XS1-U6A-64-FB96 Datasheet

2015/04/14 Document Number: XM002430,

Table of Contents

TO OUR VALUED CUSTOMERS

It is our intention to provide you with accurate and comprehensive documentation for the hardware and software components used in this product. To subscribe to receive updates, visit http://www.xmos.com/.

1 xCORE Multicore Microcontrollers

The XS1-U Series is a comprehensive range of 32-bit multicore microcontrollers that brings the low latency and timing determinism of the xCORE architecture to mainstream embedded applications. Unlike conventional microcontrollers, xCORE multicore microcontrollers execute multiple real-time tasks simultaneously and communicate between tasks using a high speed network. Because xCORE multicore microcontrollers are completely deterministic, you can write software to implement functions that traditionally require dedicated hardware.

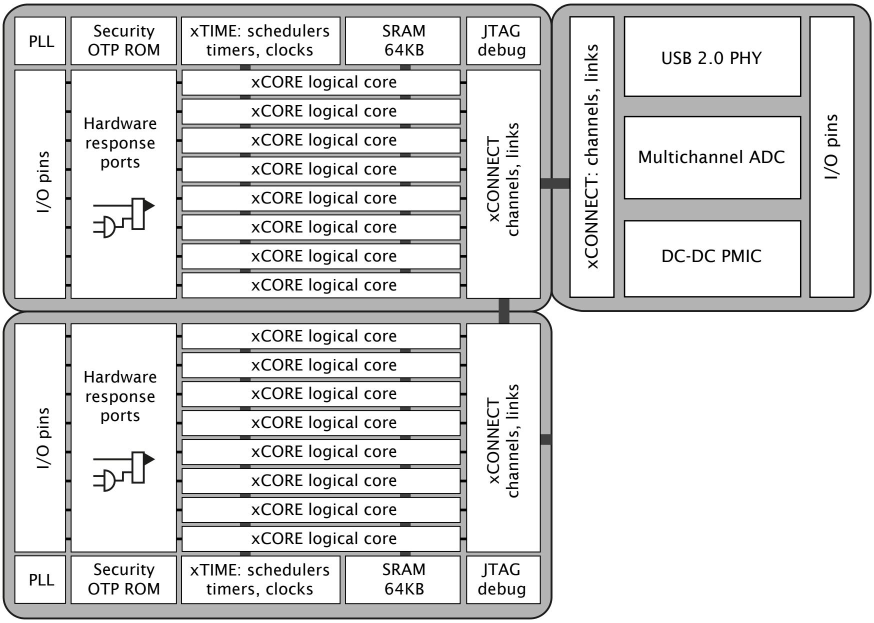


Figure 1: XS1-U Series: 6-16 core devices

Key features of the XS1-U6A-64-FB96 include:

- **Tiles**: Devices consist of one or more xCORE tiles. Each tile contains between four and eight 32-bit xCOREs with highly integrated I/O and on-chip memory.
- **Logical cores** Each logical core can execute tasks such as computational code, DSP code, control software (including logic decisions and executing a state machine) or software that handles I/O. Section 7.1
- **xTIME scheduler** The xTIME scheduler performs functions similar to an RTOS, in hardware. It services and synchronizes events in a core, so there is no requirement for interrupt handler routines. The xTIME scheduler triggers cores on events generated by hardware resources such as the I/O pins, communication channels and timers. Once triggered, a core runs independently and concurrently to other cores, until it pauses to wait for more events. Section 7.2

- **Channels and channel ends** Tasks running on logical cores communicate using channels formed between two channel ends. Data can be passed synchronously or asynchronously between the channel ends assigned to the communicating tasks. Section 7.5
- **xCONNECT Switch and Links** Between tiles, channel communications are implemented over a high performance network of xCONNECT Links and routed through a hardware xCONNECT Switch. Section 7.6
- **Ports** The I/O pins are connected to the processing cores by Hardware Response ports. The port logic can drive its pins high and low, or it can sample the value on its pins optionally waiting for a particular condition. Section 7.3
- **Clock blocks** xCORE devices include a set of programmable clock blocks that can be used to govern the rate at which ports execute. Section 7.4
- **Memory** Each xCORE Tile integrates a bank of SRAM for instructions and data, and a block of one-time programmable (OTP) memory that can be configured for system wide security features. Section 10
- **PLL** The PLL is used to create a high-speed processor clock given a low speed external oscillator. Section 8
- **USB** The USB PHY provides High-Speed and Full-Speed, device, host, and on-the-go functionality. Data is communicated through ports on the digital node. A library is provided to implement USB device functionality. Section 11
- **JTAG** The JTAG module can be used for loading programs, boundary scan testing, in-circuit source-level debugging and programming the OTP memory. Section 15

1.1 Software

Devices are programmed using C, C++ or xC (C with multicore extensions). XMOS provides tested and proven software libraries, which allow you to quickly add interface and processor functionality such as USB, Ethernet, PWM, graphics driver, and audio EQ to your applications.

1.2 xTIMEcomposer Studio

The xTIMEcomposer Studio development environment provides all the tools you need to write and debug your programs, profile your application, and write images into flash memory or OTP memory on the device. Because xCORE devices operate deterministically, they can be simulated like hardware within xTIMEcomposer: uniquely in the embedded world, xTIMEcomposer Studio therefore includes a static timing analyzer, cycle-accurate simulator, and high-speed in-circuit instrumentation.

xTIMEcomposer can be driven from either a graphical development environment, or the command line. The tools are supported on Windows, Linux and MacOS X and available at no cost from xmos.com/downloads. Information on using the tools is provided in the xTIMEcomposer User Guide, X3766.

2 XS1-U6A-64-FB96 Features

- **Multicore Microcontroller with Advanced Multi-Core RISC Architecture**
  - Six real-time logical cores
  - Core share up to 500 MIPS
  - Each logical core has:
    - Guaranteed throughput of between 1/4 and 1/6 of tile MIPS
    - 16x32bit dedicated registers
  - 159 high-density 16/32-bit instructions
    - All have single clock-cycle execution (except for divide)
    - 32x32→64-bit MAC instructions for DSP, arithmetic and user-definable cryptographic functions
- **USB PHY, fully compliant with USB 2.0 specification**
- **12b 1MSPS 4-channel SAR Analog-to-Digital Converter**
- **1 x LDO**
- **2 x DC-DC converters and Power Management Unit**
- **Watchdog Timer**
- **Onchip clocks/oscillators**
  - Crystal oscillator
  - 20MHz/31kHz silicon oscillators
- **Programmable I/O**
  - 38 general-purpose I/O pins, configurable as input or output
    - Up to 9 x 1bit port, 2 x 4bit port, 1 x 8bit port
    - 3 xCONNECT links
  - Port sampling rates of up to 60 MHz with respect to an external clock
  - 32 channel ends for communication with other cores, on or off-chip
- **Memory**
  - 64KB internal single-cycle SRAM for code and data storage
  - 8KB internal OTP for application boot code
  - 128 bytes Deep Sleep Memory
- **Hardware resources**
  - 6 clock blocks
  - 10 timers
  - 4 locks
- **JTAG Module for On-Chip Debug**
- **Security Features**
  - Programming lock disables debug and prevents read-back of memory contents
  - AES bootloader ensures secrecy of IP held on external flash memory
- **Ambient Temperature Range**
  - Commercial qualification: 0 °C to 70 °C
  - Industrial qualification: -40 °C to 85 °C
- **Speed Grade**
  - 5: 500 MIPS
- **Power Consumption with USB running (typical)**
  - 300 mW (typical)
  - Sleep Mode: 500 µW
- **96-pin FBGA package 0.8 mm pitch**

3 Pin Configuration

	1	2	3	4	5	6	7	8	9	10	11	12
A	AVDD	ADC0	ADC2	NC	USB_DP	USB_DN	USB_VBUS	1L X0D35	1A X0D00	1C X0D10	1E X0D12	32A X0D49
B	TDO	ADC1	ADC3	NC	MODE[2]	MODE[3]	USB_ID	1I X0D24	1B X0D01	1D X0D11	32A X0D50	32A X0D51
C	TCK	RST_N									32A X0D52	32A X0D53
D	TMS	TDI									32A X0D54	32A X0D55
E	XI/CLK	DEBUG_N			AVSS	GND	GND	GND			32A X0D56	32A X0D57
F	XO	OSC_EXT_N			GND	GND	GND	GND			32A X0D58	32A X0D61
G	8D X0D43/WAKE	NC			GND	GND	GND	GND			32A X0D62	32A X0D63
H	VSUP	NC			GND	GND	GND	GND			32A X0D64	32A X0D65
J	SW1	SW1									32A X0D66	32A X0D67
K	VDDCORE	VDDCORE									32A X0D68	32A X0D69
L	PGND	PGND	NC	MODE[1]	MODE[0]	VDDIO	1G X0D22	4C X0D20	4D X0D18	4D X0D16	4C X0D14	32A X0D70
M	VSUP	VSUP	PGND	VDD1V8	SW2	VDDIO	VDDIO	4C X0D21	4D X0D19	4D X0D17	4C X0D15	1F X0D13

4 Signal Description

This section lists the signals and I/O pins available on the XS1-U6A-64-FB96. The device provides a combination of 1bit, 4bit, 8bit and 16bit ports, as well as wider ports that are fully or partially (gray) bonded out. All pins of a port provide either output or input, but signals in different directions cannot be mapped onto the same port.

Pins may have one or more of the following properties:

- PD/PU: The IO pin a weak pull-down or pull-up resistor. On GPIO pins this resistor can be enabled.
- ST: The IO pin has a Schmitt Trigger on its input.

Power pins (9)

Signal	**Function**	**Type**	**Properties**
AVSS	Digital ground	GND	
GND	Digital ground	GND	
PGND	Power ground	GND	
SW1	DCDC1 switched output voltage	PWR	
SW2	DCDC2 switched output voltage	PWR	
VDD1V8	1v8 voltage supply	PWR	
VDDCORE	Core voltage supply	PWR	
VDDIO	Digital I/O power	PWR	
VSUP	Power supply (3V3/5V0)	PWR	

Analog pins (5)

Signal	**Function**	**Type**	**Properties**
ADC0	Analog input	Input	
ADC1	Analog input	Input	
ADC2	Analog input	Input	
ADC3	Analog input	Input	
AVDD	Supply and reference voltage	PWR	

USB pins (4)

Signal	**Function**	**Type**	**Properties**
USB_DN	USB Serial Data Inverted	I/O	
USB_DP	USB Serial Data	I/O	
USB_ID	USB Device ID (OTG) - Reserved	Output	
USB_VBUS	USB Power Detect Pin	Input	

Clocks pins (4)			
Signal	**Function**	**Type**	**Properties**
MODE[3:0]	Boot mode select	Input	PU, ST
OSC_EXT_N	Use Silicon Oscillator	Input	ST
XI/CLK	Crystal Oscillator/Clock Input	Input	
XO	Crystal Oscillator Output	Output	

JTAG pins (5)			
Signal	**Function**	**Type**	**Properties**
DEBUG_N	Multi-chip debug	I/O	PU
TCK	Test clock	Input	PU, ST
TDI	Test data input	Input	PU, ST
TDO	Test data output	Output	PD, OT
TMS	Test mode select	Input	PU, ST

Misc pins (1)			
Signal	**Function**	**Type**	**Properties**
RST_N	Global reset input	Input	PU, ST

I/O pins (38)								
Signal	**Function**						**Type**	**Properties**
X0D00		$1A^0$					I/O	PD_S, R_S
X0D01		$1B^0$					I/O	PD_S, R_S
X0D10		$1C^0$					I/O	PD_S, R_S
X0D11		$1D^0$					I/O	PD_S, R_S
X0D12		$1E^0$					I/O	PD_S
X0D13	XLB^4_{out}	$1F^0$					I/O	PD_S
X0D14	XLB^3_{out}		$4C^0$	$8B^0$	$16A^8$	$32A^{28}$	I/O	PD_S
X0D15	XLB^2_{out}		$4C^1$	$8B^1$	$16A^9$	$32A^{29}$	I/O	PD_S
X0D16	XLB^1_{out}		$4D^0$	$8B^2$	$16A^{10}$		I/O	PD_S
X0D17	XLB^0_{out}		$4D^1$	$8B^3$	$16A^{11}$		I/O	PD_S
X0D18	XLB^0_{in}		$4D^2$	$8B^4$	$16A^{12}$		I/O	PD_S
X0D19	XLB^1_{in}		$4D^3$	$8B^5$	$16A^{13}$		I/O	PD_S
X0D20	XLB^2_{in}		$4C^2$	$8B^6$	$16A^{14}$	$32A^{30}$	I/O	PD_S
X0D21	XLB^3_{in}		$4C^3$	$8B^7$	$16A^{15}$	$32A^{31}$	I/O	PD_S
X0D22	XLB^4_{in}	$1G^0$					I/O	PD_S
X0D24		$1I^0$					I/O	PD_S
X0D35		$1L^0$					I/O	PD_S
X0D43/WAKE				$8D^7$	$16B^{15}$		I/O	PU_S
X0D49	XLC^4_{out}					$32A^0$	I/O	PD_S

(continued)

Signal	Function		Type	Properties
X0D50	XLC^3_{out}	$32A^1$	I/O	PD_S
X0D51	XLC^2_{out}	$32A^2$	I/O	PD_S
X0D52	XLC^1_{out}	$32A^3$	I/O	PD_S
X0D53	XLC^0_{out}	$32A^4$	I/O	PD_S
X0D54	XLC^0_{in}	$32A^5$	I/O	PD_S
X0D55	XLC^1_{in}	$32A^6$	I/O	PD_S
X0D56	XLC^2_{in}	$32A^7$	I/O	PD_S
X0D57	XLC^3_{in}	$32A^8$	I/O	PD_S
X0D58	XLC^4_{in}	$32A^9$	I/O	PD_S
X0D61	XLD^4_{out}	$32A^{10}$	I/O	PD_S
X0D62	XLD^3_{out}	$32A^{11}$	I/O	PD_S
X0D63	XLD^2_{out}	$32A^{12}$	I/O	PD_S
X0D64	XLD^1_{out}	$32A^{13}$	I/O	PD_S
X0D65	XLD^0_{out}	$32A^{14}$	I/O	PD_S
X0D66	XLD^0_{in}	$32A^{15}$	I/O	PD_S
X0D67	XLD^1_{in}	$32A^{16}$	I/O	PD_S
X0D68	XLD^2_{in}	$32A^{17}$	I/O	PD_S
X0D69	XLD^3_{in}	$32A^{18}$	I/O	PD_S
X0D70	XLD^4_{in}	$32A^{19}$	I/O	PD_S

5 Example Application Diagram

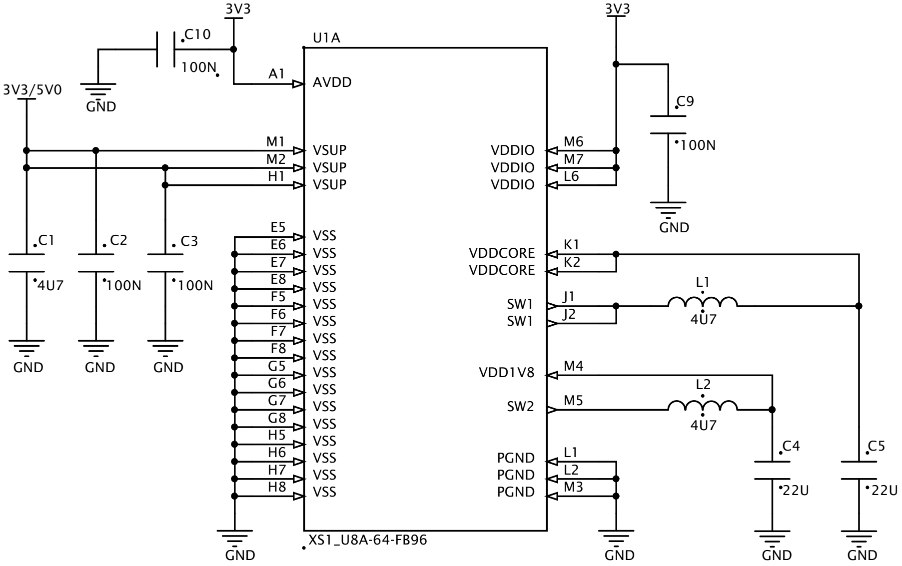


Figure 2: Simplified Reference Schematic

6 Product Overview

The XS1-U6A-64-FB96 comprises a digital and an analog node, as shown in Figure 3. The digital node comprises an xCORE Tile, a Switch, and a PLL (Phase-locked-loop). The analog node comprises the USB PHY, a multi-channel ADC (Analog to Digital Converter), deep sleep memory, an oscillator, a real-time counter, and power supply control.

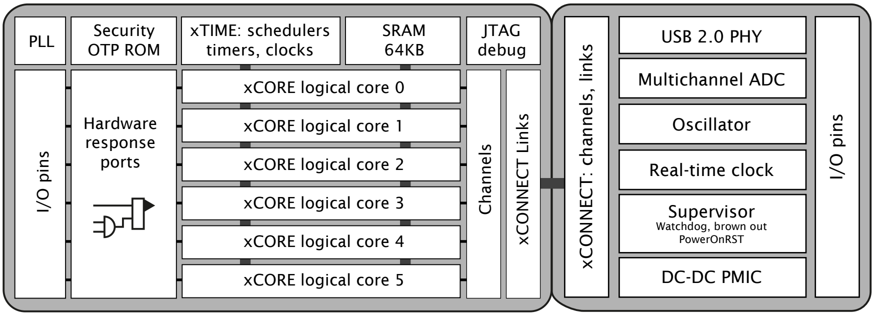


Figure 3: Block Diagram

All communication between the digital and analog node takes place over a link that is connected to the Switch of the digital node. As such, the analog node can be controlled from any node on the system. The analog functions can be configured using a set of node configuration registers, and a set of registers for each of the peripherals.

The device can be programmed using high-level languages such as C/C++ and the XMOS-originated XC language, which provides extensions to C that simplify the control over concurrency, I/O and timing, or low-level assembler.

6.1 XCore Tile

The xCORE Tile is a flexible multicore microcontroller component with tightly integrated I/O and on-chip memory. The tile contains multiple logical cores that run simultaneously, each of which is guaranteed a slice of processing power and can execute computational code, control software and I/O interfaces. The logical cores use channels to exchange data within a tile or across tiles. Multiple devices can be deployed and connected using an integrated switching network, enabling more resources to be added to a design. The I/O pins are driven using intelligent ports that can serialize data, interpret strobe signals and wait for scheduled times or events, making the device ideal for real-time control applications.

6.2 USB PHY

The USB PHY is fully compliant with the USB 2.0 specification. It supports high speed (480-Mbps) and full speed (12Mbps) operation.

The XMOS XUD software component performs all the low-level I/O operations required to meet the USB 2.0 specification, removing all low-level timing requirements from the application.

6.3 ADC and Power Management

Each XS1-U6A-64-FB96 device includes a set of analog components, including a 12b, 4-channel ADC, power management unit, watchdog timer, real-time counter and deep sleep memory. The device reduces the number of additional external components required and allows designs to be implemented using simple 2-layer boards.

7 xCORE Tile Resources

7.1 Logical cores

The tile has 6 active logical cores, which issue instructions down a shared four-stage pipeline. Instructions from the active cores are issued round-robin. If up to four logical cores are active, each core is allocated a quarter of the processing cycles. If more than four logical cores are active, each core is allocated at least $^1/_n$ cycles (for n cores). Figure 4 shows the guaranteed core performance depending on the number of cores used.

Figure 4: Logical core performance

Speed grade	MIPS	Frequency	Minimum MIPS per core (for *n* cores)							
			1	2	3	4	5	6		
5	500 MIPS	500 MHz	125	125	125	125	100	83		

There is no way that the performance of a logical core can be reduced below these predicted levels. Because cores may be delayed on I/O, however, their unused processing cycles can be taken by other cores. This means that for more than four logical cores, the performance of each core is often higher than the predicted minimum but cannot be guaranteed.

The logical cores are triggered by events instead of interrupts and run to completion. A logical core can be paused to wait for an event.

7.2 xTIME scheduler

The xTIME scheduler handles the events generated by xCORE Tile resources, such as channel ends, timers and I/O pins. It ensures that all events are serviced and synchronized, without the need for an RTOS. Events that occur at the I/O pins are handled by the Hardware-Response ports and fed directly to the appropriate xCORE Tile. An xCORE Tile can also choose to wait for a specified time to elapse, or for data to become available on a channel.

Tasks do not need to be prioritised as each of them runs on their own logical xCORE. It is possible to share a set of low priority tasks on a single core using cooperative multitasking.

7.3 Hardware Response Ports

Hardware Response ports connect an xCORE tile to one or more physical pins and as such define the interface between hardware attached to the XS1-U6A-64-FB96, and the software running on it. A combination of 1bit, 4bit, 8bit, 16bit and 32bit ports are available. All pins of a port provide either output or input. Signals in different directions cannot be mapped onto the same port.

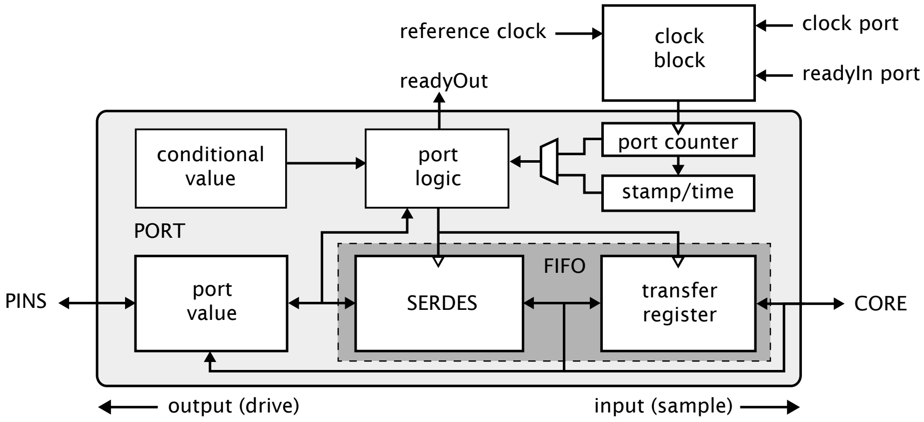


Figure 5: Port block diagram

The port logic can drive its pins high or low, or it can sample the value on its pins, optionally waiting for a particular condition. Ports are accessed using dedicated instructions that are executed in a single processor cycle.

Data is transferred between the pins and core using a FIFO that comprises a SERDES and transfer register, providing options for serialization and buffered data.

Each port has a 16-bit counter that can be used to control the time at which data is transferred between the port value and transfer register. The counter values can be obtained at any time to find out when data was obtained, or used to delay I/O until some time in the future. The port counter value is automatically saved as a timestamp, that can be used to provide precise control of response times.

The ports and xCONNECT links are multiplexed onto the physical pins. If an xConnect Link is enabled, the pins of the underlying ports are disabled. If a port is enabled, it overrules ports with higher widths that share the same pins. The pins on the wider port that are not shared remain available for use when the narrower port is enabled. Ports always operate at their specified width, even if they share pins with another port.

7.4 Clock blocks

xCORE devices include a set of programmable clocks called clock blocks that can be used to govern the rate at which ports execute. Each xCORE tile has six clock blocks: the first clock block provides the tile reference clock and runs at a default frequency of 100MHz; the remaining clock blocks can be set to run at different frequencies.

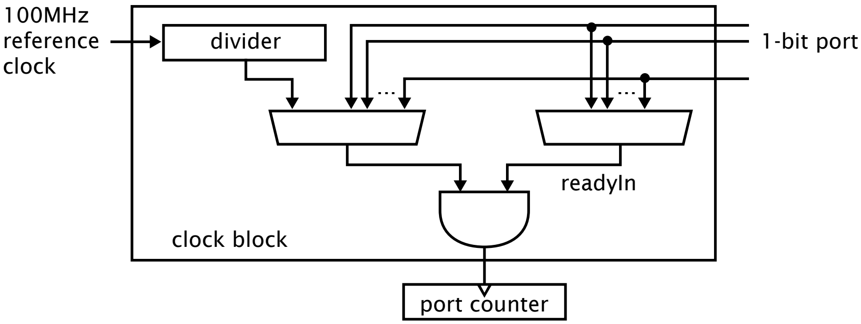


Figure 6: Clock block diagram

A clock block can use a 1-bit port as its clock source allowing external application clocks to be used to drive the input and output interfaces.

In many cases I/O signals are accompanied by strobing signals. The xCORE ports can input and interpret strobe (known as readyIn and readyOut) signals generated by external sources, and ports can generate strobe signals to accompany output data.

On reset, each port is connected to clock block 0, which runs from the processor reference clock.

7.5 Channels and Channel Ends

Logical cores communicate using point-to-point connections, formed between two channel ends. A channel-end is a resource on an xCORE tile, that is allocated by the program. Each channel-end has a unique system-wide identifier that comprises a unique number and their tile identifier. Data is transmitted to a channel-end by an output-instruction; and the other side executes an input-instruction. Data can be passed synchronously or asynchronously between the channel ends.

7.6 xCONNECT Switch and Links

XMOS devices provide a scalable architecture, where multiple xCORE devices can be connected together to form one system. Each xCORE device has an xCONNECT interconnect that provides a communication infrastructure for all tasks that run on the various xCORE tiles on the system.

The interconnect relies on a collection of switches and XMOS links. Each xCORE device has an on-chip switch that can set up circuits or route data. The switches are connected by xConnect Links. An XMOS link provides a physical connection between two switches. The switch has a routing algorithm that supports many different topologies, including lines, meshes, trees, and hypercubes.

The links operate in either 2 wires per direction or 5 wires per direction mode, depending on the amount of bandwidth required. Circuit switched, streaming

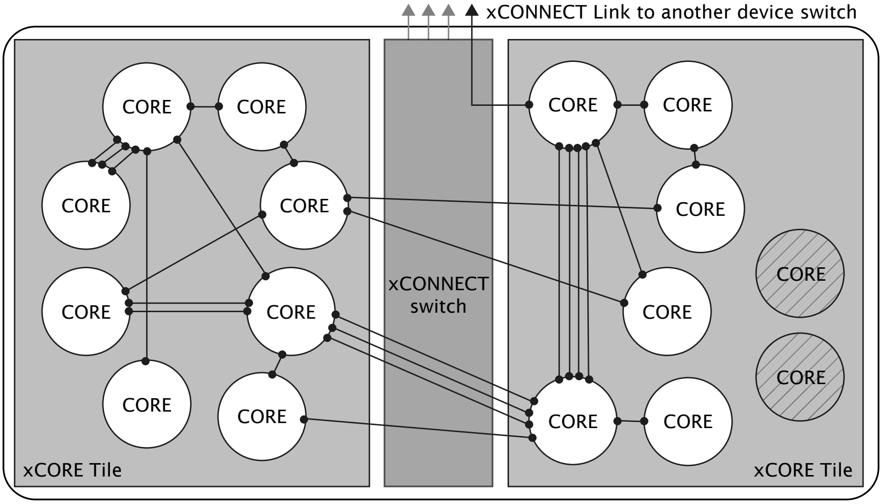


Figure 7: Switch, links and channel ends

and packet switched data can both be supported efficiently. Streams provide the fastest possible data rates between tiles (up to 313 MBit/s), but each stream requires a single link to be reserved between switches on two tiles. All packet communications can be multiplexed onto a single link.

Information on the supported routing topologies that can be used to connect multiple devices together can be found in the XS1-L Link Performance and Design Guide, X2999.

8 Oscillator

The oscillator block provides:

- An oscillator circuit. Together with an external resonator (crystal or ceramic), the oscillator circuit can provide a clock-source for both the real-time counter and the xCORE Tile. The external resonator can be chosen by the designer to have the appropriate frequency and accuracy. If desired, an external oscillator can be used on the XI/CLK input pin, this must be a 1.8 V oscillator.
- A 20 MHz silicon oscillator. This enables the device to boot and execute code without requiring an external crystal. The silicon oscillator is not as accurate as an external crystal.
- A 31,250 Hz oscillator. This enables the real-time counter to operate whilst the device is in low-power mode. This oscillator is not as accurate as an external crystal.

The oscillator can be controlled through package pins, a set of peripheral registers, and a digital node control register.

A package pin OSC_EXT_N is used to select the oscillator to use on boot. It must be grounded to select an external resonator or connected to VDDIO to select the on-chip 20 MHz oscillator. If an external resonator is used, then it must be in the range 5-100 MHz. If the USB PHY is used, then an external crystal (12 or 24 MHz) or an external oscillator (12, 24, 48, or 96 MHz) is required in order to provide a stable USB clock. Two more package pins, MODE0 and MODE1 are used to inform the node of the frequency.

The analog node runs at the frequency provided by the oscillator. Hence, increasing the clock frequency will speed up operation of the analog node, and will speed up communicating data with the digital node. The digital node has a PLL.

The PLL creates a high-speed clock that is used for the switch, tile, and reference clock.

The PLL multiplication value is selected through the two MODE pins, and can be changed by software to speed up the tile or use less power. The MODE pins are set as shown in Figure 8:

Figure 8: PLL multiplier values and MODE pins

Oscillator Frequency	MODE 1	MODE 0	Tile Frequency	PLL Ratio	PLL settings OD	F	R
5-13 MHz	0	0	130-399.75 MHz	30.75	1	122	0
13-20 MHz	1	1	260-400.00 MHz	20	2	119	0
20-48 MHz	1	0	167-400.00 MHz	8.33	2	49	0
48-100 MHz	0	1	196-400.00 MHz	4	2	23	0

Figure 8 also lists the values of OD, F and R, which are the registers that define the ratio of the tile frequency to the oscillator frequency:

$$F_{core} = F_{osc} \times \frac{F+1}{2} \times \frac{1}{R+1} \times \frac{1}{OD+1}$$

OD, F and R must be chosen so that $0 \leq R \leq 63$, $0 \leq F \leq 4095$, $0 \leq OD \leq 7$, and $260MHz \leq F_{osc} \times \frac{F+1}{2} \times \frac{1}{R+1} \leq 1.3GHz$. The OD, F, and R values can be modified by writing to the digital node PLL configuration register.

The MODE pins must be held at a static value during and after deassertion of the system reset.

If a different tile frequency is required (eg, 500 MHz), then the PLL must be reprogrammed after boot to provide the required tile frequency. The XMOS tools perform this operation by default. Further details on configuring the clock can be found in the XS1-L Clock Frequency Control document, X1433.

9 Boot Procedure

The device is kept in reset by driving RST_N low. When in reset, all GPIO pins are high impedance. When the device is taken out of reset by releasing RST_N the processor starts its internal reset process. After approximately 750,000 input clocks, all GPIO pins have their internal pull-resistor enabled, and the processor boots at a clock speed that depends on MODE0 and MODE1.

The processor boot procedure is illustrated in Figure 9. In normal usage, MODE[3:2] controls the boot source according to the table in Figure 10. If bit 5 of the security register (*see* §10.1) is set, the device boots from OTP.

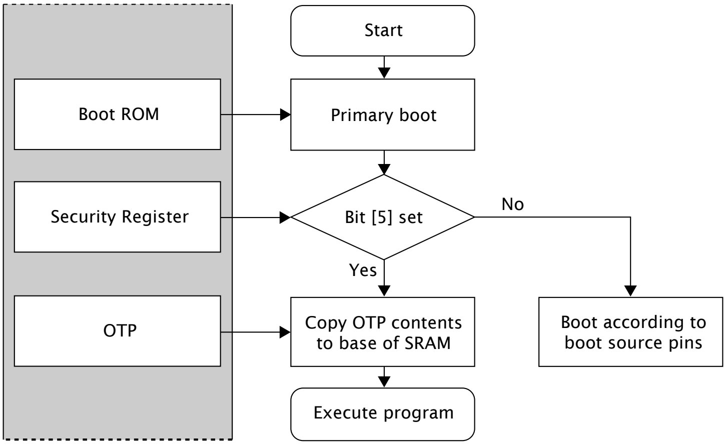


Figure 9: Boot procedure

Figure 10: Boot source pins

MODE[3]	MODE[2]	Boot Source
0	0	None: Device waits to be booted via JTAG
0	1	Reserved
1	0	xConnect Link B
1	1	SPI

The boot image has the following format:

- A 32-bit program size s in words.
- Program consisting of $s \times 4$ bytes.
- A 32-bit CRC, or the value 0x0D15AB1E to indicate that no CRC check should be performed.

The program size and CRC are stored least significant byte first. The program is loaded into the lowest memory address of RAM, and the program is started from that address. The CRC is calculated over the byte stream represented by the program size and the program itself. The polynomial used is 0xEDB88320 (IEEE 802.3); the CRC register is initialized with 0xFFFFFFFF and the residue is inverted to produce the CRC.

9.1 Boot from SPI master

If set to boot from SPI master, the processor enables the four pins specified in Figure 11, and drives the SPI clock at 2.5 MHz (assuming a 400 MHz core clock). A

READ command is issued with a 24-bit address 0x000000. The clock polarity and phase are 0 / 0.

Figure 11: SPI master pins

Pin	Signal	Description
X0D00	MISO	Master In Slave Out (Data)
X0D01	SS	Slave Select
X0D10	SCLK	Clock
X0D11	MOSI	Master Out Slave In (Data)

The xCORE Tile expects each byte to be transferred with the *least-significant bit first*. Programmers who write bytes into an SPI interface using the most significant bit first may have to reverse the bits in each byte of the image stored in the SPI device.

If a large boot image is to be read in, it is faster to first load a small boot-loader that reads the large image using a faster SPI clock, for example 50 MHz or as fast as the flash device supports.

The pins used for SPI boot are hardcoded in the boot ROM and cannot be changed. If required, an SPI boot program can be burned into OTP that uses different pins.

9.2 Boot from xConnect Link

If set to boot from an xConnect Link, the processor enables Link B around 200 ns after the boot process starts. Enabling the Link switches off the pull-down on resistors X0D16..X0D19, drives X0D16 and X0D17 low (the initial state for the Link), and monitors pins X0D18 and X0D19 for boot-traffic. X0D18 and X0D19 must be low at this stage. If the internal pull-down is too weak to drain any residual charge, external pull-downs of 10K may be required on those pins.

The boot-rom on the core will then:

1. Allocate channel-end 0.
2. Input a word on channel-end 0. It will use this word as a channel to acknowledge the boot. Provide the null-channel-end 0x0000FF02 if no acknowledgment is required.
3. Input the boot image specified above, including the CRC.
4. Input an END control token.
5. Output an END control token to the channel-end received in step 2.
6. Free channel-end 0.
7. Jump to the loaded code.

9.3 Boot from OTP

If an xCORE tile is set to use secure boot (see Figure 9), the boot image is read from address 0 of the OTP memory in the tile's security module.

This feature can be used to implement a secure bootloader which loads an encrypted image from external flash, decrypts and CRC checks it with the processor, and discontinues the boot process if the decryption or CRC check fails. XMOS provides a default secure bootloader that can be written to the OTP along with secret decryption keys.

Each tile has its own individual OTP memory, and hence some tiles can be booted from OTP while others are booted from SPI or the channel interface. This enables systems to be partially programmed, dedicating one or more tiles to perform a particular function, leaving the other tiles user-programmable.

9.4 Security register

The security register enables security features on the xCORE tile. The features shown in Figure 12 provide a strong level of protection and are sufficient for providing strong IP security.

Feature	**Bit**	**Description**
Disable JTAG	0	The JTAG interface is disabled, making it impossible for the tile state or memory content to be accessed via the JTAG interface.
Disable Link access	1	Other tiles are forbidden access to the processor state via the system switch. Disabling both JTAG and Link access transforms an xCORE Tile into a "secure island" with other tiles free for non-secure user application code.
Secure Boot	5	The processor is forced to boot from address 0 of the OTP, allowing the processor boot ROM to be bypassed (*see* §9).
Redundant rows	7	Enables redundant rows in OTP.
Sector Lock 0	8	Disable programming of OTP sector 0.
Sector Lock 1	9	Disable programming of OTP sector 1.
Sector Lock 2	10	Disable programming of OTP sector 2.
Sector Lock 3	11	Disable programming of OTP sector 3.
OTP Master Lock	12	Disable OTP programming completely: disables updates to all sectors and security register.
Disable JTAG-OTP	13	Disable all (read & write) access from the JTAG interface to this OTP.
Disable Global Debug	14	Disables access to the DEBUG_N pin.
	21..15	General purpose software accessable security register available to end-users.
	31..22	General purpose user programmable JTAG UserID code extension.

Figure 12: Security register features

10 Memory

10.1 OTP

The xCORE Tile integrates 8 KB one-time programmable (OTP) memory along with a security register that configures system wide security features. The OTP holds data in four sectors each containing 512 rows of 32 bits which can be used to implement secure bootloaders and store encryption keys. Data for the security register is loaded from the OTP on power up. All additional data in OTP is copied from the OTP to SRAM and executed first on the processor.

The OTP memory is programmed using three special I/O ports: the OTP address port is a 16-bit port with resource ID 0x100200, the OTP data is written via a 32-bit port with resource ID 0x200100, and the OTP control is on a 16-bit port with ID 0x100300. Programming is performed through `libotp` and `xburn`.

10.2 SRAM

The xCORE Tile integrates a single 64KB SRAM bank for both instructions and data. All internal memory is 32 bits wide, and instructions are either 16-bit or 32-bit. Byte (8-bit), half-word (16-bit) or word (32-bit) accesses are supported and are executed within one tile clock cycle. There is no dedicated external memory interface, although data memory can be expanded through appropriate use of the ports.

10.3 Deep Sleep Memory

The XS1-U6A-64-FB96 device includes 128 bytes of deep sleep memory for state storage during sleep mode. Deep sleep memory is volatile and if device input power is remove, the data will be lost.

11 USB PHY

The USB PHY provides High-Speed and Full-Speed, device, host, and on-the-go functionality. The PHY is configured through a set of peripheral registers (Appendix F), and data is communicated through ports on the digital node. A library, libxud_s.a, is provided to implement USB device functionality.

11.1 Logical Core Requirements

The XMOS XUD software component runs in a single logical core with endpoint and application cores communicating with it via a combination of channel communication and shared memory variables.

Each IN (host requests data from device) or OUT (data transferred from host to device) endpoint requires one logical core.

To guarantee correct operation the USB logical core must run at at least 80 MIPS, and the logical cores that communicate with the USB core must also run at 80

MIPS. This means that no more than six logical cores execute at any one time on a 500MHz device.

12 Analog-to-Digital Converter

The device has a 12-bit 1MSample/second Successive Approximation Register (SAR) Analogue to Digital Converter (ADC). It has 4 input pins which are multiplexed into the ADC. The sampling of the ADC is controlled using GPIO pin X0D24 that is triggered either by writing to port 1I, or by driving the pin externally. On each rising edge of the sample pin the ADC samples, holds and converts the data value from one of the analog input pins. Each of the 4 inputs can be enabled individually. Each of the enabled analog inputs is sampled in turn, on successive rising edges of the sample pin. The data is transmitted to the channel-end that the user configures during initialization of the ADC. Data is transmitted over the channel in individual packets, or in packets that contain multiple consecutive samples. The ADC uses an external reference voltage, nominally 3V3, which represents the full range of the ADC. The ADC configuration registers are documented in Appendix G.

The minimum latency for reading a value from the ADC into the xCORE register is shown in Figure 13:

Figure 13: Minimum latency to read sample from ADC to xCORE

Sample	Tile clock frequency	Start of packet	Subsequent samples
32-bit	500 MHz	840 ns	710 ns
32-bit	400 MHz	870 ns	740 ns
16-bit	500 MHz	770 ns	640 ns
16-bit	400 MHz	800 ns	670 ns

13 Supervisor Logic

An independent supervisor circuit provides power-on-reset, brown-out, and watchdog capabilities. This facilitates the design of systems that fail gracefully, whilst keeping BOM costs down.

The reset supervisor holds the chip in reset until all power supplies are good. This provides a power-on-reset (POR). An external reset is optional and the pin RST_N can be left not-connected.

If at any time any of the power supplies drop because of too little supply or too high a demand, the power supervisor will bring the chip into reset until the power supplies have been restored. This will reboot the system as if a cold-start has happened.

The 16-bit watchdog timer provides 1ms accuracy and runs independently of the real-time counter. It can be programmed with a time-out of between 1 ms and 65 seconds (Appendix E). If the watchdog is not set before it times out, the XS1-U6A-64-FB96 is reset. On boot, the program can read a register to test whether the reset was due to the watchdog. The watchdog timer is only enabled and clocked whilst the processor is in the AWAKE power state.

14 Energy management

XS1-U6A-64-FB96 devices can be powered by:

- An external 5v core and 3.3v I/O supply, increasing efficiency for USB bus powered applications.
- A single 3.3v supply.

14.1 DC-DC

XS1-U6A-64-FB96 devices include two DC-DC buck converters which can be configured to take input voltages between 3.3-5V power supply and output circuit voltages (nominally 1.8V and 1.0V) required by the analog peripherals and digital node.

14.2 Power mode controller

The device transitions through multiple states during the power-up and powerdown process.

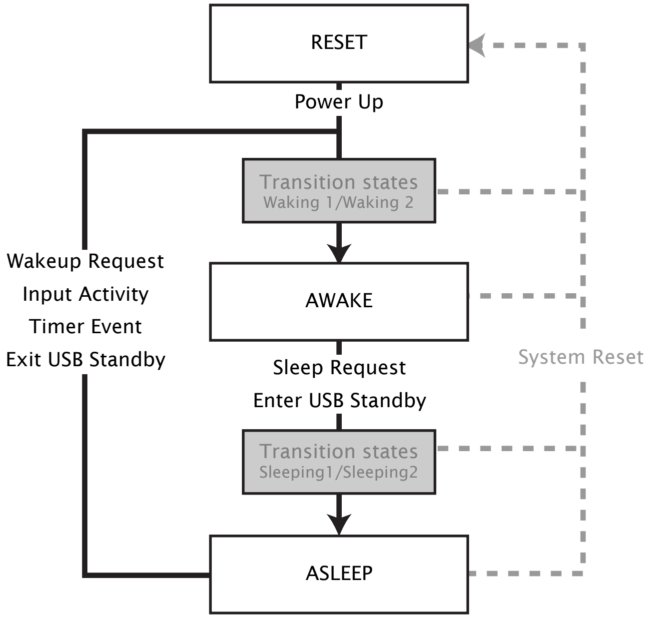


Figure 14: XS1-U6A-64-FB96 Power Up States and Transitions

The device is quiescent in the ASLEEP state, and is running in the AWAKE state. The other states allow a controlled transition between AWAKE and ASLEEP.

A transition from AWAKE state to ASLEEP state is instigated by a sleep request: either a write to the general control register or from the USB block requesting entry to standby mode. Sleep requests must only be made in the AWAKE state.

A transition from the ASLEEP state into the AWAKE state is instigated by a wakeup request triggered by a request from the USB block to exit standby mode an input, or a timer. The device only responds to a wakeup stimulus in the ASLEEP state. If wakeup stimulus occurs whilst transitioning from AWAKE to ASLEEP, the appropriate response occurs when the ASLEEP state is reached.

Configuration is through a set of registers documented in Appendix K.

14.3 Deep Sleep Modes and Real-Time Counter

The normal mode in which the XS1-U6A-64-FB96 operates is the AWAKE mode. In this mode, all cores, memory, and peripherals operate as normal. To save power, the XS1-U6A-64-FB96 can be put into a deep sleep mode, called ASLEEP, where the digital node is powered down, and most peripherals are powered down. The XS1-U6A-64-FB96 will stay in the ASLEEP mode until one of three conditions:

1. An external pin is asserted or deasserted (set by the program);
2. The 64-bit real-time counter reaches a value set by the program; or
3. The USB host (if USB is enabled) performs a wakeup.

When the chip is awake, the real-time counter counts the number of clock ticks on the oscillator. As such, the real-time counter will run at a fixed ratio, but synchronously with the 100 MHz timers on the xCORE Tile. When asleep, the real-time counter can be automatically switched to the 31,250 Hz silicon oscillator to save power (see Appendix I). To ensure that the real-time counter increases linearly over time, a programmable value is added to the counter on every 31,250 Hz clock-tick. This means that the clock will run at a granularity of 31,250 Hz but still maintain real-time in terms of the frequency of the main oscillator. If an accurate clock is required, even whilst asleep, then an external crystal or oscillator shall be provided that is used in both AWAKE and ASLEEP state.

The designer has to make a trade-off between accuracy of clocks when asleep and awake, costs, and deep-sleep power consumption. Four example designs are shown in Figure 15.

Figure 15: Example trade-offs in oscillator selection

Clocks used		Power	BOM	Accuracy	
Awake	**Asleep**	**Asleep**	**costs**	**Awake**	**Asleep**
20 Mhz SiOsc	31,250 SiOsc	lowest	lowest	lowest	lowest
24 MHz Crystal	31,250 SiOsc	lowest	medium	highest	lowest
5 MHz ext osc	5 MHz ext osc	medium	highest	highest	highest
24 MHz Crystal	24 MHz crystal	highest	medium	highest	highest

During deep-sleep, the program can store some state in 128 bytes of Deep Sleep Memory.

14.4 Requirements during sleep mode

Whilst in sleep mode, the device must still be powered as normal over 3V3 or 5V0 on VSUP, and 3V3 on VDDIO; however it will draw less power on both VSUP and VDDIO.

For best results (lowest power):

- The XTAL bias and XTAL oscillators should be switched off.
- The sleep register should be configured to
  - Disable all power supplies except DCDC2.
  - Set all power supplies to PFM mode
  - Mask the clock
  - Assert reset
- All GPIO and JTAG pins should be quiescent, and none should be driven against a pull-up or pull-down.
- 3V3 should be supplied as the input voltage to VSUP.

This will result in a power consumption of less than 100 uA on both VSUP and VDDIO.

If any power supply loses power-good status during the asleep-to-awake or awake-to-asleep transitions, a system reset is issued.

15 JTAG

The JTAG module can be used for loading programs, boundary scan testing, in-circuit source-level debugging and programming the OTP memory.

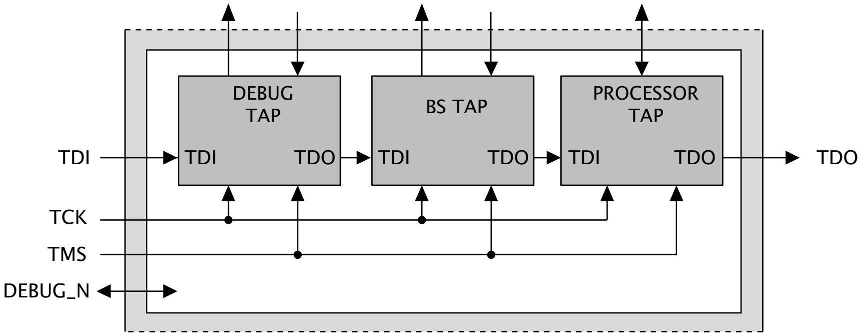


Figure 16: JTAG chain structure

The JTAG chain structure is illustrated in Figure 16. Directly after reset, three TAP controllers are present in the JTAG chain: the debug TAP, the boundary scan TAP and the processor TAP. The debug TAP provides access into the peripherals including the ADC and USB. The boundary scan TAP is a standard 1149.1 compliant TAP that can be used for boundary scan of the I/O pins. The processor TAP provides access into the xCORE Tile, switch and OTP for loading code and debugging.

The JTAG module can be reset by holding TMS high for five clock cycles.

The DEBUG_N pin is used to synchronize the debugging of multiple processors. This pin can operate in both output and input mode. In output mode and when configured to do so, DEBUG_N is driven low by the device when the processor hits a debug break point. Prior to this point the pin will be tri-stated. In input mode and when configured to do so, driving this pin low will put the processor into debug mode. Software can set the behavior of the processor based on this pin. This pin should have an external pull up of 4K7-47KΩ or left not connected in single core applications.

The JTAG device identification register can be read by using the IDCODE instruction. Its contents are specified in Figure 17.

Figure 17: IDCODE return value

<table>
<tr><td>Bit31</td><td colspan="30">Device Identification Register</td><td>Bit0</td></tr>
<tr><td colspan="4">Version</td><td colspan="16">Part Number</td><td colspan="11">Manufacturer Identity</td><td>1</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td colspan="4">0</td><td colspan="4">0</td><td colspan="4">0</td><td colspan="4">0</td><td colspan="4">3</td><td colspan="4">6</td><td colspan="4">3</td><td colspan="4">3</td></tr>
</table>

The JTAG usercode register can be read by using the USERCODE instruction. Its contents are specified in Figure 18. The OTP User ID field is read from bits [22:31] of the security register , *see* §10.1 (all zero on unprogrammed devices).

Figure 18: USERCODE return value

<table>
<tr><td>Bit31</td><td colspan="30">Usercode Register</td><td>Bit0</td></tr>
<tr><td colspan="10">OTP User ID</td><td colspan="4">Unused</td><td colspan="18">Silicon Revision</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td colspan="4">0</td><td colspan="4">0</td><td colspan="4">0</td><td colspan="4">2</td><td colspan="4">C</td><td colspan="4">0</td><td colspan="4">0</td><td colspan="4">0</td></tr>
</table>

16 Board Integration

XS1-U6A-64-FB96 devices are optimized for layout on low cost PCBs using standard design rules. Careful layout is required to maximize the device performance. XMOS therefore recommends that the guidelines in this section are followed when laying out boards using the device.

The XS1-U6A-64-FB96 includes two DC-DC buck converters that take input voltages between 3.3-5V and output the 1.8V and 1.0V circuits required by the digital core and analogue peripherals. The DC-DC converters should have a 4.7uF X5R or X7R ceramic capacitor and a 100nF X5R or X7R ceramic capacitor on the VSUP input pins M1 and M2. These capacitors must be placed as close as possible to the those pins (within a maximum of 5mm), with the routing optimized to minimize the inductance and resistance of the traces.

The SW output pin must have an LC filter on the output with a 4.7uH inductor and 22uF X5R capacitor. The capacitor must have maximum ESR value of 0.015R, and the inductor should have a maximum DCR value of 0.07R, to meet the efficiency specifications of the DC-DC converter, although this requirement may be relaxed if a drop in efficiency is acceptable. A list of suggested inductors is in Figure 19.

	Part number	Current	Max DCR	Package
Yuden	CBC2518T4R7M	680 mA	260 $m\Omega$	2518 (1007)
TDK	NLCV32T-4R7M-PFR	620 mA	200 $m\Omega$	3225 (1210)
Murata	LQM2HPN4R7MGC	800 mA	225 $m\Omega$	2520 (1008)
Sumida	0420CDMCBDS-4R7MC	3400 mA	80 $m\Omega$	4.7 x 4.3 mm
Wurth	744043004	1550 mA	70 $m\Omega$	4.8 x 4.8 mm
Murata	LQH55DN4R7M03L	2700 mA	57 $m\Omega$	5750 (2220)

Figure 19: Example 4.7 μH inductors

The traces from the SW output pins to the inductor and from the output capacitor back to the VDD pins must be routed to minimize the coupling between them.

The power supplies must be brought up monotonically and input voltages must not exceed specification at any time.

The VDDIO supply to the XS1-U6A-64-FB96 requires a 100nF X5R or X7R ceramic decoupling capacitor placed as close as possible to the supply pins.

If the ADC Is used, it requires a 100nF X5R or X7R ceramic decoupling capacitor placed as close as possible to the AVDD pin. Care should be taken to minimize noise on these inputs, and if necessary an extra 10uF decoupling capacitor and ferrite bead can be used to remove noise from this supply.

The crystal oscillator requires careful routing of the XI / XO nodes as these are high impedance and very noise sensitive. Hence, the traces should be as wide and short as possible, and routed over a continuous ground plane. They should not be routed near noisy supply lines or clocks. The device has a load capacitance of 18pF for the crystal. Care must be taken, so that the inductance and resistance of the ground returns from the capacitors to the ground of the device is minimized.

16.1 USB connections

USB_VBUS should be connected to the VBUS pin of the USB connector. A 2.2 uF capacitor to ground is required on the VBUS pin. A ferrite bead may be used to reduce HF noise.

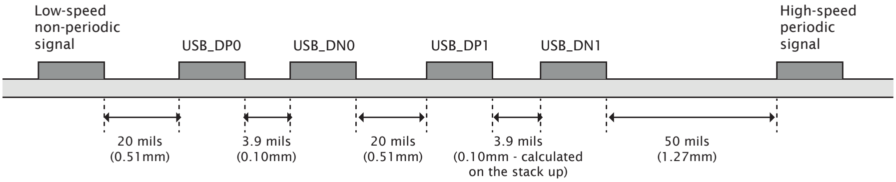
For self-powered systems, a bleeder resistor may be required to stop VBUS from floating when no USB cable is attached.

USB_DP and USB_DN should be connected to the USB connector. USB_ID does not need to be connected.

16.2 USB signal routing and placement

The USB_DP and USB_DN lines are the positive and negative data polarities of a high speed USB signal respectively. Their high-speed differential nature implies that they must be coupled and properly isolated. The board design must ensure that the board traces for USB_DP and USB_DN are tightly matched. In addition, according to the USB 2.0 specification, the USB_DP and USB_DN differential impedance must be 90 Ω.

Figure 20: USB trace separation showing a low speed signal, two differential pairs and a high-speed clock



16.2.1 General routing and placement guidelines

The following guidelines will help to avoid signal quality and EMI problems on high speed USB designs. They relate to a four-layer (Signal, GND, Power, Signal) PCB.

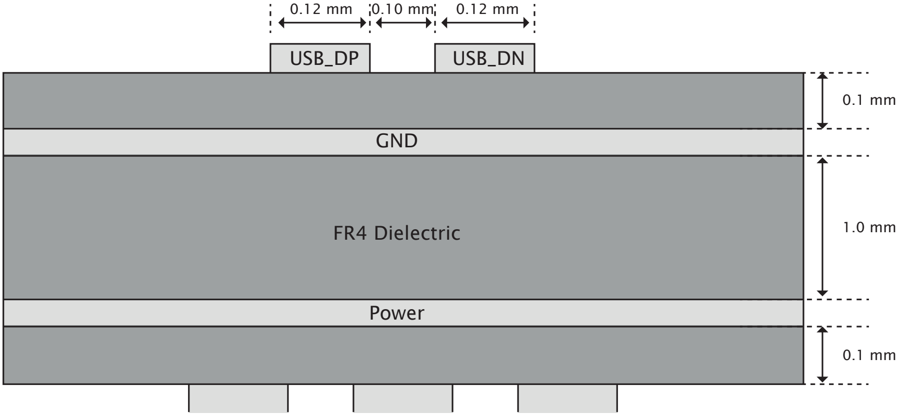


Figure 21: Example USB board stack

For best results, most of the routing should be done on the top layer (assuming the USB connector and XS1-U6A-64-FB96 are on the top layer) closest to GND. Reference planes should be below the transmission lines in order to maintain control of the trace impedance.

We recommend that the high-speed clock and high-speed USB differential pairs are routed first before any other routing. When routing high speed USB signals, the following guidelines should be followed:

- High speed differential pairs should be routed together.
- High-speed USB signal pair traces should be trace-length matched. Maximum trace-length mismatch should be no greater than 4mm.
- Ensure that high speed signals (clocks, USB differential pairs) are routed as far away from off-board connectors as possible.
- High-speed clock and periodic signal traces that run parallel should be at least 1.27mm away from USB_DP/USB_DN (see Figure 20).
- Low-speed and non-periodic signal traces that run parallel should be at least 0.5mm away from USB_DP/USB_DN (see Figure 20).
- Route high speed USB signals on the top of the PCB wherever possible.
- Route high speed USB traces over continuous power planes, with no breaks. If a trade-off must be made, changing signal layers is preferable to crossing plane splits.
- Follow the $20 \times h$ rule; keep traces $20 \times h$ (the height above the power plane) away from the edge of the power plane.
- Use a minimum of vias in high speed USB traces.
- Avoid corners in the trace. Where necessary, rather than turning through a 90 degree angle, use two 45 degree turns or an arc.
- DO NOT route USB traces near clock sources, clocked circuits or magnetic devices.
- Avoid stubs on high speed USB signals.

16.3 Land patterns and solder stencils

The land pattern recommendations in this document are based on a RoHS compliant process and derived, where possible, from the nominal *Generic Requirements for Surface Mount Design and Land Pattern Standards* IPC-7351B specifications. This standard aims to achieve desired targets of heel, toe and side fillets for solder-joints.

Solder paste and ground via recommendations are based on our engineering and development kit board production. They have been found to work and optimized as appropriate to achieve a high yield. These factors should be taken into account during design and manufacturing of the PCB.

The following land patterns and solder paste contains recommendations. Final land pattern and solder paste decisions are the responsibility of the customer. These should be tuned during manufacture to suit the manufacturing process.

The package is a 96 pin Ball Grid Array package on a 0.8mm pitch with 0.4mm balls.

An example land pattern is shown in Figure 22.

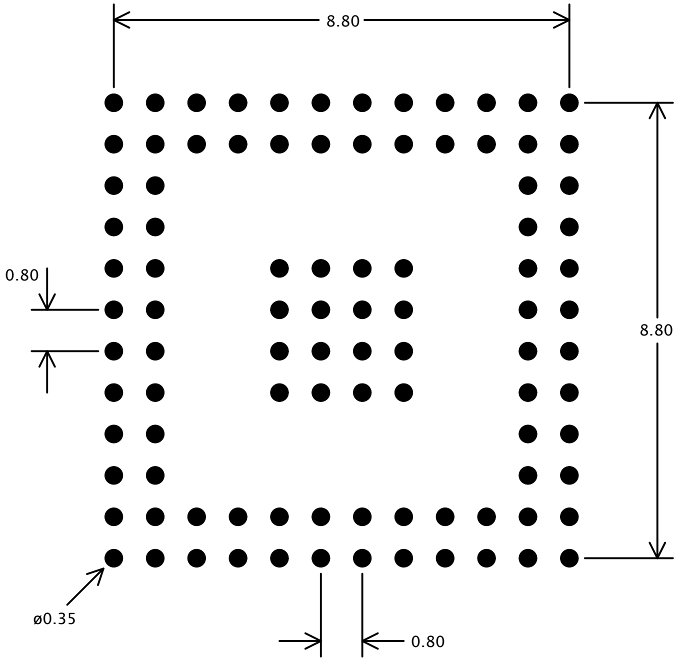


Figure 22: Example land pattern

Pad widths and spacings are such that solder mask can still be applied between the pads using standard design rules. This is highly recommended to reduce solder shorts.

16.4 Ground and Thermal Vias

Vias next to each ground ball into the ground plane of the PCB are recommended for a low inductance ground connection and good thermal performance. Vias with with a 0.6mm diameter annular ring and a 0.3mm drill would be suitable.

16.5 Moisture Sensitivity

XMOS devices are, like all semiconductor devices, susceptible to moisture absorption. When removed from the sealed packaging, the devices slowly absorb moisture from the surrounding environment. If the level of moisture present in the device is too high during reflow, damage can occur due to the increased internal vapour pressure of moisture. Example damage can include bond wire damage, die lifting, internal or external package cracks and/or delamination.

All XMOS devices are Moisture Sensitivity Level (MSL) 3 - devices have a shelf life of 168 hours between removal from the packaging and reflow, provided they are stored below 30C and 60% RH. If devices have exceeded these values or an included moisture indicator card shows excessive levels of moisture, then the parts should be baked as appropriate before use. This is based on information from *Joint IPC/JEDEC Standard For Moisture/Reflow Sensitivity Classification For Nonhermetic Solid State Surface-Mount Devices* J-STD-020 Revision D.

17 Example XS1-U6A-64-FB96 Board Designs

This section shows example schematics and layout for a 2-layer PCB.

- Figures 23 shows example schematics and layout. It uses a 24 MHz crystal for the clock, and an SPI flash for booting. The XS1-U6A-64-FB96 is powered directly from 5V. An optional ESD protection device is included to increase ESD protection from 2 to 15 kV.
- Figures 24 shows example schematics and layout for a design that uses an oscillator rather than a crystal. If required a 3V3 oscillator can be used (for example when sharing an oscillator with other parts of the design), but a resistor bridge must be included to reduce the XI/CLK input from 3V3 to 1V8.
- Figure 25 shows example schematics and layout for a design that does not use USB and that runs off the internal 20 MHz oscillator. The XS1-U6A-64-FB96 is powered directly from 3V3.

Flash, AVDD, RST, and JTAG connectivity are all optional. Flash can be removed if the processor boots from OTP. The AVDD decoupler and wiring can be removed if the ADC is not used. RST_N and all JTAG wiring can be removed if debugging is not required (see Appendix M)

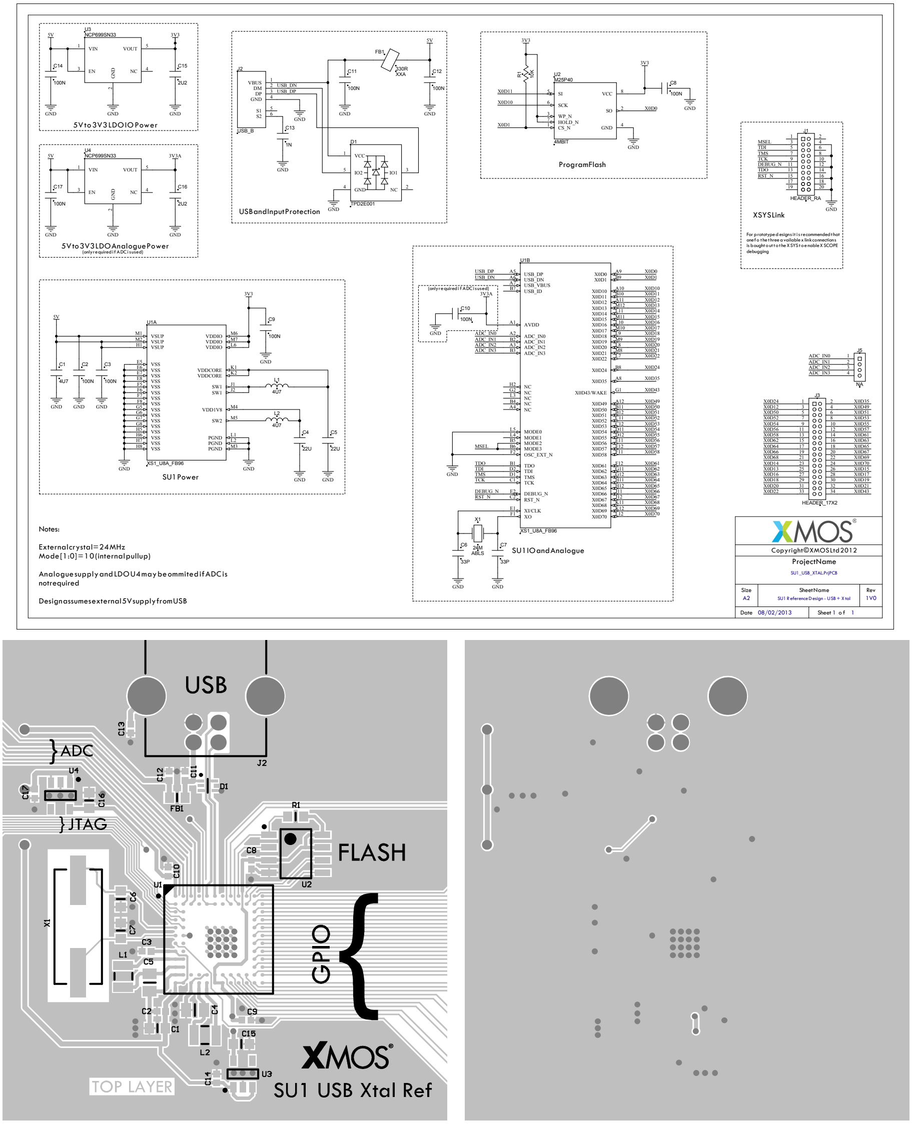


Figure 23: Example XTAL schematic, with top and bottom layout of a 2-layer PCB

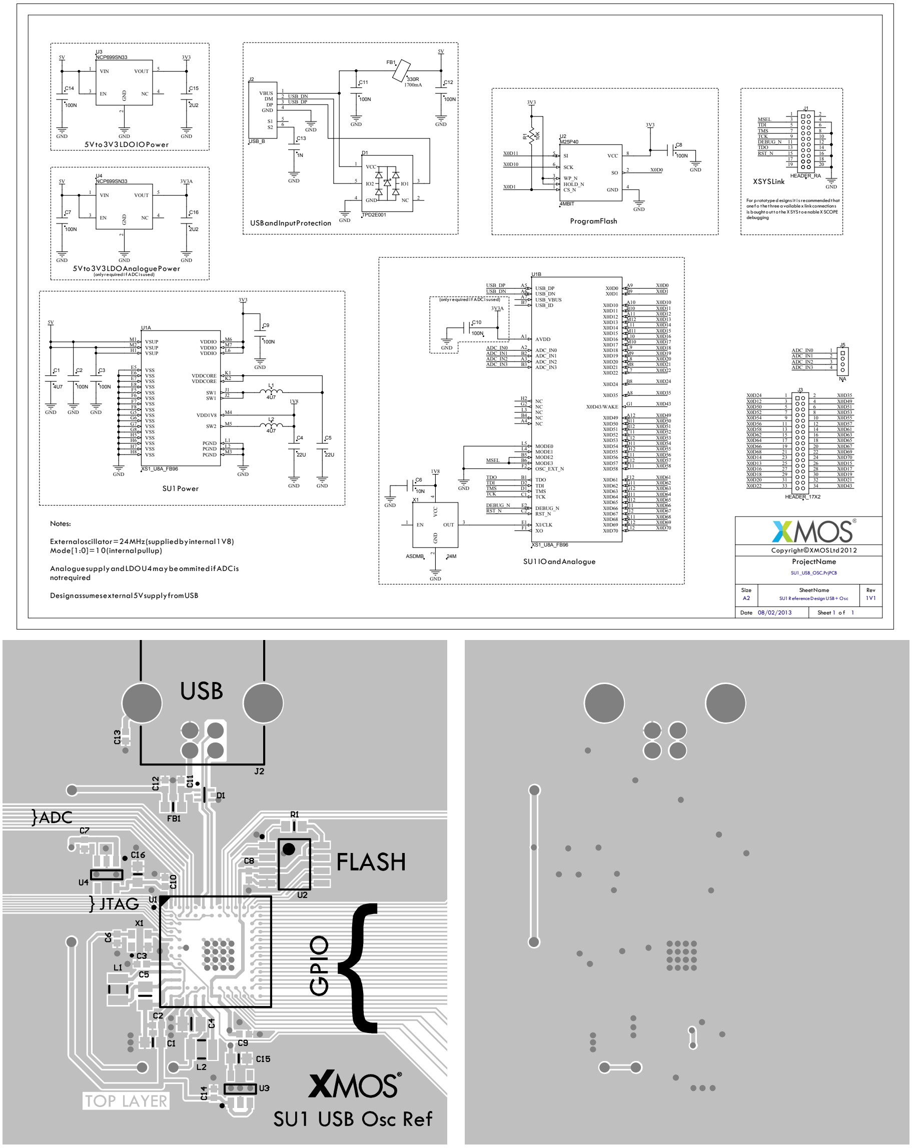
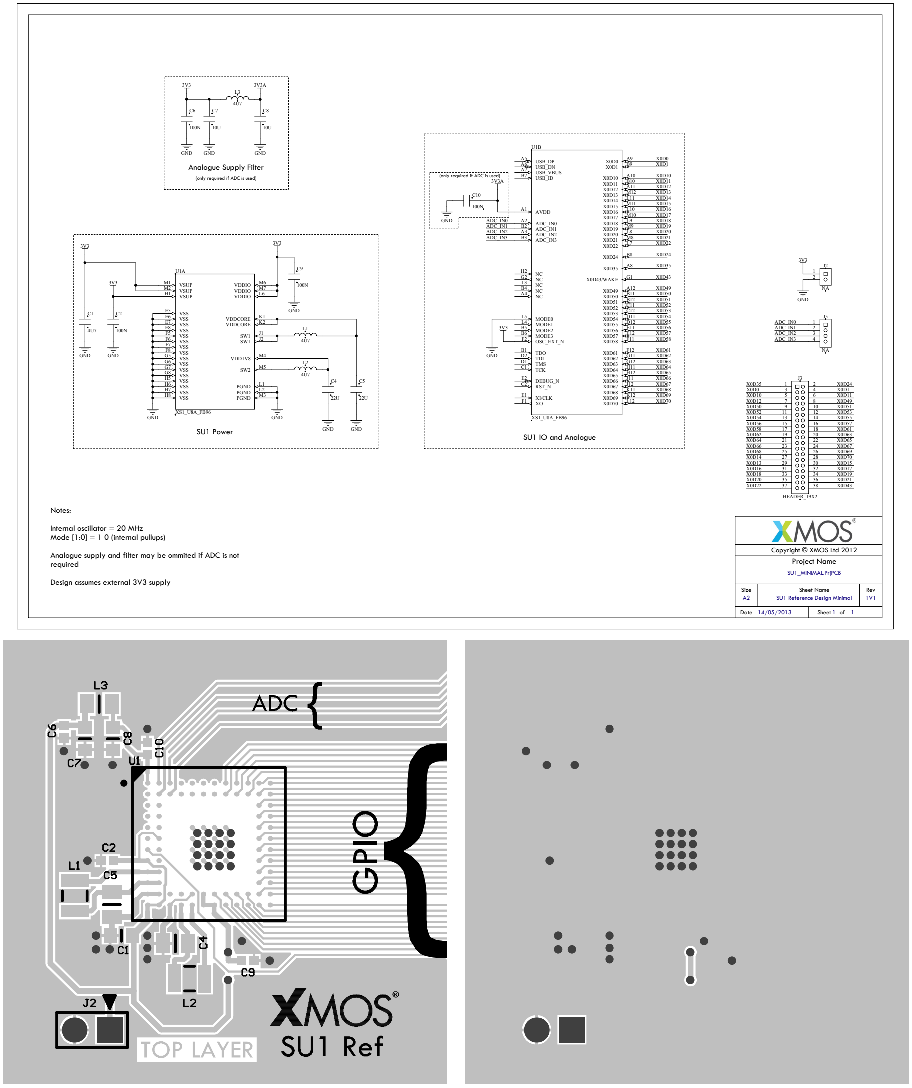


Figure 24: Example Oscillator schematic, with top and bottom layout of a 2-layer PCB



Notes:

Internal oscillator = 20 MHz
Mode [1:0] = 1 0 (internal pullups)

Analogue supply and filter may be ommited if ADC is not required

Design assumes external 3V3 supply

Figure 25: Example minimal system schematic, with top and bottom layout of a 2-layer PCB

18 DC and Switching Characteristics

18.1 Operating Conditions

Symbol	Parameter	MIN	TYP	MAX	UNITS	Notes
VSUP	Power Supply (3.3V Mode)	3.00	3.30	3.60	V	
	Power Supply (5V Mode)	4.50	5.00	5.50	V	
VDDIO	I/O supply voltage	3.00	3.30	3.60	V	
AVDD	Analog Supply and Reference Voltage	3.00	3.30	3.60	V	
Cl	xCORE Tile I/O load capacitance			25	pF	
Ta	Ambient operating temperature (Commercial)	0		70	°C	
	Ambient operating temperature (Industrial)	-40		85	°C	
Tj	Junction temperature			125	°C	
Tstg	Storage temperature	-65		150	°C	

Figure 26: Operating conditions

18.2 DC1 Characteristics

Symbol	Parameter	MIN	TYP	MAX	UNITS	Notes
VDDCORE	Tile Supply Voltage	0.95	1.00	1.05	V	
V(RIPPLE)	Ripple Voltage (peak to peak)		10	40	mV	
V(ACC)	Voltage Accuracy	-5		5	%	A
F(S)	Switching Frequency		1		MHz	
F(SVAR)	Variation in Switching Frequency	-10		10	%	
Effic	Efficiency		80		%	
PGT(HIGH)	Powergood Threshold (High)		95		%/VDDCORE	
PGT(LOW)	Powergood Threshold (Low)		80		%/VDDCORE	

A If supplied externally.

Figure 27: DC1 characteristics

18.3 DC2 Characteristics

Symbol	Parameter	MIN	TYP	MAX	UNITS	Notes
VDD1V8	1V8 Supply Voltage		1.80		V	
V(RIPPLE)	Ripple Voltage (peak to peak)		10	40	mV	
V(ACC)	Voltage Accuracy	-5		5	%	A
F(S)	Switching Frequency		1		MHz	
F(SVAR)	Variation in Switching Frequency	-10		10	%	
Effic	Efficiency		80		%	
PGT(HIGH)	Powergood Threshold (High)		95		%/VDD1V8	
PGT(LOW)	Powergood Threshold (Low)		80		%/VDD1V8	

A If supplied externally.

Figure 28: DC2 characteristics

18.4 ADC Characteristics

Symbol	Parameter	MIN	TYP	MAX	UNITS	Notes
N	Resolution		12		bits	
Fs	Conversion Speed			1	MSPS	
Nch	Number of Channels		4			
Vin	Input Range	0		AVDD	V	
DNL	Differential Non Linearity	-1		1.5	LSB	
INL	Integral Non Linearity	-4		4	LSB	
E(GAIN)	Gain Error	-10		10	LSB	
E(OFFSET)	Offset Error	-3		3	mV	
T(PWRUP)	Power time for ADC Clock Fclk			7	1/Fclk	
ENOB	Effective Number of bits		10			

Figure 29: ADC characteristics

18.5 USB Characteristics

Symbol	Parameter	MIN	TYP	MAX	UNITS	Notes
VBUS	Power supply	0	5	5.25	V	A
ID	Device ID (OTG)	0		3.3	V	
DP	Data positive	0		3.3	V	
DN	Data negative (inverted)	0		3.3	V	

A The VBUS pin is used for measuring the VBUS voltage only.

Figure 30: USB characteristics

Contact XMOS for further details on USB characteristics.

18.6 Digital I/O Characteristics

Figure 31: Digital I/O characteristics

Symbol	Parameter	MIN	TYP	MAX	UNITS	Notes
V(IH)	Input high voltage	2.00		3.60	V	A
V(IL)	Input low voltage	-0.30		0.70	V	A
V(OH)	Output high voltage	2.00			V	B, C
V(OL)	Output low voltage			0.60	V	B, C
R(PU)	Pull-up resistance		35K		Ω	D
R(PD)	Pull-down resistance		35K		Ω	D

A All pins except power supply pins.

B Ports 1A, 1D, 1E, 1H, 1I, 1J, 1K and 1L are nominal 8 mA drivers, the remainder of the general-purpose I/Os are 4 mA.

C Measured with 4 mA drivers sourcing 4 mA, 8 mA drivers sourcing 8 mA.

D Used to guarantee logic state for an I/O when high impedance. The internal pull-ups/pull-downs should not be used to pull external circuitry.

18.7 ESD Stress Voltage

Figure 32: ESD stress voltage

Symbol	Parameter	MIN	TYP	MAX	UNITS	Notes
HBM	Human body model			2.00	kV	
CDM	Charged Device Model			500	V	

18.8 Device Timing Characteristics

Figure 33: Device timing characteristics

Symbol	Parameter	MIN	TYP	MAX	UNITS	Notes
T(RST)	Reset pulse width	5			µs	
T(INIT)	Initialisation (On Silicon Oscillator)			TBC	ms	A
	Initialisation (Crystal Oscillator)			TBC	ms	
T(WAKE)	Wake up time (Sleep to Active)			TBC	ms	
T(SLEEP)	Sleep Time (Active to Sleep)			TBC	ms	

A Shows the time taken to start booting after RST_N has gone high.

18.9 Crystal Oscillator Characteristics

Figure 34: Crystal oscillator characteristics

Symbol	Parameter	MIN	TYP	MAX	UNITS	Notes
F(FO)	Input Frequency	5		30	MHz	A

A For use with USB, the design should use a 12 or 24 MHz +/- 150 ppm crystal.

18.10 External Oscillator Characteristics

Figure 35: External oscillator characteristics

Symbol	Parameter	MIN	TYP	MAX	UNITS	Notes
F(EXT)	External Frequency			100	MHz	A
V(IH)	Input high voltage	1.62		1.98	V	
V(IL)	Input low voltage			0.4	V	

A For use with USB, the design should use a 12 or 24 MHz +/- 150 ppm crystal.

18.11 Power Consumption

Figure 36: xCORE Tile currents

Symbol	Parameter	MIN	TYP	MAX	UNITS	Notes
P(AWAKE)	Active Power for awake states	TBC	300	TBC	mW	
P(SLEEP)	Power when asleep	TBC	500	TBC	µW	

18.12 Clock

Figure 37: Clock

Symbol	Parameter	MIN	TYP	MAX	UNITS	Notes
f(MAX)	Processor clock frequency			500	MHz	A

A Assumes typical tile and I/O voltages with nominal activity.

18.13 Processor I/O AC Characteristics

Figure 38: I/O AC characteristics

Symbol	Parameter	MIN	TYP	MAX	UNITS	Notes
T(XOVALID)	Input data valid window	8			ns	
T(XOINVALID)	Output data invalid window	9			ns	
T(XIFMAX)	Rate at which data can be sampled with respect to an external clock			60	MHz	

The input valid window parameter relates to the capability of the device to capture data input to the chip with respect to an external clock source. It is calculated as the sum of the input setup time and input hold time with respect to the external clock as measured at the pins. The output invalid window specifies the time for which an output is invalid with respect to the external clock. Note that these parameters are specified as a window rather than absolute numbers since the device provides functionality to delay the incoming clock with respect to the incoming data.

Information on interfacing to high-speed synchronous interfaces can be found in the XS1 Port I/O Timing document, X5821.

18.14 xConnect Link Performance

Figure 39: Link performance

Symbol	Parameter	MIN	TYP	MAX	UNITS	Notes
B(2blinkP)	2b link bandwidth (packetized)			103	MBit/s	A, B
B(5blinkP)	5b link bandwidth (packetized)			271	MBit/s	A, B
B(2blinkS)	2b link bandwidth (streaming)			125	MBit/s	B
B(5blinkS)	5b link bandwidth (streaming)			313	MBit/s	B

A Assumes 32-byte packet in 3-byte header mode. Actual performance depends on size of the header and payload.

B 7.5 ns symbol time.

The asynchronous nature of links means that the relative phasing of CLK clocks is not important in a multi-clock system, providing each meets the required stability criteria.

18.15 JTAG Timing

Figure 40: JTAG timing

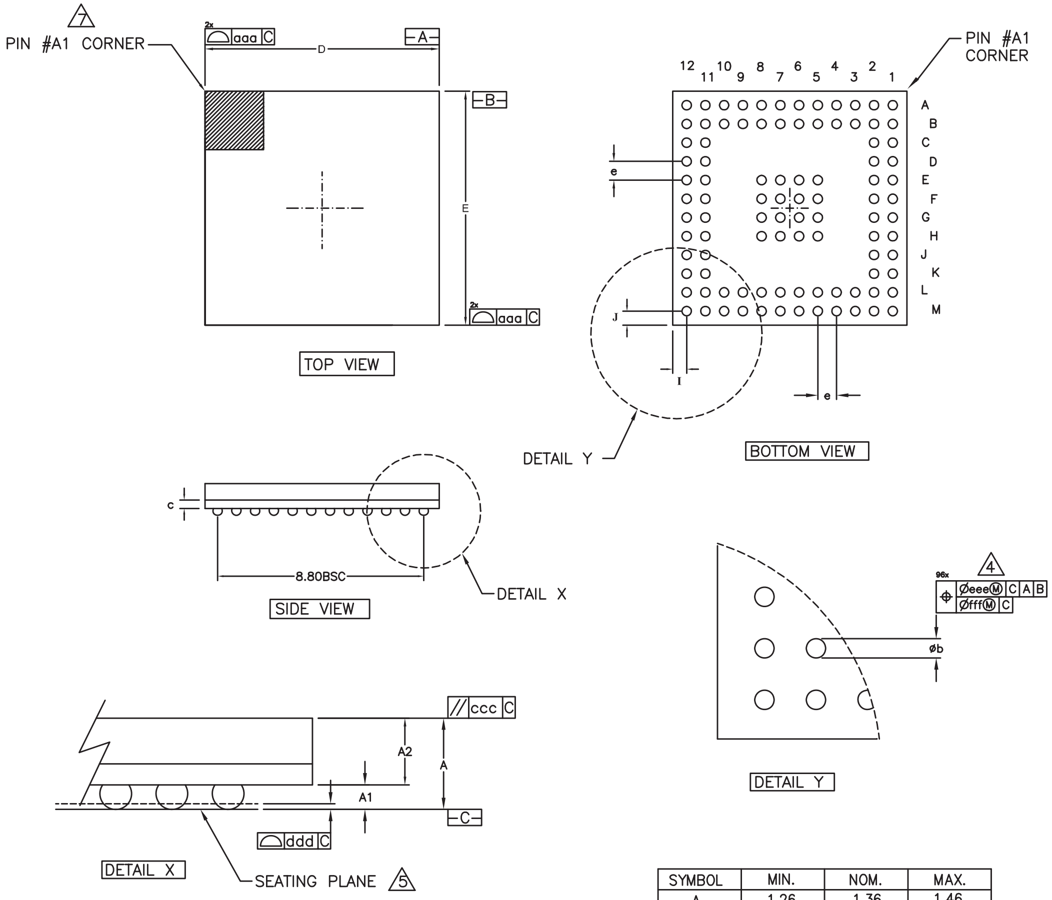
Symbol	Parameter	MIN	TYP	MAX	UNITS	Notes
f(TCK_D)	TCK frequency (debug)			TBC	MHz	
f(TCK_B)	TCK frequency (boundary scan)			TBC	MHz	
T(SETUP)	TDO to TCK setup time	TBC			ns	A
T(HOLD)	TDO to TCK hold time	TBC			ns	A
T(DELAY)	TCK to output delay			TBC	ns	B

A Timing applies to TMS and TDI inputs.

B Timing applies to TDO output from negative edge of TCK.

All JTAG operations are synchronous to TCK.

19 Package Information



NOTE:

1. ALL DIMENSIONS ARE IN MILLIMETERS, ANGLE IS DEGREES.
2. "e" REPRESENTS THE BASIC SOLDER BALL GRID PITCH.
3. "M" REPRESENTS THE MAXIMUM SOLDER BALL MATRIX SIZE.
4. DEMENSIONS "b" IS MEASURED AT THE MAXIMUM SOLDER BALL DIAMETER PARALLEL TO PRIMARY DATUM -C- .
5. PRIMARY DATUM -C- AND SEATING PLANE ARE DESIGNED BY THE SPHERICAL CROWNS OF THE SOLDER BALLS.
6. DIMENSIONING AND TOLERANCING PER ASME Y14.5M–1994
7. A1 CORNER MUST BE IDENTIFIED BY LASER MARK.
8. PACKAGE DIMENSIONS CONFORM TO JEDEC REGISTRATION MO–275.

SYMBOL	MIN.	NOM.	MAX.
A	1.26	1.36	1.46
A1	0.25	0.30	0.35
A2	1.01	1.06	1.11
D	9.90	10.00	10.10
E	9.90	10.00	10.10
I	0.60 REF.		
J	0.60 REF.		
M	10x10<DEPOPULATED>		
aaa			0.15
ccc			0.20
ddd			0.10
eee			0.15
fff			0.08
b	0.35	0.40	0.45
e	0.80 BSC.		
c	0.36 REF.		

19.1 Part Marking

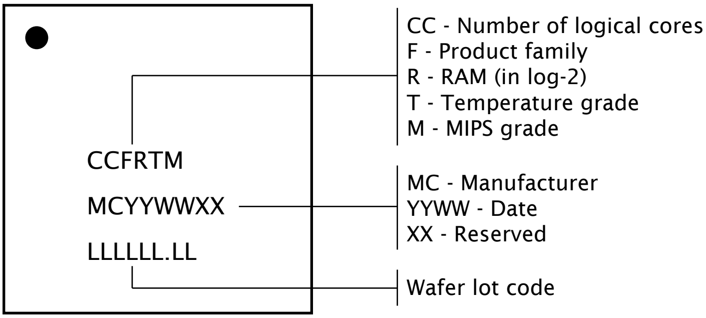


Figure 41: Part marking scheme

20 Ordering Information

Figure 42: Orderable part numbers

Product Code	Marking	Qualification	Speed Grade
XS1-U6A-64-FB96-C5	6U6C5	Commercial	500 MIPS
XS1-U6A-64-FB96-I5	6U6I5	Industrial	500 MIPS

Appendices

A Configuring the device

The device is configured through ten banks of registers, as shown in Figure 43.

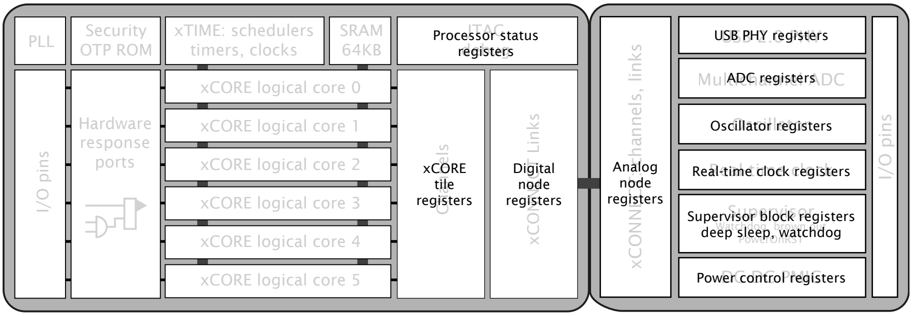


Figure 43: Registers

A.1 Accessing a processor status register

The processor status registers are accessed directly from the processor instruction set. The instructions GETPS and SETPS read and write a word. The register number should be translated into a processor-status resource identifier by shifting the register number left 8 places, and ORing it with 0x0C. Alternatively, the functions `getps(reg)` and `setps(reg,value)` can be used from XC.

A.2 Accessing an xCORE Tile configuration register

xCORE Tile configuration registers can be accessed through the interconnect using the functions `write_tile_config_reg(tileref, ...)` and `read_tile_config_reg(tileref, ...)`, where `tileref` is the name of the xCORE Tile, e.g. `tile[1]`. These functions implement the protocols described below.

Instead of using the functions above, a channel-end can be allocated to communicate with the xCORE tile configuration registers. The destination of the channel-end should be set to `0xnnnnC20C` where `nnnnnn` is the tile-identifier.

A write message comprises the following:

control-token 192	24-bit response channel-end identifier	16-bit register number	32-bit data	control-token 1

The response to a write message comprises either control tokens 3 and 1 (for success), or control tokens 4 and 1 (for failure).

A read message comprises the following:

control-token 193	24-bit response channel-end identifier	16-bit register number	control-token 1

The response to the read message comprises either control token 3, 32-bit of data, and control-token 1 (for success), or control tokens 4 and 1 (for failure).

A.3 Accessing digital and analogue node configuration registers

Node configuration registers can be accessed through the interconnect using the functions `write_node_config_reg(device, ...)` and `read_node_config_reg(device, ↪ ...)`, where `device` is the name of the node. These functions implement the protocols described below.

Instead of using the functions above, a channel-end can be allocated to communicate with the node configuration registers. The destination of the channel-end should be set to `0xnnnnC30C` where `nnnn` is the node-identifier.

A write message comprises the following:

control-token 192	24-bit response channel-end identifier	16-bit register number	32-bit data	control-token 1

The response to a write message comprises either control tokens 3 and 1 (for success), or control tokens 4 and 1 (for failure).

A read message comprises the following:

control-token 193	24-bit response channel-end identifier	16-bit register number	control-token 1

The response to a read message comprises either control token 3, 32-bit of data, and control-token 1 (for success), or control tokens 4 and 1 (for failure).

A.4 Accessing a register of an analogue peripheral

Peripheral registers can be accessed through the interconnect using the functions `write_periph_32(device, peripheral, ...)`, `read_periph_32(device, peripheral, ...)` ↪ , `write_periph_8(device, peripheral, ...)`, and `read_periph_8(device, peripheral` ↪ `, ...)`; where device is the name of the analogue device, and peripheral is the number of the peripheral. These functions implement the protocols described below.

A channel-end should be allocated to communicate with the configuration registers. The destination of the channel-end should be set to `0xnnnnpp02` where `nnnn` is the node-identifier and `pp` is the peripheral identifier.

A write message comprises the following:

control-token 36	24-bit response channel-end identifier	8-bit register number	8-bit size	data	control-token 1

The response to a write message comprises either control tokens 3 and 1 (for success), or control tokens 4 and 1 (for failure).

A read message comprises the following:

control-token 37	24-bit response channel-end identifier	8-bit register number	8-bit size	control-token 1

The response to the read message comprises either control token 3, data, and control-token 1 (for success), or control tokens 4 and 1 (for failure).

B Processor Status Configuration

The processor status control registers can be accessed directly by the processor using processor status reads and writes (use `getps(reg)` and `setps(reg,value)` for reads and writes).

Number	Perm	Description
0x00	RW	RAM base address
0x01	RW	Vector base address
0x02	RW	xCORE Tile control
0x03	RO	xCORE Tile boot status
0x05	RO	Security configuration
0x06	RW	Ring Oscillator Control
0x07	RO	Ring Oscillator Value
0x08	RO	Ring Oscillator Value
0x09	RO	Ring Oscillator Value
0x0A	RO	Ring Oscillator Value
0x10	DRW	Debug SSR
0x11	DRW	Debug SPC
0x12	DRW	Debug SSP
0x13	DRW	DGETREG operand 1
0x14	DRW	DGETREG operand 2
0x15	DRW	Debug interrupt type
0x16	DRW	Debug interrupt data
0x18	DRW	Debug core control
0x20 .. 0x27	DRW	Debug scratch
0x30 .. 0x33	DRW	Instruction breakpoint address
0x40 .. 0x43	DRW	Instruction breakpoint control
0x50 .. 0x53	DRW	Data watchpoint address 1
0x60 .. 0x63	DRW	Data watchpoint address 2
0x70 .. 0x73	DRW	Data breakpoint control register
0x80 .. 0x83	DRW	Resources breakpoint mask
0x90 .. 0x93	DRW	Resources breakpoint value
0x9C .. 0x9F	DRW	Resources breakpoint control register

Figure 44: Summary

B.1 RAM base address: 0x00

This register contains the base address of the RAM. It is initialized to 0x00010000.

0x00:
RAM base address

Bits	Perm	Init	Description
31:2	RW		Most significant 16 bits of all addresses.
1:0	RO	-	Reserved

B.2 Vector base address: 0x01

Base address of event vectors in each resource. On an interrupt or event, the 16 most significant bits of the destination address are provided by this register; the least significant 16 bits come from the event vector.

0x01:
Vector base address

Bits	Perm	Init	Description
31:16	RW		The most significant bits for all event and interrupt vectors.
15:0	RO	-	Reserved

B.3 xCORE Tile control: 0x02

Register to control features in the xCORE tile

0x02:
xCORE Tile control

Bits	Perm	Init	Description
31:6	RO	-	Reserved
5	RW	0	Set to 1 to select the dynamic mode for the clock divider when the clock divider is enabled. In dynamic mode the clock divider is only activated when all active logical cores are paused. In static mode the clock divider is always enabled.
4	RW	0	Set to 1 to enable the clock divider. This slows down the xCORE tile clock in order to use less power.
3:0	RO	-	Reserved

B.4 xCORE Tile boot status: 0x03

This read-only register describes the boot status of the xCORE tile.

0x03: xCORE Tile boot status

Bits	Perm	Init	Description
31:24	RO	-	Reserved
23:16	RO		xCORE tile number on the switch.
15:9	RO	-	Reserved
8	RO		Set to 1 if boot from OTP is enabled.
7:0	RO		The boot mode pins MODE0, MODE1, ..., specifying the boot frequency, boot source, etc.

B.5 Security configuration: 0x05

Copy of the security register as read from OTP.

0x05: Security configuration

Bits	Perm	Init	Description
31:0	RO		Value.

B.6 Ring Oscillator Control: 0x06

There are four free-running oscillators that clock four counters. The oscillators can be started and stopped using this register. The counters should only be read when the ring oscillator is stopped. The counter values can be read using four subsequent registers. The ring oscillators are asynchronous to the xCORE tile clock and can be used as a source of random bits.

0x06: Ring Oscillator Control

Bits	Perm	Init	Description
31:2	RO	-	Reserved
1	RW	0	Set to 1 to enable the xCORE tile ring oscillators
0	RW	0	Set to 1 to enable the peripheral ring oscillators

B.7 Ring Oscillator Value: 0x07

This register contains the current count of the xCORE Tile Cell ring oscillator. This value is not reset on a system reset.

0x07: Ring Oscillator Value

Bits	Perm	Init	Description
31:16	RO	-	Reserved
15:0	RO	-	Ring oscillator counter data.

B.8 Ring Oscillator Value: 0x08

This register contains the current count of the xCORE Tile Wire ring oscillator. This value is not reset on a system reset.

0x08: Ring Oscillator Value

Bits	Perm	Init	Description
31:16	RO	-	Reserved
15:0	RO	-	Ring oscillator counter data.

B.9 Ring Oscillator Value: 0x09

This register contains the current count of the Peripheral Cell ring oscillator. This value is not reset on a system reset.

0x09: Ring Oscillator Value

Bits	Perm	Init	Description
31:16	RO	-	Reserved
15:0	RO	-	Ring oscillator counter data.

B.10 Ring Oscillator Value: 0x0A

This register contains the current count of the Peripheral Wire ring oscillator. This value is not reset on a system reset.

0x0A: Ring Oscillator Value

Bits	Perm	Init	Description
31:16	RO	-	Reserved
15:0	RO	-	Ring oscillator counter data.

B.11 Debug SSR: 0x10

This register contains the value of the SSR register when the debugger was called.

0x10: Debug SSR

Bits	Perm	Init	Description
31:0	RO	-	Reserved

B.12 Debug SPC: 0x11

This register contains the value of the SPC register when the debugger was called.

0x11: Debug SPC

Bits	Perm	Init	Description
31:0	DRW		Value.

B.13 Debug SSP: 0x12

This register contains the value of the SSP register when the debugger was called.

0x12: Debug SSP

Bits	Perm	Init	Description
31:0	DRW		Value.

B.14 DGETREG operand 1: 0x13

The resource ID of the logical core whose state is to be read.

0x13: DGETREG operand 1

Bits	Perm	Init	Description
31:8	RO	-	Reserved
7:0	DRW		Thread number to be read

B.15 DGETREG operand 2: 0x14

Register number to be read by DGETREG

0x14: DGETREG operand 2

Bits	Perm	Init	Description
31:5	RO	-	Reserved
4:0	DRW		Register number to be read

B.16 Debug interrupt type: 0x15

Register that specifies what activated the debug interrupt.

Bits	Perm	Init	Description
31:18	RO	-	Reserved
17:16	DRW		If the debug interrupt was caused by a hardware breakpoint or hardware watchpoint, this field contains the number of the breakpoint or watchpoint. If multiple breakpoints or watchpoints trigger at once, the lowest number is taken.
15:8	DRW		If the debug interrupt was caused by a logical core, this field contains the number of that core. Otherwise this field is 0.
7:3	RO	-	Reserved
2:0	DRW	0	Indicates the cause of the debug interrupt 1: Host initiated a debug interrupt through JTAG 2: Program executed a DCALL instruction 3: Instruction breakpoint 4: Data watch point 5: Resource watch point

0x15: Debug interrupt type

B.17 Debug interrupt data: 0x16

On a data watchpoint, this register contains the effective address of the memory operation that triggered the debugger. On a resource watchpoint, it countains the resource identifier.

0x16: Debug interrupt data

Bits	Perm	Init	Description
31:0	DRW		Value.

B.18 Debug core control: 0x18

This register enables the debugger to temporarily disable logical cores. When returning from the debug interrupts, the cores set in this register will not execute. This enables single stepping to be implemented.

0x18: Debug core control

Bits	Perm	Init	Description
31:8	RO	-	Reserved
7:0	DRW		1-hot vector defining which logical cores are stopped when not in debug mode. Every bit which is set prevents the respective logical core from running.

B.19 Debug scratch: 0x20 .. 0x27

A set of registers used by the debug ROM to communicate with an external debugger, for example over JTAG. This is the same set of registers as the Debug Scratch registers in the xCORE tile configuration.

0x20 .. 0x27: Debug scratch

Bits	Perm	Init	Description
31:0	DRW		Value.

B.20 Instruction breakpoint address: 0x30 .. 0x33

This register contains the address of the instruction breakpoint. If the PC matches this address, then a debug interrupt will be taken. There are four instruction breakpoints that are controlled individually.

0x30 .. 0x33: Instruction breakpoint address

Bits	Perm	Init	Description
31:0	DRW		Value.

B.21 Instruction breakpoint control: 0x40 .. 0x43

This register controls which logical cores may take an instruction breakpoint, and under which condition.

0x40 .. 0x43: Instruction breakpoint control

Bits	Perm	Init	Description
31:24	RO	-	Reserved
23:16	DRW	0	A bit for each logical core in the tile allowing the breakpoint to be enabled individually for each logical core.
15:2	RO	-	Reserved
1	DRW	0	Set to 1 to cause an instruction breakpoint if the PC is not equal to the breakpoint address. By default, the breakpoint is triggered when the PC is equal to the breakpoint address.
0	DRW	0	When 1 the instruction breakpoint is enabled.

B.22 Data watchpoint address 1: 0x50 .. 0x53

This set of registers contains the first address for the four data watchpoints.

0x50 .. 0x53: Data watchpoint address 1

Bits	Perm	Init	Description
31:0	DRW		Value.

B.23 Data watchpoint address 2: 0x60 .. 0x63

This set of registers contains the second address for the four data watchpoints.

0x60 .. 0x63: Data watchpoint address 2

Bits	Perm	Init	Description
31:0	DRW		Value.

B.24 Data breakpoint control register: 0x70 .. 0x73

This set of registers controls each of the four data watchpoints.

0x70 .. 0x73: Data breakpoint control register

Bits	Perm	Init	Description
31:24	RO	-	Reserved
23:16	DRW	0	A bit for each logical core in the tile allowing the breakpoint to be enabled individually for each logical core.
15:3	RO	-	Reserved
2	DRW	0	Set to 1 to enable breakpoints to be triggered on loads. Breakpoints always trigger on stores.
1	DRW	0	By default, data watchpoints trigger if memory in the range [Address1..Address2] is accessed (the range is inclusive of Address1 and Address2). If set to 1, data watchpoints trigger if memory outside the range (Address2..Address1) is accessed (the range is exclusive of Address2 and Address1).
0	DRW	0	When 1 the instruction breakpoint is enabled.

B.25 Resources breakpoint mask: 0x80 .. 0x83

This set of registers contains the mask for the four resource watchpoints.

0x80 .. 0x83: Resources breakpoint mask

Bits	Perm	Init	Description
31:0	DRW		Value.

B.26 Resources breakpoint value: 0x90 .. 0x93

This set of registers contains the value for the four resource watchpoints.

0x90 .. 0x93: Resources breakpoint value

Bits	Perm	Init	Description
31:0	DRW		Value.

B.27 Resources breakpoint control register: 0x9C .. 0x9F

This set of registers controls each of the four resource watchpoints.

0x9C .. 0x9F: Resources breakpoint control register

Bits	Perm	Init	Description
31:24	RO	-	Reserved
23:16	DRW	0	A bit for each logical core in the tile allowing the breakpoint to be enabled individually for each logical core.
15:2	RO	-	Reserved
1	DRW	0	By default, resource watchpoints trigger when the resource id masked with the set Mask equals the Value. If set to 1, resource watchpoints trigger when the resource id masked with the set Mask is not equal to the Value.
0	DRW	0	When 1 the instruction breakpoint is enabled.

C xCORE Tile Configuration

The xCORE Tile control registers can be accessed using configuration reads and writes (use `write_tile_config_reg(tileref, ...)` and `read_tile_config_reg(tileref,` ↪ `...)` for reads and writes).

Number	Perm	Description
0x00	RO	Device identification
0x01	RO	xCORE Tile description 1
0x02	RO	xCORE Tile description 2
0x04	CRW	Control PSwitch permissions to debug registers
0x05	CRW	Cause debug interrupts
0x06	RW	xCORE Tile clock divider
0x07	RO	Security configuration
0x10 .. 0x13	RO	PLink status
0x20 .. 0x27	CRW	Debug scratch
0x40	RO	PC of logical core 0
0x41	RO	PC of logical core 1
0x42	RO	PC of logical core 2
0x43	RO	PC of logical core 3
0x44	RO	PC of logical core 4
0x45	RO	PC of logical core 5
0x60	RO	SR of logical core 0
0x61	RO	SR of logical core 1
0x62	RO	SR of logical core 2
0x63	RO	SR of logical core 3
0x64	RO	SR of logical core 4
0x65	RO	SR of logical core 5
0x80 .. 0x9F	RO	Chanend status

Figure 45: Summary

C.1 Device identification: 0x00

0x00:
Device identification

Bits	Perm	Init	Description
31:24	RO		Processor ID of this xCORE tile.
23:16	RO		Number of the node in which this xCORE tile is located.
15:8	RO		xCORE tile revision.
7:0	RO		xCORE tile version.

C.2 xCORE Tile description 1: 0x01

This register describes the number of logical cores, synchronisers, locks and channel ends available on this xCORE tile.

0x01:
xCORE Tile description 1

Bits	Perm	Init	Description
31:24	RO		Number of channel ends.
23:16	RO		Number of locks.
15:8	RO		Number of synchronisers.
7:0	RO	-	Reserved

C.3 xCORE Tile description 2: 0x02

This register describes the number of timers and clock blocks available on this xCORE tile.

0x02:
xCORE Tile description 2

Bits	Perm	Init	Description
31:16	RO	-	Reserved
15:8	RO		Number of clock blocks.
7:0	RO		Number of timers.

C.4 Control PSwitch permissions to debug registers: 0x04

This register can be used to control whether the debug registers (marked with permission CRW) are accessible through the tile configuration registers. When this bit is set, write -access to those registers is disabled, preventing debugging of the xCORE tile over the interconnect.

0x04: Control PSwitch permissions to debug registers

Bits	Perm	Init	Description
31:1	RO	-	Reserved
0	CRW		Set to 1 to restrict PSwitch access to all CRW marked registers to become read-only rather than read-write.

C.5 Cause debug interrupts: 0x05

This register can be used to raise a debug interrupt in this xCORE tile.

0x05: Cause debug interrupts

Bits	Perm	Init	Description
31:2	RO	-	Reserved
1	RO	0	Set to 1 when the processor is in debug mode.
0	CRW	0	Set to 1 to request a debug interrupt on the processor.

C.6 xCORE Tile clock divider: 0x06

This register contains the value used to divide the PLL clock to create the xCORE tile clock. The divider is enabled under control of the tile control register

0x06: xCORE Tile clock divider

Bits	Perm	Init	Description
31:8	RO	-	Reserved
7:0	RW		Value of the clock divider minus one.

C.7 Security configuration: 0x07

Copy of the security register as read from OTP.

0x07: Security configuration

Bits	Perm	Init	Description
31:0	RO		Value.

C.8 PLink status: 0x10 .. 0x13

Status of each of the four processor links; connecting the xCORE tile to the switch.

0x10 .. 0x13: PLink status

Bits	Perm	Init	Description
31:26	RO	-	Reserved
25:24	RO		00 - ChannelEnd, 01 - ERROR, 10 - PSCTL, 11 - Idle.
23:16	RO		Based on SRC_TARGET_TYPE value, it represents channelEnd ID or Idle status.
15:6	RO	-	Reserved
5:4	RO		Two-bit network identifier
3	RO	-	Reserved
2	RO		1 when the current packet is considered junk and will be thrown away.
1	RO	0	Set to 1 if the switch is routing data into the link, and if a route exists from another link.
0	RO	0	Set to 1 if the link is routing data into the switch, and if a route is created to another link on the switch.

C.9 Debug scratch: 0x20 .. 0x27

A set of registers used by the debug ROM to communicate with an external debugger, for example over the switch. This is the same set of registers as the Debug Scratch registers in the processor status.

0x20 .. 0x27: Debug scratch

Bits	Perm	Init	Description
31:0	CRW		Value.

C.10 PC of logical core 0: 0x40

Value of the PC of logical core 0.

0x40: PC of logical core 0

Bits	Perm	Init	Description
31:0	RO		Value.

C.11 PC of logical core 1: 0x41

0x41:
PC of logical core 1

Bits	Perm	Init	Description
31:0	RO		Value.

C.12 PC of logical core 2: 0x42

0x42:
PC of logical core 2

Bits	Perm	Init	Description
31:0	RO		Value.

C.13 PC of logical core 3: 0x43

0x43:
PC of logical core 3

Bits	Perm	Init	Description
31:0	RO		Value.

C.14 PC of logical core 4: 0x44

0x44:
PC of logical core 4

Bits	Perm	Init	Description
31:0	RO		Value.

C.15 PC of logical core 5: 0x45

0x45:
PC of logical core 5

Bits	Perm	Init	Description
31:0	RO		Value.

C.16 SR of logical core 0: 0x60

Value of the SR of logical core 0

0x60: SR of logical core 0

Bits	Perm	Init	Description
31:0	RO		Value.

C.17 SR of logical core 1: 0x61

0x61: SR of logical core 1

Bits	Perm	Init	Description
31:0	RO		Value.

C.18 SR of logical core 2: 0x62

0x62: SR of logical core 2

Bits	Perm	Init	Description
31:0	RO		Value.

C.19 SR of logical core 3: 0x63

0x63: SR of logical core 3

Bits	Perm	Init	Description
31:0	RO		Value.

C.20 SR of logical core 4: 0x64

0x64: SR of logical core 4

Bits	Perm	Init	Description
31:0	RO		Value.

C.21 SR of logical core 5: 0x65

0x65: SR of logical core 5

Bits	Perm	Init	Description
31:0	RO		Value.

C.22 Chanend status: 0x80 .. 0x9F

These registers record the status of each channel-end on the tile.

0x80 .. 0x9F: Chanend status

Bits	Perm	Init	Description
31:26	RO	-	Reserved
25:24	RO		00 - ChannelEnd, 01 - ERROR, 10 - PSCTL, 11 - Idle.
23:16	RO		Based on SRC_TARGET_TYPE value, it represents channelEnd ID or Idle status.
15:6	RO	-	Reserved
5:4	RO		Two-bit network identifier
3	RO	-	Reserved
2	RO		1 when the current packet is considered junk and will be thrown away.
1	RO	0	Set to 1 if the switch is routing data into the link, and if a route exists from another link.
0	RO	0	Set to 1 if the link is routing data into the switch, and if a route is created to another link on the switch.

D Digital Node Configuration

The digital node control registers can be accessed using configuration reads and writes (use `write_node_config_reg(device, ...)` and `read_node_config_reg(device,` ↪ `...)` for reads and writes).

Number	Perm	Description
0x00	RO	Device identification
0x01	RO	System switch description
0x04	RW	Switch configuration
0x05	RW	Switch node identifier
0x06	RW	PLL settings
0x07	RW	System switch clock divider
0x08	RW	Reference clock
0x0C	RW	Directions 0-7
0x0D	RW	Directions 8-15
0x10	RW	DEBUG_N configuration
0x1F	RO	Debug source
0x20 .. 0x27	RW	Link status, direction, and network
0x40 .. 0x43	RW	PLink status and network
0x80 .. 0x87	RW	Link configuration and initialization
0xA0 .. 0xA7	RW	Static link configuration

Figure 46: Summary

D.1 Device identification: 0x00

This register contains version and revision identifiers and the mode-pins as sampled at boot-time.

Bits	Perm	Init	Description
31:24	RO	0x00	Chip identifier.
23:16	RO		Sampled values of pins MODE0, MODE1, ... on reset.
15:8	RO		SSwitch revision.
7:0	RO		SSwitch version.

0x00: Device identification

D.2 System switch description: 0x01

This register specifies the number of processors and links that are connected to this switch.

0x01: System switch description

Bits	Perm	Init	Description
31:24	RO	-	Reserved
23:16	RO		Number of links on the switch.
15:8	RO		Number of cores that are connected to this switch.
7:0	RO		Number of links per processor.

D.3 Switch configuration: 0x04

This register enables the setting of two security modes (that disable updates to the PLL or any other registers) and the header-mode.

0x04: Switch configuration

Bits	Perm	Init	Description
31	RO	0	Set to 1 to disable any write access to the configuration registers in this switch.
30:9	RO	-	Reserved
8	RO	0	Set to 1 to disable updates to the PLL configuration register.
7:1	RO	-	Reserved
0	RO	0	Header mode. Set to 1 to enable 1-byte headers. This must be performed on all nodes in the system.

D.4 Switch node identifier: 0x05

This register contains the node identifier.

0x05: Switch node identifier

Bits	Perm	Init	Description
31:16	RO	-	Reserved
15:0	RW	0	The unique 16-bit ID of this node. This ID is matched most-significant-bit first with incoming messages for routing purposes.

D.5 PLL settings: 0x06

An on-chip PLL multiplies the input clock up to a higher frequency clock, used to clock the I/O, processor, and switch, see Oscillator. Note: a write to this register will cause the tile to be reset.

0x06: PLL settings

Bits	Perm	Init	Description
31:26	RO	-	Reserved
25:23	RW		OD: Output divider value The initial value depends on pins MODE0 and MODE1.
22:21	RO	-	Reserved
20:8	RW		F: Feedback multiplication ratio The initial value depends on pins MODE0 and MODE1.
7	RO	-	Reserved
6:0	RW		R: Oscilator input divider value The initial value depends on pins MODE0 and MODE1.

D.6 System switch clock divider: 0x07

Sets the ratio of the PLL clock and the switch clock.

0x07: System switch clock divider

Bits	Perm	Init	Description
31:16	RO	-	Reserved
15:0	RW	0	Switch clock divider. The PLL clock will be divided by this value plus one to derive the switch clock.

D.7 Reference clock: 0x08

Sets the ratio of the PLL clock and the reference clock used by the node.

0x08: Reference clock

Bits	Perm	Init	Description
31:16	RO	-	Reserved
15:0	RW	3	Architecture reference clock divider. The PLL clock will be divided by this value plus one to derive the 100 MHz reference clock.

D.8 Directions 0-7: 0x0C

This register contains eight directions, for packets with a mismatch in bits 7..0 of the node-identifier. The direction in which a packet will be routed is goverened by the most significant mismatching bit.

0x0C: Directions 0-7

Bits	Perm	Init	Description
31:28	RW	0	The direction for packets whose first mismatching bit is 7.
27:24	RW	0	The direction for packets whose first mismatching bit is 6.
23:20	RW	0	The direction for packets whose first mismatching bit is 5.
19:16	RW	0	The direction for packets whose first mismatching bit is 4.
15:12	RW	0	The direction for packets whose first mismatching bit is 3.
11:8	RW	0	The direction for packets whose first mismatching bit is 2.
7:4	RW	0	The direction for packets whose first mismatching bit is 1.
3:0	RW	0	The direction for packets whose first mismatching bit is 0.

D.9 Directions 8-15: 0x0D

This register contains eight directions, for packets with a mismatch in bits 15..8 of the node-identifier. The direction in which a packet will be routed is goverened by the most significant mismatching bit.

0x0D: Directions 8-15

Bits	Perm	Init	Description
31:28	RW	0	The direction for packets whose first mismatching bit is 15.
27:24	RW	0	The direction for packets whose first mismatching bit is 14.
23:20	RW	0	The direction for packets whose first mismatching bit is 13.
19:16	RW	0	The direction for packets whose first mismatching bit is 12.
15:12	RW	0	The direction for packets whose first mismatching bit is 11.
11:8	RW	0	The direction for packets whose first mismatching bit is 10.
7:4	RW	0	The direction for packets whose first mismatching bit is 9.
3:0	RW	0	The direction for packets whose first mismatching bit is 8.

D.10 DEBUG_N configuration: 0x10

Configures the behavior of the DEBUG_N pin.

0x10: DEBUG_N configuration

Bits	Perm	Init	Description
31:2	RO	-	Reserved
1	RW	0	Set to 1 to enable signals on DEBUG_N to generate DCALL on the core.
0	RW	0	When set to 1, the DEBUG_N wire will be pulled down when the node enters debug mode.

D.11 Debug source: 0x1F

Contains the source of the most recent debug event.

0x1F:
Debug source

Bits	Perm	Init	Description
31:5	RO	-	Reserved
4	RW		If set, the external DEBUG_N pin is the source of the most recent debug interrupt.
3:1	RO	-	Reserved
0	RW		If set, the xCORE Tile is the source of the most recent debug interrupt.

D.12 Link status, direction, and network: 0x20 .. 0x27

These registers contain status information for low level debugging (read-only), the network number that each link belongs to, and the direction that each link is part of. The registers control links C, D, A, B, G, H, E, and F in that order.

0x20 .. 0x27:
Link status, direction, and network

Bits	Perm	Init	Description
31:26	RO	-	Reserved
25:24	RO		If this link is currently routing data into the switch, this field specifies the type of link that the data is routed to: 0: plink 1: external link 2: internal control link
23:16	RO	0	If the link is routing data into the switch, this field specifies the destination link number to which all tokens are sent.
15:12	RO	-	Reserved
11:8	RW	0	The direction that this this link is associated with; set for routing.
7:6	RO	-	Reserved
5:4	RW	0	Determines the network to which this link belongs, set for quality of service.
3	RO	-	Reserved
2	RO	0	Set to 1 if the current packet is junk and being thrown away. A packet is considered junk if, for example, it is not routable.
1	RO	0	Set to 1 if the switch is routing data into the link, and if a route exists from another link.
0	RO	0	Set to 1 if the link is routing data into the switch, and if a route is created to another link on the switch.

D.13 PLink status and network: 0x40 .. 0x43

These registers contain status information and the network number that each processor-link belongs to.

0x40 .. 0x43: PLink status and network

Bits	Perm	Init	Description
31:26	RO	-	Reserved
25:24	RO		If this link is currently routing data into the switch, this field specifies the type of link that the data is routed to: 0: plink 1: external link 2: internal control link
23:16	RO	0	If the link is routing data into the switch, this field specifies the destination link number to which all tokens are sent.
15:6	RO	-	Reserved
5:4	RW	0	Determines the network to which this link belongs, set for quality of service.
3	RO	-	Reserved
2	RO	0	Set to 1 if the current packet is junk and being thrown away. A packet is considered junk if, for example, it is not routable.
1	RO	0	Set to 1 if the switch is routing data into the link, and if a route exists from another link.
0	RO	0	Set to 1 if the link is routing data into the switch, and if a route is created to another link on the switch.

D.14 Link configuration and initialization: 0x80 .. 0x87

These registers contain configuration and debugging information specific to external links. The link speed and width can be set, the link can be initialized, and the link status can be monitored. The registers control links C, D, A, B, G, H, E, and F in that order.

0x80 .. 0x87: Link configuration and initialization

Bits	Perm	Init	Description
31	RW	0	Write '1' to this bit to enable the link, write '0' to disable it. This bit controls the muxing of ports with overlapping links.
30	RW	0	Set to 0 to operate in 2 wire mode or 1 to operate in 5 wire mode
29:28	RO	-	Reserved
27	RO	0	Set to 1 on error: an RX buffer overflow or illegal token encoding has been received. This bit clears on reading.
26	RO	0	1 if this end of the link has issued credit to allow the remote end to transmit.
25	RO	0	1 if this end of the link has credits to allow it to transmit.
24	WO	0	Set to 1 to initialize a half-duplex link. This clears this end of the link's credit and issues a HELLO token; the other side of the link will reply with credits. This bit is self-clearing.
23	WO	0	Set to 1 to reset the receiver. The next symbol that is detected will be assumed to be the first symbol in a token. This bit is self-clearing.
22	RO	-	Reserved
21:11	RW	0	The number of system clocks between two subsequent transitions within a token
10:0	RW	0	The number of system clocks between two subsequent transmit tokens.

D.15 Static link configuration: 0xA0 .. 0xA7

These registers are used for static (ie, non-routed) links. When a link is made static, all traffic is forwarded to the designated channel end and no routing is attempted. The registers control links C, D, A, B, G, H, E, and F in that order.

0xA0 .. 0xA7: Static link configuration

Bits	Perm	Init	Description
31	RW	0	Enable static forwarding.
30:5	RO	-	Reserved
4:0	RW	0	The destination channel end on this node that packets received in static mode are forwarded to.

E Analogue Node Configuration

The analogue node control registers can be accessed using configuration reads and writes (use `write_node_config_reg(device, ...)` and `read_node_config_reg(device,` ↪ `...)` for reads and writes).

Figure 47: Summary

Number	Perm	Description
0x00	RO	Device identification register
0x04	RW	Node configuration register
0x05	RW	Node identifier
0x50	RW	Reset and Mode Control
0x51	RW	System clock frequency
0x80	RW	Link Control and Status
0xD6	RW	1 KHz Watchdog Control
0xD7	RW	Watchdog Disable

E.1 Device identification register: 0x00

This register contains version information, and information on power-on behavior.

0x00: Device identification register

Bits	Perm	Init	Description
31:24	RO	0x0F	Chip identifier
23:17	RO	-	Reserved
16	RO	pin	Oscillator used on power-up. This is set by the OSC_EXT_N pin: 0: boot from crystal; 1: boot from on-silicon 20 MHz oscillator.
15:8	RO	0x02	Revision number of the analogue block
7:0	RO	0x00	Version number of the analogue block

E.2 Node configuration register: 0x04

This register is used to set the communication model to use (1 or 3 byte headers), and to prevent any further updates.

0x04: Node configuration register

Bits	Perm	Init	Description
31	RW	0	Set to 1 to disable further updates to the node configuration and link control and status registers.
30:1	RO	-	Reserved
0	RW	0	Header mode. 0: 3-byte headers; 1: 1-byte headers.

E.3 Node identifier: 0x05

0x05: Node identifier

Bits	Perm	Init	Description
31:16	RO	-	Reserved
15:0	RW	0	16-bit node identifier. This does not need to be set, and is present for compatibility with XS1-switches.

E.4 Reset and Mode Control: 0x50

The XS1-S has two main reset signals: a system-reset and an xCORE Tile-reset. System-reset resets the whole system including external devices, whilst xCORE Tile-reset resets the xCORE Tile(s) only. The resets are induced either by software (by a write to the register below) or by one of the following:

* External reset on RST_N (System reset)

* Brown out on one of the power supplies (System reset)

* Watchdog timer (System reset)

* Sleep sequence (xCORE Tile reset)

* Clock source change (xCORE Tile reset)

The minimum system reset duration is achieved when the fastest permissible clock is used. The reset durations will be proportionately longer when a slower clock is used. Note that the minimum system reset duration allows for all power rails except the VOUT2 to turn off, and decay.

The length of the system reset comes from an internal counter, counting 524,288 oscillator clock cycles which gives the maximum time allowable for the supply rails to discharge. The system reset duration is a balance between leaving a long time for the supply rails to discharge, and a short time for the system to boot. Example reset times are 44 ms with a 12 MHz oscillator or 5.5 ms with a 96 MHz oscillator.

Bits	Perm	Init	Description
31:25	RO	-	Reserved
24	RW		Tristate processor mode pins.
23:18	RO	-	Reserved
17:16	RW		Processor mode pins.
15:4	RO	-	Reserved
3	RW	0	USB peripheral register access enable.
2	RW	0	USB interface block enable. Set to 1 to enable. Set to 0 to disable and reset all USB interface registers
1	WO	0	xCORE Tile reset. Set to 1 to initiate a reset of the xCORE Tile. This bit is self clearing. A write to this configuration register with this bit asserted results in no response packet being sent to the sender regardless of whether or not a response was requested.
0	WO	0	System reset. Set to 1 to initiate a reset whose scope includes most configuration and peripheral control registers. This bit is self clearing. A write to this configuration register with this bit asserted results in no response packet being sent to the sender regardless of whether or not a response was requested.

0x50: Reset and Mode Control

E.5 System clock frequency: 0x51

Bits	Perm	Init	Description
31:7	RO	-	Reserved
6:0	RW	25	Oscillator clock frequency in MHz rounded up to the nearest integer value. Only values between 5 and 100 MHz are valid - writes outside this range are ignored and will be NACKed. This field must be set on start up of the device and any time that the input oscillator clock frequency is changed. It must contain the system clock frequency in MHz rounded up to the nearest integer value. The following functions depend on the correct frequency settings: * Processor reset delay * The watchdog clock * The real-time clock when running in sleep mode * The USB clock (USB requires a 12, 24, 48, or 96 MHz oscillator)

0x51: System clock frequency

E.6 Link Control and Status: 0x80

0x80:
Link Control
and Status

Bits	Perm	Init	Description
31:28	RO	-	Reserved
27	RO	0	Set to 1 on error: an RX buffer overflow or illegal token encoding has been received. This bit clears on reading.
26	RO	0	1 if this end of the link has issued credit to allow the remote end to transmit.
25	RO	0	1 if this end of the link has credits to allow it to transmit.
24	WO	0	Set to 1 to initialize a half-duplex link. This clears this end of the link's credit and issues a HELLO token; the other side of the link will reply with credits. This bit is self-clearing.
23	WO	0	Set to 1 to reset the receiver. The next symbol that is detected will be assumed to be the first symbol in a token. This bit is self-clearing.
22	RO	-	Reserved
21:11	RW	1	The number of system clocks between two subsequent transitions within a token
10:0	RW	1	The number of system clocks between two subsequent transmit tokens.

E.7 1 KHz Watchdog Control: 0xD6

The watchdog provides a mechanism to prevent programs from hanging by resetting the xCORE Tile after a pre-set time. The watchdog should be periodically "kicked" by the application, causing the count-down to be restarted. If the watchdog expires, it may be due to a program hanging, for example because of a (transient) hardware issue.

The watchdog timeout is measured in 1 ms clock ticks, meaning that a time between 1 ms and 65 seconds can be set for the timeout. The watchdog timer is only clocked during the AWAKE power state. When writing the timeout value, both the timeout and its one's complement should be written. This reduces the chances of accidentally setting kicking the watchdog. If the written value does not comprise a 16-bit value with a 16-bit one's complement, the request will be NACKed, otherwise an ACK will be sent.

If the watchdog expires, the xCORE Tile is reset.

0xD6:
1 KHz
Watchdog
Control

Bits	Perm	Init	Description
31:16	RO	0	Current value of watchdog timer.
15:0	RW	1000	Number of 1kHz cycles after which the watchdog should expire and initiate a system reset.

E.8 Watchdog Disable: 0xD7

To enable the watchdog, write 0 to this register. To disable the watchdog, write the value 0x0D1SAB1E to this register.

0xD7:
Watchdog
Disable

Bits	Perm	Init	Description
31:0	RW	0x0D15AB1E	A value of 0x0D15AB1E written to this register resets and disables the watchdog timer.

F USB PHY Configuration

The USB PHY is connected to the following ports:

XS1_PORT_1J
Clk

XS1_PORT_1K
Tx ready out (Tx valid)

XS1_PORT_1H
Tx ready in

XS1_PORT_8A
Tx data

XS1_PORT_1M
Rx ready

XS1_PORT_8C
Rx data

XS1_PORT_1N
flag1

XS1_PORT_1O
flag2

XS1_PORT_1P
flag3

The *USB PHY* is peripheral 1. The control registers are accessed using 32-bit reads and writes (use `write_periph_32(device, 1, ...)` and `read_periph_32(device, ↪ 1, ...)` for reads and writes).

Number	Perm	Description
0x00	WO	UIFM reset
0x04	RW	UIFM IFM control
0x08	RW	UIFM Device Address
0x0C	RW	UIFM functional control
0x10	RW	UIFM on-the-go control
0x14	RO	UIFM on-the-go flags
0x18	RW	UIFM Serial Control
0x1C	RW	UIFM signal flags
0x20	RW	UIFM Sticky flags
0x24	RW	UIFM port masks
0x28	RW	UIFM SOF value
0x2C	RO	UIFM PID
0x30	RO	UIFM Endpoint
0x34	RW	UIFM Endpoint match
0x38	RW	UIFM power signalling
0x3C	RW	UIFM PHY control

Figure 48: Summary

F.1 UIFM reset: 0x00

A write to this register with any data resets all UIFM state, but does not otherwise affect the phy.

0x00: UIFM reset

Bits	Perm	Init	Description
31:0	WO		Value.

F.2 UIFM IFM control: 0x04

General settings of the UIFM IFM state machine.

0x04: UIFM IFM control

Bits	Perm	Init	Description
31:8	RO	-	Reserved
7	RW	0	Set to 1 to enable XEVACKMODE mode.
6	RW	0	Set to 1 to enable SOFISTOKEN mode.
5	RW	0	Set to 1 to enable UIFM power signalling mode.
4	RW	0	Set to 1 to enable IF timing mode.
3	RO	-	Reserved
2	RW	0	Set to 1 to enable UIFM linestate decoder.
1	RW	0	Set to 1 to enable UIFM CHECKTOKENS mode.
0	RW	0	Set to 1 to enable UIFM DOTOKENS mode.

F.3 UIFM Device Address: 0x08

The device address whose packets should be received. 0 until enumeration, it should be set to the assigned value after enumeration.

0x08: UIFM Device Address

Bits	Perm	Init	Description
31:7	RO	-	Reserved
6:0	RW	0	The enumerated USB device address must be stored here. Only packets to this address are passed on.

F.4 UIFM functional control: 0x0C

0x0C: UIFM functional control

Bits	Perm	Init	Description
31:5	RO	-	Reserved
4:2	RW	1	Set to 0 to disable UIFM to UTMI+ OPMODE mode.
1	RW	1	Set to 1 to switch UIFM to UTMI+ TERMSELECT mode.
0	RW	1	Set to 1 to switch UIFM to UTMI+ XCVRSELECT mode.

F.5 UIFM on-the-go control: 0x10

This register is used to negotiate an on-the-go connection.

0x10: UIFM on-the-go control

Bits	Perm	Init	Description
31:8	RO	-	Reserved
7	RW	0	Set to 1 to switch UIFM to EXTVBUSIND mode.
6	RW	0	Set to 1 to switch UIFM to DRVVBUSEXT mode.
5	RO	-	Reserved
4	RW	0	Set to 1 to switch UIFM to UTMI+ CHRGVBUS mode.
3	RW	0	Set to 1 to switch UIFM to UTMI+ DISCHRGVBUS mode.
2	RW	0	Set to 1 to switch UIFM to UTMI+ DMPULLDOWN mode.
1	RW	0	Set to 1 to switch UIFM to UTMI+ DPPULLDOWN mode.
0	RW	0	Set to 1 to switch UIFM to IDPULLUP mode.

F.6 UIFM on-the-go flags: 0x14

Status flags used for on-the-go negotiation

0x14: UIFM on-the-go flags

Bits	Perm	Init	Description
31:6	RO	-	Reserved
5	RO	0	Value of UTMI+ Bvalid flag.
4	RO	0	Value of UTMI+ IDGND flag.
3	RO	0	Value of UTMI+ HOSTDIS flag.
2	RO	0	Value of UTMI+ VBUSVLD flag.
1	RO	0	Value of UTMI+ SESSVLD flag.
0	RO	0	Value of UTMI+ SESSEND flag.

F.7 UIFM Serial Control: 0x18

0x18: UIFM Serial Control

Bits	Perm	Init	Description
31:7	RO	-	Reserved
6	RO	0	1 if UIFM is in UTMI+ RXRCV mode.
5	RO	0	1 if UIFM is in UTMI+ RXDM mode.
4	RO	0	1 if UIFM is in UTMI+ RXDP mode.
3	RW	0	Set to 1 to switch UIFM to UTMI+ TXSE0 mode.
2	RW	0	Set to 1 to switch UIFM to UTMI+ TXDATA mode.
1	RW	1	Set to 0 to switch UIFM to UTMI+ TXENABLE mode.
0	RW	0	Set to 1 to switch UIFM to UTMI+ FSLSSERIAL mode.

F.8 UIFM signal flags: 0x1C

Set of flags that monitor line and error states. These flags normally clear on the next packet, but they may be made sticky by using PER_UIFM_FLAGS_STICKY, in which they must be cleared explicitly.

0x1C: UIFM signal flags

Bits	Perm	Init	Description
31:7	RO	-	Reserved
6	RW	0	Set to 1 when the UIFM decodes a token successfully (e.g. it passes CRC5, PID check and has matching device address).
5	RW	0	Set to 1 when linestate indicates an SE0 symbol.
4	RW	0	Set to 1 when linestate indicates a K symbol.
3	RW	0	Set to 1 when linestate indicates a J symbol.
2	RW	0	Set to 1 if an incoming datapacket fails the CRC16 check.
1	RW	0	Set to the value of the UTMI_RXACTIVE input signal.
0	RW	0	Set to the value of the UTMI_RXERROR input signal

F.9 UIFM Sticky flags: 0x20

These bits define the sticky-ness of the bits in the UIFM IFM FLAGS register. A 1 means that bit will be sticky (hold its value until a 1 is written to that bitfield), or normal, in which case signal updates to the UIFM IFM FLAGS bits may be over-written by subsequent changes in those signals.

0x20: UIFM Sticky flags

Bits	Perm	Init	Description
31:7	RO	-	Reserved
6:0	RW	0	Stickyness for each flag.

F.10 UIFM port masks: 0x24

Set of masks that identify how port 1N, port 1O and port 1P are affected by changes to the flags in FLAGS

0x24: UIFM port masks

Bits	Perm	Init	Description
31:23	RO	-	Reserved
22:16	RW	0	Bit mask that determines which flags in UIFM_IFM_FLAG[6:0] contribute to port 1P. If any flag listed in this bitmask is high, port 1P will be high.
15	RO	-	Reserved
14:8	RW	0	Bit mask that determines which flags in UIFM_IFM_FLAG[6:0] contribute to port 1O. If any flag listed in this bitmask is high, port 1O will be high.
7	RO	-	Reserved
6:0	RW	0	Bit mask that determines which flags in UIFM_IFM_FLAG[6:0] contribute to port 1N. If any flag listed in this bitmask is high, port 1N will be high.

F.11 UIFM SOF value: 0x28

USB Start-Of-Frame counter

0x28: UIFM SOF value

Bits	Perm	Init	Description
31:11	RO	-	Reserved
10:8	RW	0	Most significant 3 bits of SOF counter
7:0	RW	0	Least significant 8 bits of SOF counter

F.12 UIFM PID: 0x2C

The last USB packet identifier received

0x2C: UIFM PID

Bits	Perm	Init	Description
31:4	RO	-	Reserved
3:0	RO	0	Value of the last received PID.

F.13 UIFM Endpoint: 0x30

The last endpoint seen

0x30: UIFM Endpoint

Bits	Perm	Init	Description
31:5	RO	-	Reserved
4	RO	0	1 if endpoint contains a valid value.
3:0	RO	0	A copy of the last received endpoint.

F.14 UIFM Endpoint match: 0x34

This register can be used to mark UIFM endpoints as special.

0x34: UIFM Endpoint match

Bits	Perm	Init	Description
31:16	RO	-	Reserved
15:0	RW	0	This register contains a bit for each endpoint. If its bit is set, the endpoint will be supplied on the RX port when ORed with 0x10.

F.15 UIFM power signalling: 0x38

0x38: UIFM power signalling

Bits	Perm	Init	Description
31:9	RO	-	Reserved
8	RW	0	Valid
7:0	RW	0	Data

F.16 UIFM PHY control: 0x3C

Bits	Perm	Init	Description
31:19	RO	-	Reserved
18	RW	0	Set to 1 to disable pulldowns on ports 8A and 8B.
17:14	RO	-	Reserved
13	RW	0	After an auto-resume, this bit is set to indicate that the resume signalling was for reset (se0). Set to 0 to clear.
12	RW	0	After an auto-resume, this bit is set to indicate that the resume signalling was for resume (K). Set to 0 to clear.
11:8	RW	0	Log-2 number of clocks before any linestate change is propagated.
7	RW	0	Set to 1 to use the suspend controller handle to resume from suspend. Otherwise, the program has to poll the linestate_filt field in phy_teststatus.
6:4	RW	0	Control the the conf1,2,3 input pins of the PHY.
3:0	RO	-	Reserved

0x3C: UIFM PHY control

G ADC Configuration

The device has a 12-bit Analogue to Digital Converter (ADC). It has multiple input pins, and on each positive clock edge on port 1I, it samples and converts a value on the next input pin. The data is transmitted to a channel-end that must be set on enabling the ADC input pin.

The *ADC* is peripheral 2. The control registers are accessed using 32-bit reads and writes (use `write_periph_32(device, 2, ...)` and `read_periph_32(device, 2, ...)` for reads and writes).

Number	Perm	Description
0x00	RW	ADC Control input pin 0
0x04	RW	ADC Control input pin 1
0x08	RW	ADC Control input pin 2
0x0C	RW	ADC Control input pin 3
0x20	RW	ADC General Control

Figure 49: Summary

G.1 ADC Control input pin 0: 0x00

Controls specific to ADC input pin 0.

0x00:
ADC Control
input pin 0

Bits	Perm	Init	Description
31:8	RW	0	The node and channel-end identifier to which data for this ADC input pin should be send to. This is the top 24 bits of the channel-end identifier as allocated on an xCORE Tile.
7:1	RO	-	Reserved
0	RW	0	Set to 1 to enable this input pin on the ADC.

G.2 ADC Control input pin 1: 0x04

Controls specific to ADC input pin 1.

0x04:
ADC Control
input pin 1

Bits	Perm	Init	Description
31:8	RW	0	The node and channel-end identifier to which data for this ADC input pin should be send to. This is the top 24 bits of the channel-end identifier as allocated on an xCORE Tile.
7:1	RO	-	Reserved
0	RW	0	Set to 1 to enable this input pin on the ADC.

G.3 ADC Control input pin 2: 0x08

Controls specific to ADC input pin 2.

0x08:
ADC Control
input pin 2

Bits	Perm	Init	Description
31:8	RW	0	The node and channel-end identifier to which data for this ADC input pin should be send to. This is the top 24 bits of the channel-end identifier as allocated on an xCORE Tile.
7:1	RO	-	Reserved
0	RW	0	Set to 1 to enable this input pin on the ADC.

G.4 ADC Control input pin 3: 0x0C

Controls specific to ADC input pin 3.

0x0C:
ADC Control input pin 3

Bits	Perm	Init	Description
31:8	RW	0	The node and channel-end identifier to which data for this ADC input pin should be send to. This is the top 24 bits of the channel-end identifier as allocated on an xCORE Tile.
7:1	RO	-	Reserved
0	RW	0	Set to 1 to enable this input pin on the ADC.

G.5 ADC General Control: 0x20

General ADC control.

Bits	Perm	Init	Description
31:25	RO	-	Reserved
24	RO	1	Indicates that an ADC sample has been dropped. This bit is cleared on a read.
23:18	RO	-	Reserved
17:16	RW	1	Number of bits per ADC sample. The ADC values are always left aligned: 0: 8 bits samples - the least significant four bits of each sample are discarded. 1: 16 bits samples - the sample is padded with four zero bits in bits 3..0. The most significant byte is transmitted first. 2: reserved 3: 32 bits samples - the sample is padded with 20 zero bits in bits 19..0. The most significant byte is transmitted first, hence the word can be input with a single 32-bit IN instruction.
15:8	RW	1	Number of samples to be transmitted per packet. The value 0 indicates that the packet will not be terminated until interrupted by an ADC control register access.
7:2	RO	-	Reserved
1	RW	0	Set to 1 to switch the ADC to sample a 0.8V signal rather than the external voltage. This can be used to calibrate the ADC. When switching to and from calibration mode, one sample value should be discarded. If a sample value `x` is measured in calibration mode, then a scale factor `800000/x` can be used to translate subsequent measurements into microvolts (using integer arithmetic).
0	RW	0	Set to 1 to enable the ADC. Note that when enabled, the ADC control registers above are read-only. The ADC must be disabled whilst setting up the per-input-pin control. On enabling the ADC, six pulses must be generated to calibrate the ADC. These pulses will not generate packets on the selected channel-end. The seventh and further pulses will deliver samples to the selected channel-end. These six pulses have to be issued every time that this bit is changed from 0 to 1.

0x20:
ADC General
Control

H Deep sleep memory Configuration

This peripheral contains a 128 byte RAM that retains state whilst the main processor is put to sleep.

The *Deep sleep memory* is peripheral 3. The control registers are accessed using 8-bit reads and writes (use `write_periph_8(device, 3, ...)` and `read_periph_8` ↪ `(device, 3, ...)` for reads and writes).

Figure 50: Summary

Number	Perm	Description
0x00 .. 0x7F	RW	Deep sleep memory
0xFF	RW	Deep sleep memory valid

H.1 Deep sleep memory: 0x00 .. 0x7F

128 bytes of memory that can be used to hold data when the xCORE Tile is powered down.

0x00 .. 0x7F: Deep sleep memory

Bits	Perm	Init	Description
7:0	RW		User defined data

H.2 Deep sleep memory valid: 0xFF

One byte of memory that is reset to 0. The program can write a non zero value in this register to indicate that the data in deep sleep memory is valid.

0xFF: Deep sleep memory valid

Bits	Perm	Init	Description
7:0	RW	0	User defined data, reset to 0.

I Oscillator Configuration

The *Oscillator* is peripheral 4. The control registers are accessed using 8-bit reads and writes (use `write_periph_8(device, 4, ...)` and `read_periph_8(device, 4, ...)` for reads and writes).

Figure 51: Summary

Number	Perm	Description
0x00	RW	General oscillator control
0x01	RW	On-silicon-oscillator control
0x02	RW	Crystal-oscillator control

I.1 General oscillator control: 0x00

0x00: General oscillator control

Bits	Perm	Init	Description
7:2	RO	-	Reserved
1	RW	0	Set to 1 to reset the xCORE Tile when the value of the oscillator select control register (bit 0) is changed.
0	RW	pin	Selects the oscillator to use: 0: Crystal oscillator 1: On-silicon oscillator

I.2 On-silicon-oscillator control: 0x01

This register controls the on-chip logic that implements an on-chip oscillator. The on-chip oscillator does not require an external crystal, but does not provide an accurate timing source. The nominal frequency of the on-silicon-oscillator is given below, but the actual frequency are temperature, voltage, and chip dependent.

0x01: On-silicon-oscillator control

Bits	Perm	Init	Description
7:2	RO	-	Reserved
1	RW	0	Selects the clock speed of the on-chip oscillator: 0: approximately 20 Mhz (fast clock) 1: approximately 31,250 Hz (slow clock)
0	RW	1	Set to 0 to disable the on-chip oscillator. Do not do this unless the xCORE Tile is running off the crystal oscillator.

I.3 Crystal-oscillator control: 0x02

This register controls the on-chip logic that implements the crystal oscillator; the crystal-oscillator requires an external crystal.

0x02: Crystal-oscillator control

Bits	Perm	Init	Description
7:2	RO	-	Reserved
1	RW	1	Set to 0 to disable the crystal bias circuit. Only switch the bias off if an external oscillator rather than a crystal is connected.
0	RW	1	Set to 0 to disable the crystal oscillator. Do not do this unless the xCORE Tile is running off the on-silicon oscillator.

J Real time clock Configuration

The *Real time clock* is peripheral 5. The control registers are accessed using 32-bit reads and writes (use `write_periph_32(device, 5, ...)` and `read_periph_32(device, ↪ 5, ...)` for reads and writes).

Figure 52: Summary

Number	Perm	Description
0x00	RW	Real time counter least significant 32 bits
0x04	RW	Real time counter most significant 32 bits

J.1 Real time counter least significant 32 bits: 0x00

This registers contains the lower 32-bits of the real-time counter.

0x00: Real time counter least significant 32 bits

Bits	Perm	Init	Description
31:0	RO	0	Least significant 32 bits of real-time counter.

J.2 Real time counter most significant 32 bits: 0x04

This registers contains the upper 32-bits of the real-time counter.

0x04: Real time counter most significant 32 bits

Bits	Perm	Init	Description
31:0	RO	0	Most significant 32 bits of real-time counter.

K Power control block Configuration

The *Power control block* is peripheral 6. The control registers are accessed using 32-bit reads and writes (use `write_periph_32(device, 6, ...)` and `read_periph_32( ↪ device, 6, ...)` for reads and writes).

Number	Perm	Description
0x00	RW	General control
0x04	RW	Time to wake-up, least significant 32 bits
0x08	RW	Time to wake-up, most significant 32 bits
0x0C	RW	Power supply states whilst ASLEEP
0x10	RW	Power supply states whilst WAKING1
0x14	RW	Power supply states whilst WAKING2
0x18	RW	Power supply states whilst AWAKE
0x1C	RW	Power supply states whilst SLEEPING1
0x20	RW	Power supply states whilst SLEEPING2
0x24	RW	Power sequence status
0x2C	RW	DCDC control
0x30	RW	Power supply status
0x34	RW	VDDCORE level control
0x40	RW	LDO5 level control

Figure 53: Summary

K.1 General control: 0x00

This register controls the basic settings for power modes.

0x00: General control

Bits	Perm	Init	Description
31:10	RO	-	Reserved
9	RW	0	Set to 1 to switch USB suspend controller to USB power up enable.
8	RW	0	Set to 1 to switch USB suspend controller to power down enable.
7	RW	0	By default, when waking up, the voltage levels stored in the LEVEL CONTROL registers are used. Set to 1 to use the power-on voltage levels.
6	WO		Set to 1 to re-apply the current contents of the AWAKE state. Use this when the program has changed the contents of the AWAKE state register. Self clearing.
5	RW	0	Set to 1 to use a 64-bit timer.
4	RW	0	Set to 1 to wake-up on the timer.
3	RW	1	If waking on the WAKE pin is enabled (see above), then by default the device wakes up when the WAKE pin is pulled high. Set to 0 to wake-up when the WAKE pin is pulled low.
2	RW	0	Set to 1 to wake-up when the WAKE pin is at the right level.
1	RW	0	Set to 1 to initiate sleep sequence - self clearing. Only set this bit when in AWAKE state.
0	RW	0	Sleep clock select. Set to 1 to use the default clock rather than the internal 31.25 kHz oscillator. Note: this bit is only effective in the ASLEEP state.

K.2 Time to wake-up, least significant 32 bits: 0x04

This register stores the time to wake-up. The value is only used if wake-up from the real-time clock is enabled, and the device is asleep.

0x04: Time to wake-up, least significant 32 bits

Bits	Perm	Init	Description
31:0	RW	0	Least significant 32 bits of time to wake-up.

K.3 Time to wake-up, most significant 32 bits: 0x08

This register stores the time to wake-up. The value is only used if wake-up from the real-time clock is enabled, if 64-bit comparisons are enabled, and the device is asleep. In most cases, 32-bit comparisons suffice.

0x08: Time to wake-up, most significant 32 bits

Bits	Perm	Init	Description
31:0	RW	0	Most significant 32 bits of time to wake-up (ignored unless 64-bit timer comparison is enabled).

K.4 Power supply states whilst ASLEEP: 0x0C

This register controls the state the power control block should be in when in the ASLEEP state. It also defines the minimum time that the system shall stay in this state. When the minimum time is expired, the next state may be entered if either of the wake conditions (real-time counter or WAKE pin) happens. Note that the minimum number of cycles is counted in according to the currently enabled clock, which may be the slow 31 KHz clock.

0x0C: Power supply states whilst ASLEEP

Bits	Perm	Init	Description
31:21	RO	-	Reserved
20:16	RW	16	Log2 number of cycles to stay in this state: 0: 1 clock cycles 1: 2 clock cycles 2: 4 clock cycles ... 31: 2147483648 clock cycles
15	RO	-	Reserved
14	RW	0	Set to 1 to disable clock to the xCORE Tile.
13:10	RO	-	Reserved
9	RW	0	Sets modulation used by DCDC2: 0: PWM modulation (max 475 mA) 1: PFM modulation (max 50 mA)
8	RW	0	Sets modulation used by DCDC1: 0: PWM modulation (max 700 mA) 1: PFM modulation (max 50 mA)
7:6	RO	-	Reserved
5	RW	0	Set to 1 to enable VOUT6 (IO supply).
4	RW	0	Set to 1 to enable LDO5 (core PLL supply).
3:2	RO	-	Reserved
1	RO	0	Set to 1 to enable DCDC2 (analogue supply).
0	RW	0	Set to 1 to enable DCDC1 (core supply).

K.5 Power supply states whilst WAKING1: 0x10

This register controls what state the power control block should be in when in the WAKING1 state. It also defines the minimum time that the system shall stay in this state. When the minimum time is expired, the next state is entered if all enabled power supplies are good.

0x10: Power supply states whilst WAKING1

Bits	Perm	Init	Description
31:21	RO	-	Reserved
20:16	RW	16	Log2 number of cycles to stay in this state: 0: 1 clock cycles 1: 2 clock cycles 2: 4 clock cycles ... 31: 2147483648 clock cycles
15	RO	-	Reserved
14	RW	0	Set to 1 to disable clock to the xCORE Tile.
13:10	RO	-	Reserved
9	RW	0	Sets modulation used by DCDC2: 0: PWM modulation (max 475 mA) 1: PFM modulation (max 50 mA)
8	RW	0	Sets modulation used by DCDC1: 0: PWM modulation (max 700 mA) 1: PFM modulation (max 50 mA)
7:6	RO	-	Reserved
5	RW	1	Set to 1 to enable VOUT6 (IO supply).
4	RW	0	Set to 1 to enable LDO5 (core PLL supply).
3:2	RO	-	Reserved
1	RO	0	Set to 1 to enable DCDC2 (analogue supply).
0	RW	0	Set to 1 to enable DCDC1 (core supply).

K.6 Power supply states whilst WAKING2: 0x14

This register controls what state the power control block should be in when in the WAKING2 state. It also defines the minimum time that the system shall stay in this state. When the minimum time is expired, the next state is entered if all enabled power supplies are good.

0x14: Power supply states whilst WAKING2

Bits	Perm	Init	Description
31:21	RO	-	Reserved
20:16	RW	16	Log2 number of cycles to stay in this state: 0: 1 clock cycles 1: 2 clock cycles 2: 4 clock cycles ... 31: 2147483648 clock cycles
15	RO	-	Reserved
14	RW	0	Set to 1 to disable clock to the xCORE Tile.
13:10	RO	-	Reserved
9	RW	0	Sets modulation used by DCDC2: 0: PWM modulation (max 475 mA) 1: PFM modulation (max 50 mA)
8	RW	0	Sets modulation used by DCDC1: 0: PWM modulation (max 700 mA) 1: PFM modulation (max 50 mA)
7:6	RO	-	Reserved
5	RW	1	Set to 1 to enable VOUT6 (IO supply).
4	RW	1	Set to 1 to enable LDO5 (core PLL supply).
3:2	RO	-	Reserved
1	RO	1	Set to 1 to enable DCDC2 (analogue supply).
0	RW	1	Set to 1 to enable DCDC1 (core supply).

K.7 Power supply states whilst AWAKE: 0x18

This register controls what state the power control block should be in when in the AWAKE state.

0x18: Power supply states whilst AWAKE

Bits	Perm	Init	Description
31:15	RO	-	Reserved
14	RW	0	Set to 1 to disable clock to the xCORE Tile.
13:10	RO	-	Reserved
9	RW	0	Sets modulation used by DCDC2: 0: PWM modulation (max 475 mA) 1: PFM modulation (max 50 mA)
8	RW	0	Sets modulation used by DCDC1: 0: PWM modulation (max 700 mA) 1: PFM modulation (max 50 mA)
7:6	RO	-	Reserved
5	RW	1	Set to 1 to enable VOUT6 (IO supply).
4	RW	1	Set to 1 to enable LDO5 (core PLL supply).
3:2	RO	-	Reserved
1	RO	1	Set to 1 to enable DCDC2 (analogue supply).
0	RW	1	Set to 1 to enable DCDC1 (core supply).

K.8 Power supply states whilst SLEEPING1: 0x1C

This register controls what state the power control block should be in when in the SLEEPING1 state. It also defines the time that the system shall stay in this state.

0x1C: Power supply states whilst SLEEPING1

Bits	Perm	Init	Description
31:21	RO	-	Reserved
20:16	RW	16	Log2 number of cycles to stay in this state: 0: 1 clock cycles 1: 2 clock cycles 2: 4 clock cycles ... 31: 2147483648 clock cycles
15	RO	-	Reserved
14	RW	0	Set to 1 to disable clock to the xCORE Tile.
13:10	RO	-	Reserved
9	RW	0	Sets modulation used by DCDC2: 0: PWM modulation (max 475 mA) 1: PFM modulation (max 50 mA)
8	RW	0	Sets modulation used by DCDC1: 0: PWM modulation (max 700 mA) 1: PFM modulation (max 50 mA)
7:6	RO	-	Reserved
5	RW	1	Set to 1 to enable VOUT6 (IO supply).
4	RW	0	Set to 1 to enable LDO5 (core PLL supply).
3:2	RO	-	Reserved
1	RO	1	Set to 1 to enable DCDC2 (analogue supply).
0	RW	0	Set to 1 to enable DCDC1 (core supply).

K.9 Power supply states whilst SLEEPING2: 0x20

This register controls what state the power control block should be in when in the SLEEPING2 state. It also defines the time that the system shall stay in this state.

0x20:
Power supply states whilst SLEEPING2

Bits	Perm	Init	Description
31:21	RO	-	Reserved
20:16	RW	16	Log2 number of cycles to stay in this state: 0: 1 clock cycles 1: 2 clock cycles 2: 4 clock cycles ... 31: 2147483648 clock cycles
15	RO	-	Reserved
14	RW	0	Set to 1 to disable clock to the xCORE Tile.
13:10	RO	-	Reserved
9	RW	0	Sets modulation used by DCDC2: 0: PWM modulation (max 475 mA) 1: PFM modulation (max 50 mA)
8	RW	0	Sets modulation used by DCDC1: 0: PWM modulation (max 700 mA) 1: PFM modulation (max 50 mA)
7:6	RO	-	Reserved
5	RW	0	Set to 1 to enable VOUT6 (IO supply).
4	RW	0	Set to 1 to enable LDO5 (core PLL supply).
3:2	RO	-	Reserved
1	RO	1	Set to 1 to enable DCDC2 (analogue supply).
0	RW	0	Set to 1 to enable DCDC1 (core supply).

K.10 Power sequence status: 0x24

This register defines the current status of the power supply controller.

0x24: Power sequence status

Bits	Perm	Init	Description
31:30	RO	-	Reserved
29	RO	0	1 if VOUT6 was enabled in the previous state.
28	RO	0	1 if LDO5 was enabled in the previous state.
27:26	RO	-	Reserved
25	RO	1	1 if DCDC2 was enabled in the previous state.
24	RO	0	1 if DCDC1 was enabled in the previous state.
23:19	RO	-	Reserved
18:16	RO		Current state of the power sequence state machine 0: Reset 1: Asleep 2: Waking 1 3: Waking 2 4: Awake Wait 5: Awake 6: Sleeping 1 7: Sleeping 2
15	RO	-	Reserved
14	RO	0	Set to 1 to disable clock to the xCORE Tile.
13:10	RO	-	Reserved
9	RO	0	Sets modulation used by DCDC2: 0: PWM modulation (max 475 mA) 1: PFM modulation (max 50 mA)
8	RO	0	Sets modulation used by DCDC1: 0: PWM modulation (max 700 mA) 1: PFM modulation (max 50 mA)
7:6	RO	-	Reserved
5	RO	0	Set to 1 to enable VOUT6 (IO supply).
4	RO	0	Set to 1 to enable LDO5 (core PLL supply).
3:2	RO	-	Reserved
1	RO	0	Set to 1 to enable DCDC2 (analogue supply).
0	RO	0	Set to 1 to enable DCDC1 (core supply).

K.11 DCDC control: 0x2C

This register controls the two DC-DC converters.

0x2C: DCDC control

Bits	Perm	Init	Description
31:26	RO	-	Reserved
25:24	RW	2	Sets the power good level for VDDCORE and VDD1V8: 0: 0.80 x VDDCORE, 0.80 x VDD1V8 1: 0.85 x VDDCORE, 0.85 x VDD1V8 2: 0.90 x VDDCORE, 0.90 x VDD1V8 3: 0.75 x VDDCORE, 0.75 x VDD1V8
23:17	RO	-	Reserved
16	RW	0	Clear DCDC1 and DCDC2 error flags, not self clearing.
15	RO	-	Reserved
14:13	RW	0	Sets the DCDC2 current limit: 0: 1A 1: 1.5A 2: 2A 3: 0.5A
12:10	RO	-	Reserved
9:8	RW	1	Sets the clock used by DCDC2 to generate VDD1V8: 0: 0.9 MHz 1: 1.0 MHz 2: 1.1 MHz 3: 1.2 MHz
7	RO	-	Reserved
6:5	RW	0	Sets the DCDC1 current limit: 0: 1.2A 1: 1.8A 2: 2.5A 3: 0.8A
4:2	RO	-	Reserved
1:0	RW	1	Sets the clock used by DCDC1 to generate VDDCORE: 0: 0.9 MHz 1: 1.0 MHz 2: 1.1 MHz 3: 1.2 MHz

K.12 Power supply status: 0x30

This register provides the current status of the power supplies.

0x30:
Power supply status

Bits	Perm	Init	Description
31:25	RO	-	Reserved
24	RO		1 if on-silicon oscillator is stable.
23:20	RO	-	Reserved
19	RO		1 if VDDPLL is good.
18:17	RO	-	Reserved
16	RO		1 if VDDCORE is good.
15:10	RO	-	Reserved
9	RO		1 if DCDC2 is in current limiting mode.
8	RO		1 if DCDC1 is in current limiting mode.
7:2	RO	-	Reserved
1	RO		1 if DCDC2 is in soft-start mode.
0	RO		1 if DCDC1 is in soft-start mode.

K.13 VDDCORE level control: 0x34

This register can be used to set the desired voltage on VDDCORE. If the level is to be raised or lowered, it should be raised in steps of no more than 10 mV per microsecond in order to prevent overshoot and undershoot. The default value depends on the MODE pins.

0x34:
VDDCORE level control

Bits	Perm	Init	Description
31:7	RO	-	Reserved
6:0	RW	pin	The required voltage in 10 mV steps: 0: 0.60V 1: 0.61V 2: 0.62V ... 69: 1.29V 70: 1.30V

K.14 LDO5 level control: 0x40

This register can be used to set the desired voltage on LDO5. If the level is to be raised, it should be raised in steps of 1 (100 mV). The default value depends on the MODE pins.

0x40: LDO5 level control

Bits	Perm	Init	Description
31:3	RO	-	Reserved
2:0	RW	pin	The required voltage in 100 mV steps: 0: 0.6V 1: 0.7V 2: 0.8V ... 6: 1.2V 7: 1.3V

L Device Errata

This section describes minor operational differences from the data sheet and recommended workarounds. As device and documentation issues become known, this section will be updated the document revised.

To guarantee a logic low is seen on the pins DEBUG_N, MODE[3:0], TMS, TCK and TDI, the driving circuit should present an impedance of less than 100 Ω to ground. Usually this is not a problem for CMOS drivers driving single inputs. If one or more of these inputs are placed in parallel, however, additional logic buffers may be required to guarantee correct operation.

For static inputs tied high or low, the relevant input pin should be tied directly to GND or VDDIO.

M JTAG, xSCOPE and Debugging

If you intend to design a board that can be used with the XMOS toolchain and xTAG debugger, you will need an xSYS header on your board. Figure 54 shows a decision diagram which explains what type of xSYS connectivity you need. The three subsections below explain the options in detail.

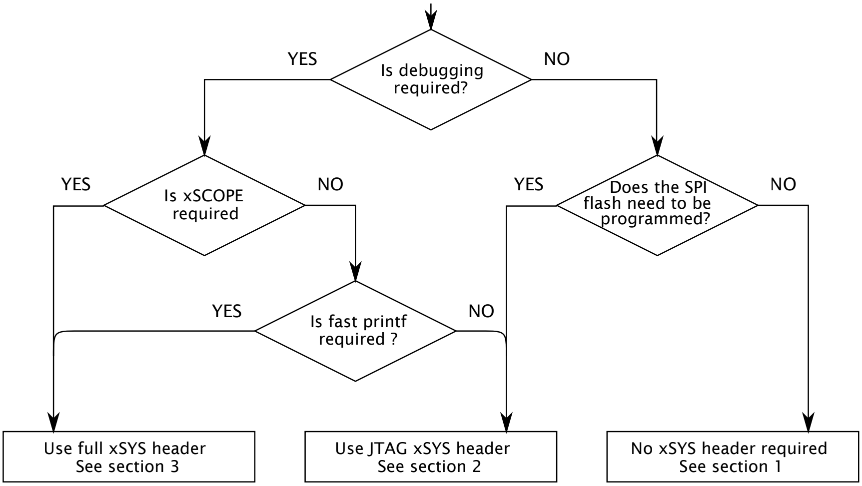


Figure 54: Decision diagram for the xSYS header

M.1 No xSYS header

The use of an xSYS header is optional, and may not be required for volume production designs. However, the XMOS toolchain expects the xSYS header; if you do not have an xSYS header then you must provide your own method for writing to flash/OTP and for debugging.

M.2 JTAG-only xSYS header

The xSYS header connects to an xTAG debugger, which has a 20-pin 0.1" female IDC header. The design will hence need a male IDC header. We advise to use a boxed header to guard against incorrect plug-ins. If you use a 90 degree angled header, make sure that pins 2, 4, 6, ..., 20 are along the edge of the PCB.

Connect pins 4, 8, 12, 16, 20 of the xSYS header to ground, and then connect:

- TDI to pin 5 of the xSYS header
- TMS to pin 7 of the xSYS header
- TCK to pin 9 of the xSYS header
- DEBUG_N to pin 11 of the xSYS header
- TDO to pin 13 of the xSYS header
- RST_N to pin 15 of the xSYS header
- If MODE2 is configured high, connect MODE2 to pin 3 of the xSYS header. Do not connect to VDDIO.
- If MODE3 is configured high, connect MODE3 to pin 3 of the xSYS header. Do not connect to VDDIO.

The RST_N net should be open-drain, active-low, and have a pull-up to VDDIO.

M.3 Full xSYS header

For a full xSYS header you will need to connect the pins as discussed in Section M.2, and then connect a 2-wire xCONNECT Link to the xSYS header. The links can be found in the Signal description table (Section 4): they are labelled XLA, XLB, etc in the function column. The 2-wire link comprises two inputs and outputs, labelled $^1_{out}$, $^0_{out}$, $^0_{in}$, and $^1_{in}$. For example, if you choose to use XLB of tile 0 for xSCOPE I/O, you need to connect up XLB^1_{out}, XLB^0_{out}, XLB^0_{in}, XLB^1_{in} as follows:

- XLB^1_{out} (X0D16) to pin 6 of the xSYS header with a 33R series resistor close to the device.
- XLB^0_{out} (X0D17) to pin 10 of the xSYS header with a 33R series resistor close to the device.
- XLB^0_{in} (X0D18) to pin 14 of the xSYS header.
- XLB^1_{in} (X0D19) to pin 18 of the xSYS header.

N Schematics Design Check List

☑ This section is a checklist for use by schematics designers using the XS1-U6A-64-FB96. Each of the following sections contains items to check for each design.

N.1 Clock

☐ If you use USB, then your clock frequency is one of 12, 24, 48, or 96 MHz (Section 8).

☐ Pins MODE0 and MODE1 are set to the correct value for the chosen frequency. The MODE settings are shown in the Oscillator section, Section 8. If you have a choice between two values, choose the value with the highest multiplier ratio since that will boot faster.

☐ OSC_EXT_N is tied to ground (for use with a crystal or oscillator) or tied to VDDIO (for use with the internal oscillator). If using the internal oscillator, set MODE0 and MODE1 to be for the 20-48 MHz range (Section 8).

☐ If you have used an oscillator, it is a 1V8 oscillator. (Section 17)

N.2 Boot

☐ The device is connected to a SPI flash for booting, connected to X0D0, X0D01, X0D10, and X0D11 (Section 9). If not, you must boot the device through OTP or JTAG.

☐ The device that is connected to flash has both MODE2 and MODE3 connected to pin 3 on the xSYS Header (MSEL). If no debug adapter connection is supported (not recommended) MODE2 and MODE3 are to be left NC (Section 9).

☐ The SPI flash that you have chosen is supported by **xflash**, or you have created a specification file for it.

N.3 JTAG, XScope, and debugging

☐ You have decided as to whether you need an XSYS header or not (Section M)

☐ If you included an XSYS header, you connected pin 3 to any MODE2/MODE3 pin that would otherwise be NC (Section M).

☐ If you have not included an XSYS header, you have devised a method to program the SPI-flash or OTP (Section M).

N.4 GPIO

☐ You have not mapped both inputs and outputs to the same multi-bit port.

N.5 Multi device designs

Skip this section if your design only includes a single XMOS device.

☐ One device is connected to a SPI flash for booting.

☐ Devices that boot from link have MODE2 grounded and MODE3 NC. These device must have link XLB connected to a device to boot from (see 9).

☐ If you included an XSYS header, you have included buffers for RST_N, TMS, TCK, MODE2, and MODE3 (Section L).

O PCB Layout Design Check List

☑ This section is a checklist for use by PCB designers using the XS1-U6A-64-FB96. Each of the following sections contains items to check for each design.

O.1 Ground Balls and Ground Plane

☐ There is one via for each ground ball to minimize impedance and conduct heat away from the device (Section 16.3).

☐ There are only few non-ground vias around the square of ground balls, to creating a good, solid, ground plane.

O.2 Power supply decoupling

☐ VSUP has a ceramic X5R or X7R bulk decoupler as close as possible to the VSUP and PGND (VDDCORE) pins; right next to the device (Section 16).

☐ The 1V0 decoupling cap is close to the VDDCORE and PGND pins (Section 16).

☐ The 1V8 decoupling cap is close to the VDD1V8 and PGND pins (Section 16).

☐ All PGND nets are connected together prior to connection to the main ground plane (Section 16).

An example PCB layout is shown in Section 17. Placing the decouplers too far away may lead to the device not coming up, or not operating properly.

P Associated Design Documentation

Document Title	Information	Document Number
Programming XC on XMOS Devices	Timers, ports, clocks, cores and channels	X9577
xTIMEcomposer User Guide	Compilers, assembler and linker/mapper Timing analyzer, xScope, debugger Flash and OTP programming utilities	X3766

Q Related Documentation

Document Title	Information	Document Number
The XMOS XS1 Architecture	ISA manual	X7879
XS1 Port I/O Timing	Port timings	X5821
xCONNECT Architecture	Link, switch and system information	X4249
XS1-L Link Performance and Design Guidelines	Link timings	X2999
XS1-L Clock Frequency Control	Advanced clock control	X1433

R Revision History

Date	**Description**
2013-01-30	New datasheet - revised part numbering
2013-02-26	New multicore microcontroller introduction Moved configuration sections to appendices
2013-03-27	Added connection details for USB_VBUS/USB_ID - Section 11 VDDCORE parameters - Section 18.2
2013-04-16	OSC_REF_EXT_N Properties - Section 4 Sleep mode requirements include JTAG - Section 14.4
2013-07-19	Updated Features list with available ports and links - Section 2 Simplified link bits in Signal Description - Section 4 New JTAG, xSCOPE and Debugging appendix - Section M New Schematics Design Check List - Section N New PCB Layout Design Check List - Section O Updated USB_VBUS pin connection - Section 11
2013-12-09	Added Industrial Ambient Temperature - Section 18.1 Annotated V(ACC) parameter - Section 18.2 Updated V(IH) parameter - Section 18.10 Updated V(OH) parameter - Section 18.6
2014-03-25	Added footnotes to DC and Switching Characteristics - Section 18
2014-06-25	New PCB guidelines for high-speed USB designs - Section 16.2
2014-08-29	Moved USB pin data to Section 16.1; added additional PHY information Added USB characterisation data - Section 18.5
2015-04-14	Updated Introduction - Section 1; Pin Configuration - Section 3; Signal Description - Section 4