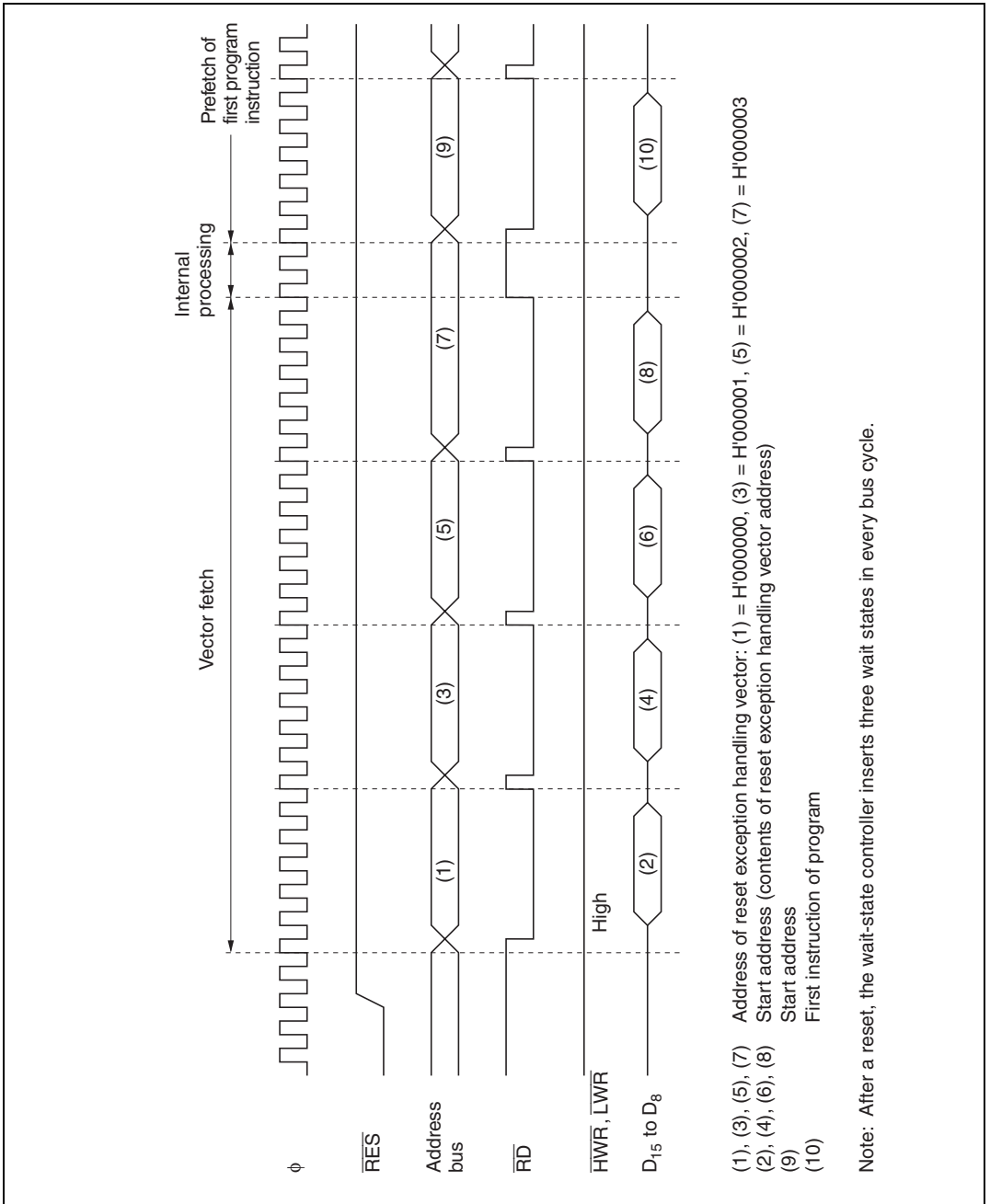


Figure 4.2 Reset Sequence (Modes 1 and 3)

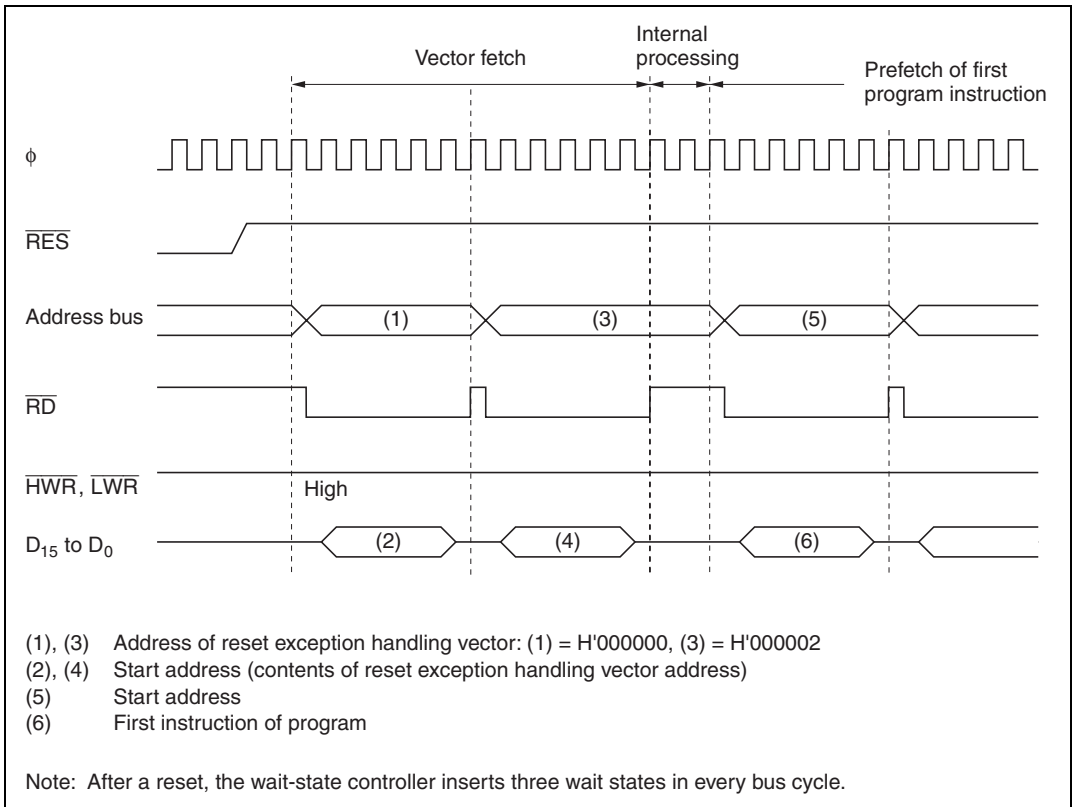


Figure 4.3 Reset Sequence (Modes 2 and 4)

4.2.3 Interrupts after Reset

If an interrupt is accepted after a reset but before the stack pointer (SP) is initialized, PC and CCR will not be saved correctly, leading to a program crash. To prevent this, all interrupt requests, including NMI, are disabled immediately after a reset exception handling. The first instruction of the program is always executed immediately after the reset state ends. This instruction should initialize the stack pointer (example: `MOV.L #xx:32, SP`).

4.3 Interrupts

Interrupt exception handling can be requested by seven external sources (NMI, IRQ₀ to IRQ₅), and 27 internal sources in the on-chip supporting modules. Figure 4.4 classifies the interrupt sources and indicates the number of interrupts of each type.

The on-chip supporting modules that can request interrupts are the watchdog timer (WDT), 16-bit timer, 8-bit timer, serial communication interface (SCI), and A/D converter. Each interrupt source has a separate vector address.

NMI is the highest-priority interrupt and is always accepted. Interrupts are controlled by the interrupt controller. The interrupt controller can assign interrupts other than NMI to two priority levels, and arbitrate between simultaneous interrupts. Interrupt priorities are assigned in interrupt priority registers A and B (IPRA and IPRB) in the interrupt controller.

For details on interrupts see section 5, Interrupt Controller.

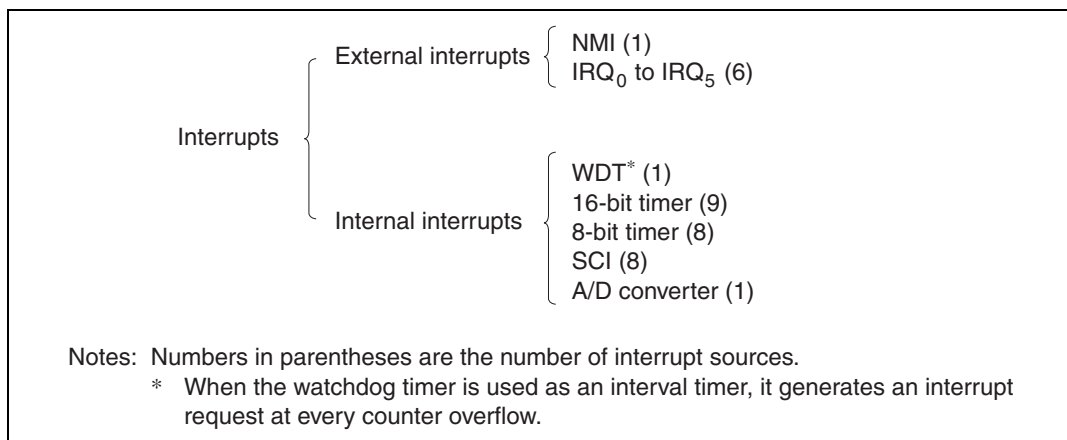


Figure 4.4 Interrupt Sources and Number of Interrupts

4.4 Trap Instruction

Trap instruction exception handling starts when a TRAPA instruction is executed. If the UE bit is set to 1 in the system control register (SYSCR), the exception handling sequence sets the I bit to 1 in CCR. If the UE bit is 0, the I and UI bits are both set to 1 in CCR. The TRAPA instruction fetches a start address from a vector table entry corresponding to a vector number from 0 to 3, which is specified in the instruction code.

4.5 Stack Status after Exception Handling

Figure 4.5 shows the stack after completion of trap instruction exception handling and interrupt exception handling.

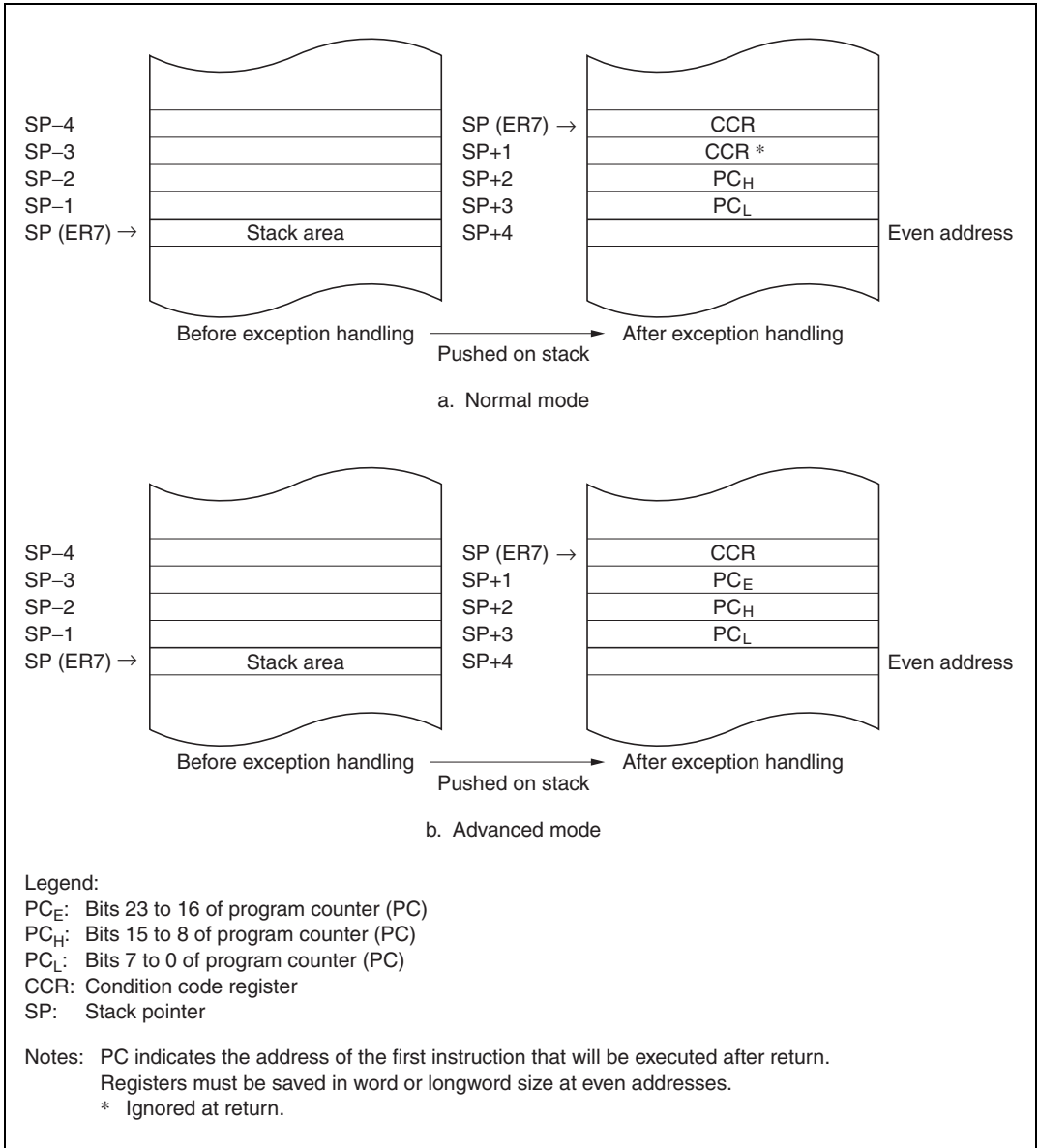


Figure 4.5 Stack after Completion of Exception Handling

4.6 Notes on Stack Usage

When accessing word data or longword data, the H8/3008 regards the lowest address bit as 0. The stack should always be accessed by word access or longword access, and the value of the stack pointer (SP:ER7) should always be kept even.

Use the following instructions to save registers:

PUSH.W Rn (or MOV.W Rn, @-SP)

PUSH.L ERn (or MOV.L ERn, @-SP)

Use the following instructions to restore registers:

POP.W Rn (or MOV.W @SP+, Rn)

POP.L ERn (or MOV.L @SP+, ERn)

Setting SP to an odd value may lead to a malfunction. Figure 4.6 shows an example of what happens when the SP value is odd.

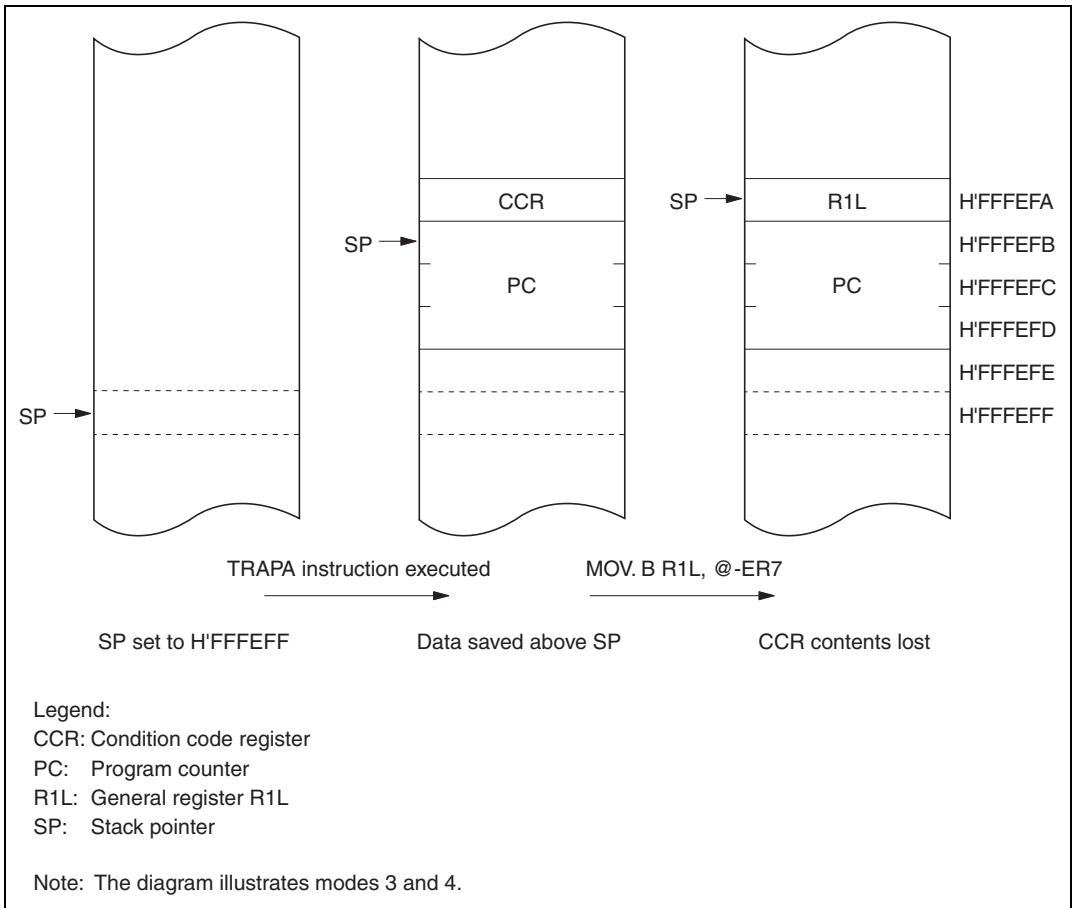


Figure 4.6 Operation when SP Value is Odd

Section 5 Interrupt Controller

5.1 Overview

5.1.1 Features

The interrupt controller has the following features:

- Interrupt priority registers (IPRs) for setting interrupt priorities
Interrupts other than NMI can be assigned to two priority levels on a module-by-module basis in interrupt priority registers A and B (IPRA and IPRB).
- Three-level enabling/disabling by the I and UI bits in the CPU's condition code register (CCR) and the UE bit in the system control register (SYSCR)
- Seven external interrupt pins

NMI has the highest priority and is always accepted; either the rising or falling edge can be selected. For each of IRQ_5 to IRQ_0 , sensing of the falling edge or level sensing can be selected independently.

5.1.2 Block Diagram

Figure 5.1 shows a block diagram of the interrupt controller.

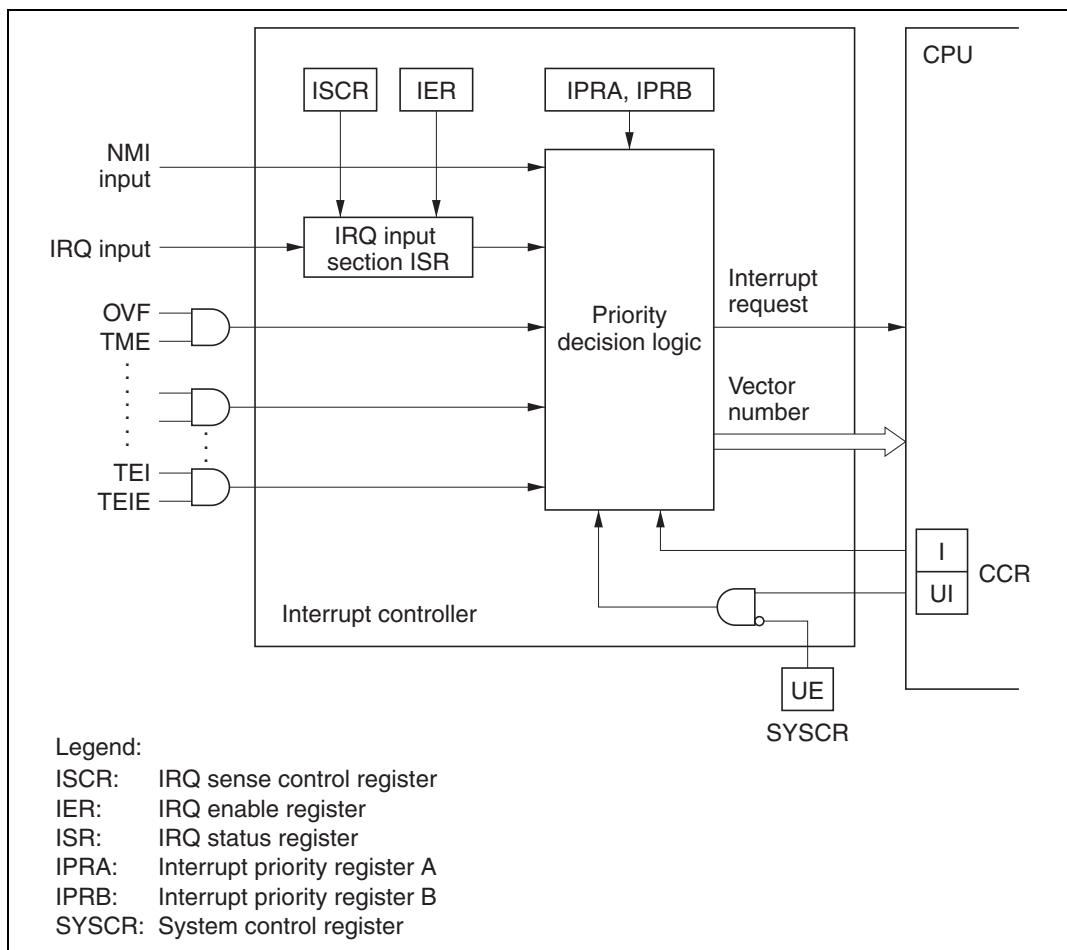


Figure 5.1 Interrupt Controller Block Diagram

5.1.3 Pin Configuration

Table 5.1 lists the interrupt pins.

Table 5.1 Interrupt Pins

| Name | Abbreviation | I/O | Function |
|-----------------------------------|--|-------|---|
| Nonmaskable interrupt | NMI | Input | Nonmaskable interrupt, rising edge or falling edge selectable |
| External interrupt request 5 to 0 | $\overline{\text{IRQ}}_5$ to $\overline{\text{IRQ}}_0$ | Input | Maskable interrupts, falling edge or level sensing selectable |

5.1.4 Register Configuration

Table 5.2 lists the registers of the interrupt controller.

Table 5.2 Interrupt Controller Registers

| Address* ¹ | Name | Abbreviation | R/W | Initial Value |
|-----------------------|-------------------------------|--------------|---------------------|---------------|
| H'EE012 | System control register | SYSCR | R/W | H'09 |
| H'EE014 | IRQ sense control register | ISCR | R/W | H'00 |
| H'EE015 | IRQ enable register | IER | R/W | H'00 |
| H'EE016 | IRQ status register | ISR | R/(W)* ² | H'00 |
| H'EE018 | Interrupt priority register A | IPRA | R/W | H'00 |
| H'EE019 | Interrupt priority register B | IPRB | R/W | H'00 |

Notes: 1. Lower 20 bits of the address in advanced mode.

2. Only 0 can be written, to clear flags.

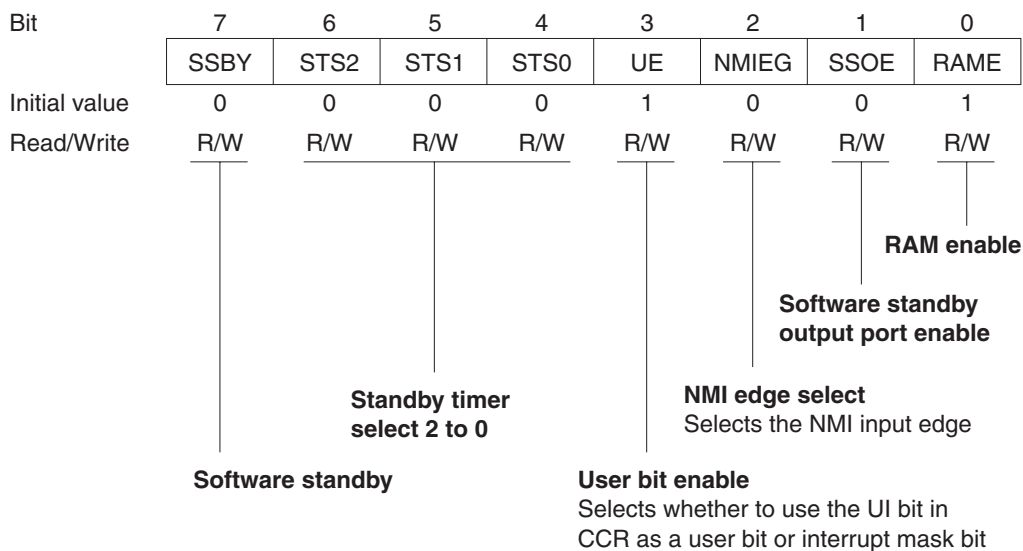
5.2 Register Descriptions

5.2.1 System Control Register (SYSCR)

SYSCR is an 8-bit readable/writable register that controls software standby mode, selects the action of the UI bit in CCR, selects the NMI edge, and enables or disables the on-chip RAM.

Only bits 3 and 2 are described here. For the other bits, see section 3.3, System Control Register (SYSCR).

SYSCR is initialized to H'09 by a reset and in hardware standby mode. It is not initialized in software standby mode.



Bit 3—User Bit Enable (UE): Selects whether to use the UI bit in CCR as a user bit or an interrupt mask bit.

| Bit 3 UE | Description |
|-------------|---|
| 0 | UI bit in CCR is used as interrupt mask bit |
| 1 | UI bit in CCR is used as user bit (Initial value) |

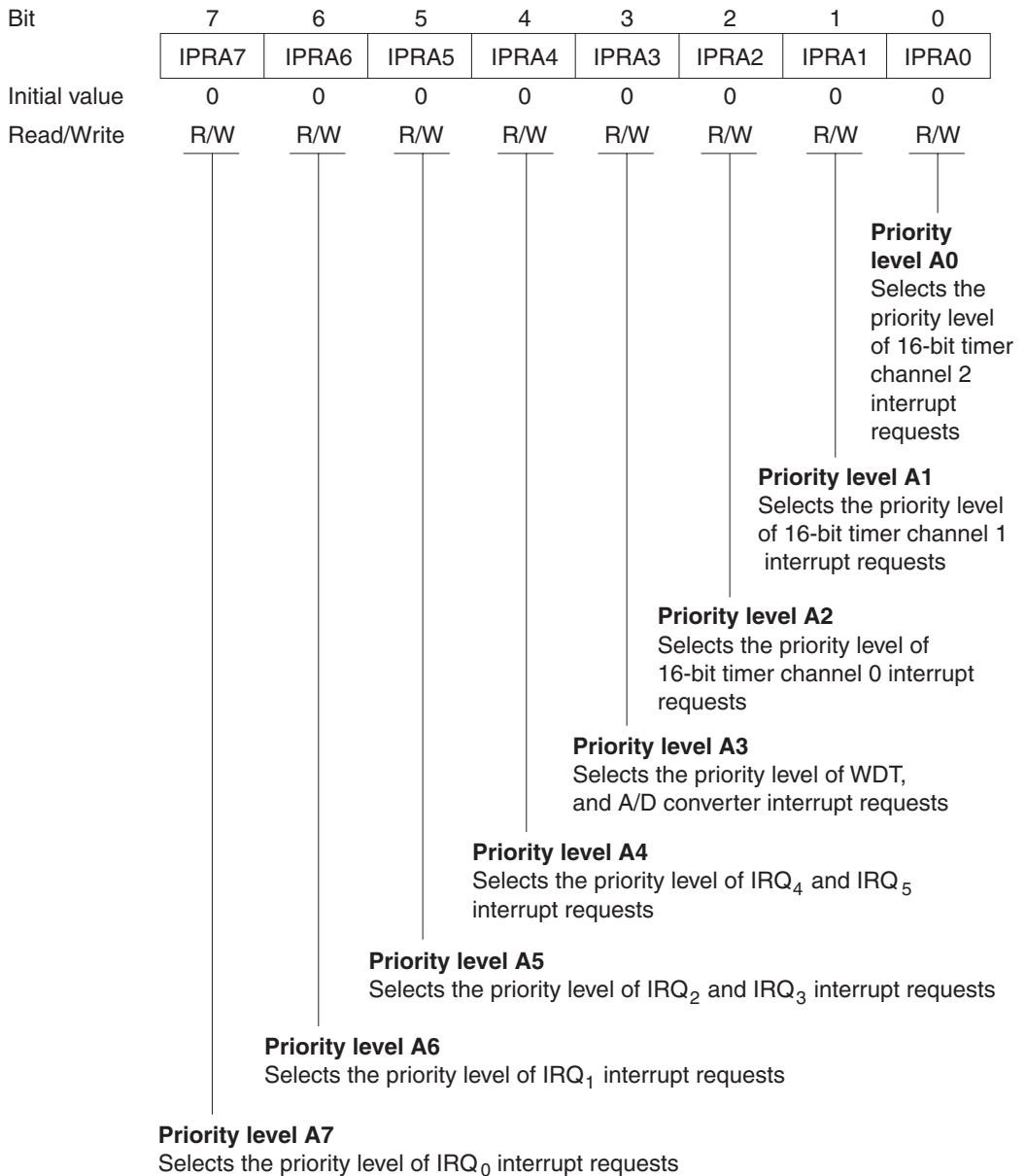
Bit 2—NMI Edge Select (NMIEG): Selects the NMI input edge.

| Bit 2 NMIEG | Description |
|----------------|---|
| 0 | Interrupt is requested at falling edge of NMI input (Initial value) |
| 1 | Interrupt is requested at rising edge of NMI input |

5.2.2 Interrupt Priority Registers A and B (IPRA, IPRB)

IPRA and IPRB are 8-bit readable/writable registers that control interrupt priority.

Interrupt Priority Register A (IPRA): IPRA is an 8-bit readable/writable register in which interrupt priority levels can be set.



IPRA is initialized to H'00 by a reset and in hardware standby mode.

Bit 7—Priority Level A7 (IPRA7): Selects the priority level of IRQ_0 interrupt requests.

Bit 7

| IPRA7 | Description |
|--------------|---|
| 0 | IRQ_0 interrupt requests have priority level 0 (low priority) (Initial value) |
| 1 | IRQ_0 interrupt requests have priority level 1 (high priority) |

Bit 6—Priority Level A6 (IPRA6): Selects the priority level of IRQ_1 interrupt requests.

Bit 6

| IPRA6 | Description |
|--------------|---|
| 0 | IRQ_1 interrupt requests have priority level 0 (low priority) (Initial value) |
| 1 | IRQ_1 interrupt requests have priority level 1 (high priority) |

Bit 5—Priority Level A5 (IPRA5): Selects the priority level of IRQ_2 and IRQ_3 interrupt requests.

Bit 5

| IPRA5 | Description |
|--------------|---|
| 0 | IRQ_2 and IRQ_3 interrupt requests have priority level 0 (low priority) (Initial value) |
| 1 | IRQ_2 and IRQ_3 interrupt requests have priority level 1 (high priority) |

Bit 4—Priority Level A4 (IPRA4): Selects the priority level of IRQ_4 and IRQ_5 interrupt requests.

Bit 4

| IPRA4 | Description |
|--------------|---|
| 0 | IRQ_4 and IRQ_5 interrupt requests have priority level 0 (low priority) (Initial value) |
| 1 | IRQ_4 and IRQ_5 interrupt requests have priority level 1 (high priority) |

Bit 3—Priority Level A3 (IPRA3): Selects the priority level of WDT, and A/D converter interrupt requests.

Bit 3

| IPRA3 | Description |
|--------------|--|
| 0 | WDT, and A/D converter interrupt requests have priority level 0 (low priority) (Initial value) |
| 1 | WDT, and A/D converter interrupt requests have priority level 1 (high priority) |

Bit 2—Priority Level A2 (IPRA2): Selects the priority level of 16-bit timer channel 0 interrupt requests.

| Bit 2 IPRA2 | Description |
|----------------|--|
| 0 | 16-bit timer channel 0 interrupt requests have priority level 0 (low priority) (Initial value) |
| 1 | 16-bit timer channel 0 interrupt requests have priority level 1 (high priority) |

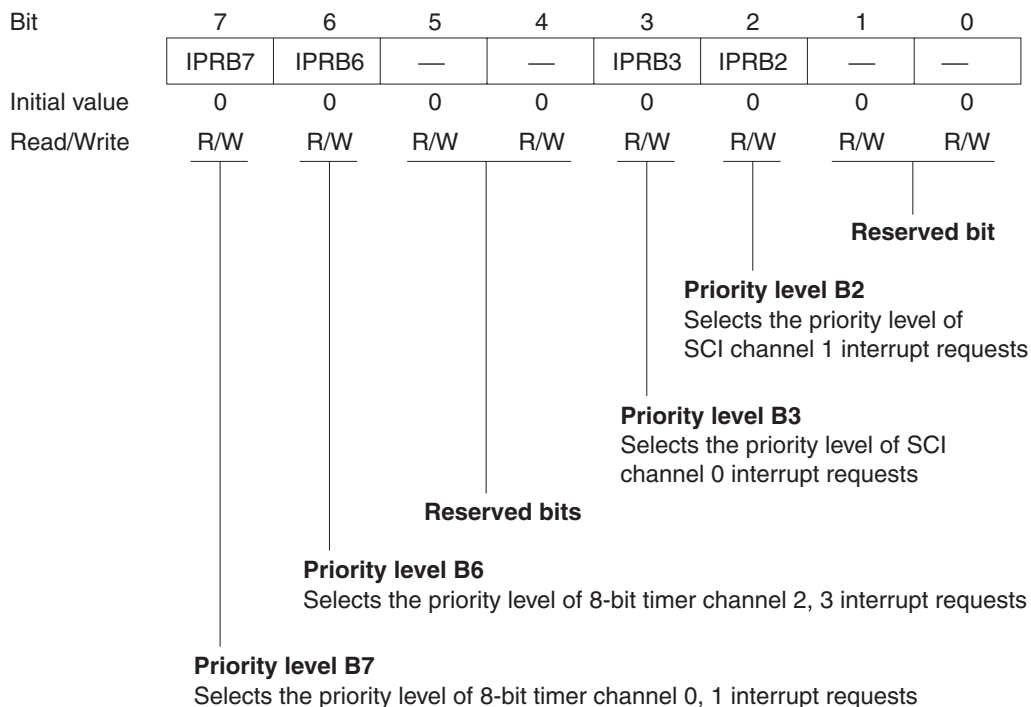
Bit 1—Priority Level A1 (IPRA1): Selects the priority level of 16-bit timer channel 1 interrupt requests.

| Bit 1 IPRA1 | Description |
|----------------|--|
| 0 | 16-bit timer channel 1 interrupt requests have priority level 0 (low priority) (Initial value) |
| 1 | 16-bit timer channel 1 interrupt requests have priority level 1 (high priority) |

Bit 0—Priority Level A0 (IPRA0): Selects the priority level of 16-bit timer channel 2 interrupt requests.

| Bit 0 IPRA0 | Description |
|----------------|--|
| 0 | 16-bit timer channel 2 interrupt requests have priority level 0 (low priority) (Initial value) |
| 1 | 16-bit timer channel 2 interrupt requests have priority level 1 (high priority) |

Interrupt Priority Register B (IPRB): IPRB is an 8-bit readable/writable register in which interrupt priority levels can be set.



IPRB is initialized to H'00 by a reset and in hardware standby mode.

Bit 7—Priority Level B7 (IPRB7): Selects the priority level of 8-bit timer channel 0, 1 interrupt requests.

Bit 7

| IPRB7 | Description |
|-------|--|
| 0 | 8-bit timer channel 0 and 1 interrupt requests have priority level 0 (low priority) (Initial value) |
| 1 | 8-bit timer channel 0 and 1 interrupt requests have priority level 1 (high priority) |

Bit 6—Priority Level B6 (IPRB6): Selects the priority level of 8-bit timer channel 2, 3 interrupt requests.

| Bit 6 IPRB6 | Description |
|----------------|--|
| 0 | 8-bit timer channel 2 and 3 interrupt requests have priority level 0 (low priority) (Initial value) |
| 1 | 8-bit timer channel 2 and 3 interrupt requests have priority level 1 (high priority) |

Bits 5 and 4—Reserved: These bits can be written and read, but they do not affect interrupt priority.

Bit 3—Priority Level B3 (IPRB3): Selects the priority level of SCI channel 0 interrupt requests.

| Bit 3 IPRB3 | Description |
|----------------|---|
| 0 | SCI channel 0 interrupt requests have priority level 0 (low priority) (Initial value) |
| 1 | SCI channel 0 interrupt requests have priority level 1 (high priority) |

Bit 2—Priority Level B2 (IPRB2): Selects the priority level of SCI channel 1 interrupt requests.

| Bit 2 IPRB2 | Description |
|----------------|---|
| 0 | SCI channel 1 interrupt requests have priority level 0 (low priority) (Initial value) |
| 1 | SCI channel 1 interrupt requests have priority level 1 (high priority) |

Bits 1 and 0—Reserved: These bits can be written and read, but they do not affect interrupt priority.

5.2.3 IRQ Status Register (ISR)

ISR is an 8-bit readable/writable register that indicates the status of IRQ_5 to IRQ_0 interrupt requests.

| | | | | | | | | |
|---------------|---|---|--------|--------|--------|--------|--------|--------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | IRQ5F | IRQ4F | IRQ3F | IRQ2F | IRQ1F | IRQ0F |
| Initial value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Read/Write | — | — | R/(W)* | R/(W)* | R/(W)* | R/(W)* | R/(W)* | R/(W)* |

Reserved bits
IRQ₅ to IRQ₀ flags
 These bits indicate IRQ₅ to IRQ₀ flag interrupt request status

Note: * Only 0 can be written, to clear flags.

ISR is initialized to H'00 by a reset and in hardware standby mode.

Bits 7 and 6—Reserved: These bits can not be modified and are always read as 0.

Bits 5 to 0—IRQ₅ to IRQ₀ Flags (IRQ5F to IRQ0F): These bits indicate the status of IRQ₅ to IRQ₀ interrupt requests.

Bits 5 to 0

IRQ5F to IRQ0F Description

| | | |
|---|---|-----------------|
| 0 | [Clearing conditions] 0 is written in IRQnF after reading the IRQnF flag when IRQnF = 1. IRQnSC = 0, $\overline{\text{IRQn}}$ input is high, and interrupt exception handling is carried out. IRQnSC = 1 and IRQn interrupt exception handling is carried out. | (Initial value) |
| 1 | [Setting conditions] IRQnSC = 0 and $\overline{\text{IRQn}}$ input is low. IRQnSC = 1 and $\overline{\text{IRQn}}$ input changes from high to low. | |

Note: n = 5 to 0

5.2.4 IRQ Enable Register (IER)

IER is an 8-bit readable/writable register that enables or disables IRQ₅ to IRQ₀ interrupt requests.

| | | | | | | | | |
|---------------|-----|-----|-------|-------|-------|-------|-------|-------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | IRQ5E | IRQ4E | IRQ3E | IRQ2E | IRQ1E | IRQ0E |
| Initial value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Reserved bits
IRQ₅ to IRQ₀ enable
 These bits enable or disable IRQ₅ to IRQ₀ interrupts

IER is initialized to H'00 by a reset and in hardware standby mode.

Bits 7 and 6—Reserved: These bits can be written and read, but they do not enable or disable interrupts.

Bits 5 to 0—IRQ₅ to IRQ₀ Enable (IRQ5E to IRQ0E): These bits enable or disable IRQ₅ to IRQ₀ interrupts.

Bits 5 to 0
IRQ5E to IRQ0E Description

| | | |
|---|--|-----------------|
| 0 | IRQ ₅ to IRQ ₀ interrupts are disabled | (Initial value) |
| 1 | IRQ ₅ to IRQ ₀ interrupts are enabled | |

5.2.5 IRQ Sense Control Register (ISCR)

ISCR is an 8-bit readable/writable register that selects level sensing or falling-edge sensing of the inputs at pins $\overline{\text{IRQ}}_5$ to $\overline{\text{IRQ}}_0$.

| | | | | | | | | |
|---------------|-----|-----|--------|--------|--------|--------|--------|--------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | IRQ5SC | IRQ4SC | IRQ3SC | IRQ2SC | IRQ1SC | IRQ0SC |
| Initial value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Reserved bits
IRQ₅ to IRQ₀ sense control
 These bits select level sensing or falling-edge sensing for IRQ₅ to IRQ₀ interrupts

ISCR is initialized to H'00 by a reset and in hardware standby mode.

Bits 7 and 6—Reserved: These bits can be written and read, but they do not select level or falling-edge sensing.

Bits 5 to 0—IRQ₅ to IRQ₀ Sense Control (IRQ5SC to IRQ0SC): These bits select whether interrupts IRQ₅ to IRQ₀ are requested by level sensing of pins $\overline{\text{IRQ}}_5$ to $\overline{\text{IRQ}}_0$, or by falling-edge sensing.

Bits 5 to 0
IRQ5SC to IRQ0SC Description

| | | |
|---|--|-----------------|
| 0 | Interrupts are requested when $\overline{\text{IRQ}}_5$ to $\overline{\text{IRQ}}_0$ inputs are low | (Initial value) |
| 1 | Interrupts are requested by falling-edge input at $\overline{\text{IRQ}}_5$ to $\overline{\text{IRQ}}_0$ | |

5.3 Interrupt Sources

The interrupt sources include external interrupts (NMI, IRQ₅ to IRQ₀) and 27 internal interrupts.

5.3.1 External Interrupts

There are seven external interrupts: NMI, and IRQ₅ to IRQ₀. Of these, NMI, IRQ₂, IRQ₁, and IRQ₀ can be used to exit software standby mode.

NMI: NMI is the highest-priority interrupt and is always accepted, regardless of the states of the I and UI bits in CCR. The NMIEG bit in SYSCR selects whether an interrupt is requested by the rising or falling edge of the input at the NMI pin. NMI interrupt exception handling has vector number 7.

IRQ₅ to IRQ₀ Interrupts: These interrupts are requested by input signals at pins $\overline{\text{IRQ}}_5$ to $\overline{\text{IRQ}}_0$. The IRQ₅ to IRQ₀ interrupts have the following features.

- ISCR settings can select whether an interrupt is requested by the low level of the input at pins $\overline{\text{IRQ}}_5$ to $\overline{\text{IRQ}}_0$, or by the falling edge.
- IER settings can enable or disable the IRQ₅ to IRQ₀ interrupts. Interrupt priority levels can be assigned by four bits in IPRA (IPRA7 to IPRA4).
- The status of IRQ₅ to IRQ₀ interrupt requests is indicated in ISR. The ISR flags can be cleared to 0 by software.

Figure 5.2 shows a block diagram of interrupts IRQ₅ to IRQ₀.

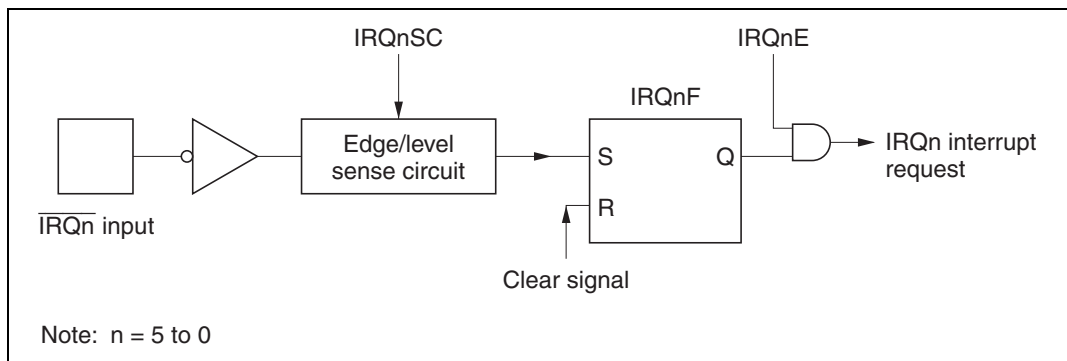


Figure 5.2 Block Diagram of Interrupts IRQ₅ to IRQ₀

Figure 5.3 shows the timing of the setting of the interrupt flags (IRQnF).

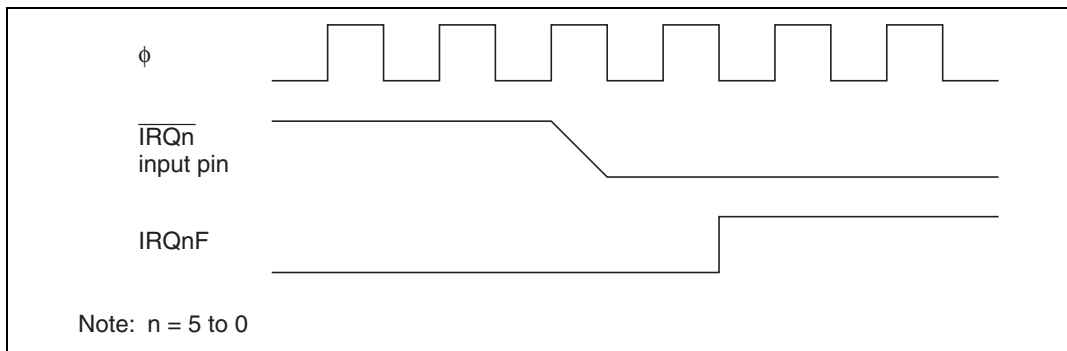


Figure 5.3 Timing of Setting of IRQnF

Interrupts IRQ_0 to IRQ_5 have vector numbers 12 to 17. These interrupts are detected regardless of whether the corresponding pin is set for input or output. When using a pin for external interrupt input, clear its DDR bit to 0 and do not use the pin for chip select output, SCI input/output, or A/D external trigger input.

5.3.2 Internal Interrupts

Twenty-Seven internal interrupts are requested from the on-chip supporting modules.

- Each on-chip supporting module has status flags for indicating interrupt status, and enable bits for enabling or disabling interrupts.
- Interrupt priority levels can be assigned in IPRA and IPRB.

5.3.3 Interrupt Exception Handling Vector Table

Table 5.3 lists the interrupt exception handling sources, their vector addresses, and their default priority order. In the default priority order, smaller vector numbers have higher priority. The priority of interrupts other than NMI can be changed in IPRA and IPRB. The priority order after a reset is the default order shown in table 5.3.

Table 5.3 Interrupt Sources, Vector Addresses, and Priority

| Interrupt Source | Origin | Vector Number | Vector Address* | | IPR | Priority |
|---|---------------------------|---------------|--------------------------------------|--------------------------------------|-------|-----------|
| | | | Advanced Mode | Normal Mode | | |
| NMI | External pins | 7 | H'001C to H'001F | H'000E to H'000F | — | High ↑ |
| IRQ ₀ | | 12 | H'0030 to H'0033 | H'0018 to H'0019 | IPRA7 | |
| IRQ ₁ | | 13 | H'0034 to H'0037 | H'001A to H'001B | IPRA6 | |
| IRQ ₂ | | 14 | H'0038 to H'003B | H'001C to H'001D | IPRA5 | |
| IRQ ₃ | | 15 | H'003C to H'003F | H'001E to H'001F | | |
| IRQ ₄ IRQ ₅ | | 16 17 | H'0040 to H'0043 H'0044 to H'0047 | H'0020 to H'0021 H'0022 to H'0023 | IPRA4 | |
| Reserved | — | 18 | H'0048 to H'004B | H'0024 to H'0025 | | |
| | | 19 | H'004C to H'004F | H'0026 to H'0027 | | |
| WOVI (interval timer) | Watchdog timer | 20 | H'0050 to H'0053 | H'0028 to H'0029 | IPRA3 | |
| Reserved | — | 21 | H'0054 to H'0057 | H'002A to H'002B | | |
| | | 22 | H'0058 to H'005B | H'002C to H'002D | | |
| ADI (A/D end) | A/D | 23 | H'005C to H'005F | H'002E to H'002F | | |
| IMIA0 (compare match/ input capture A0) | 16-bit timer channel 0 | 24 | H'0060 to H'0063 | H'0030 to H'0031 | IPRA2 | |
| IMIB0 (compare match/ input capture B0) | | 25 | H'0064 to H'0067 | H'0032 to H'0033 | | |
| OVI0 (overflow 0) | | 26 | H'0068 to H'006B | H'0034 to H'0035 | | |
| Reserved | — | 27 | H'006C to H'006F | H'0036 to H'0037 | | |
| IMIA1 (compare match/ inputcapture A1) | 16-bit timer channel 1 | 28 | H'0070 to H'0073 | H'0038 to H'0039 | IPRA1 | |
| IMIB1 (compare match/ input capture B1) | | 29 | H'0074 to H'0077 | H'003A to H'003B | | |
| OVI1 (overflow 1) | | 30 | H'0078 to H'007B | H'003C to H'003D | | |
| Reserved | — | 31 | H'007C to H'007F | H'003E to H'003F | | Low |

5. Interrupt Controller

| Interrupt Source | Origin | Vector Number | Vector Address* | | IPR | Priority |
|---|-------------------------------|---------------|------------------|------------------|-------|----------|
| | | | Advanced Mode | Normal Mode | | |
| IMIA2 (compare match/ input capture A2) | 16-bit timer channel 2 | 32 | H'0080 to H'0083 | H'0040 to H'0041 | IPRA0 | High |
| IMIB2 (compare match/ input capture B2) | | 33 | H'0084 to H'0087 | H'0042 to H'0043 | | |
| OVI2 (overflow 2) | | 34 | H'0088 to H'008B | H'0044 to H'0045 | | |
| Reserved | — | 35 | H'008C to H'008F | H'0046 to H'0047 | | |
| CMIA0 (compare match A0) | 8-bit timer channel 0/1 | 36 | H'0090 to H'0093 | H'0048 to H'0049 | IPRB7 | |
| CMIB0 (compare match B0) | | 37 | H'0094 to H'0097 | H'004A to H'004B | | |
| CMIA1/CMIB1 (compare match A1/B1) | | 38 | H'0098 to H'009B | H'004C to H'004D | | |
| TOVI0/TOVI1 (overflow 0/1) | | 39 | H'009C to H'009F | H'004E to H'004F | | |
| CMIA2 (compare match A2) | 8-bit timer channel 2/3 | 40 | H'00A0 to H'00A3 | H'0050 to H'0051 | IPRB6 | |
| CMIB2 (compare match B2) | | 41 | H'00A4 to H'00A7 | H'0052 to H'0053 | | |
| CMIA3/CMIB3 (compare match A3/B3) | | 42 | H'00A8 to H'00AB | H'0054 to H'0055 | | |
| TOVI2/TOVI3 (overflow 2/3) | | 43 | H'00AC to H'00AF | H'0056 to H'0057 | | |
| Reserved | — | 44 | H'00B0 to H'00B3 | H'0058 to H'0059 | — | |
| | | 45 | H'00B4 to H'00B7 | H'005A to H'005B | | |
| | | 46 | H'00B8 to H'00BB | H'005C to H'005D | | |
| | | 47 | H'00BC to H'00BF | H'005E to H'005F | | |
| | | 48 | H'00C0 to H'00C3 | H'0060 to H'0061 | | |
| | | 49 | H'00C4 to H'00C7 | H'0062 to H'0063 | | |
| | | 50 | H'00C8 to H'00CB | H'0064 to H'0065 | | |
| | | 51 | H'00CC to H'00CF | H'0066 to H'0067 | | Low |

| Interrupt Source | Origin | Vector Number | Vector Address* | | IPR | Priority |
|---------------------------------|------------------|---------------|------------------|------------------|-------|----------|
| | | | Advanced Mode | Normal Mode | | |
| ERI0 (receive error 0) | SCI channel 0 | 52 | H'00D0 to H'00D3 | H'0068 to H'0069 | IPRB3 | High |
| RX10 (receive data full 0) | | 53 | H'00D4 to H'00D7 | H'006A to H'006B | | ↑ |
| TX10 (transmit data empty 0) | | 54 | H'00D8 to H'00DB | H'006C to H'006D | | |
| TEI0 (transmit end 0) | | 55 | H'00DC to H'00DF | H'006E to H'006F | | |
| ERI1 (receive error 1) | SCI channel 1 | 56 | H'00E0 to H'00E3 | H'0070 to H'0071 | IPRB2 | |
| RX11 (receive data full 1) | | 57 | H'00E4 to H'00E7 | H'0072 to H'0073 | | ↓ |
| TX11 (transmit data empty 1) | | 58 | H'00E8 to H'00EB | H'0074 to H'0075 | | |
| TEI1 (transmit end 1) | | 59 | H'00EC to H'00EF | H'0076 to H'0077 | | |
| Reserved | — | 60 | H'00F0 to H'00F3 | H'0078 to H'0079 | — | |
| | | 61 | H'00F4 to H'00F7 | H'007A to H'007B | | Low |
| | | 62 | H'00F8 to H'00FB | H'007C to H'007D | | |
| | | 63 | H'00FC to H'00FF | H'007E to H'007F | | |

Note: * Lower 16 bits of the address.

5.4 Interrupt Operation

5.4.1 Interrupt Handling Process

The H8/3008 handles interrupts differently depending on the setting of the UE bit. When $UE = 1$, interrupts are controlled by the I bit. When $UE = 0$, interrupts are controlled by the I and UI bits. Table 5.4 indicates how interrupts are handled for all setting combinations of the UE, I, and UI bits.

NMI interrupts are always accepted except in the reset and hardware standby states. IRQ interrupts and interrupts from the on-chip supporting modules have their own enable bits. Interrupt requests are ignored when the enable bits are cleared to 0.

Table 5.4 UE, I, and UI Bit Settings and Interrupt Handling

| SYSCR | CCR | | Description |
|-------|-----|----|---|
| | I | UI | |
| 1 | 0 | — | All interrupts are accepted. Interrupts with priority level 1 have higher priority. |
| | 1 | — | No interrupts are accepted except NMI. |
| 0 | 0 | — | All interrupts are accepted. Interrupts with priority level 1 have higher priority. |
| | 1 | 0 | NMI and interrupts with priority level 1 are accepted. |
| | | 1 | No interrupts are accepted except NMI. |

UE = 1: Interrupts IRQ_5 to IRQ_0 and interrupts from the on-chip supporting modules can all be masked by the I bit in the CPU's CCR. Interrupts are masked when the I bit is set to 1, and unmasked when the I bit is cleared to 0. Interrupts with priority level 1 have higher priority. Figure 5.4 is a flowchart showing how interrupts are accepted when $UE = 1$.

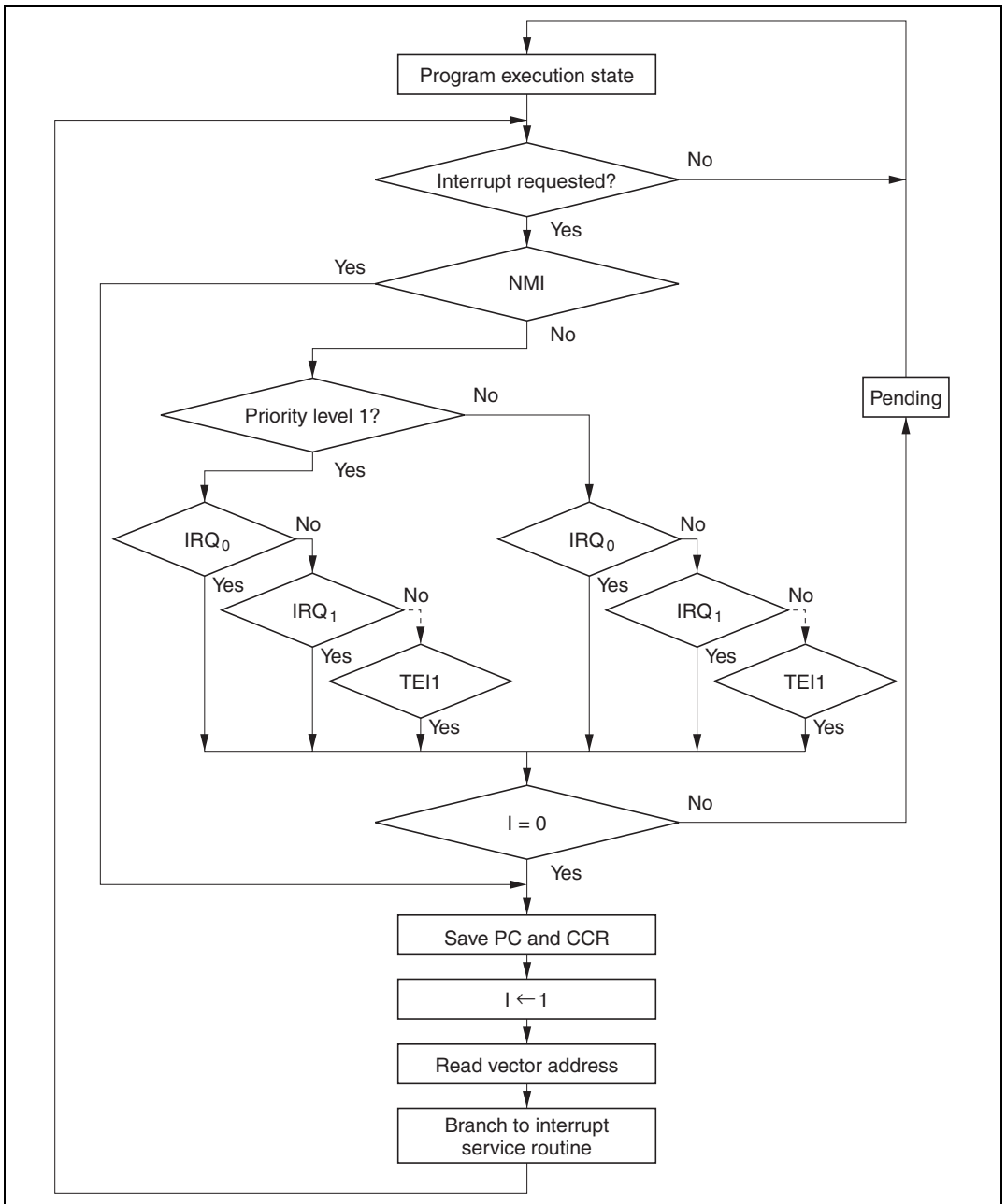


Figure 5.4 Process Up to Interrupt Acceptance when UE = 1

- If an interrupt condition occurs and the corresponding interrupt enable bit is set to 1, an interrupt request is sent to the interrupt controller.
- When the interrupt controller receives one or more interrupt requests, it selects the highest-priority request, following the IPR interrupt priority settings, and holds other requests pending. If two or more interrupts with the same IPR setting are requested simultaneously, the interrupt controller follows the priority order shown in table 5.3.
- The interrupt controller checks the I bit. If the I bit is cleared to 0, the selected interrupt request is accepted. If the I bit is set to 1, only NMI is accepted; other interrupt requests are held pending.
- When an interrupt request is accepted, interrupt exception handling starts after execution of the current instruction has been completed.
- In interrupt exception handling, PC and CCR are saved to the stack area. The PC value that is saved indicates the address of the first instruction that will be executed after the return from the interrupt service routine.
- Next the I bit is set to 1 in CCR, masking all interrupts except NMI.
- The vector address of the accepted interrupt is generated, and the interrupt service routine starts executing from the address indicated by the contents of the vector address.

UE = 0: The I and UI bits in the CPU's CCR and the IPR bits enable three-level masking of IRQ_0 to IRQ_5 interrupts and interrupts from the on-chip supporting modules.

- Interrupt requests with priority level 0 are masked when the I bit is set to 1, and are unmasked when the I bit is cleared to 0.
- Interrupt requests with priority level 1 are masked when the I and UI bits are both set to 1, and are unmasked when either the I bit or the UI bit is cleared to 0.

For example, if the interrupt enable bits of all interrupt requests are set to 1, IPRA is set to H'20, and IPRB is set to H'00 (giving IRQ_2 and IRQ_3 interrupt requests priority over other interrupts), interrupts are masked as follows:

- a. If $I = 0$, all interrupts are unmasked (priority order: $NMI > IRQ_2 > IRQ_3 > IRQ_0 \dots$).
- b. If $I = 1$ and $UI = 0$, only NMI, IRQ_2 , and IRQ_3 are unmasked.
- c. If $I = 1$ and $UI = 1$, all interrupts are masked except NMI.

Figure 5.5 shows the transitions among the above states.

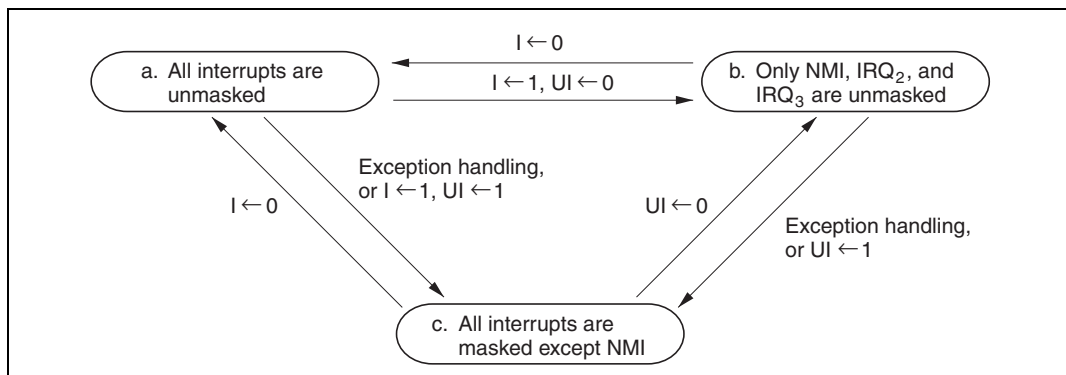


Figure 5.5 Interrupt Masking State Transitions (Example)

Figure 5.6 is a flowchart showing how interrupts are accepted when $UE = 0$.

- If an interrupt condition occurs and the corresponding interrupt enable bit is set to 1, an interrupt request is sent to the interrupt controller.
- When the interrupt controller receives one or more interrupt requests, it selects the highest-priority request, following the IPR interrupt priority settings, and holds other requests pending. If two or more interrupts with the same IPR setting are requested simultaneously, the interrupt controller follows the priority order shown in table 5.3.
- The interrupt controller checks the I bit. If the I bit is cleared to 0, the selected interrupt request is accepted regardless of its IPR setting, and regardless of the UI bit. If the I bit is set to 1 and the UI bit is cleared to 0, only interrupts with priority level 1 are accepted; interrupt requests with priority level 0 are held pending. If the I bit and UI bit are both set to 1, all other interrupt requests are held pending.
- When an interrupt request is accepted, interrupt exception handling starts after execution of the current instruction has been completed.
- In interrupt exception handling, PC and CCR are saved to the stack area. The PC value that is saved indicates the address of the first instruction that will be executed after the return from the interrupt service routine.
- The I and UI bits are set to 1 in CCR, masking all interrupts except NMI.
- The vector address of the accepted interrupt is generated, and the interrupt service routine starts executing from the address indicated by the contents of the vector address.

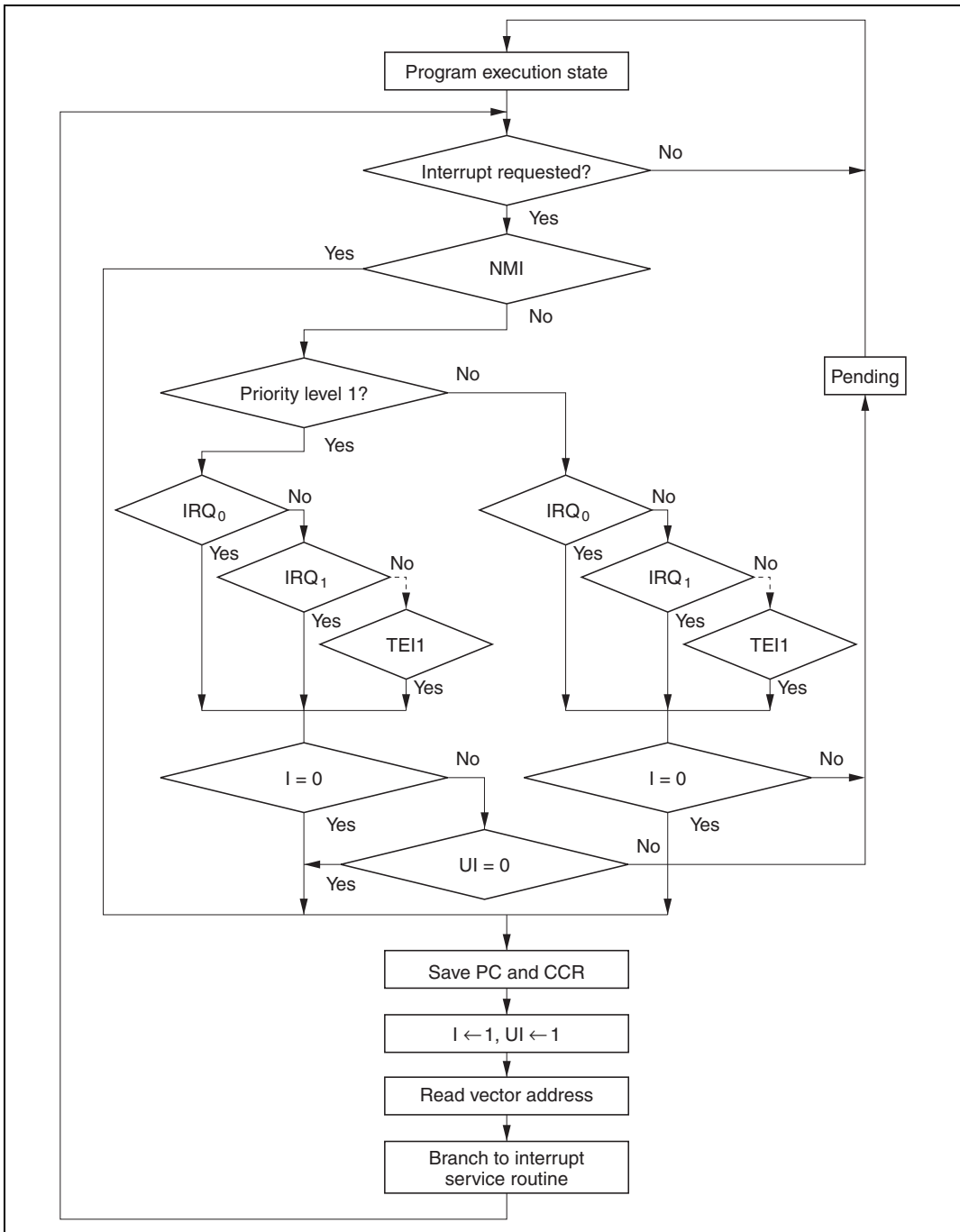


Figure 5.6 Process Up to Interrupt Acceptance when UE = 0

5.4.2 Interrupt Exception Handling Sequence

Figure 5.7 shows the interrupt exception handling sequence in mode 2 when the program code and stack are in an external memory area accessed in two states via a 16-bit bus.

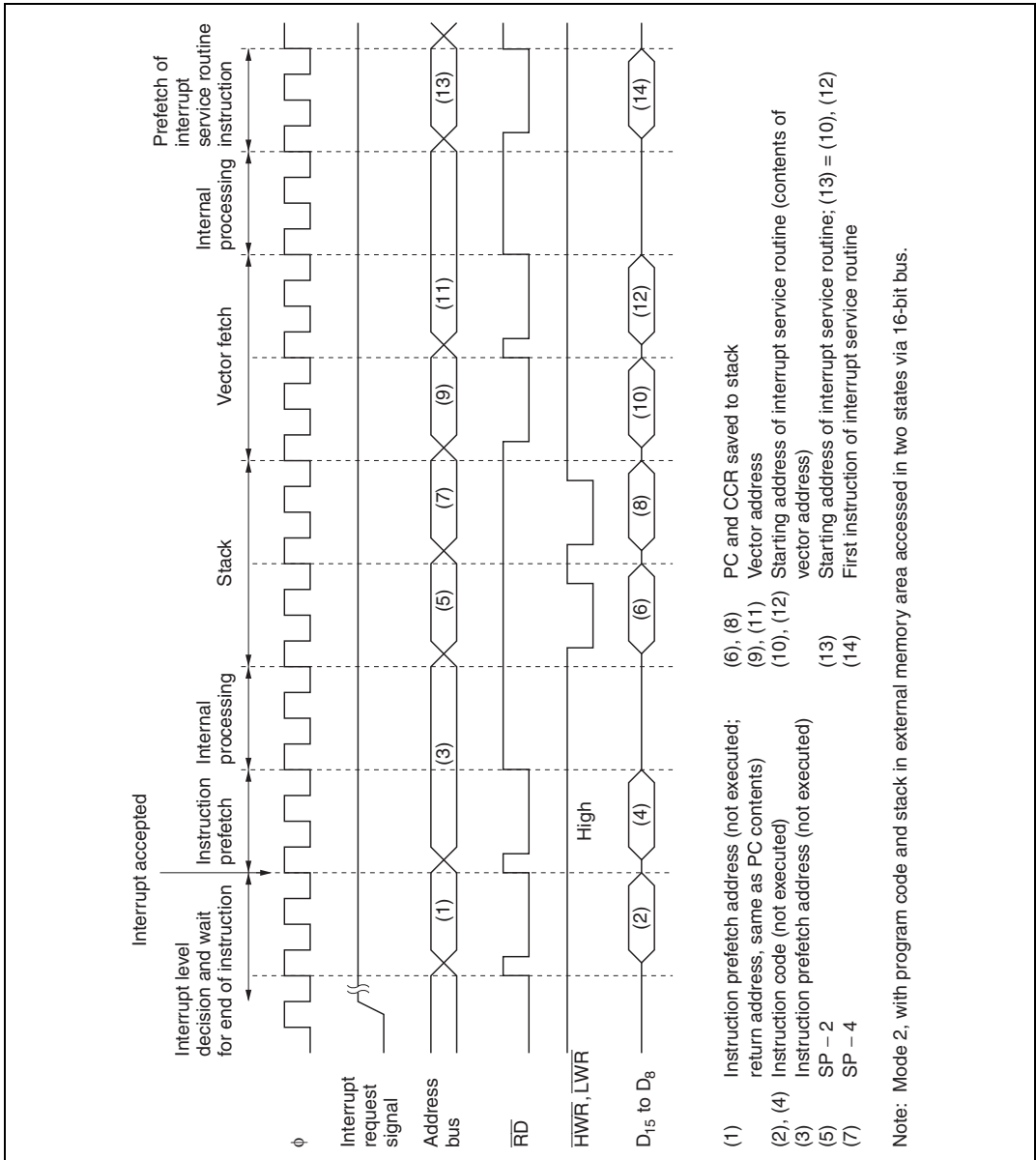


Figure 5.7 Interrupt Exception Handling Sequence

5.4.3 Interrupt Response Time

Table 5.5 indicates the interrupt response time from the occurrence of an interrupt request until the first instruction of the interrupt service routine is executed.

Table 5.5 Interrupt Response Time

| No. | Item | On-Chip Memory | External Memory | | | |
|-------|---|-------------------|-----------------|-----------------------|-----------------|-----------------------|
| | | | 8-Bit Bus | | 16-Bit Bus | |
| | | | 2 States | 3 States | 2 States | 3 States |
| 1 | Interrupt priority decision | 2* ¹ | 2* ¹ | 2* ¹ | 2* ¹ | 2* ¹ |
| 2 | Maximum number of states until end of current instruction | 1 to 23 | 1 to 27 | 1 to 31* ⁴ | 1 to 23 | 1 to 25* ⁴ |
| 3 | Saving PC and CCR to stack | 4 | 8 | 12* ⁴ | 4 | 6* ⁴ |
| 4 | Vector fetch | 4 | 8 | 12* ⁴ | 4 | 6* ⁴ |
| 5 | Instruction fetch* ² | 4 | 8 | 12* ⁴ | 4 | 6* ⁴ |
| 6 | Internal processing* ³ | 4 | 4 | 4 | 4 | 4 |
| Total | | 19 to 41 | 31 to 57 | 43 to 73 | 19 to 41 | 25 to 49 |

Notes: 1. 1 state for internal interrupts.

2. Prefetch after the interrupt is accepted and prefetch of the first instruction in the interrupt service routine.

3. Internal processing after the interrupt is accepted and internal processing after vector fetch.

4. The number of states increases if wait states are inserted in external memory access.

5.5 Usage Notes

5.5.1 Contention between Interrupt and Interrupt-Disabling Instruction

When an instruction clears an interrupt enable bit to 0 to disable the interrupt, the interrupt is not disabled until after execution of the instruction is completed. If an interrupt occurs while a BCLR, MOV, or other instruction is being executed to clear its interrupt enable bit to 0, at the instant when execution of the instruction ends the interrupt is still enabled, so its interrupt exception handling is carried out. If a higher-priority interrupt is also requested, however, interrupt exception handling for the higher-priority interrupt is carried out, and the lower-priority interrupt is ignored. This also applies to the clearing of an interrupt flag to 0.

Figure 5.8 shows an example in which an IMIEA bit is cleared to 0 in the 16-bit timer's TISRA register.

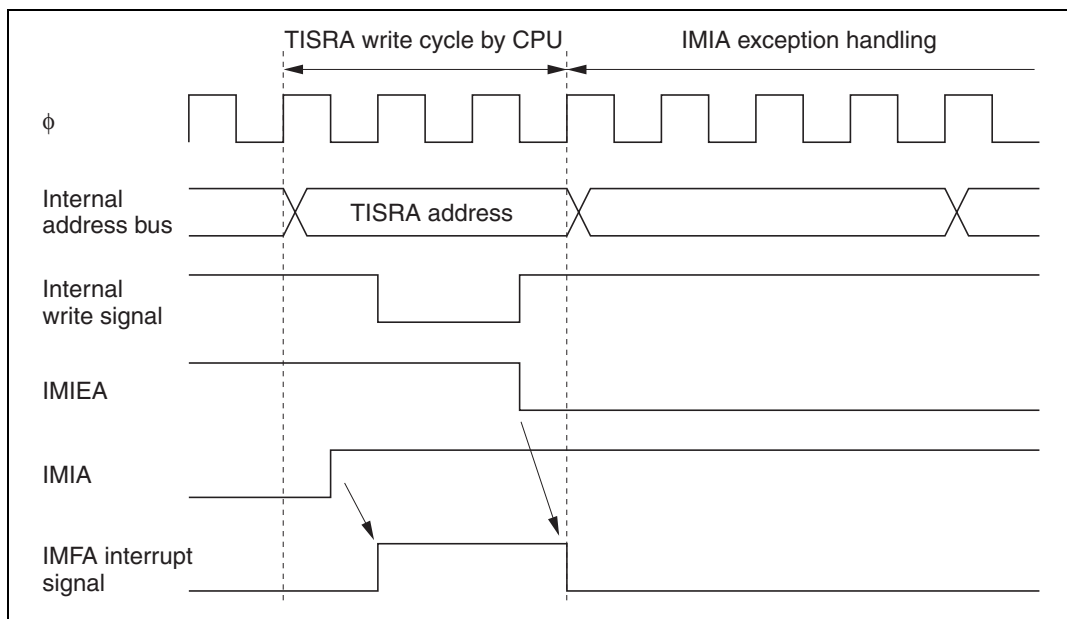


Figure 5.8 Contention between Interrupt and Interrupt-Disabling Instruction

This type of contention will not occur if the interrupt is masked when the interrupt enable bit or flag is cleared to 0.

5.5.2 Instructions that Inhibit Interrupts

The LDC, ANDC, ORC, and XORC instructions inhibit interrupts. When an interrupt occurs, after determining the interrupt priority, the interrupt controller requests a CPU interrupt. If the CPU is currently executing one of these interrupt-inhibiting instructions, however, when the instruction is completed the CPU always continues by executing the next instruction.

5.5.3 Interrupts during EEPMOV Instruction Execution

The EEPMOV.B and EEPMOV.W instructions differ in their reaction to interrupt requests.

When the EEPMOV.B instruction is executing a transfer, no interrupts are accepted until the transfer is completed, not even NMI.

When the EEPMOV.W instruction is executing a transfer, interrupt requests other than NMI are not accepted until the transfer is completed. If NMI is requested, NMI exception handling starts at a transfer cycle boundary. The PC value saved on the stack is the address of the next instruction. Programs should be coded as follows to allow for NMI interrupts during EEPMOV.W execution:

```
L1 : EEPMOV.W  
    MOV.W R4, R4  
    BNE L1
```

Section 6 Bus Controller

6.1 Overview

The H8/3008 has an on-chip bus controller (BSC) that manages the external address space divided into eight areas. The bus specifications, such as bus width and number of access states, can be set independently for each area, enabling multiple memories to be connected easily.

The bus controller also has a bus arbitration function that controls the operation of the internal bus masters—the CPU can release the bus to an external device.

6.1.1 Features

The features of the bus controller are listed below.

- Manages external address space in area units
 - Manages the external space as eight areas (0 to 7) of 128 kbytes in 1M-byte modes, or 2 Mbytes in 16-Mbyte modes
 - Bus specifications can be set independently for each area
- Basic bus interface
 - Chip select (\overline{CS}_0 to \overline{CS}_7) can be output for areas 0 to 7
 - 8-bit access or 16-bit access can be selected for each area
 - Two-state access or three-state access can be selected for each area
 - Program wait states can be inserted for each area
 - Pin wait insertion capability is provided
- Idle cycle insertion
 - An idle cycle can be inserted in case of an external read cycle between different areas
 - An idle cycle can be inserted when an external read cycle is immediately followed by an external write cycle
- Bus arbitration function
 - A built-in bus arbiter grants the bus right to the CPU, or an external bus master
- Other features
 - Choice of two address update modes

6.1.2 Block Diagram

Figure 6.1 shows a block diagram of the bus controller.

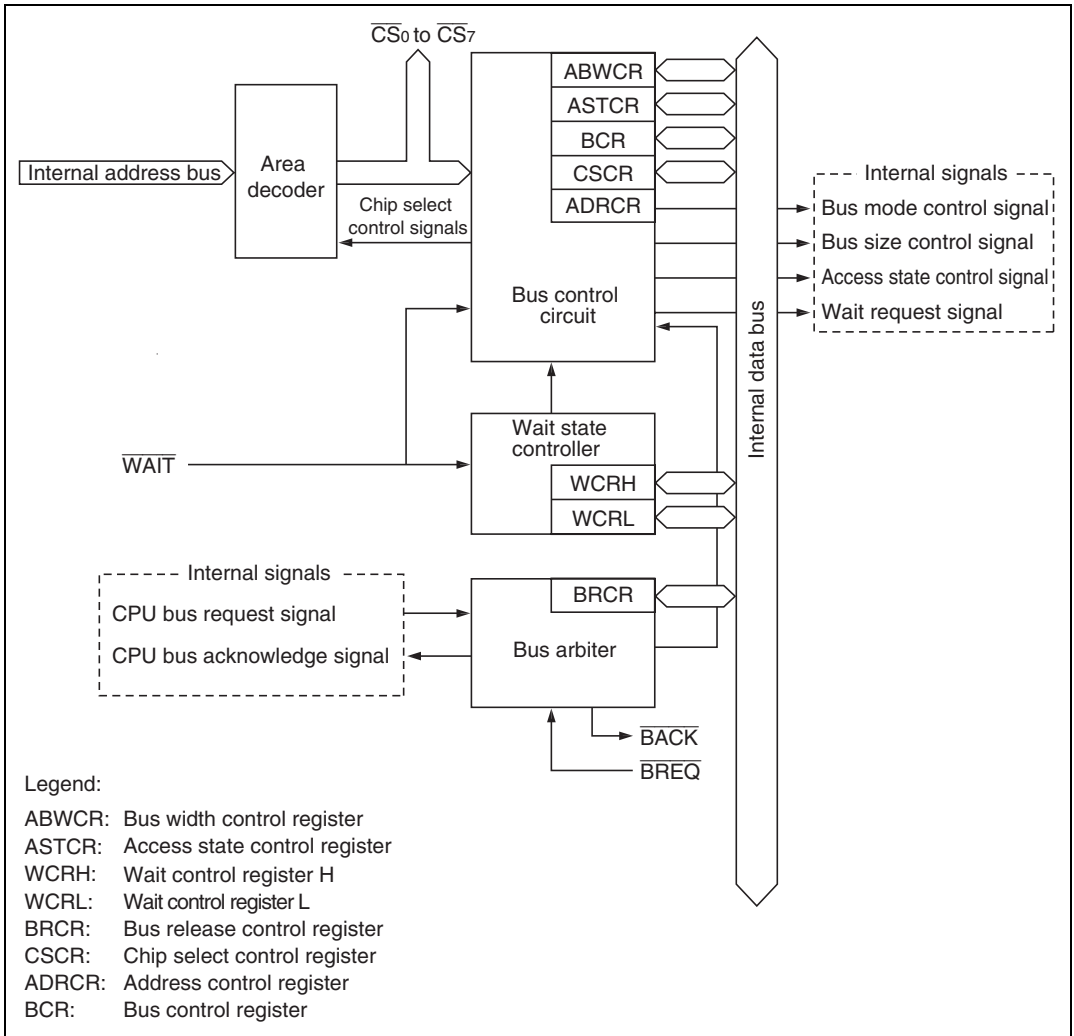


Figure 6.1 Block Diagram of Bus Controller

6.1.3 Pin Configuration

Table 6.1 summarizes the input/output pins of the bus controller.

Table 6.1 Bus Controller Pins

| Name | Abbreviation | I/O | Function |
|--------------------|--|--------|---|
| Chip select 0 to 7 | \overline{CS}_0 to \overline{CS}_7 | Output | Strobe signals selecting areas 0 to 7 |
| Address strobe | \overline{AS} | Output | Strobe signal indicating valid address output on the address bus |
| Read | \overline{RD} | Output | Strobe signal indicating reading from the external address space |
| High write | \overline{HWR} | Output | Strobe signal indicating writing to the external address space, with valid data on the upper data bus (D_{15} to D_8) |
| Low write | \overline{LWR} | Output | Strobe signal indicating writing to the external address space, with valid data on the lower data bus (D_7 to D_0) |
| Wait | \overline{WAIT} | Input | Wait request signal for access to external three-state access areas |
| Bus request | \overline{BREQ} | Input | Request signal for releasing the bus to an external device |
| Bus acknowledge | \overline{BACK} | Output | Acknowledge signal indicating release of the bus to an external device |

6.1.4 Register Configuration

Table 6.2 summarizes the bus controller's registers.

Table 6.2 Bus Controller Registers

| Address* ¹ | Name | Abbreviation | R/W | Initial Value |
|-----------------------|-------------------------------|--------------|-----|--------------------|
| H'EE020 | Bus width control register | ABWCR | R/W | H'FF* ² |
| H'EE021 | Access state control register | ASTCR | R/W | H'FF |
| H'EE022 | Wait control register H | WCRH | R/W | H'FF |
| H'EE023 | Wait control register L | WCRL | R/W | H'FF |
| H'EE013 | Bus release control register | BRCR | R/W | H'FE* ³ |
| H'EE01F | Chip select control register | CSCR | R/W | H'0F |
| H'EE01E | Address control register | ADRCR | R/W | H'FF |
| H'EE024 | Bus control register | BCR | R/W | H'C6 |

Notes: 1. Lower 20 bits of the address in advanced mode.

2. In modes 2 and 4, the initial value is H'00.

3. In modes 3 and 4, the initial value is H'EE.

6.2 Register Descriptions

6.2.1 Bus Width Control Register (ABWCR)

ABWCR is an 8-bit readable/writable register that selects 8-bit or 16-bit access for each area.

| Bit | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|------------------|---------------|------|------|------|------|------|------|------|------|
| | | ABW7 | ABW6 | ABW5 | ABW4 | ABW3 | ABW2 | ABW1 | ABW0 |
| Modes 1 and 3 | Initial value | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Modes 2 and 4 | Initial value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

When ABWCR contains H'FF (selecting 8-bit access for all areas), the chip operates in 8-bit bus mode: the upper data bus (D_{15} to D_8) is valid, and port 4 is an input/output port. When at least one bit is cleared to 0 in ABWCR, the chip operates in 16-bit bus mode with a 16-bit data bus (D_{15} to D_0). In modes 1 and 3, ABWCR is initialized to H'FF by a reset and in hardware standby mode. In modes 2 and 4, ABWCR is initialized to H'00 by a reset and in hardware standby mode. It is not initialized in software standby mode.

Bits 7 to 0—Area 7 to 0 Bus Width Control (ABW7 to ABW0): These bits select 8-bit access or 16-bit access for the corresponding areas.

Bits 7 to 0

ABW7 to ABW0 Description

| | |
|---|--------------------------------------|
| 0 | Areas 7 to 0 are 16-bit access areas |
| 1 | Areas 7 to 0 are 8-bit access areas |

ABWCR specifies the data bus width of external memory areas. The data bus width of on-chip memory and registers is fixed, and does not depend on ABWCR settings.

6.2.2 Access State Control Register (ASTCR)

ASTCR is an 8-bit readable/writable register that selects whether each area is accessed in two states or three states.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|------|------|------|------|------|------|------|------|
| | AST7 | AST6 | AST5 | AST4 | AST3 | AST2 | AST1 | AST0 |
| Initial value | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Bits selecting number of states for access to each area

ASTCR is initialized to H'FF by a reset and in hardware standby mode. It is not initialized in software standby mode.

Bits 7 to 0—Area 7 to 0 Access State Control (AST7 to AST0): These bits select whether the corresponding area is accessed in two or three states.

Bits 7 to 0

AST7 to AST0 Description

| | |
|---|---|
| 0 | Areas 7 to 0 are accessed in two states |
| 1 | Areas 7 to 0 are accessed in three states (Initial value) |

ASTCR specifies the number of states in which external areas are accessed. On-chip memory and registers are accessed in a fixed number of states that does not depend on ASTCR settings.

6.2.3 Wait Control Registers H and L (WCRH, WCRL)

WCRH and WCRL are 8-bit readable/writable registers that select the number of program wait states for each area.

On-chip memory and registers are accessed in a fixed number of states that does not depend on WCRH/WCRL settings.

WCRH and WCRL are initialized to H'FF by a reset and in hardware standby mode. They are not initialized in software standby mode.

WCRH

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|-----|-----|-----|-----|-----|-----|-----|-----|
| | W71 | W70 | W61 | W60 | W51 | W50 | W41 | W40 |
| Initial value | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Bits 7 and 6—Area 7 Wait Control 1 and 0 (W71, W70): These bits select the number of program wait states when area 7 in external space is accessed while the AST7 bit in ASTCR is set to 1.

| Bit 7 W71 | Bit 6 W70 | Description |
|--------------|--------------|--|
| 0 | 0 | Program wait not inserted when external space area 7 is accessed |
| | 1 | 1 program wait state inserted when external space area 7 is accessed |
| 1 | 0 | 2 program wait states inserted when external space area 7 is accessed |
| | 1 | 3 program wait states inserted when external space area 7 is accessed (Initial value) |

Bits 5 and 4—Area 6 Wait Control 1 and 0 (W61, W60): These bits select the number of program wait states when area 6 in external space is accessed while the AST6 bit in ASTCR is set to 1.

| Bit 5 W61 | Bit 4 W60 | Description |
|--------------|--------------|--|
| 0 | 0 | Program wait not inserted when external space area 6 is accessed |
| | 1 | 1 program wait state inserted when external space area 6 is accessed |
| 1 | 0 | 2 program wait states inserted when external space area 6 is accessed |
| | 1 | 3 program wait states inserted when external space area 6 is accessed (Initial value) |

Bits 3 and 2—Area 5 Wait Control 1 and 0 (W51, W50): These bits select the number of program wait states when area 5 in external space is accessed while the AST5 bit in ASTCR is set to 1.

| Bit 3 W51 | Bit 2 W50 | Description |
|--------------|--------------|--|
| 0 | 0 | Program wait not inserted when external space area 5 is accessed |
| | 1 | 1 program wait state inserted when external space area 5 is accessed |
| 1 | 0 | 2 program wait states inserted when external space area 5 is accessed |
| | 1 | 3 program wait states inserted when external space area 5 is accessed (Initial value) |

Bits 1 and 0—Area 4 Wait Control 1 and 0 (W41, W40): These bits select the number of program wait states when area 4 in external space is accessed while the AST4 bit in ASTCR is set to 1.

| Bit 1 W41 | Bit 0 W40 | Description |
|--------------|--------------|--|
| 0 | 0 | Program wait not inserted when external space area 4 is accessed |
| | 1 | 1 program wait state inserted when external space area 4 is accessed |
| 1 | 0 | 2 program wait states inserted when external space area 4 is accessed |
| | 1 | 3 program wait states inserted when external space area 4 is accessed (Initial value) |

WCRL

| | | | | | | | | |
|---------------|-----|-----|-----|-----|-----|-----|-----|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | W31 | W30 | W21 | W20 | W11 | W10 | W01 | W00 |
| Initial value | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Bits 7 and 6—Area 3 Wait Control 1 and 0 (W31, W30): These bits select the number of program wait states when area 3 in external space is accessed while the AST3 bit in ASTCR is set to 1.

| Bit 7 W31 | Bit 6 W30 | Description |
|--------------|--------------|--|
| 0 | 0 | Program wait not inserted when external space area 3 is accessed |
| | 1 | 1 program wait state inserted when external space area 3 is accessed |
| 1 | 0 | 2 program wait states inserted when external space area 3 is accessed |
| | 1 | 3 program wait states inserted when external space area 3 is accessed (Initial value) |

Bits 5 and 4—Area 2 Wait Control 1 and 0 (W21, W20): These bits select the number of program wait states when area 2 in external space is accessed while the AST2 bit in ASTCR is set to 1.

| Bit 5 W21 | Bit 4 W20 | Description |
|--------------|--------------|--|
| 0 | 0 | Program wait not inserted when external space area 2 is accessed |
| | 1 | 1 program wait state inserted when external space area 2 is accessed |
| 1 | 0 | 2 program wait states inserted when external space area 2 is accessed |
| | 1 | 3 program wait states inserted when external space area 2 is accessed (Initial value) |

Bits 3 and 2—Area 1 Wait Control 1 and 0 (W11, W10): These bits select the number of program wait states when area 1 in external space is accessed while the AST1 bit in ASTCR is set to 1.

| Bit 3 W11 | Bit 2 W10 | Description |
|--------------|--------------|--|
| 0 | 0 | Program wait not inserted when external space area 1 is accessed |
| | 1 | 1 program wait state inserted when external space area 1 is accessed |
| 1 | 0 | 2 program wait states inserted when external space area 1 is accessed |
| | 1 | 3 program wait states inserted when external space area 1 is accessed (Initial value) |

Bits 1 and 0—Area 0 Wait Control 1 and 0 (W01, W00): These bits select the number of program wait states when area 0 in external space is accessed while the AST0 bit in ASTCR is set to 1.

| Bit 1 W01 | Bit 0 W00 | Description |
|--------------|--------------|--|
| 0 | 0 | Program wait not inserted when external space area 0 is accessed |
| | 1 | 1 program wait state inserted when external space area 0 is accessed |
| 1 | 0 | 2 program wait states inserted when external space area 0 is accessed |
| | 1 | 3 program wait states inserted when external space area 0 is accessed (Initial value) |

6.2.4 Bus Release Control Register (BRCR)

BRCR is an 8-bit readable/writable register that enables address output on bus lines A_{23} to A_{20} and enables or disables release of the bus to an external device.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|---------------|------|------|------|---|---|---|------|
| | A23E | A22E | A21E | A20E | — | — | — | BRLE |
| Modes 1 and 2 | Initial value | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| | Read/Write | — | — | — | — | — | — | R/W |
| Modes 3 and 4 | Initial value | 1 | 1 | 0 | 1 | 1 | 1 | 0 |
| | Read/Write | R/W | R/W | R/W | — | — | — | R/W |

Address 23 to 20 enable
These bits enable PA_7 to PA_4 to be used for A_{23} to A_{20} address output

Reserved bits

Bus release enable
Enables or disables release of the bus to an external device

BRCR is initialized to H'FE in modes 1 and 2, and to H'EE in modes 3 and 4, by a reset and in hardware standby mode. It is not initialized in software standby mode.

Bit 7—Address 23 Enable (A23E): Enables PA_4 to be used as the A_{23} address output pin. Writing 0 in this bit enables A_{23} output from PA_4 . In modes other than 3 and 4, this bit cannot be modified and PA_4 has its ordinary port functions.

Bit 7

| A23E | Description |
|------|---|
| 0 | PA_4 is the A_{23} address output pin |
| 1 | PA_4 is an input/output pin (Initial value) |

Bit 6—Address 22 Enable (A22E): Enables PA_5 to be used as the A_{22} address output pin. Writing 0 in this bit enables A_{22} output from PA_5 . In modes other than 3 and 4, this bit cannot be modified and PA_5 has its ordinary port functions.

Bit 6

| A22E | Description |
|------|---|
| 0 | PA_5 is the A_{22} address output pin |
| 1 | PA_5 is an input/output pin (Initial value) |

Bit 5—Address 21 Enable (A21E): Enables PA₆ to be used as the A₂₁ address output pin. Writing 0 in this bit enables A₂₁ output from PA₆. In modes other than 3 and 4, this bit cannot be modified and PA₆ has its ordinary port functions.

| Bit 5 A21E | Description |
|---------------|---|
| 0 | PA ₆ is the A ₂₁ address output pin |
| 1 | PA ₆ is an input/output pin (Initial value) |

Bit 4—Address 20 Enable (A20E): Enables PA₇ to be used as an address output pin. When 0 is written to this bit, PA₇ functions as address output A₂₀. In modes 3 and 4, PA₇ functions as an address output pin, and in modes 1 and 2, as a normal port pin.

| Bit 4 A20E | Description |
|---------------|--|
| 0 | PA ₇ is the A ₂₀ address output pin (In mode 3 or 4) |
| 1 | PA ₇ is an input/output pin (In mode 1 or 2) |

Bits 3 to 1—Reserved: These bits cannot be modified and are always read as 1.

Bit 0—Bus Release Enable (BRLE): Enables or disables release of the bus to an external device.

| Bit 0 BRLE | Description |
|---------------|---|
| 0 | The bus cannot be released to an external device BRE \bar{Q} and BACK can be used as input/output pins (Initial value) |
| 1 | The bus can be released to an external device |

6.2.5 Bus Control Register (BCR)

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|-------|-------|-----|-----|-----|-----|------|-------|
| | ICIS1 | ICIS0 | — | — | — | — | RDEA | WAITE |
| Initial value | 1 | 1 | 0*1 | 0*1 | 0*1 | 1*2 | 1 | 0 |
| Read/Write | R/W | R/W | — | — | — | — | R/W | R/W |

- Notes: 1. 1 must not be written in bits 5 to 3.
2. 0 must not be written in bit 2.

BCR is an 8-bit readable/writable register that enables or disables idle cycle insertion, selects the area division unit, selects the extended memory map, and enables or disables WAIT pin input.

BCR is initialized to H'C6 by a reset and in hardware standby mode. It is not initialized in software standby mode.

Bit 7—Idle Cycle Insertion 1 (ICIS1): Selects whether one idle cycle state is to be inserted between bus cycles in case of consecutive external read cycles for different areas.

| Bit 7 ICIS1 | Description |
|----------------|--|
| 0 | No idle cycle inserted in case of consecutive external read cycles for different areas |
| 1 | Idle cycle inserted in case of consecutive external read cycles for different areas (Initial value) |

Bit 6—Idle Cycle Insertion 0 (ICIS0): Selects whether one idle cycle state is to be inserted between bus cycles in case of consecutive external read and write cycles.

| Bit 6 ICIS0 | Description |
|----------------|--|
| 0 | No idle cycle inserted in case of consecutive external read and write cycles |
| 1 | Idle cycle inserted in case of consecutive external read and write cycles (Initial value) |

Bits 5 to 3—Reserved (must not be set to 1): These bits can be read and written, but must not be set to 1. Normal operation cannot be guaranteed if 1 is written in these bits.

Bit 2—Reserved (must not be set to 0): This bit can be read and written, but must not be set to 0. Normal operation cannot be guaranteed if 0 is written in this bit.

Bit 1—Area Division Unit Select (RDEA): Selects the memory map area division units. This bit is valid in modes 3 and 4, and is invalid in modes 1 and 2.

| Bit 1 RDEA | Description | | | | | | | | |
|------------------|--|------------------|---------------------|------------------|------------------|------------------|----------------------|------------------|------------------|
| 0 | Area divisions are as follows: <table style="margin-left: 40px; border: none;"> <tr> <td style="padding-right: 20px;">Area 0: 2 Mbytes</td> <td>Area 4: 1.93 Mbytes</td> </tr> <tr> <td>Area 1: 2 Mbytes</td> <td>Area 5: 4 kbytes</td> </tr> <tr> <td>Area 2: 8 Mbytes</td> <td>Area 6: 23.75 kbytes</td> </tr> <tr> <td>Area 3: 2 Mbytes</td> <td>Area 7: 22 bytes</td> </tr> </table> | Area 0: 2 Mbytes | Area 4: 1.93 Mbytes | Area 1: 2 Mbytes | Area 5: 4 kbytes | Area 2: 8 Mbytes | Area 6: 23.75 kbytes | Area 3: 2 Mbytes | Area 7: 22 bytes |
| Area 0: 2 Mbytes | Area 4: 1.93 Mbytes | | | | | | | | |
| Area 1: 2 Mbytes | Area 5: 4 kbytes | | | | | | | | |
| Area 2: 8 Mbytes | Area 6: 23.75 kbytes | | | | | | | | |
| Area 3: 2 Mbytes | Area 7: 22 bytes | | | | | | | | |
| 1 | Areas 0 to 7 are the same size (2 Mbytes) (Initial value) | | | | | | | | |

Bit 0—WAIT Pin Enable (WAITE): Enables or disables wait insertion by means of the $\overline{\text{WAIT}}$ pin.

| Bit 0 WAITE | Description |
|----------------|---|
| 0 | $\overline{\text{WAIT}}$ pin wait input is disabled, and the $\overline{\text{WAIT}}$ pin can be used as an input/output port (Initial value) |
| 1 | $\overline{\text{WAIT}}$ pin wait input is enabled |

6.2.6 Chip Select Control Register (CSCR)

CSCR is an 8-bit readable/writable register that enables or disables output of chip select signals (\overline{CS}_7 to \overline{CS}_4).

If output of a chip select signal \overline{CS}_7 to \overline{CS}_4 is enabled by a setting in this register, the corresponding pin functions a chip select signal (\overline{CS}_7 to \overline{CS}_4) output regardless of any other settings.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|------|------|------|------|---|---|---|---|
| | CS7E | CS6E | CS5E | CS4E | — | — | — | — |
| Initial value | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| Read/Write | R/W | R/W | R/W | R/W | — | — | — | — |

|
Chip select 7 to 4 enable
 These bits enable or disable
 chip select signal output

|
Reserved bits

CSCR is initialized to H'0F by a reset and in hardware standby mode. It is not initialized in software standby mode.

Bits 7 to 4—Chip Select 7 to 4 Enable (CS7E to CS4E): These bits enable or disable output of the corresponding chip select signal.

| Bit n | Description |
|-------|--|
| CSnE | |
| 0 | Output of chip select signal \overline{CS}_n is disabled (Initial value) |
| 1 | Output of chip select signal \overline{CS}_n is enabled |

Note: n = 7 to 4

Bits 3 to 0—Reserved: These bits cannot be modified and are always read as 1.

6.2.7 Address Control Register (ADRCR)

ADRCR is an 8-bit readable/writable register that selects either address update mode 1 or address update mode 2 as the address output method.

| | | | | | | | | |
|---------------|---|---|---|---|---|---|---|--------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | ADRCTL |
| Initial value | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Read/Write | — | — | — | — | — | — | — | R/W |

Reserved bits
Address control
Selects address update mode 1 or address update mode 2

ADRCR is initialized to H'FF by a reset and in hardware standby mode. It is not initialized in software standby mode.

Bits 7 to 1—Reserved: Read-only bits, always read as 1.

Bit 0—Address Control (ADRCTL): Selects the address output method.

| Bit 0 ADRCTL | Description |
|-----------------|---|
| 0 | Address update mode 2 is selected |
| 1 | Address update mode 1 is selected (Initial value) |

6.3 Operation

6.3.1 Area Division

The external address space is divided into areas 0 to 7. Each area has a size of 128 kbytes in the 1-Mbyte modes, or 2 Mbytes in the 16-Mbyte modes. Figure 6.2 shows a general view of the memory map.

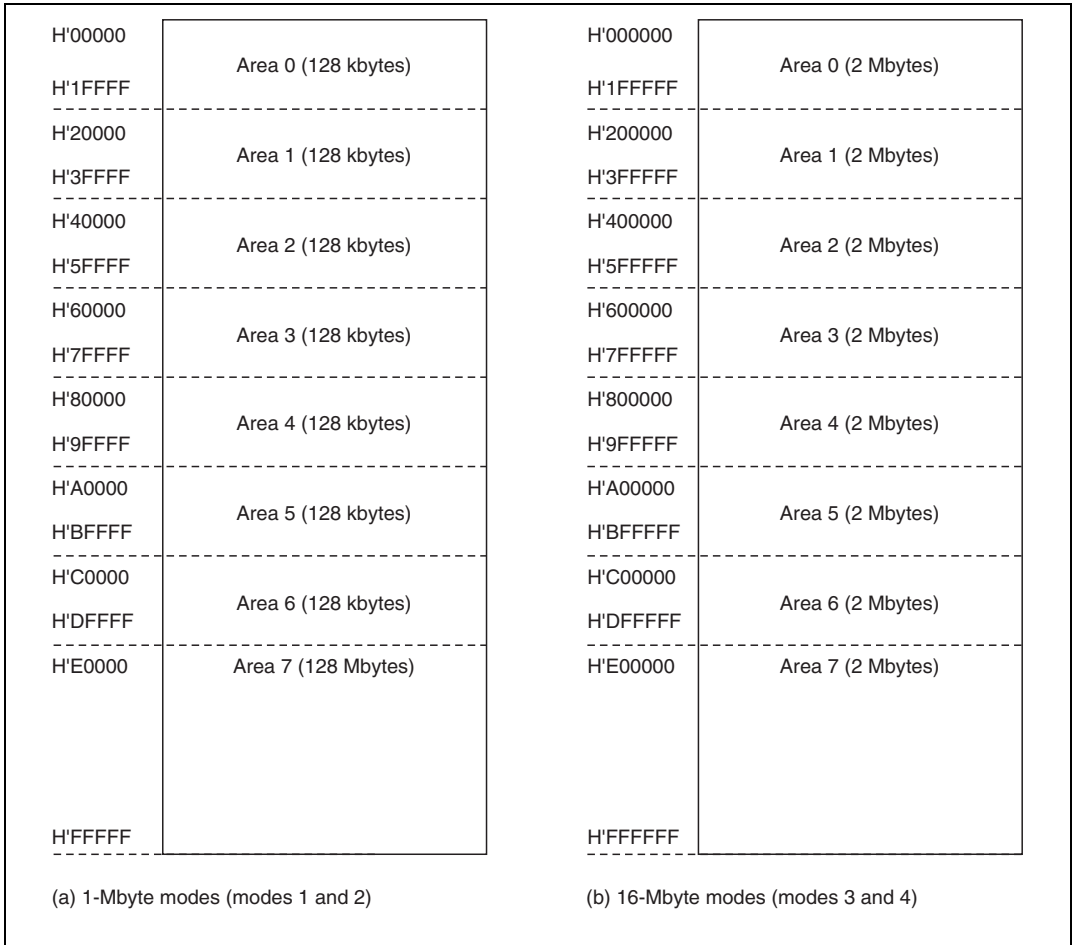
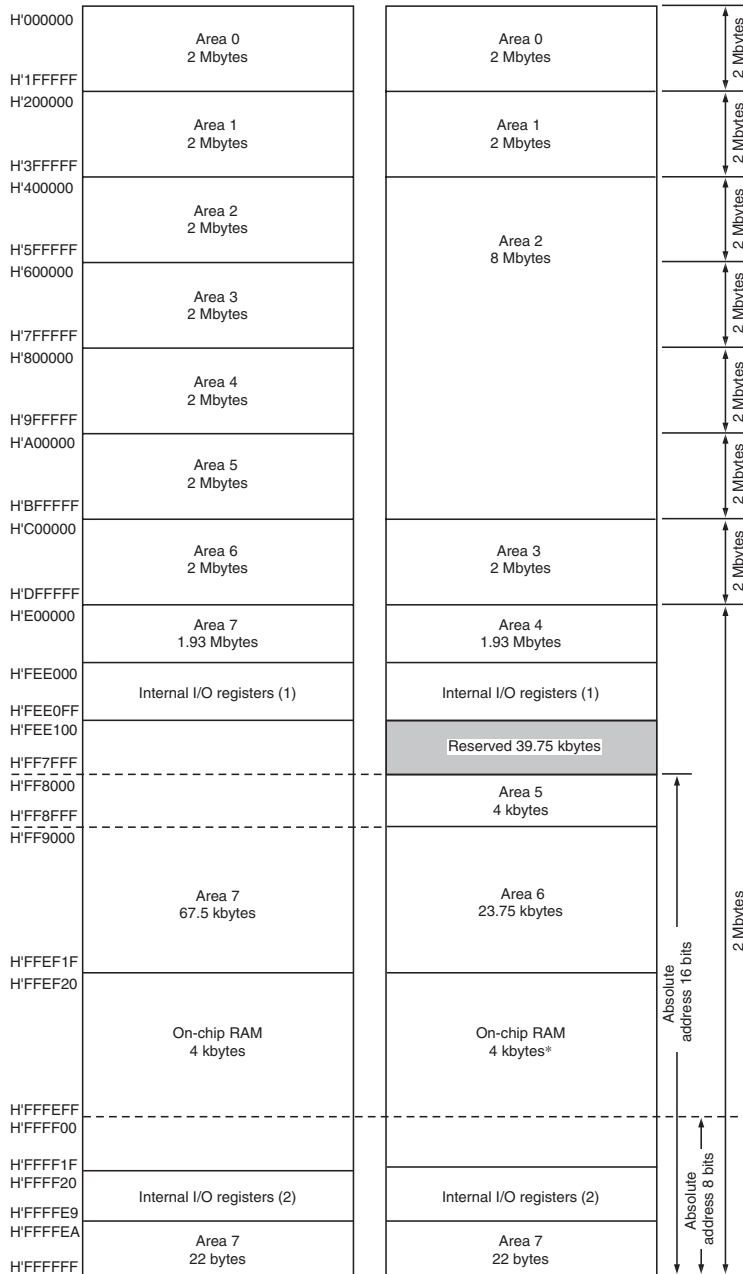


Figure 6.2 Access Area Map for Each Operating Mode

Chip select signals (\overline{CS}_0 to \overline{CS}_7) can be output for areas 0 to 7. The bus specifications for each area are selected in ABWCR, ASTCR, WCRH, and WCRL.

In 16-Mbyte mode, the area division units can be selected with the RDEA bit in BCR.



Note: * Area 6 when the RAME bit is cleared.

Figure 6.3 Memory Map in 16-Mbyte Mode

6.3.2 Bus Specifications

The external space bus specifications consist of three elements: bus width, number of access states, and number of program wait states.

The bus width and number of access states for on-chip memory and internal I/O registers are fixed, and are not affected by the bus controller.

Bus Width: A bus width of 8 or 16 bits can be selected with ABWCR. An area for which an 8-bit bus is selected functions as an 8-bit access space, and an area for which a 16-bit bus is selected functions as a 16-bit access space.

If all areas are designated for 8-bit access, 8-bit bus mode is set; if any area is designated for 16-bit access, 16-bit bus mode is set.

Number of Access States: Two or three access states can be selected with ASTCR. An area for which two-state access is selected functions as a two-state access space, and an area for which three-state access is selected functions as a three-state access space.

When two-state access space is designated, wait insertion is disabled.

Number of Program Wait States: When three-state access space is designated in ASTCR, the number of program wait states to be inserted automatically is selected with WCRH and WCRL. From 0 to 3 program wait states can be selected.

Table 6.3 shows the bus specifications for each basic bus interface area.

Table 6.3 Bus Specifications for Each Area (Basic Bus Interface)

| ABWCR | | ASTCR | | WCRH/WCRL | | Bus Specifications (Basic Bus Interface) | | |
|-------|------|-------|-----|-----------|---------------|--|--|--|
| ABWn | ASTn | Wn1 | Wn0 | Bus Width | Access States | Program Wait States | | |
| 0 | 0 | — | — | 16 | 2 | 0 | | |
| | 1 | 0 | 0 | | 3 | 0 | | |
| | | | 1 | | 1 | | | |
| | 1 | 0 | 0 | | 2 | | | |
| | | | 1 | | 3 | | | |
| 1 | 0 | — | — | 8 | 2 | 0 | | |
| | 1 | 0 | 0 | | 3 | 0 | | |
| | | | 1 | | 1 | | | |
| | 1 | 0 | 0 | | 2 | | | |
| | | | 1 | | 3 | | | |

Note: n = 0 to 7

6.3.3 Memory Interfaces

As its memory interface, the H8/3008 has only a basic bus interface that allows direct connection of ROM, SRAM, and so on. It is not possible to select a DRAM interface that allows direct connection of DRAM, or a burst ROM interface that allows direct connection of burst ROM.

6.3.4 Chip Select Signals

For each of areas 0 to 7, the H8/3008 can output a chip select signal (\overline{CS}_0 to \overline{CS}_7) that goes low when the corresponding area is selected in expanded mode. Figure 6.4 shows the output timing of a \overline{CS}_n signal.

Output of \overline{CS}_0 to \overline{CS}_3 : Output of \overline{CS}_0 to \overline{CS}_3 is enabled or disabled in the data direction register (DDR) of the corresponding port.

In the expanded modes with on-chip ROM disabled, a reset leaves pin \overline{CS}_0 in the output state and pins \overline{CS}_1 to \overline{CS}_3 in the input state. To output chip select signals \overline{CS}_1 to \overline{CS}_3 , the corresponding DDR bits must be set to 1. In the expanded modes with on-chip ROM enabled, a reset leaves pins \overline{CS}_0 to \overline{CS}_3 in the input state. To output chip select signals \overline{CS}_0 to \overline{CS}_3 , the corresponding DDR bits must be set to 1. For details, see section 7, I/O Ports.

Output of \overline{CS}_4 to \overline{CS}_7 : Output of \overline{CS}_4 to \overline{CS}_7 is enabled or disabled in the chip select control register (CSCR). A reset leaves pins \overline{CS}_4 to \overline{CS}_7 in the input state. To output chip select signals \overline{CS}_4 to \overline{CS}_7 , the corresponding CSCR bits must be set to 1. For details, see section 7, I/O Ports.

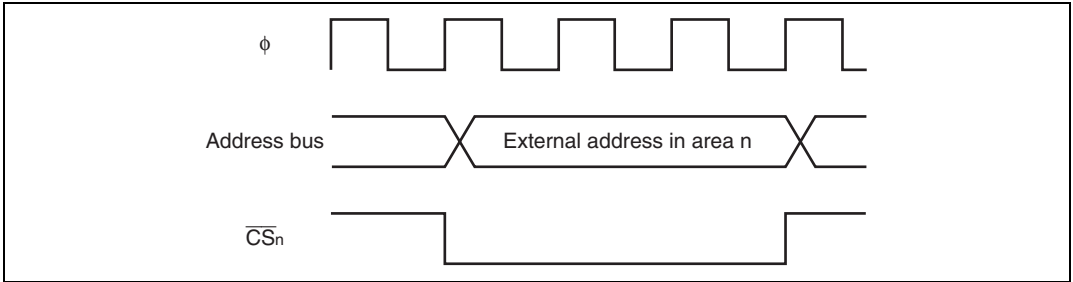


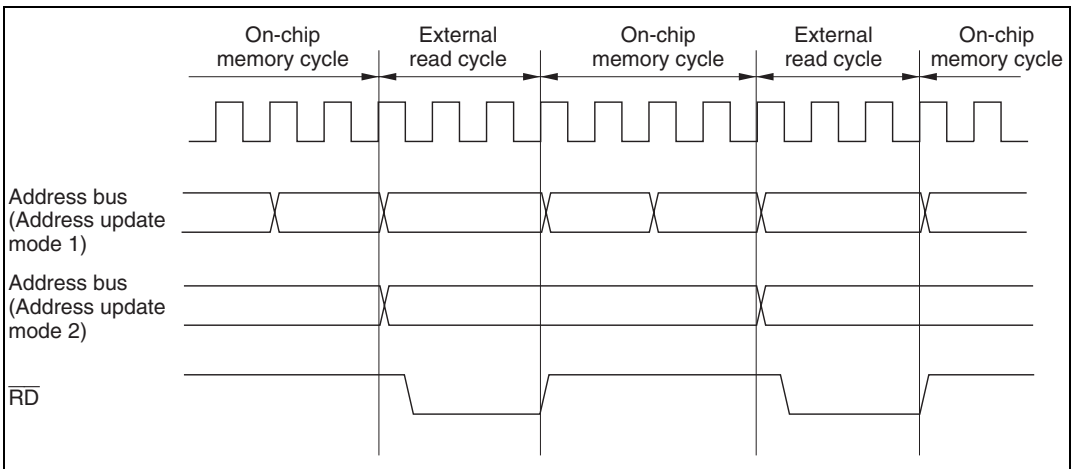
Figure 6.4 \overline{CS}_n Signal Output Timing (n = 0 to 7)

When the on-chip ROM, on-chip RAM, and internal I/O registers are accessed, \overline{CS}_0 to \overline{CS}_7 remain high. The \overline{CS}_n signals are decoded from the address signals. They can be used as chip select signals for SRAM and other devices.

6.3.5 Address Output Method

The H8/3008 provides a choice of two address update methods: either the same method as in the previous H8/300H Series (address update mode 1), or a method in which address updating is restricted to external space accesses (address update mode 2).

Figure 6.5 shows examples of address output in these two update modes.



**Figure 6.5 Sample Address Output in Each Address Update Mode
(Basic Bus Interface, 3-State Space)**

Address Update Mode 1: Address update mode 1 is compatible with the previous H8/300H Series. Addresses are always updated between bus cycles.

Address Update Mode 2: In address update mode 2, address updating is performed only in external space accesses. In this mode, the address can be retained between an external space read cycle and an instruction fetch cycle (on-chip memory) by placing the program in on-chip memory. Address update mode 2 is therefore useful when connecting a device that requires address hold time with respect to the rise of the \overline{RD} strobe.

Switching between address update modes 1 and 2 is performed by means of the ADRCTL bit in ADRCR. The initial value of ADRCR is the address update mode 1 setting, providing compatibility with the previous H8/300H Series.

Cautions: The address output methods are designed so that the initial state with the bit selection method is compatible with the H8/3062F-ZTAT (HD64F3062) (i.e. address update mode 1). However, the following points should be noted.

- ADRCR is allocated to address H'FEE01E. In the H8/3062F-ZTAT, the corresponding address is empty space, but it is necessary to confirm that no accesses are made to H'FEE01E in the program.
- When address update mode 2 is selected, the address in an internal space (on-chip memory or internal I/O) access cycle is not output externally.
- In order to secure address holding with respect to the rise of \overline{RD} , when address update mode 2 is used an external space read access must be completed within a single access cycle. For example, in a word access to 8-bit access space, the bus cycle is split into two as shown in figure 6.6, and so there is not a single access cycle. In this case, address holding is not guaranteed at the rise of \overline{RD} between the first (even address) and second (odd address) access cycles (area inside the ellipse in the figure).

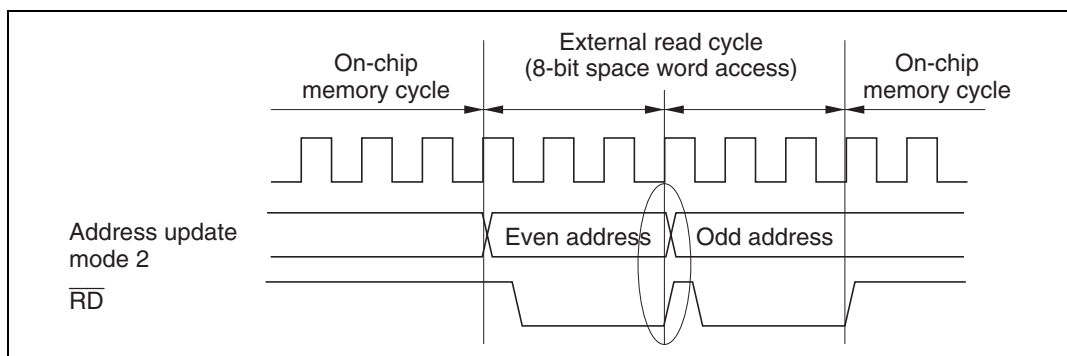


Figure 6.6 Example of Consecutive External Space Accesses in Address Update Mode 2

6.4 Basic Bus Interface

6.4.1 Overview

The basic bus interface enables direct connection of ROM, SRAM, and so on.

The bus specifications can be selected with ABWCR, ASTCR, WCRH, and WCRL (see table 6.3).

6.4.2 Data Size and Data Alignment

Data sizes for the CPU and other internal bus masters are byte, word, and longword. The bus controller has a data alignment function, and when accessing external space, controls whether the upper data bus (D_{15} to D_8) or lower data bus (D_7 to D_0) is used according to the bus specifications for the area being accessed (8-bit access area or 16-bit access area) and the data size.

8-Bit Access Areas: Figure 6.7 illustrates data alignment control for 8-bit access space. With 8-bit access space, the upper data bus (D_{15} to D_8) is always used for accesses. The amount of data that can be accessed at one time is one byte: a word access is performed as two byte accesses, and a longword access, as four byte accesses.

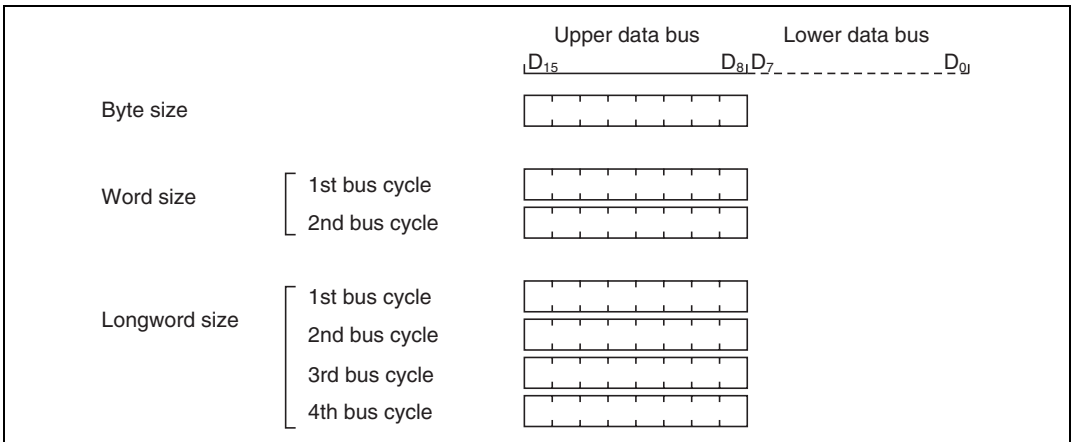


Figure 6.7 Access Sizes and Data Alignment Control (8-Bit Access Area)

16-Bit Access Areas: Figure 6.8 illustrates data alignment control for 16-bit access areas. With 16-bit access areas, the upper data bus (D_{15} to D_8) and lower data bus (D_7 to D_0) are used for accesses. The amount of data that can be accessed at one time is one byte or one word, and a longword access is executed as two word accesses.

In byte access, whether the upper or lower data bus is used is determined by whether the address is even or odd. The upper data bus is used for an even address, and the lower data bus for an odd address.

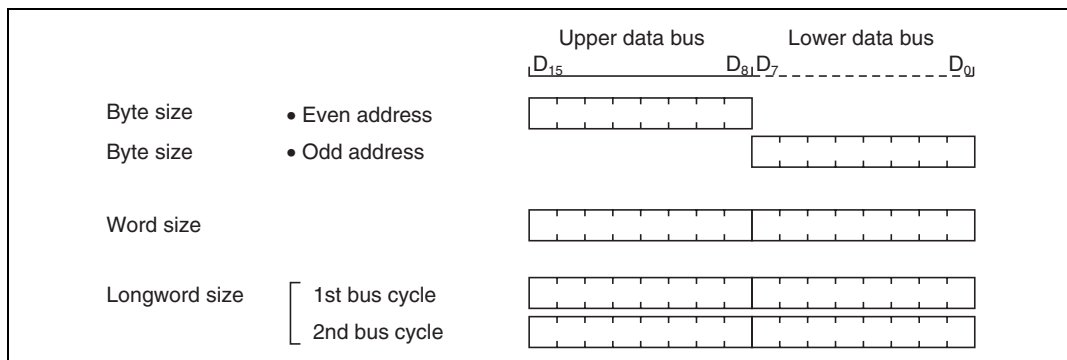


Figure 6.8 Access Sizes and Data Alignment Control (16-Bit Access Area)

6.4.3 Valid Strobes

Table 6.4 shows the data buses used, and the valid strobes, for the access spaces.

In a read, the \overline{RD} signal is valid for both the upper and the lower half of the data bus.

In a write, the \overline{HWR} signal is valid for the upper half of the data bus, and the \overline{LWR} signal for the lower half.

Table 6.4 Data Buses Used and Valid Strobes

| Area | Access Size | Read/Write | Address | Valid Strobe | Upper Data Bus (D ₁₅ to D ₈) | Lower Data Bus (D ₇ to D ₀) |
|--------------------|-------------|------------|---------|------------------|---|--|
| 8-bit access area | Byte | Read | — | \overline{RD} | Valid | Invalid |
| | | Write | — | \overline{HWR} | | Undetermined data |
| 16-bit access area | Byte | Read | Even | \overline{RD} | Valid | Invalid |
| | | | Odd | | Invalid | Valid |
| | | Write | Even | \overline{HWR} | Valid | Undetermined data |
| | | | Odd | \overline{LWR} | Undetermined data | Valid |
| | Word | Read | — | \overline{RD} | Valid | Valid |
| | | Write | — | | \overline{HWR} , \overline{LWR} | Valid |

Notes: 1. Undetermined data means that unpredictable data is output.
2. Invalid means that the bus is in the input state and the input is ignored.

6.4.4 Memory Areas

The initial state of each area is basic bus interface, three-state access space. The initial bus width is selected according to the operating mode.

Areas 0 to 6: In the H8/3008, the entire space of areas 0 to 6 is external space.

When area 0 to 6 external space is accessed, the \overline{CS}_0 to \overline{CS}_6 pin signals respectively can be output. The size of areas 0 to 6 is 128 kbytes in modes 1 and 2, and 2 Mbytes in modes 3 and 4.

Area 7: Area 7 includes the on-chip RAM and internal I/O registers. In the H8/3008, the space excluding the on-chip RAM and I/O registers is external space. The on-chip RAM is enabled when the RAME bit in the system control register (SYSCR) is set to 1; when the RAME bit is cleared to 0, the on-chip RAM is disabled and the corresponding space becomes external space .

When area 7 external space is accessed, the \overline{CS}_7 signal can be output.

The size of area 7 is 128 kbytes in modes 1 and 2, and 2 Mbytes in modes 3 and 4.

6.4.5 Basic Bus Control Signal Timing

8-Bit, Three-State-Access Areas: Figure 6.9 shows the timing of bus control signals for an 8-bit, three-state-access area. The upper data bus (D_{15} to D_8) is used in accesses to these areas. The \overline{LWR} pin is always high. Wait states can be inserted.

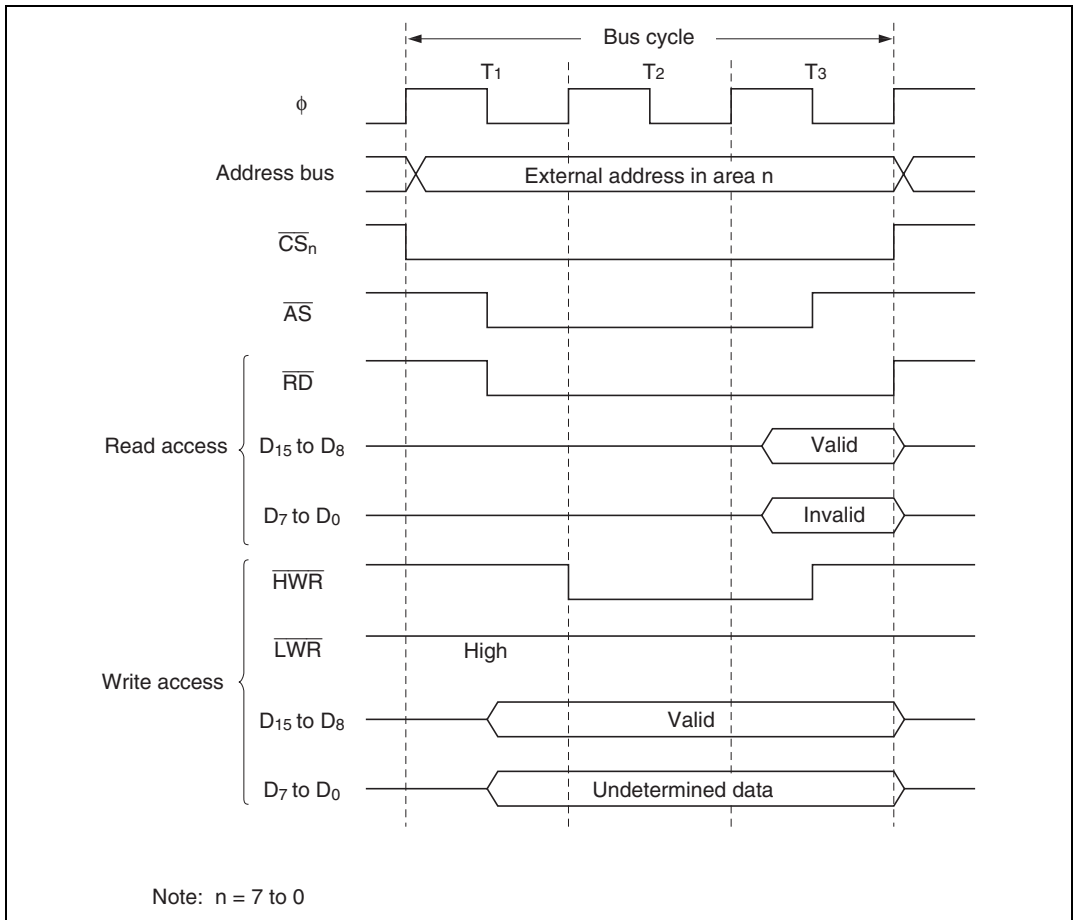


Figure 6.9 Bus Control Signal Timing for 8-Bit, Three-State-Access Area

8-Bit, Two-State-Access Areas: Figure 6.10 shows the timing of bus control signals for an 8-bit, two-state-access area. The upper data bus (D_{15} to D_8) is used in accesses to these areas. The \overline{LWR} pin is always high. Wait states cannot be inserted.

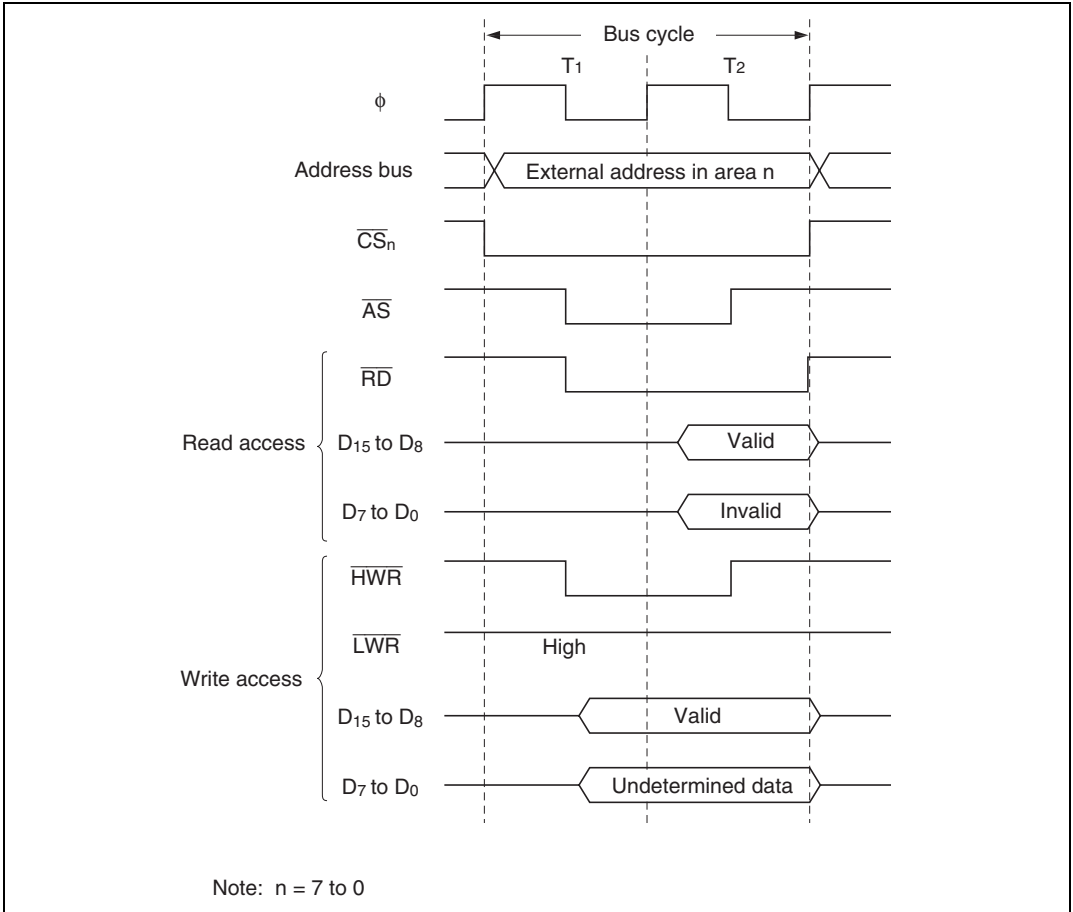


Figure 6.10 Bus Control Signal Timing for 8-Bit, Two-State-Access Area

16-Bit, Three-State-Access Areas: Figures 6.11 to 6.13 show the timing of bus control signals for a 16-bit, three-state-access area. In these areas, the upper data bus (D_{15} to D_8) is used in accesses to even addresses and the lower data bus (D_7 to D_0) in accesses to odd addresses. Wait states can be inserted.

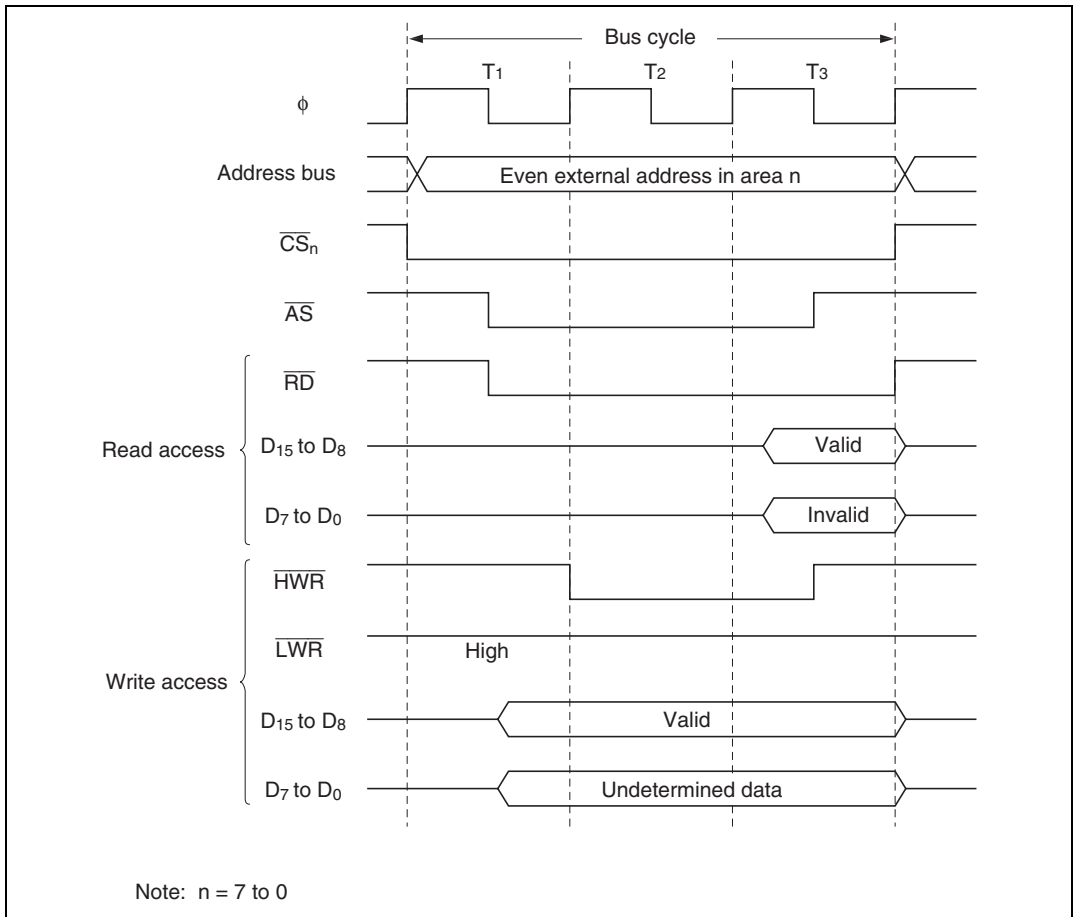
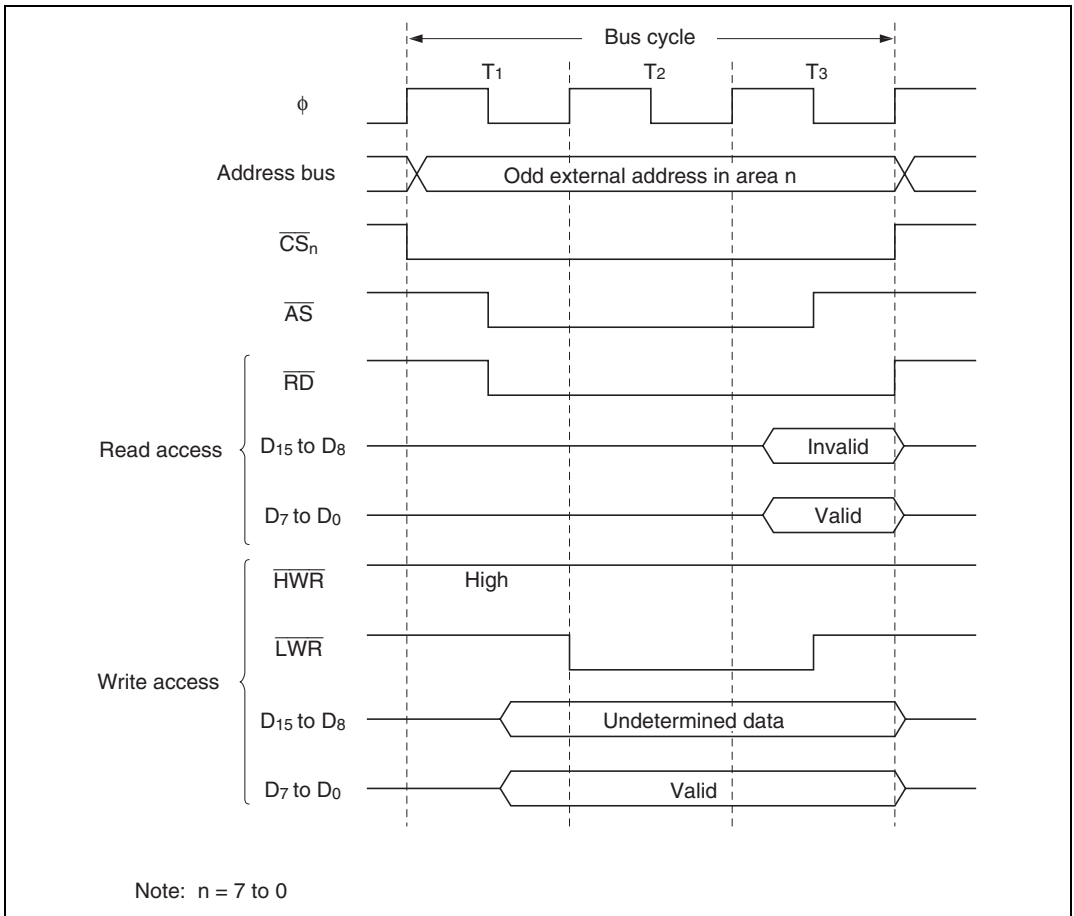


Figure 6.11 Bus Control Signal Timing for 16-Bit, Three-State-Access Area (1)
(Byte Access to Even Address)



**Figure 6.12 Bus Control Signal Timing for 16-Bit, Three-State-Access Area (2)
(Byte Access to Odd Address)**

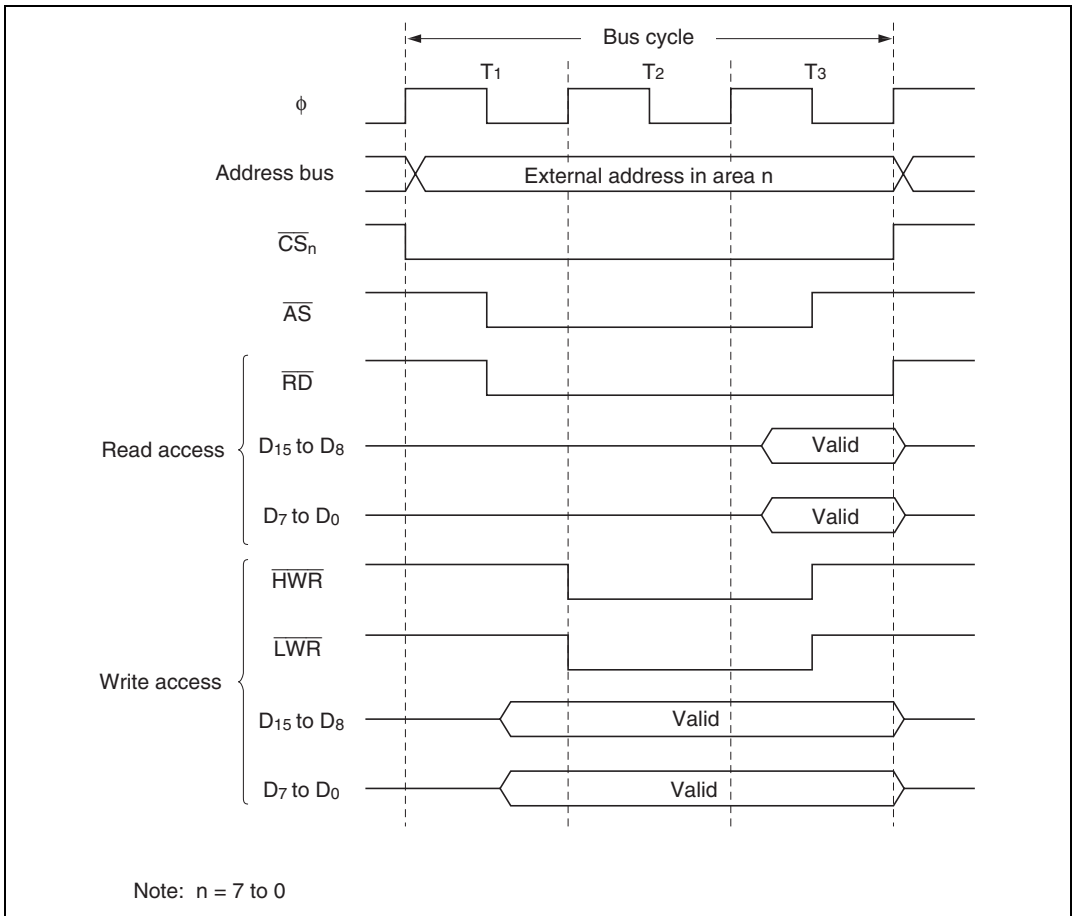
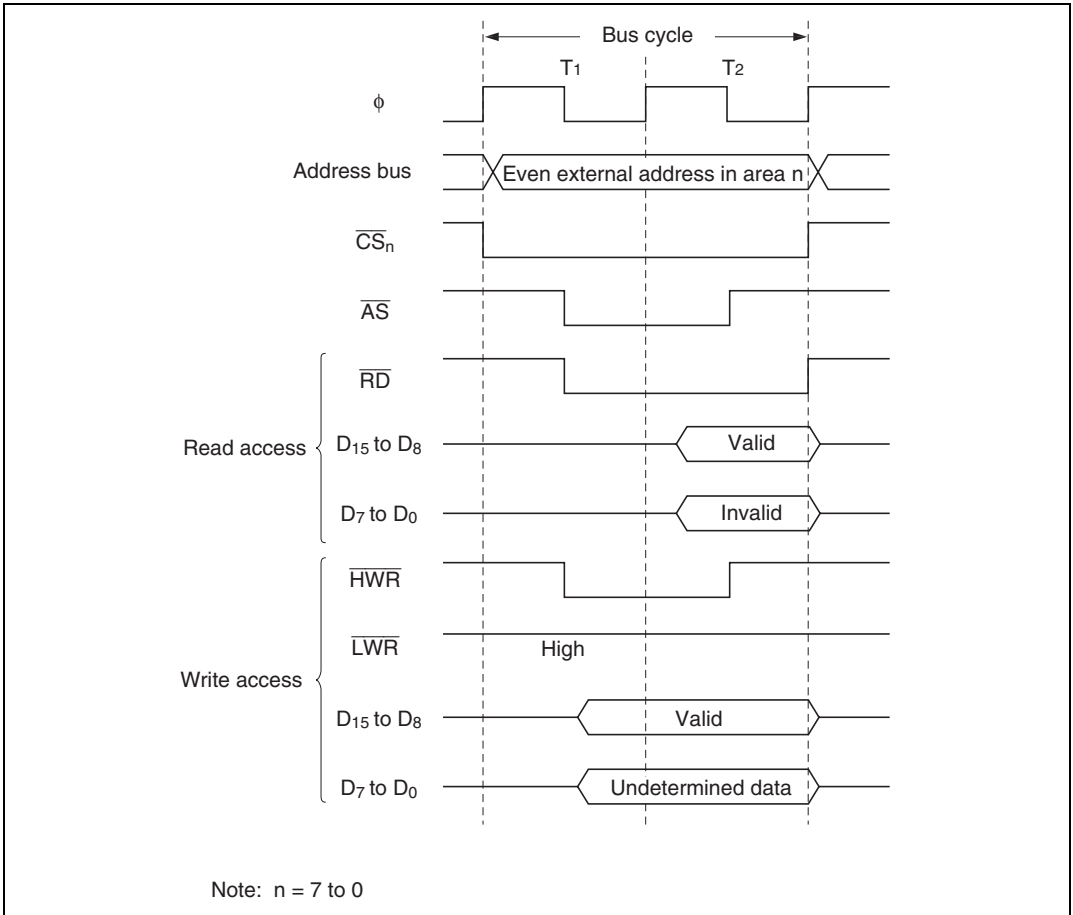
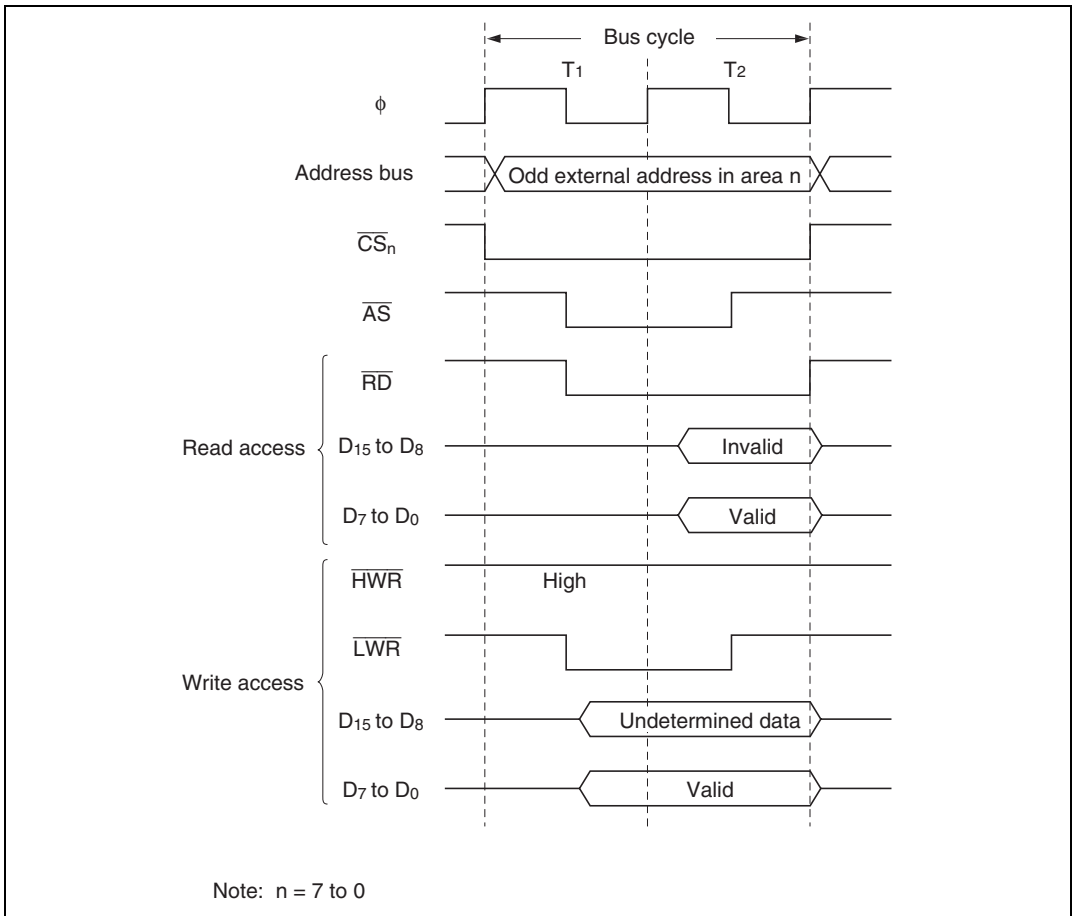


Figure 6.13 Bus Control Signal Timing for 16-Bit, Three-State-Access Area (3) (Word Access)

16-Bit, Two-State-Access Areas: Figures 6.14 to 6.16 show the timing of bus control signals for a 16-bit, two-state-access area. In these areas, the upper data bus (D_{15} to D_8) is used in accesses to even addresses and the lower data bus (D_7 to D_0) in accesses to odd addresses. Wait states cannot be inserted.



**Figure 6.14 Bus Control Signal Timing for 16-Bit, Two-State-Access Area (1)
(Byte Access to Even Address)**



**Figure 6.15 Bus Control Signal Timing for 16-Bit, Two-State-Access Area (2)
(Byte Access to Odd Address)**

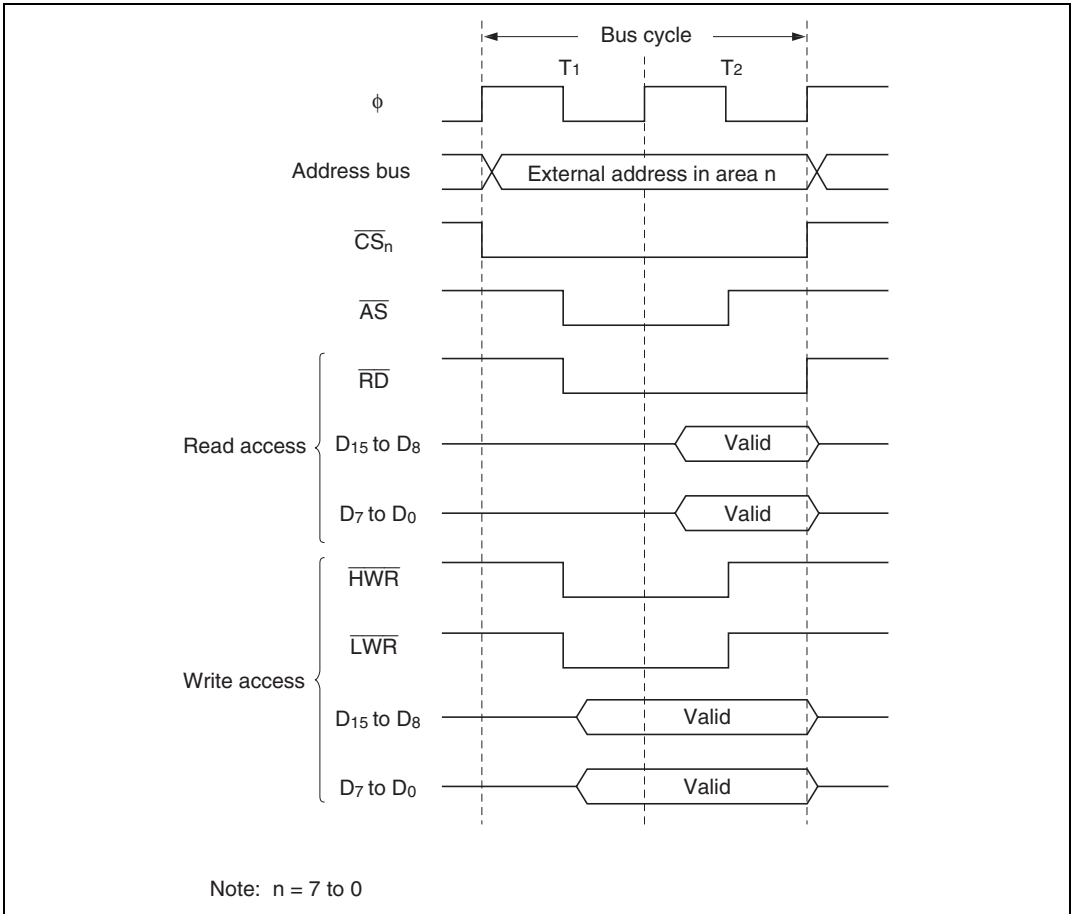


Figure 6.16 Bus Control Signal Timing for 16-Bit, Two-State-Access Area (3) (Word Access)

6.4.6 Wait Control

When accessing external space, the H8/3008 can extend the bus cycle by inserting wait states (T_w). There are two ways of inserting wait states: program wait insertion and pin wait insertion using the WAIT pin.

Program Wait Insertion: From 0 to 3 wait states can be inserted automatically between the T_2 state and T_3 state on an individual area basis in three-state access space, according to the settings of WCRH and WCRL.

Pin Wait Insertion: Setting the WAITE bit in BCR to 1 enables wait insertion by means of the $\overline{\text{WAIT}}$ pin. When external space is accessed in this state, a program wait is first inserted. If the $\overline{\text{WAIT}}$ pin is low at the falling edge of ϕ in the last T_2 or T_w state, another T_w state is inserted. If the $\overline{\text{WAIT}}$ pin is held low, T_w states are inserted until it goes high.

This is useful when inserting four or more T_w states, or when changing the number of T_w states for different external devices.

The WAITE bit setting applies to all areas.

Figure 6.17 shows an example of the timing for insertion of one program wait state in 3-state space.

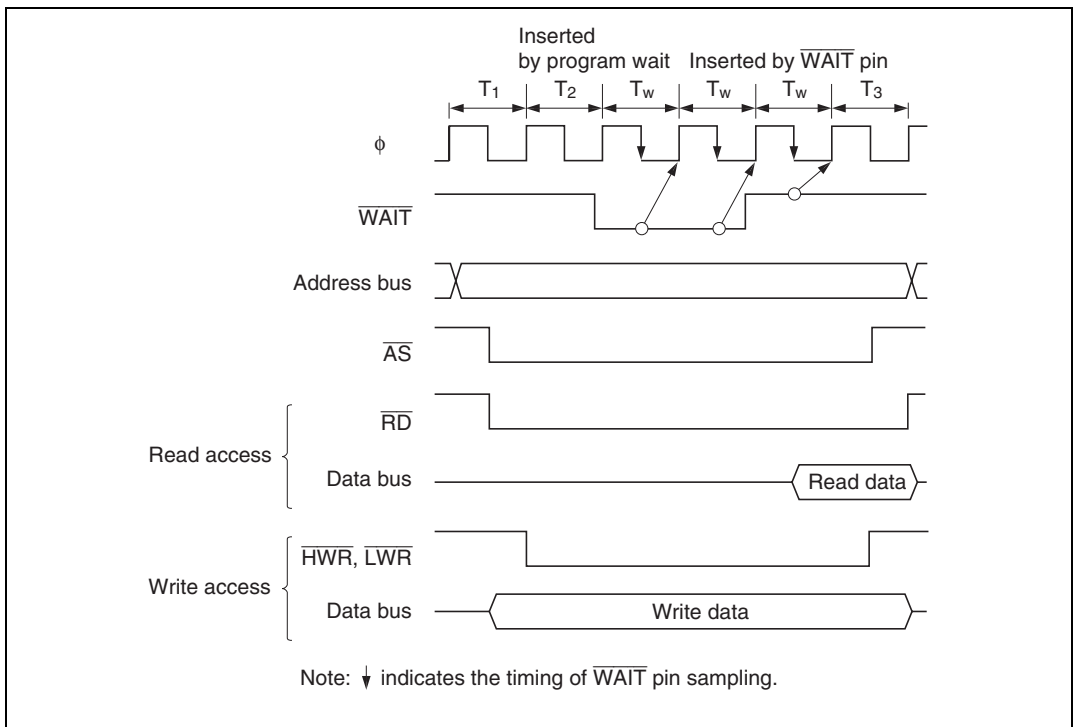


Figure 6.17 Example of Wait State Insertion Timing

6.5 Idle Cycle

6.5.1 Operation

When the H8/3008 chip accesses external space, it can insert a 1-state idle cycle (T_i) between bus cycles in the following cases: when read accesses between different areas occur consecutively, and when a write cycle occurs immediately after a read cycle. By inserting an idle cycle it is possible, for example, to avoid data collisions between ROM, which has a long output floating time, and high-speed memory, I/O interfaces, and so on.

The initial value of the ICIS1 and ICIS0 bits in BCR is 1, so that idle cycle insertion is performed in the initial state. If there are no data collisions, the ICIS bits can be cleared.

Consecutive Reads between Different Areas: If consecutive reads between different areas occur while the ICIS1 bit is set to 1 in BCR, an idle cycle is inserted at the start of the second read cycle.

Figure 6.18 shows an example of the operation in this case. In this example, bus cycle A is a read cycle from ROM with a long output floating time, and bus cycle B is a read cycle from SRAM, each being located in a different area. In (a), an idle cycle is not inserted, and a collision occurs in bus cycle B between the read data from ROM and that from SRAM. In (b), an idle cycle is inserted, and a data collision is prevented.

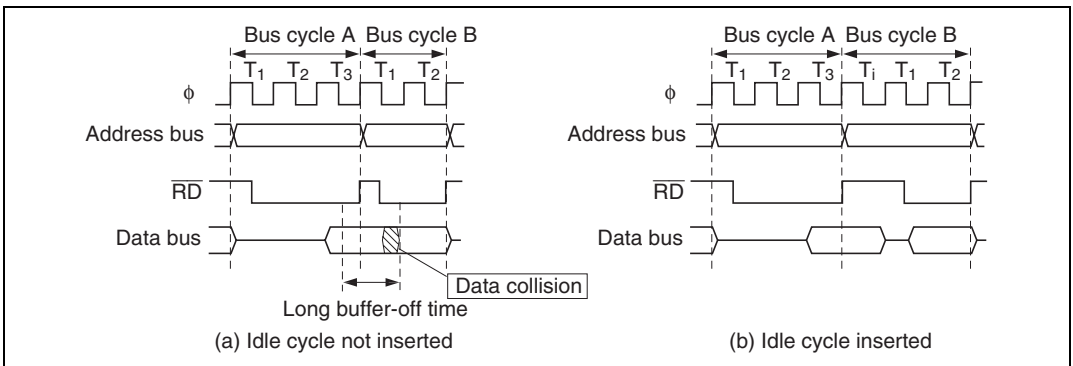


Figure 6.18 Example of Idle Cycle Operation (ICIS1 = 1)

Write after Read: If an external write occurs after an external read while the ICIS0 bit is set to 1 in BCR, an idle cycle is inserted at the start of the write cycle.

Figure 6.19 shows an example of the operation in this case. In this example, bus cycle A is a read cycle from ROM with a long output floating time, and bus cycle B is a CPU write cycle.

In (a), an idle cycle is not inserted, and a collision occurs in bus cycle B between the read data from ROM and the CPU write data. In (b), an idle cycle is inserted, and a data collision is prevented.

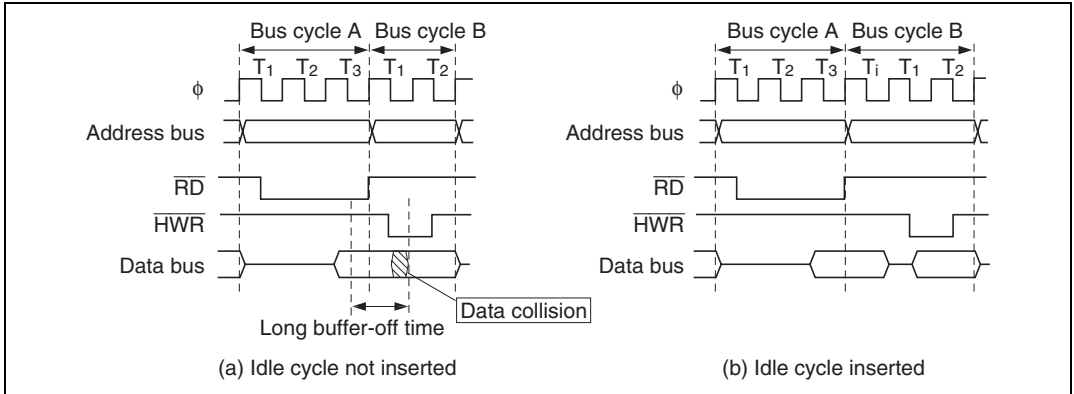


Figure 6.19 Example of Idle Cycle Operation (ICIS0 = 1)

Usage Note: When non-insertion of an idle cycle is specified, the rise (negation) of \overline{RD} and fall (assertion) of \overline{CS}_n may occur simultaneously. Figure 6.20 shows an example of the operation in this case.

If consecutive reads to a different external area occur while the ICIS1 bit in BCR is cleared to 0, or if an external read is followed by a write cycle for a different external area while the ICIS0 bit is cleared to 0, negation of \overline{RD} in the first read cycle and assertion of \overline{CS}_n in the following bus cycle will occur simultaneously. Depending on the output delay time of each signal, therefore, it is possible that the \overline{RD} low output in the previous read cycle and the \overline{CS}_n low output in the following bus cycle will overlap.

As long as \overline{RD} and \overline{CS}_n do not change simultaneously, or if there is no problem even if they do, non-insertion of an idle cycle can be specified.

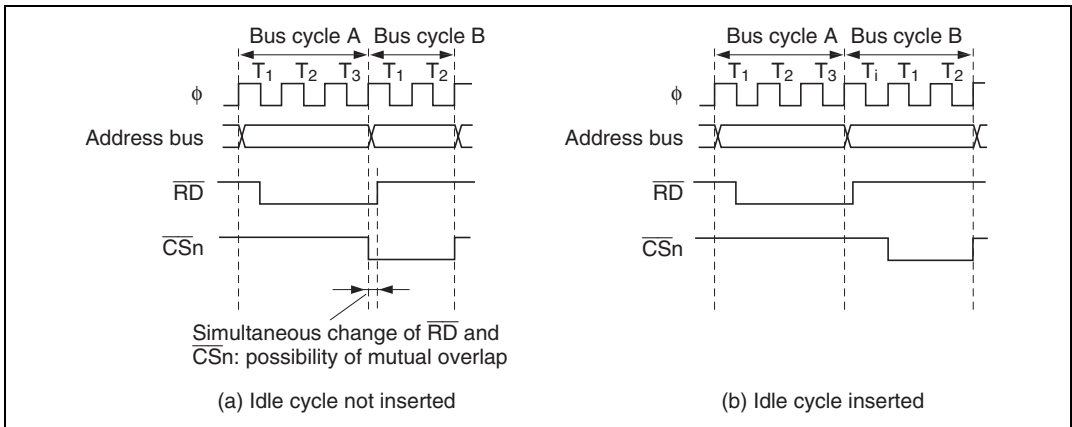


Figure 6.20 Example of Idle Cycle Operation

6.5.2 Pin States in Idle Cycle

Table 6.5 shows the pin states in an idle cycle.

Table 6.5 Pin States in Idle Cycle

| Pins | Pin State |
|-------------------|--------------------------|
| A_{23} to A_0 | Next cycle address value |
| D_{15} to D_0 | High impedance |
| \overline{CSn} | High |
| \overline{AS} | High |
| \overline{RD} | High |
| \overline{HWR} | High |
| \overline{LWR} | High |

6.6 Bus Arbiter

The bus controller has a built-in bus arbiter that arbitrates between different bus masters. The bus master can be either the CPU or an external bus master. When a bus master has the bus right it can carry out read and write operations. Each bus master uses a bus request signal to request the bus right. At fixed times the bus arbiter determines priority and uses a bus acknowledge signal to grant the bus to a bus master, which can then operate using the bus.

The bus arbiter checks whether the bus request signal from a bus master is active or inactive, and returns an acknowledge signal to the bus master. When two or more bus masters request the bus, the highest-priority bus master receives an acknowledge signal. The bus master that receives an acknowledge signal can continue to use the bus until the acknowledge signal is deactivated.

The bus master priority order is:

(High) External bus master > CPU (Low)

The bus arbiter samples the bus request signals and determines priority at all times, but it does not always grant the bus immediately, even when it receives a bus request from a bus master with higher priority than the current bus master. Each bus master has certain times at which it can release the bus to a higher-priority bus master.

6.6.1 Operation

CPU: The CPU is the lowest-priority bus master. If an external bus master requests the bus while the CPU has the bus right, the bus arbiter transfers the bus right to the bus master that requested it. The bus right is transferred at the following times:

- The bus right is transferred at the boundary of a bus cycle. If word data is accessed by two consecutive byte accesses, however, the bus right is not transferred between the two byte accesses.
- If another bus master requests the bus while the CPU is performing internal operations, such as executing a multiply or divide instruction, the bus right is transferred immediately. The CPU continues its internal operations.
- If another bus master requests the bus while the CPU is in sleep mode, the bus right is transferred immediately.

External Bus Master: When the BRLE bit is set to 1 in BRCR, the bus can be released to an external bus master. The external bus master has highest priority, and requests the bus right from the bus arbiter driving the $\overline{\text{BREQ}}$ signal low. Once the external bus master acquires the bus, it keeps the bus until the $\overline{\text{BREQ}}$ signal goes high. While the bus is released to an external bus master, the H8/3008 chip holds the address bus, data bus, bus control signals ($\overline{\text{AS}}$, $\overline{\text{RD}}$, $\overline{\text{HWR}}$, and $\overline{\text{LWR}}$), and chip select signals ($\overline{\text{CS}}_n$: $n = 7$ to 0) in the high-impedance state, and holds the $\overline{\text{BACK}}$ pin in the low output state.

The bus arbiter samples the $\overline{\text{BREQ}}$ pin at the rise of the system clock (ϕ). If $\overline{\text{BREQ}}$ is low, the bus is released to the external bus master at the appropriate opportunity. The $\overline{\text{BREQ}}$ signal should be held low until the $\overline{\text{BACK}}$ signal goes low.

When the $\overline{\text{BREQ}}$ pin is high in two consecutive samples, the $\overline{\text{BACK}}$ pin is driven high to end the bus-release cycle.

Figure 6.21 shows the timing when the bus right is requested by an external bus master during a read cycle in a two-state access area. There is a minimum interval of three states from when the $\overline{\text{BREQ}}$ goes low until the bus is released.

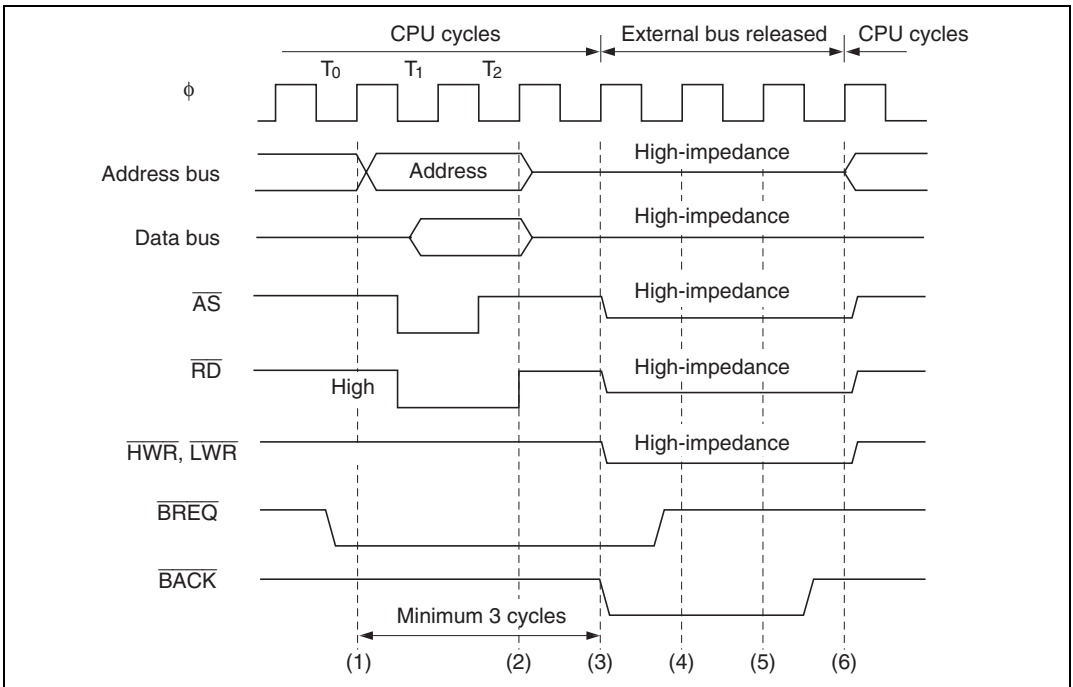


Figure 6.21 Example of External Bus Master Operation

When making a transition to software standby mode, if there is contention with a bus request from an external bus master, the $\overline{\text{BACK}}$ and strobe states may be indefinite when the transition is made.

When using software standby mode, clear the BRLE bit to 0 in BRCR before executing the SLEEP instruction.

6.7 Register and Pin Input Timing

6.7.1 Register Write Timing

ABWCR, ASTCR, WCRH, and WCRL Write Timing: Data written to ABWCR, ASTCR, WCRH, and WCRL takes effect starting from the next bus cycle. Figure 6.22 shows the timing when an instruction fetched from area 0 changes area 0 from three-state access to two-state access.

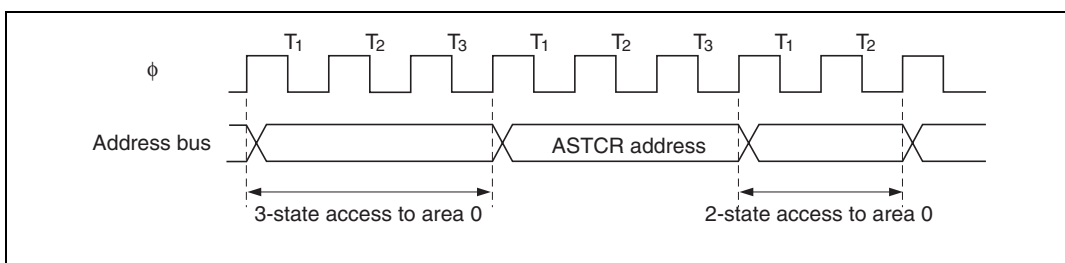


Figure 6.22 ASTCR Write Timing

DDR and CSCR Write Timing: Data written to DDR or CSCR for the port corresponding to the $\overline{\text{CS}}_n$ pin to switch between $\overline{\text{CS}}_n$ output and generic input takes effect starting from the T_3 state of the DDR write cycle. Figure 6.23 shows the timing when the $\overline{\text{CS}}_1$ pin is changed from generic input to $\overline{\text{CS}}_1$ output.

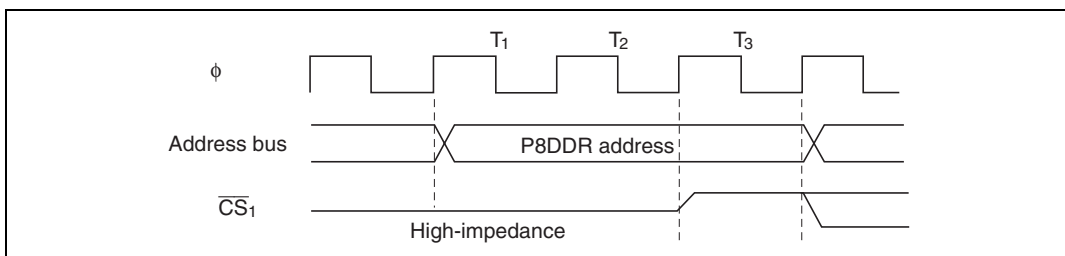


Figure 6.23 DDR Write Timing

BRCR Write Timing: Data written to BRCR to switch between A_{23} , A_{22} , A_{21} , or A_{20} output and generic input or output takes effect starting from the T_3 state of the BRCR write cycle. Figure 6.24 shows the timing when a pin is changed from generic input to A_{23} , A_{22} , A_{21} , or A_{20} output.

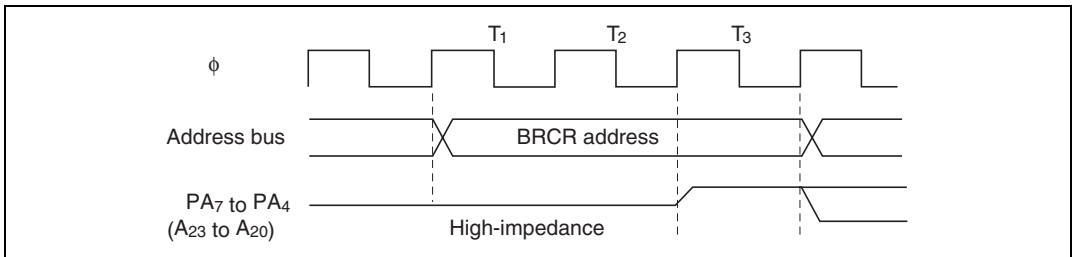


Figure 6.24 BRCR Write Timing

6.7.2 $\overline{\text{BREQ}}$ Pin Input Timing

After driving the $\overline{\text{BREQ}}$ pin low, hold it low until $\overline{\text{BACK}}$ goes low. If $\overline{\text{BREQ}}$ returns to the high level before $\overline{\text{BACK}}$ goes low, the bus arbiter may operate incorrectly.

To terminate the external-bus-released state, hold the $\overline{\text{BREQ}}$ signal high for at least three states. If $\overline{\text{BREQ}}$ is high for too short an interval, the bus arbiter may operate incorrectly.

Section 7 I/O Ports

7.1 Overview

The H8/3008 has six input/output ports (ports 4, 6, 8, 9, A, and B) and one input-only port (port 7). Table 7.1 summarizes the port functions. The pins in each port are multiplexed as shown in table 7.1.

Each port has a data direction register (DDR) for selecting input or output, and a data register (DR) for storing output data. In addition to these registers, port 4 has an input pull-up MOS control register (PCR) for switching input pull-up MOS transistors on and off.

Ports 4, 6, and 8 can drive one TTL load and a 90-pF capacitive load. Ports 9, A, and B can drive one TTL load and a 30-pF capacitive load. Ports 4, 6, 8, 9, A, and B can drive a darlington pair. Pins P8₂ to P8₀, PA₇ to PA₀ have Schmitt-trigger input circuits.

For block diagrams of the ports see appendix C, I/O Port Block Diagrams.

Table 7.1 Port Functions

| Port | Description | Pins | Expanded Modes | | | |
|--------|--|--|--|--------------------------------------|------------------------------------|--------|
| | | | Mode 1 | Mode 2 | Mode 3 | Mode 4 |
| Port 4 | <ul style="list-style-type: none"> 8-bit I/O port Built-in input pull-up MOS | P4 ₇ to P4 ₀ /D ₇ to D ₀ | Data input/output (D ₇ to D ₀) and 8-bit generic input/output | 8-bit bus mode: generic input/output | 16-bit bus mode: data input/output | |
| Port 6 | <ul style="list-style-type: none"> 8-bit I/O port | P6 ₇ /φ | Clock output (φ) and generic input | | | |
| | | P6 ₆ /LWR | Bus control signal output (LWR, HWR, RD, AS) | | | |
| | | P6 ₆ /HWR | | | | |
| | | P6 ₆ /RD | | | | |
| | | P6 ₆ /AS | | | | |
| Port 7 | <ul style="list-style-type: none"> 8-bit I/O port | P6 ₇ /BACK | Bus control signal input/output (BACK, BREQ, WAIT) and 3-bit generic input/output | | | |
| | | P6 ₇ /BREQ | | | | |
| | | P6 ₇ /WAIT | | | | |
| Port 7 | <ul style="list-style-type: none"> 8-bit I/O port | P7 ₇ /AN ₇ /DA ₁ | Analog input (AN ₇ , AN ₆) to A/D converter, analog output (DA ₁ , DA ₀) from D/A converter, and generic input | | | |
| | | P7 ₆ /AN ₆ /DA ₀ | | | | |
| Port 7 | <ul style="list-style-type: none"> 8-bit I/O port | P7 ₅ to P7 ₀ /AN ₅ to AN ₀ | Analog input (AN ₅ to AN ₀) to A/D converter, and generic input | | | |
| | | | | | | |

| Port | Description | Pins | Expanded Modes | | | |
|---|--|--|---|--------|-----------------------------------|--------|
| | | | Mode 1 | Mode 2 | Mode 3 | Mode 4 |
| Port 8 | <ul style="list-style-type: none"> 5-bit I/O port P8₂ to P8₀ have schmitt inputs | P8 ₄ / \overline{CS}_0 | DDR = 0: generic input | | | |
| | | | DDR = 1 (reset value): \overline{CS}_0 output | | | |
| | | P8 ₃ / \overline{IRQ}_3 / \overline{CS}_1 / ADTRG | \overline{IRQ}_3 input, \overline{CS}_1 output, external trigger input (\overline{ADTRG}) to A/D converter, and generic input | | | |
| | | | DDR = 0 (after reset): generic input | | | |
| | | | DDR = 1: \overline{CS}_1 output | | | |
| | | P8 ₂ / \overline{IRQ}_2 / \overline{CS}_2 P8 ₁ / \overline{IRQ}_1 / \overline{CS}_3 | \overline{IRQ}_2 and \overline{IRQ}_1 input, \overline{CS}_2 and \overline{CS}_3 output, and generic input | | | |
| | | | DDR = 0 (after reset): generic input | | | |
| | | | DDR = 1: \overline{CS}_2 and \overline{CS}_3 output | | | |
| | | P8 ₀ / \overline{IRQ}_0 | \overline{IRQ}_0 input, and generic input/output | | | |
| Port 9 | <ul style="list-style-type: none"> 6-bit I/O port | P9 ₅ / \overline{IRQ}_5 /SCK ₁ | Input and output (SCK ₁ , SCK ₀ , RxD ₁ , RxD ₀ , TxD ₁ , TxD ₀) for serial communication interfaces 1 and 0 (SCI1/0), \overline{IRQ}_5 and \overline{IRQ}_4 input, and 6-bit generic input/output | | | |
| | | P9 ₄ / \overline{IRQ}_4 /SCK ₀ | | | | |
| | | P9 ₃ /RxD ₁ | | | | |
| | | P9 ₂ /RxD ₀ | | | | |
| | | P9 ₁ /TxD ₁ | | | | |
| | | P9 ₀ /TxD ₀ | | | | |
| Port A | <ul style="list-style-type: none"> 8-bit I/O port Schmitt inputs | PA ₇ /TP ₇ / TIOCB ₂ /A ₂₀ | Output (TP ₇) from programmable timing pattern controller (TPC), input or output (TIOCB ₂) for 16-bit timer and generic input/output | | Address output (A ₂₀) | |
| | | PA ₆ /TP ₆ /TIOCA ₂ /A ₂₁ | TPC output (TP ₆ to TP ₁), 16-bit timer input and output (TIOCA ₂ , TIOCB ₁ , TIOCA ₁), and generic input/output | | | |
| | | PA ₅ /TP ₅ /TIOCB ₁ /A ₂₂ | | | | |
| | | PA ₄ /TP ₄ /TIOCA ₁ /A ₂₃ | | | | |
| | | PA ₃ /TP ₃ /TIOCB ₀ / TCLKD | TPC output (TP ₃ to TP ₀), 16-bit timer input and output (TIOCB ₀ , TIOCA ₀ , TCLKD, TCLKC, TCLKB, TCLKA), 8-bit timer input (TCLKD, TCLKC, TCLKB, TCLKA), and generic input/output | | | |
| PA ₂ /TP ₂ /TIOCA ₀ / TCLKC | | | | | | |
| PA ₁ /TP ₁ /TCLKB | | | | | | |
| PA ₀ /TP ₀ /TCLKA | | | | | | |

| Port | Description | Pins | Expanded Modes | | | |
|--------|------------------|--|---|--------|--------|--------|
| | | | Mode 1 | Mode 2 | Mode 3 | Mode 4 |
| Port B | • 8-bit I/O port | PB ₇ /TP ₁₅ PB ₆ /TP ₁₄ PB ₅ /TP ₁₃ PB ₄ /TP ₁₂ | TPC output (TP ₁₅ to TP ₁₂) and generic input/output | | | |
| | | PB ₃ /TP ₁₁ /TMIO ₃ / \overline{CS}_4 PB ₂ /TP ₁₀ /TMO ₂ / \overline{CS}_5 PB ₁ /TP ₉ /TMIO ₁ / \overline{CS}_6 PB ₀ /TP ₈ /TMO ₀ / \overline{CS}_7 | TPC output (TP ₁₁ to TP ₈), 8-bit timer input and output (TMIO ₃ , TMO ₂ , TMIO ₁ , TMO ₀), \overline{CS}_7 to \overline{CS}_4 output, and generic input/output | | | |

Legend:

SCIO: Serial communication interface channel 0

SCI1: Serial communication interface channel 1

TPC: Programmable timing pattern controller

16TIM: 16-bit timer

8TIM: 8-bit timer

7.2 Port 4

7.2.1 Overview

Port 4 is an 8-bit input/output port which also functions as a data bus. Its pin configuration is shown in figure 7.1. The pin functions differ depending on the operating mode.

In the H8/3008, when the bus width control register (ABWCR) designates areas 0 to 7 all as 8-bit-access areas, the chip operates in 8-bit bus mode and port 4 is a generic input/output port. When at least one of areas 0 to 7 is designated as a 16-bit-access area, the chip operates in 16-bit bus mode and port 4 becomes part of the data bus.

Port 4 has software-programmable built-in pull-up MOS.

Pins in port 4 can drive one TTL load and a 90-pF capacitive load. They can also drive a darlington transistor pair.

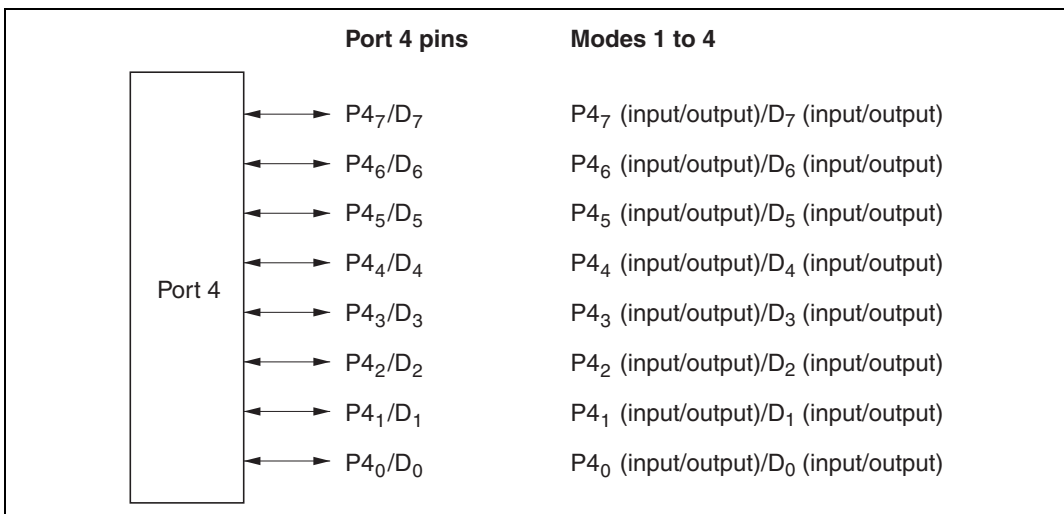


Figure 7.1 Port 4 Pin Configuration

7.2.2 Register Descriptions

Table 7.2 summarizes the registers of port 4.

Table 7.2 Port 4 Registers

| Address* | Name | Abbreviation | R/W | Initial Value |
|----------|---|--------------|-----|---------------|
| H'EE003 | Port 4 data direction register | P4DDR | W | H'00 |
| H'FFFD3 | Port 4 data register | P4DR | R/W | H'00 |
| H'EE03E | Port 4 input pull-up MOS control register | P4PCR | R/W | H'00 |

Note: * Lower 20 bits of the address in advanced mode.

Port 4 Data Direction Register (P4DDR): P4DDR is an 8-bit write-only register that can select input or output for each pin in port 4.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|
| | P ₄₇ DDR | P ₄₆ DDR | P ₄₅ DDR | P ₄₄ DDR | P ₄₃ DDR | P ₄₂ DDR | P ₄₁ DDR | P ₄₀ DDR |
| Initial value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Read/Write | W | W | W | W | W | W | W | W |

Port 4 data direction 7 to 0

These bits select input or output for port 4 pins

When all areas are designated as 8-bit-access areas by the bus controller's bus width control register (ABWCR), selecting 8-bit bus mode, port 4 functions as an input/output port. In this case, a pin in port 4 becomes an output port if the corresponding P4DDR bit is set to 1, and an input port if this bit is cleared to 0.

When at least one area is designated as a 16-bit-access area, selecting 16-bit bus mode, port 4 functions as part of the data bus, regardless of the P4DDR settings.

P4DDR is a write-only register. Its value cannot be read. All bits return 1 when read.

P4DDR is initialized to H'00 by a reset and in hardware standby mode. In software standby mode it retains its previous setting.

ABWCR and P4DDR are not initialized in software standby mode. Therefore, if a transition is made to software standby mode while port 4 is functioning as an input/output port and a P4DDR bit is set to 1, the corresponding pin maintains its output state.

Port 4 Data Register (P4DR): P4DR is an 8-bit readable/writable register that stores output data for port 4. When port 4 functions as an output port, the value of this register is output. When a bit in P4DDR is set to 1, if port 4 is read the value of the corresponding P4DR bit is returned. When a bit in P4DDR is cleared to 0, if port 4 is read the corresponding pin logic level is read.

| | | | | | | | | |
|---------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | P4 ₇ | P4 ₆ | P4 ₅ | P4 ₄ | P4 ₃ | P4 ₂ | P4 ₁ | P4 ₀ |
| Initial value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Port 4 data 7 to 0

These bits store data for port 4 pins

P4DR is initialized to H'00 by a reset and in hardware standby mode. In software standby mode it retains its previous setting.

Port 4 Input Pull-Up MOS Control Register (P4PCR): P4PCR is an 8-bit readable/writable register that controls the MOS input pull-up transistors in port 4.

| | | | | | | | | |
|---------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | P4 ₇ PCR | P4 ₆ PCR | P4 ₅ PCR | P4 ₄ PCR | P4 ₃ PCR | P4 ₂ PCR | P4 ₁ PCR | P4 ₀ PCR |
| Initial value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Port 4 input pull-up MOS control 7 to 0

These bits control input pull-up MOS transistors built into port 4

In 8-bit bus mode in modes 1 to 4 (expanded modes), when a P4DDR bit is cleared to 0 (selecting generic input), if the corresponding P4PCR bit is set to 1, the input pull-up MOS transistor is turned on.

P4PCR is initialized to H'00 by a reset and in hardware standby mode. In software standby mode it retains its previous setting.

Table 7.3 summarizes the states of the input pull-up MOS in each operating mode.

Table 7.3 Input Pull-Up MOS Transistor States (Port 4)

| Mode | Reset | Hardware Standby Mode | Software Standby Mode | Other Modes |
|-------------|-----------------|------------------------------|------------------------------|--------------------|
| 1 to 4 | 8-bit bus mode | Off | Off | On/off |
| | 16-bit bus mode | | | Off |

Legend:

Off: The input pull-up MOS transistor is always off.

On/off: The input pull-up MOS transistor is on if P4PCR = 1 and P4DDR = 0. Otherwise, it is off.

7.3 Port 6

7.3.1 Overview

Port 6 is an 8-bit input/output port that is also used for input and output of bus control signals (\overline{LWR} , \overline{HWR} , \overline{RD} , \overline{AS} , \overline{BACK} , \overline{BREQ} , \overline{WAIT}) and for clock (ϕ) output.

The port 6 pin configuration is shown in figure 7.2.

See table 7.5 for the selection of the pin functions.

Pins in port 6 can drive one TTL load and a 90-pF capacitive load. They can also drive a darlington transistor pair.

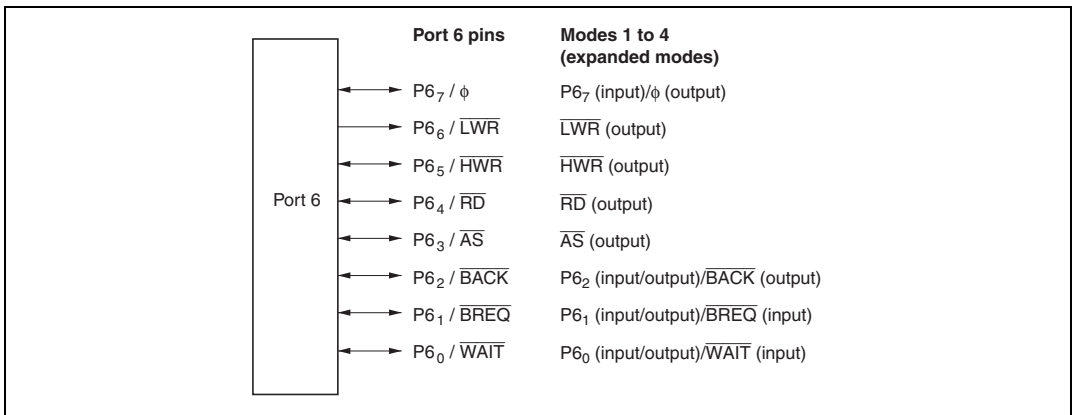


Figure 7.2 Port 6 Pin Configuration

7.3.2 Register Descriptions

Table 7.4 summarizes the registers of port 6.

Table 7.4 Port 6 Registers

| Address* | Name | Abbreviation | R/W | Initial Value |
|----------|--------------------------------|--------------|-----|---------------|
| H'EE005 | Port 6 data direction register | P6DDR | W | H'80 |
| H'FFFD5 | Port 6 data register | P6DR | R/W | H'80 |

Note: * Lower 20 bits of the address in advanced mode.

Port 6 Data Direction Register (P6DDR): P6DDR is an 8-bit write-only register that can select input or output for each pin in port 6.

Bit 7 is reserved. It is fixed at 1, and cannot be modified.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|---|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|
| | — | P6 ₆ DDR | P6 ₅ DDR | P6 ₄ DDR | P6 ₃ DDR | P6 ₂ DDR | P6 ₁ DDR | P6 ₀ DDR |
| Initial value | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Read/Write | — | W | W | W | W | W | W | W |

Reserved bit

Port 6 data direction 6 to 0

These bits select input or output for port 6 pins

- Modes 1 to 4 (Expanded Modes)

P6₇ functions as the clock output pin (ϕ) or an input port. P6₇ is the clock output pin (ϕ) if the PSTOP bit in MSTRCH is cleared to 0 (initial value), and an input port if this bit is set to 1.

P6₆ to P6₃ function as bus control output pins ($\overline{\text{LWR}}$, $\overline{\text{HWR}}$, $\overline{\text{RD}}$, and $\overline{\text{AS}}$), regardless of the settings of bits P6₆DDR to P6₃DDR.

P6₂ to P6₀ function as bus control input/output pins ($\overline{\text{BACK}}$, $\overline{\text{BREQ}}$, and $\overline{\text{WAIT}}$) or input/output ports. For the method of selecting the pin functions, see table 7.7.

When P6₂ to P6₀ function as input/output ports, the pin becomes an output port if the corresponding P6DDR bit is set to 1, and an input port if this bit is cleared to 0.

Port 6 Data Register (P6DR): P6DR is an 8-bit readable/writable register that stores output data for port 6. When port 6 functions as an output port, the value of this register is output. For bit 7, a value of 1 is returned if the bit is read while the PSTOP bit in MSTRCH is cleared to 0, and the P6₇ pin logic level is returned if the bit is read while the PSTOP bit is set to 1. Bit 7 cannot be modified. For bits 6 to 0, the pin logic level is returned if the bit is read while the corresponding bit in P6DDR is cleared to 0, and the P6DR value is returned if the bit is read while the corresponding bit in P6DDR is set to 1.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| | P6 ₇ | P6 ₆ | P6 ₅ | P6 ₄ | P6 ₃ | P6 ₂ | P6 ₁ | P6 ₀ |
| Initial value | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Read/Write | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Port 6 data 7 to 0

These bits store data for port 6 pins

P6DR is initialized to H'80 by a reset and in hardware standby mode. In software standby mode it retains its previous setting.

Table 7.5 Port 6 Pin Functions in Modes 1 to 4

| Pin | Pin Functions and Selection Method | | |
|------------------------|---|-------------------------|---------------|
| $P6_7/\phi$ | Bit PSTOP in MSTCRH selects the pin function. | | |
| | PSTOP | 0 | 1 |
| | Pin function | ϕ output | $P6_7$ input |
| \overline{LWR} | Functions as \overline{LWR} regardless of the setting of bit $P6_6$ DDR | | |
| | $P6_6$ DDR | 0 | 1 |
| | Pin function | \overline{LWR} output | |
| \overline{HWR} | Functions as \overline{HWR} regardless of the setting of bit $P6_5$ DDR | | |
| | $P6_5$ DDR | 0 | 1 |
| | Pin function | \overline{HWR} output | |
| \overline{RD} | Functions as \overline{RD} regardless of the setting of bit $P6_4$ DDR | | |
| | $P6_4$ DDR | 0 | 1 |
| | Pin function | \overline{RD} output | |
| \overline{AS} | Functions as \overline{AS} regardless of the setting of bit $P6_3$ DDR | | |
| | $P6_3$ DDR | 0 | 1 |
| | Pin function | \overline{AS} output | |
| $P6_2/\overline{BACK}$ | Bit BRLE in BR CR and bit $P6_2$ DDR select the pin function as follows. | | |
| | BRLE | 0 | 1 |
| | $P6_2$ DDR | 0 | 1 |
| | Pin function | $P6_2$ input | $P6_2$ output |
| $P6_1/\overline{BREQ}$ | Bit BRLE in BR CR and bit $P6_1$ DDR select the pin function as follows. | | |
| | BRLE | 0 | 1 |
| | $P6_1$ DDR | 0 | 1 |
| | Pin function | $P6_1$ input | $P6_1$ output |
| $P6_0/\overline{WAIT}$ | Bit WAITE in BCR and bit $P6_0$ DDR select the pin function as follows. | | |
| | WAITE | 0 | 1 |
| | $P6_0$ DDR | 0 | 1 |
| | Pin function | $P6_0$ input | $P6_0$ output |

Note: * Do not set bit $P6_0$ DDR to 1.

7.4 Port 7

7.4.1 Overview

Port 7 is an 8-bit input port that is also used for analog input to the A/D converter and analog output from the D/A converter. The pin functions are the same in all operating modes. Figure 7.3 shows the pin configuration of port 7.

See section 14, A/D Converter, for details of the A/D converter analog input pins, and section 15, D/A Converter, for details of the D/A converter analog output pins.

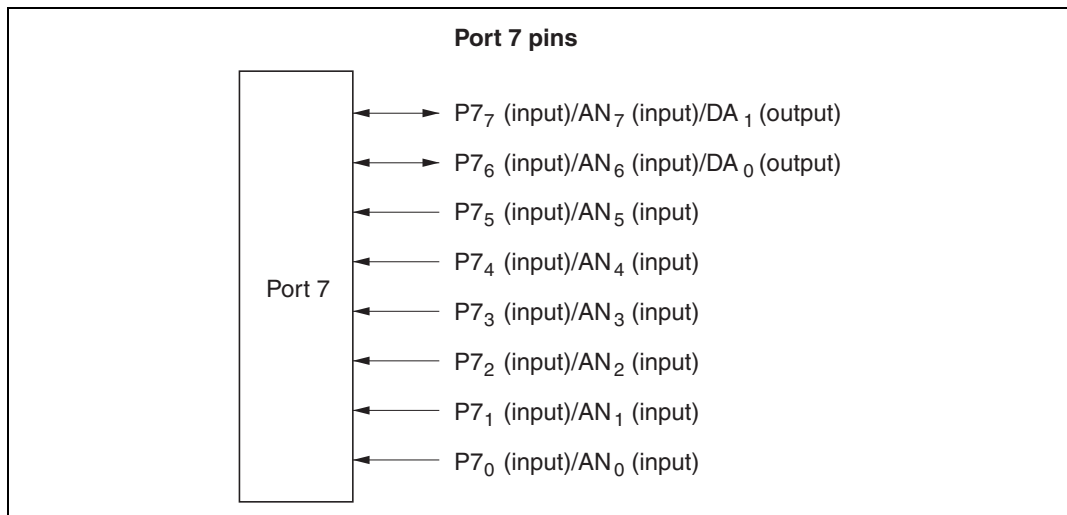


Figure 7.3 Port 7 Pin Configuration

7.4.2 Register Description

Table 7.6 summarizes the port 7 register. Port 7 is an input port, and port 7 has no data direction register.

Table 7.6 Port 7 Data Register

| Address* | Name | Abbreviation | R/W | Initial Value |
|----------|----------------------|--------------|-----|---------------|
| H'FFFD6 | Port 7 data register | P7DR | R | Undetermined |

Note: * Lower 20 bits of the address in advanced mode.

Port 7 Data Register (P7DR)

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| | P7 ₇ | P7 ₆ | P7 ₅ | P7 ₄ | P7 ₃ | P7 ₂ | P7 ₁ | P7 ₀ |
| Initial value | —* | —* | —* | —* | —* | —* | —* | —* |
| Read/Write | R | R | R | R | R | R | R | R |

Note: * Determined by pins P7₇ to P7₀.

When port 7 is read, the pin logic levels are always read. P7DR cannot be modified.

7.5 Port 8**7.5.1 Overview**

Port 8 is a 5-bit input/output port that is also used for \overline{CS}_3 to \overline{CS}_0 output, \overline{IRQ}_3 to \overline{IRQ}_0 input, and A/D converter \overline{ADTRG} input. Figure 7.4 shows the pin configuration of port 8.

In the H8/3008, port 8 can provide \overline{CS}_3 to \overline{CS}_0 output, \overline{IRQ}_3 to \overline{IRQ}_0 input, and \overline{ADTRG} input. See table 7.8 for the selection of pin functions in expanded modes.

See section 14, A/D Converter, for a description of the A/D converter's \overline{ADTRG} input pin.

The \overline{IRQ}_3 to \overline{IRQ}_0 functions are selected by IER settings, regardless of whether the pin is used for input or output. Caution is therefore required. For details see section 5.3.1, External Interrupts.

Pins in port 8 can drive one TTL load and a 90-pF capacitive load. They can also drive a darlington transistor pair.

Pins P8₂ to P8₀ have Schmitt-trigger inputs.

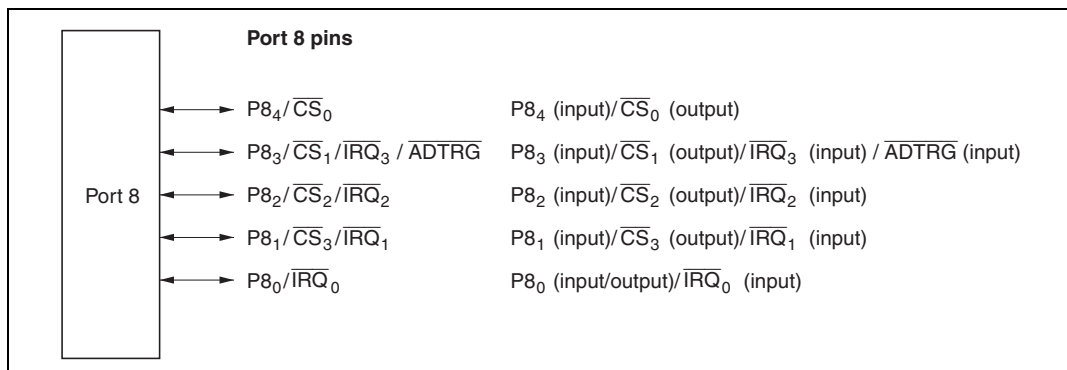


Figure 7.4 Port 8 Pin Configuration

7.5.2 Register Descriptions

Table 7.7 summarizes the registers of port 8.

Table 7.7 Port 8 Registers

| Address* | Name | Abbreviation | R/W | Initial Value |
|----------|--------------------------------|--------------|-----|---------------|
| H'EE07 | Port 8 data direction register | P8DDR | W | H'F0 |
| H'FFFD7 | Port 8 data register | P8DR | R/W | H'E0 |

Note: * Lower 20 bits of the address in advanced mode.

Port 8 Data Direction Register (P8DDR): P8DDR is an 8-bit write-only register that can select input or output for each pin in port 8.

Bits 7 to 5 are reserved. They are fixed at 1, and cannot be modified.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|---|---|---|---------------------|---------------------|---------------------|---------------------|---------------------|
| | — | — | — | P8 ₄ DDR | P8 ₃ DDR | P8 ₂ DDR | P8 ₁ DDR | P8 ₀ DDR |
| Initial value | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| Read/Write | — | — | — | W | W | W | W | W |

Reserved bits
Port 8 data direction 4 to 0
These bits select input or output for port 8 pins

When bits in P8DDR bit are set to 1, $P8_4$ to $P8_1$ become \overline{CS}_0 to \overline{CS}_3 output pins. When bits in P8DDR are cleared to 0, the corresponding pins become input ports.

In the H8/3008, following a reset P8₄ functions as the \overline{CS}_0 output, while \overline{CS}_1 to \overline{CS}_3 are input ports.

P8DDR is a write-only register. Its value cannot be read. All bits return 1 when read.

P8DDR is initialized to H'F0 by a reset and in hardware standby mode. In software standby mode P8DDR retains its previous setting. Therefore, if a transition is made to software standby mode while port 8 is functioning as an input/output port and a P8DDR bit is set to 1, the corresponding pin maintains its output state.

Port 8 Data Register (P8DR): P8DR is an 8-bit readable/writable register that stores output data for port 8. When port 8 functions as an output port, the value of this register is output. When a bit in P8DDR is set to 1, if port 8 is read the value of the corresponding P8DR bit is returned. When a bit in P8DDR is cleared to 0, if port 8 is read the corresponding pin logic level is read.

Bits 7 to 5 are reserved. They are fixed at 1, and cannot be modified.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|---|---|---|-----------------|-----------------|-----------------|-----------------|-----------------|
| | — | — | — | P8 ₄ | P8 ₃ | P8 ₂ | P8 ₁ | P8 ₀ |
| Initial value | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| Read/Write | — | — | — | R/W | R/W | R/W | R/W | R/W |

Reserved bits
Port 8 data 4 to 0
These bits store data
for port 8 pins

P8DR is initialized to H'E0 by a reset and in hardware standby mode. In software standby mode it retains its previous setting.

Table 7.8 Port 8 Pin Functions in Modes 1 to 4

| Pin | Pin Functions and Selection Method | | |
|---|--|--------------------------|--------------------------|
| $P8_4/\overline{CS}_0$ | Bit P8 ₄ DDR selects the pin function as follows. | | |
| | P8 ₄ DDR | 0 | 1 |
| | Pin function | P8 ₄ input | \overline{CS}_0 output |
| $P8_3/\overline{CS}_1/\overline{IRQ}_3/ADTRG$ | Bit P8 ₃ DDR selects the pin function as follows | | |
| | P8 ₃ DDR | 0 | 1 |
| | Pin function | P8 ₃ input | \overline{CS}_1 output |
| | | \overline{IRQ}_3 input | |
| | | ADTRG input | |
| $P8_2/\overline{CS}_2/\overline{IRQ}_2$ | Bit P8 ₂ DDR selects the pin function as follows. | | |
| | P8 ₂ DDR | 0 | 1 |
| | Pin function | P8 ₂ input | \overline{CS}_2 output |
| \overline{IRQ}_2 input | | | |
| $P8_1/\overline{CS}_3/\overline{IRQ}_1$ | Bit P8 ₁ DDR selects the pin function as follows. | | |
| | P8 ₁ DDR | 0 | 1 |
| | Pin function | P8 ₁ input | \overline{CS}_3 output |
| \overline{IRQ}_1 input | | | |
| $P8_0/\overline{IRQ}_0$ | Bit P8 ₀ DDR selects the pin function as follows. | | |
| | P8 ₀ DDR | 0 | 1 |
| | Pin function | P8 ₀ input | P8 ₀ output |
| \overline{IRQ}_0 input | | | |

7.6 Port 9

7.6.1 Overview

Port 9 is a 6-bit input/output port that is also used for input and output (TxD₀, TxD₁, RxD₀, RxD₁, SCK₀, SCK₁) by serial communication interface channels 0 and 1 (SCI0 and SCI1), and for $\overline{\text{IRQ}}_5$ and $\overline{\text{IRQ}}_4$ input. See table 7.10 for the selection of pin functions.

The $\overline{\text{IRQ}}_5$ and $\overline{\text{IRQ}}_4$ functions are selected by IER settings, regardless of whether the pin is used for input or output. Caution is therefore required. For details see section 5.3.1, External Interrupts.

Port 9 has the same set of pin functions in all operating modes. Figure 7.5 shows the pin configuration of port 9.

Pins in port 9 can drive one TTL load and a 30-pF capacitive load. They can also drive a darlington transistor pair.

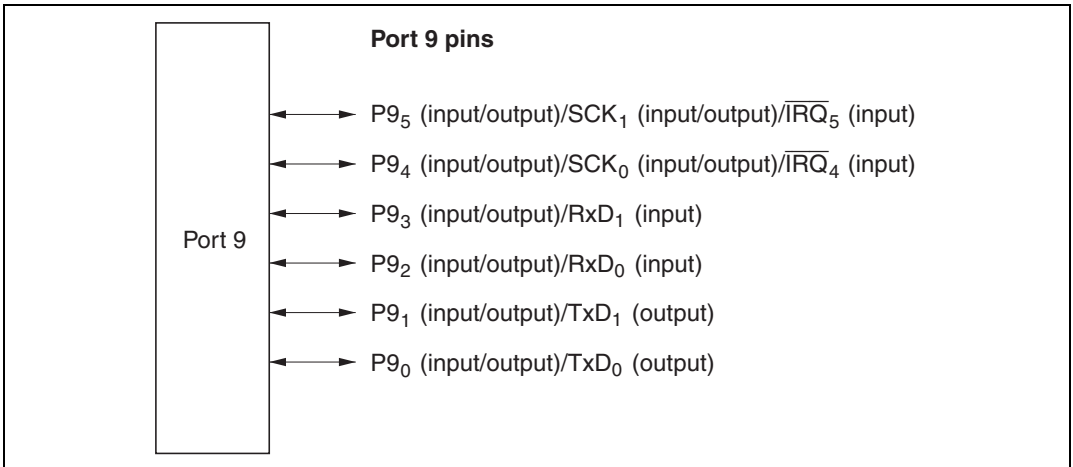


Figure 7.5 Port 9 Pin Configuration

7.6.2 Register Descriptions

Table 7.9 summarizes the registers of port 9.

Table 7.9 Port 9 Registers

| Address* | Name | Abbreviation | R/W | Initial Value |
|----------|--------------------------------|--------------|-----|---------------|
| H'EE008 | Port 9 data direction register | P9DDR | W | H'C0 |
| H'FFFD8 | Port 9 data register | P9DR | R/W | H'C0 |

Note: * Lower 20 bits of the address in advanced mode.

Port 9 Data Direction Register (P9DDR): P9DDR is an 8-bit write-only register that can select input or output for each pin in port 9.

Bits 7 and 6 are reserved. They are fixed at 1, and cannot be modified.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|----------------------|---|--|---------------------|---------------------|---------------------|---------------------|---------------------|
| | — | — | P9 ₅ DDR | P9 ₄ DDR | P9 ₃ DDR | P9 ₂ DDR | P9 ₁ DDR | P9 ₀ DDR |
| Initial value | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| Read/Write | — | — | W | W | W | W | W | W |
| | Reserved bits | | Port 9 data direction 5 to 0 These bits select input or output for port 9 pins | | | | | |

When port 9 functions as an input/output port, a pin in port 9 becomes an output port if the corresponding P9DDR bit is set to 1, and an input port if this bit is cleared to 0. For the method of selecting the pin functions, see table 7.10.

P9DDR is a write-only register. Its value cannot be read. All bits return 1 when read.

P9DDR is initialized to H'C0 by a reset and in hardware standby mode. In software standby mode it retains its previous setting. Therefore, if a transition is made to software standby mode while port 9 is functioning as an input/output port and a P9DDR bit is set to 1, the corresponding pin maintains its output state.

Port 9 Data Register (P9DR): P9DR is an 8-bit readable/writable register that stores output data for port 9. When port 9 functions as an output port, the value of this register is output. When a bit in P9DDR is set to 1, if port 9 is read the value of the corresponding P9DR bit is returned. When a bit in P9DDR is cleared to 0, if port 9 is read the corresponding pin logic level is read.

Bits 7 and 6 are reserved. They are fixed at 1, and cannot be modified.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|---|---|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| | — | — | P9 ₅ | P9 ₄ | P9 ₃ | P9 ₂ | P9 ₁ | P9 ₀ |
| Initial value | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| Read/Write | — | — | R/W | R/W | R/W | R/W | R/W | R/W |

Reserved bits
Port 9 data 5 to 0
These bits store data
for port 9 pins

P9DR is initialized to H'C0 by a reset and in hardware standby mode. In software standby mode it retains its previous setting.

Table 7.10 Port 9 Pin Functions**Pin Pin Functions and Selection Method**

$P9_5/SCK_1/\overline{IRQ}_5$ Bit C/\overline{A} in SMR of SCI1, bits CKE0 and CKE1 in SCR, and bit $P9_5DDR$ select the pin function as follows.

| | | | | | |
|------------------|--------------------------|---------------|----------------|----------------|---------------|
| CKE1 | 0 | | | 1 | |
| C/\overline{A} | 0 | | 1 | — | |
| CKE0 | 0 | | 1 | — | — |
| $P9_5DDR$ | 0 | 1 | — | — | — |
| Pin function | $P9_5$ input | $P9_5$ output | SCK_1 output | SCK_1 output | SCK_1 input |
| | \overline{IRQ}_5 input | | | | |

$P9_4/SCK_0/\overline{IRQ}_4$ Bit C/\overline{A} in SMR of SCI0, bits CKE0 and CKE1 in SCR, and bit $P9_4DDR$ select the pin function as follows.

| | | | | | |
|------------------|--------------------------|---------------|----------------|----------------|---------------|
| CKE1 | 0 | | | 1 | |
| C/\overline{A} | 0 | | 1 | — | |
| CKE0 | 0 | | 1 | — | — |
| $P9_4DDR$ | 0 | 1 | — | — | — |
| Pin function | $P9_4$ input | $P9_4$ output | SCK_0 output | SCK_0 output | SCK_0 input |
| | \overline{IRQ}_4 input | | | | |

$P9_3/RxD_1$ Bit RE in SCR of SCI1, bit SMIF in SCMR, and bit $P9_3DDR$ select the pin function as follows.

| | | | | | |
|--------------|--------------|---------------|---------------|---------------|--|
| SMIF | 0 | | | 1 | |
| RE | 0 | | 1 | — | |
| $P9_3DDR$ | 0 | 1 | — | — | |
| Pin function | $P9_3$ input | $P9_3$ output | RxD_1 input | RxD_1 input | |

$P9_2/RxD_0$ Bit RE in SCR of SCI0, bit SMIF in SCMR, and bit $P9_2DDR$ select the pin function as follows.

| | | | | | |
|--------------|--------------|---------------|---------------|---------------|--|
| SMIF | 0 | | | 1 | |
| RE | 0 | | 1 | — | |
| $P9_2DDR$ | 0 | 1 | — | — | |
| Pin function | $P9_2$ input | $P9_2$ output | RxD_0 input | RxD_0 input | |

Pin Functions and Selection MethodP9₁/TxD₁

Bit TE in SCR of SCI1, bit SMIF in SCMR, and bit P9₁DDR select the pin function as follows.

| | | | | |
|---------------------|-----------------------|------------------------|-------------------------|--------------------------|
| SMIF | 0 | | | 1 |
| TE | 0 | | 1 | — |
| P9 ₁ DDR | 0 | 1 | — | — |
| Pin function | P9 ₁ input | P9 ₁ output | TxD ₁ output | TxD ₁ output* |

Note: * Functions as the TxD₁ output pin, but there are two states: one in which the pin is driven, and another in which the pin is at high-impedance.

P9₀/TxD₀

Bit TE in SCR of SCI0, bit SMIF in SCMR, and bit P9₀DDR select the pin function as follows.

| | | | | |
|---------------------|-----------------------|------------------------|-------------------------|--------------------------|
| SMIF | 0 | | | 1 |
| TE | 0 | | 1 | — |
| P9 ₀ DDR | 0 | 1 | — | — |
| Pin function | P9 ₀ input | P9 ₀ output | TxD ₀ output | TxD ₀ output* |

Note: * Functions as the TxD₀ output pin, but there are two states: one in which the pin is driven, and another in which the pin is at high-impedance.

7.7 Port A

7.7.1 Overview

Port A is an 8-bit input/output port that is also used for output (TP_7 to TP_0) from the programmable timing pattern controller (TPC), input and output ($TIOCB_2$, $TIOCA_2$, $TIOCB_1$, $TIOCA_1$, $TIOCB_0$, $TIOCA_0$, $TCLKD$, $TCLKC$, $TCLKB$, $TCLKA$) by the 16-bit timer, clock input ($TCLKD$, $TCLKC$, $TCLKB$, $TCLKA$) to the 8-bit timer, and address output (A_{23} to A_{20}). A reset or hardware standby transition leaves port A as an input port, except that in modes 3 and 4, one pin is always used for A_{20} output. See tables 7.12 to 7.14 for the selection of pin functions.

Usage of pins for TPC, 16-bit timer, and 8-bit timer input and output is described in the sections on those modules. For output of address bits A_{23} to A_{20} in modes 3 and 4, see section 6.2.4, Bus Release Control Register (BRCR). Pins not assigned to any of these functions are available for generic input/output. Figure 7.6 shows the pin configuration of port A.

Pins in port A can drive one TTL load and a 30-pF capacitive load. They can also drive a darlington transistor pair. Port A has Schmitt-trigger inputs.

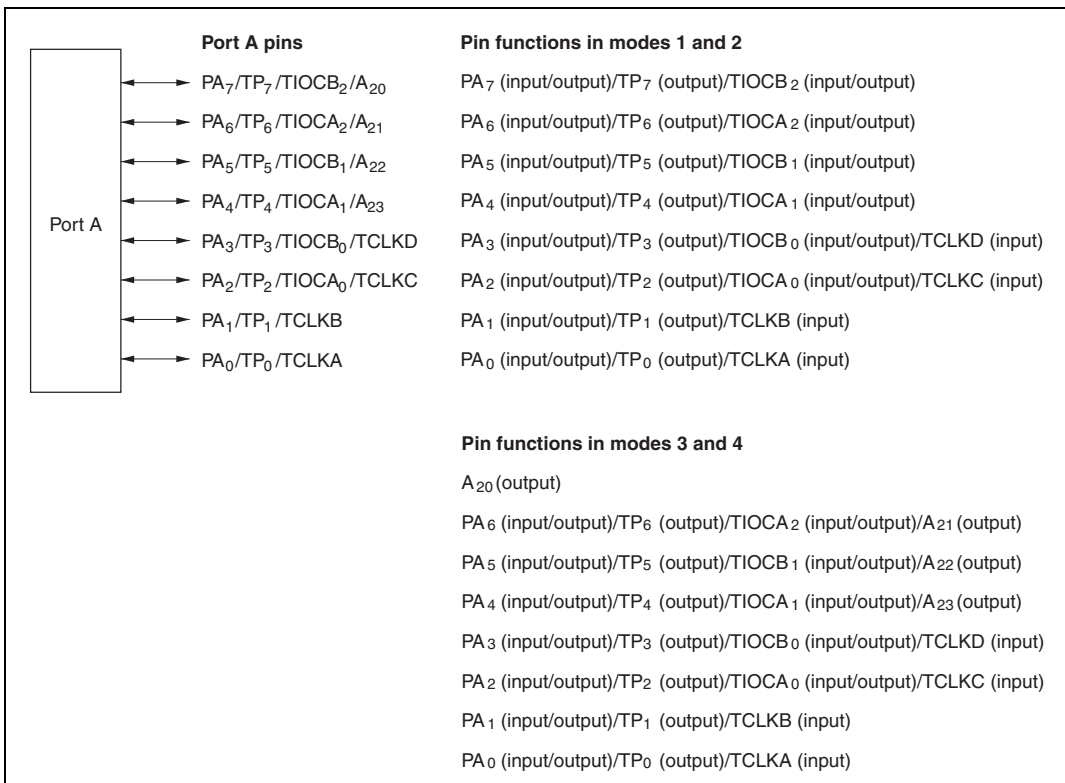


Figure 7.6 Port A Pin Configuration

7.7.2 Register Descriptions

Table 7.11 summarizes the registers of port A.

Table 7.11 Port A Registers

| Address* | Name | PADDR | R/W | Initial Value | |
|----------|--------------------------------|-------|-----|---------------|---------------|
| | | | | Modes 1 and 2 | Modes 3 and 4 |
| H'EE009 | Port A data direction register | PADDR | W | H'00 | H'80 |
| H'FFFD9 | Port A data register | PADR | R/W | H'00 | H'00 |

Note: * Lower 20 bits of the address in advanced mode.

Port A Data Direction Register (PADDR): PADDR is an 8-bit write-only register that can select input or output for each pin in port A. When pins are used for TPC output, the corresponding PADDR bits must also be set.

| Bit | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|---------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|
| | | PA ₇ DDR | PA ₆ DDR | PA ₅ DDR | PA ₄ DDR | PA ₃ DDR | PA ₂ DDR | PA ₁ DDR | PA ₀ DDR |
| Modes 3 and 4 | Initial value | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | Read/Write | — | W | W | W | W | W | W | W |
| Modes 1 and 2 | Initial value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | Read/Write | W | W | W | W | W | W | W | W |

Port A data direction 7 to 0

These bits select input or output for port A pins

The pin functions that can be selected for pins PA₇ to PA₄ differ between modes 1 and 2, and modes 3 and 4. For the method of selecting the pin functions, see tables 7.12 and 7.13.

The pin functions that can be selected for pins PA₃ to PA₀ are the same in modes 1 to 4. For the method of selecting the pin functions, see table 7.14.

When port A functions as an input/output port, a pin in port A becomes an output port if the corresponding PADDR bit is set to 1, and an input port if this bit is cleared to 0. In modes 3 and 4, PA₇DDR is fixed at 1 and PA₇ functions as the A₂₀ address output pin.

PADDR is a write-only register. Its value cannot be read. All bits return 1 when read.

PADDR is initialized to H'00 by a reset and in hardware standby mode in modes 1 and 2. It is initialized to H'80 by a reset and in hardware standby mode in modes 3 and 4. In software standby mode it retains its previous setting. Therefore, if a transition is made to software standby mode while port A is functioning as an input/output port and a PADDR bit is set to 1, the corresponding pin maintains its output state.

Port A Data Register (PADR): PADR is an 8-bit readable/writable register that stores output data for port A. When port A functions as an output port, the value of this register is output. When a bit in PADDR is set to 1, if port A is read the value of the corresponding PADR bit is returned. When a bit in PADDR is cleared to 0, if port A is read the corresponding pin logic level is read.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| | PA ₇ | PA ₆ | PA ₅ | PA ₄ | PA ₃ | PA ₂ | PA ₁ | PA ₀ |
| Initial value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Port A data 7 to 0

These bits store data for port A pins

PADR is initialized to H'00 by a reset and in hardware standby mode. In software standby mode it retains its previous setting.

Table 7.12 Port A Pin Functions (Modes 1 and 2)**Pin Pin Functions and Selection Method**

PA₇/TP₇/
TIOCB₂ Bit PWM2 in TMDR, bits IOB2 to IOB0 in TIOR2, bit NDER7 in NDERA, and bit PA7DDR select the pin function as follows.

| | | | | |
|---------------------------------|---------------------------|---------------------------|------------------------|------------------------|
| 16-bit timer channel 2 settings | (1) in table below | (2) in table below | | |
| PA ₇ DDR | — | 0 | 1 | 1 |
| NDER7 | — | — | 0 | 1 |
| Pin function | TIOCB ₂ output | PA ₇ input | PA ₇ output | TP ₇ output |
| | | TIOCB ₂ input* | | |

Note: * TIOCB₂ input when IOB2 = 1 and PWM2 = 0.

| | | | | |
|---------------------------------|-----|-----|---|-----|
| 16-bit timer channel 2 settings | (2) | (1) | | (2) |
| IOB2 | 0 | | | 1 |
| IOB1 | 0 | 0 | 1 | — |
| IOB0 | 0 | 1 | — | — |

PA₆/TP₆/
TIOCA₂ Bit PWM2 in TMDR, bits IOA2 to IOA0 in TIOR2, bit NDER6 in NDERA, and bit PA₆DDR select the pin function as follows.

| | | | | |
|---------------------------------|---------------------------|---------------------------|------------------------|------------------------|
| 16-bit timer channel 2 settings | (1) in table below | (2) in table below | | |
| PA ₆ DDR | — | 0 | 1 | 1 |
| NDER6 | — | — | 0 | 1 |
| Pin function | TIOCA ₂ output | PA ₆ input | PA ₆ output | TP ₆ output |
| | | TIOCA ₂ input* | | |

Note: * TIOCA₂ input when IOA2 = 1.

| | | | | | |
|---------------------------------|-----|-----|---|-----|-----|
| 16-bit timer channel 2 settings | (2) | (1) | | (2) | (1) |
| PWM2 | 0 | | | | 1 |
| IOA2 | 0 | | | 1 | — |
| IOA1 | 0 | 0 | 1 | — | — |
| IOA0 | 0 | 1 | — | — | — |

Pin Pin Functions and Selection Method

PA₅/TP₅/
TIOCB₁ Bit PWM1 in TMDR, bits IOB2 to IOB0 in TIOR1, bit NDER5 in NDERA, and bit PA₅DDR select the pin function as follows.

| 16-bit timer channel 1 settings | (1) in table below | (2) in table below | | |
|---------------------------------|---------------------------|---------------------------|------------------------|------------------------|
| PA ₅ DDR | — | 0 | 1 | 1 |
| NDER5 | — | — | 0 | 1 |
| Pin function | TIOCB ₁ output | PA ₅ input | PA ₅ output | TP ₅ output |
| | | TIOCB ₁ input* | | |

Note: * TIOCB₁ input when IOB2 = 1 and PWM1 = 0.

| 16-bit timer channel 1 settings | (2) | (1) | | (2) |
|---------------------------------|-----|-----|---|-----|
| IOB2 | 0 | | | 1 |
| IOB1 | 0 | 0 | 1 | — |
| IOB0 | 0 | 1 | — | — |

PA₄/TP₄/
TIOCA₁ Bit PWM1 in TMDR, bits IOA2 to IOA0 in TIOR1, bit NDER4 in NDERA, and bit PA₄DDR select the pin function as follows.

| 16-bit timer channel 1 settings | (1) in table below | (2) in table below | | |
|---------------------------------|---------------------------|---------------------------|------------------------|------------------------|
| PA ₄ DDR | — | 0 | 1 | 1 |
| NDER4 | — | — | 0 | 1 |
| Pin function | TIOCA ₁ output | PA ₄ input | PA ₄ output | TP ₄ output |
| | | TIOCA ₁ input* | | |

Note: * TIOCA₁ input when IOA2 = 1.

| 16-bit timer channel 1 settings | (2) | (1) | | (2) | (1) |
|---------------------------------|-----|-----|---|-----|-----|
| PWM1 | 0 | | | | 1 |
| IOA2 | 0 | | | 1 | — |
| IOA1 | 0 | 0 | 1 | — | — |
| IOA0 | 0 | 1 | — | — | — |

Table 7.13 Port A Pin Functions (Modes 3 and 4)**Pin Pin Functions and Selection Method**

A_{20} Always used as A_{20} output.

| | |
|--------------|-----------------|
| Pin function | A_{20} output |
|--------------|-----------------|

$PA_6/TP_6/TIOCA_2/A_{21}$ Bit PWM2 in TMDR, bits IOA2 to IOA0 in TIOR2, bit NDER6 in NDERA, bit A21E in BRCCR, and bit PA_6 DDR select the pin function as follows.

| | | | | | |
|---------------------------------|--------------------|--------------------|---------------|---------------|-----------------|
| A21E | 1 | | | | 0 |
| 16-bit timer channel 2 settings | (1) in table below | (2) in table below | | | — |
| PA_6 DDR | — | 0 | 1 | 1 | — |
| NDER6 | — | — | 0 | 1 | — |
| Pin function | $TIOCA_2$ output | PA_6 input | PA_6 output | TP_6 output | A_{21} output |
| | | $TIOCA_2$ input* | | | |

Note: * $TIOCA_2$ input when IOA2 = 1.

| | | | | | |
|---------------------------------|-----|-----|-----|-----|---|
| 16-bit timer channel 2 settings | (2) | (1) | (2) | (1) | |
| PWM2 | 0 | | | 1 | |
| IOA2 | 0 | | | 1 | — |
| IOA1 | 0 | 0 | 1 | — | — |
| IOA0 | 0 | 1 | — | — | — |

Pin Pin Functions and Selection Method

PA₅/TP₅/
TIOCB₁/A₂₂ Bit PWM1 in TMDR, bits IOB2 to IOB0 in TIOR1, bit NDER5 in NDERA, bit A22E in BRCR, and bit PA₅DDR select the pin function as follows.

| | | | | | |
|---------------------------------|---------------------------|---------------------------|------------------------|------------------------|------------------------|
| A22E | 1 | | | | 0 |
| 16-bit timer channel 1 settings | (1) in table below | (2) in table below | | | — |
| PA ₅ DDR | — | 0 | 1 | 1 | — |
| NDER5 | — | — | 0 | 1 | — |
| Pin function | TIOCB ₁ output | PA ₅ input | PA ₅ output | TP ₅ output | A ₂₂ output |
| | | TIOCB ₁ input* | | | |

Note: * TIOCB₁ input when IOB2 = 1 and PWM1 = 0.

| | | | | |
|---------------------------------|-----|-----|---|-----|
| 16-bit timer channel 1 settings | (2) | (1) | | (2) |
| IOB2 | 0 | | | 1 |
| IOB1 | 0 | 0 | 1 | — |
| IOB0 | 0 | 1 | — | — |

PA₄/TP₄/
TIOCA₁/A₂₃ Bit PWM1 in TMDR, bits IOA2 to IOA0 in TIOR1, bit NDER4 in NDERA, bit A23E in BRCR, and bit PA₄DDR select the pin function as follows.

| | | | | | |
|---------------------------------|---------------------------|---------------------------|------------------------|------------------------|------------------------|
| A23E | 1 | | | | 0 |
| 16-bit timer channel 1 settings | (1) in table below | (2) in table below | | | — |
| PA ₄ DDR | — | 0 | 1 | 1 | — |
| NDER4 | — | — | 0 | 1 | — |
| Pin function | TIOCA ₁ output | PA ₄ input | PA ₄ output | TP ₄ output | A ₂₃ output |
| | | TIOCA ₁ input* | | | |

Note: * TIOCA₁ input when IOA2 = 1.

| | | | | | |
|---------------------------------|-----|-----|---|-----|-----|
| 16-bit timer channel 1 settings | (2) | (1) | | (2) | (1) |
| PWM1 | 0 | | | | 1 |
| IOA2 | 0 | | 1 | | — |
| IOA1 | 0 | 0 | 1 | — | — |
| IOA0 | 0 | 1 | — | — | — |

Table 7.14 Port A Pin Functions (Modes 1 to 4)**Pin Pin Functions and Selection Method**

PA₃/TP₃/
TIOCB₀/
TCLKD

Bit PWM0 in TMDR, bits IOB2 to IOB0 in TIOR0, bits TPSC2 to TPSC0 in 16TCR2 to 16TCR0 of the 16-bit timer, bits CKS2 to CKS0 in 8TCR2 of the 8-bit timer, bit NDER3 in NDERA, and bit PA₃DDR select the pin function as follows.

| | | | | |
|---------------------------------|---------------------------|--|------------------------|------------------------|
| 16-bit timer channel 0 settings | (1) in table below | (2) in table below | | |
| PA ₃ DDR | — | 0 | 1 | 1 |
| NDER3 | — | — | 0 | 1 |
| Pin function | TIOCB ₀ output | PA ₃ input | PA ₃ output | TP ₃ output |
| | | TIOCB ₀ input* ¹ | | |
| | TCLKD input* ² | | | |

Notes: 1. TIOCB₀ input when IOB2 = 1 and PWM0 = 0.

2. TCLKD input when TPSC2 = TPSC1 = TPSC0 = 1 in any of 16TCR2 to 16TCR0, or bits CKS2 to CKS0 in 8TCR2 are as shown in (3) in the table below.

| | | | | |
|---------------------------------|-----|-----|---|-----|
| 16-bit timer channel 0 settings | (2) | (1) | | (2) |
| IOB2 | 0 | | | 1 |
| IOB1 | 0 | 0 | 1 | — |
| IOB0 | 0 | 1 | — | — |

| | | | | |
|--------------------------------|-----|-----|---|---|
| 8-bit timer channel 2 settings | (4) | (3) | | |
| CKS2 | 0 | 1 | | |
| CKS1 | — | 0 | | 1 |
| CKS0 | — | 0 | 1 | — |

Pin Pin Functions and Selection Method

PA₂/TP₂/
TIOCA₀/
TCLKC

Bit PWM0 in TMDR, bits IOA2 to IOA0 in TIOR0, bits TPSC2 to TPSC0 in 16TCR2 to 16TCR0 of the 16-bit timer, bits CKS2 to CKS0 in 8TCR0 of the 8-bit timer, bit NDER2 in NDERA, and bit PA₂DDR select the pin function as follows.

| 16-bit timer channel 0 settings | (1) in table below | (2) in table below | | |
|---------------------------------|---------------------------|--|------------------------|------------------------|
| PA ₂ DDR | — | 0 | 1 | 1 |
| NDER2 | — | — | 0 | 1 |
| Pin function | TIOCA ₀ output | PA ₂ input | PA ₂ output | TP ₂ output |
| | | TIOCA ₀ input* ¹ | | |
| TCLKC input* ² | | | | |

Notes: 1. TIOCA₀ input when IOA2 = 1.

2. TCLKC input when TPSC2 = TPSC1 = 1 and TPSC0 = 0 in any of 16TCR2 to 16TCR0, or bits CKS2 to CKS0 in 8TCR0 are as shown in (3) in the table below.

| 16-bit timer channel 0 settings | (2) | (1) | | (2) | (1) |
|---------------------------------|-----|-----|---|-----|-----|
| PWM0 | 0 | | | | 1 |
| IOA2 | 0 | | | 1 | — |
| IOA1 | 0 | 0 | 1 | — | — |
| IOA0 | 0 | 1 | — | — | — |

| 8-bit timer channel 0 settings | (4) | | (3) | |
|--------------------------------|-----|---|-----|---|
| CKS2 | 0 | 1 | | |
| CKS1 | — | 0 | | 1 |
| CKS0 | — | 0 | 1 | — |

Pin Pin Functions and Selection Method

$PA_i/TP_i/$
TCLKB Bit MDF in TMDR, bits TPSC2 to TPSC0 in 16TCR2 to 16TCR0 of the 16-bit timer, bits CKS2 to CKS0 in 8TCR3 of the 8-bit timer, bit NDER1 in NDERA, and bit PA_i DDR select the pin function as follows.

| | | | |
|--------------|--------------|---------------|---------------|
| PA_i DDR | 0 | 1 | 1 |
| NDER1 | — | 0 | 1 |
| Pin function | PA_i input | PA_i output | TP_i output |
| | TCLKB input* | | |

Note: * TCLKB input when MDF = 1 in TMDR, or TPSC2 = 1, TPSC1 = 0, and TPSC0 = 1 in any of 16TCR2 to 16TCR0, or bits CKS2 to CKS0 in 8TCR3 are as shown in (1) in the table below.

| | | | | |
|--------------------------------|-----|---|-----|---|
| 8-bit timer channel 3 settings | (2) | | (1) | |
| CKS2 | 0 | 1 | | |
| CKS1 | — | 0 | | 1 |
| CKS0 | — | 0 | 1 | — |

$PA_0/TP_0/$
TCLKA Bit MDF in TMDR, bits TPSC2 to TPSC0 in 16TCR2 to 16TCR0 of the 16-bit timer, bits CKS2 to CKS0 in 8TCR1 of the 8-bit timer, bit NDER0 in NDERA, and bit PA_0 DDR select the pin function as follows.

| | | | |
|--------------|--------------|---------------|---------------|
| PA_0 DDR | 0 | 1 | |
| NDER0 | — | 0 | 1 |
| Pin function | PA_0 input | PA_0 output | TP_0 output |
| | TCLKA input* | | |

Note: * TCLKA input when MDF = 1 in TMDR, or TPSC2 = 1 and TPSC1 = 0, and TPSC0 = 0 in any of 16TCR2 to 16TCR0, or bits CKS2 to CKS0 in 8TCR1 are as shown in (1) in the table below.

| | | | | |
|--------------------------------|-----|---|-----|---|
| 8-bit timer channel 1 settings | (2) | | (1) | |
| CKS2 | 0 | 1 | | |
| CKS1 | — | 0 | | 1 |
| CKS0 | — | 0 | 1 | — |

7.8 Port B

7.8.1 Overview

Port B is an 8-bit input/output port that is also used for output (TP₁₅ to TP₈) from the programmable timing pattern controller (TPC), input/output (TMIO₃, TMO₂, TMIO₁, TMO₀) by the 8-bit timer, and \overline{CS}_7 to \overline{CS}_4 output. See table 7.16 for the selection of pin functions. A reset or hardware standby transition leaves port B as an input/output port.

For output of \overline{CS}_7 to \overline{CS}_4 in modes 1 to 4, see section 6.3.4, Chip Select Signals. Pins not assigned to any of these functions are available for generic input/output. Figure 7.7 shows the pin configuration of port B.

Pins in port B can drive one TTL load and a 30-pF capacitive load. They can also drive darlington transistor pair.

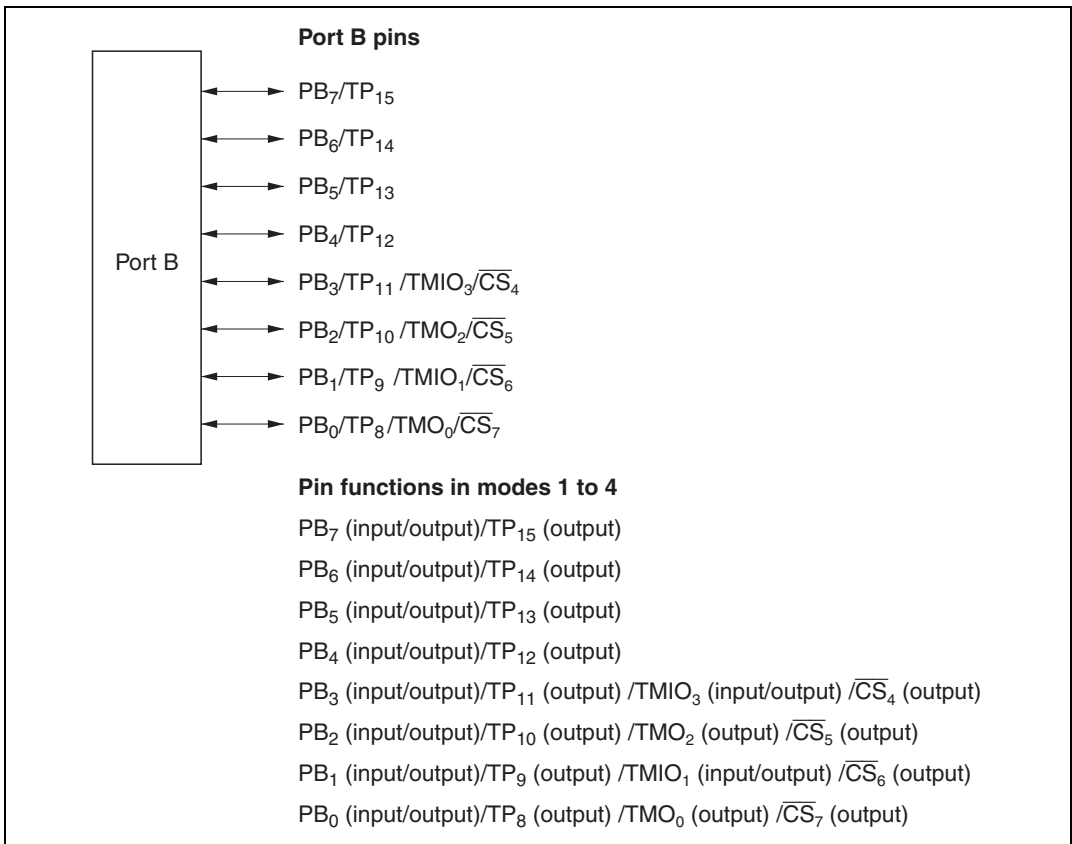


Figure 7.7 Port B Pin Configuration

7.8.2 Register Descriptions

Table 7.15 summarizes the registers of port B.

Table 7.15 Port B Registers

| Address* | Name | Abbreviation | R/W | Initial Value |
|----------|--------------------------------|--------------|-----|---------------|
| H'EE00A | Port B data direction register | PBDDR | W | H'00 |
| H'FFFDA | Port B data register | PBDR | R/W | H'00 |

Note: * Lower 20 bits of the address in advanced mode.

Port B Data Direction Register (PBDDR): PBDDR is an 8-bit write-only register that can select input or output for each pin in port B. When pins are used for TPC output, the corresponding PBDDR bits must also be set.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|
| | PB ₇ DDR | PB ₆ DDR | PB ₅ DDR | PB ₄ DDR | PB ₃ DDR | PB ₂ DDR | PB ₁ DDR | PB ₀ DDR |
| Initial value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Read/Write | W | W | W | W | W | W | W | W |

Port B data direction 7 to 0

These bits select input or output for port B pins

For the method of selecting the pin functions, see table 7.16.

When port B functions as an input/output port, a pin in port B becomes an output port if the corresponding PBDDR bit is set to 1, and an input port if this bit is cleared to 0.

PBDDR is a write-only register. Its value cannot be read. All bits return 1 when read.

PBDDR is initialized to H'00 by a reset and in hardware standby mode. In software standby mode it retains its previous setting. Therefore, if a transition is made to software standby mode while port B is functioning as an input/output port and a PBDDR bit is set to 1, the corresponding pin maintains its output state.

Port B Data Register (PBDR): PBDR is an 8-bit readable/writable register that stores output data for pins port B. When port B functions as an output port, the value of this register is output. When a bit in PBDDR is set to 1, if port B is read the value of the corresponding PBDR bit is returned. When a bit in PBDDR is cleared to 0, if port B is read the corresponding pin logic level is read.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| | PB ₇ | PB ₆ | PB ₅ | PB ₄ | PB ₃ | PB ₂ | PB ₁ | PB ₀ |
| Initial value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

|
Port B data 7 to 0

These bits store data for port B pins

PBDR is initialized to H'00 by a reset and in hardware standby mode. In software standby mode it retains its previous setting.

Table 7.16 Port B Pin Functions (Modes 1 to 4)**Pin Pin Functions and Selection Method**

PB_7/TP_{15} Bit NDER15 in NDERB and bit PB_7DDR select the pin function as follows.

| | | | |
|--------------|--------------|---------------|------------------|
| PB_7DDR | 0 | 1 | 1 |
| NDER15 | — | 0 | 1 |
| Pin function | PB_7 input | PB_7 output | TP_{15} output |

PB_6/TP_{14} Bit NDER14 in NDERB and bit PB_6DDR select the pin function as follows.

| | | | |
|--------------|--------------|---------------|------------------|
| PB_6DDR | 0 | 1 | 1 |
| NDER14 | — | 0 | 1 |
| Pin function | PB_6 input | PB_6 output | TP_{14} output |

PB_5/TP_{13} Bit NDER13 in NDERB and bit PB_5DDR select the pin function as follows.

| | | | |
|--------------|--------------|---------------|------------------|
| PB_5DDR | 0 | 1 | 1 |
| NDER13 | — | 0 | 1 |
| Pin function | PB_5 input | PB_5 output | TP_{13} output |

PB_4/TP_{12} Bit NDER12 in NDERB and bit PB_4DDR select the pin function as follows.

| | | | |
|--------------|--------------|---------------|------------------|
| PB_4DDR | 0 | 1 | 1 |
| NDER12 | — | 0 | 1 |
| Pin function | PB_4 input | PB_4 output | TP_{12} output |

$PB_3/TP_{11}/$
 $TMIO_3/\overline{CS}_4$ Bits OIS3/2 and OS1/0 in 8TCSR3, bits CCLR1/0 in 8TCR3, bit CS4E in CSCR, bit NDER11 in NDERB, and bit PB_3DDR select the pin function as follows.

| | | | | | |
|------------------|-----------------|---------------|------------------|--------------------------|-----------------|
| OIS3/2 and OS1/0 | All 0 | | | | Not all 0 |
| CS4E | 0 | | 1 | | — |
| PB_3DDR | 0 | 1 | 1 | — | — |
| NDER11 | — | 0 | 1 | — | — |
| Pin function | PB_3 input | PB_3 output | TP_{11} output | \overline{CS}_4 output | $TMIO_3$ output |
| | $TMIO_3$ input* | | | | |

Note: * $TMIO_3$ input when bit ICE = 1 in 8TCSR3.

Pin Pin Functions and Selection Method

$PB_2/TP_{10}/TMIO_2/\overline{CS}_5$ Bits OIS3/2 and OS1/0 in 8TCSR2, bit CS5E in CSCR, bit NDER10 in NDERB, and bit PB_2 DDR select the pin function as follows.

| | | | | | |
|------------------|-----------------|------------------|---------------------|-----------------------------|--------------------|
| OIS3/2 and OS1/0 | All 0 | | | | Not all 0 |
| CS5E | 0 | | | 1 | — |
| PB_2 DDR | 0 | 1 | 1 | — | — |
| NDER10 | — | 0 | 1 | — | — |
| Pin function | PB_2 input | PB_2 output | TP_{10} output | \overline{CS}_5 output | $TMIO_2$ output |

$PB_1/TP_9/TMIO_1/\overline{CS}_6$ Bits OIS3/2 and OS1/0 in 8TCSR1, bits CCLR1/0 in 8TCR1, bit CS6E in CSCR, bit NDER9 in NDERB, and bit PB_1 DDR select the pin function as follows.

| | | | | | |
|------------------|-----------------|------------------|------------------|-----------------------------|--------------------|
| OIS3/2 and OS1/0 | All 0 | | | | Not all 0 |
| CS6E | 0 | | | 1 | — |
| PB_1 DDR | 0 | 1 | 1 | — | — |
| NDER9 | — | 0 | 1 | — | — |
| Pin function | PB_1 input | PB_1 output | TP_9 output | \overline{CS}_6 output | $TMIO_1$ output |
| | $TMIO_1$ input* | | | | |

Note: * $TMIO_1$ input when bit ICE = 1 in 8TCSR1.

$PB_0/TP_8/TMO_0/\overline{CS}_7$ Bits OIS3/2 and OS1/0 in 8TCSR0, bit CS7E in CSCR, bit NDER8 in NDERB, and bit PB_0 DDR select the pin function as follows.

| | | | | | |
|------------------|-----------------|------------------|------------------|-----------------------------|-------------------|
| OIS3/2 and OS1/0 | All 0 | | | | Not all 0 |
| CS7E | 0 | | | 1 | — |
| PB_0 DDR | 0 | 1 | 1 | — | — |
| NDER8 | — | 0 | 1 | — | — |
| Pin function | PB_0 input | PB_0 output | TP_8 output | \overline{CS}_7 output | TMO_0 output |

Section 8 16-Bit Timer

8.1 Overview

The H8/3008 has built-in 16-bit timer module with three 16-bit counter channels.

8.1.1 Features

16-bit timer features are listed below.

- Capability to process up to 6 pulse outputs or 6 pulse inputs
- Six general registers (GRs, two per channel) with independently-assignable output compare or input capture functions
- Selection of eight counter clock sources for each channel:
Internal clocks: ϕ , $\phi/2$, $\phi/4$, $\phi/8$
External clocks: TCLKA, TCLKB, TCLKC, TCLKD
- Five operating modes selectable in all channels:
 - Waveform output by compare match
Selection of 0 output, 1 output, or toggle output (only 0 or 1 output in channel 2)
 - Input capture function
Rising edge, falling edge, or both edges (selectable)
 - Counter clearing function
Counters can be cleared by compare match or input capture
 - Synchronization
Two or more timer counters (16TCNTs) can be preset simultaneously, or cleared simultaneously by compare match or input capture. Counter synchronization enables synchronous register input and output.
 - PWM mode
PWM output can be provided with an arbitrary duty cycle. With synchronization, up to three-phase PWM output is possible
- Phase counting mode selectable in channel 2
Two-phase encoder output can be counted automatically.
- High-speed access via internal 16-bit bus
The 16TCNTs and GRs can be accessed at high speed via a 16-bit bus.
- Any initial timer output value can be set
- Nine interrupt sources

Each channel has two compare match/input capture interrupts and an overflow interrupt. All interrupts can be requested independently.

- Output triggering of programmable timing pattern controller (TPC)
Compare match/input capture signals from channels 0 to 2 can be used as TPC output triggers.

Table 8.1 summarizes the 16-bit timer functions.

Table 8.1 16-bit timer Functions

| Item | | Channel 0 | Channel 1 | Channel 2 |
|--|--------|---|--|--|
| Clock sources | | Internal clocks: ϕ , $\phi/2$, $\phi/4$, $\phi/8$ External clocks: TCLKA, TCLKB, TCLKC, TCLKD, selectable independently | | |
| General registers (output compare/input capture registers) | | GRA0, GRB0 | GRA1, GRB1 | GRA2, GRB2 |
| Input/output pins | | TIOCA ₀ , TIOCB ₀ | TIOCA ₁ , TIOCB ₁ | TIOCA ₂ , TIOCB ₂ |
| Counter clearing function | | GRA0/GRB0 compare match or input capture | GRA1/GRB1 compare match or input capture | GRA2/GRB2 compare match or input capture |
| Initial output value setting function | | Available | Available | Available |
| Compare match output | 0 | Available | Available | Available |
| | 1 | Available | Available | Available |
| | Toggle | Available | Available | Not available |
| Input capture function | | Available | Available | Available |
| Synchronization | | Available | Available | Available |
| PWM mode | | Available | Available | Available |
| Phase counting mode | | Not available | Not available | Available |
| Interrupt sources | | Three sources | Three sources | Three sources |
| | | • Compare match/input capture A0 | • Compare match/input capture A1 | • Compare match/input capture A2 |
| | | • Compare match/input capture B0 | • Compare match/input capture B1 | • Compare match/input capture B2 |
| | | • Overflow | • Overflow | • Overflow |

8.1.2 Block Diagrams

16-bit timer Block Diagram (Overall): Figure 8.1 is a block diagram of the 16-bit timer.

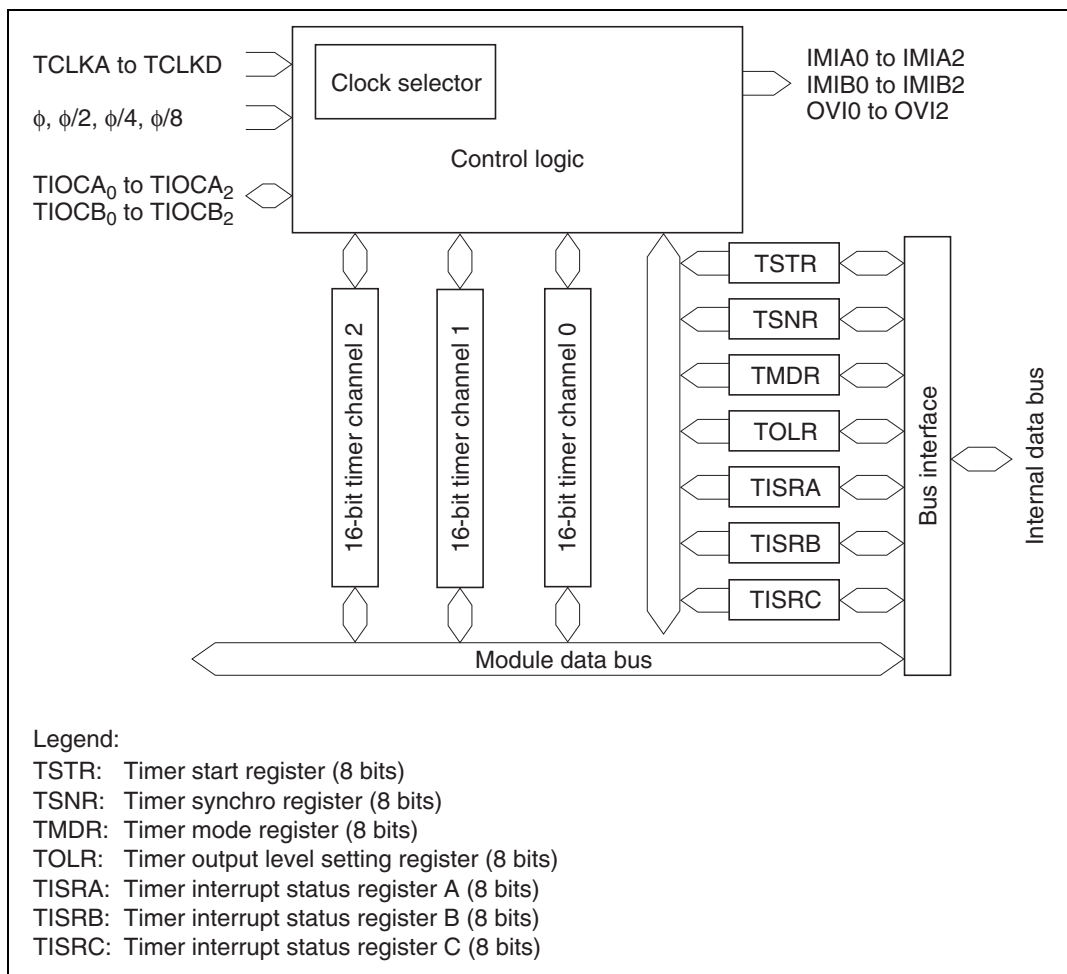


Figure 8.1 16-bit timer Block Diagram (Overall)

Block Diagram of Channels 0 and 1: 16-bit timer channels 0 and 1 are functionally identical. Both have the structure shown in figure 8.2.

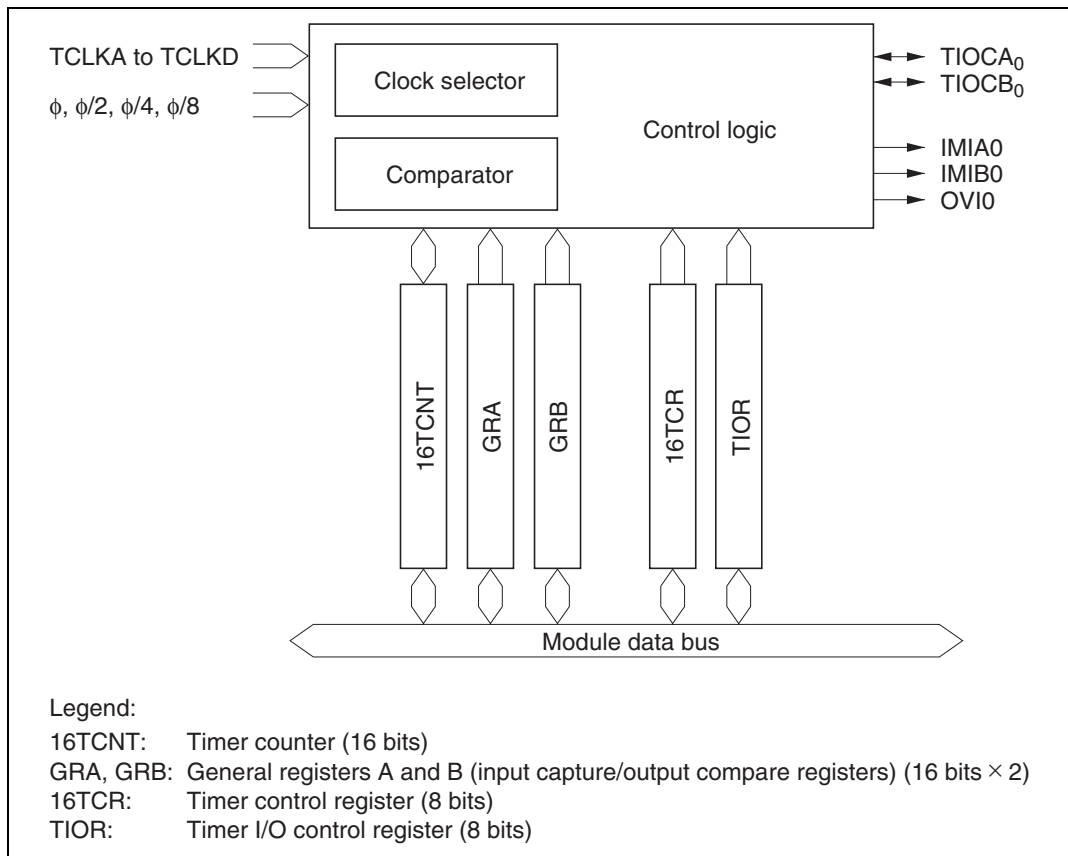


Figure 8.2 Block Diagram of Channels 0 and 1

Block Diagram of Channel 2: Figure 8.3 is a block diagram of channel 2

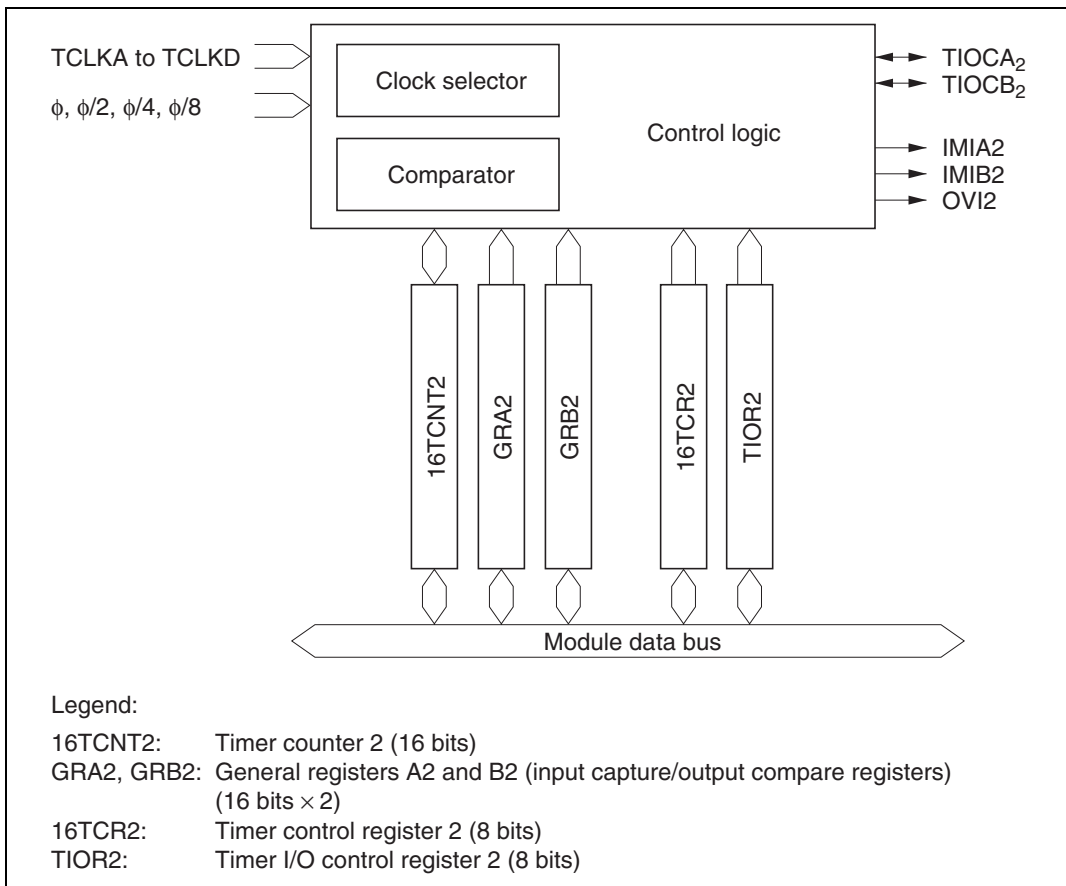


Figure 8.3 Block Diagram of Channel 2

8.1.3 Pin Configuration

Table 8.2 summarizes the 16-bit timer pins.

Table 8.2 16-bit timer Pins

| Channel | Name | Abbreviation | Input/Output | Function |
|---------|---------------------------------|--------------------|--------------|--|
| Common | Clock input A | TCLKA | Input | External clock A input pin (phase-A input pin in phase counting mode) |
| | Clock input B | TCLKB | Input | External clock B input pin (phase-B input pin in phase counting mode) |
| | Clock input C | TCLKC | Input | External clock C input pin |
| | Clock input D | TCLKD | Input | External clock D input pin |
| 0 | Input capture/output compare A0 | TIOCA ₀ | Input/output | GRA0 output compare or input capture pin PWM output pin in PWM mode |
| | Input capture/output compare B0 | TIOCB ₀ | Input/output | GRB0 output compare or input capture pin |
| 1 | Input capture/output compare A1 | TIOCA ₁ | Input/output | GRA1 output compare or input capture pin PWM output pin in PWM mode |
| | Input capture/output compare B1 | TIOCB ₁ | Input/output | GRB1 output compare or input capture pin |
| 2 | Input capture/output compare A2 | TIOCA ₂ | Input/output | GRA2 output compare or input capture pin PWM output pin in PWM mode |
| | Input capture/output compare B2 | TIOCB ₂ | Input/output | GRB2 output compare or input capture pin |

8.1.4 Register Configuration

Table 8.3 summarizes the 16-bit timer registers.

Table 8.3 16-bit timer Registers

| Channel | Address* ¹ | Name | Abbreviation | R/W | Initial Value |
|---------|-----------------------|-------------------------------------|--------------|---------------------|---------------|
| Common | H'FFF60 | Timer start register | TSTR | R/W | H'F8 |
| | H'FFF61 | Timer synchro register | TSNC | R/W | H'F8 |
| | H'FFF62 | Timer mode register | TMDR | R/W | H'98 |
| | H'FFF63 | Timer output level setting register | TOLR | W | H'00 |
| | H'FFF64 | Timer interrupt status register A | TISRA | R/(W)* ² | H'88 |
| | H'FFF65 | Timer interrupt status register B | TISRB | R/(W)* ² | H'88 |
| | H'FFF66 | Timer interrupt status register C | TISRC | R/(W)* ² | H'88 |
| 0 | H'FFF68 | Timer control register 0 | 16TCR0 | R/W | H'80 |
| | H'FFF69 | Timer I/O control register 0 | TIOR0 | R/W | H'88 |
| | H'FFF6A | Timer counter 0H | 16TCNT0H | R/W | H'00 |
| | H'FFF6B | Timer counter 0L | 16TCNT0L | R/W | H'00 |
| | H'FFF6C | General register A0H | GRA0H | R/W | H'FF |
| | H'FFF6D | General register A0L | GRA0L | R/W | H'FF |
| | H'FFF6E | General register B0H | GRB0H | R/W | H'FF |
| | H'FFF6F | General register B0L | GRB0L | R/W | H'FF |
| 1 | H'FFF70 | Timer control register 1 | 16TCR1 | R/W | H'80 |
| | H'FFF71 | Timer I/O control register 1 | TIOR1 | R/W | H'88 |
| | H'FFF72 | Timer counter 1H | 16TCNT1H | R/W | H'00 |
| | H'FFF73 | Timer counter 1L | 16TCNT1L | R/W | H'00 |
| | H'FFF74 | General register A1H | GRA1H | R/W | H'FF |
| | H'FFF75 | General register A1L | GRA1L | R/W | H'FF |
| | H'FFF76 | General register B1H | GRB1H | R/W | H'FF |
| | H'FFF77 | General register B1L | GRB1L | R/W | H'FF |

| Channel | Address* ¹ | Name | Abbreviation | R/W | Initial Value |
|---------|-----------------------|------------------------------|--------------|-----|---------------|
| 2 | H'FFF78 | Timer control register 2 | 16TCR2 | R/W | H'80 |
| | H'FFF79 | Timer I/O control register 2 | TIOR2 | R/W | H'88 |
| | H'FFF7A | Timer counter 2H | 16TCNT2H | R/W | H'00 |
| | H'FFF7B | Timer counter 2L | 16TCNT2L | R/W | H'00 |
| | H'FFF7C | General register A2H | GRA2H | R/W | H'FF |
| | H'FFF7D | General register A2L | GRA2L | R/W | H'FF |
| | H'FFF7E | General register B2H | GRB2H | R/W | H'FF |
| | H'FFF7F | General register B2L | GRB2L | R/W | H'FF |

- Notes: 1. The lower 20 bits of the address in advanced mode are indicated.
 2. Only 0 can be written in bits 3 to 0, to clear the flags.

8.2 Register Descriptions

8.2.1 Timer Start Register (TSTR)

TSTR is an 8-bit readable/writable register that starts and stops the timer counter (16TCNT) in channels 0 to 2.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|---|---|---|---|---|------|------|------|
| | — | — | — | — | — | STR2 | STR1 | STR0 |
| Initial value | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| Read/Write | — | — | — | — | — | R/W | R/W | R/W |

Reserved bits
Counter start 2 to 0
 These bits start and stop 16TCNT2 to 16TCNT0

TSTR is initialized to H'F8 by a reset and in standby mode.

Bits 7 to 3—Reserved: These bits cannot be modified and are always read as 1.

Bit 2—Counter Start 2 (STR2): Starts and stops timer counter 2 (16TCNT2).

| Bit 2 STR2 | Description | |
|---------------|---------------------|-----------------|
| 0 | 16TCNT2 is halted | (Initial value) |
| 1 | 16TCNT2 is counting | |

Bit 1—Counter Start 1 (STR1): Starts and stops timer counter 1 (16TCNT1).

| Bit 1 STR1 | Description | |
|---------------|---------------------|-----------------|
| 0 | 16TCNT1 is halted | (Initial value) |
| 1 | 16TCNT1 is counting | |

Bit 0—Counter Start 0 (STR0): Starts and stops timer counter 0 (16TCNT0).

| Bit 0 STR0 | Description | |
|---------------|---------------------|-----------------|
| 0 | 16TCNT0 is halted | (Initial value) |
| 1 | 16TCNT0 is counting | |

8.2.2 Timer Synchro Register (TSNC)

TSNC is an 8-bit readable/writable register that selects whether channels 0 to 2 operate independently or synchronously. Channels are synchronized by setting the corresponding bits to 1.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|---|---|---|---|---|-------|-------|-------|
| | — | — | — | — | — | SYNC2 | SYNC1 | SYNC0 |
| Initial value | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| Read/Write | — | — | — | — | — | R/W | R/W | R/W |

Reserved bits
Timer sync 2 to 0
 These bits synchronize channels 2 to 0

TSNC is initialized to H'F8 by a reset and in standby mode.

Bits 7 to 3—Reserved: These bits cannot be modified and are always read as 1.

Bit 2—Timer Sync 2 (SYNC2): Selects whether channel 2 operates independently or synchronously.

Bit 2**SYNC2 Description**

| | | |
|---|---|-----------------|
| 0 | Channel 2's timer counter (16TCNT2) operates independently 16TCNT2 is preset and cleared independently of other channels | (Initial value) |
| 1 | Channel 2 operates synchronously 16TCNT2 can be synchronously preset and cleared | |

Bit 1—Timer Sync 1 (SYNC1): Selects whether channel 1 operates independently or synchronously.

Bit 1**SYNC1 Description**

| | | |
|---|---|-----------------|
| 0 | Channel 1's timer counter (16TCNT1) operates independently 16TCNT1 is preset and cleared independently of other channels | (Initial value) |
| 1 | Channel 1 operates synchronously 16TCNT1 can be synchronously preset and cleared | |

Bit 0—Timer Sync 0 (SYNC0): Selects whether channel 0 operates independently or synchronously.

Bit 0**SYNC0 Description**

| | | |
|---|---|-----------------|
| 0 | Channel 0's timer counter (16TCNT0) operates independently 16TCNT0 is preset and cleared independently of other channels | (Initial value) |
| 1 | Channel 0 operates synchronously 16TCNT0 can be synchronously preset and cleared | |

8.2.3 Timer Mode Register (TMDR)

TMDR is an 8-bit readable/writable register that selects PWM mode for channels 0 to 2. It also selects phase counting mode and the overflow flag (OVF) setting conditions for channel 2.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|---|-----|------|---|---|------|------|------|
| | — | MDF | FDIR | — | — | PWM2 | PWM1 | PWM0 |
| Initial value | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| Read/Write | — | R/W | R/W | — | — | R/W | R/W | R/W |

Reserved bit
 These bits select PWM mode for channels 2 to 0

Flag direction
 Selects the setting condition for the overflow flag (OVF) in TISRC

Phase counting mode flag
 Selects phase counting mode for channel 2

Reserved bit

TMDR is initialized to H'98 by a reset and in standby mode.

Bit 7—Reserved: This bit cannot be modified and is always read as 1.

Bit 6—Phase Counting Mode Flag (MDF): Selects whether channel 2 operates normally or in phase counting mode.

Bit 6

| MDF | Description |
|-----|---|
| 0 | Channel 2 operates normally (Initial value) |
| 1 | Channel 2 operates in phase counting mode |

When MDF is set to 1 to select phase counting mode, 16TCNT2 operates as an up/down-counter and pins TCLKA and TCLKB become counter clock input pins. 16TCNT2 counts both rising and falling edges of TCLKA and TCLKB, and counts up or down as follows.

| Counting Direction | Down-Counting | | | | Up-Counting | | | |
|--------------------|---------------|------|------|-----|-------------|------|------|-----|
| | | | | | | | | |
| TCLKA pin | ↑ | High | ↓ | Low | Low | ↑ | High | ↓ |
| TCLKB pin | Low | ↑ | High | ↓ | ↑ | High | ↓ | Low |

In phase counting mode, external clock edge selection by bits CKEG1 and CKEG0 in 16TCR2 and counter clock selection by bits TPSC2 to TPSC0 are invalid, and the above phase counting mode operations take precedence.

The counter clearing condition selected by the CCLR1 and CCLR0 bits in 16TCR2 and the compare match/input capture settings and interrupt functions of TIOR2, TISRA, TISRB, TISRC remain effective in phase counting mode.

Bit 5—Flag Direction (FDIR): Designates the setting condition for the OVF flag in TISRC. The FDIR designation is valid in all modes in channel 2.

| Bit 5 FDIR | Description | |
|---------------|---|-----------------|
| 0 | OVF is set to 1 in TISRC when 16TCNT2 overflows or underflows | (Initial value) |
| 1 | OVF is set to 1 in TISRC when 16TCNT2 overflows | |

Bits 4 and 3—Reserved: These bits cannot be modified and are always read as 1.

Bit 2—PWM Mode 2 (PWM2): Selects whether channel 2 operates normally or in PWM mode.

| Bit 2 PWM2 | Description | |
|---------------|--------------------------------|-----------------|
| 0 | Channel 2 operates normally | (Initial value) |
| 1 | Channel 2 operates in PWM mode | |

When bit PWM2 is set to 1 to select PWM mode, pin TIOCA₂ becomes a PWM output pin. The output goes to 1 at compare match with GRA2, and to 0 at compare match with GRB2.

Bit 1—PWM Mode 1 (PWM1): Selects whether channel 1 operates normally or in PWM mode.

| Bit 1 PWM1 | Description | |
|---------------|--------------------------------|-----------------|
| 0 | Channel 1 operates normally | (Initial value) |
| 1 | Channel 1 operates in PWM mode | |

When bit PWM1 is set to 1 to select PWM mode, pin TIOCA₁ becomes a PWM output pin. The output goes to 1 at compare match with GRA₁, and to 0 at compare match with GRB₁.

Bit 0—PWM Mode 0 (PWM0): Selects whether channel 0 operates normally or in PWM mode.

Bit 0

| PWM0 | Description | |
|------|--------------------------------|-----------------|
| 0 | Channel 0 operates normally | (Initial value) |
| 1 | Channel 0 operates in PWM mode | |

When bit PWM0 is set to 1 to select PWM mode, pin TIOCA₀ becomes a PWM output pin. The output goes to 1 at compare match with GRA₀, and to 0 at compare match with GRB₀.

8.2.4 Timer Interrupt Status Register A (TISRA)

TISRA is an 8-bit readable/writable register that indicates GRA compare match or input capture and enables or disables GRA compare match and input capture interrupt requests.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|---|--------|--------|--------|---|--------|--------|--------|
| | — | IMIEA2 | IMIEA1 | IMIEA0 | — | IMFA2 | IMFA1 | IMFA0 |
| Initial value | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| Read/Write | — | R/W | R/W | R/W | — | R/(W)* | R/(W)* | R/(W)* |

Reserved bit
Input capture/compare match interrupt enable A2 to A0
 These bits enable or disable interrupts by the IMFA flags

Reserved bit
Input capture/compare match flags A2 to A0
 Status flags indicating GRA compare match or input capture

Note: * Only 0 can be written, to clear the flag.

TISRA is initialized to H'88 by a reset and in standby mode.

Bit 7—Reserved: This bit cannot be modified and is always read as 1.

Bit 6—Input Capture/Compare Match Interrupt Enable A2 (IMIEA2): Enables or disables the interrupt requested by the IMFA2 when IMFA2 flag is set to 1.

Bit 6**IMIEA2 Description**

| | | |
|---|---|-----------------|
| 0 | IMIA2 interrupt requested by IMFA2 flag is disabled | (Initial value) |
| 1 | IMIA2 interrupt requested by IMFA2 flag is enabled | |

Bit 5—Input Capture/Compare Match Interrupt Enable A1 (IMIEA1): Enables or disables the interrupt requested by the IMFA1 flag when IMFA1 is set to 1.

Bit 5**IMIEA1 Description**

| | | |
|---|---|-----------------|
| 0 | IMIA1 interrupt requested by IMFA1 flag is disabled | (Initial value) |
| 1 | IMIA1 interrupt requested by IMFA1 flag is enabled | |

Bit 4—Input Capture/Compare Match Interrupt Enable A0 (IMIEA0): Enables or disables the interrupt requested by the IMFA0 flag when IMFA0 is set to 1.

Bit 4**IMIEA0 Description**

| | | |
|---|---|-----------------|
| 0 | IMIA0 interrupt requested by IMFA0 flag is disabled | (Initial value) |
| 1 | IMIA0 interrupt requested by IMFA0 flag is enabled | |

Bit 3—Reserved: This bit cannot be modified and is always read as 1.

Bit 2—Input Capture/Compare Match Flag A2 (IMFA2): This status flag indicates GRA2 compare match or input capture events.

Bit 2**IMFA2 Description**

| | | |
|---|--|-----------------|
| 0 | [Clearing condition] Read IMFA2 flag when IMFA2 = 1, then write 0 in IMFA2 flag | (Initial value) |
| 1 | [Setting conditions] <ul style="list-style-type: none">• 16TCNT2 = GRA2 when GRA2 functions as an output compare register• 16TCNT2 value is transferred to GRA2 by an input capture signal when GRA2 functions as an input capture register | |

Bit 1—Input Capture/Compare Match Flag A1 (IMFA1): This status flag indicates GRA1 compare match or input capture events.

| Bit 1 IMFA1 | Description | |
|----------------|---|-----------------|
| 0 | [Clearing condition] Read IMFA1 flag when IMFA1 = 1, then write 0 in IMFA1 flag | (Initial value) |
| 1 | [Setting conditions] <ul style="list-style-type: none"> • 16TCNT1 = GRA1 when GRA1 functions as an output compare register • 16TCNT1 value is transferred to GRA1 by an input capture signal when GRA1 functions as an input capture register | |

Bit 0—Input Capture/Compare Match Flag A0 (IMFA0): This status flag indicates GRA0 compare match or input capture events.

| Bit 0 IMFA0 | Description | |
|----------------|---|-----------------|
| 0 | [Clearing condition] Read IMFA0 flag when IMFA0 = 1, then write 0 in IMFA0 flag | (Initial value) |
| 1 | [Setting conditions] <ul style="list-style-type: none"> • 16TCNT0 = GRA0 when GRA0 functions as an output compare register • 16TCNT0 value is transferred to GRA0 by an input capture signal when GRA0 functions as an input capture register | |

8.2.5 Timer Interrupt Status Register B (TISR_B)

TISR_B is an 8-bit readable/writable register that indicates GRB compare match or input capture and enables or disables GRB compare match and input capture interrupt requests.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|---|--------|--------|--------|---|--------|--------|--------|
| | — | IMIEB2 | IMIEB1 | IMIEB0 | — | IMFB2 | IMFB1 | IMFB0 |
| Initial value | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| Read/Write | — | R/W | R/W | R/W | — | R/(W)* | R/(W)* | R/(W)* |

Reserved bit
 Input capture/compare match interrupt enable B2 to B0
 These bits enable or disable interrupts by the IMFB flags

Reserved bit
 Input capture/compare match flags B2 to B0
 Status flags indicating GRB compare match or input capture

Note: * Only 0 can be written, to clear the flag.

TISR_B is initialized to H'88 by a reset and in standby mode.

Bit 7—Reserved: This bit cannot be modified and is always read as 1.

Bit 6—Input Capture/Compare Match Interrupt Enable B2 (IMIEB2): Enables or disables the interrupt requested by the IMFB2 when IMFB2 flag is set to 1.

Bit 6

IMIEB2 Description

| | | |
|---|--|-----------------|
| 0 | IMIEB2 interrupt requested by IMFB2 flag is disabled | (Initial value) |
| 1 | IMIEB2 interrupt requested by IMFB2 flag is enabled | |

Bit 5—Input Capture/Compare Match Interrupt Enable B1 (IMIEB1): Enables or disables the interrupt requested by the IMFB1 when IMFB1 flag is set to 1.

Bit 5**IMIEB1 Description**

| | | |
|---|---|-----------------|
| 0 | IMIB1 interrupt requested by IMFB1 flag is disabled | (Initial value) |
| 1 | IMIB1 interrupt requested by IMFB1 flag is enabled | |

Bit 4—Input Capture/Compare Match Interrupt Enable B0 (IMIEB0): Enables or disables the interrupt requested by the IMFB0 when IMFB0 flag is set to 1.

Bit 4**IMIEB0 Description**

| | | |
|---|---|-----------------|
| 0 | IMIB0 interrupt requested by IMFB0 flag is disabled | (Initial value) |
| 1 | IMIB0 interrupt requested by IMFB0 flag is enabled | |

Bit 3—Reserved: This bit cannot be modified and is always read as 1.

Bit 2—Input Capture/Compare Match Flag B2 (IMFB2): This status flag indicates GRB2 compare match or input capture events.

Bit 2**IMFB2 Description**

| | | |
|---|---|-----------------|
| 0 | [Clearing condition] Read IMFB2 flag when IMFB2 = 1, then write 0 in IMFB2 flag | (Initial value) |
| 1 | [Setting conditions] <ul style="list-style-type: none"> • 16TCNT2 = GRB2 when GRB2 functions as an output compare register • 16TCNT2 value is transferred to GRB2 by an input capture signal when GRB2 functions as an input capture register | |

Bit 1—Input Capture/Compare Match Flag B1 (IMFB1): This status flag indicates GRB1 compare match or input capture events.

| Bit 1 IMFB1 | Description | |
|----------------|--|-----------------|
| 0 | [Clearing condition] Read IMFB1 flag when IMFB1 = 1, then write 0 in IMFB1 flag | (Initial value) |
| 1 | [Setting conditions] <ul style="list-style-type: none">• 16TCNT1 = GRB1 when GRB1 functions as an output compare register• 16TCNT1 value is transferred to GRB1 by an input capture signal when GRB1 functions as an input capture register | |

Bit 0—Input Capture/Compare Match Flag B0 (IMFB0): This status flag indicates GRB0 compare match or input capture events.

| Bit 0 IMFB0 | Description | |
|----------------|--|-----------------|
| 0 | [Clearing condition] Read IMFB0 flag when IMFB0 = 1, then write 0 in IMFB0 flag | (Initial value) |
| 1 | [Setting conditions] <ul style="list-style-type: none">• 16TCNT0 = GRB0 when GRB0 functions as an output compare register• 16TCNT0 value is transferred to GRB0 by an input capture signal when GRB0 functions as an input capture register | |

8.2.6 Timer Interrupt Status Register C (TISRC)

TISRC is an 8-bit readable/writable register that indicates 16TCNT overflow or underflow and enables or disables overflow interrupt requests.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|---|-------|-------|-------|---|--------|--------|--------|
| | — | OVIE2 | OVIE1 | OVIE0 | — | OVF2 | OVF1 | OVF0 |
| Initial value | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| Read/Write | — | R/W | R/W | R/W | — | R/(W)* | R/(W)* | R/(W)* |

Reserved bit
 Overflow interrupt enable 2 to 0
 These bits enable or disable interrupts by the OVF flags

Reserved bit
 Overflow flags 2 to 0
 Status flags indicating interrupts by OVF flags

Note: * Only 0 can be written, to clear the flag.

TISRC is initialized to H'88 by a reset and in standby mode.

Bit 7—Reserved: This bit cannot be modified and is always read as 1.

Bit 6—Overflow Interrupt Enable 2 (OVIE2): Enables or disables the interrupt requested by the OVF2 when OVF2 flag is set to 1.

Bit 6

| OVIE2 | Description |
|-------|---|
| 0 | OVI2 interrupt requested by OVF2 flag is disabled (Initial value) |
| 1 | OVI2 interrupt requested by OVF2 flag is enabled |

Bit 5—Overflow Interrupt Enable 1 (OVIE1): Enables or disables the interrupt requested by the OVF1 when OVF1 flag is set to 1.

| Bit 5 OVIE1 | Description | |
|----------------|---|-----------------|
| 0 | OVI1 interrupt requested by OVF1 flag is disabled | (Initial value) |
| 1 | OVI1 interrupt requested by OVF1 flag is enabled | |

Bit 4—Overflow Interrupt Enable 0 (OVIE0): Enables or disables the interrupt requested by the OVF0 when OVF0 flag is set to 1.

| Bit 4 OVIE0 | Description | |
|----------------|---|-----------------|
| 0 | OVI0 interrupt requested by OVF0 flag is disabled | (Initial value) |
| 1 | OVI0 interrupt requested by OVF0 flag is enabled | |

Bit 3—Reserved: This bit cannot be modified and is always read as 1.

Bit 2—Overflow Flag 2 (OVF2): This status flag indicates 16TCNT2 overflow.

| Bit 2 OVF2 | Description | |
|---------------|---|-----------------|
| 0 | [Clearing condition] Read OVF2 flag when OVF2 = 1, then write 0 in OVF2 flag | (Initial value) |
| 1 | [Setting condition] 16TCNT2 overflowed from H'FFFF to H'0000, or underflowed from H'0000 to H'FFFF | |

Note: 16TCNT underflow occurs when 16TCNT operates as an up/down-counter. Underflow occurs only when channel 2 operates in phase counting mode (MDF = 1 in TMDR).

Bit 1—Overflow Flag 1 (OVF1): This status flag indicates 16TCNT1 overflow.

| Bit 1 OVF1 | Description | |
|---------------|---|-----------------|
| 0 | [Clearing condition] Read OVF1 flag when OVF1 = 1, then write 0 in OVF1 flag | (Initial value) |
| 1 | [Setting condition] 16TCNT1 overflowed from H'FFFF to H'0000 | |

Bit 0—Overflow Flag 0 (OVF0): This status flag indicates 16TCNT0 overflow.

Bit 0

| OVF0 | Description | |
|------|---|-----------------|
| 0 | [Clearing condition] Read OVF0 flag when OVF0 = 1, then write 0 in OVF0 flag | (Initial value) |
| 1 | [Setting condition] 16TCNT0 overflowed from H'FFFF to H'0000 | |

8.2.7 Timer Counters (16TCNT)

16TCNT is a 16-bit counter. The 16-bit timer has three 16TCNTs, one for each channel.

| Channel | Abbreviation | Function |
|---------|--------------|---|
| 0 | 16TCNT0 | Up-counter |
| 1 | 16TCNT1 | |
| 2 | 16TCNT2 | Phase counting mode: up/down-counter Other modes: up-counter |

| | | | | | | | | | | | | | | | | |
|---------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Initial value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Each 16TCNT is a 16-bit readable/writable register that counts pulse inputs from a clock source. The clock source is selected by bits TPSC2 to TPSC0 in 16TCR.

16TCNT0 and 16TCNT1 are up-counters. 16TCNT2 is an up/down-counter in phase counting mode and an up-counter in other modes.

16TCNT can be cleared to H'0000 by compare match with GRA or GRB or by input capture to GRA or GRB (counter clearing function).

When 16TCNT overflows (changes from H'FFFF to H'0000), the OVF flag is set to 1 in TISRC of the corresponding channel.

When 16TCNT underflows (changes from H'0000 to H'FFFF), the OVF flag is set to 1 in TISRC of the corresponding channel.

The 16TCNTs are linked to the CPU by an internal 16-bit bus and can be written or read by either word access or byte access.

Each 16TCNT is initialized to H'0000 by a reset and in standby mode.

8.2.8 General Registers (GRA, GRB)

The general registers are 16-bit registers. The 16-bit timer has 6 general registers, two in each channel.

| Channel | Abbreviation | Function |
|---------|--------------|---------------------------------------|
| 0 | GRA0, GRB0 | Output compare/input capture register |
| 1 | GRA1, GRB1 | |
| 2 | GRA2, GRB2 | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Initial value | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

A general register is a 16-bit readable/writable register that can function as either an output compare register or an input capture register. The function is selected by settings in TIOR.

When a general register is used as an output compare register, its value is constantly compared with the 16TCNT value. When the two values match (compare match), the IMFA or IMFB flag is set to 1 in TISRA/TISRB. Compare match output can be selected in TIOR.

When a general register is used as an input capture register, an external input capture signal are detected and the current 16TCNT value is stored in the general register. The corresponding IMFA or IMFB flag in TISRA/TISRB is set to 1 at the same time. The edges of the input capture signal are selected in TIOR.

TIOR settings are ignored in PWM mode.

General registers are linked to the CPU by an internal 16-bit bus and can be written or read by either word access or byte access.

General registers are set as output compare registers (with no pin output) and initialized to H'FFFF by a reset and in standby mode.

8.2.9 Timer Control Registers (16TCR)

16TCR is an 8-bit register. The 16-bit timer has three 16TCRs, one in each channel.

| Channel | Abbreviation | Function |
|---------|--------------|---|
| 0 | 16TCR0 | 16TCR controls the timer counter. The 16TCRs in all channels are functionally identical. When phase counting mode is selected in channel 2, the settings of bits CKEG1 and CKEG0 and TPSC2 to TPSC0 in 16TCR2 are ignored |
| 1 | 16TCR1 | |
| 2 | 16TCR2 | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|---|-------|-------|-------|-------|-------|-------|-------|
| | — | CCLR1 | CCLR0 | CKEG1 | CKEG0 | TPSC2 | TPSC1 | TPSC0 |
| Initial value | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Read/Write | — | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Reserved bit
Counter clear 1/0
 These bits select the counter clear source

Clock edge 1/0
 These bits select external clock edges

Timer prescaler 2 to 0
 These bits select the timer counter clock

Each 16TCR is an 8-bit readable/writable register that selects the timer counter clock source, selects the edge or edges of external clock sources, and selects how the counter is cleared.

16TCR is initialized to H'80 by a reset and in standby mode.

Bit 7—Reserved: This bit cannot be modified and is always read as 1.

Bits 6 and 5—Counter Clear 1 and 0 (CCLR1, CCLR0): These bits select how 16TCNT is cleared.

| Bit 6 CCLR1 | Bit 5 CCLR0 | Description |
|----------------|----------------|--|
| 0 | 0 | 16TCNT is not cleared (Initial value) |
| | 1 | 16TCNT is cleared by GRA compare match or input capture* ¹ |
| 1 | 0 | 16TCNT is cleared by GRB compare match or input capture* ¹ |
| | 1 | Synchronous clear: 16TCNT is cleared in synchronization with other synchronized timers* ² |

Notes: 1. 16TCNT is cleared by compare match when the general register functions as an output compare register, and by input capture when the general register functions as an input capture register.
2. Selected in TSNC.

Bits 4 and 3—Clock Edge 1 and 0 (CKEG1, CKEG0): These bits select external clock input edges when an external clock source is used.

| Bit 4 CKEG1 | Bit 3 CKEG0 | Description |
|----------------|----------------|------------------------------------|
| 0 | 0 | Count rising edges (Initial value) |
| | 1 | Count falling edges |
| 1 | — | Count both edges |

When channel 2 is set to phase counting mode, bits CKEG1 and CKEG0 in 16TCR2 are ignored. Phase counting takes precedence.

Bits 2 to 0—Timer Prescaler 2 to 0 (TPSC2 to TPSC0): These bits select the counter clock source.

| Bit 2 TPSC2 | Bit 1 TPSC1 | Bit 0 TPSC0 | Function |
|----------------|----------------|----------------|--|
| 0 | 0 | 0 | Internal clock: ϕ (Initial value) |
| | | 1 | Internal clock: $\phi/2$ |
| | 1 | 0 | Internal clock: $\phi/4$ |
| | | 1 | Internal clock: $\phi/8$ |
| 1 | 0 | 0 | External clock A: TCLKA input |
| | | 1 | External clock B: TCLKB input |
| | 1 | 0 | External clock C: TCLKC input |
| | | 1 | External clock D: TCLKD input |

When bit TPSC2 is cleared to 0 an internal clock source is selected, and the timer counts only falling edges. When bit TPSC2 is set to 1 an external clock source is selected, and the timer counts the edges selected by bits CKEG1 and CKEG0.

When channel 2 is set to phase counting mode (MDF = 1 in TMDR), the settings of bits TPSC2 to TPSC0 in 16TCR2 are ignored. Phase counting takes precedence.

8.2.10 Timer I/O Control Register (TIOR)

TIOR is an 8-bit register. The 16-bit timer has three TIORs, one in each channel.

| Channel | Abbreviation | Function |
|---------|--------------|---|
| 0 | TIOR0 | TIOR controls the general registers. Some functions differ in PWM mode. |
| 1 | TIOR1 | |
| 2 | TIOR2 | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|---|------|------|------|---|------|------|------|
| | — | IOB2 | IOB1 | IOB0 | — | IOA2 | IOA1 | IOA0 |
| Initial value | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| Read/Write | — | R/W | R/W | R/W | — | R/W | R/W | R/W |

Reserved bit (Bit 7)
I/O control B2 to B0 (Bits 6, 5, 4)
 These bits select GRB functions
Reserved bit (Bit 3)
I/O control A2 to A0 (Bits 2, 1, 0)
 These bits select GRA functions

Each TIOR is an 8-bit readable/writable register that selects the output compare or input capture function for GRA and GRB, and specifies the functions of the TIORA and TIORB pins. If the output compare function is selected, TIOR also selects the type of output. If input capture is selected, TIOR also selects the edges of the input capture signal.

TIOR is initialized to H'88 by a reset and in standby mode.

Bit 7—Reserved: This bit cannot be modified and is always read as 1.

Bits 6 to 4—I/O Control B2 to B0 (IOB2 to IOB0): These bits select the GRB function.

| Bit 6 IOB2 | Bit 5 IOB1 | Bit 4 IOB0 | Function | |
|---------------|---------------|---------------|--------------------------------------|---|
| 0 | 0 | 0 | GRB is an output compare register | No output at compare match (Initial value) |
| | | 1 | | 0 output at GRB compare match* ¹ |
| | 1 | 0 | | 1 output at GRB compare match* ¹ |
| | | 1 | | Output toggles at GRB compare match (1 output in channel 2)* ¹ , * ² |
| 1 | 0 | 0 | GRB is an input capture register | GRB captures rising edge of input |
| | | 1 | | GRB captures falling edge of input |
| | 1 | 0 | | GRB captures both edges of input |
| | | 1 | | |

- Notes: 1. After a reset, the output conforms to the TOLR setting until the first compare match.
2. Channel 2 output cannot be toggled by compare match. When this setting is made, 1 output is selected automatically.

Bit 3—Reserved: This bit cannot be modified and is always read as 1.

Bits 2 to 0—I/O Control A2 to A0 (IOA2 to IOA0): These bits select the GRA function.

| Bit 2 IOA2 | Bit 1 IOA1 | Bit 0 IOA0 | Function | |
|---------------|---------------|---------------|--------------------------------------|---|
| 0 | 0 | 0 | GRA is an output compare register | No output at compare match (Initial value) |
| | | 1 | | 0 output at GRA compare match* ¹ |
| | 1 | 0 | | 1 output at GRA compare match* ¹ |
| | | 1 | | Output toggles at GRA compare match (1 output in channel 2)* ¹ , * ² |
| 1 | 0 | 0 | GRA is an input capture register | GRA captures rising edge of input |
| | | 1 | | GRA captures falling edge of input |
| | 1 | 0 | | GRA captures both edges of input |
| | | 1 | | |

- Notes: 1. After a reset, the output conforms to the TOLR setting until the first compare match.
2. Channel 2 output cannot be toggled by compare match. When this setting is made, 1 output is selected automatically.

8.2.11 Timer Output Level Setting Register C (TOLR)

TOLR is an 8-bit write-only register that selects the timer output level for channels 0 to 2.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|---|---|------|------|------|------|------|------|
| | — | — | TOB2 | TOA2 | TOB1 | TOA1 | TOB0 | TOA0 |
| Initial value | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| Read/Write | — | — | W | W | W | W | W | W |

Reserved bits

Output level setting A2 to A0, B2 to B0

These bits set the levels of the timer outputs (TIOCA₂ to TIOCA₀, and TIOCB₂ to TIOCB₀)

A TOLR setting can only be made when the corresponding bit in TSTR is 0.

TOLR is a write-only register, and cannot be read. If it is read, all bits will return a value of 1.

TOLR is initialized to H'C0 by a reset and in standby mode.

Bits 7 and 6—Reserved: These bits cannot be modified.

Bit 5—Output Level Setting B2 (TOB2): Sets the value of timer output TIOCB₂.

Bit 5

| TOB2 | Description |
|------|---|
| 0 | TIOCB ₂ is 0 (Initial value) |
| 1 | TIOCB ₂ is 1 |

Bit 4—Output Level Setting A2 (TOA2): Sets the value of timer output TIOCA₂.

Bit 4

| TOA2 | Description |
|------|---|
| 0 | TIOCA ₂ is 0 (Initial value) |
| 1 | TIOCA ₂ is 1 |

Bit 3—Output Level Setting B1 (TOB1): Sets the value of timer output TIOCB₁.

| Bit 3 TOB1 | Description | |
|---------------|-------------------------|-----------------|
| 0 | TIOCB ₁ is 0 | (Initial value) |
| 1 | TIOCB ₁ is 1 | |

Bit 2—Output Level Setting A1 (TOA1): Sets the value of timer output TIOCA₁.

| Bit 2 TOA1 | Description | |
|---------------|-------------------------|-----------------|
| 0 | TIOCA ₁ is 0 | (Initial value) |
| 1 | TIOCA ₁ is 1 | |

Bit 1—Output Level Setting B0 (TOB0): Sets the value of timer output TIOCB₀.

| Bit 0 TOB0 | Description | |
|---------------|-------------------------|-----------------|
| 0 | TIOCB ₀ is 0 | (Initial value) |
| 1 | TIOCB ₀ is 1 | |

Bit 0—Output Level Setting A0 (TOA0): Sets the value of timer output TIOCA₀.

| Bit 0 TOA0 | Description | |
|---------------|-------------------------|-----------------|
| 0 | TIOCA ₀ is 0 | (Initial value) |
| 1 | TIOCA ₀ is 1 | |

8.3 CPU Interface

8.3.1 16-Bit Accessible Registers

The timer counters (16TCNTs), general registers A and B (GRAs and GRBs) are 16-bit registers, and are linked to the CPU by an internal 16-bit data bus. These registers can be written or read a word at a time, or a byte at a time.

Figures 8.4 and 8.5 show examples of word read/write access to a timer counter (16TCNT). Figures 8.6 to 8.9 show examples of byte read/write access to 16TCNTH and 16TCNTL.

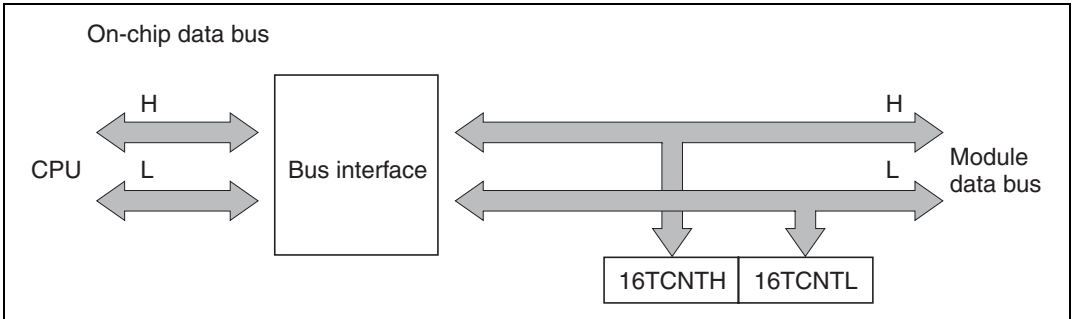


Figure 8.4 16TCNT Access Operation [CPU → 16TCNT (Word)]

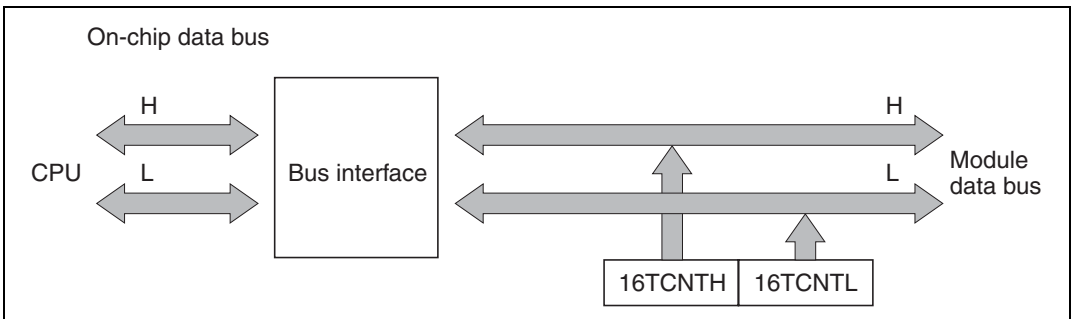


Figure 8.5 Access to Timer Counter (CPU Reads 16TCNT, Word)

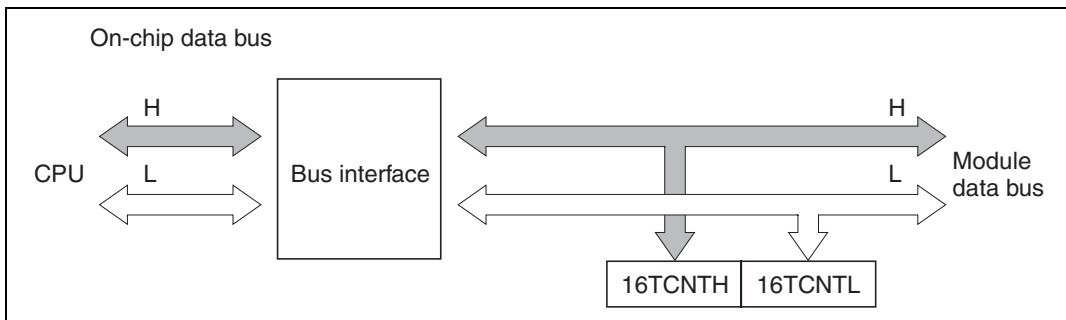


Figure 8.6 Access to Timer Counter H (CPU Writes to 16TCNTH, Upper Byte)

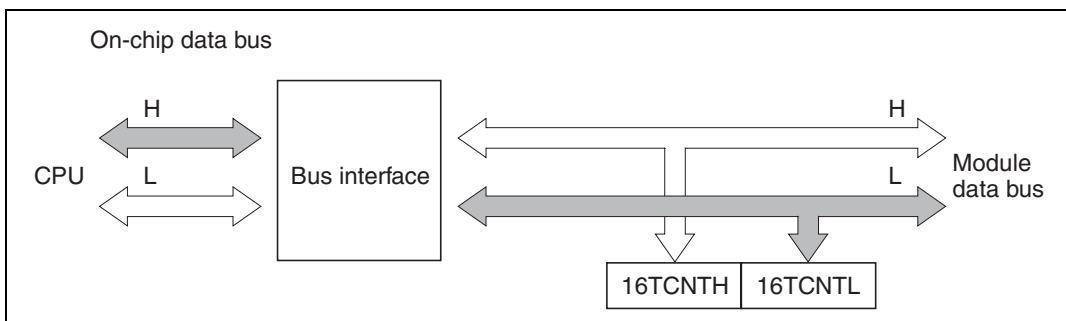


Figure 8.7 Access to Timer Counter L (CPU Writes to 16CNTL, Lower Byte)

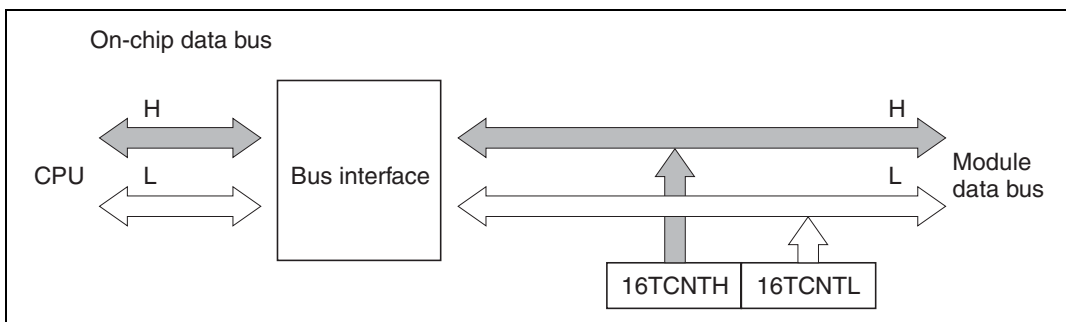


Figure 8.8 Access to Timer Counter H (CPU Reads 16TCNTH, Upper Byte)

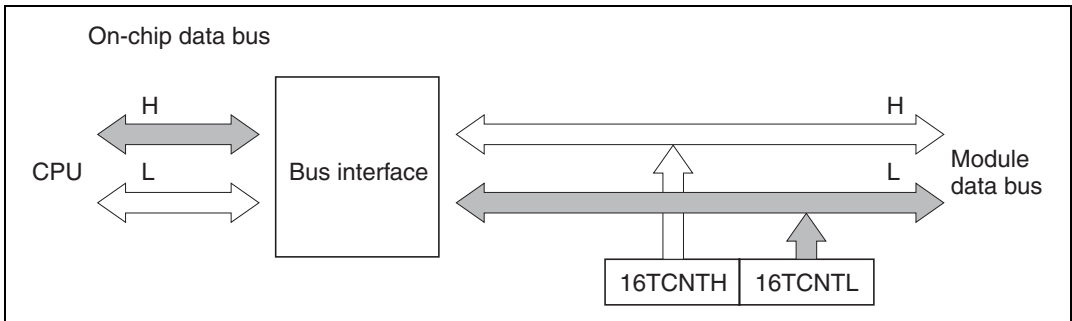


Figure 8.9 Access to Timer Counter L (CPU Reads 16TCNTL, Lower Byte)

8.3.2 8-Bit Accessible Registers

The registers other than the timer counters and general registers are 8-bit registers. These registers are linked to the CPU by an internal 8-bit data bus.

Figures 8.10 and 8.11 show examples of byte read and write access to a 16TCR.

If a word-size data transfer instruction is executed, two byte transfers are performed.

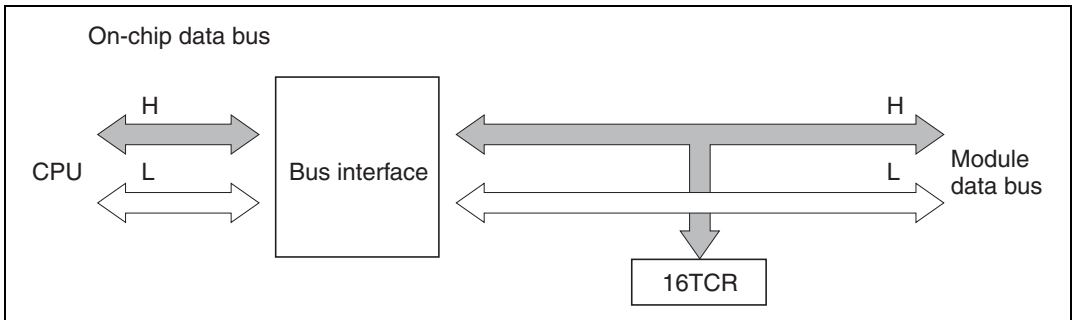


Figure 8.10 16TCR Access (CPU Writes to 16TCR)

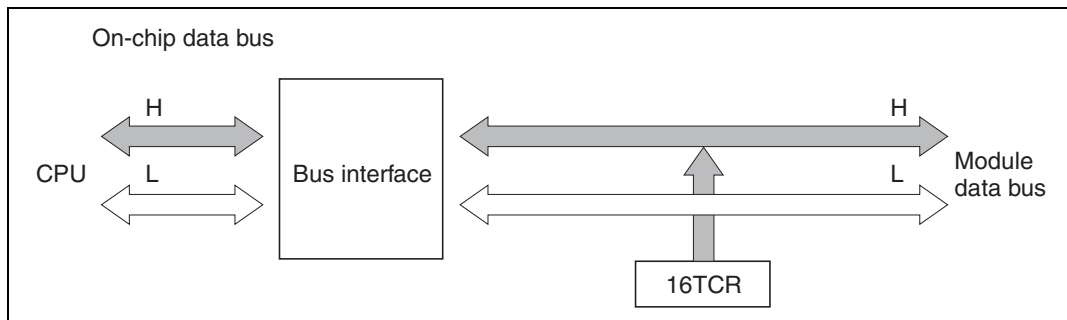


Figure 8.11 16TCR Access (CPU Reads 16TCR)

8.4 Operation

8.4.1 Overview

A summary of operations in the various modes is given below.

Normal Operation: Each channel has a timer counter and general registers. The timer counter counts up, and can operate as a free-running counter, periodic counter, or external event counter. GRA and GRB can be used for input capture or output compare.

Synchronous Operation: The timer counters in designated channels are preset synchronously. Data written to the timer counter in any one of these channels is simultaneously written to the timer counters in the other channels as well. The timer counters can also be cleared synchronously if so designated by the CCLR1 and CCLR0 bits in the TCRs.

PWM Mode: A PWM waveform is output from the TIOCA pin. The output goes to 1 at compare match A and to 0 at compare match B. The duty cycle can be varied from 0% to 100% depending on the settings of GRA and GRB. When a channel is set to PWM mode, its GRA and GRB automatically become output compare registers.

Phase Counting Mode: The phase relationship between two clock signals input at TCLKA and TCLKB is detected and 16TCNT2 counts up or down accordingly. When phase counting mode is selected TCLKA and TCLKB become clock input pins and 16TCNT2 operates as an up/down-counter.

8.4.2 Basic Functions

Counter Operation: When one of bits STR0 to STR2 is set to 1 in the timer start register (TSTR), the timer counter (16TCNT) in the corresponding channel starts counting. The counting can be free-running or periodic.

- Sample setup procedure for counter

Figure 8.12 shows a sample procedure for setting up a counter.

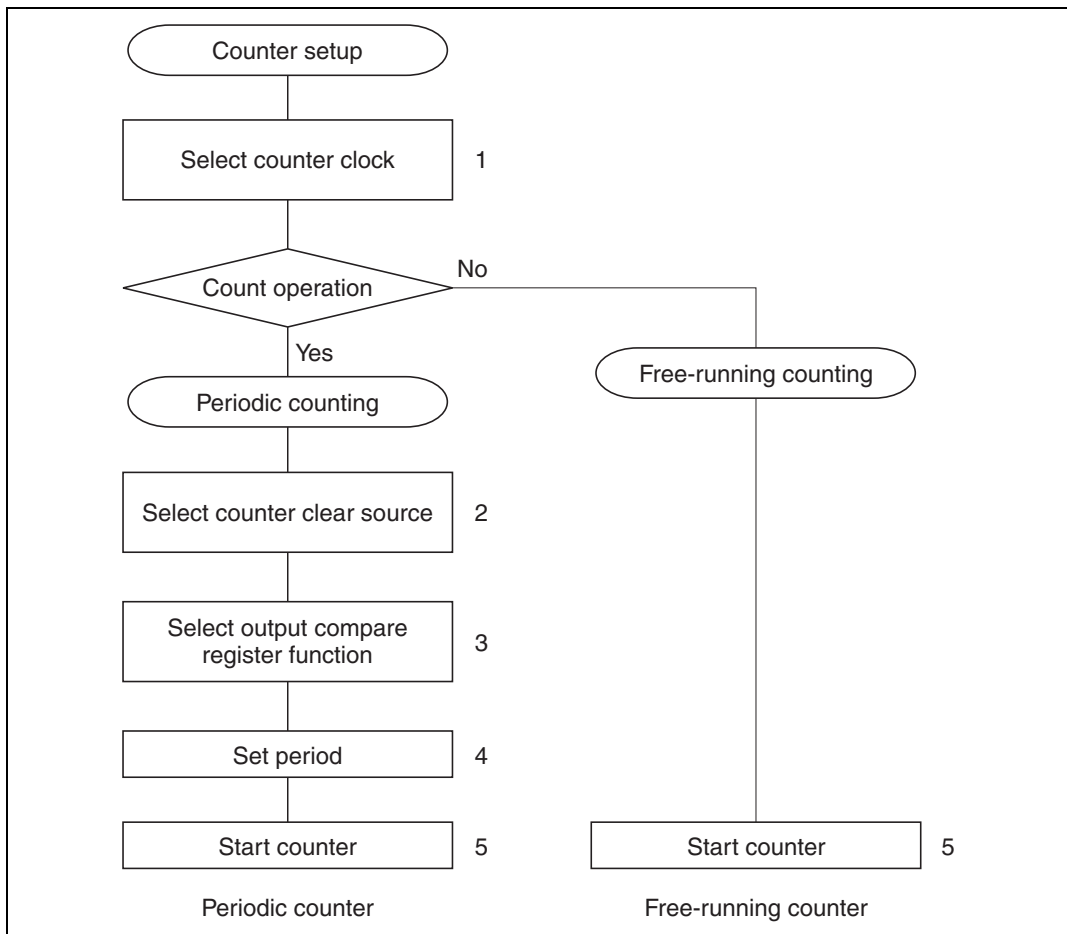


Figure 8.12 Counter Setup Procedure (Example)

1. Set bits TPSC2 to TPSC0 in 16TCR to select the counter clock source. If an external clock source is selected, set bits CKEG1 and CKEG0 in 16TCR to select the desired edge(s) of the external clock signal.
2. For periodic counting, set CCLR1 and CCLR0 in 16TCR to have 16TCNT cleared at GRA compare match or GRB compare match.
3. Set TIOR to select the output compare function of GRA or GRB, whichever was selected in step 2.
4. Write the count period in GRA or GRB, whichever was selected in step 2.

5. Set the STR bit to 1 in TSTR to start the timer counter.

- Free-running and periodic counter operation

A reset leaves the counters (16TCNTs) in 16-bit timer channels 0 to 2 all set as free-running counters. A free-running counter starts counting up when the corresponding bit in TSTR is set to 1. When the count overflows from H'FFFF to H'0000, the OVF flag is set to 1 in TISRC. After the overflow, the counter continues counting up from H'0000. Figure 8.13 illustrates free-running counting.

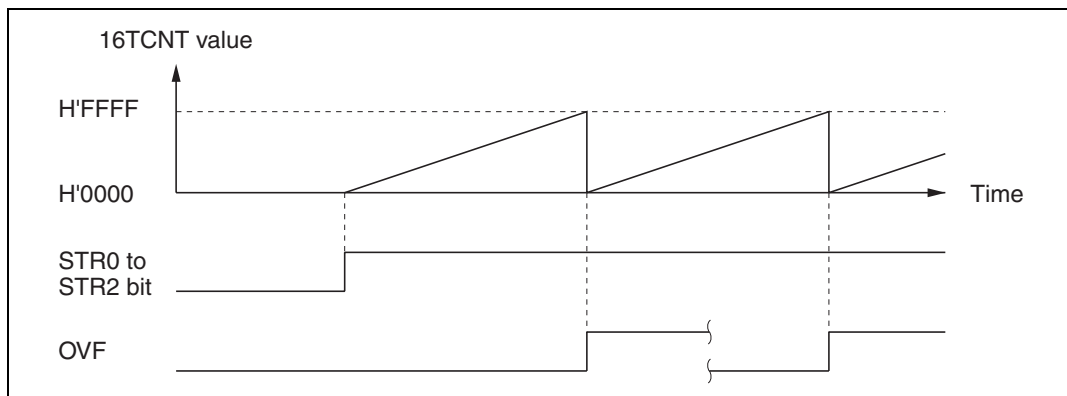


Figure 8.13 Free-Running Counter Operation

When a channel is set to have its counter cleared by compare match, in that channel 16TCNT operates as a periodic counter. Select the output compare function of GRA or GRB, set bit CCLR1 or CCLR0 in 16TCR to have the counter cleared by compare match, and set the count period in GRA or GRB. After these settings, the counter starts counting up as a periodic counter when the corresponding bit is set to 1 in TSTR. When the count matches GRA or GRB, the IMFA or IMFB flag is set to 1 in TISRA/TISRB and the counter is cleared to H'0000. If the corresponding IMIEA or IMIEB bit is set to 1 in TISRA/TISRB, a CPU interrupt is requested at this time. After the compare match, 16TCNT continues counting up from H'0000. Figure 8.14 illustrates periodic counting.

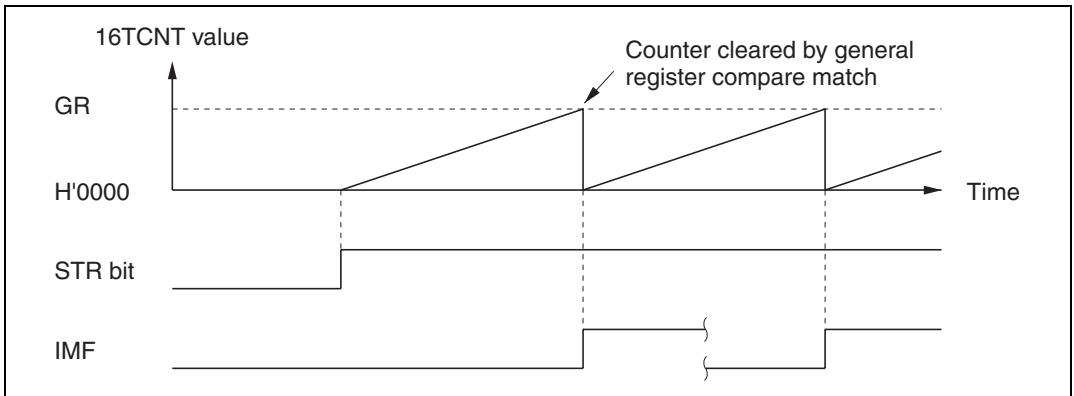


Figure 8.14 Periodic Counter Operation

- 16TCNT count timing

- Internal clock source

Bits TPSC2 to TPSC0 in 16TCR select the system clock (ϕ) or one of three internal clock sources obtained by prescaling the system clock ($\phi/2$, $\phi/4$, $\phi/8$).

Figure 8.15 shows the timing.

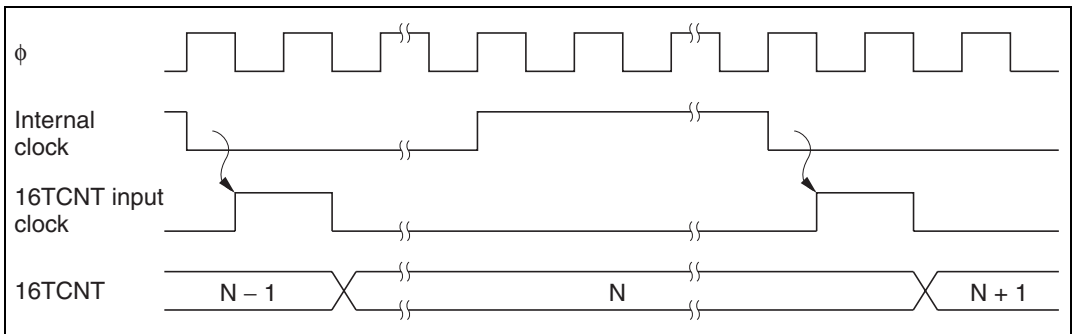


Figure 8.15 Count Timing for Internal Clock Sources

- External clock source

The external clock pin (TCLKA to TCLKD) can be selected by bits TPSC2 to TPSC0 in 16TCR, and the detected edge by bits CKEG1 and CKEG0. The rising edge, falling edge, or both edges can be selected.

The pulse width of the external clock signal must be at least 1.5 system clocks when a single edge is selected, and at least 2.5 system clocks when both edges are selected. Shorter pulses will not be counted correctly.

Figure 8.16 shows the timing when both edges are detected.

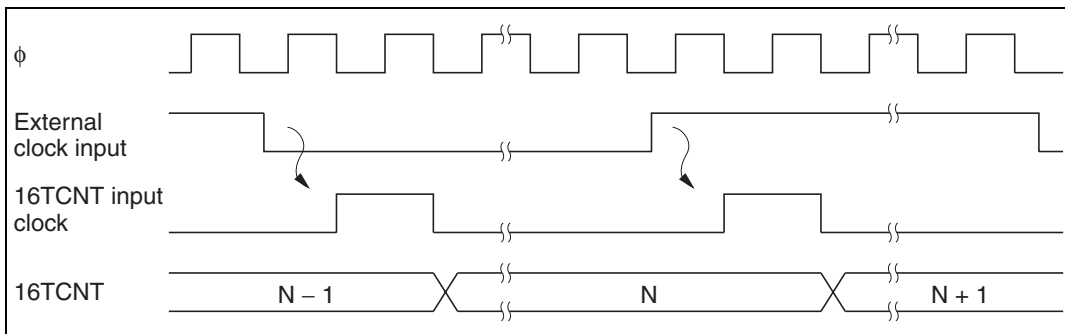


Figure 8.16 Count Timing for External Clock Sources (when Both Edges are Detected)

Waveform Output by Compare Match: In 16-bit timer channels 0, 1 compare match A or B can cause the output at the TIOCA or TIOCB pin to go to 0, go to 1, or toggle. In channel 2 the output can only go to 0 or go to 1.

- Sample setup procedure for waveform output by compare match

Figure 8.17 shows an example of the setup procedure for waveform output by compare match.

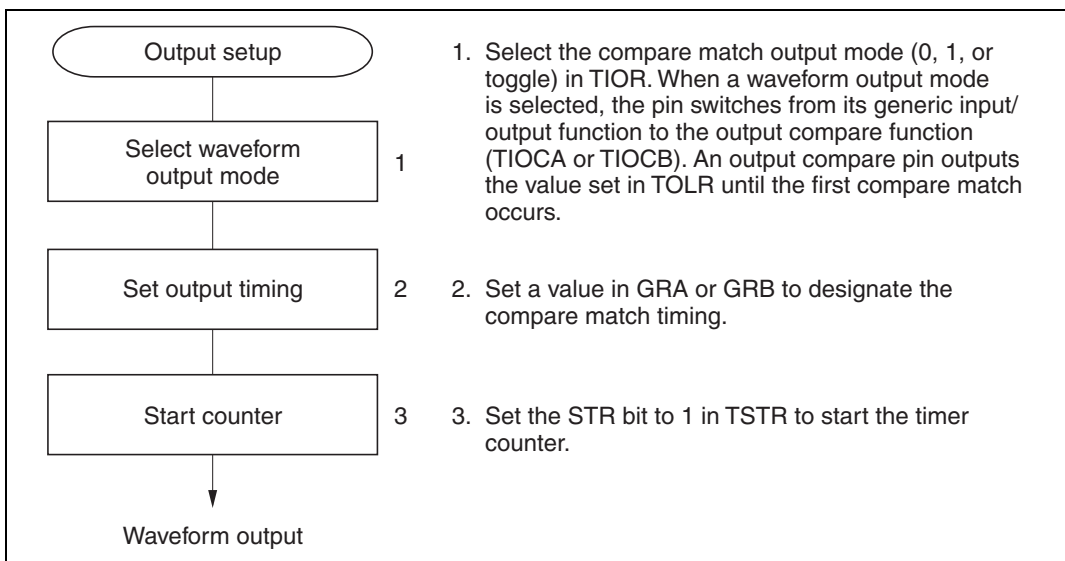


Figure 8.17 Setup Procedure for Waveform Output by Compare Match (Example)

- Examples of waveform output

Figure 8.18 shows examples of 0 and 1 output. 16TCNT operates as a free-running counter, 0 output is selected for compare match A, and 1 output is selected for compare match B. When the pin is already at the selected output level, the pin level does not change.

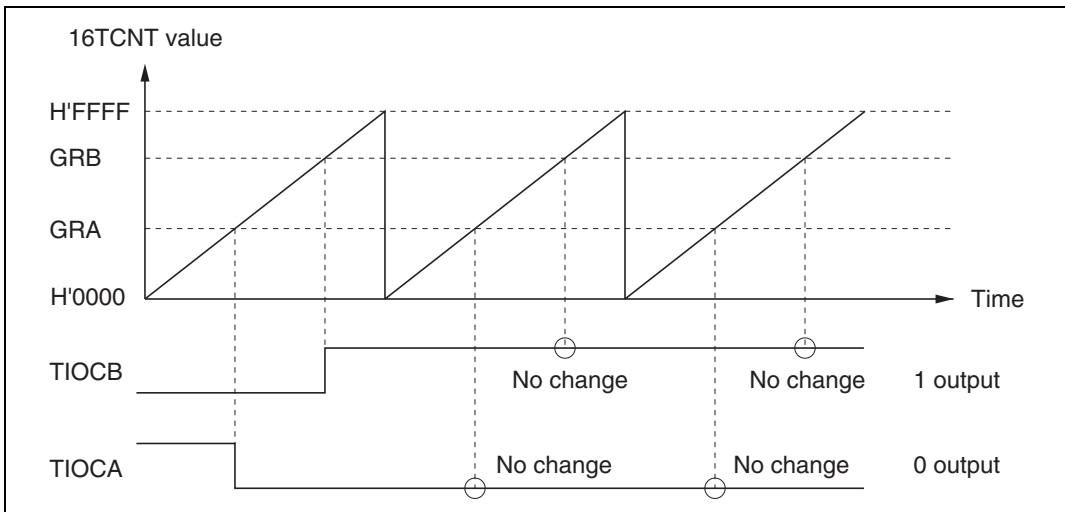


Figure 8.18 0 and 1 Output (TOA = 1, TOB = 0)

Figure 8.19 shows examples of toggle output. 16TCNT operates as a periodic counter, cleared by compare match B. Toggle output is selected for both compare match A and B.

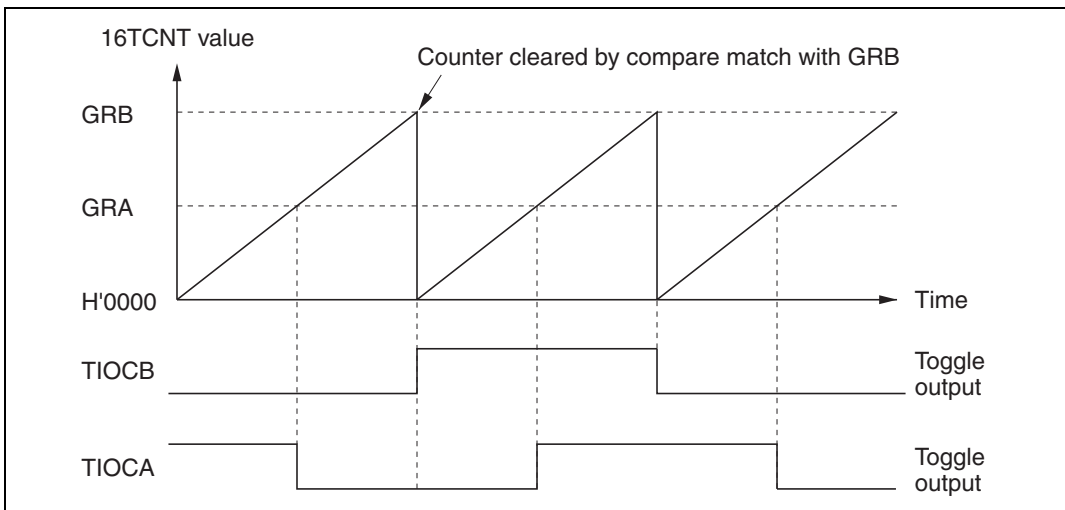


Figure 8.19 Toggle Output (TOA = 1, TOB = 0)

- Output compare output timing

The compare match signal is generated in the last state in which 16TCNT and the general register match (when 16TCNT changes from the matching value to the next value). When the compare match signal is generated, the output value selected in TIOR is output at the output compare pin (TIOCA or TIOCB). When 16TCNT matches a general register, the compare match signal is not generated until the next counter clock pulse.

Figure 8.20 shows the output compare timing.

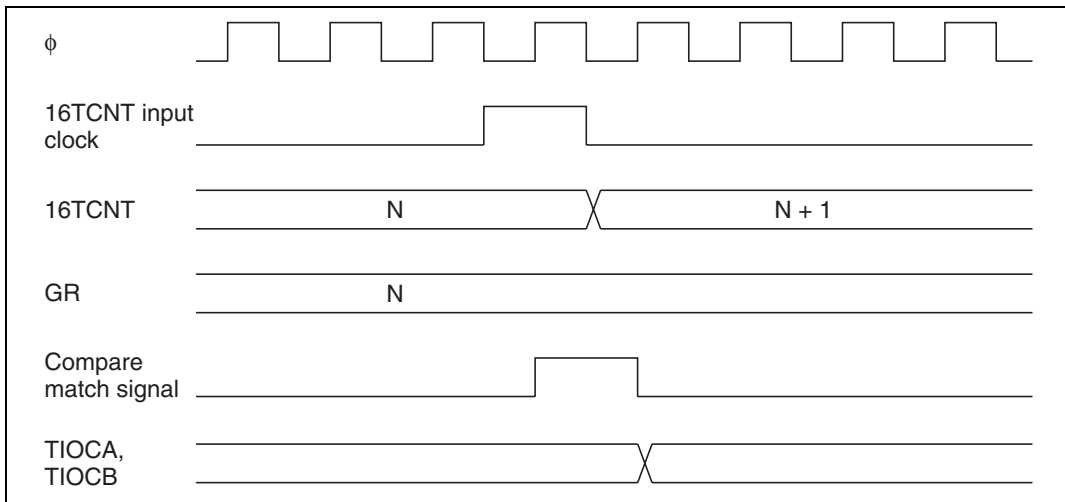


Figure 8.20 Output Compare Output Timing

Input Capture Function: The 16TCNT value can be transferred to a general register when an input edge is detected at an input capture input/output compare pin (TIOCA or TIOCB). Rising-edge, falling-edge, or both-edge detection can be selected. The input capture function can be used to measure pulse width or period.

- Sample setup procedure for input capture

Figure 8.21 shows a sample procedure for setting up input capture.

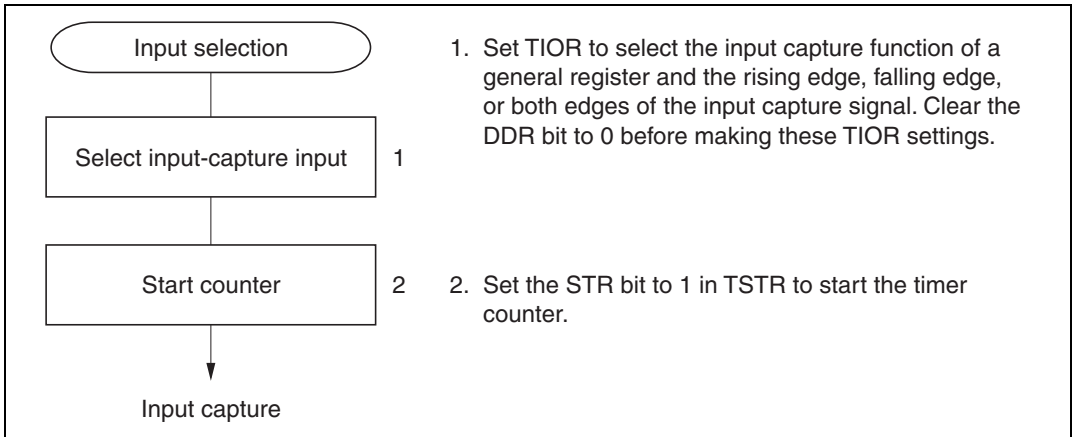


Figure 8.21 Setup Procedure for Input Capture (Example)

- Examples of input capture

Figure 8.22 illustrates input capture when the falling edge of TIOCB and both edges of TIOCA are selected as capture edges. 16TCNT is cleared by input capture into GRB.

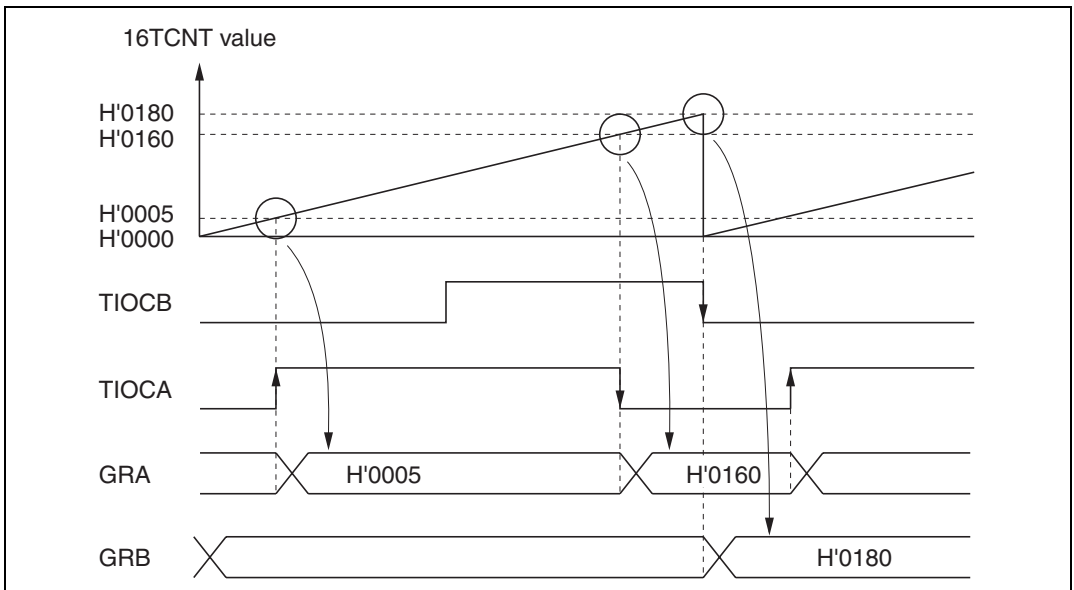


Figure 8.22 Input Capture (Example)

- Input capture signal timing

Input capture on the rising edge, falling edge, or both edges can be selected by settings in TIOR. Figure 8.23 shows the timing when the rising edge is selected. The pulse width of the input capture signal must be at least 1.5 system clocks for single-edge capture, and 2.5 system clocks for capture of both edges.

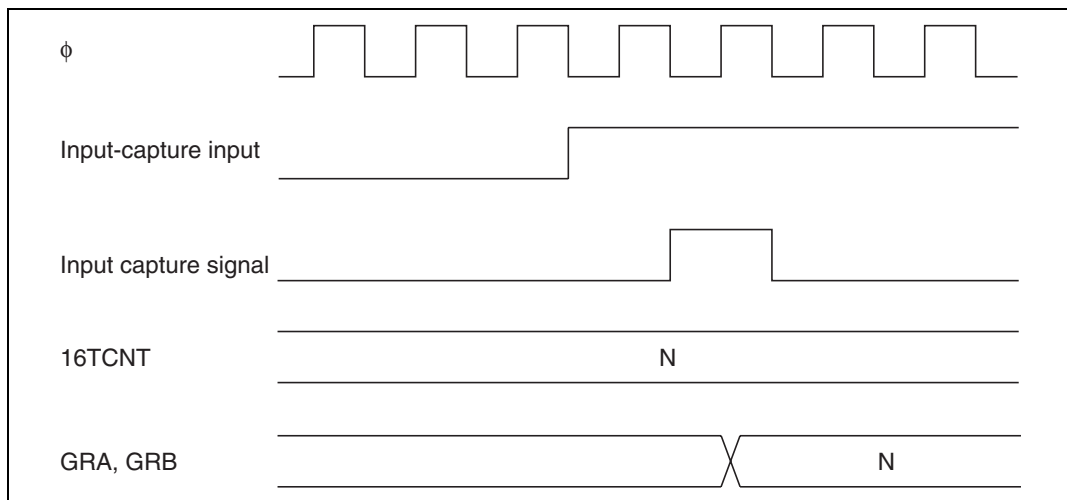


Figure 8.23 Input Capture Signal Timing

8.4.3 Synchronization

The synchronization function enables two or more timer counters to be synchronized by writing the same data to them simultaneously (synchronous preset). With appropriate 16TCR settings, two or more timer counters can also be cleared simultaneously (synchronous clear). Synchronization enables additional general registers to be associated with a single time base. Synchronization can be selected for all channels (0 to 2).

Sample Setup Procedure for Synchronization: Figure 8.24 shows a sample procedure for setting up synchronization.

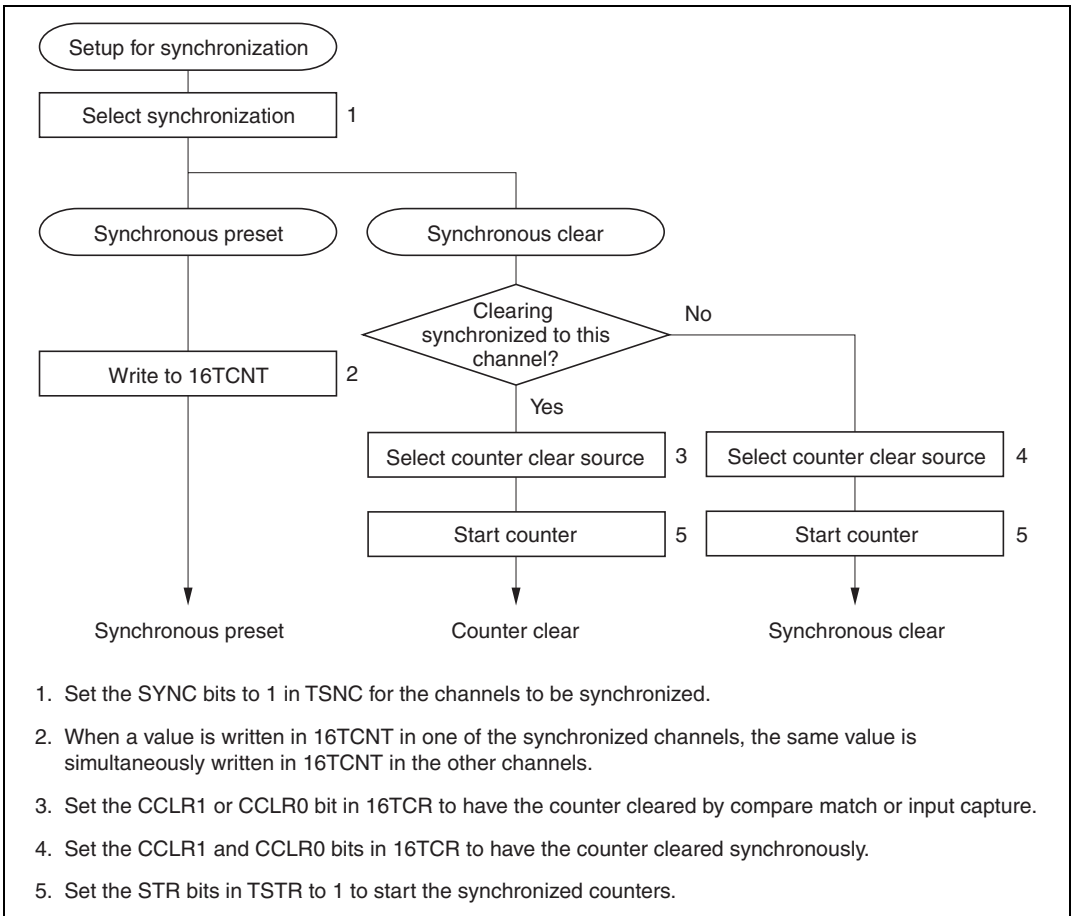


Figure 8.24 Setup Procedure for Synchronization (Example)

Example of Synchronization: Figure 8.25 shows an example of synchronization. Channels 0, 1, and 2 are synchronized, and are set to operate in PWM mode. Channel 0 is set for counter clearing by compare match with GRB0. Channels 1 and 2 are set for synchronous counter clearing. The timer counters in channels 0, 1, and 2 are synchronously preset, and are synchronously cleared by compare match with GRB0. A three-phase PWM waveform is output from pins TIOCA₀, TIOCA₁, and TIOCA₂. For further information on PWM mode, see section 8.4.4, PWM Mode.

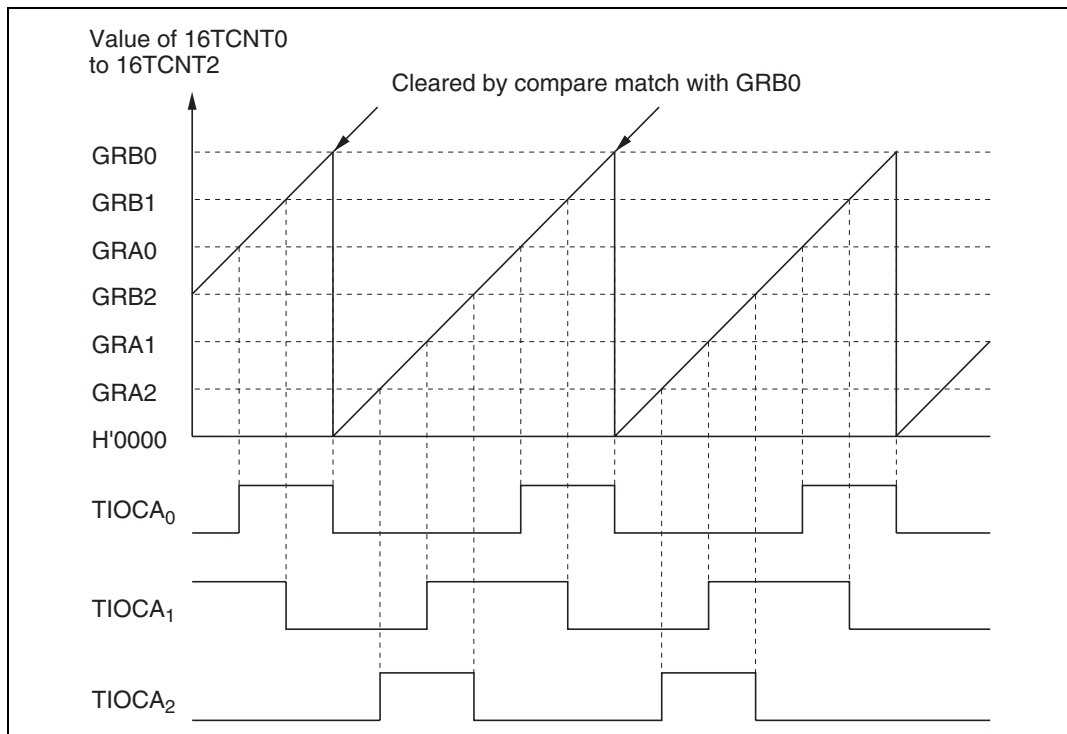


Figure 8.25 Synchronization (Example)

8.4.4 PWM Mode

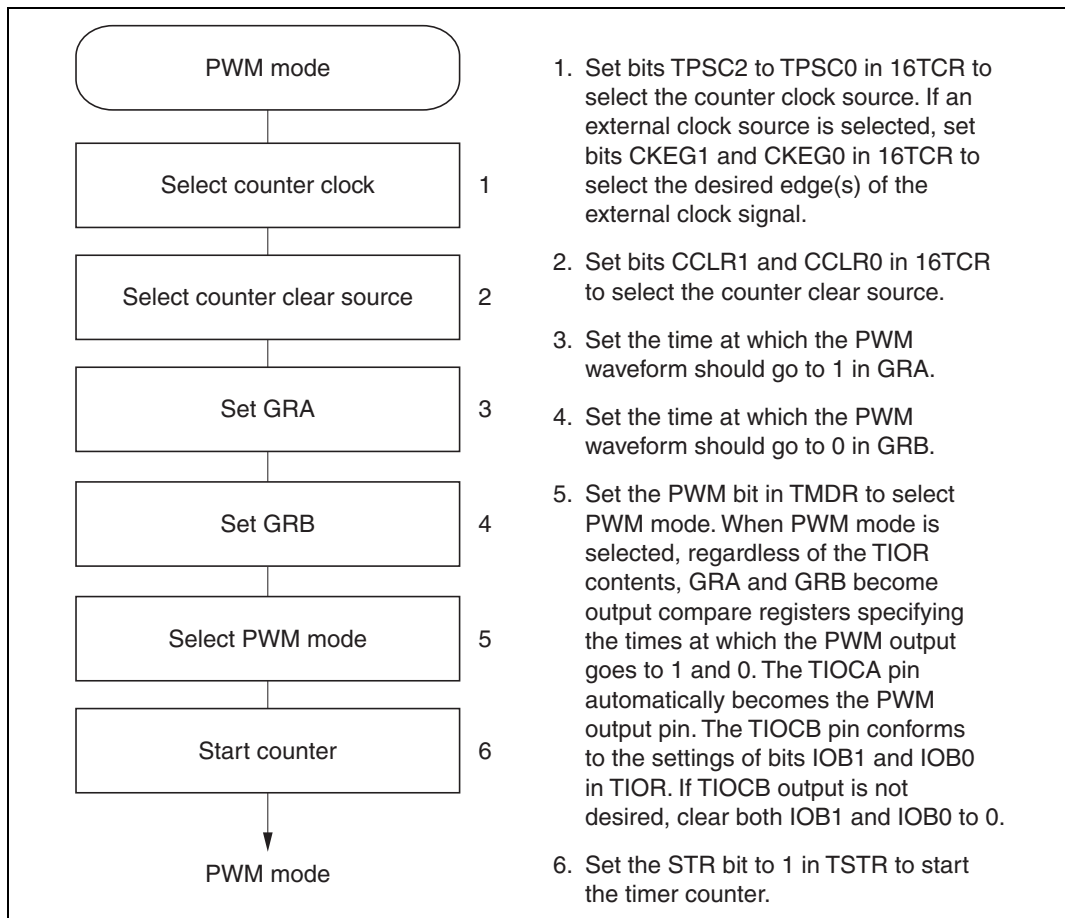
In PWM mode GRA and GRB are paired and a PWM waveform is output from the TIOCA pin. GRA specifies the time at which the PWM output changes to 1. GRB specifies the time at which the PWM output changes to 0. If either GRA or GRB compare match is selected as the counter clear source, a PWM waveform with a duty cycle from 0% to 100% is output at the TIOCA pin. PWM mode can be selected in all channels (0 to 2).

Table 8.4 summarizes the PWM output pins and corresponding registers. If the same value is set in GRA and GRB, the output does not change when compare match occurs.

Table 8.4 PWM Output Pins and Registers

| Channel | Output Pin | 1 Output | 0 Output |
|---------|--------------------|----------|----------|
| 0 | TIOCA ₀ | GRA0 | GRB0 |
| 1 | TIOCA ₁ | GRA1 | GRB1 |
| 2 | TIOCA ₂ | GRA2 | GRB2 |

Sample Setup Procedure for PWM Mode: Figure 8.26 shows a sample procedure for setting up PWM mode.

**Figure 8.26 Setup Procedure for PWM Mode (Example)**

Examples of PWM Mode: Figure 8.27 shows examples of operation in PWM mode. In PWM mode TIOCA becomes an output pin. The output goes to 1 at compare match with GRA, and to 0 at compare match with GRB.

In the examples shown, 16TCNT is cleared by compare match with GRA or GRB. Synchronized operation and free-running counting are also possible.

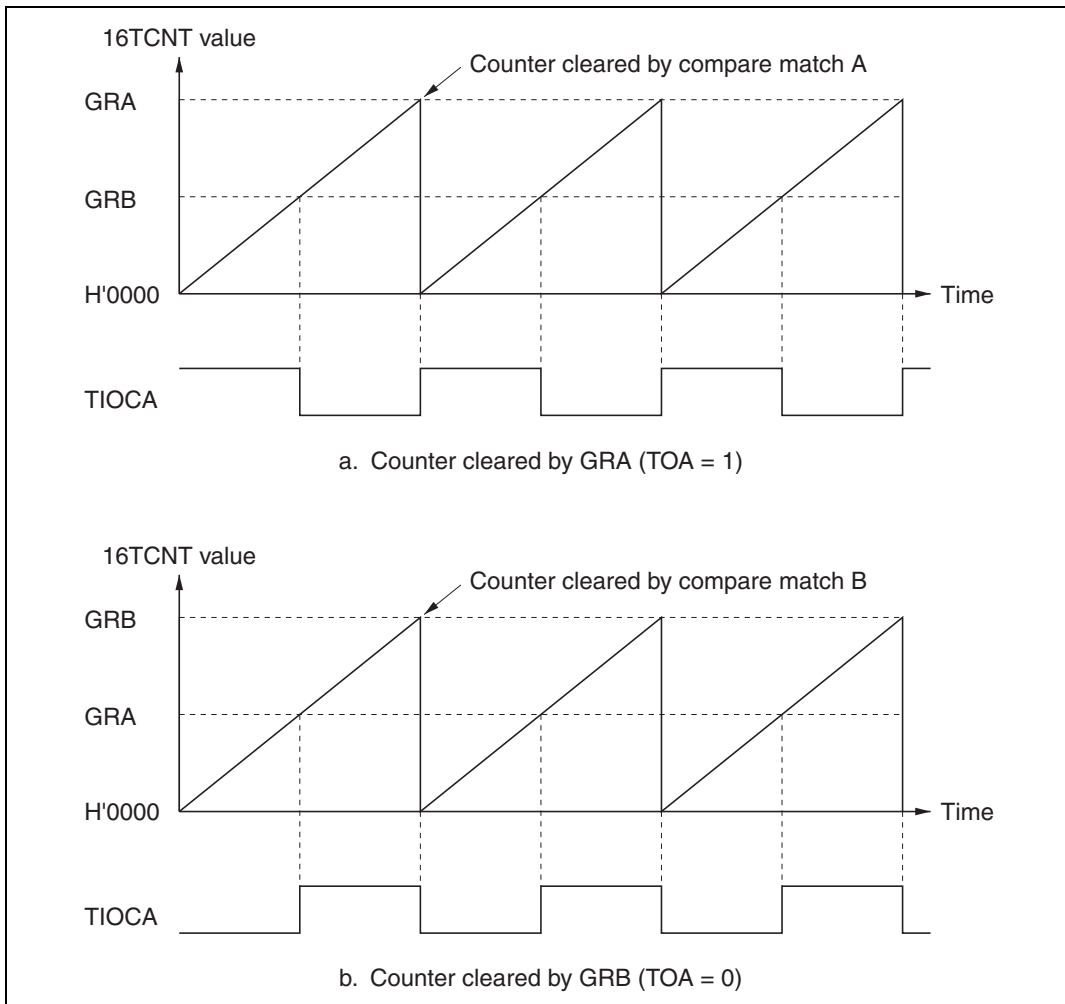


Figure 8.27 PWM Mode (Example 1)

Figure 8.28 shows examples of the output of PWM waveforms with duty cycles of 0% and 100%. If the counter is cleared by compare match with GRB, and GRA is set to a higher value than GRB, the duty cycle is 0%. If the counter is cleared by compare match with GRA, and GRB is set to a higher value than GRA, the duty cycle is 100%.

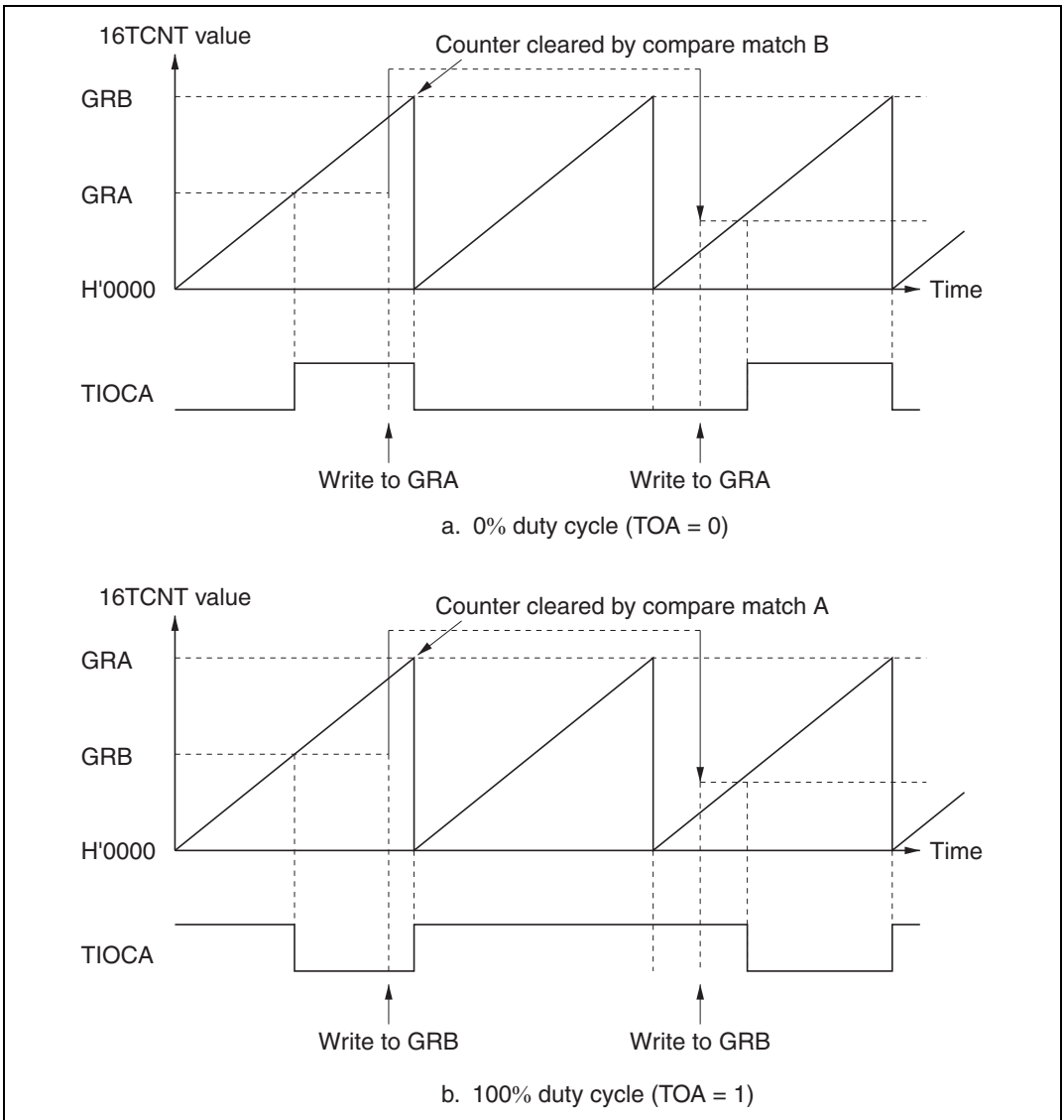


Figure 8.28 PWM Mode (Example 2)

8.4.5 Phase Counting Mode

In phase counting mode the phase difference between two external clock inputs (at the TCLKA and TCLKB pins) is detected, and 16TCNT2 counts up or down accordingly.

In phase counting mode, the TCLKA and TCLKB pins automatically function as external clock input pins and 16TCNT2 becomes an up/down-counter, regardless of the settings of bits TPSC2 to TPSC0, CKEG1, and CKEG0 in 16TCR2. Settings of bits CCLR1, CCLR0 in 16TCR2, and settings in TIOR2, TISRA, TISRB, TISRC, setting of STR2 bit in TSTR, GRA2, and GRB2 are valid. The input capture and output compare functions can be used, and interrupts can be generated.

Phase counting is available only in channel 2.

Sample Setup Procedure for Phase Counting Mode: Figure 8.29 shows a sample procedure for setting up phase counting mode.

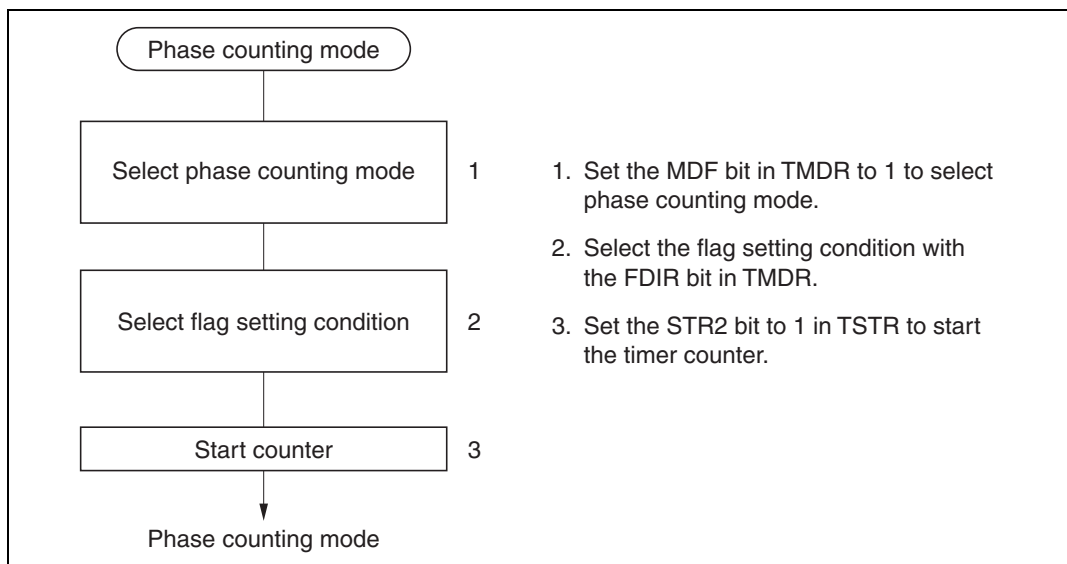


Figure 8.29 Setup Procedure for Phase Counting Mode (Example)

Example of Phase Counting Mode: Figure 8.30 shows an example of operations in phase counting mode. Table 8.5 lists the up-counting and down-counting conditions for 16TCNT2.

In phase counting mode both the rising and falling edges of TCLKA and TCLKB are counted. The phase difference between TCLKA and TCLKB must be at least 1.5 states, the phase overlap must also be at least 1.5 states, and the pulse width must be at least 2.5 states.

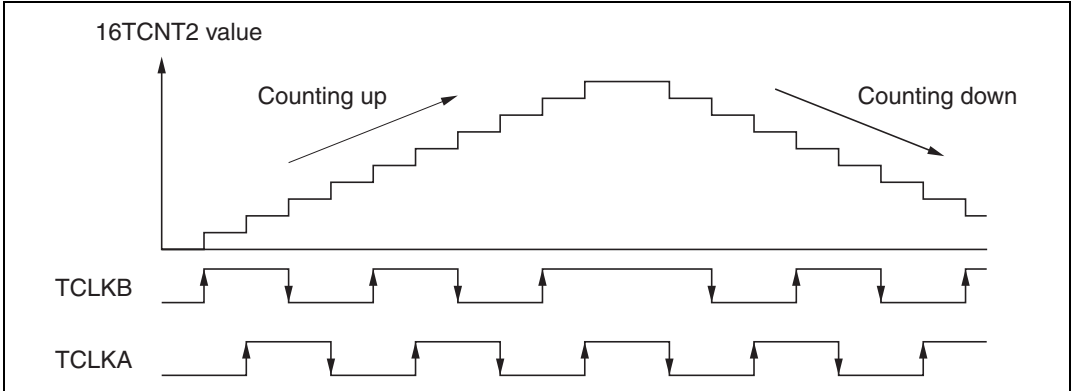


Figure 8.30 Operation in Phase Counting Mode (Example)

Table 8.5 Up/Down Counting Conditions

| Counting Direction | Up-Counting | | | | Down-Counting | | | |
|--------------------|-------------|------|------|-----|---------------|-----|-----|------|
| TCLKB pin | ↑ | High | ↓ | Low | High | ↓ | Low | ↑ |
| TCLKA pin | Low | ↑ | High | ↓ | ↓ | Low | ↑ | High |

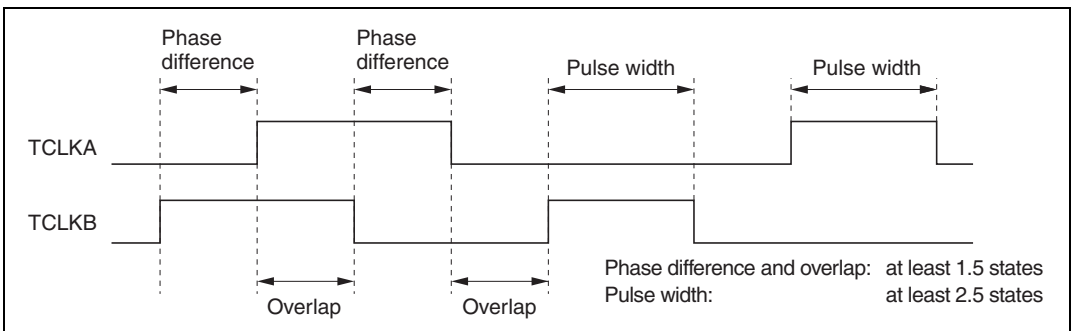


Figure 8.31 Phase Difference, Overlap, and Pulse Width in Phase Counting Mode

8.4.6 16-Bit Timer Output Timing

The initial value of 16-bit timer output when a timer count operation begins can be specified arbitrarily by making a setting in TOLR.

Figure 8.32 shows the timing for setting the initial value with TOLR.

Only write to TOLR when the corresponding bit in TSTR is cleared to 0.

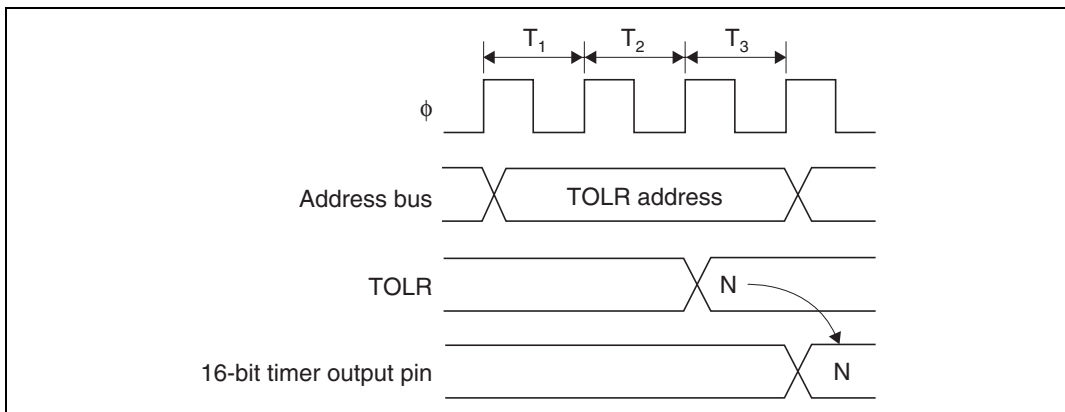


Figure 8.32 Timing for Setting 16-Bit Timer Output Level by Writing to TOLR

8.5 Interrupts

The 16-bit timer has two types of interrupts: input capture/compare match interrupts, and overflow interrupts.

8.5.1 Setting of Status Flags

Timing of Setting of IMFA and IMFB at Compare Match: IMFA and IMFB are set to 1 by a compare match signal generated when 16TCNT matches a general register (GR). The compare match signal is generated in the last state in which the values match (when 16TCNT is updated from the matching count to the next count). Therefore, when 16TCNT matches a general register, the compare match signal is not generated until the next 16TCNT clock input. Figure 8.33 shows the timing of the setting of IMFA and IMFB.

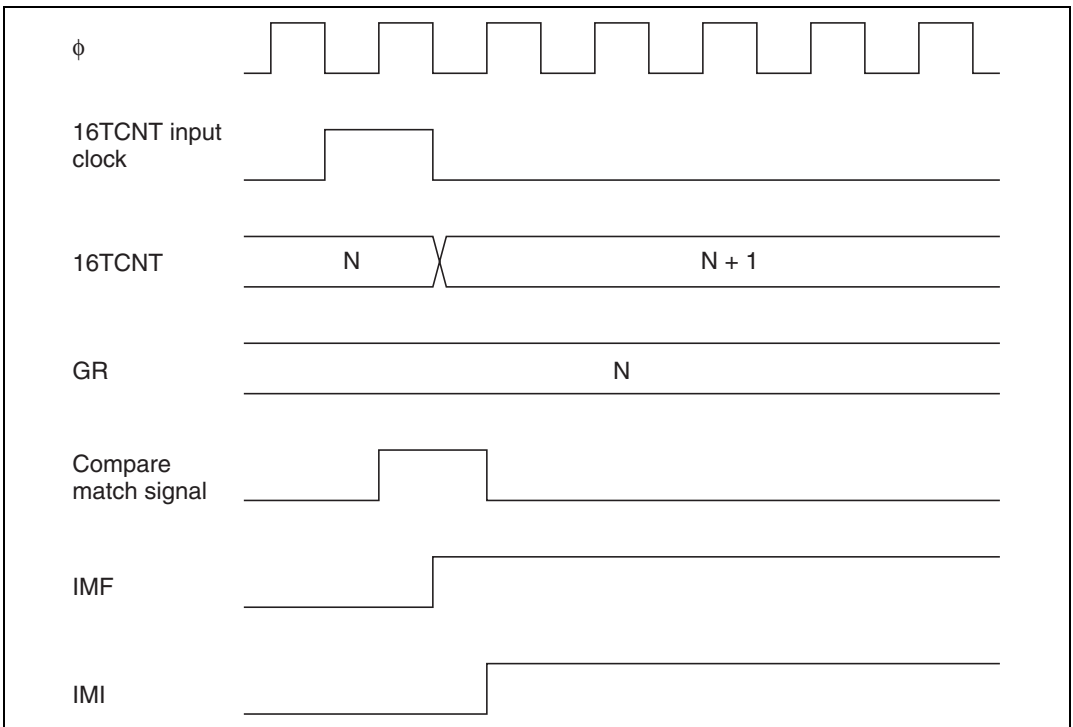


Figure 8.33 Timing of Setting of IMFA and IMFB by Compare Match

Timing of Setting of IMFA and IMFB by Input Capture: IMFA and IMFB are set to 1 by an input capture signal. The 16TCNT contents are simultaneously transferred to the corresponding general register. Figure 8.34 shows the timing.

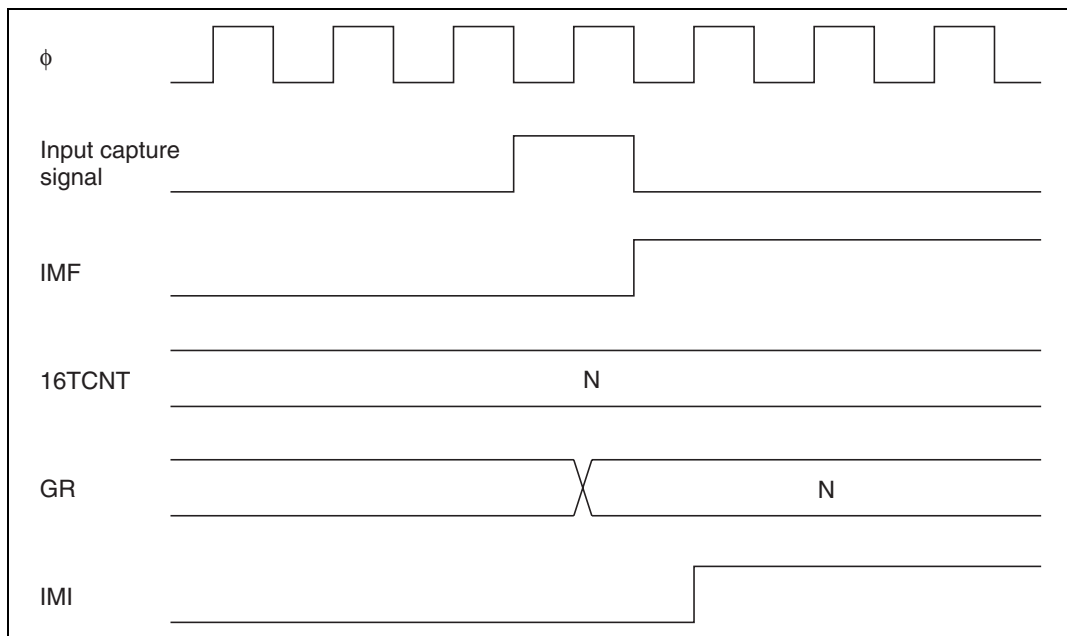


Figure 8.34 Timing of Setting of IMFA and IMFB by Input Capture

Timing of Setting of Overflow Flag (OVF): OVF is set to 1 when 16TCNT overflows from H'FFFF to H'0000 or underflows from H'0000 to H'FFFF. Figure 8.35 shows the timing.

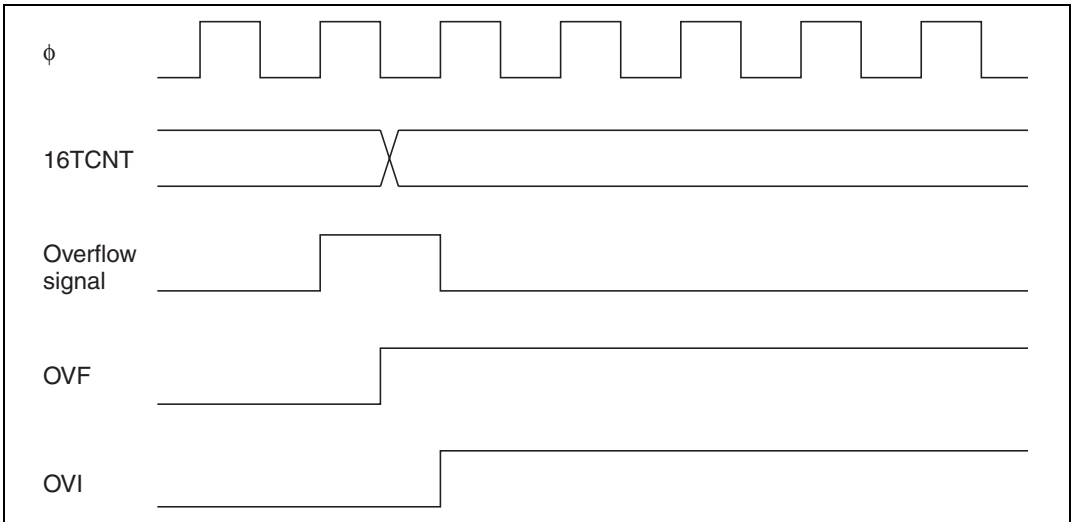


Figure 8.35 Timing of Setting of OVF

8.5.2 Timing of Clearing of Status Flags

If the CPU reads a status flag while it is set to 1, then writes 0 in the status flag, the status flag is cleared. Figure 8.36 shows the timing.

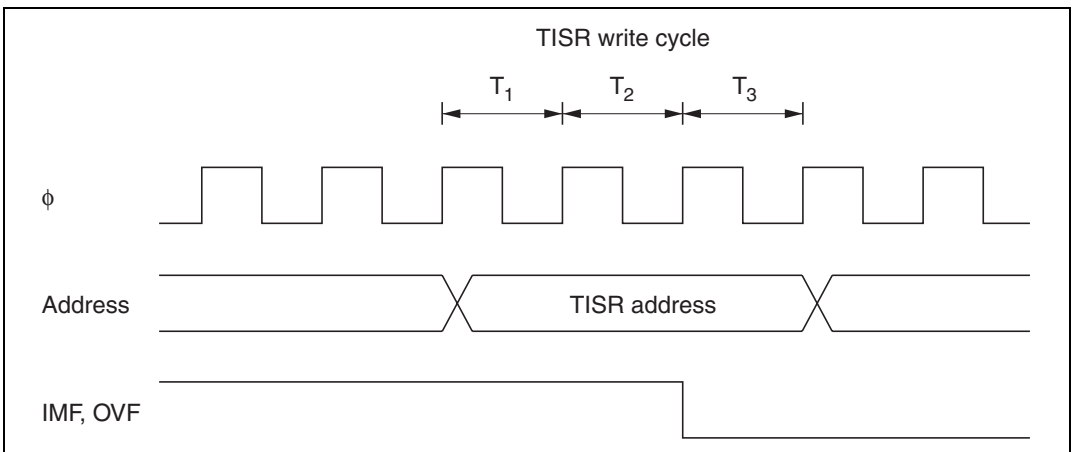


Figure 8.36 Timing of Clearing of Status Flags

8.5.3 Interrupt Sources

Each 16-bit timer channel can generate a compare match/input capture A interrupt, a compare match/input capture B interrupt, and an overflow interrupt. In total there are nine interrupt sources of three kinds, all independently vectored. An interrupt is requested when the interrupt request flag are set to 1.

The priority order of the channels can be modified in interrupt priority registers A (IPRA). For details see section 5, Interrupt Controller.

Table 8.6 lists the interrupt sources.

Table 8.6 16-bit timer Interrupt Sources

| Channel | Interrupt Source | Description | Priority* |
|---------|------------------|--------------------------------|-----------|
| 0 | IMIA0 | Compare match/input capture A0 | |
| | IMIB0 | Compare match/input capture B0 | |
| | OVI0 | Overflow 0 | |
| 1 | IMIA1 | Compare match/input capture A1 | |
| | IMIB1 | Compare match/input capture B1 | |
| | OVI1 | Overflow 1 | |
| 2 | IMIA2 | Compare match/input capture A2 | |
| | IMIB2 | Compare match/input capture B2 | |
| | OVI2 | Overflow 2 | |

Note: * The priority immediately after a reset is indicated. Inter-channel priorities can be changed by settings in IPRA.

8.6 Usage Notes

This section describes contention and other matters requiring special attention during 16-bit timer operations.

Contention between 16TCNT Write and Clear: If a counter clear signal occurs in the T_3 state of a 16TCNT write cycle, clearing of the counter takes priority and the write is not performed. See figure 8.37.

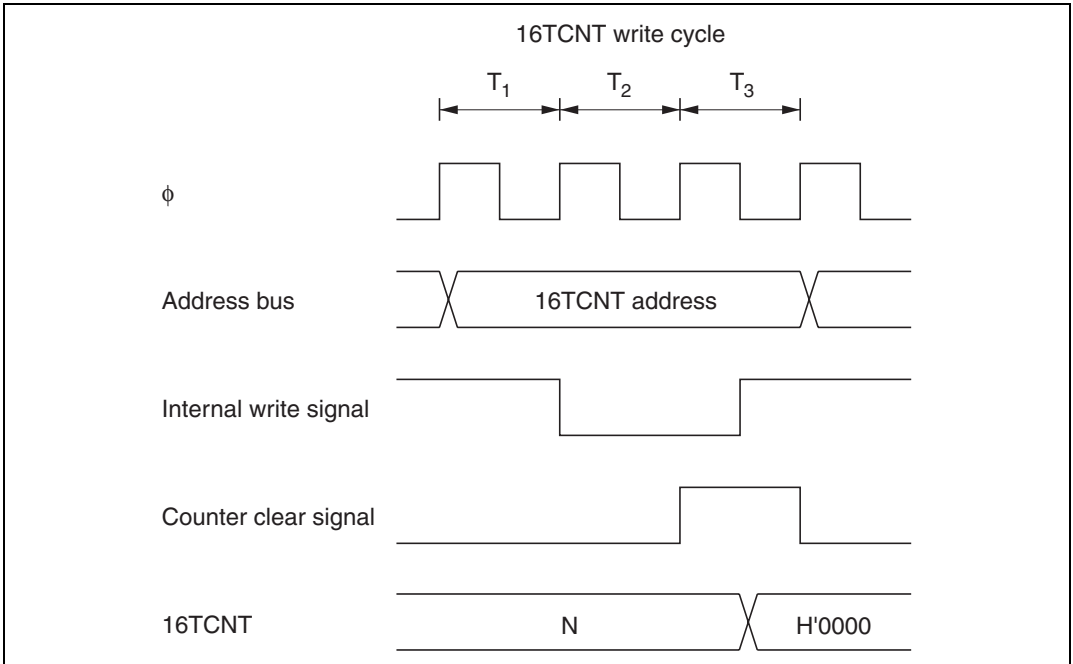


Figure 8.37 Contention between 16TCNT Write and Clear

Contention between 16TCNT Word Write and Increment: If an increment pulse occurs in the T_3 state of a 16TCNT word write cycle, writing takes priority and 16TCNT is not incremented. Figure 8.38 shows the timing in this case.

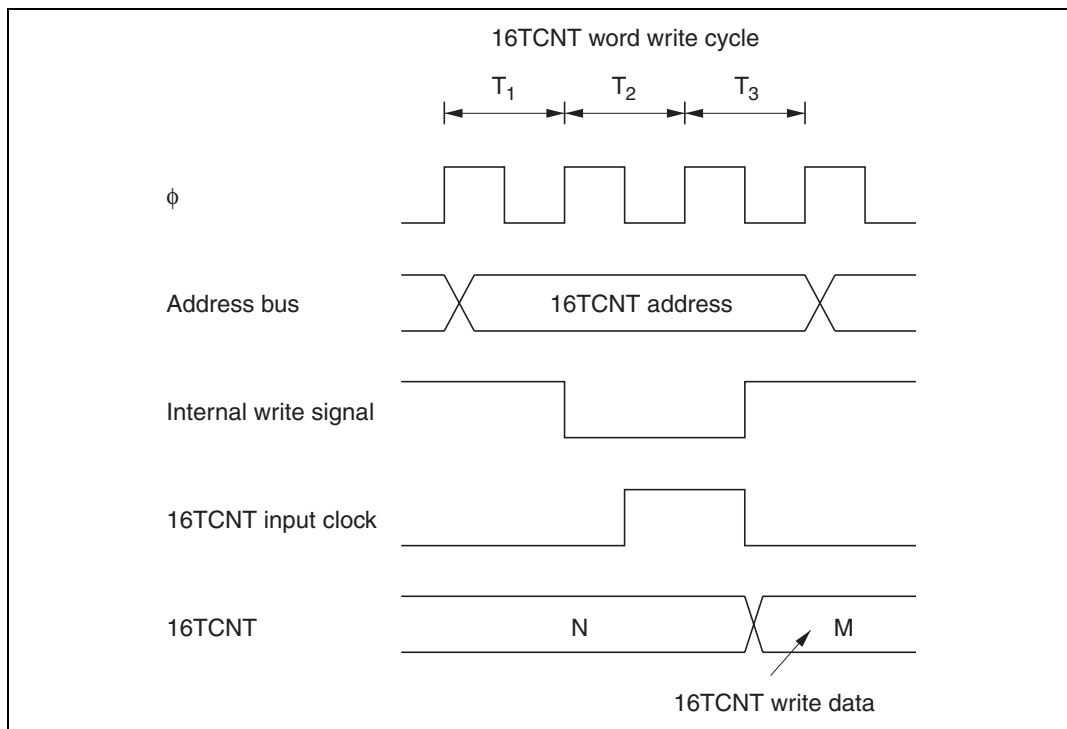


Figure 8.38 Contention between 16TCNT Word Write and Increment

Contention between 16TCNT Byte Write and Increment: If an increment pulse occurs in the T_2 or T_3 state of a 16TCNT byte write cycle, writing takes priority and 16TCNT is not incremented. The byte data for which a write was not performed is not incremented, and retains its pre-write value. See figure 8.39, which shows an increment pulse occurring in the T_2 state of a byte write to 16TCNTH.

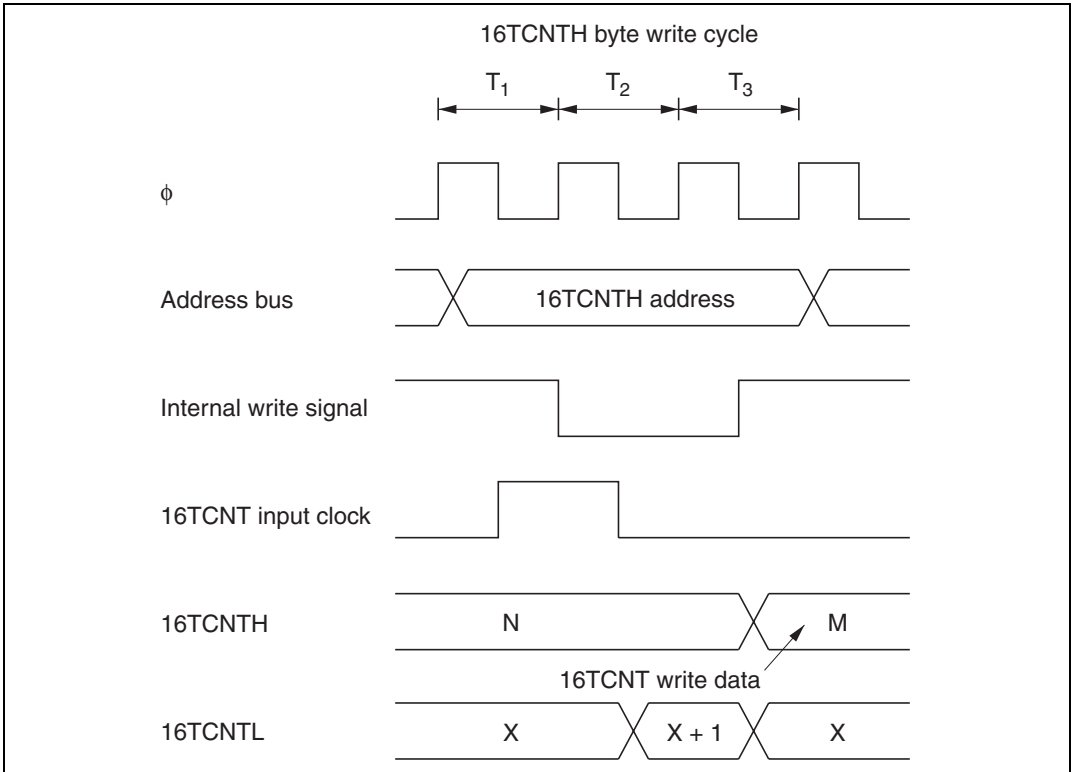


Figure 8.39 Contention between 16TCNT Byte Write and Increment

Contention between General Register Write and Compare Match: If a compare match occurs in the T_3 state of a general register write cycle, writing takes priority and the compare match signal is inhibited. See figure 8.40.

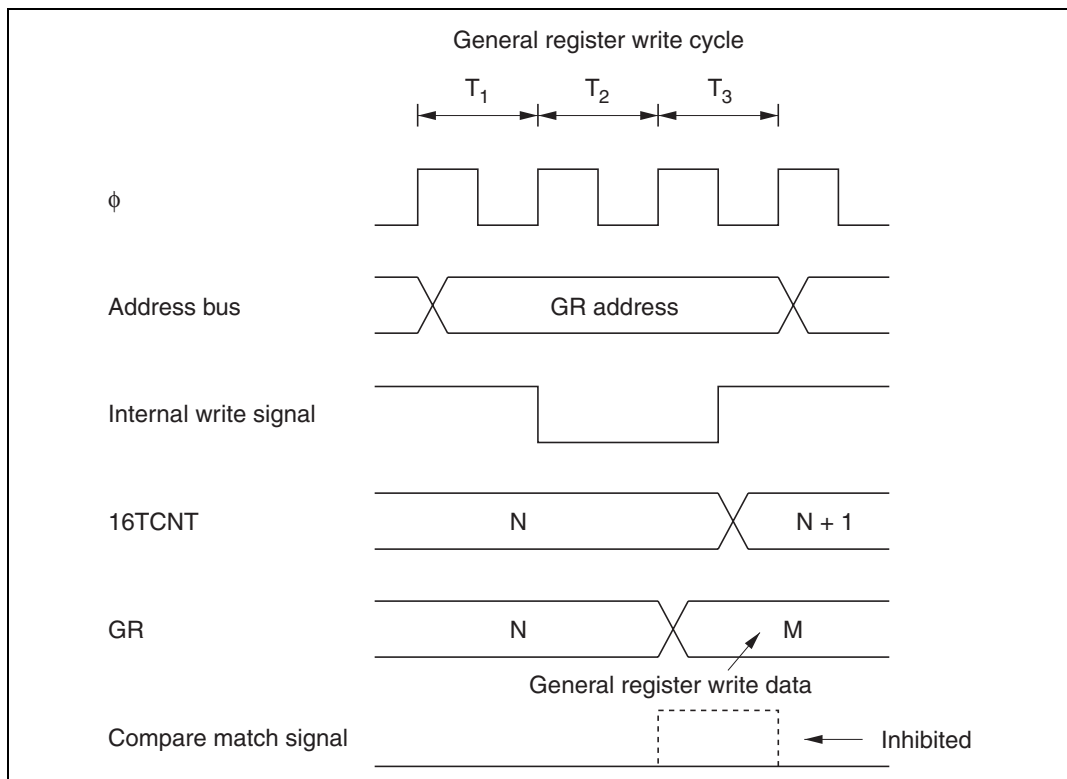


Figure 8.40 Contention between General Register Write and Compare Match

Contention between 16TCNT Write and Overflow or Underflow: If an overflow occurs in the T_3 state of a 16TCNT write cycle, writing takes priority and the counter is not incremented. OVF is set to 1. The same holds for underflow. See figure 8.41.

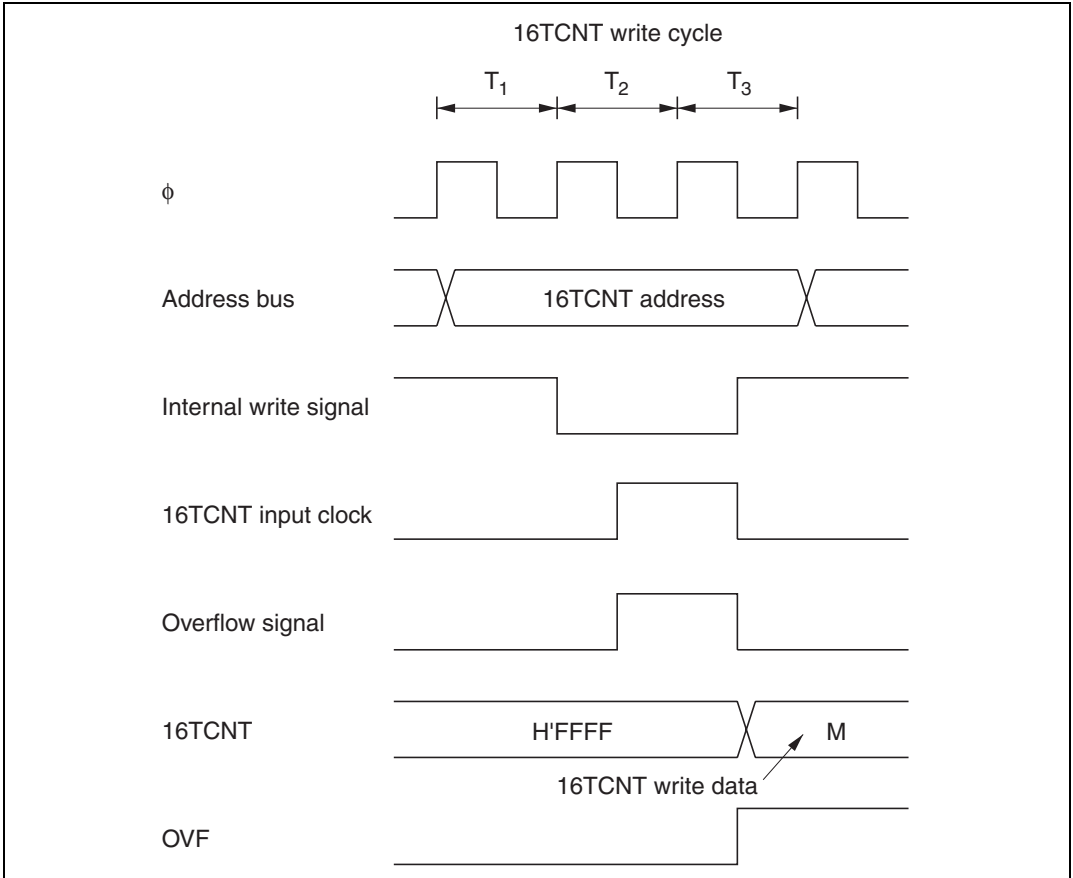


Figure 8.41 Contention between 16TCNT Write and Overflow

Contention between General Register Read and Input Capture: If an input capture signal occurs during the T_3 state of a general register read cycle, the value before input capture is read. See figure 8.42.

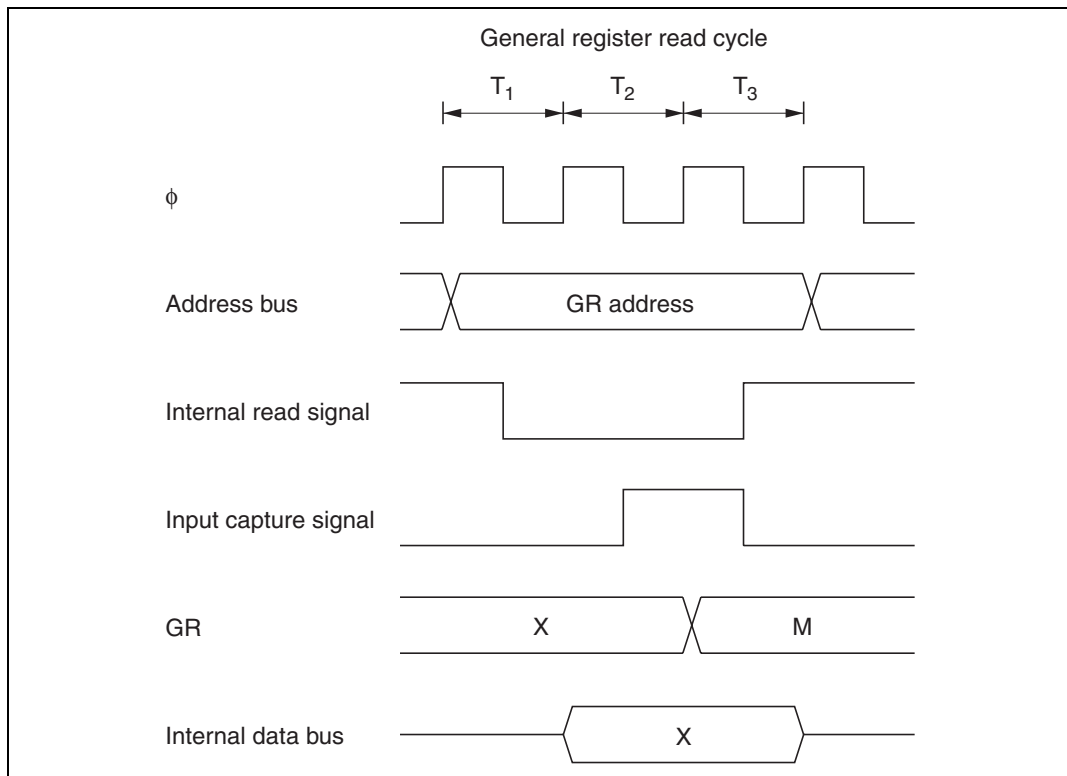


Figure 8.42 Contention between General Register Read and Input Capture

Contention between Counter Clearing by Input Capture and Counter Increment: If an input capture signal and counter increment signal occur simultaneously, the counter is cleared according to the input capture signal. The counter is not incremented by the increment signal. The value before the counter is cleared is transferred to the general register. See figure 8.43.

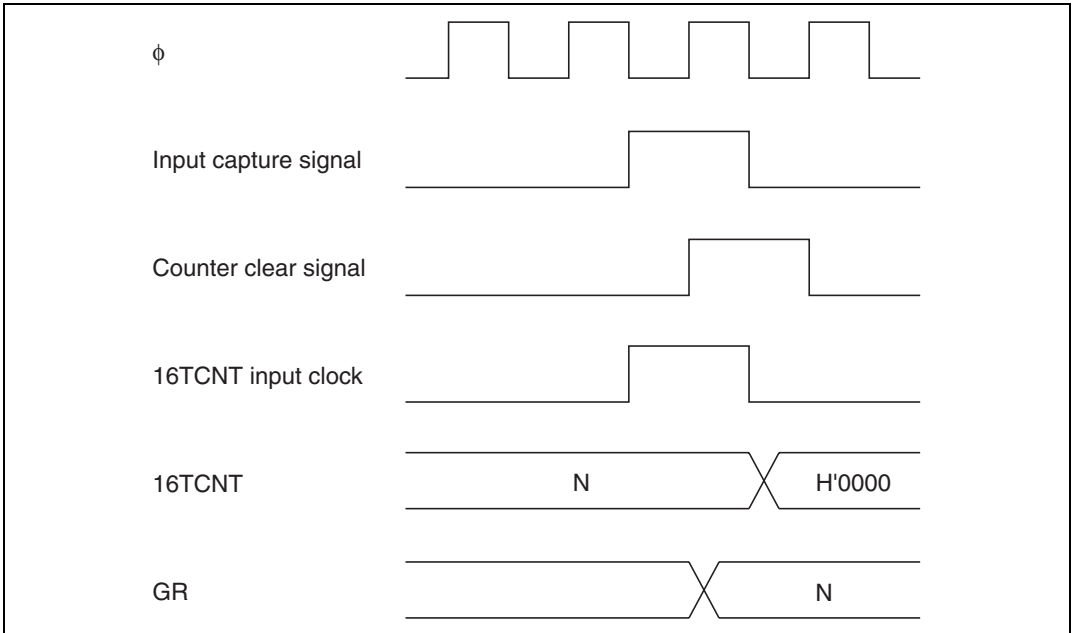


Figure 8.43 Contention between Counter Clearing by Input Capture and Counter Increment

Contention between General Register Write and Input Capture: If an input capture signal occurs in the T_3 state of a general register write cycle, input capture takes priority and the write to the general register is not performed. See figure 8.44.

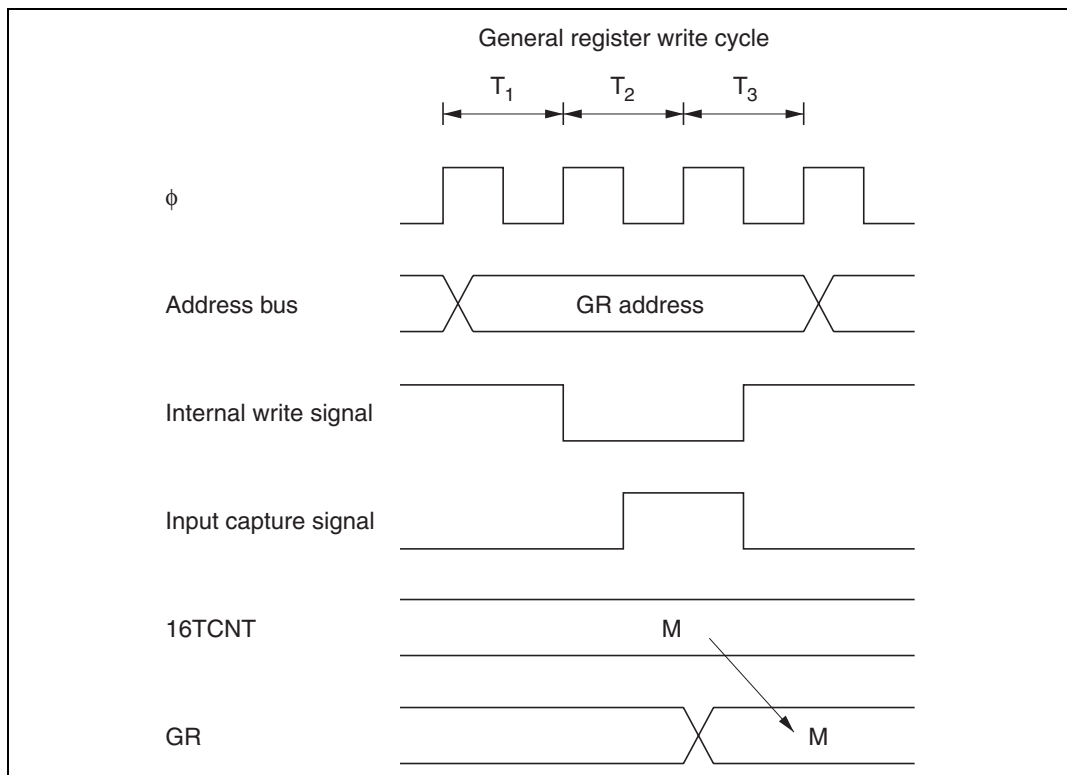


Figure 8.44 Contention between General Register Write and Input Capture

Note on Waveform Period Setting: When a counter is cleared by compare match, the counter is cleared in the last state at which the 16TCNT value matches the general register value, at the time when this value would normally be updated to the next count. The actual counter frequency is therefore given by the following formula:

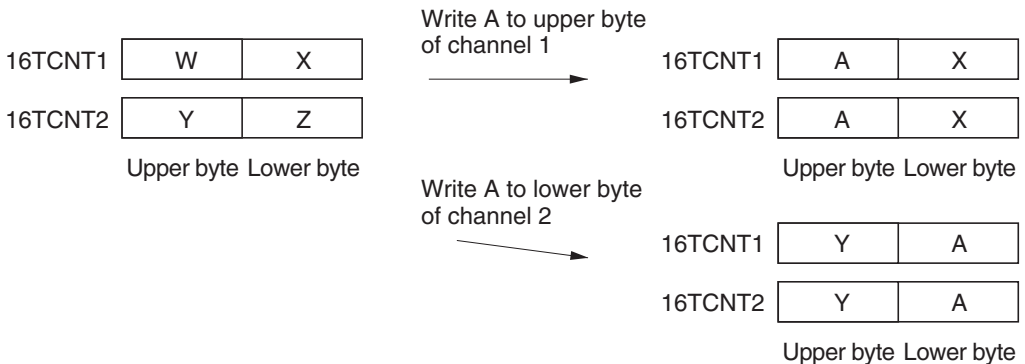
$$f = \frac{\phi}{(N + 1)}$$

(f: counter frequency. ϕ : system clock frequency. N: value set in general register.)

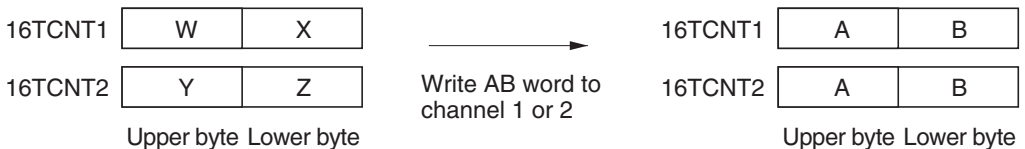
Note on Writes in Synchronized Operation: When channels are synchronized, if a 16TCNT value is modified by byte write access, all 16 bits of all synchronized counters assume the same value as the counter that was addressed.

(Example) When channels 1 and 2 are synchronized

- Byte write to channel 1 or byte write to channel 2



- Word write to channel 1 or word write to channel 2



16-bit timer Operating Modes

Table 8.7 (a) 16-bit timer Operating Modes (Channel 0)

| Operating Mode | | Register Settings | | | | | | | |
|---------------------|--|----------------------|------|------|----------|--|--|------------------------|-----------------|
| | | TSNC | TMDR | | | TIOR0 | | 16TCR0 | |
| | | Synchro- nization | MDF | FDIR | PWM | IOA | IOB | Clear Select | Clock Select |
| Synchronous preset | | SYNC0 = 1 | — | — | ○ | ○ | ○ | ○ | ○ |
| PWM mode | | ○ | — | — | PWM0 = 1 | — | ○* | ○ | ○ |
| Output compare A | | ○ | — | — | PWM0 = 0 | IOA2 = 0 Other bits unrestricted | ○ | ○ | ○ |
| Output compare B | | ○ | — | — | ○ | ○ | IOB2 = 0 Other bits unrestricted | ○ | ○ |
| Input capture A | | ○ | — | — | PWM0 = 0 | IOA2 = 1 Other bits unrestricted | ○ | ○ | ○ |
| Input capture B | | ○ | — | — | PWM0 = 0 | ○ | IOB2 = 1 Other bits unrestricted | ○ | ○ |
| Counter clearing | By compare match/input capture A | ○ | — | — | ○ | ○ | ○ | CCLR1 = 0 CCLR0 = 1 | ○ |
| | By compare match/input capture B | ○ | — | — | ○ | ○ | ○ | CCLR1 = 1 CCLR0 = 0 | ○ |
| | Syn- chronous clear | SYNC0 = 1 | — | — | ○ | ○ | ○ | CCLR1 = 1 CCLR0 = 1 | ○ |

Legend:

- : Setting available (valid).
- : Setting does not affect this mode.

Note: * The input capture function cannot be used in PWM mode. If compare match A and compare match B occur simultaneously, the compare match signal is inhibited.

Table 8.7 (b) 16-bit timer Operating Modes (Channel 1)

| Operating Mode | Register Settings | | | | | | | |
|--------------------|----------------------------------|-----------|------|----------|--|--|------------------------|-----------------|
| | TSNC | TMDR | | | TIOR1 | | 16TCR1 | |
| | Syn- chro- nization | MDF | FDIR | PWM | IOA | IOB | Clear Select | Clock Select |
| Synchronous preset | SYNC1 = 1 | — | — | ○ | ○ | ○ | ○ | ○ |
| PWM mode | ○ | — | — | PWM1 = 1 | — | ○* | ○ | ○ |
| Output compare A | ○ | — | — | PWM1 = 0 | IOA2 = 0 Other bits unrestricted | ○ | ○ | ○ |
| Output compare B | ○ | — | — | ○ | ○ | IOB2 = 0 Other bits unrestricted | ○ | ○ |
| Input capture A | ○ | — | — | PWM1 = 0 | IOA2 = 1 Other bits unrestricted | ○ | ○ | ○ |
| Input capture B | ○ | — | — | PWM1 = 0 | ○ | IOB2 = 1 Other bits unrestricted | ○ | ○ |
| Counter clearing | By compare match/input capture A | ○ | — | — | ○ | ○ | CCLR1 = 0 CCLR0 = 1 | ○ |
| | By compare match/input capture B | ○ | — | — | ○ | ○ | CCLR1 = 1 CCLR0 = 0 | ○ |
| | Syn-chronous clear | SYNC1 = 1 | — | — | ○ | ○ | CCLR1 = 1 CCLR0 = 1 | ○ |

Legend:

- : Setting available (valid).
 —: Setting does not affect this mode.

Note: * The input capture function cannot be used in PWM mode. If compare match A and compare match B occur simultaneously, the compare match signal is inhibited.

Table 8.7 (c) 16-bit timer Operating Modes (Channel 2)

| Operating Mode | Register Settings | | | | | | | |
|---------------------|----------------------------------|-----------|------|----------|--|--|------------------------|-----------------|
| | TSNC | TMDR | | | TIOR2 | | 16TCR2 | |
| | Synchro- nization | MDF | FDIR | PWM | IOA | IOB | Clear Select | Clock Select |
| Synchronous preset | SYNC2 = 1 | ○ | — | ○ | ○ | ○ | ○ | ○ |
| PWM mode | ○ | ○ | — | PWM2 = 1 | — | ○* | ○ | ○ |
| Output compare A | ○ | ○ | — | PWM2 = 0 | IOA2 = 0 Other bits unrestricted | ○ | ○ | ○ |
| Output compare B | ○ | ○ | — | ○ | ○ | IOB2 = 0 Other bits unrestricted | ○ | ○ |
| Input capture A | ○ | ○ | — | PWM2 = 0 | IOA2 = 1 Other bits unrestricted | ○ | ○ | ○ |
| Input capture B | ○ | ○ | — | PWM2 = 0 | ○ | IOB2 = 1 Other bits unrestricted | ○ | ○ |
| Counter clearing | By compare match/input capture A | ○ | ○ | — | ○ | ○ | CCLR1 = 0 CCLR0 = 1 | ○ |
| | By compare match/input capture B | ○ | ○ | — | ○ | ○ | CCLR1 = 1 CCLR0 = 0 | ○ |
| | Syn-chronous clear | SYNC2 = 1 | ○ | — | ○ | ○ | CCLR1 = 1 CCLR0 = 1 | ○ |
| Phase counting mode | ○ | MDF = 1 | ○ | ○ | ○ | ○ | ○ | — |

Legend:

○: Setting available (valid).

—: Setting does not affect this mode.

Note: * The input capture function cannot be used in PWM mode. If compare match A and compare match B occur simultaneously, the compare match signal is inhibited.

Section 9 8-Bit Timers

9.1 Overview

The H8/3008 has a built-in 8-bit timer module with four channels (TMR0, TMR1, TMR2, and TMR3), based on 8-bit counters. Each channel has an 8-bit timer counter (8TCNT) and two 8-bit time constant registers (TCORA and TCORB) that are constantly compared with the 8TCNT value to detect compare match events. The timers can be used as multifunctional timers in a variety of applications, including the generation of a rectangular-wave output with an arbitrary duty cycle.

9.1.1 Features

The features of the 8-bit timer module are listed below.

- Selection of four clock sources
The counters can be driven by one of three internal clock signals ($\phi/8$, $\phi/64$, or $\phi/8192$) or an external clock input (enabling use as an external event counter).
- Selection of three ways to clear the counters
The counters can be cleared on compare match A or B, or input capture B.
- Timer output controlled by two compare match signals
The timer output signal in each channel is controlled by two independent compare match signals, enabling the timer to generate output waveforms with an arbitrary duty cycle or PWM output.
- A/D converter can be activated by a compare match
- Two channels can be cascaded
 - Channels 0 and 1 can be operated as the upper and lower halves of a 16-bit timer (16-bit count mode).
 - Channels 2 and 3 can be operated as the upper and lower halves of a 16-bit timer (16-bit count mode).
 - Channel 1 can count channel 0 compare match events (compare match count mode).
 - Channel 3 can count channel 2 compare match events (compare match count mode).
- Input capture function can be set
8-bit or 16-bit input capture operation is available.
- Twelve interrupt sources
There are twelve interrupt sources: four compare match sources, four compare match/input capture sources, four overflow sources.

Two of the compare match sources and two of the combined compare match/input capture sources each have an independent interrupt vector. The remaining compare match interrupts, combined compare match/input capture interrupts, and overflow interrupts have one interrupt vector for two sources.

9.1.2 Block Diagram

The 8-bit timers are divided into two groups of two channels each: group 0 comprising channels 0 and 1, and group 1 comprising channels 2 and 3. Figure 9.1 shows a block diagram of 8-bit timer group 0.

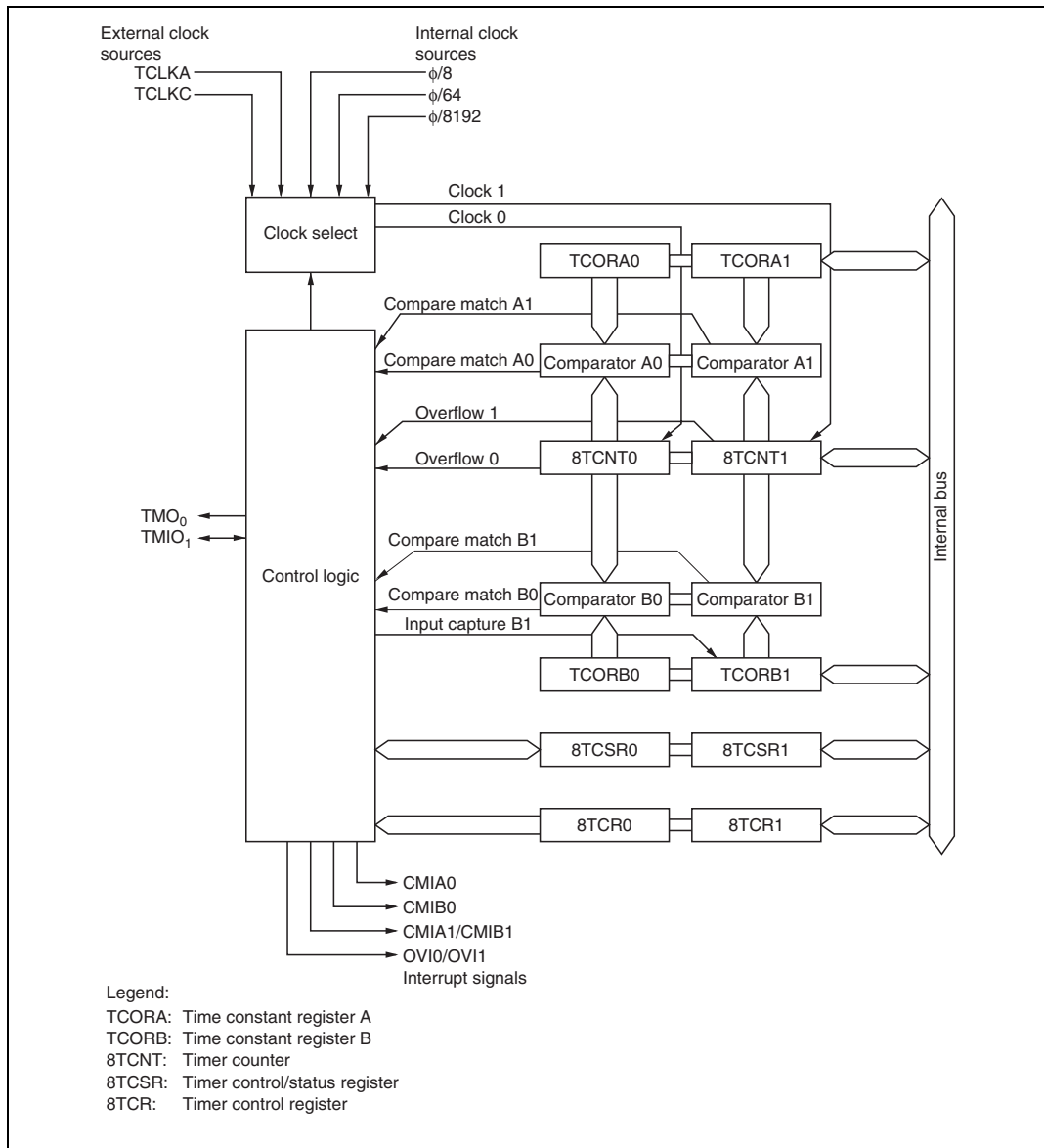


Figure 9.1 Block Diagram of 8-Bit Timer Unit (Two Channels: Group 0)

9.1.3 Pin Configuration

Table 9.1 summarizes the input/output pins of the 8-bit timer module.

Table 9.1 8-Bit Timer Pins

| Group | Channel | Name | Abbreviation | I/O | Function |
|-------|---------|--------------------|-------------------|--------|--|
| 0 | 0 | Timer output | TMO ₀ | Output | Compare match output |
| | | Timer clock input | TCLKC | Input | Counter external clock input |
| | 1 | Timer input/output | TMIO ₁ | I/O | Compare match output/input capture input |
| | | Timer clock input | TCLKA | Input | Counter external clock input |
| 1 | 2 | Timer output | TMO ₂ | Output | Compare match output |
| | | Timer clock input | TCLKD | Input | Counter external clock input |
| | 3 | Timer input/output | TMIO ₃ | I/O | Compare match output/input capture input |
| | | Timer clock input | TCLKB | Input | Counter external clock input |

9.1.4 Register Configuration

Table 9.2 summarizes the registers of the 8-bit timer module.

Table 9.2 8-Bit Timer Registers

| Channel | Address* ¹ | Name | Abbreviation | R/W | Initial value |
|---------|-----------------------|---------------------------------|--------------|---------------------|---------------|
| 0 | H'FFF80 | Timer control register 0 | 8TCR0 | R/W | H'00 |
| | H'FFF82 | Timer control/status register 0 | 8TCSR0 | R/(W)* ² | H'00 |
| | H'FFF84 | Time constant register A0 | TCORA0 | R/W | H'FF |
| | H'FFF86 | Time constant register B0 | TCORB0 | R/W | H'FF |
| | H'FFF88 | Timer counter 0 | 8TCNT0 | R/W | H'00 |
| 1 | H'FFF81 | Timer control register 1 | 8TCR1 | R/W | H'00 |
| | H'FFF83 | Timer control/status register 1 | 8TCSR1 | R/(W)* ² | H'00 |
| | H'FFF85 | Time constant register A1 | TCORA1 | R/W | H'FF |
| | H'FFF87 | Time constant register B1 | TCORB1 | R/W | H'FF |
| | H'FFF89 | Timer counter 1 | 8TCNT1 | R/W | H'00 |
| 2 | H'FFF90 | Timer control register 2 | 8TCR2 | R/W | H'00 |
| | H'FFF92 | Timer control/status register 2 | 8TCSR2 | R/(W)* ² | H'10 |
| | H'FFF94 | Time constant register A2 | TCORA2 | R/W | H'FF |
| | H'FFF96 | Time constant register B2 | TCORB2 | R/W | H'FF |
| | H'FFF98 | Timer counter 2 | 8TCNT2 | R/W | H'00 |
| 3 | H'FFF91 | Timer control register 3 | 8TCR3 | R/W | H'00 |
| | H'FFF93 | Timer control/status register 3 | 8TCSR3 | R/(W)* ² | H'00 |
| | H'FFF95 | Time constant register A3 | TCORA3 | R/W | H'FF |
| | H'FFF97 | Time constant register B3 | TCORB3 | R/W | H'FF |
| | H'FFF99 | Timer counter 3 | 8TCNT3 | R/W | H'00 |

Notes: 1. Indicates the lower 20 bits of the address in advanced mode.

2. Only 0 can be written to bits 7 to 5, to clear these flags.

Each pair of registers for channel 0 and channel 1 comprises a 16-bit register with the channel 0 register as the upper 8 bits and the channel 1 register as the lower 8 bits, so they can be accessed together by word access.

Similarly, each pair of registers for channel 2 and channel 3 comprises a 16-bit register with the channel 2 register as the upper 8 bits and the channel 3 register as the lower 8 bits, so they can be accessed together by word access.

9.2 Register Descriptions

9.2.1 Timer Counters (8TCNT)

| 8TCNT0 | | | | | | | | 8TCNT1 | | | | | | | | |
|---------------|-----|-----|-----|-----|-----|-----|-----|--------|-----|-----|-----|-----|-----|-----|-----|-----|
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Initial value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

| 8TCNT2 | | | | | | | | 8TCNT3 | | | | | | | | |
|---------------|-----|-----|-----|-----|-----|-----|-----|--------|-----|-----|-----|-----|-----|-----|-----|-----|
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Initial value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

The timer counters (8TCNT) are 8-bit readable/writable up-counters that increment on pulses generated from an internal or external clock source. The clock source is selected by clock select bits 2 to 0 (CKS2 to CKS0) in the timer control register (8TCR). The CPU can always read or write to the timer counters.

The 8TCNT0 and 8TCNT1 pair, and the 8TCNT2 and 8TCNT3 pair, can each be accessed as a 16-bit register by word access.

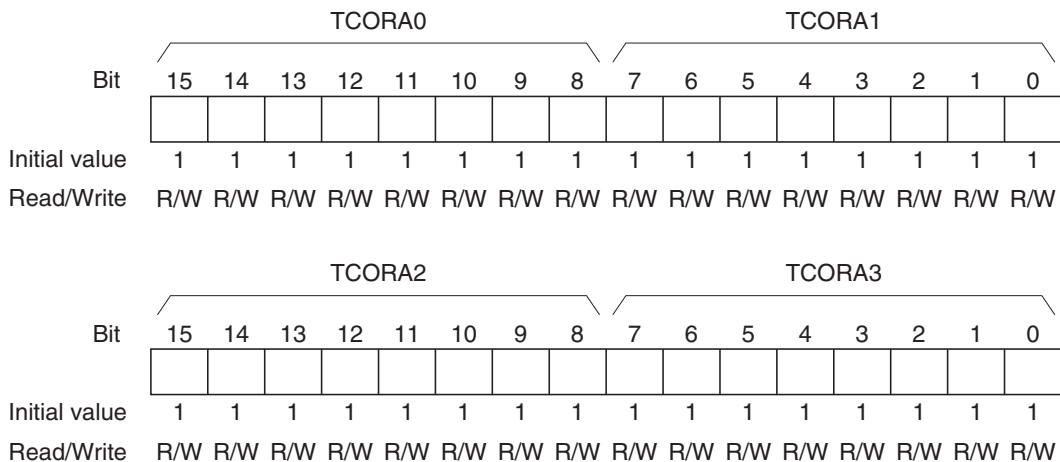
8TCNT can be cleared by an input capture signal or compare match signal. Counter clear bits 1 and 0 (CCLR1 and CCLR0) in 8TCR select the method of clearing.

When 8TCNT overflows from H'FF to H'00, the overflow flag (OVF) in the timer control/status register (8TCSR) is set to 1.

Each 8TCNT is initialized to H'00 by a reset and in standby mode.

9.2.2 Time Constant Registers A (TCORA)

TCORA0 to TCORA3 are 8-bit readable/writable registers.



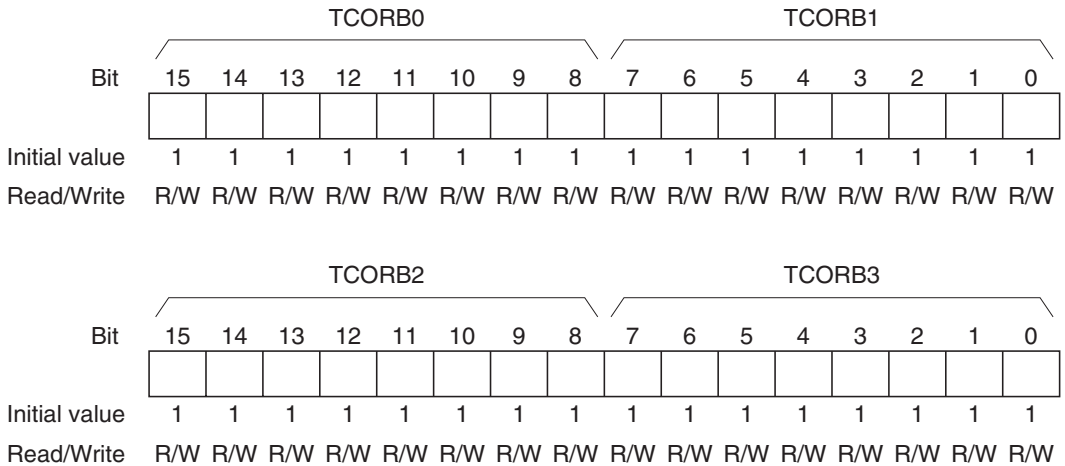
The TCORA0 and TCORA1 pair, and the TCORA2 and TCORA3 pair, can each be accessed as a 16-bit register by word access.

The TCORA value is constantly compared with the 8TCNT value. When a match is detected, the corresponding compare match flag A (CMFA) is set to 1 in 8TCSR.

The timer output can be freely controlled by these compare match signals and the settings of output select bits 1 and 0 (OS1, OS0) in 8TCSR.

Each TCORA register is initialized to H'FF by a reset and in standby mode.

9.2.3 Time Constant Registers B (TCORB)



TCORB0 to TCORB3 are 8-bit readable/writable registers. The TCORB0 and TCORB1 pair, and the TCORB2 and TCORB3 pair, can each be accessed as a 16-bit register by word access.

The TCORB value is constantly compared with the 8TCNT value. When a match is detected, the corresponding compare match flag B (CMFB) is set to 1 in 8TCSR*.

The timer output can be freely controlled by these compare match signals and the settings of output/input capture edge select bits 3 and 2 (OIS3, OIS2) in 8TCSR.

When TCORB is used for input capture, it stores the 8TCNT value on detection of an external input capture signal. At this time, the CMFB flag is set to 1 in the corresponding 8TCSR register. The detected edge of the input capture signal is set in 8TCSR.

Each TCORB register is initialized to H'FF by a reset and in standby mode.

Note: * When channel 1 and channel 3 are designated for TCORB input capture, the CMFB flag is not set by a channel 0 or channel 2 compare match B.

9.2.4 Timer Control Register (8TCR)

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|-------|-------|------|-------|-------|------|------|------|
| | CMIEB | CMIEA | OVIE | CCLR1 | CCLR0 | CKS2 | CKS1 | CKS0 |
| Initial value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

8TCR is an 8-bit readable/writable register that selects the 8TCNT input clock, gives the 8TCNT clearing specification, and enables interrupt requests.

8TCR is initialized to H'00 by a reset and in standby mode.

For the timing, see section 9.4, Operation.

Bit 7—Compare Match Interrupt Enable B (CMIEB): Enables or disables the CMIB interrupt request when the CMFB flag is set to 1 in 8TCSR.

Bit 7

| CMIEB | Description |
|-------|--|
| 0 | CMIB interrupt requested by CMFB is disabled (Initial value) |
| 1 | CMIB interrupt requested by CMFB is enabled |

Bit 6—Compare Match Interrupt Enable A (CMIEA): Enables or disables the CMIA interrupt request when the CMFA flag is set to 1 in 8TCSR.

Bit 6

| CMIEA | Description |
|-------|--|
| 0 | CMIA interrupt requested by CMFA is disabled (Initial value) |
| 1 | CMIA interrupt requested by CMFA is enabled |

Bit 5—Timer Overflow Interrupt Enable (OVIE): Enables or disables the OVI interrupt request when the OVF flag is set to 1 in 8TCSR.

Bit 5

| OVIE | Description |
|------|--|
| 0 | OVI interrupt requested by OVF is disabled (Initial value) |
| 1 | OVI interrupt requested by OVF is enabled |

Bits 4 and 3—Counter Clear 1 and 0 (CCLR1, CCLR0): These bits specify the 8TCNT clearing source. Compare match A or B, or input capture B, can be selected as the clearing source.

| Bit 4 CCLR1 | Bit 3 CCLR0 | Description | |
|----------------|----------------|--|-----------------|
| 0 | 0 | Clearing is disabled | (Initial value) |
| | 1 | Cleared by compare match A | |
| 1 | 0 | Cleared by compare match B/input capture B | |
| | 1 | Cleared by input capture B | |

Note: When input capture B is set as the 8TCNT1 and 8TCNT3 counter clear source, 8TCNT0 and 8TCNT2 are not cleared by compare match B.

Bits 2 to 0—Clock Select 2 to 0 (CSK2 to CSK0): These bits select whether the clock input to 8TCNT is an internal or external clock.

Three internal clocks can be selected, all divided from the system clock (ϕ): $\phi/8$, $\phi/64$, and $\phi/8192$. The rising edge of the selected internal clock triggers the count.

When use of an external clock is selected, three types of count can be selected: at the rising edge, the falling edge, and both rising and falling edges.

When $CKS2, CKS1, CKS0 = 1, 0, 0$, channels 0 and 1 and channels 2 and 3 are cascaded.

The incrementing clock source is different when 8TCR0 and 8TCR2 are set, and when 8TCR1 and 8TCR3 are set.

| Bit 2 CSK2 | Bit 1 CSK1 | Bit 0 CSK0 | Description | |
|---------------|---------------|---------------|--|--|
| 0 | 0 | 0 | Clock input disabled (Initial value) | |
| | | 1 | Internal clock, counted on falling edge of $\phi/8$ | |
| | 1 | 0 | Internal clock, counted on falling edge of $\phi/64$ | |
| | | 1 | Internal clock, counted on falling edge of $\phi/8192$ | |
| 1 | 0 | 0 | Channel 0 (16-bit count mode): Count on 8TCNT1 overflow signal* ¹ Channel 1 (compare match count mode): Count on 8TCNT0 compare match A* ¹ Channel 2 (16-bit count mode): Count on 8TCNT3 overflow signal* ² Channel 3 (compare match count mode): Count on 8TCNT2 compare match A* ² | |
| | | 1 | External clock, counted on rising edge | |
| | | 1 | 0 | External clock, counted on falling edge |
| | | | 1 | External clock, counted on both rising and falling edges |

- Notes:
1. If the clock input of channel 0 is the 8TCNT1 overflow signal and that of channel 1 is the 8TCNT0 compare match signal, no incrementing clock is generated. Do not use this setting.
 2. If the clock input of channel 2 is the 8TCNT3 overflow signal and that of channel 3 is the 8TCNT2 compare match signal, no incrementing clock is generated. Do not use this setting.

9.2.5 Timer Control/Status Registers (8TCSR)

8TCSR0

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|--------|--------|--------|------|------|------|-----|-----|
| | CMFB | CMFA | OVF | ADTE | OIS3 | OIS2 | OS1 | OS0 |
| Initial value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Read/Write | R/(W)* | R/(W)* | R/(W)* | R/W | R/W | R/W | R/W | R/W |

8TCSR2

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|--------|--------|--------|---|------|------|-----|-----|
| | CMFB | CMFA | OVF | — | OIS3 | OIS2 | OS1 | OS0 |
| Initial value | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| Read/Write | R/(W)* | R/(W)* | R/(W)* | — | R/W | R/W | R/W | R/W |

8TCSR1, 8TCSR3

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|--------|--------|--------|-----|------|------|-----|-----|
| | CMFB | CMFA | OVF | ICE | OIS3 | OIS2 | OS1 | OS0 |
| Initial value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Read/Write | R/(W)* | R/(W)* | R/(W)* | R/W | R/W | R/W | R/W | R/W |

Note: * Only 0 can be written to bits 7 to 5, to clear these flags.

The timer control/status registers 8TCSR are 8-bit registers that indicate compare match/input capture and overflow statuses, and control compare match output/input capture edge selection.

8TCSR2 is initialized to H'10, and 8TCSR0, 8TCSR1, and 8TCSR3 to H'00, by a reset and in standby mode.

Bit 7—Compare Match/Input Capture Flag B (CMFB): Status flag that indicates the occurrence of a TCORB compare match or input capture.

| Bit 7 CMFB | Description | |
|---------------|---|-----------------|
| 0 | [Clearing condition] Read CMFB when CMFB = 1, then write 0 in CMFB | (Initial value) |
| 1 | [Setting conditions] <ul style="list-style-type: none"> • 8TCNT = TCORB* • The 8TCNT value is transferred to TCORB by an input capture signal when TCORB functions as an input capture register | |

Note: * When bit ICE is set to 1 in 8TCSR1 and 8TCSR3, the CMFB flag is not set when 8TCNT0 = TCORB0 or 8TCNT2 = TCORB2.

Bit 6—Compare Match Flag A (CMFA): Status flag that indicates the occurrence of a TCORA compare match.

| Bit 6 CMFA | Description | |
|---------------|---|-----------------|
| 0 | [Clearing condition] Read CMFA when CMFA = 1, then write 0 in CMFA | (Initial value) |
| 1 | [Setting condition] 8TCNT = TCORA | |

Bit 5—Timer Overflow Flag (OVF): Status flag that indicates that the 8TCNT has overflowed from H'FF to H'00.

| Bit 5 OVF | Description | |
|--------------|--|-----------------|
| 0 | [Clearing condition] Read OVF when OVF = 1, then write 0 in OVF | (Initial value) |
| 1 | [Setting condition] 8TCNT overflows from H'FF to H'00 | |

Bit 4—A/D Trigger Enable (ADTE) (In 8TCSR0): In combination with TRGE in the A/D control register (ADCR), enables or disables A/D converter start requests by compare match A or an external trigger.

| TRGE* | Bit 4 ADTE | Description |
|-------|---------------|--|
| 0 | 0 | A/D converter start requests by compare match A or external trigger pin (ADTRG) input are disabled (Initial value) |
| | 1 | A/D converter start requests by compare match A or external trigger pin (ADTRG) input are disabled |
| 1 | 0 | A/D converter start requests by external trigger pin (ADTRG) input are enabled, and A/D converter start requests by compare match A are disabled |
| | 1 | A/D converter start requests by compare match A are enabled, and A/D converter start requests by external trigger pin (ADTRG) input are disabled |

Note: * TRGE is bit 7 of the A/D control register (ADCR).

Bit 4—Reserved (In 8TCSR1): This bit is a reserved bit, but can be read and written.

Bit 4—Input Capture Enable (ICE) (In 8TCSR1 and 8TCSR3): Selects the function of TCORB1 and TCORB3.

| Bit 4 ICE | Description |
|--------------|---|
| 0 | TCORB1 and TCORB3 are compare match registers (Initial value) |
| 1 | TCORB1 and TCORB3 are input capture registers |

When bit ICE is set to 1 in 8TCSR1 or 8TCSR3, the operation of the TCORA and TCORB registers in channels 0 to 3 is as shown in the tables below.

Table 9.3 Operation of Channels 0 and 1 when Bit ICE is Set to 1 in 8TCSR1 Register

| Register | Register Function | Status Flag Change | Timer Output Capture Input | Interrupt Request |
|-----------------|--------------------------|---|--|--|
| TCORA0 | Compare match operation | CMFA changed from 0 to 1 in 8TCSR0 by compare match | TMO ₀ output controllable | CMIA0 interrupt request generated by compare match |
| TCORB0 | Compare match operation | CMFB not changed from 0 to 1 in 8TCSR0 by compare match | No output from TMO ₀ | CMIB0 interrupt request not generated by compare match |
| TCORA1 | Compare match operation | CMFA changed from 0 to 1 in 8TCSR1 by compare match | TMIO ₁ is dedicated input capture pin | CMIA1 interrupt request generated by compare match |
| TCORB1 | Input capture operation | CMFB changed from 0 to 1 in 8TCSR1 by input capture | TMIO ₁ is dedicated input capture pin | CMIB1 interrupt request generated by input capture |

Table 9.4 Operation of Channels 2 and 3 when Bit ICE is Set to 1 in 8TCSR3 Register

| Register | Register Function | Status Flag Change | Timer Output Capture Input | Interrupt Request |
|-----------------|--------------------------|---|--|--|
| TCORA2 | Compare match operation | CMFA changed from 0 to 1 in 8TCSR2 by compare match | TMO ₂ output controllable | CMIA2 interrupt request generated by compare match |
| TCORB2 | Compare match operation | CMFB not changed from 0 to 1 in 8TCSR2 by compare match | No output from TMO ₂ | CMIB2 interrupt request not generated by compare match |
| TCORA3 | Compare match operation | CMFA changed from 0 to 1 in 8TCSR3 by compare match | TMIO ₃ is dedicated input capture pin | CMIA3 interrupt request generated by compare match |
| TCORB3 | Input capture operation | CMFB changed from 0 to 1 in 8TCSR3 by input capture | TMIO ₃ is dedicated input capture pin | CMIB3 interrupt request generated by input capture |

Bits 3 and 2—Output/Input Capture Edge Select B3 and B2 (OIS3, OIS2): In combination with the ICE bit in 8TCSR1 (8TCSR3), these bits select the compare match B output level or the input capture input detected edge.

The function of TCORB1 (TCORB3) depends on the setting of bit 4 of 8TCSR1 (8TCSR3).

ICE Bit in

| 8TCSR1 (8TCSR3) | Bit 3 OIS3 | Bit 2 OIS2 | Description |
|--------------------|---------------|---------------|--|
| 0 | 0 | 0 | No change when compare match B occurs (Initial value) |
| | | 1 | 0 is output when compare match B occurs |
| | 1 | 0 | 1 is output when compare match B occurs |
| | | 1 | Output is inverted when compare match B occurs (toggle output) |
| 1 | 0 | 0 | TCORB input capture on rising edge |
| | | 1 | TCORB input capture on falling edge |
| | 1 | 0 | TCORB input capture on both rising and falling edges |
| | | 1 | |

- When the compare match register function is used, the timer output priority order is: toggle output > 1 output > 0 output.
- If compare match A and B occur simultaneously, the output changes in accordance with the higher-priority compare match.
- When bits OIS3, OIS2, OS1, and OS0 are all cleared to 0, timer output is disabled.

Bits 1 and 0—Output Select A1 and A0 (OS1, OS0): These bits select the compare match A output level.

| Bit 1 OS1 | Bit 0 OS0 | Description |
|--------------|--------------|--|
| 0 | 0 | No change when compare match A occurs (Initial value) |
| | 1 | 0 is output when compare match A occurs |
| 1 | 0 | 1 is output when compare match A occurs |
| | 1 | Output is inverted when compare match A occurs (toggle output) |

- When the compare match register function is used, the timer output priority order is: toggle output > 1 output > 0 output.
- If compare match A and B occur simultaneously, the output changes in accordance with the higher-priority compare match.
- When bits OIS3, OIS2, OS1, and OS0 are all cleared to 0, timer output is disabled.

9.3 CPU Interface

9.3.1 8-Bit Registers

8TCNT, TCORA, TCORB, 8TCR, and 8TCSR are 8-bit registers. These registers are connected to the CPU by an internal 16-bit data bus and can be read and written a word at a time or a byte at a time.

Figures 9.2 and 9.3 show the operation in word read and write accesses to 8TCNT.

Figures 9.4 to 9.7 show the operation in byte read and write accesses to 8TCNT0 and 8TCNT1.

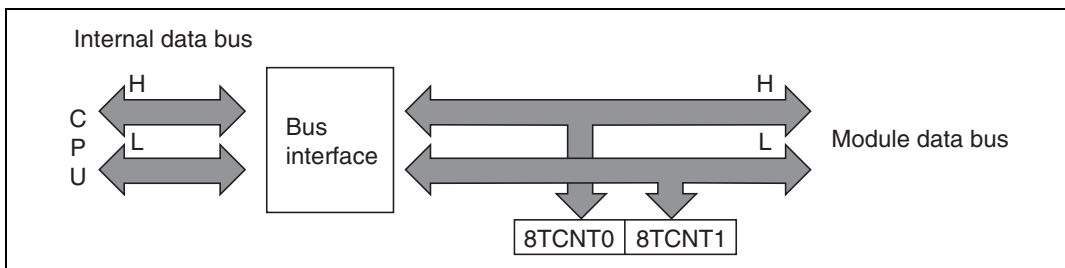


Figure 9.2 8TCNT Access Operation (CPU Writes to 8TCNT, Word)

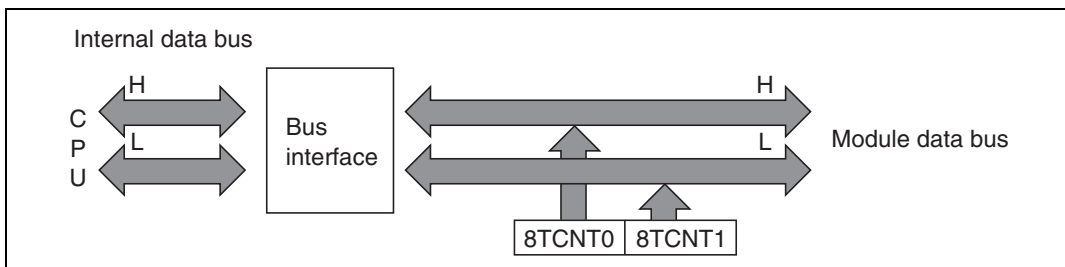


Figure 9.3 8TCNT Access Operation (CPU Reads 8TCNT, Word)

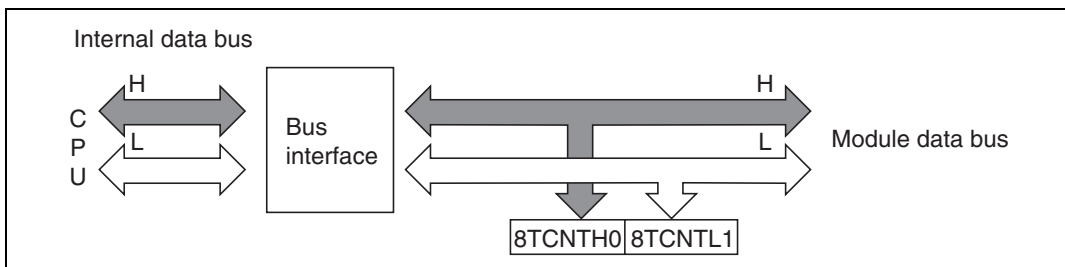


Figure 9.4 8TCNT0 Access Operation (CPU Writes to 8TCNT0, Upper Byte)

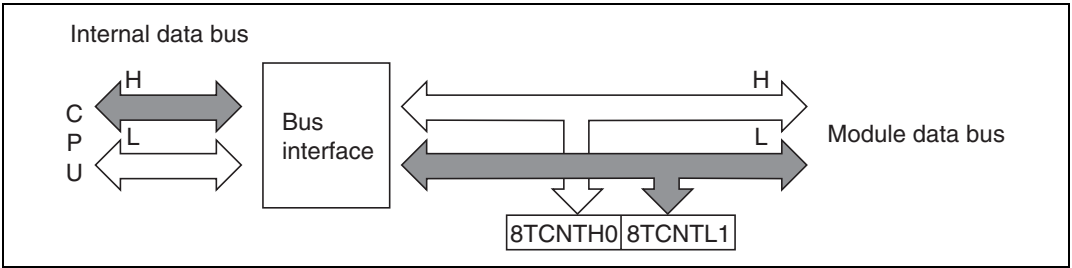


Figure 9.5 8TCNT1 Access Operation (CPU Writes to 8TCNT1, Lower Byte)

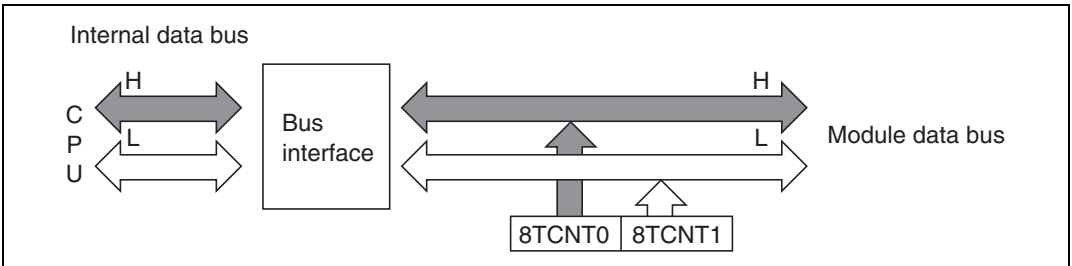


Figure 9.6 8TCNT0 Access Operation (CPU Reads 8TCNT0, Upper Byte)

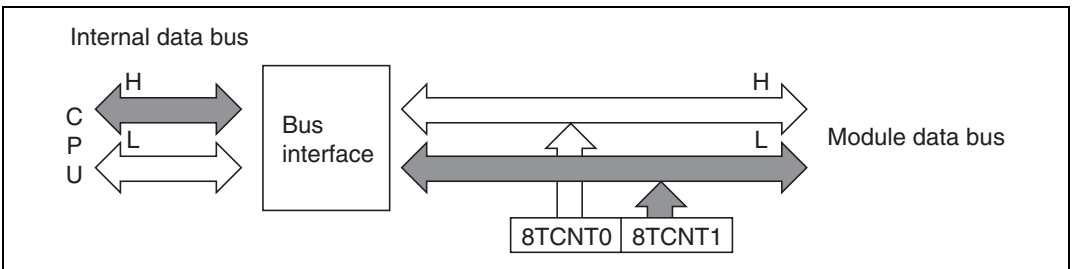


Figure 9.7 8TCNT1 Access Operation (CPU Reads 8TCNT1, Lower Byte)

9.4 Operation

9.4.1 8TCNT Count Timing

8TCNT is incremented by input clock pulses (either internal or external).

Internal Clock: Three different internal clock signals ($\phi/8$, $\phi/64$, or $\phi/8192$) divided from the system clock (ϕ) can be selected, by setting bits CKS2 to CKS0 in 8TCR. Figure 9.8 shows the count timing.

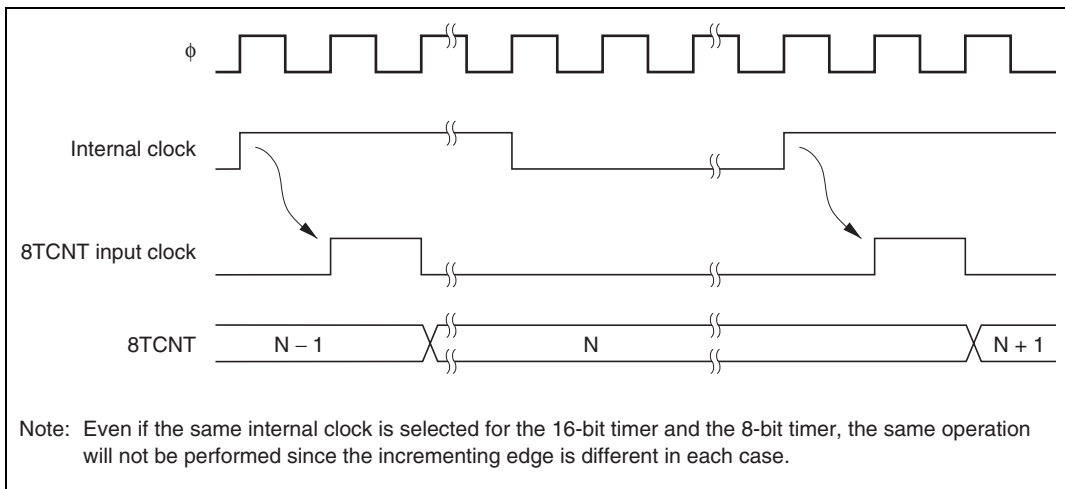


Figure 9.8 Count Timing for Internal Clock Input

External Clock: Three incrementation methods can be selected by setting bits CKS2 to CKS0 in 8TCR: on the rising edge, the falling edge, and both rising and falling edges.

The pulse width of the external clock signal must be at least 1.5 system clocks when a single edge is selected, and at least 2.5 system clocks when both edges are selected. Shorter pulses will not be counted correctly.

Figure 9.9 shows the timing for incrementation on both edges of the external clock signal.

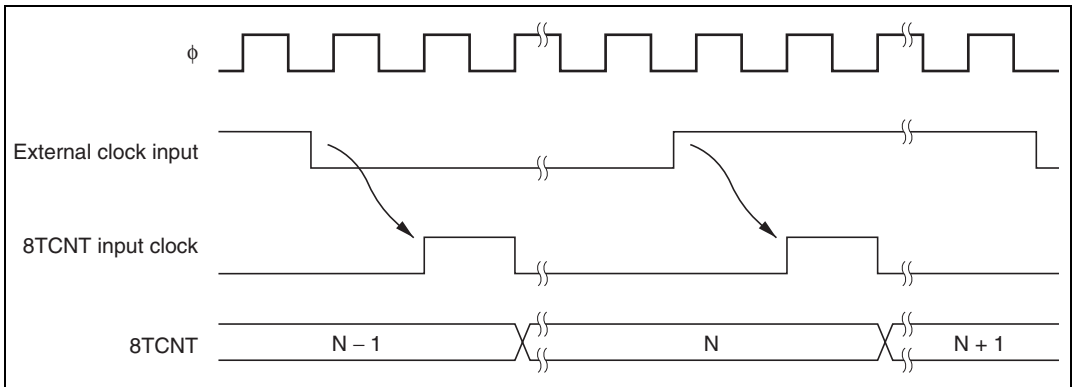


Figure 9.9 Count Timing for External Clock Input (Both-Edge Detection)

9.4.2 Compare Match Timing

Timer Output Timing: When compare match A or B occurs, the timer output is as specified by the OIS3, OIS2, OS1, and OS0 bits in 8TCSR (unchanged, 0 output, 1 output, or toggle output).

Figure 9.10 shows the timing when the output is set to toggle on compare match A.

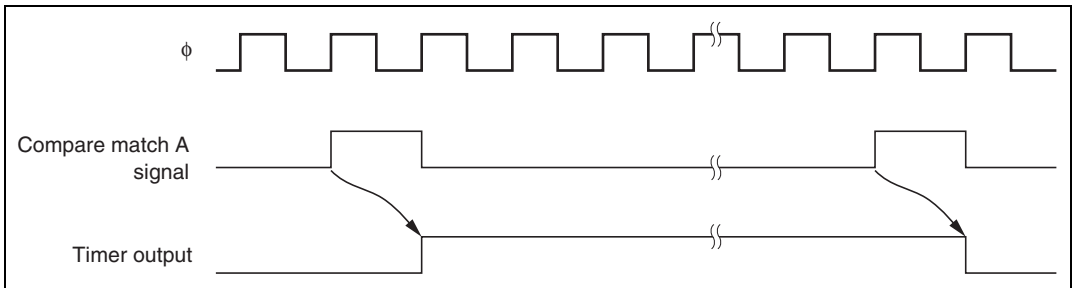


Figure 9.10 Timing of Timer Output

Clear by Compare Match: Depending on the setting of the CCLR1 and CCLR0 bits in 8TCR, 8TCNT can be cleared when compare match A or B occurs, Figure 9.11 shows the timing of this operation.

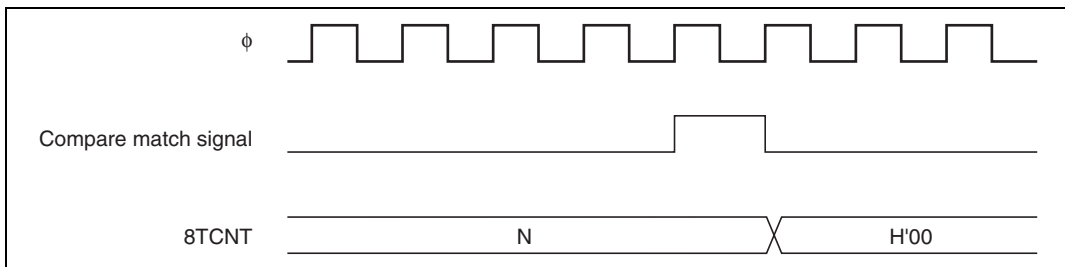


Figure 9.11 Timing of Clear by Compare Match

Clear by Input Capture: Depending on the setting of the CCLR1 and CCLR0 bits in 8TCR, 8TCNT can be cleared when input capture B occurs. Figure 9.12 shows the timing of this operation.

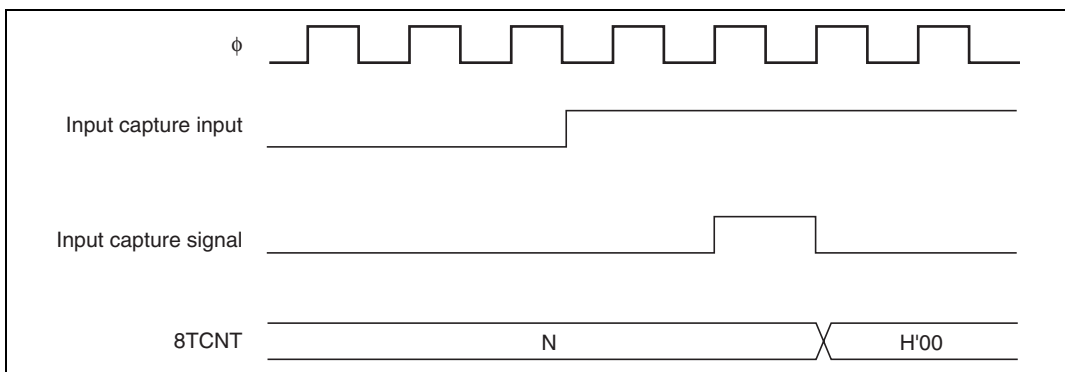


Figure 9.12 Timing of Clear by Input Capture

9.4.3 Input Capture Signal Timing

Input capture on the rising edge, falling edge, or both edges can be selected by settings in 8TCSR.

Figure 9.13 shows the timing when the rising edge is selected.

The pulse width of the input capture input signal must be at least 1.5 system clocks when a single edge is selected, and at least 2.5 system clocks when both edges are selected.

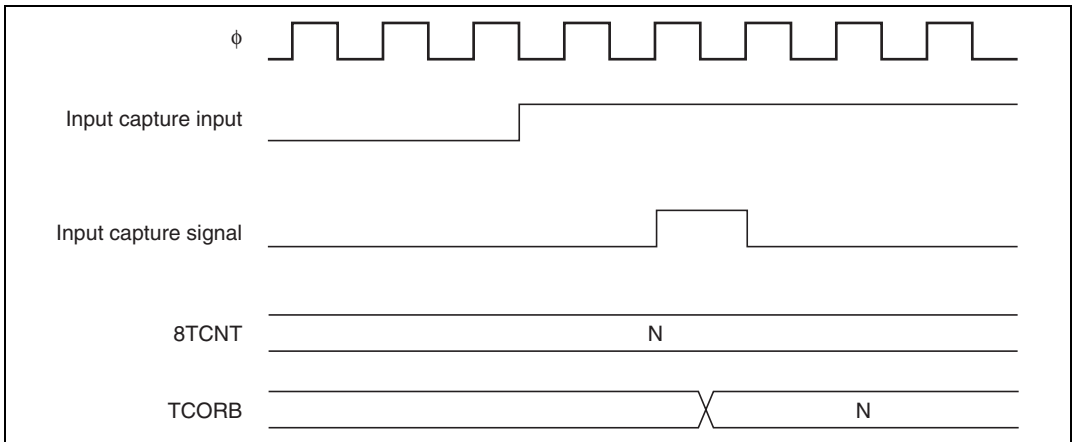


Figure 9.13 Timing of Input Capture Input Signal

9.4.4 Timing of Status Flag Setting

Timing of CMFA/CMFB Flag Setting when Compare Match Occurs: The CMFA and CMFB flags in 8TCSR are set to 1 by the compare match signal output when the TCORA or TCORB and 8TCNT values match. The compare match signal is generated in the last state of the match (when the matched 8TCNT count value is updated). Therefore, after the 8TCNT and TCORA or TCORB values match, the compare match signal is not generated until an incrementing clock pulse signal is generated. Figure 9.14 shows the timing in this case.

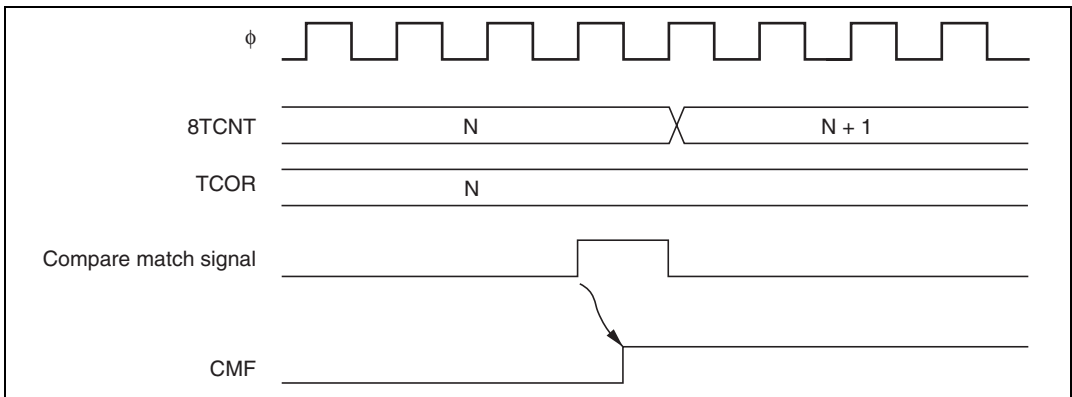


Figure 9.14 CMF Flag Setting Timing when Compare Match Occurs

Timing of CMFB Flag Setting when Input Capture Occurs: On generation of an input capture signal, the CMFB flag is set to 1 and at the same time the 8TCNT value is transferred to TCORB. Figure 9.15 shows the timing in this case.

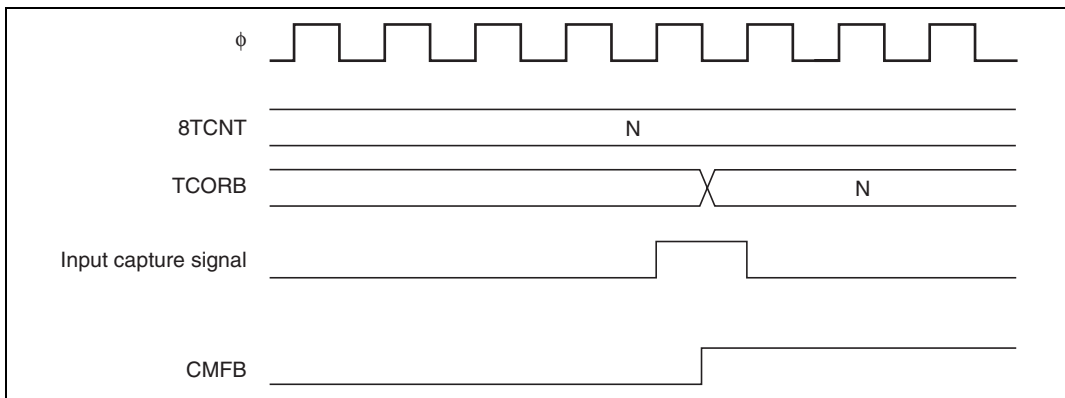


Figure 9.15 CMFB Flag Setting Timing when Input Capture Occurs

Timing of Overflow Flag (OVF) Setting: The OVF flag in 8TCSR is set to 1 by the overflow signal generated when 8TCNT overflows (from H'FF to H'00). Figure 9.16 shows the timing in this case.

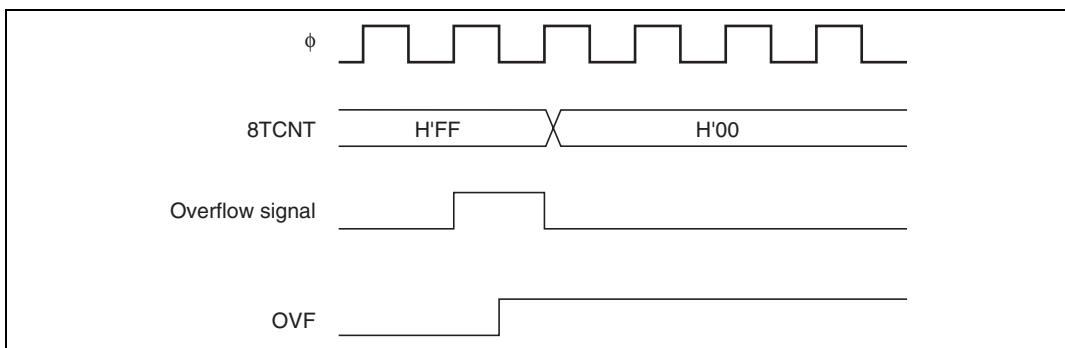


Figure 9.16 Timing of OVF Setting

9.4.5 Operation with Cascaded Connection

If bits CKS2 to CKS0 are set to (100) in either 8TCR0 or 8TCR1, the 8-bit timers of channels 0 and 1 are cascaded. With this configuration, the two timers can be used as a single 16-bit timer (16-bit timer mode), or channel 0 8-bit timer compare matches can be counted in channel 1 (compare match count mode). Similarly, if bits CKS2 to CKS0 are set to (100) in either 8TCR2 or 8TCR3, the 8-bit timers of channels 2 and 3 are cascaded. With this configuration, the two timers can be used as a single 16-bit timer (16-bit timer mode), or channel 2 8-bit timer compare matches can be counted in channel 3 (compare match count mode). In this case, the timer operates as below.

16-Bit Count Mode

- Channels 0 and 1:

When bits CKS2 to CKS0 are set to (100) in 8TCR0, the timer functions as a single 16-bit timer with channel 0 occupying the upper 8 bits and channel 1 occupying the lower 8 bits.

— Setting when Compare Match Occurs

- The CMFA or CMFB flag is set to 1 in 8TCSR0 when a 16-bit compare match occurs.
- The CMFA or CMFB flag is set to 1 in 8TCSR1 when a lower 8-bit compare match occurs.
- TMO₀ pin output control by bits OIS3, OIS2, OS1, and OS0 in 8TCSR0 is in accordance with the 16-bit compare match conditions.
- TMIO₁ pin output control by bits OIS3, OIS2, OS1, and OS0 in 8TCSR1 is in accordance with the lower 8-bit compare match conditions.

— Setting when Input Capture Occurs

- The CMFB flag is set to 1 in 8TCSR0 and 8TCSR1 when the ICE bit is 1 in TCSR1 and input capture occurs.
- TMIO₁ pin input capture input signal edge detection is selected by bits OIS3 and OIS2 in 8TCSR0.

— Counter Clear Specification

- If counter clear on compare match or input capture has been selected by the CCLR1 and CCLR0 bits in 8TCR0, the 16-bit counter (both 8TCNT0 and 8TCNT1) is cleared.
- The settings of the CCLR1 and CCLR0 bits in 8TCR1 are ignored. The lower 8 bits cannot be cleared independently.

— OVF Flag Operation

- The OVF flag is set to 1 in 8TCSR0 when the 16-bit counter (8TCNT0 and 8TCNT1) overflows (from H'FFFF to H'0000).
- The OVF flag is set to 1 in 8TCSR1 when the 8-bit counter (8TCNT1) overflows (from H'FF to H'00).

- Channels 2 and 3:

When bits CKS2 to CKS0 are set to (100) in 8TCR2, the timer functions as a single 16-bit timer with channel 2 occupying the upper 8 bits and channel 3 occupying the lower 8 bits.

— Setting when Compare Match Occurs

- The CMFA or CMFB flag is set to 1 in 8TCSR2 when a 16-bit compare match occurs.
- The CMFA or CMFB flag is set to 1 in 8TCSR3 when a lower 8-bit compare match occurs.
- TMO₂ pin output control by bits OIS3, OIS2, OS1, and OS0 in 8TCSR2 is in accordance with the 16-bit compare match conditions.

- TMIO₃ pin output control by bits OIS3, OIS2, OS1, and OS0 in 8TCSR3 is in accordance with the lower 8-bit compare match conditions.
- Setting when Input Capture Occurs
- The CMFB flag is set to 1 in 8TCSR2 and 8TCSR3 when the ICE bit is 1 in TCSR3 and input capture occurs.
 - TMIO₃ pin input capture input signal edge detection is selected by bits OIS3 and OIS2 in 8TCSR2.
- Counter Clear Specification
- If counter clear on compare match has been selected by the CCLR1 and CCLR0 bits in 8TCR2, the 16-bit counter (both 8TCNT2 and 8TCNT3) is cleared.
 - The settings of the CCLR1 and CCLR0 bits in 8TCR3 are ignored. The lower 8 bits cannot be cleared independently.
- OVF Flag Operation
- The OVF flag is set to 1 in 8TCSR2 when the 16-bit counter (8TCNT2 and 8TCNT3) overflows (from H'FFFF to H'0000).
 - The OVF flag is set to 1 in 8TCSR3 when the 8-bit counter (8TCNT3) overflows (from H'FF to H'00).

Compare Match Count Mode

- Channels 0 and 1:

When bits CKS2 to CKS0 are set to (100) in 8TCR1, 8TCNT1 counts channel 0 compare match A events.

Channels 0 and 1 are controlled independently.

CMF flag setting, interrupt generation, TMO pin output, counter clearing, and so on, is in accordance with the settings for each channel.

Note: When bit ICE = 1 in 8TCSR1, the compare match register function of TCORB0 in channel 0 cannot be used.
- Channels 2 and 3:

When bits CKS2 to CKS0 are set to (100) in 8TCR3, 8TCNT3 counts channel 2 compare match A events.

Channels 2 and 3 are controlled independently.

CMF flag setting, interrupt generation, TMO pin output, counter clearing, and so on, is in accordance with the settings for each channel.

Note: When bit ICE = 1 in 8TCSR3, the compare match register function of TCORB2 in channel 2 cannot be used.

Caution

Do not set 16-bit counter mode and compare match count mode simultaneously within the same group, as the 8TCNT input clock will not be generated and the counters will not operate.

9.4.6 Input Capture Setting

The 8TCNT value can be transferred to TCORB on detection of an input edge on the input capture/output compare pin (TMIO₁ or TMIO₃). Rising edge, falling edge, or both edge detection can be selected. In 16-bit count mode, 16-bit input capture can be used.

Setting Input Capture Operation in 8-Bit Timer Mode (Normal Operation)

- Channel 1:
 - Set TCORB1 as an 8-bit input capture register with the ICE bit in 8TCSR1.
 - Select rising edge, falling edge, or both edges as the input edge(s) for the input capture signal (TMIO₁) with bits OIS3 and OIS2 in 8TCSR1.
 - Select the input clock with bits CKS2 to CKS0 in 8TCR1, and start the 8TCNT count.
- Channel 3:
 - Set TCORB3 as an 8-bit input capture register with the ICE bit in 8TCSR3.
 - Select rising edge, falling edge, or both edges as the input edge(s) for the input capture signal (TMIO₃) with bits OIS3 and OIS2 in 8TCSR3.
 - Select the input clock with bits CKS2 to CKS0 in 8TCR3, and start the 8TCNT count.

Note: When TCORB1 in channel 1 is used for input capture, TCORB0 in channel 0 cannot be used as a compare match register.

Similarly, when TCORB3 in channel 3 is used for input capture, TCORB2 in channel 2 cannot be used as a compare match register.

Setting Input Capture Operation in 16-Bit Count Mode

- Channels 0 and 1:
 - In 16-bit count mode, TCORB0 and TCORB1 function as a 16-bit input capture register when the ICE bit is set to 1 in 8TCSR1.
 - Select rising edge, falling edge, or both edges as the input edge(s) for the input capture signal (TMIO₁) with bits OIS3 and OIS2 in 8TCSR0. (In 16-bit count mode, the settings of bits OIS3 and OIS2 in 8TCSR1 are ignored.)
 - Select the input clock with bits CKS2 to CKS0 in 8TCR1, and start the 8TCNT count.
- Channels 2 and 3:
 - In 16-bit count mode, TCORB2 and TCORB3 function as a 16-bit input capture register when the ICE bit is set to 1 in 8TCSR3.

- Select rising edge, falling edge, or both edges as the input edge(s) for the input capture signal (TMIO₃) with bits OIS3 and OIS2 in 8TCSR2. (In 16-bit count mode, the settings of bits OIS3 and OIS2 in 8TCSR3 are ignored.)
- Select the input clock with bits CKS2 to CKS0 in 8TCR3, and start the 8TCNT count.

9.5 Interrupt

9.5.1 Interrupt Sources

The 8-bit timer unit can generate three types of interrupt: compare match A and B (CMIA and CMIB) and overflow (TOVI). Table 9.5 shows the interrupt sources and their priority order. Each interrupt source is enabled or disabled by the corresponding interrupt enable bit in 8TCR. A separate interrupt request signal is sent to the interrupt controller by each interrupt source.

Table 9.5 Types of 8-Bit Timer Interrupt Sources and Priority Order

| Interrupt Source | Description | Priority |
|------------------|-------------------|----------|
| CMIA | Interrupt by CMFA | High |
| CMIB | Interrupt by CMFB | ↑ |
| TOVI | Interrupt by OVF | Low |

For compare match interrupts CMIA1/CMIB1 and CMIA3/CMIB3 and the overflow interrupts (TOVI0/TOVI1 and TOVI2/TOVI3), one vector is shared by two interrupts.

Table 9.6 lists the interrupt sources.

Table 9.6 8-Bit Timer Interrupt Sources

| Channel | Interrupt Source | Description |
|---------|------------------|---|
| 0 | CMIA0 | TCORA0 compare match |
| | CMIB0 | TCORB0 compare match/input capture |
| 1 | CMIA1/CMIB1 | TCORA1 compare match, or TCORB1 compare match/input capture |
| 0, 1 | TOVI0/TOVI1 | Counter 0 or counter 1 overflow |
| 2 | CMIA2 | TCORA2 compare match |
| | CMIB2 | TCORB2 compare match/input capture |
| 3 | CMIA3/CMIB3 | TCORA3 compare match, or TCORB3 compare match/input capture |
| 2, 3 | TOVI2/TOVI3 | Counter 2 or counter 3 overflow |

9.5.2 A/D Converter Activation

The A/D converter can only be activated by channel 0 compare match A.

If the ADTE bit setting is 1 when the CMFA flag in 8TCSR0 is set to 1 by generation of channel 0 compare match A, an A/D conversion start request will be issued to the A/D converter. If the TRGE bit in ADCR is 1 at this time, the A/D converter will be started. If the ADTE bit in 8TCSR0 is 1, A/D converter external trigger pin ($\overline{\text{ADTRG}}$) input is disabled.

9.6 8-Bit Timer Application Example

Figure 9.17 shows how the 8-bit timer module can be used to output pulses with any desired duty cycle. The settings for this example are as follows:

- Clear the CCLR1 bit to 0 and set the CCLR0 bit to 1 in 8TCR so that 8TCNT is cleared by a TCORA compare match.
- Set bits OIS3, OIS2, OS1, and OS0 to (0110) in 8TCSR so that 1 is output on a TCORA compare match and 0 is output on a TCORB compare match.

The above settings enable a waveform with the cycle determined by TCORA and the pulse width detected by TCORB to be output without software intervention.

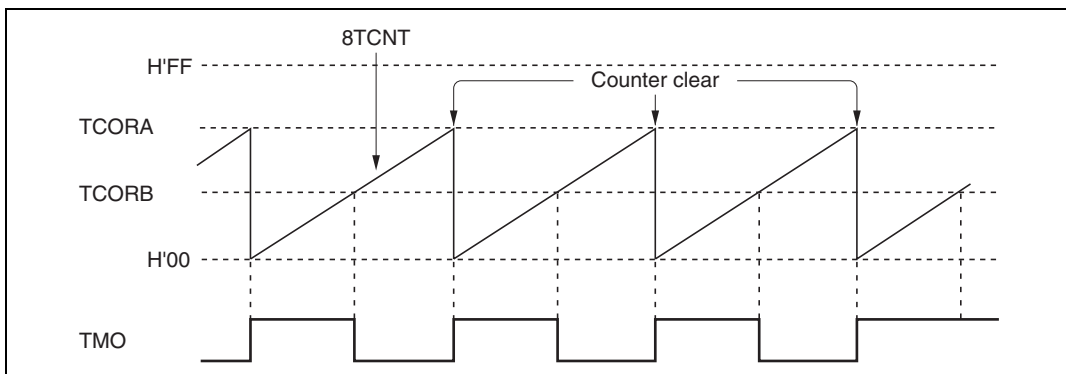


Figure 9.17 Example of Pulse Output

9.7 Usage Notes

Note that the following kinds of contention can occur in 8-bit timer operation.

9.7.1 Contention between 8TCNT Write and Clear

If a timer counter clear signal occurs in the T_3 state of a 8TCNT write cycle, clearing of the counter takes priority and the write is not performed. Figure 9.18 shows the timing in this case.

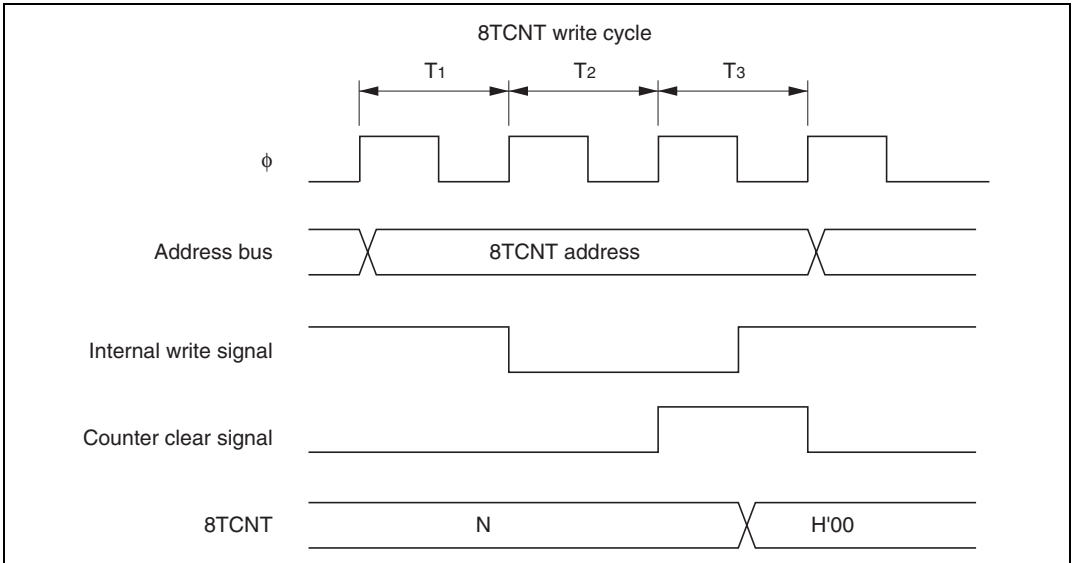


Figure 9.18 Contention between 8TCNT Write and Clear

9.7.2 Contention between 8TCNT Write and Increment

If an increment pulse occurs in the T_3 state of a 8TCNT write cycle, writing takes priority and 8TCNT is not incremented. Figure 9.19 shows the timing in this case.

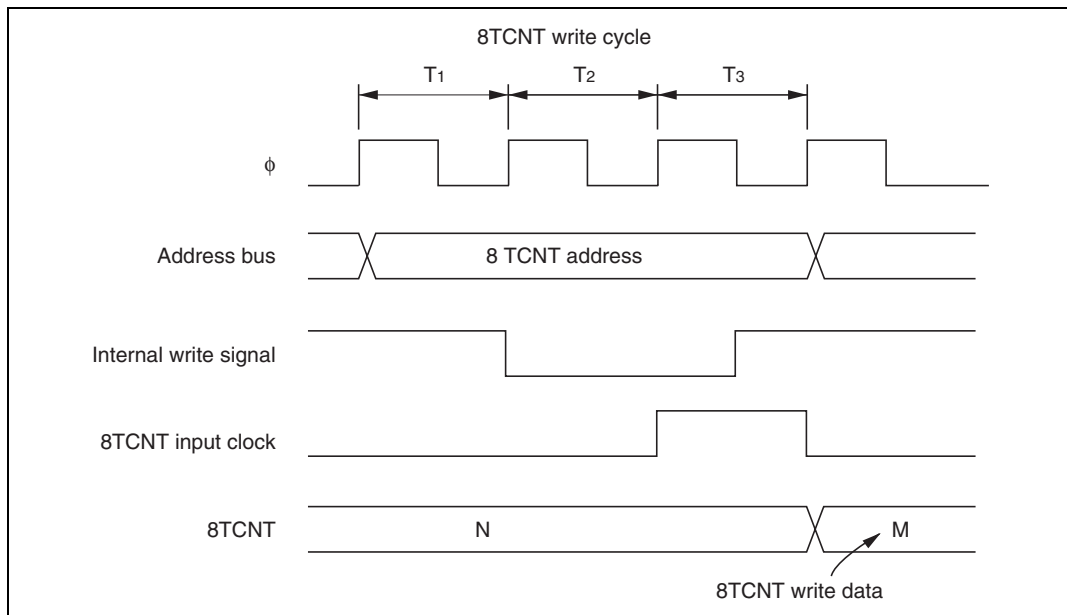


Figure 9.19 Contention between 8TCNT Write and Increment

9.7.3 Contention between TCOR Write and Compare Match

If a compare match occurs in the T_3 state of a TCOR write cycle, writing takes priority and the compare match signal is inhibited. Figure 9.20 shows the timing in this case.

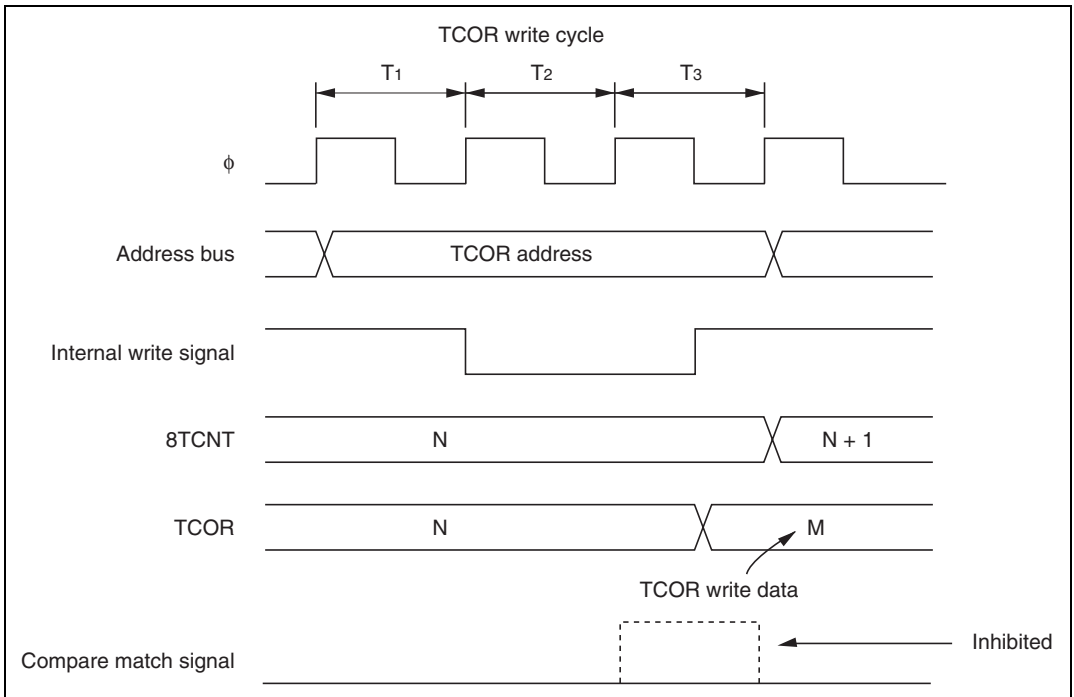


Figure 9.20 Contention between TCOR Write and Compare Match

9.7.4 Contention between TCOR Read and Input Capture

If an input capture signal occurs in the T_3 state of a TCOR read cycle, the value before input capture is read. Figure 9.21 shows the timing in this case.

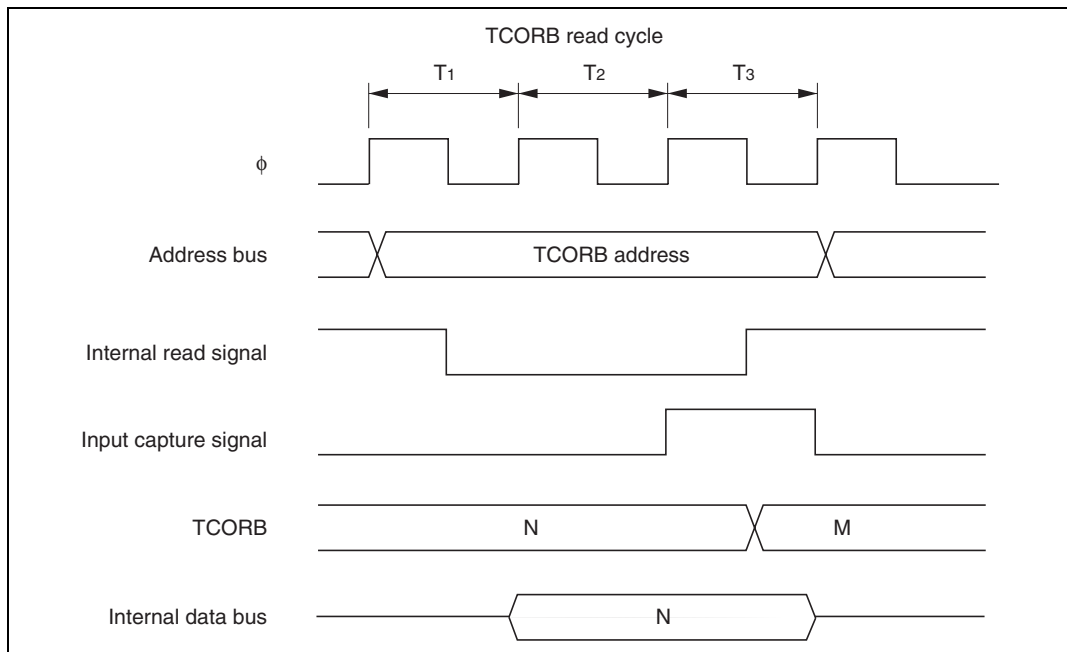


Figure 9.21 Contention between TCOR Read and Input Capture

9.7.5 Contention between Counter Clearing by Input Capture and Counter Increment

If an input capture signal and counter increment signal occur simultaneously, counter clearing by the input capture signal takes priority and the counter is not incremented. The value before the counter is cleared is transferred to TCORB. Figure 9.22 shows the timing in this case.

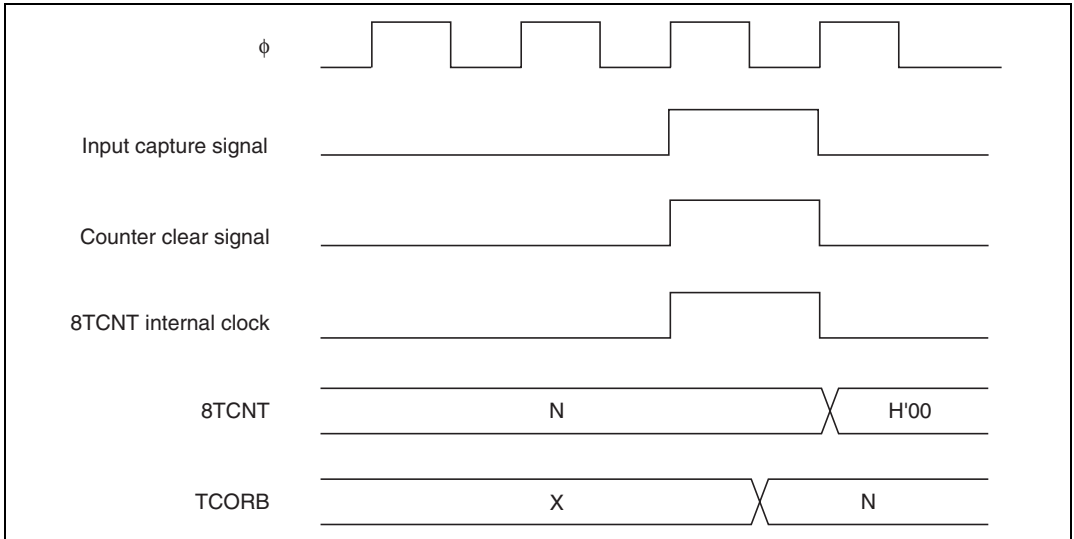


Figure 9.22 Contention between Counter Clearing by Input Capture and Counter Increment

9.7.6 Contention between TCOR Write and Input Capture

If an input capture signal occurs in the T_3 state of a TCOR write cycle, input capture takes priority and the write to TCOR is not performed. Figure 9.23 shows the timing in this case.

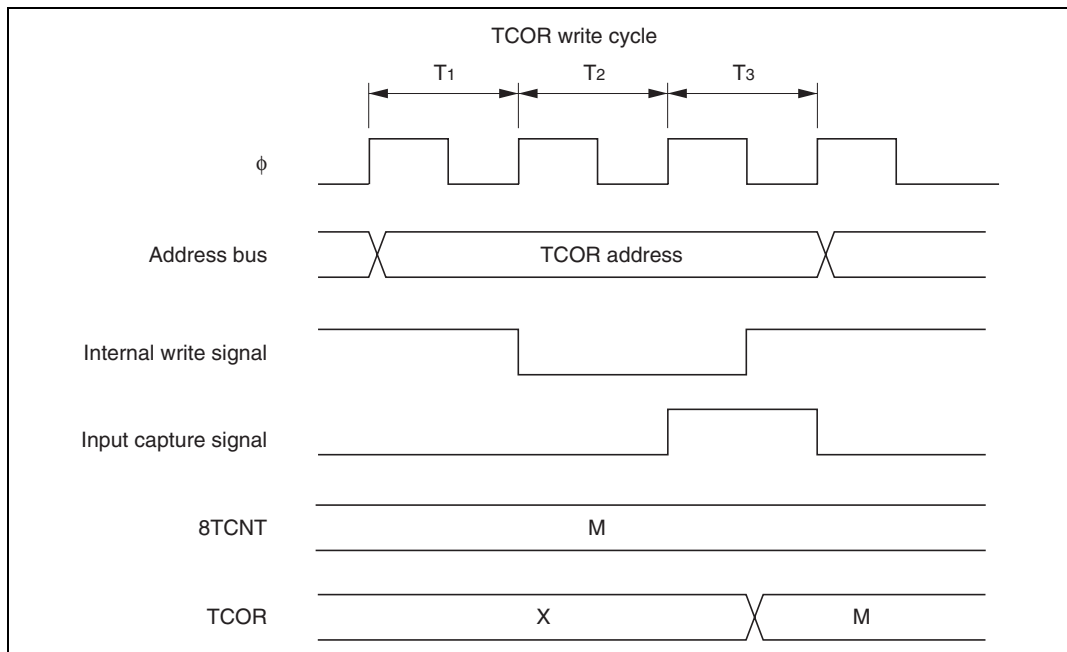


Figure 9.23 Contention between TCOR Write and Input Capture

9.7.7 Contention between 8TCNT Byte Write and Increment in 16-Bit Count Mode (Cascaded Connection)

If an increment pulse occurs in the T_2 or T_3 state of an 8TCNT byte write cycle in 16-bit count mode, the counter write takes priority and the byte data for which the write was performed is not incremented. The byte data for which a write was not performed is incremented. Figure 9.24 shows the timing when an increment pulse occurs in the T_2 state of a byte write to 8TCNT (upper byte). If an increment pulse occurs in the T_2 state, on the other hand, the increment takes priority.

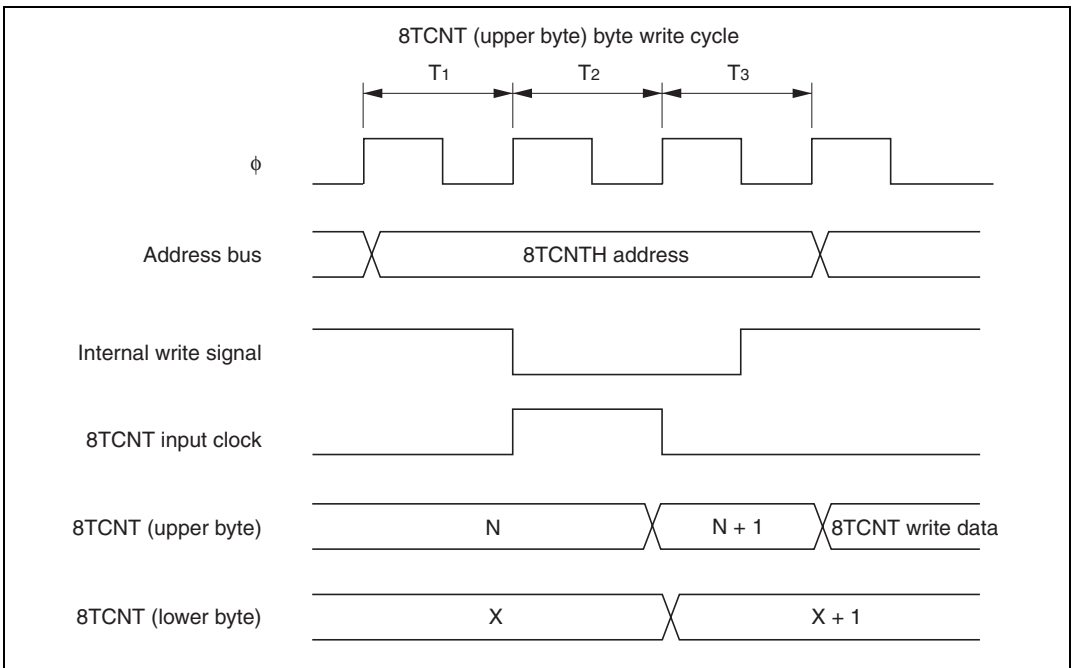


Figure 9.24 Contention between 8TCNT Byte Write and Increment in 16-Bit Count Mode

9.7.8 Contention between Compare Matches A and B

If compare matches A and B occur at the same time, the 8-bit timer operates according to the relative priority of the output states set for compare match A and compare match B, as shown in Table 9.7.

Table 9.7 Timer Output Priority Order

| Output Setting | Priority |
|----------------|----------|
| Toggle output | High |
| 1 output | ↑ |
| 0 output | |
| No change | Low |

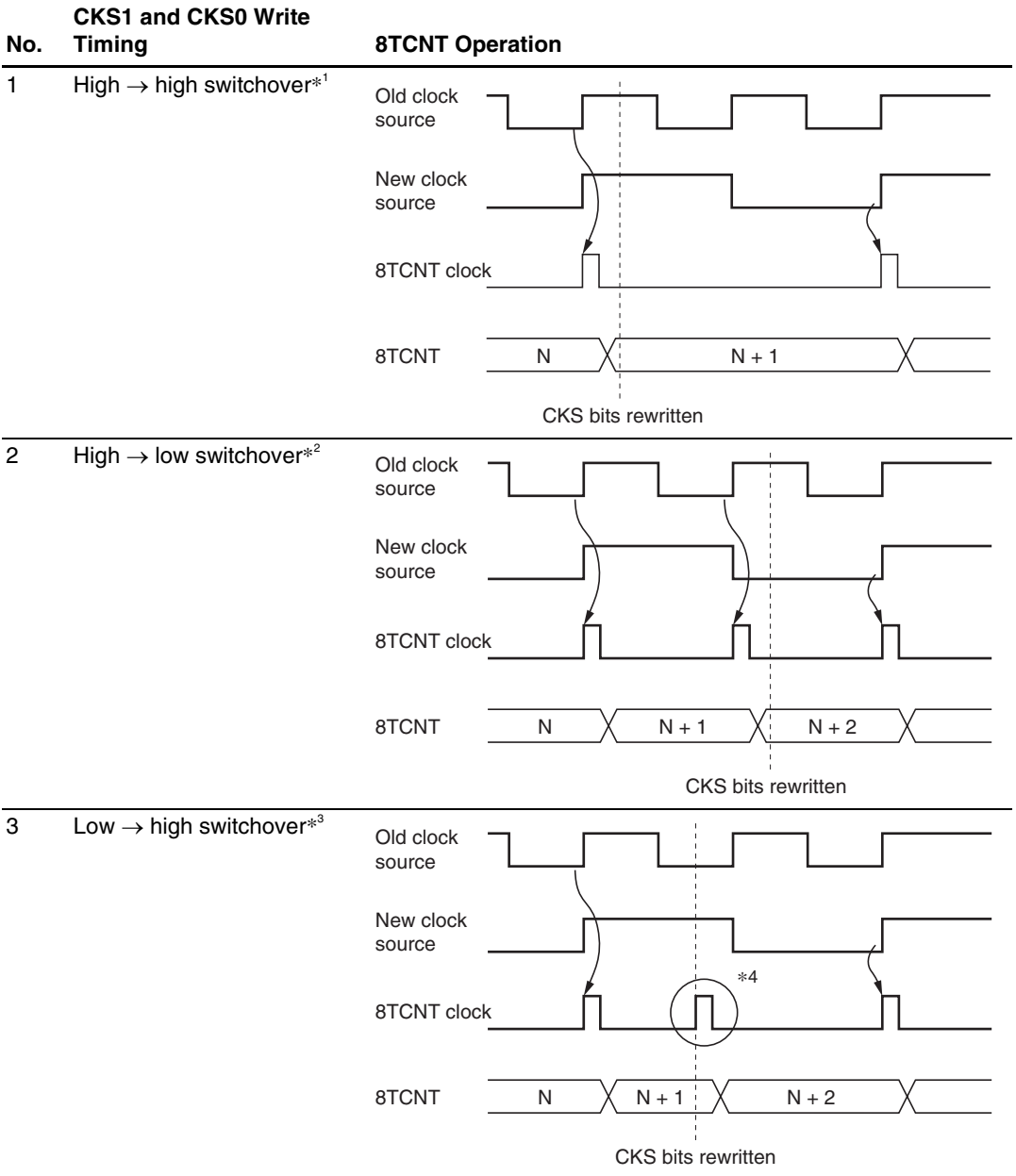
9.7.9 8TCNT Operation and Internal Clock Source Switchover

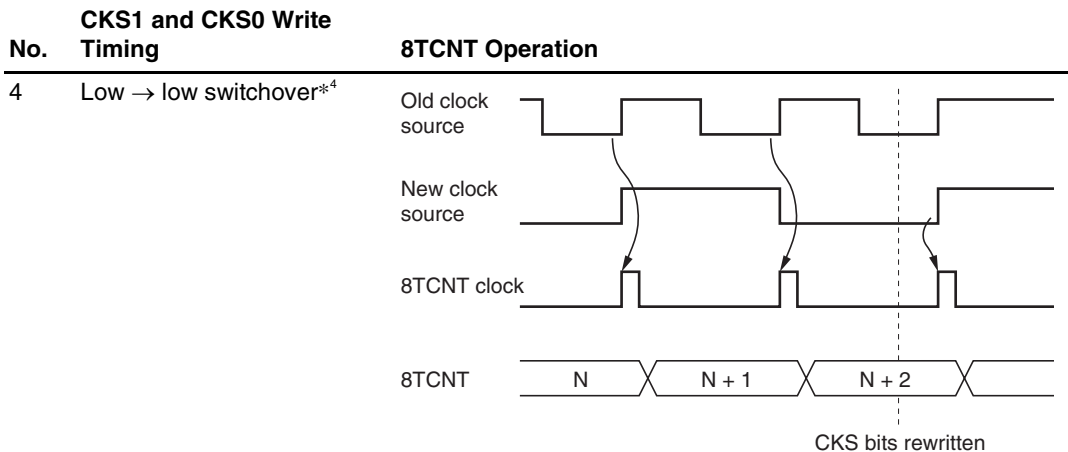
Switching internal clock sources may cause 8TCNT to increment, depending on the switchover timing. Table 9.8 shows the relation between the time of the switchover (by writing to bits CKS1 and CKS0) and the operation of 8TCNT.

The 8TCNT input clock is generated from the internal clock source by detecting the rising edge of the internal clock. If a switchover is made from a low clock source to a high clock source, as in case No. 3 in Table 9.8, the switchover will be regarded as a falling edge, a 8TCNT clock pulse will be generated, and 8TCNT will be incremented.

8TCNT may also be incremented when switching between internal and external clocks.

Table 9.8 Internal Clock Switchover and 8TCNT Operation





- Notes:
1. Including switchovers from the high level to the halted state, and from the halted state to the high level.
 2. Including switchover from the halted state to the low level.
 3. Including switchover from the low level to the halted state.
 4. The switchover is regarded as a rising edge, causing 8TCNT to increment.

Section 10 Programmable Timing Pattern Controller (TPC)

10.1 Overview

The H8/3008 has a built-in programmable timing pattern controller (TPC) that provides pulse outputs by using the 16-bit timer as a time base. The TPC pulse outputs are divided into 4-bit groups (group 3 to group 0) that can operate simultaneously and independently.

10.1.1 Features

TPC features are listed below.

- 16-bit output data
Maximum 16-bit data can be output. TPC output can be enabled on a bit-by-bit basis.
- Four output groups
Output trigger signals can be selected in 4-bit groups to provide up to four different 4-bit outputs.
- Selectable output trigger signals
Output trigger signals can be selected for each group from the compare match signals of three 16-bit timer channels.
- Non-overlap mode
A non-overlap margin can be provided between pulse outputs.

10.1.2 Block Diagram

Figure 10.1 shows a block diagram of the TPC.

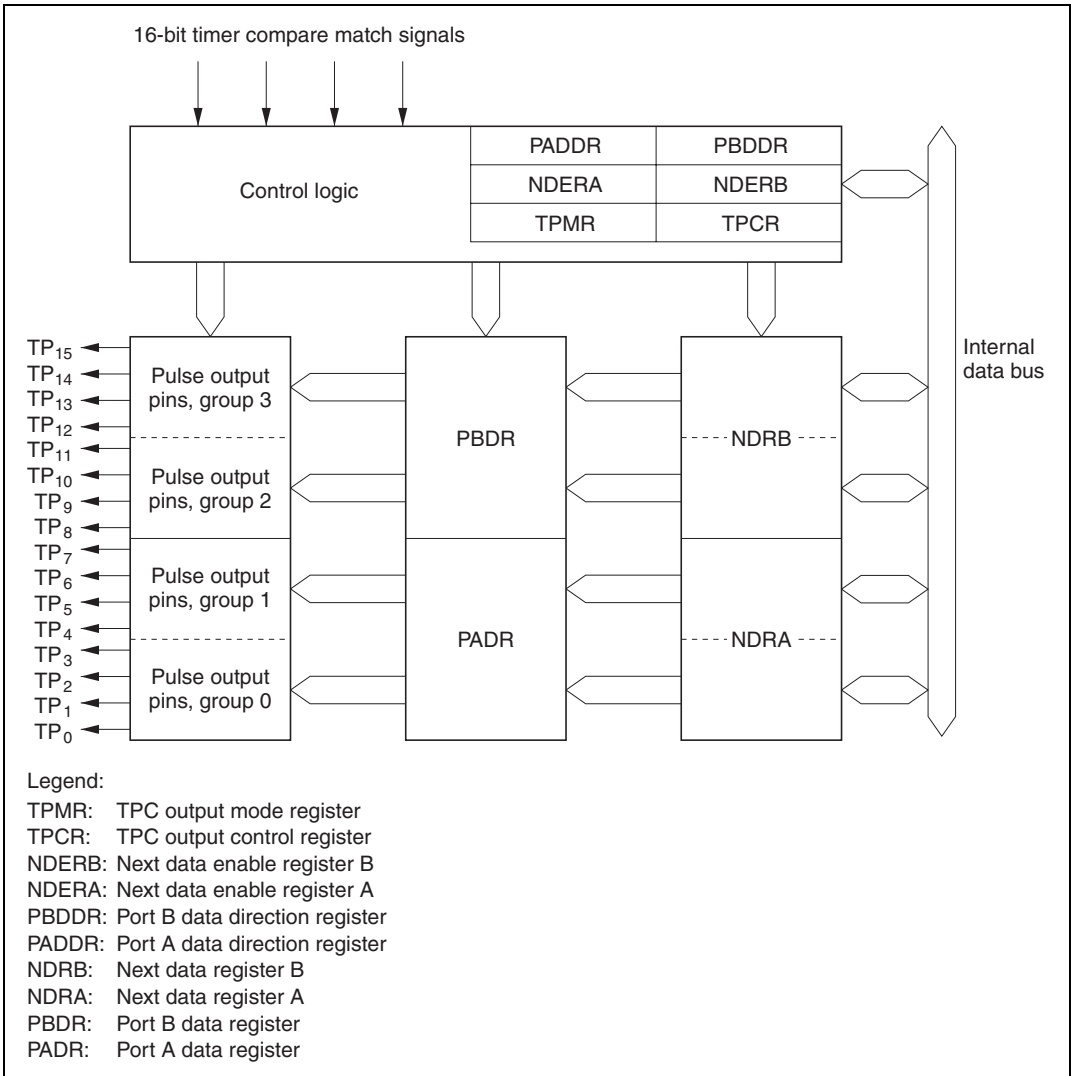


Figure 10.1 TPC Block Diagram

10.1.3 Pin Configuration

Table 10.1 summarizes the TPC output pins.

Table 10.1 TPC Pins

| Name | Symbol | I/O | Function |
|---------------|------------------|------------|----------------------|
| TPC output 0 | TP ₀ | Output | Group 0 pulse output |
| TPC output 1 | TP ₁ | Output | |
| TPC output 2 | TP ₂ | Output | |
| TPC output 3 | TP ₃ | Output | Group 1 pulse output |
| TPC output 4 | TP ₄ | Output | |
| TPC output 5 | TP ₅ | Output | |
| TPC output 6 | TP ₆ | Output | |
| TPC output 7 | TP ₇ | Output | |
| TPC output 8 | TP ₈ | Output | Group 2 pulse output |
| TPC output 9 | TP ₉ | Output | |
| TPC output 10 | TP ₁₀ | Output | |
| TPC output 11 | TP ₁₁ | Output | Group 3 pulse output |
| TPC output 12 | TP ₁₂ | Output | |
| TPC output 13 | TP ₁₃ | Output | |
| TPC output 14 | TP ₁₄ | Output | |
| TPC output 15 | TP ₁₅ | Output | |

10.1.4 Register Configuration

Table 10.2 summarizes the TPC registers.

Table 10.2 TPC Registers

| Address* ¹ | Name | Abbreviation | R/W | Initial Value |
|-----------------------------------|--------------------------------|--------------|---------------------|---------------|
| H'EE009 | Port A data direction register | PADDR | W | H'00 |
| H'FFFD9 | Port A data register | PADR | R/(W)* ² | H'00 |
| H'EE00A | Port B data direction register | PBDDR | W | H'00 |
| H'FFFDA | Port B data register | PBDR | R/(W)* ² | H'00 |
| H'FFFA0 | TPC output mode register | TPMR | R/W | H'F0 |
| H'FFFA1 | TPC output control register | TPCR | R/W | H'FF |
| H'FFFA2 | Next data enable register B | NDERB | R/W | H'00 |
| H'FFFA3 | Next data enable register A | NDERA | R/W | H'00 |
| H'FFFA5/ H'FFFA7* ³ | Next data register A | NDRA | R/W | H'00 |
| H'FFFA4/ H'FFFA6* ³ | Next data register B | NDRB | R/W | H'00 |

Notes: 1. Lower 20 bits of the address in advanced mode.

2. Bits used for TPC output cannot be written.

3. The NDRA address is H'FFFA5 when the same output trigger is selected for TPC output groups 0 and 1 by settings in TPCR. When the output triggers are different, the NDRA address is H'FFFA7 for group 0 and H'FFFA5 for group 1. Similarly, the address of NDRB is H'FFFA4 when the same output trigger is selected for TPC output groups 2 and 3 by settings in TPCR. When the output triggers are different, the NDRB address is H'FFFA6 for group 2 and H'FFFA4 for group 3.

10.2 Register Descriptions

10.2.1 Port A Data Direction Register (PADDDR)

PADDDR is an 8-bit write-only register that selects input or output for each pin in port A.

| | | | | | | | | |
|---------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | PA ₇ DDR | PA ₆ DDR | PA ₅ DDR | PA ₄ DDR | PA ₃ DDR | PA ₂ DDR | PA ₁ DDR | PA ₀ DDR |
| Initial value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Read/Write | W | W | W | W | W | W | W | W |

Port A data direction 7 to 0

These bits select input or output for port A pins

Port A is multiplexed with pins TP₇ to TP₀. Bits corresponding to pins used for TPC output must be set to 1. For further information about PADDDR, see section 7.11, Port A.

10.2.2 Port A Data Register (PADR)

PADR is an 8-bit readable/writable register that stores TPC output data for groups 0 and 1, when these TPC output groups are used.

| | | | | | | | | |
|---------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | PA ₇ | PA ₆ | PA ₅ | PA ₄ | PA ₃ | PA ₂ | PA ₁ | PA ₀ |
| Initial value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Read/Write | R/(W)* | R/(W)* | R/(W)* | R/(W)* | R/(W)* | R/(W)* | R/(W)* | R/(W)* |

Port A data 7 to 0

These bits store output data for TPC output groups 0 and 1

Note: * Bits selected for TPC output by NDERA settings become read-only bits.

For further information about PADR, see section 7.11, Port A.

10.2.3 Port B Data Direction Register (PBDDR)

PBDDR is an 8-bit write-only register that selects input or output for each pin in port B.

| | | | | | | | | |
|---------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | PB ₇ DDR | PB ₆ DDR | PB ₅ DDR | PB ₄ DDR | PB ₃ DDR | PB ₂ DDR | PB ₁ DDR | PB ₀ DDR |
| Initial value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Read/Write | W | W | W | W | W | W | W | W |

Port B data direction 7 to 0

These bits select input or output for port B pins

Port B is multiplexed with pins TP₁₅ to TP₈. Bits corresponding to pins used for TPC output must be set to 1. For further information about PBDDR, see section 7.12, Port B.

10.2.4 Port B Data Register (PBDR)

PBDR is an 8-bit readable/writable register that stores TPC output data for groups 2 and 3, when these TPC output groups are used.

| | | | | | | | | |
|---------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | PB ₇ | PB ₆ | PB ₅ | PB ₄ | PB ₃ | PB ₂ | PB ₁ | PB ₀ |
| Initial value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Read/Write | R/(W)* | R/(W)* | R/(W)* | R/(W)* | R/(W)* | R/(W)* | R/(W)* | R/(W)* |

Port B data 7 to 0

These bits store output data for TPC output groups 2 and 3

Note: * Bits selected for TPC output by NDERB settings become read-only bits.

For further information about PBDR, see section 7.12, Port B.

10.2.5 Next Data Register A (NDRA)

NDRA is an 8-bit readable/writable register that stores the next output data for TPC output groups 1 and 0 (pins TP₇ to TP₀). During TPC output, when an 16-bit timer compare match event specified in TPCR occurs, NDRA contents are transferred to the corresponding bits in PADR. The address of NDRA differs depending on whether TPC output groups 0 and 1 have the same output trigger or different output triggers.

NDRA is initialized to H'00 by a reset and in hardware standby mode. It is not initialized in software standby mode.

Same Trigger for TPC Output Groups 0 and 1: If TPC output groups 0 and 1 are triggered by the same compare match event, the NDRA address is H'FFFA5. The upper 4 bits belong to group 1 and the lower 4 bits to group 0. Address H'FFFA7 consists entirely of reserved bits that cannot be modified and always read 1.

Address H'FFFA5

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|------|------|------|------|------|------|------|------|
| | NDR7 | NDR6 | NDR5 | NDR4 | NDR3 | NDR2 | NDR1 | NDR0 |
| Initial value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

| | |
|--|--|
| Next data 7 to 4 | Next data 3 to 0 |
| These bits store the next output data for TPC output group 1 | These bits store the next output data for TPC output group 0 |

Address H'FFFA7

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — |
| Initial value | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Read/Write | — | — | — | — | — | — | — | — |

Reserved bits

Different Triggers for TPC Output Groups 0 and 1: If TPC output groups 0 and 1 are triggered by different compare match events, the address of the upper 4 bits of NDRA (group 1) is H'FFFA5 and the address of the lower 4 bits (group 0) is H'FFFA7. Bits 3 to 0 of address H'FFFA5 and bits 7 to 4 of address H'FFFA7 are reserved bits that cannot be modified and always read 1.

Address H'FFFA5

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|------|------|------|------|---|---|---|---|
| | NDR7 | NDR6 | NDR5 | NDR4 | — | — | — | — |
| Initial value | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| Read/Write | R/W | R/W | R/W | R/W | — | — | — | — |

Next data 7 to 4
Reserved bits

These bits store the next output data for TPC output group 1

Address H'FFFA7

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|---|---|---|---|------|------|------|------|
| | — | — | — | — | NDR3 | NDR2 | NDR1 | NDR0 |
| Initial value | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| Read/Write | — | — | — | — | R/W | R/W | R/W | R/W |

Reserved bits
Next data 3 to 0

These bits store the next output data for TPC output group 0

10.2.6 Next Data Register B (NDRB)

NDRB is an 8-bit readable/writable register that stores the next output data for TPC output groups 3 and 2 (pins TP₁₅ to TP₈). During TPC output, when an 16-bit timer compare match event specified in TPCR occurs, NDRB contents are transferred to the corresponding bits in PBDR. The address of NDRB differs depending on whether TPC output groups 2 and 3 have the same output trigger or different output triggers.

NDRB is initialized to H'00 by a reset and in hardware standby mode. It is not initialized in software standby mode.

Same Trigger for TPC Output Groups 2 and 3: If TPC output groups 2 and 3 are triggered by the same compare match event, the NDRB address is H'FFFA4. The upper 4 bits belong to group 3 and the lower 4 bits to group 2. Address H'FFFA6 consists entirely of reserved bits that cannot be modified and always read 1.

Address H'FFFA4

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|-------|-------|-------|-------|-------|-------|------|------|
| | NDR15 | NDR14 | NDR13 | NDR12 | NDR11 | NDR10 | NDR9 | NDR8 |
| Initial value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

| | |
|--|--|
| Next data 15 to 12 | Next data 11 to 8 |
| These bits store the next output data for TPC output group 3 | These bits store the next output data for TPC output group 2 |

Address H'FFFA6

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — |
| Initial value | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Read/Write | — | — | — | — | — | — | — | — |

Reserved bits

Different Triggers for TPC Output Groups 2 and 3: If TPC output groups 2 and 3 are triggered by different compare match events, the address of the upper 4 bits of NDRB (group 3) is H'FFFA4 and the address of the lower 4 bits (group 2) is H'FFFA6. Bits 3 to 0 of address H'FFFA4 and bits 7 to 4 of address H'FFFA6 are reserved bits that cannot be modified and always read 1.

Address H'FFFA4

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|-------|-------|-------|-------|---|---|---|---|
| | NDR15 | NDR14 | NDR13 | NDR12 | — | — | — | — |
| Initial value | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| Read/Write | R/W | R/W | R/W | R/W | — | — | — | — |

Next data 15 to 12
Reserved bits

These bits store the next output data for TPC output group 3

Address H'FFFA6

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|---|---|---|---|-------|-------|------|------|
| | — | — | — | — | NDR11 | NDR10 | NDR9 | NDR8 |
| Initial value | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| Read/Write | — | — | — | — | R/W | R/W | R/W | R/W |

Reserved bits
Next data 11 to 8

These bits store the next output data for TPC output group 2

10.2.7 Next Data Enable Register A (NDERA)

NDERA is an 8-bit readable/writable register that enables or disables TPC output groups 1 and 0 (TP₇ to TP₀) on a bit-by-bit basis.

| | | | | | | | | |
|---------------|-------|-------|-------|-------|-------|-------|-------|-------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | NDER7 | NDER6 | NDER5 | NDER4 | NDER3 | NDER2 | NDER1 | NDER0 |
| Initial value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Next data enable 7 to 0

These bits enable or disable
TPC output groups 1 and 0

If a bit is enabled for TPC output by NDERA, then when the 16-bit timer compare match event selected in the TPC output control register (TPCR) occurs, the NDRA value is automatically transferred to the corresponding PADR bit, updating the output value. If TPC output is disabled, the bit value is not transferred from NDRA to PADR and the output value does not change.

NDERA is initialized to H'00 by a reset and in hardware standby mode. It is not initialized in software standby mode.

Bits 7 to 0—Next Data Enable 7 to 0 (NDER7 to NDER0): These bits enable or disable TPC output groups 1 and 0 (TP₇ to TP₀) on a bit-by-bit basis.

Bits 7 to 0

NDER7 to NDER0

Description

| | | |
|---|--|-----------------|
| 0 | TPC outputs TP ₇ to TP ₀ are disabled (NDR7 to NDR0 are not transferred to PA ₇ to PA ₀) | (Initial value) |
| 1 | TPC outputs TP ₇ to TP ₀ are enabled (NDR7 to NDR0 are transferred to PA ₇ to PA ₀) | |

10.2.8 Next Data Enable Register B (NDERB)

NDERB is an 8-bit readable/writable register that enables or disables TPC output groups 3 and 2 (TP₁₅ to TP₈) on a bit-by-bit basis.

| | | | | | | | | |
|---------------|--------|--------|--------|--------|--------|--------|-------|-------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | NDER15 | NDER14 | NDER13 | NDER12 | NDER11 | NDER10 | NDER9 | NDER8 |
| Initial value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Next data enable 15 to 8

These bits enable or disable
TPC output groups 3 and 2

If a bit is enabled for TPC output by NDERB, then when the 16-bit timer compare match event selected in the TPC output control register (TPCR) occurs, the NDRB value is automatically transferred to the corresponding PBDR bit, updating the output value. If TPC output is disabled, the bit value is not transferred from NDRB to PBDR and the output value does not change.

NDERB is initialized to H'00 by a reset and in hardware standby mode. It is not initialized in software standby mode.

Bits 7 to 0—Next Data Enable 15 to 8 (NDER15 to NDER8): These bits enable or disable TPC output groups 3 and 2 (TP₁₅ to TP₈) on a bit-by-bit basis.

Bits 7 to 0

NDER15 to NDER8

Description

| | | |
|---|--|-----------------|
| 0 | TPC outputs TP ₁₅ to TP ₈ are disabled (NDR15 to NDR8 are not transferred to PB ₇ to PB ₀) | (Initial value) |
| 1 | TPC outputs TP ₁₅ to TP ₈ are enabled (NDR15 to NDR8 are transferred to PB ₇ to PB ₀) | |

10.2.9 TPC Output Control Register (TPCR)

TPCR is an 8-bit readable/writable register that selects output trigger signals for TPC outputs on a group-by-group basis.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|--------|--------|--------|--------|--------|--------|--------|--------|
| | G3CMS1 | G3CMS0 | G2CMS1 | G2CMS0 | G1CMS1 | G1CMS0 | G0CMS1 | G0CMS0 |
| Initial value | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

| | | | |
|--|---|--|--|
| <p>Group 3 compare match select 1 and 0 These bits select the compare match event that triggers TPC output group 3 (TP₁₅ to TP₁₂)</p> | <p>Group 2 compare match select 1 and 0 These bits select the compare match event that triggers TPC output group 2 (TP₁₁ to TP₈)</p> | <p>Group 1 compare match select 1 and 0 These bits select the compare match event that triggers TPC output group 1 (TP₇ to TP₄)</p> | <p>Group 0 compare match select 1 and 0 These bits select the compare match event that triggers TPC output group 0 (TP₃ to TP₀)</p> |
|--|---|--|--|

TPCR is initialized to H'FF by a reset and in hardware standby mode. It is not initialized in software standby mode.

Bits 7 and 6—Group 3 Compare Match Select 1 and 0 (G3CMS1, G3CMS0): These bits select the compare match event that triggers TPC output group 3 (TP₁₅ to TP₁₂).

| Bit 7 G3CMS1 | Bit 6 G3CMS0 | Description |
|-----------------|-----------------|--|
| 0 | 0 | TPC output group 3 (TP ₁₅ to TP ₁₂) is triggered by compare match in 16-bit timer channel 0 |
| | 1 | TPC output group 3 (TP ₁₅ to TP ₁₂) is triggered by compare match in 16-bit timer channel 1 |
| 1 | 0 | TPC output group 3 (TP ₁₅ to TP ₁₂) is triggered by compare match in 16-bit timer channel 2 |
| | 1 | TPC output group 3 (TP ₁₅ to TP ₁₂) is triggered by compare match in 16-bit timer channel 2 (Initial value) |

Bits 5 and 4—Group 2 Compare Match Select 1 and 0 (G2CMS1, G2CMS0): These bits select the compare match event that triggers TPC output group 2 (TP₁₁ to TP₈).

| Bit 5 G2CMS1 | Bit 4 G2CMS0 | Description |
|-----------------|-----------------|---|
| 0 | 0 | TPC output group 2 (TP ₁₁ to TP ₈) is triggered by compare match in 16-bit timer channel 0 |
| | 1 | TPC output group 2 (TP ₁₁ to TP ₈) is triggered by compare match in 16-bit timer channel 1 |
| 1 | 0 | TPC output group 2 (TP ₁₁ to TP ₈) is triggered by compare match in 16-bit timer channel 2 |
| | 1 | TPC output group 2 (TP ₁₁ to TP ₈) is triggered by compare match in 16-bit timer channel 2 (Initial value) |

Bits 3 and 2—Group 1 Compare Match Select 1 and 0 (G1CMS1, G1CMS0): These bits select the compare match event that triggers TPC output group 1 (TP₇ to TP₄).

| Bit 3 G1CMS1 | Bit 2 G1CMS0 | Description |
|-----------------|-----------------|--|
| 0 | 0 | TPC output group 1 (TP ₇ to TP ₄) is triggered by compare match in 16-bit timer channel 0 |
| | 1 | TPC output group 1 (TP ₇ to TP ₄) is triggered by compare match in 16-bit timer channel 1 |
| 1 | 0 | TPC output group 1 (TP ₇ to TP ₄) is triggered by compare match in 16-bit timer channel 2 |
| | 1 | TPC output group 1 (TP ₇ to TP ₄) is triggered by compare match in 16-bit timer channel 2 (Initial value) |

Bits 1 and 0—Group 0 Compare Match Select 1 and 0 (G0CMS1, G0CMS0): These bits select the compare match event that triggers TPC output group 0 (TP₃ to TP₀).

| Bit 1 G0CMS1 | Bit 0 G0CMS0 | Description |
|-----------------|-----------------|--|
| 0 | 0 | TPC output group 0 (TP ₃ to TP ₀) is triggered by compare match in 16-bit timer channel 0 |
| | 1 | TPC output group 0 (TP ₃ to TP ₀) is triggered by compare match in 16-bit timer channel 1 |
| 1 | 0 | TPC output group 0 (TP ₃ to TP ₀) is triggered by compare match in 16-bit timer channel 2 |
| | 1 | TPC output group 0 (TP ₃ to TP ₀) is triggered by compare match in 16-bit timer channel 2 (Initial value) |

10.2.10 TPC Output Mode Register (TPMR)

TPMR is an 8-bit readable/writable register that selects normal or non-overlapping TPC output for each group.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|---|---|---|---|-------|-------|-------|-------|
| | — | — | — | — | G3NOV | G2NOV | G1NOV | G0NOV |
| Initial value | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| Read/Write | — | — | — | — | R/W | R/W | R/W | R/W |

Reserved bits

Group 3 non-overlap
Selects non-overlapping TPC output for group 3 (TP₁₅ to TP₁₂)

Group 2 non-overlap
Selects non-overlapping TPC output for group 2 (TP₁₁ to TP₈)

Group 1 non-overlap
Selects non-overlapping TPC output for group 1 (TP₇ to TP₄)

Group 0 non-overlap
Selects non-overlapping TPC output for group 0 (TP₃ to TP₀)

The output trigger period of a non-overlapping TPC output waveform is set in general register B (GRB) in the 16-bit timer channel selected for output triggering. The non-overlap margin is set in general register A (GRA). The output values change at compare match A and B.

For details see section 10.3.4, Non-Overlapping TPC Output.

TPMR is initialized to H'F0 by a reset and in hardware standby mode. It is not initialized in software standby mode.

Bits 7 to 4—Reserved: These bits cannot be modified and are always read as 1.

Bit 3—Group 3 Non-Overlap (G3NOV): Selects normal or non-overlapping TPC output for group 3 (TP₁₅ to TP₁₂).

| Bit 3 G3NOV | Description | |
|----------------|--|-----------------|
| 0 | Normal TPC output in group 3 (output values change at compare match A in the selected 16-bit timer channel) | (Initial value) |
| 1 | Non-overlapping TPC output in group 3 (independent 1 and 0 output at compare match A and B in the selected 16-bit timer channel) | |

Bit 2—Group 2 Non-Overlap (G2NOV): Selects normal or non-overlapping TPC output for group 2 (TP₁₁ to TP₈).

| Bit 2 G2NOV | Description | |
|----------------|--|-----------------|
| 0 | Normal TPC output in group 2 (output values change at compare match A in the selected 16-bit timer channel) | (Initial value) |
| 1 | Non-overlapping TPC output in group 2 (independent 1 and 0 output at compare match A and B in the selected 16-bit timer channel) | |

Bit 1—Group 1 Non-Overlap (G1NOV): Selects normal or non-overlapping TPC output for group 1 (TP₇ to TP₄).

| Bit 1 G1NOV | Description | |
|----------------|--|-----------------|
| 0 | Normal TPC output in group 1 (output values change at compare match A in the selected 16-bit timer channel) | (Initial value) |
| 1 | Non-overlapping TPC output in group 1 (independent 1 and 0 output at compare match A and B in the selected 16-bit timer channel) | |

Bit 0—Group 0 Non-Overlap (G0NOV): Selects normal or non-overlapping TPC output for group 0 (TP₃ to TP₀).

| Bit 0 G0NOV | Description | |
|----------------|--|-----------------|
| 0 | Normal TPC output in group 0 (output values change at compare match A in the selected 16-bit timer channel) | (Initial value) |
| 1 | Non-overlapping TPC output in group 0 (independent 1 and 0 output at compare match A and B in the selected 16-bit timer channel) | |

10.3 Operation

10.3.1 Overview

When corresponding bits in PADDR or PBDDR and NDERA or NDERB are set to 1, TPC output is enabled. The TPC output initially consists of the corresponding PADDR or PBDDR contents. When a compare-match event selected in TPCR occurs, the corresponding NDRA or NDRB bit contents are transferred to PADDR or PBDDR to update the output values.

Figure 10.2 illustrates the TPC output operation. Table 10.3 summarizes the TPC operating conditions.

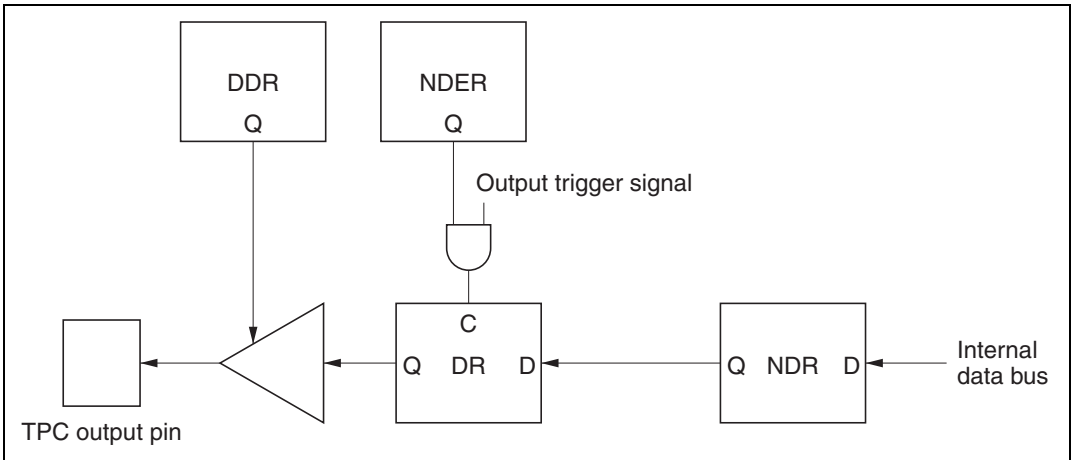


Figure 10.2 TPC Output Operation

Table 10.3 TPC Operating Conditions

| NDR | DDR | Pin Function |
|-----|-----|---|
| 0 | 0 | Generic input port |
| | 1 | Generic output port |
| 1 | 0 | Generic input port (but the DR bit is a read-only bit, and when compare match occurs, the NDR bit value is transferred to the DR bit) |
| | 1 | TPC pulse output |

Sequential output of up to 16-bit patterns is possible by writing new output data to NDRA and NDRB before the next compare match. For information on non-overlapping operation, see section 10.3.4, Non-Overlapping TPC Output.

10.3.2 Output Timing

If TPC output is enabled, NDRA/NDRB contents are transferred to PADR/PBDR and output when the selected compare match event occurs. Figure 10.3 shows the timing of these operations for the case of normal output in groups 2 and 3, triggered by compare match A.

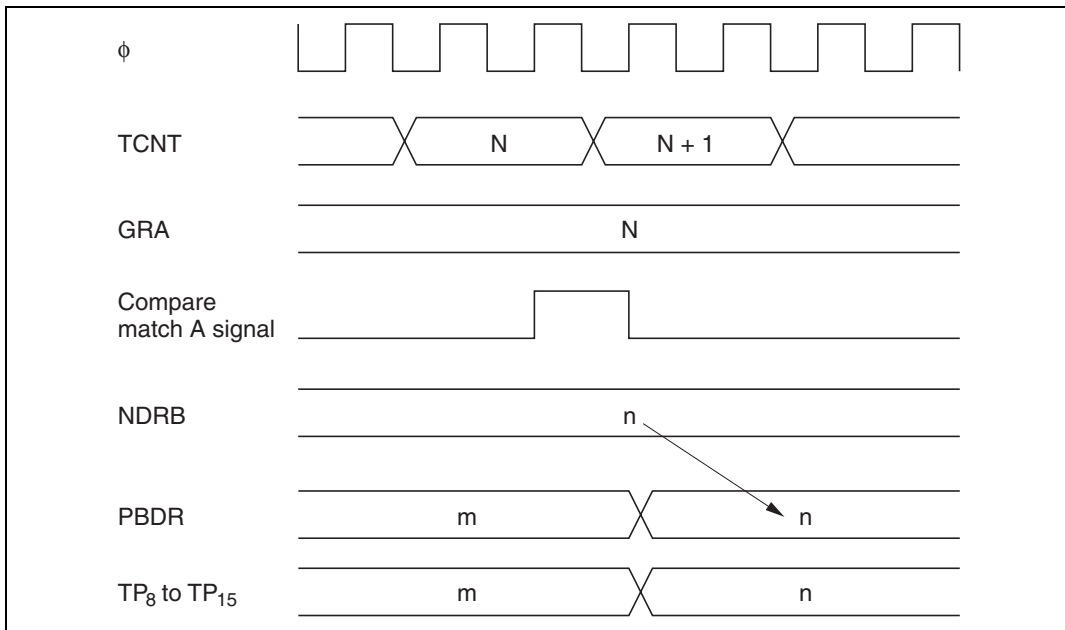


Figure 10.3 Timing of Transfer of Next Data Register Contents and Output (Example)

10.3.3 Normal TPC Output

Sample Setup Procedure for Normal TPC Output: Figure 10.4 shows a sample procedure for setting up normal TPC output.

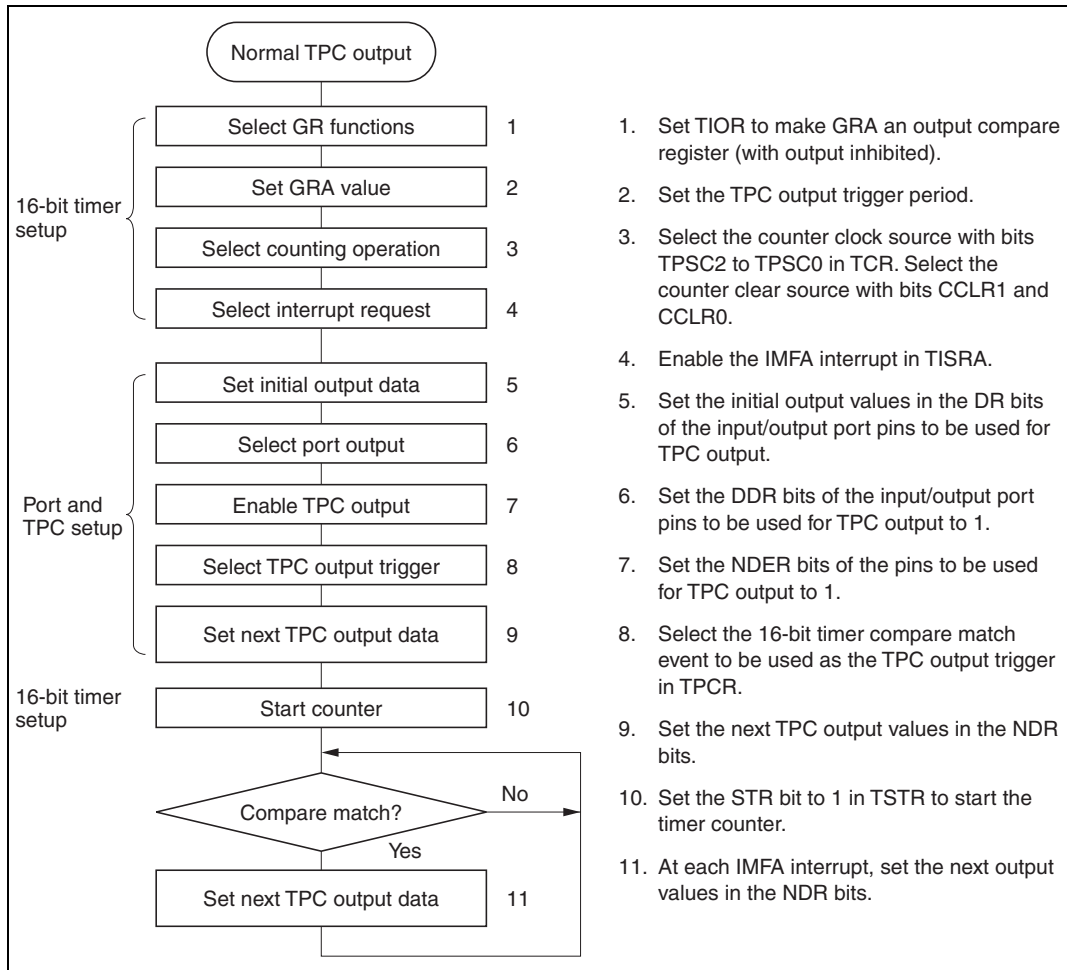


Figure 10.4 Setup Procedure for Normal TPC Output (Example)

Example of Normal TPC Output (Example of Five-Phase Pulse Output): Figure 10.5 shows an example in which the TPC is used for cyclic five-phase pulse output.

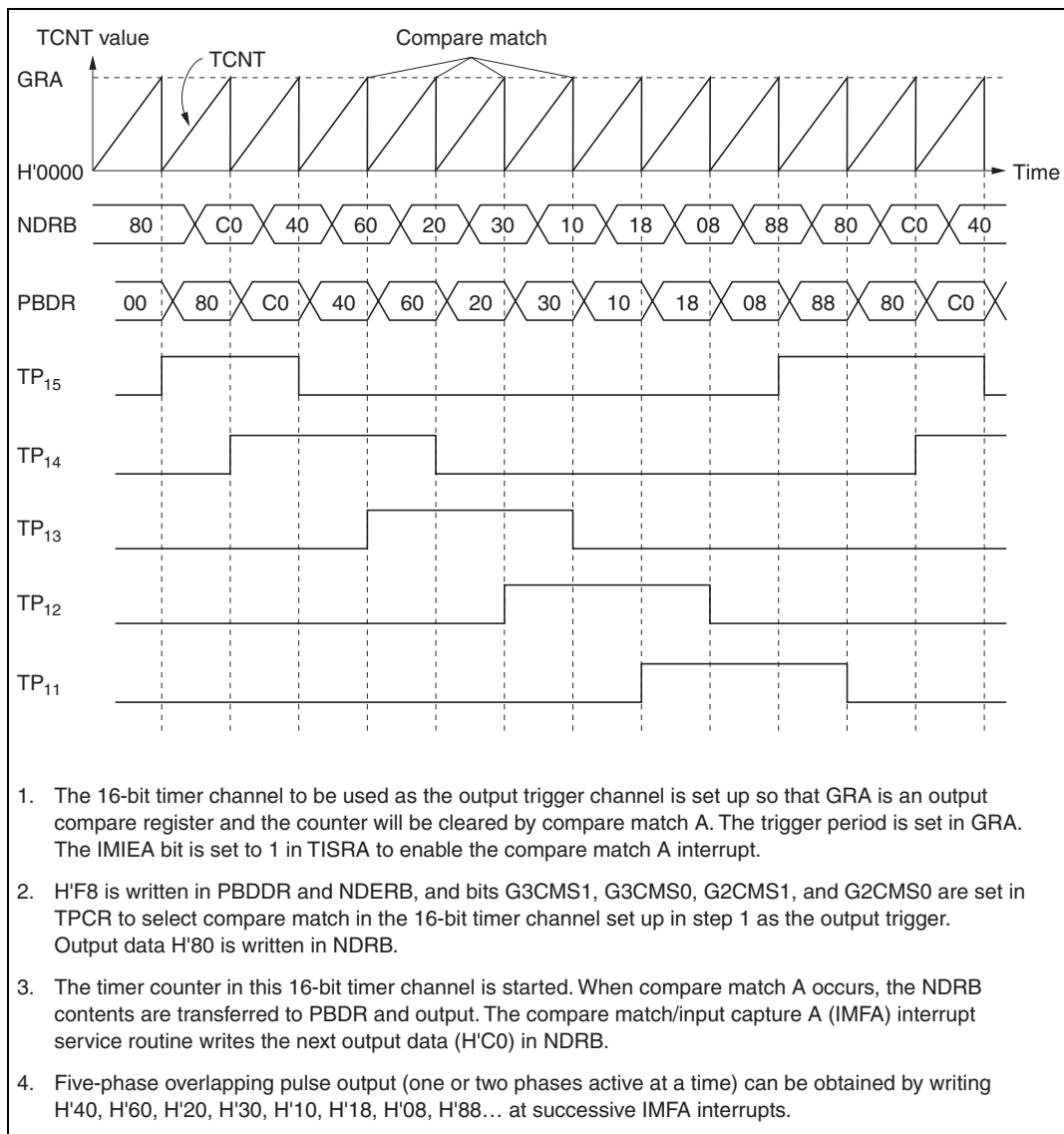


Figure 10.5 Normal TPC Output Example (Five-Phase Pulse Output)

10.3.4 Non-Overlapping TPC Output

Sample Setup Procedure for Non-Overlapping TPC Output: Figure 10.6 shows a sample procedure for setting up non-overlapping TPC output.

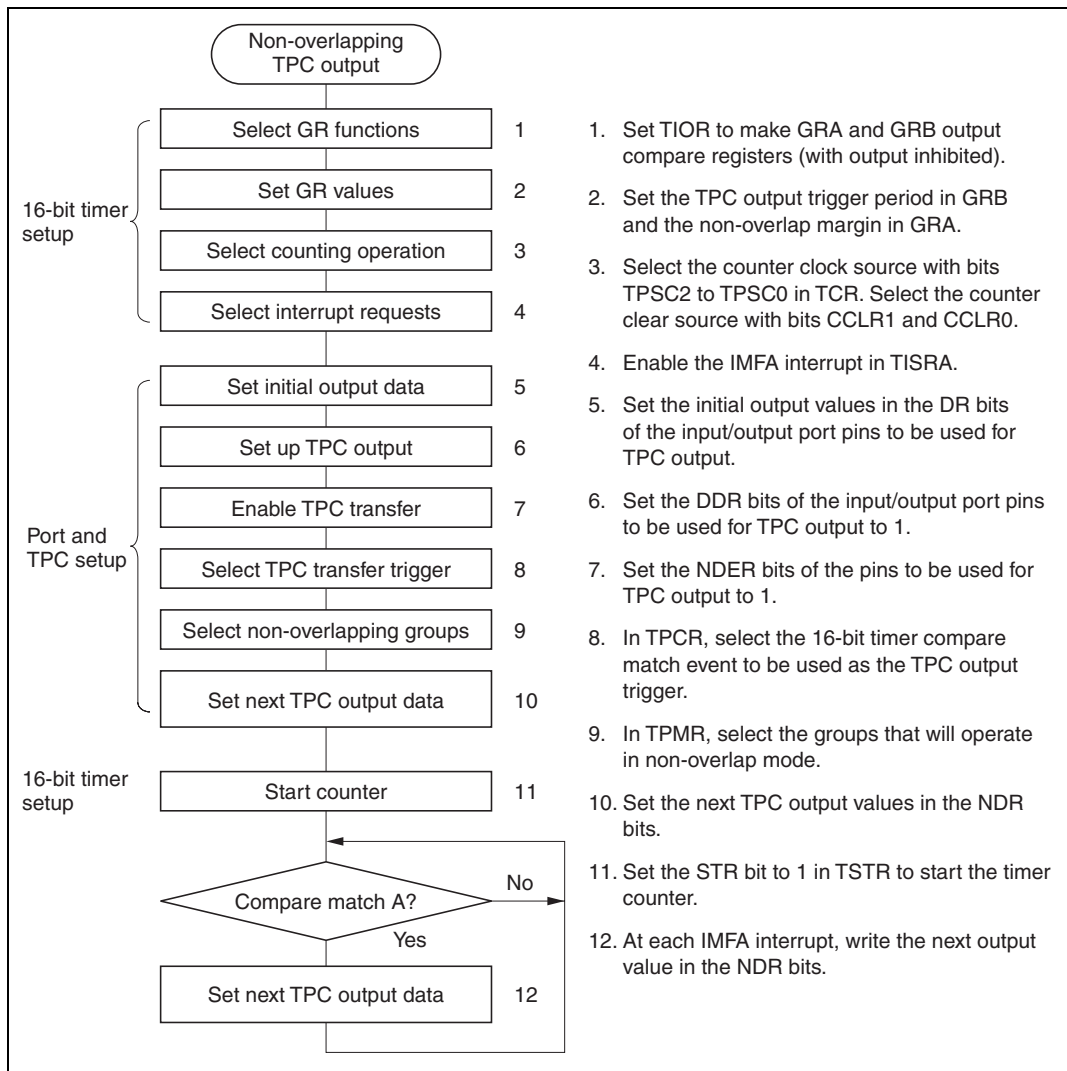


Figure 10.6 Setup Procedure for Non-Overlapping TPC Output (Example)

Example of Non-Overlapping TPC Output (Example of Four-Phase Complementary Non-Overlapping Output): Figure 10.7 shows an example of the use of TPC output for four-phase complementary non-overlapping pulse output.

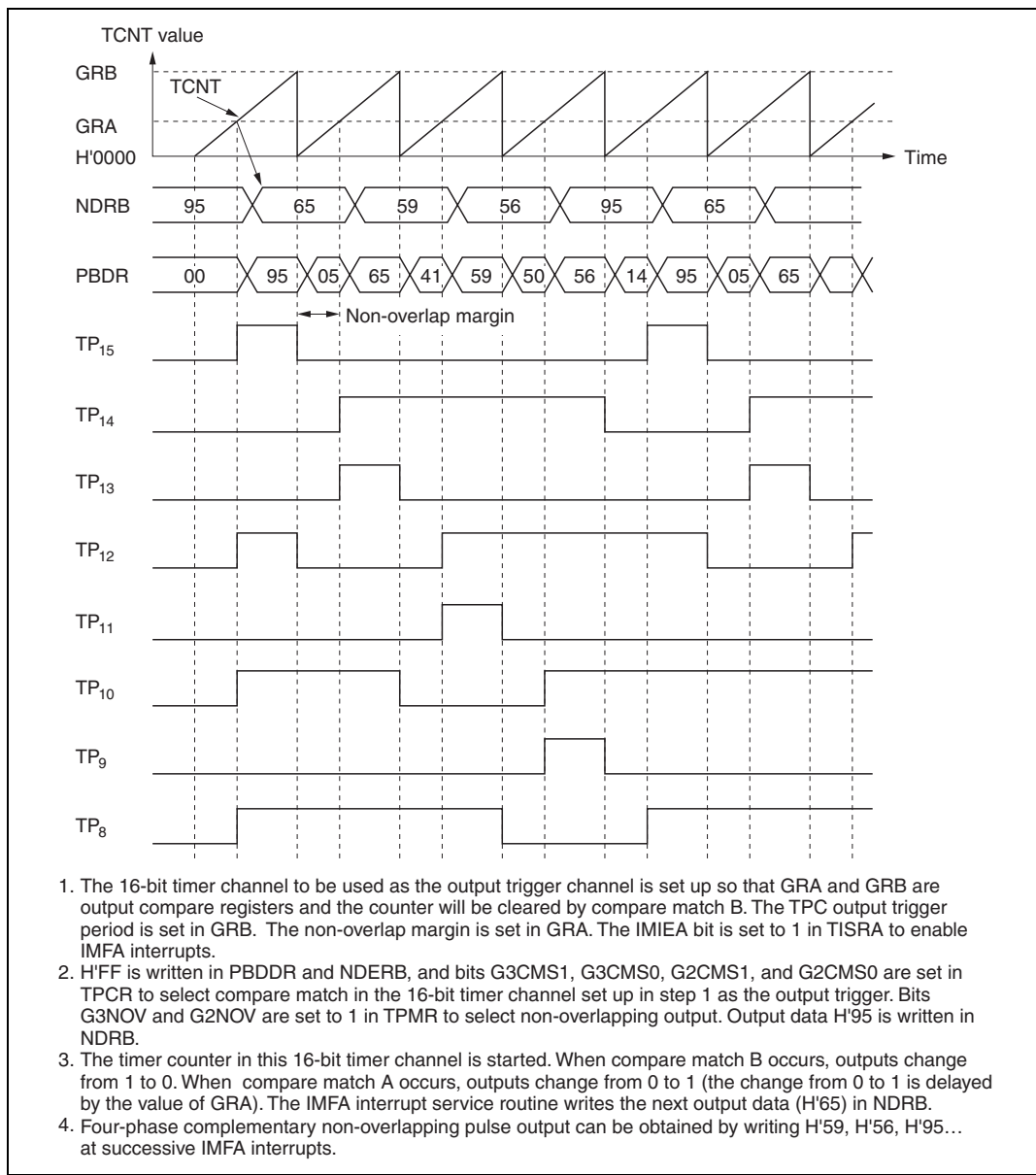


Figure 10.7 Non-Overlapping TPC Output Example (Four-Phase Complementary Non-Overlapping Pulse Output)

10.3.5 TPC Output Triggering by Input Capture

TPC output can be triggered by 16-bit timer input capture as well as by compare match. If GRA functions as an input capture register in the 16-bit timer channel selected in TPCR, TPC output will be triggered by the input capture signal. Figure 10.8 shows the timing.

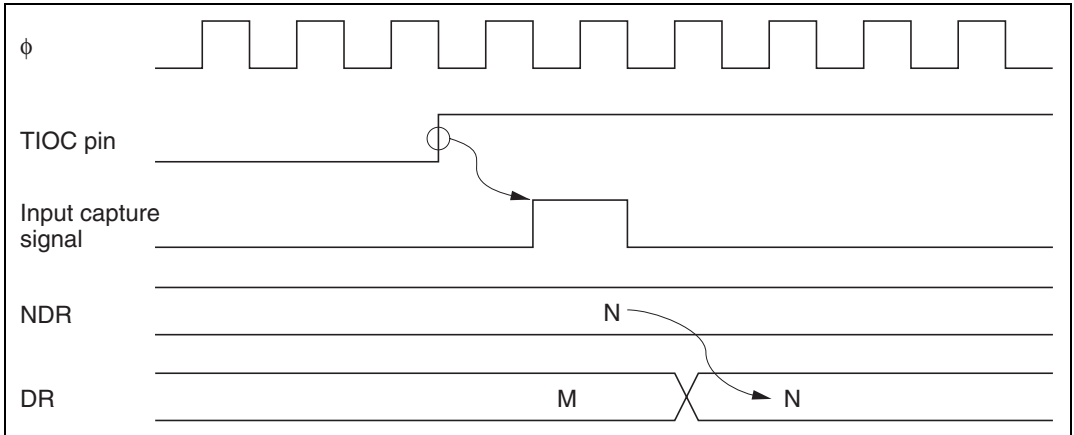


Figure 10.8 TPC Output Triggering by Input Capture (Example)

10.4 Usage Notes

10.4.1 Operation of TPC Output Pins

TP₀ to TP₁₅ are multiplexed with 16-bit timer, address bus, and other pin functions. When 16-bit timer, or address bus output is enabled, the corresponding pins cannot be used for TPC output. The data transfer from NDR bits to DR bits takes place, however, regardless of the usage of the pin.

Pin functions should be changed only under conditions in which the output trigger event will not occur.

10.4.2 Note on Non-Overlapping Output

During non-overlapping operation, the transfer of NDR bit values to DR bits takes place as follows.

1. NDR bits are always transferred to DR bits at compare match A.
2. At compare match B, NDR bits are transferred only if their value is 0. Bits are not transferred if their value is 1.

Figure 10.9 illustrates the non-overlapping TPC output operation.

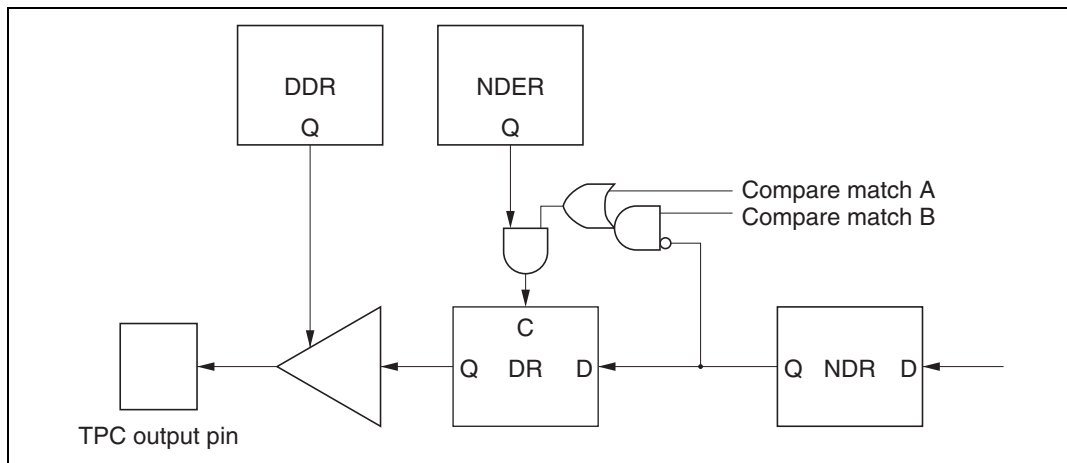


Figure 10.9 Non-Overlapping TPC Output

Therefore, 0 data can be transferred ahead of 1 data by making compare match B occur before compare match A. NDR contents should not be altered during the interval from compare match B to compare match A (the non-overlap margin).

This can be accomplished by having the IMFA interrupt service routine write the next data in NDR. The next data must be written before the next compare match B occurs.

Figure 10.10 shows the timing relationships.

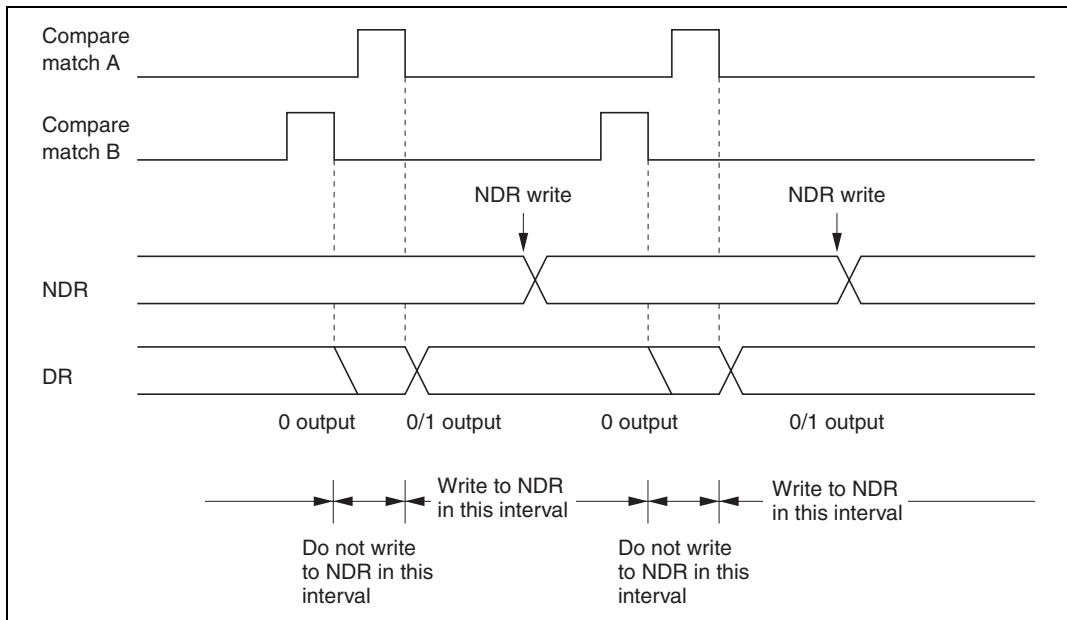


Figure 10.10 Non-Overlapping Operation and NDR Write Timing

Section 11 Watchdog Timer

11.1 Overview

The H8/3008 has an on-chip watchdog timer (WDT). The WDT has two selectable functions: it can operate as a watchdog timer to supervise system operation, or it can operate as an interval timer. As a watchdog timer, it generates a reset signal for the H8/3008 chip if a system crash allows the timer counter (TCNT) to overflow before being rewritten. In interval timer operation, an interval timer interrupt is requested at each TCNT overflow.

11.1.1 Features

WDT features are listed below.

- Selection of eight counter clock sources
 $\phi/2$, $\phi/32$, $\phi/64$, $\phi/128$, $\phi/256$, $\phi/512$, $\phi/2048$, or $\phi/4096$
- Interval timer option
- Timer counter overflow generates a reset signal or interrupt.

The reset signal is generated in watchdog timer operation. An interval timer interrupt is generated in interval timer operation.

- Watchdog timer reset signal resets the entire H8/3008 internally, and can also be output externally.

The reset signal generated by timer counter overflow during watchdog timer operation resets the entire H8/3008 internally. An external reset signal can be output from the $\overline{\text{RESO}}$ pin to reset other system devices simultaneously.

11.1.2 Block Diagram

Figure 11.1 shows a block diagram of the WDT.

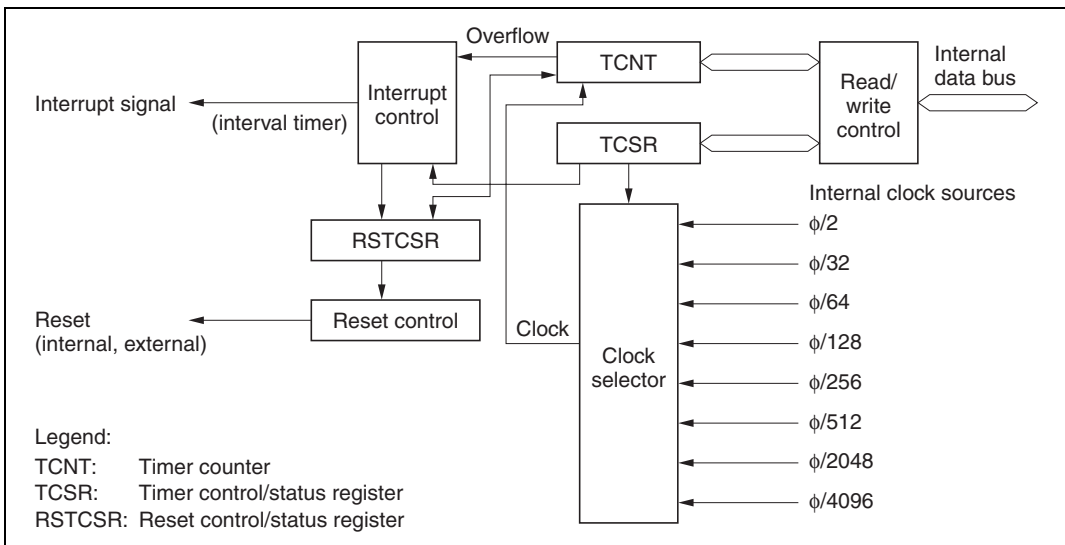


Figure 11.1 WDT Block Diagram

11.1.3 Pin Configuration

Table 11.1 describes the WDT output pin.

Table 11.1 WDT Pin

| Name | Abbreviation | I/O | Function |
|--------------|--------------------------|---------|--|
| Reset output | $\overline{\text{RES0}}$ | Output* | External output of the watchdog timer reset signal |

Note: * Open-drain output.

11.1.4 Register Configuration

Table 11.2 summarizes the WDT registers.

Table 11.2 WDT Registers

| Address* ¹ | | Name | Abbreviation | R/W | Initial Value |
|-----------------------|---------|-------------------------------|--------------|---------------------|---------------|
| Write* ² | Read | | | | |
| H'FFF8C | H'FFF8C | Timer control/status register | TCSR | R/(W)* ³ | H'18 |
| | H'FFF8D | Timer counter | TCNT | R/W | H'00 |
| H'FFF8E | H'FFF8F | Reset control/status register | RSTCSR | R/(W)* ³ | H'3F |

Notes: 1. Lower 20 bits of the address in advanced mode.

2. Write word data starting at this address.

3. Only 0 can be written in bit 7, to clear the flag.

11.2 Register Descriptions

11.2.1 Timer Counter (TCNT)

TCNT is an 8-bit readable and writable up-counter.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|-----|-----|-----|-----|-----|-----|-----|-----|
| Initial value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Note: The method for writing to TCNT is different from that for general registers to prevent inadvertent overwriting. For details see section 11.2.4, Notes on Register Access.

When the TME bit is set to 1 in TCSR, TCNT starts counting pulses generated from an internal clock source selected by bits CKS2 to CKS0 in TCSR. When the count overflows (changes from H'FF to H'00), the OVF bit is set to 1 in TCSR. TCNT is initialized to H'00 by a reset and when the TME bit is cleared to 0.

11.2.2 Timer Control/Status Register (TCSR)

TCSR is an 8-bit readable and writable register. Its functions include selecting the timer mode and clock source.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|--------|---------------------|-----|---|---|------|------|------|
| | OVF | WT/ \overline{IT} | TME | — | — | CKS2 | CKS1 | CKS0 |
| Initial value | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| Read/Write | R/(W)* | R/W | R/W | — | — | R/W | R/W | R/W |

Overflow flag
 Status flag indicating overflow

Timer mode select
 Selects the mode

Timer enable
 Selects whether TCNT runs or halts

Reserved bits

Clock select
 These bits select the TCNT clock source

Notes: The method for writing to TCSR is different from that for general registers to prevent inadvertent overwriting. For details see section 11.2.4, Notes on Register Access.

* Only 0 can be written, to clear the flag.

Bits 7 to 5 are initialized to 0 by a reset and in standby mode. Bits 2 to 0 are initialized to 0 by a reset. In software standby mode bits 2 to 0 are not initialized, but retain their previous values.

Bit 7—Overflow Flag (OVF): This status flag indicates that the timer counter has overflowed from H'FF to H'00.

| Bit 7 | Description |
|------------|--|
| OVF | |
| 0 | [Clearing condition] Cleared by reading OVF when OVF = 1, then writing 0 in OVF (Initial value) |
| 1 | [Setting condition] Set when TCNT changes from H'FF to H'00 |

Bit 6—Timer Mode Select (WT/IT $\bar{}$): Selects whether to use the WDT as a watchdog timer or interval timer. If used as an interval timer, the WDT generates an interval timer interrupt request when TCNT overflows. If used as a watchdog timer, the WDT generates a reset signal when TCNT overflows.

Bit 6

| WT/IT $\bar{}$ | Description | |
|----------------|--|-----------------|
| 0 | Interval timer: requests interval timer interrupts | (Initial value) |
| 1 | Watchdog timer: generates a reset signal | |

Bit 5—Timer Enable (TME): Selects whether TCNT runs or is halted. When WT/IT $\bar{}$ = 1, clear the software standby bit (SSBY) to 0 in SYSCR before setting TME. When setting SSBY to 1, TME should be cleared to 0.

Bit 5

| TME | Description | |
|-----|--|-----------------|
| 0 | TCNT is initialized to H'00 and halted | (Initial value) |
| 1 | TCNT is counting | |

Bits 4 and 3—Reserved: These bits cannot be modified and are always read as 1.

Bits 2 to 0—Clock Select 2 to 0 (CKS2 to CKS0): These bits select one of eight internal clock sources, obtained by prescaling the system clock (ϕ), for input to TCNT.

| Bit 2 CKS2 | Bit 1 CKS1 | Bit 0 CKS0 | Description | |
|---------------|---------------|---------------|-------------|-----------------|
| 0 | 0 | 0 | $\phi/2$ | (Initial value) |
| | | 1 | $\phi/32$ | |
| | 1 | 0 | $\phi/64$ | |
| | | 1 | $\phi/128$ | |
| 1 | 0 | 0 | $\phi/256$ | |
| | | 1 | $\phi/512$ | |
| | 1 | 0 | $\phi/2048$ | |
| | | 1 | $\phi/4096$ | |

11.2.3 Reset Control/Status Register (RSTCSR)

RSTCSR is an 8-bit readable and writable register that indicates when a reset signal has been generated by watchdog timer overflow, and controls external output of the reset signal.

| | | | | | | | | |
|---------------|--------|-------|---|---|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | WRST | RSTOE | — | — | — | — | — | — |
| Initial value | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| Read/Write | R/(W)* | R/W | — | — | — | — | — | — |

Reserved bits

Reset output enable
Enables or disables external output of the reset signal

Watchdog timer reset
Indicates that a reset signal has been generated

Notes: The method for writing to RSTCSR is different from that for general registers to prevent inadvertent overwriting. For details see section 11.2.4, Notes on Register Access.

* Only 0 can be written in bit 7, to clear the flag.

Bits 7 and 6 are initialized by input of a reset signal at the $\overline{\text{RES}}$ pin. They are not initialized by reset signals generated by watchdog timer overflow.

Bit 7—Watchdog Timer Reset (WRST): During watchdog timer operation, this bit indicates that TCNT has overflowed and generated a reset signal. This reset signal resets the entire H8/3008 chip internally. If bit RSTOE is set to 1, this reset signal is also output (low) at the $\overline{\text{RESO}}$ pin to initialize external system devices. Note that there is no $\overline{\text{RESO}}$ pin in the versions with on-chip flash memory.

Bit 7

WRST Description

| | | |
|---|--|--|
| 0 | [Clearing conditions] | |
| | <ul style="list-style-type: none"> Reset signal at $\overline{\text{RES}}$ pin. Read WRST when WRST = 1, then write 0 in WRST. (Initial value) | |
| 1 | [Setting condition] | |
| | Set when TCNT overflow generates a reset signal during watchdog timer operation | |

Bit 6—Reset Output Enable (RSTOE): Enables or disables external output at the $\overline{\text{RESO}}$ pin of the reset signal generated if TCNT overflows during watchdog timer operation. Note that there is no $\overline{\text{RESO}}$ pin in the versions with on-chip flash memory.

Bit 6**RSTOE Description**

| | | |
|---|---------------------------------------|-----------------|
| 0 | Reset signal is not output externally | (Initial value) |
| 1 | Reset signal is output externally | |

Bits 5 to 0—Reserved: These bits cannot be modified and are always read as 1.

11.2.4 Notes on Register Access

The watchdog timer's TCNT, TCSR, and RSTCSR registers differ from other registers in being more difficult to write. The procedures for writing and reading these registers are given below.

Writing to TCNT and TCSR: These registers must be written by a word transfer instruction. They cannot be written by byte instructions. Figure 11.2 shows the format of data written to TCNT and TCSR. TCNT and TCSR both have the same write address. The write data must be contained in the lower byte of the written word. The upper byte must contain H'5A (password for TCNT) or H'A5 (password for TCSR). This transfers the write data from the lower byte to TCNT or TCSR.

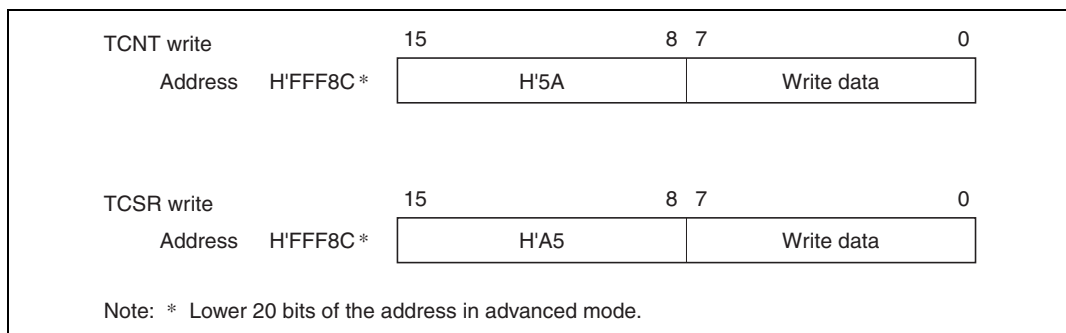


Figure 11.2 Format of Data Written to TCNT and TCSR

Writing to RSTCSR: RSTCSR must be written by a word transfer instruction. It cannot be written by byte transfer instructions. Figure 11.3 shows the format of data written to RSTCSR. To write 0 in the WRST bit, the write data must have H'A5 in the upper byte and H'00 in the lower byte. The data (H'00) in the lower byte is written to RSTCSR, clearing the WRST bit to 0. To write to the RSTOE bit, the upper byte must contain H'5A and the lower byte must contain the write data. Writing this word transfers a write data value into the RSTOE bit.

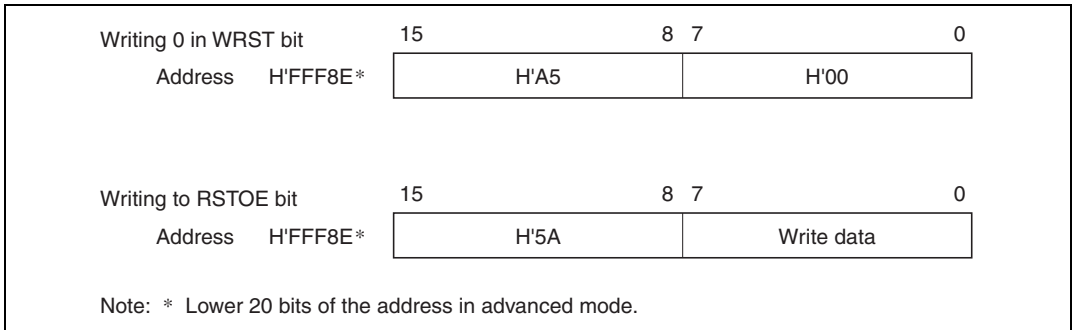


Figure 11.3 Format of Data Written to RSTCSR

Reading TCNT, TCSR, and RSTCSR: For reads of TCNT, TCSR, and RSTCSR, address H'FFF8C is assigned to TCSR, address H'FFF8D to TCNT, and address H'FFF8F to RSTCSR. These registers are therefore read like other registers. Byte transfer instructions can be used for reading. Table 11.3 lists the read addresses of TCNT, TCSR, and RSTCSR.

Table 11.3 Read Addresses of TCNT, TCSR, and RSTCSR

| Address* | Register |
|----------|----------|
| H'FFF8C | TCSR |
| H'FFF8D | TCNT |
| H'FFF8F | RSTCSR |

Note: * Lower 20 bits of the address in advanced mode.

11.3 Operation

Operations when the WDT is used as a watchdog timer and as an interval timer are described below.

11.3.1 Watchdog Timer Operation

Figure 11.4 illustrates watchdog timer operation. To use the WDT as a watchdog timer, set the WT/IT and TME bits to 1 in TCSR. Software must prevent TCNT overflow by rewriting the TCNT value (normally by writing H'00) before overflow occurs. If TCNT fails to be rewritten and overflows due to a system crash etc., the H8/3008 is internally reset for a duration of 518 states.

The watchdog reset signal can be externally output from the $\overline{\text{RESO}}$ pin to reset external system devices. The reset signal is output externally for 132 states. External output can be enabled or disabled by the RSTOE bit in RSTCSR.

A watchdog reset has the same vector as a reset generated by input at the $\overline{\text{RES}}$ pin. Software can distinguish a $\overline{\text{RES}}$ reset from a watchdog reset by checking the WRST bit in RSTCSR.

If a $\overline{\text{RES}}$ reset and a watchdog reset occur simultaneously, the $\overline{\text{RES}}$ reset takes priority.

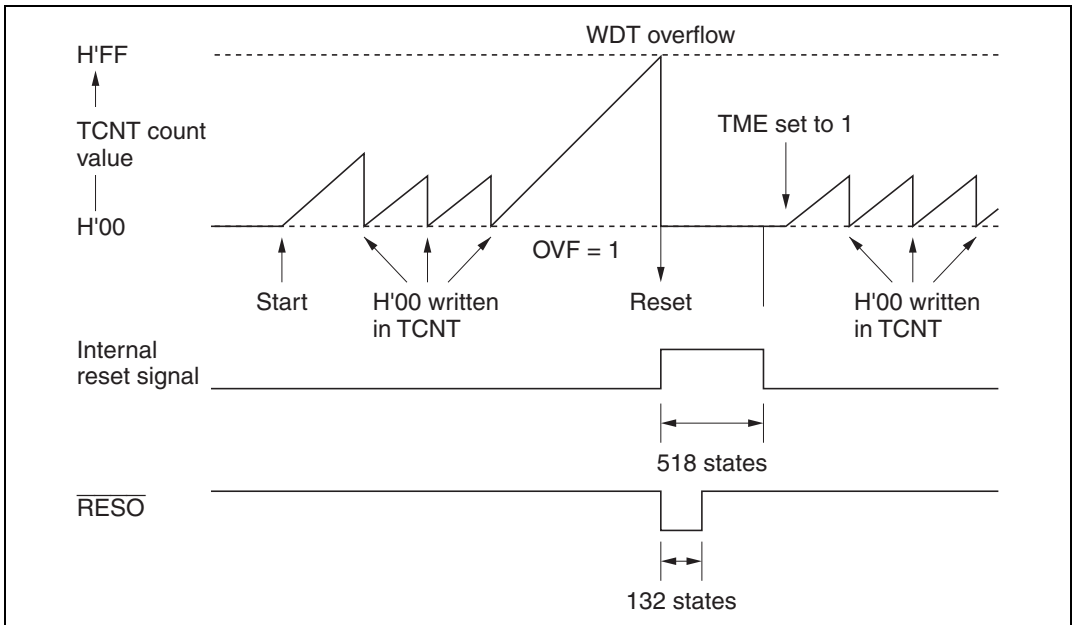


Figure 11.4 Operation in Watchdog Timer Mode

11.3.2 Interval Timer Operation

Figure 11.5 illustrates interval timer operation. To use the WDT as an interval timer, clear bit WT/\overline{IT} to 0 and set bit TME to 1 in $TCSR$. An interval timer interrupt request is generated at each $TCNT$ overflow. This function can be used to generate interval timer interrupts at regular intervals.

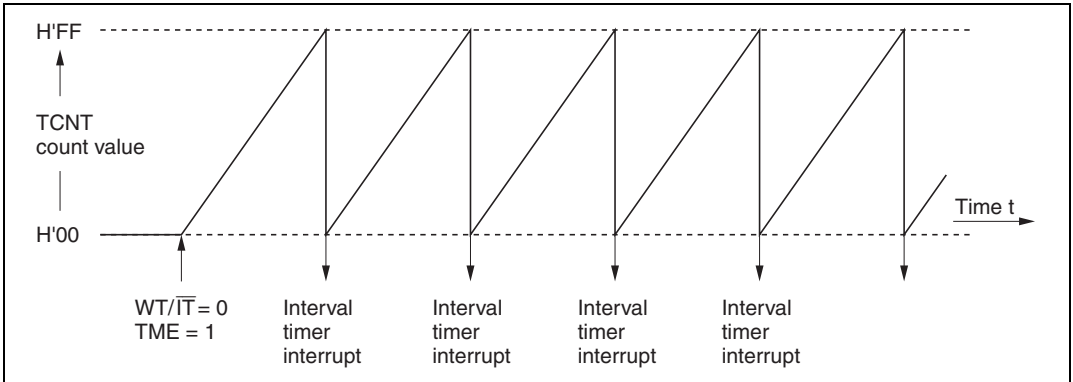


Figure 11.5 Interval Timer Operation

11.3.3 Timing of Setting of Overflow Flag (OVF)

Figure 11.6 shows the timing of setting of the OVF flag. The OVF flag is set to 1 when $TCNT$ overflows. At the same time, a reset signal is generated in watchdog timer operation, or an interval timer interrupt is generated in interval timer operation.

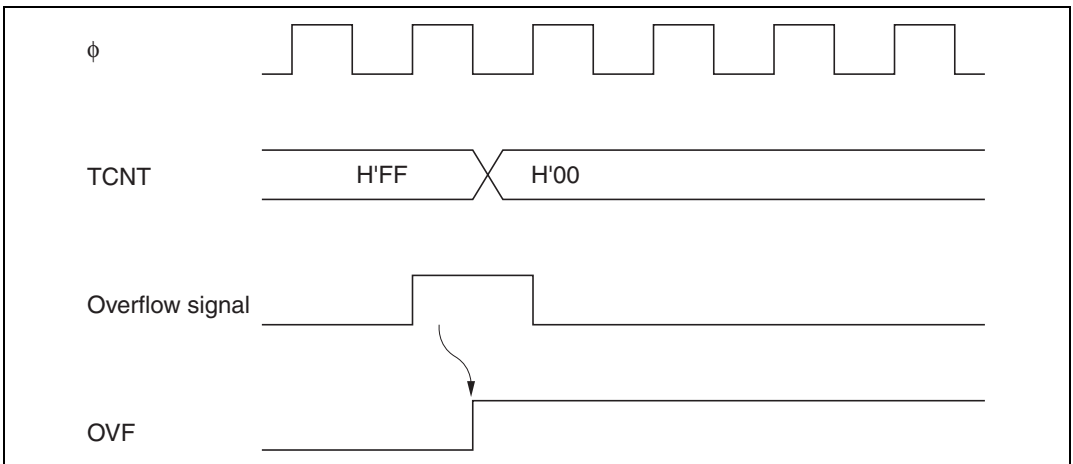


Figure 11.6 Timing of Setting of OVF

11.3.4 Timing of Setting of Watchdog Timer Reset Bit (WRST)

The WRST bit in RSTCSR is valid when bits $\overline{WT/IT}$ and \overline{TME} are both set to 1 in TCSR.

Figure 11.7 shows the timing of setting of WRST and the internal reset timing. The WRST bit is set to 1 when TCNT overflows and OVF is set to 1. At the same time an internal reset signal is generated for the entire H8/3008 chip. This internal reset signal clears OVF to 0, but the WRST bit remains set to 1. The reset routine must therefore clear the WRST bit.

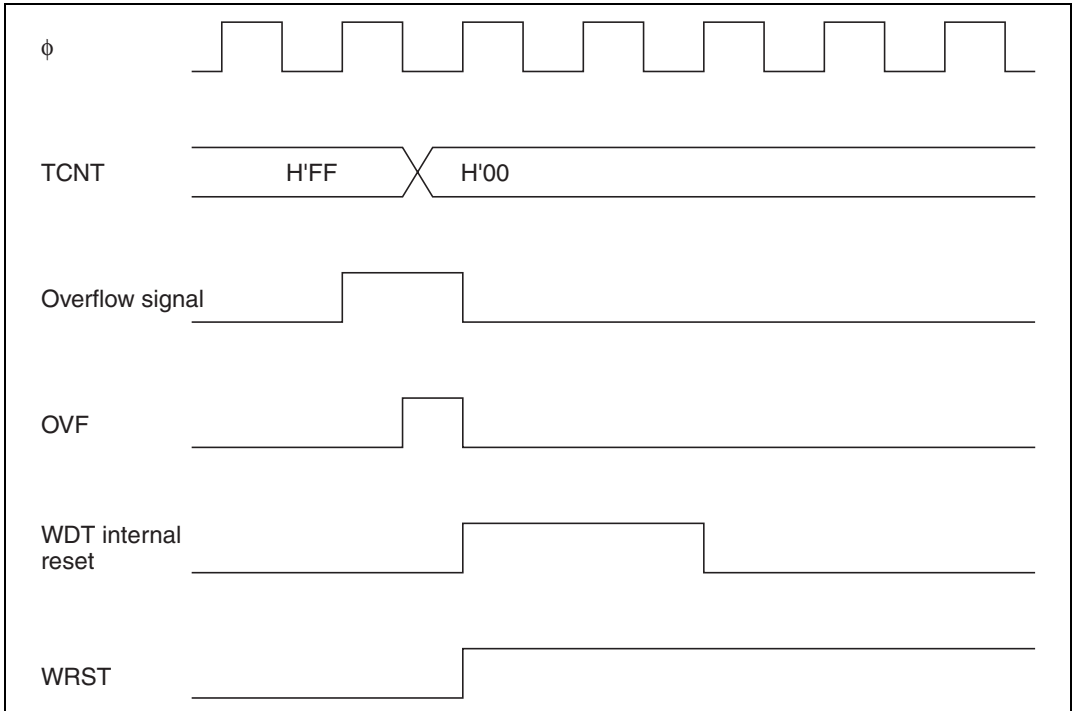


Figure 11.7 Timing of Setting of WRST Bit and Internal Reset

11.4 Interrupts

During interval timer operation, an overflow generates an interval timer interrupt (WOVI). The interval timer interrupt is requested whenever the OVF flag is set to 1 in TCSR.

11.5 Usage Notes

Contention between TCNT Write and Increment: If a timer counter clock pulse is generated during the T_3 state of a write cycle to TCNT, the write takes priority and the timer count is not incremented. See figure 11.8.

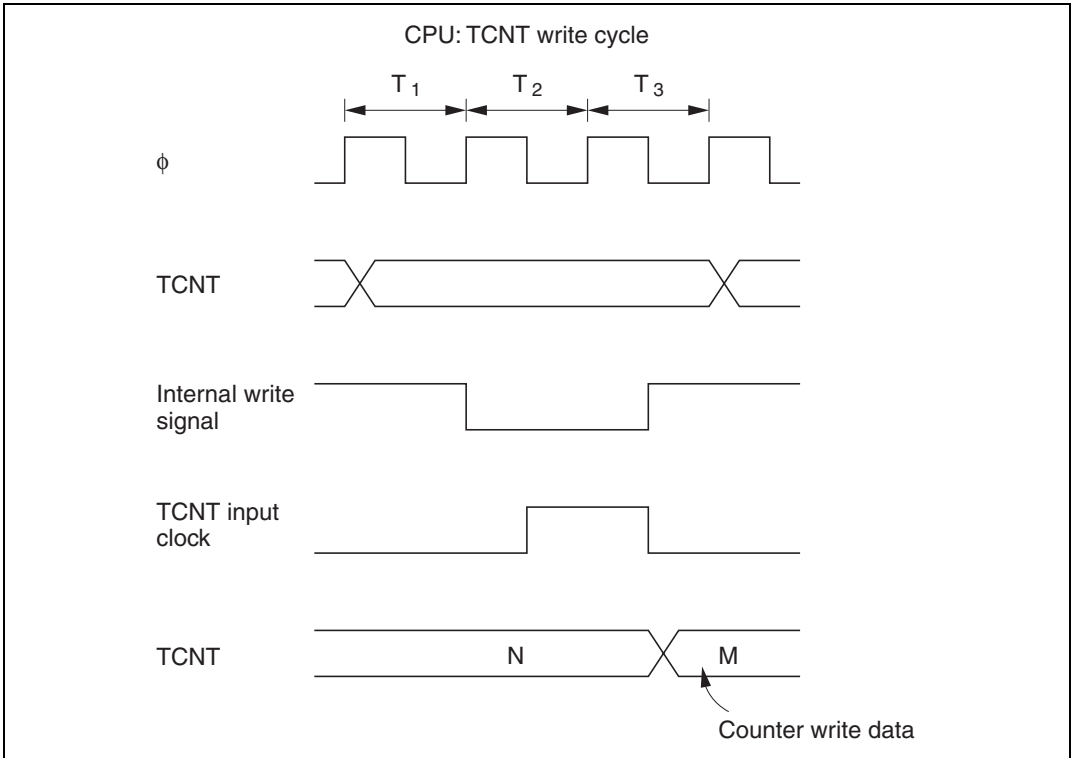


Figure 11.8 Contention between TCNT Write and Count up

Changing CKS2 to CKS0 Bit: Halt TCNT by clearing the TME bit to 0 in TCSR before changing the values of bits CKS2 to CKS0.

Section 12 Serial Communication Interface

12.1 Overview

The H8/3008 has a serial communication interface (SCI) with two independent channels. The two channels have identical functions. The SCI can communicate in both asynchronous and synchronous mode. It also has a multiprocessor communication function for serial communication among two or more processors.

When the SCI is not used, it can be halted to conserve power. Each SCI channel can be halted independently. For details, see section 18.6, Module Standby Function.

The SCI also has a smart card interface function conforming to the ISO/IEC 7816-3 (Identification Card) standard. This function supports serial communication with a smart card. Switching between the normal serial communication interface and the smart card interface is carried out by means of a register setting.

12.1.1 Features

SCI features are listed below.

- Selection of synchronous or asynchronous mode for serial communication

Asynchronous mode

Serial data communication is synchronized one character at a time. The SCI can communicate with a universal asynchronous receiver/transmitter (UART), asynchronous communication interface adapter (ACIA), or other chip that employs standard asynchronous communication. It can also communicate with two or more other processors using the multiprocessor communication function. There are twelve selectable serial data transfer formats.

- Data length: 7 or 8 bits
- Stop bit length: 1 or 2 bits
- Parity: even/odd/none
- Multiprocessor bit: 1 or 0
- Receive error detection: parity, overrun, and framing errors
- Break detection: by reading the RxD level directly when a framing error occurs

Synchronous mode

Serial data communication is synchronized with a clock signal. The SCI can communicate with other chips having a synchronous communication function.

There is a single serial data communication format.

- Data length: 8 bits
- Receive error detection: overrun errors
- Full-duplex communication

The transmitting and receiving sections are independent, so the SCI can transmit and receive simultaneously. The transmitting and receiving sections are both double-buffered, so serial data can be transmitted and received continuously.
- The following settings can be made for the serial data to be transferred:
 - LSB-first or MSB-first transfer
 - Inversion of data logic level
- Built-in baud rate generator with selectable bit rates
- Selectable transmit/receive clock sources: internal clock from baud rate generator, or external clock from the SCK pin
- Four types of interrupts

Transmit-data-empty, transmit-end, receive-data-full, and receive-error interrupts are requested independently.

Features of the smart card interface are listed below.

- Asynchronous communication
 - Data length: 8 bits
 - Parity bits generated and checked
 - Error signal output in receive mode (parity error)
 - Error signal detect and automatic data retransmit in transmit mode
 - Supports both direct convention and inverse convention
- Built-in baud rate generator with selectable bit rates
- Three types of interrupts

Transmit-data-empty, receive-data-full, and transmit/receive-error interrupts are requested independently.

12.1.2 Block Diagram

Figure 12.1 shows a block diagram of the SCI.

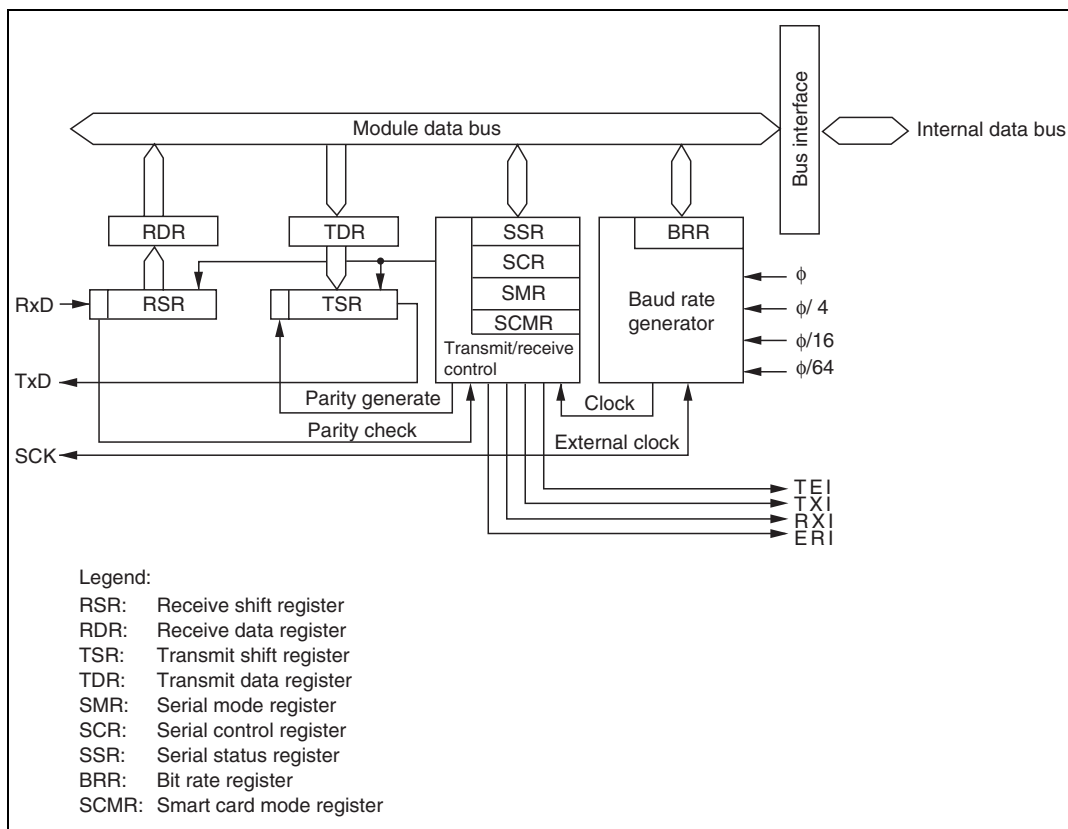


Figure 12.1 SCI Block Diagram

12.1.3 Pin Configuration

The SCI has serial pins for each channel as listed in table 12.1.

Table 12.1 SCI Pins

| Channel | Name | Abbreviation | I/O | Function |
|----------------|-------------------|---------------------|--------------|---------------------------------------|
| 0 | Serial clock pin | SCK ₀ | Input/output | SCI ₀ clock input/output |
| | Receive data pin | RxD ₀ | Input | SCI ₀ receive data input |
| | Transmit data pin | TxD ₀ | Output | SCI ₀ transmit data output |
| 1 | Serial clock pin | SCK ₁ | Input/output | SCI ₁ clock input/output |
| | Receive data pin | RxD ₁ | Input | SCI ₁ receive data input |
| | Transmit data pin | TxD ₁ | Output | SCI ₁ transmit data output |

12.1.4 Register Configuration

The SCI has internal registers as listed in table 12.2. These registers select asynchronous or synchronous mode, specify the data format and bit rate, control the transmitter and receiver sections, and specify switching between the serial communication interface and smart card interface.

Table 12.2 SCI Registers

| Channel | Address* ¹ | Name | Abbreviation | R/W | Initial Value |
|---------|-----------------------|--------------------------|--------------|---------------------|---------------|
| 0 | H'FFFB0 | Serial mode register | SMR | R/W | H'00 |
| | H'FFFB1 | Bit rate register | BRR | R/W | H'FF |
| | H'FFFB2 | Serial control register | SCR | R/W | H'00 |
| | H'FFFB3 | Transmit data register | TDR | R/W | H'FF |
| | H'FFFB4 | Serial status register | SSR | R/(W)* ² | H'84 |
| | H'FFFB5 | Receive data register | RDR | R | H'00 |
| | H'FFFB6 | Smart card mode register | SCMR | R/W | H'F2 |
| 1 | H'FFFB8 | Serial mode register | SMR | R/W | H'00 |
| | H'FFFB9 | Bit rate register | BRR | R/W | H'FF |
| | H'FFBFA | Serial control register | SCR | R/W | H'00 |
| | H'FFFBFB | Transmit data register | TDR | R/W | H'FF |
| | H'FFFBFC | Serial status register | SSR | R/(W)* ² | H'84 |
| | H'FFFBFD | Receive data register | RDR | R | H'00 |
| | H'FFFBFE | Smart card mode register | SCMR | R/W | H'F2 |

Notes: 1. Indicates the lower 20 bits of the address in advanced mode.

2. Only 0 can be written, to clear flags.

12.2 Register Descriptions

12.2.1 Receive Shift Register (RSR)

RSR is the register that receives serial data.

| | | | | | | | | |
|------------|---|---|---|---|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | | | | | | | | |
| Read/Write | — | — | — | — | — | — | — | — |

The SCI loads serial data input at the RxD pin into RSR in the order received, LSB (bit 0) first, thereby converting the data to parallel data. When one byte of data has been received, it is automatically transferred to RDR. The CPU cannot read or write RSR directly.

12.2.2 Receive Data Register (RDR)

RDR is the register that stores received serial data.

| | | | | | | | | |
|---------------|---|---|---|---|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | | | | | | | | |
| Initial value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Read/Write | R | R | R | R | R | R | R | R |

When the SCI has received one byte of serial data, it transfers the received data from RSR into RDR for storage, completing the receive operation. RSR is then ready to receive the next data. This double-buffering allows data to be received continuously.

RDR is a read-only register. Its contents cannot be modified by the CPU. RDR is initialized to H'00 by a reset and in standby mode.

12.2.3 Transmit Shift Register (TSR)

TSR is the register that transmits serial data.

| | | | | | | | | |
|------------|---|---|---|---|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | | | | | | | | |
| Read/Write | — | — | — | — | — | — | — | — |

The SCI loads transmit data from TDR to TSR, then transmits the data serially from the TxD pin, LSB (bit 0) first. After transmitting one data byte, the SCI automatically loads the next transmit data from TDR into TSR and starts transmitting it. If the TDRE flag is set to 1 in SSR, however, the SCI does not load the TDR contents into TSR. The CPU cannot read or write RSR directly.

12.2.4 Transmit Data Register (TDR)

TDR is an 8-bit register that stores data for serial transmission.

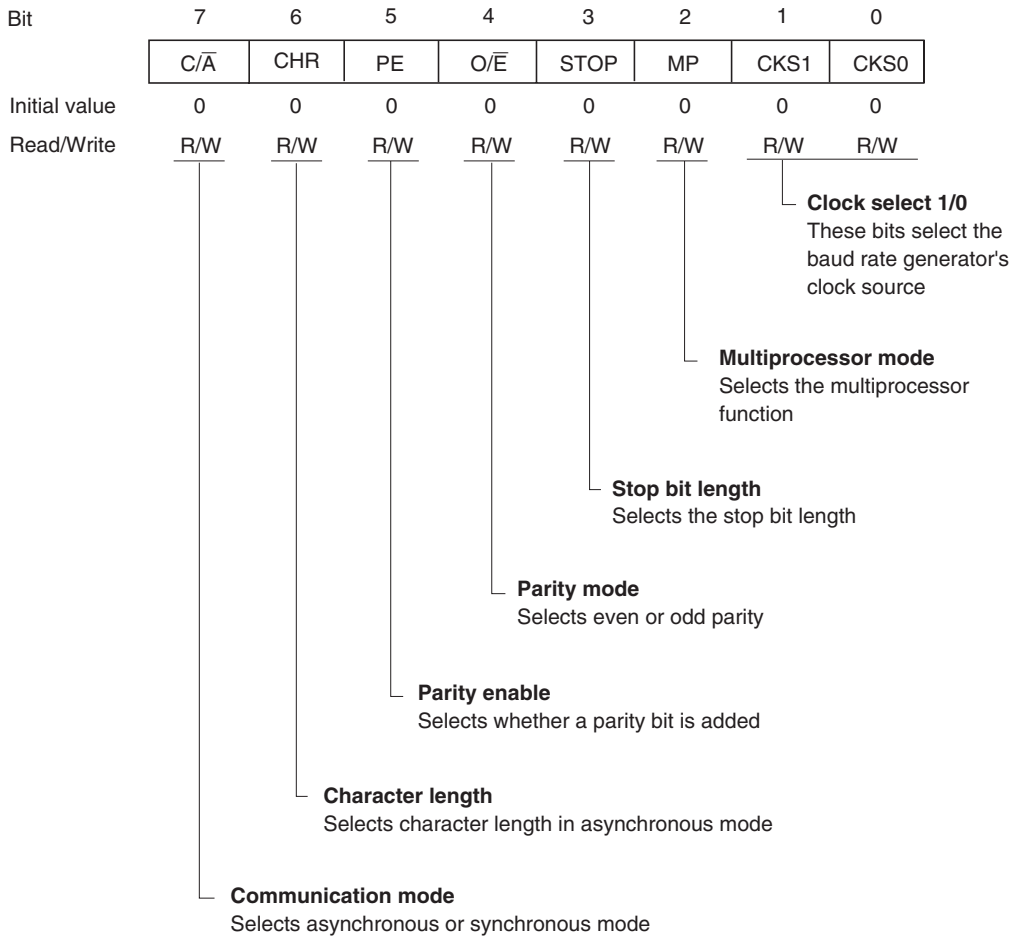
| | | | | | | | | |
|---------------|-----|-----|-----|-----|-----|-----|-----|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | | | | | | | | |
| Initial value | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

When the SCI detects that TSR is empty, it moves transmit data written in TDR from TDR into TSR and starts serial transmission. Continuous serial transmission is possible by writing the next transmit data in TDR during serial transmission from TSR.

The CPU can always read and write TDR. TDR is initialized to H'FF by a reset and in standby mode.

12.2.5 Serial Mode Register (SMR)

SMR is an 8-bit register that specifies the SCI's serial communication format and selects the clock source for the baud rate generator.



The CPU can always read and write SMR. SMR is initialized to H'00 by a reset and in standby mode.

Bit 7—Communication Mode (C/ \bar{A})/GSM Mode (GM): The function of this bit differs for the normal serial communication interface and for the smart card interface. Its function is switched with the SMIF bit in SCMR.

For Serial Communication Interface (SMIF Bit in SCMR Cleared to 0): Selects whether the SCI operates in asynchronous or synchronous mode.

| Bit 7 C/ \bar{A} | Description | |
|-----------------------|-------------------|-----------------|
| 0 | Asynchronous mode | (Initial value) |
| 1 | Synchronous mode | |

For Smart Card Interface (SMIF Bit in SCMR Set to 1): Selects GSM mode for the smart card interface.

| Bit 7 GM | Description | |
|-------------|---|-----------------|
| 0 | The TEND flag is set 12.5 etu after the start bit | (Initial value) |
| 1 | The TEND flag is set 11.0 etu after the start bit | |

Note: etu: Elementary time unit (time required to transmit one bit)

Bit 6—Character Length (CHR): Selects 7-bit or 8-bits data length in asynchronous mode. In synchronous mode, the data length is 8 bits regardless of the CHR setting,

| Bit 6 CHR | Description | |
|--------------|-------------|-----------------|
| 0 | 8-bit data | (Initial value) |
| 1 | 7-bit data* | |

Note: * When 7-bit data is selected, the MSB (bit 7) of TDR is not transmitted.

Bit 5—Parity Enable (PE): In asynchronous mode, this bit enables or disables the addition of a parity bit to transmit data, and the checking of the parity bit in receive data. In synchronous mode, the parity bit is neither added nor checked, regardless of the PE bit setting.

| Bit 5 PE | Description | |
|-------------|---------------------------------|-----------------|
| 0 | Parity bit not added or checked | (Initial value) |
| 1 | Parity bit added and checked* | |

Note: * When PE bit is set to 1, an even or odd parity bit is added to transmit data according to the even or odd parity mode selection by the O/ \bar{E} bit, and the parity bit in receive data is checked to see that it matches the even or odd mode selected by the O/ \bar{E} bit.

Bit 4—Parity Mode (O/ \bar{E}): Specifies whether even parity or odd parity is used for parity addition and checking. The O/ \bar{E} bit setting is only valid when the PE bit is set to 1, enabling parity bit addition and checking, in asynchronous mode. The O/ \bar{E} bit setting is ignored in synchronous mode, or when parity addition and checking is disabled in asynchronous mode.

| Bit 4 O/ \bar{E} | Description | |
|-----------------------|---------------------------|-----------------|
| 0 | Even parity* ¹ | (Initial value) |
| 1 | Odd parity* ² | |

- Notes:
1. When even parity is selected, the parity bit added to transmit data makes an even number of 1s in the transmitted character and parity bit combined. Receive data must have an even number of 1s in the received character and parity bit combined.
 2. When odd parity is selected, the parity bit added to transmit data makes an odd number of 1s in the transmitted character and parity bit combined. Receive data must have an odd number of 1s in the received character and parity bit combined.

Bit 3—Stop Bit Length (STOP): Selects one or two stop bits in asynchronous mode. This setting is used only in asynchronous mode. In synchronous mode no stop bit is added, so the STOP bit setting is ignored.

| Bit 3 STOP | Description | |
|---------------|---------------------------|-----------------|
| 0 | 1 stop bit* ¹ | (Initial value) |
| 1 | 2 stop bits* ² | |

- Notes:
1. One stop bit (with value 1) is added to the end of each transmitted character.
 2. Two stop bits (with value 1) are added to the end of each transmitted character.

In receiving, only the first stop bit is checked, regardless of the STOP bit setting. If the second stop bit is 1, it is treated as a stop bit. If the second stop bit is 0, it is treated as the start bit of the next incoming character.

Bit 2—Multiprocessor Mode (MP): Selects a multiprocessor format. When a multiprocessor format is selected, parity settings made by the PE and O/\bar{E} bits are ignored. The MP bit setting is valid only in asynchronous mode. It is ignored in synchronous mode.

For further information on the multiprocessor communication function, see section 12.3.3, Multiprocessor Communication.

| Bit 2 MP | Description | |
|-------------|----------------------------------|-----------------|
| 0 | Multiprocessor function disabled | (Initial value) |
| 1 | Multiprocessor format selected | |

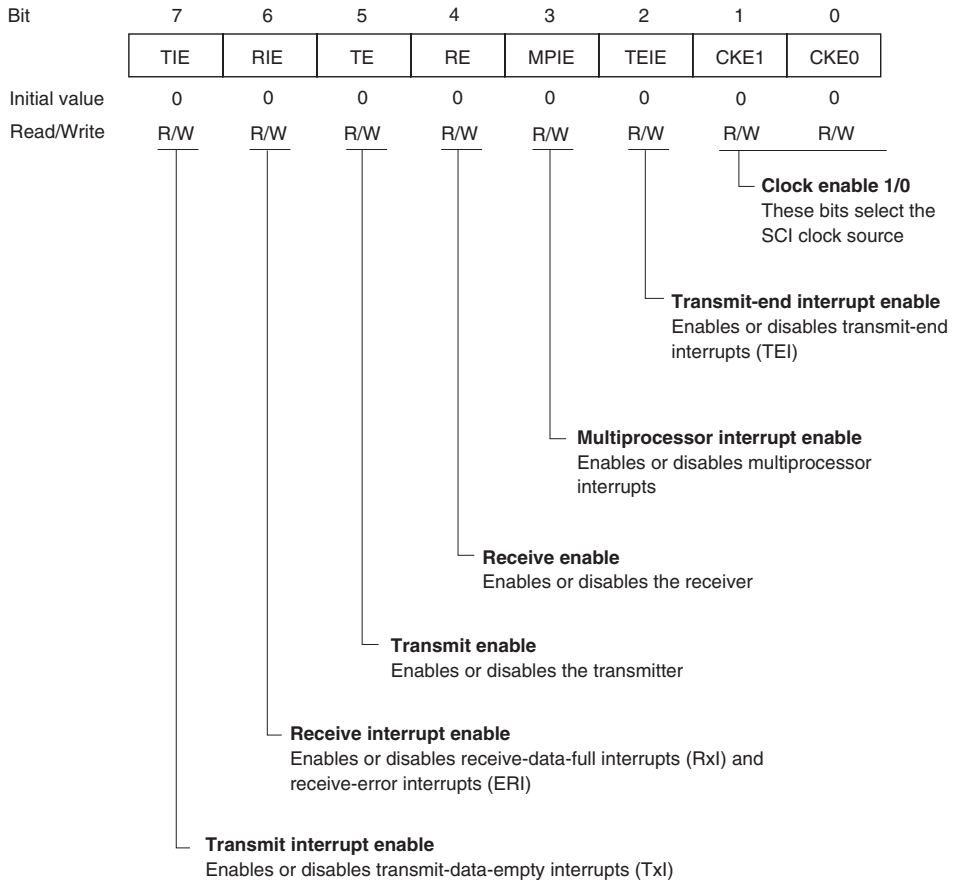
Bits 1 and 0—Clock Select 1 and 0 (CKS1, CKS0): These bits select the clock source for the on-chip baud rate generator. Four clock sources can be selected by the CKS1 and CKS0 bits: ϕ , $\phi/4$, $\phi/16$, and $\phi/64$.

For the relationship between the clock source, bit rate register setting, and baud rate, see section 12.2.8, Bit Rate Register (BRR).

| Bit 1 CKS1 | Bit 0 CKS0 | Description | |
|---------------|---------------|-------------|-----------------|
| 0 | 0 | ϕ | (Initial value) |
| 0 | 1 | $\phi/4$ | |
| 1 | 0 | $\phi/16$ | |
| 1 | 1 | $\phi/64$ | |

12.2.6 Serial Control Register (SCR)

SCR register enables or disables the SCI transmitter and receiver, enables or disables serial clock output in asynchronous mode, enables or disables interrupts, and selects the transmit/receive clock source.



The CPU can always read and write SCR. SCR is initialized to H'00 by a reset and in standby mode.

Bit 7—Transmit Interrupt Enable (TIE): Enables or disables the transmit-data-empty interrupt (TXI) requested when the TDRE flag in SSR is set to 1 due to transfer of serial transmit data from TDR to TSR.

| Bit 7 TIE | Description |
|--------------|--|
| 0 | Transmit-data-empty interrupt request (TXI) is disabled* (Initial value) |
| 1 | Transmit-data-empty interrupt request (TXI) is enabled |

Note: * TXI interrupt requests can be cleared by reading the value 1 from the TDRE flag, then clearing it to 0; or by clearing the TIE bit to 0.

Bit 6—Receive Interrupt Enable (RIE): Enables or disables the receive-data-full interrupt (RXI) requested when the RDRF flag in SSR is set to 1 due to transfer of serial receive data from RSR to RDR; also enables or disables the receive-error interrupt (ERI).

| Bit 6 RIE | Description |
|--------------|--|
| 0 | Receive-data-full (RXI) and receive-error (ERI) interrupt requests are disabled* (Initial value) |
| 1 | Receive-data-full (RXI) and receive-error (ERI) interrupt requests are enabled |

Note: * RXI and ERI interrupt requests can be cleared by reading the value 1 from the RDRF, FER, PER, or ORER flag, then clearing the flag to 0; or by clearing the RIE bit to 0.

Bit 5—Transmit Enable (TE): Enables or disables the start of SCI serial transmitting operations.

| Bit 5 TE | Description |
|-------------|---|
| 0 | Transmitting disabled* ¹ (Initial value) |
| 1 | Transmitting enabled* ² |

Notes: 1. The TDRE flag is fixed at 1 in SSR.
2. In the enabled state, serial transmission starts when the TDRE flag in SSR is cleared to 0 after writing of transmit data into TDR. Select the transmit format in SMR before setting the TE bit to 1.

Bit 4—Receive Enable (RE): Enables or disables the start of SCI serial receiving operations.

Bit 4

| RE | Description |
|----|--|
| 0 | Receiving disabled* ¹ (Initial value) |
| 1 | Receiving enabled* ² |

- Notes:
1. Clearing the RE bit to 0 does not affect the RDRF, FER, PER, and ORER flags. These flags retain their previous values.
 2. In the enabled state, serial receiving starts when a start bit is detected in asynchronous mode, or serial clock input is detected in synchronous mode. Select the receive format in SMR before setting the RE bit to 1.

Bit 3—Multiprocessor Interrupt Enable (MPIE): Enables or disables multiprocessor interrupts. The MPIE bit setting is valid only in asynchronous mode, and only if the MP bit is set to 1 in SMR. The MPIE bit setting is ignored in synchronous mode or when the MP bit is cleared to 0.

Bit 3

| MPIE | Description |
|------|--|
| 0 | Multiprocessor interrupts are disabled (normal receive operation) (Initial value) [Clearing conditions] <ul style="list-style-type: none"> • The MPIE bit is cleared to 0 • MPB = 1 in received data |
| 1 | Multiprocessor interrupts are enabled* Receive-data-full interrupts (RXI), receive-error interrupts (ERI), and setting of the RDRF, FER, and ORER status flags in SSR are disabled until data with the multiprocessor bit set to 1 is received. |

- Note: * The SCI does not transfer receive data from RSR to RDR, does not detect receive errors, and does not set the RDRF, FER, and ORER flags in SSR. When it receives data in which MPB = 1, the SCI sets the MPB bit to 1 in SSR, automatically clears the MPIE bit to 0, enables RXI and ERI interrupts (if the TIE and RIE bits in SCR are set to 1), and allows the FER and ORER flags to be set.

Bit 2—Transmit-End interrupt Enable (TEIE): Enables or disables the transmit-end interrupt (TEI) requested if TDR does not contain valid transmit data when the MSB is transmitted.

| Bit 2 TEIE | Description |
|---------------|---|
| 0 | Transmit-end interrupt requests (TEI) are disabled* (Initial value) |
| 1 | Transmit-end interrupt requests (TEI) are enabled* |

Note: * TEI interrupt requests can be cleared by reading the value 1 from the TDRE flag in SSR, then clearing the TDRE flag to 0, thereby also clearing the TEND flag to 0; or by clearing the TEIE bit to 0.

Bits 1 and 0—Clock Enable 1 and 0 (CKE1, CKE0): The function of these bits differs for the normal serial communication interface and for the smart card interface. Their function is switched with the SMIF bit in SCMR.

For serial communication interface (SMIF bit in SCMR cleared to 0): These bits select the SCI clock source and enable or disable clock output from the SCK pin. Depending on the settings of CKE1 and CKE0, the SCK pin can be used for generic input/output, serial clock output, or serial clock input.

The CKE0 setting is valid only in asynchronous mode, and only when the SCI is internally clocked (CKE1 = 0). The CKE0 setting is ignored in synchronous mode, or when an external clock source is selected (CKE1 = 1). Select the SCI operating mode in SMR before setting the CKE1 and CKE0 bits. For further details on selection of the SCI clock source, see table 12.9 in section 12.3, Operation.

| Bit 1 CKE1 | Bit 0 CKE0 | Description |
|---------------|---------------|--|
| 0 | 0 | Asynchronous mode Internal clock, SCK pin available for generic input/output* ¹ |
| | | Synchronous mode Internal clock, SCK pin used for serial clock output* ¹ |
| 0 | 1 | Asynchronous mode Internal clock, SCK pin used for clock output* ² |
| | | Synchronous mode Internal clock, SCK pin used for serial clock output |
| 1 | 0 | Asynchronous mode External clock, SCK pin used for clock input* ³ |
| | | Synchronous mode External clock, SCK pin used for serial clock input |
| 1 | 1 | Asynchronous mode External clock, SCK pin used for clock input* ³ |
| | | Synchronous mode External clock, SCK pin used for serial clock input |

- Notes: 1. Initial value
 2. The output clock frequency is the same as the bit rate.
 3. The input clock frequency is 16 times the bit rate.

For smart card interface (SMIF bit in SCMR set to 1): These bits, together with the GM bit in SMR, determine whether the SCK pin is used for generic input/output or as the serial clock output pin.

| SMR GM | Bit 1 CKE1 | Bit 0 CKE0 | Description | |
|-------------------|-----------------------|-----------------------|--|-----------------|
| 0 | 0 | 0 | SCK pin available for generic input/output | (Initial value) |
| 0 | 0 | 1 | SCK pin used for clock output | |
| 1 | 0 | 0 | SCK pin output fixed low | |
| 1 | 0 | 1 | SCK pin used for clock output | |
| 1 | 1 | 0 | SCK pin output fixed high | |
| 1 | 1 | 1 | SCK pin used for clock output | |

12.2.7 Serial Status Register (SSR)

SSR is an 8-bit register containing multiprocessor bit values, and status flags that indicate the operating status of the SCI.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|---------------------|---------------------|---------------------|---------------------|---------------------|------|-----|------|
| | TDRE | RDRF | ORER | FER/ERS | PER | TEND | MPB | MPBT |
| Initial value | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| Read/Write | R/(W)* ¹ | R/(W)* ¹ | R/(W)* ¹ | R/(W)* ¹ | R/(W)* ¹ | R | R | R/W |

Multiprocessor bit transfer
 Value of multiprocessor bit to be transmitted

Multiprocessor bit
 Stores the received multiprocessor bit value

Transmit end*²
 Status flag indicating end of transmission

Parity error
 Status flag indicating detection of a receive parity error

Framing error (FER)/Error signal status (ERS)*²
 Status flag indicating detection of a receive framing error, or flag indicating detection of an error signal

Overrun error
 Status flag indicating detection of a receive overrun error

Receive data register full
 Status flag indicating that data has been received and stored in RDR

Transmit data register empty
 Status flag indicating that transmit data has been transferred from TDR into TSR and new data can be written in TDR

Notes: 1. Only 0 can be written, to clear the flag.

2. Function differs between the normal serial communication interface and the smart card interface.

The CPU can always read and write SSR, but cannot write 1 in the TDRE, RDRF, ORER, PER, and FER flags. These flags can be cleared to 0 only if they have first been read while set to 1. The TEND and MPB flags are read-only bits that cannot be written.

SSR is initialized to H'84 by a reset and in standby mode.

Bit 7—Transmit Data Register Empty (TDRE): Indicates that the SCI has loaded transmit data from TDR into TSR and the next serial data can be written in TDR.

Bit 7

| TDRE | Description |
|------|---|
| 0 | TDR contains valid transmit data [Clearing condition] Read TDRE when TDRE = 1, then write 0 in TDRE |
| 1 | TDR does not contain valid transmit data (Initial value) [Setting conditions] <ul style="list-style-type: none">• The chip is reset or enters standby mode• The TE bit in SCR is cleared to 0• TDR contents are loaded into TSR, so new data can be written in TDR |

Bit 6—Receive Data Register Full (RDRF): Indicates that RDR contains new receive data.

Bit 6

| RDRF | Description |
|------|---|
| 0 | RDR does not contain new receive data (Initial value) [Clearing conditions] <ul style="list-style-type: none">• The chip is reset or enters standby mode• Read RDRF when RDRF = 1, then write 0 in RDRF |
| 1 | RDR contains new receive data [Setting condition] Serial data is received normally and transferred from RSR to RDR |

Note: The RDR contents and the RDRF flag are not affected by detection of receive errors or by clearing of the RE bit to 0 in SCR. They retain their previous values. If the RDRF flag is still set to 1 when reception of the next data ends, an overrun error will occur and the receive data will be lost.

Bit 5—Overrun Error (ORER): Indicates that data reception ended abnormally due to an overrun error.

| Bit 5 ORER | Description |
|---------------|---|
| 0 | Receiving is in progress or has ended normally* ¹ (Initial value) [Clearing conditions] <ul style="list-style-type: none"> The chip is reset or enters standby mode Read ORER when ORER = 1, then write 0 in ORER |
| 1 | A receive overrun error occurred* ² [Setting condition] Reception of the next serial data ends when RDRF = 1 |

- Notes:
1. Clearing the RE bit to 0 in SCR does not affect the ORER flag, which retains its previous value.
 2. RDR continues to hold the receive data prior to the overrun error, so subsequent receive data is lost. Serial receiving cannot continue while the ORER flag is set to 1. In synchronous mode, serial transmitting is also disabled.

Bit 4—Framing Error (FER)/Error Signal Status (ERS): The function of this bit differs for the normal serial communication interface and for the smart card interface. Its function is switched with the SMIF bit in SCMR.

For serial communication interface (SMIF bit in SCMR cleared to 0): Indicates that data reception ended abnormally due to a framing error in asynchronous mode.

| Bit 4 FER | Description |
|--------------|--|
| 0 | Receiving is in progress or has ended normally* ¹ (Initial value) [Clearing conditions] <ul style="list-style-type: none"> The chip is reset or enters standby mode Read FER when FER = 1, then write 0 in FER |
| 1 | A receive framing error occurred [Setting condition] The stop bit at the end of the receive data is checked for a value of 1, and is found to be 0.* ² |

- Notes:
1. Clearing the RE bit to 0 in SCR does not affect the FER flag, which retains its previous value.
 2. When the stop bit length is 2 bits, only the first bit is checked for a value of 1. The second stop bit is not checked. When a framing error occurs the SCI transfers the receive data into RDR but does not set the RDRF flag. Serial receiving cannot continue while the FER flag is set to 1. In synchronous mode, serial transmitting is also disabled.

For Smart Card Interface (SMIF Bit in SCMR Set to 1): Indicates the status of the error signal sent back from the receiving side during transmission. Framing errors are not detected in smart card interface mode.

| Bit 4 ERS | Description | |
|--------------|---|-----------------|
| 0 | Normal reception, no error signal* [Clearing conditions] | (Initial value) |
| | <ul style="list-style-type: none"> The chip is reset or enters standby mode Read ERS when ERS = 1, then write 0 in ERS | |
| 1 | An error signal has been sent from the receiving side indicating detection of a parity error [Setting condition] The error signal is low when sampled | |

Note: * Clearing the TE bit to 0 in SCR does not affect the ERS flag, which retains its previous value.

Bit 3—Parity Error (PER): Indicates that reception of data with parity added ended abnormally due to a parity error in asynchronous mode.

| Bit 3 PER | Description | |
|--------------|--|-----------------|
| 0 | Receiving is in progress or has ended normally* ¹ [Clearing conditions] | (Initial value) |
| | <ul style="list-style-type: none"> The chip is reset or enters standby mode Read PER when PER = 1, then write 0 in PER | |
| 1 | A receive parity error occurred* ² [Setting condition] The number of 1s in receive data, including the parity bit, does not match the even or odd parity setting of O/ \bar{E} in SMR | |

Notes: 1. Clearing the RE bit to 0 in SCR does not affect the PER flag, which retains its previous value.
2. When a parity error occurs the SCI transfers the receive data into RDR but does not set the RDRF flag. Serial receiving cannot continue while the PER flag is set to 1. In synchronous mode, serial transmitting is also disabled.

Bit 2—Transmit End (TEND): The function of this bit differs for the normal serial communication interface and for the smart card interface. Its function is switched with the SMIF bit in SCMR.

For Serial Communication Interface (SMIF Bit in SCMR Cleared to 0): Indicates that when the last bit of a serial character was transmitted TDR did not contain valid transmit data, so transmission has ended. The TEND flag is a read-only bit and cannot be written.

| Bit 2 TEND | Description |
|---------------|---|
| 0 | Transmission is in progress [Clearing condition] Read TDRE when TDRE = 1, then write 0 in TDRE |
| 1 | End of transmission (Initial value) [Setting conditions] <ul style="list-style-type: none"> • The chip is reset or enters standby mode • The TE bit in SCR is cleared to 0 • TDRE is 1 when the last bit of a 1-byte serial transmit character is transmitted |

For Smart Card Interface (SMIF Bit in SCMR Set to 1): Indicates that when the last bit of a serial character was transmitted TDR did not contain valid transmit data, so transmission has ended. The TEND flag is a read-only bit and cannot be written.

| Bit 2 TEND | Description |
|---------------|--|
| 0 | Transmission is in progress [Clearing condition] Read TDRE when TDRE = 1, then write 0 in TDRE |
| 1 | End of transmission (Initial value) [Setting conditions] <ul style="list-style-type: none"> • The chip is reset or enters standby mode • The TE bit is cleared to 0 in SCR and the FER/ERS bit is also cleared to 0 • TDRE is 1 and FER/ERS is 0 (normal transmission) 2.5 etu (when GM = 0) or 1.0 etu (when GM = 1) after a 1-byte serial character is transmitted |

Note: etu: Elementary time unit (time required to transmit one bit)

Bit 1—Multiprocessor bit (MPB): Stores the value of the multiprocessor bit in the receive data when a multiprocessor format is used in asynchronous mode. MPB is a read-only bit, and cannot be written.

| Bit 1 MPB | Description | |
|--------------|--|-----------------|
| 0 | Multiprocessor bit value in receive data is 0* | (Initial value) |
| 1 | Multiprocessor bit value in receive data is 1 | |

Note: * If the RE bit in SCR is cleared to 0 when a multiprocessor format is selected, MPB retains its previous value.

Bit 0—Multiprocessor Bit Transfer (MPBT): Stores the value of the multiprocessor bit added to transmit data when a multiprocessor format is selected for transmitting in asynchronous mode.

The MPBT bit setting is ignored in synchronous mode, when a multiprocessor format is not selected, or when the SCI cannot transmit.

| Bit 0 MPBT | Description | |
|---------------|--|-----------------|
| 0 | Multiprocessor bit value in transmit data is 0 | (Initial value) |
| 1 | Multiprocessor bit value in transmit data is 1 | |

12.2.8 Bit Rate Register (BRR)

BRR is an 8-bit register that sets the serial transmit/receive bit rate in accordance with the baud rate generator operating clock selected by bits CKS0 and CKS1 in SMR.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|-----|-----|-----|-----|-----|-----|-----|-----|
| | | | | | | | | |
| Initial value | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

BRR can be read or written to by the CPU at all times.

BRR is initialized to H'FF by a reset and in standby mode.

As baud rate generator control is performed independently for each channel, different values can be set for each channel.

Table 12.3 shows examples of BRR settings in asynchronous mode. Table 12.4 shows examples of BRR settings in synchronous mode.

Table 12.3 Examples of Bit Rates and BRR Settings in Asynchronous Mode

| Bit Rate (bit/s) | ϕ (MHz) | | | | | | | | | | | |
|---------------------|--------------|-----|-----------|----------|-----|-----------|--------|-----|-----------|---|-----|-----------|
| | 2 | | | 2.097152 | | | 2.4576 | | | 3 | | |
| | n | N | Error (%) | n | N | Error (%) | n | N | Error (%) | n | N | Error (%) |
| 110 | 1 | 141 | 0.03 | 1 | 148 | -0.04 | 1 | 174 | -0.26 | 1 | 212 | 0.03 |
| 150 | 1 | 103 | 0.16 | 1 | 108 | 0.21 | 1 | 127 | 0.00 | 1 | 155 | 0.16 |
| 300 | 0 | 207 | 0.16 | 0 | 217 | 0.21 | 0 | 255 | 0.00 | 1 | 77 | 0.16 |
| 600 | 0 | 103 | 0.16 | 0 | 108 | 0.21 | 0 | 127 | 0.00 | 0 | 155 | 0.16 |
| 1200 | 0 | 51 | 0.16 | 0 | 54 | -0.70 | 0 | 63 | 0.00 | 0 | 77 | 0.16 |
| 2400 | 0 | 25 | 0.16 | 0 | 26 | 1.14 | 0 | 31 | 0.00 | 0 | 38 | 0.16 |
| 4800 | 0 | 12 | 0.16 | 0 | 13 | -2.48 | 0 | 15 | 0.00 | 0 | 19 | -2.34 |
| 9600 | 0 | 6 | -6.99 | 0 | 6 | -2.48 | 0 | 7 | 0.00 | 0 | 9 | -2.34 |
| 19200 | 0 | 2 | 8.51 | 0 | 2 | 13.78 | 0 | 3 | 0.00 | 0 | 4 | -2.34 |
| 31250 | 0 | 1 | 0.00 | 0 | 1 | 4.86 | 0 | 1 | 22.88 | 0 | 2 | 0.00 |
| 38400 | 0 | 1 | -18.62 | 0 | 1 | -14.67 | 0 | 1 | 0.00 | — | — | — |

| Bit Rate (bit/s) | ϕ (MHz) | | | | | | | | | | | |
|---------------------|--------------|-----|-----------|---|-----|-----------|--------|-----|-----------|---|-----|-----------|
| | 3.6864 | | | 4 | | | 4.9152 | | | 5 | | |
| | n | N | Error (%) | n | N | Error (%) | n | N | Error (%) | n | N | Error (%) |
| 110 | 2 | 64 | 0.70 | 2 | 70 | 0.03 | 2 | 86 | 0.31 | 2 | 88 | -0.25 |
| 150 | 1 | 191 | 0.00 | 1 | 207 | 0.16 | 1 | 255 | 0.00 | 2 | 64 | 0.16 |
| 300 | 1 | 95 | 0.00 | 1 | 103 | 0.16 | 1 | 127 | 0.00 | 1 | 129 | 0.16 |
| 600 | 0 | 191 | 0.00 | 0 | 207 | 0.16 | 0 | 255 | 0.00 | 1 | 64 | 0.16 |
| 1200 | 0 | 95 | 0.00 | 0 | 103 | 0.16 | 0 | 127 | 0.00 | 0 | 129 | 0.16 |
| 2400 | 0 | 47 | 0.00 | 0 | 51 | 0.16 | 0 | 63 | 0.00 | 0 | 64 | 0.16 |
| 4800 | 0 | 23 | 0.00 | 0 | 25 | 0.16 | 0 | 31 | 0.00 | 0 | 32 | -1.36 |
| 9600 | 0 | 11 | 0.00 | 0 | 12 | 0.16 | 0 | 15 | 0.00 | 0 | 15 | 1.73 |
| 19200 | 0 | 5 | 0.00 | 0 | 6 | -6.99 | 0 | 7 | 0.00 | 0 | 7 | 1.73 |
| 31250 | — | — | — | 0 | 3 | 0.00 | 0 | 4 | -1.70 | 0 | 4 | 0.00 |
| 38400 | 0 | 2 | 0.00 | 0 | 2 | 8.51 | 0 | 3 | 0.00 | 0 | 3 | 1.73 |

ϕ (MHz)

| Bit Rate (bit/s) | 6 | | | 6.144 | | | 7.3728 | | | 8 | | |
|---------------------|---|-----|-----------|-------|-----|-----------|--------|-----|-----------|---|-----|-----------|
| | n | N | Error (%) | n | N | Error (%) | n | N | Error (%) | n | N | Error (%) |
| 110 | 2 | 106 | -0.44 | 2 | 108 | 0.08 | 2 | 130 | -0.07 | 2 | 141 | 0.03 |
| 150 | 2 | 77 | 0.16 | 2 | 79 | 0.00 | 2 | 95 | 0.00 | 2 | 103 | 0.16 |
| 300 | 1 | 155 | 0.16 | 1 | 159 | 0.00 | 1 | 191 | 0.00 | 1 | 207 | 0.16 |
| 600 | 1 | 77 | 0.16 | 1 | 79 | 0.00 | 1 | 95 | 0.00 | 1 | 103 | 0.16 |
| 1200 | 0 | 155 | 0.16 | 0 | 159 | 0.00 | 0 | 191 | 0.00 | 0 | 207 | 0.16 |
| 2400 | 0 | 77 | 0.16 | 0 | 79 | 0.00 | 0 | 95 | 0.00 | 0 | 103 | 0.16 |
| 4800 | 0 | 38 | 0.16 | 0 | 39 | 0.00 | 0 | 47 | 0.00 | 0 | 51 | 0.16 |
| 9600 | 0 | 19 | -2.34 | 0 | 19 | 0.00 | 0 | 23 | 0.00 | 0 | 25 | 0.16 |
| 19200 | 0 | 9 | -2.34 | 0 | 9 | 0.00 | 0 | 11 | 0.00 | 0 | 12 | 0.16 |
| 31250 | 0 | 5 | 0.00 | 0 | 5 | 2.40 | 0 | 6 | 5.33 | 0 | 7 | 0.00 |
| 38400 | 0 | 4 | -2.34 | 0 | 4 | 0.00 | 0 | 5 | 0.00 | 0 | 6 | -6.99 |

 ϕ (MHz)

| Bit Rate (bit/s) | 9.8304 | | | 10 | | | 12 | | | 12.288 | | |
|---------------------|--------|-----|-----------|----|-----|-----------|----|-----|-----------|--------|-----|-----------|
| | n | N | Error (%) | n | N | Error (%) | n | N | Error (%) | n | N | Error (%) |
| 110 | 2 | 174 | -0.26 | 2 | 177 | -0.25 | 2 | 212 | 0.03 | 2 | 217 | 0.08 |
| 150 | 2 | 127 | 0.00 | 2 | 129 | 0.16 | 2 | 155 | 0.16 | 2 | 159 | 0.00 |
| 300 | 1 | 255 | 0.00 | 2 | 64 | 0.16 | 2 | 77 | 0.16 | 2 | 79 | 0.00 |
| 600 | 1 | 127 | 0.00 | 1 | 129 | 0.16 | 1 | 155 | 0.16 | 1 | 159 | 0.00 |
| 1200 | 0 | 255 | 0.00 | 1 | 64 | 0.16 | 1 | 77 | 0.16 | 1 | 79 | 0.00 |
| 2400 | 0 | 127 | 0.00 | 0 | 129 | 0.16 | 0 | 155 | 0.16 | 0 | 159 | 0.00 |
| 4800 | 0 | 63 | 0.00 | 0 | 64 | 0.16 | 0 | 77 | 0.16 | 0 | 79 | 0.00 |
| 9600 | 0 | 31 | 0.00 | 0 | 32 | -1.36 | 0 | 38 | 0.16 | 0 | 39 | 0.00 |
| 19200 | 0 | 15 | 0.00 | 0 | 15 | 1.73 | 0 | 19 | -2.34 | 0 | 19 | 0.00 |
| 31250 | 0 | 9 | -1.70 | 0 | 9 | 0.00 | 0 | 11 | 0.00 | 0 | 11 | 2.40 |
| 38400 | 0 | 7 | 0.00 | 0 | 7 | 1.73 | 0 | 9 | -2.34 | 0 | 9 | 0.00 |

| Bit Rate (bit/s) | ϕ (MHz) | | | | | | | | | | | |
|---------------------|--------------|-----|-----------|----|-----|-----------|---------|-----|-----------|----|-----|-----------|
| | 13 | | | 14 | | | 14.7456 | | | 16 | | |
| | n | N | Error (%) | n | N | Error (%) | n | N | Error (%) | n | N | Error (%) |
| 110 | 2 | 230 | -0.08 | 2 | 248 | -0.17 | 3 | 64 | 0.70 | 3 | 70 | 0.03 |
| 150 | 2 | 168 | 0.16 | 2 | 181 | 0.16 | 2 | 191 | 0.00 | 2 | 207 | 0.16 |
| 300 | 2 | 84 | -0.43 | 2 | 90 | 0.16 | 2 | 95 | 0.00 | 2 | 103 | 0.16 |
| 600 | 1 | 168 | 0.16 | 1 | 181 | 0.16 | 1 | 191 | 0.00 | 1 | 207 | 0.16 |
| 1200 | 1 | 84 | -0.43 | 1 | 90 | 0.16 | 1 | 95 | 0.00 | 1 | 103 | 0.16 |
| 2400 | 0 | 168 | 0.16 | 0 | 181 | 0.16 | 0 | 191 | 0.00 | 0 | 207 | 0.16 |
| 4800 | 0 | 84 | -0.43 | 0 | 90 | 0.16 | 0 | 95 | 0.00 | 0 | 103 | 0.16 |
| 9600 | 0 | 41 | 0.76 | 0 | 45 | -0.93 | 0 | 47 | 0.00 | 0 | 51 | 0.16 |
| 19200 | 0 | 20 | 0.76 | 0 | 22 | -0.93 | 0 | 23 | 0.00 | 0 | 25 | 0.16 |
| 31250 | 0 | 12 | 0.00 | 0 | 13 | 0.00 | 0 | 14 | -1.70 | 0 | 15 | 0.00 |
| 38400 | 0 | 10 | -3.82 | 0 | 10 | 3.57 | 0 | 11 | 0.00 | 0 | 12 | 0.16 |

| Bit Rate (bit/s) | ϕ (MHz) | | | | | | | | |
|---------------------|--------------|-----|-----------|----|-----|-----------|----|-----|-----------|
| | 18 | | | 20 | | | 25 | | |
| | n | N | Error (%) | n | N | Error (%) | n | N | Error (%) |
| 110 | 3 | 79 | -0.12 | 3 | 88 | -0.25 | 3 | 110 | -0.02 |
| 150 | 2 | 233 | 0.16 | 3 | 64 | 0.16 | 3 | 80 | -0.47 |
| 300 | 2 | 116 | 0.16 | 2 | 129 | 0.16 | 2 | 162 | 0.15 |
| 600 | 1 | 233 | 0.16 | 2 | 64 | 0.16 | 2 | 80 | -0.47 |
| 1200 | 1 | 116 | 0.16 | 1 | 129 | 0.16 | 1 | 162 | 0.15 |
| 2400 | 0 | 233 | 0.16 | 1 | 64 | 0.16 | 1 | 80 | -0.47 |
| 4800 | 0 | 116 | 0.16 | 0 | 129 | 0.16 | 0 | 162 | 0.15 |
| 9600 | 0 | 58 | -0.69 | 0 | 64 | 0.16 | 0 | 80 | -0.47 |
| 19200 | 0 | 28 | 1.02 | 0 | 32 | -1.36 | 0 | 40 | -0.76 |
| 31250 | 0 | 17 | 0.00 | 0 | 19 | 0.00 | 0 | 24 | 0.00 |
| 38400 | 0 | 14 | -2.34 | 0 | 15 | 1.73 | 0 | 19 | 1.73 |

Table 12.4 Examples of Bit Rates and BRR Settings in Synchronous Mode

| Bit Rate | ϕ (MHz) | | | | | | | | | | | | | | | | | |
|----------|--------------|-----|---|-----|---|-----|----|-----|----|-----|----|-----|----|-----|----|-----|----|-----|
| | 2 | | 4 | | 8 | | 10 | | 13 | | 16 | | 18 | | 20 | | 25 | |
| (bit/s) | n | N | n | N | n | N | n | N | n | N | n | N | n | N | n | N | n | N |
| 110 | 3 | 70 | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| 250 | 2 | 124 | 2 | 249 | 3 | 124 | — | — | 3 | 202 | 3 | 249 | — | — | — | — | — | — |
| 500 | 1 | 249 | 2 | 124 | 2 | 249 | — | — | 3 | 101 | 3 | 124 | 3 | 140 | 3 | 155 | — | — |
| 1k | 1 | 124 | 1 | 249 | 2 | 124 | — | — | 2 | 202 | 2 | 249 | 3 | 69 | 3 | 77 | 3 | 97 |
| 2.5k | 0 | 199 | 1 | 99 | 1 | 199 | 1 | 249 | 2 | 80 | 2 | 99 | 2 | 112 | 2 | 124 | 2 | 155 |
| 5k | 0 | 99 | 0 | 199 | 1 | 99 | 1 | 124 | 1 | 162 | 1 | 199 | 1 | 224 | 1 | 249 | 2 | 77 |
| 10k | 0 | 49 | 0 | 99 | 0 | 199 | 0 | 249 | 1 | 80 | 1 | 99 | 1 | 112 | 1 | 124 | 1 | 155 |
| 25k | 0 | 19 | 0 | 39 | 0 | 79 | 0 | 99 | 0 | 129 | 0 | 159 | 0 | 179 | 0 | 199 | 0 | 249 |
| 50k | 0 | 9 | 0 | 19 | 0 | 39 | 0 | 49 | 0 | 64 | 0 | 79 | 0 | 89 | 0 | 99 | 0 | 124 |
| 100k | 0 | 4 | 0 | 9 | 0 | 19 | 0 | 24 | — | — | 0 | 39 | 0 | 44 | 0 | 49 | 0 | 62 |
| 250k | 0 | 1 | 0 | 3 | 0 | 7 | 0 | 9 | 0 | 12 | 0 | 15 | 0 | 17 | 0 | 19 | 0 | 24 |
| 500k | 0 | 0* | 0 | 1 | 0 | 3 | 0 | 4 | — | — | 0 | 7 | 0 | 8 | 0 | 9 | — | — |
| 1M | | | 0 | 0* | 0 | 1 | — | — | — | — | 0 | 3 | 0 | 4 | 0 | 4 | — | — |
| 2M | | | | | 0 | 0* | — | — | — | — | 0 | 1 | — | — | — | — | — | — |
| 2.5M | | | | | — | — | 0 | 0* | — | — | — | — | — | — | — | — | — | — |
| 4M | | | | | | | | | | 0 | 0* | — | — | — | — | — | — | — |

Legend:

Blank: No setting available

—: Setting possible, but error occurs

*: Continuous transmission/reception not possible

Note: Settings with an error of 1% or less are recommended.

The BRR setting is calculated as follows:

Asynchronous mode:

$$N = \frac{\phi}{64 \times 2^{2n-1} \times B} \times 10^6 - 1$$

Synchronous mode:

$$N = \frac{\phi}{8 \times 2^{2n-1} \times B} \times 10^6 - 1$$

Legend:

B: Bit rate (bit/s)

N: BRR setting for baud rate generator ($0 \leq N \leq 255$)

ϕ : System clock frequency (MHz)

n: Baud rate generator input clock ($n = 0, 1, 2, 3$)

(For the clock sources and values of n, see the following table.)

| n | Clock Source | SMR Settings | |
|---|--------------|--------------|------|
| | | CKS1 | CKS0 |
| 0 | ϕ | 0 | 0 |
| 1 | $\phi/4$ | 0 | 1 |
| 2 | $\phi/16$ | 1 | 0 |
| 3 | $\phi/64$ | 1 | 1 |

The bit rate error in asynchronous mode is calculated as follows:

$$\text{Error (\%)} = \left\{ \frac{\phi \times 10^6}{(N + 1) \times B \times 64 \times 2^{2n-1}} - 1 \right\} \times 100$$

Table 12.5 shows the maximum bit rates in asynchronous mode for various system clock frequencies. Tables 12.6 and 12.7 show the maximum bit rates with external clock input.

Table 12.5 Maximum Bit Rates for Various Frequencies (Asynchronous Mode)

| ϕ (MHz) | Maximum Bit Rate (bit/s) | Settings | |
|--------------|--------------------------|----------|---|
| | | n | N |
| 2 | 62500 | 0 | 0 |
| 2.097152 | 65536 | 0 | 0 |
| 2.4576 | 76800 | 0 | 0 |
| 3 | 93750 | 0 | 0 |
| 3.6864 | 115200 | 0 | 0 |
| 4 | 125000 | 0 | 0 |
| 4.9152 | 153600 | 0 | 0 |
| 5 | 156250 | 0 | 0 |
| 6 | 187500 | 0 | 0 |
| 6.144 | 192000 | 0 | 0 |
| 7.3728 | 230400 | 0 | 0 |
| 8 | 250000 | 0 | 0 |
| 9.8304 | 307200 | 0 | 0 |
| 10 | 312500 | 0 | 0 |
| 12 | 375000 | 0 | 0 |
| 12.288 | 384000 | 0 | 0 |
| 14 | 437500 | 0 | 0 |
| 14.7456 | 460800 | 0 | 0 |
| 16 | 500000 | 0 | 0 |
| 17.2032 | 537600 | 0 | 0 |
| 18 | 562500 | 0 | 0 |
| 20 | 625000 | 0 | 0 |
| 25 | 781250 | 0 | 0 |

Table 12.6 Maximum Bit Rates with External Clock Input (Asynchronous Mode)

| ϕ (MHz) | External Input Clock (MHz) | Maximum Bit Rate (bit/s) |
|--------------|----------------------------|--------------------------|
| 2 | 0.5000 | 31250 |
| 2.097152 | 0.5243 | 32768 |
| 2.4576 | 0.6144 | 38400 |
| 3 | 0.7500 | 46875 |
| 3.6864 | 0.9216 | 57600 |
| 4 | 1.0000 | 62500 |
| 4.9152 | 1.2288 | 76800 |
| 5 | 1.2500 | 78125 |
| 6 | 1.5000 | 93750 |
| 6.144 | 1.5360 | 96000 |
| 7.3728 | 1.8432 | 115200 |
| 8 | 2.0000 | 125000 |
| 9.8304 | 2.4576 | 153600 |
| 10 | 2.5000 | 156250 |
| 12 | 3.0000 | 187500 |
| 12.288 | 3.0720 | 192000 |
| 14 | 3.5000 | 218750 |
| 14.7456 | 3.6864 | 230400 |
| 16 | 4.0000 | 250000 |
| 17.2032 | 4.3008 | 268800 |
| 18 | 4.5000 | 281250 |
| 20 | 5.0000 | 312500 |
| 25 | 6.2500 | 390625 |

Table 12.7 Maximum Bit Rates with External Clock Input (Synchronous Mode)

| ϕ (MHz) | External Input Clock (MHz) | Maximum Bit Rate (bit/s) |
|--------------|----------------------------|--------------------------|
| 2 | 0.3333 | 333333.3 |
| 4 | 0.6667 | 666666.7 |
| 6 | 1.0000 | 1000000.0 |
| 8 | 1.3333 | 1333333.3 |
| 10 | 1.6667 | 1666666.7 |
| 12 | 2.0000 | 2000000.0 |
| 14 | 2.3333 | 2333333.3 |
| 16 | 2.6667 | 2666666.7 |
| 18 | 3.0000 | 3000000.0 |
| 20 | 3.3333 | 3333333.3 |
| 25 | 4.1667 | 4166666.7 |

12.3 Operation

12.3.1 Overview

The SCI can carry out serial communication in two modes: asynchronous mode in which synchronization is achieved character by character, and synchronous mode in which synchronization is achieved with clock pulses. A smart card interface is also supported as a serial communication function for an IC card interface.

Selection of asynchronous or synchronous mode and the transmission format for the normal serial communication interface is made in SMR, as shown in table 12.8. The SCI clock source is selected by the C/\bar{A} bit in SMR and the CKE1 and CKE0 bits in SCR, as shown in table 12.9.

For details of the procedures for switching between LSB-first and MSB-first mode and inverting the data logic level, see section 13.2.1, Smart Card Mode Register (SCMR).

For selection of the smart card interface format, see section 13.3.3, Data Format.

Asynchronous Mode

- Data length is selectable: 7 or 8 bits
- Parity and multiprocessor bits are selectable, and so is the stop bit length (1 or 2 bits). These selections determine the communication format and character length.
- In receiving, it is possible to detect framing errors, parity errors, overrun errors, and the break state.
- An internal or external clock can be selected as the SCI clock source.
 - When an internal clock is selected, the SCI operates using the on-chip baud rate generator, and can output a serial clock signal with a frequency matching the bit rate.
 - When an external clock is selected, the external clock input must have a frequency 16 times the bit rate. (The on-chip baud rate generator is not used.)

Synchronous Mode

- The communication format has a fixed 8-bit data length.
- In receiving, it is possible to detect overrun errors.
- An internal or external clock can be selected as the SCI clock source.
 - When an internal clock is selected, the SCI operates using the on-chip baud rate generator, and can output a serial clock signal to external devices.
 - When an external clock is selected, the SCI operates on the input serial clock. The on-chip baud rate generator is not used.

Smart Card Interface

- One frame consists of 8-bit data and a parity bit.
- In transmitting, a guard time of at least two elementary time units (2 etu) is provided between the end of the parity bit and the start of the next frame. (An elementary time unit is the time required to transmit one bit.)
- In receiving, if a parity error is detected, a low error signal level is output for 1 etu, beginning 10.5 etu after the start bit..
- In transmitting, if an error signal is received, the same data is automatically transmitted again after at least 2 etu.
- Only asynchronous communication is supported. There is no synchronous communication function.

For details of smart card interface operation, see section 13, Smart Card Interface.

Table 12.8 SMR Settings and Serial Communication Formats

| SMR Settings | | | | | | SCI Communication Format | | | | | |
|-----------------------|--------------|-------------|-------------|---|---|--------------------------|----------------------------|---------------|--------------------|---------|-------|
| Bit 7 C/ \bar{A} | Bit 6 CHR | Bit 2 MP | Bit 5 PE | Bit 3 STOP | Mode | Data Length | Multi- processor Bit | Parity Bit | Stop Bit Length | | |
| 0 | 0 | 0 | 0 | 0 | Asyn- chronous mode | 8-bit data | Absent | Absent | 1 bit | | |
| | | | | 1 | | | | | 2 bits | | |
| | | | | 1 | | | | | 0 | Present | 1 bit |
| | | | | 1 | | | | | 2 bits | | |
| | 1 | 0 | 0 | 0 | Asyn- chronous mode (multi- processor format) | 7-bit data | Absent | Absent | 1 bit | | |
| | | | | 1 | | | | | 2 bits | | |
| | | | | 1 | | | | | 0 | Present | 1 bit |
| | | | | 1 | | | | | 2 bits | | |
| 0 | 1 | — | 0 | Asyn- chronous mode (multi- processor format) | 8-bit data | Present | Absent | 1 bit | | | |
| | | | 1 | | | | | 2 bits | | | |
| | | | 1 | | | | | 0 | 7-bit data | 1 bit | |
| | | | 1 | | | | | 2 bits | | | |
| 1 | — | — | — | — | Syn- chronous mode | 8-bit data | Absent | None | | | |

Table 12.9 SMR and SCR Settings and SCI Clock Source Selection

| SMR Bit 7 C/ \bar{A} | SCR Setting | | | SCI Transmit/Receive clock | |
|------------------------------|---------------|---------------|----------------------|----------------------------|--|
| | Bit 1 CKE1 | Bit 0 CKE0 | Mode | Clock Source | SCK Pin Function |
| 0 | 0 | 0 | Asynchronous mode | Internal | SCI does not use the SCK pin |
| | | 1 | | | Outputs clock with frequency matching the bit rate |
| 1 | 1 | 0 | Synchronous mode | External | Inputs clock with frequency 16 times the bit rate |
| | | 1 | | | Outputs the serial clock |
| 1 | 1 | 0 | Synchronous mode | Internal | Outputs the serial clock |
| | | 1 | | | External |

12.3.2 Operation in Asynchronous Mode

In asynchronous mode, each transmitted or received character begins with a start bit and ends with one or two stop bits. Serial communication is synchronized one character at a time.

The transmitting and receiving sections of the SCI are independent, so full-duplex communication is possible. The transmitter and the receiver are both double-buffered, so data can be written and read while transmitting and receiving are in progress, enabling continuous transmitting and receiving.

Figure 12.2 shows the general format of asynchronous serial communication. In asynchronous serial communication the communication line is normally held in the mark (high) state. The SCI monitors the line and starts serial communication when the line goes to the space (low) state, indicating a start bit. One serial character consists of a start bit (low), data (LSB first), parity bit (high or low), and one or two stop bits (high), in that order.

When receiving in asynchronous mode, the SCI synchronizes at the falling edge of the start bit. The SCI samples each data bit on the eighth pulse of a clock with a frequency 16 times the bit rate. Receive data is latched at the center of each bit.

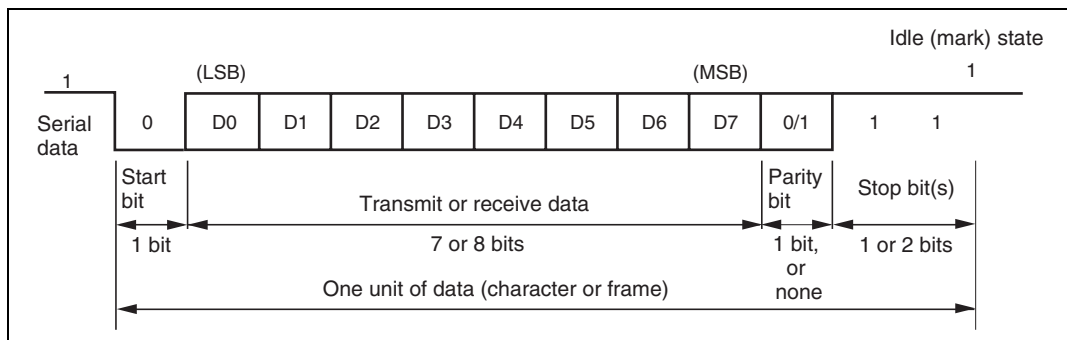


Figure 12.2 Data Format in Asynchronous Communication
(Example: 8-Bit Data with Parity and 2 Stop Bits)

Communication Formats: Table 12.10 shows the 12 communication formats that can be selected in asynchronous mode. The format is selected by settings in SMR.

Table 12.10 Serial Communication Formats (Asynchronous Mode)

| SMR Settings | | | | Serial Communication Format and Frame Length | | | | | | | | | | | | |
|--------------|----|----|------|--|------------|---|---|---|---|---|---|------|------|------|------|--|
| CHR | PE | MP | STOP | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | |
| 0 | 0 | 0 | 0 | S | 8-bit data | | | | | | | | STOP | | | |
| 0 | 0 | 0 | 1 | S | 8-bit data | | | | | | | | STOP | STOP | | |
| 0 | 1 | 0 | 0 | S | 8-bit data | | | | | | | | P | STOP | | |
| 0 | 1 | 0 | 1 | S | 8-bit data | | | | | | | | P | STOP | STOP | |
| 1 | 0 | 0 | 0 | S | 7-bit data | | | | | | | STOP | | | | |
| 1 | 0 | 0 | 1 | S | 7-bit data | | | | | | | STOP | STOP | | | |
| 1 | 1 | 0 | 0 | S | 7-bit data | | | | | | | P | STOP | | | |
| 1 | 1 | 0 | 1 | S | 7-bit data | | | | | | | P | STOP | STOP | | |
| 0 | — | 1 | 0 | S | 8-bit data | | | | | | | | MPB | STOP | | |
| 0 | — | 1 | 1 | S | 8-bit data | | | | | | | | MPB | STOP | STOP | |
| 1 | — | 1 | 0 | S | 7-bit data | | | | | | | MPB | STOP | | | |
| 1 | — | 1 | 1 | S | 7-bit data | | | | | | | MPB | STOP | STOP | | |

Legend:

S: Start bit

STOP: Stop bit

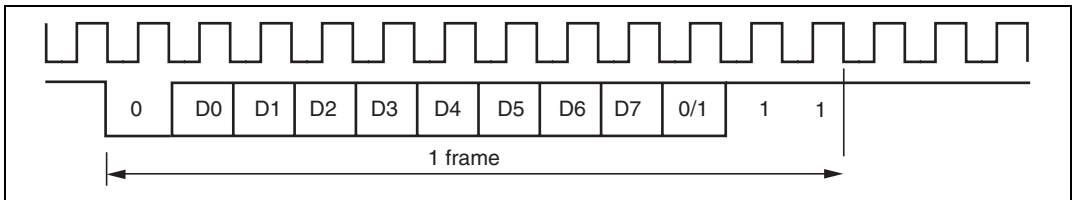
P: Parity bit

MPB: Multiprocessor bit

Clock: An internal clock generated by the on-chip baud rate generator or an external clock input from the SCK pin can be selected as the SCI transmit/receive clock. The clock source is selected by the C/\bar{A} bit in SMR and bits CKE1 and CKE0 in SCR. For details of SCI clock source selection, see table 12.9.

When an external clock is input at the SCK pin, it must have a frequency 16 times the desired bit rate.

When the SCI is operated on an internal clock, it can output a clock signal at the SCK pin. The frequency of this output clock is equal to the bit rate. The phase is aligned as shown in figure 12.3 so that the rising edge of the clock occurs at the center of each transmit data bit.



**Figure 12.3 Phase Relationship between Output Clock and Serial Data
(Asynchronous Mode)**

Transmitting and Receiving Data:

- **SCI Initialization (Asynchronous Mode):** Before transmitting or receiving data, clear the TE and RE bits to 0 in SCR, then initialize the SCI as follows.

When changing the communication mode or format, always clear the TE and RE bits to 0 before following the procedure given below. Clearing TE to 0 sets the TDRE flag to 1 and initializes TSR. Clearing RE to 0, however, does not initialize the RDRF, PER, FER, and ORER flags, or RDR, which retain their previous contents.

When an external clock is used the clock should not be stopped during initialization or subsequent operation, since operation will be unreliable in this case.

Figure 12.4 shows a sample flowchart for initializing the SCI.

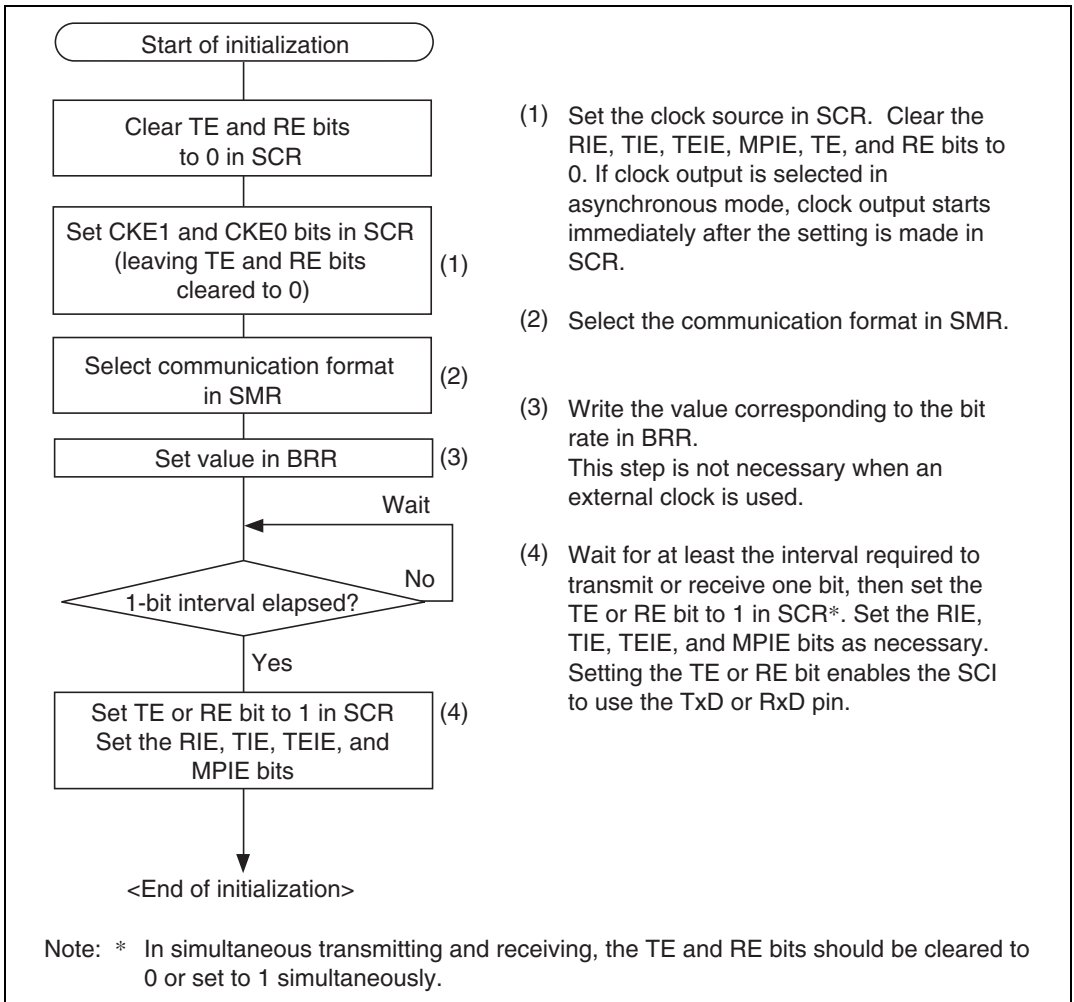


Figure 12.4 Sample Flowchart for SCI Initialization

- **Transmitting Serial Data (Asynchronous Mode):** Figure 12.5 shows a sample flowchart for transmitting serial data and indicates the procedure to follow.

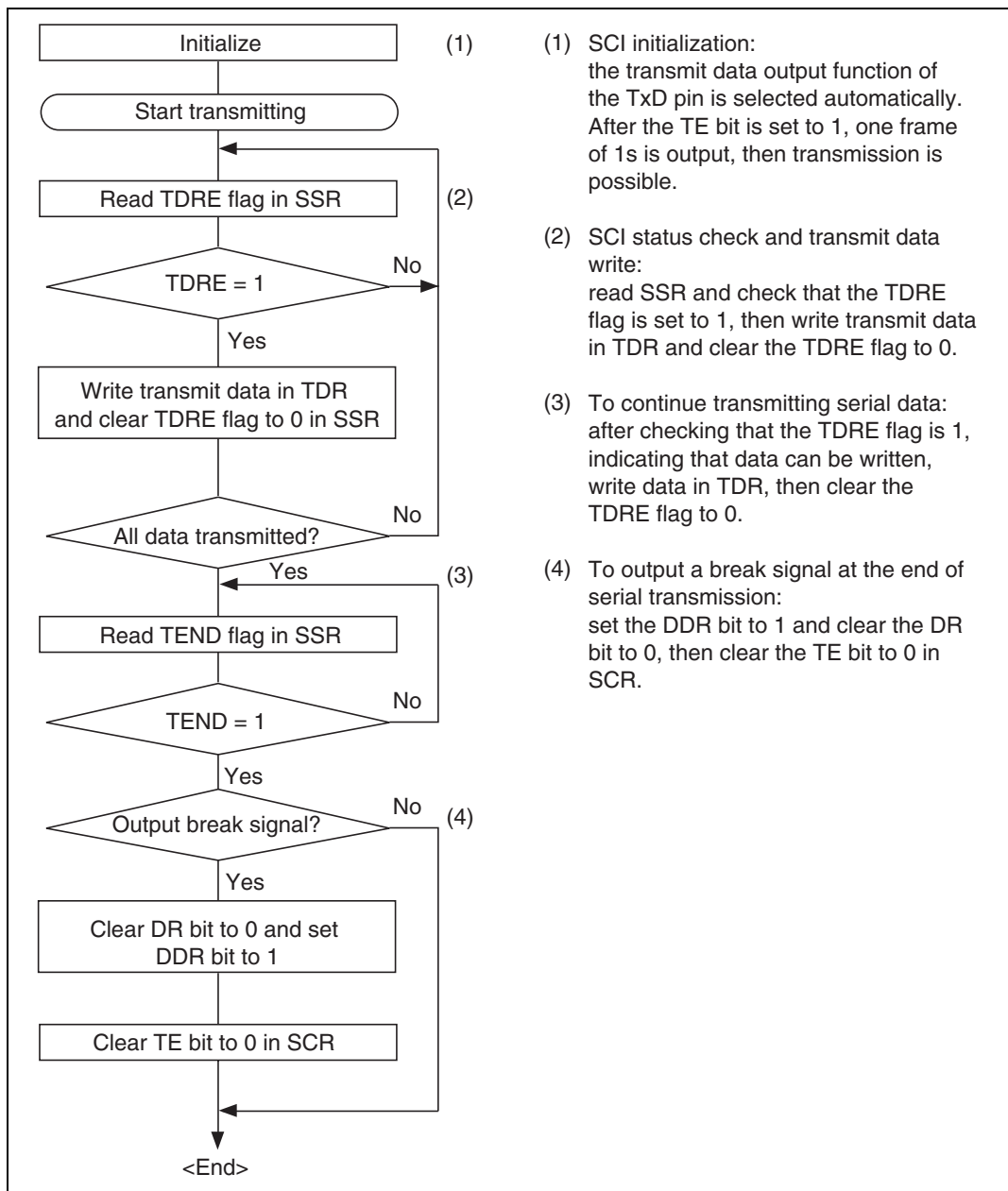


Figure 12.5 Sample Flowchart for Transmitting Serial Data

In transmitting serial data, the SCI operates as follows:

- The SCI monitors the TDRE flag in SSR. When the TDRE flag is cleared to 0, the SCI recognizes that TDR contains new data, and loads this data from TDR into TSR.
- After loading the data from TDR to TSR, the SCI sets the TDRE flag to 1 and starts transmitting. If the TIE bit is set to 1 in SCR, the SCI requests a transmit-data-empty interrupt (TXI) at this time.

Serial transmit data is transmitted in the following order from the TxD pin:

— Start bit: One 0 bit is output.

— Transmit data: 7 or 8 bits are output, LSB first.

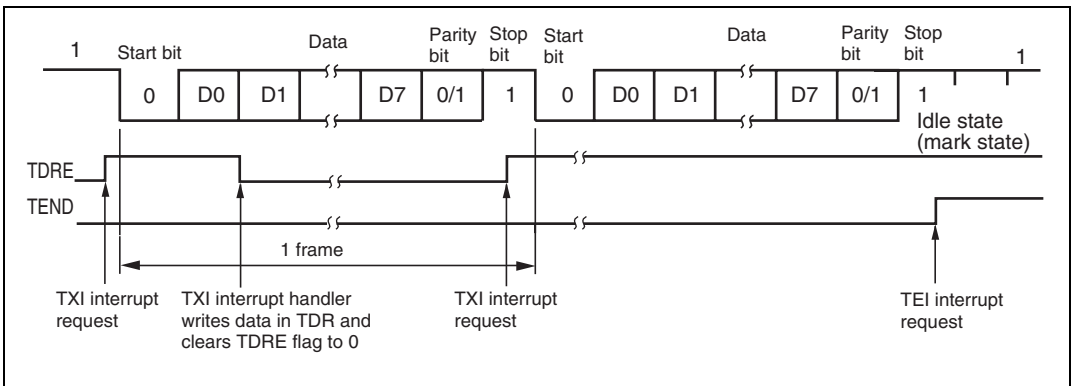
— Parity bit or multiprocessor bit: One parity bit (even or odd parity), or one multiprocessor bit is output. Formats in which neither a parity bit nor a multiprocessor bit is output can also be selected.

— Stop bit(s): One or two 1 bits (stop bits) are output.

— Mark state: Output of 1 bits continues until the start bit of the next transmit data.

- The SCI checks the TDRE flag when it outputs the stop bit. If the TDRE flag is 0, the SCI loads new data from TDR into TSR, outputs the stop bit, then begins serial transmission of the next frame. If the TDRE flag is 1, the SCI sets the TEND flag to 1 in SSR, outputs the stop bit, then continues output of 1 bits in the mark state. If the TEIE bit is set to 1 in SCR, a transmit-end interrupt (TEI) is requested at this time

Figure 12.6 shows an example of SCI transmit operation in asynchronous mode.



**Figure 12.6 Example of SCI Transmit Operation in Asynchronous Mode
(8-Bit Data with Parity and One Stop Bit)**

- Receiving Serial Data (Asynchronous Mode): Figure 12.7 shows a sample flowchart for receiving serial data and indicates the procedure to follow.

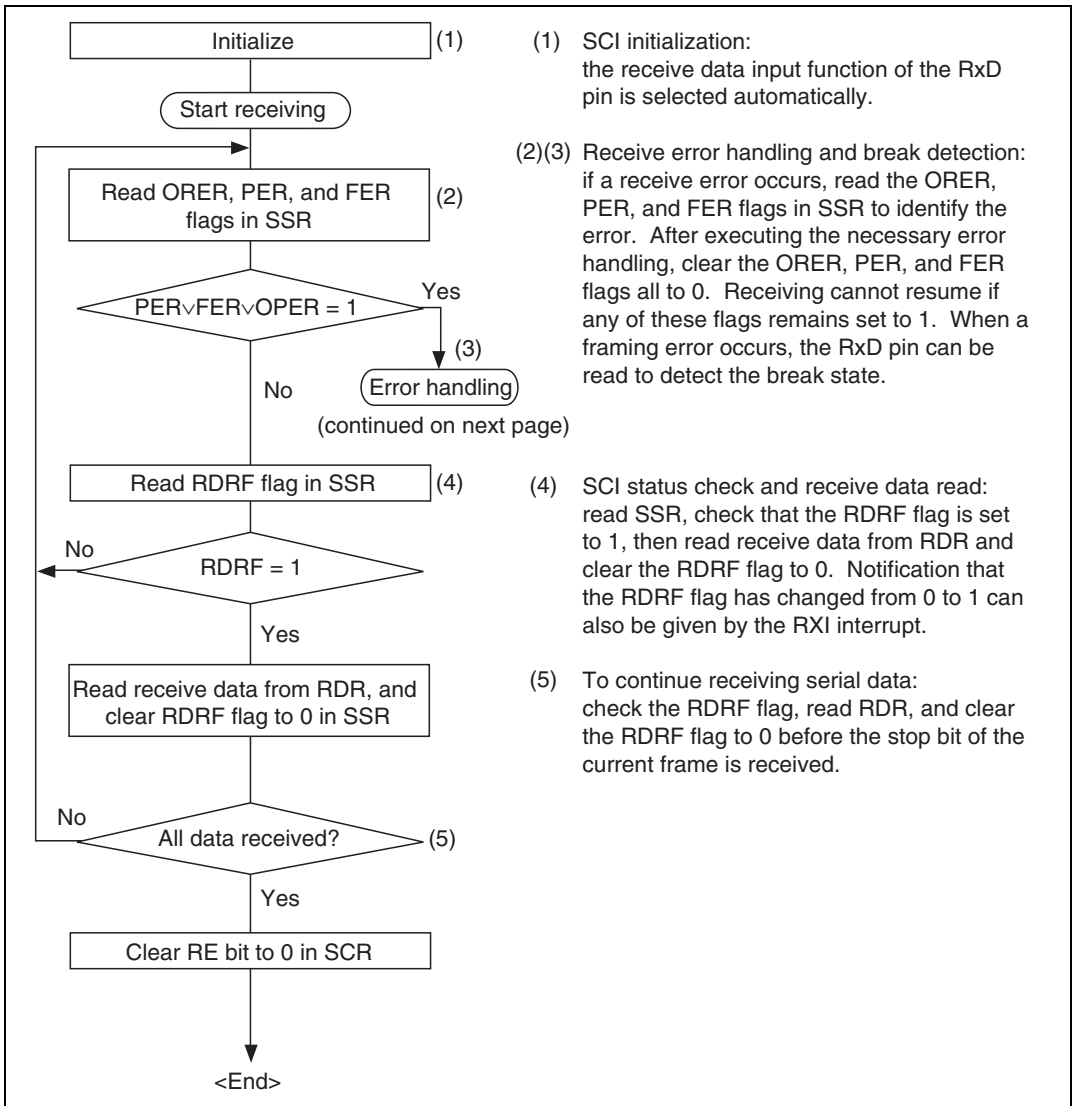


Figure 12.7 Sample Flowchart for Receiving Serial Data (1)

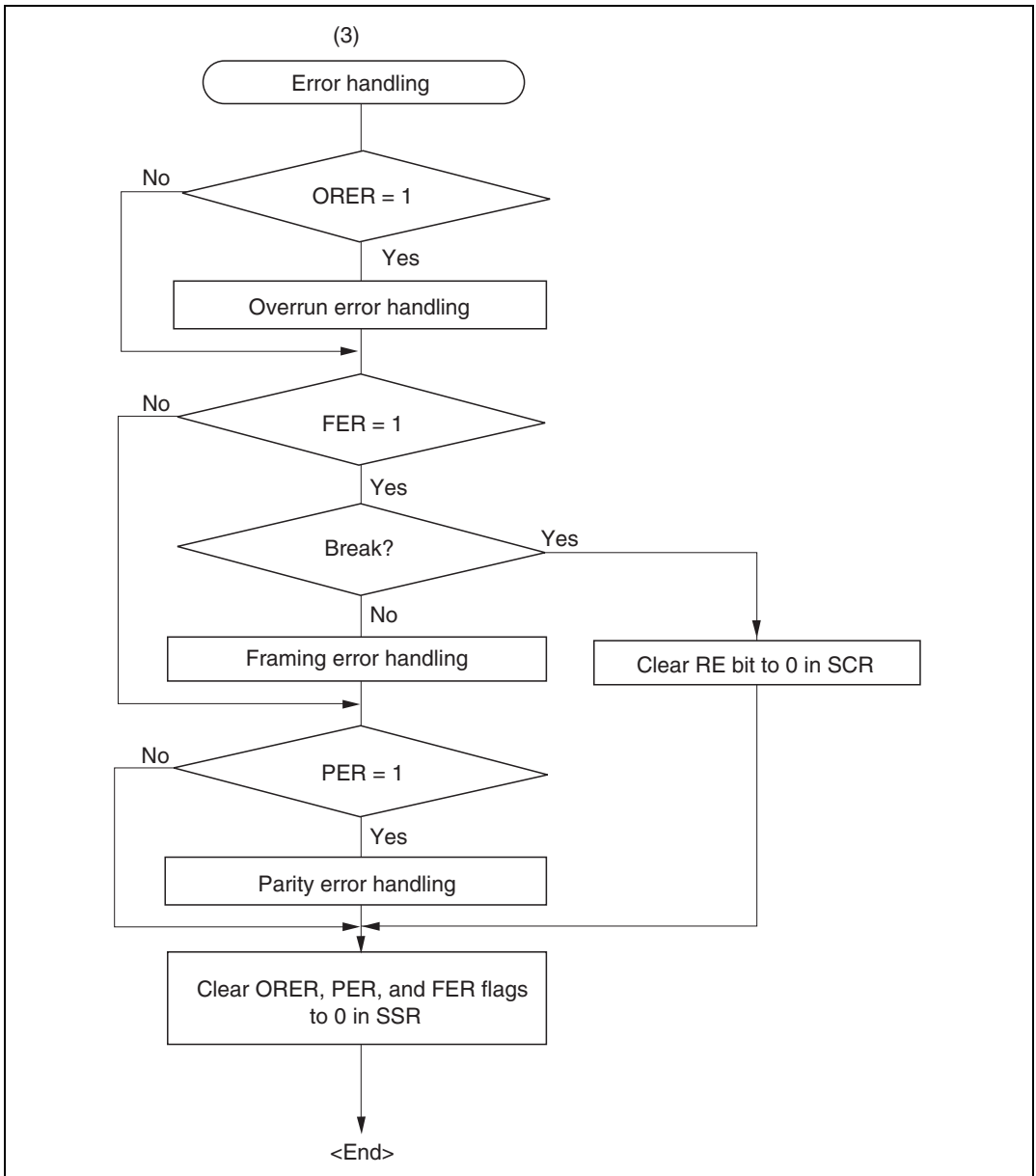


Figure 12.7 Sample Flowchart for Receiving Serial Data (2)

In receiving, the SCI operates as follows:

- The SCI monitors the communication line. When it detects a start bit (0 bit), the SCI synchronizes internally and starts receiving.
- Receive data is stored in RSR in order from LSB to MSB.
- The parity bit and stop bit are received.

After receiving these bits, the SCI carries out the following checks:

- Parity check: The number of 1s in the receive data must match the even or odd parity setting of in the O/\bar{E} bit in SMR.
- Stop bit check: The stop bit value must be 1. If there are two stop bits, only the first is checked.
- Status check: The RDRF flag must be 0, indicating that the receive data can be transferred from RSR into RDR.

If these all checks pass, the RDRF flag is set to 1 and the received data is stored in RDR. If one of the checks fails (receive error*), the SCI operates as shown in table 12.11.

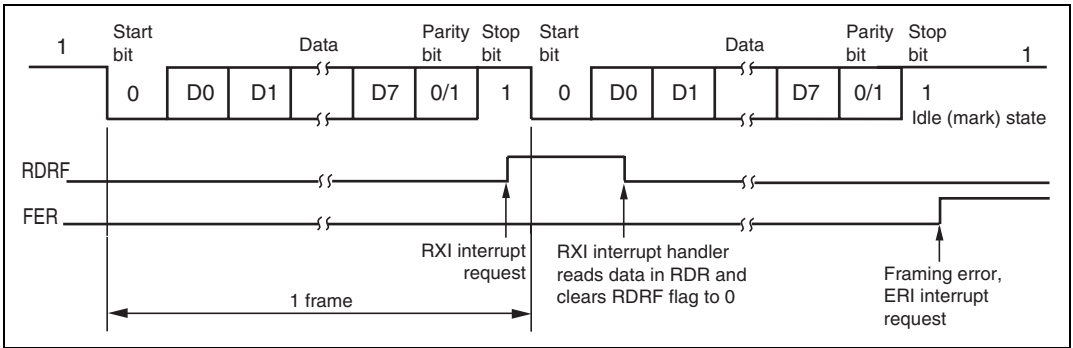
Note: * When a receive error occurs, further receiving is disabled. In receiving, the RDRF flag is not set to 1. Be sure to clear the error flags to 0.

- When the RDRF flag is set to 1, if the RIE bit is set to 1 in SCR, a receive-data-full interrupt (RXI) is requested. If the ORER, PER, or FER flag is set to 1 and the RIE bit in SCR is also set to 1, a receive-error interrupt (ERI) is requested.

Table 12.11 Receive Error Conditions

| Receive Error | Abbreviation | Condition | Data Transfer |
|---------------|--------------|--|---|
| Overrun error | ORER | Receiving of next data ends while RDRF flag is still set to 1 in SSR | Receive data is not transferred from RSR to RDR |
| Framing error | FER | Stop bit is 0 | Receive data is transferred from RSR to RDR |
| Parity error | PER | Parity of received data differs from even/odd parity setting in SMR | Receive data is transferred from RSR to RDR |

Figure 12.8 shows an example of SCI receive operation in asynchronous mode.



**Figure 12.8 Example of SCI Receive Operation
(8-Bit Data with Parity and One Stop Bit)**

12.3.3 Multiprocessor Communication

The multiprocessor communication function enables several processors to share a single serial communication line. The processors communicate in asynchronous mode using a format with an additional multiprocessor bit (multiprocessor format).

In multiprocessor communication, each receiving processor is addressed by an ID. A serial communication cycle consists of an ID-sending cycle that identifies the receiving processor, and a data-sending cycle. The multiprocessor bit distinguishes ID-sending cycles from data-sending cycles.

The transmitting processor starts by sending the ID of the receiving processor with which it wants to communicate as data with the multiprocessor bit set to 1. Next the transmitting processor sends transmit data with the multiprocessor bit cleared to 0.

Receiving processors skip incoming data until they receive data with the multiprocessor bit set to 1. When they receive data with the multiprocessor bit set to 1, receiving processors compare the data with their IDs. Processors with IDs not matching the received data skip further incoming data until they again receive data with the multiprocessor bit set to 1. Multiple processors can send and receive data in this way.

Figure 12.9 shows an example of communication among different processors using a multiprocessor format.

Communication Formats: Four formats are available. Parity bit settings are ignored when a multiprocessor format is selected. For details see table 12.10.

Clock: See the description of asynchronous mode.

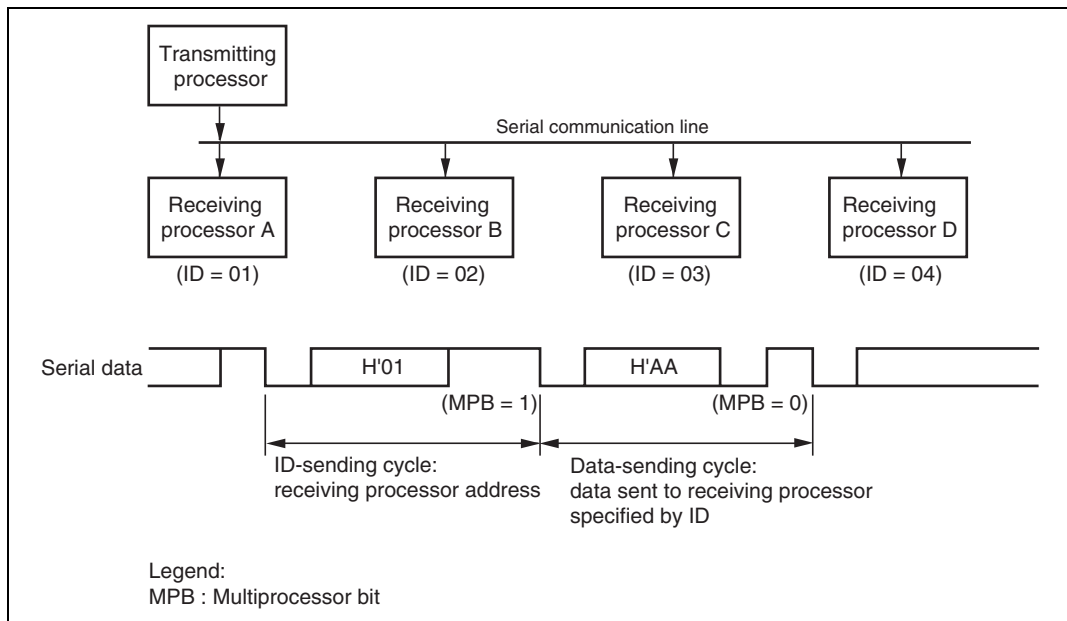


Figure 12.9 Example of Communication among Processors using Multiprocessor Format (Sending Data H'AA to Receiving Processor A)

Transmitting and Receiving Data:

- **Transmitting Multiprocessor Serial Data:** Figure 12.10 shows a sample flowchart for transmitting multiprocessor serial data and indicates the procedure to follow.

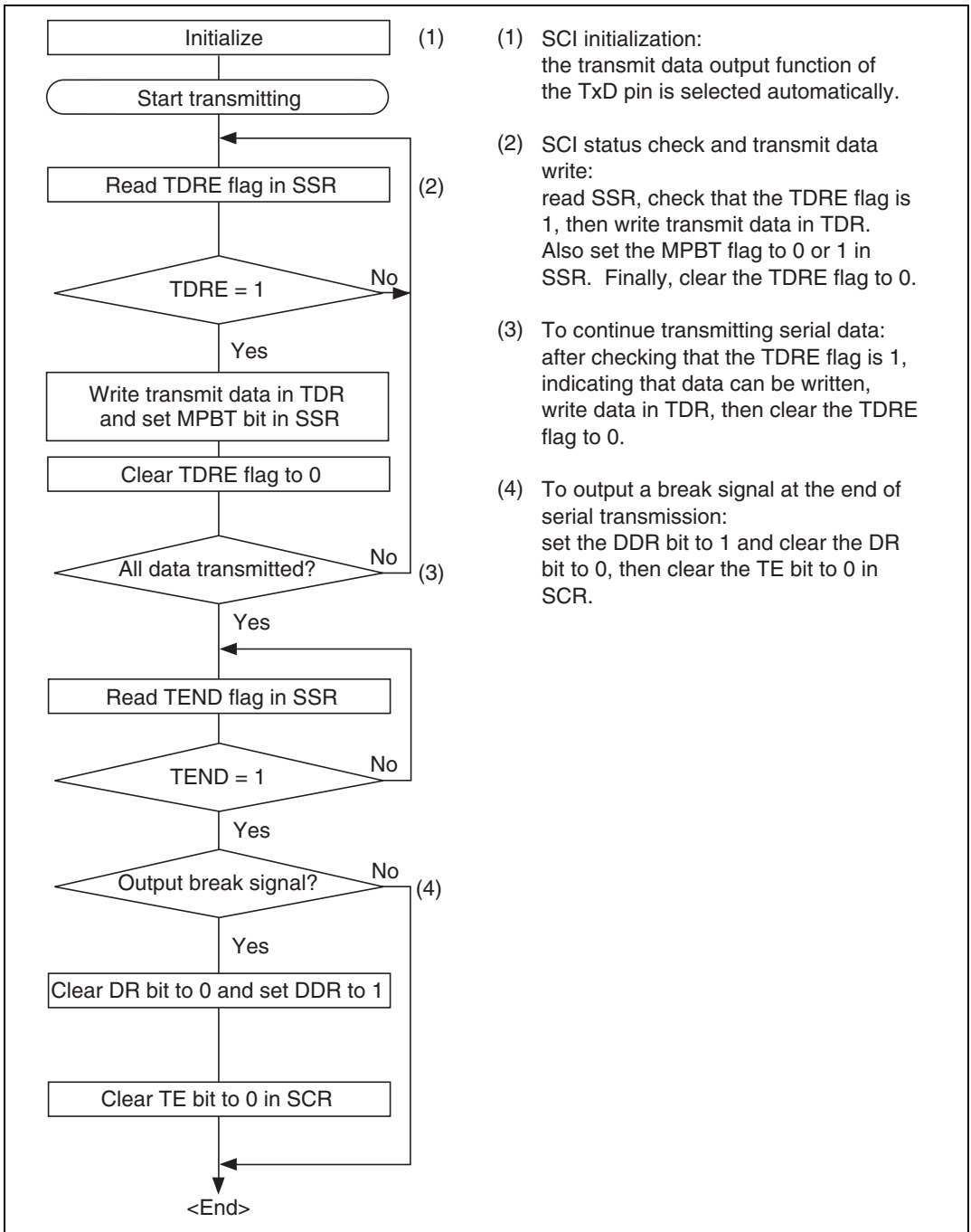


Figure 12.10 Sample Flowchart for Transmitting Multiprocessor Serial Data

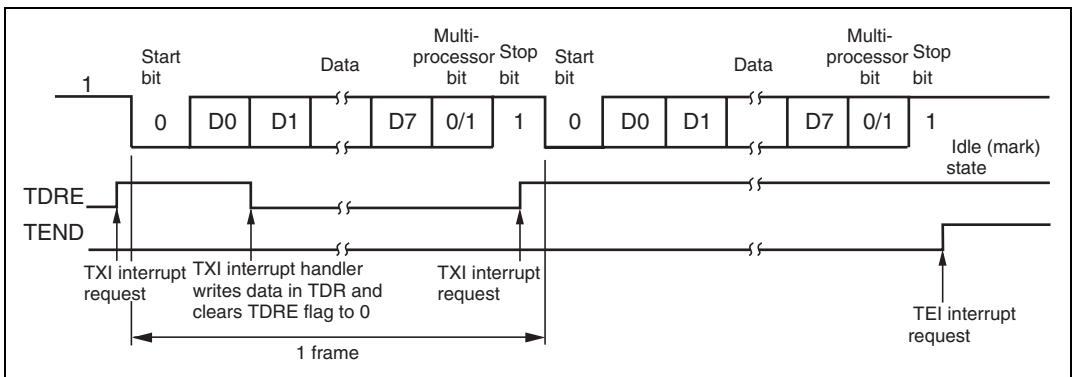
In transmitting serial data, the SCI operates as follows:

- The SCI monitors the TDRE flag in SSR. When the TDRE flag is cleared to 0, the SCI recognizes that TDR contains new data, and loads this data from TDR into TSR.
- After loading the data from TDR to TSR, the SCI sets the TDRE flag to 1 and starts transmitting. If the TIE bit is set to 1 in SCR, the SCI requests a transmit-data-empty interrupt (TXI) at this time.

Serial transmit data is transmitted in the following order from the TxD pin:

- Start bit: One 0 bit is output.
- Transmit data: 7 or 8 bits are output, LSB first.
- Multiprocessor bit: One multiprocessor bit (MPBT value) is output.
- Stop bit(s): One or two 1 bits (stop bits) are output.
- Mark state: Output of 1 bits continues until the start bit of the next transmit data.
- The SCI checks the TDRE flag when it outputs the stop bit. If the TDRE flag is 0, the SCI loads new data from TDR into TSR, outputs the stop bit, then begins serial transmission of the next frame. If the TDRE flag is 1, the SCI sets the TEND flag to 1 in SSR, outputs the stop bit, then continues output of 1 bits in the mark state. If the TEIE bit is set to 1 in SCR, a transmit-end interrupt (TEI) is requested at this time

Figure 12.11 shows an example of SCI transmit operation using a multiprocessor format.



**Figure 12.11 Example of SCI Transmit Operation
(8-Bit Data with Multiprocessor Bit and One Stop Bit)**

- Receiving Multiprocessor Serial Data: Figure 12.12 shows a sample flowchart for receiving multiprocessor serial data and indicates the procedure to follow.

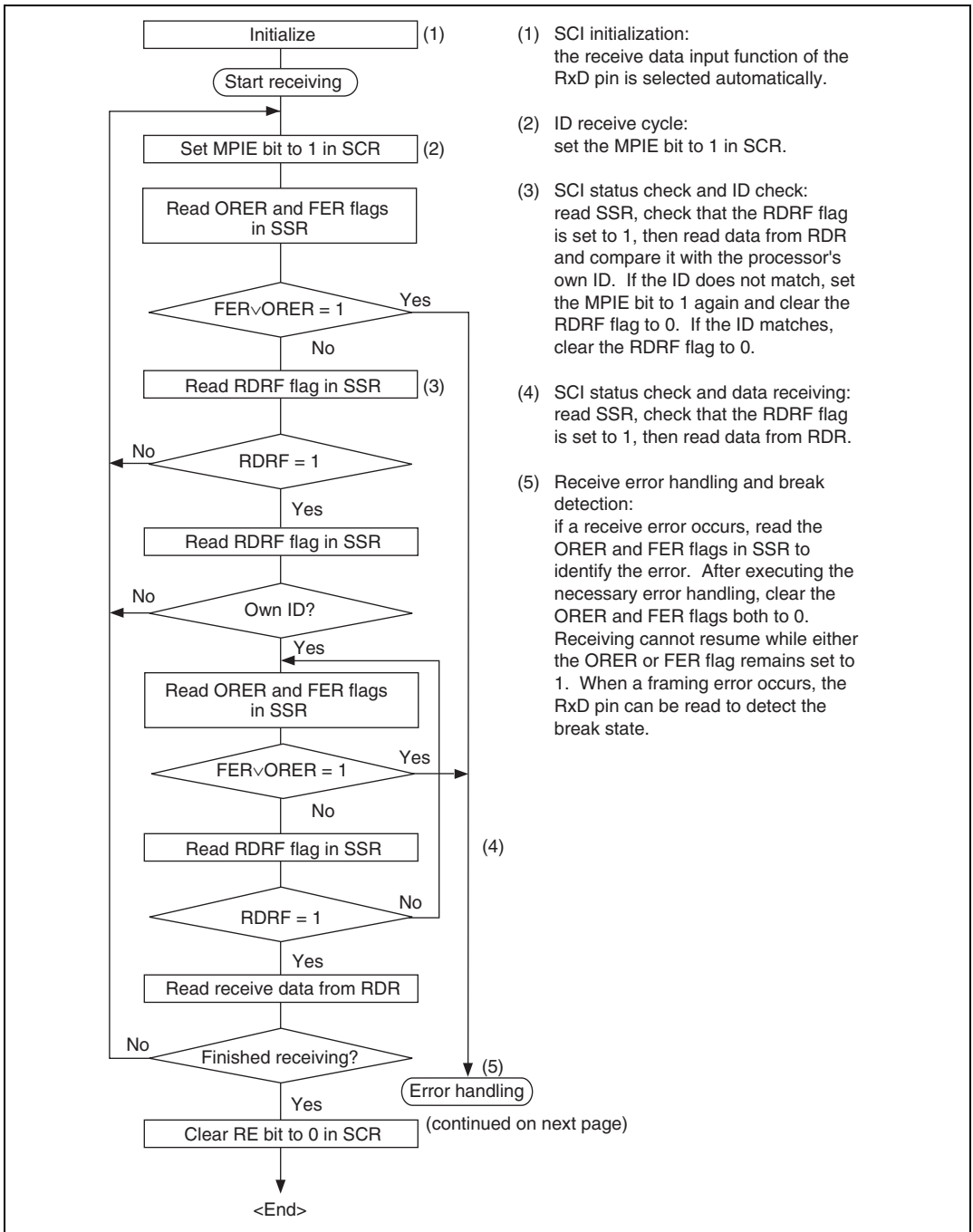


Figure 12.12 Sample Flowchart for Receiving Multiprocessor Serial Data (1)

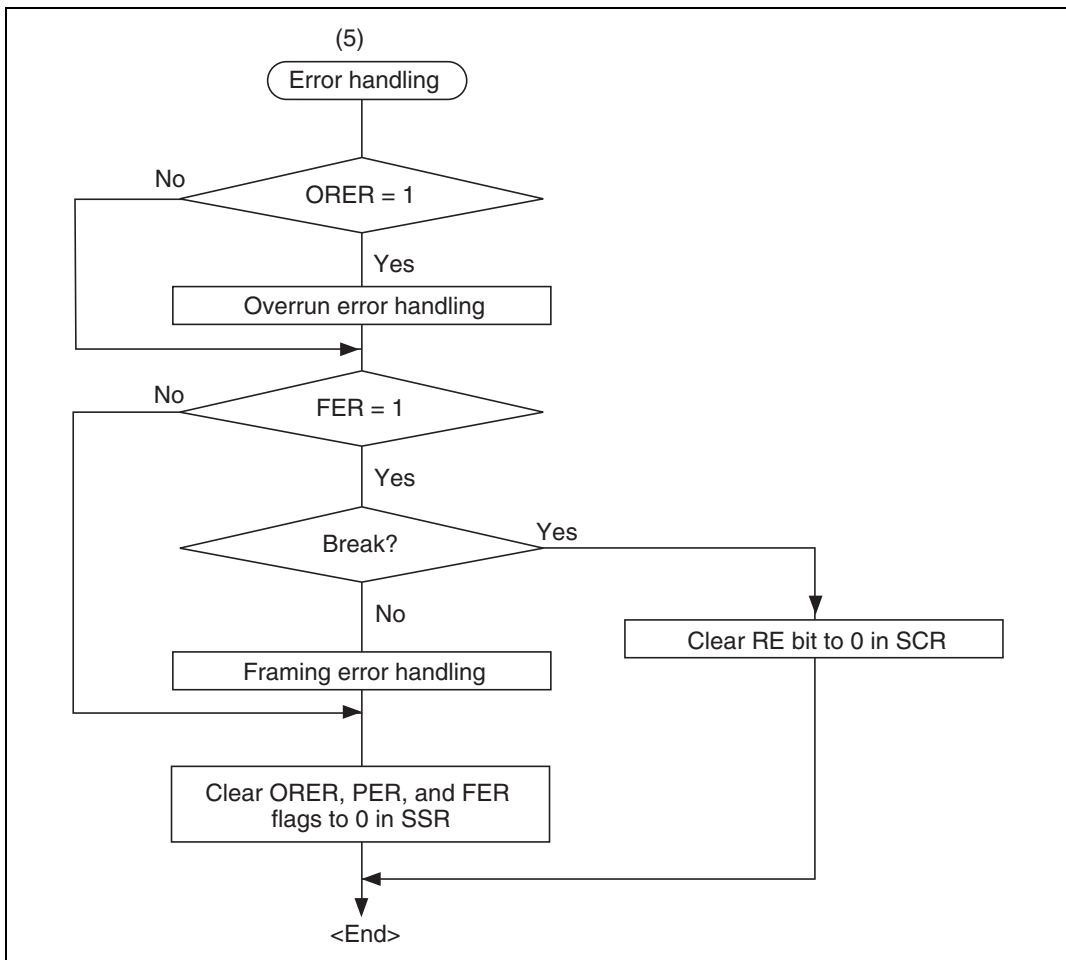


Figure 12.12 Sample Flowchart for Receiving Multiprocessor Serial Data (2)

Figure 12.13 shows an example of SCI receive operation using a multiprocessor format.

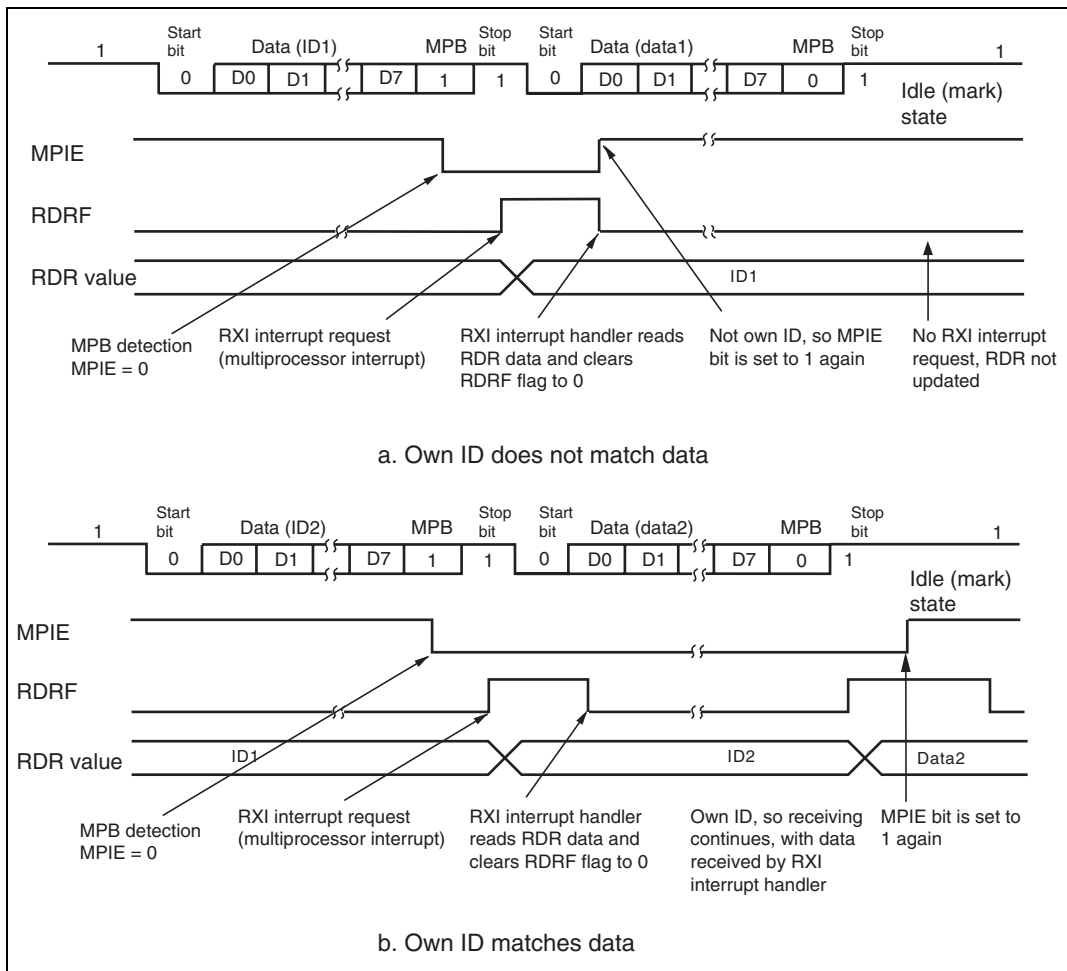


Figure 12.13 Example of SCI Receive Operation (8-Bit Data with Multiprocessor Bit and One Stop Bit)

12.3.4 Synchronous Operation

In synchronous mode, the SCI transmits and receives data in synchronization with clock pulses. This mode is suitable for high-speed serial communication.

The SCI transmitter and receiver share the same clock but are otherwise independent, so full-duplex communication is possible. The transmitter and the receiver are also double-buffered, so continuous transmitting or receiving is possible by reading or writing data while transmitting or receiving is in progress.

Figure 12.14 shows the general format in synchronous serial communication.

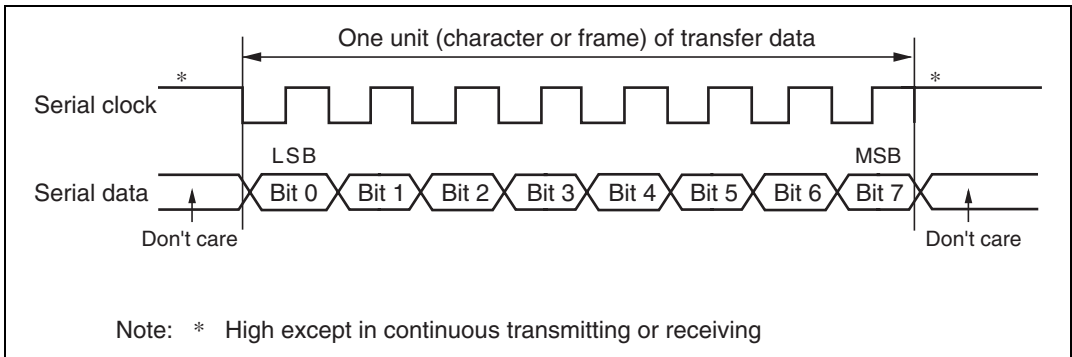


Figure 12.14 Data Format in Synchronous Communication

In synchronous serial communication, each data bit is placed on the communication line from one falling edge of the serial clock to the next. Data is guaranteed valid at the rise of the serial clock. In each character, the serial data bits are transferred in order from LSB (first) to MSB (last). After output of the MSB, the communication line remains in the state of the MSB. In synchronous mode the SCI receives data by synchronizing with the rise of the serial clock.

Communication Format: The data length is fixed at 8 bits. No parity bit or multiprocessor bit can be added.

Clock: An internal clock generated by the on-chip baud rate generator or an external clock input from the SCK pin can be selected by means of the C/A bit in SMR and the CKE1 and CKE0 bits in SCR. See table 12.6 for details of SCI clock source selection.

When the SCI operates on an internal clock, it outputs the clock source at the SCK pin. Eight clock pulses are output per transmitted or received character. When the SCI is not transmitting or receiving, the clock signal remains in the high state. If receiving in single-character units is required, an external clock should be selected.

Transmitting and Receiving Data:

- SCI Initialization (Synchronous Mode): Before transmitting or receiving data, clear the TE and RE bits to 0 in SCR, then initialize the SCI as follows.

When changing the communication mode or format, always clear the TE and RE bits to 0 before following the procedure given below. Clearing TE to 0 sets the TDRE flag to 1 and initializes TSR. Clearing RE to 0, however, does not initialize the RDRF, PER, FER, and ORER flags, or RDR, which retain their previous contents.

Figure 12.15 shows a sample flowchart for initializing the SCI.

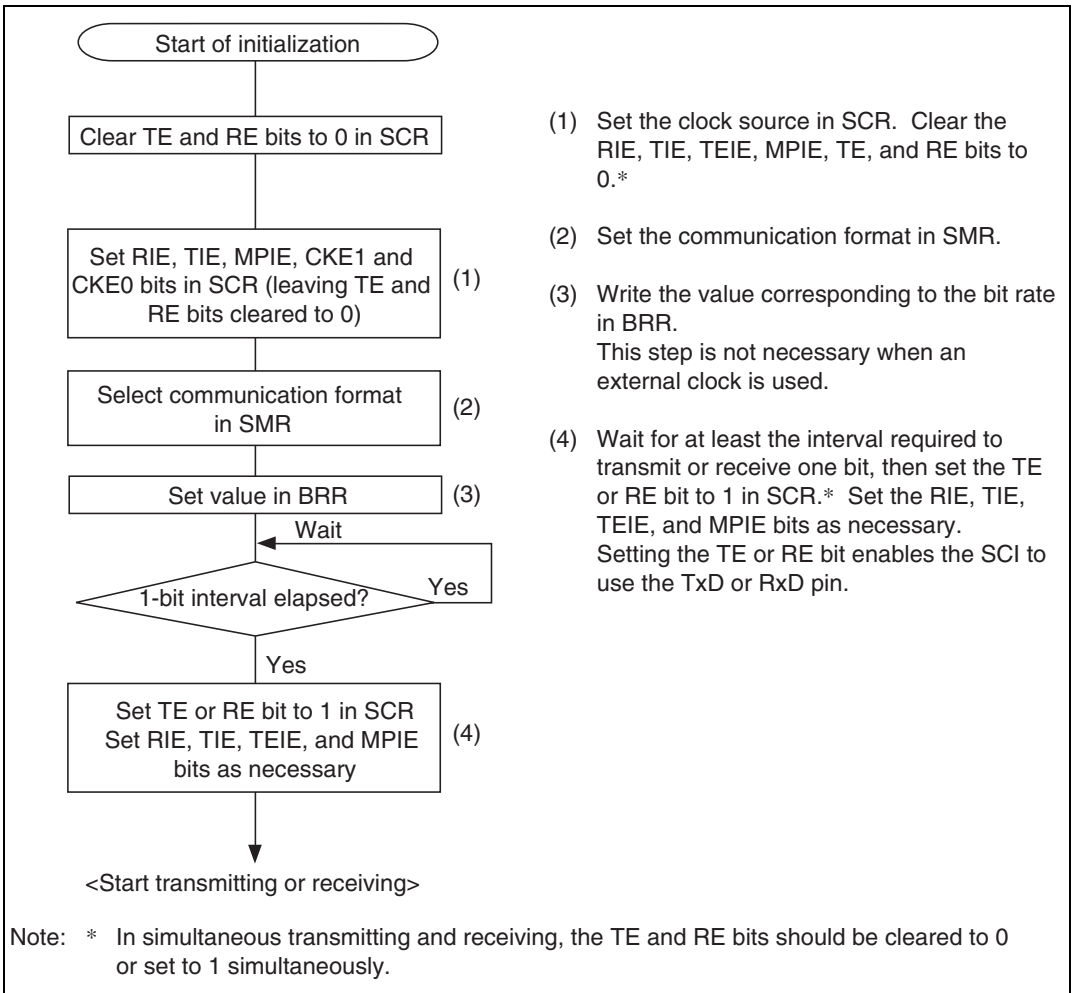


Figure 12.15 Sample Flowchart for SCI Initialization

- Transmitting Serial Data (Synchronous Mode): Figure 12.16 shows a sample flowchart for transmitting serial data and indicates the procedure to follow.

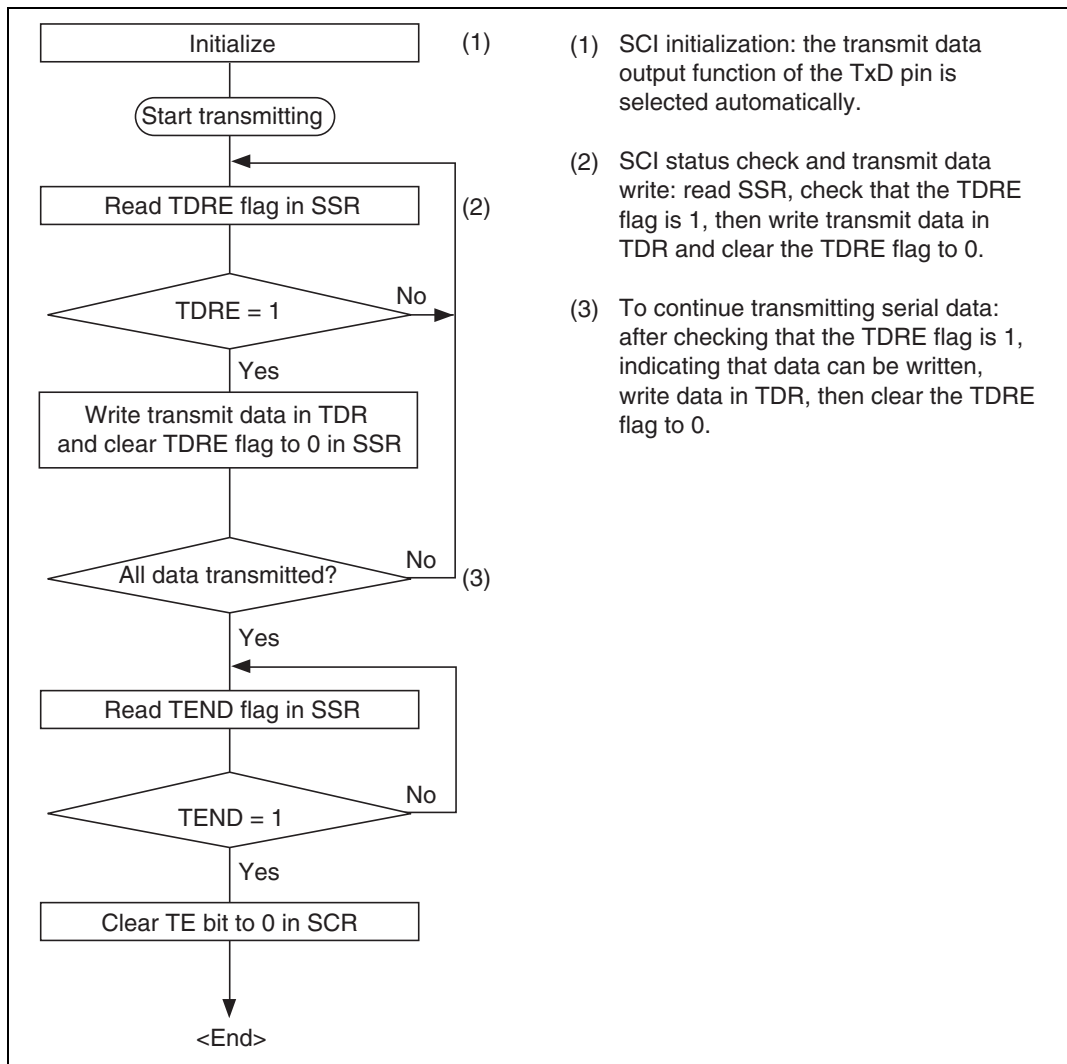


Figure 12.16 Sample Flowchart for Serial Transmitting

In transmitting serial data, the SCI operates as follows.

- The SCI monitors the TDRE flag in SSR. When the TDRE flag is cleared to 0, the SCI recognizes that TDR contains new data, and loads this data from TDR into TSR.
- After loading the data from TDR to TSR, the SCI sets the TDRE flag to 1 and starts transmitting. If the TIE bit is set to 1 in SCR, the SCI requests a transmit-data-empty interrupt (TXI) at this time.

If clock output is selected, the SCI outputs eight serial clock pulses. If an external clock source is selected, the SCI outputs data in synchronization with the input clock. Data is output from the TxD pin in order from LSB (bit 0) to MSB (bit 7).

- The SCI checks the TDRE flag when it outputs the MSB (bit 7). If the TDRE flag is 0, the SCI loads data from TDR into TSR and begins serial transmission of the next frame. If the TDRE flag is 1, the SCI sets the TEND flag to 1 in SSR, and after transmitting the MSB, holds the TxD pin in the MSB state. If the TEIE bit is set to 1 in SCR, a transmit-end interrupt (TEI) is requested at this time
- After the end of serial transmission, the SCK pin is held in a constant state.

Figure 12.17 shows an example of SCI transmit operation.

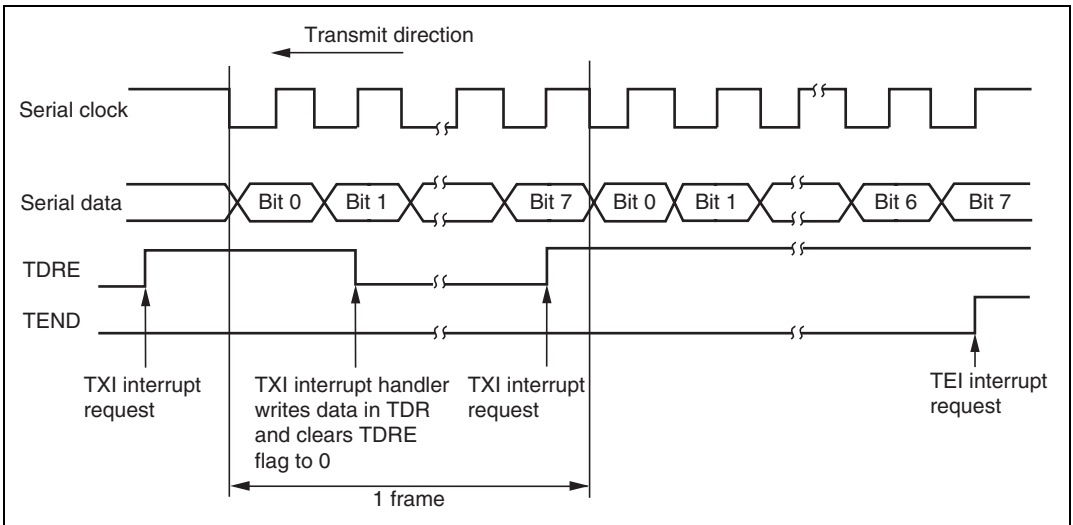


Figure 12.17 Example of SCI Transmit Operation

- Receiving Serial Data (Synchronous Mode): Figure 12.18 shows a sample flowchart for receiving serial data and indicates the procedure to follow. When switching from asynchronous to synchronous mode, make sure that the ORER, PER, and FER flags are cleared to 0. If the FER or PER flag is set to 1 the RDRF flag will not be set and both transmitting and receiving will be disabled.

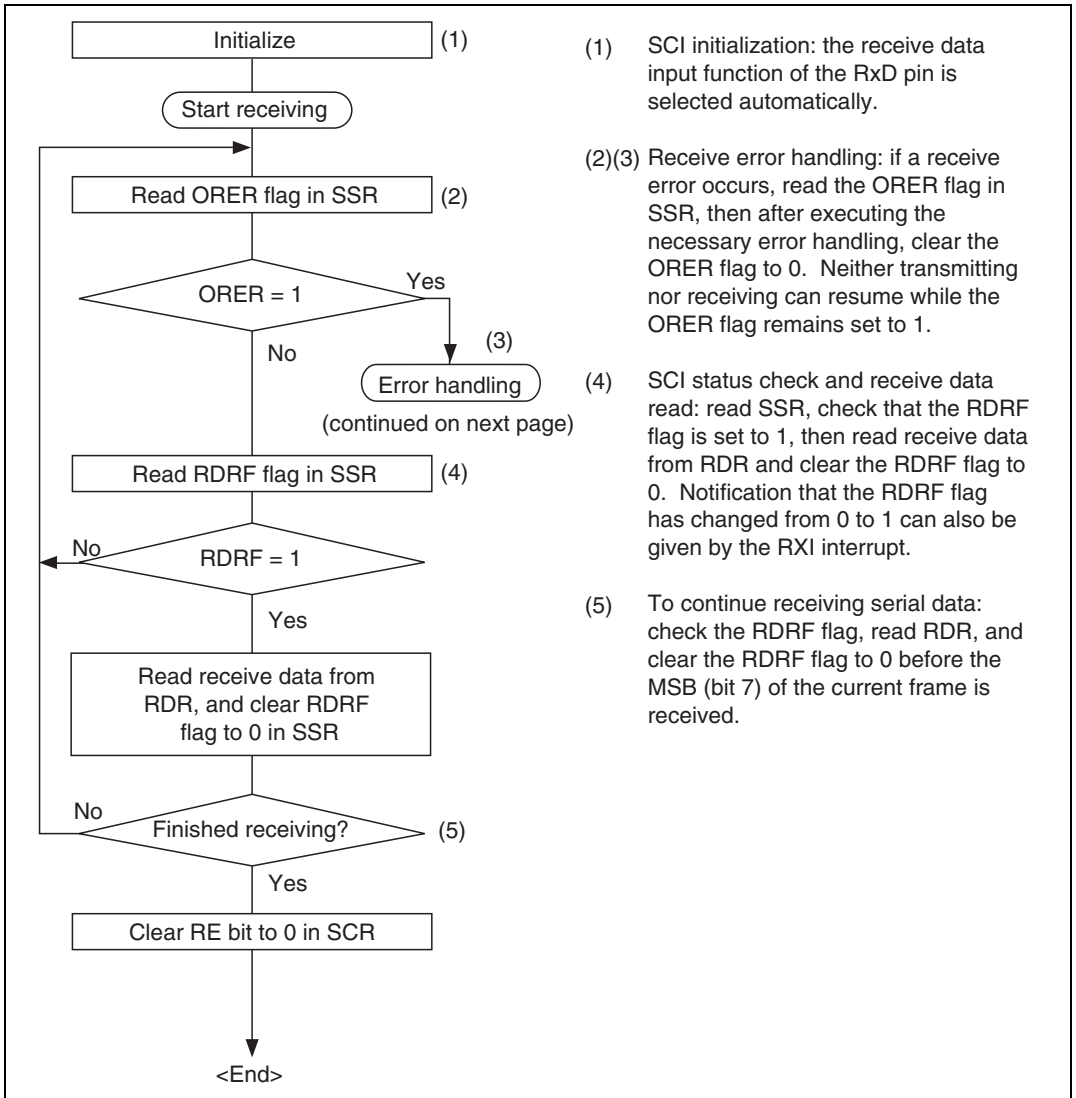


Figure 12.18 Sample Flowchart for Serial Receiving (1)

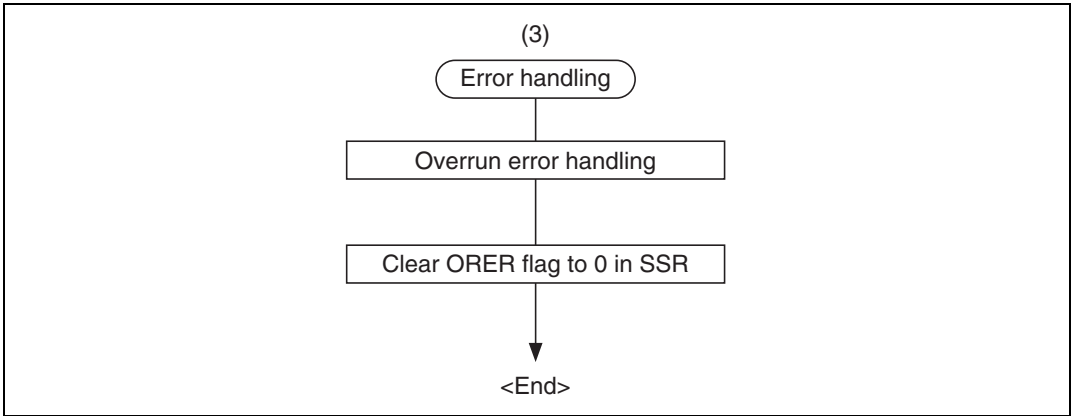


Figure 12.18 Sample Flowchart for Serial Receiving (2)

In receiving, the SCI operates as follows:

- The SCI synchronizes with serial clock input or output and synchronizes internally.
- Receive data is stored in RSR in order from LSB to MSB.
After receiving the data, the SCI checks that the RDRF flag is 0, so that receive data can be transferred from RSR to RDR. If this check passes, the RDRF flag is set to 1 and the received data is stored in RDR. If the checks fails (receive error), the SCI operates as shown in table 12.11.
When a receive error has been identified in the error check, subsequent transmit and receive operations are disabled.
- When the RDRF flag is set to 1, if the RIE bit is set to 1 in SCR, a receive-data-full interrupt (RXI) is requested. If the ORER flag is set to 1 and the RIE bit in SCR is also set to 1, a receive-error interrupt (ERI) is requested.

Figure 12.19 shows an example of SCI receive operation.

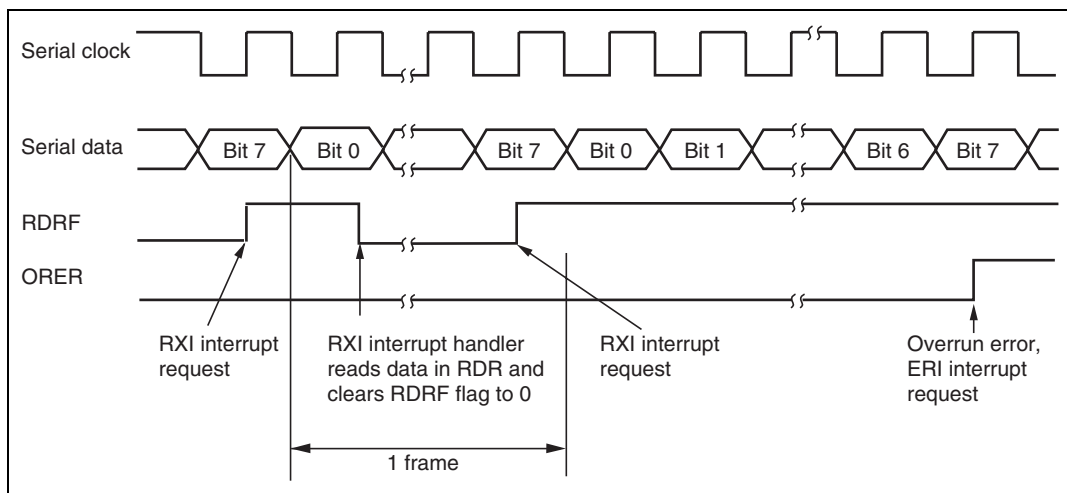


Figure 12.19 Example of SCI Receive Operation

- Transmitting and Receiving Data Simultaneously (Synchronous Mode): Figure 12.20 shows a sample flowchart for transmitting and receiving serial data simultaneously and indicates the procedure to follow.

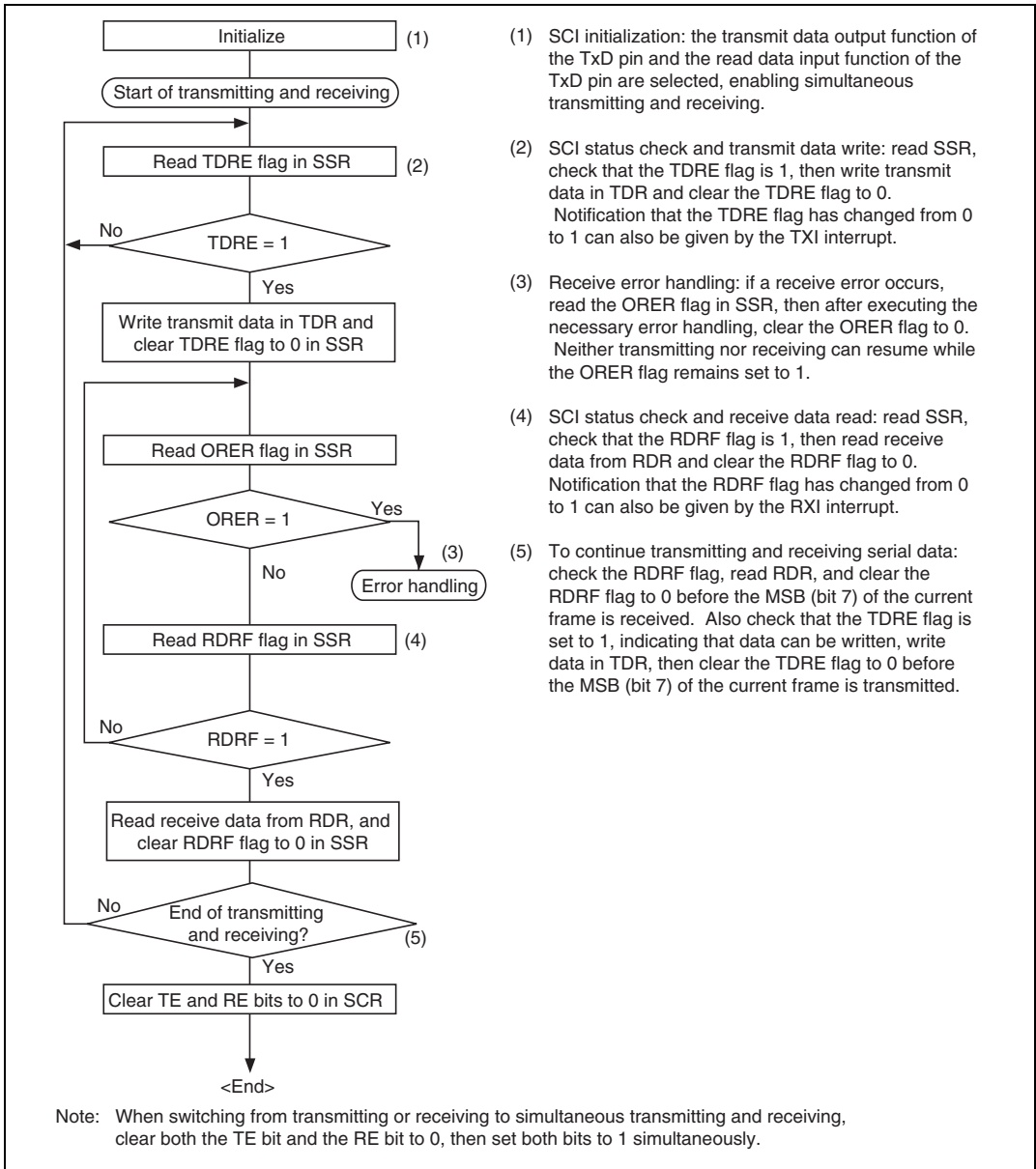


Figure 12.20 Sample Flowchart for Simultaneous Serial Transmitting and Receiving

12.4 SCI Interrupts

The SCI has four interrupt request sources: transmit-end interrupt (TEI), receive-error (ERI), receive-data-full (RXI), and transmit-data-empty interrupt (TXI). Table 12.12 lists the interrupt sources and indicates their priority. These interrupts can be enabled or disabled by the TIE, RIE, and TEIE bits in SCR. Each interrupt request is sent separately to the interrupt controller.

A TXI interrupt is requested when the TDRE flag is set to 1 in SSR. A TEI interrupt is requested when the TEND flag is set to 1 in SSR.

An RXI interrupt is requested when the RDRF flag is set to 1 in SSR. An ERI interrupt is requested when the ORER, PER, or FER flag is set to 1 in SSR.

Table 12.12 SCI Interrupt Sources

| Interrupt Source | Description | Priority |
|------------------|-------------------------------------|----------|
| ERI | Receive error (ORER, FER, or PER) | High |
| RXI | Receive data register full (RDRF) | ↑ |
| TXI | Transmit data register empty (TDRE) | |
| TEI | Transmit end (TEND) | |

12.5 Usage Notes

12.5.1 Notes on Use of SCI

Note the following points when using the SCI.

TDR Write and TDRE Flag: The TDRE flag in SSR is a status flag indicating the loading of transmit data from TDR to TSR. The SCI sets the TDRE flag to 1 when it transfers data from TDR to TSR.

Data can be written into TDR regardless of the state of the TDRE flag. If new data is written in TDR when the TDRE flag is 0, the old data stored in TDR will be lost because this data has not yet been transferred to TSR. Before writing transmit data in TDR, be sure to check that the TDRE flag is set to 1.

Simultaneous Multiple Receive Errors: Table 12.13 shows the state of the SSR status flags when multiple receive errors occur simultaneously. When an overrun error occurs the RSR contents are not transferred to RDR, so receive data is lost.

Table 12.13 SSR Status Flags and Transfer of Receive Data

| SSR Status Flags | | | | Receive Data Transfer | Receive Errors |
|------------------|------|-----|-----|-----------------------|--|
| RDRF | ORER | FER | PER | RSR → RDR | |
| 1 | 1 | 0 | 0 | × | Overrun error |
| 0 | 0 | 1 | 0 | ○ | Framing error |
| 0 | 0 | 0 | 1 | ○ | Parity error |
| 1 | 1 | 1 | 0 | × | Overrun error + framing error |
| 1 | 1 | 0 | 1 | × | Overrun error + parity error |
| 0 | 0 | 1 | 1 | ○ | Framing error + parity error |
| 1 | 1 | 1 | 1 | × | Overrun error + framing error + parity error |

Legend:

- : Receive data is transferred from RSR to RDR.
- ×: Receive data is not transferred from RSR to RDR.

Break Detection and Processing: Break signals can be detected by reading the RxD pin directly when a framing error (FER) is detected. In the break state the input from the RxD pin consists of all 0s, so the FER flag is set and the parity error flag (PER) may also be set. In the break state the SCI receiver continues to operate, so if the FER flag is cleared to 0 it will be set to 1 again.

Sending a Break Signal: The input/output condition and level of the TxD pin are determined by DR and DDR bits. This feature can be used to send a break signal.

After the serial transmitter is initialized, the DR value substitutes for the mark state until the TE bit is set to 1 (the TxD pin function is not selected until the TE bit is set to 1). The DDR and DR bits should therefore be set to 1 beforehand.

To send a break signal during serial transmission, clear the DR bit to 0, then clear the TE bit to 0. When the TE bit is cleared to 0 the transmitter is initialized, regardless of its current state, so the TxD pin becomes an input/output outputting the value 0.

Receive Error Flags and Transmitter Operation (Synchronous Mode Only): When a receive error flag (ORER, PER, or FER) is set to 1 the SCI will not start transmitting, even if the TDRE flag is cleared to 0. Be sure to clear the receive error flags to 0 when starting to transmit. Note that clearing the RE bit to 0 does not clear the receive error flags to 0.

Receive Data Sampling Timing in Asynchronous Mode and Receive Margin: In asynchronous mode the SCI operates on a base clock with 16 times the bit rate frequency. In receiving, the SCI synchronizes internally with the fall of the start bit, which it samples on the base clock. Receive data is latched at the rising edge of the eighth base clock pulse. See figure 12.21.

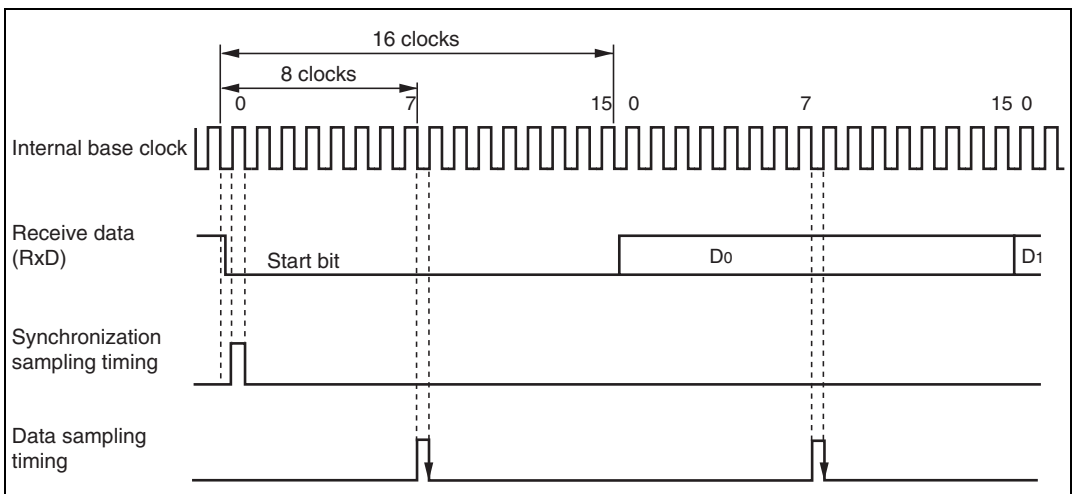


Figure 12.21 Receive Data Sampling Timing in Asynchronous Mode

The receive margin in asynchronous mode can therefore be expressed as shown in equation (1).

$$M = \left| \left(0.5 - \frac{1}{2N} \right) - (L - 0.5) F - \frac{|D - 0.5|}{N} (1 + F) \right| \times 100\% \quad \dots \dots \dots (1)$$

Legend:

- M: Receive margin (%)
- N: Ratio of clock frequency to bit rate (N = 16)
- D: Clock duty cycle (D = 0 to 1.0)
- L: Frame length (L = 9 to 12)
- F: Absolute deviation of clock frequency

From equation (1), if F = 0 and D = 0.5, the receive margin is 46.875%, as given by equation (2).

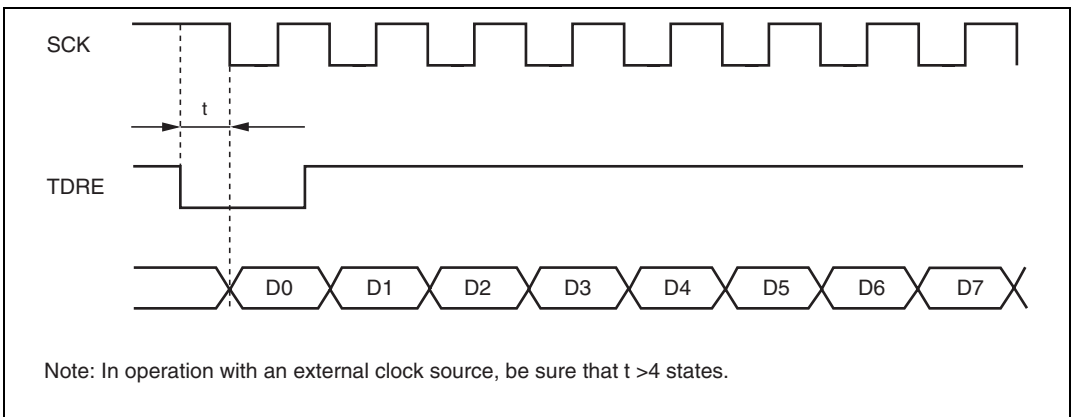
When D = 0.5 and F = 0:

$$M = \left(0.5 - \frac{1}{2 \times 16} \right) \times 100\% = 46.875\% \quad \dots \dots \dots (2)$$

This is a theoretical value. A reasonable margin to allow in system designs is 20% to 30%.

Restrictions on Use of an External Clock Source:

- When an external clock source is used for the serial clock, after updates TDR, allow an inversion of at least five system clock (ϕ) cycles before input of the serial clock to start transmitting. If the serial clock is input within four states of the TDR update, a malfunction may occur. (See figure 12.22)



Note: In operation with an external clock source, be sure that t > 4 states.

Figure 12.22 Example of Synchronous Transmission

Switching from SCK Pin Function to Port Pin Function:

- Problem in Operation: When switching the SCK pin function to the output port function (high-level output) by making the following settings while $DDR = 1$, $DR = 1$, $C/\bar{A} = 1$, $CKE1 = 0$, $CKE0 = 0$, and $TE = 1$ (synchronous mode), low-level output occurs for one half-cycle.
1. End of serial data transmission
 2. TE bit = 0
 3. C/\bar{A} bit = 0 ... switchover to port output
 4. Occurrence of low-level output (see figure 12.23)

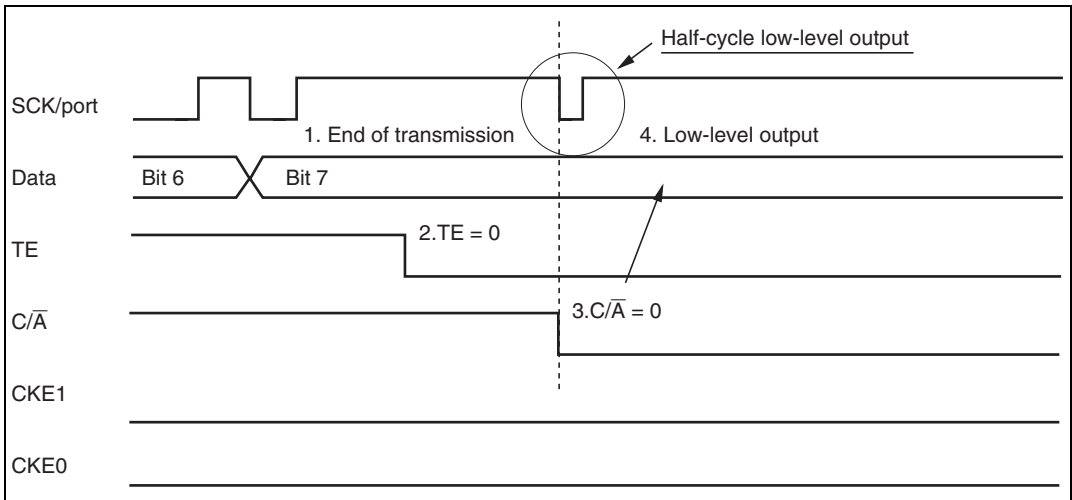


Figure 12.23 Operation when Switching from SCK Pin Function to Port Pin Function

- Sample Procedure for Avoiding Low-Level Output: As this sample procedure temporarily places the SCK pin in the input state, the SCK/port pin should be pulled up beforehand with an external circuit.

With $DDR = 1$, $DR = 1$, $C/\bar{A} = 1$, $CKE1 = 0$, $CKE0 = 0$, and $TE = 1$, make the following settings in the order shown.

1. End of serial data transmission
2. TE bit = 0
3. CKE1 bit = 1
4. C/\bar{A} bit = 0 ... switchover to port output
5. CKE1 bit = 0

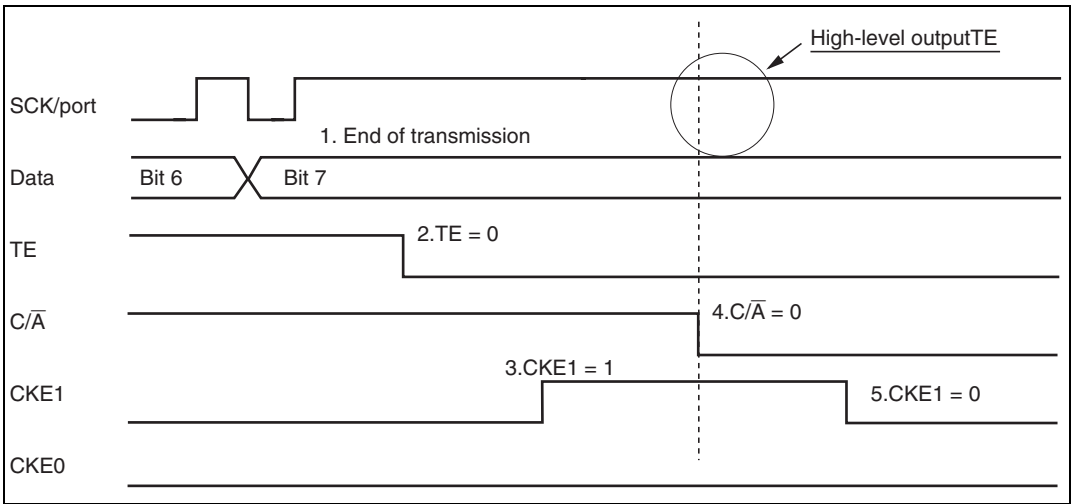


Figure 12.24 Operation when Switching from SCK Pin Function to Port Pin Function (Example of Preventing Low-Level Output)

Section 13 Smart Card Interface

13.1 Overview

The SCI supports an IC card (smart card) interface handling ISO/IEC7816-3 (Identification Card) character transmission as a serial communication interface expansion function.

Switchover between the normal serial communication interface and the smart card interface is controlled by a register setting.

13.1.1 Features

Features of the smart card interface supported by the H8/3008 are listed below.

- Asynchronous communication
 - Data length: 8 bits
 - Parity bit generation and checking
 - Transmission of error signal (parity error) in receive mode
 - Error signal detection and automatic data retransmission in transmit mode
 - Direct convention and inverse convention both supported
- Built-in baud rate generator allows any bit rate to be selected
- Three interrupt sources
 - There are three interrupt sources—transmit-data-empty, receive-data-full, and transmit/receive error—that can issue requests independently.

13.1.2 Block Diagram

Figure 13.1 shows a block diagram of the smart card interface.

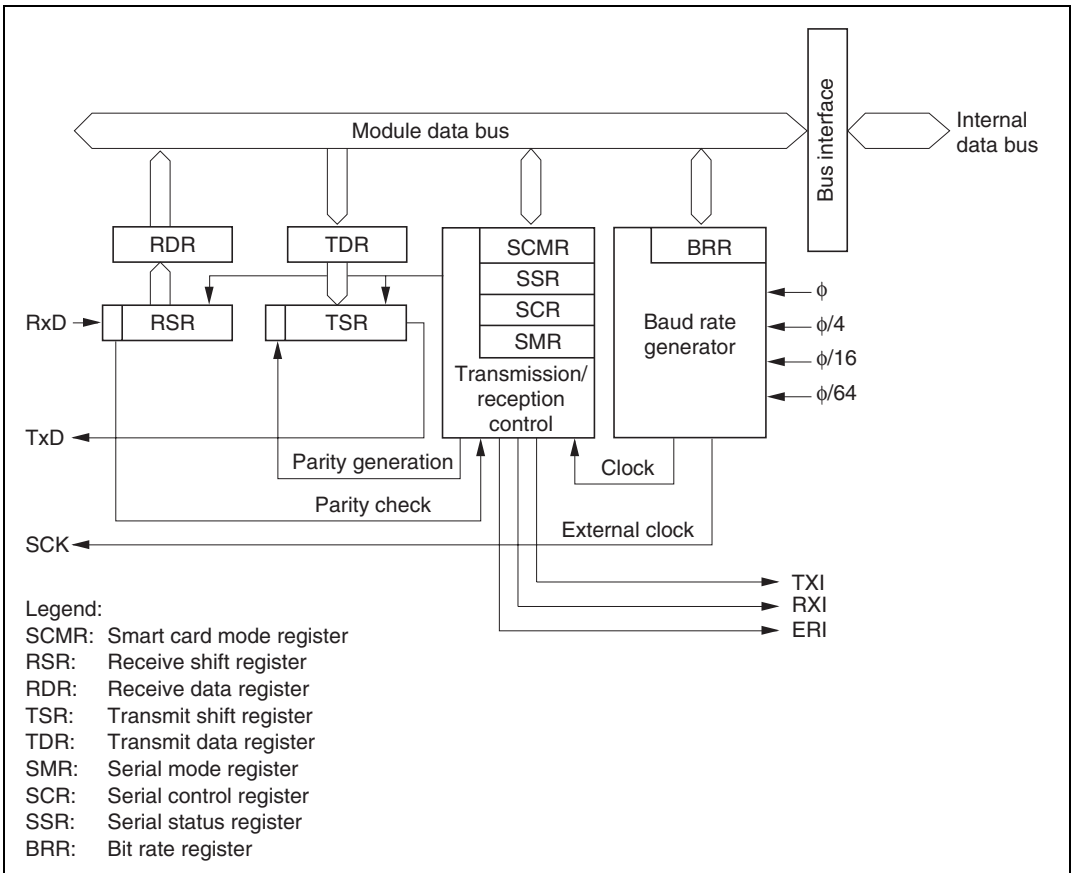


Figure 13.1 Block Diagram of Smart Card Interface

13.1.3 Pin Configuration

Table 13.1 shows the smart card interface pins.

Table 13.1 Smart Card Interface Pins

| Pin Name | Abbreviation | I/O | Function |
|-------------------|--------------|--------|----------------------|
| Serial clock pin | SCK | I/O | Clock input/output |
| Receive data pin | RxD | Input | Receive data input |
| Transmit data pin | TxD | Output | Transmit data output |

13.1.4 Register Configuration

The smart card interface has the internal registers listed in table 13.2. The BRR, TDR, and RDR registers have their normal serial communication interface functions, as described in section 12, Serial Communication Interface.

Table 13.2 Smart Card Interface Registers

| Channel | Address* ¹ | Name | Abbreviation | R/W | Initial Value |
|---------|-----------------------|--------------------------|--------------|---------------------|---------------|
| 0 | H'FFFB0 | Serial mode register | SMR | R/W | H'00 |
| | H'FFFB1 | Bit rate register | BRR | R/W | H'FF |
| | H'FFFB2 | Serial control register | SCR | R/W | H'00 |
| | H'FFFB3 | Transmit data register | TDR | R/W | H'FF |
| | H'FFFB4 | Serial status register | SSR | R/(W)* ² | H'84 |
| | H'FFFB5 | Receive data register | RDR | R | H'00 |
| | H'FFFB6 | Smart card mode register | SCMR | R/W | H'F2 |
| 1 | H'FFFB8 | Serial mode register | SMR | R/W | H'00 |
| | H'FFFB9 | Bit rate register | BRR | R/W | H'FF |
| | H'FFFB A | Serial control register | SCR | R/W | H'00 |
| | H'FFFB B | Transmit data register | TDR | R/W | H'FF |
| | H'FFFB C | Serial status register | SSR | R/(W)* ² | H'84 |
| | H'FFFB D | Receive data register | RDR | R | H'00 |
| | H'FFFB E | Smart card mode register | SCMR | R/W | H'F2 |

- Notes: 1. Lower 20 bits of the address in advanced mode.
 2. Only 0 can be written in bits 7 to 3, to clear the flags.

13.2 Register Descriptions

This section describes the new or modified registers and bit functions in the smart card interface.

13.2.1 Smart Card Mode Register (SCMR)

SCMR is an 8-bit readable/writable register that selects smart card interface functions.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|---|---|---|---|------|------|---|------|
| | — | — | — | — | SDIR | SINV | — | SMIF |
| Initial value | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 |
| Read/Write | — | — | — | — | R/W | R/W | — | R/W |

Reserved bits (Bits 7, 6, 5, 4)
Reserved bit (Bit 1)
Smart card interface mode select
 Enables or disables the smart card interface function
Smart card data invert
 Inverts data logic levels
Smart card data transfer direction
 Selects the serial/parallel conversion format

SCMR is initialized to H'F2 by a reset and in standby mode.

Bits 7 to 4—Reserved: Read-only bits, always read as 1.

Bit 3—Smart Card Data Transfer Direction (SDIR): Selects the serial/parallel conversion format.*1

Bit 3

| SDIR | Description |
|------|--|
| 0 | TDR contents are transmitted LSB-first Receive data is stored LSB-first in RDR (Initial value) |
| 1 | TDR contents are transmitted MSB-first Receive data is stored MSB-first in RDR |

Bit 2—Smart Card Data Invert (SINV): Specifies inversion of the data logic level. This function is used in combination with the SDIR bit to communicate with inverse-convention cards.*2 The SINV bit does not affect the logic level of the parity bit. For parity settings, see section 13.3.4, Register Settings.

| Bit 2 SINV | Description | |
|---------------|---|-----------------|
| 0 | Unmodified TDR contents are transmitted Receive data is stored unmodified in RDR | (Initial value) |
| 1 | Inverted TDR contents are transmitted Receive data is inverted before storage in RDR | |

Bit 1—Reserved: Read-only bit, always read as 1.

Bit 0—Smart Card Interface Mode Select (SMIF): Enables the smart card interface function.

| Bit 0 SMIF | Description | |
|---------------|---|-----------------|
| 0 | Smart card interface function is disabled | (Initial value) |
| 1 | Smart card interface function is enabled | |

- Notes:
1. The function for switching between LSB-first and MSB-first mode can also be used with the normal serial communication interface. Note that when the communication format data length is set to 7 bits and MSB-first mode is selected for the serial data to be transferred, bit 0 of TDR is not transmitted, and only bits 7 to 1 of the received data are valid.
 2. The data logic level inversion function can also be used with the normal serial communication interface. Note that, when inverting the serial data to be transferred, parity transmission and parity checking is based on the number of high-level periods at the serial data I/O pin, and not on the register value.

13.2.2 Serial Status Register (SSR)

The function of SSR bit 4 is modified in smart card interface mode. This change also causes a modification to the setting conditions for bit 2 (TEND).

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|--------|--------|--------|--------|--------|------|-----|------|
| | TDRE | RDRF | ORER | ERS | PER | TEND | MPB | MPBT |
| Initial value | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| Read/Write | R/(W)* | R/(W)* | R/(W)* | R/(W)* | R/(W)* | R | R | R/W |

Error signal status (ERS)
 Status flag indicating that an error signal has been received

Transmit end
 Status flag indicating end of transmission

Note: * Only 0 can be written, to clear the flag.

Bits 7 to 5: These bits operate as in normal serial communication. For details see section 12.2.7, Serial Status Register (SSR).

Bit 4—Error Signal Status (ERS): In smart card interface mode, this flag indicates the status of the error signal sent from the receiving device to the transmitting device. The smart card interface does not detection framing errors.

Bit 4

| ERS | Description |
|-----|--|
| 0 | Indicates normal transmission, with no error signal returned (Initial value) [Clearing conditions] <ul style="list-style-type: none"> • The chip is reset, or enters standby mode or module stop mode • Software reads ERS while it is set to 1, then writes 0. |
| 1 | Indicates that the receiving device sent an error signal reporting a parity error [Setting condition] A low error signal was sampled. |

Note: Clearing the TE bit to 0 in SCR does not affect the ERS flag, which retains its previous value.

Bits 3 to 0: These bits operate as in normal serial communication. For details see section 12.2.7, Serial Status Register (SSR). The setting conditions for transmit end (TEND), however, are modified as follows.

| Bit | Description |
|-----|---|
| 0 | Transmission is in progress [Clearing condition] Software reads TDRE while it is set to 1, then writes 0 in the TDRE flag. |
| 1 | End of transmission [Setting conditions] (Initial value) <ul style="list-style-type: none"> The chip is reset or enters standby mode. The TE bit and FER/ERS bit are both cleared to 0 in SCR. TDRE is 1 and FER/ERS is 0 at a time 2.5 etu after the last bit of a 1-byte serial character is transmitted (normal transmission). |

Note: An etu (elementary time unit) is the time needed to transmit one bit.

13.2.3 Serial Mode Register (SMR)

The function of SMR bit 7 is modified in smart card interface mode. This change also causes a modification to the function of bits 1 and 0 in the serial control register (SCR).

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|-----|-----|-----|--------------|------|-----|------|------|
| | GM | CHR | PE | O/ \bar{E} | STOP | MP | CKS1 | CKS0 |
| Initial value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Bit 7—GSM Mode (GM): With the normal smart card interface, this bit is cleared to 0. Setting this bit to 1 selects GSM mode, an additional mode for controlling the timing for setting the TEND flag that indicates completion of transmission, and the type of clock output used. The details of the additional clock output control mode are specified by the CKE1 and CKE0 bits in the serial control register (SCR).

Bit 7

| GM | Description |
|-----------|---|
| 0 | Normal smart card interface mode operation <ul style="list-style-type: none"> The TEND flag is set 12.5 etu after the beginning of the start bit. Clock output on/off control only. (Initial value) |
| 1 | GSM mode smart card interface mode operation <ul style="list-style-type: none"> The TEND flag is set 11.0 etu after the beginning of the start bit. Clock output on/off and fixed-high/fixed-low control. |

Bits 6 to 0: These bits operate as in normal serial communication. For details see section 12.2.5, Serial Mode Register (SMR).

13.2.4 Serial Control Register (SCR)

The function of SCR bits 1 and 0 is modified in smart card interface mode.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|-----|-----|-----|-----|------|------|------|------|
| | TIE | RIE | TE | RE | MPIE | TEIE | CKE1 | CKE0 |
| Initial value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Bits 7 to 2: These bits operate as in normal serial communication. For details see section 12.2.6, Serial Control Register (SCR).

Bits 1 and 0—Clock Enable 1 and 0 (CKE1, CKE0): These bits select the SCI clock source and enable or disable clock output from the SCK pin. In smart card interface mode, it is possible to specify a fixed high level or fixed low level for the clock output, in addition to the usual switching between enabling and disabling of the clock output.

| Bit 7 GM | Bit 1 CKE1 | Bit 0 CKE0 | Description |
|---------------------|-----------------------|-----------------------|--|
| 0 | 0 | 0 | Internal clock/SCK pin is I/O port (Initial value) |
| | | 1 | Internal clock/SCK pin is clock output |
| 1 | 0 | 0 | Internal clock/SCK pin is fixed at low output |
| | | 1 | Internal clock/SCK pin is clock output |
| | 1 | 0 | Internal clock/SCK pin is fixed at high output |
| | | 1 | Internal clock/SCK pin is clock output |

13.3 Operation

13.3.1 Overview

The main features of the smart card interface are as follows.

- One frame consists of 8-bit data plus a parity bit.
- In transmission, a guard time of at least 2 etu (elementary time units: the time for transfer of one bit) is provided between the end of the parity bit and the start of the next frame.
- If a parity error is detected during reception, a low error signal level is output for 1 etu period 10.5 etu after the start bit.
- If an error signal is detected during transmission, the same data is transmitted automatically after the elapse of 2 etu or longer.
- Only asynchronous communication is supported; there is no synchronous communication function.

13.3.2 Pin Connections

Figure 13.2 shows a pin connection diagram for the smart card interface.

In communication with a smart card, since both transmission and reception are carried out on a single data transmission line, the TxD pin and RxD pin should both be connected to this line. The data transmission line should be pulled up to V_{cc} with a resistor.

When the smart card uses the clock generated on the smart card interface, the SCK pin output is input to the CLK pin of the smart card. If the smart card uses an internal clock, this connection is unnecessary.

The reset signal should be output from one of the H8/3008's generic ports.

In addition to these pin connections, power and ground connections will normally also be necessary.

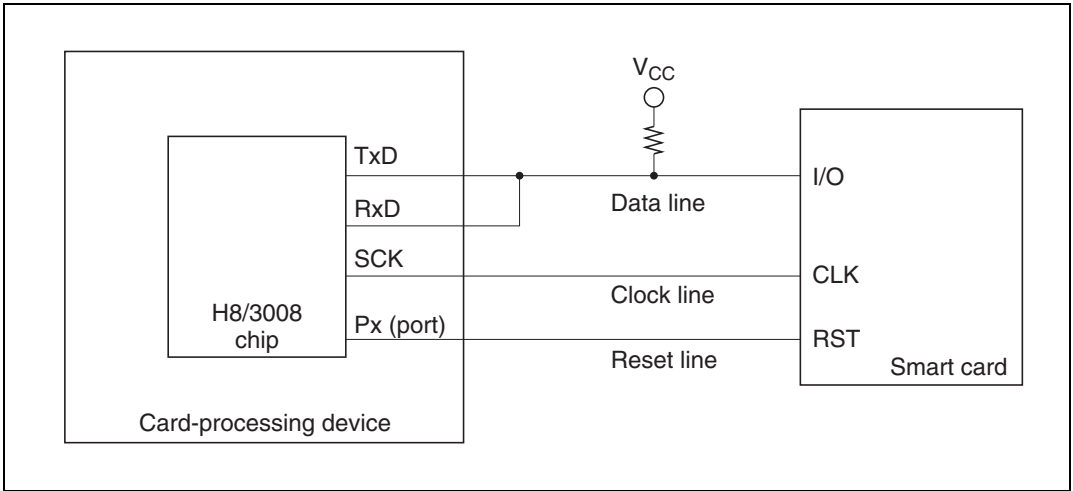


Figure 13.2 Smart Card Interface Connection Diagram

Note: Setting both TE and RE to 1 without connecting a smart card enables closed transmission/reception, allowing self-diagnosis to be carried out.

13.3.3 Data Format

Figure 13.3 shows the smart card interface data format. In reception in this mode, a parity check is carried out on each frame, and if an error is detected an error signal is sent back to the transmitting device to request retransmission of the data. In transmission, the error signal is sampled and the same data is retransmitted.

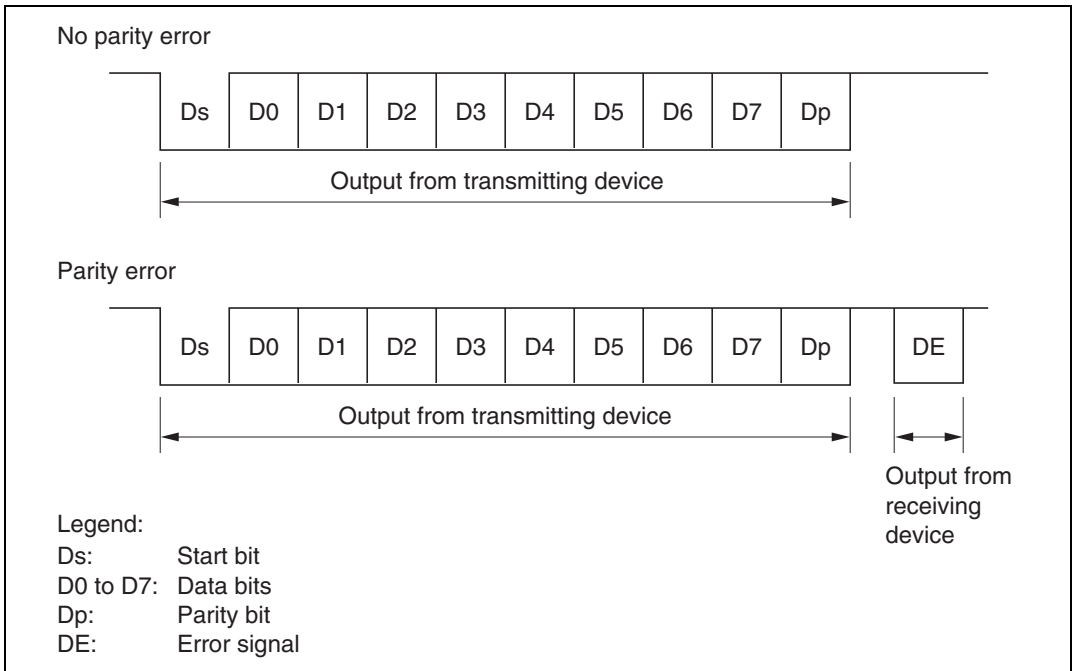


Figure 13.3 Smart Card Interface Data Format

The operating sequence is as follows.

1. When the data line is not in use it is in the high-impedance state, and is fixed high with a pull-up resistor.
2. The transmitting device starts transfer of one frame of data. The data frame starts with a start bit (Ds, low-level), followed by 8 data bits (D0 to D7) and a parity bit (Dp).
3. With the smart card interface, the data line then returns to the high-impedance state. The data line is pulled high with a pull-up resistor.
4. The receiving device carries out a parity check. If there is no parity error and the data is received normally, the receiving device waits for reception of the next data. If a parity error occurs, however, the receiving device outputs an error signal (DE, low-level) to request retransmission of the data. After outputting the error signal for the prescribed length of time, the receiving device places the signal line in the high-impedance state again. The signal line is pulled high again by a pull-up resistor.
5. If the transmitting device does not receive an error signal, it proceeds to transmit the next data frame. If it receives an error signal, however, it returns to step 2 and transmits the same data again.

13.3.4 Register Settings

Table 13.3 shows a bit map of the registers used in the smart card interface. Bits indicated as 0 or 1 must be set to the value shown. The setting of other bits is described in this section.

Table 13.3 Smart Card Interface Register Settings

| Register | Address* ¹ | Bit | | | | | | | |
|----------|-----------------------|-------|-------|-------|-------|-------|-------|--------------------|-------|
| | | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| SMR | H'FFFB0 | GM | 0 | 1 | O/E | 1 | 0 | CKS1 | CKS0 |
| BRR | H'FFFB1 | BRR7 | BRR6 | BRR5 | BRR4 | BRR3 | BRR2 | BRR1 | BRR0 |
| SCR | H'FFFB2 | TIE | RIE | TE | RE | 0 | 0 | CKE1* ² | CKE0 |
| TDR | H'FFFB3 | TDR7 | TDR6 | TDR5 | TDR4 | TDR3 | TDR2 | TDR1 | TDR0 |
| SSR | H'FFFB4 | TDRE | RDRF | ORER | ERS | PER | TEND | 0 | 0 |
| RDR | H'FFFB5 | RDR7 | RDR6 | RDR5 | RDR4 | RDR3 | RDR2 | RDR1 | RDR0 |
| SCMR | H'FFFB6 | — | — | — | — | SDIR | SINV | — | SMIF |

Legend:

—: Unused bit.

Notes: 1. Lower 20 bits of the address in advanced mode.

2. When GM is cleared to 0 in SMR, the CKE1 bit must also be cleared to 0.

Serial Mode Register (SMR) Settings: Clear the GM bit to 0 when using the normal smart card interface mode, or set to 1 when using GSM mode. Clear the $\overline{O/E}$ bit to 0 if the smart card is of the direct convention type, or set to 1 if of the inverse convention type.

Bits CKS1 and CKS0 select the clock source of the built-in baud rate generator. See section 13.3.5, Clock.

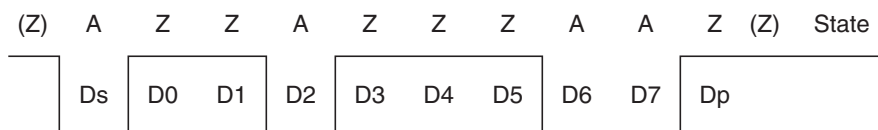
Bit Rate Register (BRR) Settings: BRR is used to set the bit rate. See section 13.3.5, Clock, for the method of calculating the value to be set.

Serial Control Register (SCR) Settings: The TIE, RIE, TE, and RE bits have their normal serial communication functions. See section 12, Serial Communication Interface, for details. The CKE1 and CKE0 bits specify clock output. To disable clock output, clear these bits to 00; to enable clock output, set these bits to 01. Clock output is performed when the GM bit is set to 1 in SMR. Clock output can also be fixed low or high.

Smart Card Mode Register (SCMR) Settings: Clear both the SDIR bit and SINV bit cleared to 0 if the smart card is of the direct convention type, and set both to 1 if of the inverse convention type. To use the smart card interface, set the SMIF bit to 1.

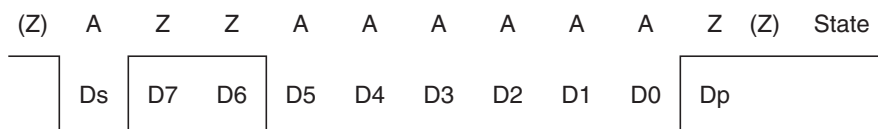
The register settings and examples of starting character waveforms are shown below for two smart cards, one following the direct convention and one the inverse convention.

1. Direct Convention ($SDIR = SINV = O/\bar{E} = 0$)



With the direct convention type, the logic 1 level corresponds to state Z and the logic 0 level to state A, and transfer is performed in LSB-first order. In the example above, the first character data is H'3B. The parity bit is 1, following the even parity rule designated for smart cards.

2. Inverse Convention ($SDIR = SINV = O/\bar{E} = 1$)



With the inverse convention type, the logic 1 level corresponds to state A and the logic 0 level to state Z, and transfer is performed in MSB-first order. In the example above, the first character data is H'3F. The parity bit is 0, corresponding to state Z, following the even parity rule designated for smart cards.

In the H8/3008, inversion specified by the SINV bit applies only to the data bits, D7 to D0. For parity bit inversion, the O/\bar{E} bit in SMR must be set to odd parity mode. This applies to both transmission and reception.

13.3.5 Clock

Only an internal clock generated by the on-chip baud rate generator can be used as the transmit/receive clock for the smart card interface. The bit rate is set with the bit rate register (BRR) and the CKS1 and CKS0 bits in the serial mode register (SMR). The equation for calculating the bit rate is shown below. Table 13.5 shows some sample bit rates.

If clock output is selected with CKE0 set to 1, a clock with a frequency of 372 times the bit rate is output from the SCK pin.

$$B = \frac{\phi}{1488 \times 2^{2n-1} \times (N + 1)} \times 10^6$$

where, N: BRR setting ($0 \leq N \leq 255$)

B: Bit rate (bit/s)

ϕ : Operating frequency (MHz)

n: See table 13.4

Table 13.4 n-Values of CKS1 and CKS0 Settings

| n | CKS1 | CKS0 |
|---|------|------|
| 0 | 0 | 0 |
| 1 | | 1 |
| 2 | 1 | 0 |
| 3 | | 1 |

Note: If the gear function is used to divide the clock frequency, use the divided frequency to calculate the bit rate. The equation above applies directly to 1/1 frequency division.

Table 13.5 Bit Rates (bits/s) for Various BRR Settings (When n = 0)

| N | ϕ (MHz) | | | | | | | | |
|---|--------------|---------|---------|---------|---------|---------|---------|---------|---------|
| | 7.1424 | 10.00 | 10.7136 | 13.00 | 14.2848 | 16.00 | 18.00 | 20.00 | 25.00 |
| 0 | 9600.0 | 13440.9 | 14400.0 | 17473.1 | 19200.0 | 21505.4 | 24193.5 | 26881.7 | 33602.2 |
| 1 | 4800.0 | 6720.4 | 7200.0 | 8736.6 | 9600.0 | 10752.7 | 12096.8 | 13440.9 | 16801.1 |
| 2 | 3200.0 | 4480.3 | 4800.0 | 5824.4 | 6400.0 | 7168.5 | 8064.5 | 8960.6 | 11200.7 |

Note: Bit rates are rounded off to two decimal places.

The following equation calculates the bit rate register (BRR) setting from the operating frequency and bit rate. N is an integer from 0 to 255, specifying the value with the smaller error.

$$N = \frac{\phi}{1488 \times 2^{2n-1} \times B} \times 10^6 - 1$$

Table 13.6 BRR Settings for Typical Bit Rates (bits/s) (When n = 0)

| bit/s | ϕ (MHz) | | | | | | | | | | | | | | | | | |
|-------|--------------|-------|-------|-------|---------|-------|-------|-------|---------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| | 7.1424 | | 10.00 | | 10.7136 | | 13.00 | | 14.2848 | | 16.00 | | 18.00 | | 20.00 | | 25.00 | |
| | N | Error | N | Error | N | Error | N | Error | N | Error | N | Error | N | Error | N | Error | N | Error |
| 9600 | 0 | 0.00 | 1 | 30 | 1 | 25 | 1 | 8.99 | 1 | 0.00 | 1 | 12.01 | 2 | 15.99 | 2 | 6.66 | 3 | 12.49 |

Table 13.7 Maximum Bit Rates for Various Frequencies (Smart Card Interface Mode)

| ϕ (MHz) | Maximum Bit Rate (bits/s) | N | n |
|--------------|---------------------------|---|---|
| 7.1424 | 9600 | 0 | 0 |
| 10.00 | 13441 | 0 | 0 |
| 10.7136 | 14400 | 0 | 0 |
| 13.00 | 17473 | 0 | 0 |
| 14.2848 | 19200 | 0 | 0 |
| 16.00 | 21505 | 0 | 0 |
| 18.00 | 24194 | 0 | 0 |
| 20.00 | 26882 | 0 | 0 |
| 25.00 | 33602 | 0 | 0 |

The bit rate error is given by the following equation:

$$\text{Error (\%)} = \left(\frac{\phi}{1488 \times 2^{2n-1} \times B \times (N + 1)} \times 10^6 - 1 \right) \times 100$$

13.3.6 Transmitting and Receiving Data

Initialization: Before transmitting or receiving data, the smart card interface must be initialized as described below. Initialization is also necessary when switching from transmit mode to receive mode, or vice versa.

1. Clear the TE and RE bits to 0 in the serial control register (SCR).
2. Clear error flags ERS, PER, and ORER to 0 in the serial status register (SSR).
3. Set the parity bit (O/\bar{E}) and baud rate generator select bits (CKS1 and CKS0) in the serial mode register (SMR). Clear the C/\bar{A} , CHR, and MP bits to 0, and set the STOP and PE bits to 1.
4. Set the SMIF, SDIR, and SINV bits in the smart card mode register (SCMR).
When the SMIF bit is set to 1, the TxD pin and RxD pin are both switched from port to SCI pin functions and go to the high-impedance state.
5. Set a value corresponding to the desired bit rate in the bit rate register (BRR).
6. Set the CKE0 bit in SCR. Clear the TIE, RIE, TE, RE, MPIE, TEIE, and CKE1 bits to 0. If the CKE0 bit is set to 1, the clock is output from the SCK pin.
7. Wait at least one bit interval, then set the TIE, RIE, TE, and RE bits in SCR. Do not set the TE bit and RE bit at the same time, except for self-diagnosis.

Transmitting Serial Data: As data transmission in smart card mode involves error signal sampling and retransmission processing, the processing procedure is different from that for the normal SCI. Figure 13.5 shows a sample transmission processing flowchart.

1. Perform smart card interface mode initialization as described in Initialization above.
2. Check that the ERS error flag is cleared to 0 in SSR.
3. Repeat steps 2 and 3 until it can be confirmed that the TEND flag is set to 1 in SSR.
4. Write the transmit data in TDR, clear the TDRE flag to 0, and perform the transmit operation.
The TEND flag is cleared to 0.
5. To continue transmitting data, go back to step 2.
6. To end transmission, clear the TE bit to 0.

The above processing may include interrupt handling.

If transmission ends and the TEND flag is set to 1 while the TIE bit is set to 1 and interrupt requests are enabled, a transmit-data-empty interrupt (TXI) will be requested. If an error occurs in transmission and the ERS flag is set to 1 while the RIE bit is set to 1 and interrupt requests are enabled, a transmit/receive-error interrupt (ERI) will be requested.

The timing of TEND flag setting depends on the GM bit in SMR.

Figure 13.4 shows timing of TEND flag setting.

For details, see Interrupt Operations in this section.

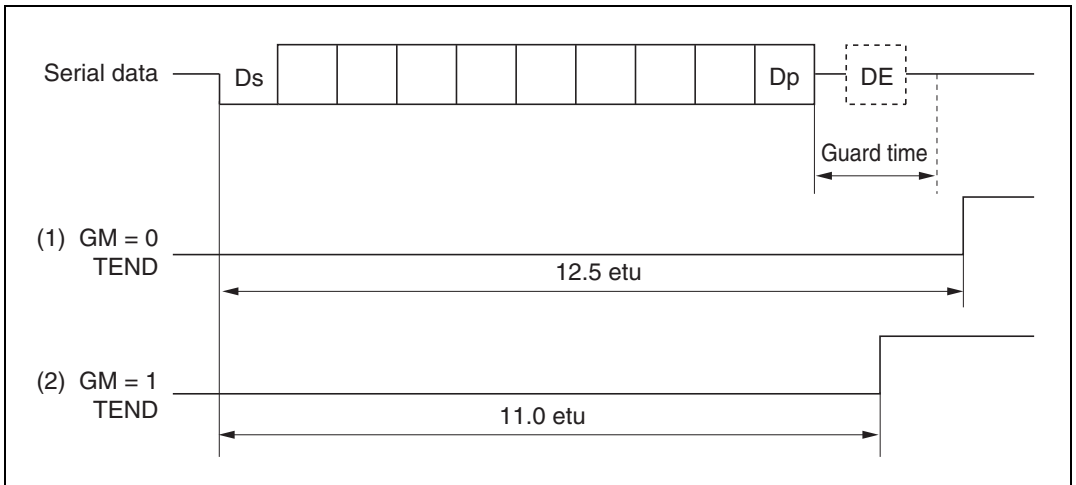


Figure 13.4 Timing of TEND Flag Setting

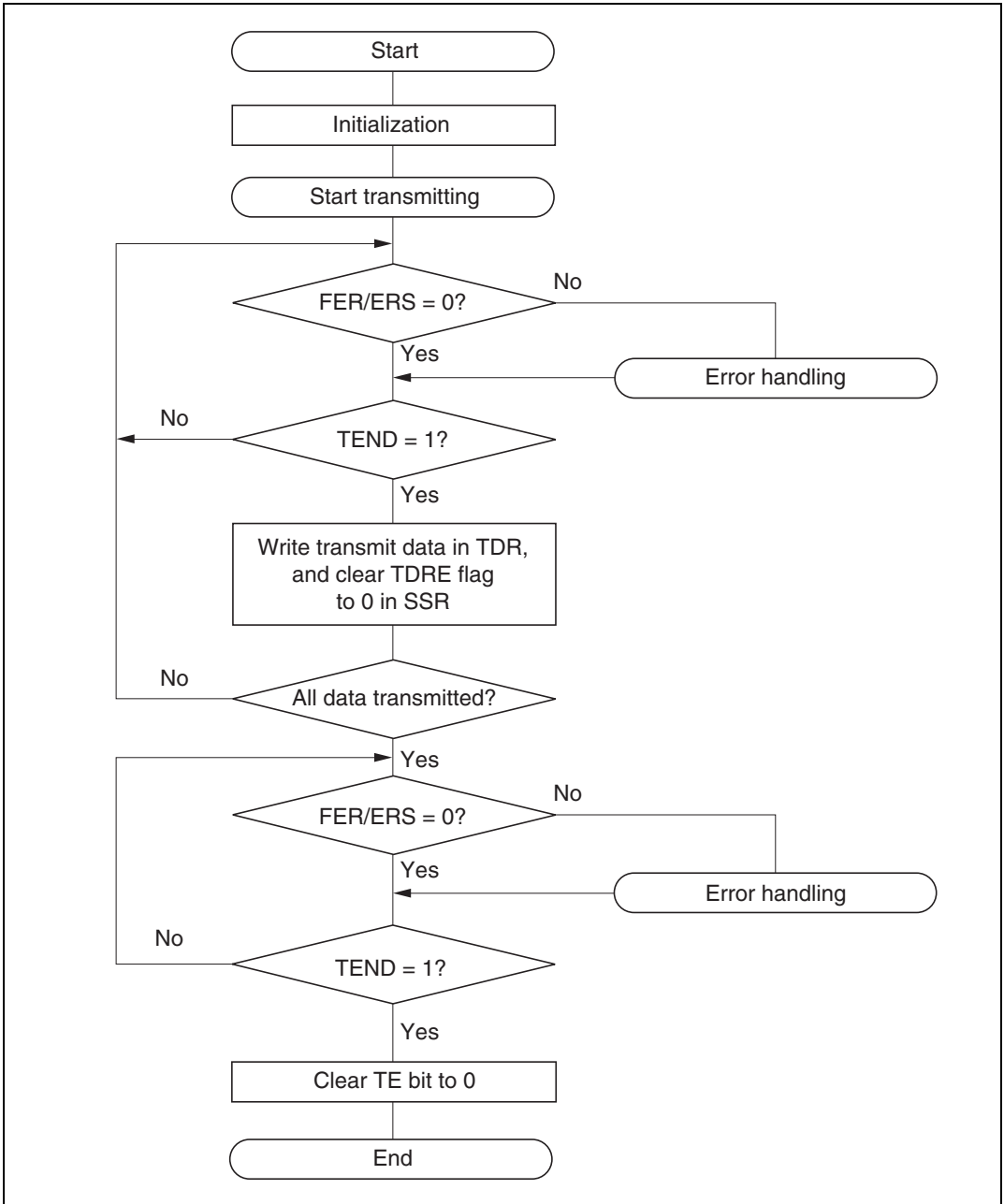


Figure 13.5 Sample Transmission Processing Flowchart

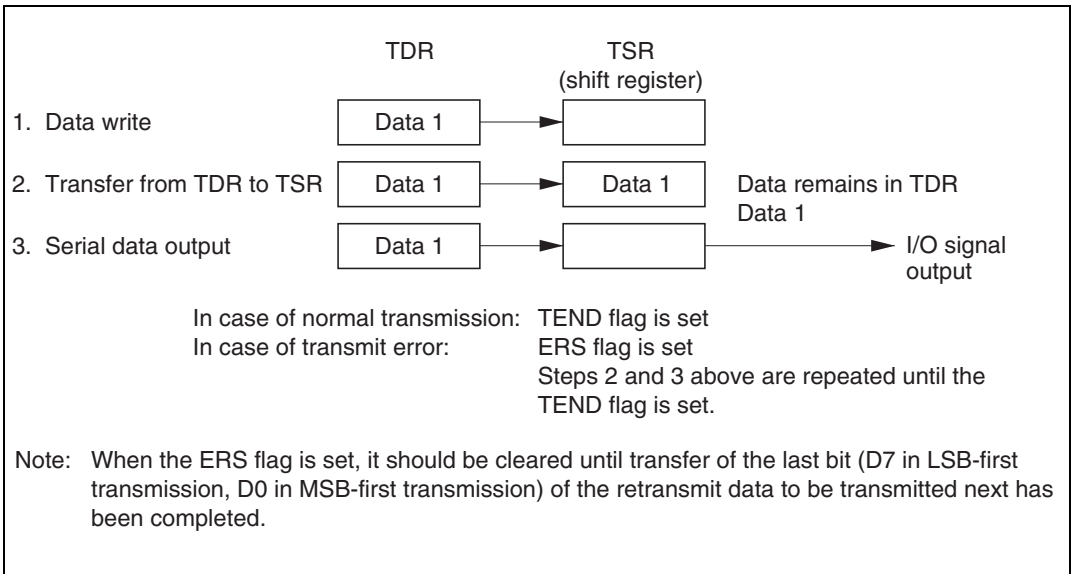


Figure 13.6 Relation Between Transmit Operation and Internal Registers

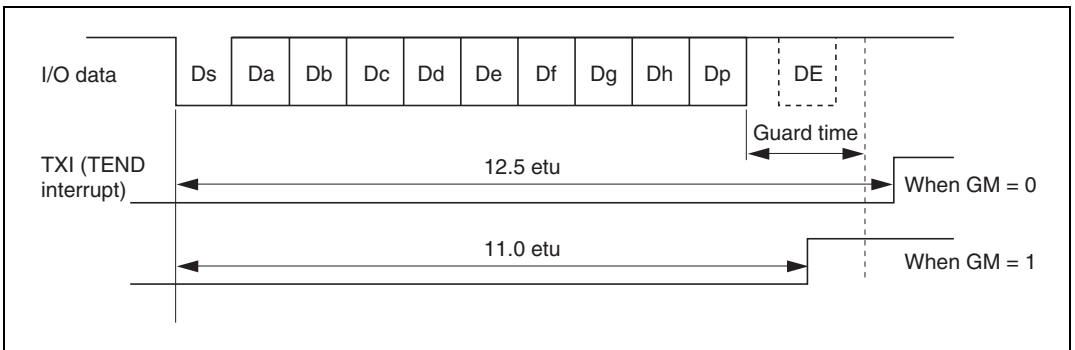


Figure 13.7 Timing of TEND Flag Setting

Receiving Serial Data: Data reception in smart card mode uses the same processing procedure as for the normal SCI. Figure 13.8 shows a sample reception processing flowchart.

1. Perform smart card interface mode initialization as described in Initialization above.
2. Check that the ORER flag and PER flag are cleared to 0 in SSR. If either is set, perform the appropriate receive error handling, then clear both the ORER and the PER flag to 0.
3. Repeat steps 2 and 3 until it can be confirmed that the RDRF flag is set to 1.
4. Read the receive data from RDR.
5. To continue receiving data, clear the RDRF flag to 0 and go back to step 2.
6. To end reception, clear the RE bit to 0.

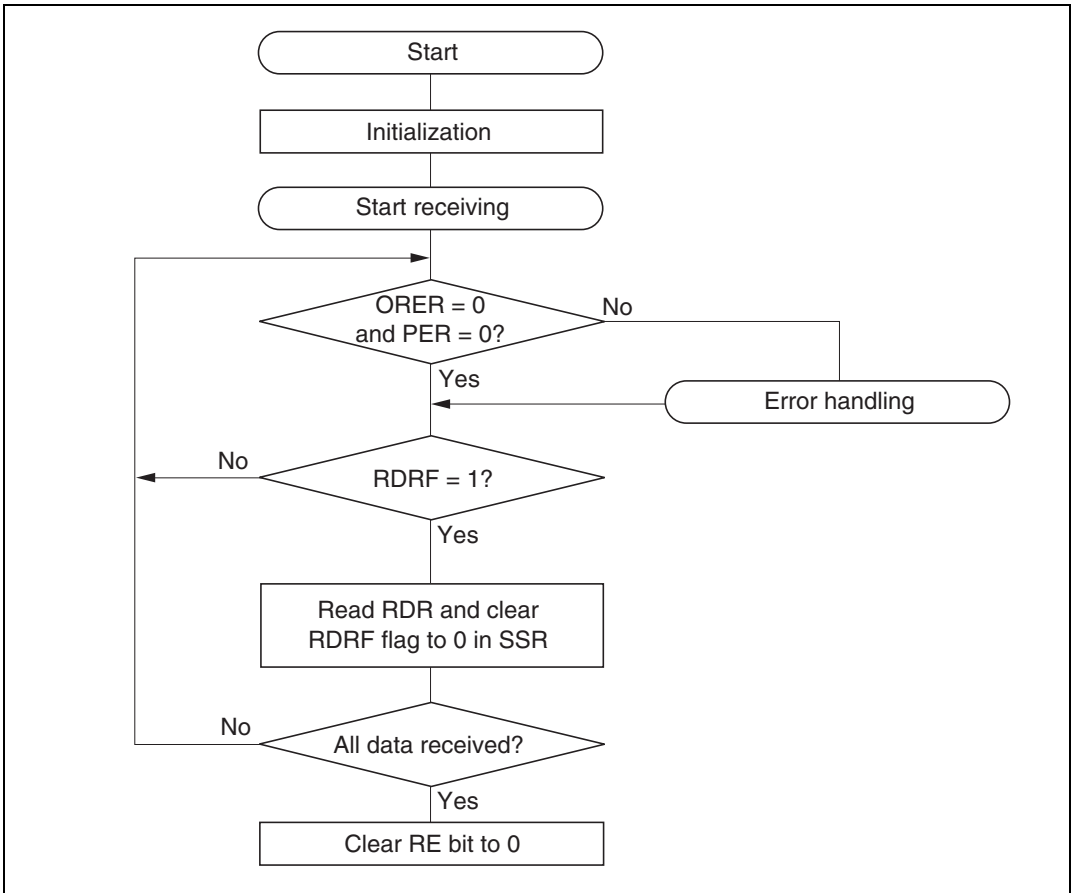


Figure 13.8 Sample Reception Processing Flowchart

The above procedure may include interrupt handling.

If reception ends and the RDRF flag is set to 1 while the RIE bit is set to 1 and interrupt requests are enabled, a receive-data-full interrupt (RXI) will be requested. If an error occurs in reception and either the ORER flag or the PER flag is set to 1, a transmit/receive-error interrupt (ERI) will be requested.

For details, see Interrupt Operations in this section.

If a parity error occurs during reception and the PER flag is set to 1, the received data is transferred to RDR, so the erroneous data can be read.

Switching Modes: When switching from receive mode to transmit mode, first confirm that the receive operation has been completed, then start from initialization, clearing RE to 0 and setting TE to 1. The RDRF, PER, or ORER flag can be used to check that the receive operation has been completed.

When switching from transmit mode to receive mode, first confirm that the transmit operation has been completed, then start from initialization, clearing TE to 0 and setting RE to 1. The TEND flag can be used to check that the transmit operation has been completed.

Fixing Clock Output: When the GM bit is set to 1 in SMR, clock output can be fixed by means of the CKE1 and CKE0 bits in SCR. The minimum clock pulse width can be set to the specified width in this case.

Figure 13.9 shows the timing for fixing clock output. In this example, GM = 1, CKE1 = 0, and the CKE0 bit is controlled.

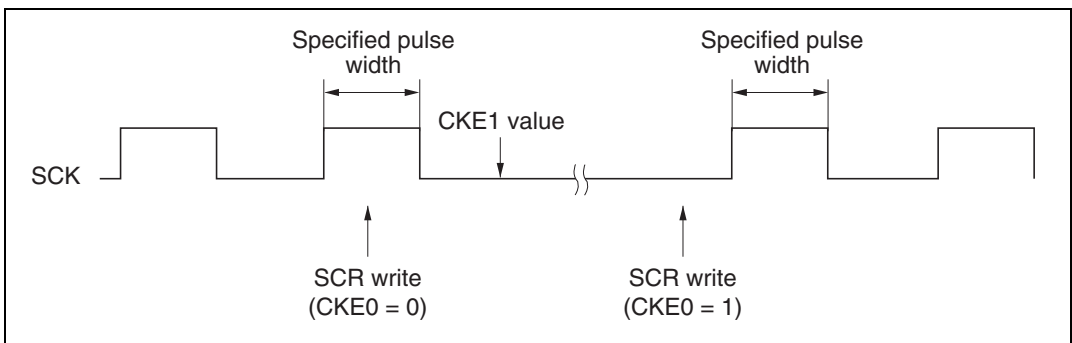


Figure 13.9 Timing for Fixing Clock Output

Interrupt Operations: The smart card interface has three interrupt sources: transmit-data-empty (TXI), transmit/receive-error (ERI), and receive-data-full (RXI). The transmit-end interrupt request (TEI) is not available in smart card mode.

A TXI interrupt is requested when the TEND flag is set to 1 in SSR. An RXI interrupt is requested when the RDRF flag is set to 1 in SSR. An ERI interrupt is requested when the ORER, PER, or ERS flag is set to 1 in SSR. These relationships are shown in table 13.8.

Table 13.8 Smart Card Interface Mode Operating States and Interrupt Sources

| Operating State | | Flag | Enable Bit | Interrupt Source |
|-----------------|------------------|-----------|------------|------------------|
| Transmit Mode | Normal operation | TEND | TIE | TXI |
| | Error | ERS | RIE | ERI |
| Receive Mode | Normal operation | RDRF | RIE | RXI |
| | Error | PER, ORER | RIE | ERI |

Examples of Operation in GSM Mode: When switching between smart card interface mode and software standby mode, use the following procedures to maintain the clock duty cycle.

- Switching from smart card interface mode to software standby mode
 1. Set the P9₄ data register (DR) and data direction register (DDR) to the values for the fixed output state in software standby mode.
 2. Write 0 in the TE and RE bits in the serial control register (SCR) to stop transmit/receive operations. At the same time, set the CKE1 bit to the value for the fixed output state in software standby mode.
 3. Write 0 in the CKE0 bit in SCR to stop the clock.
 4. Wait for one serial clock cycle. During this period, the duty cycle is preserved and clock output is fixed at the specified level.
 5. Write H'00 in the serial mode register (SMR) and smart card mode register (SCMR).
 6. Make the transition to the software standby state.
- Returning from software standby mode to smart card interface mode
 - 1'. Clear the software standby state.
 - 2'. Set the CKE1 bit in SCR to the value for the fixed output state at the start of software standby (the current P9₄ pin state).
 - 3'. Set smart card interface mode and output the clock. Clock signal generation is started with the normal duty cycle.

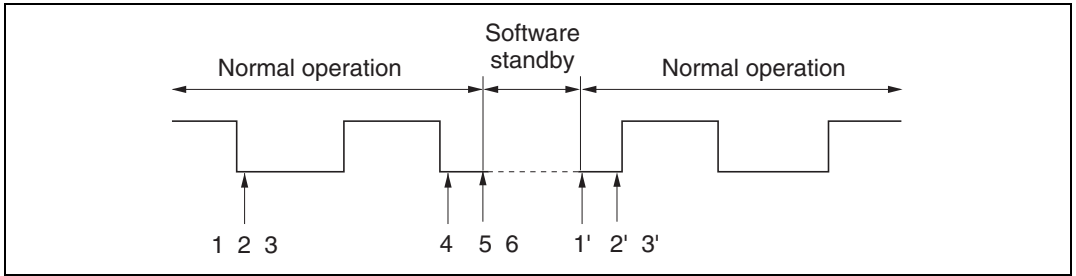


Figure 13.10 Procedure for Stopping and Restarting the Clock

Use the following procedure to secure the clock duty cycle after powering on.

1. The initial state is port input and high impedance. Use pull-up or pull-down resistors to fix the potential.
2. Fix at the output specified by the CKE1 bit in SCR.
3. Set SMR and SCMR, and switch to smart card interface mode operation.
4. Set the CKE0 bit to 1 in SCR to start clock output.

13.4 Usage Notes

The following points should be noted when using the SCI as a smart card interface.

Receive Data Sampling Timing and Receive Margin in Smart Card Interface Mode: In smart card interface mode, the SCI operates on a base clock with a frequency of 372 times the transfer rate. In reception, the SCI synchronizes internally with the fall of the start bit, which it samples on the base clock. Receive data is latched at the rising edge of the 186th base clock pulse. The timing is shown in figure 13.11.

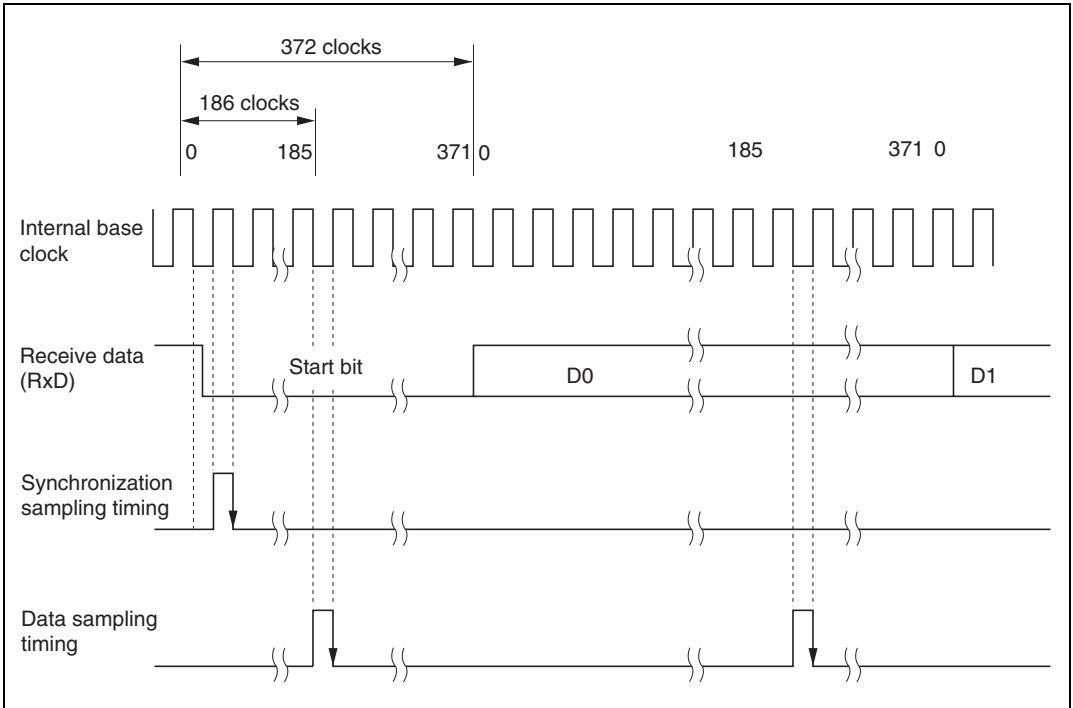


Figure 13.11 Receive Data Sampling Timing in Smart Card Interface Mode

The receive margin can therefore be expressed as follows.

Receive margin in smart card interface mode:

$$M = \left| \left(0.5 - \frac{1}{2N} \right) - (L - 0.5) F - \frac{|D - 0.5|}{N} (1 + F) \right| \times 100\%$$

Legend:

M: Receive margin (%)

N: Ratio of clock frequency to bit rate (N = 372)

D: Clock duty cycle (D = 0 to 1.0)

L: Frame length (L = 10)

F: Absolute deviation of clock frequency

From the above equation, if F = 0 and D = 0.5, the receive margin is as follows.

When D = 0.5 and F = 0:

$$\begin{aligned} M &= (0.5 - 1/2 \times 372) \times 100\% \\ &= 49.866\% \end{aligned}$$

Retransmission: Retransmission is performed by the SCI in receive mode and transmit mode as described below.

- Retransmission when SCI is in Receive Mode

Figure 13.12 illustrates retransmission when the SCI is in receive mode.

1. If an error is found when the received parity bit is checked, the PER bit is automatically set to 1. If the RIE bit in SCR is set to the enable state, an ERI interrupt is requested. The PER bit should be cleared to 0 in SSR before the next parity bit sampling timing.
2. The RDRF bit in SSR is not set for the frame in which the error has occurred.
3. If an error is found when the received parity bit is checked, the PER bit is not set to 1 in SSR.
4. If no error is found when the received parity bit is checked, the receive operation is assumed to have been completed normally, and the RDRF bit is automatically set to 1 in SSR. If the RIE bit in SCR is set to the enable state, an RXI interrupt is requested.
5. When a normal frame is received, the data pin is held in three-state at the error signal transmission timing.

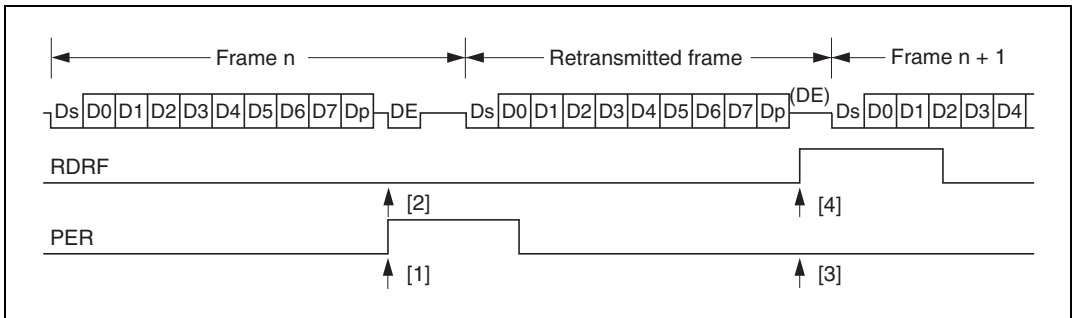


Figure 13.12 Retransmission in SCI Receive Mode

- Retransmission when SCI is in Transmit Mode

Figure 13.13 illustrates retransmission when the SCI is in transmit mode.

6. If an error signal is sent back from the receiving device after transmission of one frame is completed, the ERS bit is set to 1 in SSR. If the RIE bit in SCR is set to the enable state, an ERI interrupt is requested. The ERS bit should be cleared to 0 in SSR before the next parity bit sampling timing.
7. The TEND bit in SSR is not set for the frame for which the error signal was received.
8. If an error signal is not sent back from the receiving device, the ERS flag is not set in SSR.
9. If an error signal is not sent back from the receiving device, transmission of one frame, including retransmission, is assumed to have been completed, and the TEND bit is set to 1 in SSR. If the TIE bit in SCR is set to the enable state, a TXI interrupt is requested.

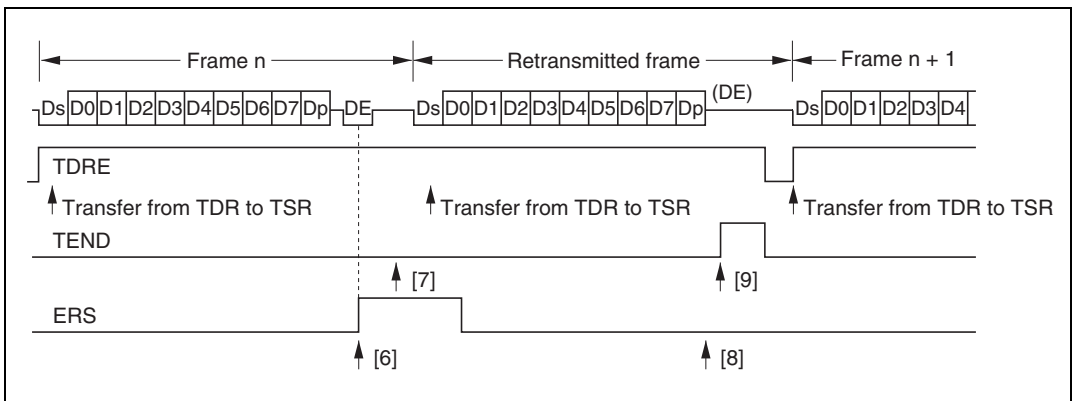


Figure 13.13 Retransmission in SCI Transmit Mode

Note on Block Transfer Mode Support: The smart card interface installed in the H8/3008 supports an IC card (smart card) interface with provision for ISO/IEC7816-3 T = 0 (character transmission). Therefore, block transfer operations are not supported (error signal transmission, detection, and automatic data retransmission are not performed).

Section 14 A/D Converter

14.1 Overview

The H8/3008 includes a 10-bit successive-approximations A/D converter with a selection of up to eight analog input channels.

When the A/D converter is not used, it can be halted independently to conserve power. For details see section 18.6, Module Standby Function.

The H8/3008 supports 70/134-state conversion as a high-speed conversion mode. Note that it differs in this respect from the H8/3048 Group, which supports 134/266-state conversion.

14.1.1 Features

A/D converter features are listed below.

- 10-bit resolution
- Eight input channels
- Selectable analog conversion voltage range
The analog voltage conversion range can be programmed by input of an analog reference voltage at the V_{REF} pin.
- High-speed conversion
Conversion time: minimum 5.36 μ s per channel
- Two conversion modes
Single mode: A/D conversion of one channel
Scan mode: continuous A/D conversion on one to four channels
- Four 16-bit data registers
A/D conversion results are transferred for storage into data registers corresponding to the channels.
- Sample-and-hold function
- Three conversion start sources
The A/D converter can be activated by software, an external trigger, or an 8-bit timer compare match.
- A/D interrupt requested at end of conversion
At the end of A/D conversion, an A/D end interrupt (ADI) can be requested.

14.1.2 Block Diagram

Figure 14.1 shows a block diagram of the A/D converter.

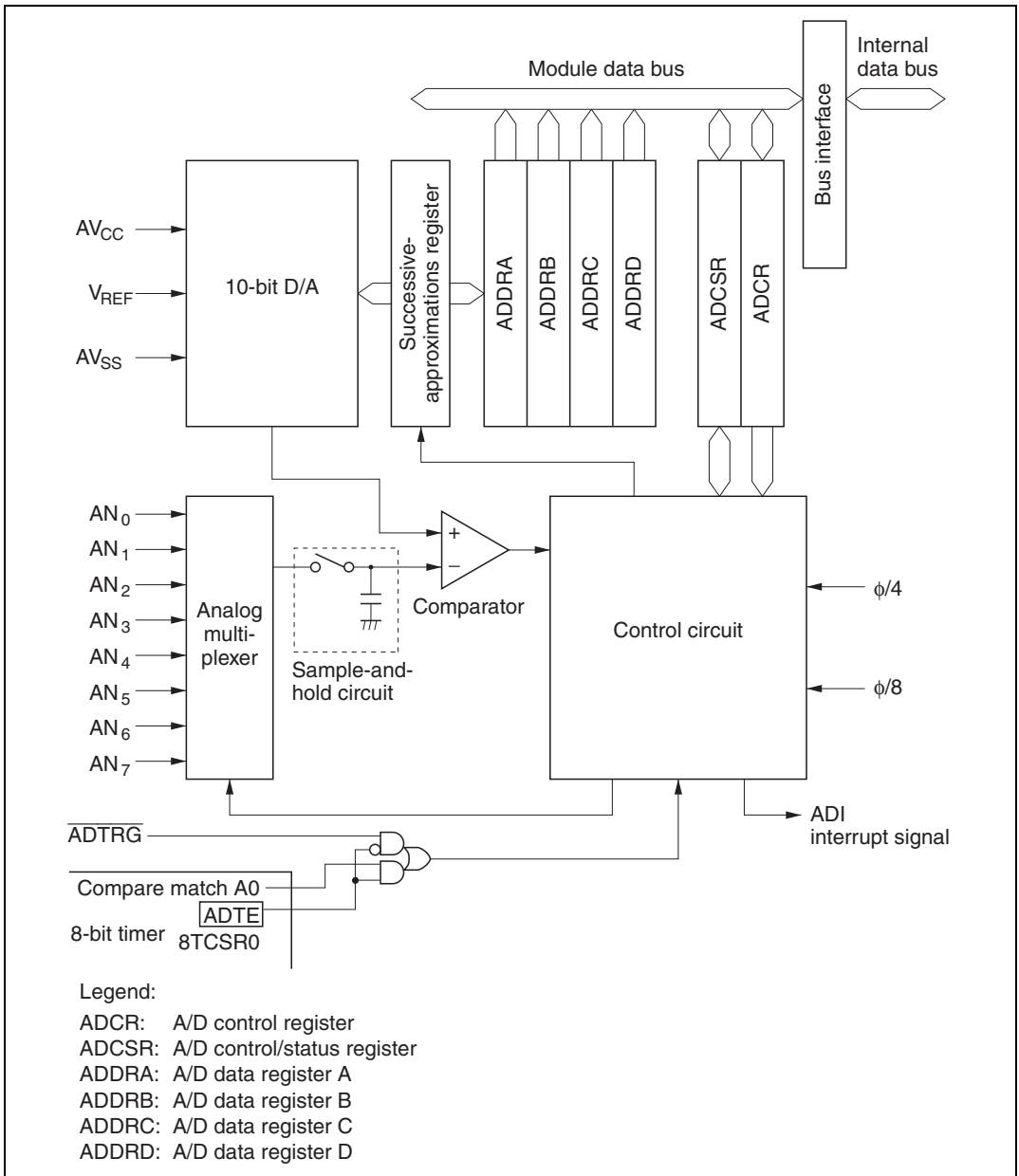


Figure 14.1 A/D Converter Block Diagram

14.1.3 Pin Configuration

Table 14.1 summarizes the A/D converter's input pins. The eight analog input pins are divided into two groups: group 0 (AN_0 to AN_3), and group 1 (AN_4 to AN_7). AV_{CC} and AV_{SS} are the power supply for the analog circuits in the A/D converter. V_{REF} is the A/D conversion reference voltage.

Table 14.1 A/D Converter Pins

| Pin Name | Abbreviation | I/O | Function |
|--------------------------------|--------------------|-------|--|
| Analog power supply pin | AV_{CC} | Input | Analog power supply |
| Analog ground pin | AV_{SS} | Input | Analog ground and reference voltage |
| Reference voltage pin | V_{REF} | Input | Analog reference voltage |
| Analog input pin 0 | AN_0 | Input | Group 0 analog inputs |
| Analog input pin 1 | AN_1 | Input | |
| Analog input pin 2 | AN_2 | Input | |
| Analog input pin 3 | AN_3 | Input | |
| Analog input pin 4 | AN_4 | Input | Group 1 analog inputs |
| Analog input pin 5 | AN_5 | Input | |
| Analog input pin 6 | AN_6 | Input | |
| Analog input pin 7 | AN_7 | Input | |
| A/D external trigger input pin | \overline{ADTRG} | Input | External trigger input for starting A/D conversion |

14.1.4 Register Configuration

Table 14.2 summarizes the A/D converter's registers.

Table 14.2 A/D Converter Registers

| Address* ¹ | Name | Abbreviation | R/W | Initial Value |
|-----------------------|-----------------------------|--------------|---------------------|---------------|
| H'FFFE0 | A/D data register A H | ADDRAH | R | H'00 |
| H'FFFE1 | A/D data register A L | ADDRAL | R | H'00 |
| H'FFFE2 | A/D data register B H | ADDRBH | R | H'00 |
| H'FFFE3 | A/D data register B L | ADDRBL | R | H'00 |
| H'FFFE4 | A/D data register C H | ADDRCH | R | H'00 |
| H'FFFE5 | A/D data register C L | ADDRCL | R | H'00 |
| H'FFFE6 | A/D data register D H | ADDRDH | R | H'00 |
| H'FFFE7 | A/D data register D L | ADDRDL | R | H'00 |
| H'FFFE8 | A/D control/status register | ADCSR | R/(W)* ² | H'00 |
| H'FFFE9 | A/D control register | ADCR | R/W | H'7E |

- Notes: 1. Lower 20 bits of the address in advanced mode.
 2. Only 0 can be written in bit 7, to clear the flag.

14.2 Register Descriptions

14.2.1 A/D Data Registers A to D (ADDRA to ADDR D)

| | | | | | | | | | | | | | | | | |
|---------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|---|---|---|---|---|---|
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ADDRn | AD9 | AD8 | AD7 | AD6 | AD5 | AD4 | AD3 | AD2 | AD1 | AD0 | — | — | — | — | — | — |
| Initial value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Read/Write | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

A/D conversion data
10-bit data giving an
A/D conversion result
Reserved bits

Note: n = A to D

The four A/D data registers (ADDRA to ADDR D) are 16-bit read-only registers that store the results of A/D conversion.

An A/D conversion produces 10-bit data, which is transferred for storage into the A/D data register corresponding to the selected channel. The upper 8 bits of the result are stored in the upper byte of the A/D data register. The lower 2 bits are stored in the lower byte. Bits 5 to 0 of an A/D data register are reserved bits that are always read as 0. Table 14.3 indicates the pairings of analog input channels and A/D data registers.

The CPU can always read the A/D data registers. The upper byte can be read directly, but the lower byte is read through a temporary register (TEMP). For details see section 14.3, CPU Interface.

The A/D data registers are initialized to H'0000 by a reset and in standby mode.

Table 14.3 Analog Input Channels and A/D Data Registers (ADDRA to ADDR D)

| Analog Input Channel | | A/D Data Register |
|----------------------|-----------------|-------------------|
| Group 0 | Group 1 | |
| AN ₀ | AN ₄ | ADDRA |
| AN ₁ | AN ₅ | ADDRB |
| AN ₂ | AN ₆ | ADDRC |
| AN ₃ | AN ₇ | ADDRD |

14.2.2 A/D Control/Status Register (ADCSR)

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|--------|------|------|------|-----|-----|-----|-----|
| | ADF | ADIE | ADST | SCAN | CKS | CH2 | CH1 | CH0 |
| Initial value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Read/Write | R/(W)* | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

A/D end flag
Indicates end of A/D conversion

A/D interrupt enable
Enables and disables A/D end interrupts

A/D start
Starts or stops A/D conversion

Scan mode
Selects single mode or scan mode

Clock select
Selects the A/D conversion time

Channel select 2 to 0
These bits select analog input channels

Note: * Only 0 can be written, to clear the flag.

ADCSR is an 8-bit readable/writable register that selects the mode and controls the A/D converter. ADCSR is initialized to H'00 by a reset and in standby mode.

Bit 7—A/D End Flag (ADF): Indicates the end of A/D conversion.

Bit 7

| ADF | Description |
|-----|--|
| 0 | [Clearing condition] Read ADF when ADF = 1, then write 0 in ADF. (Initial value) |
| 1 | [Setting conditions] <ul style="list-style-type: none"> • Single mode: A/D conversion ends • Scan mode: A/D conversion ends in all selected channels |

Bit 6—A/D Interrupt Enable (ADIE): Enables or disables the interrupt (ADI) requested at the end of A/D conversion.

| Bit 6 ADIE | Description | |
|---------------|---|-----------------|
| 0 | A/D end interrupt request (ADI) is disabled | (Initial value) |
| 1 | A/D end interrupt request (ADI) is enabled | |

Bit 5—A/D Start (ADST): Starts or stops A/D conversion. The ADST bit remains set to 1 during A/D conversion. It can also be set to 1 by external trigger input at the $\overline{\text{ADTRG}}$ pin, or by an 8-bit timer compare match.

| Bit 5 ADST | Description | |
|---------------|--|-----------------|
| 0 | A/D conversion is stopped | (Initial value) |
| 1 | Single mode: A/D conversion starts; ADST is automatically cleared to 0 when conversion ends. Scan mode: A/D conversion starts and continues, cycling among the selected channels, until ADST is cleared to 0 by software, by a reset, or by a transition to standby mode. | |

Bit 4—Scan Mode (SCAN): Selects single mode or scan mode. For further information on operation in these modes, see section 14.4, Operation. Clear the ADST bit to 0 before switching the conversion mode.

| Bit 4 SCAN | Description | |
|---------------|-------------|-----------------|
| 0 | Single mode | (Initial value) |
| 1 | Scan mode | |

Bit 3—Clock Select (CKS): Selects the A/D conversion time. Clear the ADST bit to 0 before switching the conversion time.

| Bit 3 CKS | Description | |
|--------------|--|-----------------|
| 0 | Conversion time = 134 states (maximum) | (Initial value) |
| 1 | Conversion time = 70 states (maximum) | |

Bits 2 to 0—Channel Select 2 to 0 (CH2 to CH0): These bits and the SCAN bit select the analog input channels. Clear the ADST bit to 0 before changing the channel selection.

| Group Selection | Channel Selection | | Description | |
|-----------------|-------------------|-----|---------------------------------|------------------------------------|
| | CH1 | CH0 | Single Mode | Scan Mode |
| 0 | 0 | 0 | AN ₀ (Initial value) | AN ₀ |
| | | 1 | AN ₁ | AN ₀ , AN ₁ |
| | 1 | 0 | AN ₂ | AN ₀ to AN ₂ |
| | | 1 | AN ₃ | AN ₀ to AN ₃ |
| 1 | 0 | 0 | AN ₄ | AN ₄ |
| | | 1 | AN ₅ | AN ₄ , AN ₅ |
| | 1 | 0 | AN ₆ | AN ₄ to AN ₆ |
| | | 1 | AN ₇ | AN ₄ to AN ₇ |

14.2.3 A/D Control Register (ADCR)

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|------|---|---|---|---|---|---|-----|
| | TRGE | — | — | — | — | — | — | — |
| Initial value | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| Read/Write | R/W | — | — | — | — | — | — | R/W |

Reserved bits

Trigger enable

Enables or disables starting of A/D conversion by an external trigger or 8-bit timer compare match

ADCR is an 8-bit readable/writable register that enables or disables starting of A/D conversion by external trigger input or an 8-bit timer compare match signal. ADCR is initialized to H'7F by a reset and in standby mode.

Bit 7—Trigger Enable (TRGE): Enables or disables starting of A/D conversion by an external trigger or 8-bit timer compare match.

| Bit 7 TRGE | Description | |
|---------------|---|-----------------|
| 0 | Starting of A/D conversion by an external trigger or 8-bit timer compare match is disabled | (Initial value) |
| 1 | A/D conversion is started at the falling edge of the external trigger signal (ADTRG) or by an 8-bit timer compare match | |

External trigger pin and 8-bit timer selection is performed by the 8-bit timer. For details, see section 9, 8-Bit Timers.

Bits 6 to 1—Reserved: These bits cannot be modified and are always read as 1.

Bit 0—Reserved: This bit can be read or written, but must not be set to 1.

14.3 CPU Interface

ADDRA to ADDR D are 16-bit registers, but they are connected to the CPU by an 8-bit data bus. Therefore, although the upper byte can be accessed directly by the CPU, the lower byte is read through an 8-bit temporary register (TEMP).

An A/D data register is read as follows. When the upper byte is read, the upper-byte value is transferred directly to the CPU and the lower-byte value is transferred into TEMP. Next, when the lower byte is read, the TEMP contents are transferred to the CPU.

When reading an A/D data register, always read the upper byte before the lower byte. It is possible to read only the upper byte, but if only the lower byte is read, incorrect data may be obtained.

Figure 14.2 shows the data flow for access to an A/D data register.

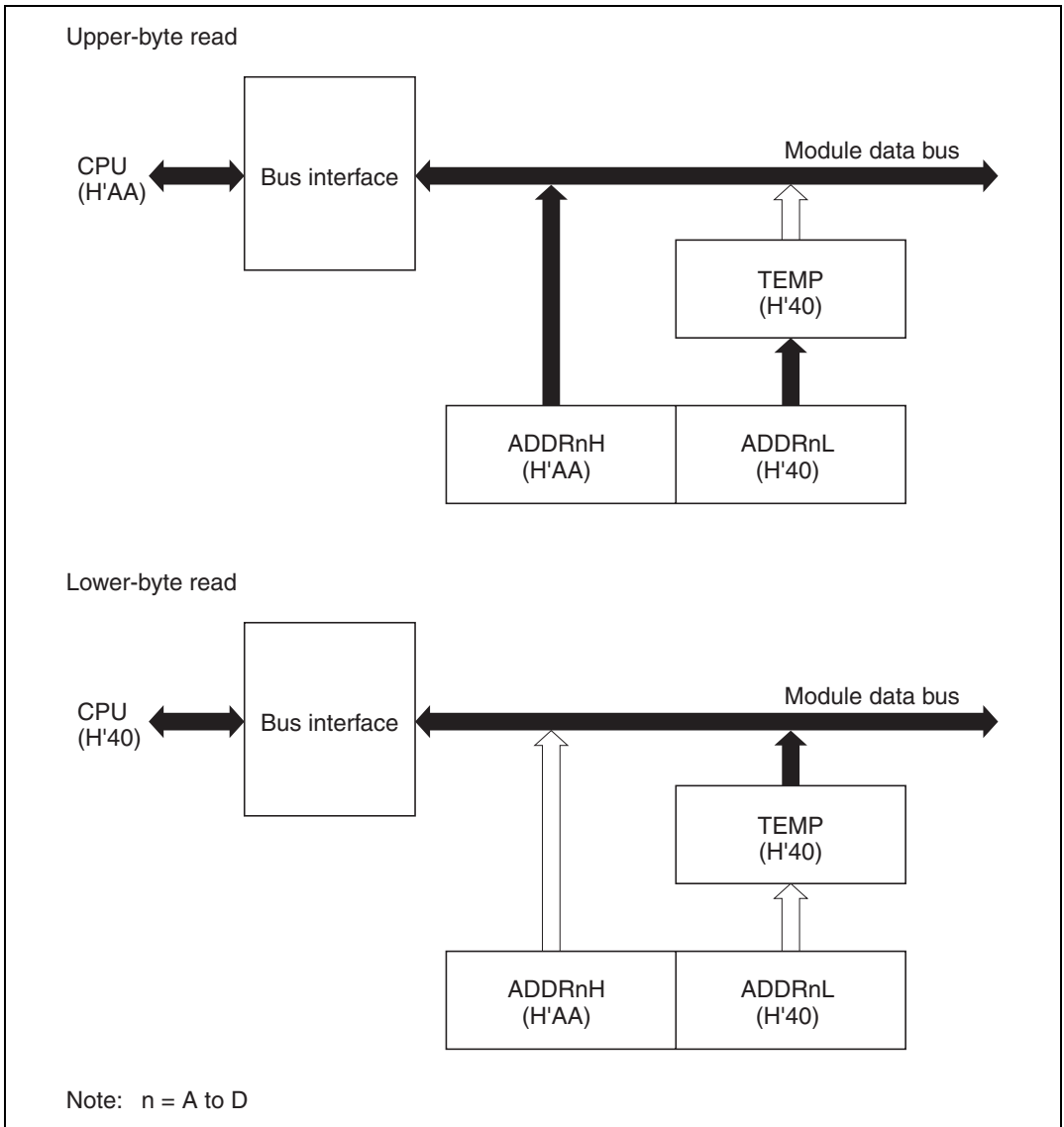


Figure 14.2 A/D Data Register Access Operation (Reading H'AA40)

14.4 Operation

The A/D converter operates by successive approximations with 10-bit resolution. It has two operating modes: single mode and scan mode.

14.4.1 Single Mode (SCAN = 0)

Single mode should be selected when only one A/D conversion on one channel is required. A/D conversion starts when the ADST bit is set to 1 by software, or by external trigger input. The ADST bit remains set to 1 during A/D conversion and is automatically cleared to 0 when conversion ends.

When conversion ends the ADF flag is set to 1. If the ADIE bit is also set to 1, an ADI interrupt is requested at this time. To clear the ADF flag to 0, first read ADCSR, then write 0 in ADF.

When the mode or analog input channel must be switched during analog conversion, to prevent incorrect operation, first clear the ADST bit to 0 in ADCSR to halt A/D conversion. After making the necessary changes, set the ADST bit to 1 to start A/D conversion again. The ADST bit can be set at the same time as the mode or channel is changed.

Typical operations when channel 1 (AN_1) is selected in single mode are described next.

Figure 14.3 shows a timing diagram for this example.

1. Single mode is selected (SCAN = 0), input channel AN_1 is selected (CH2 = CH1 = 0, CH0 = 1), the A/D interrupt is enabled (ADIE = 1), and A/D conversion is started (ADST = 1).
2. When A/D conversion is completed, the result is transferred into ADDR0. At the same time the ADF flag is set to 1, the ADST bit is cleared to 0, and the A/D converter becomes idle.
3. Since ADF = 1 and ADIE = 1, an ADI interrupt is requested.
4. The A/D interrupt handling routine starts.
5. The routine reads ADCSR, then writes 0 in the ADF flag.
6. The routine reads and processes the conversion result (ADDR0).
7. Execution of the A/D interrupt handling routine ends. After that, if the ADST bit is set to 1, A/D conversion starts again and steps 2 to 7 are repeated.

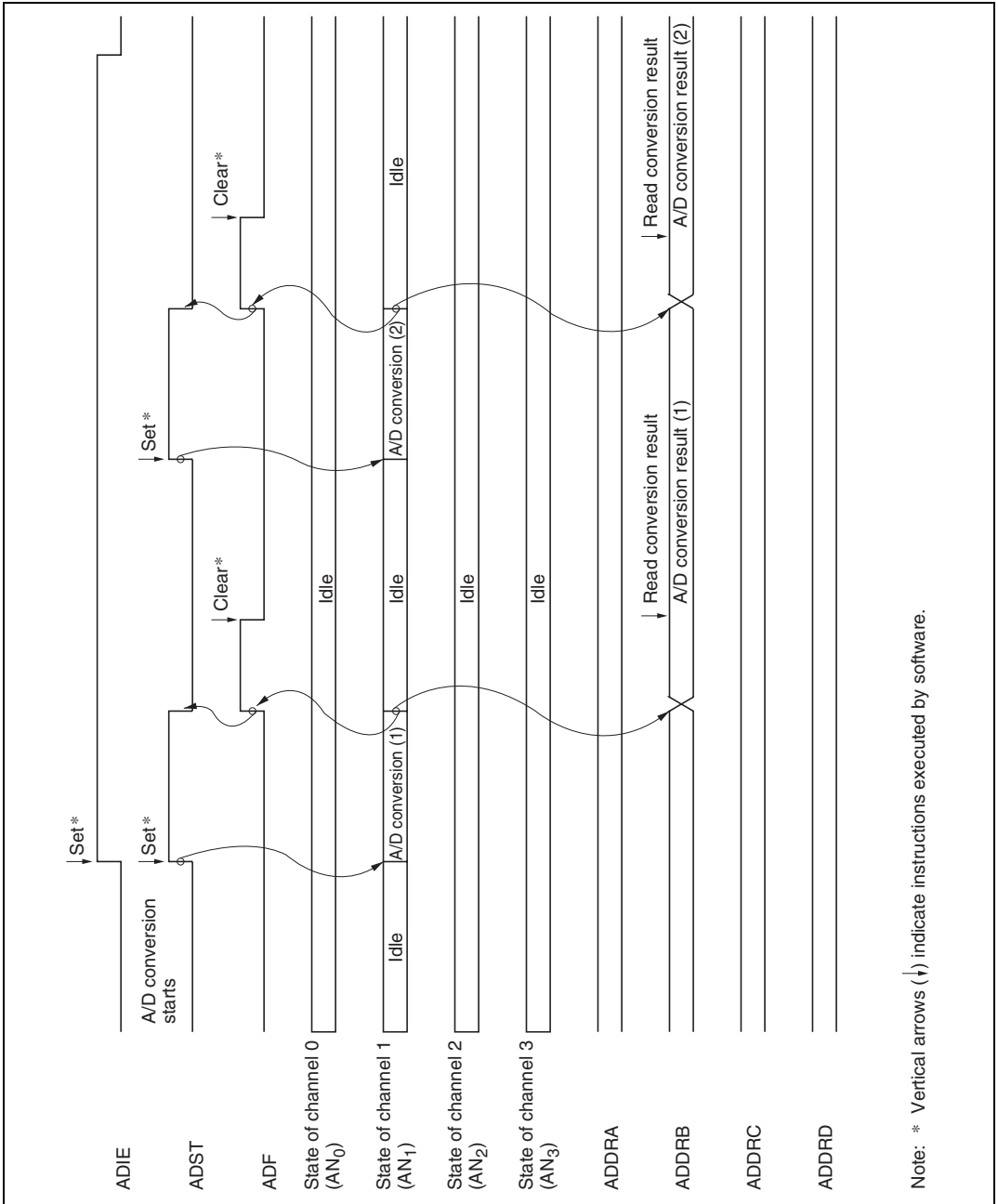


Figure 14.3 Example of A/D Converter Operation (Single Mode, Channel 1 Selected)

14.4.2 Scan Mode (SCAN = 1)

Scan mode is useful for monitoring analog inputs in a group of one or more channels. When the ADST bit is set to 1 by software or external trigger input, A/D conversion starts on the first channel in the group (AN_0 when $CH2 = 0$, AN_4 when $CH2 = 1$). When two or more channels are selected, after conversion of the first channel ends, conversion of the second channel (AN_1 or AN_5) starts immediately. A/D conversion continues cyclically on the selected channels until the ADST bit is cleared to 0. The conversion results are transferred for storage into the A/D data registers corresponding to the channels.

When the mode or analog input channel selection must be changed during analog conversion, to prevent incorrect operation, first clear the ADST bit to 0 in ADCSR to halt A/D conversion. After making the necessary changes, set the ADST bit to 1. A/D conversion will start again from the first channel in the group. The ADST bit can be set at the same time as the mode or channel selection is changed.

Typical operations when three channels in group 0 (AN_0 to AN_2) are selected in scan mode are described next. Figure 14.4 shows a timing diagram for this example.

1. Scan mode is selected ($SCAN = 1$), scan group 0 is selected ($CH2 = 0$), analog input channels AN_0 to AN_2 are selected ($CH1 = 1$, $CH0 = 0$), and A/D conversion is started ($ADST = 1$).
2. When A/D conversion of the first channel (AN_0) is completed, the result is transferred into ADDRA. Next, conversion of the second channel (AN_1) starts automatically.
3. Conversion proceeds in the same way through the third channel (AN_2).
4. When conversion of all selected channels (AN_0 to AN_2) is completed, the ADF flag is set to 1 and conversion of the first channel (AN_0) starts again. If the ADIE bit is set to 1, an ADI interrupt is requested when A/D conversion ends.
5. Steps 2 to 4 are repeated as long as the ADST bit remains set to 1. When the ADST bit is cleared to 0, A/D conversion stops. After that, if the ADST bit is set to 1, A/D conversion starts again from the first channel (AN_0).

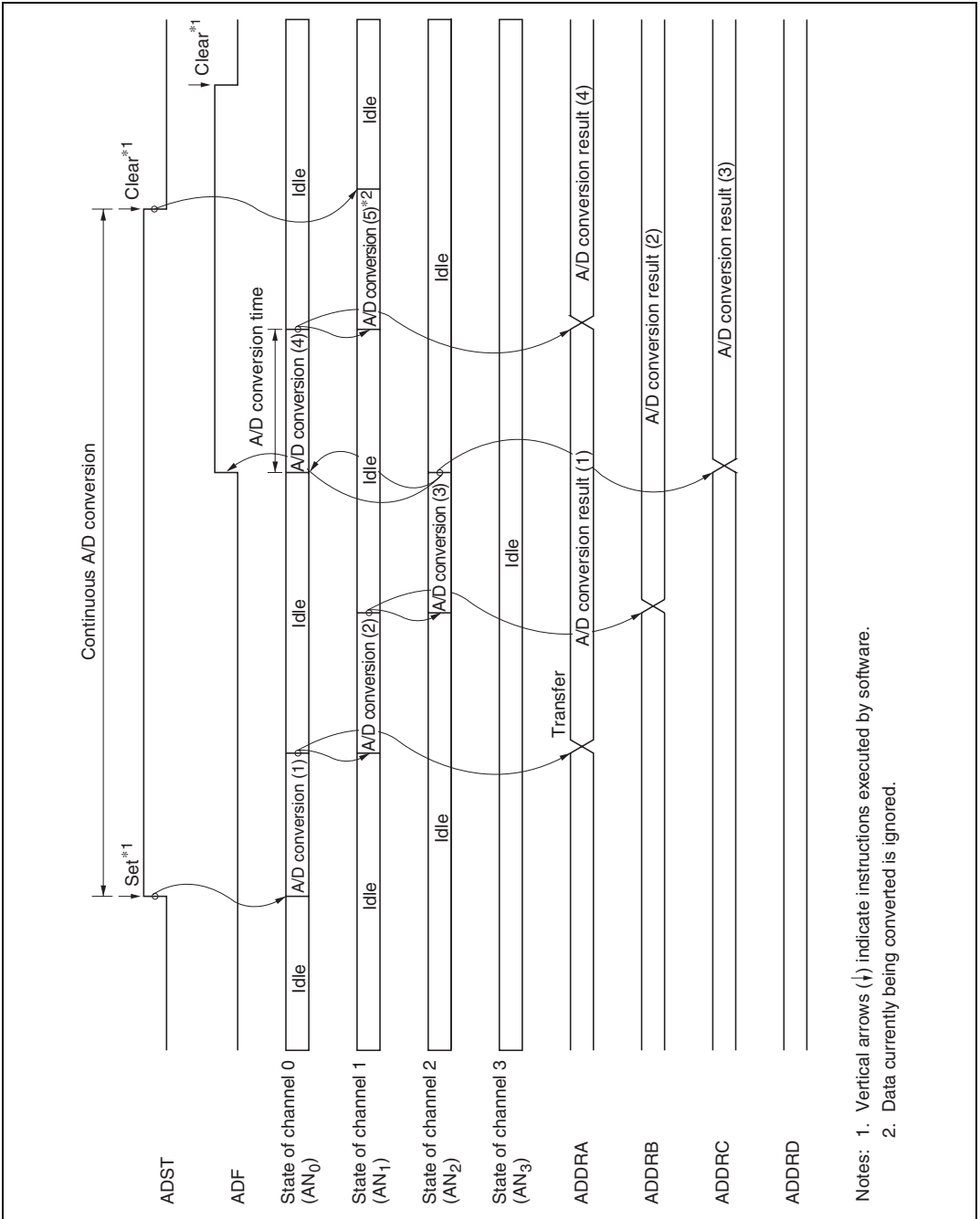


Figure 14.4 Example of A/D Converter Operation (Scan Mode, Channels AN₀ to AN₂ Selected)

- Notes:
1. Vertical arrows (↓) indicate instructions executed by software.
 2. Data currently being converted is ignored.

14.4.3 Input Sampling and A/D Conversion Time

The A/D converter has a built-in sample-and-hold circuit. The A/D converter samples the analog input at a time t_p after the ADST bit is set to 1, then starts conversion. Figure 14.5 shows the A/D conversion timing. Table 14.4 indicates the A/D conversion time.

As indicated in figure 14.5, the A/D conversion time includes t_p and the input sampling time. The length of t_p varies depending on the timing of the write access to ADCSR. The total conversion time therefore varies within the ranges indicated in table 14.4.

In scan mode, the values given in table 14.4 apply to the first conversion. In the second and subsequent conversions the conversion time is fixed at 128 states when CKS = 0 or 66 states when CKS = 1.

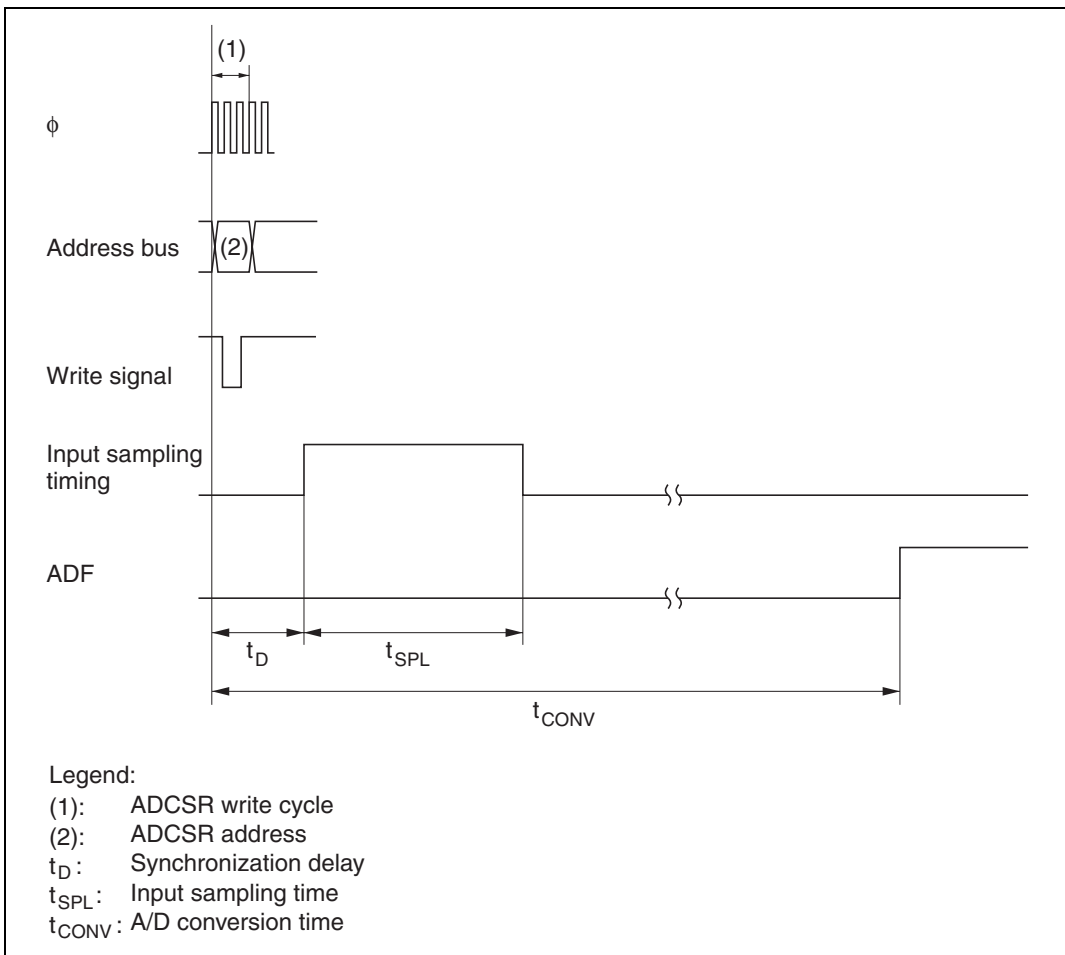


Figure 14.5 A/D Conversion Timing

Table 14.4 A/D Conversion Time (Single Mode)

| Symbol | CKS = 0 | | | CKS = 1 | | | |
|-----------------------|------------|-----|-----|---------|-----|-----|----|
| | Min | Typ | Max | Min | Typ | Max | |
| Synchronization delay | t_D | 6 | — | 9 | 4 | — | 5 |
| Input sampling time | t_{SPL} | — | 31 | — | 15 | — | — |
| A/D conversion time | t_{CONV} | 131 | — | 134 | 69 | — | 70 |

Note: Values in the table are numbers of states.

14.4.4 External Trigger Input Timing

A/D conversion can be externally triggered. When the TRGE bit is set to 1 in ADCR and the 8-bit timer's ADTE bit is cleared to 0, external trigger input is enabled at the $\overline{\text{ADTRG}}$ pin. A high-to-low transition at the $\overline{\text{ADTRG}}$ pin sets the ADST bit to 1 in ADCSR, starting A/D conversion. Other operations, in both single and scan modes, are the same as if the ADST bit had been set to 1 by software. Figure 14.6 shows the timing.

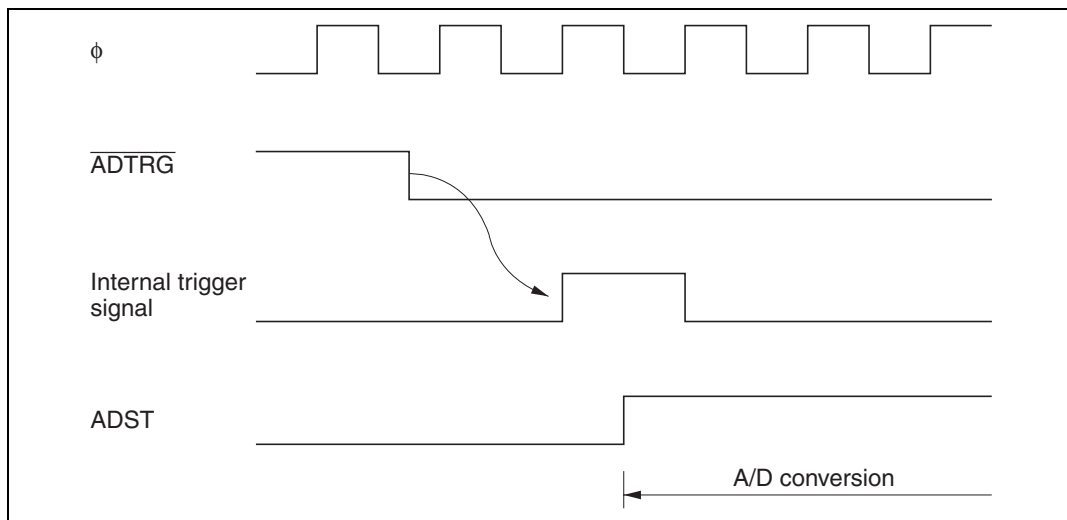


Figure 14.6 External Trigger Input Timing

14.5 Interrupts

The A/D converter generates an interrupt (ADI) at the end of A/D conversion. The ADI interrupt request can be enabled or disabled by the ADIE bit in ADCSR.

14.6 Usage Notes

When using the A/D converter, note the following points:

1. Analog Input Voltage Range

During A/D conversion, the voltages input to the analog input pins AN_n should be in the range $AV_{SS} \leq AN_n \leq V_{REF}$.

2. Relationships of AV_{CC} and AV_{SS} to V_{CC} and V_{SS}

AV_{CC} , AV_{SS} , V_{CC} , and V_{SS} should be related as follows: $AV_{SS} = V_{SS}$. AV_{CC} and AV_{SS} must not be left open, even if the A/D converter is not used.

3. V_{REF} Programming Range

The reference voltage input at the V_{REF} pin should be in the range $V_{REF} \leq AV_{CC}$.

4. Note on Board Design

In board layout, separate the digital circuits from the analog circuits as much as possible. Particularly avoid layouts in which the signal lines of digital circuits cross or closely approach the signal lines of analog circuits. Induction and other effects may cause the analog circuits to operate incorrectly, or may adversely affect the accuracy of A/D conversion.

The analog input signals (AN_0 to AN_7), analog reference voltage (V_{REF}), and analog supply voltage (AV_{CC}) must be separated from digital circuits by the analog ground (AV_{SS}). The analog ground (AV_{SS}) should be connected to a stable digital ground (V_{SS}) at one point on the board.

5. Note on Noise

To prevent damage from surges and other abnormal voltages at the analog input pins (AN_0 to AN_7) and analog reference voltage pin (V_{REF}), connect a protection circuit like the one in figure 14.7 between AV_{CC} and AV_{SS} . The bypass capacitors connected to AV_{CC} and V_{REF} and the filter capacitors connected to AN_0 to AN_7 must be connected to AV_{SS} . If filter capacitors like the ones in figure 14.7 are connected, the voltage values input to the analog input pins (AN_0 to AN_7) will be smoothed, which may give rise to error. Error can also occur if A/D conversion is frequently performed in scan mode so that the current that charges and discharges the capacitor in the sample-and-hold circuit of the A/D converter becomes greater than that input to the analog input pins via input impedance (R_{in}). The circuit constants should therefore be selected carefully.

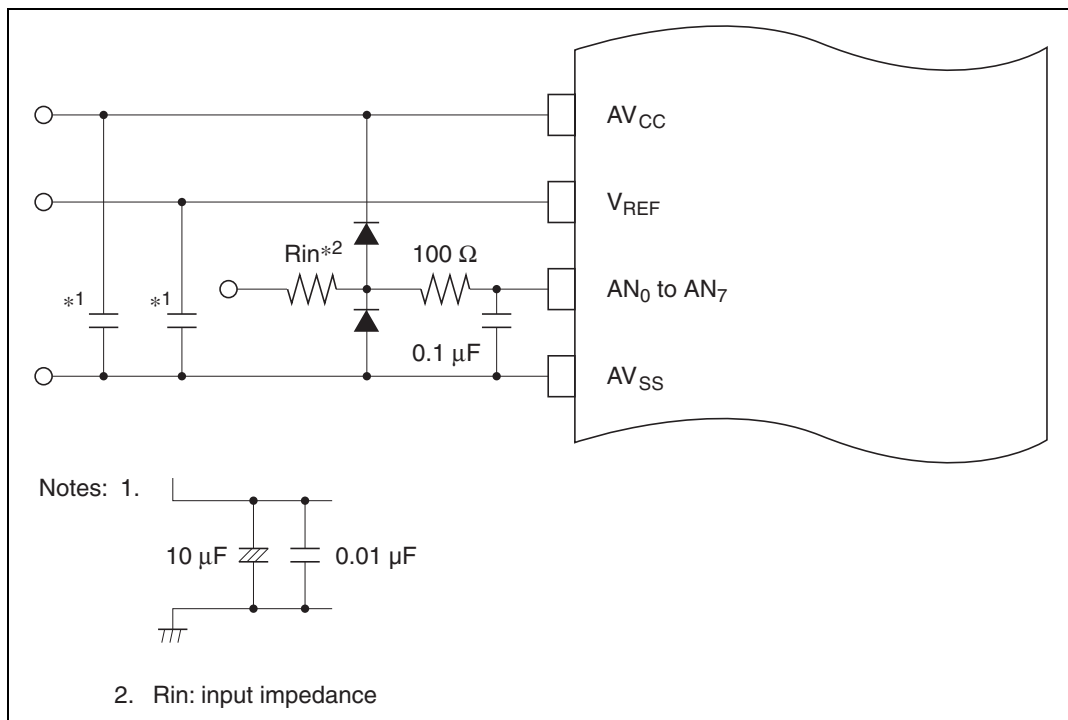


Figure 14.7 Example of Analog Input Protection Circuit

Table 14.5 Analog Input Pin Ratings

| Item | Min | Max | Unit |
|-----------------------------------|-----|-----|------|
| Analog input capacitance | — | 20 | pF |
| Allowable signal-source impedance | — | 10* | kΩ |

Note: * When conversion time = 134 states, $V_{CC} = 4.0\text{ V to }5.5\text{ V}$, and $\phi \leq 13\text{ MHz}$. For details, see section 19. Electrical Characteristics.

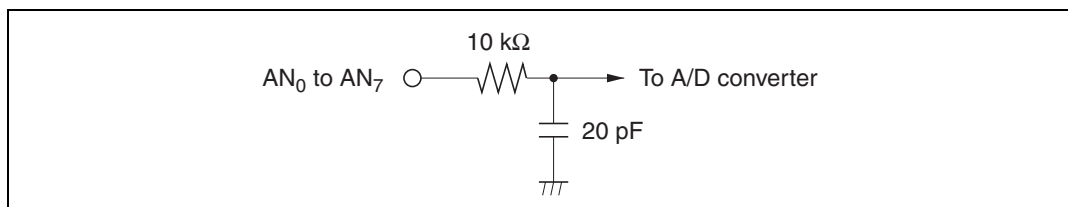


Figure 14.8 Analog Input Pin Equivalent Circuit

Note: Numeric values are approximate, except in table 14.5

6. A/D Conversion Accuracy Definitions

A/D conversion accuracy in the H8/3008 is defined as follows:

— Resolution

Digital output code length of A/D converter

— Offset error

Deviation from ideal A/D conversion characteristic of analog input voltage required to raise digital output from minimum voltage value 0000000000 to 0000000001 (figure 14.10)

— Full-scale error

Deviation from ideal A/D conversion characteristic of analog input voltage required to raise digital output from 1111111110 to 1111111111 (figure 14.10)

— Quantization error

Intrinsic error of the A/D converter; 1/2 LSB (figure 14.9)

— Nonlinearity error

Deviation from ideal A/D conversion characteristic in range from zero volts to full scale, exclusive of offset error, full-scale error, and quantization error.

— Absolute accuracy

Deviation of digital value from analog input value, including offset error, full-scale error, quantization error, and nonlinearity error.

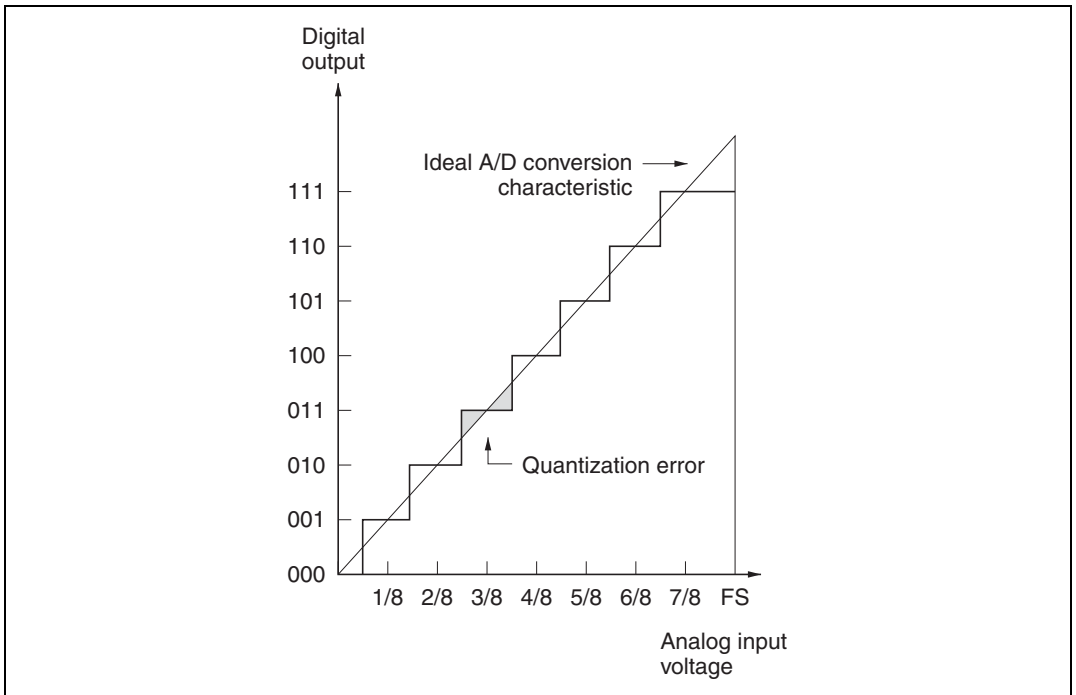


Figure 14.9 A/D Converter Accuracy Definitions (1)

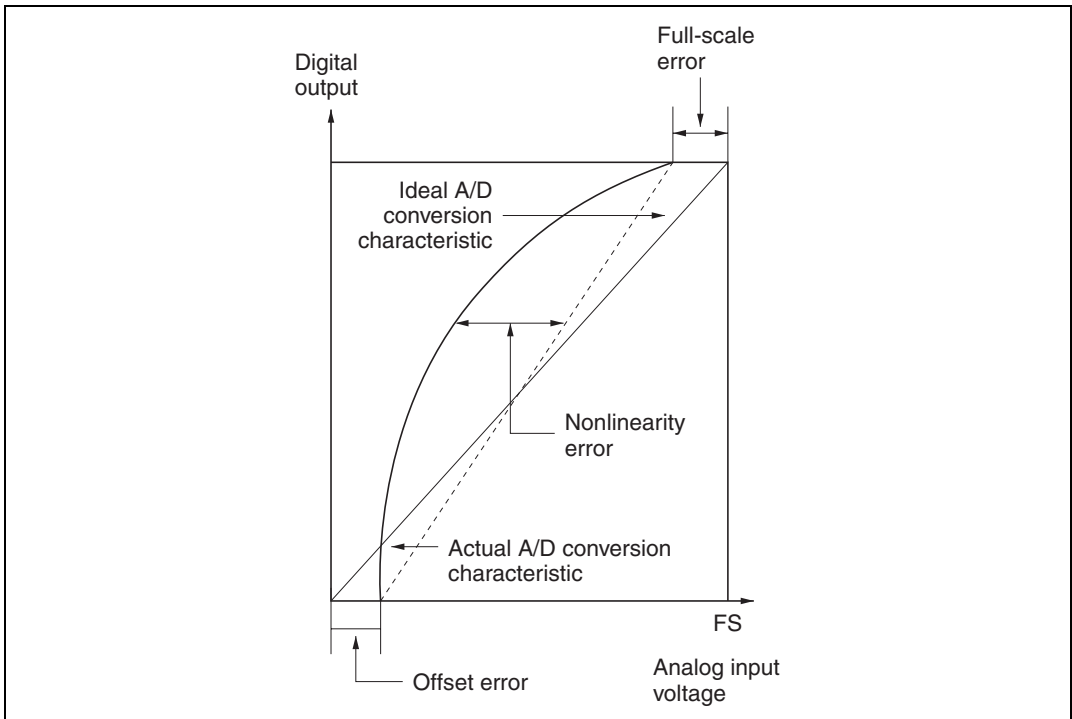


Figure 14.10 A/D Converter Accuracy Definitions (2)

7. Allowable Signal-Source Impedance

The analog inputs of the H8/3008 are designed to assure accurate conversion of input signals with a signal-source impedance not exceeding 10 k Ω . The reason for this rating is that it enables the input capacitor in the sample-and-hold circuit in the A/D converter to charge within the sampling time. If the sensor output impedance exceeds 10 k Ω , charging may be inadequate and the accuracy of A/D conversion cannot be guaranteed.

If a large external capacitor is provided in single mode, then the internal 10-k Ω input resistance becomes the only significant load on the input. In this case the impedance of the signal source is not a problem.

A large external capacitor, however, acts as a low-pass filter. This may make it impossible to track analog signals with high dv/dt (e.g. a variation of 5 mV/ μ s) (figure 14.11). To convert high-speed analog signals or to use scan mode, insert a low-impedance buffer.

8. Effect on Absolute Accuracy

Attaching an external capacitor creates a coupling with ground, so if there is noise on the ground line, it may degrade absolute accuracy. The capacitor must be connected to an electrically stable ground, such as AV_{SS} .

If a filter circuit is used, be careful of interference with digital signals on the same board, and make sure the circuit does not act as an antenna.

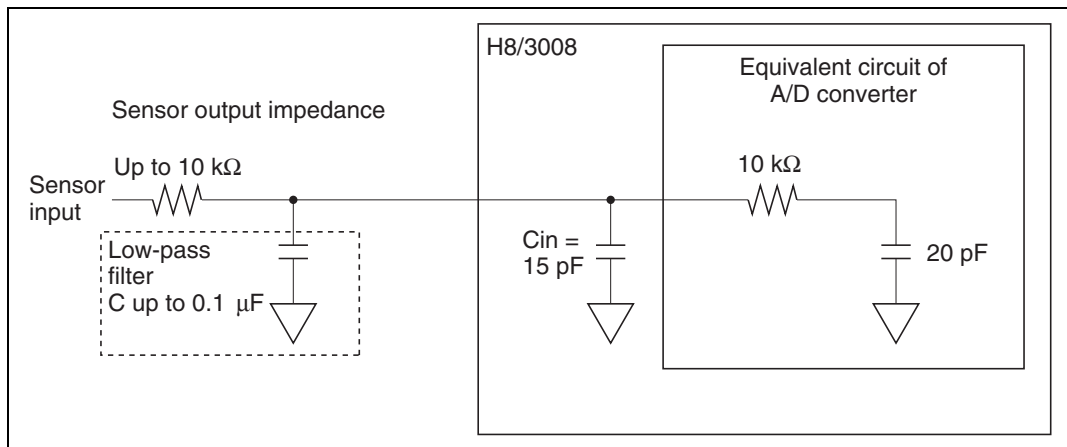


Figure 14.11 Analog Input Circuit (Example)

Section 15 D/A Converter

15.1 Overview

The H8/3008 includes a D/A converter with two channels.

15.1.1 Features

D/A converter features are listed below.

- Eight-bit resolution
- Two output channels
- Conversion time: maximum 10 μ s (with 20-pF capacitive load)
- Output voltage: 0 V to V_{REF}
- D/A outputs can be sustained in software standby mode

15.1.2 Block Diagram

Figure 15.1 shows a block diagram of the D/A converter.

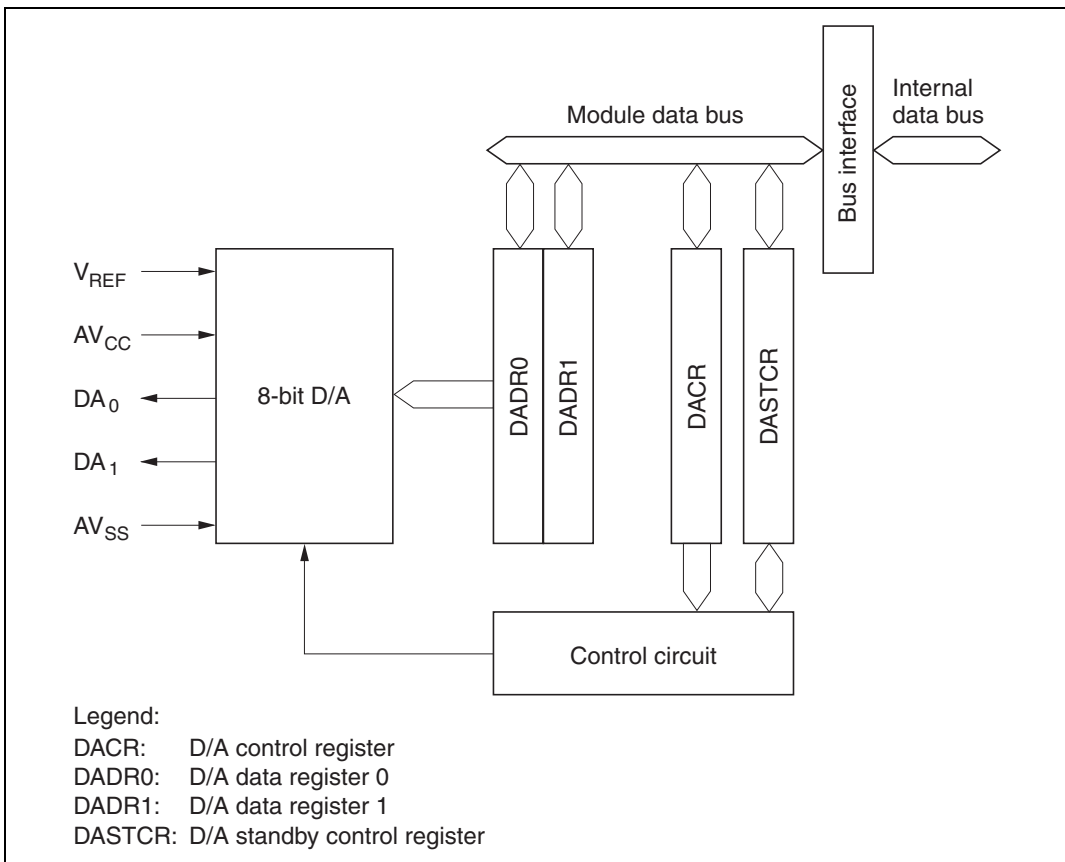


Figure 15.1 D/A Converter Block Diagram

15.1.3 Pin Configuration

Table 15.1 summarizes the D/A converter's input and output pins.

Table 15.1 D/A Converter Pins

| Pin Name | Abbreviation | I/O | Function |
|-----------------------------|--------------|--------|---|
| Analog power supply pin | AV_{SS} | Input | Analog power supply and reference voltage |
| Analog ground pin | AV_{SS} | Input | Analog ground and reference voltage |
| Analog output pin 0 | DA_0 | Output | Analog output, channel 0 |
| Analog output pin 1 | DA_1 | Output | Analog output, channel 1 |
| Reference voltage input pin | V_{REF} | Input | Analog reference voltage |

15.1.4 Register Configuration

Table 15.2 summarizes the D/A converter's registers.

Table 15.2 D/A Converter Registers

| Address* | Name | Abbreviation | R/W | Initial Value |
|----------|------------------------------|--------------|-----|---------------|
| H'FFF9C | D/A data register 0 | DADR0 | R/W | H'00 |
| H'FFF9D | D/A data register 1 | DADR1 | R/W | H'00 |
| H'FFF9E | D/A control register | DACR | R/W | H'1F |
| H'EE01A | D/A standby control register | DASTCR | R/W | H'FE |

Note: * Lower 20 bits of the address in advanced mode.

15.2 Register Descriptions

15.2.1 D/A Data Registers 0 and 1 (DADR0, DADR1)

| | | | | | | | | |
|---------------|-----|-----|-----|-----|-----|-----|-----|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | | | | | | | | |
| Initial value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

The D/A data registers (DADR0 and DADR1) are 8-bit readable/writable registers that store the data to be converted. When analog output is enabled, the D/A data register values are constantly converted and output at the analog output pins.

The D/A data registers are initialized to H'00 by a reset and in standby mode.

When the DASTE bit is set to 1 in the D/A standby control register (DASTCR), the D/A registers are not initialized in software standby mode.

15.2.2 D/A Control Register (DACR)

| | | | | | | | | |
|---------------|-------|-------|-----|---|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | DAOE1 | DAOE0 | DAE | — | — | — | — | — |
| Initial value | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |
| Read/Write | R/W | R/W | R/W | — | — | — | — | — |

D/A enable
 Controls D/A conversion

D/A output enable 0
 Controls D/A conversion and analog output

D/A output enable 1
 Controls D/A conversion and analog output

DACR is an 8-bit readable/writable register that controls the operation of the D/A converter. DACR is initialized to H'1F by a reset and in standby mode.

When the DASTE bit is set to 1 in the D/A standby control register (DASTCR), the D/A registers are not initialized in software standby mode.

Bit 7—D/A Output Enable 1 (DAOE1): Controls D/A conversion and analog output.

Bit 7**DAOE1 Description**

| | |
|---|--|
| 0 | DA ₁ analog output is disabled |
| 1 | Channel-1 D/A conversion and DA ₁ analog output are enabled |

Bit 6—D/A Output Enable 0 (DAOE0): Controls D/A conversion and analog output.

Bit 6**DAOE0 Description**

| | |
|---|--|
| 0 | DA ₀ analog output is disabled |
| 1 | Channel-0 D/A conversion and DA ₀ analog output are enabled |

Bit 5—D/A Enable (DAE): Controls D/A conversion, together with bits DAOE0 and DAOE1. When the DAE bit is cleared to 0, analog conversion is controlled independently in channels 0 and 1. When the DAE bit is set to 1, analog conversion is controlled together in channels 0 and 1. Output of the conversion results is always controlled independently by DAOE0 and DAOE1.

Bit 7 Bit 6 Bit 5**DAOE1 DAOE0 DAE Description**

| | | | |
|---|---|---|---|
| 0 | 0 | — | D/A conversion is disabled in channels 0 and 1 |
| 0 | 1 | 0 | D/A conversion is enabled in channel 0 D/A conversion is disabled in channel 1 |
| 0 | 1 | 1 | D/A conversion is enabled in channels 0 and 1 |
| 1 | 0 | 0 | D/A conversion is disabled in channel 0 D/A conversion is enabled in channel 1 |
| 1 | 0 | 1 | D/A conversion is enabled in channels 0 and 1 |
| 1 | 1 | — | D/A conversion is enabled in channels 0 and 1 |

When the DAE bit is set to 1, even if bits DAOE0 and DAOE1 in DACR and the ADST bit in ADCSR are cleared to 0, the same current is drawn from the analog power supply as during A/D and D/A conversion.

Bits 4 to 0—Reserved: These bits cannot be modified and are always read as 1.

15.2.3 D/A Standby Control Register (DASTCR)

DASTCR is an 8-bit readable/writable register that enables or disables D/A output in software standby mode.

| | | | | | | | | |
|---------------|---|---|---|---|---|---|---|-------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | DASTE |
| Initial value | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| Read/Write | — | — | — | — | — | — | — | R/W |

Reserved bits

D/A standby enable
Enables or disables D/A output
in software standby mode

DASTCR is initialized to H'FE by a reset and in hardware standby mode. It is not initialized in software standby mode.

Bits 7 to 1—Reserved: These bits cannot be modified and are always read as 1.

Bit 0—D/A Standby Enable (DASTE): Enables or disables D/A output in software standby mode.

Bit 0

| DASTE | Description | |
|-------|---|-----------------|
| 0 | D/A output is disabled in software standby mode | (Initial value) |
| 1 | D/A output is enabled in software standby mode | |

15.3 Operation

The D/A converter has two built-in D/A conversion circuits that can perform conversion independently.

D/A conversion is performed constantly while enabled in DACR. If the DADR0 or DADR1 value is modified, conversion of the new data begins immediately. The conversion results are output when bits DAOE0 and DAOE1 are set to 1.

An example of D/A conversion on channel 0 is given next. Timing is indicated in figure 15.2.

1. Data to be converted is written in DADR0.
2. Bit DAOE0 is set to 1 in DACR. D/A conversion starts and DA₀ becomes an output pin. The converted result is output after the conversion time.

$$\text{The output value is } \frac{\text{DADR contents}}{256} \times V_{\text{REF}}$$

Output of this conversion result continues until the value in DADR0 is modified or the DAOE0 bit is cleared to 0.

3. If the DADR0 value is modified, conversion starts immediately, and the result is output after the conversion time.
4. When the DAOE0 bit is cleared to 0, DA0 becomes an input pin.

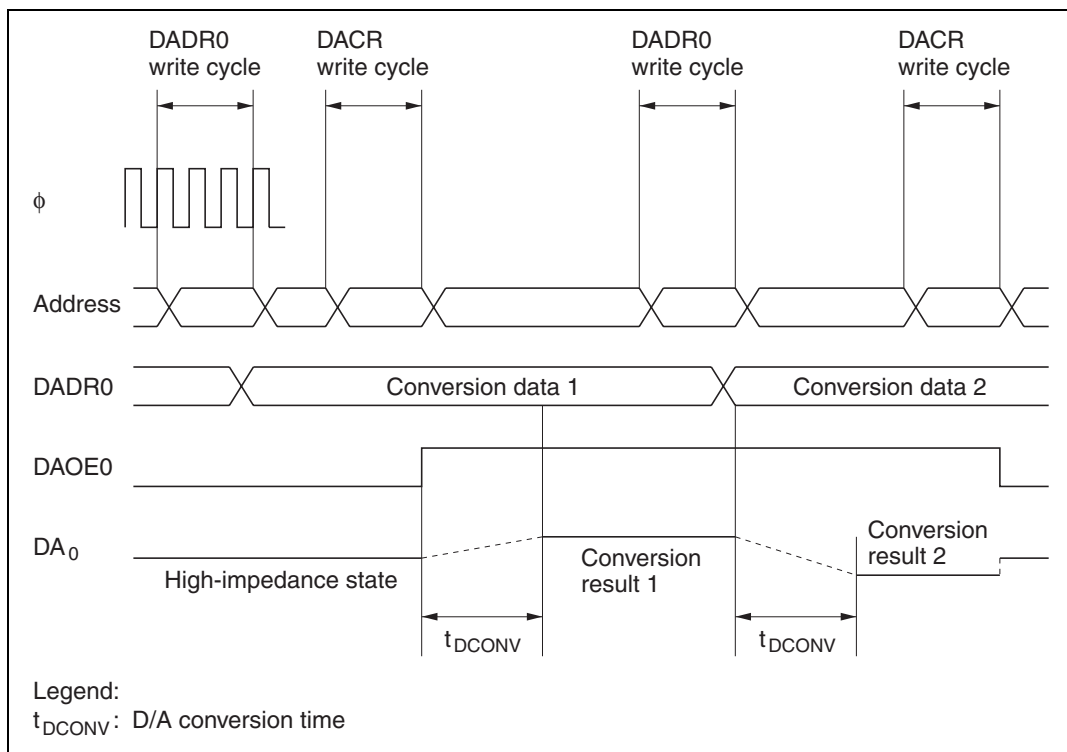


Figure 15.2 Example of D/A Converter Operation

15.4 D/A Output Control

In the H8/3008, D/A converter output can be enabled or disabled in software standby mode.

When the DASTE bit is set to 1 in DASTCR, D/A converter output is enabled in software standby mode. The D/A converter registers retain the values they held prior to the transition to software standby mode.

When D/A output is enabled in software standby mode, the reference supply current is the same as during normal operation.

Section 16 RAM

16.1 Overview

The H8/3008 has high-speed static RAM on-chip. The RAM is connected to the CPU by a 16-bit data bus. The CPU accesses both byte data and word data in two states, making the RAM useful for rapid data transfer.

The on-chip RAM can be enabled or disabled with the RAM enable bit (RAME) in the system control register (SYSCR). When the on-chip RAM is disabled, that area is assigned to external space in the expanded modes. The on-chip RAM specifications for the H8/3008 are shown in table 16.1.

Table 16.1 H8/3008 On-Chip RAM Specifications

| RAM size | 4 kbytes | |
|--------------------|-----------------|--------------------|
| Address assignment | Modes 1, 2 | H'FEF20 to H'FFF1F |
| | Modes 3, 4 | H'FFE20 to H'FFF1F |

16.1.1 Block Diagram

Figure 16.1 shows a block diagram of the on-chip RAM.

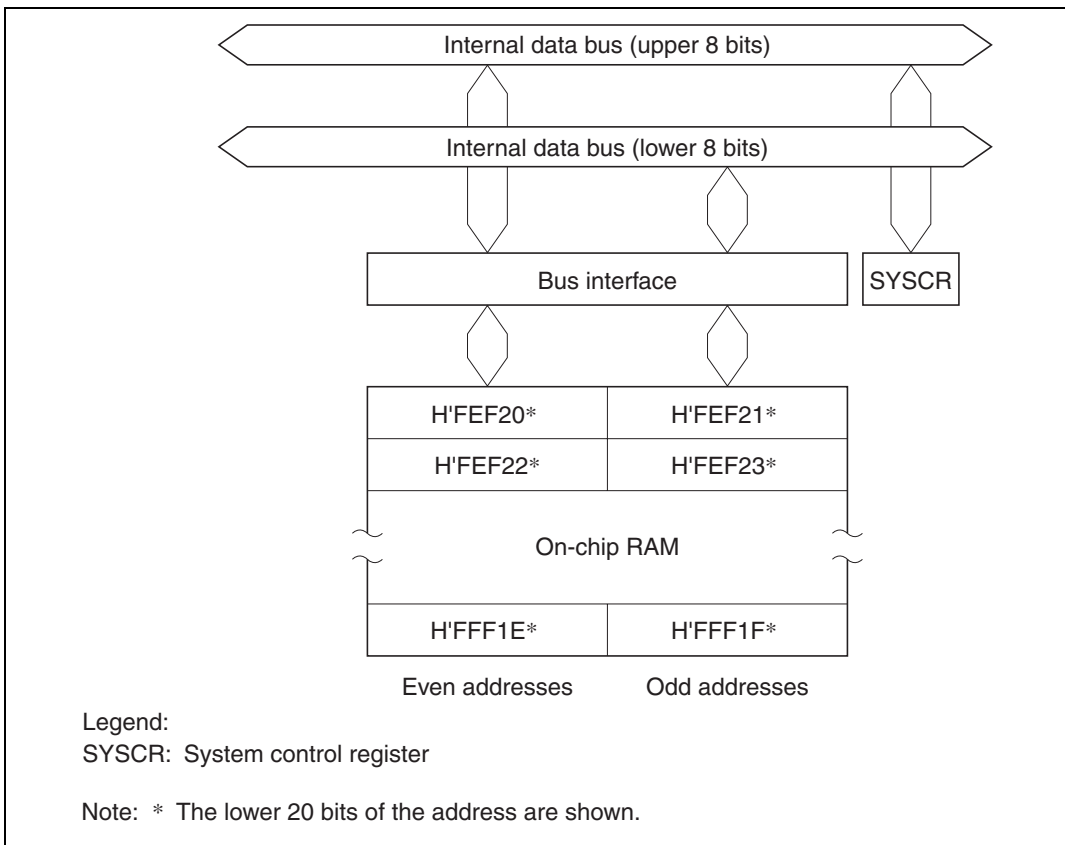


Figure 16.1 RAM Block Diagram

16.1.2 Register Configuration

The on-chip RAM is controlled by SYSCR. Table 16.2 gives the address and initial value of SYSCR.

Table 16.2 System Control Register

| Address* | Name | Abbreviation | R/W | Initial Value |
|----------|-------------------------|--------------|-----|---------------|
| H'EE012 | System control register | SYSCR | R/W | H'09 |

Note: * Lower 20 bits of the address in advanced mode.

16.2 System Control Register (SYSCR)

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|------|------|------|------|-----|-------|------|------|
| | SSBY | STS2 | STS1 | STS0 | UE | NMIEG | SSOE | RAME |
| Initial value | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Software standby
 Standby timer select 2 to 0
 User bit enable
 NMI edge select
 Software standby output port enable
 RAM enable bit
 Enables or disables on-chip RAM

One function of SYSCR is to enable or disable access to the on-chip RAM. The on-chip RAM is enabled or disabled by the RAME bit in SYSCR. For details about the other bits, see section 3.3, System Control Register (SYSCR).

Bit 0—RAM Enable (RAME): Enables or disables the on-chip RAM. The RAME bit is initialized at the rising edge of the input at the $\overline{\text{RES}}$ pin. It is not initialized in software standby mode.

Bit 0

| RAME | Description |
|------|--|
| 0 | On-chip RAM is disabled |
| 1 | On-chip RAM is enabled (Initial value) |

16.3 Operation

When the RAME bit is set to 1, the on-chip RAM is enabled. Accesses to the addresses shown in table 16.1 are directed to the on-chip RAM. In modes 1 to 4 (expanded modes), when the RAME bit is cleared to 0, the off-chip address space is accessed.

Since the on-chip RAM is connected to the CPU by an internal 16-bit data bus, it can be written and read by word access. It can also be written and read by byte access. Byte data is accessed in two states using the upper 8 bits of the data bus. Word data starting at an even address is accessed in two states using all 16 bits of the data bus.

Section 17 Clock Pulse Generator

17.1 Overview

The H8/3008 has a built-in clock pulse generator (CPG) that generates the system clock (ϕ) and other internal clock signals ($\phi/2$ to $\phi/4096$). After duty adjustment, a frequency divider divides the clock frequency to generate the system clock (ϕ). The system clock is output at the ϕ pin*¹ and furnished as a master clock to prescalers that supply clock signals to the on-chip supporting modules. Frequency division ratios of 1/1, 1/2, 1/4, and 1/8 can be selected for the frequency divider by settings in a division control register (DIVCR)*². Power consumption in the chip is reduced in almost direct proportion to the frequency division ratio.

- Notes:
1. Usage of the ϕ pin differs depending on the chip operating mode and the PSTOP bit setting in the module standby control register (MSTCR). For details, see section 18.7, System Clock Output Disabling Function.
 2. The division ratio of the frequency divider can be changed dynamically during operation. The clock output at the ϕ pin also changes when the division ratio is changed. The frequency output at the ϕ pin is shown below.

$$\phi = \text{EXTAL} \times n$$

where, EXTAL: Frequency of crystal resonator or external clock signal

n: Frequency division ratio (n = 1/1, 1/2, 1/4, or 1/8)

17.1.1 Block Diagram

Figure 17.1 shows a block diagram of the clock pulse generator.

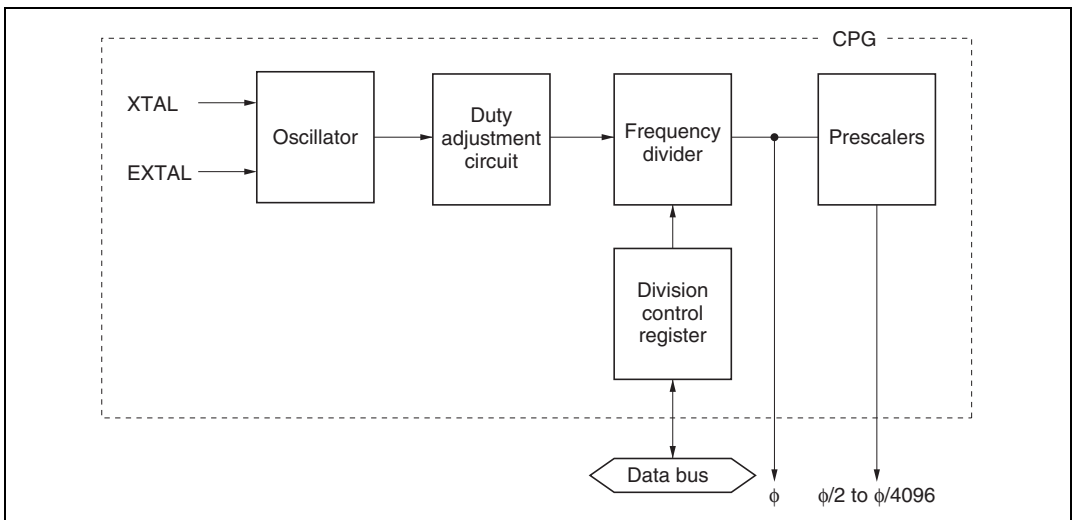


Figure 17.1 Block Diagram of Clock Pulse Generator

17.2 Oscillator Circuit

Clock pulses can be supplied by connecting a crystal resonator, or by input of an external clock signal.

17.2.1 Connecting a Crystal Resonator

Circuit Configuration: A crystal resonator can be connected as in the example in figure 17.2. Damping resistance R_d should be selected according to table 17.1 (1), and external capacitances C_{L1} and C_{L2} according to table 17.1 (2). An AT-cut parallel-resonance crystal should be used.

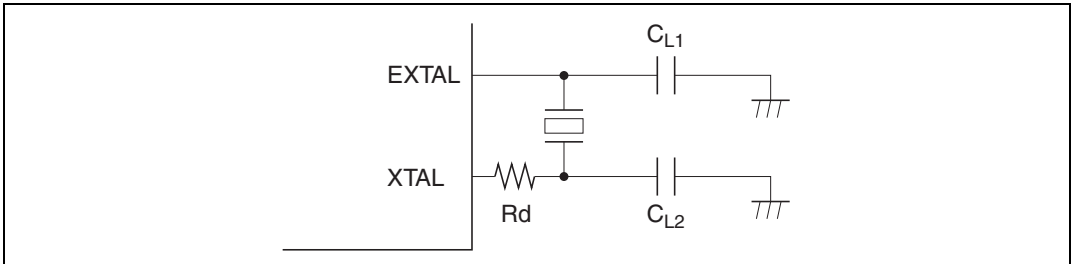


Figure 17.2 Connection of Crystal Resonator (Example)

If a crystal resonator with a frequency higher than 20 MHz is connected, the external load capacitance values in table 17.1 (2) should not exceed 10 [pF]. Also, in order to improve the accuracy of the oscillation frequency, a thorough study of oscillation matching evaluation, etc., should be carried out when deciding the circuit constants.

Table 17.1 (1) Damping Resistance Value

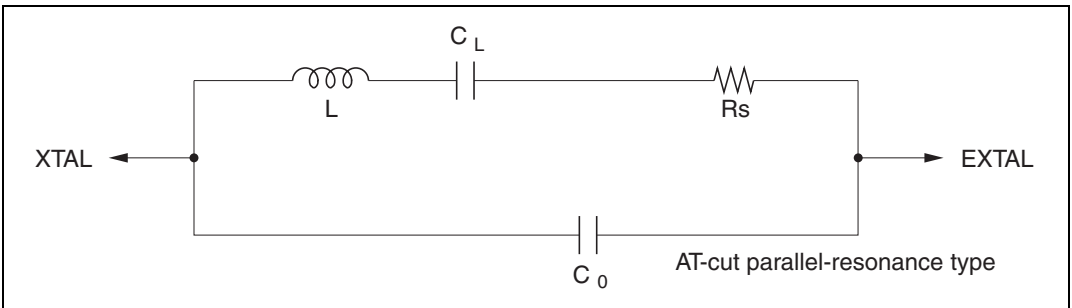
| Damping Resistance Value | Frequency f (MHz) | | | | | | | |
|--------------------------|---------------------|----------------|----------------|-----------------|------------------|------------------|------------------|------------------|
| | 2 | $2 < f \leq 4$ | $4 < f \leq 8$ | $8 < f \leq 10$ | $10 < f \leq 13$ | $13 < f \leq 16$ | $16 < f \leq 18$ | $18 < f \leq 25$ |
| R_d (Ω) | 1 k | 500 | 200 | 0 | 0 | 0 | 0 | 0 |

Note: A crystal resonator between 2 MHz and 25 MHz can be used. If the chip is to be operated at less than 2 MHz, the on-chip frequency divider should be used. (A crystal resonator of less than 2 MHz cannot be used.)

Table 17.1 (2) External Capacitance Values

| External Capacitance Value | 5 V Version | | 3 V Version | |
|----------------------------|------------------|--------------------|--------------------|---------------------|
| Frequency f (MHz) | $20 < f \leq 25$ | $2 \leq f \leq 20$ | $2 \leq f \leq 16$ | $16 \leq f \leq 25$ |
| $C_{L1} = C_{L2}$ (pF) | 10 | 10 to 22 | 22 | 10 |

Crystal Resonator: Figure 17.3 shows an equivalent circuit of the crystal resonator. The crystal resonator should have the characteristics listed in table 17.2.

**Figure 17.3 Crystal Resonator Equivalent Circuit****Table 17.2 Crystal Resonator Parameters**

| Frequency (MHz) | 2 | 4 | 8 | 10 | 12 | 16 | 18 | 20 | 25 |
|------------------------|----------|-----|----|----|----|----|----|----|----|
| R_s max (Ω) | 500 | 120 | 80 | 70 | 60 | 50 | 40 | 40 | 40 |
| C_o (pF) | 7 pF max | | | | | | | | |

Use a crystal resonator with a frequency equal to the system clock frequency (ϕ).

Notes on Board Design: When a crystal resonator is connected, the following points should be noted:

Other signal lines should be routed away from the oscillator circuit to prevent induction from interfering with correct oscillation. See figure 17.4.

When the board is designed, the crystal resonator and its load capacitors should be placed as close as possible to the XTAL and EXTAL pins.

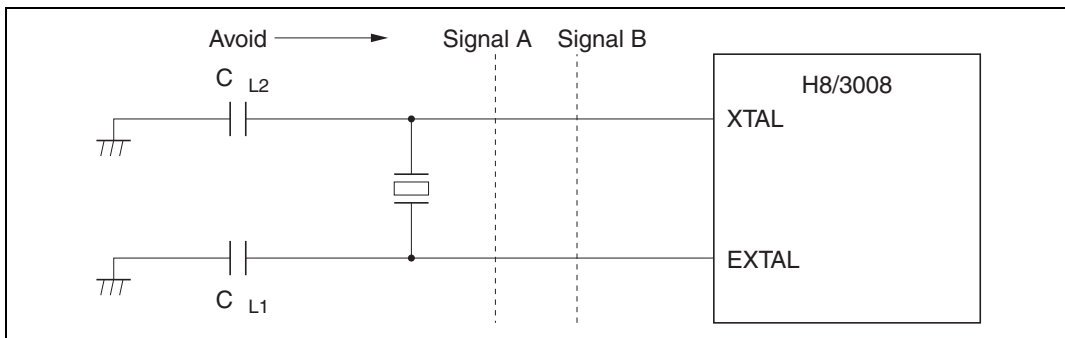


Figure 17.4 Oscillator Circuit Block Board Design Precautions

17.2.2 External Clock Input

Circuit Configuration: An external clock signal can be input as shown in the examples in figure 17.5. If the XTAL pin is left open, the stray capacitance should not exceed 10 pF. If the stray capacitance at the XTAL pin exceeds 10 pF in configuration a, use the connection shown in configuration b instead, and hold the external clock high in standby mode.

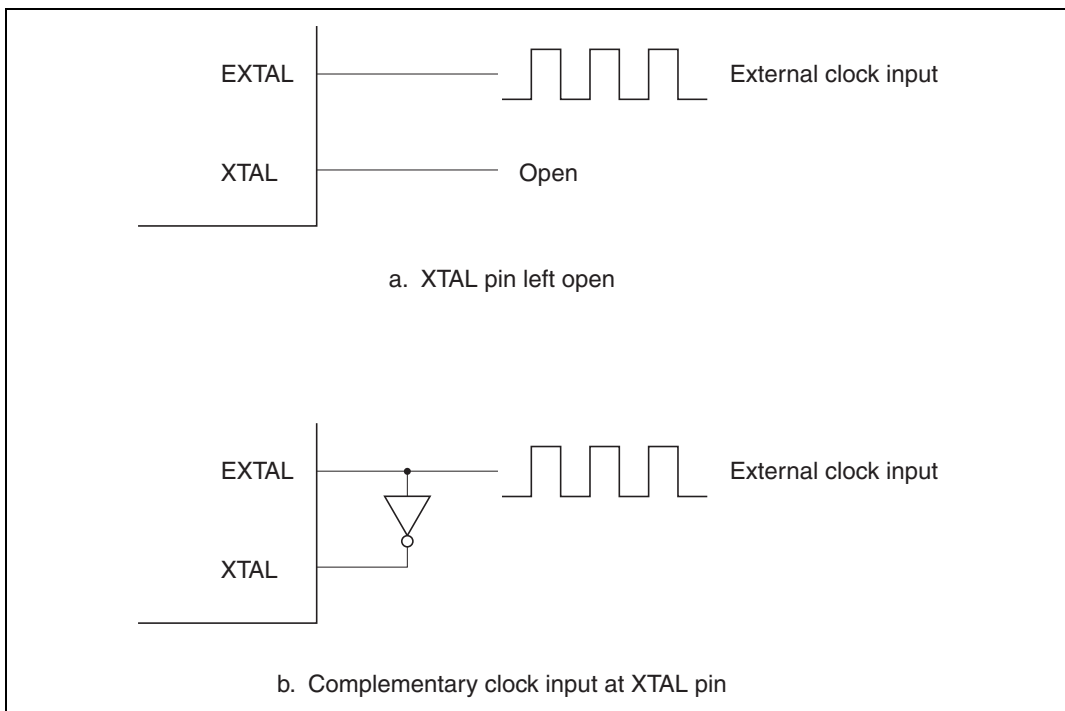


Figure 17.5 External Clock Input (Examples)

External Clock: The external clock frequency should be equal to the system clock frequency when not divided by the on-chip frequency divider. Table 17.3 shows the clock timing, figure 17.6 shows the external clock input timing, and figure 17.7 shows the external clock output settling delay timing. When the appropriate external clock is input via the EXTAL pin, its waveform is corrected by the on-chip oscillator and duty adjustment circuit.

When the appropriate external clock is input via the EXTAL pin, its waveform is corrected by the on-chip oscillator and duty adjustment circuit. The resulting stable clock is output to external devices after the external clock settling time (t_{DEXT}) has passed after the clock input. The system must remain reset with the reset signal low during t_{DEXT} , while the clock output is unstable.

Table 17.3 Clock Timing (Preliminary)

| Item | Symbol | $V_{\text{CC}} = 3.0 \text{ V}$ to 3.6 V | | $V_{\text{CC}} = 5.0 \text{ V}$ $\pm 10 \%$ | | Unit | Test Conditions | |
|---|---------------------|---|-----|--|-----|------------------|---------------------------|-------------|
| | | Min | Max | Min | Max | | | |
| External clock input low pulse width | t_{EXL} | 15 | — | 15 | — | ns | Figure 17.6 | |
| External clock input high pulse width | t_{EXH} | 15 | — | 15 | — | ns | | |
| External clock rise time | t_{EXr} | — | 5 | — | 5 | ns | | |
| External clock fall time | t_{EXf} | — | 5 | — | 5 | ns | | |
| Clock low pulse width | t_{CL} | 0.4 | 0.6 | 0.4 | 0.6 | t_{cyc} | $\phi \geq 5 \text{ MHz}$ | Figure 19.7 |
| | | 80 | — | 80 | — | ns | $\phi < 5 \text{ MHz}$ | |
| Clock high pulse width | t_{CH} | 0.4 | 0.6 | 0.4 | 0.6 | t_{cyc} | $\phi \geq 5 \text{ MHz}$ | |
| | | 80 | — | 80 | — | ns | $\phi < 5 \text{ MHz}$ | |
| External clock output settling delay time | t_{DEXT}^* | 500 | — | 500 | — | μs | Figure 17.7 | |

Note: * t_{DEXT} includes a $\overline{\text{RES}}$ pulse width (t_{RESW}). $t_{\text{RESW}} = 20 t_{\text{cyc}}$

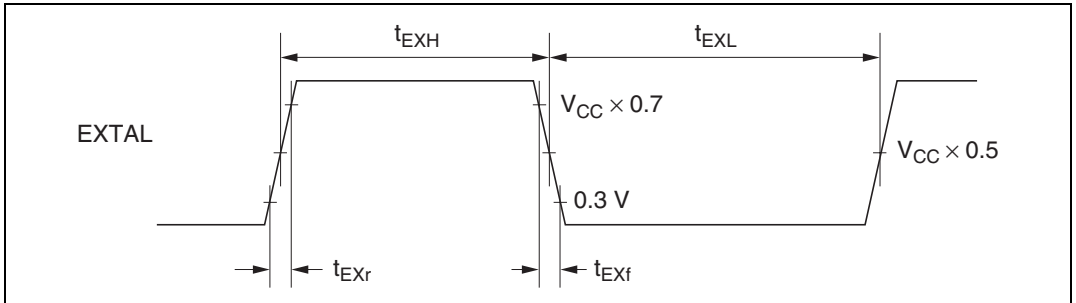


Figure 17.6 External Clock Input Timing

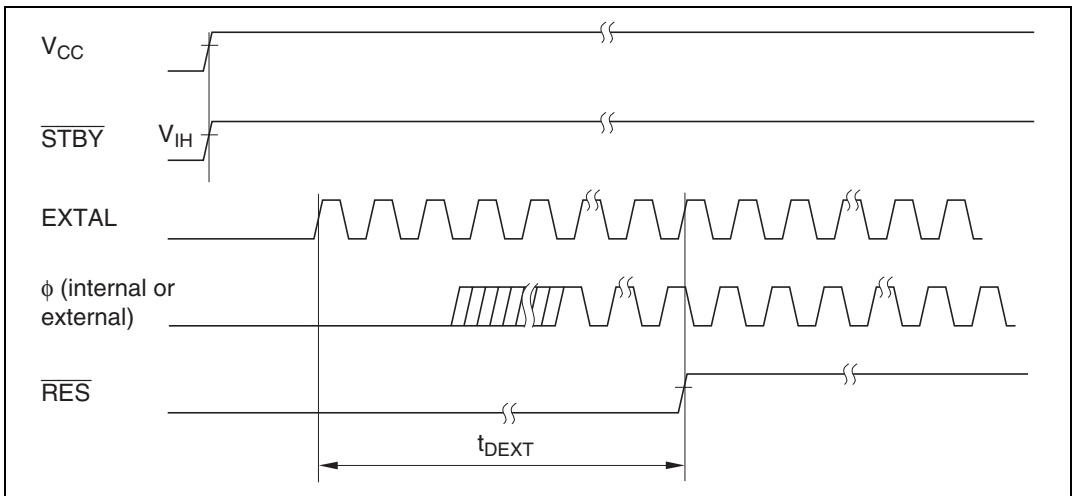


Figure 17.7 External Clock Output Settling Delay Timing

17.3 Duty Adjustment Circuit

When the oscillator frequency is 5 MHz or higher, the duty adjustment circuit adjusts the duty cycle of the clock signal from the oscillator to generate ϕ .

17.4 Prescalers

The prescalers divide the system clock (ϕ) to generate internal clocks ($\phi/2$ to $\phi/4096$).

17.5 Frequency Divider

The frequency divider divides the duty-adjusted clock signal to generate the system clock (ϕ). The frequency division ratio can be changed dynamically by modifying the value in DIVCR, as described below. Power consumption in the chip is reduced in almost direct proportion to the frequency division ratio. The system clock generated by the frequency divider can be output at the ϕ pin.

17.5.1 Register Configuration

Table 17.4 summarizes the frequency division register.

Table 17.4 Frequency Division Register

| Address* | Name | Abbreviation | R/W | Initial Value |
|----------|---------------------------|--------------|-----|---------------|
| H'EE01B | Division control register | DIVCR | R/W | H'FC |

Note: * Lower 20 bits of the address in advanced mode.

17.5.2 Division Control Register (DIVCR)

DIVCR is an 8-bit readable/writable register that selects the division ratio of the frequency divider.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|---|---|---|---|---|---|------|------|
| | — | — | — | — | — | — | DIV1 | DIV0 |
| Initial value | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |
| Read/Write | — | — | — | — | — | — | R/W | R/W |

Reserved bits

Divide bits 1 and 0
 These bits select the frequency division ratio

DIVCR is initialized to H'FC by a reset and in hardware standby mode. It is not initialized in software standby mode.

Bits 7 to 2—Reserved: These bits cannot be modified and are always read as 1.

Bits 1 and 0—Divide (DIV1, DIV0): These bits select the frequency division ratio, as follows.

| Bit 1 DIV1 | Bit 0 DIV0 | Frequency Division Ratio | |
|---------------|---------------|--------------------------|-----------------|
| 0 | 0 | 1/1 | (Initial value) |
| 0 | 1 | 1/2 | |
| 1 | 0 | 1/4 | |
| 1 | 1 | 1/8 | |

17.5.3 Usage Notes

The DIVCR setting changes the ϕ frequency, so note the following points.

- Select a frequency division ratio that stays within the assured operation range specified for the clock cycle time t_{cyc} in the AC electrical characteristics. Note that ϕ_{min} = lower limit of the operating frequency range. Ensure that ϕ is not below this lower limit.
- All on-chip module operations are based on ϕ . Note that the timing of timer operations, serial communication, and other time-dependent processing differs before and after any change in the division ratio. The waiting time for exit from software standby mode also changes when the division ratio is changed. For details, see section 18.4.3, Selection of Waiting Time for Exit from Software Standby Mode.

Section 18 Power-Down State

18.1 Overview

The H8/3008 has a power-down state that greatly reduces power consumption by halting the CPU, and a module standby function that reduces power consumption by selectively halting on-chip modules.

The power-down state includes the following three modes:

- Sleep mode
- Software standby mode
- Hardware standby mode

The module standby function can halt on-chip supporting modules independently of the power-down state. The modules that can be halted are the 16-bit timer, 8-bit timer, SCI0, SCI1, and A/D converter.

Table 18.1 indicates the methods of entering and exiting the power-down modes and module standby mode, and gives the status of the CPU and on-chip supporting modules in each mode.

Table 18.1 Power-Down State and Module Standby Function

| Mode | State | | | | | | | | | | Exiting Conditions | | | |
|-----------------------|--|--------|--------|---------------|--------------------|--------------------|--------------------|--------------------|--------------------|------------------|--------------------|------------------|-----------------------|--|
| | Entering Conditions | Clock | CPU | CPU Registers | 16-Bit Timer | 8-Bit Timer | SCIO | SC11 | A/D | Other Modules | | RAM | ϕ clock Output*3 | I/O Ports |
| Sleep mode | SLEEP instruction executed while SSBY = 0 in SYSCR | Active | Halted | Held | Active | Active | Active | Active | Active | Active | Held | ϕ output | Held | Interrupt • RES • STBY |
| Software standby mode | SLEEP instruction executed while SSBY = 1 in SYSCR | Halted | Halted | Held | Halted and reset | Halted and reset | Halted and reset | Halted and reset | Halted and reset | Halted and reset | Held | High output | Held | NMI • \overline{IRQ}_0 to \overline{IRQ}_2 • RES • STBY |
| Hardware standby mode | Low input at STBY pin | Halted | Halted | Undetermined | Halted and reset | Halted and reset | Halted and reset | Halted and reset | Halted and reset | Halted and reset | Held*2 | High impedance | High impedance | • STBY • RES |
| Module standby | Corresponding bit set to 1 in MSTRH and MSTRCL | Active | Active | — | Halted*1 and reset | Halted*1 and reset | Halted*1 and reset | Halted*1 and reset | Halted*1 and reset | Active | — | High impedance*1 | — | • STBY • RES • Clear MSTR bit to 0*4 |

- Notes: 1. State in which the corresponding MSTR bit was set to 1. For details see section 18.2.2, Module Standby Control Register H (MSTRH) and section 18.2.3, Module Standby Control Register L (MSTRCL).
2. The RAME bit must be cleared to 0 in SYSCR before the transition from the program execution state to hardware standby mode.
3. When P67 is used as the ϕ output pin.
4. When a MSTR bit is set to 1, the registers of the corresponding on-chip supporting module are initialized. To restart the module, first clear the MSTR bit to 0, then set up the module registers again.

Legend:

SYSCR: System control register
 SSBY: Software standby bit
 MSTRH: Module standby control register H
 MSTRCL: Module standby control register L

18.2 Register Configuration

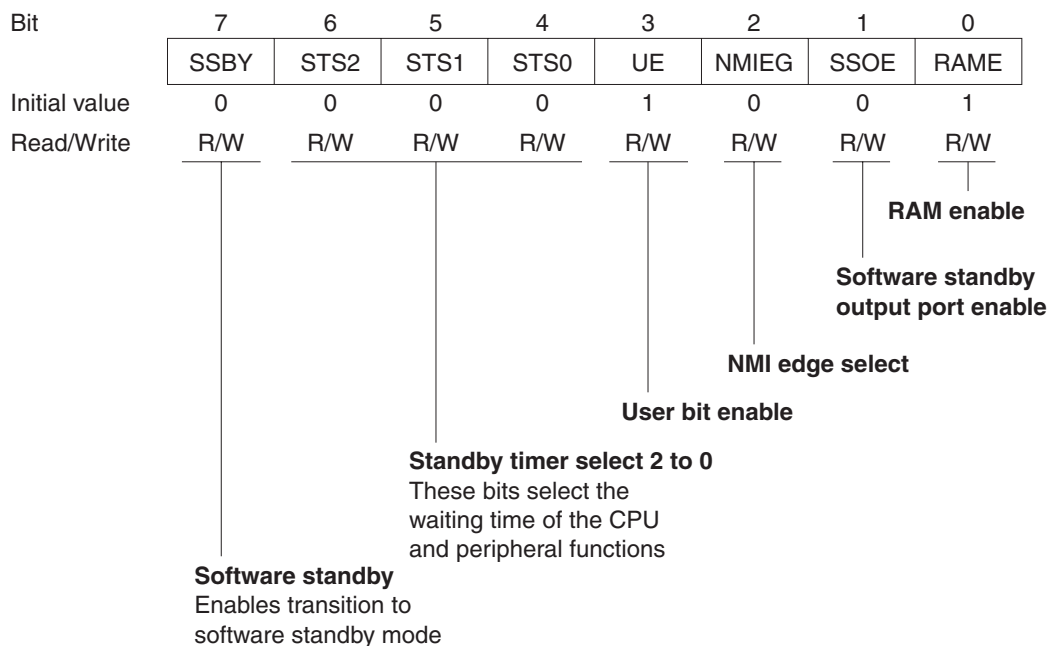
The H8/3008 has a system control register (SYSCR) that controls the power-down state, and module standby control registers H (MSTCRH) and L (MSTCRL) that control the module standby function. Table 18.2 summarizes these registers.

Table 18.2 Control Register

| Address* | Name | Abbreviation | R/W | Initial Value |
|----------|-----------------------------------|--------------|-----|---------------|
| H'EE012 | System control register | SYSCR | R/W | H'09 |
| H'EE01C | Module standby control register H | MSTCRH | R/W | H'78 |
| H'EE01D | Module standby control register L | MSTCRL | R/W | H'00 |

Note: * Lower 20 bits of the address in advanced mode.

18.2.1 System Control Register (SYSCR)



SYSCR is an 8-bit readable/writable register. Bit 7 (SSBY), bits 6 to 4 (STS2 to STS0), and bit 1 (SSOE) control the power-down state. For information on the other SYSCR bits, see section 3.3, System Control Register (SYSCR).

Bit 7—Software Standby (SSBY): Enables transition to software standby mode. When software standby mode is exited by an external interrupt, this bit remains set to 1 after the return to normal operation. To clear this bit, write 0.

Bit 7**SSBY****Description**

| | | |
|---|--|-----------------|
| 0 | SLEEP instruction causes transition to sleep mode | (Initial value) |
| 1 | SLEEP instruction causes transition to software standby mode | |

Bits 6 to 4—Standby Timer Select (STS2 to STS0): These bits select the length of time the CPU and on-chip supporting modules wait for the clock to settle when software standby mode is exited by an external interrupt. If the clock is generated by a crystal resonator, set these bits according to the clock frequency so that the waiting time will be at least 7 ms. See table 18.3. Set these bits according to the operating frequency so that the waiting time will be at least 100 μ s.

Bit 6**STS2****Bit 5****STS1****Bit 4****STS0****Description**

| | | | | |
|---|---|---|-------------------------------|-----------------|
| 0 | 0 | 0 | Waiting time = 8,192 states | (Initial value) |
| | | 1 | Waiting time = 16,384 states | |
| | 1 | 0 | Waiting time = 32,768 states | |
| | | 1 | Waiting time = 65,536 states | |
| 1 | 0 | 0 | Waiting time = 131,072 states | |
| 1 | 0 | 1 | Waiting time = 262,144 states | |
| 1 | 1 | 0 | Waiting time = 1,024 states | |
| 1 | 1 | 1 | Illegal setting | |

Bit 1—Software Standby Output Port Enable (SSOE): Specifies whether the address bus and bus control signals (\overline{CS}_0 to \overline{CS}_7 , \overline{AS} , \overline{RD} , \overline{HWR} , and \overline{LWR}) are kept as outputs or fixed high, or placed in the high-impedance state in software standby mode.

Bit 1**SSOE****Description**

| | | |
|---|---|-----------------|
| 0 | In software standby mode, the address bus and bus control signals are all high-impedance | (Initial value) |
| 1 | In software standby mode, the address bus retains its output state and bus control signals are fixed high | |

18.2.2 Module Standby Control Register H (MSTCRH)

MSTCRH is an 8-bit readable/writable register that controls output of the system clock (ϕ). It also controls the module standby function, which places individual on-chip supporting modules in the standby state. Module standby can be designated for the SCIO, SCI1.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|-------|---|---|---|---|-----|--------|--------|
| | PSTOP | — | — | — | — | — | MSTPH1 | MSTPH0 |
| Initial value | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| Read/Write | R/W | — | — | — | — | R/W | R/W | R/W |

Reserved bits

Module standby H1 to H0
These bits select modules to be placed in standby

ϕ **clock stop**
Enables or disables output of the system clock

MSTCRH is initialized to H'78 by a reset and in hardware standby mode. It is not initialized in software standby mode.

Bit 7— ϕ Clock Stop (PSTOP): Enables or disables output of the system clock (ϕ).

Bit 7

PSTOP

Description

| | | |
|---|---------------------------------|-----------------|
| 0 | System clock output is enabled | (Initial value) |
| 1 | System clock output is disabled | |

Bits 6 to 3—Reserved: These bits cannot be modified and are always read as 1.

Bit 2—Reserved: This bit can be written and read.

Bit 1—Module Standby H1 (MSTPH1): Selects whether to place the SCI1 in standby.

Bit 1

MSTPH1

Description

| | | |
|---|--------------------------|-----------------|
| 0 | SCI1 operates normally | (Initial value) |
| 1 | SCI1 is in standby state | |

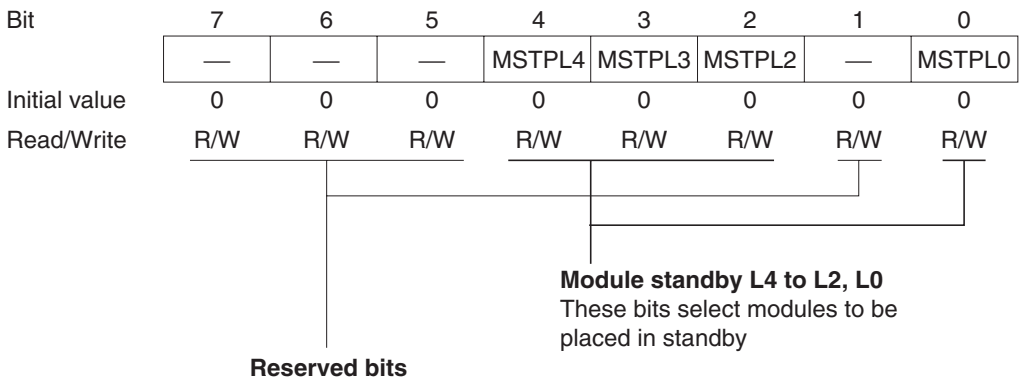
Bit 0—Module Standby H0 (MSTPH0): Selects whether to place the SCI0 in standby.

Bit 0**MSTPH0 Description**

| Bit | Description | (Initial value) |
|-----|--------------------------|-----------------|
| 0 | SCI0 operates normally | (Initial value) |
| 1 | SCI0 is in standby state | |

18.2.3 Module Standby Control Register L (MSTCRL)

MSTCRL is an 8-bit readable/writable register that controls the module standby function, which places individual on-chip supporting modules in the standby state. Module standby can be designated for 16-bit timer, 8-bit timer, and A/D converter modules.



MSTCRL is initialized to H'00 by a reset and in hardware standby mode. It is not initialized in software standby mode.

Bits 7 to 5—Reserved: This bit can be written and read.

Bit 4—Module Standby L4 (MSTPL4): Selects whether to place the 16-bit timer in standby.

Bit 4**MSTPL4 Description**

| Bit | Description | (Initial value) |
|-----|----------------------------------|-----------------|
| 0 | 16-bit timer operates normally | (Initial value) |
| 1 | 16-bit timer is in standby state | |

Bit 3—Module Standby L3 (MSTPL3): Selects whether to place 8-bit timer channels 0 and 1 in standby.

Bit 3**MSTPL3 Description**

| | | |
|---|---|-----------------|
| 0 | 8-bit timer channels 0 and 1 operate normally | (Initial value) |
| 1 | 8-bit timer channels 0 and 1 are in standby state | |

Bit 2—Module Standby L2 (MSTPL2): Selects whether to place 8-bit timer channels 2 and 3 in standby.

Bit 2**MSTPL2 Description**

| | | |
|---|---|-----------------|
| 0 | 8-bit timer channels 2 and 3 operate normally | (Initial value) |
| 1 | 8-bit timer channels 2 and 3 are in standby state | |

Bit 1—Reserved: This bit can be written and read.

Bit 0—Module Standby L0 (MSTPL0): Selects whether to place the A/D converter in standby.

Bit 0**MSTPL0 Description**

| | | |
|---|-----------------------------------|-----------------|
| 0 | A/D converter operates normally | (Initial value) |
| 1 | A/D converter is in standby state | |

18.3 Sleep Mode

18.3.1 Transition to Sleep Mode

When the SSBY bit is cleared to 0 in SYSCR, execution of the SLEEP instruction causes a transition from the program execution state to sleep mode. Immediately after executing the SLEEP instruction the CPU halts, but the contents of its internal registers are retained. On-chip supporting modules do not halt in sleep mode. Modules which have been placed in standby by the module standby function, however, remain halted.

18.3.2 Exit from Sleep Mode

Sleep mode is exited by an interrupt, or by input at the $\overline{\text{RES}}$ or $\overline{\text{STBY}}$ pin.

Exit by Interrupt: An interrupt terminates sleep mode and causes a transition to the interrupt exception handling state. Sleep mode is not exited by an interrupt source in an on-chip supporting module if the interrupt is disabled in the on-chip supporting module. Sleep mode is not exited by an interrupt other than NMI if the interrupt is masked by interrupt priority settings and the settings of the I and UI bits in CCR, IPR.

Exit by $\overline{\text{RES}}$ Input: Low input at the $\overline{\text{RES}}$ pin exits from sleep mode to the reset state.

Exit by $\overline{\text{STBY}}$ Input: Low input at the $\overline{\text{STBY}}$ pin exits from sleep mode to hardware standby mode.

18.4 Software Standby Mode

18.4.1 Transition to Software Standby Mode

To enter software standby mode, execute the SLEEP instruction while the SSBY bit is set to 1 in SYSCR.

In software standby mode, current dissipation is reduced to an extremely low level because the CPU, clock, and on-chip supporting modules all halt. On-chip supporting modules are reset and halted. As long as the specified voltage is supplied, however, CPU register contents and on-chip RAM data are retained. The settings of the I/O ports also held. When the WDT is used as a watchdog timer ($\text{WT}/\overline{\text{IT}} = 1$), the TME bit must be cleared to 0 before setting SSBY. Also, when setting TME to 1, SSBY should be cleared to 0.

Clear the BRLE bit in BRCCR (inhibiting bus release) before making a transition to software standby mode.

18.4.2 Exit from Software Standby Mode

Software standby mode can be exited by input of an external interrupt at the NMI, $\overline{\text{IRQ}}_0$, $\overline{\text{IRQ}}_1$, or $\overline{\text{IRQ}}_2$ pin, or by input at the RES or STBY pin.

Exit by Interrupt: When an NMI, IRQ_0 , IRQ_1 , or IRQ_2 interrupt request signal is received, the clock oscillator begins operating. After the oscillator settling time selected by bits STS2 to STS0 in SYSCR, stable clock signals are supplied to the entire chip, software standby mode ends, and interrupt exception handling begins. Software standby mode is not exited if the interrupt enable bits of interrupts IRQ_0 , IRQ_1 , and IRQ_2 are cleared to 0, or if these interrupts are masked in the CPU.

Exit by $\overline{\text{RES}}$ Input: When the $\overline{\text{RES}}$ input goes low, the clock oscillator starts and clock pulses are supplied immediately to the entire chip. The $\overline{\text{RES}}$ signal must be held low long enough for the clock oscillator to stabilize. When $\overline{\text{RES}}$ goes high, the CPU starts reset exception handling.

Exit by $\overline{\text{STBY}}$ Input: Low input at the $\overline{\text{STBY}}$ pin causes a transition to hardware standby mode.

18.4.3 Selection of Waiting Time for Exit from Software Standby Mode

Bits STS2 to STS0 in SYSCR and bits DIV1 and DIV0 in DIVCR should be set as follows.

Crystal Resonator: Set STS2 to STS0, DIV1, and DIV0 so that the waiting time (for the clock to stabilize) is at least 7 ms. Table 18.3 indicates the waiting times that are selected by STS2 to STS0, DIV1, and DIV0 settings at various system clock frequencies.

When Using an External Clock: Set the STS2 to STS0, DIV0, and DIV1 bits so that the waiting time is at least 100 μs .

Table 18.3 Clock Frequency and Waiting Time for Clock to Settle

| DIV1 | DIV0 | STS2 | STS1 | STS0 | Waiting Time | 25 MHz | 20 MHz | 18 MHz | 16 MHz | 12 MHz | 10 MHz | 8 MHz | 6 MHz | 4 MHz | 2 MHz | 1MHz | Unit | |
|------|------|------|------|------|---------------|--------|-------------|--------|--------|-----------------|--------|-------|-------|-------|--------|--------|-------|-------|
| 0 | 0 | 0 | 0 | 0 | 8192 states | 0.3 | 0.4 | 0.46 | 0.51 | 0.65 | 0.8 | 1.0 | 1.3 | 2.0 | 4.1 | 8.2* | ms | |
| | | 0 | 0 | 1 | 16384 states | 0.7 | 0.8 | 0.91 | 1.0 | 1.3 | 1.6 | 2.0 | 2.7 | 4.1 | 8.2* | 16.4 | | |
| | | 0 | 1 | 0 | 32768 states | 1.3 | 1.6 | 1.8 | 2.0 | 2.7 | 3.3 | 4.1 | 5.5 | 8.2* | 16.4 | 32.8 | | |
| | | 0 | 1 | 1 | 65536 states | 2.6 | 3.3 | 3.6 | 4.1 | 5.5 | 6.6 | 8.2* | 10.9* | 16.4 | 32.8 | 65.5 | | |
| | | 1 | 0 | 0 | 131072 states | 5.2 | 6.6 | 7.3* | 8.2* | 10.9* | 13.1* | 16.4 | 21.8 | 32.8 | 65.5 | 131.1 | | |
| | | 1 | 0 | 1 | 262144 states | 10.5* | 13.1* | 14.6 | 16.4 | 21.8 | 26.2 | 32.8 | 43.7 | 65.5 | 131.1 | 262.1 | | |
| | | 1 | 1 | 0 | 1024 states | 0.04 | 0.05 | 0.057 | 0.064 | 0.085 | 0.10 | 0.13 | 0.17 | 0.26 | 0.51 | 1.0 | | |
| | | 1 | 1 | 1 | | | | | | Illegal setting | | | | | | | | |
| | | 0 | 1 | 0 | 0 | 0 | 8192 states | 0.7 | 0.8 | 0.91 | 1.02 | 1.4 | 1.6 | 2.0 | 2.7 | 4.1 | | 8.2* |
| 0 | 0 | | | 1 | 16384 states | 1.3 | 1.6 | 1.8 | 2.0 | 2.7 | 3.3 | 4.1 | 5.5 | 8.2* | 16.4 | 32.8 | | |
| 0 | 1 | | | 0 | 32768 states | 2.6 | 3.3 | 3.6 | 4.1 | 5.5 | 6.6 | 8.2* | 10.9* | 16.4 | 32.8 | 65.5 | | |
| 0 | 1 | | | 1 | 65536 states | 5.2 | 6.6 | 7.3* | 8.2* | 10.9* | 13.1* | 16.4 | 21.8 | 32.8 | 65.5 | 131.1 | | |
| 1 | 0 | | | 0 | 131072 states | 10.5* | 13.1* | 14.6 | 16.4 | 21.8 | 26.2 | 32.8 | 43.7 | 65.5 | 131.1 | 262.1 | | |
| 1 | 0 | | | 1 | 262144 states | 21.0 | 26.2 | 29.1 | 32.8 | 43.7 | 52.4 | 65.5 | 87.4 | 131.1 | 262.1 | 524.3 | | |
| 1 | 1 | | | 0 | 1024 states | 0.08 | 0.10 | 0.11 | 0.13 | 0.17 | 0.20 | 0.26 | 0.34 | 0.51 | 1.0 | 2.0 | | |
| 1 | 1 | | | 1 | | | | | | Illegal setting | | | | | | | | |
| 1 | 0 | | | 0 | 0 | 0 | 8192 states | 1.3 | 1.6 | 1.8 | 2.0 | 2.7 | 3.3 | 4.1 | 5.5 | 8.2* | 16.4* | 32.8* |
| | | 0 | 0 | 1 | 16384 states | 2.6 | 3.3 | 3.6 | 4.1 | 5.5 | 6.6 | 8.2* | 10.9* | 16.4 | 32.8 | 65.5 | | |
| | | 0 | 1 | 0 | 32768 states | 5.2 | 6.6 | 7.3* | 8.2* | 10.9* | 13.1* | 16.4 | 21.8 | 32.8 | 65.5 | 131.1 | | |
| | | 0 | 1 | 1 | 65536 states | 10.5* | 13.1* | 14.6 | 16.4 | 21.8 | 26.2 | 32.8 | 43.7 | 65.5 | 131.1 | 262.1 | | |
| | | 1 | 0 | 0 | 131072 states | 21.0 | 26.2 | 29.1 | 32.8 | 43.7 | 52.4 | 65.5 | 87.4 | 131.1 | 262.1 | 524.3 | | |
| | | 1 | 0 | 1 | 262144 states | 41.9 | 52.4 | 58.3 | 65.5 | 87.4 | 104.9 | 131.1 | 174.8 | 262.1 | 524.3 | 1048.6 | | |
| | | 1 | 1 | 0 | 1024 states | 0.16 | 0.20 | 0.23 | 0.26 | 0.34 | 0.41 | 0.51 | 0.68 | 1.02 | 2.0 | 4.1 | | |
| | | 1 | 1 | 1 | | | | | | Illegal setting | | | | | | | | |
| | | 1 | 1 | 0 | 0 | 0 | 8192 states | 2.6 | 3.3 | 3.6 | 4.1 | 5.5 | 6.6 | 8.2* | 10.9* | 16.4* | 32.8* | 65.5 |
| 0 | 0 | | | 1 | 16384 states | 5.2 | 6.6 | 7.3* | 8.2* | 10.9* | 13.1* | 16.4 | 21.8 | 32.8 | 65.5 | 131.1 | | |
| 0 | 1 | | | 0 | 32768 states | 10.5* | 13.1 | 14.6 | 16.4 | 21.8 | 26.2 | 32.8 | 43.7 | 65.5 | 131.1 | 262.1 | | |
| 0 | 1 | | | 1 | 65536 states | 21.0 | 26.2 | 29.1 | 32.8 | 43.7 | 52.4 | 65.5 | 87.4 | 131.1 | 262.1 | 524.3 | | |
| 1 | 0 | | | 0 | 131072 states | 41.9 | 52.4 | 58.3 | 65.5 | 87.4 | 104.9 | 131.1 | 174.8 | 262.1 | 524.3 | 1048.6 | | |
| 1 | 0 | | | 1 | 262144 states | 83.9 | 104.9 | 116.5 | 131.1 | 174.8 | 209.7 | 262.1 | 349.5 | 524.3 | 1048.6 | 2097.1 | | |
| 1 | 1 | | | 0 | 1024 states | 0.33 | 0.41 | 0.46 | 0.51 | 0.68 | 0.82 | 1.0 | 1.4 | 2.0 | 4.1 | 8.2* | | |
| 1 | 1 | | | 1 | | | | | | Illegal setting | | | | | | | | |

Note: * Recommended setting

18.4.4 Sample Application of Software Standby Mode

Figure 18.1 shows an example in which software standby mode is entered at the fall of NMI and exited at the rise of NMI.

With the NMI edge select bit (NMIEG) cleared to 0 in SYSCR (selecting the falling edge), an NMI interrupt occurs. Next the NMIEG bit is set to 1 (selecting the rising edge) and the SSBY bit is set to 1; then the SLEEP instruction is executed to enter software standby mode.

Software standby mode is exited at the next rising edge of the NMI signal.

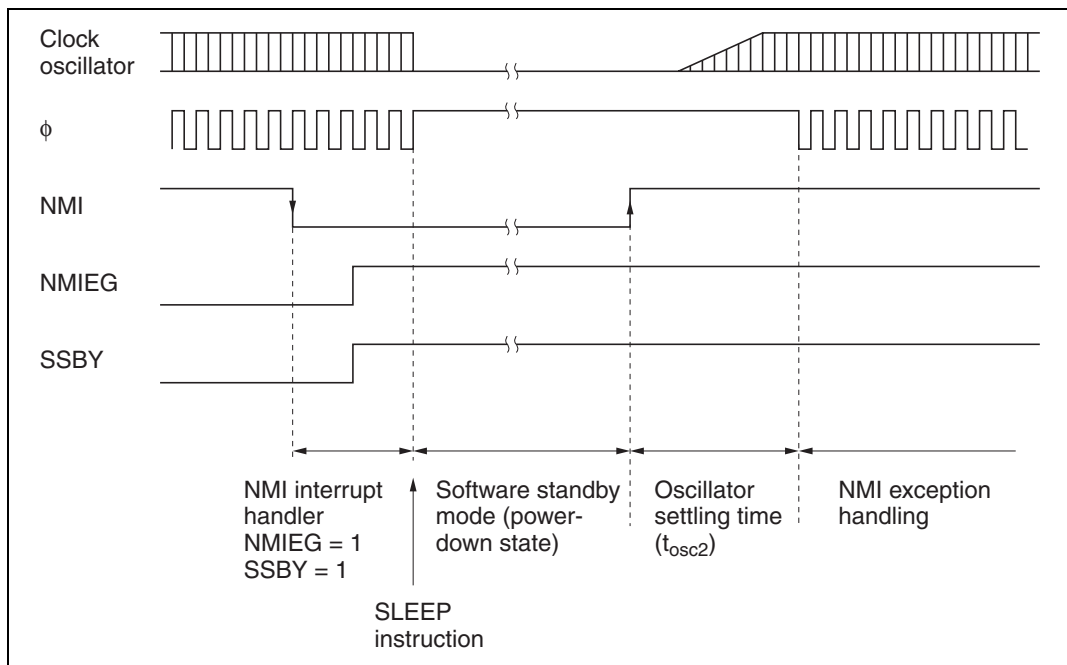


Figure 18.1 NMI Timing for Software Standby Mode (Example)

18.4.5 Note

The I/O ports retain their existing states in software standby mode. If a port is in the high output state, its output current is not reduced.

18.5 Hardware Standby Mode

18.5.1 Transition to Hardware Standby Mode

Regardless of its current state, the chip enters hardware standby mode whenever the $\overline{\text{STBY}}$ pin goes low. Hardware standby mode reduces power consumption drastically by halting all functions of the CPU, and on-chip supporting modules. All modules are reset except the on-chip RAM. As long as the specified voltage is supplied, on-chip RAM data is retained. I/O ports are placed in the high-impedance state.

Clear the RAME bit to 0 in SYSCR before $\overline{\text{STBY}}$ goes low to retain on-chip RAM data.

The inputs at the mode pins (MD2 to MD0) should not be changed during hardware standby mode.

18.5.2 Exit from Hardware Standby Mode

Hardware standby mode is exited by inputs at the $\overline{\text{STBY}}$ and $\overline{\text{RES}}$ pins. While $\overline{\text{RES}}$ is low, when $\overline{\text{STBY}}$ goes high, the clock oscillator starts running. $\overline{\text{RES}}$ should be held low long enough for the clock oscillator to settle. When $\overline{\text{RES}}$ goes high, reset exception handling begins, followed by a transition to the program execution state.

18.5.3 Timing for Hardware Standby Mode

Figure 18.2 shows the timing relationships for hardware standby mode. To enter hardware standby mode, first drive $\overline{\text{RES}}$ low, then drive $\overline{\text{STBY}}$ low. To exit hardware standby mode, first drive $\overline{\text{STBY}}$ high, wait for the clock to settle, then bring $\overline{\text{RES}}$ from low to high.

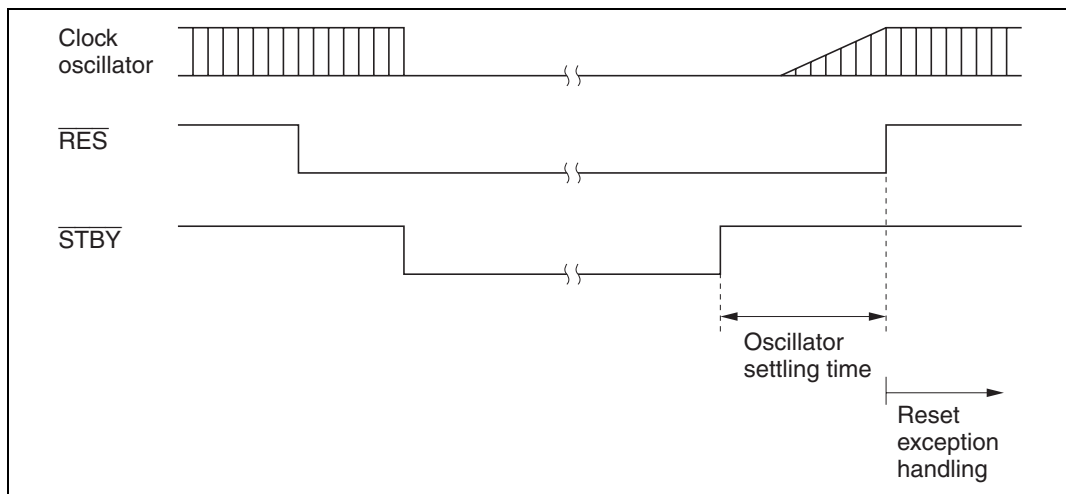


Figure 18.2 Hardware Standby Mode Timing

18.6 Module Standby Function

18.6.1 Module Standby Timing

The module standby function can halt several of the on-chip supporting modules (SCI1, SCI0, 16-bit timer, 8-bit timer, and A/D converter) independently in the power-down state. This standby function is controlled by bits MSTPH2 to MSTPH0 in MSTCRH and bits MSTPL7 to MSTPL0 in MSTCRL. When one of these bits is set to 1, the corresponding on-chip supporting module is placed in standby and halts at the beginning of the next bus cycle after the MSTCR write cycle.

18.6.2 Read/Write in Module Standby

When an on-chip supporting module is in module standby, read/write access to its registers is disabled. Read access always results in H'FF data. Write access is ignored.

18.6.3 Usage Notes

When using the module standby function, note the following points.

On-chip Supporting Module Interrupts: Before setting a module standby bit, first disable interrupts by that module. When an on-chip supporting module is placed in standby by the module standby function, its registers are initialized, including registers with interrupt request flags.

Pin States: Pins used by an on-chip supporting module lose their module functions when the module is placed in module standby. What happens after that depends on the particular pin. For details, see section 7, I/O Ports. Pins that change from the input to the output state require special care. For example, if SCI1 is placed in module standby, the receive data pin loses its receive data

function and becomes a port pin. If its port DDR bit is set to 1, the pin becomes a data output pin, and its output may collide with external SCI transmit data. Data collision should be prevented by clearing the port DDR bit to 0 or taking other appropriate action.

Register Resetting: When an on-chip supporting module is halted by the module standby function, all its registers are initialized. To restart the module, after its MSTCR bit is cleared to 0, its registers must be set up again. It is not possible to write to the registers while the MSTCR bit is set to 1.

18.7 System Clock Output Disabling Function

Output of the system clock (ϕ) can be controlled by the PSTOP bit in MSTCRH. When the PSTOP bit is set to 1, output of the system clock halts and the ϕ pin is placed in the high-impedance state. Figure 18.3 shows the timing of the stopping and starting of system clock output. When the PSTOP bit is cleared to 0, output of the system clock is enabled. Table 18.4 indicates the state of the ϕ pin in various operating states.

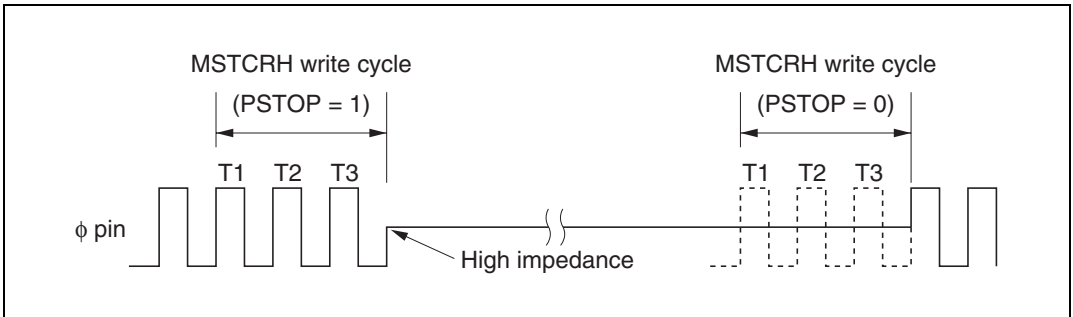


Figure 18.3 Starting and Stopping of System Clock Output

Table 18.4 ϕ Pin State in Various Operating States

| Operating State | PSTOP = 0 | PSTOP = 1 |
|------------------|---------------------|----------------|
| Hardware standby | High impedance | High impedance |
| Software standby | Always high | High impedance |
| Sleep mode | System clock output | High impedance |
| Normal operation | System clock output | High impedance |

Section 19 Electrical Characteristics

19.1 Absolute Maximum Ratings

Table 19.1 lists the absolute maximum ratings.

Table 19.1 Absolute Maximum Ratings

| Item | Symbol | Value | Unit |
|-----------------------------------|-----------|---|------|
| Power supply voltage | V_{CC} | 5 V version: -0.3 to $+7.0$ | V |
| | | 3 V version: -0.3 to $+4.6$ | V |
| Input voltage (except for port 7) | V_{in} | -0.3 to $V_{CC} + 0.3$ | V |
| Input voltage (port 7) | V_{in} | -0.3 to $AV_{CC} + 0.3$ | V |
| Reference voltage | V_{REF} | -0.3 to $AV_{CC} + 0.3$ | V |
| Analog power supply voltage | AV_{CC} | 5 V version: -0.3 to $+7.0$ | V |
| | | 3 V version: -0.3 to $+4.6$ | V |
| Analog input voltage | V_{AN} | -0.3 to $AV_{CC} + 0.3$ | V |
| Operating temperature | T_{opr} | Regular specifications: -20 to $+75$ | °C |
| | | Wide-range specifications: -40 to $+85$ | °C |
| Storage temperature | T_{stg} | -55 to $+125$ | °C |

Caution: Permanent damage to the chip may result if absolute maximum ratings are exceeded.

19.2 DC Characteristics

Table 19.2 lists the DC characteristics. Table 19.3 lists the permissible output currents.

Table 19.2 DC Characteristics (1)

Conditions: $V_{CC} = 5.0\text{ V} \pm 10\%$, $AV_{CC} = 5.0\text{ V} \pm 10\%$, $V_{REF} = 4.5\text{ V}$ to AV_{CC}^{*1} ,
 $V_{SS} = AV_{SS} = 0\text{ V}^{*1}$, $T_a = -20^\circ\text{C}$ to $+75^\circ\text{C}$ (regular specifications),
 $T_a = -40^\circ\text{C}$ to $+85^\circ\text{C}$ (wide-range specifications)

| Item | Symbol | Min | Typ | Max | Unit | Test Conditions | |
|--------------------------------|---|-----------------|---------------------|-----|---------------------|-----------------|---|
| Schmitt trigger input voltages | Port A, P8 ₀ to P8 ₂ | V_T^- | 1.0 | — | — | V | |
| | | V_T^+ | — | — | $V_{CC} \times 0.7$ | V | |
| | | $V_T^+ - V_T^-$ | 0.4 | — | — | V | |
| Input high voltage | RES, STBY, NMI, MD ₂ to MD ₀ | V_{IH} | $V_{CC} - 0.7$ | — | $V_{CC} + 0.3$ | V | |
| | EXTAL | | $V_{CC} \times 0.7$ | — | $V_{CC} + 0.3$ | V | |
| | Port 7 | | 2.0 | — | $AV_{CC} + 0.3$ | V | |
| | Ports 4 to 6, P8 ₃ , P8 ₄ , P9 ₀ to P9 ₅ , port B | | 2.0 | — | $V_{CC} + 0.3$ | V | |
| Input low voltage | RES, STBY, MD ₂ to MD ₀ | V_{IL} | -0.3 | — | 0.5 | V | |
| | NMI, EXTAL, ports 4 to 7, P8 ₃ , P8 ₄ , P9 ₀ to P9 ₅ , port B | | -0.3 | — | 0.8 | V | |
| Output high voltage | All output pins (except RESO) | V_{OH} | $V_{CC} - 0.5$ | — | — | V | $I_{OH} = -200\ \mu\text{A}$ |
| | | | 3.5 | — | — | V | $I_{OH} = -1\ \text{mA}$ |
| Output low voltage | All output pins (except RESO) | V_{OL} | — | — | 0.4 | V | $I_{OL} = 1.6\ \text{mA}$ |
| | A ₀ to A ₁₉ | | — | — | 1.0 | V | $I_{OL} = 10\ \text{mA}$ |
| | RESO | | — | — | 0.4 | V | $I_{OL} = 1.6\ \text{mA}$ |
| Input leakage current | STBY, NMI, RES, MD ₂ to MD ₀ | $ I_{in} $ | — | — | 1.0 | μA | $V_{in} = 0.5\ \text{V}$ to $V_{CC} - 0.5\ \text{V}$ |
| | Port 7 | | — | — | 1.0 | μA | $V_{in} = 0.5\ \text{V}$ to $AV_{CC} - 0.5\ \text{V}$ |

| Item | | Symbol | Min | Typ | Max | Unit | Test Conditions | | |
|-----------------------------------|--|--------------------------------|-------------------------------|------------------|------|------|--|----|--|
| Three-state leakage current | Ports 4 to 6, A ₀ to A ₁₉ , Ports 8 to B | $ I_{TSI} $ | — | — | 1.0 | μA | V _{in} = 0.5 V to V _{CC} - 0.5 V | | |
| | RES0 | | — | — | 10.0 | μA | V _{in} = 0 V | | |
| Input pull-up MOS current | Ports 4 and 5 | -I _p | 50 | — | 300 | μA | V _{in} = 0 V | | |
| Input capacitance | NMI | C _{in} | — | — | 50 | pF | V _{in} = 0 V f = f _{min} T _a = 25°C | | |
| | All input pins except NMI | | — | — | 15 | pF | | | |
| Current dissipation* ² | Normal operation | I _{CC} * ³ | — | 32 (5.0 V) | 47 | mA | f = 20 MHz | | |
| | | | — | 37 (5.0 V) | 58 | mA | f = 25 MHz | | |
| | Sleep mode | | — | 24 (5.0 V) | 38 | mA | f = 20 MHz | | |
| | | | — | 29 (5.0 V) | 47 | mA | f = 25 MHz | | |
| | Module standby mode | | — | 19 (5.0 V) | 31 | mA | f = 20 MHz | | |
| | | | — | 21 (5.0 V) | 37 | mA | f = 25 MHz | | |
| | Standby mode | | — | 1.0 | 10 | μA | T _a ≤ 50°C | | |
| | | | — | — | 80 | μA | 50°C < T _a | | |
| | Analog power supply current | | During A/D conversion | AI _{CC} | — | 0.6 | 1.5 | mA | |
| | | | During A/D and D/A conversion | | — | 0.6 | 1.5 | mA | |
| Idle | | | — | 0.01 | 5.0 | μA | DASTE = 0 | | |
| Reference current | During A/D conversion | AI _{CC} | — | 0.45 | 0.8 | mA | | | |
| | During A/D and D/A conversion | | — | 2.0 | 3.0 | mA | | | |
| | Idle | | — | 0.01 | 5.0 | μA | DASTE = 0 | | |
| RAM standby voltage | | V _{RAM} | 2.0 | — | — | V | | | |

Notes: 1. Do not open the pin connections of the AV_{CC} , V_{REF} and AV_{SS} pins while the A/D converter is not in use.

Connect the AV_{CC} and V_{REF} pins to the V_{CC} and connect the AV_{SS} pin to the V_{SS} , respectively.

2. Given current consumption values are when all the output pins are made to unloaded state and, furthermore, when the on-chip pull-up MOS is turned off under conditions that $V_{IH\ min} = V_{CC} - 0.5\ V$ and $V_{IL\ max} = 0.5\ V$.

Also, the aforesaid current consumption values are when $V_{IH\ min} = V_{CC} \times 0.9$ and $V_{IL\ max} = 0.3\ V$ under the condition of $V_{RAM} \leq V_{CC} < 4.5\ V$.

3. $I_{CC\ max.}$ (under normal operations) = $3.0\ (mA) + 0.40\ (mA/(MHz \times V)) \times V_{CC} \times f$
 $I_{CC\ max.}$ (when using the sleeve) = $3.0\ (mA) + 0.32\ (mA/(MHz \times V)) \times V_{CC} \times f$
 $I_{CC\ max.}$ (when the sleeve + module are standing by)
= $3.0\ (mA) + 0.25\ (mA/(MHz \times V)) \times V_{CC} \times f$

Also, the typ. values for current dissipation are reference values.

Table 19.2 DC Characteristics (2)

Conditions: $V_{CC} = 3.0$ to 3.6 V, $AV_{CC} = 3.0$ to 3.6 V, $V_{REF} = 3.0$ V to AV_{CC}^{*1} ,
 $V_{SS} = AV_{SS} = 0$ V^{*1}, $T_a = -20^{\circ}\text{C}$ to $+75^{\circ}\text{C}$ (regular specifications),
 $T_a = -40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$ (wide-range specifications)

| Item | | Symbol | Min | Typ | Max | Unit | Test Conditions | |
|--------------------------------|--|-----------------|-----------------------------------|-----|---------------------|---------------|--|---------------------------|
| Schmitt trigger input voltages | Port A, P8 ₀ to P8 ₂ | V_T^- | $V_{CC} \times 0.2$ | — | — | V | | |
| | | V_T^+ | — | — | $V_{CC} \times 0.7$ | V | | |
| | | $V_T^+ - V_T^-$ | $V_{CC} \times 0.05$ | — | — | V | | |
| Input high voltage | RES, STBY, NMI, MD ₂ to MD ₀ | V_{IH} | $V_{CC} \times 0.9$ | — | $V_{CC} + 0.3$ | V | | |
| | EXTAL | | $V_{CC} \times 0.7$ | — | $V_{CC} + 0.3$ | V | | |
| | Port 7 | | $V_{CC} \times 0.7$ | — | $AV_{CC} + 0.3$ | V | | |
| | Ports 4 to 6 P8 ₃ , P8 ₄ , P9 ₀ to P9 ₅ , port B | | $V_{CC} \times 0.7$ | — | $V_{CC} + 0.3$ | V | | |
| Input low voltage | RES, STBY, MD ₂ to MD ₀ | V_{IL} | -0.3 | — | $V_{CC} \times 0.1$ | V | | |
| | NMI, EXTAL, ports 4 to 7 P8 ₃ , P8 ₄ , P9 ₀ to P9 ₅ , port B | | -0.3 | — | $V_{CC} \times 0.2$ | V | | |
| Output high voltage | All output pins (except RESO) | V_{OH} | $V_{CC} - 0.5$ | — | — | V | $I_{OH} = -200 \mu\text{A}$ | |
| | | | $V_{CC} - 1.0$ | — | — | V | $I_{OH} = -1 \text{ mA}$ | |
| Output low voltage | All output pins (except RESO) | V_{OL} | — | — | 0.4 | V | $I_{OL} = 1.6 \text{ mA}$ | |
| | | | A ₀ to A ₁₉ | — | — | 1.0 | V | $I_{OL} = 5 \text{ mA}$ |
| | | | RESO | — | — | 0.4 | V | $I_{OL} = 1.6 \text{ mA}$ |
| Input leakage current | STBY, NMI, RES, MD ₂ to MD ₀ | $ I_{in} $ | — | — | 1.0 | μA | $V_{in} = 0.5 \text{ V to } V_{CC} - 0.5 \text{ V}$ | |
| | Port 7 | | — | — | 1.0 | μA | $V_{in} = 0.5 \text{ V to } AV_{CC} - 0.5 \text{ V}$ | |

19. Electrical Characteristics

| Item | | Symbol | Min | Typ | Max | Unit | Test Conditions | |
|-----------------------------|--|--------------------|-----|---------------|------|------|--|-----------------------|
| Three-state leakage current | Ports 4 to 6, A ₀ to A ₁₉ , Ports 8 to B | $ I_{TSI} $ | — | — | 1.0 | μA | V _{in} = 0.5 V to V _{CC} - 0.5 V | |
| | \overline{RESO} | | — | — | 10.0 | μA | V _{in} = 0 V | |
| Input pull-up MOS current | Ports 4 and 5 | -I _p | 10 | — | 300 | μA | V _{in} = 0 V | |
| Input capacitance | NMI | C _{in} | — | — | 50 | pF | V _{in} = 0 V | |
| | All input pins except NMI | | — | — | 15 | pF | f = f _{min} T _a = 25°C | |
| Current dissipation*2 | Normal operation | I _{CC} *3 | — | 37 (3.3 V) | 58 | mA | f = 25 MHz | |
| | Sleep mode | | — | 29 (3.3 V) | 47 | mA | f = 25 MHz | |
| | Module standby mode | | — | 21 (3.3 V) | 37 | mA | f = 25 MHz | |
| | Standby mode | | | — | 1.0 | 10 | μA | T _a ≤ 50°C |
| | | | | — | — | 80 | μA | 50°C < T _a |
| Analog power supply current | During A/D conversion | AI _{CC} | — | 0.6 | 1.5 | mA | AV _{CC} = 3.0 V | |
| | During A/D and D/A conversion | | — | 0.6 | 1.5 | mA | AV _{CC} = 3.0 V | |
| | Idle | | — | 0.01 | 5.0 | μA | DASTE = 0 | |
| Reference current | During A/D conversion | AI _{CC} | — | 0.45 | 0.8 | mA | V _{REF} = 3.0 V | |
| | During A/D and D/A conversion | | — | 2.0 | 3.0 | mA | V _{REF} = 3.0 V | |
| | Idle | | — | 0.01 | 5.0 | μA | DASTE = 0 | |
| RAM standby voltage | | V _{RAM} | 2.0 | — | — | V | | |

Notes: 1. Do not open the pin connections of the AV_{CC}, V_{REF} and AV_{SS} pins while the A/D converter is not in use.

Connect the AV_{CC} and V_{REF} pins to the V_{CC} and connect the AV_{SS} pin to the V_{SS}, respectively.

2. Given current consumption values are when all the output pins are made to unloaded state and, furthermore, when the on-chip pull-up MOS is turned off under conditions that V_{IH} min = V_{CC} - 0.5 V and V_{IL} max = 0.5 V.

Also, the aforesaid current consumption values are when $V_{IH\ min} = V_{CC} \times 0.9$ and $V_{IL\ max} = 0.3\ V$ under the condition of $V_{RAM} \leq V_{CC} < 3.0\ V$.

3. $I_{CC\ max.}$ (under normal operations) = $3.0\ (mA) + 0.61\ (mA/(MHz \times V)) \times V_{CC} \times f$
- $I_{CC\ max.}$ (when using the sleeve) = $3.0\ (mA) + 0.49\ (mA/(MHz \times V)) \times V_{CC} \times f$
- $I_{CC\ max.}$ (when the sleeve + module are standing by) = $3.0\ (mA) + 0.38\ (mA/(MHz \times V)) \times V_{CC} \times f$

Also, the typ. values for current dissipation are reference values.

Table 19.3 Permissible Output Currents

Condition: $T_a = -20^\circ C$ to $+75^\circ C$ (regular specifications), $T_a = -40^\circ C$ to $+85^\circ C$ (wide-range specifications)

Condition A: $V_{CC} = 3.0$ to $3.6\ V$, $AV_{CC} = 3.0$ to $3.6\ V$, $V_{REF} = 3.0$ to AV_{CC} , $V_{SS} = AV_{SS} = 0\ V$

Condition B: $V_{CC} = 4.5$ to $5.5\ V$, $AV_{CC} = 4.5$ to $5.5\ V$, $V_{REF} = 4.5$ to AV_{CC} , $V_{SS} = AV_{SS} = 0\ V$

| Item | Symbol | Condition | | | Unit |
|---|---|-----------|-----|-----|------|
| | | Min | Typ | Max | |
| Permissible output low current (per pin) | A_{19} to A_0 | — | — | 10 | mA |
| | Other output pins | — | — | 2.0 | mA |
| Permissible output low current (total) | Total of 20 pins in A_{19} to A_0 | — | — | 80 | mA |
| | Total of all output pins, including the above | — | — | 120 | mA |
| Permissible output high current (per pin) | All output pins | — | — | 2.0 | mA |
| Permissible output high current (total) | Total of all output pins | — | — | 40 | mA |

Notes: 1. To protect chip reliability, do not exceed the output current values in table 19.3.

2. When directly driving a darlington pair or LED, always insert a current-limiting resistor in the output line, as shown in figure 19.1.

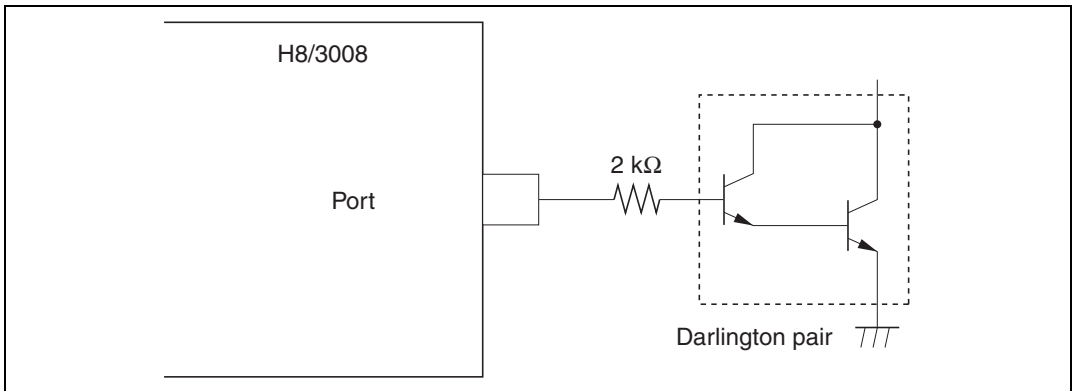


Figure 19.1 Darlington Pair Drive Circuit (Example)

19.3 AC Characteristics

Clock timing parameters are listed in table 19.4, control signal timing parameters in table 19.5, and bus timing parameters in table 19.6. Timing parameters of the on-chip supporting modules are listed in table 19.7.

Table 19.4 Clock Timing

Condition: $T_a = -20^{\circ}\text{C}$ to $+75^{\circ}\text{C}$ (regular specifications), $T_a = -40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$ (wide-range specifications)

Condition A: $V_{CC} = 3.0$ to 3.6 V, $AV_{CC} = 3.0$ to 3.6 V, $V_{REF} = 3.0$ to AV_{CC} , $V_{SS} = AV_{SS} = 0$ V, $f_{max} = 25$ MHz

Condition B: $V_{CC} = 5.0$ V $\pm 10\%$, $AV_{CC} = 5.0$ V $\pm 10\%$, $V_{REF} = 4.5$ to AV_{CC} , $V_{SS} = AV_{SS} = 0$ V, $f_{max} = 20$ MHz

Condition C: $V_{CC} = 5.0$ V $\pm 10\%$, $AV_{CC} = 5.0$ V $\pm 10\%$, $V_{REF} = 4.5$ to AV_{CC} , $V_{SS} = AV_{SS} = 0$ V, $f_{max} = 25$ MHz

| Item | Symbol | Condition | | | | | | Unit | Test Conditions |
|--|------------|-----------|-----|-----|-----|-----|-----|------|-----------------------------|
| | | A | | B | | C | | | |
| | | Min | Max | Min | Max | Min | Max | | |
| Clock cycle time | t_{cyc} | 40 | 500 | 50 | 500 | 40 | 500 | ns | Figure 19.3 to figure 19.15 |
| Clock pulse low width | t_{CL} | 10 | — | 15 | — | 10 | — | ns | |
| Clock pulse high width | t_{CH} | 10 | — | 15 | — | 10 | — | ns | |
| Clock rise time | t_{Cr} | — | 10 | — | 10 | — | 10 | ns | |
| Clock fall time | t_{Cf} | — | 10 | — | 10 | — | 10 | ns | |
| Clock oscillator settling time at reset | t_{OSC1} | 20 | — | 20 | — | 20 | — | ms | Figure 19.3 |
| Clock oscillator settling time in software standby | t_{OSC2} | 7 | — | 7 | — | 7 | — | ms | Figure 18.1 |

Table 19.5 Control Signal Timing

Condition: $T_a = -20^{\circ}\text{C}$ to $+75^{\circ}\text{C}$ (regular specifications), $T_a = -40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$ (wide-range specifications)

Condition A: $V_{CC} = 3.0$ to 3.6 V, $AV_{CC} = 3.0$ to 3.6 V, $V_{REF} = 3.0$ to AV_{CC} , $V_{SS} = AV_{SS} = 0$ V, $f_{max} = 25$ MHz

Condition B: $V_{CC} = 5.0$ V $\pm 10\%$, $AV_{CC} = 5.0$ V $\pm 10\%$, $V_{REF} = 4.5$ to AV_{CC} , $V_{SS} = AV_{SS} = 0$ V, $f_{max} = 20$ MHz

Condition C: $V_{CC} = 5.0$ V $\pm 10\%$, $AV_{CC} = 5.0$ V $\pm 10\%$, $V_{REF} = 4.5$ to AV_{CC} , $V_{SS} = AV_{SS} = 0$ V, $f_{max} = 25$ MHz

| Item | Symbol | Condition | | | | Unit | Test Conditions |
|--|--------------------|-----------|-----|---------|-----|------------------|-----------------|
| | | A | | B and C | | | |
| | | Min | Max | Min | Max | | |
| $\overline{\text{RES}}$ setup time | t_{RESS} | 150 | — | 150 | — | ns | Figure 19.4 |
| $\overline{\text{RES}}$ pulse width | t_{RESW} | 10 | — | 10 | — | t_{cyc} | |
| Mode programming setup time | t_{MDS} | 200 | — | 200 | — | ns | |
| $\overline{\text{RESO}}$ output delay time | t_{RESD} | — | 50 | — | 50 | ns | Figure 19.5 |
| $\overline{\text{RESO}}$ output pulse width | t_{RESOW} | 132 | — | 132 | — | t_{cyc} | |
| NMI, $\overline{\text{IRQ}}$ setup time | t_{NMIS} | 150 | — | 150 | — | ns | Figure 19.6 |
| NMI, $\overline{\text{IRQ}}$ hold time | t_{NMIH} | 10 | — | 10 | — | ns | |
| NMI, $\overline{\text{IRQ}}$ pulse width (in recovery from software standby mode) | t_{NMIW} | 200 | — | 200 | — | ns | |

Table 19.6 Bus Timing

Condition: $T_a = -20^{\circ}\text{C}$ to $+75^{\circ}\text{C}$ (regular specifications), $T_a = -40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$ (wide-range specifications)

Condition A: $V_{CC} = 3.0$ to 3.6 V, $AV_{CC} = 3.0$ to 3.6 V, $V_{REF} = 3.0$ to AV_{CC} , $V_{SS} = AV_{SS} = 0$ V, $f_{max} = 25$ MHz

Condition B: $V_{CC} = 5.0$ V $\pm 10\%$, $AV_{CC} = 5.0$ V $\pm 10\%$, $V_{REF} = 4.5$ to AV_{CC} , $V_{SS} = AV_{SS} = 0$ V, $f_{max} = 20$ MHz

Condition C: $V_{CC} = 5.0$ V $\pm 10\%$, $AV_{CC} = 5.0$ V $\pm 10\%$, $V_{REF} = 4.5$ to AV_{CC} , $V_{SS} = AV_{SS} = 0$ V, $f_{max} = 25$ MHz

| Item | Symbol | Condition | | | | Unit | Test Conditions |
|----------------------------|------------|----------------------|-----|----------------------|-----|------|-----------------------------|
| | | A | | B and C | | | |
| | | Min | Max | Min | Max | | |
| Address delay time | t_{AD} | — | 25 | — | 25 | ns | Figure 19.7, figure 19.8 |
| Address hold time | t_{AH} | $0.5 t_{cyc}$ -20 | — | $0.5 t_{cyc}$ -20 | — | ns | |
| Read strobe delay time | t_{RSD} | — | 25 | — | 25 | ns | |
| Address strobe delay time | t_{ASD} | — | 25 | — | 25 | ns | |
| Write strobe delay time | t_{WSD} | — | 25 | — | 25 | ns | |
| Strobe delay time | t_{SD} | — | 25 | — | 25 | ns | |
| Write strobe pulse width 1 | t_{WSW1} | $1.0 t_{cyc}$ -25 | — | $1.0 t_{cyc}$ -25 | — | ns | |
| Write strobe pulse width 2 | t_{WSW2} | $1.5 t_{cyc}$ -25 | — | $1.5 t_{cyc}$ -25 | — | ns | |
| Address setup time 1 | t_{AS1} | $0.5 t_{cyc}$ -20 | — | $0.5 t_{cyc}$ -20 | — | ns | |
| Address setup time 2 | t_{AS2} | $1.0 t_{cyc}$ -20 | — | $1.0 t_{cyc}$ -20 | — | ns | |
| Read data setup time | t_{RDS} | 25 | — | 25 | — | ns | |
| Read data hold time | t_{RDH} | 0 | — | 0 | — | ns | |
| Write data delay time | t_{WDD} | — | 35 | — | 35 | ns | |
| Write data setup time 1 | t_{WDS1} | $1.0 t_{cyc}$ -30 | — | $1.0 t_{cyc}$ -30 | — | ns | |
| Write data setup time 2 | t_{WDS2} | $2.0 t_{cyc}$ -30 | — | $2.0 t_{cyc}$ -30 | — | ns | |

| Item | Symbol | Condition | | | | Unit | Test Conditions |
|------------------------------|-------------|----------------------|----------------------|----------------------|----------------------|------|-----------------------------|
| | | A | | B and C | | | |
| | | Min | Max | Min | Max | | |
| Write data hold time | t_{WDH} | $0.5 t_{cyc}$ -15 | — | $0.5 t_{cyc}$ -15 | — | ns | Figure 19.7, figure 19.8 |
| Read data access time 1 | t_{ACC1} | — | $2.0 t_{cyc}$ -45 | — | $2.0 t_{cyc}$ -45 | ns | |
| Read data access time 2 | t_{ACC2} | — | $3.0 t_{cyc}$ -45 | — | $3.0 t_{cyc}$ -45 | ns | |
| Read data access time 3 | t_{ACC3} | — | $1.5 t_{cyc}$ -45 | — | $1.5 t_{cyc}$ -45 | ns | |
| Read data access time 4 | t_{ACC4} | — | $2.5 t_{cyc}$ -45 | — | $2.5 t_{cyc}$ -45 | ns | |
| Precharge time 1 | t_{PCH1} | $1.0 t_{cyc}$ -20 | — | $1.0 t_{cyc}$ -20 | — | ns | |
| Precharge time 2 | t_{PCH2} | $0.5 t_{cyc}$ -20 | — | $0.5 t_{cyc}$ -20 | — | ns | |
| Wait setup time | t_{WTS} | 25 | — | 25 | — | ns | Figure 19.9 |
| Wait hold time | t_{WTH} | 5 | — | 5 | — | ns | |
| Bus request setup time | t_{BRQS} | 25 | — | 25 | — | ns | Figure 19.10 |
| Bus acknowledge delay time 1 | t_{BACD1} | — | 30 | — | 30 | ns | |
| Bus acknowledge delay time 2 | t_{BACD2} | — | 30 | — | 30 | ns | |
| Bus-floating time | t_{BZD} | — | 30 | — | 30 | ns | |

Note: In order to secure the address hold time relative to the rise of the \overline{RD} strobe, address update mode 2 should be used. For details see section 6.3.5, Address Output Method.

Table 19.7 Timing of On-Chip Supporting Modules

Condition: $T_a = -20^{\circ}\text{C}$ to $+75^{\circ}\text{C}$ (regular specifications), $T_a = -40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$ (wide-range specifications)

Condition A: $V_{CC} = 3.0$ to 3.6 V, $AV_{CC} = 3.0$ to 3.6 V, $V_{REF} = 3.0$ to AV_{CC} , $V_{SS} = AV_{SS} = 0$ V, $f_{max} = 25$ MHz

Condition B: $V_{CC} = 5.0$ V $\pm 10\%$, $AV_{CC} = 5.0$ V $\pm 10\%$, $V_{REF} = 4.5$ to AV_{CC} , $V_{SS} = AV_{SS} = 0$ V, $f_{max} = 20$ MHz

Condition C: $V_{CC} = 5.0$ V $\pm 10\%$, $AV_{CC} = 5.0$ V $\pm 10\%$, $V_{REF} = 4.5$ to AV_{CC} , $V_{SS} = AV_{SS} = 0$ V, $f_{max} = 25$ MHz

| Module Item | | Symbol | Condition | | | | Unit | Test Conditions | |
|---------------|------------------------------|-------------|-------------|-----|---------|-----|------|-----------------|--|
| | | | A | | B and C | | | | |
| | | | Min | Max | Min | Max | | | |
| Ports and TPC | Output data delay time | t_{PVD} | — | 50 | — | 50 | ns | Figure 19.11 | |
| | Input data setup time | t_{PRS} | 50 | — | 50 | — | ns | | |
| | Input data hold time | t_{PRH} | 50 | — | 50 | — | ns | | |
| 16-bit timer | Timer output delay time | t_{TOCD} | — | 50 | — | 50 | ns | Figure 19.12 | |
| | Timer input setup time | t_{TICS} | 50 | — | 50 | — | ns | | |
| | Timer clock input setup time | t_{TCKS} | 50 | — | 50 | — | ns | Figure 19.13 | |
| | Timer clock pulse width | Single edge | t_{TCKWH} | 1.5 | — | 1.5 | — | t_{cyc} | |
| | | Both edges | t_{TCKWL} | 2.5 | — | 2.5 | — | t_{cyc} | |
| 8-bit timer | Timer output delay time | t_{TOCD} | — | 50 | — | 50 | ns | Figure 19.12 | |
| | Timer input setup time | t_{TICS} | 50 | — | 50 | — | ns | | |
| | Timer clock input setup time | t_{TCKS} | 50 | — | 50 | — | ns | Figure 19.13 | |
| | Timer clock pulse width | Single edge | t_{TCKWH} | 1.5 | — | 1.5 | — | t_{cyc} | |
| | | Both edges | t_{TCKWL} | 2.5 | — | 2.5 | — | t_{cyc} | |

| Module | Item | Symbol | Condition | | | | Unit | Test Conditions | |
|--------|---------------------------------------|--------------|---------------|-----|-----|-----|---------------|-----------------|--------------|
| | | | Min | Max | Min | Max | | | |
| SCI | Input clock cycle | Asynchronous | $t_{S_{cyc}}$ | 4 | — | 4 | — | t_{cyc} | Figure 19.14 |
| | | Synchronous | | 6 | — | 6 | — | t_{cyc} | |
| | Input clock rise time | t_{SCKr} | — | 1.5 | — | 1.5 | t_{cyc} | | |
| | Input clock fall time | t_{SCKf} | — | 1.5 | — | 1.5 | t_{cyc} | | |
| | Input clock pulse width | t_{SCKW} | 0.4 | 0.6 | 0.4 | 0.6 | $t_{S_{cyc}}$ | | |
| | Transmit data delay time | t_{TXD} | — | 100 | — | 100 | ns | Figure 19.15 | |
| | Receive data setup time (synchronous) | t_{RXS} | 100 | — | 100 | — | ns | | |
| | Receive data hold time (synchronous) | t_{RXH} | 100 | — | 100 | — | ns | | |
| | | | | 0 | — | 0 | — | ns | |

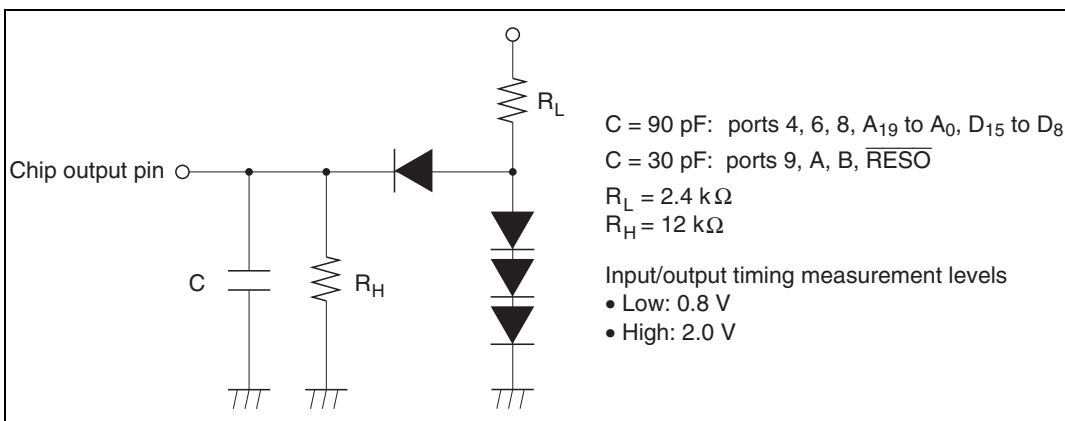


Figure 19.2 Output Load Circuit

19.4 A/D Conversion Characteristics

Table 19.8 lists the A/D conversion characteristics.

Table 19.8 A/D Conversion Characteristics

Condition: $T_a = -20^{\circ}\text{C}$ to $+75^{\circ}\text{C}$ (regular specifications), $T_a = -40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$ (wide-range specifications)

Condition A: $V_{CC} = 3.0$ to 3.6 V, $AV_{CC} = 3.0$ to 3.6 V, $V_{REF} = 3.0$ to AV_{CC} , $V_{SS} = AV_{SS} = 0$ V, $f_{max} = 25$ MHz

Condition B: $V_{CC} = 5.0$ V $\pm 10\%$, $AV_{CC} = 5.0$ V $\pm 10\%$, $V_{REF} = 4.5$ to AV_{CC} , $V_{SS} = AV_{SS} = 0$ V, $f_{max} = 25$ MHz

Condition C: $V_{CC} = 5.0$ V $\pm 10\%$, $AV_{CC} = 5.0$ V $\pm 10\%$, $V_{REF} = 4.5$ to AV_{CC} , $V_{SS} = AV_{SS} = 0$ V, $f_{max} = 25$ MHz

| Item | | Condition | | | | | | Unit | |
|-----------------------------|-------------------------------------|--------------------|-----|-----------|---------|-----|-----------|---------------|------------|
| | | A | | | B and C | | | | |
| | | Min | Typ | Max | Min | Typ | Max | | |
| Conversion time: 134 states | Resolution | 10 | 10 | 10 | 10 | 10 | 10 | bits | |
| | Conversion time (single mode) | 5.36 | — | — | 5.36 | — | — | μs | |
| | Analog input capacitance | — | — | 20 | — | — | 20 | pF | |
| | Permissible signal-source impedance | $\phi \leq 13$ MHz | — | — | 10 | — | — | 10 | k Ω |
| | | $\phi > 13$ MHz | — | — | 5 | — | — | 5 | k Ω |
| | Nonlinearity error | — | — | ± 3.5 | — | — | ± 3.5 | LSB | |
| | Offset error | — | — | ± 3.5 | — | — | ± 3.5 | LSB | |
| | Full-scale error | — | — | ± 3.5 | — | — | ± 3.5 | LSB | |
| | Quantization error | — | — | ± 0.5 | — | — | ± 0.5 | LSB | |
| | Absolute accuracy | — | — | ± 4.0 | — | — | ± 4.0 | LSB | |

| Item | | Condition | | | | | | Unit | |
|-------------------------------|-------------------------------------|--------------------|-----------|-----------|---------|-----------|-----------|---------|------------|
| | | A | | | B and C | | | | |
| | | Min | Typ | Max | Min | Typ | Max | | |
| Conversion time: 70 states | Resolution | 10 | 10 | 10 | 10 | 10 | 10 | bits | |
| | Conversion time (single mode) | 5.36 | — | — | 5.36 | — | — | μ s | |
| | Analog input capacitance | — | — | 20 | — | — | 20 | pF | |
| | Permissible signal-source impedance | $\phi \leq 13$ MHz | — | — | 5 | — | — | 5 | k Ω |
| | | $\phi > 13$ MHz | — | — | 3 | — | — | 3 | k Ω |
| | Nonlinearity error | — | — | ± 7.5 | — | — | ± 7.5 | LSB | |
| | Offset error | — | — | ± 7.5 | — | — | ± 7.5 | LSB | |
| | Full-scale error | — | — | ± 7.5 | — | — | ± 7.5 | LSB | |
| | Quantization error | — | — | ± 0.5 | — | — | ± 0.5 | LSB | |
| Absolute accuracy | — | — | ± 8.0 | — | — | ± 8.0 | LSB | | |

Note: * Do not select 70 states as the conversion time when using an operating frequency that exceeds $f = 70$ (states)/5.36 (μ s) \approx 13.0 (MHz).

19.5 D/A Conversion Characteristics

Table 19.9 lists the D/A conversion characteristics.

Table 19.9 D/A Conversion Characteristics

Condition: $T_a = -20^{\circ}\text{C}$ to $+75^{\circ}\text{C}$ (regular specifications), $T_a = -40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$ (wide-range specifications)

Condition A: $V_{CC} = 3.0$ to 3.6 V, $AV_{CC} = 3.0$ to 3.6 V, $V_{REF} = 3.0$ to AV_{CC} , $V_{SS} = AV_{SS} = 0$ V, $f_{max} = 25$ MHz

Condition B: $V_{CC} = 5.0$ V $\pm 10\%$, $AV_{CC} = 5.0$ V $\pm 10\%$, $V_{REF} = 4.5$ to AV_{CC} , $V_{SS} = AV_{SS} = 0$ V, $f_{max} = 20$ MHz

Condition C: $V_{CC} = 5.0$ V $\pm 10\%$, $AV_{CC} = 5.0$ V $\pm 10\%$, $V_{REF} = 4.5$ to AV_{CC} , $V_{SS} = AV_{SS} = 0$ V, $f_{max} = 25$ MHz

| Item | Condition | | | | | | Unit | Conditions |
|--------------------------------|-----------|-----------|-----------|---------|-----------|-----------|---------------|-----------------------------|
| | A | | | B and C | | | | |
| | Min | Typ | Max | Min | Typ | Max | | |
| Resolution | 8 | 8 | 8 | 8 | 8 | 8 | bits | |
| Conversion time (setting time) | — | — | 10 | — | — | 10 | μs | 20 pF capacitive load |
| Absolute accuracy | — | ± 2.0 | ± 3.0 | — | ± 1.5 | ± 2.0 | LSB | 2 M Ω resistive load |
| | — | — | ± 2.0 | — | — | ± 1.5 | LSB | 4 M Ω resistive load |

19.6 Operational Timing

This section shows timing diagrams.

19.6.1 Clock Timing

Clock timing is shown as follows:

- Oscillator settling timing

Figure 19.3 shows the oscillator settling timing.

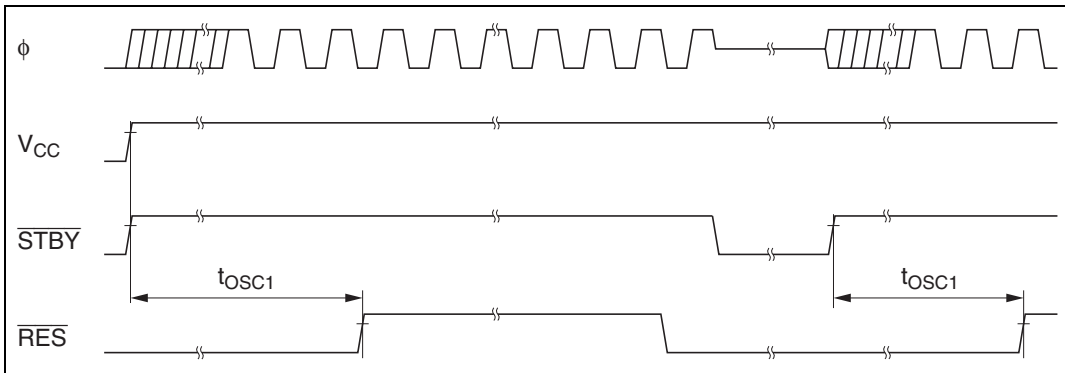


Figure 19.3 Oscillator Settling Timing

19.6.2 Control Signal Timing

Control signal timing is shown as follows:

- Reset input timing
Figure 19.4 shows the reset input timing.
- Reset output timing
Figure 19.5 shows the reset output timing.
- Interrupt input timing
Figure 19.6 shows the interrupt input timing for NMI and \overline{IRQ}_5 to \overline{IRQ}_0 .

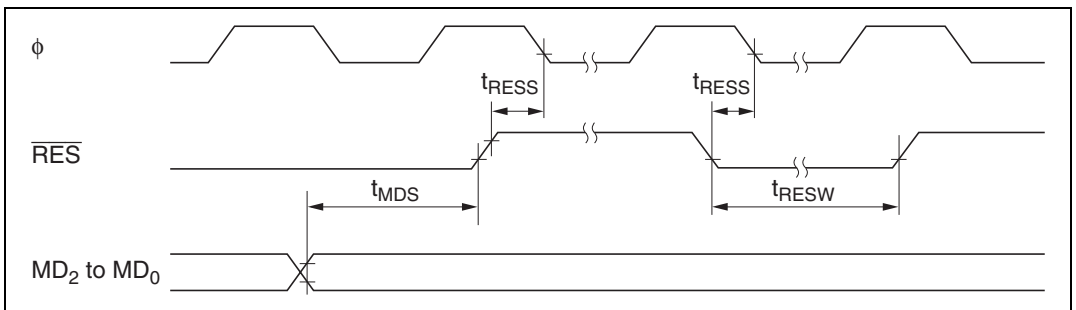


Figure 19.4 Reset Input Timing

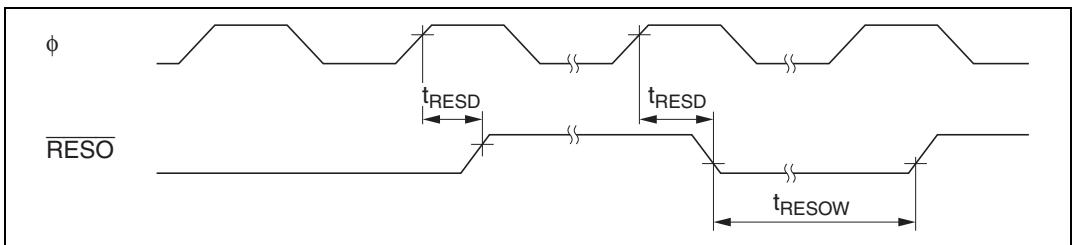


Figure 19.5 Reset Output Timing

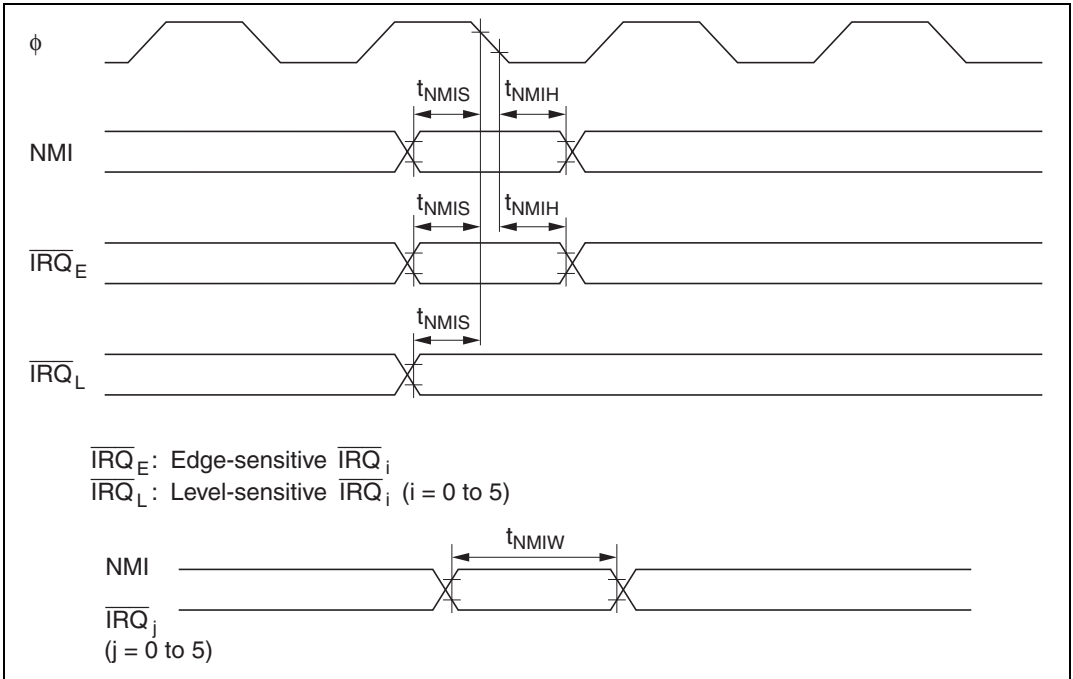


Figure 19.6 Interrupt Input Timing

19.6.3 Bus Timing

Bus timing is shown as follows:

- Basic bus cycle: two-state access
Figure 19.7 shows the timing of the external two-state access cycle.
- Basic bus cycle: three-state access
Figure 19.8 shows the timing of the external three-state access cycle.
- Basic bus cycle: three-state access with one wait state
Figure 19.9 shows the timing of the external three-state access cycle with one wait state inserted.
- Bus-release mode timing
Figure 19.10 shows the bus-release mode timing.

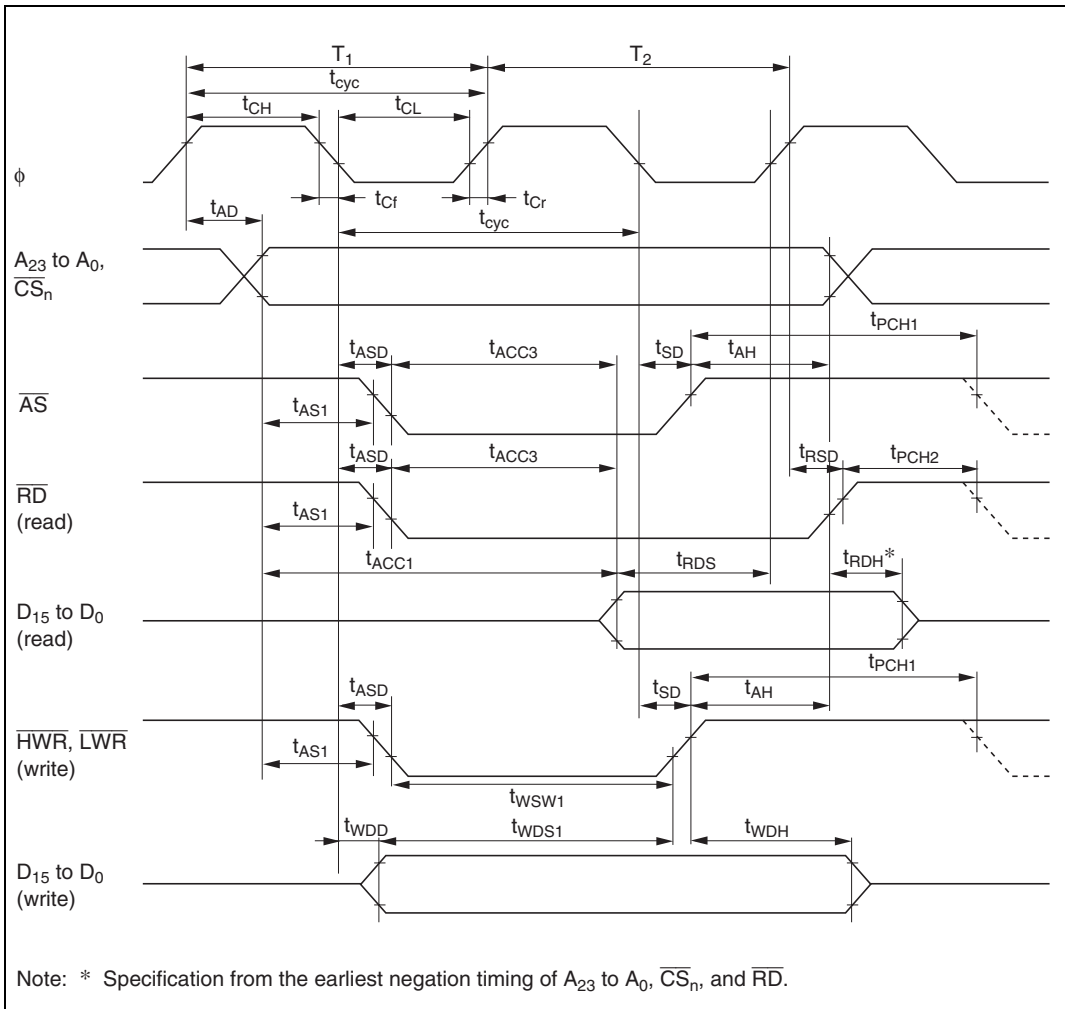


Figure 19.7 Basic Bus Cycle: Two-State Access

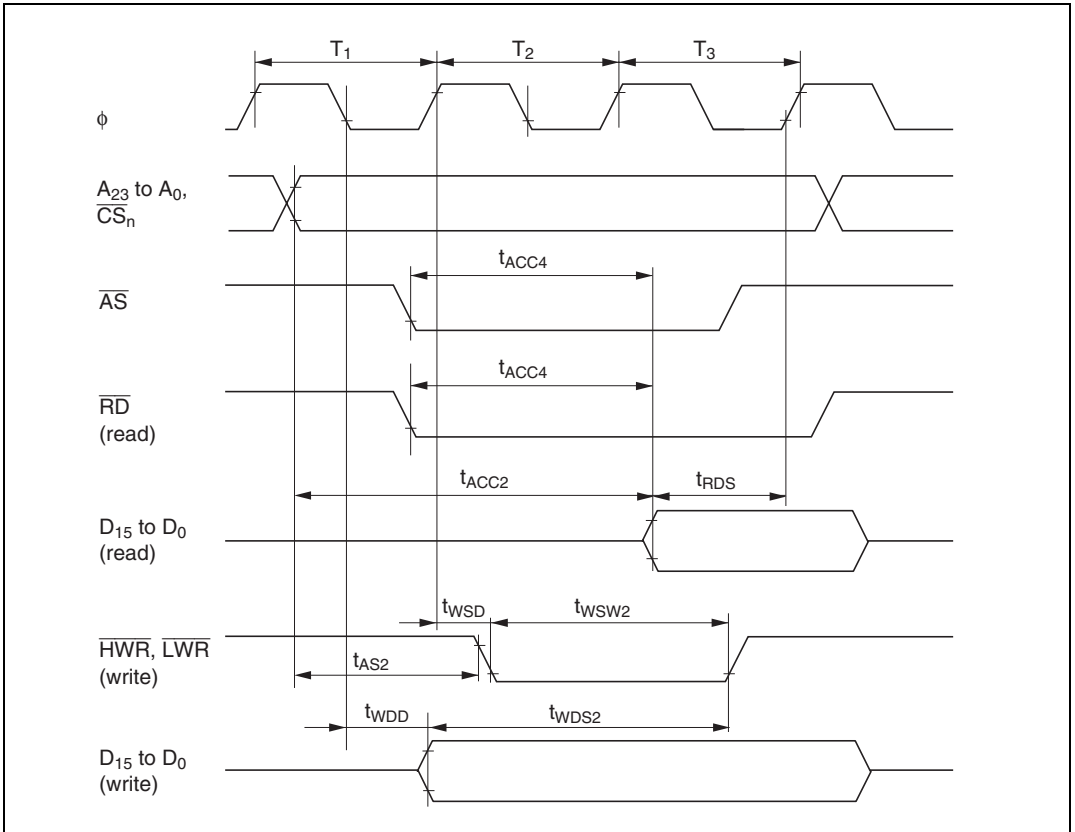


Figure 19.8 Basic Bus Cycle: Three-State Access

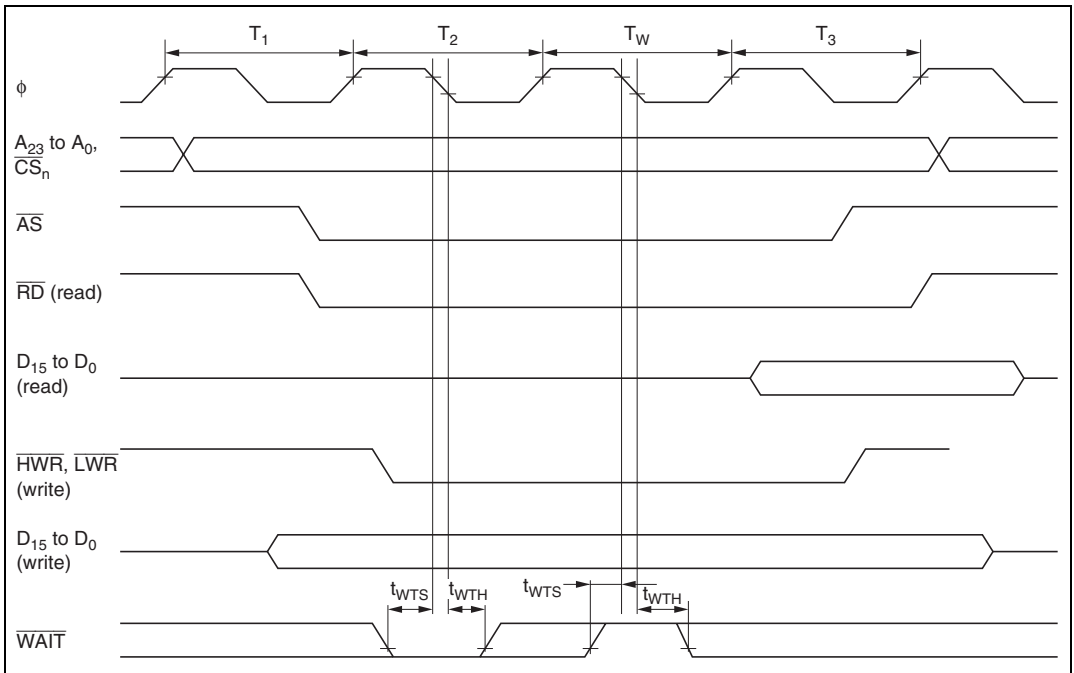


Figure 19.9 Basic Bus Cycle: Three-State Access with One Wait State

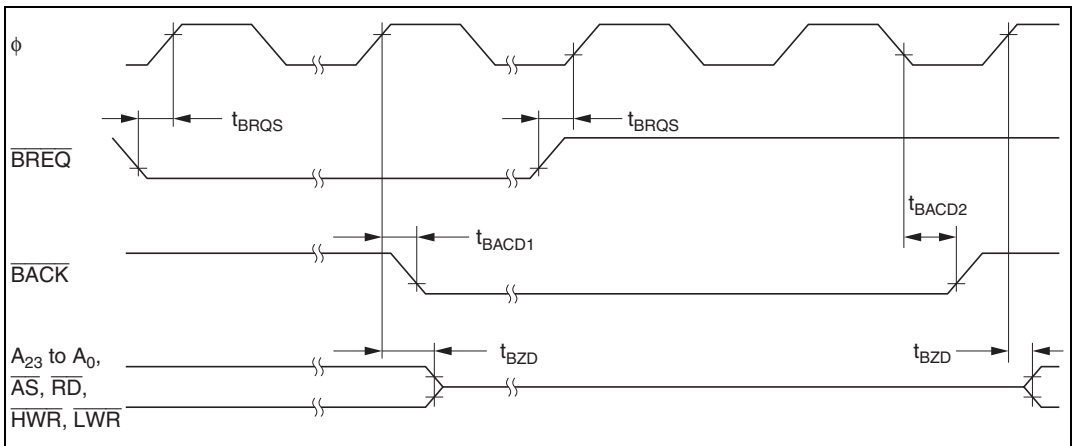


Figure 19.10 Bus-Release Mode Timing

19.6.4 TPC and I/O Port Timing

Figure 19.11 shows the TPC and I/O port input/output timing.

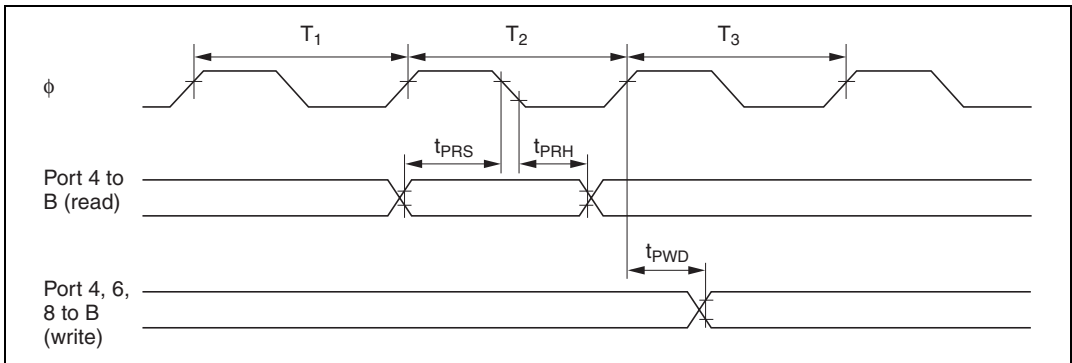


Figure 19.11 TPC and I/O Port Input/Output Timing

19.6.5 Timer Input/Output Timing

16-bit timer and 8-bit timer timing are shown below.

- Timer input/output timing

Figure 19.12 shows the timer input/output timing.

- Timer external clock input timing

Figure 19.13 shows the timer external clock input timing.

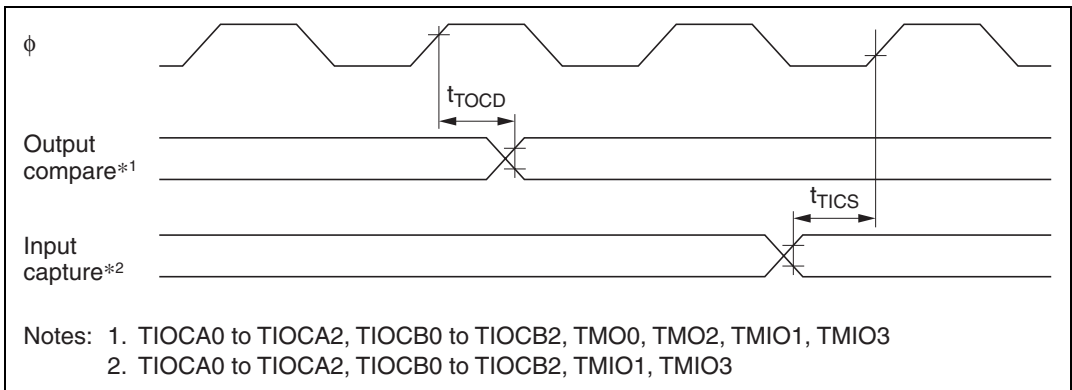


Figure 19.12 Timer Input/Output Timing

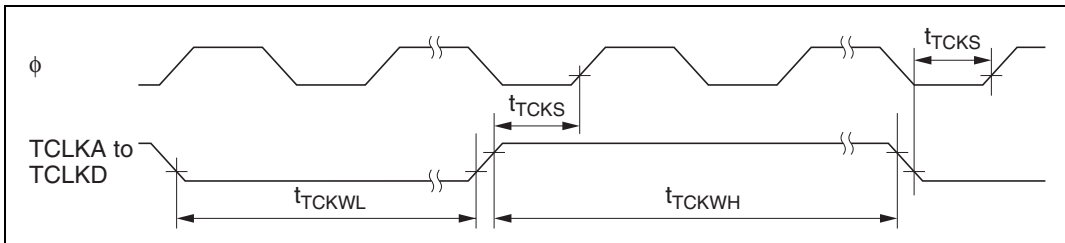


Figure 19.13 Timer External Clock Input Timing

19.6.6 SCI Input/Output Timing

SCI timing is shown as follows:

- SCI input clock timing

Figure 19.14 shows the SCI input clock timing.

- SCI input/output timing (synchronous mode)

Figure 19.15 shows the SCI input/output timing in synchronous mode.

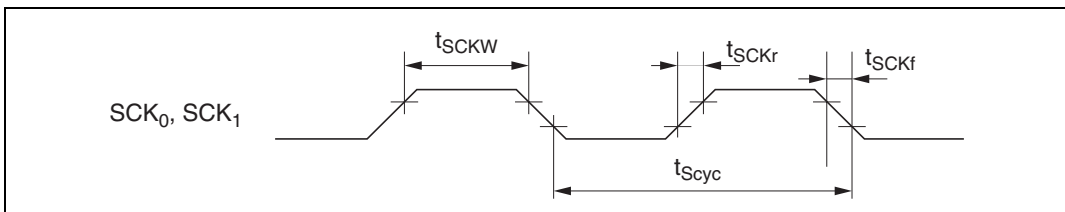


Figure 19.14 SCI Input Clock Timing

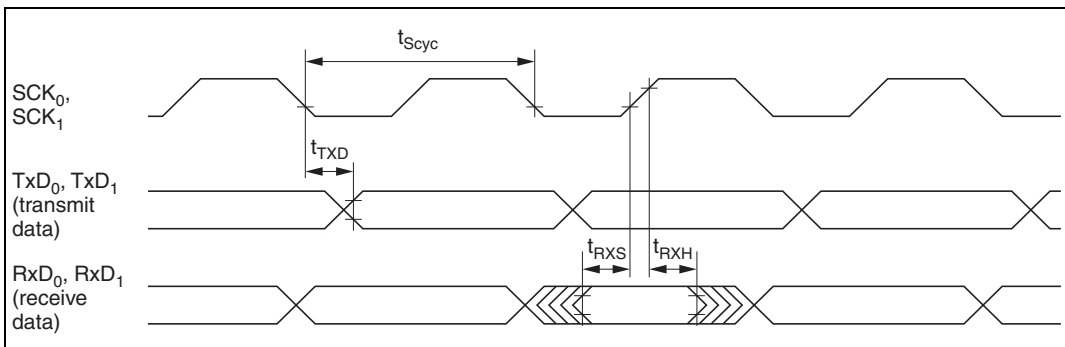


Figure 19.15 SCI Input/Output Timing in Synchronous Mode

Appendix A Instruction Set

A.1 Instruction List

Operand Notation

| Symbol | Description |
|----------|---|
| Rd | General destination register |
| Rs | General source register |
| Rn | General register |
| ERd | General destination register (address register or 32-bit register) |
| ERs | General source register (address register or 32-bit register) |
| ERn | General register (32-bit register) |
| (EAd) | Destination operand |
| (EAs) | Source operand |
| PC | Program counter |
| SP | Stack pointer |
| CCR | Condition code register |
| N | N (negative) flag in CCR |
| Z | Z (zero) flag in CCR |
| V | V (overflow) flag in CCR |
| C | C (carry) flag in CCR |
| disp | Displacement |
| → | Transfer from the operand on the left to the operand on the right, or transition from the state on the left to the state on the right |
| + | Addition of the operands on both sides |
| − | Subtraction of the operand on the right from the operand on the left |
| × | Multiplication of the operands on both sides |
| ÷ | Division of the operand on the left by the operand on the right |
| ^ | Logical AND of the operands on both sides |
| ∨ | Logical OR of the operands on both sides |
| ⊕ | Exclusive logical OR of the operands on both sides |
| ¬ | NOT (logical complement) |
| (), < > | Contents of operand |

Note: General registers include 8-bit registers (R0H to R7H and R0L to R7L) and 16-bit registers (R0 to R7 and E0 to E7).

Condition Code Notation

| Symbol | Description |
|---------------|--|
| ↕ | Changed according to execution result |
| * | Undetermined (no guaranteed value) |
| 0 | Cleared to 0 |
| 1 | Set to 1 |
| — | Not affected by execution of the instruction |
| Δ | Varies depending on conditions, described in notes |

Table A.1 Instruction Set

1. Data transfer instructions

| Mnemonic | Operand Size | Addressing Mode and Instruction Length (bytes) | | | | | | | Operation | Condition Code | | | | | | No. of States ^{*1} | | |
|------------------------|--------------|--|----|------|-----------|-------------|-----|----------|---------------------------------|----------------|---|---|---|---|---|-----------------------------|--------|----------|
| | | #xx | Rn | @ERn | @(d, ERn) | @-ERn/@ERn+ | @aa | @(d, PC) | | @aa | I | H | N | Z | V | C | Normal | Advanced |
| | | | | | | | | | | | | | | | | | | |
| MOV.B #xx:8, Rd | B | 2 | | | | | | | #xx:8 → Rd8 | — | — | ↓ | ↓ | 0 | — | 2 | | |
| MOV.B Rs, Rd | B | | 2 | | | | | | Rs8 → Rd8 | — | — | ↓ | ↓ | 0 | — | 2 | | |
| MOV.B @ERs, Rd | B | | | 2 | | | | | @ERs → Rd8 | — | — | ↓ | ↓ | 0 | — | 4 | | |
| MOV.B @(d:16, ERs), Rd | B | | | | 4 | | | | @(d:16, ERs) → Rd8 | — | — | ↓ | ↓ | 0 | — | 6 | | |
| MOV.B @(d:24, ERs), Rd | B | | | | | 8 | | | @(d:24, ERs) → Rd8 | — | — | ↓ | ↓ | 0 | — | 10 | | |
| MOV.B @ERs+, Rd | B | | | | | | 2 | | @ERs → Rd8 ERs32+1 → ERs32 | — | — | ↓ | ↓ | 0 | — | 6 | | |
| MOV.B @aa:8, Rd | B | | | | | | | 2 | @aa:8 → Rd8 | — | — | ↓ | ↓ | 0 | — | 4 | | |
| MOV.B @aa:16, Rd | B | | | | | | | 4 | @aa:16 → Rd8 | — | — | ↓ | ↓ | 0 | — | 6 | | |
| MOV.B @aa:24, Rd | B | | | | | | | 6 | @aa:24 → Rd8 | — | — | ↓ | ↓ | 0 | — | 8 | | |
| MOV.B Rs, @ERd | B | | | 2 | | | | | Rs8 → @ERd | — | — | ↓ | ↓ | 0 | — | 4 | | |
| MOV.B Rs, @(d:16, ERd) | B | | | | 4 | | | | Rs8 → @(d:16, ERd) | — | — | ↓ | ↓ | 0 | — | 6 | | |
| MOV.B Rs, @(d:24, ERd) | B | | | | | 8 | | | Rs8 → @(d:24, ERd) | — | — | ↓ | ↓ | 0 | — | 10 | | |
| MOV.B Rs, @-ERd | B | | | | | | 2 | | ERd32-1 → ERd32 Rs8 → @ERd | — | — | ↓ | ↓ | 0 | — | 6 | | |
| MOV.B Rs, @aa:8 | B | | | | | | | 2 | Rs8 → @aa:8 | — | — | ↓ | ↓ | 0 | — | 4 | | |
| MOV.B Rs, @aa:16 | B | | | | | | | 4 | Rs8 → @aa:16 | — | — | ↓ | ↓ | 0 | — | 6 | | |
| MOV.B Rs, @aa:24 | B | | | | | | | 6 | Rs8 → @aa:24 | — | — | ↓ | ↓ | 0 | — | 8 | | |
| MOV.W #xx:16, Rd | W | 4 | | | | | | | #xx:16 → Rd16 | — | — | ↓ | ↓ | 0 | — | 4 | | |
| MOV.W Rs, Rd | W | | 2 | | | | | | Rs16 → Rd16 | — | — | ↓ | ↓ | 0 | — | 2 | | |
| MOV.W @ERs, Rd | W | | | 2 | | | | | @ERs → Rd16 | — | — | ↓ | ↓ | 0 | — | 4 | | |
| MOV.W @(d:16, ERs), Rd | W | | | | 4 | | | | @(d:16, ERs) → Rd16 | — | — | ↓ | ↓ | 0 | — | 6 | | |
| MOV.W @(d:24, ERs), Rd | W | | | | | 8 | | | @(d:24, ERs) → Rd16 | — | — | ↓ | ↓ | 0 | — | 10 | | |
| MOV.W @ERs+, Rd | W | | | | | | 2 | | @ERs → Rd16 ERs32+2 → @ERd32 | — | — | ↓ | ↓ | 0 | — | 6 | | |
| MOV.W @aa:16, Rd | W | | | | | | | 4 | @aa:16 → Rd16 | — | — | ↓ | ↓ | 0 | — | 6 | | |

| Mnemonic | Operand Size | Addressing Mode and Instruction Length (bytes) | | | | | | | | Operation | Condition Code | | | | | | No. of States ^{*1} | |
|------------------------|--------------|--|----|------|-----------|-------------|-----|----------|---------------------------------|-----------|----------------|---|---|---|---|----|-----------------------------|----------|
| | | #xx | Rn | @ERn | @(d, ERn) | @-ERn/@ERn+ | @aa | @(d, PC) | @@aa | | I | H | N | Z | V | C | Normal | Advanced |
| | | | | | | | | | | | | | | | | | | |
| MOV.W @aa:24, Rd | W | | | | | 6 | | | @aa:24 → Rd16 | — | — | ⇕ | ⇕ | 0 | — | 8 | | |
| MOV.W Rs, @ERd | W | | 2 | | | | | | Rs16 → @ERd | — | — | ⇕ | ⇕ | 0 | — | 4 | | |
| MOV.W Rs, @(d:16, ERd) | W | | | 4 | | | | | Rs16 → @(d:16, ERd) | — | — | ⇕ | ⇕ | 0 | — | 6 | | |
| MOV.W Rs, @(d:24, ERd) | W | | | 8 | | | | | Rs16 → @(d:24, ERd) | — | — | ⇕ | ⇕ | 0 | — | 10 | | |
| MOV.W Rs, @-ERd | W | | | | 2 | | | | ERd32-2 → ERd32 Rs16 → @ERd | — | — | ⇕ | ⇕ | 0 | — | 6 | | |
| MOV.W Rs, @aa:16 | W | | | | | 4 | | | Rs16 → @aa:16 | — | — | ⇕ | ⇕ | 0 | — | 6 | | |
| MOV.W Rs, @aa:24 | W | | | | | 6 | | | Rs16 → @aa:24 | — | — | ⇕ | ⇕ | 0 | — | 8 | | |
| MOVL #xx:32, Rd | L | 6 | | | | | | | #xx:32 → Rd32 | — | — | ⇕ | ⇕ | 0 | — | 6 | | |
| MOVL ERs, ERd | L | | 2 | | | | | | ERs32 → ERd32 | — | — | ⇕ | ⇕ | 0 | — | 2 | | |
| MOVL @ERs, ERd | L | | | 4 | | | | | @ERs → ERd32 | — | — | ⇕ | ⇕ | 0 | — | 8 | | |
| MOVL @(d:16, ERs), ERd | L | | | | 6 | | | | @(d:16, ERs) → ERd32 | — | — | ⇕ | ⇕ | 0 | — | 10 | | |
| MOVL @(d:24, ERs), ERd | L | | | | 10 | | | | @(d:24, ERs) → ERd32 | — | — | ⇕ | ⇕ | 0 | — | 14 | | |
| MOVL @ERs+, ERd | L | | | | | 4 | | | @ERs → ERd32 ERs32+4 → ERs32 | — | — | ⇕ | ⇕ | 0 | — | 10 | | |
| MOVL @aa:16, ERd | L | | | | | 6 | | | @aa:16 → ERd32 | — | — | ⇕ | ⇕ | 0 | — | 10 | | |
| MOVL @aa:24, ERd | L | | | | | 8 | | | @aa:24 → ERd32 | — | — | ⇕ | ⇕ | 0 | — | 12 | | |
| MOVL ERs, @ERd | L | | | 4 | | | | | ERs32 → @ERd | — | — | ⇕ | ⇕ | 0 | — | 8 | | |
| MOVL ERs, @(d:16, ERd) | L | | | | 6 | | | | ERs32 → @(d:16, ERd) | — | — | ⇕ | ⇕ | 0 | — | 10 | | |
| MOVL ERs, @(d:24, ERd) | L | | | | 10 | | | | ERs32 → @(d:24, ERd) | — | — | ⇕ | ⇕ | 0 | — | 14 | | |
| MOVL ERs, @-ERd | L | | | | | 4 | | | ERd32-4 → ERd32 ERs32 → @ERd | — | — | ⇕ | ⇕ | 0 | — | 10 | | |
| MOVL ERs, @aa:16 | L | | | | | 6 | | | ERs32 → @aa:16 | — | — | ⇕ | ⇕ | 0 | — | 10 | | |
| MOVL ERs, @aa:24 | L | | | | | 8 | | | ERs32 → @aa:24 | — | — | ⇕ | ⇕ | 0 | — | 12 | | |
| POP.W Rn | W | | | | | | | 2 | @SP → Rn16 SP+2 → SP | — | — | ⇕ | ⇕ | 0 | — | 6 | | |
| POP.L ERn | L | | | | | | | 4 | @SP → ERn32 SP+4 → SP | — | — | ⇕ | ⇕ | 0 | — | 10 | | |

| Mnemonic | Operand Size | Addressing Mode and Instruction Length (bytes) | | | | | | | Operation | Condition Code | | | | | | No. of States ^{*1} | | |
|--------------------|--------------|--|----|------|------------|--------------|-----|-----------|-----------|-------------------------------|-------------------------------|---|---|---|---|-----------------------------|--------|----------|
| | | #xx | Rn | @ERn | @ (d, ERn) | @ -ERn/@ERn+ | @aa | @ (d, PC) | | @@aa | I | H | N | Z | V | C | Normal | Advanced |
| | | | | | | | | | | | | | | | | | | |
| PUSH.W Rn | W | | | | | | | | 2 | SP-2 → SP Rn16 → @SP | — | — | ↑ | ↓ | 0 | — | 6 | |
| PUSH.L ERn | L | | | | | | | | 4 | SP-4 → SP ERn32 → @SP | — | — | ↑ | ↓ | 0 | — | 10 | |
| MOVFPPE @aa:16, Rd | B | | | | | 4 | | | | Cannot be used in the H8/3008 | Cannot be used in the H8/3008 | | | | | | | |
| MOVTPE Rs, @aa:16 | B | | | | | 4 | | | | Cannot be used in the H8/3008 | Cannot be used in the H8/3008 | | | | | | | |

2. Arithmetic instructions

| Mnemonic | Operand Size | Addressing Mode and Instruction Length (bytes) | | | | | | | Operation | Condition Code | | | | | | No. of States ^{*1} | | | |
|-------------------|--------------|--|----|------|------------|--------------|-----|-----------|-----------|----------------------|---|-----|---|-----|---|-----------------------------|--------|----------|--|
| | | #xx | Rn | @ERn | @ (d, ERn) | @ -ERn/@ERn+ | @aa | @ (d, PC) | | @@aa | I | H | N | Z | V | C | Normal | Advanced | |
| | | | | | | | | | | | | | | | | | | | |
| ADD.B #xx:8, Rd | B | 2 | | | | | | | | Rd8+#xx:8 → Rd8 | — | ↑ | ↓ | ↓ | ↓ | ↓ | ↓ | 2 | |
| ADD.B Rs, Rd | B | 2 | | | | | | | | Rd8+Rs8 → Rd8 | — | ↑ | ↓ | ↓ | ↓ | ↓ | ↓ | 2 | |
| ADD.W #xx:16, Rd | W | 4 | | | | | | | | Rd16+#xx:16 → Rd16 | — | (1) | ↑ | ↓ | ↓ | ↓ | ↓ | 4 | |
| ADD.W Rs, Rd | W | 2 | | | | | | | | Rd16+Rs16 → Rd16 | — | (1) | ↑ | ↓ | ↓ | ↓ | ↓ | 2 | |
| ADD.L #xx:32, ERd | L | 6 | | | | | | | | ERd32+#xx:32 → ERd32 | — | (2) | ↑ | ↓ | ↓ | ↓ | ↓ | 6 | |
| ADD.L ERs, ERd | L | 2 | | | | | | | | ERd32+ERs32 → ERd32 | — | (2) | ↑ | ↓ | ↓ | ↓ | ↓ | 2 | |
| ADDX.B #xx:8, Rd | B | 2 | | | | | | | | Rd8+#xx:8 +C → Rd8 | — | ↑ | ↓ | (3) | ↓ | ↓ | ↓ | 2 | |
| ADDX.B Rs, Rd | B | 2 | | | | | | | | Rd8+Rs8 +C → Rd8 | — | ↑ | ↓ | (3) | ↓ | ↓ | ↓ | 2 | |
| ADDS.L #1, ERd | L | 2 | | | | | | | | ERd32+1 → ERd32 | — | — | — | — | — | — | — | 2 | |
| ADDS.L #2, ERd | L | 2 | | | | | | | | ERd32+2 → ERd32 | — | — | — | — | — | — | — | 2 | |
| ADDS.L #4, ERd | L | 2 | | | | | | | | ERd32+4 → ERd32 | — | — | — | — | — | — | — | 2 | |
| INC.B Rd | B | 2 | | | | | | | | Rd8+1 → Rd8 | — | — | ↑ | ↓ | ↓ | — | — | 2 | |
| INC.W #1, Rd | W | 2 | | | | | | | | Rd16+1 → Rd16 | — | — | ↑ | ↓ | ↓ | — | — | 2 | |
| INC.W #2, Rd | W | 2 | | | | | | | | Rd16+2 → Rd16 | — | — | ↑ | ↓ | ↓ | — | — | 2 | |

| Mnemonic | Operand Size | Addressing Mode and Instruction Length (bytes) | | | | | | | | Operation | Condition Code | | | | | | No. of States ^{*1} | |
|-------------------|--------------|--|----|------|----------|-------------|-----|---------|------|--|----------------|-----|-----|-----|---|---|-----------------------------|----------|
| | | #xx | Rn | @ERn | @(d,ERn) | @_ERn/@ERn+ | @aa | @(d,PC) | @@aa | | I | H | N | Z | V | C | Normal | Advanced |
| | | | | | | | | | | | | | | | | | | |
| INC.L #1, ERd | L | 2 | | | | | | | | ERd32+1 → ERd32 | — | — | ↓ | ↓ | ↓ | — | 2 | |
| INC.L #2, ERd | L | 2 | | | | | | | | ERd32+2 → ERd32 | — | — | ↓ | ↓ | ↓ | — | 2 | |
| DAA Rd | B | 2 | | | | | | | | Rd8 decimal adjust → Rd8 | — | * | ↓ | ↓ | * | — | 2 | |
| SUB.B Rs, Rd | B | 2 | | | | | | | | Rd8-Rs8 → Rd8 | — | ↓ | ↓ | ↓ | ↓ | ↓ | 2 | |
| SUB.W #xx:16, Rd | W | 4 | | | | | | | | Rd16-#xx:16 → Rd16 | — | (1) | ↓ | ↓ | ↓ | ↓ | 4 | |
| SUB.W Rs, Rd | W | 2 | | | | | | | | Rd16-Rs16 → Rd16 | — | (1) | ↓ | ↓ | ↓ | ↓ | 2 | |
| SUB.L #xx:32, ERd | L | 6 | | | | | | | | ERd32-#xx:32 → ERd32 | — | (2) | ↓ | ↓ | ↓ | ↓ | 6 | |
| SUB.L ERs, ERd | L | 2 | | | | | | | | ERd32-ERs32 → ERd32 | — | (2) | ↓ | ↓ | ↓ | ↓ | 2 | |
| SUBX.B #xx:8, Rd | B | 2 | | | | | | | | Rd8-#xx:8-C → Rd8 | — | ↓ | ↓ | (3) | ↓ | ↓ | 2 | |
| SUBX.B Rs, Rd | B | 2 | | | | | | | | Rd8-Rs8-C → Rd8 | — | ↓ | ↓ | (3) | ↓ | ↓ | 2 | |
| SUBS.L #1, ERd | L | 2 | | | | | | | | ERd32-1 → ERd32 | — | — | — | — | — | — | 2 | |
| SUBS.L #2, ERd | L | 2 | | | | | | | | ERd32-2 → ERd32 | — | — | — | — | — | — | 2 | |
| SUBS.L #4, ERd | L | 2 | | | | | | | | ERd32-4 → ERd32 | — | — | — | — | — | — | 2 | |
| DEC.B Rd | B | 2 | | | | | | | | Rd8-1 → Rd8 | — | — | ↓ | ↓ | ↓ | — | 2 | |
| DEC.W #1, Rd | W | 2 | | | | | | | | Rd16-1 → Rd16 | — | — | ↓ | ↓ | ↓ | — | 2 | |
| DEC.W #2, Rd | W | 2 | | | | | | | | Rd16-2 → Rd16 | — | — | ↓ | ↓ | ↓ | — | 2 | |
| DEC.L #1, ERd | L | 2 | | | | | | | | ERd32-1 → ERd32 | — | — | ↓ | ↓ | ↓ | — | 2 | |
| DEC.L #2, ERd | L | 2 | | | | | | | | ERd32-2 → ERd32 | — | — | ↓ | ↓ | ↓ | — | 2 | |
| DAS.Rd | B | 2 | | | | | | | | Rd8 decimal adjust → Rd8 | — | * | ↓ | ↓ | * | — | 2 | |
| MULXU.B Rs, Rd | B | 2 | | | | | | | | Rd8 × Rs8 → Rd16 (unsigned multiplication) | — | — | — | — | — | — | 14 | |
| MULXU.W Rs, ERd | W | 2 | | | | | | | | Rd16 × Rs16 → ERd32 (unsigned multiplication) | — | — | — | — | — | — | 22 | |
| MULXS.B Rs, Rd | B | 4 | | | | | | | | Rd8 × Rs8 → Rd16 (signed multiplication) | — | — | ↓ | ↓ | — | — | 16 | |
| MULXS.W Rs, ERd | W | 4 | | | | | | | | Rd16 × Rs16 → ERd32 (signed multiplication) | — | — | ↓ | ↓ | — | — | 24 | |
| DIVXU.B Rs, Rd | B | 2 | | | | | | | | Rd16 ÷ Rs8 → Rd16 (RdH: remainder, RdL: quotient) (unsigned division) | — | — | (6) | (7) | — | — | 14 | |

| Mnemonic | Operand Size | Addressing Mode and Instruction Length (bytes) | | | | | | | Operation | Condition Code | | | | | | No. of States ^{*1} | | |
|-------------------|--------------|--|----|------|------------|--------------|-----|-----------|-----------|----------------|---|---|---|---|---|-----------------------------|--------|----------|
| | | #xx | Rn | @ERn | @ (d, ERn) | @ -ERn/@ERn+ | @aa | @ (d, PC) | | @ @aa | I | H | N | Z | V | C | Normal | Advanced |
| | | | | | | | | | | | | | | | | | | |
| DIVXU.W Rs, ERd | W | | 2 | | | | | | | | | | | | | | | 22 |
| DIVXS.B Rs, Rd | B | | 4 | | | | | | | | | | | | | | | 16 |
| DIVXS.W Rs, ERd | W | | 4 | | | | | | | | | | | | | | | 24 |
| CMP.B #xx:8, Rd | B | 2 | | | | | | | | | | | | | | | | 2 |
| CMP.B Rs, Rd | B | | 2 | | | | | | | | | | | | | | | 2 |
| CMP.W #xx:16, Rd | W | 4 | | | | | | | | | | | | | | | | 4 |
| CMP.W Rs, Rd | W | | 2 | | | | | | | | | | | | | | | 2 |
| CMP.L #xx:32, ERd | L | 6 | | | | | | | | | | | | | | | | 6 |
| CMP.L ERs, ERd | L | | 2 | | | | | | | | | | | | | | | 2 |
| NEG.B Rd | B | | 2 | | | | | | | | | | | | | | | 2 |
| NEG.W Rd | W | | 2 | | | | | | | | | | | | | | | 2 |
| NEG.L ERd | L | | 2 | | | | | | | | | | | | | | | 2 |
| EXTU.W Rd | W | | 2 | | | | | | | | | | | | | | | 2 |
| EXTU.L ERd | L | | 2 | | | | | | | | | | | | | | | 2 |
| EXTS.W Rd | W | | 2 | | | | | | | | | | | | | | | 2 |
| EXTS.L ERd | L | | 2 | | | | | | | | | | | | | | | 2 |

3. Logic instructions

| Mnemonic | Operand Size | Addressing Mode and Instruction Length (bytes) | | | | | | | | Operation | Condition Code | | | | | | No. of States ^{*1} | | |
|-------------------|--------------|--|----|------|------------|--------------|-----|-----------|-------|-----------|----------------|---|---|---|---|---|-----------------------------|--------|----------|
| | | #xx | Rn | @ERn | @ (d, ERn) | @ -ERn/@ERn+ | @aa | @ (d, PC) | @ @aa | | | I | H | N | Z | V | C | Normal | Advanced |
| | | | | | | | | | | | | | | | | | | | |
| AND.B #xx:8, Rd | B | 2 | | | | | | | | | | — | — | ⇕ | ⇕ | 0 | — | 2 | |
| AND.B Rs, Rd | B | | 2 | | | | | | | | | — | — | ⇕ | ⇕ | 0 | — | 2 | |
| AND.W #xx:16, Rd | W | 4 | | | | | | | | | | — | — | ⇕ | ⇕ | 0 | — | 4 | |
| AND.W Rs, Rd | W | | 2 | | | | | | | | | — | — | ⇕ | ⇕ | 0 | — | 2 | |
| AND.L #xx:32, ERd | L | 6 | | | | | | | | | | — | — | ⇕ | ⇕ | 0 | — | 6 | |
| AND.L ERs, ERd | L | | 4 | | | | | | | | | — | — | ⇕ | ⇕ | 0 | — | 4 | |
| OR.B #xx:8, Rd | B | 2 | | | | | | | | | | — | — | ⇕ | ⇕ | 0 | — | 2 | |
| OR.B Rs, Rd | B | | 2 | | | | | | | | | — | — | ⇕ | ⇕ | 0 | — | 2 | |
| OR.W #xx:16, Rd | W | 4 | | | | | | | | | | — | — | ⇕ | ⇕ | 0 | — | 4 | |
| OR.W Rs, Rd | W | | 2 | | | | | | | | | — | — | ⇕ | ⇕ | 0 | — | 2 | |
| OR.L #xx:32, ERd | L | 6 | | | | | | | | | | — | — | ⇕ | ⇕ | 0 | — | 6 | |
| OR.L ERs, ERd | L | | 4 | | | | | | | | | — | — | ⇕ | ⇕ | 0 | — | 4 | |
| XOR.B #xx:8, Rd | B | 2 | | | | | | | | | | — | — | ⇕ | ⇕ | 0 | — | 2 | |
| XOR.B Rs, Rd | B | | 2 | | | | | | | | | — | — | ⇕ | ⇕ | 0 | — | 2 | |
| XOR.W #xx:16, Rd | W | 4 | | | | | | | | | | — | — | ⇕ | ⇕ | 0 | — | 4 | |
| XOR.W Rs, Rd | W | | 2 | | | | | | | | | — | — | ⇕ | ⇕ | 0 | — | 2 | |
| XOR.L #xx:32, ERd | L | 6 | | | | | | | | | | — | — | ⇕ | ⇕ | 0 | — | 6 | |
| XOR.L ERs, ERd | L | | 4 | | | | | | | | | — | — | ⇕ | ⇕ | 0 | — | 4 | |
| NOT.B Rd | B | | 2 | | | | | | | | | — | — | ⇕ | ⇕ | 0 | — | 2 | |
| NOT.W Rd | W | | 2 | | | | | | | | | — | — | ⇕ | ⇕ | 0 | — | 2 | |
| NOT.L ERd | L | | 2 | | | | | | | | | — | — | ⇕ | ⇕ | 0 | — | 2 | |

4. Shift instructions

| Mnemonic | Operand Size | Addressing Mode and Instruction Length (bytes) | | | | | | | Operation | Condition Code | | | | | | No. of States ^{*1} | | |
|-------------|--------------|--|----|------|------------|-------------|-----|-----------|-----------|----------------|---|---|---|---|---|-----------------------------|--------|----------|
| | | #xx | Rn | @ERn | @ (d, ERn) | @-ERn/@ERn+ | @aa | @ (d, PC) | | @@aa | I | H | N | Z | V | C | Normal | Advanced |
| | | | | | | | | | | | | | | | | | | |
| SHAL.B Rd | B | 2 | | | | | | | | — | — | ↓ | ↓ | ↓ | ↓ | 2 | | |
| SHAL.W Rd | W | 2 | | | | | | | | — | — | ↓ | ↓ | ↓ | ↓ | 2 | | |
| SHAL.L ERd | L | 2 | | | | | | | | — | — | ↓ | ↓ | ↓ | ↓ | 2 | | |
| SHAR.B Rd | B | 2 | | | | | | | | — | — | ↓ | ↓ | 0 | ↓ | 2 | | |
| SHAR.W Rd | W | 2 | | | | | | | | — | — | ↓ | ↓ | 0 | ↓ | 2 | | |
| SHAR.L ERd | L | 2 | | | | | | | | — | — | ↓ | ↓ | 0 | ↓ | 2 | | |
| SHLL.B Rd | B | 2 | | | | | | | | — | — | ↓ | ↓ | 0 | ↓ | 2 | | |
| SHLL.W Rd | W | 2 | | | | | | | | — | — | ↓ | ↓ | 0 | ↓ | 2 | | |
| SHLL.L ERd | L | 2 | | | | | | | | — | — | ↓ | ↓ | 0 | ↓ | 2 | | |
| SHLR.B Rd | B | 2 | | | | | | | | — | — | ↓ | ↓ | 0 | ↓ | 2 | | |
| SHLR.W Rd | W | 2 | | | | | | | | — | — | ↓ | ↓ | 0 | ↓ | 2 | | |
| SHLR.L ERd | L | 2 | | | | | | | | — | — | ↓ | ↓ | 0 | ↓ | 2 | | |
| ROTXL.B Rd | B | 2 | | | | | | | | — | — | ↓ | ↓ | 0 | ↓ | 2 | | |
| ROTXL.W Rd | W | 2 | | | | | | | | — | — | ↓ | ↓ | 0 | ↓ | 2 | | |
| ROTXL.L ERd | L | 2 | | | | | | | | — | — | ↓ | ↓ | 0 | ↓ | 2 | | |
| ROTXR.B Rd | B | 2 | | | | | | | | — | — | ↓ | ↓ | 0 | ↓ | 2 | | |
| ROTXR.W Rd | W | 2 | | | | | | | | — | — | ↓ | ↓ | 0 | ↓ | 2 | | |
| ROTXR.L ERd | L | 2 | | | | | | | | — | — | ↓ | ↓ | 0 | ↓ | 2 | | |
| ROTL.B Rd | B | 2 | | | | | | | | — | — | ↓ | ↓ | 0 | ↓ | 2 | | |
| ROTL.W Rd | W | 2 | | | | | | | | — | — | ↓ | ↓ | 0 | ↓ | 2 | | |
| ROTL.L ERd | L | 2 | | | | | | | | — | — | ↓ | ↓ | 0 | ↓ | 2 | | |
| ROTR.B Rd | B | 2 | | | | | | | | — | — | ↓ | ↓ | 0 | ↓ | 2 | | |
| ROTR.W Rd | W | 2 | | | | | | | | — | — | ↓ | ↓ | 0 | ↓ | 2 | | |
| ROTR.L ERd | L | 2 | | | | | | | | — | — | ↓ | ↓ | 0 | ↓ | 2 | | |

5. Bit manipulation instructions

| Mnemonic | Operand Size | Addressing Mode and Instruction Length (bytes) | | | | | | | | Operation | Condition Code | | | | | | No. of States ^{*1} | | |
|-------------------|--------------|--|----|------|------------|--------------|-----|-----------|-------|-----------|----------------|---|---|---|---|---|-----------------------------|--------|----------|
| | | #xx | Rn | @ERn | @ (d, ERn) | @ -ERn/@ERn+ | @aa | @ (d, PC) | @ @aa | | | I | H | N | Z | V | C | Normal | Advanced |
| | | | | | | | | | | | | | | | | | | | |
| BSET #xx:3, Rd | B | 2 | | | | | | | | | | | | | | | 2 | | |
| BSET #xx:3, @ERd | B | | 4 | | | | | | | | | | | | | | 8 | | |
| BSET #xx:3, @aa:8 | B | | | | | 4 | | | | | | | | | | | 8 | | |
| BSET Rn, Rd | B | 2 | | | | | | | | | | | | | | | 2 | | |
| BSET Rn, @ERd | B | | 4 | | | | | | | | | | | | | | 8 | | |
| BSET Rn, @aa:8 | B | | | | | 4 | | | | | | | | | | | 8 | | |
| BCLR #xx:3, Rd | B | 2 | | | | | | | | | | | | | | | 2 | | |
| BCLR #xx:3, @ERd | B | | 4 | | | | | | | | | | | | | | 8 | | |
| BCLR #xx:3, @aa:8 | B | | | | | 4 | | | | | | | | | | | 8 | | |
| BCLR Rn, Rd | B | 2 | | | | | | | | | | | | | | | 2 | | |
| BCLR Rn, @ERd | B | | 4 | | | | | | | | | | | | | | 8 | | |
| BCLR Rn, @aa:8 | B | | | | | 4 | | | | | | | | | | | 8 | | |
| BNOT #xx:3, Rd | B | 2 | | | | | | | | | | | | | | | 2 | | |
| BNOT #xx:3, @ERd | B | | 4 | | | | | | | | | | | | | | 8 | | |
| BNOT #xx:3, @aa:8 | B | | | | | 4 | | | | | | | | | | | 8 | | |
| BNOT Rn, Rd | B | 2 | | | | | | | | | | | | | | | 2 | | |
| BNOT Rn, @ERd | B | | 4 | | | | | | | | | | | | | | 8 | | |
| BNOT Rn, @aa:8 | B | | | | | 4 | | | | | | | | | | | 8 | | |
| BTST #xx:3, Rd | B | 2 | | | | | | | | | | | ↓ | | | | 2 | | |
| BTST #xx:3, @ERd | B | | 4 | | | | | | | | | | ↓ | | | | 6 | | |
| BTST #xx:3, @aa:8 | B | | | | | 4 | | | | | | | ↓ | | | | 6 | | |
| BTST Rn, Rd | B | 2 | | | | | | | | | | | ↓ | | | | 2 | | |
| BTST Rn, @ERd | B | | 4 | | | | | | | | | | ↓ | | | | 6 | | |
| BTST Rn, @aa:8 | B | | | | | 4 | | | | | | | ↓ | | | | 6 | | |
| BLD #xx:3, Rd | B | 2 | | | | | | | | | | | | | ↓ | | 2 | | |

| Mnemonic | Operand Size | Addressing Mode and Instruction Length (bytes) | | | | | | | Operation | Condition Code | | | | | | No. of States ^{*1} | | |
|--------------------|--------------|--|----|------|------------|--------------|-----|-----------|----------------------------|----------------|---|---|---|---|---|-----------------------------|--------|----------|
| | | #xx | Rn | @ERn | @ (d, ERn) | @ -ERn/@ERn+ | @aa | @ (d, PC) | | @aa | I | H | N | Z | V | C | Normal | Advanced |
| | | | | | | | | | | | | | | | | | | |
| BLD #xx:3, @ERd | B | | 4 | | | | | | (#xx:3 of @ERd) → C | — | — | — | — | — | ↓ | 6 | | |
| BLD #xx:3, @aa:8 | B | | | | | 4 | | | (#xx:3 of @aa:8) → C | — | — | — | — | — | ↓ | 6 | | |
| BILD #xx:3, Rd | B | 2 | | | | | | | ¬(#xx:3 of Rd8) → C | — | — | — | — | — | ↓ | 2 | | |
| BILD #xx:3, @ERd | B | | 4 | | | | | | ¬(#xx:3 of @ERd) → C | — | — | — | — | — | ↓ | 6 | | |
| BILD #xx:3, @aa:8 | B | | | | | 4 | | | ¬(#xx:3 of @aa:8) → C | — | — | — | — | — | ↓ | 6 | | |
| BST #xx:3, Rd | B | 2 | | | | | | | C → (#xx:3 of Rd8) | — | — | — | — | — | — | 2 | | |
| BST #xx:3, @ERd | B | | 4 | | | | | | C → (#xx:3 of @ERd24) | — | — | — | — | — | — | 8 | | |
| BST #xx:3, @aa:8 | B | | | | | 4 | | | C → (#xx:3 of @aa:8) | — | — | — | — | — | — | 8 | | |
| BIST #xx:3, Rd | B | 2 | | | | | | | ¬C → (#xx:3 of Rd8) | — | — | — | — | — | — | 2 | | |
| BIST #xx:3, @ERd | B | | 4 | | | | | | ¬C → (#xx:3 of @ERd24) | — | — | — | — | — | — | 8 | | |
| BIST #xx:3, @aa:8 | B | | | | | 4 | | | ¬C → (#xx:3 of @aa:8) | — | — | — | — | — | — | 8 | | |
| BAND #xx:3, Rd | B | 2 | | | | | | | C ∧ (#xx:3 of Rd8) → C | — | — | — | — | — | ↓ | 2 | | |
| BAND #xx:3, @ERd | B | | 4 | | | | | | C ∧ (#xx:3 of @ERd24) → C | — | — | — | — | — | ↓ | 6 | | |
| BAND #xx:3, @aa:8 | B | | | | | 4 | | | C ∧ (#xx:3 of @aa:8) → C | — | — | — | — | — | ↓ | 6 | | |
| BIAND #xx:3, Rd | B | 2 | | | | | | | C ∧ ¬(#xx:3 of Rd8) → C | — | — | — | — | — | ↓ | 2 | | |
| BIAND #xx:3, @ERd | B | | 4 | | | | | | C ∧ ¬(#xx:3 of @ERd24) → C | — | — | — | — | — | ↓ | 6 | | |
| BIAND #xx:3, @aa:8 | B | | | | | 4 | | | C ∧ ¬(#xx:3 of @aa:8) → C | — | — | — | — | — | ↓ | 6 | | |
| BOR #xx:3, Rd | B | 2 | | | | | | | C ∨ (#xx:3 of Rd8) → C | — | — | — | — | — | ↓ | 2 | | |
| BOR #xx:3, @ERd | B | | 4 | | | | | | C ∨ (#xx:3 of @ERd24) → C | — | — | — | — | — | ↓ | 6 | | |
| BOR #xx:3, @aa:8 | B | | | | | 4 | | | C ∨ (#xx:3 of @aa:8) → C | — | — | — | — | — | ↓ | 6 | | |
| BIOR #xx:3, Rd | B | 2 | | | | | | | C ∨ ¬(#xx:3 of Rd8) → C | — | — | — | — | — | ↓ | 2 | | |
| BIOR #xx:3, @ERd | B | | 4 | | | | | | C ∨ ¬(#xx:3 of @ERd24) → C | — | — | — | — | — | ↓ | 6 | | |
| BIOR #xx:3, @aa:8 | B | | | | | 4 | | | C ∨ ¬(#xx:3 of @aa:8) → C | — | — | — | — | — | ↓ | 6 | | |
| BXOR #xx:3, Rd | B | 2 | | | | | | | C ⊕ (#xx:3 of Rd8) → C | — | — | — | — | — | ↓ | 2 | | |
| BXOR #xx:3, @ERd | B | | 4 | | | | | | C ⊕ (#xx:3 of @ERd24) → C | — | — | — | — | — | ↓ | 6 | | |
| BXOR #xx:3, @aa:8 | B | | | | | 4 | | | C ⊕ (#xx:3 of @aa:8) → C | — | — | — | — | — | ↓ | 6 | | |
| BIXOR #xx:3, Rd | B | 2 | | | | | | | C ⊕ ¬(#xx:3 of Rd8) → C | — | — | — | — | — | ↓ | 2 | | |
| BIXOR #xx:3, @ERd | B | | 4 | | | | | | C ⊕ ¬(#xx:3 of @ERd24) → C | — | — | — | — | — | ↓ | 6 | | |
| BIXOR #xx:3, @aa:8 | B | | | | | 4 | | | C ⊕ ¬(#xx:3 of @aa:8) → C | — | — | — | — | — | ↓ | 6 | | |

6. Branching instructions

| Mnemonic | Operand Size | Addressing Mode and Instruction Length (bytes) | | | | | | | | Operation | Branch Condition | Condition Code | | | | | | No. of States ^{*1} | | |
|---------------------|--------------|--|----|------|------------|--------------|-----|-----------|-------|--|------------------|----------------|---|---|---|---|---|-----------------------------|--------|----------|
| | | #xx | Rn | @ERn | @ (d, ERn) | @ -ERn/@ERn+ | @aa | @ (d, PC) | @ @aa | | | | I | H | N | Z | V | C | Normal | Advanced |
| | | | | | | | | | | | | | | | | | | | | |
| BRA d:8 (BT d:8) | — | | | | | | | | 2 | If condition is true then PC ← PC+d else next; | Always | — | — | — | — | — | — | 4 | | |
| BRA d:16 (BT d:16) | — | | | | | | | | 4 | | | — | — | — | — | — | — | — | 6 | |
| BRN d:8 (BF d:8) | — | | | | | | | | 2 | | Never | — | — | — | — | — | — | 4 | | |
| BRN d:16 (BF d:16) | — | | | | | | | | 4 | | | — | — | — | — | — | — | — | 6 | |
| BHI d:8 | — | | | | | | | | 2 | | C ∨ Z = 0 | — | — | — | — | — | — | 4 | | |
| BHI d:16 | — | | | | | | | | 4 | | | — | — | — | — | — | — | — | 6 | |
| BLS d:8 | — | | | | | | | | 2 | | C ∨ Z = 1 | — | — | — | — | — | — | 4 | | |
| BLS d:16 | — | | | | | | | | 4 | | | — | — | — | — | — | — | — | 6 | |
| BCC d:8 (BHS d:8) | — | | | | | | | | 2 | | C = 0 | — | — | — | — | — | — | 4 | | |
| BCC d:16 (BHS d:16) | — | | | | | | | | 4 | | | — | — | — | — | — | — | — | 6 | |
| BCS d:8 (BLO d:8) | — | | | | | | | | 2 | | C = 1 | — | — | — | — | — | — | 4 | | |
| BCS d:16 (BLO d:16) | — | | | | | | | | 4 | | | — | — | — | — | — | — | — | 6 | |
| BNE d:8 | — | | | | | | | | 2 | | Z = 0 | — | — | — | — | — | — | 4 | | |
| BNE d:16 | — | | | | | | | | 4 | | | — | — | — | — | — | — | — | 6 | |
| BEQ d:8 | — | | | | | | | | 2 | | Z = 1 | — | — | — | — | — | — | 4 | | |
| BEQ d:16 | — | | | | | | | | 4 | | | — | — | — | — | — | — | — | 6 | |
| BVC d:8 | — | | | | | | | | 2 | | V = 0 | — | — | — | — | — | — | 4 | | |
| BVC d:16 | — | | | | | | | | 4 | | | — | — | — | — | — | — | — | 6 | |
| BVS d:8 | — | | | | | | | | 2 | | V = 1 | — | — | — | — | — | — | 4 | | |
| BVS d:16 | — | | | | | | | | 4 | | | — | — | — | — | — | — | — | 6 | |
| BPL d:8 | — | | | | | | | | 2 | | N = 0 | — | — | — | — | — | — | 4 | | |
| BPL d:16 | — | | | | | | | | 4 | | | — | — | — | — | — | — | — | 6 | |
| BMI d:8 | — | | | | | | | | 2 | | N = 1 | — | — | — | — | — | — | 4 | | |
| BMI d:16 | — | | | | | | | | 4 | | | — | — | — | — | — | — | — | 6 | |
| BGE d:8 | — | | | | | | | | 2 | | N ⊕ V = 0 | — | — | — | — | — | — | 4 | | |
| BGE d:16 | — | | | | | | | | 4 | | | — | — | — | — | — | — | — | 6 | |
| BLT d:8 | — | | | | | | | | 2 | | N ⊕ V = 1 | — | — | — | — | — | — | 4 | | |
| BLT d:16 | — | | | | | | | | 4 | | | — | — | — | — | — | — | — | 6 | |
| BGT d:8 | — | | | | | | | | 2 | | Z ∨ (N ⊕ V) = 0 | — | — | — | — | — | — | 4 | | |
| BGT d:16 | — | | | | | | | | 4 | | | — | — | — | — | — | — | — | 6 | |

| Mnemonic | Addressing Mode and Instruction Length (bytes) | | | | | | | | Operation | Branch Condition | Condition Code | | | | | | No. of States ^{*1} | | |
|------------|--|-----|----|------|------------|--------------|-----|-----------|--|------------------|----------------|---|---|---|---|---|-----------------------------|--------|----------|
| | Operand Size | #xx | Rn | @ERn | @ (d, ERn) | @ -ERn/@ERn+ | @aa | @ (d, PC) | | | @aa | I | H | N | Z | V | C | Normal | Advanced |
| | | | | | | | | | | | | | | | | | | | |
| BLE d:8 | — | | | | | | 2 | | If condition is true then PC ← PC+d else next; | Z ∨ (N⊕V) = 1 | — | — | — | — | — | — | 4 | | |
| BLE d:16 | — | | | | | | 4 | | | | — | — | — | — | — | — | — | 6 | |
| JMP @ERn | — | | | 2 | | | | | PC ← ERn | | — | — | — | — | — | — | 4 | | |
| JMP @aa:24 | — | | | | | | 4 | | PC ← aa:24 | | — | — | — | — | — | — | 6 | | |
| JMP @@aa:8 | — | | | | | | | 2 | PC ← @aa:8 | | — | — | — | — | — | — | 8 | 10 | |
| BSR d:8 | — | | | | | | 2 | | PC → @-SP PC ← PC+d:8 | | — | — | — | — | — | — | 6 | 8 | |
| BSR d:16 | — | | | | | | 4 | | PC → @-SP PC ← PC+d:16 | | — | — | — | — | — | — | 8 | 10 | |
| JSR @ERn | — | | | 2 | | | | | PC → @-SP PC ← @ERn | | — | — | — | — | — | — | 6 | 8 | |
| JSR @aa:24 | — | | | | | | 4 | | PC → @-SP PC ← @aa:24 | | — | — | — | — | — | — | 8 | 10 | |
| JSR @@aa:8 | — | | | | | | | 2 | PC → @-SP PC ← @aa:8 | | — | — | — | — | — | — | 8 | 12 | |
| RTS | — | | | | | | | 2 | PC ← @SP+ | | — | — | — | — | — | — | 8 | 10 | |

7. System control instructions

| Mnemonic | Operand Size | Addressing Mode and Instruction Length (bytes) | | | | | | | | Operation | Condition Code | | | | | | No. of States ^{*1} | | |
|-----------------------|--------------|--|----|------|------------|-------------|-----|-----------|------|-----------|--|---|---|---|---|---|-----------------------------|--------|----------|
| | | #xx | Rn | @ERn | @ (d, ERn) | @-ERn/@ERn+ | @aa | @ (d, PC) | @@aa | | | I | H | N | Z | V | C | Normal | Advanced |
| | | | | | | | | | | | | | | | | | | | |
| TRAPA #x:2 | — | | | | | | | | | 2 | PC → @-SP CCR → @-SP <vector> → PC | 1 | — | — | — | — | — | 14 | 16 |
| RTE | — | | | | | | | | | | CCR ← @SP+ PC ← @SP+ | ↑ | ↑ | ↑ | ↑ | ↑ | ↑ | 10 | |
| SLEEP | — | | | | | | | | | | Transition to powerdown state | — | — | — | — | — | — | 2 | |
| LDC #xx:8, CCR | B | 2 | | | | | | | | | #xx:8 → CCR | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | 2 | |
| LDC Rs, CCR | B | | 2 | | | | | | | | Rs8 → CCR | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | 2 | |
| LDC @ERs, CCR | W | | | 4 | | | | | | | @ERs → CCR | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | 6 | |
| LDC @(d:16, ERs), CCR | W | | | | 6 | | | | | | @(d:16, ERs) → CCR | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | 8 | |
| LDC @(d:24, ERs), CCR | W | | | | | 10 | | | | | @(d:24, ERs) → CCR | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | 12 | |
| LDC @ERs+, CCR | W | | | | | | 4 | | | | @ERs → CCR ERs32+2 → ERs32 | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | 8 | |
| LDC @aa:16, CCR | W | | | | | | | 6 | | | @aa:16 → CCR | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | 8 | |
| LDC @aa:24, CCR | W | | | | | | | | 8 | | @aa:24 → CCR | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | 10 | |
| STC CCR, Rd | B | | 2 | | | | | | | | CCR → Rd8 | — | — | — | — | — | — | 2 | |
| STC CCR, @ERd | W | | | 4 | | | | | | | CCR → @ERd | — | — | — | — | — | — | 6 | |
| STC CCR, @(d:16, ERd) | W | | | | 6 | | | | | | CCR → @(d:16, ERd) | — | — | — | — | — | — | 8 | |
| STC CCR, @(d:24, ERd) | W | | | | | 10 | | | | | CCR → @(d:24, ERd) | — | — | — | — | — | — | 12 | |
| STC CCR, @-ERd | W | | | | | | 4 | | | | ERd32-2 → ERd32 CCR → @ERd | — | — | — | — | — | — | 8 | |
| STC CCR, @aa:16 | W | | | | | | | 6 | | | CCR → @aa:16 | — | — | — | — | — | — | 8 | |
| STC CCR, @aa:24 | W | | | | | | | | 8 | | CCR → @aa:24 | — | — | — | — | — | — | 10 | |
| ANDC #xx:8, CCR | B | 2 | | | | | | | | | CCR ∧ #xx:8 → CCR | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | 2 | |
| ORC #xx:8, CCR | B | 2 | | | | | | | | | CCR ∨ #xx:8 → CCR | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | 2 | |
| XORC #xx:8, CCR | B | 2 | | | | | | | | | CCR ⊕ #xx:8 → CCR | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | 2 | |
| NOP | — | | | | | | | | | 2 | PC ← PC+2 | — | — | — | — | — | — | 2 | |

8. Block transfer instructions

| Mnemonic | Operand Size | Addressing Mode and Instruction Length (bytes) | | | | | | | | Operation | Condition Code | | | | | | No. of States*1 | |
|----------|--------------|--|----|------|------------|-------------|-----|-----------|------|-----------|--|---|---|---|---|---|-----------------|----------|
| | | #xx | Rn | @ERn | @ (d, ERn) | @-ERn/@ERn+ | @aa | @ (d, PC) | @@aa | | I | H | N | Z | V | C | Normal | Advanced |
| | | | | | | | | | | | | | | | | | | |
| EEPMOV.B | — | | | | | | | | | 4 | if R4L ≠ 0 repeat @R5 → @R6 R5+1 → R5 R6+1 → R6 R4L-1 → R4L until R4L=0 else next; | — | — | — | — | — | — | 8+4n*2 |
| EEPMOV.W | — | | | | | | | | | 4 | if R4 ≠ 0 repeat @R5 → @R6 R5+1 → R5 R6+1 → R6 R4-1 → R4 until R4L=0 else next; | — | — | — | — | — | — | 8+4n*2 |

- Notes:
- The number of states is the number of states required for execution when the instruction and its operands are located in on-chip memory. For other cases see section A.3, Number of States Required for Execution.
 - n is the value set in register R4L or R4.
 - Set to 1 when a carry or borrow occurs at bit 11; otherwise cleared to 0.
 - Set to 1 when a carry or borrow occurs at bit 27; otherwise cleared to 0.
 - Retains its previous value when the result is zero; otherwise cleared to 0.
 - Set to 1 when the adjustment produces a carry; otherwise retains its previous value.
 - The number of states required for execution of an instruction that transfers data in synchronization with the E clock is variable.
 - Set to 1 when the divisor is negative; otherwise cleared to 0.
 - Set to 1 when the divisor is zero; otherwise cleared to 0.
 - Set to 1 when the quotient is negative; otherwise cleared to 0.

A.2 Operation Code Maps

Table A.2 Operation Code Map (1)

Instruction code:

| | | |
|----------|----------|----|
| 1st byte | 2nd byte | |
| AH | AL | BL |

Instruction when most significant bit of BH is 0.

Instruction when most significant bit of BH is 1.

| AL | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|----|------------------|------------------|------------------|------------------|------|-------|-------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------|------------------|
| AH | NOP | Table A.2 (2) | STC | LDC | ORC | XORC | ANDC | LDC | ADD | Table A.2 (2) | Table A.2 (2) | Table A.2 (2) | Table A.2 (2) | MOV | ADDX | Table A.2 (2) |
| | Table A.2 (2) | Table A.2 (2) | Table A.2 (2) | Table A.2 (2) | OR.B | XOR.B | AND.B | Table A.2 (2) | SUB | Table A.2 (2) | Table A.2 (2) | Table A.2 (2) | Table A.2 (2) | CMP | SUBX | Table A.2 (2) |
| 2 | MOV.B | | | | | | | | | | | | | | | |
| 3 | MOV | | | | | | | | | | | | | | | |
| 4 | BRA | BRN | BHI | BLS | BCC | BCS | BNE | BNQ | BVC | BVS | BPL | BMI | BGE | BLT | BGT | BLE |
| 5 | MULXU | DIVXU | MULXU | DIVXU | RTS | BSR | RTE | TRAPA | Table A.2 (2) | JMP | BSR | | JSR | | | |
| 6 | BSET | BNOT | BCLR | BTST | OR | XOR | AND | BST | MOV | | | | | | | |
| 7 | | | | | BOR | BXOR | BAND | BISL | MOV | Table A.2 (2) | Table A.2 (2) | Table A.2 (2) | Table A.2 (2) | Table A.2 (3) | | |
| 8 | ADD | | | | | | | | | | | | | | | |
| 9 | ADDX | | | | | | | | | | | | | | | |
| A | CMP | | | | | | | | | | | | | | | |
| B | SUBX | | | | | | | | | | | | | | | |
| C | OR | | | | | | | | | | | | | | | |
| D | XOR | | | | | | | | | | | | | | | |
| E | AND | | | | | | | | | | | | | | | |
| F | MOV | | | | | | | | | | | | | | | |

Table A.2 Operation Code Map (2)

Instruction code:

| | | | |
|----------|----|----------|----|
| 1st byte | | 2nd byte | |
| AH | AL | BH | BL |

| BH AH AL | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|-------------|-------|-----|-----|-------|---------|-----|-----|-----|-------|-----|-----|------|------------------|------------------|-----|------------------|
| 01 | MOV | | | | LDC/STC | | | | SLEEP | | | | Table A.2 (3) | Table A.2 (3) | | Table A.2 (3) |
| 0A | INC | | | | | | | | | | | | ADD | | | |
| 0B | ADDS | | | | | INC | | INC | ADDS | | | | | INC | | INC |
| 0F | DAA | | | | | | | | | | | | MOV | | | |
| 10 | SHLL | | | SHLL | | | | | SHAL | | | SHAL | | | | |
| 11 | SHLR | | | SHLR | | | | | SHAR | | | SHAR | | | | |
| 12 | ROTXL | | | ROTXL | | | | | ROTL | | | ROTL | | | | |
| 13 | ROTXR | | | ROTXR | | | | | ROTR | | | ROTR | | | | |
| 17 | NOT | | | NOT | | | | | EXTU | | | NEG | | | | EXTS |
| 1A | DEC | | | | | | | | | | | | SUB | | | |
| 1B | SUBS | | | | | DEC | | DEC | SUBS | | | | | DEC | | DEC |
| 1F | DAS | | | | | | | | | | | | CMP | | | |
| 58 | BRA | BRN | BHI | BLS | BCC | BCS | BNE | BEQ | BVC | BVS | BPL | BMI | BGE | BLT | BGT | BLE |
| 79 | MOV | ADD | CMP | SUB | OR | XOR | AND | | | | | | | | | |
| 7A | MOV | ADD | CMP | SUB | OR | XOR | AND | | | | | | | | | |

Table A.2 Operation Code Map (3)

Instruction code:

| | | | |
|----------|----------|----------|----------|
| 1st byte | 2nd byte | 3rd byte | 4th byte |
| AH AL | BH BL | CH CL | DH DL |

Instruction when most significant bit of DH is 0.

Instruction when most significant bit of DH is 1.

| CL | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|--------------------|-------|-------|-------|-------|------|-------|-------|-----|------|------|------|------|------|-------|-------|------|
| AH ALBH BLCH | | | | | | | | | | LDC | | LDC | | LDC | | LDC |
| 01406 | | | | | | | | | | LDC | | LDC | | LDC | | LDC |
| 01C05 | MULXS | | MULXS | | | | | | | | | | | | | |
| 01D05 | | DIVXS | | DIVXS | | | | | | | | | | | | |
| 01F06 | | | | | OR | XOR | AND | | | | | | | | | |
| 7C06 *1 | | | | BTST | | | | | | | | | | | | |
| 7C07 *1 | | | | BTST | BOR | BXOR | BAND | BLD | BOR | BXOR | BAND | BILD | BOR | BXOR | BAND | BILD |
| 7D06 *1 | BSET | BNOT | BCLR | | BIOR | BIXOR | BIAND | BST | BSET | BNOT | BCLR | | BIOR | BIXOR | BIAND | BST |
| 7D07 *1 | BSET | BNOT | BCLR | | | | | | | | | | | | | |
| 7Eaa6 *2 | | | | BTST | | | | | | | | | | | | |
| 7Eaa7 *2 | | | | BTST | BOR | BXOR | BAND | BLD | BOR | BXOR | BAND | BILD | BOR | BXOR | BAND | BILD |
| 7Faa6 *2 | BSET | BNOT | BCLR | | BIOR | BIXOR | BIAND | BST | BSET | BNOT | BCLR | | BIOR | BIXOR | BIAND | BST |
| 7Faa7 *2 | BSET | BNOT | BCLR | | | | | | | | | | | | | |

Notes: 1. r is the register designation field.
 2. aa is the absolute address field.

A.3 Number of States Required for Execution

The tables in this section can be used to calculate the number of states required for instruction execution by the H8/300H CPU. Table A.4 indicates the number of instruction fetch, data read/write, and other cycles occurring in each instruction. Table A.3 indicates the number of states required per cycle according to the bus size. The number of states required for execution of an instruction can be calculated from these two tables as follows:

$$\text{Number of states} = I \times S_I + J \times S_J + K \times S_K + L \times S_L + M \times S_M + N \times S_N$$

Examples of Calculation of Number of States Required for Execution

Examples: Advanced mode, stack located in external address space, on-chip supporting modules accessed with 8-bit bus width, external devices accessed in three states with one wait state and 16-bit bus width.

BSET #0, @FFFFC7:8

From table A.4, $I = L = 2$ and $J = K = M = N = 0$

From table A.3, $S_I = 4$ and $S_L = 3$

Number of states = $2 \times 4 + 2 \times 3 = 14$

JSR @@30

From table A.4, $I = J = K = 2$ and $L = M = N = 0$

From table A.3, $S_I = S_J = S_K = 4$

Number of states = $2 \times 4 + 2 \times 4 + 2 \times 4 = 24$

Table A.3 Number of States per Cycle

| Cycle | On-Chip Memory | Access Conditions | | | | | | |
|---------------------|----------------|---------------------------|------------|-----------------|----------------|----------------|----------------|---------|
| | | On-Chip Supporting Module | | External Device | | | | |
| | | 8-Bit Bus | 16-Bit Bus | 8-Bit Bus | | 16-Bit Bus | | |
| | | | | 2-State Access | 3-State Access | 2-State Access | 3-State Access | |
| Instruction fetch | S_I | 2 | 6 | 3 | 4 | $6 + 2m$ | 2 | $3 + m$ |
| Branch address read | S_J | | | | | | | |
| Stack operation | S_K | | | | | | | |
| Byte data access | S_L | | 3 | | 2 | $3 + m$ | | |
| Word data access | S_M | | 6 | | 4 | $6 + 2m$ | | |
| Internal operation | S_N | 1 | | | | | | |

Legend:

m: Number of wait states inserted into external device access

Table A.4 Number of Cycles per Instruction

| Instruction | Mnemonic | Instruction | Branch | Stack | Byte Data | Word Data | Internal |
|-------------|-------------------|-------------|------------|-----------|-----------|-----------|-----------|
| | | Fetch | Addr. Read | Operation | Access | Access | Operation |
| | | I | J | K | L | M | N |
| ADD | ADD.B #xx:8, Rd | 1 | | | | | |
| | ADD.B Rs, Rd | 1 | | | | | |
| | ADD.W #xx:16, Rd | 2 | | | | | |
| | ADD.W Rs, Rd | 1 | | | | | |
| | ADD.L #xx:32, ERd | 3 | | | | | |
| | ADD.L ERs, ERd | 1 | | | | | |
| ADDS | ADDS #1/2/4, ERd | 1 | | | | | |
| ADDX | ADDX #xx:8, Rd | 1 | | | | | |
| | ADDX Rs, Rd | 1 | | | | | |
| AND | AND.B #xx:8, Rd | 1 | | | | | |
| | AND.B Rs, Rd | 1 | | | | | |
| | AND.W #xx:16, Rd | 2 | | | | | |
| | AND.W Rs, Rd | 1 | | | | | |
| | AND.L #xx:32, ERd | 3 | | | | | |
| | AND.L ERs, ERd | 2 | | | | | |
| ANDC | ANDC #xx:8, CCR | 1 | | | | | |
| BAND | BAND #xx:3, Rd | 1 | | | | | |
| | BAND #xx:3, @ERd | 2 | | | 1 | | |
| | BAND #xx:3, @aa:8 | 2 | | | 1 | | |
| Bcc | BRA d:8 (BT d:8) | 2 | | | | | |
| | BRN d:8 (BF d:8) | 2 | | | | | |
| | BHI d:8 | 2 | | | | | |
| | BLS d:8 | 2 | | | | | |
| | BCC d:8 (BHS d:8) | 2 | | | | | |
| | BCS d:8 (BLO d:8) | 2 | | | | | |
| | BNE d:8 | 2 | | | | | |
| | BEQ d:8 | 2 | | | | | |
| | BVC d:8 | 2 | | | | | |
| | BVS d:8 | 2 | | | | | |
| | BPL d:8 | 2 | | | | | |
| | BMI d:8 | 2 | | | | | |
| | BGE d:8 | 2 | | | | | |
| | BLT d:8 | 2 | | | | | |
| | BGT d:8 | 2 | | | | | |
| | BLE d:8 | 2 | | | | | |

| Instruction | Mnemonic | Instruction | Branch | Stack | Byte Data | Word Data | Internal |
|-------------|---------------------|-------------|------------|-----------|-----------|-----------|-----------|
| | | Fetch | Addr. Read | Operation | Access | Access | Operation |
| | | I | J | K | L | M | N |
| Bcc | BRA d:16 (BT d:16) | 2 | | | | | 2 |
| | BRN d:16 (BF d:16) | 2 | | | | | 2 |
| | BHI d:16 | 2 | | | | | 2 |
| | BLS d:16 | 2 | | | | | 2 |
| | BCC d:16 (BHS d:16) | 2 | | | | | 2 |
| | BCS d:16 (BLO d:16) | 2 | | | | | 2 |
| | BNE d:16 | 2 | | | | | 2 |
| | BEQ d:16 | 2 | | | | | 2 |
| | BVC d:16 | 2 | | | | | 2 |
| | BVS d:16 | 2 | | | | | 2 |
| | BPL d:16 | 2 | | | | | 2 |
| | BMI d:16 | 2 | | | | | 2 |
| | BGE d:16 | 2 | | | | | 2 |
| | BLT d:16 | 2 | | | | | 2 |
| | BGT d:16 | 2 | | | | | 2 |
| BLE d:16 | 2 | | | | | 2 | |
| BCLR | BCLR #xx:3, Rd | 1 | | | | | |
| | BCLR #xx:3, @ERd | 2 | | | 2 | | |
| | BCLR #xx:3, @aa:8 | 2 | | | 2 | | |
| | BCLR Rn, Rd | 1 | | | | | |
| | BCLR Rn, @ERd | 2 | | | 2 | | |
| | BCLR Rn, @aa:8 | 2 | | | 2 | | |
| BIAND | BIAND #xx:3, Rd | 1 | | | | | |
| | BIAND #xx:3, @ERd | 2 | | | 1 | | |
| | BIAND #xx:3, @aa:8 | 2 | | | 1 | | |
| BILD | BILD #xx:3, Rd | 1 | | | | | |
| | BILD #xx:3, @ERd | 2 | | | 1 | | |
| | BILD #xx:3, @aa:8 | 2 | | | 1 | | |
| BIOR | BIOR #xx:8, Rd | 1 | | | | | |
| | BIOR #xx:8, @ERd | 2 | | | 1 | | |
| | BIOR #xx:8, @aa:8 | 2 | | | 1 | | |
| BIST | BIST #xx:3, Rd | 1 | | | | | |
| | BIST #xx:3, @ERd | 2 | | | 2 | | |
| | BIST #xx:3, @aa:8 | 2 | | | 2 | | |
| BIXOR | BIXOR #xx:3, Rd | 1 | | | | | |
| | BIXOR #xx:3, @ERd | 2 | | | 1 | | |
| | BIXOR #xx:3, @aa:8 | 2 | | | 1 | | |
| BLD | BLD #xx:3, Rd | 1 | | | | | |
| | BLD #xx:3, @ERd | 2 | | | 1 | | |
| | BLD #xx:3, @aa:8 | 2 | | | 1 | | |

| Instruction | Mnemonic | Instruction | | Branch | Stack | Byte Data | Word Data | Internal |
|-------------|-------------------|-------------|-------|--------|-----------|-----------|-----------|-----------|
| | | Fetch | Addr. | Read | Operation | Access | Access | Operation |
| | | I | J | | K | L | M | N |
| BNOT | BNOT #xx:3, Rd | 1 | | | | | | |
| | BNOT #xx:3, @ERd | 2 | | | | 2 | | |
| | BNOT #xx:3, @aa:8 | 2 | | | | 2 | | |
| | BNOT Rn, Rd | 1 | | | | | | |
| | BNOT Rn, @ERd | 2 | | | | 2 | | |
| | BNOT Rn, @aa:8 | 2 | | | | 2 | | |
| BOR | BOR #xx:3, Rd | 1 | | | | | | |
| | BOR #xx:3, @ERd | 2 | | | | 1 | | |
| | BOR #xx:3, @aa:8 | 2 | | | | 1 | | |
| BSET | BSET #xx:3, Rd | 1 | | | | | | |
| | BSET #xx:3, @ERd | 2 | | | | 2 | | |
| | BSET #xx:3, @aa:8 | 2 | | | | 2 | | |
| | BSET Rn, Rd | 1 | | | | | | |
| | BSET Rn, @ERd | 2 | | | | 2 | | |
| | BSET Rn, @aa:8 | 2 | | | | 2 | | |
| BSR | BSR d:8 | Normal | | | 1 | | | |
| | | Advanced | 2 | | 2 | | | |
| | BSR d:16 | Normal | 2 | | | 1 | | 2 |
| | | Advanced | 2 | | | 2 | | 2 |
| BST | BST #xx:3, Rd | 1 | | | | | | |
| | BST #xx:3, @ERd | 2 | | | | 2 | | |
| | BST #xx:3, @aa:8 | 2 | | | | 2 | | |
| BTST | BTST #xx:3, Rd | 1 | | | | | | |
| | BTST #xx:3, @ERd | 2 | | | | 1 | | |
| | BTST #xx:3, @aa:8 | 2 | | | | 1 | | |
| | BTST Rn, Rd | 1 | | | | | | |
| | BTST Rn, @ERd | 2 | | | | 1 | | |
| | BTST Rn, @aa:8 | 2 | | | | 1 | | |
| BXOR | BXOR #xx:3, Rd | 1 | | | | | | |
| | BXOR #xx:3, @ERd | 2 | | | | 1 | | |
| | BXOR #xx:3, @aa:8 | 2 | | | | 1 | | |
| CMP | CMP.B #xx:8, Rd | 1 | | | | | | |
| | CMP.B Rs, Rd | 1 | | | | | | |
| | CMP.W #xx:16, Rd | 2 | | | | | | |
| | CMP.W Rs, Rd | 1 | | | | | | |
| | CMP.L #xx:32, ERd | 3 | | | | | | |
| | CMP.L ERs, ERd | 1 | | | | | | |
| DAA | DAA Rd | 1 | | | | | | |
| DAS | DAS Rd | 1 | | | | | | |

| Instruction Mnemonic | | Instruction | | Branch | Stack | Byte Data | Word Data | Internal | |
|----------------------|-----------------------|-------------|------------|--------|-----------|---------------|-----------|-----------|---|
| | | Fetch | Addr. Read | Read | Operation | Access | Access | Operation | |
| | | I | J | K | L | M | N | | |
| DEC | DEC.B Rd | 1 | | | | | | | |
| | DEC.W #1/2, Rd | 1 | | | | | | | |
| | DEC.L #1/2, ERd | 1 | | | | | | | |
| DIVXS | DIVXS.B Rs, Rd | 2 | | | | | | 12 | |
| | DIVXS.W Rs, ERd | 2 | | | | | | 20 | |
| DIVXU | DIVXU.B Rs, Rd | 1 | | | | | | 12 | |
| | DIVXU.W Rs, ERd | 1 | | | | | | 20 | |
| EEPMOV | EEPMOV.B | 2 | | | | $2n + 2^{*1}$ | | | |
| | EEPMOV.W | 2 | | | | $2n + 2^{*1}$ | | | |
| EXTS | EXTS.W Rd | 1 | | | | | | | |
| | EXTS.L ERd | 1 | | | | | | | |
| EXTU | EXTU.W Rd | 1 | | | | | | | |
| | EXTU.L ERd | 1 | | | | | | | |
| INC | INC.B Rd | 1 | | | | | | | |
| | INC.W #1/2, Rd | 1 | | | | | | | |
| | INC.L #1/2, ERd | 1 | | | | | | | |
| JMP | JMP @ERn | 2 | | | | | | | |
| | JMP @aa:24 | 2 | | | | | | 2 | |
| | JMP @ @aa:8 | Normal | 2 | 1 | | | | | 2 |
| | | Advanced | 2 | 2 | | | | | 2 |
| JSR | JSR @ERn | Normal | 2 | | 1 | | | | |
| | | Advanced | 2 | | 2 | | | | |
| | JSR @aa:24 | Normal | 2 | | 1 | | | 2 | |
| | | Advanced | 2 | | 2 | | | 2 | |
| | JSR @ @aa:8 | Normal | 2 | 1 | 1 | | | | |
| | | Advanced | 2 | 2 | 2 | | | | |
| LDC | LDC #xx:8, CCR | 1 | | | | | | | |
| | LDC Rs, CCR | 1 | | | | | | | |
| | LDC @ERs, CCR | 2 | | | | 1 | | | |
| | LDC @(d:16, ERs), CCR | 3 | | | | 1 | | | |
| | LDC @(d:24, ERs), CCR | 5 | | | | 1 | | | |
| | LDC @ERs+, CCR | 2 | | | | 1 | | 2 | |
| | LDC @aa:16, CCR | 3 | | | | 1 | | | |
| | LDC @aa:24, CCR | 4 | | | | 1 | | | |

| Instruction Mnemonic | Instruction Fetch | Branch Addr. Read | Stack Operation | Byte Data Access | Word Data Access | Internal Operation |
|----------------------|-------------------------|----------------------|--------------------|---------------------|---------------------|-----------------------|
| | | | | | | |
| MOV | MOV.B #xx:8, Rd | 1 | | | | |
| | MOV.B Rs, Rd | 1 | | | | |
| | MOV.B @ERs, Rd | 1 | | 1 | | |
| | MOV.B @(d:16, ERs), Rd | 2 | | 1 | | |
| | MOV.B @(d:24, ERs), Rd | 4 | | 1 | | |
| | MOV.B @ERs+, Rd | 1 | | 1 | | 2 |
| | MOV.B @aa:8, Rd | 1 | | 1 | | |
| | MOV.B @aa:16, Rd | 2 | | 1 | | |
| | MOV.B @aa:24, Rd | 3 | | 1 | | |
| | MOV.B Rs, @ERd | 1 | | 1 | | |
| | MOV.B Rs, @(d:16, ERd) | 2 | | 1 | | |
| | MOV.B Rs, @(d:24, ERd) | 4 | | 1 | | |
| | MOV.B Rs, @-ERd | 1 | | 1 | | 2 |
| | MOV.B Rs, @aa:8 | 1 | | 1 | | |
| | MOV.B Rs, @aa:16 | 2 | | 1 | | |
| | MOV.B Rs, @aa:24 | 3 | | 1 | | |
| | MOV.W #xx:16, Rd | 2 | | | | |
| | MOV.W Rs, Rd | 1 | | | | |
| | MOV.W @ERs, Rd | 1 | | | 1 | |
| | MOV.W @(d:16, ERs), Rd | 2 | | | 1 | |
| | MOV.W @(d:24, ERs), Rd | 4 | | | 1 | |
| | MOV.W @ERs+, Rd | 1 | | | 1 | 2 |
| | MOV.W @aa:16, Rd | 2 | | | 1 | |
| | MOV.W @aa:24, Rd | 3 | | | 1 | |
| | MOV.W Rs, @ERd | 1 | | | 1 | |
| | MOV.W Rs, @(d:16, ERd) | 2 | | | 1 | |
| | MOV.W Rs, @(d:24, ERd) | 4 | | | 1 | |
| | MOV.W Rs, @-ERd | 1 | | | 1 | 2 |
| | MOV.W Rs, @aa:16 | 2 | | | 1 | |
| | MOV.W Rs, @aa:24 | 3 | | | 1 | |
| | MOV.L #xx:32, ERd | 3 | | | | |
| | MOV.L ERs, ERd | 1 | | | | |
| | MOV.L @ERs, ERd | 2 | | | 2 | |
| | MOV.L @(d:16, ERs), ERd | 3 | | | 2 | |
| | MOV.L @(d:24, ERs), ERd | 5 | | | 2 | |
| | MOV.L @ERs+, ERd | 2 | | | 2 | 2 |
| | MOV.L @aa:16, ERd | 3 | | | 2 | |
| | MOV.L @aa:24, ERd | 4 | | | 2 | |
| | MOV.L ERs, @ERd | 2 | | | 2 | |
| | MOV.L ERs, @(d:16, ERd) | 3 | | | 2 | |
| | MOV.L ERs, @(d:24, ERd) | 5 | | | 2 | |
| | MOV.L ERs, @-ERd | 2 | | | 2 | 2 |
| | MOV.L ERs, @aa:16 | 3 | | | 2 | |
| | MOV.L ERs, @aa:24 | 4 | | | 2 | |

| Instruction | Mnemonic | Instruction | Branch | Stack | Byte Data | Word Data | Internal |
|-------------|--------------------------------|-------------|------------|-----------|-----------|-----------|-----------|
| | | Fetch | Addr. Read | Operation | Access | Access | Operation |
| | | I | J | K | L | M | N |
| MOVFP | MOVFP @aa:16, Rd ^{*2} | | | | 1 | | |
| MOVTP | MOVTP Rs, @aa:16 ^{*2} | | | | 1 | | |
| MULXS | MULXS.B Rs, Rd | 2 | | | | | 12 |
| | MULXS.W Rs, ERd | 2 | | | | | 20 |
| MULXU | MULXU.B Rs, Rd | 1 | | | | | 12 |
| | MULXU.W Rs, ERd | 1 | | | | | 20 |
| NEG | NEG.B Rd | 1 | | | | | |
| | NEG.W Rd | 1 | | | | | |
| | NEG.L ERd | 1 | | | | | |
| NOP | NOP | 1 | | | | | |
| NOT | NOT.B Rd | 1 | | | | | |
| | NOT.W Rd | 1 | | | | | |
| | NOT.L ERd | 1 | | | | | |
| OR | OR.B #xx:8, Rd | 1 | | | | | |
| | OR.B Rs, Rd | 1 | | | | | |
| | OR.W #xx:16, Rd | 2 | | | | | |
| | OR.W Rs, Rd | 1 | | | | | |
| | OR.L #xx:32, ERd | 3 | | | | | |
| | OR.L ERs, ERd | 2 | | | | | |
| ORC | ORC #xx:8, CCR | 1 | | | | | |
| POP | POP.W Rn | 1 | | | | 1 | 2 |
| | POP.L ERn | 2 | | | | 2 | 2 |
| PUSH | PUSH.W Rn | 1 | | | | 1 | 2 |
| | PUSH.L ERn | 2 | | | | 2 | 2 |
| ROTL | ROTL.B Rd | 1 | | | | | |
| | ROTL.W Rd | 1 | | | | | |
| | ROTL.L ERd | 1 | | | | | |
| ROTR | ROTR.B Rd | 1 | | | | | |
| | ROTR.W Rd | 1 | | | | | |
| | ROTR.L ERd | 1 | | | | | |
| ROTXL | ROTXL.B Rd | 1 | | | | | |
| | ROTXL.W Rd | 1 | | | | | |
| | ROTXL.L ERd | 1 | | | | | |
| ROTXR | ROTXR.B Rd | 1 | | | | | |
| | ROTXR.W Rd | 1 | | | | | |
| | ROTXR.L ERd | 1 | | | | | |
| RTE | RTE | 2 | | 2 | | 2 | |

| Instruction | Mnemonic | | Instruction | Branch | Stack | Byte Data | Word Data | Internal |
|-------------|-----------------------|----------|-------------|------------|-----------|-----------|-----------|-----------|
| | | | Fetch | Addr. Read | Operation | Access | Access | Operation |
| | | | I | J | K | L | M | N |
| RTS | RTS | Normal | 2 | | 1 | | | 2 |
| | | Advanced | 2 | | 2 | | | 2 |
| SHAL | SHAL.B Rd | | 1 | | | | | |
| | SHAL.W Rd | | 1 | | | | | |
| | SHAL.L ERd | | 1 | | | | | |
| SHAR | SHAR.B Rd | | 1 | | | | | |
| | SHAR.W Rd | | 1 | | | | | |
| | SHAR.L ERd | | 1 | | | | | |
| SHLL | SHLL.B Rd | | 1 | | | | | |
| | SHLL.W Rd | | 1 | | | | | |
| | SHLL.L ERd | | 1 | | | | | |
| SHLR | SHLR.B Rd | | 1 | | | | | |
| | SHLR.W Rd | | 1 | | | | | |
| | SHLR.L ERd | | 1 | | | | | |
| SLEEP | SLEEP | | 1 | | | | | |
| STC | STC CCR, Rd | | 1 | | | | | |
| | STC CCR, @ERd | | 2 | | | | 1 | |
| | STC CCR, @(d:16, ERd) | | 3 | | | | 1 | |
| | STC CCR, @(d:24, ERd) | | 5 | | | | 1 | |
| | STC CCR, @-ERd | | 2 | | | | 1 | 2 |
| | STC CCR, @aa:16 | | 3 | | | | 1 | |
| | STC CCR, @aa:24 | | 4 | | | | 1 | |
| SUB | SUB.B Rs, Rd | | 1 | | | | | |
| | SUB.W #xx:16, Rd | | 2 | | | | | |
| | SUB.W Rs, Rd | | 1 | | | | | |
| | SUB.L #xx:32, ERd | | 3 | | | | | |
| | SUB.L ERs, ERd | | 1 | | | | | |
| SUBS | SUBS #1/2/4, ERd | | 1 | | | | | |
| SUBX | SUBX #xx:8, Rd | | 1 | | | | | |
| | SUBX Rs, Rd | | 1 | | | | | |
| TRAPA | TRAPA #x:2 | Normal | 2 | 1 | 2 | | | 4 |
| | | Advanced | 2 | 2 | 2 | | | 4 |
| XOR | XOR.B #xx:8, Rd | | 1 | | | | | |
| | XOR.B Rs, Rd | | 1 | | | | | |
| | XOR.W #xx:16, Rd | | 2 | | | | | |
| | XOR.W Rs, Rd | | 1 | | | | | |
| | XOR.L #xx:32, ERd | | 3 | | | | | |
| | XOR.L ERs, ERd | | 2 | | | | | |
| XORC | XORC #xx:8, CCR | | 1 | | | | | |

Notes: 1. n is the value set in register R4L or R4. The source and destination are accessed n + 1 times each.

2. Not available in the H8/3008.

Appendix B Internal I/O Registers

B.1 Address List

| Address (Low) | Register Name | Data Bus Width | Bit Names | | | | | | | | Module Name | |
|------------------|------------------|----------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|----------------------|--|
| | | | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | | |
| H'EE000 | — | — | — | — | — | — | — | — | — | — | — | |
| H'EE001 | — | — | — | — | — | — | — | — | — | — | — | |
| H'EE002 | — | — | — | — | — | — | — | — | — | — | — | |
| H'EE003 | P4DDR | 8 | P4 ₇ DDR | P4 ₆ DDR | P4 ₅ DDR | P4 ₄ DDR | P4 ₃ DDR | P4 ₂ DDR | P4 ₁ DDR | P4 ₀ DDR | Port 4 | |
| H'EE004 | — | — | — | — | — | — | — | — | — | — | — | |
| H'EE005 | P6DDR | 8 | — | P6 ₆ DDR | P6 ₅ DDR | P6 ₄ DDR | P6 ₃ DDR | P6 ₂ DDR | P6 ₁ DDR | P6 ₀ DDR | Port 6 | |
| H'EE006 | — | — | — | — | — | — | — | — | — | — | — | |
| H'EE007 | P8DDR | 8 | — | — | — | P8 ₇ DDR | P8 ₆ DDR | P8 ₅ DDR | P8 ₄ DDR | P8 ₃ DDR | Port 8 | |
| H'EE008 | P9DDR | 8 | — | — | P9 ₅ DDR | P9 ₄ DDR | P9 ₃ DDR | P9 ₂ DDR | P9 ₁ DDR | P9 ₀ DDR | Port 9 | |
| H'EE009 | PADDR | 8 | PA ₇ DDR | PA ₆ DDR | PA ₅ DDR | PA ₄ DDR | PA ₃ DDR | PA ₂ DDR | PA ₁ DDR | PA ₀ DDR | Port A | |
| H'EE00A | PBDDR | 8 | PB ₇ DDR | PB ₆ DDR | PB ₅ DDR | PB ₄ DDR | PB ₃ DDR | PB ₂ DDR | PB ₁ DDR | PB ₀ DDR | Port B | |
| H'EE00B | — | — | — | — | — | — | — | — | — | — | — | |
| H'EE00C | — | — | — | — | — | — | — | — | — | — | — | |
| H'EE00D | — | — | — | — | — | — | — | — | — | — | — | |
| H'EE00E | — | — | — | — | — | — | — | — | — | — | — | |
| H'EE00F | — | — | — | — | — | — | — | — | — | — | — | |
| H'EE010 | — | — | — | — | — | — | — | — | — | — | — | |
| H'EE011 | MDCR | 8 | — | — | — | — | — | MDS2 | MDS1 | MDS0 | System control | |
| H'EE012 | SYSCR | 8 | SSBY | STS2 | STS1 | STS0 | UE | NMIEG | SSOE | RAME | | |
| H'EE013 | BRCR | 8 | A23E | A22E | A21E | A20E | — | — | — | BRLE | Bus controller | |
| H'EE014 | ISCR | 8 | — | — | IRQ5SC | IRQ4SC | IRQ3SC | IRQ2SC | IRQ1SC | IRQ0SC | Interrupt controller | |
| H'EE015 | IER | 8 | — | — | IRQ5E | IRQ4E | IRQ3E | IRQ2E | IRQ1E | IRQ0E | | |
| H'EE016 | ISR | 8 | — | — | IRQ5F | IRQ4F | IRQ3F | IRQ2F | IRQ1F | IRQ0F | | |
| H'EE017 | — | — | — | — | — | — | — | — | — | — | — | |
| H'EE018 | IPRA | 8 | IPRA7 | IPRA6 | IPRA5 | IPRA4 | IPRA3 | IPRA2 | IPRA1 | IPRA0 | | |
| H'EE019 | IPRB | 8 | IPRB7 | IPRB6 | — | — | IPRB3 | IPRB2 | — | — | | |
| H'EE01A | DASTCR | 8 | — | — | — | — | — | — | — | DASTE | D/A converter | |
| H'EE01B | DIVCR | 8 | — | — | — | — | — | — | DIV1 | DIV0 | System control | |
| H'EE01C | MSTCRH | 8 | PSTOP | — | — | — | — | — | MSTPH1 | MSTPH0 | | |
| H'EE01D | MSTCRL | 8 | — | — | — | MSTPL4 | MSTPL3 | MSTPL2 | — | MSTPL0 | | |
| H'EE01E | ADRCR | 8 | — | — | — | — | — | — | — | ADRCTL | Bus controller | |
| H'EE01F | CSCR | 8 | CS7E | CS6E | CS5E | CS4E | — | — | — | — | | |

| Address (Low) | Register Name | Data Bus Width | Bit Names | | | | | | | | Module Name |
|------------------|-----------------------------------|----------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|----------------|
| | | | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | |
| H'EE020 | ABWCR | 8 | ABW7 | ABW6 | ABW5 | ABW4 | ABW3 | ABW2 | ABW1 | ABW0 | Bus controller |
| H'EE021 | ASTCR | 8 | AST7 | AST6 | AST5 | AST4 | AST3 | AST2 | AST1 | AST0 | |
| H'EE022 | WCRH | 8 | W71 | W70 | W61 | W60 | W51 | W50 | W41 | W40 | |
| H'EE023 | WCRL | 8 | W31 | W30 | W21 | W20 | W11 | W10 | W01 | W00 | |
| H'EE024 | BCR | 8 | ICIS1 | ICIS0 | —* ¹ | —* ¹ | —* ¹ | —* ¹ | RDEA | WAITE | |
| H'EE025 | — | — | — | — | — | — | — | — | — | — | |
| H'EE026 | Reserved area (access prohibited) | | | | | | | | | | |
| H'EE027 | | | | | | | | | | | |
| H'EE028 | | | | | | | | | | | |
| H'EE029 | | | | | | | | | | | |
| H'EE02A | | | | | | | | | | | |
| H'EE02B | | | | | | | | | | | |
| H'EE02C | | | | | | | | | | | |
| H'EE02D | | | | | | | | | | | |
| H'EE02E | | | | | | | | | | | |
| H'EE02F | | | | | | | | | | | |
| H'EE030 | Reserved area (access prohibited) | | | | | | | | | | |
| H'EE031 | | | | | | | | | | | |
| H'EE032 | | | | | | | | | | | |
| H'EE033 | | | | | | | | | | | |
| H'EE034 | | | | | | | | | | | |
| H'EE035 | | | | | | | | | | | |
| H'EE036 | | | | | | | | | | | |
| H'EE037 | | | | | | | | | | | |
| H'EE038 | | | | | | | | | | | |
| H'EE039 | | | | | | | | | | | |
| H'EE03A | | | | | | | | | | | |
| H'EE03B | | | | | | | | | | | |
| H'EE03C | | | | | | | | | | | |
| H'EE03D | | | | | | | | | | | |
| H'EE03E | P4PCR | 8 | P4 ₇ PCR | P4 ₆ PCR | P4 ₅ PCR | P4 ₄ PCR | P4 ₃ PCR | P4 ₂ PCR | P4 ₁ PCR | P4 ₀ PCR | Port 4 |
| H'EE03F | Reserved area (access prohibited) | | | | | | | | | | |

| Address (Low) | Register Name | Data Bus Width | Bit Names | | | | | | | | Module Name |
|------------------|-----------------------------------|----------------------|-----------|-------|-------|-------|-------|-------|-------|-------|-------------|
| | | | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | |
| H'EE040 | Reserved area (access prohibited) | | | | | | | | | | |
| H'EE041 | | | | | | | | | | | |
| H'EE042 | | | | | | | | | | | |
| H'EE043 | | | | | | | | | | | |
| H'EE044 | | | | | | | | | | | |
| H'EE045 | | | | | | | | | | | |
| H'EE046 | | | | | | | | | | | |
| H'EE047 | | | | | | | | | | | |
| H'EE048 | | | | | | | | | | | |
| H'EE049 | | | | | | | | | | | |
| H'EE04A | | | | | | | | | | | |
| H'EE04B | | | | | | | | | | | |
| H'EE04C | | | | | | | | | | | |
| H'EE04D | | | | | | | | | | | |
| H'EE04E | | | | | | | | | | | |
| H'EE04F | | | | | | | | | | | |
| H'EE050 | Reserved area (access prohibited) | | | | | | | | | | |
| H'EE051 | | | | | | | | | | | |
| H'EE052 | | | | | | | | | | | |
| H'EE053 | | | | | | | | | | | |
| H'EE054 | | | | | | | | | | | |
| H'EE055 | | | | | | | | | | | |
| H'EE056 | | | | | | | | | | | |
| H'EE057 | | | | | | | | | | | |
| H'EE058 | | | | | | | | | | | |
| H'EE059 | | | | | | | | | | | |
| H'EE05A | | | | | | | | | | | |
| H'EE05B | | | | | | | | | | | |
| H'EE05C | | | | | | | | | | | |
| H'EE05D | | | | | | | | | | | |
| H'EE05E | | | | | | | | | | | |
| H'EE05F | | | | | | | | | | | |

| Address (Low) | Register Name | Data Bus Width | Bit Names | | | | | | | | Module Name |
|------------------|-----------------------------------|----------------------|-----------|-------|-------|-------|-------|-------|-------|-------|-------------|
| | | | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | |
| H'EE060 | Reserved area (access prohibited) | | | | | | | | | | |
| H'EE061 | | | | | | | | | | | |
| H'EE062 | | | | | | | | | | | |
| H'EE063 | | | | | | | | | | | |
| H'EE064 | | | | | | | | | | | |
| H'EE065 | | | | | | | | | | | |
| H'EE066 | | | | | | | | | | | |
| H'EE067 | | | | | | | | | | | |
| H'EE068 | | | | | | | | | | | |
| H'EE069 | | | | | | | | | | | |
| H'EE06A | | | | | | | | | | | |
| H'EE06B | | | | | | | | | | | |
| H'EE06C | | | | | | | | | | | |
| H'EE06D | | | | | | | | | | | |
| H'EE06E | | | | | | | | | | | |
| H'EE06F | | | | | | | | | | | |
| H'EE070 | Reserved area (access prohibited) | | | | | | | | | | |
| H'EE071 | | | | | | | | | | | |
| H'EE072 | | | | | | | | | | | |
| H'EE073 | | | | | | | | | | | |
| H'EE074 | | | | | | | | | | | |
| H'EE075 | | | | | | | | | | | |
| H'EE076 | | | | | | | | | | | |
| H'EE077 | | | | | | | | | | | |
| H'EE078 | | | | | | | | | | | |
| H'EE079 | | | | | | | | | | | |
| H'EE07A | | | | | | | | | | | |
| H'EE07B | | | | | | | | | | | |
| H'EE07C | | | | | | | | | | | |
| H'EE07D | | | | | | | | | | | |
| H'EE07E | | | | | | | | | | | |
| H'EE07F | | | | | | | | | | | |

| Address (Low) | Register Name | Data Bus Width | Bit Names | | | | | | | | Module Name |
|------------------|-----------------------------------|----------------------|-----------|-------|-------|-------|-------|-------|-------|-------|-------------|
| | | | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | |
| H'EE080 | Reserved area (access prohibited) | | | | | | | | | | |
| H'EE081 | | | | | | | | | | | |
| H'EE082 | | | | | | | | | | | |
| H'EE083 | | | | | | | | | | | |
| H'EE084 | | | | | | | | | | | |
| H'EE085 | | | | | | | | | | | |
| H'EE086 | | | | | | | | | | | |
| H'EE087 | | | | | | | | | | | |
| H'EE088 | | | | | | | | | | | |
| H'EE089 | | | | | | | | | | | |
| H'EE08A | | | | | | | | | | | |
| H'EE08B | | | | | | | | | | | |
| H'EE08C | | | | | | | | | | | |
| H'EE08D | | | | | | | | | | | |
| H'EE08E | | | | | | | | | | | |
| H'EE08F | | | | | | | | | | | |
| H'EE090 | Reserved area (access prohibited) | | | | | | | | | | |
| H'EE091 | | | | | | | | | | | |
| H'EE092 | | | | | | | | | | | |
| H'EE093 | | | | | | | | | | | |
| H'EE094 | | | | | | | | | | | |
| H'EE095 | | | | | | | | | | | |
| H'EE096 | | | | | | | | | | | |
| H'EE097 | | | | | | | | | | | |
| H'EE098 | | | | | | | | | | | |
| H'EE099 | | | | | | | | | | | |
| H'EE09A | | | | | | | | | | | |
| H'EE09B | | | | | | | | | | | |
| H'EE09C | | | | | | | | | | | |
| H'EE09D | | | | | | | | | | | |
| H'EE09E | | | | | | | | | | | |
| H'EE09F | | | | | | | | | | | |

| Address (Low) | Register Name | Data Bus Width | Bit Names | | | | | | | | Module Name |
|------------------|-----------------------------------|----------------------|-----------|-------|-------|-------|-------|-------|-------|-------|-------------|
| | | | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | |
| H'EE0A0 | Reserved area (access prohibited) | | | | | | | | | | |
| H'EE0A1 | | | | | | | | | | | |
| H'EE0A2 | | | | | | | | | | | |
| H'EE0A3 | | | | | | | | | | | |
| H'EE0A4 | | | | | | | | | | | |
| H'EE0A5 | | | | | | | | | | | |
| H'EE0A6 | | | | | | | | | | | |
| H'EE0A7 | | | | | | | | | | | |
| H'EE0A8 | | | | | | | | | | | |
| H'EE0A9 | | | | | | | | | | | |
| H'EE0AA | | | | | | | | | | | |
| H'EE0AB | | | | | | | | | | | |
| H'EE0AC | | | | | | | | | | | |
| H'EE0AD | | | | | | | | | | | |
| H'EE0AE | | | | | | | | | | | |
| H'EE0AF | | | | | | | | | | | |
| H'EE0B0 | Reserved area (access prohibited) | | | | | | | | | | |
| H'EE0B1 | | | | | | | | | | | |
| H'EE0B2 | | | | | | | | | | | |
| H'EE0B3 | | | | | | | | | | | |
| H'EE0B4 | | | | | | | | | | | |
| H'EE0B5 | | | | | | | | | | | |
| H'EE0B6 | | | | | | | | | | | |
| H'EE0B7 | | | | | | | | | | | |
| H'EE0B8 | | | | | | | | | | | |
| H'EE0B9 | | | | | | | | | | | |
| H'EE0BA | | | | | | | | | | | |
| H'EE0BB | | | | | | | | | | | |
| H'EE0BC | | | | | | | | | | | |
| H'EE0BD | | | | | | | | | | | |
| H'EE0BE | | | | | | | | | | | |
| H'EE0BF | | | | | | | | | | | |

| Address (Low) | Register Name | Data Bus Width | Bit Names | | | | | | | | Module Name |
|------------------|-----------------------------------|----------------------|-----------|-------|-------|-------|-------|-------|-------|-------|-------------|
| | | | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | |
| H'EE0C0 | Reserved area (access prohibited) | | | | | | | | | | |
| H'EE0C1 | | | | | | | | | | | |
| H'EE0C2 | | | | | | | | | | | |
| H'EE0C3 | | | | | | | | | | | |
| H'EE0C4 | | | | | | | | | | | |
| H'EE0C5 | | | | | | | | | | | |
| H'EE0C6 | | | | | | | | | | | |
| H'EE0C7 | | | | | | | | | | | |
| H'EE0C8 | | | | | | | | | | | |
| H'EE0C9 | | | | | | | | | | | |
| H'EE0CA | | | | | | | | | | | |
| H'EE0CB | | | | | | | | | | | |
| H'EE0CC | | | | | | | | | | | |
| H'EE0CD | | | | | | | | | | | |
| H'EE0CE | | | | | | | | | | | |
| H'EE0CF | | | | | | | | | | | |
| H'EE0D0 | Reserved area (access prohibited) | | | | | | | | | | |
| H'EE0D1 | | | | | | | | | | | |
| H'EE0D2 | | | | | | | | | | | |
| H'EE0D3 | | | | | | | | | | | |
| H'EE0D4 | | | | | | | | | | | |
| H'EE0D5 | | | | | | | | | | | |
| H'EE0D6 | | | | | | | | | | | |
| H'EE0D7 | | | | | | | | | | | |
| H'EE0D8 | | | | | | | | | | | |
| H'EE0D9 | | | | | | | | | | | |
| H'EE0DA | | | | | | | | | | | |
| H'EE0DB | | | | | | | | | | | |
| H'EE0DC | | | | | | | | | | | |
| H'EE0DD | | | | | | | | | | | |
| H'EE0DE | | | | | | | | | | | |
| H'EE0DF | | | | | | | | | | | |

| Address (Low) | Register Name | Data Bus Width | Bit Names | | | | | | | | Module Name |
|------------------|-----------------------------------|----------------------|-----------|-------|-------|-------|-------|-------|-------|-------|-------------|
| | | | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | |
| H'EE0E0 | Reserved area (access prohibited) | | | | | | | | | | |
| H'EE0E1 | | | | | | | | | | | |
| H'EE0E2 | | | | | | | | | | | |
| H'EE0E3 | | | | | | | | | | | |
| H'EE0E4 | | | | | | | | | | | |
| H'EE0E5 | | | | | | | | | | | |
| H'EE0E6 | | | | | | | | | | | |
| H'EE0E7 | | | | | | | | | | | |
| H'EE0E8 | | | | | | | | | | | |
| H'EE0E9 | | | | | | | | | | | |
| H'EE0EA | | | | | | | | | | | |
| H'EE0EB | | | | | | | | | | | |
| H'EE0EC | | | | | | | | | | | |
| H'EE0ED | | | | | | | | | | | |
| H'EE0EE | | | | | | | | | | | |
| H'EE0EF | | | | | | | | | | | |
| H'EE0F0 | Reserved area (access prohibited) | | | | | | | | | | |
| H'EE0F1 | | | | | | | | | | | |
| H'EE0F2 | | | | | | | | | | | |
| H'EE0F3 | | | | | | | | | | | |
| H'EE0F4 | | | | | | | | | | | |
| H'EE0F5 | | | | | | | | | | | |
| H'EE0F6 | | | | | | | | | | | |
| H'EE0F7 | | | | | | | | | | | |
| H'EE0F8 | | | | | | | | | | | |
| H'EE0F9 | | | | | | | | | | | |
| H'EE0FA | | | | | | | | | | | |
| H'EE0FB | | | | | | | | | | | |
| H'EE0FC | | | | | | | | | | | |
| H'EE0FD | | | | | | | | | | | |
| H'EE0FE | | | | | | | | | | | |
| H'EE0FF | | | | | | | | | | | |

| Address (Low) | Register Name | Data Bus Width | Bit Names | | | | | | | | Module Name | |
|------------------|-----------------------------------|----------------------|-----------|-------|-------|-------|-------|-------|-------|-------|-------------|--|
| | | | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | | |
| H'FFF20 | Reserved area (access prohibited) | | | | | | | | | | | |
| H'FFF21 | | | | | | | | | | | | |
| H'FFF22 | | | | | | | | | | | | |
| H'FFF23 | | | | | | | | | | | | |
| H'FFF24 | | | | | | | | | | | | |
| H'FFF25 | | | | | | | | | | | | |
| H'FFF26 | | | | | | | | | | | | |
| H'FFF27 | | | | | | | | | | | | |
| H'FFF28 | | | | | | | | | | | | |
| H'FFF29 | | | | | | | | | | | | |
| H'FFF2A | | | | | | | | | | | | |
| H'FFF2B | | | | | | | | | | | | |
| H'FFF2C | | | | | | | | | | | | |
| H'FFF2D | | | | | | | | | | | | |
| H'FFF2E | | | | | | | | | | | | |
| H'FFF2F | | | | | | | | | | | | |
| H'FFF30 | Reserved area (access prohibited) | | | | | | | | | | | |
| H'FFF31 | | | | | | | | | | | | |
| H'FFF32 | | | | | | | | | | | | |
| H'FFF33 | | | | | | | | | | | | |
| H'FFF34 | | | | | | | | | | | | |
| H'FFF35 | | | | | | | | | | | | |
| H'FFF36 | | | | | | | | | | | | |
| H'FFF37 | | | | | | | | | | | | |
| H'FFF38 | | | | | | | | | | | | |
| H'FFF39 | | | | | | | | | | | | |
| H'FFF3A | | | | | | | | | | | | |
| H'FFF3B | | | | | | | | | | | | |
| H'FFF3C | | | | | | | | | | | | |
| H'FFF3D | | | | | | | | | | | | |
| H'FFF3E | | | | | | | | | | | | |
| H'FFF3F | | | | | | | | | | | | |

| Address (Low) | Register Name | Data Bus Width | Bit Names | | | | | | | | Module Name |
|------------------|------------------|----------------------|-----------|-------|-------|-------|-------|-------|-------|-------|-------------|
| | | | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | |
| H'FFF40 | — | — | — | — | — | — | — | — | — | — | |
| H'FFF41 | — | — | — | — | — | — | — | — | — | — | |
| H'FFF42 | — | — | — | — | — | — | — | — | — | — | |
| H'FFF43 | — | — | — | — | — | — | — | — | — | — | |
| H'FFF44 | — | — | — | — | — | — | — | — | — | — | |
| H'FFF45 | — | — | — | — | — | — | — | — | — | — | |
| H'FFF46 | — | — | — | — | — | — | — | — | — | — | |
| H'FFF47 | — | — | — | — | — | — | — | — | — | — | |
| H'FFF48 | — | — | — | — | — | — | — | — | — | — | |
| H'FFF49 | — | — | — | — | — | — | — | — | — | — | |
| H'FFF4A | — | — | — | — | — | — | — | — | — | — | |
| H'FFF4B | — | — | — | — | — | — | — | — | — | — | |
| H'FFF4C | — | — | — | — | — | — | — | — | — | — | |
| H'FFF4D | — | — | — | — | — | — | — | — | — | — | |
| H'FFF4E | — | — | — | — | — | — | — | — | — | — | |
| H'FFF4F | — | — | — | — | — | — | — | — | — | — | |
| H'FFF50 | — | — | — | — | — | — | — | — | — | — | |
| H'FFF51 | — | — | — | — | — | — | — | — | — | — | |
| H'FFF52 | — | — | — | — | — | — | — | — | — | — | |
| H'FFF53 | — | — | — | — | — | — | — | — | — | — | |
| H'FFF54 | — | — | — | — | — | — | — | — | — | — | |
| H'FFF55 | — | — | — | — | — | — | — | — | — | — | |
| H'FFF56 | — | — | — | — | — | — | — | — | — | — | |
| H'FFF57 | — | — | — | — | — | — | — | — | — | — | |
| H'FFF58 | — | — | — | — | — | — | — | — | — | — | |
| H'FFF59 | — | — | — | — | — | — | — | — | — | — | |
| H'FFF5A | — | — | — | — | — | — | — | — | — | — | |
| H'FFF5B | — | — | — | — | — | — | — | — | — | — | |
| H'FFF5C | — | — | — | — | — | — | — | — | — | — | |
| H'FFF5D | — | — | — | — | — | — | — | — | — | — | |
| H'FFF5E | — | — | — | — | — | — | — | — | — | — | |
| H'FFF5F | — | — | — | — | — | — | — | — | — | — | |

| Address (Low) | Register Name | Data Bus Width | Bit Names | | | | | | | | | Module Name |
|------------------|------------------|----------------------|-----------|--------|--------|--------|-------|-------|-------|-------|------------------------|------------------------------|
| | | | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | | |
| H'FFF60 | TSTR | 8 | — | — | — | — | — | — | STR2 | STR1 | STR0 | 16-bit timer, (all channels) |
| H'FFF61 | TSNC | 8 | — | — | — | — | — | — | SYNC2 | SYNC1 | SYNC0 | |
| H'FFF62 | TMDR | 8 | — | MDF | FDIR | — | — | — | PWM2 | PWM1 | PWM0 | |
| H'FFF63 | TOLR | 8 | — | — | TOB2 | TOA2 | TOB1 | TOA1 | TOB0 | TOA0 | | |
| H'FFF64 | TISRA | 8 | — | IMIEA2 | IMIEA1 | IMIEA0 | — | — | IMFA2 | IMFA1 | IMFA0 | |
| H'FFF65 | TISRB | 8 | — | IMIEB2 | IMIEB1 | IMIEB0 | — | — | IMFB2 | IMFB1 | IMFB0 | |
| H'FFF66 | TISRC | 8 | — | OVIE2 | OVIE1 | OVIE0 | — | — | OVF2 | OVF1 | OVF0 | |
| H'FFF67 | — | — | — | — | — | — | — | — | — | — | — | |
| H'FFF68 | 16TCR0 | 8 | — | CCLR1 | CCLR0 | CKEG1 | CKEG0 | TPSC2 | TPSC1 | TPSC0 | 16-bit timer channel 0 | |
| H'FFF69 | TIOR0 | 8 | — | IOB2 | IOB1 | IOB0 | — | IOA2 | IOA1 | IOA0 | | |
| H'FFF6A | 16TCNT0H | 16 | | | | | | | | | | |
| H'FFF6B | 16TCNT0L | | | | | | | | | | | |
| H'FFF6C | GRA0H | 16 | | | | | | | | | | |
| H'FFF6D | GRA0L | | | | | | | | | | | |
| H'FFF6E | GRB0H | 16 | | | | | | | | | | |
| H'FFF6F | GRB0L | | | | | | | | | | | |
| H'FFF70 | 16TCR1 | 8 | — | CCLR1 | CCLR0 | CKEG1 | CKEG0 | TPSC2 | TPSC1 | TPSC0 | 16-bit timer channel 1 | |
| H'FFF71 | TIOR1 | 8 | — | IOB2 | IOB1 | IOB0 | — | IOA2 | IOA1 | IOA0 | | |
| H'FFF72 | 16TCNT1H | 16 | | | | | | | | | | |
| H'FFF73 | 16TCNT1L | | | | | | | | | | | |
| H'FFF74 | GRA1H | 16 | | | | | | | | | | |
| H'FFF75 | GRA1L | | | | | | | | | | | |
| H'FFF76 | GRB1H | 16 | | | | | | | | | | |
| H'FFF77 | GRB1L | | | | | | | | | | | |
| H'FFF78 | 16TCR2 | 8 | — | CCLR1 | CCLR0 | CKEG1 | CKEG0 | TPSC2 | TPSC1 | TPSC0 | 16-bit timer channel 2 | |
| H'FFF79 | TIOR2 | 8 | — | IOB2 | IOB1 | IOB0 | — | IOA2 | IOA1 | IOA0 | | |
| H'FFF7A | 16TCNT2H | 16 | | | | | | | | | | |
| H'FFF7B | 16TCNT2L | | | | | | | | | | | |
| H'FFF7C | GRA2H | 16 | | | | | | | | | | |
| H'FFF7D | GRA2L | | | | | | | | | | | |
| H'FFF7E | GRB2H | 16 | | | | | | | | | | |
| H'FFF7F | GRB2L | | | | | | | | | | | |

| Address (Low) | Register Name | Data Bus Width | Bit Names | | | | | | | | Module Name |
|------------------|----------------------|----------------------|-----------|-------|-------|-------|-------|-------|-------|-------|------------------------------------|
| | | | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | |
| H'FFF80 | 8TCR0 | 16 | CMIEB | CMIEA | OVIE | CCLR1 | CCLR0 | CKS2 | CKS1 | CKS0 | 8-bit timer channels 0 and 1 |
| H'FFF81 | 8TCR1 | 16 | CMIEB | CMIEA | OVIE | CCLR1 | CCLR0 | CKS2 | CKS1 | CKS0 | |
| H'FFF82 | 8TCSR0 | 16 | CMFB | CMFA | OVF | ADTE | OIS3 | OIS2 | OS1 | OS0 | |
| H'FFF83 | 8TCSR1 | 16 | CMFB | CMFA | OVF | ICE | OIS3 | OIS2 | OS1 | OS0 | |
| H'FFF84 | TCORA0 | 16 | | | | | | | | | |
| H'FFF85 | TCORA1 | 16 | | | | | | | | | |
| H'FFF86 | TCORB0 | 16 | | | | | | | | | |
| H'FFF87 | TCORB1 | 16 | | | | | | | | | |
| H'FFF88 | 8TCNT0 | 16 | | | | | | | | | |
| H'FFF89 | 8TCNT1 | 16 | | | | | | | | | |
| H'FFF8A | — | | — | — | — | — | — | — | — | — | |
| H'FFF8B | — | | — | — | — | — | — | — | — | — | |
| H'FFF8C | TCSR* ² | 8 | OVF | WT/IT | TME | — | — | CKS2 | CKS1 | CKS0 | WDT |
| H'FFF8D | TCNT* ² | 8 | | | | | | | | | |
| H'FFF8E | — | | — | — | — | — | — | — | — | — | |
| H'FFF8F | RSTCSR* ² | 8 | WRST | RSTOE | — | — | — | — | — | — | |
| H'FFF90 | 8TCR2 | 16 | CMIEB | CMIEA | OVIE | CCLR1 | CCLR0 | CKS2 | CKS1 | CKS0 | 8-bit timer channels 2 and 3 |
| H'FFF91 | 8TCR3 | 16 | CMIEB | CMIEA | OVIE | CCLR1 | CCLR0 | CKS2 | CKS1 | CKS0 | |
| H'FFF92 | 8TCSR2 | 16 | CMFB | CMFA | OVF | — | OIS3 | OIS2 | OS1 | OS0 | |
| H'FFF93 | 8TCSR3 | 16 | CMFB | CMFA | OVF | ICE | OIS3 | OIS2 | OS1 | OS0 | |
| H'FFF94 | TCORA2 | 16 | | | | | | | | | |
| H'FFF95 | TCORA3 | 16 | | | | | | | | | |
| H'FFF96 | TCORB2 | 16 | | | | | | | | | |
| H'FFF97 | TCORB3 | 16 | | | | | | | | | |
| H'FFF98 | 8TCNT2 | 16 | | | | | | | | | |
| H'FFF99 | 8TCNT3 | 16 | | | | | | | | | |
| H'FFF9A | — | | — | — | — | — | — | — | — | — | |
| H'FFF9B | — | | — | — | — | — | — | — | — | — | |
| H'FFF9C | DADR0 | 8 | | | | | | | | | D/A converter |
| H'FFF9D | DADR1 | 8 | | | | | | | | | |
| H'FFF9E | DACR | 8 | DAOE1 | DAOE0 | DAE | — | — | — | — | — | |
| H'FFF9F | — | 8 | — | — | — | — | — | — | — | — | |

Appendix B Internal I/O Registers

| Address (Low) | Register Name | Data Bus Width | Bit Names | | | | | | | | Module Name |
|------------------|-----------------------------------|----------------------|--------------|--------|--------|--------------|--------|--------|--------|--------|---------------|
| | | | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | |
| H'FFFA0 | TPMR | 8 | — | — | — | — | G3NOV | G2NOV | G1NOV | G0NOV | TPC |
| H'FFFA1 | TPCR | 8 | G3CMS1 | G3CMS0 | G2CMS1 | G2CMS0 | G1CMS1 | G1CMS0 | G0CMS1 | G0CMS0 | |
| H'FFFA2 | NDERB | 8 | NDER15 | NDER14 | NDER13 | NDER12 | NDER11 | NDER10 | NDER9 | NDER8 | |
| H'FFFA3 | NDERA | 8 | NDER7 | NDER6 | NDER5 | NDER4 | NDER3 | NDER2 | NDER1 | NDER0 | |
| H'FFFA4 | NDRB* ³ | 8 | NDR15 | NDR14 | NDR13 | NDR12 | NDR11 | NDR10 | NDR9 | NDR8 | |
| | | | NDR15 | NDR14 | NDR13 | NDR12 | — | — | — | — | |
| H'FFFA5 | NDRA* ³ | 8 | NDR7 | NDR6 | NDR5 | NDR4 | NDR3 | NDR2 | NDR1 | NDR0 | |
| | | | NDR7 | NDR6 | NDR5 | NDR4 | — | — | — | — | |
| H'FFFA6 | NDRB* ³ | 8 | — | — | — | — | — | — | — | — | |
| | | | — | — | — | — | NDR11 | NDR10 | NDR9 | NDR8 | |
| H'FFFA7 | NDRA* ³ | 8 | — | — | — | — | — | — | — | — | |
| | | | — | — | — | — | NDR3 | NDR2 | NDR1 | NDR0 | |
| H'FFFA8 | — | 8 | — | — | — | — | — | — | — | — | |
| H'FFFA9 | — | 8 | — | — | — | — | — | — | — | — | |
| H'FFFAA | — | 8 | — | — | — | — | — | — | — | — | |
| H'FFFAB | — | 8 | — | — | — | — | — | — | — | — | |
| H'FFFAC | — | 8 | — | — | — | — | — | — | — | — | |
| H'FFFAD | — | 8 | — | — | — | — | — | — | — | — | |
| H'FFFAE | — | 8 | — | — | — | — | — | — | — | — | |
| H'FFFAF | — | 8 | — | — | — | — | — | — | — | — | |
| H'FFFB0 | SMR | 8 | C/ \bar{A} | CHR | PE | O/ \bar{E} | STOP | MP | CKS1 | CKS0 | SCI channel 0 |
| H'FFFB1 | BRR | 8 | — | — | — | — | — | — | — | — | |
| H'FFFB2 | SCR | 8 | TIE | RIE | TE | RE | MPIE | TEIE | CKE1 | CKE0 | |
| H'FFFB3 | TDR | 8 | — | — | — | — | — | — | — | — | |
| H'FFFB4 | SSR | 8 | TDRE | RDRF | ORER | FER/ERS | PER | TEND | MPB | MPBT | |
| H'FFFB5 | RDR | 8 | — | — | — | — | — | — | — | — | |
| H'FFFB6 | SCMR | 8 | — | — | — | — | SDIR | SINV | — | SMIF | |
| H'FFFB7 | Reserved area (access prohibited) | | | | | | | | | | |
| H'FFFB8 | SMR | 8 | C/ \bar{A} | CHR | PE | O/ \bar{E} | STOP | MP | CKS1 | CKS0 | SCI channel 1 |
| H'FFFB9 | BRR | 8 | — | — | — | — | — | — | — | — | |
| H'FFBFA | SCR | 8 | TIE | RIE | TE | RE | MPIE | TEIE | CKE1 | CKE0 | |
| H'FFBFB | TDR | 8 | — | — | — | — | — | — | — | — | |
| H'FFBFC | SSR | 8 | TDRE | RDRF | ORER | FER/ERS | PER | TEND | MPB | MPBT | |
| H'FFBFD | RDR | 8 | — | — | — | — | — | — | — | — | |
| H'FFBFE | SCMR | 8 | — | — | — | — | SDIR | SINV | — | SMIF | |
| H'FFBFF | Reserved area (access prohibited) | | | | | | | | | | |

| Address (Low) | Register Name | Data Bus Width | Bit Names | | | | | | | | Module Name |
|------------------|-----------------------------------|----------------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-------------|
| | | | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | |
| H'FFFC0 | Reserved area (access prohibited) | | | | | | | | | | |
| H'FFFC1 | | | | | | | | | | | |
| H'FFFC2 | | | | | | | | | | | |
| H'FFFC3 | | | | | | | | | | | |
| H'FFFC4 | | | | | | | | | | | |
| H'FFFC5 | | | | | | | | | | | |
| H'FFFC6 | | | | | | | | | | | |
| H'FFFC7 | | | | | | | | | | | |
| H'FFFC8 | | | | | | | | | | | |
| H'FFFC9 | | | | | | | | | | | |
| H'FFCA | | | | | | | | | | | |
| H'FFFCB | | | | | | | | | | | |
| H'FFCC | | | | | | | | | | | |
| H'FFCD | | | | | | | | | | | |
| H'FFCE | | | | | | | | | | | |
| H'FFCF | | | | | | | | | | | |
| H'FFFD0 | — | — | — | — | — | — | — | — | — | — | |
| H'FFFD1 | — | — | — | — | — | — | — | — | — | — | |
| H'FFFD2 | — | — | — | — | — | — | — | — | — | — | |
| H'FFFD3 | P4DR | 8 | P4 ₇ | P4 ₆ | P4 ₅ | P4 ₄ | P4 ₃ | P4 ₂ | P4 ₁ | P4 ₀ | Port 4 |
| H'FFFD4 | — | — | — | — | — | — | — | — | — | — | |
| H'FFFD5 | P6DR | 8 | P6 ₇ | P6 ₆ | P6 ₅ | P6 ₄ | P6 ₃ | P6 ₂ | P6 ₁ | P6 ₀ | Port 6 |
| H'FFFD6 | P7DR | 8 | P7 ₇ | P7 ₆ | P7 ₅ | P7 ₄ | P7 ₃ | P7 ₂ | P7 ₁ | P7 ₀ | Port 7 |
| H'FFFD7 | P8DR | 8 | — | — | — | P8 ₄ | P8 ₃ | P8 ₂ | P8 ₁ | P8 ₀ | Port 8 |
| H'FFFD8 | P9DR | 8 | — | — | P9 ₅ | P9 ₄ | P9 ₃ | P9 ₂ | P9 ₁ | P9 ₀ | Port 9 |
| H'FFFD9 | PADR | 8 | PA ₇ | PA ₆ | PA ₅ | PA ₄ | PA ₃ | PA ₂ | PA ₁ | PA ₀ | Port A |
| H'FFFDA | PBDR | 8 | PB ₇ | PB ₆ | PB ₅ | PB ₄ | PB ₃ | PB ₂ | PB ₁ | PB ₀ | Port B |
| H'FFFDB | — | — | — | — | — | — | — | — | — | — | |
| H'FFDC | — | — | — | — | — | — | — | — | — | — | |
| H'FFDD | — | — | — | — | — | — | — | — | — | — | |
| H'FFDE | — | — | — | — | — | — | — | — | — | — | |
| H'FFDF | — | — | — | — | — | — | — | — | — | — | |

| Address (Low) | Register Name | Data Bus Width | Bit Names | | | | | | | | Module Name |
|------------------|------------------|----------------------|-----------|-------|-------|-------|-------|-------|-------|-------|---------------|
| | | | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | |
| H'FFFE0 | ADDRAH | 8 | AD9 | AD8 | AD7 | AD6 | AD5 | AD4 | AD3 | AD2 | A/D converter |
| H'FFFE1 | ADDRAL | 8 | AD1 | AD0 | — | — | — | — | — | — | |
| H'FFFE2 | ADDRBH | 8 | AD9 | AD8 | AD7 | AD6 | AD5 | AD4 | AD3 | AD2 | |
| H'FFFE3 | ADDRBL | 8 | AD1 | AD0 | — | — | — | — | — | — | |
| H'FFFE4 | ADDRCH | 8 | AD9 | AD8 | AD7 | AD6 | AD5 | AD4 | AD3 | AD2 | |
| H'FFFE5 | ADDRCL | 8 | AD1 | AD0 | — | — | — | — | — | — | |
| H'FFFE6 | ADDRDH | 8 | AD9 | AD8 | AD7 | AD6 | AD5 | AD4 | AD3 | AD2 | |
| H'FFFE7 | ADDRDL | 8 | AD1 | AD0 | — | — | — | — | — | — | |
| H'FFFE8 | ADCSR | 8 | ADF | ADIE | ADST | SCAN | CKS | CH2 | CH1 | CH0 | |
| H'FFFE9 | ADCR | 8 | TRGE | — | — | — | — | — | — | — | |

Legend:

WDT: Watchdog timer

TPC: Programmable timing pattern controller

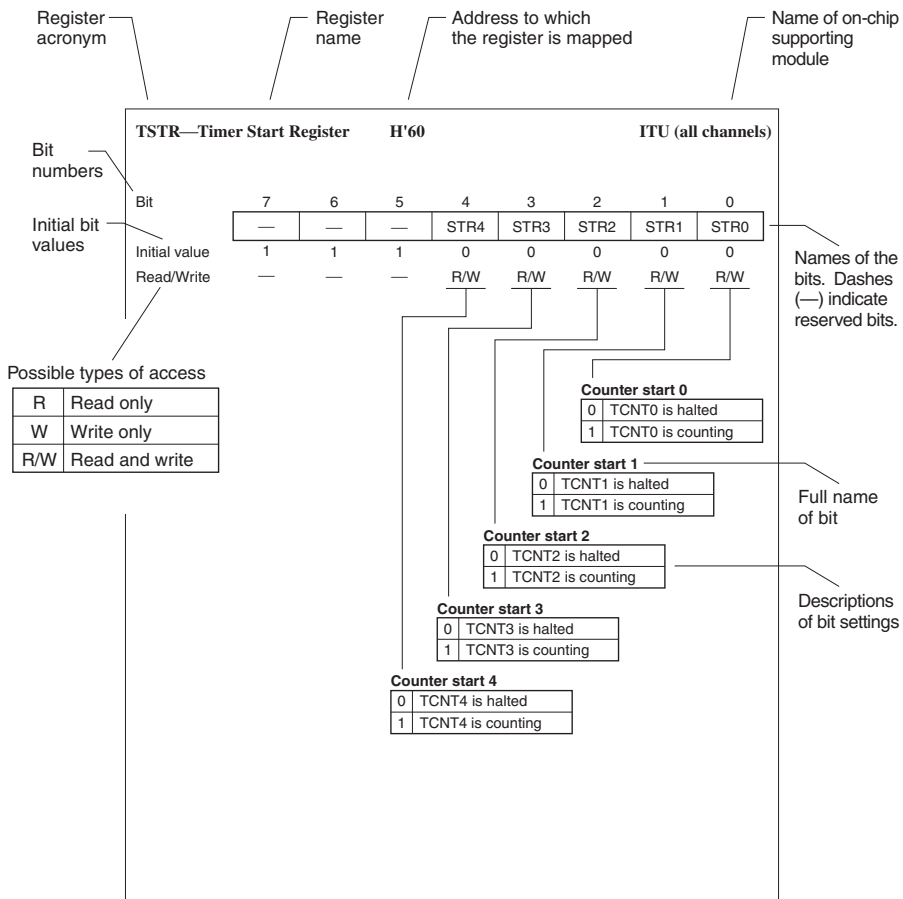
SCI: Serial communication interface

Notes: 1. Writing to bits 5 to 2 of BCR is prohibited.

2. For the procedure for writing to TCSR, TCNT, and RSTCSR, see section 11.2.4, Notes on Register Rewriting.

3. The address depends on the output trigger setting.

B.2 Functions



P4DDR—Port 4 Data Direction Register**H'EE003****Port 4**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|--------|--------|--------|--------|--------|--------|--------|--------|
| | P47DDR | P46DDR | P45DDR | P44DDR | P43DDR | P42DDR | P41DDR | P40DDR |
| Initial value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Read/Write | W | W | W | W | W | W | W | W |

Port 4 input/output select

| | |
|---|----------------|
| 0 | Generic input |
| 1 | Generic output |

P6DDR—Port 6 Data Direction Register**H'EE005****Port 6**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|---|--------|--------|--------|--------|--------|--------|--------|
| | — | P66DDR | P65DDR | P64DDR | P63DDR | P62DDR | P61DDR | P60DDR |
| Initial value | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Read/Write | — | W | W | W | W | W | W | W |

Port 6 input/output select

| | |
|---|----------------|
| 0 | Generic input |
| 1 | Generic output |

P8DDR—Port 8 Data Direction Register**H'EE007****Port 8**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------------|---|---|---|--------|--------|--------|--------|--------|
| | — | — | — | P84DDR | P83DDR | P82DDR | P81DDR | P80DDR |
| Modes 1 to 4 | { | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| Read/Write | — | — | — | W | W | W | W | W |
| Modes 5 to 7 | { | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| Read/Write | — | — | — | W | W | W | W | W |

Port 8 input/output select

| | |
|---|----------------|
| 0 | Generic input |
| 1 | Generic output |

P9DDR—Port 9 Data Direction Register**H'EE008****Port 9**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|---|---|--------|--------|--------|--------|--------|--------|
| | — | — | P95DDR | P94DDR | P93DDR | P92DDR | P91DDR | P90DDR |
| Initial value | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| Read/Write | — | — | W | W | W | W | W | W |

Port 9 input/output select

| | |
|---|----------------|
| 0 | Generic input |
| 1 | Generic output |

PADDR—Port A Data Direction Register**H'EE009****Port A**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|---------------|--------|--------|--------|--------|--------|--------|--------|
| | PA7DDR | PA6DDR | PA5DDR | PA4DDR | PA3DDR | PA2DDR | PA1DDR | PA0DDR |
| Modes 3 and 4 | Initial value | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| | Read/Write | — | W | W | W | W | W | W |
| Modes 1 and 2 | Initial value | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | Read/Write | W | W | W | W | W | W | W |

Port A input/output select

| | |
|---|----------------|
| 0 | Generic input |
| 1 | Generic output |

PBDDR—Port B Data Direction Register**H'EE00A****Port B**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|--------|--------|--------|--------|--------|--------|--------|--------|
| | PB7DDR | PB6DDR | PB5DDR | PB4DDR | PB3DDR | PB2DDR | PB1DDR | PB0DDR |
| Initial value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Read/Write | W | W | W | W | W | W | W | W |

Port B input/output select

| | |
|---|----------------|
| 0 | Generic input |
| 1 | Generic output |

MDCR—Mode Control Register

H'EE011

System control

| | | | | | | | | |
|---------------|---|---|---|---|---|------|------|------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | MDS2 | MDS1 | MDS0 |
| Initial value | 1 | 1 | 0 | 0 | 0 | —* | —* | —* |
| Read/Write | — | — | — | — | — | R | R | R |

Mode select 2 to 0

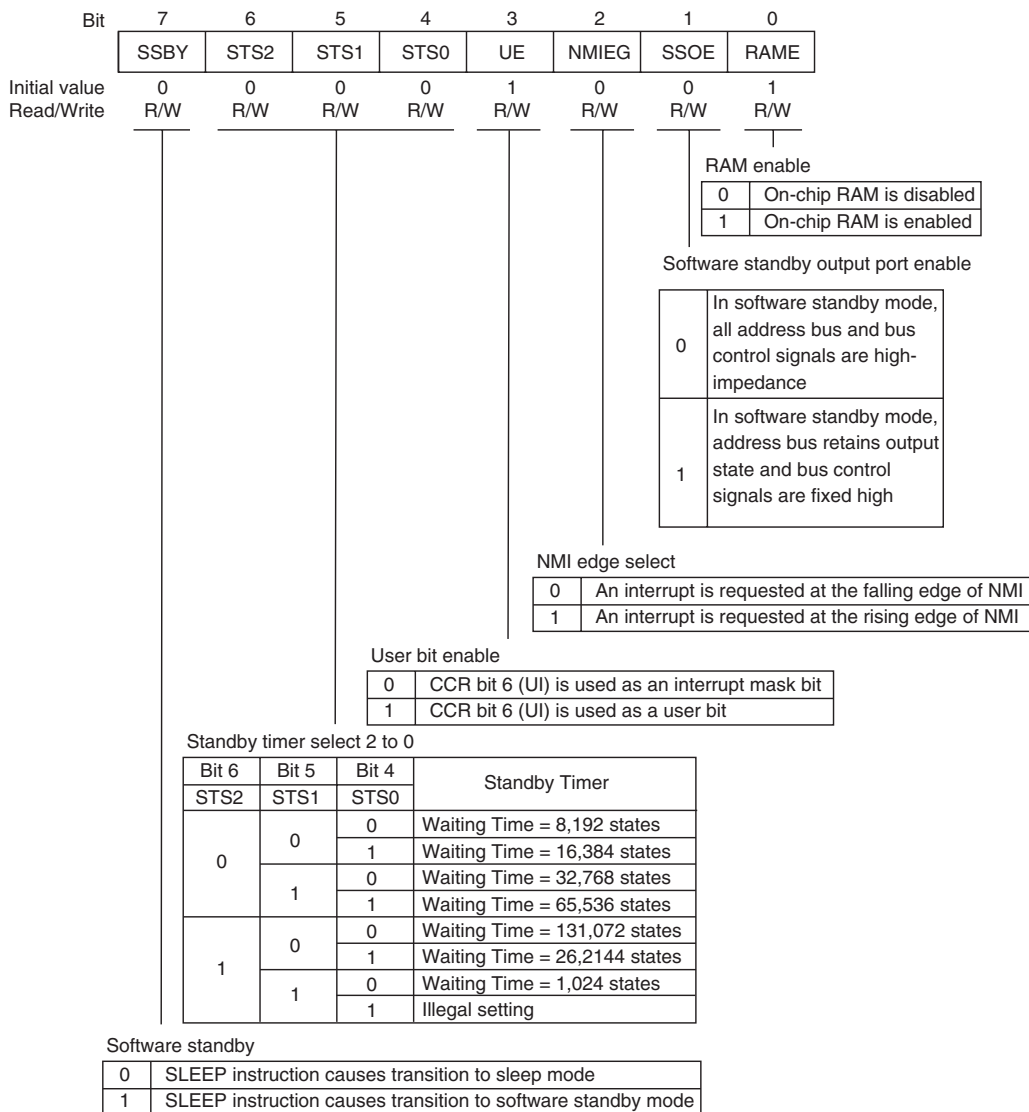
| Bit 2 | Bit 1 | Bit 0 | Operating Mode |
|-----------------|-----------------|-----------------|----------------|
| MD ₂ | MD ₁ | MD ₀ | |
| 0 | 0 | 0 | — |
| | | 1 | Mode 1 |
| | 1 | 0 | Mode 2 |
| | | 1 | Mode 3 |
| 1 | 0 | 0 | Mode 4 |
| | | 1 | Mode 5 |
| | 1 | 0 | Mode 6 |
| | | 1 | Mode 7 |

Note: * Determined by the state of the mode pins (MD₂ to MD₀).

SYSCR—System Control Register

H'EE012

System control



BRCR—Bus Release Control Register**H'EE013****Bus controller**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|
| | A23E | A22E | A21E | A20E | — | — | — | BRLE |
| Modes 1 and 2 | Initial value: 1 | Initial value: 1 | Initial value: 1 | Initial value: 1 | Initial value: 1 | Initial value: 1 | Initial value: 1 | Initial value: 0 |
| | Read/Write: — | Read/Write: — | Read/Write: — | Read/Write: — | Read/Write: — | Read/Write: — | Read/Write: — | Read/Write: R/W |
| Modes 3 and 4 | Initial value: 1 | Initial value: 1 | Initial value: 1 | Initial value: 0 | Initial value: 1 | Initial value: 1 | Initial value: 1 | Initial value: 0 |
| | Read/Write: R/W | Read/Write: R/W | Read/Write: R/W | Read/Write: — | Read/Write: — | Read/Write: — | Read/Write: — | Read/Write: R/W |

Address 23 to 20 enable

| | |
|---|--------------------|
| 0 | Address output |
| 1 | Other input/output |

Bus release enable

| | |
|---|--|
| 0 | The bus cannot be released to an external device |
| 1 | The bus can be released to an external device |

ISCR—IRQ Sense Control Register**H'EE014****Interrupt Controller**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|-----|-----|--------|--------|--------|--------|--------|--------|
| | — | — | IRQ5SC | IRQ4SC | IRQ3SC | IRQ2SC | IRQ1SC | IRQ0SC |
| Initial value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

IRQ5 to IRQ0 sense control

| | |
|---|--|
| 0 | Interrupts are requested when $\overline{\text{IRQ}}_5$ to $\overline{\text{IRQ}}_0$ are low |
| 1 | Interrupts are requested by falling-edge input at $\overline{\text{IRQ}}_5$ to $\overline{\text{IRQ}}_0$ |

IER—IRQ Enable Register**H'EE015****Interrupt Controller**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|-----|-----|-------|-------|-------|-------|-------|-------|
| | — | — | IRQ5E | IRQ4E | IRQ3E | IRQ2E | IRQ1E | IRQ0E |
| Initial value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

IRQ₅ to IRQ₀ enable

| | |
|---|--|
| 0 | IRQ ₅ to IRQ ₀ interrupts are disabled |
| 1 | IRQ ₅ to IRQ ₀ interrupts are enabled |

ISR—IRQ Status Register**H'EE016****Interrupt Controller**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|---|---|--------|--------|--------|--------|--------|--------|
| | — | — | IRQ5F | IRQ4F | IRQ3F | IRQ2F | IRQ1F | IRQ0F |
| Initial value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Read/Write | — | — | R/(W)* | R/(W)* | R/(W)* | R/(W)* | R/(W)* | R/(W)* |

IRQ₅ to IRQ₀ flags

| Bits 5 to 0 | Setting and Clearing Conditions |
|----------------|---|
| IRQ5F to IRQ0F | |
| 0 | [Clearing conditions] <ul style="list-style-type: none"> Read IRQ_nF when IRQ_nF = 1, then write 0 in IRQ_nF. IRQ_nSC = 0, $\overline{\text{IRQ}}_n$ input is high, and interrupt exception handling is being carried out. IRQ_nSC = 1 and IRQ_n interrupt exception handling is being carried out. |
| 1 | [Setting conditions] <ul style="list-style-type: none"> IRQ_nSC = 0 and $\overline{\text{IRQ}}_n$ input is low. IRQ_nSC = 1 and $\overline{\text{IRQ}}_n$ input changes from high to low. |

Note: n = 5 to 0

Note: * Only 0 can be written to clear the flag.

IPRA—Interrupt Priority Register A

H'EE018

Interrupt Controller

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|-------|-------|-------|-------|-------|-------|-------|-------|
| | IPRA7 | IPRA6 | IPRA5 | IPRA4 | IPRA3 | IPRA2 | IPRA1 | IPRA0 |
| Initial value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Priority level A7 to A0

| | |
|---|----------------------------------|
| 0 | Priority level 0 (low priority) |
| 1 | Priority level 1 (high priority) |

- Interrupt sources controlled by each bit

| IPRA | Bit | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|------|------------------|-------|-------|---------------|---------------|-----------------------|---------------------------|---------------------------|---------------------------|
| | | IPRA7 | IPRA6 | IPRA5 | IPRA4 | IPRA3 | IPRA2 | IPRA1 | IPRA0 |
| | Interrupt source | IRQ0 | IRQ1 | IRQ2, IRQ3 | IRQ4, IRQ5 | WDT, A/D converter | 16-bit timer channel 0 | 16-bit timer channel 1 | 16-bit timer channel 2 |

IPRB—Interrupt Priority Register B

H'EE019

Interrupt Controller

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|-------|-------|-----|-----|-------|-------|-----|-----|
| | IPRB7 | IPRB6 | — | — | IPRB3 | IPRB2 | — | — |
| Initial value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Priority level B7, B6, B3, and B2

| | |
|---|----------------------------------|
| 0 | Priority level 0 (low priority) |
| 1 | Priority level 1 (high priority) |

- Interrupt sources controlled by each bit

| IPRB | Bit | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|------|------------------|------------------------------------|------------------------------------|-------|-------|------------------|------------------|-------|-------|
| | | IPRB7 | IPRB6 | — | — | IPRB3 | IPRB2 | — | — |
| | Interrupt source | 8-bit timer channels 0 and 1 | 8-bit timer channels 2 and 3 | — | — | SCI channel 0 | SCI channel 1 | — | — |

DASTCR—D/A Standby Control Register**H'EE01A****D/A**

| | | | | | | | | |
|---------------|---|---|---|---|---|---|---|-------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | DASTE |
| Initial value | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| Read/Write | — | — | — | — | — | — | — | R/W |

D/A standby enable

| | | |
|---|---|-----------------|
| 0 | D/A output is disabled in software standby mode | (Initial value) |
| 1 | D/A output is enabled in software standby mode | |

DIVCR—Division Control Register**H'EE01B****System control**

| | | | | | | | | |
|---------------|---|---|---|---|---|---|------|------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | DIV1 | DIV0 |
| Initial value | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |
| Read/Write | — | — | — | — | — | — | R/W | R/W |

Division ratio bits 1 and 0

| Bit 1 | Bit 0 | Frequency Division Ratio |
|-------|-------|--------------------------|
| DIV1 | DIV0 | |
| 0 | 0 | 1/1 (Initial value) |
| | 1 | 1/2 |
| 1 | 0 | 1/4 |
| | 1 | 1/8 |

MSTCRH—Module Standby Control Register H**H'EE01C****System control**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|-------|---|---|---|---|-----|--------|--------|
| | PSTOP | — | — | — | — | — | MSTPH1 | MSTPH0 |
| Initial value | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| Read/Write | R/W | — | — | — | — | R/W | R/W | R/W |

Reserved bits

Module standby H1 to H0

Selection bits for placing modules in standby state.

 ϕ clock stopEnables or disables ϕ clock output.**MSTCRL—Module Standby Control Register L****H'EE01D****System control**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|-----|-----|-----|--------|--------|--------|-----|--------|
| | — | — | — | MSTPL4 | MSTPL3 | MSTPL2 | — | MSTPL0 |
| Initial value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Reserved bits

Module standby L4 to L2, L0

Selection bits for placing modules in standby state.

ADRCR—Address Control Register**H'EE01E****Bus controller**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|---|---|---|---|---|---|---|--------|
| | — | — | — | — | — | — | — | ADRCTL |
| Initial value | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Read/Write | — | — | — | — | — | — | — | R/W |

Reserved bits

Address control
Selects address update mode 1 or address update mode 2.

| ADRCTL | Description |
|--------|---|
| 0 | Address update mode 2 is selected |
| 1 | Address update mode 1 is selected (Initial value) |

CSCR—Chip Select Control Register**H'EE01F****Bus controller**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|------|------|------|------|---|---|---|---|
| | CS7E | CS6E | CS5E | CS4E | — | — | — | — |
| Initial value | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| Read/Write | R/W | R/W | R/W | R/W | — | — | — | — |

Chip select 7 to 4 enable

| Bit n | Description |
|-------|---|
| CSnE | |
| 0 | Output of chip select signal \overline{CSn} is disabled (Initial value) |
| 1 | Output of chip select signal \overline{CSn} is enabled |

Note: n = 7 to 4

ABWCR—Bus Width Control Register

H'EE020

Bus controller

| Bit | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|---------------|------|------|------|------|------|------|------|------|
| | | ABW7 | ABW6 | ABW5 | ABW4 | ABW3 | ABW2 | ABW1 | ABW0 |
| Modes 1 and 3 | Initial value | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Modes 2 and 4 | Initial value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Area 7 to 0 bus width control

| | |
|--------------|--------------------------------------|
| Bits 7 to 0 | Bus Width of Access Area |
| ABW7 to ABW0 | |
| 0 | Areas 7 to 0 are 16-bit access areas |
| 1 | Areas 7 to 0 are 8-bit access areas |

ASTCR—Access State Control Register

H'EE021

Bus controller

| Bit | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|--|------|------|------|------|------|------|------|------|
| | | AST7 | AST6 | AST5 | AST4 | AST3 | AST2 | AST1 | AST0 |
| Initial value | | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Read/Write | | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Area 7 to 0 access state control

| | |
|--------------|---|
| Bits 7 to 0 | Number of States in Access Area |
| AST7 to AST0 | |
| 0 | Areas 7 to 0 are two-state access areas |
| 1 | Areas 7 to 0 are three-state access areas |

WCRH—Wait Control Register H

H'EE022

Bus controller

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|-----|-----|-----|-----|-----|-----|-----|-----|
| | W71 | W70 | W61 | W60 | W51 | W50 | W41 | W40 |
| Initial value | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Area 4 wait control 1 and 0

| | | |
|---|---|------------------------------------|
| 0 | 0 | No program wait is inserted |
| | 1 | 1 program wait state is inserted |
| 1 | 0 | 2 program wait states are inserted |
| | 1 | 3 program wait states are inserted |

Area 5 wait control 1 and 0

| | | |
|---|---|------------------------------------|
| 0 | 0 | No program wait is inserted |
| | 1 | 1 program wait state is inserted |
| 1 | 0 | 2 program wait states are inserted |
| | 1 | 3 program wait states are inserted |

Area 6 wait control 1 and 0

| | | |
|---|---|------------------------------------|
| 0 | 0 | No program wait is inserted |
| | 1 | 1 program wait state is inserted |
| 1 | 0 | 2 program wait states are inserted |
| | 1 | 3 program wait states are inserted |

Area 7 wait control 1 and 0

| | | |
|---|---|------------------------------------|
| 0 | 0 | No program wait is inserted |
| | 1 | 1 program wait state is inserted |
| 1 | 0 | 2 program wait states are inserted |
| | 1 | 3 program wait states are inserted |

WCRL—Wait Control Register L

H'EE023

Bus controller

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|-----|-----|-----|-----|-----|-----|-----|-----|
| | W31 | W30 | W21 | W20 | W11 | W10 | W01 | W00 |
| Initial value | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

| | | |
|---|---|------------------------------------|
| 0 | 0 | No program wait is inserted |
| | 1 | 1 program wait state is inserted |
| 1 | 0 | 2 program wait states are inserted |
| | 1 | 3 program wait states are inserted |

| | | |
|---|---|------------------------------------|
| 0 | 0 | No program wait is inserted |
| | 1 | 1 program wait state is inserted |
| 1 | 0 | 2 program wait states are inserted |
| | 1 | 3 program wait states are inserted |

| | | |
|---|---|------------------------------------|
| 0 | 0 | No program wait is inserted |
| | 1 | 1 program wait state is inserted |
| 1 | 0 | 2 program wait states are inserted |
| | 1 | 3 program wait states are inserted |

| | | |
|---|---|------------------------------------|
| 0 | 0 | No program wait is inserted |
| | 1 | 1 program wait state is inserted |
| 1 | 0 | 2 program wait states are inserted |
| | 1 | 3 program wait states are inserted |

BCR—Bus Control Register

H'EE024

Bus controller

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|-------|-------|-----|-----|-----|-----|------|-------|
| | ICIS1 | ICIS0 | — | — | — | — | RDEA | WAITE |
| Initial value | 1 | 1 | 0*1 | 0*1 | 0*1 | 1*2 | 1 | 0 |
| Read/Write | R/W | R/W | — | — | — | — | R/W | R/W |

| Wait pin enable | |
|-----------------|---------------------------------|
| 0 | WAIT pin wait input is disabled |
| 1 | WAIT pin wait input is enabled |

| Area division unit select | |
|---------------------------|---|
| 0 | Area divisions are as follows: Area 0: 2 Mbytes Area 4: 1.93 Mbytes Area 1: 2 Mbytes Area 5: 4 kbytes Area 2: 8 Mbytes Area 6: 23.75 kbytes Area 3: 2 Mbytes Area 7: 22 bytes |
| 1 | Areas 0 to 7 are the same size (2 Mbytes) |

| Idle cycle insertion 0 | |
|------------------------|---|
| 0 | No idle cycle is inserted in case of consecutive external read and write cycles |
| 1 | Idle cycle is inserted in case of consecutive external read and write cycles |

| Idle cycle insertion 1 | |
|------------------------|---|
| 0 | No idle cycle is inserted in case of consecutive external read cycles for different areas |
| 1 | Idle cycle is inserted in case of consecutive external read cycles for different areas |

Notes: 1. These bits can be read and written, but must not be set to 1. Normal operation cannot be guaranteed if 1 is written in these bits.

2. 0 must not be written in bit 2.

P4PCR—Port 4 Input Pull-Up MOS Control Register H'EE03E**Port 4**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|--------|--------|--------|--------|--------|--------|--------|--------|
| | P47PCR | P46PCR | P45PCR | P44PCR | P43PCR | P42PCR | P41PCR | P40PCR |
| Initial value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Port 4 input pull-up MOS control 7 to 0

| | |
|---|---------------------------------|
| 0 | Input pull-up transistor is off |
| 1 | Input pull-up transistor is on |

Note: Valid when the corresponding P4DDR bit is cleared to 0 (designating generic input).

TSTR—Timer Start Register

H'FFF60 16-bit timer (all channels)

| | | | | | | | | |
|---------------|---|---|---|---|---|------|------|------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | STR2 | STR1 | STR0 |
| Initial value | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| Read/Write | — | — | — | — | — | R/W | R/W | R/W |

Reserved bits

Counter start 0

| | | |
|---|---------------------|-----------------|
| 0 | 16TCNT0 is halted | (Initial value) |
| 1 | 16TCNT0 is counting | |

Counter start 1

| | | |
|---|---------------------|-----------------|
| 0 | 16TCNT1 is halted | (Initial value) |
| 1 | 16TCNT1 is counting | |

Counter start 2

| | | |
|---|---------------------|-----------------|
| 0 | 16TCNT2 is halted | (Initial value) |
| 1 | 16TCNT2 is counting | |

TSNC—Timer Synchro Register

H'FFF61 16-bit timer (all channels)

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|---|---|---|---|---|-------|-------|-------|
| | — | — | — | — | — | SYNC2 | SYNC1 | SYNC0 |
| Initial value | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| Read/Write | — | — | — | — | — | R/W | R/W | R/W |

Reserved bits

Timer sync 0

| | |
|---|---|
| 0 | Channel 0 timer counter (16TCNT0) operates independently (16TCNT0 presetting/clearing is independent of other channels) (Initial value) |
| 1 | Channel 0 operates synchronously Synchronous presetting/synchronous clearing of 16TCNT0 is possible |

Timer sync 1

| | |
|---|---|
| 0 | Channel 1 timer counter (16TCNT1) operates independently (16TCNT1 presetting/clearing is independent of other channels) (Initial value) |
| 1 | Channel 1 operates synchronously Synchronous presetting/synchronous clearing of 16TCNT1 is possible |

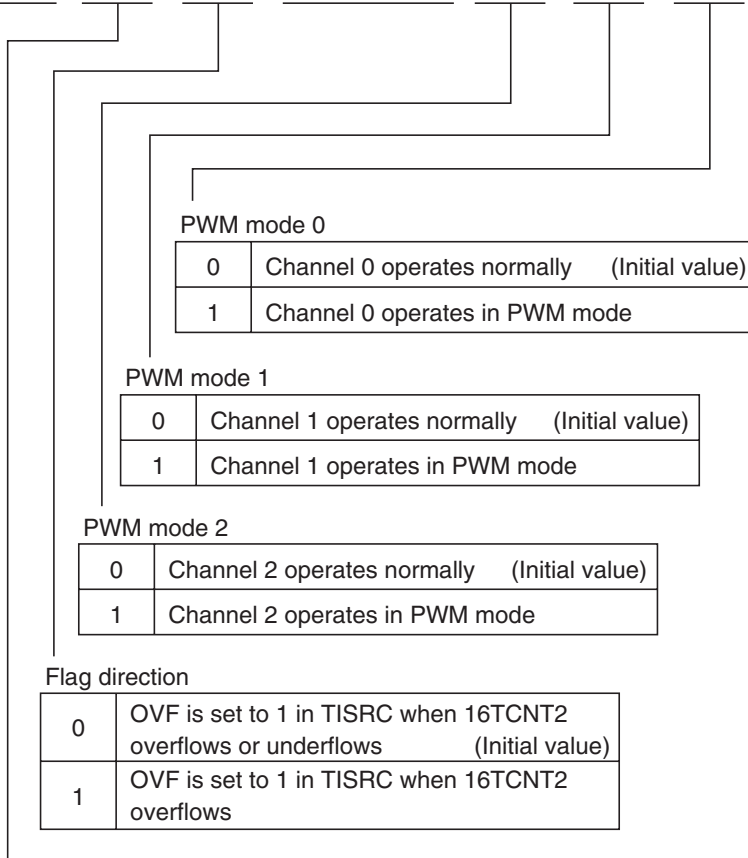
Timer sync 2

| | |
|---|---|
| 0 | Channel 2 timer counter (16TCNT2) operates independently (16TCNT2 presetting/clearing is independent of other channels) (Initial value) |
| 1 | Channel 2 operates synchronously Synchronous presetting/synchronous clearing of 16TCNT2 is possible |

TMDR—Timer Mode Register

H'FFF62 16-bit timer (all channels)

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|---|-----|------|---|---|------|------|------|
| | — | MDF | FDIR | — | — | PWM2 | PWM1 | PWM0 |
| Initial value | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| Read/Write | — | R/W | R/W | — | — | R/W | R/W | R/W |



PWM mode 0

| | |
|---|---|
| 0 | Channel 0 operates normally (Initial value) |
| 1 | Channel 0 operates in PWM mode |

PWM mode 1

| | |
|---|---|
| 0 | Channel 1 operates normally (Initial value) |
| 1 | Channel 1 operates in PWM mode |

PWM mode 2

| | |
|---|---|
| 0 | Channel 2 operates normally (Initial value) |
| 1 | Channel 2 operates in PWM mode |

Flag direction

| | |
|---|---|
| 0 | OVF is set to 1 in TISRC when 16TCNT2 overflows or underflows (Initial value) |
| 1 | OVF is set to 1 in TISRC when 16TCNT2 overflows |

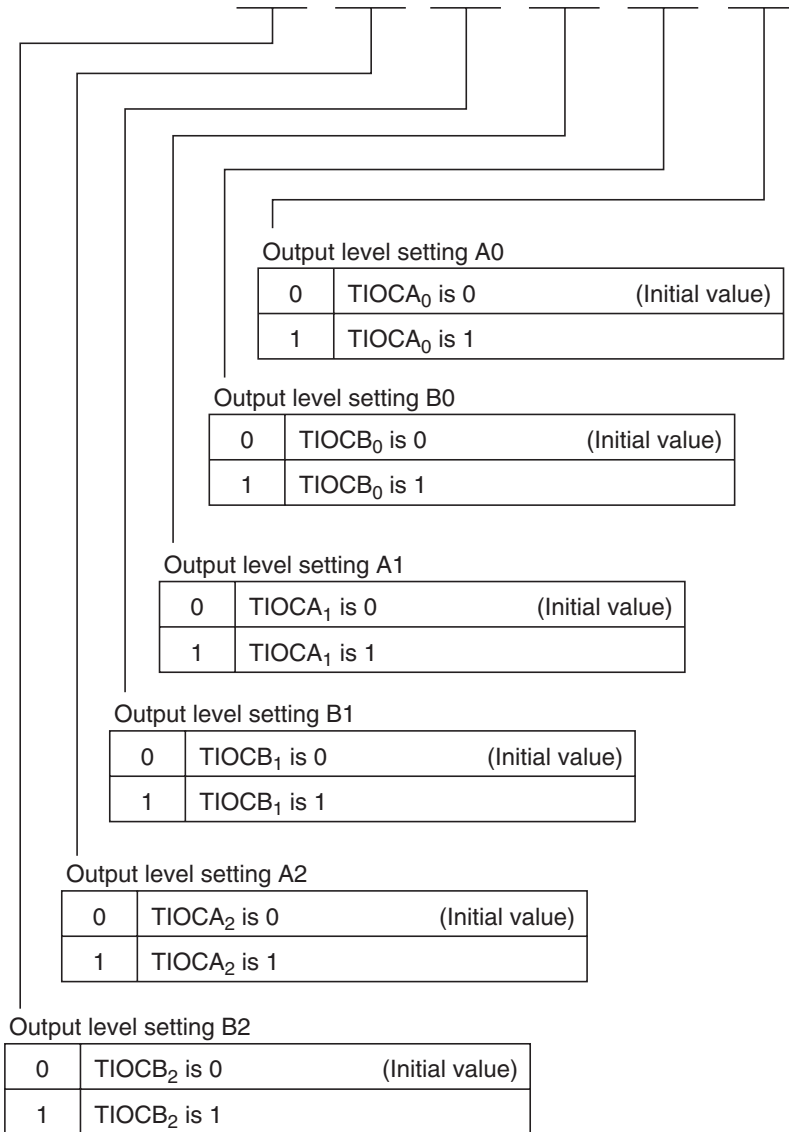
Phase counting mode

| | |
|---|---|
| 0 | Channel 2 operates normally (Initial value) |
| 1 | Channel 2 operates in phase counting mode |

TOLR—Timer Output Level Setting Register

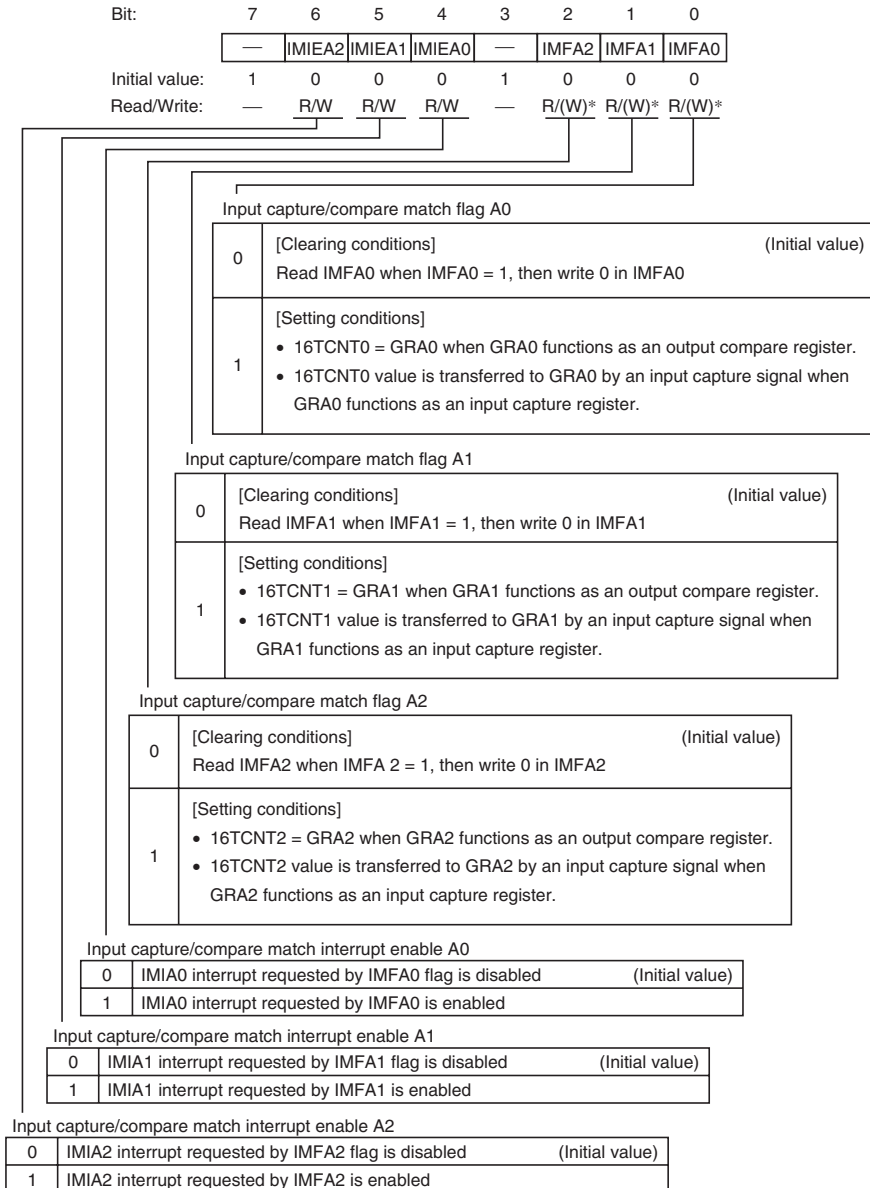
H'FFF63 16-bit timer (all channels)

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|---|---|------|------|------|------|------|------|
| | — | — | TOB2 | TOA2 | TOB1 | TOA1 | TOB0 | TOA0 |
| Initial value | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| Read/Write | — | — | W | W | W | W | W | W |



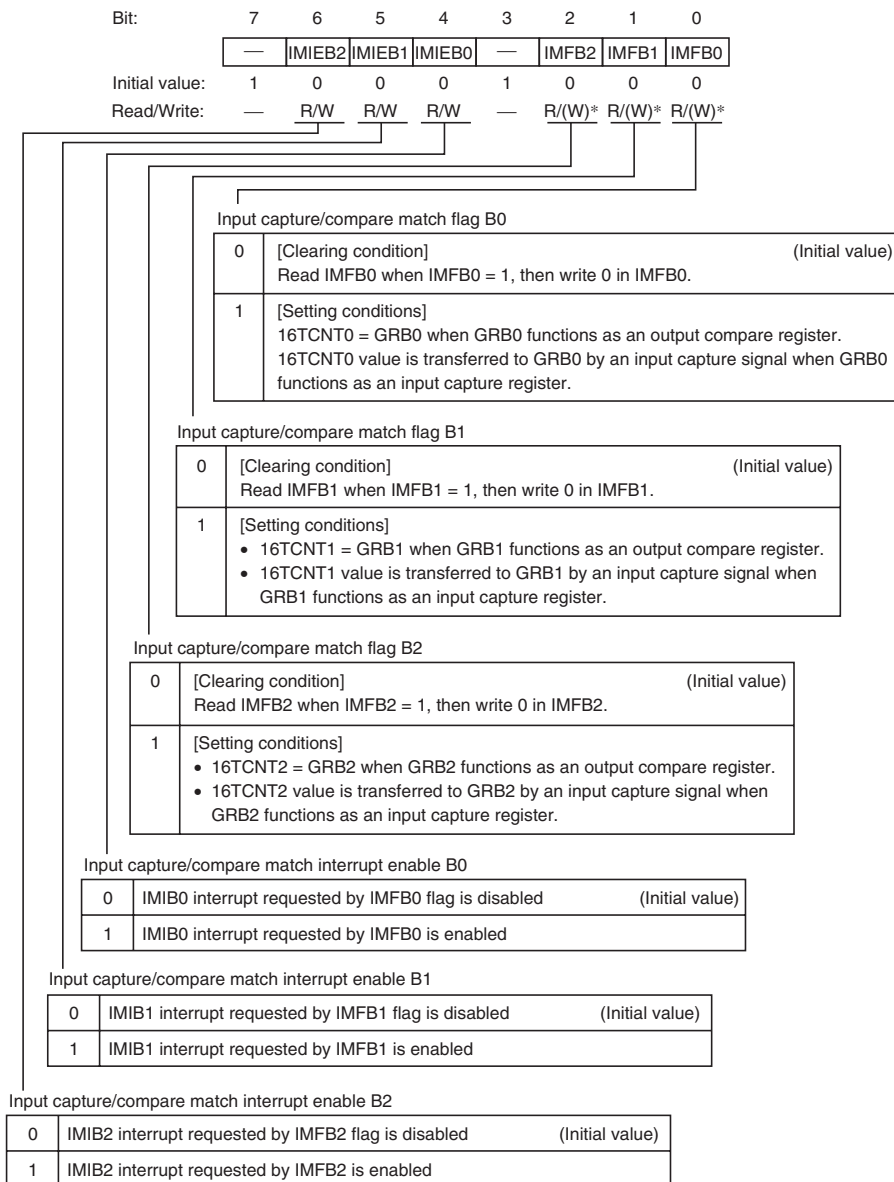
TISRA—Timer Interrupt Status Register A

H'FFF64 16-bit timer (all channels)



TISRB—Timer Interrupt Status Register B

H'FFF65 16-bit timer (all channels)



Note: * Only 0 can be written to clear the flag.

TISRC—Timer Interrupt Status Register C

H'FFF66 16-bit timer (all channels)

| | | | | | | | | |
|----------------|---|-------|-------|-------|---|--------|--------|--------|
| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | OVIE2 | OVIE1 | OVIE0 | — | OVF2 | OVF1 | OVF0 |
| Initial value: | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| Read/Write: | — | R/W | R/W | R/W | — | R/(W)* | R/(W)* | R/(W)* |

Overflow flag 0

| | | |
|---|--|-----------------|
| 0 | [Clearing condition] Read OVF0 when OVF0 = 1, then write 0 in OVF0. | (Initial value) |
| 1 | [Setting condition] 16TCNT0 overflowed from H'FFFF to H'0000. | |

Overflow flag 1

| | | |
|---|--|-----------------|
| 0 | [Clearing condition] Read OVF1 when OVF1 = 1, then write 0 in OVF1. | (Initial value) |
| 1 | [Setting condition] 16TCNT1 overflowed from H'FFFF to H'0000. | |

Overflow flag 2

| | | |
|---|--|-----------------|
| 0 | [Clearing condition] Read OVF2 when OVF2 = 1, then write 0 in OVF2. | (Initial value) |
| 1 | [Setting condition] 16TCNT2 overflowed from H'FFFF to H'0000, or underflowed from H'0000 to H'FFFF. | |

Overflow interrupt enable 0

| | | |
|---|---|-----------------|
| 0 | OVI0 interrupt requested by OVF0 flag is disabled | (Initial value) |
| 1 | OVI0 interrupt requested by OVF0 flag is enabled | |

Overflow interrupt enable 1

| | | |
|---|---|-----------------|
| 0 | OVI1 interrupt requested by OVF1 flag is disabled | (Initial value) |
| 1 | OVI1 interrupt requested by OVF1 flag is enabled | |

Overflow interrupt enable 2

| | | |
|---|---|-----------------|
| 0 | OVI2 interrupt requested by OVF2 flag is disabled | (Initial value) |
| 1 | OVI2 interrupt requested by OVF2 flag is enabled | |

Note: * Only 0 can be written to clear the flag.

16TCR0—Timer Control Register 0

H'FFF68

16-bit timer channel 0

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|---|-------|-------|-------|-------|-------|-------|-------|
| | — | CCLR1 | CCLR0 | CKEG1 | CKEG0 | TPSC2 | TPSC1 | TPSC0 |
| Initial value | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Read/Write | — | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Timer prescaler 2 to 0

| Bit 2 | Bit 1 | Bit 0 | Description |
|-------|-------|-------|---|
| TPSC2 | TPSC1 | TPSC0 | |
| 0 | 0 | 0 | Internal clock : ϕ (Initial value) |
| | | 1 | Internal clock : $\phi / 2$ |
| | 1 | 0 | Internal clock : $\phi / 4$ |
| | | 1 | Internal clock : $\phi / 8$ |
| 1 | 0 | 0 | External clock A : TCLKA input |
| | | 1 | External clock B : TCLKB input |
| | 1 | 0 | External clock C : TCLKC input |
| | | 1 | External clock D : TCLKD input |

Clock edge 1 and 0

| Bit 4 | Bit 3 | Description |
|-------|-------|--------------------------------------|
| CKEG1 | CKEG0 | |
| 0 | 0 | Rising edges counted (Initial value) |
| 0 | 1 | Falling edges counted |
| 1 | — | Both edges counted |

Counter clear 1 and 0

| Bit 6 | Bit 5 | Description |
|-------|-------|---|
| CCLR1 | CCLR0 | |
| 0 | 0 | 16TCNT is not cleared (Initial value) |
| | 1 | 16TCNT is cleared by GRA compare match or input capture |
| 1 | 0 | 16TCNT is cleared by GRB compare match or input capture |
| | 1 | Synchronous clear : 16TCNT is cleared in synchronization with other synchronized timers |

TIOR0—Timer I/O Control Register 0

H'FFF69

16-bit timer channel 0

| | | | | | | | | |
|----------------|---|------|------|------|---|------|------|------|
| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | IOB2 | IOB1 | IOB0 | — | IOA2 | IOA1 | IOA0 |
| Initial value: | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| Read/Write: | — | R/W | R/W | R/W | — | R/W | R/W | R/W |

I / O control A2 to A0

| Bit 2 | Bit 1 | Bit 0 | Description | |
|-------|-------|-------|-----------------------------------|---|
| IOA2 | IOA1 | IOA0 | | |
| 0 | 0 | 0 | GRA is an output compare register | No output at compare match (Initial value) |
| | | 1 | | 0 output at GRA compare match |
| | 1 | 0 | | 1 output at GRA compare match |
| | | 1 | | Output toggles at GRA compare match (1 output on channel 2) |
| 1 | 0 | 0 | GRA is an input capture register | GRA captures rising edges of input |
| | | 1 | | GRA captures falling edges of input |
| | 1 | 0 | | GRB captures both edges of input |
| | | 1 | | |

I / O control B2 to B0

| Bit 6 | Bit 5 | Bit 4 | Description | |
|-------|-------|-------|-----------------------------------|---|
| IOB2 | IOB1 | IOB0 | | |
| 0 | 0 | 0 | GRB is an output compare register | No output at compare match (Initial value) |
| | | 1 | | 0 output at GRB compare match |
| | 1 | 0 | | 1 output at GRB compare match |
| | | 1 | | Output toggles at GRB compare match (1 output on channel 2) |
| 1 | 0 | 0 | GRB is an input capture register | GRB captures rising edges of input |
| | | 1 | | GRB captures falling edges of input |
| | 1 | 0 | | GRB captures both edges of input |
| | | 1 | | |

16TCNT0 H/L—Timer Counter 0 H/L **H'FFF6A, H'FFF6B** **16-bit timer channel 0**

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| | | | | | | | | | | | | | | | | |
| Initial value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

|
Up-counter

GRA0 H/L—General Register A0 H/L **H'FFF6C, H'FFF6D** **16-bit timer channel 0**

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| | | | | | | | | | | | | | | | | |
| Initial value | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

|
Output compare or input capture register

GRB0 H/L—General Register B0 H/L **H'FFF6E, H'FFF6F** **16-bit timer channel 0**

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| | | | | | | | | | | | | | | | | |
| Initial value | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

|
Output compare or input capture register

16TCR1 Timer Control Register 1**H'FFF70****16-bit timer channel 1**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|---|-------|-------|-------|-------|-------|-------|-------|
| | — | CCLR1 | CCLR0 | CKEG1 | CKEG0 | TPSC2 | TPSC1 | TPSC0 |
| Initial value | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Read/Write | — | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Note: Bit functions are the same as for 16-bit timer channel 0.

TIOR1—Timer I/O Control Register 1**H'FFF71****16-bit timer channel 1**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|---|------|------|------|---|------|------|------|
| | — | IOB2 | IOB1 | IOB0 | — | IOA2 | IOA1 | IOA0 |
| Initial value | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| Read/Write | — | R/W | R/W | R/W | — | R/W | R/W | R/W |

Note: Bit functions are the same as for 16-bit timer channel 0.

16TCNT1 H/L—Timer Counter 1 H/L**H'FFF72, H'FFF73****16-bit timer channel 1**

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| | | | | | | | | | | | | | | | | |
| Initial value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Note: Bit functions are the same as for 16-bit timer channel 0.

GRA1 H/L—General Register A1 H/L **H'FFF74, H'FFF75** **16-bit timer channel 1**

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| | | | | | | | | | | | | | | | | |
| Initial value | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Note: Bit functions are the same as for 16-bit timer channel 0.

GRB1 H/L—General Register B1 H/L **H'FFF76, H'FFF77** **16-bit timer channel 1**

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| | | | | | | | | | | | | | | | | |
| Initial value | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Note: Bit functions are the same as for 16-bit timer channel 0.

16TCR2—Timer Control Register 2 **H'FFF78** **16-bit timer channel 2**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|---|-------|-------|-------|-------|-------|-------|-------|
| | — | CCLR1 | CCLR0 | CKEG1 | CKEG0 | TPSC2 | TPSC1 | TPSC0 |
| Initial value | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Read/Write | — | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

- Notes:
1. Bit functions are the same as for 16-bit timer channel 0.
 2. When phase counting mode is selected in channel 2, the settings of bits CKEG1 and CKEG0 and TPSC2 to TPSC0 in 16TCR2 are ignored.

TIOR2—Timer I/O Control Register 2**H'FFF79****16-bit timer channel 2**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|---|------|------|------|---|------|------|------|
| | — | IOB2 | IOB1 | IOB0 | — | IOA2 | IOA1 | IOA0 |
| Initial value | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| Read/Write | — | R/W | R/W | R/W | — | R/W | R/W | R/W |

Note: Bit functions are the same as for 16-bit timer channel 0.

16TCNT2 H/L—Timer Counter 2 H/L**H'FFF7A, H'FFF7B****16-bit timer channel 2**

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| | | | | | | | | | | | | | | | | |
| Initial value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Phase counting mode : up/down-counter
Other mode : up-counter

GRA2 H/L—General Register A2 H/L**H'FFF7C, H'FFF7D****16-bit timer channel 2**

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| | | | | | | | | | | | | | | | | |
| Initial value | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Note: Bit functions are the same as for 16-bit timer channel 0.

GRB2 H/L—General Register B2 H/L**H'FFF7E, H'FFF7F****16-bit timer channel 2**

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| | | | | | | | | | | | | | | | | |
| Initial value | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Note: Bit functions are the same as for 16-bit timer channel 0.

8TCR0—Timer Control Register 0

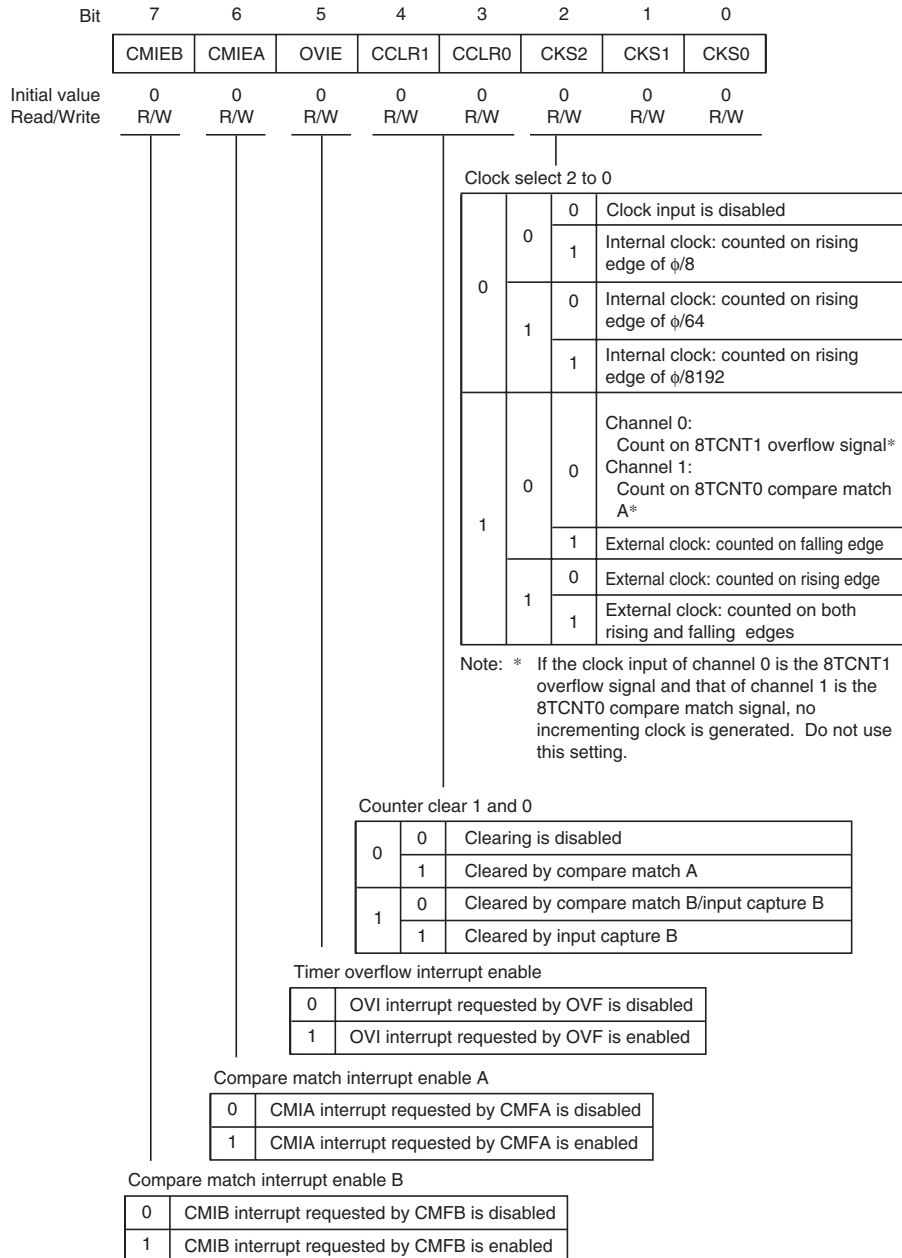
H'FFF80

8-bit timer channel 0

8TCR1—Timer Control Register 1

H'FFF81

8-bit timer channel 1



8TCSR0—Timer Control/Status Register 0

H'FFF82

8-bit timer channel 0

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|---------|---------|---------|------|------|------|-----|-----|
| | CMFB | CMFA | OVF | ADTE | OIS3 | OIS2 | OS1 | OS0 |
| Initial value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Read/Write | R/(W)*1 | R/(W)*1 | R/(W)*1 | R/W | R/W | R/W | R/W | R/W |

Output select A1 and A0

| Bit 1 | Bit 0 | Description |
|-------|-------|-----------------------------------|
| OS1 | OS0 | |
| 0 | 0 | No change at compare match A |
| | 1 | 0 output at compare match A |
| 1 | 0 | 1 output at compare match A |
| | 1 | Output toggles at compare match A |

Output/input capture edge select B3 and B2

| ICE in 8TCSR1 | Bit 3 OIS3 | Bit 2 OIS2 | Description |
|---------------|------------|------------|--|
| 0 | 0 | 0 | No change at compare match B |
| | | 1 | 0 output at compare match B |
| | 1 | 0 | 1 output at compare match B |
| | | 1 | Output toggles at compare match B |
| 1 | 0 | 0 | TCORB input capture on rising edge |
| | | 1 | TCORB input capture on falling edge |
| | 1 | 0 | TCORB input capture on both rising and falling edges |
| | | 1 | |

A/D trigger enable

| TRGE ⁻² | Bit 4 ADTE | Description |
|--------------------|------------|---|
| 0 | 0 | A/D converter start requests by compare match A or an external trigger are disabled |
| | 1 | A/D converter start requests by compare match A or an external trigger are disabled |
| 1 | 0 | A/D converter start requests by an external trigger are enabled, and A/D converter start requests by compare match A are disabled |
| | 1 | A/D converter start requests by compare match A are enabled, and A/D converter start requests by an external trigger are disabled |

Note: 2. TRGE is bit 7 of the A/D control register (ADCR).

Timer overflow flag

| | |
|---|---|
| 0 | [Clearing condition] Read OVF when OVF = 1, then write 0 in OVF. |
| 1 | [Setting condition] 8TCNT overflows from H'FF to H'00. |

Compare match flag A

| | |
|---|--|
| 0 | [Clearing condition] Read CMFA when CMFA = 1, then write 0 in CMFA. |
| 1 | [Setting condition] 8TCNT = TCORA |

Compare match/input capture flag B

| | |
|---|--|
| 0 | [Clearing condition] Read CMFB when CMFB = 1, then write 0 in CMFB. |
| 1 | [Setting conditions] <ul style="list-style-type: none"> • 8TCNT = TCORB • The 8TCNT value is transferred to TCORB by an input capture signal when TCORB functions as an input capture register. |

Note: 1. Only 0 can be written to bits 7 to 5 to clear these flags.

8TCSR1—Timer Control/Status Register 1

H'FFF83

8-bit timer channel 1

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|--------|--------|--------|-----|------|------|-----|-----|
| | CMFB | CMFA | OVF | ICE | OIS3 | OIS2 | OS1 | OS0 |
| Initial value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Read/Write | R/(W)* | R/(W)* | R/(W)* | R/W | R/W | R/W | R/W | R/W |

| Bit 1 | Bit 0 | Description |
|-------|-------|-----------------------------------|
| 0 | 0 | No change at compare match A |
| | 1 | 0 output at compare match A |
| 1 | 0 | 1 output at compare match A |
| | 1 | Output toggles at compare match A |

| ICE in 8TCSR1 | Bit 3 OIS3 | Bit 2 OIS2 | Description |
|---------------|------------|------------|--|
| 0 | 0 | 0 | No change at compare match B |
| | | 1 | 0 output at compare match B |
| | 1 | 0 | 1 output at compare match B |
| | | 1 | Output toggles at compare match B |
| 1 | 0 | 0 | TCORB input capture on rising edge |
| | | 1 | TCORB input capture on falling edge |
| | 1 | 0 | TCORB input capture on both rising and falling edges |
| | | 1 | TCORB input capture on both rising and falling edges |

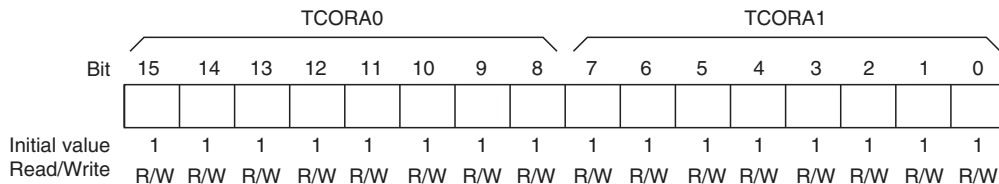
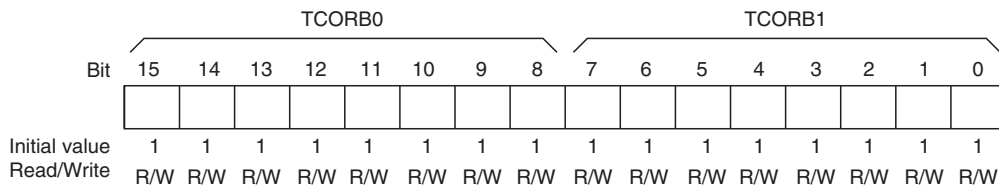
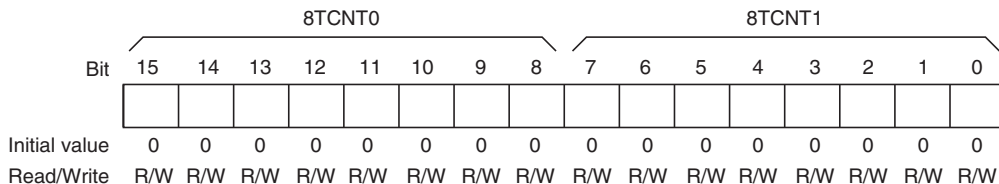
| | Description |
|---|------------------------------------|
| 0 | TCORB is a compare match register |
| 1 | TCORB is an input capture register |

| | Description |
|---|---|
| 0 | [Clearing condition] Read OVF when OVF = 1, then write 0 in OVF. |
| 1 | [Setting condition] 8TCNT overflows from H'FF to H'00. |

| | Description |
|---|--|
| 0 | [Clearing condition] Read CMFA when CMFA = 1, then write 0 in CMFA. |
| 1 | [Setting condition] 8TCNT = TCORA |

| | Description |
|---|---|
| 0 | [Clearing condition] Read CMFB when CMFB = 1, then write 0 in CMFB. |
| 1 | [Setting conditions] <ul style="list-style-type: none"> 8TCNT = TCORB The 8TCNT value is transferred to TCORB by an input capture signal when TCORB functions as an input capture register. |

Note: * Only 0 can be written to bits 7 to 5 to clear these flags.

TCORA0—Time Constant Register A0**H'FFF84****8-bit timer channel 0****TCORA1—Time Constant Register A1****H'FFF85****8-bit timer channel 1****TCORB0—Time Constant Register B0****H'FFF86****8-bit timer channel 0****TCORB1—Time Constant Register B1****H'FFF87****8-bit timer channel 1****8TCNT0—Timer Counter 0****H'FFF88****8-bit timer channel 0****8TCNT1—Timer Counter 1****H'FFF89****8-bit timer channel 1**

TCSR—Timer Control/Status Register

H'FFF8C

WDT

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|--------|----------------|-----|---|---|------|------|------|
| | OVF | WT/ $\bar{I}T$ | TME | — | — | CKS2 | CKS1 | CKS0 |
| Initial value | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| Read/Write | R/(W)* | R/W | R/W | — | — | R/W | R/W | R/W |

Clock select 2 to 0

| CKS2 | CKS1 | CKS0 | Description |
|------|------|------|-------------|
| 0 | 0 | 0 | $\phi/2$ |
| | | 1 | $\phi/32$ |
| | 1 | 0 | $\phi/64$ |
| | | 1 | $\phi/128$ |
| 1 | 0 | 0 | $\phi/256$ |
| | | 1 | $\phi/512$ |
| | 1 | 0 | $\phi/2048$ |
| | | 1 | $\phi/4096$ |

Timer enable

| | |
|---|---|
| 0 | Timer disabled: TCNT is initialized to H'00 and halted |
| 1 | Timer enabled: TCNT starts counting up |

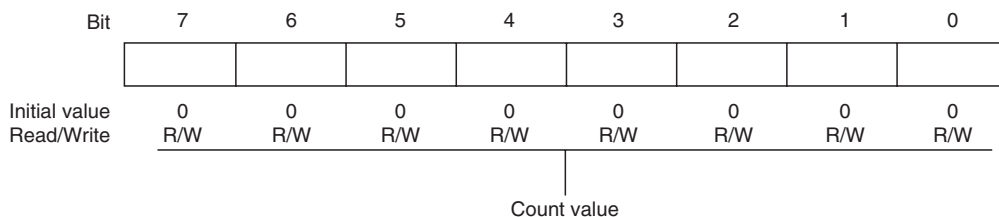
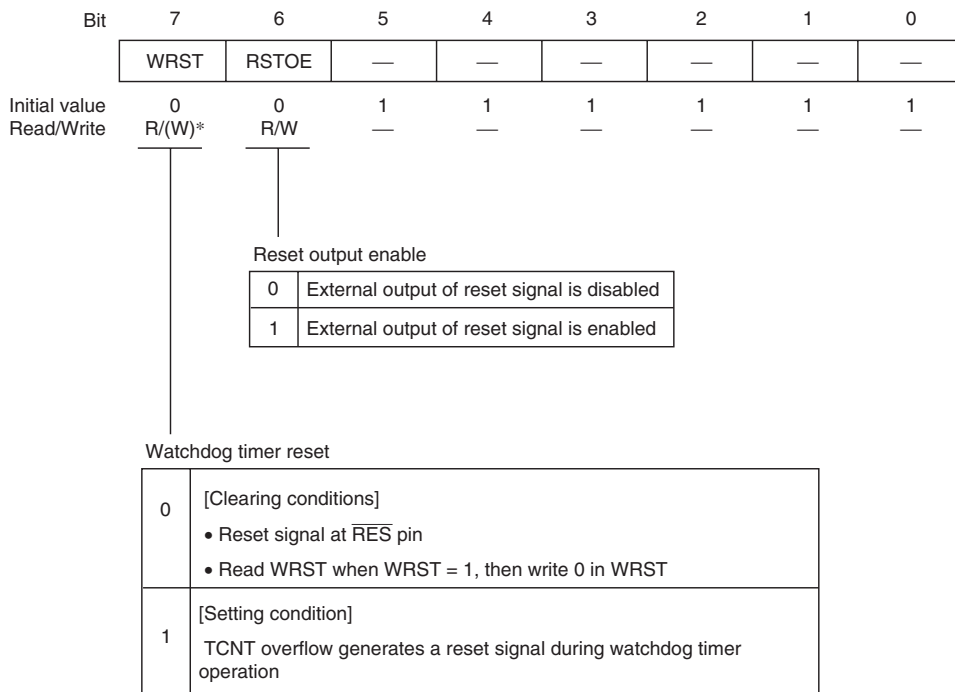
Timer mode select

| | |
|---|---|
| 0 | Interval timer: requests interval timer interrupts |
| 1 | Watchdog timer: generates a reset signal |

Overflow flag

| | |
|---|--|
| 0 | [Clearing condition] Read OVF when OVF = 1, then write 0 in OVF |
| 1 | [Setting condition] TCNT changes from H'FF to H'00 |

Note: * Only 0 can be written to clear the flag.

TCNT—Timer Counter**H'FFF8D (read), H'FFF8C (write)****WDT****RSTCSR—Reset Control/Status Register****H'FFF8F (read), H'FFF8E (write)****WDT**

Note: * Only 0 can be written in bit 7 to clear the flag.

8TCR2—Timer Control Register 2**H'FFF90****8-bit timer channel 2****8TCR3—Timer Control Register 3****H'FFF91****8-bit timer channel 3**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|-------|-------|------|-------|-------|------|------|------|
| | CMIEB | CMIEA | OVIE | CCLR1 | CCLR0 | CKS2 | CKS1 | CKS0 |
| Initial value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Clock select 2 to 0

| CKS2 | CKS1 | CKS0 | Description |
|------|------|------|--|
| 0 | 0 | 0 | Clock input is disabled |
| | | 1 | Internal clock: counted on rising edge of $\phi/8$ |
| | 1 | 0 | Internal clock: counted on rising edge of $\phi/64$ |
| | | 1 | Internal clock: counted on rising edge of $\phi/8192$ |
| 1 | 0 | 0 | Channel 2: Count on 8TCNT3 overflow signal* Channel 3: Count on 8TCNT2 compare match A* |
| | | 1 | External clock: counted on falling edge |
| | 1 | 0 | External clock: counted on rising edge |
| | | 1 | External clock: counted on both rising and falling edges |

Note: * If the clock input of channel 2 is the 8TCNT3 overflow signal and that of channel 3 is the 8TCNT2 compare match signal, no incrementing clock is generated. Do not use this setting.

Counter clear 1 and 0

| | | |
|---|---|--|
| 0 | 0 | Clearing is disabled |
| | 1 | Cleared by compare match A |
| 1 | 0 | Cleared by compare match B/input capture B |
| | 1 | Cleared by input capture B |

Timer overflow interrupt enable

| | |
|---|--|
| 0 | OVI interrupt requested by OVF is disabled |
| 1 | OVI interrupt requested by OVF is enabled |

Compare match interrupt enable A

| | |
|---|--|
| 0 | CMIA interrupt requested by CMFA is disabled |
| 1 | CMIA interrupt requested by CMFA is enabled |

Compare match interrupt enable B

| | |
|---|--|
| 0 | CMIB interrupt requested by CMFB is disabled |
| 1 | CMIB interrupt requested by CMFB is enabled |

8TCSR2—Timer Control/Status Register 2

H'FFF92

8-bit timer channel 2

8TCSR3—Timer Control/Status Register 3

H'FFF93

8-bit timer channel 3

| | | | | | | | | | |
|---------------|-----|--------|--------|--------|---|------|------|-----|-----|
| 8TCSR2 | Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | | CMFB | CMFA | OVF | — | OIS3 | OIS2 | OS1 | OS0 |
| Initial value | | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| Read/Write | | R/(W)* | R/(W)* | R/(W)* | — | R/W | R/W | R/W | R/W |

| | | | | | | | | | |
|---------------|-----|--------|--------|--------|-----|------|------|-----|-----|
| 8TCSR3 | Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | | CMFB | CMFA | OVF | ICE | OIS3 | OIS2 | OS1 | OS0 |
| Initial value | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Read/Write | | R/(W)* | R/(W)* | R/(W)* | R/W | R/W | R/W | R/W | R/W |

Output select A1 and A0

| Bit 1 OS1 | Bit 0 OS0 | Description |
|--------------|--------------|-----------------------------------|
| 0 | 0 | No change at compare match A |
| | 1 | 0 output at compare match A |
| 1 | 0 | 1 output at compare match A |
| | 1 | Output toggles at compare match A |

Output/input capture edge select B3 and B2

| ICE in 8TCSR3 | Bit 3 OIS3 | Bit 2 OIS2 | Description |
|------------------|---------------|---------------|--|
| 0 | 0 | 0 | No change at compare match B |
| | | 1 | 0 output at compare match B |
| | 1 | 0 | 1 output at compare match B |
| | | 1 | Output toggles at compare match B |
| 1 | 0 | 0 | TCORB input capture on rising edge |
| | | 1 | TCORB input capture on falling edge |
| | 1 | 0 | TCORB input capture on both rising and falling edges |

Input capture enable

| | |
|---|------------------------------------|
| 0 | TCORB is a compare match register |
| 1 | TCORB is an input capture register |

Timer overflow flag

| | |
|---|---|
| 0 | [Clearing condition] Read OVF when OVF = 1, then write 0 in OVF. |
| 1 | [Setting condition] 8TCNT overflows from H'FF to H'00. |

Compare match flag A

| | |
|---|--|
| 0 | [Clearing condition] Read CMFA when CMFA = 1, then write 0 in CMFA. |
| 1 | [Setting condition] 8TCNT = TCORA |

Compare match/input capture flag B

| | |
|---|--|
| 0 | [Clearing condition] Read CMFB when CMFB = 1, then write 0 in CMFB. |
| 1 | [Setting conditions] <ul style="list-style-type: none"> 8TCNT = TCORB The 8TCNT value is transferred to TCORB by an input capture signal when TCORB functions as an input capture register. |

Note: * Only 0 can be written to bits 7 to 5 to clear these flags.

TCORA2—Time Constant Register A2

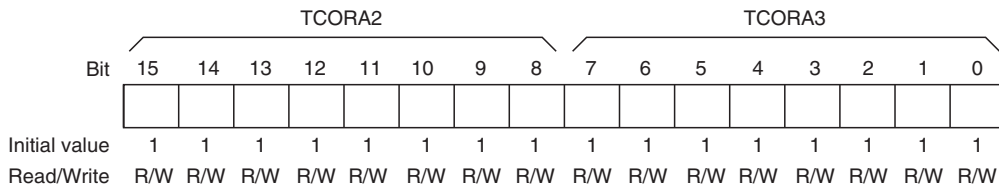
H'FFF94

8-bit timer channel 2

TCORA3—Time Constant Register A3

H'FFF95

8-bit timer channel 3



TCORB2—Time Constant Register B2

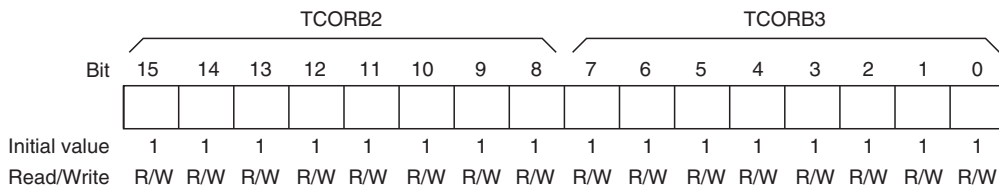
H'FFF96

8-bit timer channel 2

TCORB3—Time Constant Register B3

H'FFF97

8-bit timer channel 3



8TCNT2—Timer Counter 2

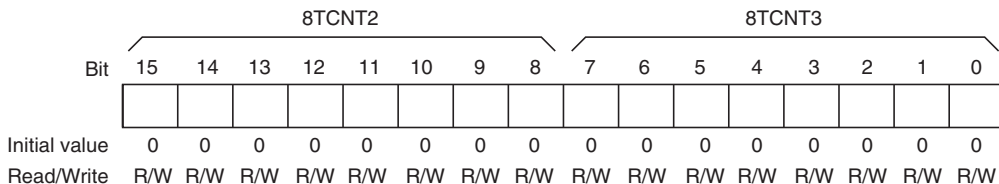
H'FFF98

8-bit timer channel 2

8TCNT3—Timer Counter 3

H'FFF99

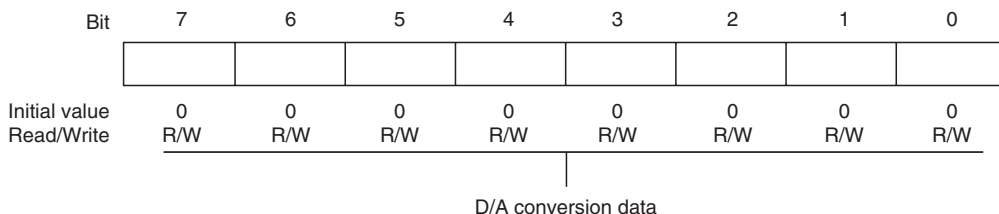
8-bit timer channel 3

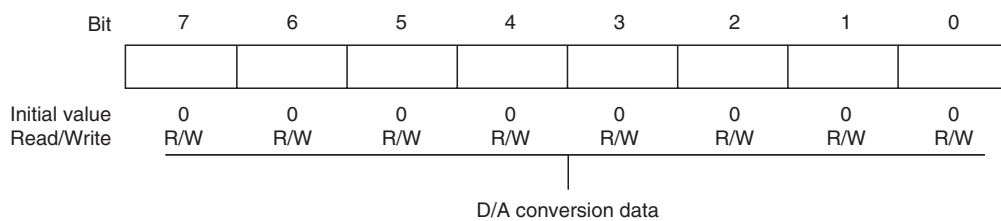


DADR0—D/A Data Register 0

H'FFF9C

D/A



DADR1—D/A Data Register 1**H'FFF9D****D/A**

DACR—D/A Control Register

H'FFF9E

D/A

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|-------|-------|-----|---|---|---|---|---|
| | DAOE1 | DAOE0 | DAE | — | — | — | — | — |
| Initial value | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |
| Read/Write | R/W | R/W | R/W | — | — | — | — | — |

D/A enable

| Bit 7 | Bit 6 | Bit 5 | Description |
|-------|-------|-------|---|
| DAOE1 | DAOE0 | DAE | |
| 0 | 0 | — | D/A conversion is disabled in channels 0 and 1 |
| 0 | 1 | 0 | D/A conversion is enabled in channel 0 D/A conversion is disabled in channel 1 |
| 0 | 1 | 1 | D/A conversion is enabled in channels 0 and 1 |
| 1 | 0 | 0 | D/A conversion is disabled in channel 0 D/A conversion is enabled in channel 1 |
| 1 | 0 | 1 | D/A conversion is enabled in channels 0 and 1 |
| 1 | 1 | — | D/A conversion is enabled in channels 0 and 1 |

D/A output enable 0

| | |
|---|--|
| 0 | DA ₀ analog output is disabled |
| 1 | Channel-0 D/A conversion and DA ₀ analog output are enabled |

D/A output enable 1

| | |
|---|--|
| 0 | DA ₁ analog output is disabled |
| 1 | Channel-1 D/A conversion and DA ₁ analog output are enabled |

TPMR—TPC Output Mode Register

H'FFFA0

TPC

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|---|---|---|---|-------|-------|-------|-------|
| | — | — | — | — | G3NOV | G2NOV | G1NOV | G0NOV |
| Initial value | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| Read/Write | — | — | — | — | R/W | R/W | R/W | R/W |

| Group 0 non-overlap | |
|---------------------|---|
| 0 | Normal TPC output in group 0. Output values change at compare match A in the selected 16-bit timer channel |
| 1 | Non-overlapping TPC output in group 0, controlled by compare match A and B in the selected 16-bit timer channel |

| Group 1 non-overlap | |
|---------------------|---|
| 0 | Normal TPC output in group 1. Output values change at compare match A in the selected 16-bit timer channel |
| 1 | Non-overlapping TPC output in group 1, controlled by compare match A and B in the selected 16-bit timer channel |

| Group 2 non-overlap | |
|---------------------|---|
| 0 | Normal TPC output in group 2. Output values change at compare match A in the selected 16-bit timer channel |
| 1 | Non-overlapping TPC output in group 2, controlled by compare match A and B in the selected 16-bit timer channel |

| Group 3 non-overlap | |
|---------------------|---|
| 0 | Normal TPC output in group 3. Output values change at compare match A in the selected 16-bit timer channel |
| 1 | Non-overlapping TPC output in group 3, controlled by compare match A and B in the selected 16-bit timer channel |

TPCR—TPC Output Control Register

H'FFFA1

TPC

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|--------|--------|--------|--------|--------|--------|--------|--------|
| | G3CMS1 | G3CMS0 | G2CMS1 | G2CMS0 | G1CMS1 | G1CMS0 | G0CMS1 | G0CMS0 |
| Initial value | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

| Group 0 compare match select 1 and 0 | | |
|--------------------------------------|--------|--|
| Bit 1 | Bit 0 | 16-Bit Timer Channel Selected as Output Trigger |
| G0CMS1 | G0CMS0 | |
| 0 | 0 | TPC output group 0 (TP ₃ to TP ₀) is triggered by compare match in 16-bit timer channel 0 |
| | 1 | TPC output group 0 (TP ₃ to TP ₀) is triggered by compare match in 16-bit timer channel 1 |
| 1 | 0 | TPC output group 0 (TP ₃ to TP ₀) is triggered by compare match in 16-bit timer channel 2 |
| | 1 | TPC output group 0 (TP ₃ to TP ₀) is triggered by compare match in 16-bit timer channel 2 |

| Group 1 compare match select 1 and 0 | | |
|--------------------------------------|--------|--|
| Bit 3 | Bit 2 | 16-Bit Timer Channel Selected as Output Trigger |
| G1CMS1 | G1CMS0 | |
| 0 | 0 | TPC output group 1 (TP ₇ to TP ₄) is triggered by compare match in 16-bit timer channel 0 |
| | 1 | TPC output group 1 (TP ₇ to TP ₄) is triggered by compare match in 16-bit timer channel 1 |
| 1 | 0 | TPC output group 1 (TP ₇ to TP ₄) is triggered by compare match in 16-bit timer channel 2 |
| | 1 | TPC output group 1 (TP ₇ to TP ₄) is triggered by compare match in 16-bit timer channel 2 |

| Group 2 compare match select 1 and 0 | | |
|--------------------------------------|--------|---|
| Bit 5 | Bit 4 | 16-Bit Timer Channel Selected as Output Trigger |
| G2CMS1 | G2CMS0 | |
| 0 | 0 | TPC output group 2 (TP ₁₁ to TP ₈) is triggered by compare match in 16-bit timer channel 0 |
| | 1 | TPC output group 2 (TP ₁₁ to TP ₈) is triggered by compare match in 16-bit timer channel 1 |
| 1 | 0 | TPC output group 2 (TP ₁₁ to TP ₈) is triggered by compare match in 16-bit timer channel 2 |
| | 1 | TPC output group 2 (TP ₁₁ to TP ₈) is triggered by compare match in 16-bit timer channel 2 |

| Group 3 compare match select 1 and 0 | | |
|--------------------------------------|--------|--|
| Bit 7 | Bit 6 | 16-Bit Timer Channel Selected as Output Trigger |
| G3CMS1 | G3CMS0 | |
| 0 | 0 | TPC output group 3 (TP ₁₅ to TP ₁₂) is triggered by compare match in 16-bit timer channel 0 |
| | 1 | TPC output group 3 (TP ₁₅ to TP ₁₂) is triggered by compare match in 16-bit timer channel 1 |
| 1 | 0 | TPC output group 3 (TP ₁₅ to TP ₁₂) is triggered by compare match in 16-bit timer channel 2 |
| | 1 | TPC output group 3 (TP ₁₅ to TP ₁₂) is triggered by compare match in 16-bit timer channel 2 |

NDERB—Next Data Enable Register B**H'FFFA2****TPC**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|--------|--------|--------|--------|--------|--------|-------|-------|
| | NDER15 | NDER14 | NDER13 | NDER12 | NDER11 | NDER10 | NDER9 | NDER8 |
| Initial value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Next data enable 15 to 8

| Bits 7 to 0 | Description |
|-----------------|--|
| NDER15 to NDER8 | |
| 0 | TPC outputs TP ₁₅ to TP ₈ are disabled (NDR15 to NDR8 are not transferred to PB ₇ to PB ₀) |
| 1 | TPC outputs TP ₁₅ to TP ₈ are enabled (NDR15 to NDR8 are transferred to PB ₇ to PB ₀) |

NDERA—Next Data Enable Register A**H'FFFA3****TPC**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|-------|-------|-------|-------|-------|-------|-------|-------|
| | NDER7 | NDER6 | NDER5 | NDER4 | NDER3 | NDER2 | NDER1 | NDER0 |
| Initial value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Next data enable 7 to 0

| Bits 7 to 0 | Description |
|----------------|--|
| NDER7 to NDER0 | |
| 0 | TPC outputs TP ₇ to TP ₀ are disabled (NDR7 to NDR0 are not transferred to PA ₇ to PA ₀) |
| 1 | TPC outputs TP ₇ to TP ₀ are enabled (NDR7 to NDR0 are transferred to PA ₇ to PA ₀) |

NDRB—Next Data Register B**H'FFFA4/H'FFFA6****TPC**

- Same trigger for TPC output groups 2 and 3

— Address H'FFFA4

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|-------|-------|-------|-------|-------|-------|------|------|
| | NDR15 | NDR14 | NDR13 | NDR12 | NDR11 | NDR10 | NDR9 | NDR8 |
| Initial value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Store the next output data for TPC output group 3 Store the next output data for TPC output group 2

— Address H'FFFA6

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — |
| Initial value | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Read/Write | — | — | — | — | — | — | — | — |

- Different triggers for TPC output groups 2 and 3

— Address H'FFFA4

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|-------|-------|-------|-------|---|---|---|---|
| | NDR15 | NDR14 | NDR13 | NDR12 | — | — | — | — |
| Initial value | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| Read/Write | R/W | R/W | R/W | R/W | — | — | — | — |

Store the next output data for TPC output group 3

— Address H'FFFA6

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|---|---|---|---|-------|-------|------|------|
| | — | — | — | — | NDR11 | NDR10 | NDR9 | NDR8 |
| Initial value | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| Read/Write | — | — | — | — | R/W | R/W | R/W | R/W |

Store the next output data for TPC output group 2

NDRA—Next Data Register A

H'FFFA5/H'FFFA7

TPC

- Same trigger for TPC output groups 0 and 1

— Address H'FFFA5

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|------|------|------|------|------|------|------|------|
| | NDR7 | NDR6 | NDR5 | NDR4 | NDR3 | NDR2 | NDR1 | NDR0 |
| Initial value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Store the next output data for TPC output group 1 Store the next output data for TPC output group 0

— Address H'FFFA7

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — |
| Initial value | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Read/Write | — | — | — | — | — | — | — | — |

- Different triggers for TPC output groups 0 and 1

— Address H'FFFA5

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|------|------|------|------|---|---|---|---|
| | NDR7 | NDR6 | NDR5 | NDR4 | — | — | — | — |
| Initial value | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| Read/Write | R/W | R/W | R/W | R/W | — | — | — | — |

Store the next output data for TPC output group 1

— Address H'FFFA7

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|---|---|---|---|------|------|------|------|
| | — | — | — | — | NDR3 | NDR2 | NDR1 | NDR0 |
| Initial value | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| Read/Write | — | — | — | — | R/W | R/W | R/W | R/W |

Store the next output data for TPC output group 0

SMR—Serial Mode Register

H'FFF0

SCIO

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|--------------|-----|-----|--------------|------|-----|------|------|
| | C/ \bar{A} | CHR | PE | O/ \bar{E} | STOP | MP | CKS1 | CKS0 |
| Initial value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Clock select 1 and 0

| Bit 1 | Bit 0 | Clock Source |
|-------|-------|-----------------|
| CKS1 | CKS0 | |
| 0 | 0 | ϕ clock |
| | 1 | $\phi/4$ clock |
| 1 | 0 | $\phi/16$ clock |
| | 1 | $\phi/64$ clock |

Multiprocessor mode

| | |
|---|----------------------------------|
| 0 | Multiprocessor function disabled |
| 1 | Multiprocessor format selected |

Stop bit length

| | |
|---|---------------|
| 0 | One stop bit |
| 1 | Two stop bits |

Parity mode

| | |
|---|-------------|
| 0 | Even parity |
| 1 | Odd parity |

Parity enable

| | |
|---|------------------------------------|
| 0 | Parity bit is not added or checked |
| 1 | Parity bit is added and checked |

Character length

| | |
|---|------------|
| 0 | 8-bit data |
| 1 | 7-bit data |

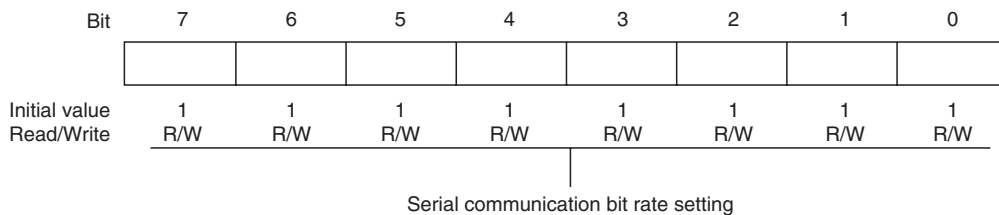
Communication mode (for serial communication interface)

| | |
|---|-------------------|
| 0 | Asynchronous mode |
| 1 | Synchronous mode |

GSM mode (for smart card interface)

| | |
|---|--|
| 0 | TEND flag is set 12.5 etu* after start bit |
| 1 | TEND flag is set 11.0 etu* after start bit |

Note: * etu: Elementary time unit (time required to transmit one bit)

BRR—Bit Rate Register**H'FFF B1****SCIO**

SCR—Serial Control Register

H'FFFFB2

SCIO

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|-----|-----|-----|-----|------|------|------|------|
| | TIE | RIE | TE | RE | MPIE | TEIE | CKE1 | CKE0 |
| Initial value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Receive enable

| | |
|---|-----------------------|
| 0 | Receiving is disabled |
| 1 | Receiving is enabled |

Transmit enable

| | |
|---|--------------------------|
| 0 | Transmitting is disabled |
| 1 | Transmitting is enabled |

Clock enable 1 and 0
(for serial communication interface)

| Bit 1 CKE1 | Bit 0 CKE0 | Description | |
|---------------|---------------|-------------------|--|
| 0 | 0 | Asynchronous mode | Internal clock: SCK pin available for generic I/O |
| | | Synchronous mode | Internal clock: SCK pin used for serial clock output |
| | 1 | Asynchronous mode | Internal clock: SCK pin used for clock output |
| | | Synchronous mode | Internal clock: SCK pin used for serial clock output |
| 1 | 0 | Asynchronous mode | External clock: SCK pin used for clock input |
| | | Synchronous mode | External clock: SCK pin used for serial clock input |
| | 1 | Asynchronous mode | External clock: SCK pin used for clock input |
| | | Synchronous mode | External clock: SCK pin used for serial clock input |

Clock enable 1 and 0 (for smart card interface)

| SMR GM | Bit 1 CKE1 | Bit 0 CKE0 | Description |
|-----------|---------------|---------------|-----------------------------------|
| 0 | 0 | 0 | SCK pin available for generic I/O |
| | | 1 | SCK pin used for clock output |
| 1 | 0 | 0 | SCK pin output fixed low |
| | | 1 | SCK pin used for clock output |
| | 1 | 0 | SCK pin output fixed high |
| | | 1 | SCK pin used for clock output |

Transmit-end interrupt enable

| | |
|---|--|
| 0 | Transmit-end interrupt requests (TEI) are disabled |
| 1 | Transmit-end interrupt requests (TEI) are enabled |

Multiprocessor interrupt enable

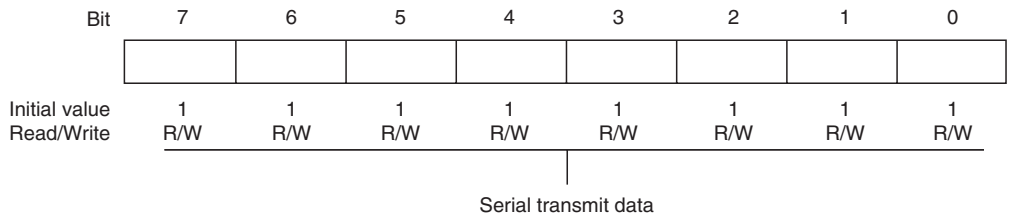
| | |
|---|---|
| 0 | Multiprocessor interrupts are disabled (normal receive operation) |
| 1 | Multiprocessor interrupts are enabled |

Receive interrupt enable

| | |
|---|---|
| 0 | Receive-data-full (RXI) and receive-error (ERI) interrupt requests are disabled |
| 1 | Receive-data-full (RXI) and receive-error (ERI) interrupt requests are enabled |

Transmit interrupt enable

| | |
|---|---|
| 0 | Transmit-data-empty interrupt request (TXI) is disabled |
| 1 | Transmit-data-empty interrupt request (TXI) is enabled |

TDR—Transmit Data Register**H'FFFB3****SCI0**

SSR—Serial Status Register

H'FFFFB4

SCIO

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|--------|--------|--------|---------|--------|------|-----|------|
| | TDRE | RDRF | ORER | FER/ERS | PER | TEND | MPB | MPBT |
| Initial value | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| Read/Write | R/(W)* | R/(W)* | R/(W)* | R/(W)* | R/(W)* | R | R | R/W |

| | |
|-----------------------------|--|
| Multiprocessor bit transfer | |
| 0 | Multiprocessor bit value in transmit data is 0 |
| 1 | Multiprocessor bit value in transmit data is 1 |

| | |
|--------------------|---|
| Multiprocessor bit | |
| 0 | Multiprocessor bit value in receive data is 0 |
| 1 | Multiprocessor bit value in receive data is 1 |

Transmit end (for serial communication interface)

| | |
|---|---|
| 0 | [Clearing condition] Read TDRE when TDRE = 1, then write 0 in TDRE. |
| 1 | [Setting conditions] <ul style="list-style-type: none"> Reset or transition to standby mode TE is cleared to 0 in SCR. TDRE is 1 when last bit of 1-byte serial character is transmitted. |

Transmit end (for smart card interface)

| | |
|---|---|
| 0 | [Clearing condition] Read TDRE when TDRE = 1, then write 0 in TDRE. |
| 1 | [Setting conditions] <ul style="list-style-type: none"> Reset or transition to standby mode TE is cleared to 0 in SCR and FER/ERS is cleared to 0. TDRE is 1 and FER/ERS is 0 (normal transmission) 2.5 etu* (when GM = 0) or 1.0 etu (when GM = 1) after 1-byte serial character is transmitted. |

Note: * etu: Elementary time unit (time required to transmit one bit)

Parity error

| | |
|---|--|
| 0 | [Clearing conditions] • Reset or transition to standby mode • Read PER when PER = 1, then write 0 in PER. |
| 1 | [Setting condition] Parity error (parity of receive data does not match parity setting of O/E bit in SMR) |

Framing error (for serial communication interface)

| | |
|---|--|
| 0 | [Clearing conditions] • Reset or transition to standby mode • Read FER when FER = 1, then write 0 in FER. |
| 1 | [Setting condition] Framing error (stop bit is 0) |

Error signal status (for smart card interface)

| | |
|---|--|
| 0 | [Clearing conditions] • Reset or transition to standby mode • Read ERS when ERS = 1, then write 0 in ERS. |
| 1 | [Setting condition] A low error signal is received. |

Overrun error

| | |
|---|---|
| 0 | [Clearing conditions] • Reset or transition to standby mode • Read ORER when ORER = 1, then write 0 in ORER. |
| 1 | [Setting condition] Overrun error (reception of the next serial data ends when RDRF = 1) |

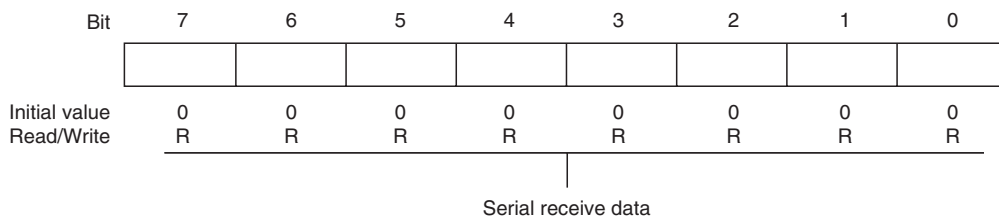
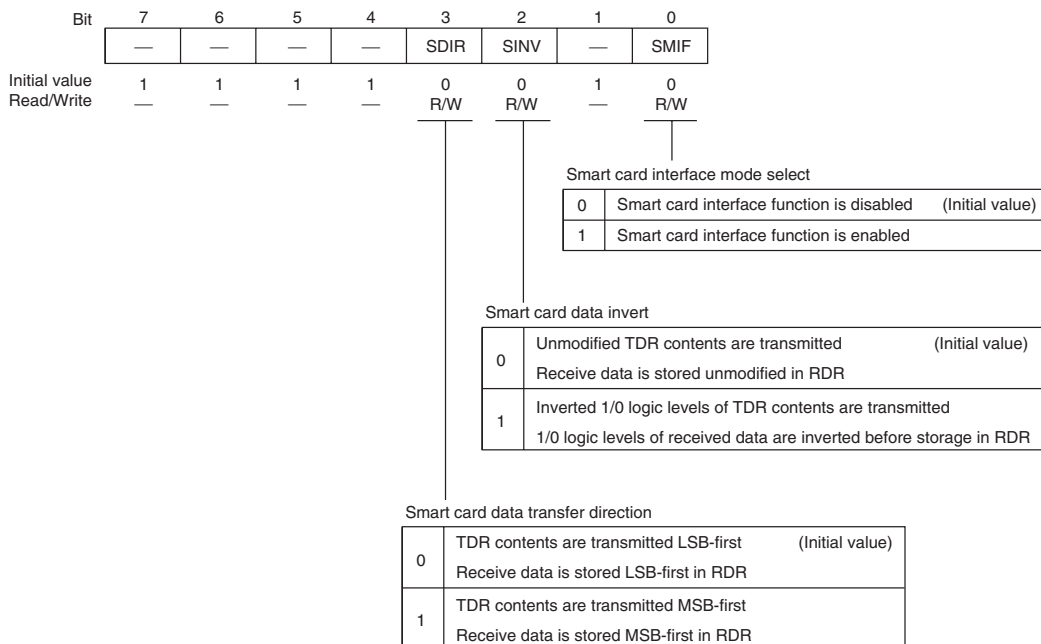
Receive data register full

| | |
|---|---|
| 0 | [Clearing conditions] • Reset or transition to standby mode • Read RDRF when RDRF = 1, then write 0 in RDRF. |
| 1 | [Setting condition] Serial data is received normally and transferred from RSR to RDR. |

Transmit data register empty

| | |
|---|---|
| 0 | [Clearing condition] Read TDRE when TDRE = 1, then write 0 in TDRE. |
| 1 | [Setting conditions] <ul style="list-style-type: none"> Reset or transition to standby mode TE is 0 in SCR. Data is transferred from TDR to TSR, enabling new data to be written in TDR |

Note: * Only 0 can be written, to clear the flag.

RDR—Receive Data Register**H'FFFB5****SCIO****SCMR—Smart Card Mode Register****H'FFFB6****SCIO**

SMR—Serial Mode Register**H'FFFFB8****SCI1**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|--------------|-----|-----|--------------|------|-----|------|------|
| | C/ \bar{A} | CHR | PE | O/ \bar{E} | STOP | MP | CKS1 | CKS0 |
| Initial value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Note: Bit functions are the same as for SCI0.

BRR—Bit Rate Register**H'FFFFB9****SCI1**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|-----|-----|-----|-----|-----|-----|-----|-----|
| | | | | | | | | |
| Initial value | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Note: Bit functions are the same as for SCI0.

SCR—Serial Control Register**H'FFFBA****SCI1**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|-----|-----|-----|-----|------|------|------|------|
| | TIE | RIE | TE | RE | MPIE | TEIE | CKE1 | CKE0 |
| Initial value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Note: Bit functions are the same as for SCI0.

TDR—Transmit Data Register**H'FFFBB****SCI1**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|-----|-----|-----|-----|-----|-----|-----|-----|
| | | | | | | | | |
| Initial value | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Note: Bit functions are the same as for SCI0.

SSR—Serial Status Register**H'FFFBC****SCI1**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|--------|--------|--------|---------|--------|------|-----|------|
| | TDRE | RDRF | ORER | FER/ERS | PER | TEND | MPB | MPBT |
| Initial value | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| Read/Write | R/(W)* | R/(W)* | R/(W)* | R/(W)* | R/(W)* | R | R | R/W |

Notes: Bit functions are the same as for SCI0.

* Only 0 can be written to clear the flag.

RDR—Receive Data Register**H'FFFBD****SCI1**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Initial value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Read/Write | R | R | R | R | R | R | R | R |

Note: Bit functions are the same as for SCI0.

SCMR—Smart Card Mode Register**H'FFFBE****SCI1**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|---|---|---|---|------|------|---|------|
| | — | — | — | — | SDIR | SINV | — | SMIF |
| Initial value | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 |
| Read/Write | — | — | — | — | R/W | R/W | — | R/W |

Note: Bit functions are the same as for SCI0.

P4DR—Port 4 Data Register**H'FFFD3****Port 4**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|-----|-----|-----|-----|-----|-----|-----|-----|
| | P47 | P46 | P45 | P44 | P43 | P42 | P41 | P40 |
| Initial value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Data for port 4 pins

P6DR—Port 6 Data Register**H'FFFD5****Port 6**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|-----|-----|-----|-----|-----|-----|-----|-----|
| | P67 | P66 | P65 | P64 | P63 | P62 | P61 | P60 |
| Initial value | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Read/Write | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Data for port 6 pins

P7DR—Port 7 Data Register**H'FFFD6****Port 7**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|-----|-----|-----|-----|-----|-----|-----|-----|
| | P77 | P76 | P75 | P74 | P73 | P72 | P71 | P70 |
| Initial value | — * | — * | — * | — * | — * | — * | — * | — * |
| Read/Write | R | R | R | R | R | R | R | R |

Data for port 7 pins

Note: * Determined by pins P7₇ to P7₀.**P8DR—Port 8 Data Register****H'FFFD7****Port 8**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|---|---|---|-----|-----|-----|-----|-----|
| | — | — | — | P84 | P83 | P82 | P81 | P80 |
| Initial value | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| Read/Write | — | — | — | R/W | R/W | R/W | R/W | R/W |

Data for port 8 pins

P9DR—Port 9 Data Register**H'FFFD8****Port 9**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|---|---|-----|-----|-----|-----|-----|-----|
| | — | — | P95 | P94 | P93 | P92 | P91 | P90 |
| Initial value | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| Read/Write | — | — | R/W | R/W | R/W | R/W | R/W | R/W |

Data for port 9 pins

PADR—Port A Data Register**H'FFFD9****Port A**

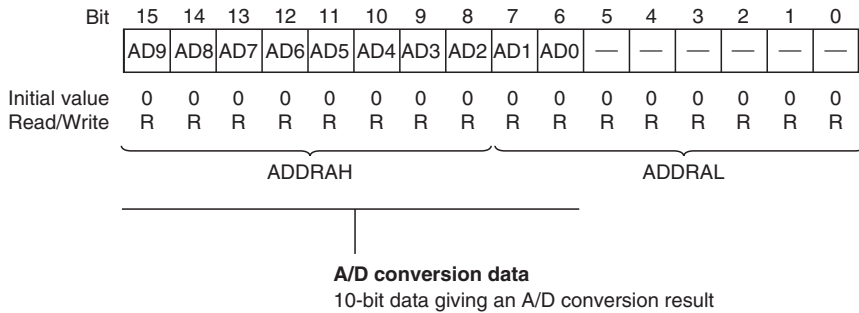
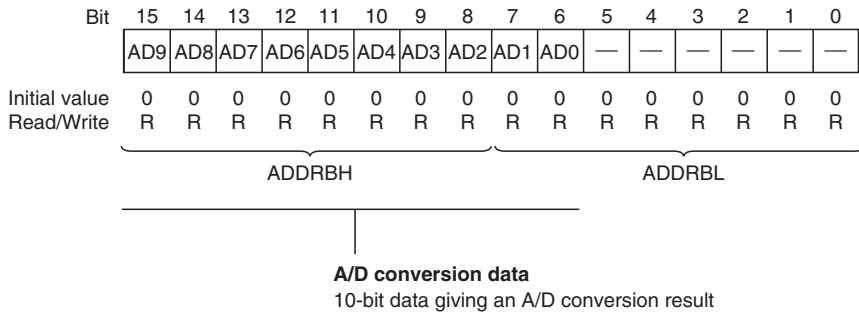
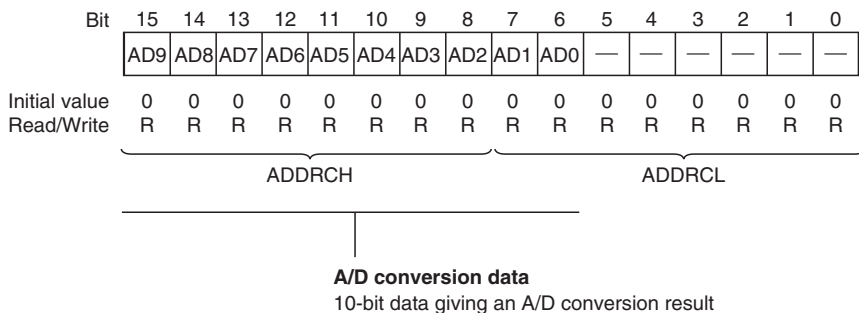
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|-----|-----|-----|-----|-----|-----|-----|-----|
| | PA7 | PA6 | PA5 | PA4 | PA3 | PA2 | PA1 | PA0 |
| Initial value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

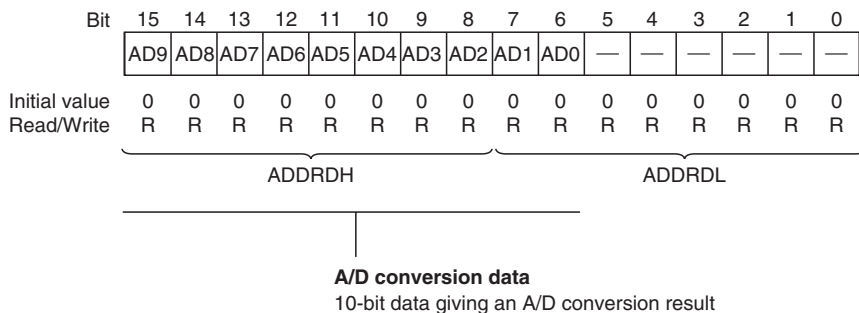
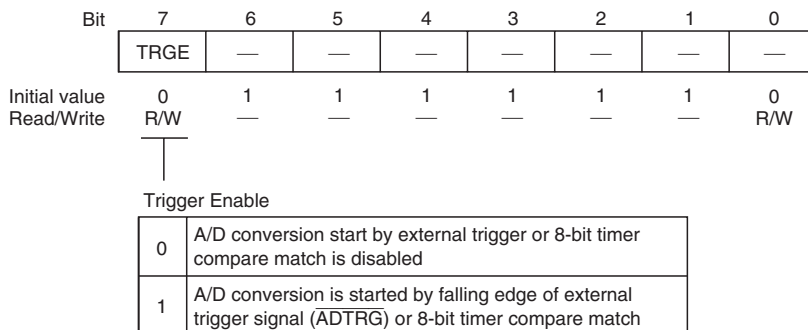
|
Data for port A pins

PBDR—Port B Data Register**H'FFFDA****Port B**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|-----|-----|-----|-----|-----|-----|-----|-----|
| | PB7 | PB6 | PB5 | PB4 | PB3 | PB2 | PB1 | PB0 |
| Initial value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

|
Data for port B pins

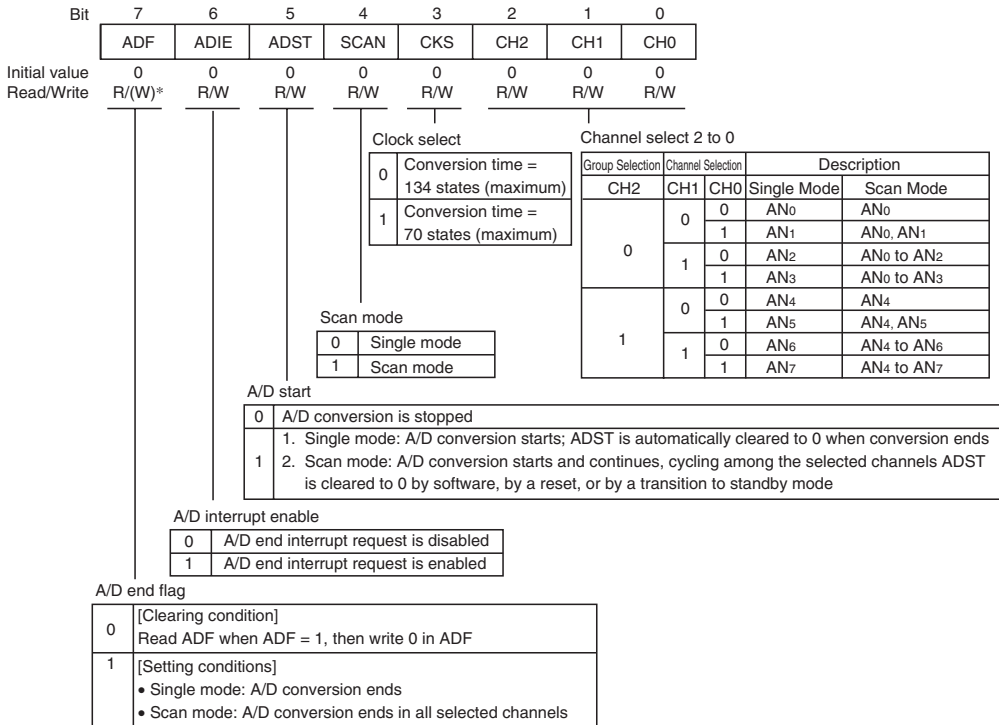
ADDRA H/L—A/D Data Register A H/L**H'FFFE0, H'FFFE1****A/D****ADDRB H/L—A/D Data Register B H/L****H'FFFE2, H'FFFE3****A/D****ADDRC H/L—A/D Data Register C H/L****H'FFFE4, H'FFFE5****A/D**

ADDRD H/L—A/D Data Register D H/L**H'FFFE6, H'FFFE7****A/D****ADCR—A/D Control Register****H'FFFE9****A/D**

ADCSR—A/D Control/Status Register

H'FFFFE8

A/D



Note: * Only 0 can be written to clear the flag.

Appendix C I/O Port Block Diagrams

C.1 Port 4 Block Diagram

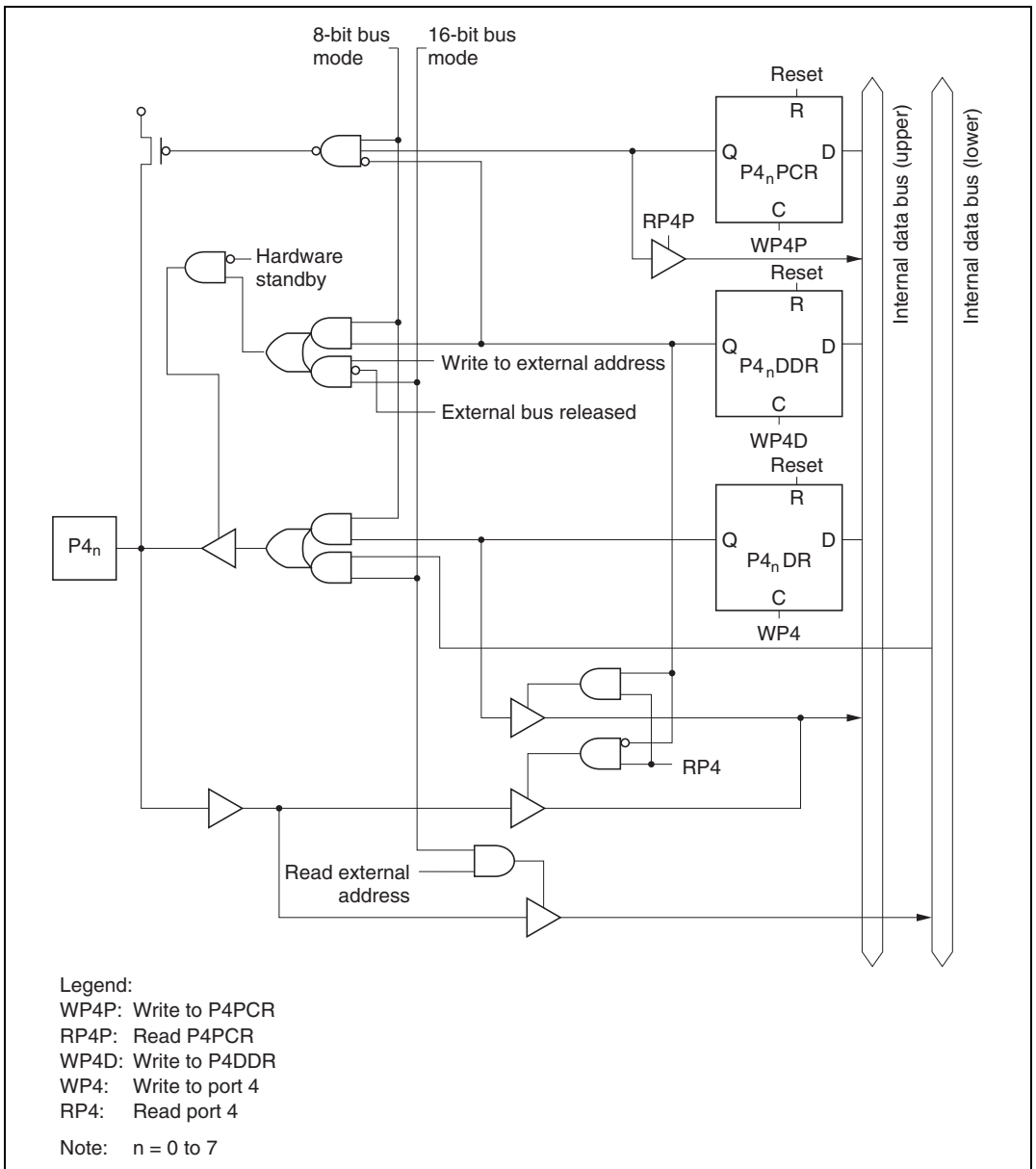


Figure C.1 Port 4 Block Diagram

C.2 Port 6 Block Diagrams

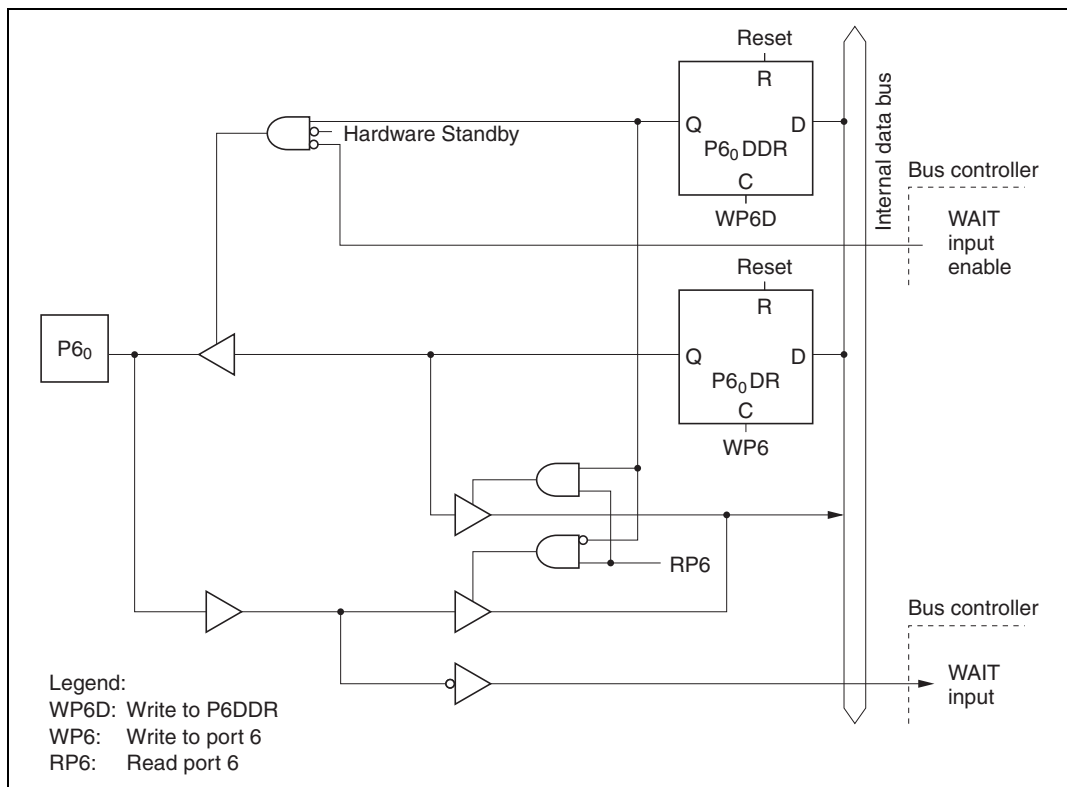
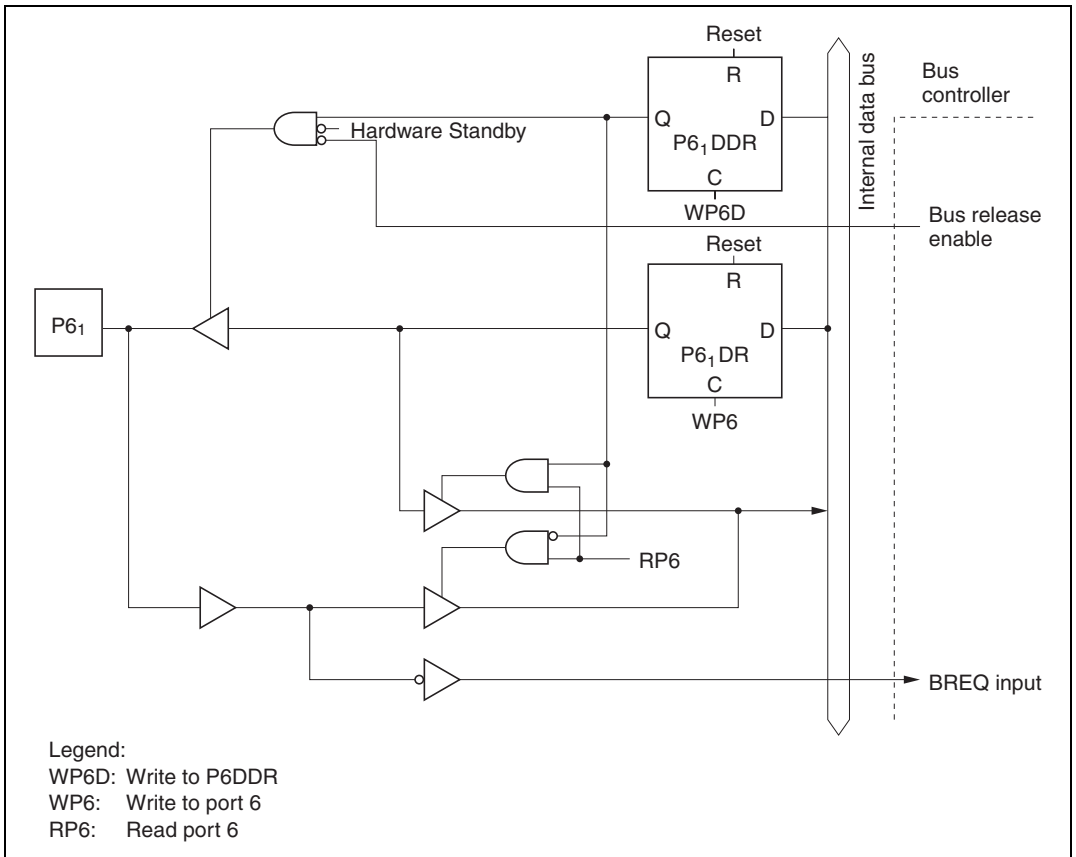


Figure C.2 (a) Port 6 Block Diagram (Pin P6₀)

Figure C.2 (b) Port 6 Block Diagram (Pin P6_i)

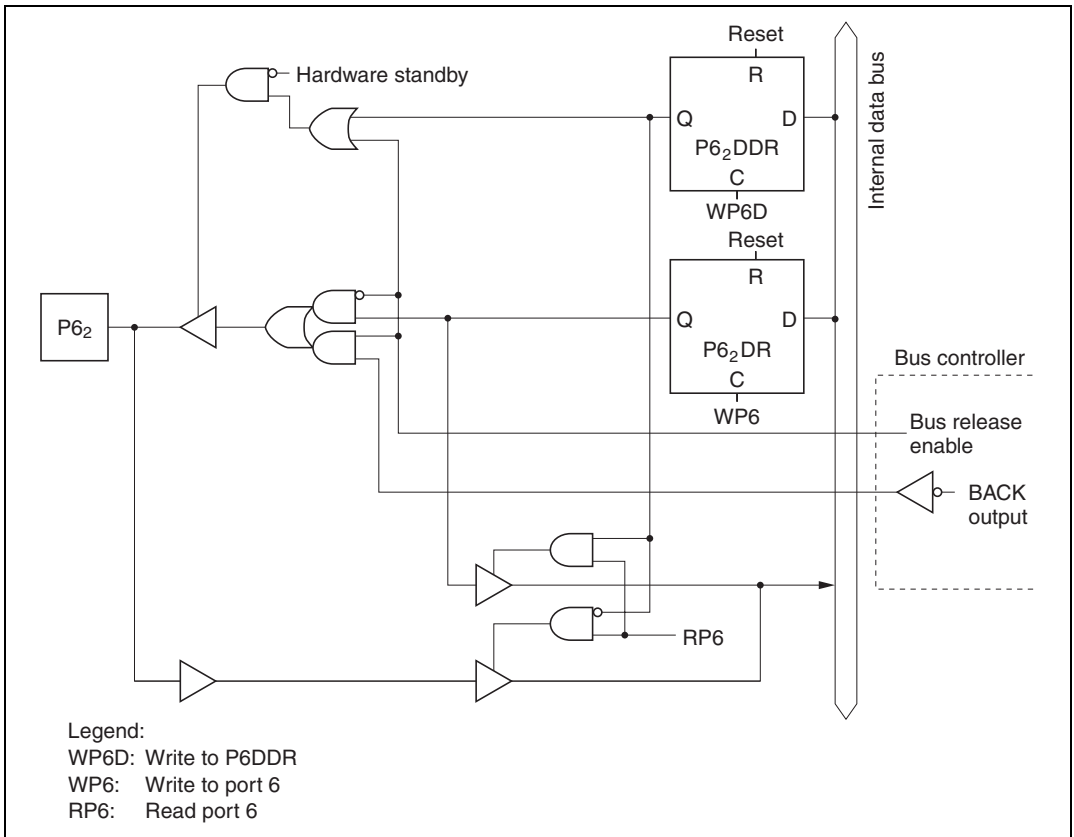
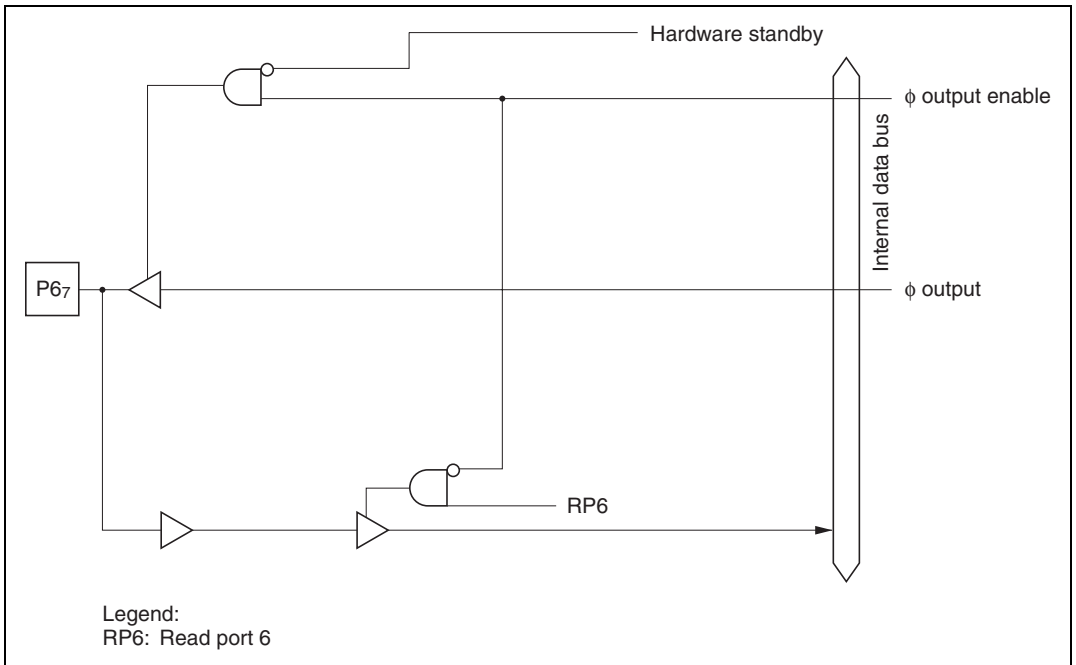


Figure C.2 (c) Port 6 Block Diagram (Pin P6₂)

**Figure C.2 (d) Port 6 Block Diagram (Pin P6₇)**

C.3 Port 7 Block Diagrams

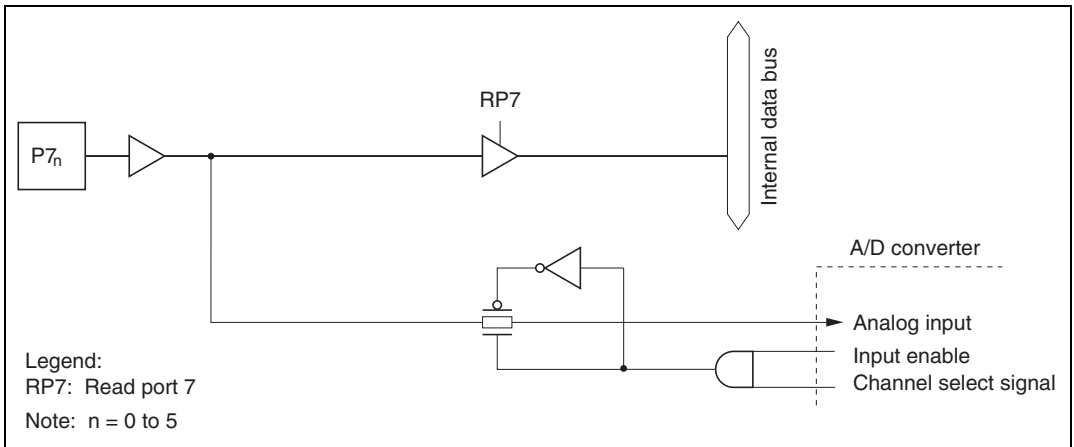


Figure C.3 (a) Port 7 Block Diagram (Pins P7₀ to P7₅)

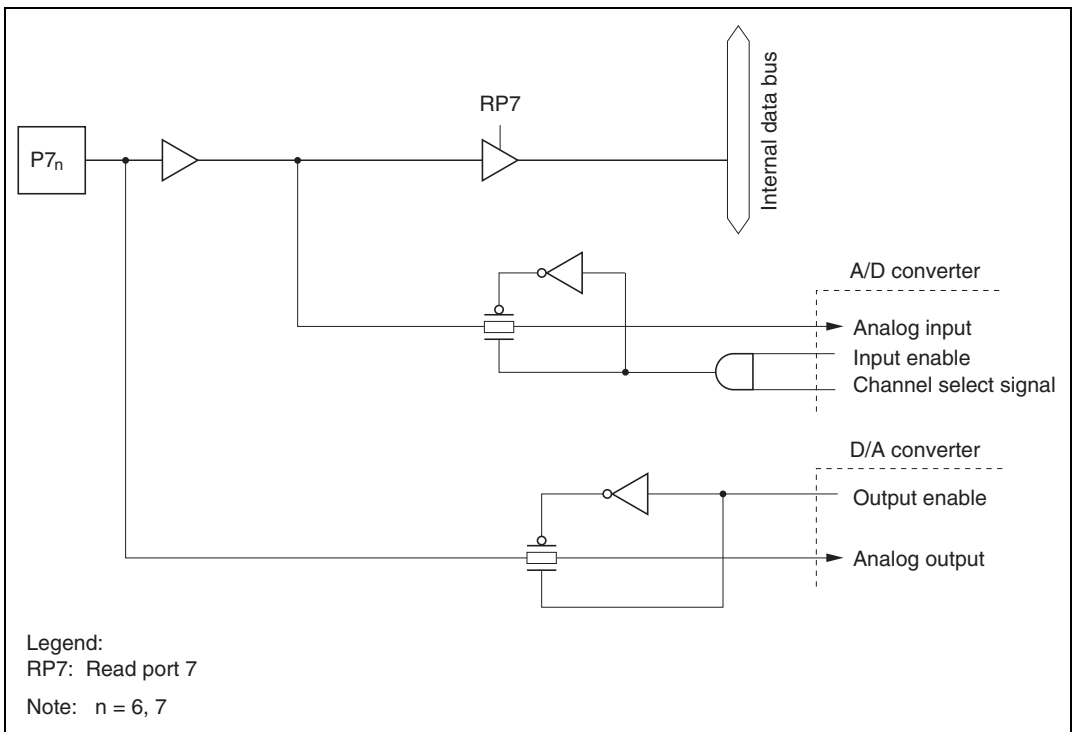


Figure C.3 (b) Port 7 Block Diagram (Pins P7₆ and P7₇)

C.4 Port 8 Block Diagrams

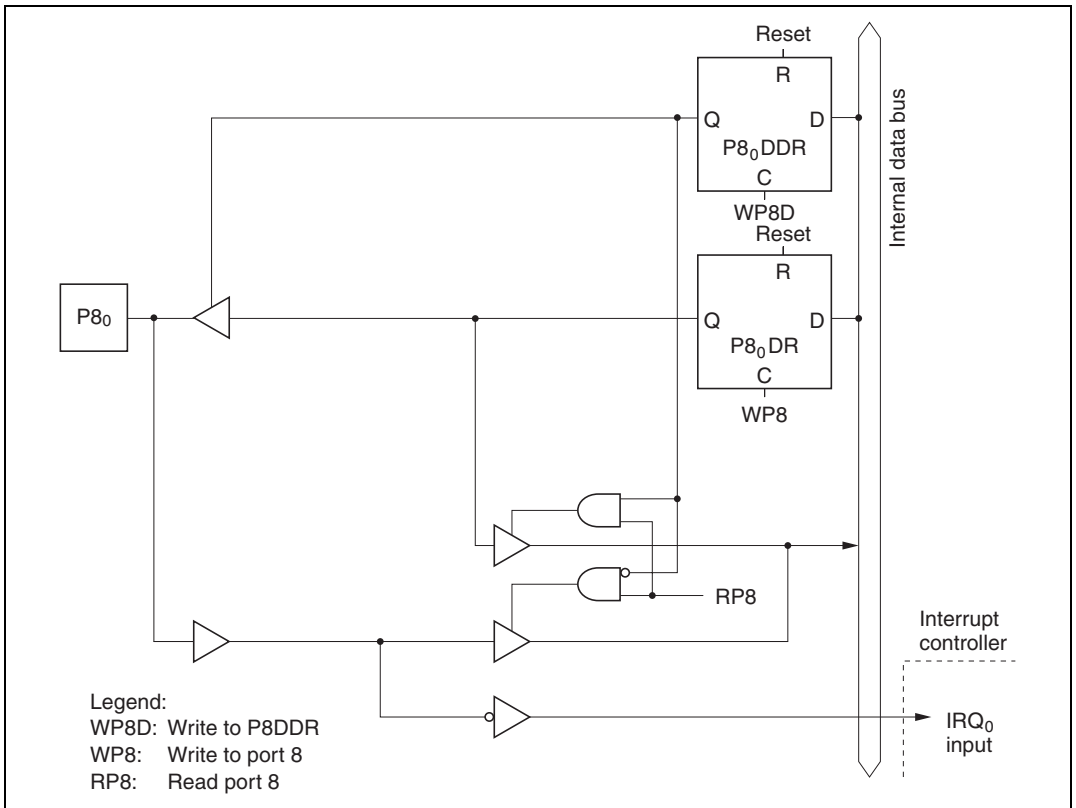


Figure C.4 (a) Port 8 Block Diagram (Pin P8₀)

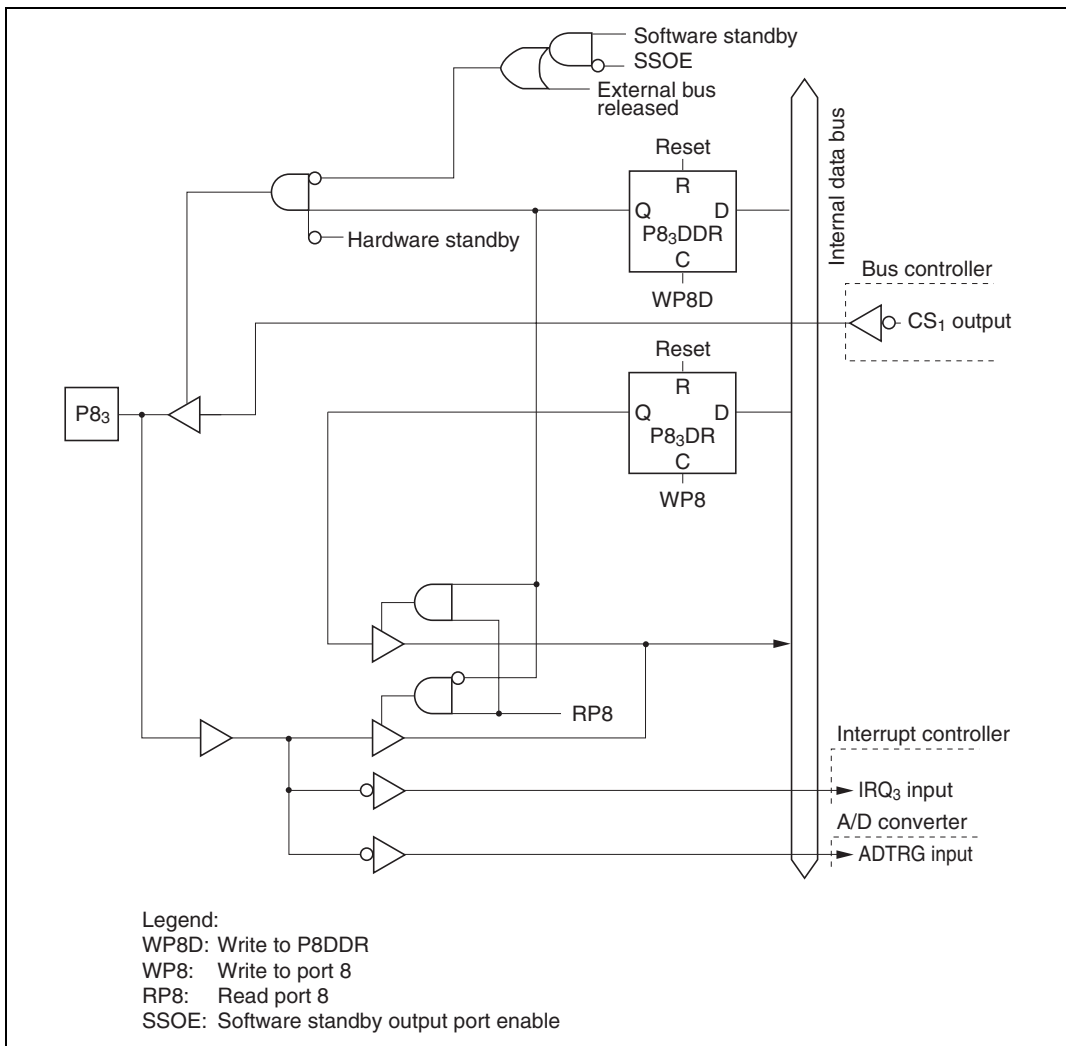


Figure C.4 (c) Port 8 Block Diagram (Pin P8₃)

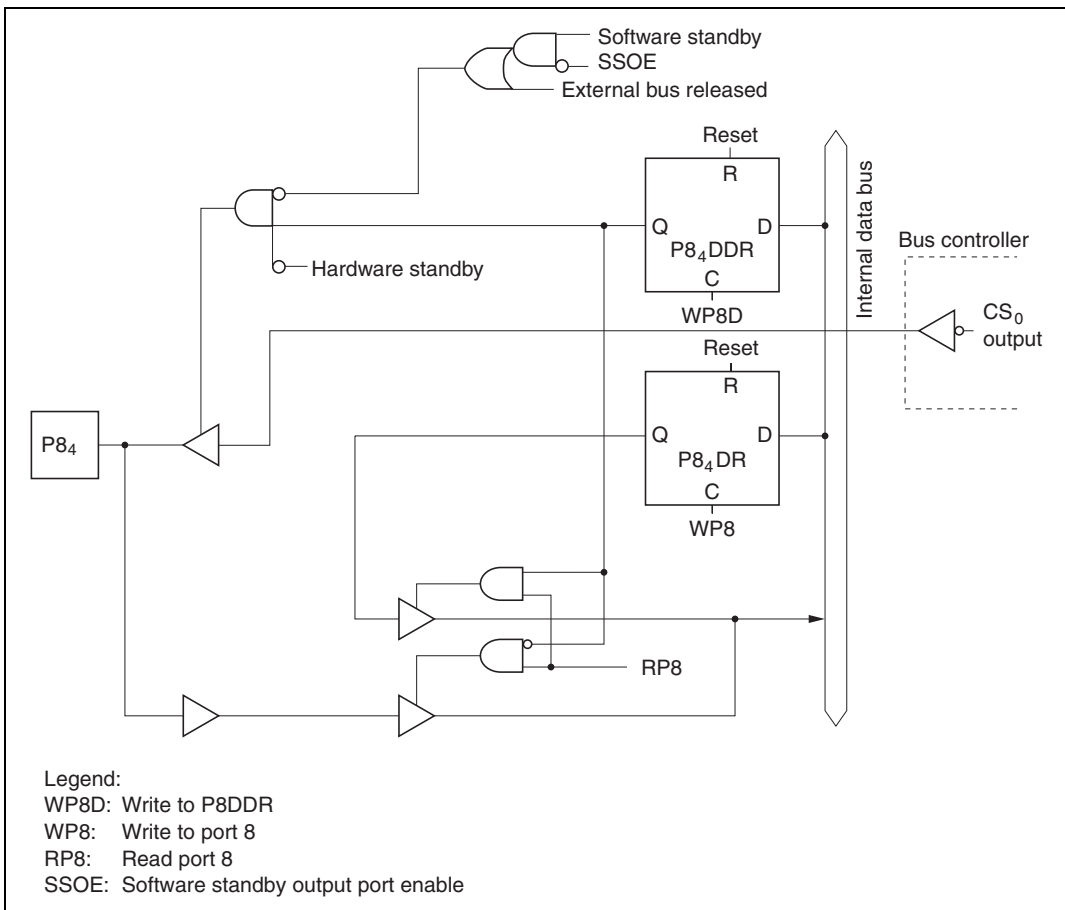


Figure C.4 (d) Port 8 Block Diagram (Pin P8₄)

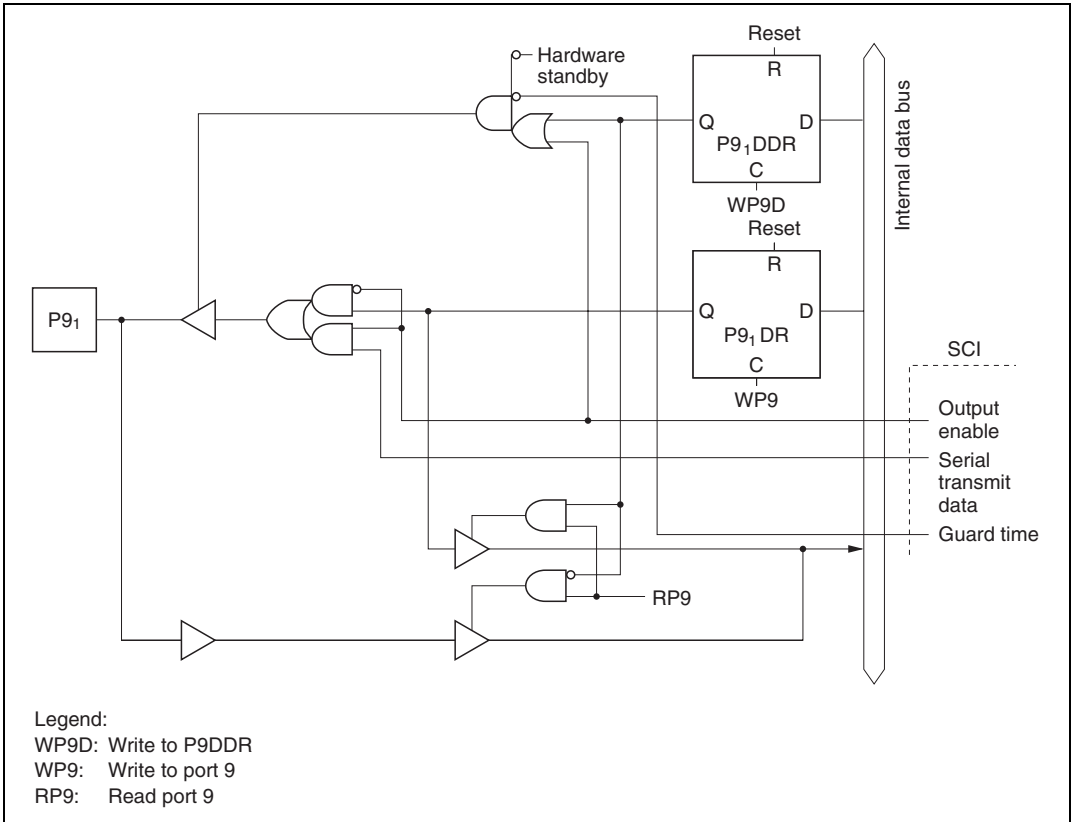


Figure C.5 (b) Port 9 Block Diagram (Pin P9₁)

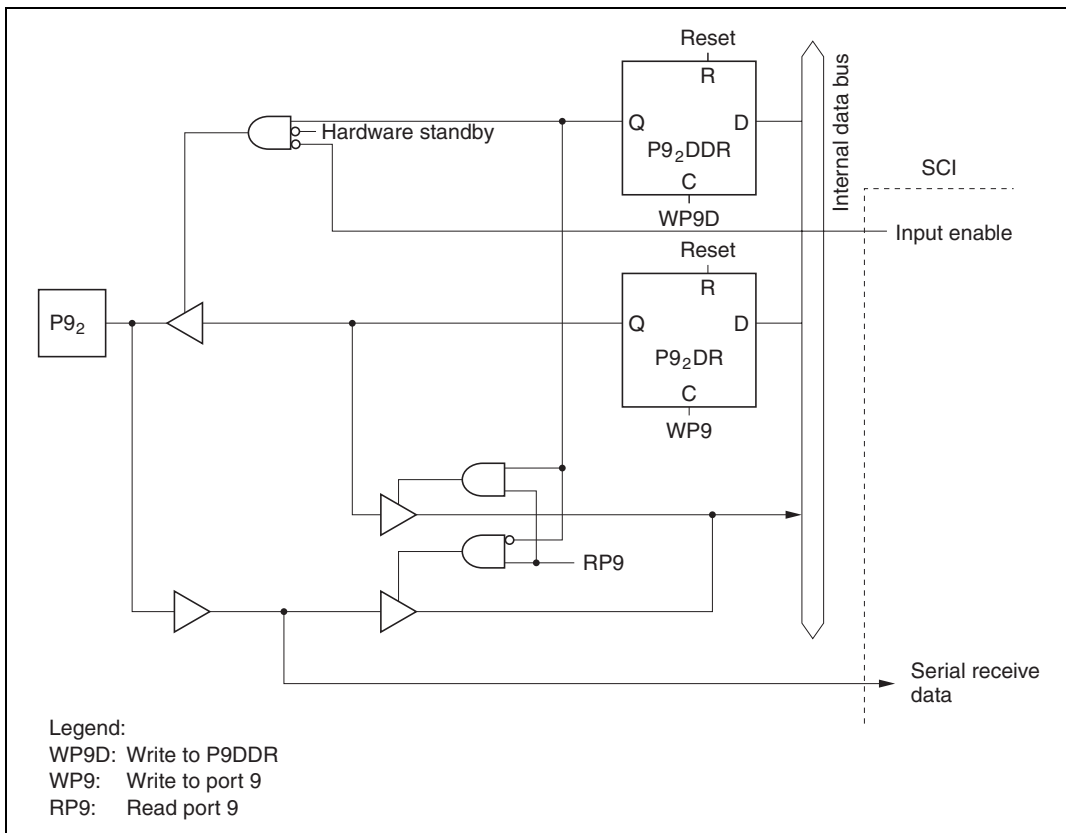


Figure C.5 (c) Port 9 Block Diagram (Pin P9₂)

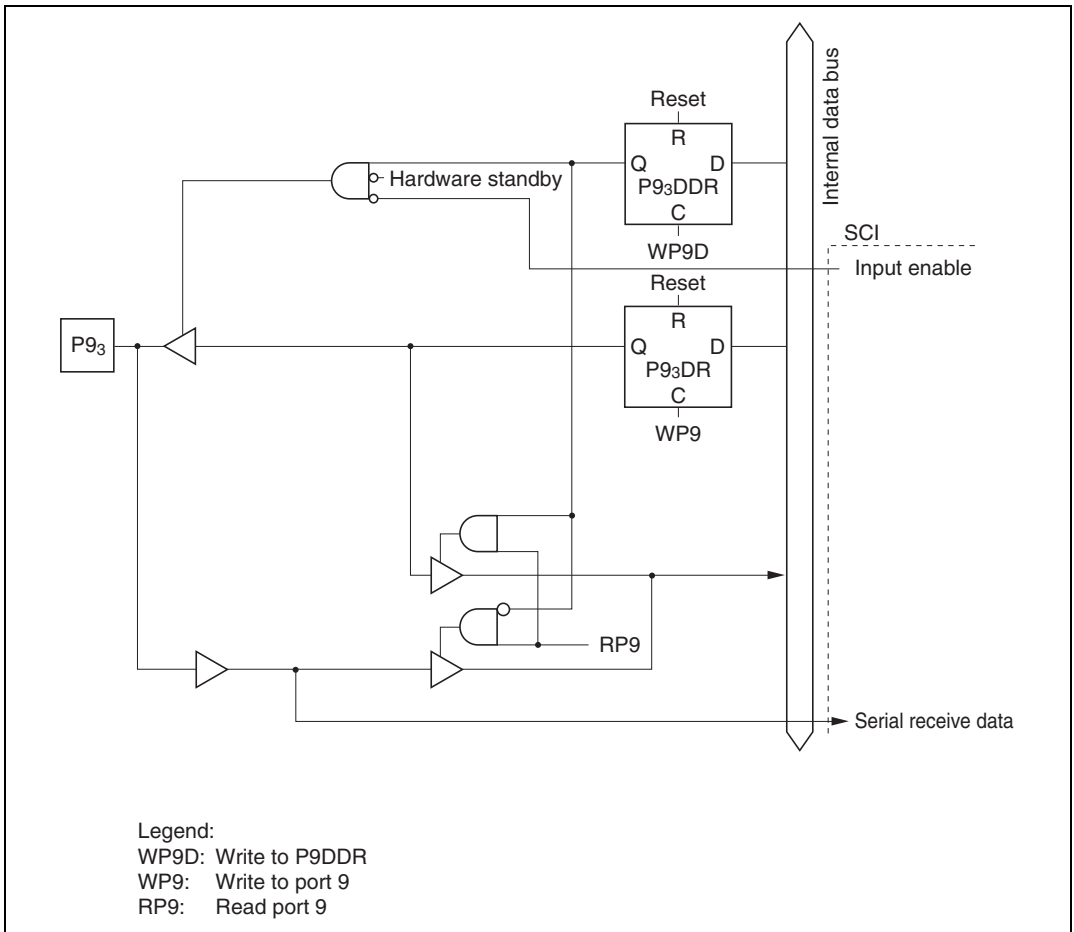
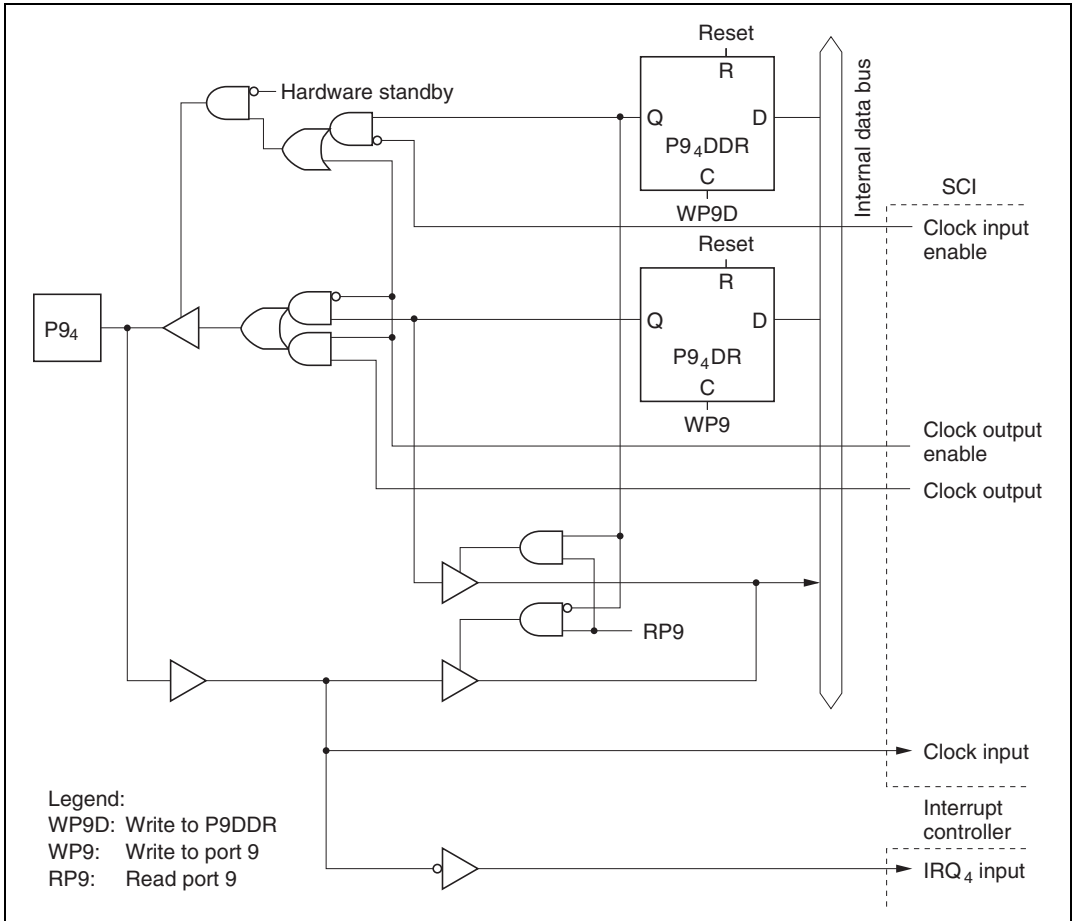


Figure C.5 (d) Port 9 Block Diagram (Pin P9₃)

Figure C.5 (e) Port 9 Block Diagram (Pin P9₄)

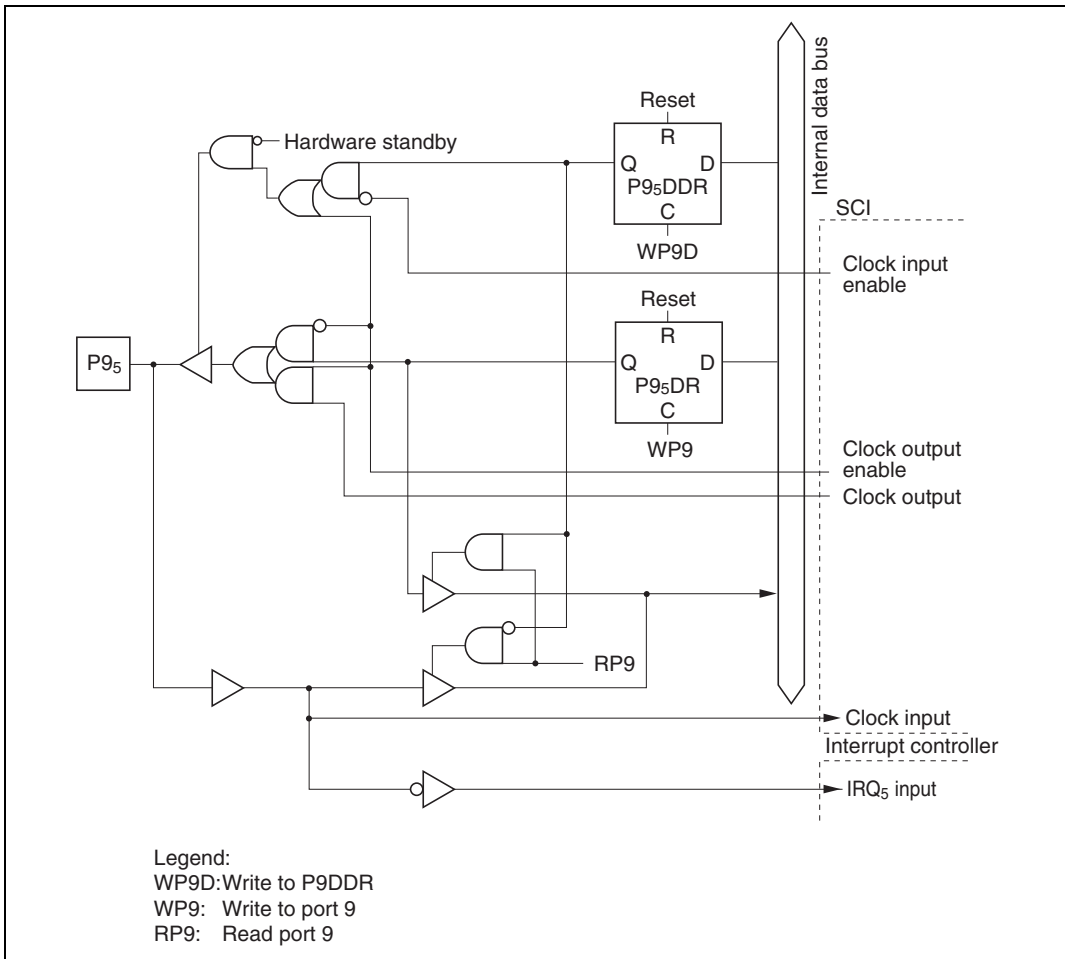


Figure C.5 (f) Port 9 Block Diagram (Pin P9₅)

C.6 Port A Block Diagrams

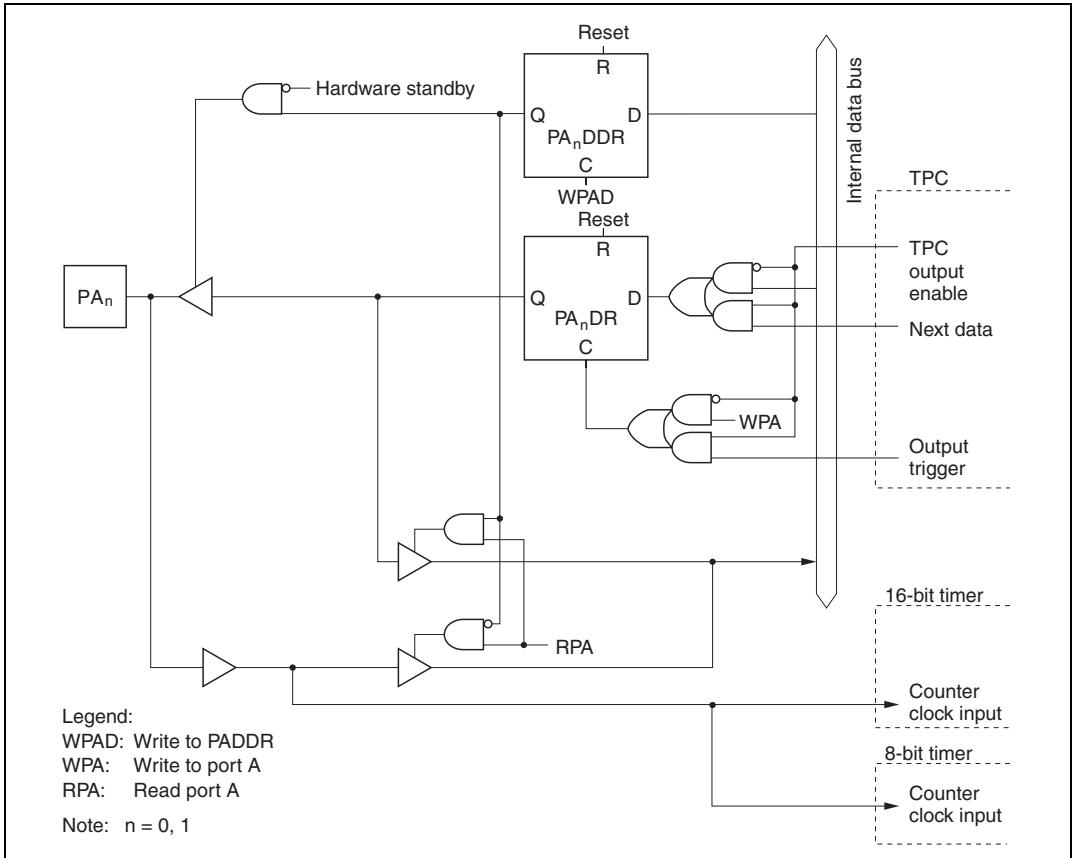


Figure C.6 (a) Port A Block Diagram (Pins PA₀ and PA₁)

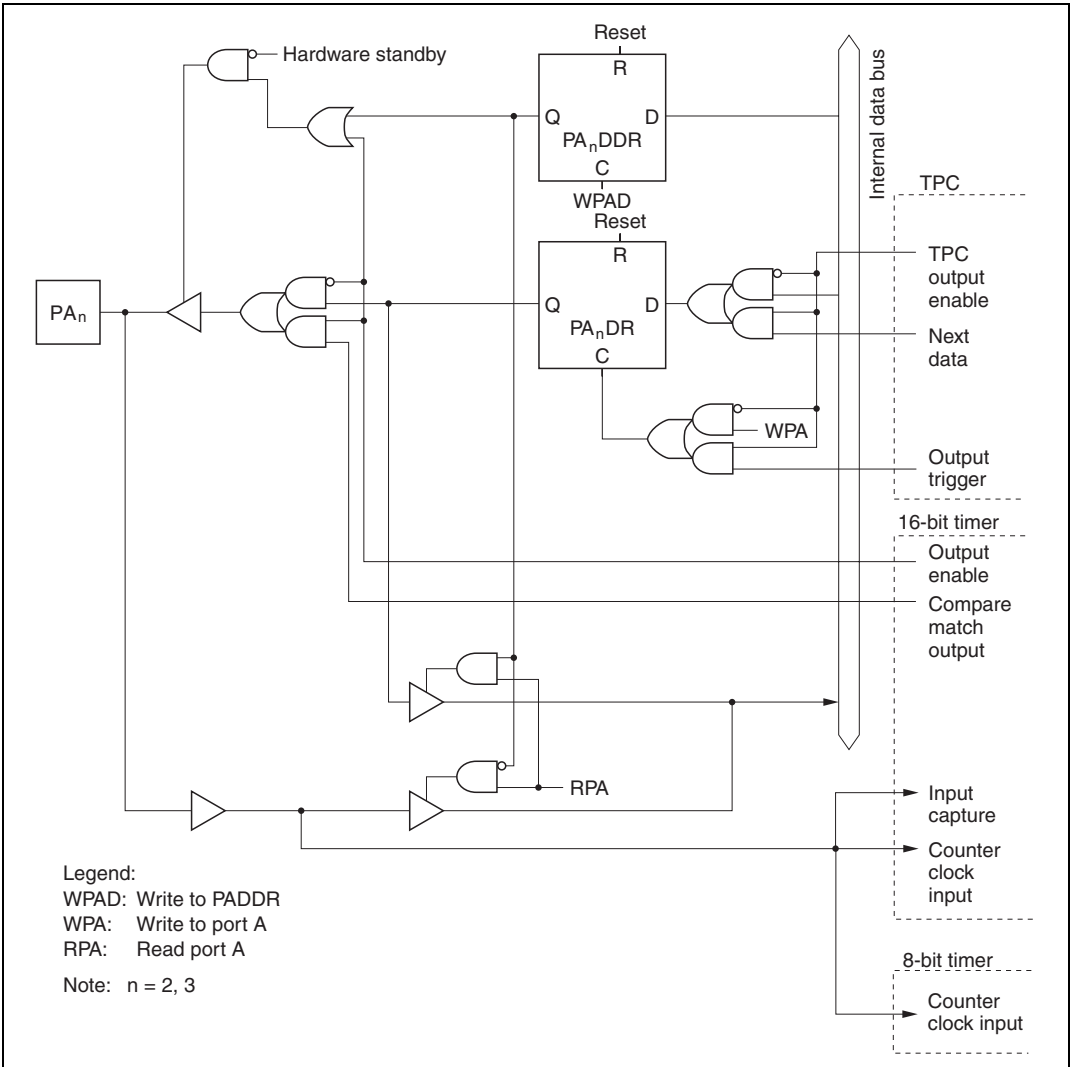


Figure C.6 (b) Port A Block Diagram (Pins PA₂ and PA₃)

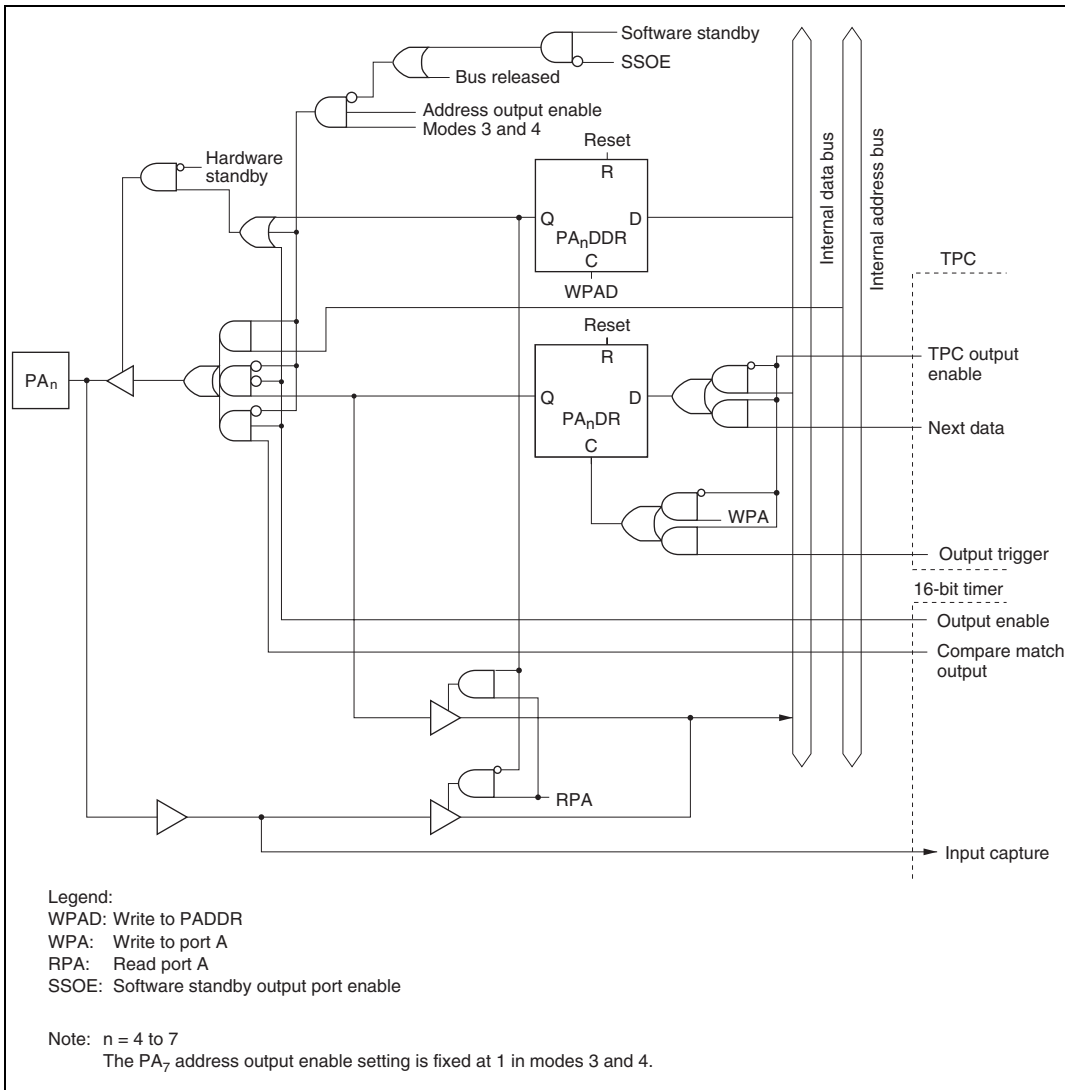


Figure C.6 (c) Port A Block Diagram (Pins PA₄ to PA₇)

C.7 Port B Block Diagrams

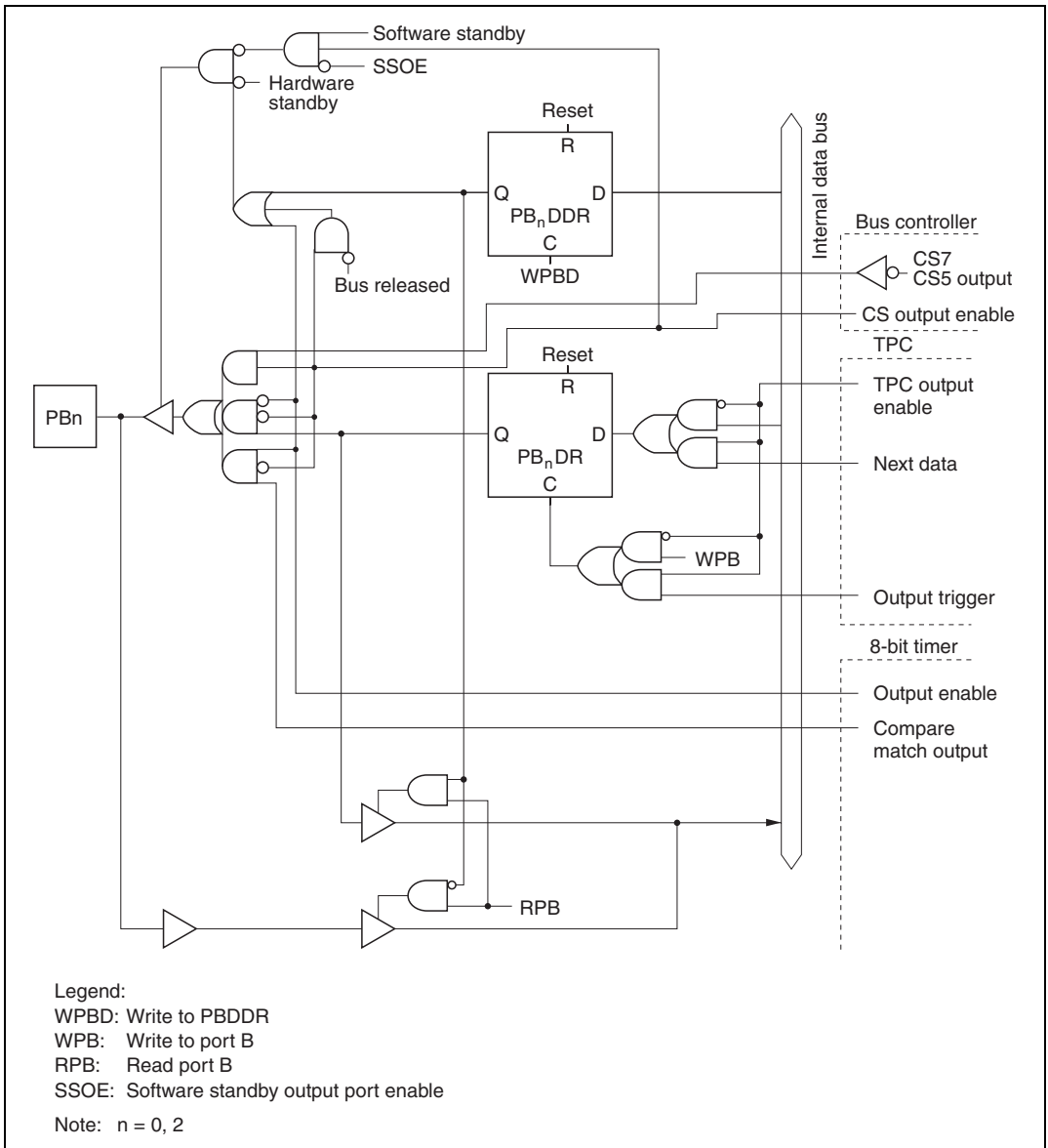


Figure C.7 (a) Port B Block Diagram (Pins PB_0 and PB_2)

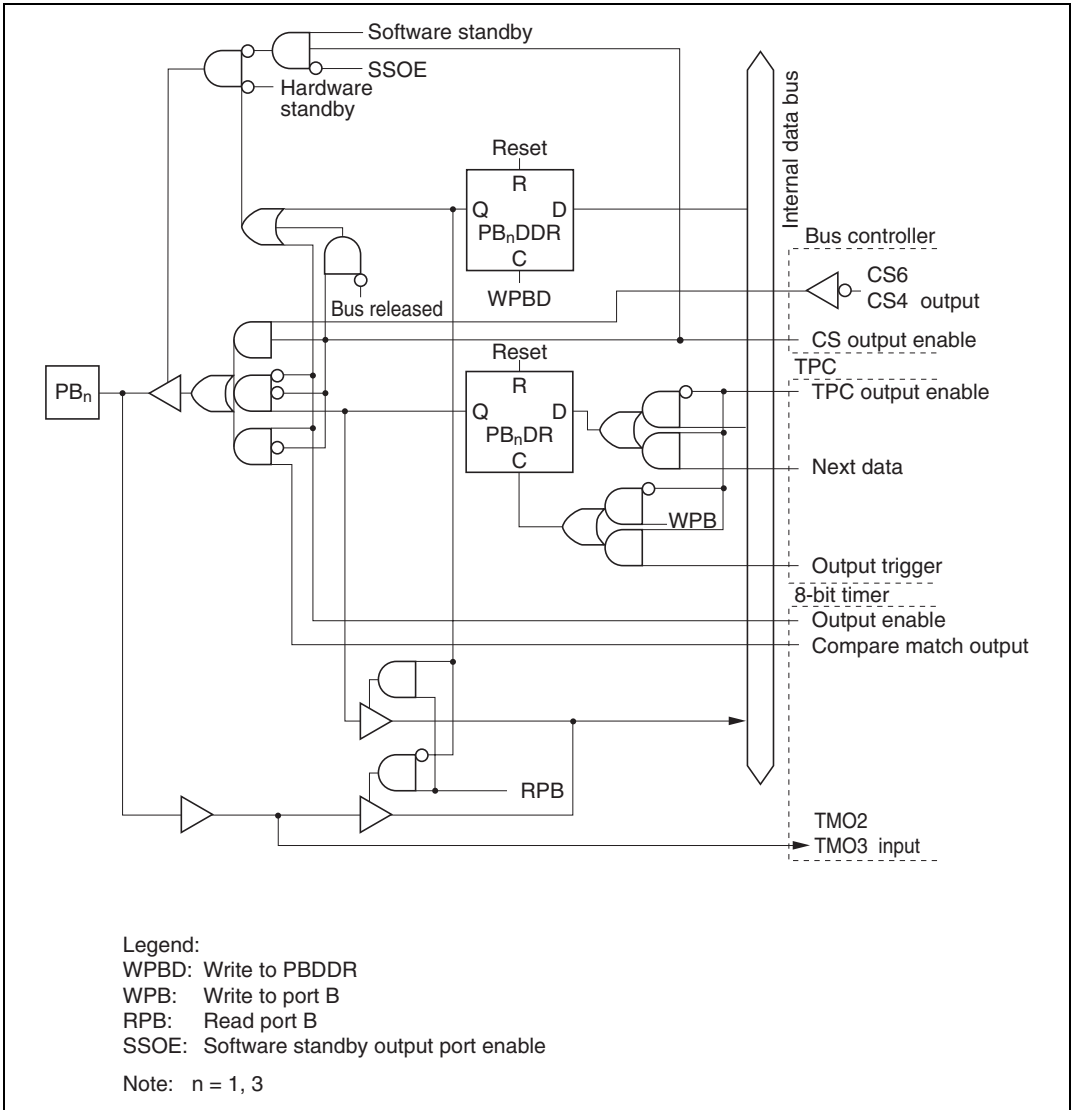


Figure C.7 (b) Port B Block Diagram (Pins PB₁ and PB₃)

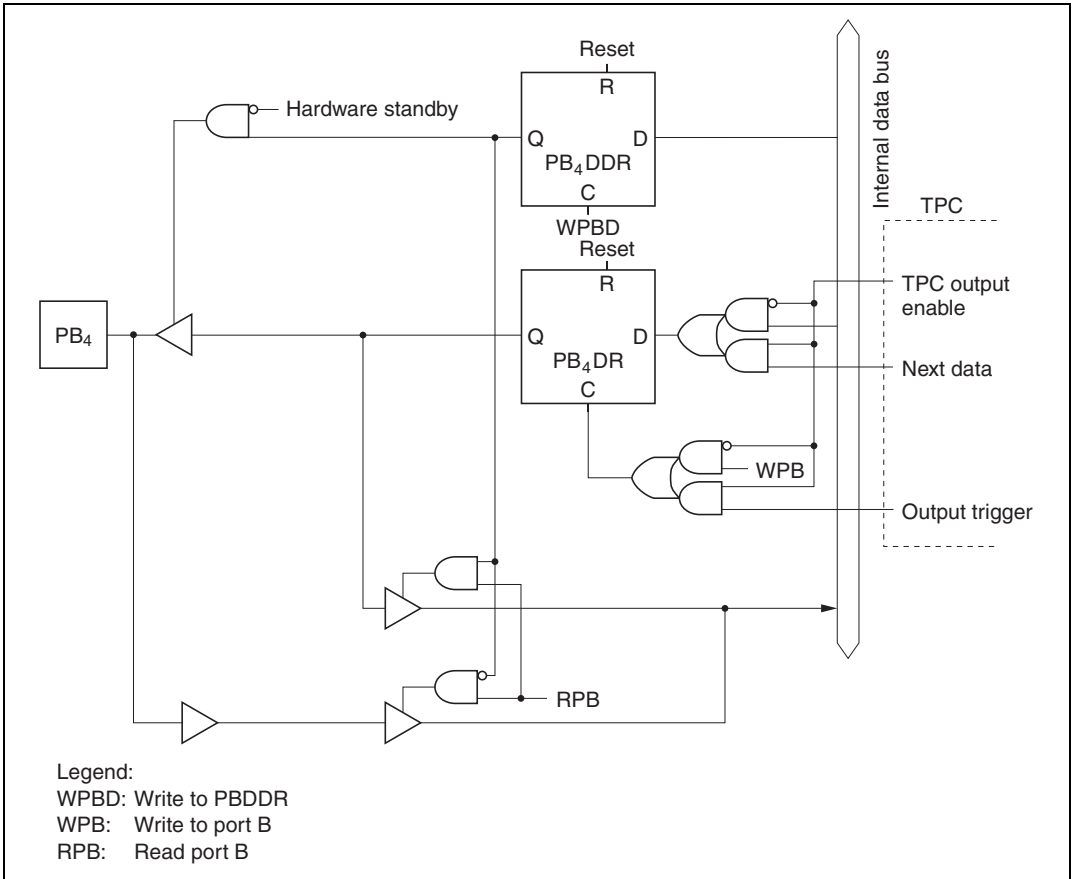


Figure C.7 (c) Port B Block Diagram (Pin PB₄)

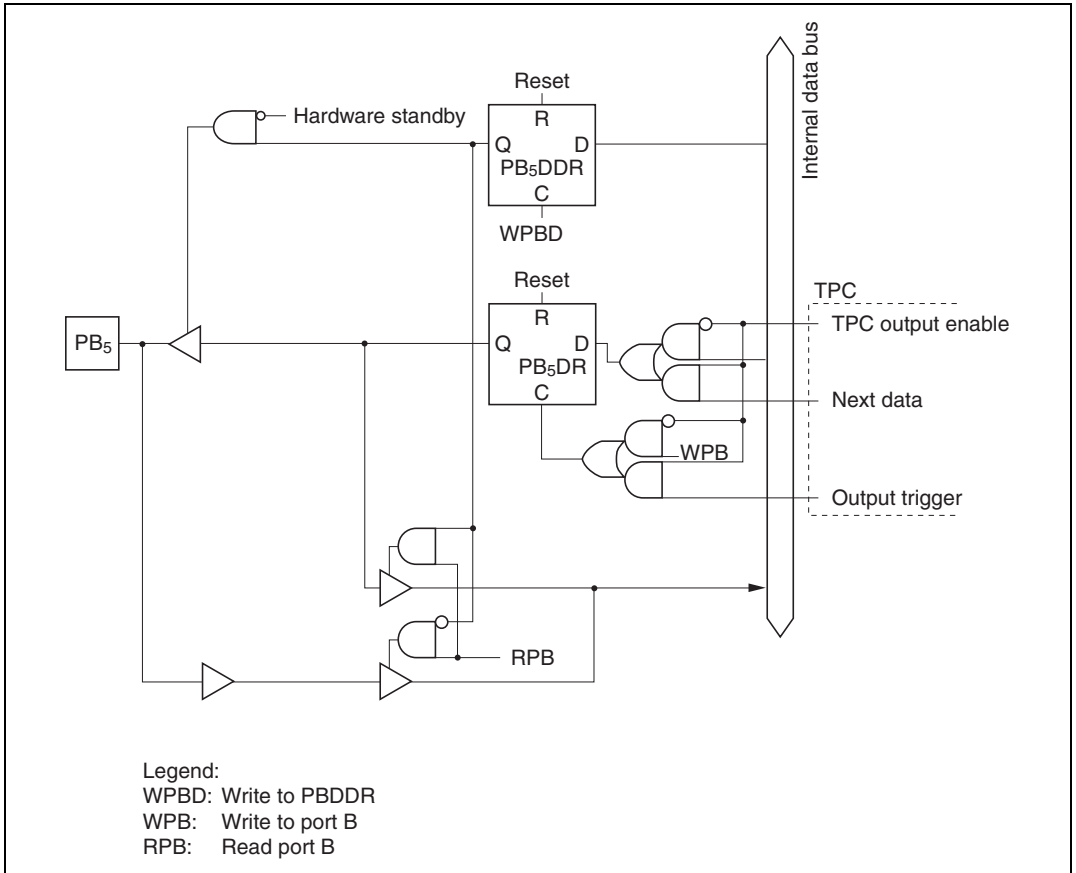


Figure C.7 (d) Port B Block Diagram (Pin PB₅)

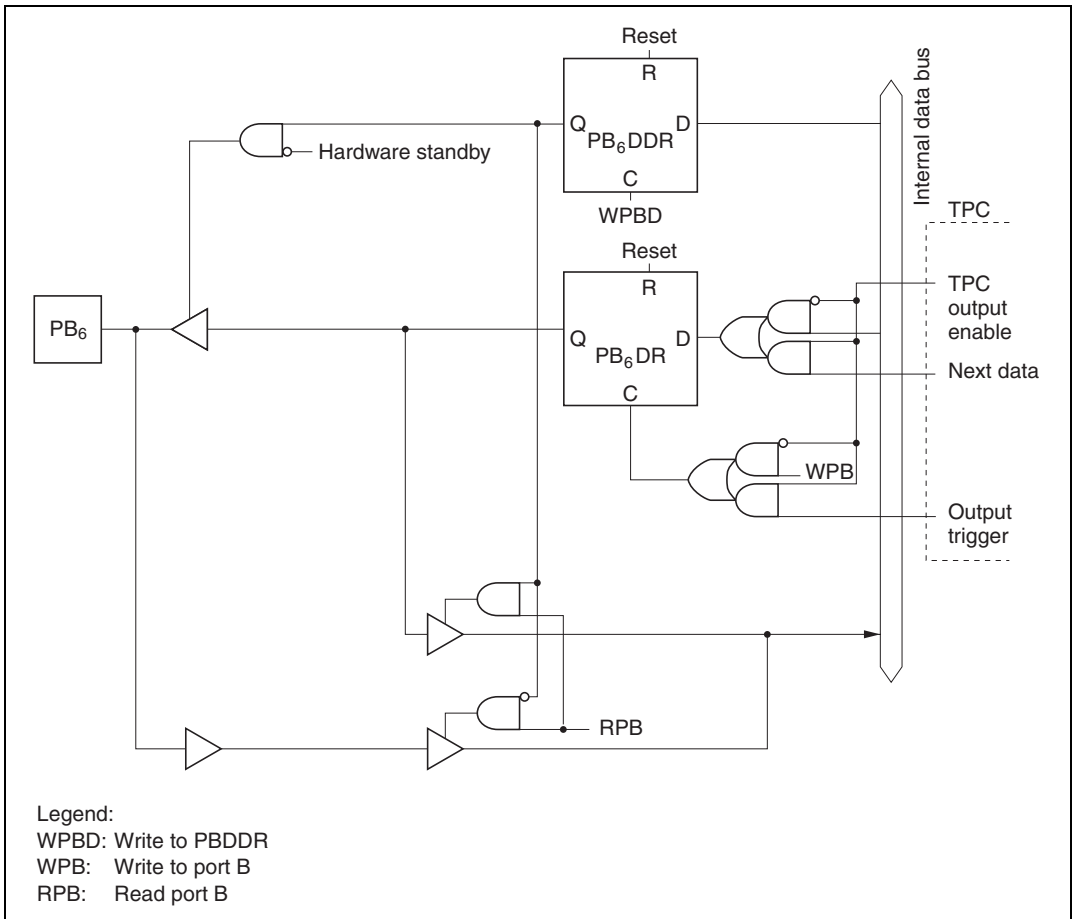


Figure C.7 (e) Port B Block Diagram (Pin PB₆)

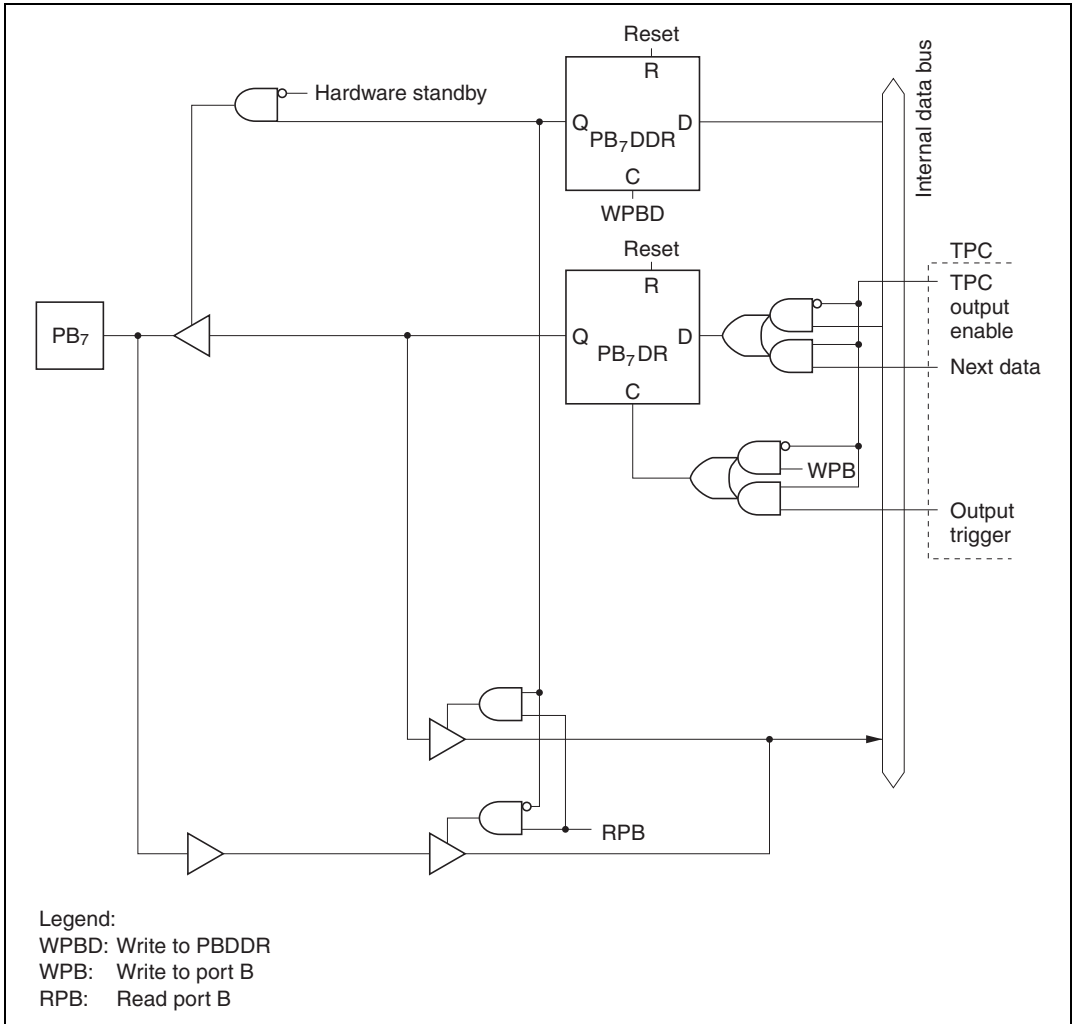


Figure C.7 (f) Port B Block Diagram (Pin PB₇)

Appendix D Pin States

D.1 Port States in Each Mode

Table D.1 Port States

| Pin Name | Mode | Reset | Hardware Standby Mode | Software Standby Mode | Bus-Released Mode | Program Execution Mode |
|------------------------------------|------|--------------|-----------------------|---------------------------------------|---------------------|--|
| A ₇ to A ₀ | — | L | T | (SSOE = 0) T (SSOE = 1) Keep | T | A ₇ to A ₀ |
| A ₁₅ to A ₈ | — | L | T | (SSOE = 0) T (SSOE = 1) Keep | T | A ₁₅ to A ₈ |
| D ₁₅ to D ₈ | — | T | T | T | T | D ₁₅ to D ₈ |
| P4 ₇ to P4 ₀ | 1, 3 | T | T | Keep | Keep | I/O port |
| | 2, 4 | T | T | T | T | D ₇ to D ₀ |
| A ₁₉ to A ₁₆ | — | L | T | (SSOE = 0) T (SSOE = 1) Keep | T | A ₁₉ to A ₁₆ |
| P6 ₀ | — | T | T | Keep | Keep | I/O port WAIT |
| P6 ₁ | — | T | T | (BRLE = 0) Keep (BRLE = 1) T | T | I/O port BREQ |
| P6 ₂ | — | T | T | (BRLE = 0) Keep (BRLE = 1) H | L | (BRLE = 0) I/O port (BRLE = 1) BACK |
| AS, RD, HWR, LWR | — | H | T | (SSOE = 0) T (SSOE = 1) H | T | AS, RD, HWR, LWR |
| P6 ₇ | — | Clock output | T | (PSTOP = 0) H | (PSTOP = 0) φ | (PSTOP = 0) φ |
| | | | T | (PSTOP = 1) Keep | (PSTOP = 1) Keep | (PSTOP = 1) Input port |
| | | | T | Keep | Keep | Input port |

| Pin Name | Mode | Reset | Hardware Standby Mode | Software Standby Mode | Bus-Released Mode | Program Execution Mode |
|------------------------------------|------|-------|-----------------------|--|--|---|
| P7 ₇ to P7 ₀ | — | T | T | T | T | Input port |
| P8 ₀ | — | T | T | Keep | — | I/O port |
| P8 ₁ | — | T | T | (DDR = 0) T (DDR = 1, SSOE = 0) T (DDR = 1, SSOE = 1) H | (DDR = 0) Keep (DDR = 1) T | (DDR = 0) Input port (DDR = 1) \overline{CS}_3 |
| P8 ₂ | — | T | T | (DDR = 0) T (DDR = 1, SSOE = 0) T (DDR = 1, SSOE = 1) H | (DDR = 0) Keep (DDR = 1) T | (DDR = 0) Input port (DDR = 1) \overline{CS}_2 |
| P8 ₃ | — | T | T | (DDR = 0) T (DDR = 1, SSOE = 0) T (DDR = 1, SSOE = 1) H | (DDR = 0) Keep (DDR = 1) T | (DDR = 0) Input port (DDR = 1) \overline{CS}_1 |
| P8 ₄ | — | H | T | (DDR = 0) T (DDR = 1, SSOE = 0) T (DDR = 1, SSOE = 1) H | (DDR = 0) Keep (DDR = 1) T | (DDR = 0) Input port (DDR = 1) \overline{CS}_0 |
| P9 ₅ to P9 ₀ | — | T | T | Keep | Keep | I/O port |
| PA ₃ to PA ₀ | — | T | T | Keep | Keep | I/O port |
| PA ₆ to PA ₄ | 1, 2 | T | T | Keep | Keep | I/O port |
| | 3, 4 | T | T | (Address output)* ¹ (SSOE = 0) T (SSOE = 1) Keep (Otherwise)* ² Keep | (Address output)* ¹ T (Otherwise)* ² Keep | (Address output)* ¹ A ₂₃ to A ₂₁ (Otherwise)* ² I/O port |

| Pin Name | Mode | Reset | Hardware Standby Mode | Software Standby Mode | Bus-Released Mode | Program Execution Mode |
|------------------------------------|------|-------|-----------------------|--|---|--|
| PA ₇ | 1, 2 | T | T | Keep | Keep | I/O port |
| | 3, 4 | L | T | (SSOE = 0) T (SSOE = 1) Keep | T | A ₂₀ |
| PB ₃ to PB ₀ | — | T | T | (CS output)* ³ (SSOE = 0) T (SSOE = 1) H (Otherwise)* ⁴ Keep | (CS output)* ³ T (Otherwise)* ⁴ Keep | (CS output)* ³ CS ₇ to CS ₄ (Otherwise)* ⁴ I/O port |
| | | | | Keep | Keep | I/O port |
| PB ₇ to PB ₄ | — | T | T | Keep | Keep | I/O port |

Legend:

H: High

L: Low

T: High-impedance state

keep: Input pins are in the high-impedance state; output pins maintain their previous state.

DDR: Data direction register

- Notes:
1. When A23E, A22E, A21E = 0 in BRCCR (bus release control register).
 2. When A23E, A22E, A21E = 1 in BRCCR (bus release control register).
 3. When CS7E, CS6E, CS5E, CS4E = 1 in CSCR (chip select control register).
 4. When CS7E, CS6E, CS5E, CS4E = 0 in CSCR (chip select control register).

D.2 Pin States at Reset

Modes 1 and 2: Figure D.1 is a timing diagram for the case in which $\overline{\text{RES}}$ goes low during an external memory access in mode 1 or 2. As soon as $\overline{\text{RES}}$ goes low, all ports are initialized to the input state. $\overline{\text{AS}}$, $\overline{\text{RD}}$, $\overline{\text{HWR}}$, $\overline{\text{LWR}}$, and $\overline{\text{CS}}_0$ go high, and D_{15} to D_0 go to the high-impedance state. The address bus is initialized to the low output level 2.5 ϕ clock cycles after the low level of $\overline{\text{RES}}$ is sampled. Clock pin $\text{P6}_7/\phi$ goes to the output state at the next rise of ϕ after $\overline{\text{RES}}$ goes low.

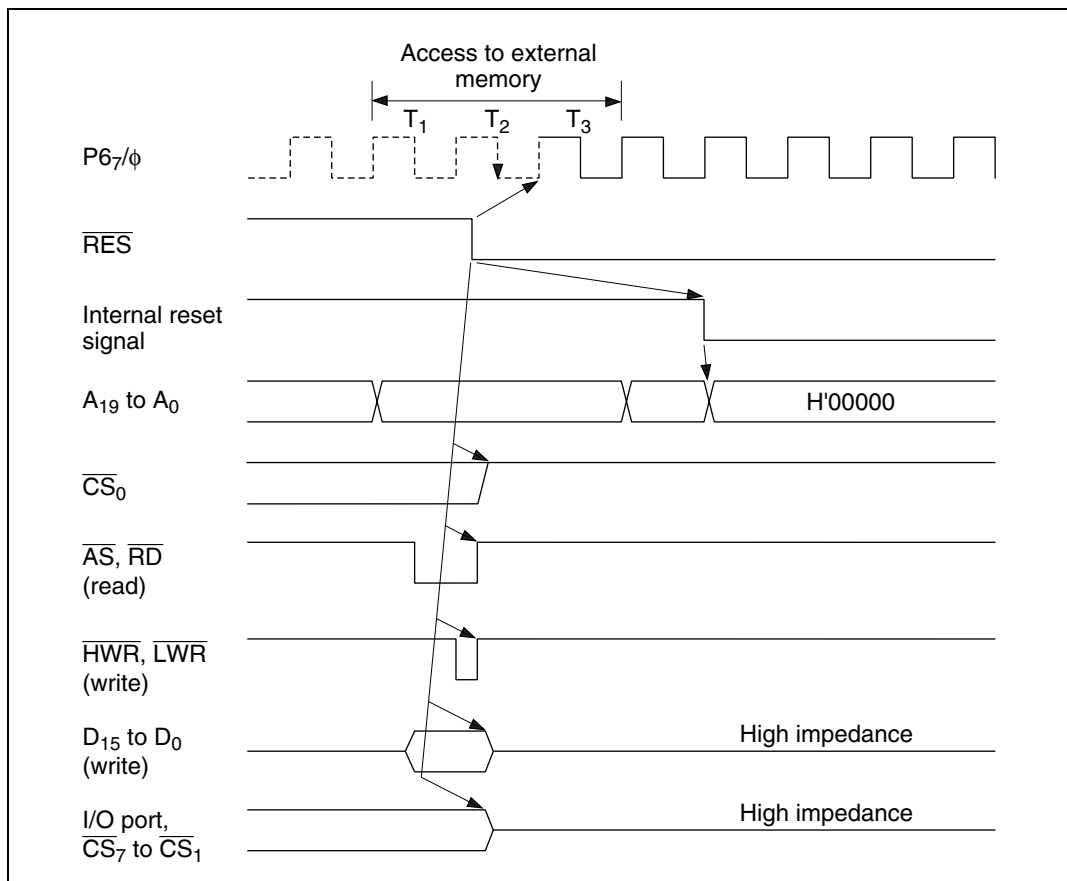


Figure D.1 Reset during Memory Access (Modes 1 and 2)

Modes 3 and 4: Figure D.2 is a timing diagram for the case in which $\overline{\text{RES}}$ goes low during an external memory access in mode 3 or 4. As soon as $\overline{\text{RES}}$ goes low, all ports are initialized to the input state. $\overline{\text{AS}}$, $\overline{\text{RD}}$, $\overline{\text{HWR}}$, $\overline{\text{LWR}}$, and $\overline{\text{CS}}_0$ go high, and D_{15} to D_0 go to the high-impedance state. The address bus is initialized to the low output level 2.5ϕ clock cycles after the low level of $\overline{\text{RES}}$ is sampled. However, when PA_4 to PA_6 are used as address bus pins, or when P8_3 to P8_1 and PB_0 to PB_3 are used as CS output pins, they go to the high-impedance state at the same time as $\overline{\text{RES}}$ goes low. Clock pin $\text{P6}_7/\phi$ goes to the output state at the next rise of ϕ after $\overline{\text{RES}}$ goes low.

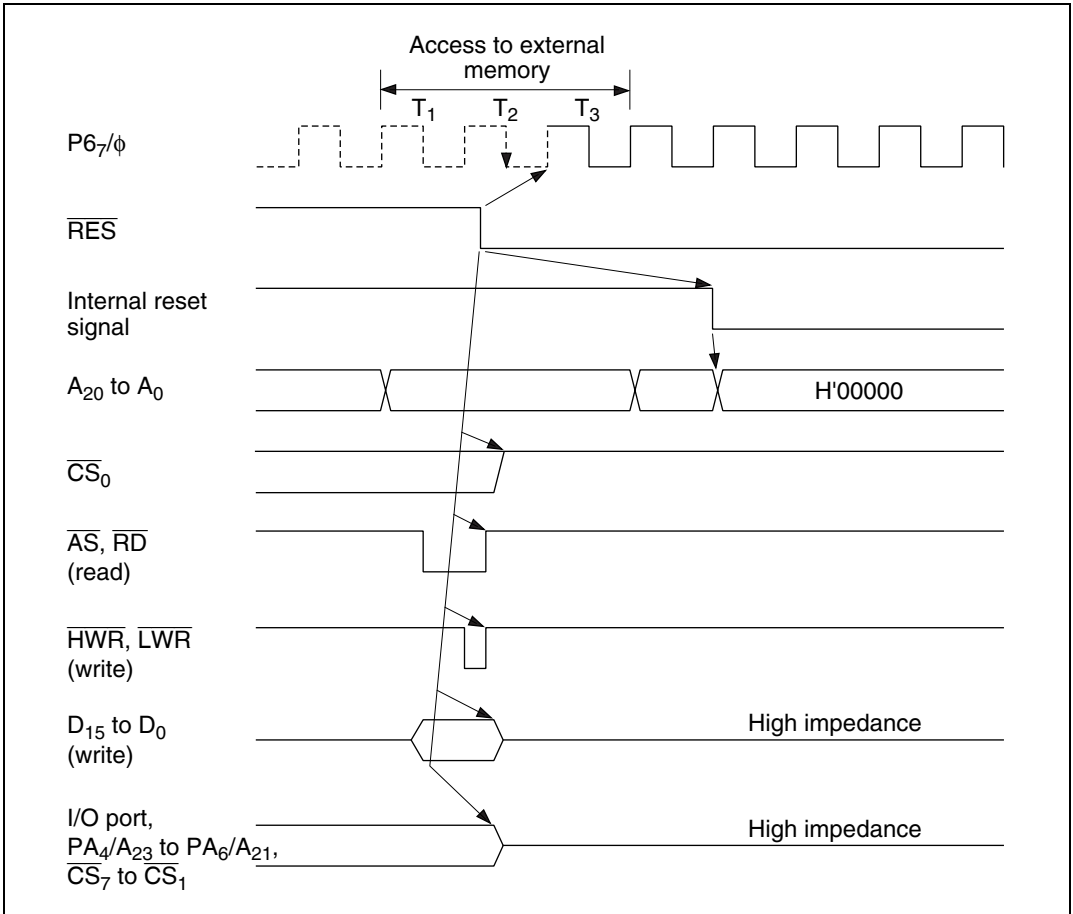
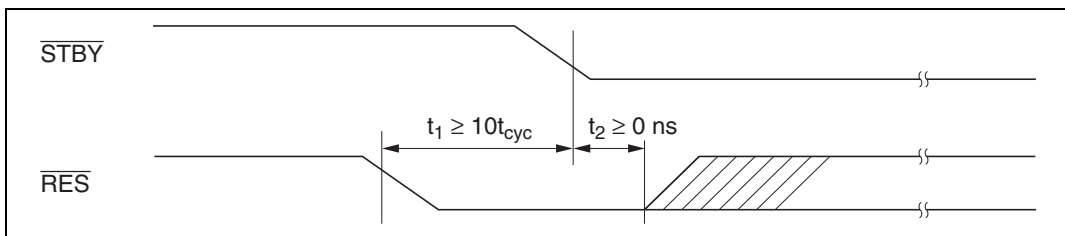


Figure D.2 Reset during Memory Access (Modes 3 and 4)

Appendix E Timing of Transition to and Recovery from Hardware Standby Mode

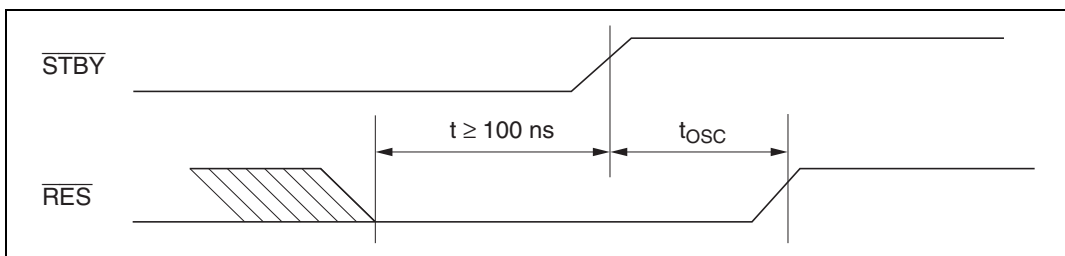
Timing of Transition to Hardware Standby Mode

1. To retain RAM contents with the RAME bit set to 1 in SYSCR, drive the $\overline{\text{RES}}$ signal low 10 system clock cycles before the $\overline{\text{STBY}}$ signal goes low, as shown below. $\overline{\text{RES}}$ must remain low until $\overline{\text{STBY}}$ goes low (minimum delay from $\overline{\text{STBY}}$ low to $\overline{\text{RES}}$ high: 0 ns).



2. To retain RAM contents with the RAME bit cleared to 0 in SYSCR, $\overline{\text{RES}}$ does not have to be driven low as in (1).

Timing of Recovery from Hardware Standby Mode: Drive the $\overline{\text{RES}}$ signal low approximately 100 ns before $\overline{\text{STBY}}$ goes high.



Appendix F Product Code Lineup

Table F.1 H8/3008 Product Code Lineup

| Product Type | | | Product Code | Mark Code | Package (Package Code) |
|---------------------|---------|-----|---------------------|------------------|-------------------------------|
| H8/3008 | ROMless | 5 V | HD6413008F | HD6413008F | 100-pin QFP (FP-100B) |
| | | | HD6413008TE | HD6413008TE | 100-pin TQFP (TFP-100B) |
| | | 3 V | HD6413008VF | HD6413008VF | 100-pin QFP (FP-100B) |
| | | | HD6413008VTE | HD6413008VTE | 100-pin TQFP (TFP-100B) |

Appendix G Package Dimensions

Figure G.1 shows the FP-100B package dimensions of the H8/3008. Figure G.2 shows the TFP-100B package dimensions.

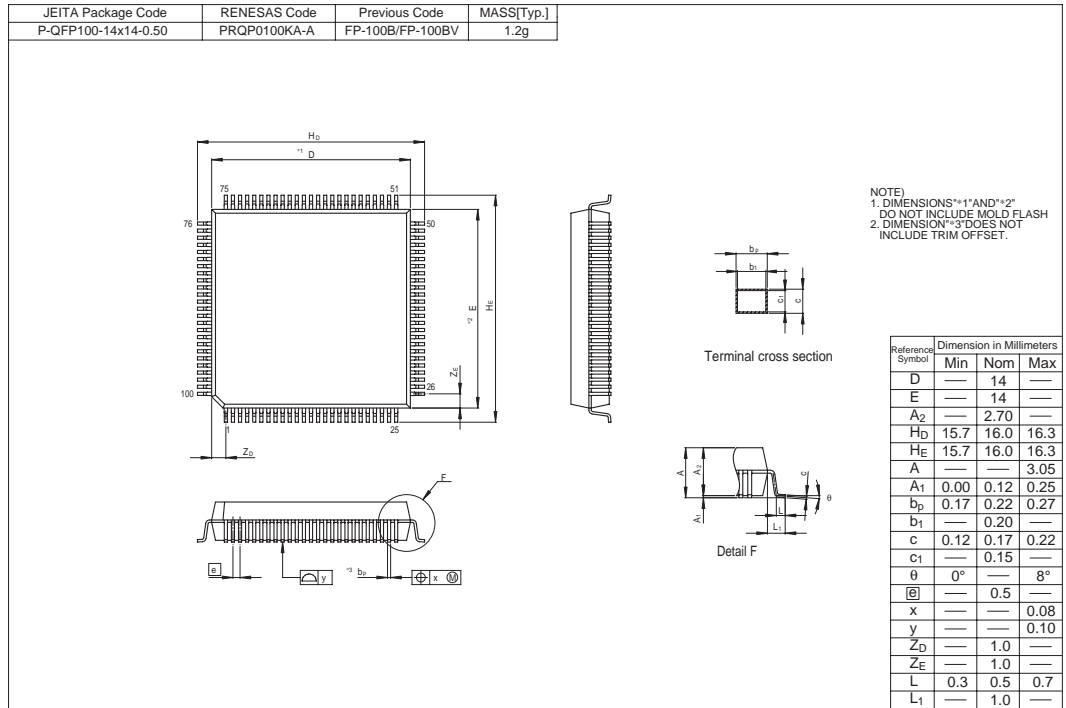
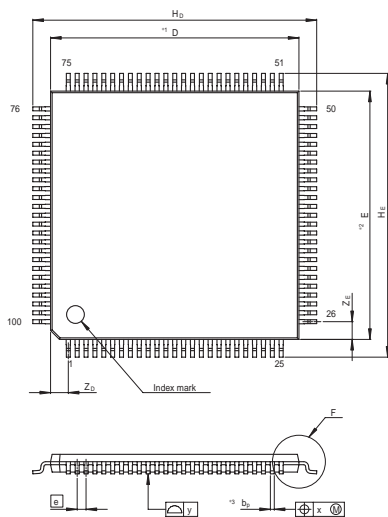


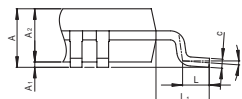
Figure G.1 Package Dimensions (FP-100B)

Appendix G Package Dimensions

| | | | |
|----------------------|--------------|--------------------|------------|
| JEITA Package Code | RENESAS Code | Previous Code | MASS[Typ.] |
| P-TQFP100-14x14-0.50 | PTQP0100KA-A | TFP-100B/TFP-100BV | 0.5g |



Terminal cross section



Detail F

NOTE)
 1. DIMENSIONS *1* AND *2* DO NOT INCLUDE MOLD FLASH
 2. DIMENSION *3* DOES NOT INCLUDE TRIM OFFSET.

| Reference Symbol | Dimension in Millimeters | | |
|------------------|--------------------------|------|------|
| | Min | Nom | Max |
| D | — | 14 | — |
| E | — | 14 | — |
| A ₂ | — | 1.00 | — |
| H _D | 15.8 | 16.0 | 16.2 |
| H _E | 15.8 | 16.0 | 16.2 |
| A | — | — | 1.20 |
| A ₁ | 0.00 | 0.10 | 0.20 |
| b _p | 0.17 | 0.22 | 0.27 |
| b ₁ | — | 0.20 | — |
| c | 0.12 | 0.17 | 0.22 |
| c ₁ | — | 0.15 | — |
| θ | 0° | — | 8° |
| e | — | 0.5 | — |
| x | — | — | 0.08 |
| y | — | — | 0.10 |
| Z _D | — | 1.00 | — |
| Z _E | — | 1.00 | — |
| L | 0.4 | 0.5 | 0.6 |
| L ₁ | — | 1.0 | — |

Figure G.2 Package Dimensions (TFP-100B)

Appendix H Comparison of H8/300H Series Product Specifications

H.1 Differences between H8/3067 and H8/3062 Group, H8/3048 Group, H8/3006 and H8/3007, and H8/3008

| Item | | H8/3067 Group, H8/3062 Group | H8/3048 Group | H8/3006, H8/3007 | H8/3008 | |
|------|-----------------------|--|--|---------------------|--------------------|-------------------------------|
| 1 | Operating Mode 5 mode | 16 Mbyte ROM enabled expanded mode | 1 Mbyte ROM enabled expanded mode | | | |
| | Mode 6 | 64 kbyte single-chip mode | 16 Mbyte ROM enabled expanded mode | | | |
| 2 | Interrupt controller | Internal interrupt sources 36 (H8/3067) 27 (H8/3062 Group) | 30 | 36 | 27 | |
| 3 | Bus controller | Burst ROM interface Yes (H8/3067) No (H8/3062 Group) | No | Yes | No | |
| | | Idle cycle insertion function | Yes | No | Yes | |
| | | Wait mode | 2 modes | 4 modes | 2 modes | 2 modes |
| | | Wait state number setting | Per area | Common to all areas | Per area | Per area |
| | | Address output method | Choice of address update mode (fixed in H8/3067F-ZTAT and H8/3062F-ZTAT) | Fixed | Fixed | Choice of address update mode |
| 4 | DRAM interface | Connectable areas Area 2/3/4/5 (H8/3067 only) | Area 3 | Area 2/3/4/5 | No | |
| | | Precharge cycle insertion function | Yes (H8/3067 only) | No | Yes | No |
| | | Fast page mode | Yes (H8/3067 only) | No | Yes | No |
| | | Address shift amount | 8 bit/9 bit/10 bit (H8/3067 only) | 8 bit/9 bit | 8 bit/9 bit/10 bit | No |

| Item | H8/3067 Group, H8/3062 Group | | H8/3048 Group | H8/3006, H8/3007 | | H8/3008 | | |
|------|---------------------------------|---------------------------------------|---|---------------------------|----------------------------|-----------------------------|----------------------------|-----------------------------|
| | 16-bit timers | 8-bit timers | ITU | 16-bit timers | 8-bit timers | 16-bit timers | 8-bit timers | |
| 5 | Timer functions | | | | | | | |
| | Number of channels | 16 bits × 3 | 8 bits × 4 (16 bits × 2) | 16 bits × 5 | 16 bits × 3 | 8 bits × 4 (16 bits × 2) | 16 bits × 3 | 8 bits × 4 (16 bits × 2) |
| | Pulse output | 6 pins | 4 pins (2 pins) | 12 pins | 6 pins | 4 pins (2 pins) | 6 pins | 4 pins (2 pins) |
| | Input capture | 6 | 2 | 10 | 6 | 2 | 6 | 2 |
| | External clock | 4 systems (selectable) | 4 systems (fixed) | 4 systems (selectable) | 4 systems (selectable) | 4 systems (fixed) | 4 systems (selectable) | 4 systems (fixed) |
| | Internal clock | φ, φ/2, φ/4, φ/8 | φ/8, φ/64, φ/8192 | φ, φ/2, φ/4, φ/8 | φ, φ/2, φ/4, φ/8 | φ/8, φ/64, φ/8192 | φ, φ/2, φ/4, φ/8 | φ/8, φ/64, φ/8192 |
| | Complementary PWM function | No | No | Yes | No | No | No | No |
| | Reset-synchronous PWM function | No | No | Yes | No | No | No | No |
| | Buffer operation | No | No | Yes | No | No | No | No |
| | Output initialization function | Yes | No | No | Yes | No | Yes | No |
| | PWM output | 3 | 4 (2) | 5 | 3 | 4 (2) | 3 | 4 (2) |
| | DMAC activation | 3 channels (H8/3067 only) | No | 4 channels | 3 channels | No | No | No |
| | A/D conversion activation | No | Yes | No | No | Yes | Yes | Yes |
| | Interrupt sources | 3 sources × 3 | 8 sources | 3 sources × 5 | 3 sources × 3 | 8 sources | 8 sources | 8 sources |
| 6 | TPC | Time base | 3 kinds, 16-bit timer base | 4 kinds, ITU base | 3 kinds, 16-bit timer base | 3 kinds, 16-bit timer base | 3 kinds, 16-bit timer base | 3 kinds, 16-bit timer base |
| 7 | WDT | Reset signal external output function | Yes (except products with on-chip flash memory) | Yes | Yes | Yes | Yes | Yes |

| Item | | H8/3067 Group, H8/3062 Group | H8/3048 Group | H8/3006, H8/3007 | H8/3008 | |
|------|------------------------|--|--|--|---|---|
| 8 | SCI | Number of channels | 3 channels (H8/3067) 2 channels (H8/3062 Group) | 2 channels | 3 channels | 2 channels |
| | | Smart card interface | Supported on all channels | Supported on SCI0 only | Supported on all channels | Supported on all channels |
| 9 | A/D converter | Conversion start trigger input | External trigger/8-bit timer compare match | External trigger | External trigger/8-bit timer compare match | External trigger/8-bit timer compare match |
| | | Conversion states | 70/134 | 134/266 | 70/134 | 70/134 |
| 10 | Pin control | ϕ pin | ϕ /input port multiplexing | ϕ output only | ϕ /input port multiplexing | ϕ output/input port |
| | | A ₂₀ in 16 MB ROM enabled expanded mode | A ₂₀ / I/O port multiplexing | A ₂₀ output | | |
| | | Address bus, AS, RD, HWR, LWR, CS ₇ –CS ₀ , RFSH in software standby state | High-level output/high-impedance selectable (RFSH: H8/3067 only) | High-level output (except CS ₀) Low-level output (CS ₀) | High-level output/high-impedance selectable | High-level output/high-impedance selectable |
| | | CS ₇ –CS ₀ in bus-released state | High-impedance | High-level output | High-impedance | High-impedance |
| 11 | Flash memory functions | Program/erase voltage | 12 V application unnecessary. Single-power-supply programming. | 12 V application from off-chip | | |
| | | Block divisions | 8 blocks (12 blocks in H8/3064F-ZTAT B-mask version) | 16 blocks | | |

H.2 Comparison of Pin Functions of 100-Pin Package Products (FP-100B, TFP-100B)

Table H.1 Pin Arrangement of Each Product (FP-100B, TFP-100B)

| Pin No. | On-chip-ROM Products | | | | ROMless Products | |
|---------|--|---|---|---|--|---|
| | H8/3067 Group | H8/3062 Group | H8/3048 Group | H8/3042 Group | H8/3006, H8/3007 | H8/3008 |
| 1 | V _{CC} | V _{CC} /V _{CL} * ² | V _{CC} | V _{CC} | V _{CC} | V _{CC} /V _{CL} * ² |
| 2 | PB ₀ /TP ₈ /TMO ₀ / CS ₇ | PB ₀ /TP ₈ /TMO ₀ / CS ₇ | PB ₀ /TP ₈ / TIOCA ₃ | PB ₀ /TP ₈ / TIOCA ₃ | PB ₀ /TP ₈ /TMO ₀ / CS ₇ | PB ₀ /TP ₈ /TMO ₀ / CS ₇ |
| 3 | PB ₁ /TP ₉ /TMIO ₁ / DREQ ₀ /CS ₆ | PB ₁ /TP ₉ /TMIO ₁ / CS ₆ | PB ₁ /TP ₉ / TIOCB ₃ | PB ₁ /TP ₉ / TIOCB ₃ | PB ₁ /TP ₉ /TMIO ₁ / DREQ ₀ /CS ₆ | PB ₁ /TP ₉ /TMIO ₁ / CS ₆ |
| 4 | PB ₂ /TP ₁₀ /TMO ₂ / CS ₅ | PB ₂ /TP ₁₀ /TMO ₂ / CS ₅ | PB ₂ /TP ₁₀ / TIOCA ₄ | PB ₂ /TP ₁₀ / TIOCA ₄ | PB ₂ /TP ₁₀ /TMO ₂ / CS ₅ | PB ₂ /TP ₁₀ /TMO ₂ / CS ₅ |
| 5 | PB ₃ /TP ₁₁ / TMIO ₃ /DREQ ₁ / CS ₄ | PB ₃ /TP ₁₁ / TMIO ₃ /CS ₄ | PB ₃ /TP ₁₁ / TIOCB ₄ | PB ₃ /TP ₁₁ / TIOCB ₄ | PB ₃ /TP ₁₁ / TMIO ₃ /DREQ ₁ / CS ₄ | PB ₃ /TP ₁₁ / TMIO ₃ /CS ₄ |
| 6 | PB ₄ /TP ₁₂ / UCAS | PB ₄ /TP ₁₂ | PB ₄ /TP ₁₂ / TOCXA ₄ | PB ₄ /TP ₁₂ / TOCXA ₄ | PB ₄ /TP ₁₂ / UCAS | PB ₄ /TP ₁₂ |
| 7 | PB ₅ /TP ₁₃ / LCAS/SCK ₂ | PB ₅ /TP ₁₃ | PB ₅ /TP ₁₃ / TOCXB ₄ | PB ₅ /TP ₁₃ / TOCXB ₄ | PB ₅ /TP ₁₃ / LCAS/SCK ₂ | PB ₅ /TP ₁₃ |
| 8 | PB ₆ /TP ₁₄ /TxD ₂ | PB ₆ /TP ₁₄ | PB ₆ /TP ₁₄ / DREQ ₀ /CS ₇ | PB ₆ /TP ₁₄ / DREQ ₀ | PB ₆ /TP ₁₄ /TxD ₂ | PB ₆ /TP ₁₄ |
| 9 | PB ₇ /TP ₁₅ /RxD ₂ | PB ₇ /TP ₁₅ | PB ₇ /TP ₁₅ / DREQ ₁ /ADTRG | PB ₇ /TP ₁₅ / DREQ ₁ /ADTRG | PB ₇ /TP ₁₅ /RxD ₂ | PB ₇ /TP ₁₅ |
| 10 | RESO/FWE* ¹ | RESO/FWE* ¹ | RESO/V _{PP} | RESO | RESO | NC/RESO |
| 11 | V _{SS} | V _{SS} | V _{SS} | V _{SS} | V _{SS} | V _{SS} |
| 12 | P9 ₀ /TxD ₀ | P9 ₀ /TxD ₀ | P9 ₀ /TxD ₀ | P9 ₀ /TxD ₀ | P9 ₀ /TxD ₀ | P9 ₀ /TxD ₀ |
| 13 | P9 ₁ /TxD ₁ | P9 ₁ /TxD ₁ | P9 ₁ /TxD ₁ | P9 ₁ /TxD ₁ | P9 ₁ /TxD ₁ | P9 ₁ /TxD ₁ |
| 14 | P9 ₂ /RxD ₀ | P9 ₂ /RxD ₀ | P9 ₂ /RxD ₀ | P9 ₂ /RxD ₀ | P9 ₂ /RxD ₀ | P9 ₂ /RxD ₀ |
| 15 | P9 ₃ /RxD ₁ | P9 ₃ /RxD ₁ | P9 ₃ /RxD ₁ | P9 ₃ /RxD ₁ | P9 ₃ /RxD ₁ | P9 ₃ /RxD ₁ |
| 16 | P9 ₄ /SCK ₀ /IRQ ₄ | P9 ₄ /SCK ₀ /IRQ ₄ | P9 ₄ /SCK ₀ /IRQ ₄ | P9 ₄ /SCK ₀ /IRQ ₄ | P9 ₄ /SCK ₀ /IRQ ₄ | P9 ₄ /SCK ₀ /IRQ ₄ |
| 17 | P9 ₅ /SCK ₁ /IRQ ₅ | P9 ₅ /SCK ₁ /IRQ ₅ | P9 ₅ /SCK ₁ /IRQ ₅ | P9 ₅ /SCK ₁ /IRQ ₅ | P9 ₅ /SCK ₁ /IRQ ₅ | P9 ₅ /SCK ₁ /IRQ ₅ |
| 18 | P4 ₀ /D ₀ | P4 ₀ /D ₀ | P4 ₀ /D ₀ | P4 ₀ /D ₀ | P4 ₀ /D ₀ | P4 ₀ /D ₀ |
| 19 | P4 ₁ /D ₁ | P4 ₁ /D ₁ | P4 ₁ /D ₁ | P4 ₁ /D ₁ | P4 ₁ /D ₁ | P4 ₁ /D ₁ |
| 20 | P4 ₂ /D ₂ | P4 ₂ /D ₂ | P4 ₂ /D ₂ | P4 ₂ /D ₂ | P4 ₂ /D ₂ | P4 ₂ /D ₂ |
| 21 | P4 ₃ /D ₃ | P4 ₃ /D ₃ | P4 ₃ /D ₃ | P4 ₃ /D ₃ | P4 ₃ /D ₃ | P4 ₃ /D ₃ |
| 22 | V _{SS} | V _{SS} | V _{SS} | V _{SS} | V _{SS} | V _{SS} |
| 23 | P4 ₄ /D ₄ | P4 ₄ /D ₄ | P4 ₄ /D ₄ | P4 ₄ /D ₄ | P4 ₄ /D ₄ | P4 ₄ /D ₄ |

| Pin No. | On-chip-ROM Products | | | | ROMless Products | |
|---------|----------------------------------|----------------------------------|----------------------------------|----------------------------------|---------------------------------|---------------------------------|
| | H8/3067 Group | H8/3062 Group | H8/3048 Group | H8/3042 Group | H8/3006, H8/3007 | H8/3008 |
| 24 | P4 ₅ /D ₅ | P4 ₅ /D ₅ | P4 ₅ /D ₅ | P4 ₅ /D ₅ | P4 ₅ /D ₅ | P4 ₅ /D ₅ |
| 25 | P4 ₆ /D ₆ | P4 ₆ /D ₆ | P4 ₆ /D ₆ | P4 ₆ /D ₆ | P4 ₆ /D ₆ | P4 ₆ /D ₆ |
| 26 | P4 ₇ /D ₇ | P4 ₇ /D ₇ | P4 ₇ /D ₇ | P4 ₇ /D ₇ | P4 ₇ /D ₇ | P4 ₇ /D ₇ |
| 27 | P3 ₀ /D ₈ | P3 ₀ /D ₈ | P3 ₀ /D ₈ | P3 ₀ /D ₈ | D ₈ | D ₈ |
| 28 | P3 ₁ /D ₉ | P3 ₁ /D ₉ | P3 ₁ /D ₉ | P3 ₁ /D ₉ | D ₉ | D ₉ |
| 29 | P3 ₂ /D ₁₀ | P3 ₂ /D ₁₀ | P3 ₂ /D ₁₀ | P3 ₂ /D ₁₀ | D ₁₀ | D ₁₀ |
| 30 | P3 ₃ /D ₁₁ | P3 ₃ /D ₁₁ | P3 ₃ /D ₁₁ | P3 ₃ /D ₁₁ | D ₁₁ | D ₁₁ |
| 31 | P3 ₄ /D ₁₂ | P3 ₄ /D ₁₂ | P3 ₄ /D ₁₂ | P3 ₄ /D ₁₂ | D ₁₂ | D ₁₂ |
| 32 | P3 ₅ /D ₁₃ | P3 ₅ /D ₁₃ | P3 ₅ /D ₁₃ | P3 ₅ /D ₁₃ | D ₁₃ | D ₁₃ |
| 33 | P3 ₆ /D ₁₄ | P3 ₆ /D ₁₄ | P3 ₆ /D ₁₄ | P3 ₆ /D ₁₄ | D ₁₄ | D ₁₄ |
| 34 | P3 ₇ /D ₁₅ | P3 ₇ /D ₁₅ | P3 ₇ /D ₁₅ | P3 ₇ /D ₁₅ | D ₁₅ | D ₁₅ |
| 35 | Vcc | Vcc | Vcc | Vcc | Vcc | Vcc |
| 36 | P1 ₀ /A ₀ | P1 ₀ /A ₀ | P1 ₀ /A ₀ | P1 ₀ /A ₀ | A ₀ | A ₀ |
| 37 | P1 ₁ /A ₁ | P1 ₁ /A ₁ | P1 ₁ /A ₁ | P1 ₁ /A ₁ | A ₁ | A ₁ |
| 38 | P1 ₂ /A ₂ | P1 ₂ /A ₂ | P1 ₂ /A ₂ | P1 ₂ /A ₂ | A ₂ | A ₂ |
| 39 | P1 ₃ /A ₃ | P1 ₃ /A ₃ | P1 ₃ /A ₃ | P1 ₃ /A ₃ | A ₃ | A ₃ |
| 40 | P1 ₄ /A ₄ | P1 ₄ /A ₄ | P1 ₄ /A ₄ | P1 ₄ /A ₄ | A ₄ | A ₄ |
| 41 | P1 ₅ /A ₅ | P1 ₅ /A ₅ | P1 ₅ /A ₅ | P1 ₅ /A ₅ | A ₅ | A ₅ |
| 42 | P1 ₆ /A ₆ | P1 ₆ /A ₆ | P1 ₆ /A ₆ | P1 ₆ /A ₆ | A ₆ | A ₆ |
| 43 | P1 ₇ /A ₇ | P1 ₇ /A ₇ | P1 ₇ /A ₇ | P1 ₇ /A ₇ | A ₇ | A ₇ |
| 44 | Vss | Vss | Vss | Vss | Vss | Vss |
| 45 | P2 ₀ /A ₈ | P2 ₀ /A ₈ | P2 ₀ /A ₈ | P2 ₀ /A ₈ | A ₈ | A ₈ |
| 46 | P2 ₁ /A ₉ | P2 ₁ /A ₉ | P2 ₁ /A ₉ | P2 ₁ /A ₉ | A ₉ | A ₉ |
| 47 | P2 ₂ /A ₁₀ | P2 ₂ /A ₁₀ | P2 ₂ /A ₁₀ | P2 ₂ /A ₁₀ | A ₁₀ | A ₁₀ |
| 48 | P2 ₃ /A ₁₁ | P2 ₃ /A ₁₁ | P2 ₃ /A ₁₁ | P2 ₃ /A ₁₁ | A ₁₁ | A ₁₁ |
| 49 | P2 ₄ /A ₁₂ | P2 ₄ /A ₁₂ | P2 ₄ /A ₁₂ | P2 ₄ /A ₁₂ | A ₁₂ | A ₁₂ |
| 50 | P2 ₅ /A ₁₃ | P2 ₅ /A ₁₃ | P2 ₅ /A ₁₃ | P2 ₅ /A ₁₃ | A ₁₃ | A ₁₃ |
| 51 | P2 ₆ /A ₁₄ | P2 ₆ /A ₁₄ | P2 ₆ /A ₁₄ | P2 ₆ /A ₁₄ | A ₁₄ | A ₁₄ |
| 52 | P2 ₇ /A ₁₅ | P2 ₇ /A ₁₅ | P2 ₇ /A ₁₅ | P2 ₇ /A ₁₅ | A ₁₅ | A ₁₅ |
| 53 | P5 ₀ /A ₁₆ | P5 ₀ /A ₁₆ | P5 ₀ /A ₁₆ | P5 ₀ /A ₁₆ | A ₁₆ | A ₁₆ |
| 54 | P5 ₁ /A ₁₇ | P5 ₁ /A ₁₇ | P5 ₁ /A ₁₇ | P5 ₁ /A ₁₇ | A ₁₇ | A ₁₇ |
| 55 | P5 ₂ /A ₁₈ | P5 ₂ /A ₁₈ | P5 ₂ /A ₁₈ | P5 ₂ /A ₁₈ | A ₁₈ | A ₁₈ |
| 56 | P5 ₃ /A ₁₉ | P5 ₃ /A ₁₉ | P5 ₃ /A ₁₉ | P5 ₃ /A ₁₉ | A ₁₉ | A ₁₉ |

| Pin No. | On-chip-ROM Products | | | | ROMless Products | |
|---------|--|--|--|--|--|--|
| | H8/3067 Group | H8/3062 Group | H8/3048 Group | H8/3042 Group | H8/3006, H8/3007 | H8/3008 |
| 57 | V _{SS} | V _{SS} | V _{SS} | V _{SS} | V _{SS} | V _{SS} |
| 58 | P6 ₀ /WAIT | P6 ₀ /WAIT | P6 ₀ /WAIT | P6 ₀ /WAIT | P6 ₀ /WAIT | P6 ₀ /WAIT |
| 59 | P6 ₁ /BREQ | P6 ₁ /BREQ | P6 ₁ /BREQ | P6 ₁ /BREQ | P6 ₁ /BREQ | P6 ₁ /BREQ |
| 60 | P6 ₂ /BACK | P6 ₂ /BACK | P6 ₂ /BACK | P6 ₂ /BACK | P6 ₂ /BACK | P6 ₂ /BACK |
| 61 | P6 ₇ /φ | P6 ₇ /φ | φ | φ | P6 ₇ /φ | P6 ₇ /φ |
| 62 | STBY | STBY | STBY | STBY | STBY | STBY |
| 63 | RES | RES | RES | RES | RES | RES |
| 64 | NMI | NMI | NMI | NMI | NMI | NMI |
| 65 | V _{SS} | V _{SS} | V _{SS} | V _{SS} | V _{SS} | V _{SS} |
| 66 | EXTAL | EXTAL | EXTAL | EXTAL | EXTAL | EXTAL |
| 67 | XTAL | XTAL | XTAL | XTAL | XTAL | XTAL |
| 68 | V _{CC} | V _{CC} | V _{CC} | V _{CC} | V _{CC} | V _{CC} |
| 69 | P6 ₃ /AS | P6 ₃ /AS | P6 ₃ /AS | P6 ₃ /AS | AS | AS |
| 70 | P6 ₄ /RD | P6 ₄ /RD | P6 ₄ /RD | P6 ₄ /RD | RD | RD |
| 71 | P6 ₅ /HWR | P6 ₅ /HWR | P6 ₅ /HWR | P6 ₅ /HWR | HWR | HWR |
| 72 | P6 ₆ /LWR | P6 ₆ /LWR | P6 ₆ /LWR | P6 ₆ /LWR | LWR | LWR |
| 73 | MD ₀ | MD ₀ | MD ₀ | MD ₀ | MD ₀ | MD ₀ |
| 74 | MD ₁ | MD ₁ | MD ₁ | MD ₁ | MD ₁ | MD ₁ |
| 75 | MD ₂ | MD ₂ | MD ₂ | MD ₂ | MD ₂ | MD ₂ |
| 76 | AV _{CC} | AV _{CC} | AV _{CC} | AV _{CC} | AV _{CC} | AV _{CC} |
| 77 | V _{REF} | V _{REF} | V _{REF} | V _{REF} | V _{REF} | V _{REF} |
| 78 | P7 ₀ /AN ₀ | P7 ₀ /AN ₀ | P7 ₀ /AN ₀ | P7 ₀ /AN ₀ | P7 ₀ /AN ₀ | P7 ₀ /AN ₀ |
| 79 | P7 ₁ /AN ₁ | P7 ₁ /AN ₁ | P7 ₁ /AN ₁ | P7 ₁ /AN ₁ | P7 ₁ /AN ₁ | P7 ₁ /AN ₁ |
| 80 | P7 ₂ /AN ₂ | P7 ₂ /AN ₂ | P7 ₂ /AN ₂ | P7 ₂ /AN ₂ | P7 ₂ /AN ₂ | P7 ₂ /AN ₂ |
| 81 | P7 ₃ /AN ₃ | P7 ₃ /AN ₃ | P7 ₃ /AN ₃ | P7 ₃ /AN ₃ | P7 ₃ /AN ₃ | P7 ₃ /AN ₃ |
| 82 | P7 ₄ /AN ₄ | P7 ₄ /AN ₄ | P7 ₄ /AN ₄ | P7 ₄ /AN ₄ | P7 ₄ /AN ₄ | P7 ₄ /AN ₄ |
| 83 | P7 ₅ /AN ₅ | P7 ₅ /AN ₅ | P7 ₅ /AN ₅ | P7 ₅ /AN ₅ | P7 ₅ /AN ₅ | P7 ₅ /AN ₅ |
| 84 | P7 ₆ /AN ₆ /DA ₀ | P7 ₆ /AN ₆ /DA ₀ | P7 ₆ /AN ₆ /DA ₀ | P7 ₆ /AN ₆ /DA ₀ | P7 ₆ /AN ₆ /DA ₀ | P7 ₆ /AN ₆ /DA ₀ |
| 85 | P7 ₇ /AN ₇ /DA ₁ | P7 ₇ /AN ₇ /DA ₁ | P7 ₇ /AN ₇ /DA ₁ | P7 ₇ /AN ₇ /DA ₁ | P7 ₇ /AN ₇ /DA ₁ | P7 ₇ /AN ₇ /DA ₁ |
| 86 | AV _{SS} | AV _{SS} | AV _{SS} | AV _{SS} | AV _{SS} | AV _{SS} |
| 87 | P8 ₀ /RFSH/IRQ ₀ | P8 ₀ /IRQ ₀ | P8 ₀ /RFSH/IRQ ₀ | P8 ₀ /RFSH/IRQ ₀ | P8 ₀ /RFSH/IRQ ₀ | P8 ₀ /IRQ ₀ |
| 88 | P8 ₁ /CS ₃ /IRQ ₁ | P8 ₁ /CS ₃ /IRQ ₁ | P8 ₁ /CS ₃ /IRQ ₁ | P8 ₁ /CS ₃ /IRQ ₁ | P8 ₁ /CS ₃ /IRQ ₁ | P8 ₁ /CS ₃ /IRQ ₁ |
| 89 | P8 ₂ /CS ₂ /IRQ ₂ | P8 ₂ /CS ₂ /IRQ ₂ | P8 ₂ /CS ₂ /IRQ ₂ | P8 ₂ /CS ₂ /IRQ ₂ | P8 ₂ /CS ₂ /IRQ ₂ | P8 ₂ /CS ₂ /IRQ ₂ |

| Pin No. | On-chip-ROM Products | | | | ROMless Products | |
|---------|--|--|--|--|--|--|
| | H8/3067 Group | H8/3062 Group | H8/3048 Group | H8/3042 Group | H8/3006, H8/3007 | H8/3008 |
| 90 | $P8_3/\overline{CS}_1/\overline{IRQ}_3/\overline{ADTRG}$ | $P8_3/\overline{CS}_1/\overline{IRQ}_3/\overline{ADTRG}$ | $P8_3/\overline{CS}_1/\overline{IRQ}_3$ | $P8_3/\overline{CS}_1/\overline{IRQ}_3$ | $P8_3/\overline{CS}_1/\overline{IRQ}_3/\overline{ADTRG}$ | $P8_3/\overline{CS}_1/\overline{IRQ}_3/\overline{ADTRG}$ |
| 91 | $P8_4/\overline{CS}_0$ | $P8_4/\overline{CS}_0$ | $P8_4/\overline{CS}_0$ | $P8_4/\overline{CS}_0$ | $P8_4/\overline{CS}_0$ | $P8_4/\overline{CS}_0$ |
| 92 | Vss | Vss | Vss | Vss | Vss | Vss |
| 93 | $PA_0/\overline{TP}_0/\overline{TEND}_0/\overline{TCLKA}$ | $PA_0/\overline{TP}_0/\overline{TCLKA}$ | $PA_0/\overline{TP}_0/\overline{TEND}_0/\overline{TCLKA}$ | $PA_0/\overline{TP}_0/\overline{TEND}_0/\overline{TCLKA}$ | $PA_0/\overline{TP}_0/\overline{TEND}_0/\overline{TCLKA}$ | $PA_0/\overline{TP}_0/\overline{TCLKA}$ |
| 94 | $PA_1/\overline{TP}_1/\overline{TEND}_1/\overline{TCLKB}$ | $PA_1/\overline{TP}_1/\overline{TCLKB}$ | $PA_1/\overline{TP}_1/\overline{TEND}_1/\overline{TCLKB}$ | $PA_1/\overline{TP}_1/\overline{TEND}_1/\overline{TCLKB}$ | $PA_1/\overline{TP}_1/\overline{TEND}_1/\overline{TCLKB}$ | $PA_1/\overline{TP}_1/\overline{TCLKB}$ |
| 95 | $PA_2/\overline{TP}_2/\overline{TIOCA}_2/\overline{TCLKC}$ | $PA_2/\overline{TP}_2/\overline{TIOCA}_2/\overline{TCLKC}$ | $PA_2/\overline{TP}_2/\overline{TIOCA}_2/\overline{TCLKC}$ | $PA_2/\overline{TP}_2/\overline{TIOCA}_2/\overline{TCLKC}$ | $PA_2/\overline{TP}_2/\overline{TIOCA}_2/\overline{TCLKC}$ | $PA_2/\overline{TP}_2/\overline{TIOCA}_2/\overline{TCLKC}$ |
| 96 | $PA_3/\overline{TP}_3/\overline{TIOCB}_3/\overline{TCLKD}$ | $PA_3/\overline{TP}_3/\overline{TIOCB}_3/\overline{TCLKD}$ | $PA_3/\overline{TP}_3/\overline{TIOCB}_3/\overline{TCLKD}$ | $PA_3/\overline{TP}_3/\overline{TIOCB}_3/\overline{TCLKD}$ | $PA_3/\overline{TP}_3/\overline{TIOCB}_3/\overline{TCLKD}$ | $PA_3/\overline{TP}_3/\overline{TIOCB}_3/\overline{TCLKD}$ |
| 97 | $PA_4/\overline{TP}_4/\overline{TIOCA}_4/A_{23}$ | $PA_4/\overline{TP}_4/\overline{TIOCA}_4/A_{23}$ | $PA_4/\overline{TP}_4/\overline{TIOCA}_4/\overline{CS}_6/A_{23}$ | $PA_4/\overline{TP}_4/\overline{TIOCA}_4/A_{23}$ | $PA_4/\overline{TP}_4/\overline{TIOCA}_4/A_{23}$ | $PA_4/\overline{TP}_4/\overline{TIOCA}_4/A_{23}$ |
| 98 | $PA_5/\overline{TP}_5/\overline{TIOCB}_5/A_{22}$ | $PA_5/\overline{TP}_5/\overline{TIOCB}_5/A_{22}$ | $PA_5/\overline{TP}_5/\overline{TIOCB}_5/\overline{CS}_5/A_{22}$ | $PA_5/\overline{TP}_5/\overline{TIOCB}_5/A_{22}$ | $PA_5/\overline{TP}_5/\overline{TIOCB}_5/A_{22}$ | $PA_5/\overline{TP}_5/\overline{TIOCB}_5/A_{22}$ |
| 99 | $PA_6/\overline{TP}_6/\overline{TIOCA}_6/A_{21}$ | $PA_6/\overline{TP}_6/\overline{TIOCA}_6/A_{21}$ | $PA_6/\overline{TP}_6/\overline{TIOCA}_6/\overline{CS}_4/A_{21}$ | $PA_6/\overline{TP}_6/\overline{TIOCA}_6/A_{21}$ | $PA_6/\overline{TP}_6/\overline{TIOCA}_6/A_{21}$ | $PA_6/\overline{TP}_6/\overline{TIOCA}_6/A_{21}$ |
| 100 | $PA_7/\overline{TP}_7/\overline{TIOCB}_7/A_{20}$ | $PA_7/\overline{TP}_7/\overline{TIOCB}_7/A_{20}$ | $PA_7/\overline{TP}_7/\overline{TIOCB}_7/A_{20}$ | $PA_7/\overline{TP}_7/\overline{TIOCB}_7/A_{20}$ | $PA_7/\overline{TP}_7/\overline{TIOCB}_7/A_{20}$ | $PA_7/\overline{TP}_7/\overline{TIOCB}_7/A_{20}$ |

Notes: 1. Functions as \overline{RESO} in the mask ROM versions, and as FWE in the on-chip flash memory versions.

2. The 5 V operation models of the H8/3064F-ZTAT B-mask version and the H8/3062F-ZTAT B-mask version have a V_{CL} pin, and require an external capacitor (0.1 μ F).

**Renesas 16-Bit Single-chip Microcomputer
Hardware Manual
H8/3008**

Publication Date: 1st Edition, September 2000
Rev.4.00, August 20, 2007
Published by: Sales Strategic Planning Div.
Renesas Technology Corp.
Edited by: Customer Support Department
Global Strategic Communication Div.
Renesas Solutions Corp.

Renesas Technology Corp. Sales Strategic Planning Div. Nippon Bldg., 2-6-2, Ohte-machi, Chiyoda-ku, Tokyo 100-0004, Japan



RENESAS SALES OFFICES

<http://www.renesas.com>

Refer to "<http://www.renesas.com/en/network>" for the latest and detailed information.

Renesas Technology America, Inc.

450 Holger Way, San Jose, CA 95134-1368, U.S.A
Tel: <1> (408) 382-7500, Fax: <1> (408) 382-7501

Renesas Technology Europe Limited

Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K.
Tel: <44> (1628) 585-100, Fax: <44> (1628) 585-900

Renesas Technology (Shanghai) Co., Ltd.

Unit 204, 205, AZIACenter, No.1233 Lujiazui Ring Rd, Pudong District, Shanghai, China 200120
Tel: <86> (21) 5877-1818, Fax: <86> (21) 6887-7898

Renesas Technology Hong Kong Ltd.

7th Floor, North Tower, World Finance Centre, Harbour City, 1 Canton Road, Tsimshatsui, Kowloon, Hong Kong
Tel: <852> 2265-6688, Fax: <852> 2730-6071

Renesas Technology Taiwan Co., Ltd.

10th Floor, No.99, Fushing North Road, Taipei, Taiwan
Tel: <886> (2) 2715-2888, Fax: <886> (2) 2713-2999

Renesas Technology Singapore Pte. Ltd.

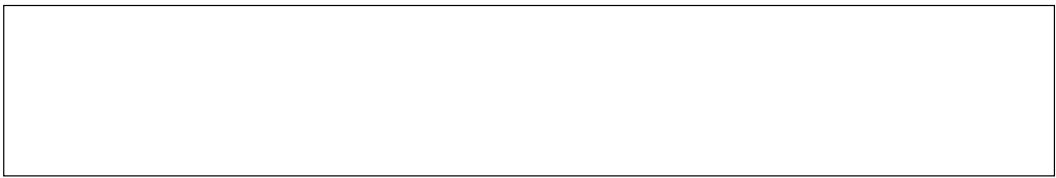
1 Harbour Front Avenue, #06-10, Keppel Bay Tower, Singapore 098632
Tel: <65> 6213-0200, Fax: <65> 6278-8001

Renesas Technology Korea Co., Ltd.

Kukje Center Bldg. 18th Fl., 191, 2-ka, Hangang-ro, Yongsan-ku, Seoul 140-702, Korea
Tel: <82> (2) 796-3115, Fax: <82> (2) 796-2145

Renesas Technology Malaysia Sdn. Bhd

Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No.18, Jalan Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: <603> 7955-9390, Fax: <603> 7955-9510



H8/3008 Hardware Manual



Renesas Electronics Corporation

1753, Shimonumabe, Nakahara-ku, Kawasaki-shi, Kanagawa 211-8668 Japan

REJ09B0395-0400