# MAX 10 FPGA (10M08S, 144-EQFP) Evaluation Kit

## User Guide

101 Innovation Drive
San Jose, CA 95134
www.altera.com

UG-01158-1.2

Feedback Subscribe

# Contents

# 1. Overview

The MAX ®10 Evaluation Kit allows is an entry-level board for evaluating the MAX 10 FPGA technology and Enpirion® PowerSoC regulators. You can use this kit to do the following:

- Develop designs for the 10M08S, 144-EQFP FPGA
- Measure FPGA power (VCC_CORE and VCC_IO)
- Bridge between different I/O voltages
- Read and write to the FPGA's NOR flash memory
- Use the FPGA's analog-to-digital converter embedded block to measure incoming analog signals
- Interface to external functions or devices via Arduino UNO R3 connectors or through-hole vias
- Reuse the kit's PCB board and schematic as a model for your design

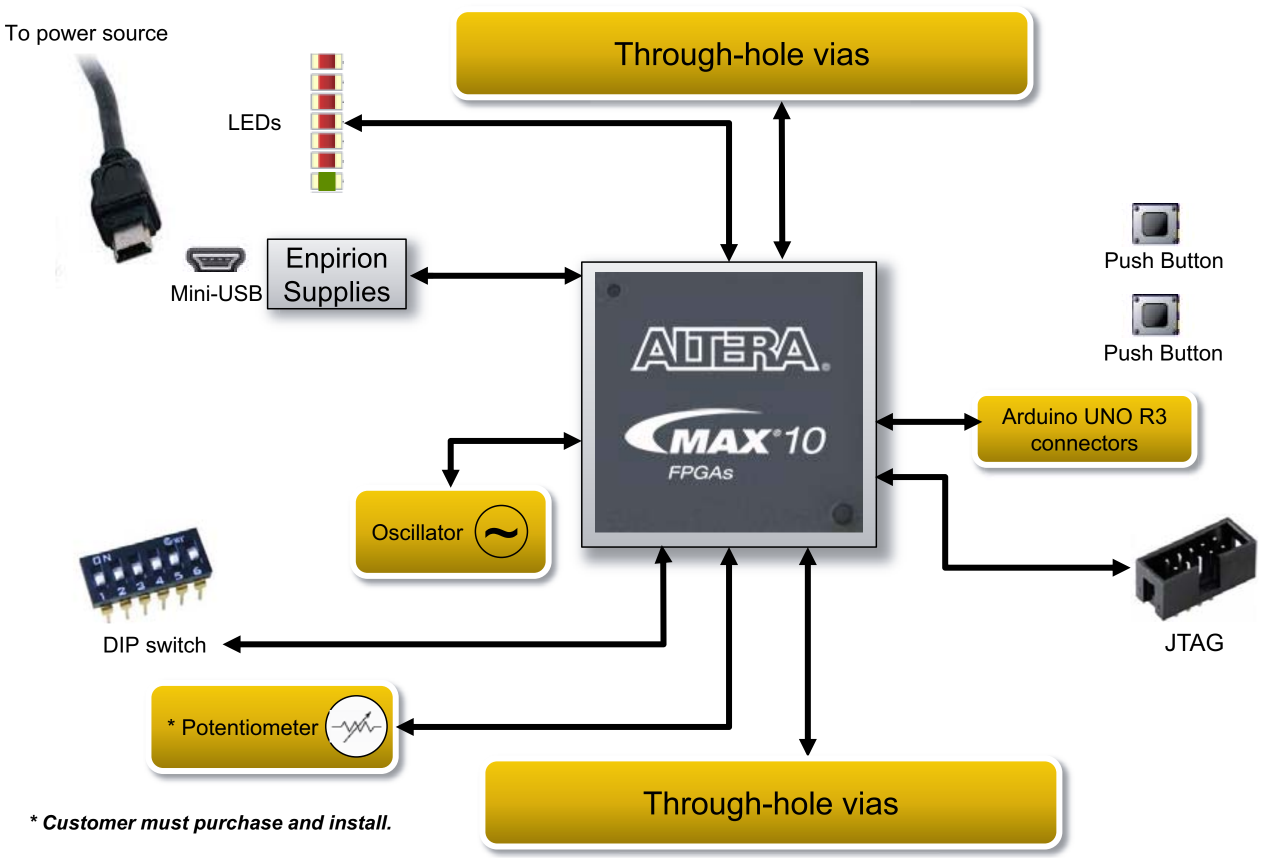
## Board Component Blocks

This evaluation kit features the following major component blocks. For a detailed description of the board components, see "Board Components" on page 3–1.

- Altera MAX 10 FPGA, 10M08SAE144C8G, (or ES variant)
  - 8,000 logic elements (LE)
  - 378 kilobits (Kb) M9K memory
  - 32 – 172 (KB) user flash memory
  - One analog-to-digital (ADC) converter, 1 million samples per second (MSPS), 12-bit
- FPGA configuration circuitry
  - JTAG header for external USB-Blaster™, USB-Blaster II, or Ethernet Blaster download cable
  - Flash storage for two configuration images (factory and user)
  - Dual-image self-configuration via Programmer Object File (**.pof**)
  - Temporary engineering debug of FPGA design via SRAM Object File (**.sof)**
- On-Board clocking circuitry
  - 50 MHz oscillator connected to FPGA global clock input
- General user I/O
  - 8 analog input I/O, 14 Arduino I/O, 40 general purpose I/O
  - 5 red user-defined LEDs
  - One green LED to show power from USB cable

- Push button and DIP switches
  - One reconfiguration push button (SW2)
  - One device-wide reset of all registers, push button (SW1)
  - User DIP switch (SW3)
- Power
  - The board is powered by USB cable (from PC or wall jack)
  - One green power-on LED (D6)
  - Probe points for manual, multi-meter measurement of current to calculate power consumption (TP2 - TP5) or to verify voltages on the selected internal nodes (TP1, TP6 - TP9)

**Figure 1–1. Example MAX 10 Evaluation Kit Block Diagram**



# Supported Items Not Included with the Kit

The following items are not included in the kit but were designed to be used in conjunction with this kit.

**Table 1–1. Additional Components Not Included with the Kit**

| Board Reference | Description | Manufacturer | Manufacturing Part Number | Manufacturer Website |
|---|---|---|---|---|
| R94 | Potentiometer | Bourns | 3362P-1-103TLF | www.bourns.com |
| J8, J9 | 2x20 0.1-inch headers | Sullins Connector Solution | PPPC202LFBN-RC | www.sullinscorp.com |
| J2, J3, J4, J5 | Optional daughter-cards: Arduino UNO R3 revision shields. | Arduino | — | www.adafruit.com<br>www.sainsmart.com<br>www.arduino.com |
| J1 | USB-Blaster Download Cable | Altera | PL-USB-BLASTER-RCN | www.altera.com |
| J1 | USB-Blaster II Download Cable | Altera | PL-USB2-BLASTER | Please call |

## Powering the Kit

You can apply power the MAX 10 FPGA Evaluation Kit by plugging in the USB cable (J1) to your PC or wall jack. When powered correctly, a pre-programmed design blinks LEDs D1 through D5 ON half a second then OFF half a second.

## Installing the USB-Blaster Driver

You can configure the evaluation kit by programming on-chip flash memory using a USB-Blaster™, USB-Blaster II, or Ethernet Blaster download cable. However, for the host computer and board to communicate, you must install the appropriate USB-Blaster, USB-Blaster II, or Ethernet Blaster driver on the host computer.

Installation instructions for the Blaster driver for your operating system are available on the Altera website. On the Altera Programming Cable Driver Information page of the Altera website, locate the table entry for your configuration and click the link to access the instructions.

You can download the Blaster drivers from the Download Cables page.

## Handling the Kit

When handling the board, it is important to observe the following static discharge precaution:

CAUTION

Without proper anti-static handling, the board can be damaged. Therefore, use anti-static handling precautions when touching the board.

The MAX 10 Evaluation Kit must be stored between –40º C and 100º C. The recommended operating temperature is between 0º C and 85º C.

# Factory Default Switch and Jumper Settings

**Figure 2–1. Switch Locations and Default Settings (Board Top)**

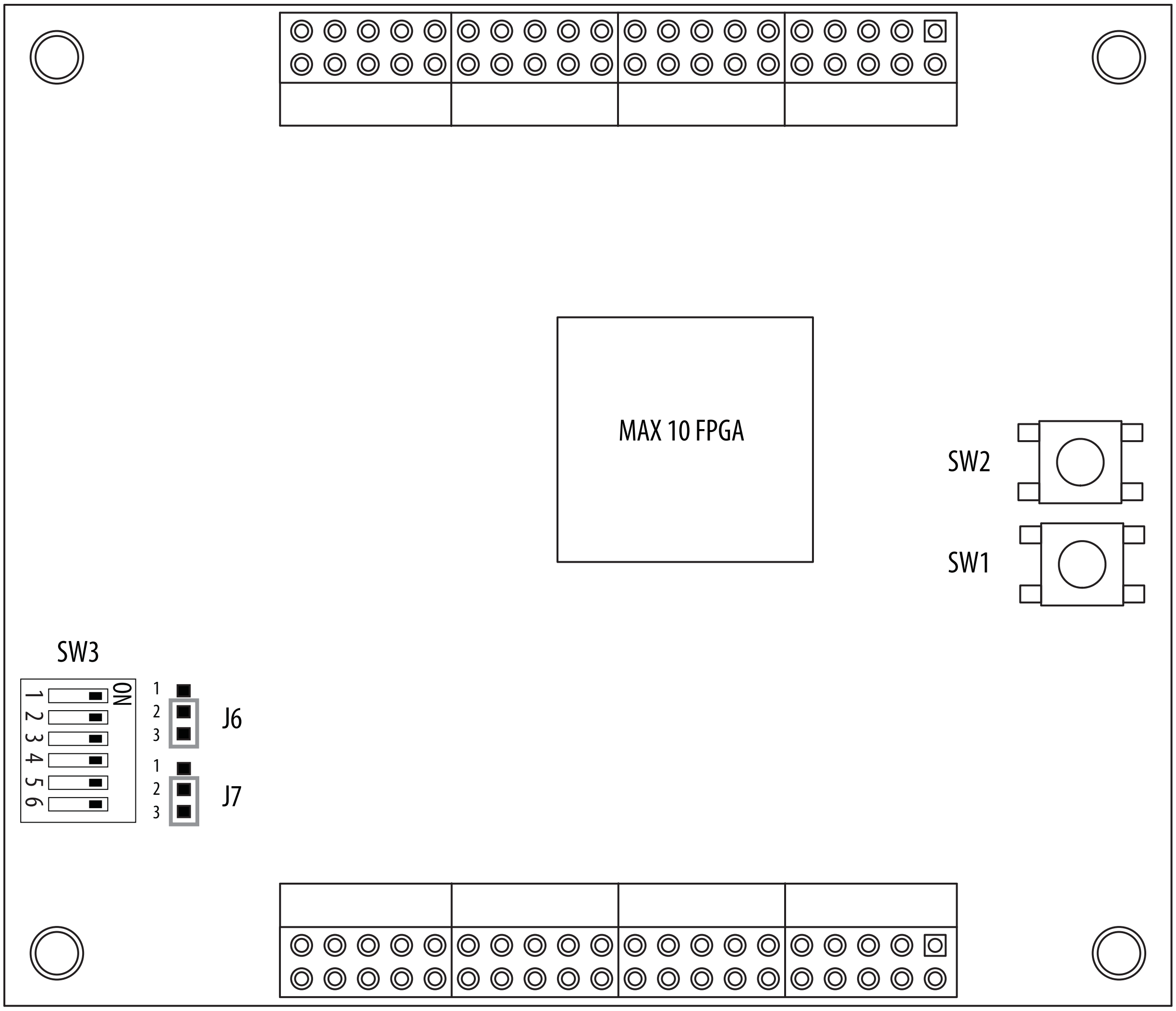



**Table 2–1. Default SW3 DIP Switch Settings (Part 1 of 2)**

| Switch | Function | Default Position |
|---|---|---|
| 1 | User-defined | On |
| 2 | User-defined | On |
| 3 | User-defined | On |
| 4 | User-defined | On |

**Table 2–1. Default SW3 DIP Switch Settings (Part 2 of 2)**

| Switch | Function | Default Position |
|---|---|---|
| 5 | User-defined | On |
| 6 | BOOT_SEL: Use this switch to choose CFM0, CFM1, or CFM2 image as the first image in a dual-image configuration.<br>If BOOT_SEL is set to low, the first boot image is CFM0 image, If set to high, the first boot image is the CFM1 or CFM2 image. By default, the FPGA setting for this pin is tri-stated. | On |

**Table 2–2. Default Jumper Settings**

| Jumper | Function | Setting |
|---|---|---|
| J6 | Jumper for analog input channel #8. Default connection is to GND. | Pins 2 and 3 |
| J7 | Jumper for analog input channel #7. Default connection is to potentiometer. | Pins 2 and 3 |

# 3. Board Components

This chapter introduces all the important components on the evaluation kit. Figure 3–1 illustrates major component locations and Table 3–1 provides a brief description of all features of the board.

## Board Overview

This section provides an overview of the evaluation kit, including an annotated board image and component descriptions.

**Figure 3–1. Overview of the MAX 10 FPGA Evaluation Kit Features**

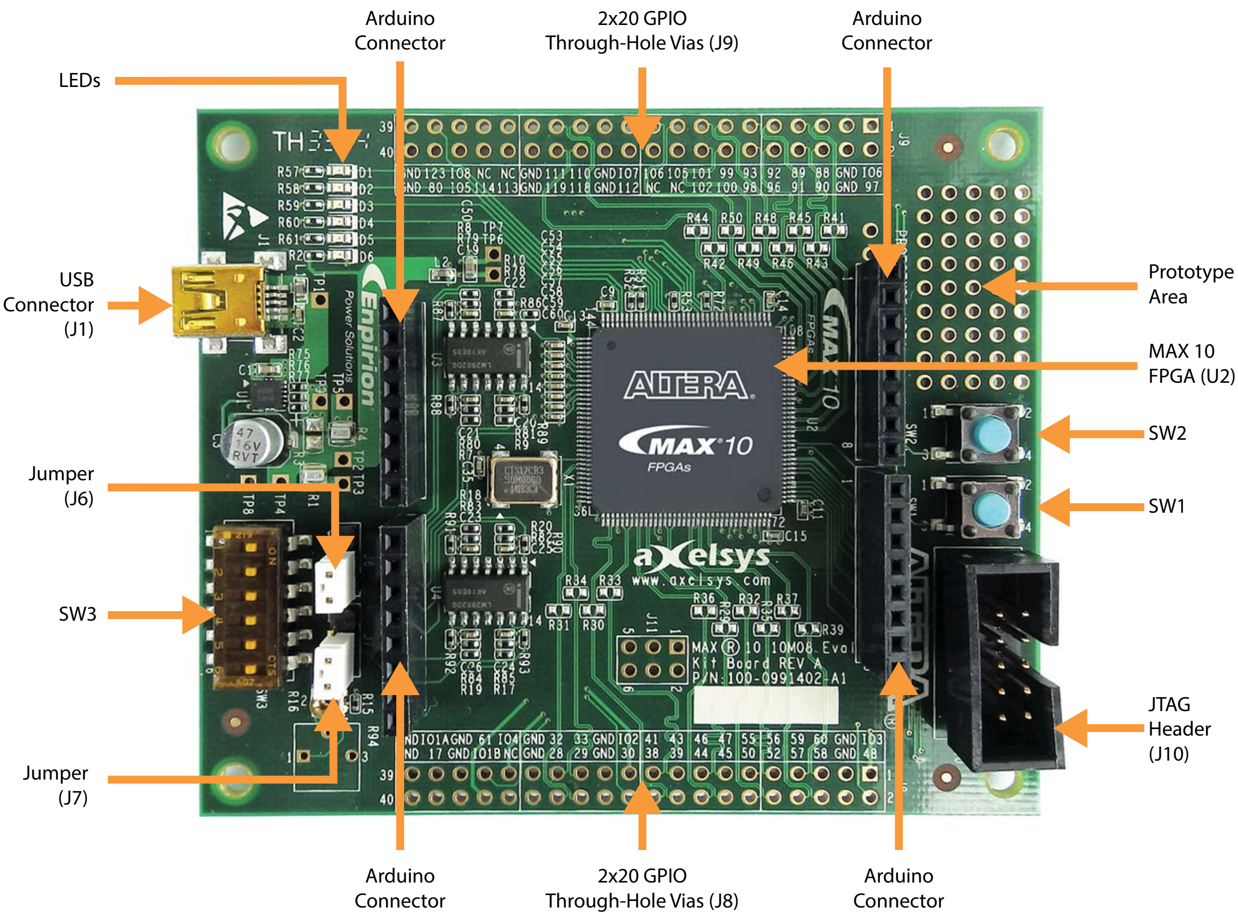



Table 3–1 describes the components and lists their corresponding board references.

**Table 3–1. MAX 10 FPGA (10M08S, 144-EQFP) Evaluation Kit Components**

| Board Reference | Type | Description |
|---|---|---|
| **Featured Device** | | |
| U2 | FPGA | 10M08SAE144C8G, (or ES variant) Plastic Enhanced Quad Flat Pack (EQFP), 144 pins, 22 mm x 22 mm.<br>For package details, refer to the Altera Device Package Information page. |
| **Configuration, Status, and Setup Elements** | | |
| J6 | Jumper for analog input channel #8 | Default connection is to GND. Change jumper to pins 1 and 2 to switch analog source to Arduino header. |
| J7 | Jumper for analog input channel #7 | Default connection is to potentiometer (customer option to purchase and install). Change jumper to pins 1 and 2 to switch analog source to Arduino header. |
| SW3 | User-defined DIP switch | 6-position switch. SW3.1 through SW3.5 are user-defined. SW3.6 is predefined for dual-image configuration. |
| D1, D2, D3, D4, D5 | LED, red | These LEDs cycle off and on when the kit is powered on. |
| D6 | Power LED, green | Illuminates when USB power is present. |
| SW2 | FPGA reconfiguration push-button | Toggling this button causes the FPGA to reconfigure from on-die Configuration Flash Memory (CFM). |
| **Clock Circuitry** | | |
| X1 | 50-MHz oscillator | 50-MHz crystal oscillator for general purpose logic. |
| **General User Input and Output** | | |
| D1, D2, D3, D4, D5 | User-defined LEDs, red | User-defined LEDs. |
| SW1 | FPGA register push-button | Toggling this button resets all registers in the FPGA. |
| R94 | Potentiometer | You must purchase and install this device to provide analog inputs signals to the MAX 10 ADC IP block (analog input channel 8). |
| **Connectors** | | |
| J2, J3, J4, J5 | Arduino UNO R3 connectors | You can purchase Arduino Uno R3 compatible Shields (i.e. daughtercards) to connect to the Arduino headers installed on the board. |
| J10 | JTAG header | Connects an Altera USB-Blaster, USB-Blaster II, or Ethernet Blaster to program or configure the FPGA. |
| — | Prototype Area | This through-hole area is not connected to the FPGA. You can use this area to connect or solder additional components. |
| **Power Supply** | | |
| J1 | USB connector | Connects a USB cable to a power source. |

# Featured Device: MAX 10 FPGA

The evaluation kit features the MAX 10 FPGA 10M08SAE144C8G device (U2) in a 144-pin Plastic Enhanced Quad Flat Pack (EQFP) package.

For more detailed information about the MAX 10 FPGA device family, refer to the *MAX 10 FPGA Device Overview.*

**Table 3–2. MAX 10 FPGA Component Reference and Manufacturing Information**

| Board Reference | Description | Manufacturer | Manufacturing Part Number | Manufacturer Website |
|---|---|---|---|---|
| U2 | 10M08SAE144C8G, (or ES variant) Plastic Enhanced Quad Flat Pack (EQFP), 144 pins, 20 mm x 20mm | Altera Corporation | 10M08SAE144C8GES | www.altera.com/max10 |

# Configuration

The evaluation kit supports two configuration methods:

- JTAG header (J10) for configuration by downloading a **.sof** file to the FPGA. Any power cycling of the FPGA or reconfiguration will power up the FPGA to a blank state.
- JTAG header (J10) for programming of the on-die FPGA Configuration Flash Memory (CFM) via a **.pof** file. Any power cycling of the FPGA or reconfiguration will power up the FPGA in self-configuration mode, using the files stored in the CFM.

## FPGA Programming over External USB-Blaster

The JTAG header provides a method for configuring the FPGA (U2) using an external USB-Blaster™, USB-Blaster II, or Ethernet Blaster download cable with the Quartus II Programmer running on a PC. The external download cable connects to the board through the JTAG header (J10).

## Configuring the FPGA Using the Quartus II Programmer

You can use the Quartus II Programmer to configure the FPGA with a **.sof**. Before configuring the FPGA, ensure that the Quartus II Programmer and the USB-Blaster driver are installed on the host computer, the USB cable is connected to the evaluation kit, power to the board is on, and no other applications that use the JTAG chain are running.

To successfully use the USB-Blaster cable, disconnect it before power cycling the board. After you power cycled the board, then reconnect the USB-Blaster cable.

To configure the MAX 10 FPGA FPGA, perform the following steps:

1. Start the Quartus II Programmer.
2. Click **Auto Detect** to display the devices in the JTAG chain.

3. Click **Add File** and select the path to the desired **.sof**.
4. Turn on the **Program/Configure** option for the added file.
5. Click **Start** to download the selected file to the FPGA. Configuration is complete when the progress bar reaches 100%.

The Quartus II Convert Programming File (CPF) GUI can be used to generate a **.pof** file that can use for internal configuration. You can directly program the MAX 10 device's flash which included Configuration Flash Memory (CFM) and User Flash Memory (UFM) by using a download cable with the Quartus II software programmer.

## Selecting Internal Configuration Scheme

For all MAX 10 devices except 10M02 device, there are total of 5 different modes can be selected when using Internal Configuration. The internal configuration scheme needs to be selected before design compilation.

To select the configuration mode, follow these steps:

1. Open the Quartus II software and load a project using MAX 10 device family.
2. On the Assignments menu, click **Settings**. The **Settings** dialog box appears.
3. In the Category list, select **Device**. The Device page appears.
4. Click **Device and Pin Options**.
5. In the **Device and Pin Options** dialog box, click the **Configuration** tab.
6. In the **Configuration Scheme** list, select **Internal Configuration**.
7. In the **Configuration Mode** list, select 1 out of 5 configuration modes except 10M02 device, which has only 2 modes available.
8. Turn on **Generate compressed bitstreams** if needed.
9. Click **OK**.

## Generating a .pof File with ICB Settings

To generate a **.pof** file from a **.sof** file for internal configuration, follow these steps:

1. On the File menu, click **Convert Programming Files**.
2. Under **Output programming file**, select Programmer Object File **(.pof)** in the Programming file type list.
3. In the **Mode** list, select **Internal Configuration**.

4. To set the ICB settings, click **Option/Boot Info** button. An ICB setting dialog box will appear.

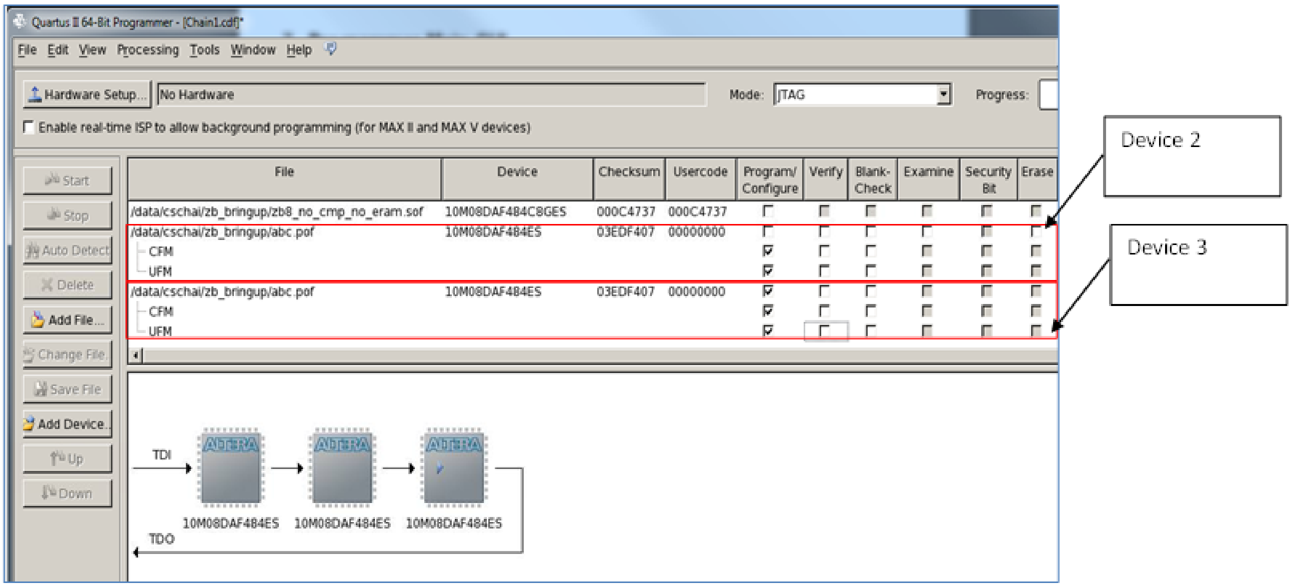
   Several ICB settings can be set through the ICB setting dialog box, including:

   - Power on Reset Scheme: Instant On, Fast POR Delay or Slow POR Delay.
   - Enable user IOs week pull up during configuration check box.
   - Enable the JTAG Security check box.
   - Verify Protect check box.
   - Enable watchdog for dual boot and watching value (initially grayed out, after add 2 sof page with 2 design that compiled with Dual Compressed Internal Images, the watchdog setting will then be enable).
   - User Flash Memory settings.
5. In the **File name** box, specify the file name for the programming file you want to create.
6. To generate a Memory Map File **(.map)**, turn on **Create Memory Map File (Auto generate output_file.map**). In the **.map** file, not only will show the address of the CFM and UFM, but also will contain the information of the ICB setting that user set through the **Option/Boot Info** dialog box.
7. You can add an SRAM Object File **(.sof)** through **Input files to convert** list. The maximum sof page is two.
8. After set all the desirable settings, click Generate to generate related programming file.

## Programming Internal Flash Memory

After generating the **.pof** file, Quartus II Programmer can be used to program the internal flash memory through JTAG connection. The following shows an example of the Quartus II Programmer.

**Figure 3–2. Quartus II Programmer**




There are 2 scenarios when using the Quartus II Programmer to program a **.pof** file into MAX 10 devices:

1. For Device 2, the Programmer will just erase and configure CFM and UFM sector in the internal flash memory, but the ICB setting will be preserved. However, before starting the programming, the makes sure the ICB setting in the device and the ICB setting in the **.pof** file are the same. If both ICB settings are different, the Programmer still erases and programs the full internal memory including the ICB setting, even though only CFM and UFM are selected in the Programmer.
2. For Device 3, the Programmer will erase and program full internal memory, which includes the ICB setting, CFM, and UFM.

Before enabling real-time ISP for internal flash memory programming, you need to ensure the MAX 10 FPGA is in user mode, otherwise the programming process will fail.

# Clock Circuitry

## General-Purpose Clock

One general-purpose clock is provided to the FPGA global clock inputs for general FPGA design. The clock source is from the following component:

- A 50-MHz oscillator to the clock input `CLK0p` of bank 2.

Figure 3–3 shows the general purpose clock going in to the evaluation kit.

**Figure 3–3. MAX 10 FPGA (10M08S, 144-EQFP) Evaluation Kit General Purpose Clock**


# Arduino Connectors

Arduino connectors J3, J4, and J5 connect to the MAX 10 FPGA. Any analog inputs signals sourced through the Arduino header J4 are first filtered by the evaluation boards op-amp based circuit. This circuit scales the maximum allowable voltage per the Arduino specification (5.0V) to the maximum allowable voltage per the MAX 10 FPGA ADC IP block (2.5V).

You can download an example design with pin locations and assignments completed according to the following table from the Altera Design Store. In the MAX 10 FPGA Evaluation Kit, under **Design Examples**, click **MAX 10 Evaluation Kit Baseline Pinout**.

**Table 3–3. Arduino Connector Pin Assignments, Signal Names and Functions (Part 1 of 2)**

| Board Reference | Schematic Signal Name | MAX 10 FPGA Device Pin Number | Description |
|---|---|---|---|
| J3.1 | `ANALOG_VREF` | 5 | Arduino analog Vref input |
| J3.2 | `GND` | — | Arduino GND input |
| J3.3 | `ARDUINO_IO13` | 70 | Arduino digital I/O input to FPGA |
| J3.4 | `ARDUINO_IO12` | 69 | Arduino digital I/O input to FPGA |
| J3.5 | `ARDUINO_IO11` | 66 | Arduino digital I/O input to FPGA |
| J3.6 | `ARDUINO_IO10` | 65 | Arduino digital I/O input to FPGA |
| J3.7 | `ARDUINO_IO9` | 64 | Arduino digital I/O input to FPGA |
| J3.8 | `ARDUINO_IO8` | 62 | Arduino digital I/O input to FPGA |
| J4.1 | `ARDUINO_A0` | 6 | Arduino analog channel input through the op-amp filter circuit to the FPGA ADC IP input channel ADCIN1 |
| J4.2 | `ARDUINO_A1` | 7 | Arduino analog channel input through the op-amp filter circuit to the FPGA ADC IP input channel ADC1IN2 |
| J4.3 | `ARDUINO_A2` | 8 | Arduino analog channel input through the op-amp filter circuit to the FPGA ADC IP input channel ADC1IN3 |
| J4.4 | `ARDUINO_A3` | 10 | Arduino analog channel input through the op-amp filter circuit to the FPGA ADC IP input channel ADC1IN4 |

**Table 3–3. Arduino Connector Pin Assignments, Signal Names and Functions (Part 2 of 2)**

| Board Reference | Schematic Signal Name | MAX 10 FPGA Device Pin Number | Description |
|---|---|---|---|
| J4.5 | ARDUINO_A4 | 11 | Arduino analog channel input through the op-amp filter circuit to the FPGA ADC IP input channel ADC1IN5 |
| J4.6 | ARDUINO_A5 | 12 | Arduino analog channel input through the op-amp filter circuit to the FPGA ADC IP input channel ADC1IN6 |
| J4.7 | ARDUINO_A6 | 13 | Arduino analog channel input through the op-amp filter circuit to the FPGA ADC IP input channel ADC1IN7 |
| J4.8 | ARDUINO_A7 | 14 | Arduino analog channel input through the op-amp filter circuit to the FPGA ADC IP input channel ADC1IN8 |
| J5.1 | ARDUINO_IO7 | 86 | Arduino digital I/O input to FPGA |
| J5.2 | ARDUINO_IO6 | 84 | Arduino digital I/O input to FPGA |
| J5.3 | ARDUINO_IO5 | 81 | Arduino digital I/O input to FPGA |
| J5.4 | ARDUINO_IO4 | 79 | Arduino digital I/O input to FPGA |
| J5.5 | ARDUINO_IO3 | 77 | Arduino digital I/O input to FPGA |
| J5.6 | ARDUINO_IO2 | 76 | Arduino digital I/O input to FPGA |
| J5.7 | ARDUINO_IO1 | 75 | Arduino digital I/O input to FPGA |
| J5.8 | ARDUINO_IO0 | 74 | Arduino digital I/O input to FPGA |

# General User Input/Output

This section describes the user I/O interface to the FPGA:

- User-defined DIP switch
- User-defined LEDs

## User-Defined DIP Switch

Board reference SW3 is a 6-pin DIP switch. Switches 1 through 5 are user-defined, and provide additional FPGA input control. When the switch is in the OPEN or OFF position, a logic 1 is selected. When the switch is in the CLOSED or ON position, a logic 0 is selected. There is no board-specific function for these switches.

Table 3–4 lists the user-defined DIP switch schematic signal names and their corresponding MAX 10 FPGA pin numbers.

**Table 3–4. User-Defined DIP Switch Schematic Signal Names and Functions**

| Board Reference SW3 | Schematic Signal Name | I/O Standard | MAX 10 FPGA Device Pin Number |
|---|---|---|---|
| 1 | `Switch 1` | 3.3-V | 120 |
| 2 | `Switch 2` | 3.3-V | 124 |
| 3 | `Switch 3` | 3.3-V | 127 |
| 4 | `Switch 4` | 3.3-V | 130 |
| 5 | `Switch 5` | 3.3-V | 131 |

## User-Defined LEDs

The development board includes five user-defined LEDs. Board references D1 through D5 are user LEDs that allow status and debugging signals to be driven to the LEDs from the designs loaded into the MAX 10 FPGA device. The LEDs illuminate when a logic 0 is driven, and turns off when a logic 1 is driven. There is no board-specific function for these LEDs.

Table 3–5 lists the user-defined LED schematic signal names and their corresponding MAX 10 FPGA pin numbers.

**Table 3–5. User-Defined LED Schematic Signal Names and Functions**

| Board Reference | Schematic Signal Name | I/O Standard | MAX 10 FPGA Device Pin Number |
|---|---|---|---|
| D1 | `LED1` | 2.0-V | 132 |
| D2 | `LED2` | 2.0-V | 134 |
| D3 | `LED3` | 2.0-V | 135 |
| D4 | `LED4` | 2.0-V | 140 |
| D5 | `LED5` | 2.0-V | 141 |

# Power Supply

The development board is powered up through a USB cable. The green LED illuminates when the board is powered up.

## Power Measurement

In order to measure the actual power of the FPGA, there are test pads on the board to be used as probe points for multi-meter probes. The user can measure the current and using the equation $P = R \times I^2$. (Power = Resistance x Current Squared), calculate the power dissipation.

Test pads TP2 and TP3 are used to measure the current consumed by the FPGA core. Test pads TP4 and TP5 are used to measure the current consumed by all of the FPGA's I/O banks. All the other test pads are used to verify the voltage levels of various nodes on the board.

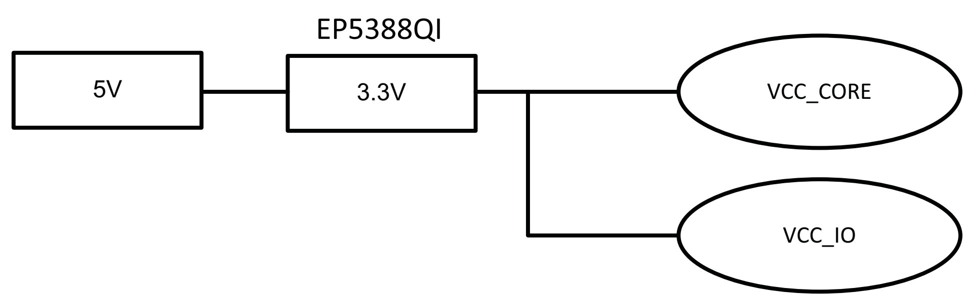
**Table 3–6. Power Measurement Details**

| Test Pad #'s | Measuring | Description | Expected Value |
|---|---|---|---|
| TP2 - TP3 | FPGA core current | Power calculation for FPGA Vcc-core power consumption. Resistor R1 = 0.1 ohms. Current measured by user's multi-meter. | ___ mWatts |
| TP4 - TP5 | FPGA I/O current | Power calculation for FPGA Vcc-io power consumption. Resistor R4 = 0.1 ohms. Current measured by user's multi-meter. | ___ mWatts |
| **Test Pad #'s** | **Measuring** | **Description** | **Expected Value** |
| TP1 | Board input voltage | Verify the USB input voltage | 5.0-volts |
| TP6 | Analog voltage | Verify the proper voltage required by the FPGA Vcca inputs | 3.3-volts |
| TP7 | Analog GND | Verify the proper voltage required by the FPGA ADC IP block GND inputs | 0-volts |
| TP8 | Digital GND | Verify the proper voltage required by the FPGA digital GND inputs | 0-volts |
| TP9 | Digital GND | Verify the proper voltage required by the FPGA digital GND inputs | 0-volts |

## Power Distribution System

Figure 3–4 shows the power distribution system on the development board.

**Figure 3–4. Power Distribution System**




## Temperature Sense

The ADCs provide the devices with built-in capability for on-die temperature monitoring and external analog signal conversion.

Temperature sensing mode—monitors external temperature data input with a sampling rate of up to 50 kilosamples per second. In dual ADC devices, only the first ADC block contains the temperature sensing diode.

For more information on the ADC, refer to the *MAX 10 FPGA Device Overview*.

This chapter provides additional information about the document and Altera.

## Document Revision History

The following table shows the revision history for this document.

| Date | Version | Changes |
|---|---|---|
| September 2015 | 1.2 | ■ Corrected sense of switches in "User-Defined DIP Switch" on page 3–8.<br>■ Added link to Altera Design Store in "Arduino Connectors" on page 3–7 |
| October 2014 | 1.1 | Corrected FPGA pin number for SW3.2 from 121 to 124. |
| September 2014 | 1.0 | Initial release. |

## How to Contact Altera

To locate the most up-to-date information about Altera products, refer to the following table.

| Contact (1) | Contact Method | Address |
|---|---|---|
| Technical support | Website | www.altera.com/support |
| Technical training | Website | www.altera.com/training |
| | Email | custrain@altera.com |
| Product literature | Website | www.altera.com/literature |
| Nontechnical support (general) | Email | nacomp@altera.com |
| (software licensing) | Email | authorization@altera.com |

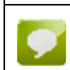
**Note to Table:**

(1) You can also contact your local Altera sales office or sales representative.

## Typographic Conventions

The following table shows the typographic conventions this document uses.

| Visual Cue | Meaning |
|---|---|
| **Bold Type with Initial Capital Letters** | Indicate command names, dialog box titles, dialog box options, and other GUI labels. For example, **Save As** dialog box. For GUI elements, capitalization matches the GUI. |
| **bold type** | Indicates directory names, project names, disk drive names, file names, file name extensions, software utility names, and GUI labels. For example, **\qdesigns** directory, **D:** drive, and **chiptrip.gdf** file. |
| *Italic Type with Initial Capital Letters* | Indicate document titles. For example, *Stratix IV Design Guidelines*. |

| Visual Cue | Meaning |
|---|---|
| *italic type* | Indicates variables. For example, *n* + 1.<br>Variable names are enclosed in angle brackets (< >). For example, <*file name*> and <*project name*>**.pof** file. |
| Initial Capital Letters | Indicate keyboard keys and menu names. For example, the Delete key and the Options menu. |
| "Subheading Title" | Quotation marks indicate references to sections in a document and titles of Quartus II Help topics. For example, "Typographic Conventions." |
| `Courier type` | Indicates signal, port, register, bit, block, and primitive names. For example, `data1`, `tdi`, and `input`. The suffix `n` denotes an active-low signal. For example, `resetn`.<br>Indicates command line commands and anything that must be typed exactly as it appears. For example, `c:\qdesigns\tutorial\chiptrip.gdf`.<br>Also indicates sections of an actual file, such as a Report File, references to parts of files (for example, the AHDL keyword `SUBDESIGN`), and logic function names (for example, `TRI`). |
| ↵ | An angled arrow instructs you to press the Enter key. |
| 1., 2., 3., and<br>a., b., c., and so on | Numbered steps indicate a list of items when the sequence of the items is important, such as the steps listed in a procedure. |
| ■ ■ ■ | Bullets indicate a list of items when the sequence of the items is not important. |
|  | The hand points to information that requires special attention. |
|  | The question mark directs you to a software help system with related information. |
|  | The feet direct you to another document or website with related information. |
|  | The multimedia icon directs you to a related multimedia presentation. |
|   | A caution calls attention to a condition or possible situation that can damage or destroy the product or your work. |
|   | A warning calls attention to a condition or possible situation that can cause you injury. |
|  | The envelope links to the Email Subscription Management Center page of the Altera website, where you can sign up to receive update notifications for Altera documents. |
|  | The feedback icon allows you to submit feedback to Altera about the document. Methods for collecting feedback vary as appropriate for each document. |

# Compliance and Conformity Statements

## CE EMI Conformity Caution

This evaluation kit is delivered conforming to relevant standards mandated by Directive 2004/108/EC. Because of the nature of programmable logic devices, it is possible for the user to modify the kit in such a way as to generate electromagnetic interference (EMI) that exceeds the limits established for this equipment. Any EMI caused as the result of modifications to the delivered material is the responsibility of the user.