

EARTH PEOPLE TECHNOLOGY, Inc

MAXPROLOGIC DEVELOPMENT SYSTEM User Manual

The MaxProLogic is an FPGA development board that is designed to be user friendly and a great introduction into digital design for Electrical Engineering students and hobbyists. This boards provides innovative method of developing and debugging programmable logic code. It has been designed from the ground up to provide the functionality needed for the demanding projects from todays students and hobbyists. The board provides a convenient, user-friendly work flow by connecting seamlessly with Altera's Quartus II software. The user will develop the code in the Quartus environment on a Windows Personal Computer. The programmable logic code is loaded into the FPGA using only the Quartus Programmer tool.

The core of the MaxProLogic is the Intel/Altera MAX10 FPGA. This powerful chip has 4,000 Logic Elements and 200 Kbits of Memory. The MAX10 is easily scalable from the entry level college student to the most advanced projects like an audio sound meter with FFT. The MAX10 is Intel/Altera's newest entrance into low cost FPGA's.

Circuit designs, software and documentation are copyright © 2018, Earth People Technology, Inc

Microsoft and Windows are both registered trademarks of Microsoft Corporation. Altera is a trademark of the Altera Corporation. All other trademarks referenced herein are the property of their respective owners and no trademark rights to the same are claimed. http://www.earthpeopletechnology.com/



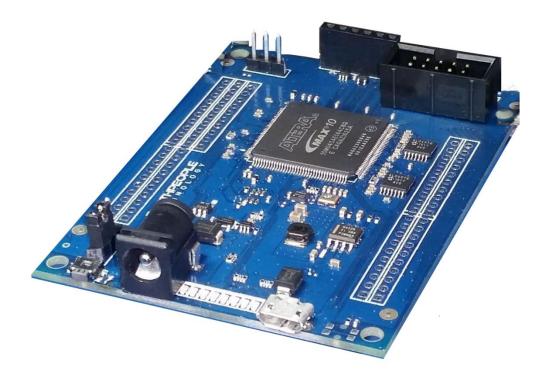
Contents

1 2			ction	
3	Ma	xPro	DLogic Description	5
	3.1	PC	B Footprint	5
	3.2	Fur	nctional Block Diagram	5
	3.3	MA	XPROLOGIC SPECIFICATIONS	5
	3.4	FPO	GA	7
	3.5	Pov	ver Supply	7
	3.6	Clo	ck Domains	9
	3.7	Dig	zital I/Os 1	0
	3.8	Ana	alog Input 1	1
	3.8	.1	Voltage Representation Conversion 1	5
	3.8	.2	Calculation Example for V_{REF} of 2.5 V	5
	3.8	.3	ADC Analog Input Pins 1	5
	3.8	.4	ADC Prescaler 1	6
	3.8	.5	ADC Clock Sources 1	6
	3.8	.6	ADC Voltage Reference 1	7
	3.8	.7	Creating Intel MAX 10 ADC Design 1	7
	3.8	.8	Parameters Settings for Generating ALTPLL IP Core 1	7
	3.8	.9	Completing ADC Design 1	7
	3.9	Ter	nperature Sensor	0
	3.10	L	.EDs	3
	3.11	S	D Card Interface	:5
	3.1	1.1	SD Card Protocol	6
	3.1	1.2	References	6
	3.1	1.3	Compatibility	6
	3.1	1.4	Protocol 2	.7
	3.1	1.5	Physical layer 2	:7
	3.1	1.6	Framing	27



	3.11.7	Commands and Responses	28
	3.11.8	Data	30
	3.11.9	Handover	30
	3.11.10) Initialisation	30
	3.11.11	Reading	32
	3.12 I	Power Input	33
	3.13	On/Off Control	33
	3.14 0	Communications Interface	36
	3.15 J	JTAG Interface	40
4 5		ng the MaxProLogicng Quartus	
	5.1 Do	wnloading Quartus	44
	5.1.1	Quartus Installer	47
6	Compi	ling, Synthesizing, and Programming the FPGA	52
	6.1 Set	tting up the Project and Compiling	52
	6.1.1	Selecting Pins and Synthesizing	58
7	Program	mming the MaxProLogic	65





1 Introduction

The MaxProLogic is Open Source Hardware based on the MAX10 FPGA. Core of the board is the MAX10 chip. It has a built in Flash for configuration and incorporates 8 channels of Analog to Digital Conversion. The board has two power options, standard USB Micro B connector and 5.5mm Barrel connector. The MaxProLogic can be powered from a laptop with 2.5W of power. Or it can be run it from the +5V @ 2A wall USB chargers for 10W of power. The barrel connector can handle up to +9V @ 3 A for 27W of power. The MaxProLogic has a MicroSD connector on the bottom of the board. The board has an optional On/Off pushbutton switch that allows the user to turn the system on and off. There are two clocking options, 50MHz oscillator and 32.768KHz oscillator.

2 User Setup

The MaxProLogic is ready to go straight out of the box. Just connect power from either a PC/Laptop, +5 VDC Universal Wall Charger, +5 to 9 VDC through barrel connector, or +5 to 9VDC through pin x on Jx. Install the On/Off jumper to select default mode or On/Off switch.



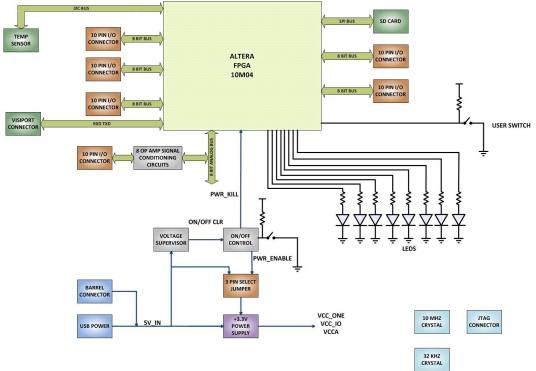
See section x for details on powering the board. Next, connect a JTAG Blaster or Altera USB Blaster to Jx for programming the MAX10 chip. Then, write some code, synthesize, and program the chip. See section x for details about programming the MaxProLogic board.

3 MaxProLogic Description

3.1 PCB Footprint

The PCB is 75mm x 50mm with four mounting holes (M2 metric screws) spaced as shown in the figure. These mounting holes are electrically isolated from all signals on the XEM6310. The two connectors (USB and DC power) overhang the PCB by approximately 1mm in order to accommodate mounting within an enclosure.

3.2 Functional Block Diagram

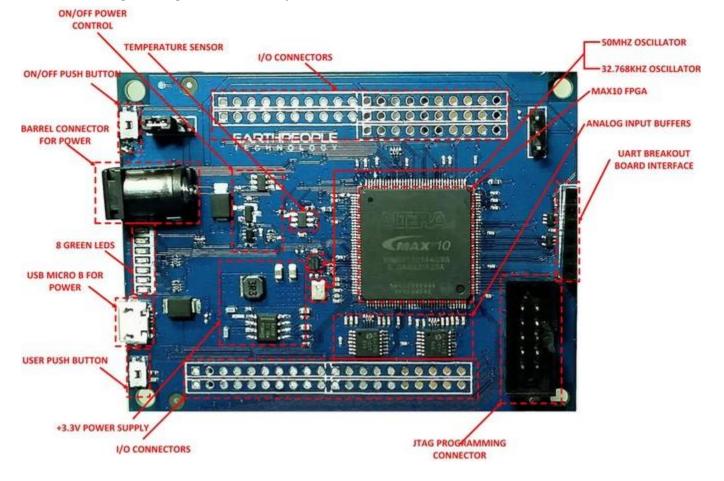


3.3 MAXPROLOGIC SPECIFICATIONS

- MAX 10 10M04SA FPGA FROM INTEL/ALTERA 4,000 Logic Elements;
- 2.2 Mbit On chip Flash; 189 Kbit On Chip SRAM 8 Analog Input
- Channels; 12 bit; 1MSamples/Second 65 Available I/O's at connectors
- 65 Inputs/Outputs available at connectors on board



- 8 Green User configurable LEDs 1 Power Pushbutton Switch; 1 User Configurable Pushbutton Switch
- On Board MicroSD Card Slot
- Two Power options: Standard USB (+5V @ 2Amp) Using Micro-B connector; 5mm Barrel Connector Accepts +12V @ 3Amp
- Switching Power Supply, Provides stable output under high load stress
- On/Off controller uses push button, optional bypass mode to allow board power up without On/Off control
- Two Clocks: 50MHz Oscillator; 32.768KHz Oscillator
- On board interface to Standard USB to Serial Adapters
- Temperature Sensor with +/-0.3C Resolution
- Standard Programming Connector fits any Altera USB Blaster





3.4 FPGA

The MaxProLogic includes the Intel/Altera 10M04SAE144C8G FPGA. It is an EQFP 144 pin package. This FPGA incorporates both the configuration flash and the ADC on the chip. This is different from conventional FPGAs as these two items are usually off chip. Access to the configuration flash is transparent to the user. The user does have access to the User Flash. This access is provided through the Altera MegaFunctions at the project level. Access to the ADC is also provided through the Altera MegaFunctions at the project level.

Parameter	10M04SA
LEs (Logic Elements)	4,000
Block memory (Kb)	189
User flash memory1 (KB)	16-156
18 x 18 multipliers	20
Phase-locked loops (PLLs)	2
Internal configuration	Dual
Analog-to-digital converter	8 Channels, 1MSa/Sec
(ADC)	
External memory interface	Yes
(EMIF)	
Inputs/Outputs	101

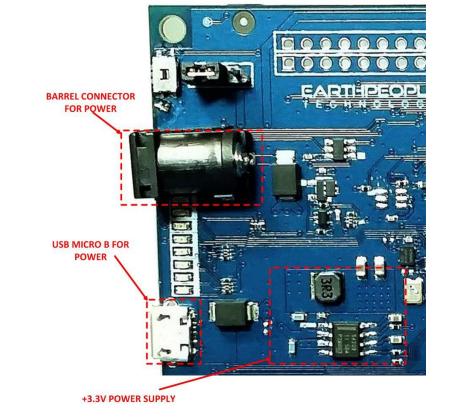
3.5 Power Supply

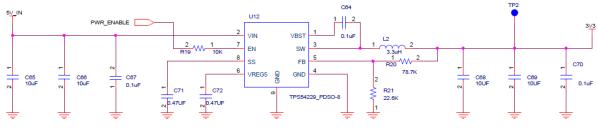
The MaxProLogic is designed to be operated from one of four different power sources:

- standard USB cable from Laptop/PC.
- +5 VDC wall charger (phone charger) through USB cable.
- +5.5 to +9 VDC supplied through the DC power jack.
- +5.5 to +9 VDC supplied through I/O connector pin.

This provides power for a high-efficiency switching regulator on-board to provide +3.3 VDC. The power supply provides reliable power under dynamic loads and high frequency switching internal to the FPGA.



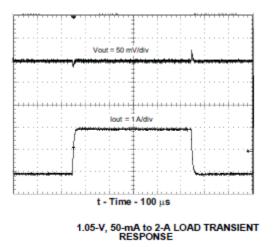




Buck Switching Regulator

The TPS54229 is a very stable synchronous buck converter. This allows it to provide a fast transient response. You can view this response here:



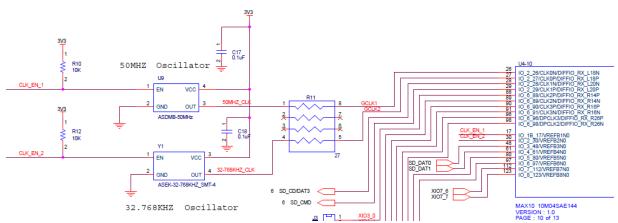


The TPS54229 has an Enable signal that allows the power supply to turn off. This signal is the net PWR_ENABLE. It is controlled by the On/Off controller MAX16054 or default. The operation of the On/Off controller is explained in section x. The default level for the PWR_ENABLE is high. The default level allows the power supply to turn on when power is applied to the board.

3.6 Clock Domains

There are two clock domains external to the MAX10 FPGA, 50 MHz and 32.768 KHz. The 50 MHz oscillator is a +3.3VDC device that provides a high speed clock to the FPGA. It is a CMOS device that provides a stable 50 MHz at \pm 50 ppm. This clock can be used directly in the user code or use it as an input to one of the PLL's internal to the FPGA. It is intended that this clock will drive the logic of the user code. If a different clock frequency is required in the user code, use the PLL scale up/down to produce the desired clocking.





The 32.768 KHz oscillator is a +3.3VDC device that provides a stable source for low speed clocking. This oscillator can be used for lower speed functions such as PWM for motor control, Audio, or controlling LEDs.

Both oscillators have an enable signal that are individually controlled by the FPGA. The oscillators are operational when the enable signal is high. The oscillators are in a low power, no output mode when the enable signals are low.

Enable Signal	High	Low
CLK_EN_1 (50MHz Oscillator)	Oscillator output On	Oscillator output Off
CLK_EN_2 (32.768 KHz Oscillator)	Oscillator output On	Oscillator output Off

3.7 Digital I/Os

The MaxProLogic has eight 10 pin headers that provide 64 digital Inputs and Outputs. All of the I/O's are +3.3 VDC only. All I/O's connect directly from the FPGA to one of the ten pin headers.



All I/O's are organized into separate banks of the FPGA. There are eight banks. These different banks provide different output speed technologies. Programmable Open Drain The optional open-drain output for each I/O pin is equivalent to an open collector output. If it is configured as



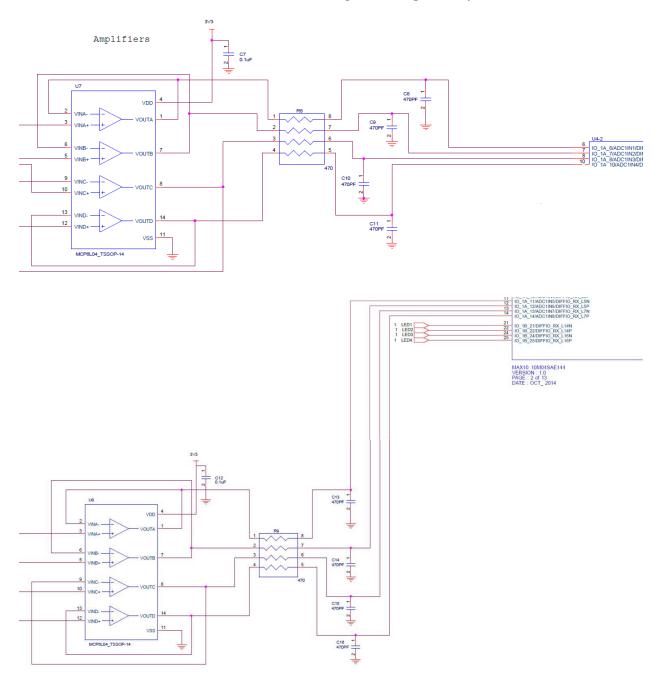
an open drain, the logic value of the output is either high-Z or logic low. Use an external resistor to pull the signal to a logic high. Programmable Bus Hold Each I/O pin provides an optional bushold feature that is active only after configuration. When the device enters user mode, the bushold circuit captures the value that is present on the pin by the end of the configuration. The bushold circuitry holds this pin state until the next input signal is present. Because of this, you do not require an external pull-up or pull-down resistor to hold a signal level when the bus is tristated. For each I/O pin, you can individually specify that the bus-hold circuitry pulls non-driven pins away from the input threshold voltage-where noise can cause unintended high-frequency switching. To prevent over-driving signals, the bus-hold circuitry drives the voltage level of the I/O pin lower than the VCCIO level. If you enable the bus-hold feature, you cannot use the programmable pull-up option. To configure the I/O pin for differential signals, disable the bushold feature. Programmable Pull-Up Resistor Each I/O pin provides an optional programmable pull-up resistor during user mode. The pull-up resistor weakly holds the I/O to the VCCIO level. If you enable the weak pull-up resistor, you cannot use the bus-hold feature. Programmable Current Strength You can use the programmable current strength to mitigate the effects of high signal attenuation that is caused by a long transmission line or a legacy backplane.

To provide maximum flexibility to system designers, all FPGA I/O are intrinsically bidirectional. Based on the I/O direction specified in the configuration file, each I/O can be configured as input-only, output-only, or bidirectional. While the output buffer of a bidirectional I/O has always included a dynamic output enable signal, MAX 10 FPGAs also include an input buffer enable/disable capability. When an input buffer is dynamically disabled, the buffer presents its last known state to the FPGA core fabric, so that no inputs inside the FPGA are left floating.

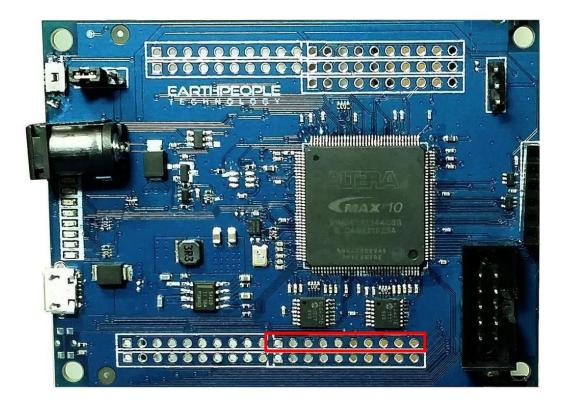
3.8 Analog Input

The MaxProLogic features an analog input of eight buffered channels. Each channel has its own 1MHz Unity Gain Amplifier to provide isolation and filtering. The Unity Gain Amplifier provides the best isolation between channels when using a Sample and Hold ADC and a high speed multiplexor.









The ADC provide the MAX 10 devices with built-in capability for on-die temperature monitoring and external analog signal conversion. The ADC solution consists of hard IP blocks in the MAX 10 device periphery and soft logic through the Altera Modular ADC IP core. The ADC solution provides you with built-in capability to translate analog quantities to digital data for information processing, computing, data transmission, and control systems. The basic function is to provide a 12 bit digital representation of the analog signal being observed. The ADC monitors up to 8 single-ended external inputs with a cumulative sampling rate of one megasymbols per second (Msps).

Intel MAX 10 ADC Conversion

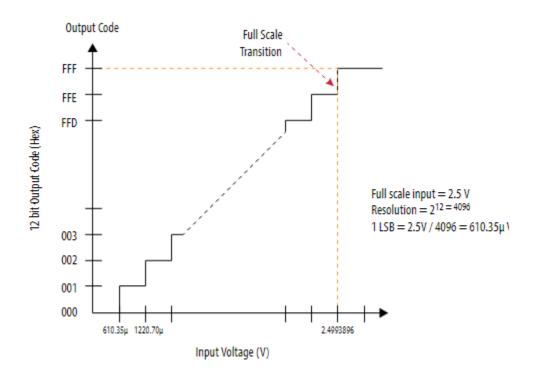
The ADC in dual supply Intel® MAX® 10 devices can measure from 0 V to 2.5 V. In single supply Intel® MAX® 10 devices, it can measure up to 3.0 V or 3.3 V, depending on your power supply voltage.



In prescaler mode, the analog input can measure up to 3.0 V in dual supply Intel® MAX® 10 devices and up to 3.6 V in single supply Intel® MAX® 10 devices.

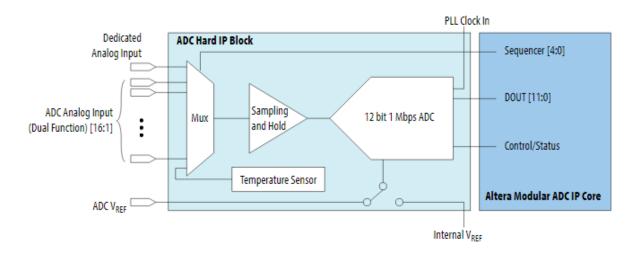
The analog input scale has full scale code from 000h to FFFh. However, the measurement can only display up to full scale -1 LSB.

For the 12 bits corresponding value calculation, use unipolar straight binary coding scheme.



The Intel® MAX® 10 ADC is a 1 MHz successive approximation register (SAR) ADC. If you set up the PLL and Altera Modular ADC IP core correctly, the ADC operates at up to 1 MHz during normal sampling and 50 kHz during temperature sensing.





3.8.1 Voltage Representation Conversion

Use the following equations to convert the voltage between analog value and digital representation.

Conversion from Analog Value to Digital Code Digital Code=(VINVREF)×212Digital Code=(VINVREF)×212

Conversion from Digital Code to Analog Value Analog Value=Digital Code×(VREF212)Analog Value=Digital Code×(VREF212)

3.8.2 Calculation Example for V_{REF} of **2.5 V**

Analog voltage value to digital code (in decimal), where signal in is 2 V:

Digital Code= $(22.5) \times 4096 = 3277$ Digital Code= $(22.5) \times 4096 = 3277$ Digital code to analog voltage value, approximation to 4 decimal points:

Analog Value = 3277×(2.54096)=2.0000

3.8.3 ADC Analog Input Pins

The analog input pins support single-ended and unipolar measurements. The ADC block in Intel[®] MAX[®] 10 devices contains two types of ADC analog input pins:

- Dedicated ADC analog input pin—pins with dedicated routing that ensures both dedicated analog input pins in a dual ADC device has the same trace length.
- Dual function ADC analog input pin—pins that share the pad with GPIO pins.



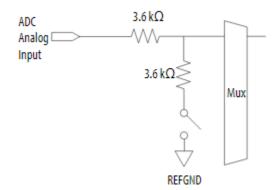
If you use bank 1A for ADC, you cannot use the bank for GPIO.

Each analog input pin in the ADC block is protected by electrostatic discharge (ESD) cell.

3.8.4 ADC Prescaler

The ADC block in Intel® MAX® 10 devices contains a prescaler function.

The prescaler function divides the analog input voltage by half. Using this function, you can measure analog input greater than 2.5 V. In prescaler mode, the analog input can handle up to 3 V input for the dual supply Intel[®] MAX[®] 10 devices and 3.6 V for the single supply Intel[®] MAX[®] 10 devices.



The prescaler feature is available on these channels in each ADC block:

• Single ADC device—channels 8 and 16 (if available)

3.8.5 ADC Clock Sources

The ADC block uses the device PLL as the clock source. The ADC clock path is a dedicated clock path. You cannot change this clock path.

For devices that support two PLLs, you can select which PLL to connect to the ADC. You can configure the ADC blocks with one of the following schemes:

- Both ADC blocks share the same clock source for synchronization.
- Both ADC blocks use different PLLs for redundancy.

If each ADC block in your design uses its own PLL, the Intel[®] Quartus[®] Prime Fitter automatically selects the clock source scheme based on the PLL clock input source:

• If each PLL that clocks its respective ADC block uses different PLL input clock source, the Intel[®] Quartus[®]Prime Fitter follows your design (two PLLs).



If both PLLs that clock their respective ADC block uses the same PLL input clock source, the Intel®Quartus® Prime Fitter merges both PLLs as one.

In dual ADC mode, both ADC instance must share the same ADC clock setting.

Related information

PLL Locations, Intel® MAX® 10 Clocking and PLL User Guide

3.8.6 ADC Voltage Reference

Each ADC block in Intel® MAX® 10 devices can independently use an internal or external voltage reference. There is only one external VREF pin in each Intel® MAX® 10 device. Therefore, if you want to assign external voltage reference for both ADC blocks in dual ADC devices, share the same external voltage reference for both ADC blocks.

Intel recommends that you use a clean external voltage reference with a maximum resistance of 100 Ω for the ADC blocks. If the ADC block uses an internal voltage reference, the ADC block is tied to its analog voltage and the conversion result is ratiometric.

3.8.7 Creating Intel MAX 10 ADC Design

To create your ADC design, you must customize and generate the ALTPLL and Altera Modular ADC IP cores.

The ALTPLL IP core provides the clock for the Altera Modular ADC IP core.

- 1. Customize and generate the ALTPLL IP core.
- 2. Customize and generate the Altera Modular ADC IP core.
- 3. Connect the ALTPLL IP core to the Altera Modular ADC IP core.
- 4. Create ADC Avalon slave interface to start the ADC.

3.8.8 Parameters Settings for Generating ALTPLL IP Core

Navigate through the ALTPLL IP core parameter editor and specify the settings required for your design. After you have specified all options as listed in the following table, you can generate the HDL files and the optional simulation files.

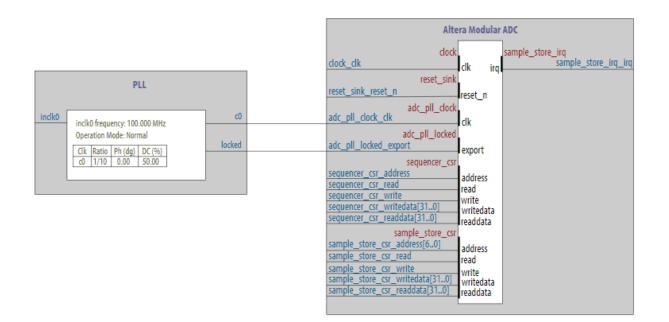
For more information about all ALTPLL parameters, refer to the related information.

3.8.9 Completing ADC Design

The ADC design requires that the ALTPLL IP core clocks the Altera Modular ADC IP core.

Generate the ALTPLL and Altera Modular ADC IP cores with the settings in the related information.





- 1. Create the design as shown in the preceding figure.
- 2. Connect the c0 signal from the ALTPLL IP core to the adc_pll_clock_clk port of the Altera Modular ADCIP core.
- 3. Connect the locked signal from the ALTPLL IP core to the adc_pll_locked_export port of the Altera Modular ADC IP core.
- 4. Create the ADC Avalon slave interface to start the ADC.

Table 12. ALTPLL Parameters Settings. To generate the PLL for the ADC, use the following settings.					
Tab	Parameter	Setting			
Parameter Settings > General/Modes	What is the frequency of the inclk0 input?	Specify the input frequency to the PLL.			



Table 12. ALTPLL Paramet	ers Settings. To generate the PL	L for the ADC, use the following settings.
Tab	Parameter	Setting
Parameter Settings > Inputs/Lock	Create an 'areset' input to asynchronously reset the PLL	Turn off this option.
	Create 'locked' output	Turn on this option. You need to connect this signal to the adc_pll_locked port of the Altera Modular ADC or Altera Modular Dual ADC IP core.
Output Clocks > clk c0	Use this clock	Turn on this option.
	Enter output clock frequency	Specify an output frequency of 2, 10, 20, 40, or 80 MHz. You can specify any of these frequencies. The ADC block runs at 1 MHz internally but it contains a clock divider that can further divide the clock by a factor of 2, 10, 20, 40, and 80.
		Use this same frequency value in your Altera Modular ADC or Altera Modular Dual ADC IP core. You need to connect this signal to the adc_pll_clock port of the Altera Modular ADC or Altera Modular Dual ADC IP core. Different ADC sampling rates support different clock frequencies. For a valid sampling rate and clock frequency combination, refer to the related information.

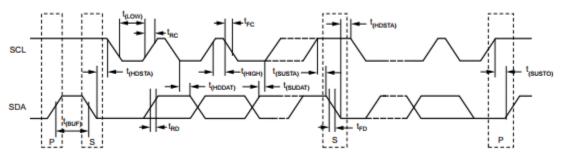


3.9 Temperature Sensor

The MaxProLogic includes a Temperature Sensor from Texas Instruments, the TMP100. This digital sensor has an accuracy of $\pm 1^{\circ}$ C. without requiring calibration or external component signal conditioning. Device temperature sensors are highly linear and do not require complex calculations or look-up tables to derive the temperature. The on-chip, 12-bit ADC offers resolutions down to 0.0625°C.

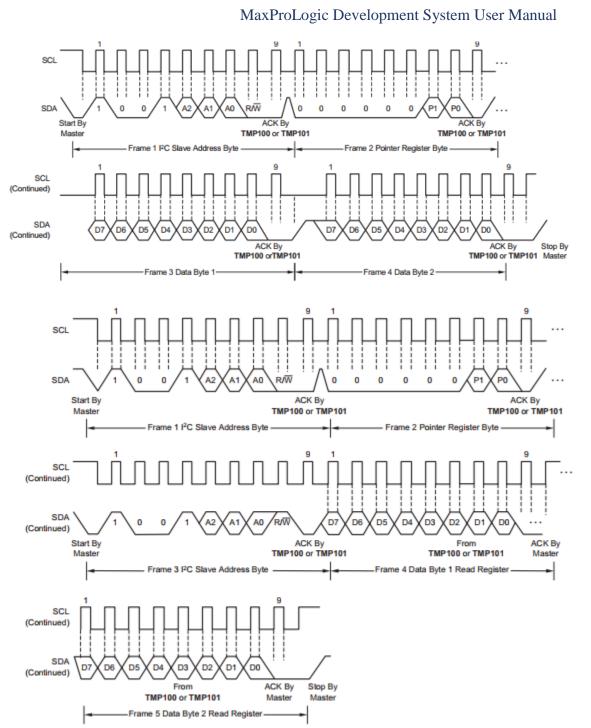
The sensor is accessed through an I2C bus. This bus is connected to the FPGA through pins 100 and 101. The bus uses 10K pullups on both the SDA and SCK signals.

Timing Diagrams The TMP100 and TMP101 devices are Two-Wire, SMBUs, and I 2C interface-compatible. Figure 6 to Figure 9 describe the various operations on the TMP100 and TMP101. The following list provides bus definitions. Parameters for Figure 6 are defined in the Timing Requirements section.



Bus Idle: Both SDA and SCL lines remain HIGH. Start Data Transfer: A change in the state of the SDA line, from HIGH to LOW, while the SCL line is HIGH, defines a START condition. Each data transfer is initiated with a START condition. Stop Data Transfer: A change in the state of the SDA line from LOW to HIGH while the SCL line is HIGH defines a STOP condition. Each data transfer is terminated with a repeated START or STOP condition. Data Transfer: The number of data bytes transferred between a START and a STOP condition is not limited and is determined by the master device. The receiver acknowledges the transfer of data. Acknowledge: Each receiving device, when addressed, is obliged to generate an Acknowledge bit. A device that acknowledges must pull down the SDA line during the Acknowledge clock pulse in such a way that the SDA line is stable LOW during the HIGH period of the Acknowledge clock pulse. Setup and hold times must be taken into account. On a master receive, the termination of the data transfer can be signaled by the master generating a Not-Acknowledge on the last byte that is transmitted by the slave. Figure 6. I 2C Timing Diagram



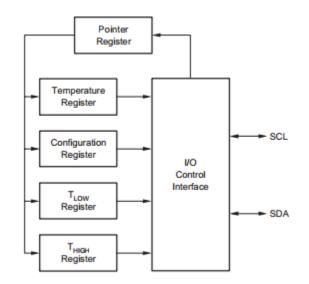


Programming 7.5.1 Pointer Register Figure 10 shows the internal register structure of the TMP100 and TMP101 devices. The 8-bit Pointer Register of the TMP100 and TMP101 devices

Page 21



is used to address a given data register. The Pointer Register uses the two LSBs to identify which of the data registers respond to a read or write command. Table 4 identifies the bits of the Pointer Register byte. Table 5 describes the pointer address of the registers available in the TMP100 and TMP101 devices. The power-up reset value of P1 and P0 is 00.



Pointer Register Byte (pointer = N/A) [reset = 00h] Table 4. Pointer Register Byte

[P7	P6	P5	P4	P3	P2	P1	P0
[0	0	0	0	0	0	Regist	er Bits

Pointer Addresses of the TMP100 and TMP101 Registers Table 5. Pointer Addresses of the TMP100 and TMP101 Registers

P1	P0	TYPE	REGISTER
0	0	R only, default	Temperature Register
0	1	R/W	Configuration Register
1	0	R/W	T _{LOW} Register
1	1	R/W	T _{HIGH} Register

Temperature Register The Temperature Register of the TMP100 or TMP101 devices is a 12-bit, read-only register that stores the output of the most recent conversion. Two bytes must be read to obtain data, and are described in Table 6 and Table 7. The first 12 bits are used to indicate temperature, with all remaining bits equal to zero. Data format for temperature is summarized in Table 1. Following power-up or reset, the Temperature Register reads 0°C until the first conversion is complete. Table 6. Byte 1 of the Temperature Register



D7	D6	D5	D4	D3	D2	D1	D0
T11	T10	Т9	Т8	T7	Т6	T5	T4

Table 7. Byte 2 of the Temperature Register

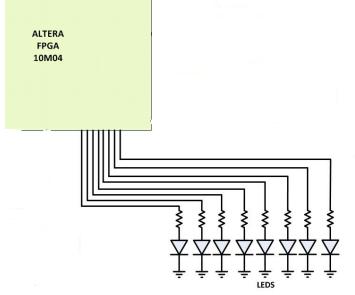
D7	D6	D5	D4	D3	D2	D1	D0
Т3	T2	T1	то	0	0	0	0

Configuration Register The Configuration Register is an 8-bit read and write register used to store bits that control the operational modes of the temperature sensor. Read and write operations are performed MSB-first. The format of the Configuration Register for the TMP100 and TMP101 devices is shown in Table 8, followed by a breakdown of the register bits. The power-up or reset value of the Configuration Register is all bits equal to 0. The OS/ALERT bit reads as 1 after power-up or reset value. Table 8. Configuration Register Format

	BYTE	D7	D6	D5	D4	D3	D2	D1	D0
[1	OS/ALERT	R1	R0	F1	F0	POL	тм	SD

3.10 LEDs

The MaxProLogic includes eight user LEDs and One LED to indicate when the FPGA has been successfully configured. The LEDs are directly driven from the FPGA.



They use the +3.3V I/O's along with a 220 Ohm series resistor for each LED. This provides the following current through the LEDS



$$I_{LED} = \frac{V_0 - V_F}{R}$$
$$I_{LED} = \frac{3.3V - 2.0V}{220}$$
$$I_{LED} = 5.9mA$$

The code to drive the LEDs is either zero (1'b0) or floating (1'bz). First, declare the LED as an output. In the example below, the vector LED is set to 'reg' because it is driven in an always block.

```
module EPT_10M04_AF_S2_Top (
    input wire
                                CLK 10MHZ,
    input wire
                                CLK 32KHZ,
    input wire
                                RST,
    output wire
                                CLK 10MHZ ENABLE,
                                CLK 32KHZ ENABLE,
    output wire
                                             11
    output wire
                   [7:0]
                                XIO 1,
                   [7:0]
                                XIO 2,
                                             11
    input wire
    input wire
                  [7:0]
                                XIO_3,
                                             11
                                XIO 4,
    output wire
                  [8:0]
                                             11
                                PWR ENABLE,
    input wire
    output reg
                   [7:0]
                                  LED
    );
```

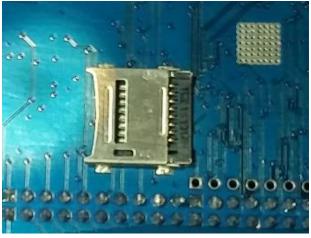
To turn the selected LED on, set the signal equal to 1'b0. This will apply a ground to the cathode side of the LED and allow current to flow through the circuit turning the LED on. To turn the selected LED off, set the signal equal to 1'bz. This will float the cathode side of the LED and no current will flow through the LED.



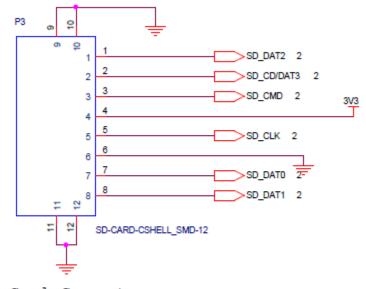
```
//-----
// Set the LED outputs
//-----
                   _____
                        _____
always @ (posedge CLK_10MHZ or negedge RST)
begin
 if(!RST)
     LED <= 8'hz;
 else
 begin
   if(state[LOAD_LEDS])
   begin
     if ( led reg[0] )
        LED[0] = 1'b0;
     else
        LED[0] = 1'bz;
     if ( led reg[1] )
        LED[1] = 1'b0;
     else
        LED[1] = 1'bz;
     if ( led_reg[2] )
        LED[2] = 1'b0;
     else
        LED[2] = 1'bz;
```

3.11 SD Card Interface

The SD Connector uses a standard cell phone SD Card Connector. This connector has a mechanism that opens out away from the board on a hinge and allows the card to be inserted into the hinged door. Close the hinged door and pull it towards the front of the MaxProLogic to secure it.







SD Card Connector

3.11.1 SD Card Protocol

There's a ton of information out there on using the MMC/SD SPI protocol to access SD cards but not much on the native protocol. This page hopes to rectify that with information helpful to those implementing a SD host or trying to understand what they're seeing on an oscilloscope.

3.11.2 References

- The full MMC Specification is available as <u>JESD84-A44</u> from the <u>JEDEC website</u>
- The <u>SD Simplified Specification</u> from the SD Card Association covers most of the protocol
- The full SD specification is available after joining the SD Card Association

3.11.3 Compatibility

MMC, SD, and SDHC cards are broadly compatible at the electrical and framing level. A properly designed controller should be able to handle them all. Some differences are:

- MMC cards are available in both High Voltage (2.7 3.6 V) and Dual Voltage (2.7 3.6 and 1.70 - 1.95 V)
- MMC is designed to support multiple cards on the same bus



- During initialisation MMC cards are clocked at 400 kHz or less
- MMC, SD, and SDHC all have different initialisation sequences

3.11.4 Protocol

The protocol is a strict master/slave arrangement where data is clocked synchronously from the host to the card or from the card to the host over digital lines. Commands are sent from the host to the card and all commands have either no response, a 48 bit response, or a 136 bit response. Some commands may also start a data transfer to or from the card.

There are three types of signal:

- CLK, carrying the clock signal from the host
- CMD, carrying commands from the host and responses from the card
- DAT, carrying data from the host or data from the card

There may be 1, 4, or 8 DAT lines. SD Cards can run at 0 - 25 MHz in Default Mode or 0 to 50 MHz in High-Speed Mode. MMC cards come in different grades that can run at up to 20, 26, or 52 MHz. No matter what all cards start up in 3.3 V, single DAT, and low speed mode with any other features negotiated during the initialisation.

3.11.5 Physical layer

All communication are at 3.3 V logic levels with 3.3 V being a high and 0 V being a low. CLK comes from the host and idles low. CMD and DAT are bidirectional and idle high. All are driven in a push/pull mode for speed.

Data is clocked into the host or card on the rising edge of CLK and changes on the falling edge. This is equivalent to the SPI (0, 0) mode.

3.11.6 Framing

The framing is a bit unusual. It feels like it was written by a embedded software engineer instead of a hardware or protocol engineer as the framing and use of CRCs is unusual and inconsistent. The advantage is that the framing maps through to a software only implementation pretty well.

All transfers start with a zero start bit and finish with a one stop bit. A card may signal that it is still working on the response by keeping the CMD line high until the response is ready.

All commands are 48 bits (6 bytes) long and all responses are either 48 bits (6 bytes) or 136 bits (17 bytes) long. The 48 bit transfers can be thought of as an 8 bit message ID, 32 bit argument, and 8 bit checksum.



Bytes are transferred most significant bit first. Words are transferred most significant byte first.

3.11.7 Commands and Responses

A command or response has the following format:

Bit	#	Value	Name
47	1	0	Start bit
46	1	1 for commands, 0 for responses	Transmitter bit
45-40	6		Command ID
39-8	32		Argument
7-1	7		CRC
0	1	1	Stop bit

The CRC is a 7 bit CRC with polynomial $x^7 + x^3 + 1$. A table driven form can be found in the Linux kernel under <u>lib/crc7.c</u>. Bitwise forms may be generated using <u>pycrc</u> with the parameters

```
--width=7 --poly=9 --reflect-in=0 --reflect-out=0 --xor-out=0 --xor-in=0 --generate c
--algorithm=bit-by-bit-fast
such as
```

```
for (int b = 0; b < 8; b++)
{
    uint bit = crc & 0x40;
    if ((data & 0x80UL) != 0)
    {
        bit ^= 0x40;
    }
    data <<= 1;
    crc <<= 1;
    if (bit != 0)
    {
        crc ^= 0x09;
    }
}</pre>
```

Note that the final CRC must be ANDed with 0x7F.



The CRC seed is zero and is calculated over the start, transmitter, command ID, and argument fields. The resulting CRC is compared for equality with the CRC from the message.

Some examples are:

An APP_CMD (55) command that prefixes a SD specific command Bytes: 0x77000000065 Fields:

- Start bit = 0
- Transmitter = 1
- Command = 55 (decimal)
- Argument = 00000000
- CRC = 0x32
- Stop bit = 1

The CRC can be generated by feeding 0x77, 0x00, 0x00, 0x00, 0x00 into the CRC function above.

The response to the APP_CMD Bytes: 0x370000012083 Fields:

- Start bit = 0
- Transmitter = 0 (this is a response)
- Response = 55 (decimal)
- Argument = 00000120
- CRC = 0x41
- Stop bit = 1

The response to CMD3 SEND_RELATIVE_ADDR Bytes: 0x03B368050019 Fields:

Start bit = 0



- Transmitter = 0 (this is a response)
- Response = 3 (decimal)
- Argument = 0xB3680500
- CRC = 0x0C
- Stop bit = 1

Note that this cards Relative Card Address (RCA) is 0xB368

3.11.8 Data

Data has the following format:

Bit	#	Value	Name
4113	1	0	Start bit
4112-17	512*8		Data bits
16-1	16		CRC
0	1	1	Stop bit

Note that this is for a typical transfer of a block of 512 bytes. The host knows from the command that was sent how many bytes to expect back. There is no other way of knowing the message length.

The CRC is is the ITU-T V.41 16 bit CRC with polynomial 0x1021. A table driven version can be found in the Linux kernel under <u>lib/crc-itu-t.c</u>. Note that there is no such thing as 'the' CRC16 so make sure you get the right one.

Unlike the commands or responses the CRC is calculated over all of the data bytes and does not include the start bit. The calculated CRC is checked for equality with the received CRC.

PENDING: Add an example data message with CRC.

3.11.9 Handover

The CMD and DAT lines are bidirectional. Handover occurs at the end of a command where both the host and the card switch to input mode for two clocks before the card starts driving in push/pull mode. Some commands must be responded to in a fixed number of clocks but most allow an arbitrary time before the response must start.

3.11.10 Initialisation

To initialise a SD or SDHC card, send the following:

• Write 74 clocks with CMD and DAT high



- Write CMD0 GO IDLE STATE. This will reset the card.
- Write CMD8 SEND_IF_COND for 3.3 V parts. If any SDHC cards are

present then you will get a wired-OR response with 0x3F as the command and 0xFF as the CRC and stop bit. Note that this must be sent or SDHC cards will not respond to the following steps

- Write CMD55 APP_CMD
- Receive a 55 response
- Write ACMD41 SD SEND OP COND
- Expect a wired-OR response with 0x3F as the command and 0xFF as the CRC and stop • bit
- Check the ready bit in the previous response. If the card is not

ready then repeat the CMD55/ACMD41 until it is

- Write CMD2 ALL SEND CID
- Expect a wired-OR response with 0x3F as the command and 0xFF as the CRC and stop • bit
- Write CMD3 SEND_RELATIVE_ADDR
- Expect a 3 response. The upper two bytes of the argument is the

Relative Card Address (RCA) which is used in the next step

- Write CMD7 SELECT_CARD with the RCA
- Expect a 7 response

The card is now selected and ready to transfer data. See Figure 4-1 'SD Memory Card State Diagram' in the simplified spec for more information.

See section 4.7.4 'Detailed Command Description' in the simplified spec for more information on the commands and responses.

MMC cards are initialised using a similar but different method.

An example flow captured from Linux on a SC2440 is:

Phase	Command	Response	Notes
CMD0	400000000 95	None	
CMD55	7700000000 65	370000012083	



ACMD41 SEND_OP_COND	6900100000 5F	3F00FF8000FF	Card is busy
CMD55	7700000000 65	37000012083	
ACMD41	6900100000 5F	3F00FF8000FF	Card is still busy
CMD55	7700000000 65	37000012083	
ACMD41 SEND_OP_COND	6900100000 5F	3F80FF8000FF	Card is ready
CMD2 ALL_SEND_CID	420000000 4D	3F1D4144534420202010A0400BC1008 8ADFF	
CMD3 SEND_RELATIVE_A DDR	4300010000 7F	03B368050019	RCA of 0xB36 8

Note the missing CMD8 as this controller does not support SDHC. I didn't capture the final CMD7.

3.11.11 Reading

Once initialised reading from a card is straight forward.

To read a single page:

- Send CMD17 READ_SINGLE_BLOCK with the offset to read from as the argument
- Receive on the DAT lines
- Send CMD12 STOP_TRANSMISSION once the block has been received

To read consecutive pages:

- Send CMD18 READ_MULTIPLE_BLOCK with the starting offset as the argument
- Receive as many blocks as you want on the DAT lines
- Send CMD12 STOP_TRANSMISSION once the done

Note that there will be a response to these commands and the response may be interleaved with the data. See Figure 3-3: (Multiple) Block Read Operation for more information.



On SDHC cards the offset is in terms of 512 byte blocks. On SD cards the offset is in bytes and the number of bytes received depends onCMD16 SET_BLOCKLEN

3.12 Power Input

The MaxProLogic is designed to be operated from one of four different power sources:

- Standard USB cable from Laptop/PC.
- +5 VDC wall charger (phone charger) through USB cable.
- +5.5 to +9 VDC supplied through the DC power jack.
- +5.5 to +9 VDC supplied through I/O connector pin.

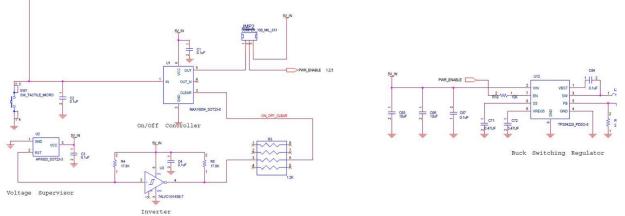
This provides power for a high-efficiency switching regulator on-board to provide +3.3 VDC. The power supply provides reliable power under dynamic loads and high frequency switching internal to the FPGA.

3.13 On/Off Control

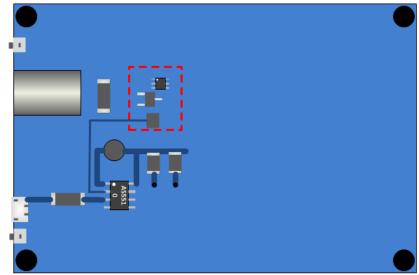
The MaxProLogic is equipped with an On/Off circuit. This circuit consists of a push button switch that is momentary contact and a MAX16054 controller chip. The controller senses the change in state of the momentary switch and drives the output "PWR_ENABLE" signal to the +3.3 VDC power supply. The enable pin of the TPS54229 has the following truth table:

Enable Pin	Output of TPS54229
High	+3.3 V
Low	0V



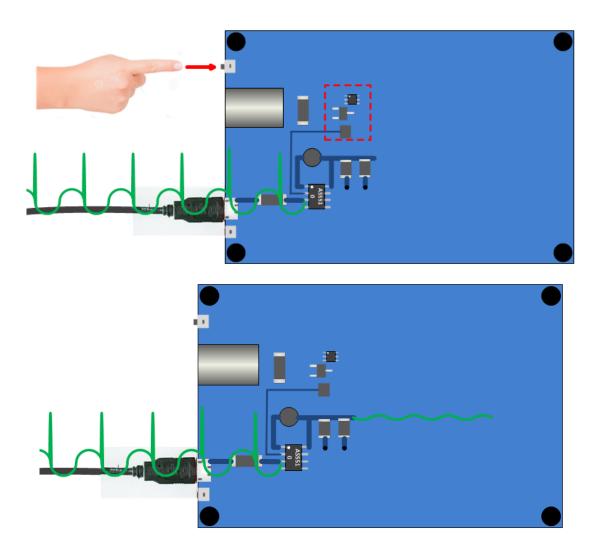


Use the Pushbutton SW1 to turn the power on to the MaxProLogic.



The TPS54229 Synchronous Buck Regulator will take any noisy input power and provide a smooth stable output. It has a fast transient response with a large inductor on the output.





The MAX16054 Controller chip also has a 'CLEAR' input. This input allows secondary device to cause a shutdown. The CLEAR input has the following truth table:

Clear Pin	Output of TPS54229
High	+3.3 V
Low	0V

The On/Off controller can be bypassed and allow the MaxProLogic to power up whenever power is applied. The JMP2 provides the bypass. Set the jumper to the '1' position to use the On/Off

Page 35

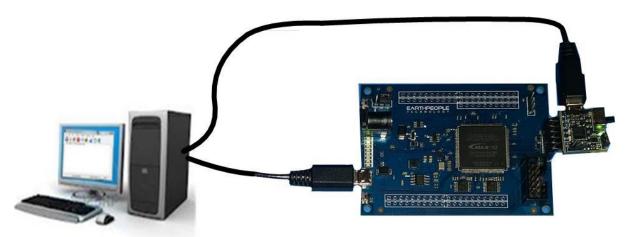


pushbutton switch. Uses the '2' position to allow the MaxProLogic to power up with board power.

3.14 Communications Interface

The MaxProLogic is equipped with a communications port that is compatible with FTDI Breakout Boards. It is a 6 pin female header that connects directly with most Breakout boards. The pinout:

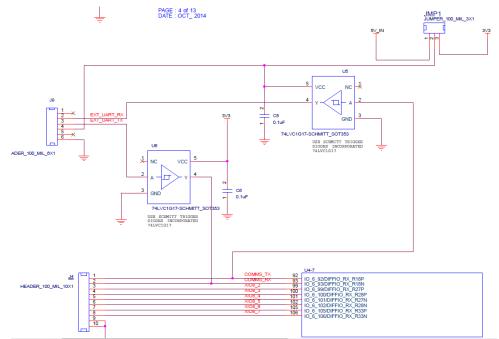
Communications Pin #	Signal
1	NC
2	RX
3	TX
4	VCC
5	NX
6	GND



The Communications Connector provides a path between the FTDI Breakout Board and the MAX10 FPGA. It brings both the RX and TX signals into the FPGA at the following pins.

FPGA Pin #	Signal
92	RX
93	TX

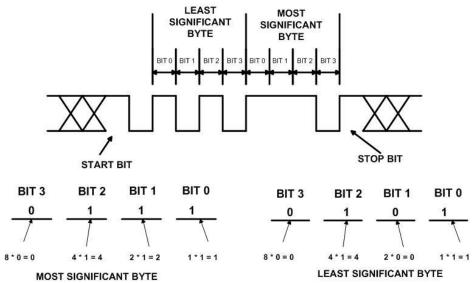




The MaxProLogic Communications Interface is compatible with both +5V and +3.3V Breakout Boards. The JMP1 provides a user selection jumper to select the +5V and +3.3V interface. There are two 74LVC1G17 Schmitt Trigger chips to provide voltage level compatibility with the FPGA.

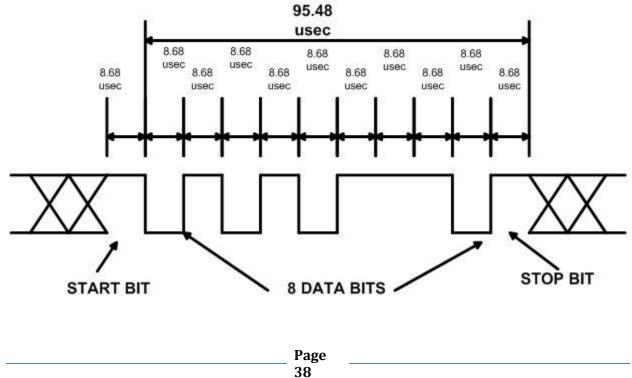
The FPGA must run Verilog code to perform the full duplex UART communications. The code must decode in the incoming RX signal and produce the outgoing TX signals with an accuracy of greater than 99%. This means up to 1% error is acceptable. The UART signals are based on a protocol. Where the UART protocol indicates one start bit, eight data bits and one stop bit.





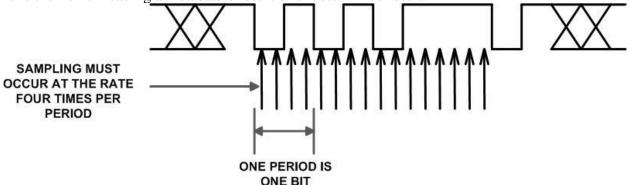
Each bit of the protocol must assert in the allocated time slot for each bit. What this means is that the clock used to time each bit of the protocol is embedded in the data. Both the sending device and receiving device must use the same clock rate. This embedding of the clock into data rate is called the baud rate.

The baud rate describes the data rate in bits per second. The timing diagram for a baud rate of 115,200 is 8.68 microseconds per bit.

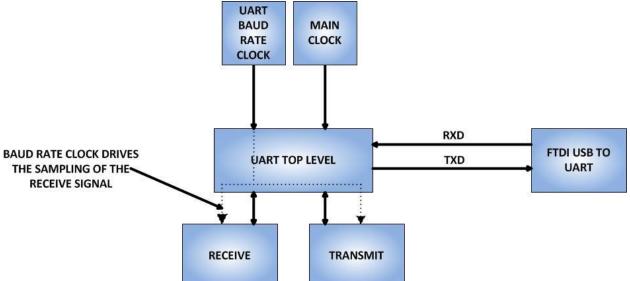




For the incoming RX signal, the FPGA must sample each bit of the protocol at four times per bit. This sampling rate is critical as the sending device clock and the receiving device clock are not synchronized. If the sampling rate is too low, a transition may be missed. If the sampling rate is only two times greater than the period of the receive device clock, samples could fall in the rising of falling edge of the received signal. Any time a sample occurs on the rising or falling edge, it runs the risk of reading a metastable result or undetermined result.



The Verilog code is written so that a special Baud Rate Clock is used to drive the UART Receive Block.



This code will use the Baud Rate Clock to drive a conditional if statement. The if statement will branch when a transition occurs and records the state of the signal (1 or 0). The Baud Rate Clock must be set to four times the incoming signal bit rate.



3.15 JTAG Interface

The MaxProLogic has a 5x2 header for use in programming the MAX10 FPGA via JTAG. The connector is located in the bottom right corner of the MaxProLogic. It is shrouded and keyed to allow easier insertion.



This connector uses the standard Altera Blaster connector pinout.

TCK	-	1	2	\vdash	GND
TDO	-	3	4	\vdash	VCC(TRGT)
TMS	-	5	6	\vdash	NC
NC	-	7	8	\vdash	NC
TDI	-	9	10	\vdash	GND

The VCC(TRGT) is set to +3.3V on the MaxProLogic. There are no jumper settings to make in order to program the MAX10 FPGA. Just connect a compatible Blaster to the connector and the PC, then use the Quartus software to program the FPGA.

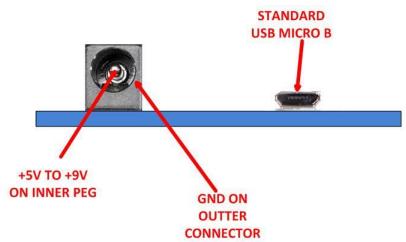




4 Powering the MaxProLogic

You can run the MaxProLogic from a laptop with 2.5W of power. Or you can run it from the +5V @ 2A wall USB chargers for 10W of power. The barrel connector can handle up to +9V @ 3 A for 27W of power.

- Standard USB cable from Laptop/PC.
- +5 VDC wall charger (phone charger) through USB cable.
- +5.5 to +9 VDC supplied through the DC power jack.
- +5.5 to +9 VDC supplied through I/O connector pin.



The barrel connector is the typical size used on many popular DIY boards such as the Arduino series. It has the following mechanical specs:

- 2.0mm Inner Diameter
- 5.5mm Outer Diameter



The barrel connector does not include a diode protection to prevent reverse polarity connection. So, care must be exercised when connecting up your cable to the barrel connector. Please ensure the correct polarity connections are made before connecting to the MaxProLogic. Also, there is no discrete protection to the power input. The power supply does include a high current protection circuit. The current limit is around 4.7Amps. But, the MaxProLogic is only designed to handle 2Amps of current. So, damage may occur to the MaxProLogic if the user does not exercise care in design and use of the Inputs/Outputs.



Power the MaxProLogic directly from the PC. +5V@0.5A

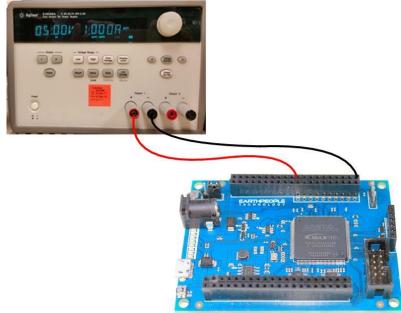


Power the MaxProLogic directly from the wall charger. +5V@2A





Power the MaxProLogic from the Barrel Connector. +9V@3A



Power the MaxProLogic directly from a power supply. +5V@2A

5 Installing Quartus

Locate the Quartus_17.1 folder on the EPT MaxProLogic Development System DVD.



rganize 🔻 💼 Open 🛛 Share with 🔻 Burn 🛛	New fold	ler			
MEGA_USB_CPLD_PROJECT_CD	^	Name	Date modified	Туре	Size
UNO_ANALOG_MONITOR_PROJECT_CD UNO USB CPLD PROJECT CD		🔇 12.1sp1_243_quartus_free_windows.exe	3/3/2013 1:12 AM	Application	3,633,187 k
Arduino IDE					
Documentation					
Drivers	E				
Projects_ActiveHost_32Bit					
Projects_ActiveHost_64Bit					
Projects_Arduino					
Projects_HDL Quartus 12.1sp1					
User Manual					
White Paper	-				

If you don't need to download Quartus, double click on the

12.1sp1_xxx_quartus_free_widows.exe (the xxx is the build number of the file, it is subject to change). The Quartus II Web Edition will start the installation process.

Vunicodedata.pyd Extracting 12.1sp1_243_quartus_free_windows\altera_installer\bin \win32api.pyd Extracting 12.1sp1_243_quartus_free_windows\altera_installer\bin \win32bip.pyd Extracting 12.1sp1_243_quartus_free_windows\altera_installer\bin \win32bip.pyd Extracting 12.1sp1_243_quartus_free_windows\altera_installer\bin \win32pip.pyd Extracting 12.1sp1_243_quartus_free_windows\altera_installer\bin \win32pip.expd Extracting 12.1sp1_243_quartus_free_windows\altera_installer\bin \win32pip.expd Extracting 12.1sp1_243_quartus_free_windows\altera_installer\bin \win32ui.pyd Extracting 12.1sp1_243_quartus_free_windows\altera_installer\bin \win32ui.pyd Extracting 12.1sp1_243_quartus_free_windows\altera_installer\bin \win32ui.pyd Extracting 12.1sp1_243_quartus_free_windows\devices\web\antiai.qda	🔇 Quartus II Web Editic	on (Free) (includes Nios II EDS)		х
C:\Users\NELSON~1\AppData\Local\Temp Browse	ADTRAA. Edi Will Will Will Will Edi Will Edi Will Will Edi Will Will Will Edi Will Will Will Will Will Will Will Wi	acting 12.1 isp1_243_quartus_free_windows\altera_installer n32api.pyd n32api.pyd n32api.pyd n32fie.pyd n32fie.pyd n32pie.pyd n32pie.pyd n32pie.oses.pyd n32pie.oses.pyd n32pie.got_21.sp1_243_quartus_free_windows\altera_installer n32pie.pt_243_quartus_free_windows\altera_installer n32pirace.pyd nacting 12.1sp1_243_quartus_free_windows\altera_installer n32pirace.pyd	\bin \bin \bin \bin \bin	^
	C:\\	Users\NELSON~1\AppData\Local\Temp 👻	Browse.	

When the install shield window pops up click "Yes" or if needed, enter the administrator password for the users PC. Click "Ok"

Next, skip down to the Quartus Installer section to complete the Quartus installation.

5.1 Downloading Quartus

The first thing to do in order build a project in Quartus is to download and install the application. You can find the latest version of Quartus at:

https://www.altera.com/download/dnl-index.jsp

Click on the Download Windows Version.





The next page will require you to sign into your "myAltera" account. If you do not have one, follow the directions under the box, "Don't have an account?"

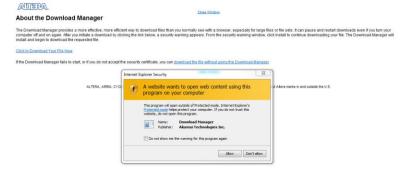
Solvices Version Tools & Services Version Address Version Version Address already exists in our system we will retrieve the associated information.) Create Your mail address already exists in our system we will retrieve the associated information.) Create Your already exists in our system we will retrieve the associated information.) Create Your already exists in our system we will retrieve the associated information.) Create Your already exists in our system we will retrieve the associated information.) Create Your already exists in our system we will retrieve the associated information.) Create Your already exists in our system we will retrieve the associated information.) Create Your and Address already exists in our system we will retrieve the associated information.) Create Your and Address already exists in our system we will retrieve the associated information.) Create Your and Address already exists in our system we will retrieve the associated information.) Create Your and Address already exists in our system we will retrieve the associated information.) Create Your Address already exists in our system we will retrieve the associated information.) Create Your Address already exists in our system we will retrieve the associated information.) Create Your Address already exists in our system we will retrieve the associated information.) Create Your Address Addr	MEASURABL			VEARS DERSHIP DERSHIP DERSHIP DERSHIP DERSHIP	Download	Center Do	Search	≗ myAltera Ac	• Q count
View myAltera Home Hanage Frofie Logout Heip myAltera Account Help Terms and Conditions Heip myAltera Account Help Terms and Conditions Change Password User Name Password Remember me Sign In Don't have an account? Create Your myAltera Account Your myAltera Account allows you to file a service request, register for a class, download software, and more. Enter your email address. (If your email address already exists in our system we will retrieve the associated information.)	Uevices	Design Tools & Servi	ices 🛛 🗎 End Markets	ڬ Technology 🛛 🗎 Tra	ining 🎽 Support	Nabout 🛛	≌ Buy		
Create Your myAlters Account Your myAlters account allows you to file a service request, register for a class, download software, and more. Enter your email address. (If your email address already exists in our system we will retrieve the associated information.)	View myAlter Manage Profi Change Pass Logout Help myAltera Acc	a Home le word ount Help	Home > myAltera Account	Sign In Jser Name Password Rememb		ot Your User	Name or Pass	word?	
Your myAltera account allows you to file a service request, register for a class, download software, and more. Enter your email address. (If your email address already exists in our system we will retrieve the associated information.)				-	on't have an accou	int?			
			Your myAltera i Enter your em (If your email a	ail address.					

Once you have created your myAltera account, enter the User Name and Password. The next window will ask you to allow pop ups so that the file download can proceed.



MEASURABLE ADVANTAGE*	Search	
■ Devices ■ Design Tools & Se	ervices 🔌 End Markets 🔌 Technology 🔌 Training 🛸 Support 🎽 About 🛸 Buy	
Design Software Quartus II Subscription Edition Quartus II Web Edition MegaCore IP Library ModelSim-Altera ModelSim-Altera Starter Niss II EDS Legacy Tools DSP Builder OS Support Archives Service Packs Design Software Licensing PAQ License Daemon Software Pogramming Software Quartus II Programmer Jam" STAPL Software Drivers Cable & Adapter Drivers	Chanch you for downloading Marce Support > Downleads > Therk you for downleading Dawnleads > Therk you for downleading Dawnleads value 12.1 for Windows (Alter Installeu) Dawnlead window Figure 1. Downlead window Wine: Software, name, see: Type: Application, (file size) Figure (marce, mare, see: Cancelle) Figure (marce, mare, see: Cancelle) Whe files from the Marent can bu weidt, ifts file pop cancelle window more context weidt, ifts file pop cancelle window more compare, Fryoud on the that the source, do ext 	icenses icensing and installation
Board Layout and Test BSDL Models SPICE Models IBIS Models IBIS Models Layout Review Worksheets Layout Review Worksheets Cadence PCB Libraries Gerber Files Lepacy Software MAX-PLUS II Other Legacy Software	un or save this software. <u>What's the rack?</u> Download TroubleShooting Steps Disable your pop-up blocker if a yellow pop-up blocker bar appears just below your Internet Explorer menu ba the yellow bar and select "Temporarily Allow Pop-ups." Any third-party pop-up blo disabled to download a file. Figure 2. Windows Explorer Pop-Up Blocker Pop-up blocket. To see this pop-up or additional options dick here You may also by-pass the pop-up blocker by clicking this link: <u>Download your file</u> .	x

Click on the "Allow Once" button. The next window will appear. It is the Download Manager.



Click on the "Allow" button. This will bring up the Save As dialog box. Save the altera_installer.external.exe to a download file.



					_
Organize - New folder				HE •	
Recent Places	^	Name	Date modified	Туре	
		\$11.1sp2_259_quartus_free_windows.exe	6/15/2012 3:18 PM	Application	
🕞 Libraries		12.0sp1_232_quartus_free_windows.exe	8/11/2012 11:20 AM	Application	
Documents		😋 12.1_177_quartus_free_windows.exe	11/25/2012 10:54	Application	
Music Fictures		😋 12.1sp1_243_quartus_free_windows.exe	3/3/2013 1:12 AM	Application	
Videos Videos Homegroup Riolly (HP-DESKTOP-3005)	E	🚱 altera_installer.external.exe	5/20/2012 8:52 PM	Application	
👰 Computer					
🚢 Gateway (C:)					
KINGSTON urDrive (H:)	-	· m			
- 10 Destand - (1.)					
File name: altera_installer.exte	mal.exe				
Save as type: *.exe					

Click the Save button. This will start the Download Manager.

	👎 Altera Download Mana	ger - Download	ing (35%)	
	File:	altera_installe	r.external.ex	e
	Progress:	7.84 MB/22.19) MB (35% d	one)
l				
	Status:	Downloading		
	Transfer Rate:	1.79 MB/sec		
	Time Left	00:08 secs		
	Connections:	4		
	Close window when d	one	□ Launch	when done
	Pau	se	🐐 Exit	v2.2.6

When it finishes, click the "Launch" button.

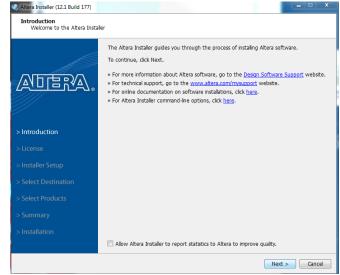


Click "Ok" and "Yes" to the following screen.

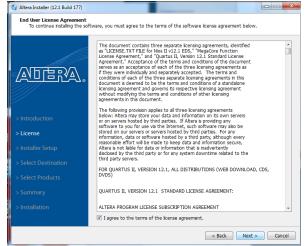
5.1.1 Quartus Installer

Click "Next" on the Introduction Window.





Click the checkbox to agree to the license terms. Then click "Next".



Click "Next" and accept the defaults.



Select the location of the	installation files.
	 Specify the Instalation Files Directory Select this option if you want the Altera Installer to retrieve installation files from your hard disk or local network. Installer Source Directory:
	Browse
	Download Installation Files from the Internet
	 Download Instalation Files from the Internet Select this option if you want the Altera Installer to retrieve installation files from the Internet.
	Select this option if you want the Altera Installer to retrieve installation files from the Internet. Specify how you are connected to the Internet:
	Select this option if you want the Altera Installer to retrieve installation files from the Internet.
	Select this option if you want the Altera Installer to retrieve installation files from the Internet. Specify how you are connected to the Internet: No proxy (recommended)
> License> Installer Setup	Select this option if you want the Altera Installer to retrieve installation files from the Internet. Specify how you are connected to the Internet: Ho proxy (recommended) Manual proxy configuration Web Proxy: Username:
 License Installer Setup Select Destination 	Select this option if you want the Altera Installer to retrieve installation files from the Internet: Specify how you are connected to the Internet: Who proxy (recommended) Manual proxy configuration Web Proxy: Port:

```
Click "Next" to accept the defaults
```

Select Destination Specify the software destination	tion directory.	
	Destination Directory:	
	c:\altera\12.1	Browse
	Available Space:	800 G
	Temporary Directory for installation files:	
	C:\Users\NELSON~1\AppData\Local\Temp	Browse
	Available Space:	800 G
	Remove saved installation files after completion.	
	Download only and install later.	
Introduction	Program Folder	
License	Specify the Program Folder:	
	Altera 12.1 Build 177 (Copy 2)	
Installer Setup	Existing Folders:	
Select Destination	Accessories	*
	Administrative Tools Altera	=
Select Products	Altera 12.1 Build 177	
C	Altera 12.1sp1 Build 243	
> Summary	AMD VISION Engine Control Center AUPEO!	
Installation	Barnes & Noble	
	Brother	*

At the Select Products Window, de-select the Quartus II Supbscription Edition by clicking on its check box so that the box is not checked. Then click on the check box by the Quartus II Web Edition (Free).



	Products		Insta	Size Do	ownload Size
		ubscription Edition (include	s Nios II EDS)	4.9 G	1.4 G
AL CONTRACTOR		II software (64-bit)		986 M	150 M
	Device R		No. T COC)	8.1 G	3.8 G
		/eb Edition (Free) (includes II software (64-bit)	NIOS II EDS)	986 M	3.8 G 150 M
	Quartus			2.5 G	2.4 G
		tera Starter Edition (Free)		3.1 G	491 M
	ModelSim-At			3.1 G	491 M
	DSP Builder			267 M	48 M
> Introduction					
> License					
> Installer Setup					
> Select Destination	Description				
	Installs the Quartu	s II Web Edition software.	The Ouertus II Web	Edition of	Augusta in A
> Select Products	a free, limited-feat	ure version of the Quartus	II Subscription Editi	on softwar	re that
		FPGA and CPLD device fam	ilies. Includes the fr	ee Nios II	
> Summary	Embedded Design	Suite.			~
		Install: [Download:		
> Installation	Space Required:		Jownload: 1.3 G	Color	t/Deselect All
	Space Available:		308 G	Delect	U Deselect All
	Share Walapie:	000 0 0	0000		
			< Back	Next >	Cancel
			< Back	Next >	Cano

🔇 Altera Installer (12.1 Build 177)	
Installation Summary Review the installation summa	ry to verify your software installation options, and then click Next to begin the installation.
	Destination Directory: c:\altern\12.1 Cache Location: C:\Users\VELSON-J\AppData\Loca\\Temp\atera_12.1_177 Program Folder: Atera 12.1 Buil 177 (Copy 2) Selected Products: Ourtus II Web Edition (Free) (includes Nos II EDS)
 > Introduction > License > Installer Setup > Select Destination 	Quartus II Web Edition (Free) (includes Nos II EDS) Quartus II software (64-bit) Arra II GX Cyclone II Cyclone II/III LS Cyclone IV E Cyclone IV 6X Cyclone V K Cyclone V K MAX II MAX V
> Select Products > Summary	ModelSim-Altera Starter Edition (Free)
> Installation	
	< Back Next > Cancel

Click "Next" to accept the defaults



Please wait while the Alter	ra sortware is installed.	os II EDS)		
				1%
	Products	Downloaded	Installed	
	Quartus II Web Edition (Free) (includes Nios II EDS) Quartus II Web Edition (Free) (includes Nios II EDS) Quartus II software (64-bit) Arris II GX Cyclone III/III LS Cyclone II/III LS Cyclone IV E			
	Cyclone IV GX Cyclone V Legacy Families			
	MAX II			
	MAX V ModelSim-Altera Starter Edition (Free)			
	Start Ouartus II 12.1			
> Select Products	Get Quartus II Subscription License			
	Create Desktop Shortcuts			
	Quartus II Web Edition (Free) and Nios II SBT / Nios II	GCC4 Toolchai	n	
> Installation	☑ ModelSim-Altera Starter Edition (Free)			
	Rate your installation experience			

Wait for the download to complete. The file is 3.5 GB, so this could take a couple of hours depending on your internet connection. When installation is complete, the following window appears.

Installation Please wait while the Altera	software is installed.		
	Installation successful.		
			10
	Products	Downloaded	Installed
	 Quartus II Web Edition (Free) (includes Nios II EDS) Quartus II Web Edition (Free) (includes Nios II EDS) Quartus II software (64-bit) Arria II GX 	~ ~ ~	<i>'</i> ,
	ltera Installer		1
> Introduction	The Altera Installer finished installing the * Quartus II Web Edition (Free) (includes Nios II EDS) * ModelSim-Altera Starter Edition (Free) software version 12.1.		~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
> License	software version 12.1.	1	¥
> Installer Setup	ОК	F	<i>'</i>
	Start Quartus II 12.1		
	Get Quartus II Subscription License		
	Create Desktop Shortcuts Image: Organization of the state of	GCC4 Toolchai	in
> Installation	ModelSim-Altera Starter Edition (Free)		
	Rate your installation experience		

Click "Ok", then click "Finish". The Quartus II is now installed and ready to be used.



6 Compiling, Synthesizing, and Programming the FPGA





The FPGA on the EPT-10M04-AF-S2can be programmed with the Active Transfer Library and custom HDL code created by the user. Programming the FPGA requires the use of the Quartus II software and a standard USB cable. There are no extra parts to buy, just plug in the USB cable. Once the user HDL code is written according to the syntax rules of the language (Verilog and VHDL) it can be compiled and synthesized using the Quartus II software. This manual will not focus on HDL coding or proper coding techniques, instead it will use the example code to compile, synthesize and program the FPGA.

6.1 Setting up the Project and Compiling

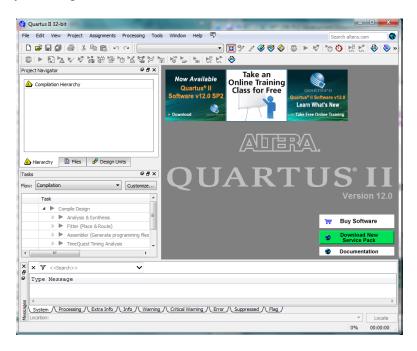
Once the HDL code (Verilog or VHDL) is written and verified using a simulator, a project can be created using Quartus II. Writing the HDL code and simulating it will be covered in later sections. Bring up Quartus II, then use Windows Explorer to browse to c:/altera/xxx/quartus/qdesigns create a new directory called: "EPT_10M04_AF_Blinky".

rganize 👻 🔭 Open 🛛 Include in library 👻	Share with	Burn New folder			800 -	
)} bin	*	Name	Date modified	Туре	Size	
🎉 bin64		EPT_Transfer_Test	3/13/2013 8:52 PM	File folder		
ja common		J fir_filter	3/3/2013 1:38 AM	File folder		
📕 cusp		incr_comp_makefile	3/3/2013 1:38 AM	File folder		
drivers	Е	whol_verilog_tutorial	3/3/2013 1:38 AM	File folder		
🎍 dsp_builder						
🎍 eda						
extlibs32						
libraries						
Imf gdesigns						
EPT_Transfer_Test						
ir_filter						
incr_comp_makefile						
whell_verilog_tutorial						
sopc_builder						
altera_inspector.log.zip						
91	-					





Open Quartus II by clicking on the icon



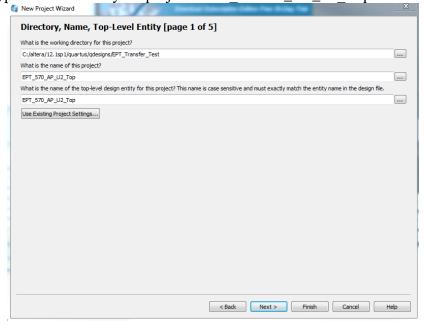
Under Quartus, Select File->New Project Wizard. The Wizard will walk you through setting up files and directories for your project.



🔮 Quartus II 32-bit File Edit View Project Assignments Processing Tools Window Help 🛡 D New... Ctrl+N - 😿 ൙ Open... Ctrl+O 🕷 🖄 🐜 🔊 🏷 🛸 👘 🐮 🖉 Close Ctrl+F4 *9*∂× New Project Wizard... Now Available ١O Quartus[®] II Software v12.0 SP2 🛃 Open Project... Ctrl+J С Save Project Close Project Save Ctrl+S Save As.. Save All Ctrl+Shift+S *9* ₽ × File Properties... omize... Create / Update • . Export... = Convert Programming Files... Page Setup... 👌 Print Preview 4 🖨 Print... Ctrl+P Recent Files • Recent Projects . Exit Alt+F4 System / Processing / Extra Info / Info / Warning / Critical Warning / Error / S ocation Starts the New Project Wizard

MaxProLogic Development System User Manual

At the Top-Level Entity page, browse to the c:/altera/xxx/quartus/qdesigns directory to store your project. Type in a name for your project "EPT_10M04_AF_S2_Top".



Page 54



Select Next. At the Add Files window: Browse to the $Projects_HDL EPT_10M04_AF_Blinky \src folder of the EPT USB-FPGA Development System CD. Copy the files from the \src directory.$

• EPT_10M04_AF_S2_Top.v

gory:		Device
General	Files	
Files Libraries Operating Settings and Conditions	Select the design files you want to include in the project. Click Add All to add all design files in the project directory to the project.	
Voltage Temperature	File name:	Add
Compilation Process Settings	Туре	Add All
Early Timing Estimate Incremental Compilation Physical Synthesis Optimizations	/EPT US8-CPLD Development System CD/Projects_HDL/EPT_Transfer_Test/src/write_control_logic.v Verilog HDL File /EPT US8-CPLD Development System CD/Projects_HDL/EPT_Transfer_Test/src/mea_control_logic.v Verilog HDL File /EPT US8-CPLD Development System CD/Projects_HDL/EPT_Transfer_Test/src/mea_mray.v Verilog HDL File	e Remove
EDA Tool Settings Design Entry/Synthesis Simulation	/EPT US8-CPLD Development System CD/Projects_HDU_EPT_Transfer_Test/src/ept/WireOR.v Verilop HDL File /EPT US8-CPLD Development System CD/Projects_HDU_EPT_Transfer_Test/src/EPT_570_AP_U2_Top.v Verilop HDL File /EPT US8-CPLD Development System CD/Projects_HDU_EPT_Transfer_Test/src/ept/src/etricy_active_trigger.vqm Verilop Quartus	e
Formal Verification Board-Level Analysis & Synthesis Settings	(PT US8-CPLD Development System CD)Projects_HDL/EPT_Transfer_Test/src/active_transfer_Ibrary.vom Verlog Quartu (PT US8-CPLD Development System CD)Projects_HDL/EPT_Transfer_Test/src/active_transfer_dom Verlog Quartu Verlog Quartu Verlog Quartu Verlog Quartu Verlog Quartu	s Ma Properties
Verlog HDL Input Default Parameters Fitter Settings TimeQuest Timing Analyzer Assembler Design Assistant SignalTap II Logic Analyzer Jougic Analyzer Interface PowerRay Power Analyzer Settings SSN Analyzer		

Select Next, at the Device Family group, select MAX II for Family. In the Available Devices group, browse down to 10M04SAE22 for Name.



Esmilue MAY IT				Show in 'Availa	able devices' lis	t	
Family: MAX II			•	Package:	Any		•
Devices: All			•	Pin count: Any			
				Speed grade:			-
Target device	Target device			Speed grade.	Ally		•
Auto device	selected by the Fitte	er		Name filter:			
🔘 Other: n/a							
Available devices:							
Available devices:	Core Voltage	LEs			UFM b	locks	_
Available devices: Name EPM570T100C3	Core Voltage	LEs 570	1		UFM b	locks	
Name EPM570T100C3	Core Voltage 3.3V		1		UFM b	locks	
Name EPM570T100C3 EPM570T100C4	Core Voltage 3.3V 3.3V	570			UFM b	locks	
Name EPM570T100C3 EPM570T100C4 EPM570T100C5	Core Voltage 3.3V 3.3V 3.3V 3.3V	570 570	1		UFM b	locks	
Name EPM570T100C3 EPM570T100C4 EPM570T100C5 EPM570T100I5	Core Voltage 3.3V 3.3V 3.3V 3.3V 3.3V	570 570 570	1 1		UFM b	locks	
Name EPM570T100C3 EPM570T100C4 EPM570T100C5 EPM570T100I5 EPM570T144A5	Core Voltage 3.3V 3.3V 3.3V 3.3V 3.3V 3.3V 3.3V	570 570 570 570	1 1 1		UFM b	locks	
Name EPM570T100C3 EPM570T100C4 EPM570T100C5	Core Voltage 3.3V 3.3V 3.3V 3.3V 3.3V 3.3V 3.3V 3.3	570 570 570 570 570 570	1 1 1 1		UFM b	locks	

Select Next, leave defaults for the EDA Tool Settings.



Tool Name <none> <none> <none> Timing Symbol Signal Integrity Boundary Scan</none></none></none>	Format(s)	Ru 	in Tool Automatically Run this tool automatically to synthesize th Run gate-level simulation automatically afte
<none> <none> Timing Symbol Signal Integrity</none></none>	<th></th> <th></th>		
<none> Timing Symbol Signal Integrity</none>	<none></none>	•	Run gate-level simulation automatically afte
Timing Symbol Signal Integrity	<none> <none></none></none>		
Symbol Signal Integrity	<none></none>		
Signal Integrity		•]	
	<none></none>		
Boundary Scan		-	
	<none></none>	•	

Select Next, then select Finish. You are done with the project level selections.



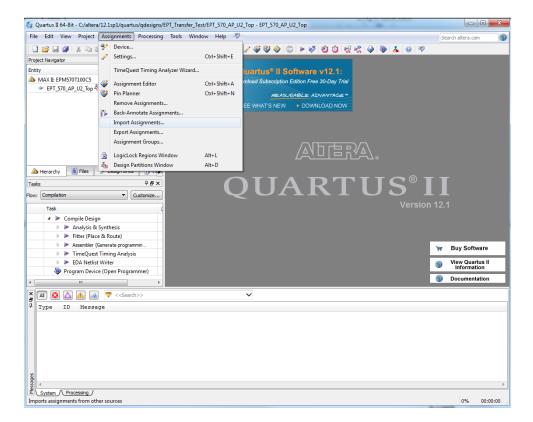
	New Project Wizard	× ×
	Summary [page 5 of 5]	
	When you click Finish, the project will be create	d with the following settings:
	Project directory:	C:/altera/12.0sp1/quartus/qdesigns
	Project name:	Active_Transfer_Example
	Top-level design entity:	Active_Transfer_Example
	Number of files added:	7
	Number of user libraries added:	0
	Device assignments:	
	Family name:	MAX II
	Device:	EPM570T100C5
	EDA tools:	
	Design entry/synthesis:	<none> (<none>)</none></none>
	Simulation:	<none> (<none>)</none></none>
	Timing analysis:	0
	Operating conditions:	
	VCCINT voltage:	3.3V
	Junction temperature range:	0-85 ℃
-		
		< Back Next > Finish Cancel Help

Next, we will select the pins and synthesize the project.

6.1.1 Selecting Pins and Synthesizing

With the project created, we need to assign pins to the project. The signals defined in the top level file (in this case: EPT_10M04_AF_S2_Top.v) will connect directly to pins on the FPGA. The Pin Planner Tool from Quartus II will add the pins and check to verify that our pin selections do not violate any restrictions of the device. In the case of this example we will import pin assignments that created at an earlier time. Under Assignments, Select Import Assignments.



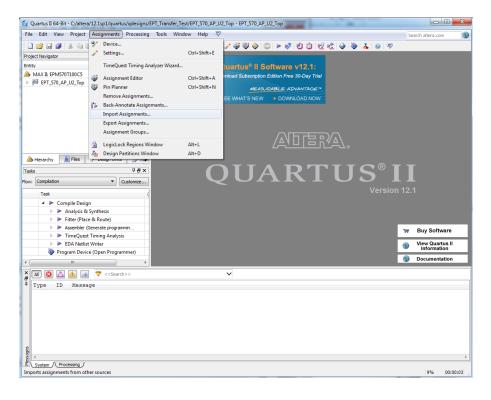


At the Import Assignment dialog box, Browse to the $Projects_HDL EPT_10M04_AF_Blinky EPT-10M04-AF-S2_Top folder of the EPT FPGA Development System DVD. Select the "EPT-10M04-AF-S2_Top.qsf" file.$

🕼 Import Assignments	× (DILA	RT	X
Specify the source and categories of assignments t) import.			
File name: echnology/EPT USB-CPLD Development	System CD/Projects_HDL/EPT_Transf	er_Test/Altera_EPM570_U2/EP	T_570_AP_U2_Top.qsf	Categories
Copy existing assignments into EPT_570_AP_Tr	ansfer_Test.qsf.bak before importing			Advanced
			OK Cancel	Help
				.4

Click Ok. Under Assignments, Select Pin Planner. Verify the pins have been imported correctly.





The pin locations should not need to be changed for EPT FPGA Development System. However, if you need to change any pin location, just click on the "location" column for the particular node you wish to change. Then, select the new pin location from the drop down box.



e	Edit V	iew Processing	Tools Window	Help 🐬			Searc	ch altera.com	¢
R	leport		₽ <i>₽</i> ×				87 8 10 8 10 10 10 10		
-	Report	not available				●▲∀●●∀●∆●●○○○	00000000		
					: 1		Ē		
					1			-	
						Top View			
					:8	Wire Bond			
					:Ă		Ē		
					ž⊻		2		
					3.₩				
E					: 💾			13	
I	asks		₽₽×				X		
		Show I/O Ban	ks ^		1	MAX II			
		Show VREF Gr	oups		=	EPM570T100C5			
		Show Edges			÷ 🖉	2111070110000			
	4	Show DQ/DQ	S Pins		:8				
		▶ x4 Mode			* O		00000VA0		
		III	*		24 27 28 20 80				
									_
Ě	× Named		Edit: 💢 🖌					Iter: Pins: all	-
	i 📃	Node Name	Direction	Location	I/O Bank	I/O Standard	Reserved	Current Strength	I.
1		LOWER[0]	Unknown	PIN_52	2	3.3-V LVdefault)		16mA (default)	
		LOWER[1]	Unknown	PIN_53	2	3.3-V LVdefault)		16mA (default)	
		LOWER[2]	Unknown	PIN_54	2	3.3-V LVdefault)		16mA (default)	
		LOWER[3]	Unknown	PIN_55	2	3.3-V LVdefault)		16mA (default)	
	🛛 🔇 🖉 LE	LOWER[4]	Unknown Unknown	PIN_56	2	3.3-V LVdefault)		16mA (default)	
	- I 🗠 1 r			PIN 57	2	3.3-V LVdefault)		16mA (default)	
		LOWER[5]		DINI FO	2	2 2 V IV Jackson (k)			
	🔷 LE	LOWER[6]	Unknown	PIN_58	2	3.3-V LVdefault)		16mA (default)	
	♦ LE ♦ LE	LOWER[6] LOWER[7]	Unknown Unknown	PIN_61	2	3.3-V LVdefault)		16mA (default)	
	 LE LE LE 	LOWER[6] LOWER[7] _UPPER[0]	Unknown Unknown Unknown	PIN_61 PIN_97	2 2	3.3-V LVdefault) 3.3-V LVdefault)		16mA (default) 16mA (default)	
	 LE LE LE LE LE 	LOWER[6] LOWER[7] _UPPER[0] _UPPER[1]	Unknown Unknown	PIN_61 PIN_97 PIN_96	2	3.3-V LVdefault) 3.3-V LVdefault) 3.3-V LVdefault)		16mA (default) 16mA (default) 16mA (default)	
	 Le Le Le Le Le Le Le 	LOWER[6] LOWER[7] _UPPER[0]	Unknown Unknown Unknown Unknown	PIN_61 PIN_97	2 2 2	3.3-V LVdefault) 3.3-V LVdefault)		16mA (default) 16mA (default)	
	 LE LE LE LE LE LE LE 	LOWER[6] LOWER[7] _UPPER[0] _UPPER[1] _UPPER[2]	Unknown Unknown Unknown Unknown Unknown	PIN_61 PIN_97 PIN_96 PIN_95	2 2 2 2	3.3-V LVdefault) 3.3-V LVdefault) 3.3-V LVdefault) 3.3-V LVdefault)		16mA (default) 16mA (default) 16mA (default) 16mA (default)	
	 LE LE LE LE LE LE LE LE 	LOWER[6] LOWER[7] _UPPER[0] _UPPER[1] _UPPER[2] _UPPER[3]	Unknown Unknown Unknown Unknown Unknown Unknown	PIN_61 PIN_97 PIN_96 PIN_95 PIN_92	2 2 2 2 2 2	3.3-V LVdefault) 3.3-V LVdefault) 3.3-V LVdefault) 3.3-V LVdefault) 3.3-V LVdefault)		16mA (default) 16mA (default) 16mA (default) 16mA (default) 16mA (default)	
	 LE 	LOWER[6] _LOWER[7] _UPPER[0] _UPPER[1] _UPPER[2] _UPPER[3] _UPPER[4]	Unknown Unknown Unknown Unknown Unknown Unknown Unknown	PIN_61 PIN_97 PIN_96 PIN_95 PIN_92 PIN_91	2 2 2 2 2 2 2 2	3.3-V LVdefault) 3.3-V LVdefault) 3.3-V LVdefault) 3.3-V LVdefault) 3.3-V LVdefault) 3.3-V LVdefault)		16mA (default) 16mA (default) 16mA (default) 16mA (default) 16mA (default) 16mA (default)	
	 LE <	LOWER[6] LOWER[7] UPPER[0] UPPER[1] UPPER[2] UPPER[3] UPPER[3] UPPER[5] UPPER[6] UPPER[7]	Unknown Unknown Unknown Unknown Unknown Unknown Unknown Unknown Unknown	PIN_61 PIN_97 PIN_96 PIN_95 PIN_92 PIN_92 PIN_89 PIN_87 PIN_86	2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	3.3-V LVdefault) 3.3-V LVdefault) 3.3-V LVdefault) 3.3-V LVdefault) 3.3-V LVdefault) 3.3-V LVdefault) 3.3-V LVdefault) 3.3-V LVdefault) 3.3-V LVdefault)		16mA (default) 16mA (default) 16mA (default) 16mA (default) 16mA (default) 16mA (default) 16mA (default) 16mA (default) 16mA (default)	
	 LE <	LOWER[6] LOWER[7] UPPER[0] UPPER[1] UPPER[2] UPPER[3] UPPER[4] UPPER[5] UPPER[6] UPPER[7] D[0]	Unknown Unknown Unknown Unknown Unknown Unknown Unknown Unknown Unknown Unknown	PIN_61 PIN_97 PIN_96 PIN_95 PIN_92 PIN_91 PIN_89 PIN_87 PIN_86 PIN_74	2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	3.3-V LVdefault) 3.3-V LVdefault) 3.3-V LVdefault) 3.3-V LVdefault) 3.3-V LVdefault) 3.3-V LVdefault) 3.3-V LVdefault) 3.3-V LVdefault) 3.3-V LVdefault)		16mA (default) 16mA (default) 16mA (default) 16mA (default) 16mA (default) 16mA (default) 16mA (default) 16mA (default) 16mA (default)	
	 2 2 4 4<	LOWER[6] LOWER[7] _UPPER[0] _UPPER[1] _UPPER[2] _UPPER[3] _UPPER[4] _UPPER[5] _UPPER[5] _UPPER[7] D[0] D[1]	Unknown Unknown Unknown Unknown Unknown Unknown Unknown Unknown Unknown Unknown	PIN_61 PIN_97 PIN_95 PIN_95 PIN_92 PIN_91 PIN_89 PIN_87 PIN_86 PIN_74 PIN_75	2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	3.3-V LVdefault) 3.3-V LVdefault) 3.3-V LVdefault) 3.3-V LVdefault) 3.3-V LVdefault) 3.3-V LVdefault) 3.3-V LVdefault) 3.3-V LVdefault) 3.3-V LVdefault)		16mA (default) 16mA (default) 16mA (default) 16mA (default) 16mA (default) 16mA (default) 16mA (default) 16mA (default) 16mA (default) 16mA (default)	
	> LE LE	LOWER[6] LOWER[7] _UPPER[0] _UPPER[1] _UPPER[2] _UPPER[3] _UPPER[5] _UPPER[6] _UPPER[6] _UPPER[7] D[0] D[1] D[2]	Unknown Unknown Unknown Unknown Unknown Unknown Unknown Unknown Unknown Unknown Unknown Unknown	PIN_61 PIN_97 PIN_96 PIN_95 PIN_92 PIN_92 PIN_89 PIN_87 PIN_86 PIN_74 PIN_75 PIN_98	2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	3.3-V LVdefault) 3.3-V LVdefault)		16mA (default) 16mA (default)	
	> LE LE	LOWER[6] LOWER[7] LOPER[0] UPPER[1] UPPER[2] UPPER[3] UPPER[3] UPPER[6] UPPER[6] UPPER[6] DIPER[7] D[0] D[1] D[2] D[3]	Unknown Unknown Unknown Unknown Unknown Unknown Unknown Unknown Unknown Unknown Unknown Unknown	PIN_61 PIN_97 PIN_95 PIN_95 PIN_92 PIN_91 PIN_89 PIN_87 PIN_86 PIN_74 PIN_74 PIN_75 PIN_98 PIN_99	2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	3.3-V LVdefault) 3.3-V LVdefault)		16mA (default) 16mA (default)	
	> LE >	LOWER[6] LOWER[7] LUPPER[1] LUPPER[1] LUPPER[2] LUPPER[3] LUPPER[5] LUPPER[5] LUPPER[5] LUPPER[5] D[0] D[1] D[2] D[3] VUSER_1	Unknown Unknown Unknown Unknown Unknown Unknown Unknown Unknown Unknown Unknown Unknown Unknown Unknown Unknown	PIN_61 PIN_97 PIN_96 PIN_95 PIN_92 PIN_91 PIN_89 PIN_87 PIN_86 PIN_74 PIN_75 PIN_98 PIN_99 PIN_15	2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	3.3-V LVdefault) 3.3-V LVdefault)		16mA (default) 16mA (default)	
	> LE > Stational > Stational > Stational	LOWER[6] LOWER[7] LUPPER[1] LUPPER[1] LUPPER[2] LUPPER[3] LUPPER[3] LUPPER[5] LUPPER[5] LUPPER[6] LUPPER[7] D[0] D[1] D[2] D[3] V USER_1 V USER_2	Unknown Unknown Unknown Unknown Unknown Unknown Unknown Unknown Unknown Unknown Unknown Unknown Unknown Unknown Unknown	PIN_61 PIN_97 PIN_95 PIN_92 PIN_91 PIN_83 PIN_86 PIN_74 PIN_98 PIN_98 PIN_99 PIN_15 PIN_100	2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	3.3-V LVdefault) 3.3-V LVdefault)		16mA (default) 16mA (default)	
	 LE L	LOWER[6] LOWER[7] LUPPER[1] LUPPER[1] LUPPER[2] LUPPER[3] LUPPER[5] LUPPER[5] LUPPER[5] LUPPER[5] D[0] D[1] D[2] D[3] VUSER_1	Unknown Unknown Unknown Unknown Unknown Unknown Unknown Unknown Unknown Unknown Unknown Unknown Unknown Unknown	PIN_61 PIN_97 PIN_96 PIN_95 PIN_92 PIN_91 PIN_89 PIN_87 PIN_86 PIN_74 PIN_75 PIN_98 PIN_99 PIN_15	2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	3.3-V LVdefault) 3.3-V LVdefault)		16mA (default) 16mA (default)	

Exit the Pin Planner. Next, we need to add the Synopsys Design Constraint file. This file contains timing constraints which forces the built in tool called TimeQuest Timing Analyzer to analyze the path of the synthesized HDL code with setup and hold times of the internal registers. It takes note of any path that may be too long to appropriately meet the timing qualifications. For more information on TimeQuest Timing Analyzer, see

http://www.altera.com/literature/hb/qts/qts_qii53018.pdf?GSA_pos=1&WT.oss_r=1&WT.oss=T imeQuest Timing Analyzer

Browse to the $Projects_HDL EPT_10M04_AF_Blinky \ EPT-10M04-AF-S2_Top folder of the EPT USB-FPGA Development System CD. Select the "EPT-10M04-AF-S2_Top.sdc" file.$



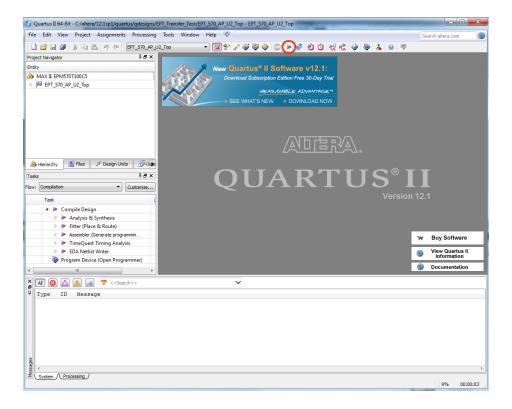
Irganize 🕶 📄 Open Share with 🕶 Burn	New fold	er				
Countertation Drives Projects, Active-loss, 3281 Projects, Active-loss, 3281 Projects, Active-loss, 3281 Projects, Active-loss, 4281 Projects, Vol. Pro		Name noremestal, db odput, life: PF 375, AP, UL, Toppid PF 375, AP, UL, Toppid PF 375, AP, UL, Toppid PF 375, AP, UL, Toppids PF 375, AP, UL, Toppids PF 375, AP, UL, Toppids PF 375, AP, UL, Toppids	Date modified 3/(2013-1.36 AM 1/30/(2013-842 PM 1/30/(2013-842 PM 1/30/(2013-842 PM 1/30/(2013-10.32 PM 1/30/(2013-10.32 PM 3/(2013-1.36 AM 1/29/(2013-10.30 PM	Type File folder File folder JDI file QPF File QSF File QMS File SDC File	Size 1 KB 2 KB 3 KB 2 KB 4 KB	
🕌 NXP						

Copy the file and browse to c: $alteraxxquartusqdesignsEPT_10M04_AF_Blinky directory.$ Paste the file.

janize 🕶 📄 Open Burn New fold	ler				
EPT_Transfer_Test fir filter	*	Name	Date modified	Туре	Size
		퉬 db	3/13/2013 9:42 PM	File folder	
incr_comp_makefile		鷆 incremental_db	3/13/2013 9:38 PM	File folder	
whole_verilog_tutorial		output_files	3/13/2013 9:42 PM	File folder	
🎉 sopc_builder		imulation	3/13/2013 9:42 PM	File folder	
altera_inspector.log.zip		EPT_570_AP_U2_Top.jdi	3/13/2013 9:42 PM	JDI File	1 KB
91		EPT_570_AP_U2_Top.qpf	3/13/2013 9:10 PM	QPF File	2 KB
book .	E	EPT_570_AP_U2_Top.qsf	3/13/2013 9:38 PM	QSF File	7 KB
Brother		EPT_570_AP_U2_Top.qsf.bak	3/13/2013 9:41 PM	BAK File	7 KB
CIMTEMP		EPT_570_AP_U2_Top.sdc	1/29/2013 10:30 PM	SDC File	4 KB
Documents and Settings					
Jolly					
Business Opportunities					
Lapitol_College					
Lode_FPGA					
Documents					
Download					

Select the Start Compilation button.





If you forget to include a file or some other error you should expect to see a screen similar to this:



🚭 Quartus II 64-Bit - C:/altera/12.1sp1/quartus/qdesigns/	PT_Transfer_Test/EPT_570_AP_U2_Top - EPT_570_AP_U2	_Top			23
File Edit View Project Assignments Processing	Tools Window Help 💎			Search altera.com	
🗋 📑 🖃 🥔 🕺 🕹 🖾 🖃 🖓 🖓 EPT_570_AP_L	12_Top 🔹 😿 🐓 🥒 📼	N 10 10 10 10 10 10 10 10 10 10 10 10 10	: 4 8 4 9 7		
Project Navigator 4 8 ×	Compilation Report - EPT_570_AP_U2_Top				
Entity	Table of Contents 4 8	Flow Summary			
MAX II: EPMd5701100C5 ▲ MAX II: EPMd5701100C5 ▲ BT_570_AP_U2_Top ■ Retrieve_transfer_LOTVE_TRANSFER_INST ■ edite=tigner_CATUR_TRANSFER_INST ■ edite=tigner_CATUR_TRANSFER_INST ■ eptWireOR.wwireOR	Table of Contents I Boy Summary If Dow Settings I Bow Settings If Pow Non-Default Global Settings I Bow Conton-Default Global Settings If Pow Ion-Default Global Settings I Bow Col Summary If Pow Ion Go Summary I Bow Col Summary If Pow Respect Time I Bow Col Summary If Pow Respect T	Flow Status Quartus II 64-Bit Version Revision Name Top-level Entity Name Family Device Timing Models	Flow Failed - Wed Mar 13 21:28 12.1 Build 243 01/31/2013 SP 1 EPT_570, AP_U2_Top EPT_570, AP_U2_Top MAX II EPM570T100C5 Final		
Task > > Compile Design > > Analysis & Synthesis > > Fitter (Place & Route) > > Assembler (Generate programmin > > TimeQuest Timing Analysis > > EDA Netlist Writer > Program Device (Open Programme)	SX Impilation was NOT successful (3 errors, 3 warnings) K				
	۱				
12128 Elaborating entity "acti 12128 Elaborating entity "acti 12128 Elaborating entity "acti 12128 Elaborating entity "acti 12106 Hode instance "BLOCK III 14001 Generated suppressed mes 2000 Quartus II 64-Bit Analys 2293001 Quartus II Full Compilat 4	reoR" for hierarchy "eptWireOR:wireO we_tringder" for hierarchy "active_tri we_tonsfer" for hierarchy "active_toick EIFO" instantiates undefined entity " ages file Civiters/12.1ps/jugartug/ is 6 Synthesis was unsuccessful. 1 er ion was unsuccessful. 3 errors, 3 wart m	gger:ACTIVE_TRIGGER ansfer:ACTIVE_TRANS BLOCK_TRANSFER_INS sync_fifo" gdesigns/EPT_Transf cor, 3 warnings	FER_INST" T"	PT_570_AP_U2_	Fop.m E
System (1) Processing (32)				9% 0	0:00:03

Click Ok, the select the "Error" tab to see the error.

File Edit View Project Assignment: Processing Tools Window Help Tools Window Help Tools Tools Window Help Tools Tools Window Help Tools Too	Quartus II 64-Bit - C:/altera/12.1sp1/quartus/qdesign	ns/EPT_Transfer_Test/EPT_570_AP_U2_Top - EPT_570_AP_U2	Тор	1. 10		= <mark>- X</mark>
Project Navgativ 9.0 × Project Navgativ 9.0 × MAX & EMMOTIDOCS Table of contents Image: Status Flow Status <	File Edit View Project Assignments Processi	ng Tools Window Help 🐬			Search altera.com	
Project Navgator 9.8 × Completion Report - BPT_STO, AP U/2, Top Table of content I Blow Summary I Blow Summary II Blow Summary III	🗋 💕 🗟 🥔 🐰 🖏 🖏 🕫 🗠 📴 EPT_570_A	P_U2_Top 🔹 🙀 🏏 🥥 🎯	► Ø Ø Ø ₩ ₹			
MAX & EMMSDT100C5 MAX & EMMSDT100C5 MAX & EMMSDT100C5 Max etc. transfer_XCTVE_TRANSFER_INST Max etc. transfer_XCTVE_TRANSFER_INST	Project Navigator 4 8 2					
MAXE EPMS70100CS Flow Sammary	Entity	Table of Contents	Flow Summary			
 Image: Strings Image: Strings<!--</td--><td>MAX II: EPM570T100C5</td><td></td><td>Flow Status</td><td>Flow Failed - Wed Mar 13 21</td><td>:28:50 2013</td><td></td>	MAX II: EPM570T100C5		Flow Status	Flow Failed - Wed Mar 13 21	:28:50 2013	
Total Comple Design Analysis & Synthesis Total Comple Design Total Comple Design Co	4 🔤 EPT_570_AP_U2_Top				1 SJ Web Edition	
Image: State (Second Section 2) Image: State (Second Section 2) Image: State (Second Section 2) Image: State (Second Section 2) Image: State (Second Section 2) Image: State (Second Section 2) Image: State (Second Section 2) Image: State (Second Section 2) Image: State (Second Second 2) Image: State (Second 2) Image: State (Second 2) Image: State (Second 2)	# active_transfer:ACTIVE_TRANSFER_INST	Flow Non-Default Global Settings				
Image: Active_things Brew Log Image: Active_thick_RDCX_TFIND Image: Active_thick_RDCX_TFINDS Image: Active_thick_RDCX_TFINDS Image: Active_thick_RDCX_TFINDS Image: Active_thick_RDCX_TFINDS Image: Active_thick_RDX_TFINDS Image: Active_thick_RDX_TFINDS Image: Active_thick_RDX_TFINDS Image: Active_thick_RDX_TFINDS <td< td=""><td><pre>## active_transfer_library:ACTIVE_TRANSFER_L</pre></td><td>Flow Elapsed Time</td><td></td><td></td><td></td><td></td></td<>	<pre>## active_transfer_library:ACTIVE_TRANSFER_L</pre>	Flow Elapsed Time				
• sync. fde BLOCK, NJFRO						
<pre># actue_block:BLOCK_[RANSER_MST # eptWercRwieGR # Herordry</pre>						
Herardry Files P Desgn Lines @0.000 Tasks 0 0 × 0 Files 0 0 × Torigon State Comple Design 0 0 > Files 0 0 × V > Analysis & Symthesis 0 > > > > > V > Assemble (Generate programm.) > > > > > > >> > EOA Netist Wider >		Analysis & Synthesis				
Tads 00% Tads 0	#99 eptWireOR:wireOR	-11				
Tads 00% Tads 0						
Table						
<pre>core Completion Castonize Task</pre>	À Hierarchy 🖹 Files 🔗 Design Units 🔂 🔅					
Tak Compile Design P Answide (Symmetry Symmetry Symmet	Tasks I de de la companya de la comp	×				
Tade Compile Design O According Design O Fitter (Place & Route) Fitter (Pl	Consistion					
Comple Design C D Analysis & Synthesis D First (Pace & Rout) D						
<pre>K > Andyn & Synthein 0 > Finter Direc Routd) > Amerika (Greenste programm.) > Trope ID Message</pre>		4				
P > Fater (Pice & Boute) P Assemble (Generate programm.) P ImeQuest Iming Analysis P ID Message Type ID Message O 12006 Node instance "BLOCK_IN_FIFO" instantiates undefined entity "ymo_fifo" Quartus II 64-Bit Analysis 6 Synthesis was unsuccessful. 1 erToty - withings Quartus II Full Complication was unsuccessful. 3 errors, 3 warnings		0(
<pre>> > Accender (Greente programme.) > > Eto A Hedinis Winder > > Eto A Hedinis Winder > > Tope ID Message 1 Type ID Message 1 Type ID Message 2 Dialow Rode instance "BLOCK IN FIFO" instantiates undefined entity sync_tifo" > O Quartus II 64-Bit Analysis 6 Synthesis was unsuccessful. 1 erfort, = withings 2 293001 Quartus II Full Compilation was unsuccessful. 3 errors, 3 warnings</pre>		0(
P > TureQuet Turing Analysis > EDA Helist Wite' Program Device (Deen Programmer) Type ID Message 3 12006 Node instance "BLOCK IN FIFO" instantiates undefined entity "Pyne_fifo" 3 12006 Node instance "BLOCK IN FIFO" instantiates undefined entity "Pyne_fifo" 3 12006 Node instance "BLOCK IN FIFO" instantiates undefined entity "Pyne_fifo" 3 223001 Quartus II 64-Bit Analysis 6 Synthesis was unsuccessful. 1 erToty - watnings 3 233001 Quartus II Full Compilation was unsuccessful. 3 errors, 3 warnings						
<pre>> EDA Helis Wiee Progum Device (Open Progumme)</pre>						
Program Device (Open Programme) m						
<pre>m</pre>						
Vesendo-> V						
 Type ID Message I2006 Node instance "BLOCK_IN_FIFO" instantiates undefined entity "pymo_fifo" Quartus II 64-Bit Analysis 6 Synthesis was unsuccessful. 1 erfor, 3 warnings 239001 Quartus II Full Compilation was unsuccessful. 3 errors, 3 warnings 		• • • • • • • • • • • • • • • • • • •				
 Type ID Message I 2006 Node instance "BLOCK IN FIFO" instantiates undefined entity (syno_fifo) Quartus II 64-Bit Analyzis 6 Synthesis was unsuccessful. 1 erfor, swinings 293001 Quartus II Full Compilation was unsuccessful. 3 errors, 3 warnings 	🕻 🖪 💽 🔝 💁 🤜 🔻 < <search>></search>	~				
 12006 Node instance "BLOCK IN FIFO" instantiates undefined entity (wyno_fifo") Quartus II 64-Bit Analysis 6 Synthesis vas unsuccessful. 1 erfor, 3 valinings 293001 Quartus II Full Compilation was unsuccessful. 3 errors, 3 varings 						-
 Quartus II 64-51: Analysis 6 Synthesis was unsuccessful. 1 erfor, s wafnings 293001 Quartus II Full Compilation was unsuccessful. 3 errors, 3 warnings 		N FIFO" instantiates undefined entity	who fife"			
	293001 Quartus II Full Compil-	ation was unsuccessful. 3 errors, 3 warn	nings			
	20					



Review the source and/or database to remedy the error.

Once the error has been fixed, re-run the Compile process. After successful completion, the screen should look like the following:

💱 Quartus II 64-Bit - C:/altera/12.1sp1/quartus/qdesigns/	/EPT_Transfer_Test/EPT_570_AP_U2_Top - EPT_570_AP_U	2_Top		
File Edit View Project Assignments Processing	Tools Window Help 💎			Search altera.com
🗋 💕 😹 🥔 🐰 🖏 🖏 🕫 (* 1975-570_AP)	U2_Top 🔹 🕅 🔀 🔮 🥥 💷) 🕨 🧭 🙆 🥲 🤘	2 🕹 😻 👗 💿 🔻	
Project Navigator 🕴 🗗 🗙	Compilation Report - EPT_570_AP_U2_To	p 🗵		
intity	Table of Contents	Flow Summary		
MAX II: EPM570T100C5	Flow Summary	Flow Status	Successful - Wed Mar 13 22:	
⊳ jiliž EPT_570_AP_U2_Top	Flow Settings Flow Non-Default Global Settings Flow Dos Summary Flow CS Summary Flow Log Analysis & Synthesis Flow Log	Quartus II 64-Bit Version Revision Name Top-level Entity Name Family Device Timing Models Total logic elements	12.1 Build 243 01/31/2013 SP EPT_570_AP_U2_Top EPT_570_AP_U2_Top MAX II EPM5701100C5 Final 557 / 570 (98 %)	1 SJ Web Edition
▲ Hierarchy 🖹 Files J ^P Design Units III	Quartus I E	Total pins Total virtual pins UFM blocks	51 / 76 (67 %) 0 0 / 1 (0 %)	
Fasks # 6	OK			
ow: Compilation Customize				
× ■ 😧 🛆 🕂 🕢 🔻 < <search>></search>	~			
 Running Quartus II 64-Bi Command: quartus dar 204018 Generated files "EFT_570 Quartus II 64-Bit EDA Ne 	est Timing Analyzer was successful. (****** gs_files=off EPT_57 p_vhd.sdo" in direc , 0 warnings	0_AP_U2_Top -c EPT_5	
System / Processing (134)	III			Þ
System Processing (134)				100% 00:00:24

At this point the project has been successfully compiled, synthesized and a programming file has been produce. See the next section on how to program the FPGA.

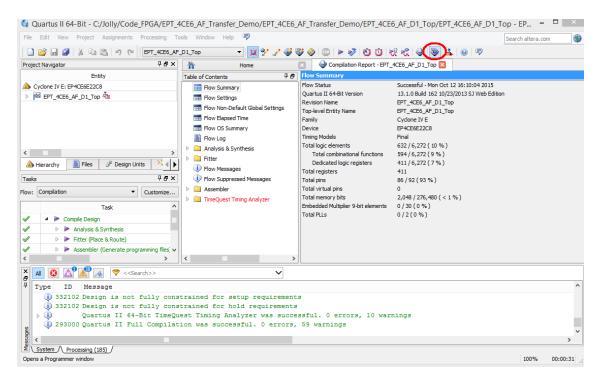
7 Programming the MaxProLogic

Configuring the FPGA is quick and easy. All that is required is a standard USB Micro B cable and a compatible Blaster Programmer. Connect the MaxProLogic to the PC, open up Quartus II, open the programmer tool, and click the Start button. To program the MAX10 Configuration Flash, connect the Blaster and ensure the JTAG Driver is loaded for Quartus II.





If the project created in the previous sections is not open, open it. Click on the Programmer button.



The Programmer Window will open up with the programming file selected. Click on the Hardware Setup button in the upper left corner.



👋 Programmer - (C:/Jolly/Code_FPGA	/EPT_4CE6_AF_Tran	sfer_Demo/El	PT_4CE6_AF	_Transfer_D	emo/El	PT_4CE6_	AF_D1_Top	D/EPT_4CE	6_AF_D	1_Top - EP		×
File Edit View Pr	ocessing Tools Windo	w Help 🐬									Search alte	ra.com	-
🔔 Hardware Setup	No Hardware					Mode:	ITAG		▼ Pro	gress:			
		ramming (for MAX II and M	AX V devices)							, coo. [
i≱ ³ ≌ Start	File	Device	Checksum	Usercode	Program/ Configure	Verify	Blank- Check	Examine	Security Bit	Erase	ISP CLAMP		
Stop													
Auto Detect													
💥 Delete													
🆄 Add File													
🕍 Change File													
Save File													
Add Device													
1 [™] u Up													
J [™] Down													

The Hardware Setup Window will open. In the "Available hardware items", double click on "EPT-Blaster v1.6b".

4	>		Hardware S	Setup		×
	Hardware Settings Select a programming hardware setup applie Currently selected ha Available hardware	es only to the rdware: No	up to use when pro		. This programming	_
	Hardware EPT-Blaster v1.5b	(64)	Server Local	Port MBUSB-0	Add Hardware Remove Hardware	
					Close	



If you successfully double clicked, the "Currently selected hardware:" dropdown box will show the "EPT-Blaster v1.6b".

	H	Hardware Se	etup	
Hardware Settings	JTAG Settings			
Select a programmir hardware setup app				This programming
Currently selected h	nardware: EPT-Bla	aster v1.5b (64)	[MBUSB-0]	>
Available hardwar	e items		_	
Hardware		Server	Port	Add Hardware
				Remove Hardware
				Close

Click on the "Add File" button

🕹 Programmer -	C:/Jolly/Code_FPGA/E	PT_4CE6_AF_Trar	nsfer_Demo/E	PT_4CE6_AF	_Transfer_D)emo/E	PT_4CE6_	AF_D1_Top	D/EPT_4CI	E6_AF_C	01_Top -	EP 🗖	×
File Edit View Pr	ocessing Tools Window	Help 🐬									Search	h altera.com	-
🔔 Hardware Setup	- EPT-Blaster v1.5b (64) [M	BUSB-0]				Mode:	Active Seria	Programming	▼ Pro	ogress:			
Enable real-time ISF	o to allow background program	nming (for MAX II and M	1AX V devices)										
⊌ ³ b Start	File	Device	Checksum	Usercode	Program/ Configure	Verify	Blank- Check	Examine	Security Bit	Erase	ISP CLAMP		
🗰 Stop													
Auto Detect													
💥 Delete													
Add File)												
🕍 Change File													
Save File													
Add Device													
1 ¹¹ Up													
J ¹ ∰ Down													
	L												



	Select Programm	ning File			×
Look in: 🎒 C: V	olly\Code_FPGA\EPT_4CE6_Ansfer_Demo\EP	T_4CE6_AF_D1_Top	• • •	00	:: 🔳
鰔 My Computer	Name	Size	Туре	Date Modified	
nelso_000	🎍 db			10/12/2015 4:	
	🕌 incremental db 📕 output_files			10/8/2015 9:20	
	\sim				
	<	_			>
File name:	•				Open
	/* 0				
Files of type: POF File:	s (*.pot)			▼	Cancel:

At the Browse window, double click on the output files folder.



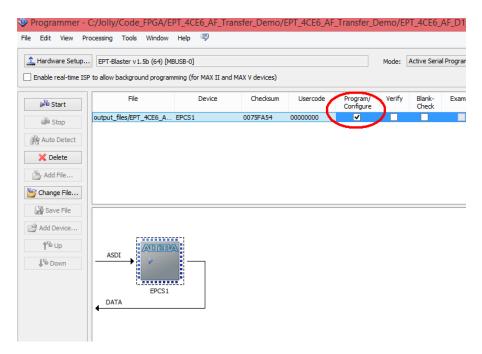
۵	Select Program	ming File	×
Look in: 🎒 C:\J	olly\Code_FPGA\EPT_4CE6_AFPT_4CE6_A	F_D1_Top\output_files 🔻 🔇 🕥 🌗 🔃	
My Computer	Name	Size Type Date Modified	
nelso_000	EPT_4CE6_AF_D1_Top.pof	128 KB pof File 10/12/2015 4:10:04 PM	1
~			
	<		>
File name: EPT_4CE	E6_AF_D1_Top.pof	Open	
Files of type: POF Files	s (*.pof)	▼ Cancel	

Double click on the "EPT_10M04_AF_Top.pof" file. Click the Open button in the lower right corner.

🔔 Hardware Setup	EPT-Blaster v1.5b (64) [MBI	JSB-0]				Mode:	Active Seria	al Prog
	P to allow background programm	-	d MAX V devices)					
Start	File	Device	Checksum	Usercode	Program/ Configure	Verify	Blank- Check	Ex
Stop	output_files/EPT_4CE6_A.	EPCS1	0075FA54	00000000				
alle								
Auto Detect								
💢 Delete								
灅 Add File								
Ghange File								
Change File								
Change File								
Add Device								
Save File								



Next, selet the checkbox under the "Program/Configure" of the Programmer Tool.



Click on the Start button to to start programming the FPGA. The Progress bar will indicate the progress of programming.

👋 Programmer -	C:/Jolly/Code_FPGA/E	PT_4CE6_AF_Tra	nsfer_Demo/	EPT_4CE6_AF	_Transfer_D)emo/E	PT_4CE6_	AF_D1_Top	D/EPT_4CE	E6_AF_C	01_Top -	EP 🗖	×
File Edit View Pr	rocessing Tools Window	Help 🔛									Searc	h altera.com	
🟦 Hardware Setup.	EPT-Blaster v1.5b (64) [M	BUSB-0]				Mode:	Active Seria	l Programming	* Pro	ogress:		2%	
Enable real-time IS	P to allow background program	nming (for MAX II and	MAX V devices)										
Start	File	Device	Checksum	Usercode	Program/ Configure	Verify	Blank- Check	Examine	Security Bit	Erase	ISP CLAMP		
🐝 Stop	output_files/EPT_4CE6_A	EPCS1	00755C55	00000000	V								
Auto Detect													
💥 Delete													
Add File													
Change File													
Save File													
Add Device													
1 ⁰⁰ Up	ADER												
J [™] Down													
	EPCS1												
	DATA												
	•												
													1.0



- C:/Jolly/Code_FPGA/EPT_4CE6_AF_Transfer_Der no/FPT 4CF6 AF Edit View Processing Tools Window Help Ţ EPT-Blaster v1.5b (64) [MBUSB-0] Enable real-time ISP to allow background programming (for MAX II and MAX V devices) Verify Blank- Examine Security Erase Check Bit File Device ISP CLAMP Checksum Usercode Program/ Configure Start **v** output_files/EPT_4CE6_A... EPCS1 00755C55 00000000 Stop Auto Detect 💢 Delete Add File. 쌸 Change File.. 🔛 Save File Add Device ********* ∱[™]Up 1 Down EPCS1 DATA

When the programming is complete, the Progress bar will indicate success.

At this point, the MAXPROLOGIC is programmed and ready for use.

The eight LEDs on the front of the MAXPROLOGIC will blink one after the other.

