AAAGENTSCIEV Cyclone 10 LP RefKit User Guide



Please read the legal disclaimer at the end of this document.

Revision 1.0

Table of Contents

Table of	f Figures	4
Chapter	r 1 - Cyclone 10 LP RefKit Development Board	5
1.1	About Arrow Cyclone 10 LP RefKit Board	5
1.2	Useful Links	5
1.3	Getting Help	6
Chapter	r 2 - Introduction to the Cyclone 10 LP RefKit Board	7
2.1	Layout and Components	7
2.2	Hardware variations	9
2.3	Block Diagram	
Chapter	r 3 - Connections and Peripherals of the Cyclone 10 LP RefKit Board	11
3.1	Board Status Elements	
3.2	Clock Circuitry	
3.3	Peripherals Connected to the FPGA	
3.3.	Communication and Configuration	
3.3.	8.2 Fast Ethernet	
3.3.	Serial Configuration Flash Memory	
3.3.	.4 HyperRAM	
3.3.	5 SDRAM Memory	
3.3.	.6 QSPI Flash Memory	
3.3.	EEPROMs	
3.3.	8.8 ADC/DAC	
3.3.	9.9 I2C Grove Connector	
3.3.	Arduino Header	21
3.3.	PMOD Connectors	
3.3.	9.12 VGA	
3.3.	8.13 LEDs	25
3.3.	9.14 Push Buttons	
3.3.	3.15 7-segment LED Display	27
3.3.	.16 Power Tree	
Chapter	r 4 - Software and Driver Installation	30
4.1	Installing Quartus Prime Software	
4.2	Installing Arrow USB Programmer2	
4.3	License	

Cha	pter 5 -	New Project with Cyclone 10 LP RefKit	34
5.	1 C	reating a new Blinky Project with Cyclone 10 LP RefKit	
5.	2 B	uilding a Blinky Project with Cyclone 10 LP RefKit	
	5.2.1	Block Diagram	
	5.2.2	Components of the Design	
	5.2.3	Catalog IP	
	5.2.4	Create and Configure PLL	
	5.2.5	Create and Configure the Counter	
	5.2.6	Create and Configure the Multiplexer	
	5.2.7	Adding the Components to the Schematic	
	5.2.8	Connecting the Components	50
	5.2.9	Add inputs, outputs to the schematic	53
	5.2.10	Analysis and Synthesis	55
	5.2.11	Adding Timing Constraints	56
	5.2.12	Pinning Assignments	57
	5.2.13	Compiling the Design	
	5.2.14	Reading the Compilation Report	61
Cha	pter 6 -	Configuring the Cyclone 10 LP RefKit	63
6.	1 C	onfigure the FPGA in JTAG mode	63
6.	2 Se	erial configuration flash memory programming	
	6.2.1	Programming File generation	67
	6.2.2	Device Programming	70
6.	3 Те	esting the Design	71
Cha	pter 7 -	Common Issues and Fixes	72
Cha	pter 8 -	- Appendix	73
8.	1 R	evision History	73
8.	2 Le	egal Disclaimer	74

Table of Figures

Chapter 1 - Cyclone 10 LP RefKit Development Board

1.1 About Arrow Cyclone 10 LP RefKit Board

The Cyclone 10 LP Reference Kit is a customizable development board that targets all kinds of applications with a wide range of interfaces. The board is based on Cyclone 10 LP FPGA, which is optimized for low-cost and low-power designs, making them ideal for high-volume and cost-sensitive applications. High-density sea of programmable gates and onboard resources allow implementation of Nios II 32-bit microcontroller IP, which provides the ideal solution for I/O expansion, chip-to-chip interfacing, industrial, automotive, and consumer applications.

The C10LP RefKit is equipped with an Arrow USB Programmer2, 2 ports 10/100Mbps Ethernet, SDRAM, HyperRAM, flash memory, VGA, 8-channel ADC/DAC, PMODs, and ARDUINO connectors making it a fully featured plug and play solution without any additional costs.

The C10LP RefKit board contains all the tools needed to use the board in conjunction with a computer that runs a 64-bit Linux / Microsoft Windows 10 operating system or later.

1.2 Useful Links

A set of useful links that can be used to get relevant information about the Cyclone 10 LP RefKit or the Cyclone 10 LP FPGA.

- <u>Cyclone 10 LP RefKit at Arrow Shop</u>
- <u>Cyclone 10 LP RefKit at Trenz Electronic Shop</u>
- Intel Cyclone 10 LP Webpage
- Cyclone 10 LP RefKit Wiki Page



1.3 Getting Help

Here are the addresses where you can get help if you encounter any problems:

• Arrow Electronics

<u>In Person</u> Arrow EMEA + 49 (0) 6102 5030 0

Online https://arrow.com

• Trenz Electronic GmbH

https://www.trenz-electronic.de/en/



Chapter 2 - Introduction to the Cyclone 10 LP RefKit Board

2.1 Layout and Components

Figure 1 and Figure 2 shows a top view and the bottom view of the board. It depicts the layout of the board and indicates the location of the various connectors and key components.



Figure 1 – Cyclone 10 LP RefKit Board (top view)



MOW



Figure 2 – Cyclone 10 LP RefKit Board (bottom view)

The following features are available on the Cyclone 10 LP RefKit board:

FPGA Device

• Intel Cyclone 10 LP 10CL055YU484C8G device. Features of the FPGA on the C10LP RefKit:

Basoursos	Device
Resources	10CL055
Logic Elements (LE)	55,856
M9K Memory (Kb)	2,340
18 x 18 Multiplier	156
PLLs	4
I/O	321

Memory Devices

- 64-256Mbit external SDRAM memory¹
- 64Mbit external HyperRAM memory
- 64-128Mbit external QSPI Flash memory¹
- 16Mbit EPCQ serial configuration flash memory
- 2× 2Kbit serial MAC-Address EEPROM memory

¹ The different board variations are equipped with different memory devices

Cyclone 10 LP RefKit User Guide

Configuration and Debug

• On-board Arrow USB Programmer2 (micro-USB type B connector) – JTAG mode

Interfaces

- $2 \times 10/100$ Mbps Ethernet PHYs with RJ45 connectors
- 8-Channel, 12-bits configurable ADC/DAC

Connectors

- 6× PMOD Headers
- Arduino Uno R3 compatible Header
- VGA with 15-pin high density D-Sub connector
- I2C Grove connector
- Optional SMA connectors for preferred frequency

Buttons and Indicators

- 4-Digit 7-Segment LED Display
- 7× Buttons
- 13× user LEDs
- 2× board status LEDs

Power

- Recommended external supply voltage range: +5.0 V (nominal)
- Recommended external supply current:
- Recommended I/O signal voltage range: 0 to +3.3 V

2.2 Hardware variations

Multiple board configurations are available with Cyclone 10 LP RefKit have different equipment. This user guide covers REV02 hardware revision with 8C and 8CA featured boards.

3 A

These two boards are the same with the exception that different memory devices have been mounted.

Ordering Code	SDRAM	SDRAM feature	QSPI Flash	QSPI Flash feature
TEI0009-02-055-8C	IS42S16400J-7BL	64Mbit up to	IS25LP064A-JBLE	64Mbit up to
		143MHz		133MHz
TEI0009-02-055-8CA	IS42S16160J-7BL	256Mbit up to	IS25LP128F-JBLE	128Mbit up to
		143MHz		166MHz

2.3 Block Diagram

Figure 3 represents the block diagram of the board. All the connections are established through the Cyclone 10 LP FPGA device to provide maximum flexibility for users. Users can configure the FPGA to implement any system design.



Figure 3 - Cyclone 10 LP RefKit Block Diagram



Chapter 3 - Connections and Peripherals of the Cyclone 10 LP RefKit Board

3.1 Board Status Elements

In addition to the 13 user LEDs that the FPGA can control, there are 2 additional board-specific status LEDs that can indicate the status of the board.



Figure 4 – Position of Indication LEDs

Board Reference	LED Name	Description
D1	3.3V	On when 3.3V power is active
D10	D10 CONF_DONE On when configuration data was loaded to Cyclo	
		device without error

3.2 Clock Circuitry

All the external clocks of the system can be seen in Figure 5. There are two default clocks which are 12MHz and 25MHz. Both clock signals are connected and driving the FPGA's user logic and other interfaces (Arrow USB Programmer2 and Ethernet). There are optional slots for other clocks that you can either add another preferred clock source to the FPGA (CLK_IN_SMA) or generate an FPGA-controlled clock (CLK_OUT_SMA). All clock signals are connected to the internal PLLs of the FPGA.

For more information on clocks and PLLs of the Cyclone 10 LP, please refer to this document.







Figure 5 – Cyclone 10 LP RefKit Clock Tree

Board Reference	FPGA Pin No.	Pin Func.	Description	I/O Std
CLK12M	PIN_G21	Input	12MHz clock input	3.3 V
CLK_25M	PIN_AA12	Input	25MHz clock input	3.3 V
CLK_IN_SMA	PIN_B11	Input	Optional clock input	3.3 V
CLK_OUT_SMA	PIN_E5	Output	Optional clock output	3.3 V

3.3 Peripherals Connected to the FPGA

3.3.1 Communication and Configuration

The C10LP RefKit board uses a single chip to perform configuration of the device and communication over USB.

3.3.1.1 JTAG Chain Configuration

There are two types of configuration methods supported by C10LP RefKit:

- JTAG Configuration: configuration using JTAG ports. JTAG configuration scheme allows you to directly configure the device core through JTAG pins (TDI, TDO, TMS and TCK pins). The Quartus Prime software automatically generates a .sof that can be downloaded to the Cyclone 10 LP with a download cable through the Quartus Prime Programmer.
- Configuration from EPCQ-A flash: configuration using external flash. Before configuration, you need to program the configuration data .jic into the configuration flash memory (EPCQ-A) which provides non-volatile storage for the bit stream. The information is retained within



EPCQ-A even if the C10LP RefKit is turned off. When the board is powered on, the configuration data in the EPCQ-A is automatically loaded into the Cyclone 10 LP FPGA.



Figure	6 – JTAG	Connections
--------	----------	-------------

Board Reference	FPGA Pin No.	Pin Func.	Description	I/O Std
ТСК	PIN_L2	Input	Test Interface Clock	3.3 V
TDO	PIN_L4	Output	Test Data Out	3.3 V
TDI	PIN_L5	Input	Test Data In	3.3 V
TMS	PIN_L1	Input	Test Mode Select	3.3 V

For detailed information about how to configure the Cyclone 10 LP, please refer to Chapter 6.

3.3.1.2 USB Communication

The FTDI chip converts signals from USB 2.0 to a variety of standard serial and parallel interfaces. Channel A of FTDI chip is used in MPPSE mode for JTAG. Channel B is routed to FPGA and is usable for other standard interfaces.



Figure 7 – FTDI Connections

Cyclone 10 LP RefKit User Guide

WUW

Board Reference	FPGA Pin No.	Pin Func.	Description	I/O Std
BDBUS0	PIN_C20	Bidir	D[0] of bidirectional data bus	3.3 V
BDBUS1	PIN_B21	Bidir	D[1] of bidirectional data bus	3.3 V
BDBUS2	PIN_B22	Bidir	D[2] of bidirectional data bus	3.3 V
BDBUS3	PIN_C21	Bidir	D[3] of bidirectional data bus	3.3 V
BDBUS4	PIN_C22	Bidir	D[4] of bidirectional data bus	3.3 V
BDBUS5	PIN_D21	Bidir	D[5] of bidirectional data bus	3.3 V
BDBUS6	PIN_D22	Bidir	D[6] of bidirectional data bus	3.3 V
BDBUS7	PIN_E21	Bidir	D[7] of bidirectional data bus	3.3 V
BCBUS0	PIN_E22	Bidir	D[0] of bidirectional data bus	3.3 V
BCBUS1	PIN_F21	Bidir	D[1] of bidirectional data bus	3.3 V
BCBUS2	PIN_F22	Bidir	D[2] of bidirectional data bus	3.3 V
BCBUS3	PIN_H21	Bidir	D[3] of bidirectional data bus	3.3 V
BCBUS4	PIN_H22	Bidir	D[4] of bidirectional data bus	3.3 V
BCBUS5	PIN_J21	Bidir	D[5] of bidirectional data bus	3.3 V
BCBUS6	PIN_J21	Bidir	D[6] of bidirectional data bus	3.3 V
BCBUS7	PIN_J19	Bidir	D[7] of bidirectional data bus	3.3 V

3.3.2 Fast Ethernet

The board has two independent 10/100Mbps Ethernet ports with RJ-45 connectors. For the physical layer, the Microchip KSZ8081 Ethernet PHY is used, which is suitable for general applications.

The MAC-to-PHY interface is configured to a MII interface connections with MDIO interface as management.



Figure 8 -	- MAC-to-PHY	connection
------------	--------------	------------

Board Reference	FPGA Pin No.	Pin Func.	Description	I/O Std
ETH1_MDIO	PIN_AA21	Bidir	Management Interface Data	3.3 V
ETH1_MDC	PIN_AA22	Output	Management Interface Clock	3.3 V
ETH1_COL	PIN_T19	Bidir	MII Collision Detect	3.3 V
ETH1_CRS	PIN_R20	Bidir	MII Carrier Sense	3.3 V



Board Reference	FPGA Pin No.	Pin Func.	Description	I/O Std
ETH1_RXDV	PIN_W21	Bidir	MII Receive Data Valid	3.3 V
ETH1_RST	PIN_R19	Output	Chip Reset	3.3 V
ETH1_INTRP	PIN_U22	Bidir	Interrupt	3.3 V
ETH1_RXC	PIN_V22	Bidir	MII Receive Clock	3.3 V
ETH1_RXER	PIN_V21	Bidir	MII Receive Error	3.3 V
ETH1_RXD0	PIN_W22	Bidir	MII Receive Data D[0]	3.3 V
ETH1_RXD1	PIN_W20	Bidir	MII Receive Data D[1]	3.3 V
ETH1_RXD2	PIN_Y21	Bidir	MII Receive Data D[2]	3.3 V
ETH1_RXD3	PIN_Y22	Bidir	MII Receive Data D[3]	3.3 V
ETH1_TXC	PIN_U21	Bidir	MII Transmit Clock	3.3 V
ETH1_TXEN	PIN_T18	Output	MII Transmit Enable	3.3 V
ETH1_TXD0	PIN_T17	Output	MII Transmit Data D[0]	3.3 V
ETH1_TXD1	PIN_U20	Output	MII Transmit Data D[1]	3.3 V
ETH1_TXD2	PIN_U19	Output	MII Transmit Data D[2]	3.3 V
ETH1_TXD3	PIN_T20	Output	MII Transmit Data D[3]	3.3 V
ETH2_MDIO	PIN_N20	Bidir	Management Interface Data	3.3 V
ETH2_MDC	PIN_N18	Output	Management Interface Clock	3.3 V
ETH2_COL	PIN_P21	Bidir	MII Collision Detect	3.3 V
ETH2_CRS	PIN_P22	Bidir	MII Carrier Sense	3.3 V
ETH2_RXDV	PIN_R18	Bidir	MII Receive Data Valid	3.3 V
ETH2_RST	PIN_M21	Output	Chip Reset	3.3 V
ETH2_INTRP	PIN_N17	Bidir	Interrupt	3.3 V
ETH2_RXC	PIN_R17	Bidir	MII Receive Clock	3.3 V
ETH2_RXER	PIN_P17	Bidir	MII Receive Error	3.3 V
ETH2_RXD0	PIN_M20	Bidir	MII Receive Data D[0]	3.3 V
ETH2_RXD1	PIN_M19	Bidir	MII Receive Data D[1]	3.3 V
ETH2_RXD2	PIN_M16	Bidir	MII Receive Data D[2]	3.3 V
ETH2_RXD3	PIN_N19	Bidir	MII Receive Data D[3]	3.3 V
ETH2_TXC	PIN_N16	Bidir	MII Transmit Clock	3.3 V
ETH2_TXEN	PIN_R22	Output	MII Transmit Enable	3.3 V
ETH2_TXD0	PIN_R21	Output	MII Transmit Data D[0]	3.3 V
ETH2_TXD1	PIN_N21	Output	MII Transmit Data D[1]	3.3 V
ETH2_TXD2	PIN_M22	Output	MII Transmit Data D[2]	3.3 V
ETH2_TXD3	PIN_N22	Output	MII Transmit Data D[3]	3.3 V

3.3.3 Serial Configuration Flash Memory

The C10LP RefKit board is integrated with a 16MBit of serial flash memory that can be used for user data and programming non-volatile storage. The configuration bitstream is downloaded into the serial configuration device which automatically loads the configuration data into the Cyclone 10 LP when the board is powered on. Device memory capacity not consumed storing configuration data can be used as general-purpose non-volatile memory, which is perfect for program and data storage. Several interfaces available with Nios II embedded processors allow you to access the serial configuration device as a memory module connected to your embedded system.





Figure 9 – Configuration Flash Connections

Board Reference	FPGA Pin No.	Pin Func.	Description	I/O Std
AS_DATA	PIN_K1	Input	Data In	3.3 V
AS_DCLK	PIN_K2	Output	Clock	3.3 V
AS_NCS	PIN_E2	Output	Chip Select	3.3 V
AS_ASDO	PIN_D1	Output	Data Out	3.3 V

3.3.4 HyperRAM

A 64Mbit Pseudo Static Random Access Memory using a self-refresh DRAM array is integrated on C10LP RefKit. The Cyclone 10 LP connects to this memory via a very low signal count interface, called HyperBus.



Figure 10 – HyperRAM Connections

Board Reference	FPGA Pin No.	Pin Func.	Description	I/O Std
HR_CLK	PIN_T16	Output	Single Ended Clock	3.3 V
HR_RW	PIN_U13	Bidir	Read Write Data Strobe	3.3 V
HR_CS	PIN_V13	Output	Chip Select	3.3 V
HR_RESET	PIN_U12	Output	Hardware Reset	3.3 V
HR_D0	PIN_T15	Bidir	Data [0]	3.3 V
HR_D1	PIN_W17	Bidir	Data [1]	3.3 V
HR_D2	PIN_U14	Bidir	Data [2]	3.3 V
HR_D3	PIN_R15	Bidir	Data [3]	3.3 V
HR_D4	PIN_R14	Bidir	Data [4]	3.3 V
HR_D5	PIN_V16	Bidir	Data [5]	3.3 V
HR_D6	PIN_U16	Bidir	Data [6]	3.3 V
HR_D7	PIN_U17	Bidir	Data [7]	3.3 V

Cyclone 10 LP RefKit User Guide

WDW

3.3.5 SDRAM Memory

The C10LP RefKit board supports single-chip SDRAM with up to 256Mbit density² which can operate up to 143 MHz clock frequency. Below are the connections and pinning of the SDRAM used in the C10LP RefKit.



Figure	11-	SDRAM	Connections
--------	-----	-------	-------------

Board Reference	FPGA Pin No.	Pin Func.	Description	I/O Std
A0	PIN_V5	Output	SDRAM Address [0]	3.3 V
A1	PIN_Y3	Output	SDRAM Address [1]	3.3 V
A2	PIN_W6	Output	SDRAM Address [2]	3.3 V
A3	PIN_Y4	Output	SDRAM Address [3]	3.3 V
A4	PIN_AB5	Output	SDRAM Address [4]	3.3 V
A5	PIN_AB6	Output	SDRAM Address [5]	3.3 V
A6	PIN_AA6	Output	SDRAM Address [6]	3.3 V
A7	PIN_AA7	Output	SDRAM Address [7]	3.3 V
A8	PIN_AB8	Output	SDRAM Address [8]	3.3 V
A9	PIN_AA5	Output	SDRAM Address [9]	3.3 V
A10	PIN_V6	Output	SDRAM Address [10]	3.3 V
A11	PIN_AA8	Output	SDRAM Address [11]	3.3 V
A12	PIN_AB8	Output	SDRAM Address [12]	3.3 V
A13	PIN_AB9	Output	SDRAM Address [13]	3.3 V
BA0	PIN_Y6	Output	SDRAM Bank Address [0]	3.3 V
BA1	PIN_V7	Output	SDRAM Bank Address [1]	3.3 V
RAS	PIN_V8	Output	SDRAM Row Address Strobe	3.3 V
CAS	PIN_Y7	Output	SDRAM Column Address Strobe	3.3 V
WE	PIN_W8	Output	SDRAM Write Enable	3.3 V
CS	PIN_W7	Output	SDRAM Chip Select	3.3 V
CLK	PIN_AA3	Output	SDRAM Input Clock	3.3 V
СКЕ	PIN_AA4	Output	SDRAM Clock Enable	3.3 V
DQ0	PIN_AB16	Bidir	SDRAM Data [0]	3.3 V
DQ1	PIN_Y17	Bidir	SDRAM Data [1]	3.3 V
DQ2	PIN_AA16	Bidir	SDRAM Data [2]	3.3 V
DQ3	PIN_AA19	Bidir	SDRAM Data [3]	3.3 V
DQ4	PIN_AB18	Bidir	SDRAM Data [4]	3.3 V
DQ5	PIN_AA20	Bidir	SDRAM Data [5]	3.3 V
DQ6	PIN_AB19	Bidir	SDRAM Data [6]	3.3 V
DQ7	PIN_AB20	Bidir	SDRAM Data [7]	3.3 V

² The size of the mounted SDRAM depends on the board variation. For detailed information, please refer to Chapter 2.2.

wow

Board Reference	FPGA Pin No.	Pin Func.	Description	I/O Std
DQ8	PIN_Y13	Bidir	SDRAM Data [8]	3.3 V
DQ9	PIN_Y15	Bidir	SDRAM Data [9]	3.3 V
DQ10	PIN_AA13	Bidir	SDRAM Data [10]	3.3 V
DQ11	PIN_AB15	Bidir	SDRAM Data [11]	3.3 V
DQ12	PIN_AB13	Bidir	SDRAM Data [12]	3.3 V
DQ13	PIN_AA15	Bidir	SDRAM Data [13]	3.3 V
DQ14	PIN_AA14	Bidir	SDRAM Data [14]	3.3 V
DQ15	PIN_AB14	Bidir	SDRAM Data [15]	3.3 V
DQM0	PIN_Y14	Output	SDRAM Lower Data Mask	3.3 V
DQM1	PIN_W13	Output	SDRAM Upper Data Mask	3.3 V

3.3.6 QSPI Flash Memory

There is a non-volatile, QSPI Flash memory with up to 128Mbit density³ which can operate on up to 166MHz on the board. It can be used to store larger size user data or software for Nios II embedded processors.



Figure 12 – QSPI Flash Connections

Board Reference	FPGA Pin No.	Pin Func.	Description	I/O Std
F_CS	PIN_F15	Output	Chip Enable	3.3 V
F_CLK	PIN_F16	Output	Serial Data Clock	3.3 V
F_100	PIN_G16	Bidir	Serial Data [0]	3.3 V
F_I01	PIN_D15	Bidir	Serial Data [1]	3.3 V
F_102	PIN_E16	Bidir	Serial Data [2]	3.3 V
F_IO3	PIN_E15	Bidir	Serial Data [3]	3.3 V

3.3.7 EEPROMs

The C10LP RefKit board has 2 pieces 2Kb serial EEPROMs that can be used for MAC address configuration. The EEPROMs are pre-programmed with a globally unique EUI-48 node address.

³ The size of the mounted QSPI Flash depends on the board variation. For detailed information, please refer to Chapter 2.2.



Figure 13 – EEPROM Connections

Board Reference	FPGA Pin No.	Pin Func.	Description	I/O Std
I2C_SDA	PIN_F17	Bidir	Serial Data Line	3.3 V
I2C_SCL	PIN_D20	Output	Serial Clock Line	3.3 V

3.3.8 ADC/DAC

The C10LP RefKit is equipped with an 8-channel, 12-bit, configurable analog-to-digital, digital-toanalog converter. There are 2 dedicated through-hole connection points on the board for 2 analog channels, while the remaining 6 channels are directly connected to the J4 header of the Arduino interface.



Figure 14 – ADC/DAC Connections

Board Reference	FPGA Pin No.	Pin Func.	Description	I/O Std
ADDA_RSTN	PIN_V4	Output	Reset	3.3 V
ADDA_SYNC	PIN_R5	Output	Synchronization	3.3 V
MCLK	PIN_T5	Output	Serial Clock Input	3.3 V
MOSI	PIN_T4	Output	Master Output Slave Input	3.3 V
MISO	PIN_R6	Input	Master Input Slave Output	3.3 V

Cyclone 10 LP RefKit User Guide

WUW

wow

Board Reference	Connector	Description
AREF	J1/8	External Reference Voltage
AIN0	J4 / 1	Analog I/O Channel 0
AIN1	J4 / 2	Analog I/O Channel 1
AIN2	J4 / 3	Analog I/O Channel 2
AIN3	J4 / 4	Analog I/O Channel 3
AIN4	J4 / 5	Analog I/O Channel 4
AIN5	J4 / 6	Analog I/O Channel 5
AIN6	TP1	Analog I/O Channel 6
AIN7	TP2	Analog I/O Channel 7

Note: The FPGA is also directly connected to the J4 connector. If AIN5..0 are used as analog input/output, make sure that the belonging FPGA I/Os are unused and configured as input tri-stated!

Note: Do not drive a voltage greater than 3.3V to the analog I/Os. Voltages greater than 3.3V can cause irreversible damage to the FPGA!

3.3.9 I2C Grove Connector

There is a Grove connector which allows external, I2C compatible devices to be connected to the C10LP RefKit board.



Figure 15 – I2C Grove Connector

Board	FPGA Pin	Grove Pin	Pin	Description	I/O Std
Reference	No.		Func.		
I2C_SCL	PIN_D20	1	Output	Serial Clock Line	3.3 V
I2C_SDA	PIN_F17	2	Bidir	Serial Data Line	3.3 V
3.3V	-	3	PWR	3.3V power to the connector	-
GND	-	4	PWR	Ground output to the connector	-

Note: The EEPROMs are also connected to this I2C bus, 0x50h and 0x51h addresses are reserved for these EEPROMs.



3.3.10 Arduino Header

The C10LP RefKit board offers connectivity to classic Arduino compatible shields that could also alternatively be used as GPIOs. The Arduino connectors offer up to 23 digital I/Os which comes with four independent headers.



Figure 16 - Arduino Header Connections

Board	FPGA Pin	Arduino	Pin	Description	I/O Std
Reference	No.	Header	Func.		
D0_RXD	PIN_J7	J2 / 1	Bidir	Digital I/O [0] or Serial In	3.3 V
D1_TXD	PIN_H7	J2 / 2	Bidir	Digital I/O [1] or Serial Out	3.3 V
D2	PIN_H6	J2 / 3	Bidir	Digital I/O [2]	3.3 V
D3	PIN_J4	J2 / 4	Bidir	Digital I/O [3]	3.3 V
D4	PIN_E4	J2 / 5	Bidir	Digital I/O [4]	3.3 V
D5	PIN_E3	J2 / 6	Bidir	Digital I/O [5]	3.3 V
D6	PIN_D2	J2 / 7	Bidir	Digital I/O [6]	3.3 V
D7	PIN_F2	J2 / 8	Bidir	Digital I/O [7]	3.3 V
D8	PIN_B2	J1 / 1	Bidir	Digital I/O [8]	3.3 V
D9	PIN_B1	J1/2	Bidir	Digital I/O [9]	3.3 V
D10	PIN_G3	J1/3	Bidir	Digital I/O [10]	3.3 V
D11	PIN_H2	J1/4	Bidir	Digital I/O [11]	3.3V
D12	PIN_F1	J1/5	Bidir	Digital I/O [12]	3.3 V
D13	PIN_J4	J1/6	Bidir	Digital I/O [13]	3.3 V
GND	-	J1 / 7	PWR	Ground output to the connector	-
AREF	-	J1/8	PWR	Input reference voltage for ADC/DAC	-

Cyclone 10 LP RefKit User Guide

WDW

Board	FPGA Pin	Arduino	Pin	Description	I/O Std
Reference	No.	Header	Func.		
D14_SDA	PIN_E1	J1/9	Bidir	Digital I/O [14] or Serial Data Line	3.3 V
D15_SCL	PIN_H3	J1 / 10	Bidir	Digital I/O [15] or Serial Clock Line	3.3 V
n.c.	-	J3 / 1	-	Not connected	-
3.3V	-	J3 / 2	PWR	3.3V power to the connector	-
EXT_RST	PIN_P7	J3 / 3	Bidir	Reset signal of the FPGA	3.3 V
3.3V	-	J3 / 4	PWR	3.3V power to the connector	-
5V	-	J3 / 5	PWR	5V power to the connector	-
GND	-	J3 / 6	PWR	Ground output to the connector	-
GND	-	J3 / 7	PWR	Ground output to the connector	-
n.c.	-	J3 / 8	-	Not connected	-
AIN0	PIN_J6	J4 / 1	Bidir	GPIO [0]	3.3 V
AIN1	PIN_H1	J4 / 2	Bidir	GPIO [1]	3.3 V
AIN2	PIN_J2	J4 / 3	Bidir	GPIO [2]	3.3 V
AIN3	PIN_J1	J4 / 4	Bidir	GPIO [3]	3.3 V
AIN4	PIN_J3	J4 / 5	Bidir	GPIO [4]	3.3 V
AIN5	PIN_J5	J4 / 6	Bidir	GPIO [5]	3.3 V

Note: The ADC/DAC is also directly connected to the J4 connector. If AIN5..0 are used as digital I/Os of the FPGA, make sure that the ADC/DAC does not drive these wires!

3.3.11 PMOD Connectors

The C10LP RefKit board offers connectivity to PMOD compatible connectors, making it possible to add a big variety of sensors or ICs to the system. The board has 6 PMOD connectors that can be configured to 2×6 pins or 1×12 pins





Cyclone 10 LP RefKit User Guide

WUW

www.arrow.com February 2022

Board	FPGA Pin	PMOD	Pin	Description	I/O Std
Reference	No.	Header	Func.		
P1_IO1	PIN_V3	P1/1	Bidir	PMOD I/O [1] of P1	3.3 V
P1_IO2	PIN_P6	P1/2	Bidir	PMOD I/O [2] of P1	3.3 V
P1_IO3	PIN_P4	P1/3	Bidir	PMOD I/O [3] of P1	3.3 V
P1_IO4	PIN_N5	P1/4	Bidir	PMOD I/O [4] of P1	3.3 V
P1_IO5	PIN_N7	P1/7	Bidir	PMOD I/O [5] of P1	3.3 V
P1_IO6	PIN_R4	P1/8	Bidir	PMOD I/O [6] of P1	3.3 V
P1_IO7	PIN_P5	P1/9	Bidir	PMOD I/O [7] of P1	3.3 V
P1_IO8	PIN_N7	P1/10	Bidir	PMOD I/O [8] of P1	3.3 V
P2_IO1	PIN_N1	P2/1	Bidir	PMOD I/O [1] of P2	3.3 V
P2_IO2	PIN_M2	P2/2	Bidir	PMOD I/O [2] of P2	3.3 V
P2_IO3	PIN_M4	P2/3	Bidir	PMOD I/O [3] of P2	3.3 V
P2_IO4	PIN_L6	P2/4	Bidir	PMOD I/O [4] of P2	3.3 V
P2_IO5	PIN_N2	P2/7	Bidir	PMOD I/O [5] of P2	3.3 V
P2_IO6	PIN_M1	P2/8	Bidir	PMOD I/O [6] of P2	3.3 V
P2_IO7	PIN_M3	P2/9	Bidir	PMOD I/O [7] of P2	3.3 V
P2_IO8	PIN_M6	P2/10	Bidir	PMOD I/O [8] of P2	3.3 V
P3_IO1	PIN_A3	P3/1	Bidir	PMOD I/O [1] of P3	3.3 V
P3_IO2	PIN_B3	P3 / 2	Bidir	PMOD I/O [2] of P3	3.3 V
P3_IO3	PIN_A4	P3/3	Bidir	PMOD I/O [3] of P3	3.3 V
P3_IO4	PIN_B4	P3/4	Bidir	PMOD I/O [4] of P3	3.3 V
P3_IO5	PIN_B6	P3 / 7	Bidir	PMOD I/O [5] of P3	3.3 V
P3_IO6	PIN_A6	P3/8	Bidir	PMOD I/O [6] of P3	3.3 V
P3_IO7	PIN_C6	P3/9	Bidir	PMOD I/O [7] of P3	3.3 V
P3_IO8	PIN_A5	P3/10	Bidir	PMOD I/O [8] of P3	3.3 V
P4_IO1	PIN_A7	P4/1	Bidir	PMOD I/O [1] of P4	3.3 V
P4_IO2	PIN_B7	P4 / 2	Bidir	PMOD I/O [2] of P4	3.3 V
P4_IO3	PIN_A8	P4/3	Bidir	PMOD I/O [3] of P4	3.3 V
P4_IO4	PIN_B8	P4 / 4	Bidir	PMOD I/O [4] of P4	3.3 V
P4_IO5	PIN_B10	P4 / 7	Bidir	PMOD I/O [5] of P4	3.3 V
P4_IO6	PIN_A10	P4/8	Bidir	PMOD I/O [6] of P4	3.3 V
P4_107	PIN_B9	P4/9	Bidir	PMOD I/O [7] of P4	3.3 V
P4_I08	PIN_A9	P4 / 10	Bidir	PMOD I/O [8] of P4	3.3 V
P5_IO1	PIN_A14	P5/1	Bidir	PMOD I/O [1] of P5	3.3 V
P5_IO2	PIN_B15	P5/2	Bidir	PMOD I/O [2] of P5	3.3 V
P5_103	PIN_A15	P5/3	Bidir	PMOD I/O [3] of P5	3.3 V
P5_104	PIN_B16	P5/4	Bidir	PMOD I/O [4] of P5	3.3 V
P5_105	PIN_B14	P5/7	Bidir	PMOD I/O [5] of P5	3.3 V
P5_106	PIN_A13	P5/8	Bidir		3.3 V
P5_107	PIN_B13	P5/9	Bidir		3.3 V
P5_108	PIN_A16	P5/10	BIGIL		3.3 V
	PIN_BTA		RIGIL		3.3 V
	PIN_A19	P0/2	BIUII		5.5 V
		PG / 3	Bidir		3.5 V 3 2 V
P6 105		D6/7	Bidir		22V
	LUN_ATO	FU//	Biuli		5.5 V

Cyclone 10 LP RefKit User Guide

WDVN



Board	FPGA Pin	PMOD	Pin	Description	I/O Std
Reference	No.	Header	Func.		
P6_IO6	PIN_B18	P6/8	Bidir	PMOD I/O [6] of P6	3.3 V
P6_IO7	PIN_A17	P6/9	Bidir	PMOD I/O [7] of P6	3.3 V
P6_IO8	PIN_B17	P6 / 10	Bidir	PMOD I/O [8] of P6	3.3 V
GND	-	5, 11 ⁴	PWR	Ground	-
3.3V	-	6, 12 ⁴	PWR	3.3 V Power to PMODs	-

3.3.12 VGA

The C10LP RefKit provides VGA connectivity that allows users to display content on a monitor. The VGA uses a 4-bit resistor-network DAC which supports up to 640×480 resolutions at a 60Hz refresh rate with 4096 colors.



Figure 18 – VGA Connections

Board Reference	FPGA Pin No.	Pin Func.	Description	I/O Std
VGA_R0	PIN_R3	Output	VGA Red [0]	3.3 V
VGA_R1	PIN_V2	Output	VGA Red [1]	3.3 V
VGA_R2	PIN_W2	Output	VGA Red [2]	3.3 V
VGA_R3	PIN_Y2	Output	VGA Red [3]	3.3 V
VGA_G0	PIN_Y1	Output	VGA Green [0]	3.3 V
VGA_G1	PIN_W1	Output	VGA Green [1]	3.3 V
VGA_G2	PIN_V1	Output	VGA Green [2]	3.3 V

⁴ Pins 5, 6, 11, and 12 applies to all, P1..6 PMOD connectors.

wow

Board Reference	FPGA Pin No.	Pin Func.	Description	I/O Std
VGA_G3	PIN_U2	Output	VGA Green [3]	3.3 V
VGA_B0	PIN_U1	Output	VGA Blue [0]	3.3 V
VGA_B1	PIN_R2	Output	VGA Blue [1]	3.3 V
VGA_B2	PIN_R1	Output	VGA Blue [2]	3.3 V
VGA_B3	PIN_P2	Output	VGA Blue [3]	3.3 V
VGA_VS	PIN_P3	Output	Vertical Synchronization	3.3 V
VGA_HS	PIN_P1	Output	Horizontal Synchronization	3.3 V

The 4, 9, 11, 12, and 15 pins of the J11 connector are not connected.

3.3.13 LEDs

There is a total of 13 red user-controllable LEDs connected to the FPGA in two types of splits. 8 LEDs are arranged in a traditional row, and an additional 5 LEDs are arranged in a joystick shape according to the location of the pushbuttons. Each LED is driven directly and individually by the Cyclone 10 LP FPGA, driving its associated pin to a high logic level for on or low logic level for off.



Figure 19 – LED Connections

Board Reference	FPGA Pin No.	Pin Func.	I/O Std
LED1	PIN_AB10	Output	3.3 V
LED2	PIN_AA10	Output	3.3 V
LED3	PIN_AA9	Output	3.3 V
LED4	PIN_Y10	Output	3.3 V
LED5	PIN_W10	Output	3.3 V
LED6	PIN_U9	Output	3.3 V

Cyclone 10 LP RefKit User Guide

WUVN

Board Reference	FPGA Pin No.	Pin Func.	I/O Std
LED7	PIN_U8	Output	3.3 V
LED8	PIN_U7	Output	3.3 V
LED_PB1	PIN_C18	Output	3.3 V
LED_PB2	PIN_D19	Output	3.3 V
LED_PB3	PIN_C19	Output	3.3 V
LED_PB4	PIN_C17	Output	3.3 V
LED_PB5	PIN_D18	Output	3.3 V

3.3.14 Push Buttons

The board has seven push buttons connected to the FPGA that allow users to interact with the Cyclone 10 LP FPGA device. 5 of them are placed in a joystick shape for better usability. Push buttons drive their associated pins low logic level when pressed and high logic level when released.



Figure 20 – Button Connections

Board Reference	FPGA Pin No.	Pin Func.	Description	I/O Std
RESET	PIN_K5	Input	nCONFIG	3.3 V
USER_BTN1	PIN_U10	Input	User button	3.3 V
USER_BTN2	PIN_U11	Input	User button	3.3 V
USER_BTN3	PIN_V11	Input	User button	3.3 V
USER_BTN4	PIN_T10	Input	User button	3.3 V
USER_BTN5	PIN_T11	Input	User button	3.3 V
RST_GPIO	PIN_V15	Input	User button	3.3 V

Cyclone 10 LP RefKit User Guide

WDW

3.3.15 7-segment LED Display

The C10LP RefKit board has a Quadruple seven-segment LED display to display numbers. This display has a multiplex common anode structure to reduce the number of control signals.

The connections and the structure of the display are shown in the figures below.



Figure 21 – 4-digit 7-segment Display Connections

Board Reference	FPGA Pin No.	LED Display	Pin Func.	Description	I/O Std
SEG_AN	PIN_H20	AN	Output	Common Anode for L1, L2 and L3	3.3 V
SEG_AN1	PIN_F20	AN1	Output	Common Anode for Digit 1	3.3 V
SEG_AN2	PIN_H16	AN2	Output	Common Anode for Digit 2	3.3 V
SEG_AN3	PIN_J20	AN3	Output	Common Anode for Digit 3	3.3 V
SEG_AN4	PIN_K17	AN4	Output	Common Anode for Digit 4	3.3 V
SEG_CA	PIN_J17	A	Output	Segment A or L1	3.3 V
SEG_CB	PIN_H17	В	Output	Segment B or L2	3.3 V
SEG_CC	PIN_G17	С	Output	Segment C or L3	3.3 V
SEG_CD	PIN_G18	D	Output	Segment D	3.3 V
SEG_CE	PIN_K18	E	Output	Segment E	3.3 V
SEG_CF	PIN_F19	F	Output	Segment F	3.3 V
SEG_CG	PIN_J18	G	Output	Segment G	3.3 V
SEG_CDP	PIN_H19	DP	Output	Decimal Point	3.3 V



wow



Figure 22 – Quadruple Seven-segment LED Display's Internal Circuit Diagram

3.3.16 Power Tree

The Cyclone 10 LP RefKit is powered by circuit Enpirion's buck regulator which provides high efficiency on a small layout. The board is powered through a 2.0mm DC Jack connector. All devices are powered by a 3.3V voltage line and the 5V and 3.3V lines are fed back to the Arduino header to power that connection if needed. The Cyclone 10 LP FPGA is powered by 2 Enpirion devices.







Figure 23 – Power Tree Connections

Cyclone 10 LP RefKit User Guide

WDW

Chapter 4 - Software and Driver Installation

Firstly, it is required to create your <u>Basic Intel Account</u> if you don't own one already. It is required to download the software. Below are guides for installing the software and drivers for Windows operating systems.

4.1 Installing Quartus Prime Software

- 4.1.1 Go to the Intel Download Center: Link.
- 4.1.2 Select **Windows** as the operating system (highlighted in red).
- 4.1.3 Select Release **21.1**, or your preferred version (highlighted in red).
- 4.1.4 Download the following files from the "Individual Files" tab (highlighted in yellow):
 - Quartus Prime Lite Edition (Free)
 - Questa Intel FPGA Edition (includes Starter Edition)
 - Cyclone 10 LP device support

se date: March, 2021 t Release: v21.1	Intel Quartus Prime Design Software
t edition: Lite v t release: 21.1 v	
ating System 👔 💿 🎊 Windows 🔿 👌 Linux	
ombined Files Individual Files Additional Software	
ownload and install instructions: <u>More</u>	
<u>lead Intel FPGA Software v21.1 Installation FAQ</u> Duick Start Guide	
Quartus Prime Lite Edition (Free)	
Quartus Prime (includes Nios II EDS) Size: 1.6 GB MD5: 4C3E00771CFE9D6DA618B2D79D5. ** Nios II EDS on Windows requires Ubuntu 18.04 LTS on Window: requires a manual installation. ** Nios II EDS requires you to install an Eclipse IDE manually.	4A5F6 Subsystem for Linux (WSL), which
Questa - Intel FPGA Edition(includes Starter Edition) Size: 961.6 MB MD5: CAB368F5A03D78F842424CB280 ** Starter edition requires free license that can be obtained <u>here</u>	DF45BB3
Devices You must install device support for at least one de	
software.	vice family to use the Quartus Prime
software. Arria II device support Size: 499.1 MB MD5: A2D16C109493C37BB5D10BCD3	A54F58C
software. Arria II device support Size: 499.1 MB MD5: A2D16C109493C37BB5D10BCD3 Cyclone IV device support Size: 466.0 MB MD5: E37015353737752218908311E2	A54F58C
software. Arria II device support Size: 499.1 MB MD5: A2D16C109493C37BB5D10BCD3 Cyclone IV device support Size: 466.0 MB MD5: E37015353737752218908311E2 Cyclone 10 LP device support Size: 265.7 MB MD5: 2061E55E14FA6419376BEA1FF8	A54F58C E915F5 BBA3C1
software. Arria II device support Size: 499.1 MB MD5: A2D16C109493C37BB5D10BCD3 Cyclone IV device support Size: 466.0 MB MD5: E37015353737752218908311E2 Cyclone 10 LP device support Size: 265.7 MB MD5: 2061E55E14FA6419376BEA1FF8 Cyclone V device support Size: 1.3 GB MD5: 5D3DF782AC7F408F8166E58AF030	A54F58C O BBA3C1 O FF9B O
software. Arria II device support Size: 499.1 MB MD5: A2D16C109493C37BB5D10BCD3 Cyclone IV device support Size: 466.0 MB MD5: E37015353737752218908311E2 Cyclone 10 LP device support Size: 265.7 MB MD5: 2061E55E14FA6419376BEA1FF8 Cyclone V device support Size: 1.3 GB MD5: 5D3DF782AC7F408F8166E58AF030 MAX II, MAX V device support Size: 11.4 MB MD5: 8657DE76CA949C8B435146F79BU	A54F58C O BBA3C1 O S39EF9 O

4.1.5 Click on \bigcirc button to begin the download and save them in the same folder.

Cyclone 10 LP RefKit User Guide

WUW

- 4.1.6 After the download is finished, run the Quartus Prime installer.
- 4.1.7 When prompted to select the components, the installer will automatically detect the Cyclone 10 LP device support and Questa packages when they are in the same folder. Make sure these components are selected:



4.1.8 Finish the installation of the Quartus Lite and proceeded to the next section to install Arrow USB Programmer2 to be able to connect to the C10LP RefKit board.

4.2 Installing Arrow USB Programmer2

The Cyclone 10 LP RefKit board uses version 2 of the Arrow USB Programmer2 programming solution, that is an FTDI FT2232H Hi-Speed USB controller plus a programmer DLL. Since this FTDI USB controller is a very common standard device, usually no specific drivers are needed to make the C10LP RefKit work.

4.2.1 Download the appropriate version⁵ of Arrow USB Programmer2 for C10LP RefKit from Trenz Electronic Wiki page or alternatively this direct <u>link.</u>

⁵Modules produced after June 2020 are no longer compatible with older drivers. Please install driver version 2.4 or newer.

Home Products Download	Company Services Jobs News Distributors
Download > Trenz_Electronic >	Software > Drivers > Arrow_USB_Programmer
Digilent	Online Documentation:
OHO-Elektronik	Notes:
SunDance	• If you did not find the necessary documents, please send a request mail to Trenz Electronic Support (support[at]trenz-electronic.de).
Trenz_Electronic	Arrow_USB_Programmer_2.0 - Arrow USB Programmer 2.0 Libraries Arrow_USB_Programmer_2.1 - Arrow USB Programmer 2.1 Libraries
-corporate	Arrow_USB_Programmer_2.2 - Arrow USB Programmer 2.2 Libraries
-obsolete_products	Arrow_USB_Programmer_2.3 - Arrow USB Programmer 2.3 Libraries
Accessories	Arrow_USB_Programmer_2.4 - Arrow USB Programmer 2.4 Libraries
CPCIS_Cards	🚘 Arrow_USB_Programmer_2.5 - Arrow USB Programmer 2.5 Libraries
CRUVI	Files
Development_Boards	+ Documents (1 Files)
Digital_IO	Arrow_USB_Programmer2-Troubleshooting_Guide_for_WinOS.pdf
FMC_Cards	Size 350,35 KB / Modified 07.03.2018 - 13:59:19
JTAG_Programmer	↓ Diagnose Tool for Win OS (1 Files)
Modules_and_Module_Carriers	Arrow USB Programmer ³ -Diagnostic Program for Win OS zin
Motherboards_and_Carriers	Size 218,36 KB / Modified 15.04.2020 - 16:32:58
PCIe_Cards	
Pinout	Other Files (0 Files)

- 4.2.2 After downloading the file, run the installer to install the Arrow USB Programmer2. The setup executable installs the programmer DLL and adds some keys to the registry of the PC.
- 4.2.3 After connecting the C10LP RefKit board to the PC, two unknown devices might appear in the "Other devices" section of device manager of the PC.



Windows usually automatically finds the appropriate drivers for these devices. After some time, the "Other devices" section should be empty. Instead, two USB Serial Converters should be listed in the section "USB Serial Bus controllers":

> ኪ System devices

- Universal Serial Bus controllers
 - 🏺 Generic USB Hub
 - 🏺 Generic USB Hub
 - Intel(R) 8 Series/C220 Series USB EHCI #1 8C26
 - Intel(R) 8 Series/C220 Series USB EHCI #2 8C2D
 - Intel(R) USB 3.0 eXtensible Host Controller 1.0 (Microsoft)
 - USB Composite Device
 - 🏺 USB Root Hub
 - 🏺 USB Root Hub
 - USB Root Hub (xHCl)
 - 🏺 USB Serial Converter A
 - USB Serial Converter B

Furthermore, a USB Serial Port should be listed in the "Ports (COM & LPT)" section.





Note that the number of the port will most probably be different from the one shown here.

In case Windows does not automatically find the appropriate drivers go to <u>http://www.ftdichip.com/Drivers/D2XX.htm</u> to download the setup executable to install the required drivers.

4.3 License

Quartus Lite does not require a license, its use is completely free. However, even though Questa Starter Edition can be used free of charge, you need to generate a free license for it.

- 4.3.1 Log in to Intel FPGA Self-Service Licensing Center
- 4.3.2 Go to Sign up for Evaluation or Free Licenses tab
- 4.3.3 Select Questa*-Intel® FPGA Starter Edition SW-QUESTA option
- 4.3.4 Set the seats and accept the terms of use this license

Intel® FPGA Self-Service Licensing Center

	Product \vee # of Seats	✓ Maintenance expiration ✓ License expiration
1	Intel® Quartus® II Software SW-QUARTUS-WE-FIX 1	2023-02-10
2	Questa*-Intel® FPGA Starter Edition SW-QUESTA	2023-02-10
3	Intel® FPGA MAXPLUS2WEB	✗ 2023-02-10
4	Intel® FPGA IP PLS-WEB	A 2023-02-10
5	Intel® FPGA EVALUATION-LIC	A 2022-05-10 2022-05-11
	Intel® FPGA IP IP-NIOSVM	<i>∎</i> * 2023-02-10

4.3.5 Click on Get License button

- 4.3.6 In the pop-up window select **+New computer** under Create a New Computer
- 4.3.7 In the Create Computer window, fill in the fields with your computer details and click on Generate License.

The license file will be provided by email, or you can also download it under Intel® FPGA Self-Service Licensing Center.



Chapter 5 - New Project with Cyclone 10 LP RefKit

5.1 Creating a new Blinky Project with Cyclone 10 LP RefKit

5.1.1 Launch Quartus Prime Lite Edition from the Start Menu.



5.1.2 In the Quartus Prime tool, create a new project: File -> New Project Wizard.

The New Project Wizard walks you through the project settings, such as the name, directories, files, directories, device family and other settings. These settings can be changed later if needed.

Introduction The New Project Wizard helps you create a new project and preliminary project settings, including the following: Project name and directory Name of the top-level design entity Project files and libraries Target device family and device EDA tool settings You can change the settings for an existing project and specify additional project-wide settings with the Settings command (Assignments menu). You can use the various pages of the Settings dialog box to add functionality to the project. Pont show me this introduction again Belp Sack Next> Einish Cancel 	🕥 New Pro		×
The New Project Wizard helps you create a new project and preliminary project settings, including the following: Project name and directory Name of the top-level design entity Project files and libraries Target device family and device EDA tool settings You can change the settings for an existing project and specify additional project-wide settings with the Settings command (Assignments menu). You can use the various pages of the Settings dialog box to add functionality to the project. Don't show me this introduction again <u>Help</u> Sack Next> Einish Cancel 	Introd	uction	
Project name and directory Name of the top-level design entity Project files and libraries Target device family and device EDA tool settings Vou can change the settings for an existing project and specify additional project-wide settings with the Settings command (Assignments menu). You can use the various pages of the Settings dialog box to add functionality to the project. Don't show me this introduction again Help	The New I	Project Wizard helps you create a new project and preliminary project settings, including the following:	
Name of the top-level design entity Project files and libraries Target device family and device EDA tool settings You can change the settings for an existing project and specify additional project-wide settings with the Settings command (Assignments menu). You can use the various pages of the Settings dialog box to add functionality to the project. Don't show me this introduction again	•	Project name and directory	
Project files and libraries Target device family and device EDA tool settings Vou can change the settings for an existing project and specify additional project-wide settings with the Settings command (Assignments menu). You can use the various pages of the Settings dialog box to add functionality to the project. Don't show me this introduction again Help <a href="https://www.edu/settings/libraries/settings/libraries/settings/settings/libraries/settings/seti</td> <td>•</td> <td>Name of the top-level design entity</td> <td></td>	•	Name of the top-level design entity	
Target device family and device EDA tool settings Vou can change the settings for an existing project and specify additional project-wide settings with the Settings command (Assignments menu). You can use the various pages of the Settings dialog box to add functionality to the project. Dent show me this introduction again Help	•	Project files and libraries	
EDA tool settings Vou can change the settings for an existing project and specify additional project-wide settings with the Settings command (Assignments menu). You can use the various pages of the Settings dialog box to add functionality to the project. Don't show me this introduction again Help < Back Next> Einish Cancel	•	Target device family and device	
You can change the settings for an existing project and specify additional project-wide settings with the Settings command (Assignments menu). You can use the various pages of the Settings dialog box to add functionality to the project. Dent show me this introduction again Help < Back	•	EDA tool settings	
Don't show me this introduction again Help < Back Next> Einish Cancel	You can c menu). Yo	hange the settings for an existing project and specify additional project-wide settings with the Settings command (Assignments ou can use the various pages of the Settings dialog box to add functionality to the project.	
Don't show me this introduction again Help < Back Next > Einish Cancel			
Don't show me this introduction again Help <back< td=""> Next></back<>			
Don't show me this introduction again Help <back< td=""> Next></back<>			
Don't show me this introduction again Help < Back Next > Einish Cancel			
Don't show me this introduction again			
Don't show me this introduction again Help < Back Next > Einish Cancel			
Don't show me this introduction again Help < Back Next > Einlish Cancel			
Don't show me this introduction again Help < Back Next > Einish Cancel			
Don't show me this introduction again Help <back next=""> Einish Cancel</back>			
Don't show me this introduction again Help < Back Next > Einish Cancel			
Don't show me this introduction again Help < Back Next > Einlish Cancel			
Don't show me this introduction again Help < Back			
Don't show me this introduction again Help < Back			
Dent show me this introduction again Help < Back			
Help < Back Next> Einlish Cancel	Don't	show me this introduction again	
Help <a>Back Einish Cancel			
	Help	< Back Next > Einish Can	el

5.1.3 Click "Next".



5.1.4 Browse in the project directory and choose a preferred location for the new project. Then create new folder named C10LPRefKit_blinky. This will be the folder containing all the project files.

New Project Wizard				>
Directory, Name, Top-Level Entity				
What is the working directory for this project?				
C:/C10LPRefKit/C10LPRefKit_blinky				
What is the name of this project?				
What is the name of the <u>t</u> op-level design entity for this project? This name	me is case sensitive and r	nust exactly ma	tch the entity na	me in the
design file.				
Use Existing Project Settings				

5.1.5 Enter the project name: "top".

New Project Wizard				
nen ngee man				
Directory, Name, Top-Level Entity				
What is the working directory for this project?				
C:/C10LPRefKit/C10LPRefKit_blinky				
What is the name of this project?				
top				
What is the name of the top-level design entity for this project? This name is ca design file.	se sensitive and	must exactly mat	ch the entity name	in the
ton				
·				
Use Existing Project Settings				

5.1.6 Click "Next".

5.1.7 Project Type

In this page you choose the Project Type. In this tutorial, a new project will be created, and thus the default settings of empty project should be selected.

New Project Wizard		
Project Type		
Select the type of project to create.		
Empty project		
Create new project by specifying project files and libraries, target device	amily and device, and EDA tool settings.	
Project <u>t</u> emplate		
Create a project from an existing design template. You can choose from download design templates from the <u>Design Store</u> .	design templates installed with the Quartus Prime soft	tware, or
Help	< <u>B</u> ack <u>N</u> ext > <u>F</u> inish	Cancel

- 5.1.8 Click "Next".
- 5.1.9 Add Project Files

The Add File window will appear. For this tutorial, new design files will be created so no files will be added. For other designs, files could be added here.

ile name:			Ac	ld
۹.			Add	Aļl
ile Name Type	Library Design Entry/Synth	esis Tool HDL Version	Rem	ove
			U	p
			Do	wn
			Prope	ertie

5.1.10 Click "Next".


5.1.11 Select the Device Part Number of the C10LPRefKit Board

In the Family and Device Settings, use the pull-down menu to select the family as Cyclone 10 LP. Then in the Name Filter enter **10CL055YU484C8G**.

Select the family and	device you want to	target for	compilation.				
You can install additi	ional device support	with the	Install Devices co	ommand o	n the Tools menu	<i>I</i> .	
To determine the ver	sion of the Quartus	Prime sof	tware in which y	our target (device is support	ed, refer to the <u>Device Support List</u> webpage	
Device family				Show i	n 'Available devic	es' list	
Eamily: Cyclone	10 LP		-	Pac <u>k</u>	age: A	ny 👻	
Device: All		Pin <u>c</u>	Pin <u>c</u> ount: Any				
Target device			Core	Core speed grade: Any 💌			
			Name filter: 10CL055YU484C8G				
Specific device	selected in 'Availab	le devices	'list	✓ s	how advanced de	evices	
O Other: n/a							
A <u>v</u> ailable devices:							
Name	Core Voltage	LEs	Total I/Os	GPIOs	Memory Bits	Embedded multiplier 9-bit elements	
10CL055YU484C8G	i 1.2V	55856	322	322	2396160	312	

Rather than entering the exact part number, the pull-down menus can be used to select the correct family, package, pin count, and speed grade. Quartus Prime will use these settings to compile the design, and also provide the programming file that you will use later to program the device.

5.1.12 Click "Next".

5.1.13 EDA Tool Settings

In the EDA tool Settings window, disable any EDA tools, if there are any present. EDA tools are third party tools that work with Quartus Prime for design entry, simulation, verification, and board-level timing. For this tutorial, no EDA software will be used, as only Quartus Prime will be used.





'ool Type	Tool Name	Format(s)		Run Tool Automatically
Design Entry/Synth	None> ``	<none></none>	Ÿ	Run this tool automatically to synthesize the current design
Simulation	<none></none>	None>	Ŧ	Run gate-level simulation automatically after compilation
Board-Level	Timing	<none></none>	*	
	Symbol	<none></none>	•	
	Signal Integrity	<none></none>	*	
	Boundary Scan	<none></none>	Ŧ	

5.1.14 Click "Next".

5.1.15 Project Summary Page

This is the Summary Page that shows the settings Quartus Prime will use for this Project. Those settings can be changed if required later.

S New Project Wizard	×
Summary	
When you click Finish, the project will be created with the following s	ettings:
Project directory:	C:/C10LPRefKit/C10LPRefKit_blinky
Project name:	top
Top-level design entity:	top
Number of files added:	0
Number of user libraries added:	0
Device assignments:	
Design template:	n/a
Family name:	Cyclone 10 LP
Device:	10CL055YU484C8G
Board:	n/a
EDA tools:	
Design entry/synthesis:	<none> (<none>)</none></none>
Simulation:	<none> (<none>)</none></none>
Timing analysis:	0
Operating conditions:	
VCCINT voltage:	1.2V
Junction temperature range:	0-85 �C
Help	< <u>Back</u> <u>N</u> ext > <u>F</u> inish Cancel

5.1.16 Click "Finish".

WDW

wow

5.2 Building a Blinky Project with Cyclone 10 LP RefKit

Overview: In this section you will create the components to a design, make connections, set the pins, and compile a project. The goal is to go through the design process of a simple blinky project, where the toggle speed of the LEDs could be controlled by one of the pushbuttons of the C10LP RefKit.

5.2.1 Block Diagram



The final system that will be built with the following steps will look as follows when complete:

5.2.2 Components of the Design

There are three components in the system: a PLL, a counter and a mux. The components, in the following steps, will be built separately and then connected together. A user push button on the board controls the mux. The mux in turn control which of the counter outputs (slow counting or fast counting) will be shown on the LEDs. There are different ways to create components, such as RTL or schematic. In this lab, schematics will be used. There are also different ways for entering schematics such as Qsys and IP Catalog. This lab will focus on the IP Catalog.



5.2.3 Catalog IP

The IP Catalog allows you to create and modify design files with custom variations. The IP Catalog window is open by default when you open Quartus Prime. If it's not present, you can open it by going to the tab **Tool** \rightarrow **IP Catalog**.



5.2.4 Create and Configure PLL

In the IP Catalog, browse for ALTPLL, via: Basic Functions \rightarrow Clocks; PLLs and Resets \rightarrow PLL or type in the search field for "PLL".

5.2.4.1 In the Search bar of the IP Catalog, type "pll" and select **ALTPLL** which stands for Altera Phase Locked Loop.





5.2.4.2 Click "Add". When the Save IP Variation window appears, enter the file name variation as PLL and select VHDL (Verilog can be used as well). Both Verilog and VHDL schematics will be created.

🕥 Save IP Variation		×
IP variation file name: C:/C10LPRefKit/C10LPRefKit_blinky/PLL IP variation file type]	OK Cancel
VHDL Verilog		

5.2.4.3 Click "OK".

The next step is to configure the PLL component that we just named.

5.2.4.4 Enter the PLL reference clock frequency to match the clock input on the C10LP RefKit Board. We have 12 MHz and 25MHz clock signals coming into the FPGA, in this example, we will use 12MHz for the inclk0 input.

The setting should look like this:

× MegaWizard Plug-In Manager [page 1 of 12]	? ×
ALTPLL	About Documentation
Parameter 2 PLL 3 Output 4 ED Settings Reconfiguration Clocks	A 5 Summary
General/Modes Inputs/Lock Bandwidth/SS	Clock switchover
	Currently selected device family:
PLL	₩ Match project/default
inclk0 inclk0 frequency: 12.000 MHz c0	Able to implement the requested PLL
Areset Operation Mode: Normal locked Clik Ratio Ph (dg) DC (%) o0 1/1 0.00 50.00	General
Cyclone 10 LP	Which device speed grade will you be using?
	Use military temperature range devices only
	What is the frequency of the inclk0 input? 12.000 MHz 💌
	I Set up PLL in LVDS mode Data rate: Not Available ▼ Mbps
	PLL Type Which PLL type will you be using?
	C Fast PLL C Enhanced PLL © Select the PLL type automatically
	Operation Mode
	How will the PLL outputs be generated? Use the feedback path inside the PLL
	In normal mode
	C In source-synchronous compensation Mode
	In zero delay buffer mode Connect the fbmimic port (bidirectional)
	C With no compensation
	C Create an 'fbin' input for an external feedback (External Feedback Mode)
	Which output clock will be compensated for?
	Cancel < Back Next > Einish

- 5.2.4.5 Click "Next".
- 5.2.4.6 Simplify the PLL, by disabling 'areset' and 'locked output'.

The setting should look like this:

Cyclone 10 LP RefKit User Guide

WDW

≫ MegaWizard Plug-In Manager [page 2 of 12]	?	×
ALTPLL About	<u>D</u> ocument	ation
Parameter PL 3 Output 4 EDA 5 Summary Settings Clods Souther the set of the set o		
General/Modes Inputs/Lock Bandwidth/SS Clock switchover		
PLL Optional Inputs Optional Inputs Create an infernation input to selectively enable the PLL Optional Inputs Create an infernation input to selectively enable the PLL Optional Inputs Create an infernation input to selectively enable the PLL Optional Inputs Create an infernation input to selectively enable the PLL Optional Inputs Create an infernation input to selectively enable the PLL Optional Inputs Create an infernation input to selectively enable the phase/frequency detector Lock Output Create Indee Self-reset on loss lock Advanced Parameters Advanced PLL parameters Using these parameters is recommended for advanced users only Create output fle(s) using the 'Advanced PLL parameters - Configurations with output dock(s) that use cascade counters are not supported Configurations with output dock(s) that use cascade counters are not supported		
Cancel < Back Next	t> Fir	nish

- 5.2.4.7 Click "Next".
- 5.2.4.8 Continue to select Next to go through the various options (from Pages 3 to Pages 5) but leaving the default options as they are. The page numbers can be seen on the top of the window.
- 5.2.4.9 On page 6, (c0-Core/External Output Clock) select "Enter output clock frequency" and set the requested setting to 20 MHz, leave the rest as default. For simplification, there is one input to the PLL (12 MHz), and one output of the PLL (20 MHz)

ℜ MegaWizard Plug-In Manager [page 6 of 12]		?
altpll		<u>A</u> bout
Parameter 2 PLL 3 Output Settings Reconfiguration Clocks	4 EDA 5 Summary	
dk c1 dk c2 dk c3 PLL PLL Incik0 Incik0 frequency: 12.000 MHz Operation Mode: Normal CB Cikit Ratio Phr. (dg) DC (%) C0 0 dd (dd (dd (dd (dd (dd (dd (dd (dd (dd	CC - Core/External Output Clor Able to implement the requested PLL Use this dock Clock Tap Settings Clock Tap Settings Finter output clock frequency: Enter output clock frequency: Clock division factor Clock division factor Clock phase shift	Requested Settings Actual Settings 20.0000000 MHz • 20.000000 1 - / 0.00 - / / 0.00 - / / /
	Clock duty cycle (%)	50.00 •
	Note: The displayed internal settings of the PLL is recommended for use by advanced users only	Description Val Primary clock VCO frequency (MHz) 48(Modulus for M counter 40 ↓ ↓
		Per Gock Feasibility Indicators c0 c1 c2 c3 c4
		Cancel < <u>B</u> ack <u>N</u> ext > <u>F</u> inisl

Cyclone 10 LP RefKit User Guide

WDW

- 5.2.4.10 Click "Next" until reaching page 12.
- 5.2.4.11 On page 12 there is a list of output files that will be generated. Since the design will be done in a schematic, you will need to select PLL.bsf checkbox. The .bsf file provides a symbol that can be used in the schematic design we will be creating later.

× MegaWizard Plug-In Manager [page 12 of 12]		? ×
		About Documentation
Parameter PLL Output Settings Reconfiguration Clocks	4 EDA 5 Summary	
PLL Inck0 Inck0 frequency: 12 000 MHzC0,	Turn on the files you wish to green checkmark indicates a checkbox is maintained in su The MegaWizard Plug-In Ma C: \C 10LPRefixit\C 10LPRefixi	o generate. A gray checkmark indicates a file that is automatically generated, and a no potonal file. Click Finish to generate the selected files. The state of each beequent MegaWizard Plug-In Manager sessions. nager creates the selected files in the following directory: Lybinky\
Operation Mode: Normal	File	Description
Clk Ratio Ph (dg) DC (%) c0 5/3 0.00 50.00	PLL.vhd	Variation file
	PLL.ppf	PinPlanner ports PPF file
Cyclone 10 LP	PLL.inc	AHDL Indude file
	PLL.cmp	VHDL component declaration file
	PLL.bsf	Quartus Prime symbol file
	PLL_inst.vhd	Instantiation template file
		Cancel < Back Mext > Einish

- 5.2.4.12 Click "Finish". The PLL (1st component) will now be created.
- 5.2.4.13 If this is the first time that you are using this version of Quartus Prime, you might see a pop-up Window for Quartus Prime IP Files, that asks if the tool should add IP files automatically after generating them.

🕥 Quartus Prime IP Files			>
When you create an Intel IP va Quartus Prime IP Files are use want to add the Quartus Prime	riation, a Quartus Pr d to represent the In e IP File to the projec	ime IP File is g tel IP in your d :t?	enerated. esign. Do you
✓ C:\C10LPRefKit\C10LPRe	fKit blinkv\PLL aip		
 Automatically add Quartus 	s Prime IP Files to all	projects	
✓ Automatically add Quartus (Note: Turning on this option p change this setting in the Opti	Prime IP Files to all permanently suppresions dialog box)	projects sses this dialoį	g box. You can

- 5.2.4.14 Select "Automatically add Quartus Prime IP Files to all projects".
- 5.2.4.15 Click "Yes" to allow all of the IP to automatically be added to the project, and so that this message will not be seen for other designs.



5.2.5 Create and Configure the Counter

The next step is to create the counter which will drive the LEDs on the C10LP RefKit board.

5.2.5.1 To create this counter, select the IP Catalog and expand the **Basics** \rightarrow **Arithmetic** and select the LPM_COUNTER or type "counter" in the search field.



Note that the LPM stands for Library of Parameterized Modules

- 5.2.5.2 Click "Add".
- 5.2.5.3 When the Save IP Variation pop up appears, enter "simple_counter" and select VHDL as below:

🕥 Save IP Variation	×
IP variation file name: C:/C10LPRefKit/C10LPRefKit_blinky/simple_counter IP variation file type	 OK Cancel
VHDL Verilog	

- 5.2.5.4 Click "OK".
- 5.2.5.5 The next step is to increase the size of the counter to a number of bits large enough to divide down the clock so we can see the LEDs toggling.
- 5.2.5.6 Change this number to 32.
- 5.2.5.7 Let the counter to be Up only, so the LEDs will show the counters counting up.

🔨 MegaWizard Plug-In M	lanager [page 1 of 5]	? ×
🧳 LPM_CC		Documentation
1 Parameter Settings	3 Summary	
General General 2	> Optional Inputs >	
simple counter	Currently selected device family: ☐ Cyclone ☐ Mate	± 10 LP h project/default
	How wide should the 'q' output bus be? 32 💌 bits	
	What should the counter direction be?	
	Up only	
	C Down only	
	 Create an 'updown' input port to allow me to do both (1 counts up) 	; 0 counts down)
	<u>-</u>	
Resource Usage		
32 lut + 32 reg	Cancel < <u>B</u> ack N	ext > <u>F</u> inish

- 5.2.5.8 Select "Next" until reaching Page 5.
- 5.2.5.9 Select simple_counter.bsf checkbox to generate a symbol for our schematic design.

% MegaWizard Plug-In Manager [page 5 of 5] ? X				
LPM_COUNTER				
1 Parameter Settings	3 Summary			
simple counter ← clock up counter q[31.0]	Turn on the files you wish t generated, and a green ch selected files. The state of Manager sessions. The MegaWizard Plug-In Ma C:\C10LPRefixit\C10LPRefix	o generate. A gray checkmark indicates a file that is automatically ecomark indicates an optional file. Click Finish to generate the each checkbox is maintained in subsequent MegaWizard Plug-In anager creates the selected files in the following directory: it_blinky\		
	File	Description		
	✓ simple_counter.vhd	Variation file		
	simple_counter.inc	AHDL Include file		
	simple_counter.cmp	VHDL component declaration file		
	✓ simple_counter.bsf	Quartus Prime symbol file		
	simple_counter_inst	Instantiation template file		
Resource Usage 32 lut + 32 reg		Cancel <back mext=""> Finish</back>		

5.2.5.10 Click "Finish".

The counter is now created.

5.2.6 Create and Configure the Multiplexer

The next step is to create a mux component. This mux will be used along with a push button on the C10LP RefKit board to control the speed of the counter, where the counter outputs will be seen on the LEDs.

5.2.6.1 To create this mux, select IP Catalog and expand **Basic Functions** → **Miscellaneous** and select LPM_MUX or type mux in the search field.



5.2.6.2 Click "Add".

5.2.6.3 In the Save IP Variation, enter the name of the counter_mux and the file type to be VHDL.

Save IP Variation	×
IP variation file name: C:/C10LPRefKit/C10LPRefKit_blinky/counter_mux	OK Cancel
VHDL Verilog	

- 5.2.6.4 Click "OK".
- 5.2.6.5 Select 2 data inputs and the width of the input and output buses to be 8 bits. The reason for 8 bits is that there are 8 LEDs to be toggled (showing count values).

The screen should look like this now:



🔌 MegaWizard Plug-In Mar	nager [page 1 of 3] ? ×
🍓 LPM_MU	X About Documentation
1 Parameter Settings	3 Summary
counter_mux data1x[7.0] data0x[7.0] g	Currently selected device family: Cydone 10 LP ✓ Match project/default How many 'data' inputs do you want? 2 How wide should the 'data' input and the 'resulf' output buses be? Do you want to pipeline the multiplexer? O you want to pipeline the multiplexer? O Yes, I want an output latency of 1 Create an asynchronous Clear input Create a Clock Enable input
Resource Usage	Cancel < <u>B</u> ack <u>N</u> ext > <u>Fi</u> nish

- 5.2.6.6 Click "Next" until Page 3.
- 5.2.6.7 Select counter_mux.bsf checkbox to generate a symbol for our schematic design.



5.2.6.8 Click "Finish".

5.2.7 Adding the Components to the Schematic

The next step would be to connect all three components together.

5.2.7.1 To do so, select File menu, then select New and select Block Diagram/Schematic File.



5.2.7.2 Click "OK".

A new schematic will be created, where the components can be added.

5.2.7.3 Right click on the schematic page and select **Insert** \rightarrow **Symbol...** as seen below.



5.2.7.4 In the new window, expand "Project" and the three components that were created can now be seen.

Cyclone 10 LP RefKit User Guide

WUV



- 5.2.7.5 Select "PLL".
- 5.2.7.6 Click "OK".
- 5.2.7.7 The PLL component can be added now by left clicking on the schematic page.
- 5.2.7.8 Just like in the steps from 5.2.7.3 to 5.2.7.6, do the same for counter_mux and simple_counter to add them to the schematic page. The order of adding the components does not matter, as the connections between them will happen in the following steps.
- 5.2.7.9 After adding three components, your schematic should look similar to the following. To place them similarly, simply drag the components to the appropriate locations.





5.2.8 Connecting the Components

Next step is to make the proper connections between the three components we just added to the schematic.





5.2.8.2 Connect the c0 of the PLL to the simple_counter as shown below:

This will mean that a single signal (c0) is connected to the simple_counter (clock).

5.2.8.3 Select the "Bus Tool".

5.2.8.4 Using the bus tool create a connection coming out of the simple_counter and one connection for each of the inputs of the counter_mux as show below.

5.2.8.5 Right click on the output bus of the simple counter that you just created and select **"Properties"**.

Set the name of the bus to: **counter[31..0]** The view of the "Bus Properties" should look like this:

🔁 Bus Prop	operties	×
General	Font Format	
<u>N</u> ame:	counter[310]	
<u>H</u> ide	e name in block design file.	
	OK Cancel	Help

- 5.2.8.6 Click "OK".
- 5.2.8.7 Do the same for input buses of the mux:

Name the top bus input:	data1x[70] \rightarrow counter[2431]
Name the bottom bus input:	data0x[70] \rightarrow counter[1926]

Schematic should look like this:

5.2.9 Add inputs, outputs to the schematic

5.2.9.1 Click on the "Pin Tool" as show below and select "Input".

5.2.9.2 Add one input pin for inclk0 of the PLL and add other one input pin for sel of counter_mux.

Your schematic should look like this:

- 5.2.9.3 Rename the pin_name1 to **CLK12M** by double clicking its current name. This is going to be the clock signal coming into the FPGA.
- 5.2.9.4 Rename the pin_name2 to **USER_BTN** by double clicking its current name. This is going to be one of the user buttons of the C10LP RefKit board to select the mux.

5.2.9.5 Using the "Node Tool" connect:

CLK12M \rightarrow inclk0 (of the PLL component)USER_BTN \rightarrow sel(of the counter_mux component)

Your schematic should look like this now:

5.2.9.6 Click on the "Pin Tool" as before, but this time select "Output".

- 5.2.9.7 Add one output pin for the LEDs.
- 5.2.9.8 Rename the pin to LED[7..0].
- 5.2.9.9 Using the **"Bus Tool"**, make the connection between counter_mux component and output pin:

result[7..0] \rightarrow LED[7..0]

The final schematic should look like the following:

Looking at the schematic, even though the buses are not connected together by wires, the names of counter tell Quartus Prime to connect the signals together. Overall, the user button will toggle between displaying higher 8 bits of the counter and 8 lower bits of the counter. The signals of the counter that are not connected will not be used by Quartus Prime.

5.2.9.10 Save your design. Open the File Menu and select "Save". Save it as top.bdf

5.2.10 Analysis and Synthesis

The next step is to run Analysis and Synthesis to ensure that there are no errors in the design. To run Analysis and synthesis open **Processing** \rightarrow **Start** \rightarrow **Analysis and Synthesis** or from clicking button on the top toolbar.

There should be no errors. If there are errors, they should be fixed before continuing and Analysis and Synthesis run again.

5.2.11 Adding Timing Constraints

Timing Constraints tell the Quartus what the timing requirements for this design are. Timing Constraints are required in every CPLD/FPGA design.

5.2.11.1 To add the timing constraints, select File \rightarrow New and under the "Other File" section, select "Synopsys Design Constraints File" and select "OK".

5.2.11.2 Type or copy the following lines into this new file:

```
#create input clock which is 12MHz
create_clock -name CLK12M -period 83.333 [get_ports {CLK12M}]
#derive PLL clocks
derive_pll_clocks
#derive clock uncertainty
derive_clock_uncertainty
#set false path
set_false_path -from [get_ports {USER_BTN}]
set_false_path -from * -to [get_ports {LED*}]
```

The first line "create_clock" tells Quartus Prime that the clock, CLK12M is 83.333 ns (12 MHz). It also assigns the CLK12M to a pin (port) in the .sdc format.

The second line "derive_pll_clocks" tells the software to look if there are any PLLs, and if so, automatically derive the clock multiplication/division of the outputs of the PLL even if they are used internally within the CPLD/FPGA.

The third line "derive_clock_uncertainty" tells the software to automatically determine the internal clock uncertainty. No clock is ideal, and thus there will be some internal jitter within the FPGA associated with it.

The fourth and fifth line "set_false_path" tells the software to not do any timing optimization to the stated paths/pins. The I/Os of this design are trivial, so they can be ignored in the Timing Analysis.

- 5.2.11.3 Use File \rightarrow Save to save it as top.sdc.
- 5.2.11.4 Ensure that the file is added to the Project: Assignments → Settings and select "Timing Analyzer". The top.sdc should have been already added by default. If it is not, it will need to be added manually.

✓ Settings - top		- 0	×
Category:		Device/B	oard
Category: General Files Libraries * IP Settings IP Catalog Search Locations Design Templates * Operating Settings and Conditions Voltage Temperature * Compilation Process Settings Incremental Compilation * EDA Tool Settings Design Entry/Synthesis Simulation Board-Level * Compiler Settings VHDL Input Verilog HDL Input Default Parameters Timing Analyzer	Timing Analyzer Specify Timing Analyzer options. SDC files to include in the project Eile name: Image: Image: <th>Device/B</th> <th>oard</th>	Device/B	oard
Design Assistant Signal Tap Logic Analyzer Logic Analyzer Interface Power Analyzer Settings SSN Analyzer	Tcl Script File for customizing reports during compilation Icl Script File name: ✓ Run default timing analysis before running custom script Metastability analysis Synchronizer identification: Auto Description:		
	Associates a Synopsys Design Constraint File (sdc) with this project. Image: Buy Software OK Cancel Apply	He	lp

5.2.12 Pinning Assignments

Before the design can be downloaded to the FPGA, pin assignments that match the hardware on the board are needed. There are different ways to do this such as the Pin Planner, Assignment Editor, and text files.

The following steps will show one of these ways, the Pin Planner. Since there are only 10 pins that need to be assigned, the Pin Planner can be used. If many pins are needed, other ways can be used such as the Quartus Assignment Editor, or by importing constraints from a text file or spreadsheet.

5.2.12.1 Open the Pin Planner: Assignments \rightarrow Pin Planner.

A new window will open as seen below:

5.2.12.2 To make pin assignments, select the CLK12M (node name) on the bottom portion and drag and drop it to pin G21 of the Top View of the FPGA or alternatively set the Location field of the CLK12M to **PIN_G21**.

Report 🖗 🛙 🕅											Pin L	rgend	
Report not available						Тор	View - W	/ire Bond			Sym	ol Pin Type	*
					C	velope 10	10 100		ACOC		0	User I/O	
					C	yclone to	LP - TUC	10551040	54066		•	User assigned V	
								10000	í.		•	Fitter assigned I	
						(1.4914)11.0914	***	a warrant warrant				Unbonded pad	
					1 2 3	4 5 6 7 8	9 10 11 12	13 14 15 16 1	17 18 19 20 21	22	•	Reserved pin	
				A	V	0.0404040	···· 11	-0-0-0-0-	O L O	A	(c)	Other _	
				8	(n) (p) (p) (○ <u>A</u>	- p - p - J _	(p) (p) (p) (p) (p *p (D (L *p	• n) B	(in the second s	DEV. OF	
				c	Veo e		VAVV		D • n • n • n • p	• n) C		DEV CIR	
				D		0.40.40.40	0 . 0 0	· /	4+ 9+ 9+ 9+ P	•n) D	0	DIEE n	
				E		D D +0+0	0 • p • n	(+n +p +p /	A 0 V	• n E	6	OVER IN	
				F		A/A-0+0	0 .pp. V	• p • p • n • n •	n A	• 6) F		CIT -	
				G	25.00	A 40 40	0.00		p +n 0 / 1	1 0		CCK_H	
				IDANSL T				VAZA C.		n H manufacture		CUK_P	
				(Trent terms)	D ($\nabla \Lambda \overline{\Lambda} \Lambda$	赤赤豆香口	D +0 +n +0	(n) d		Other PLL	
					007	0.007	XXXX	$\Theta X X \Theta$	0000			Other dual	
					KXX6		A AYAY	XXXXX				MSELO	
					1000	$ \land \land \land \leftrightarrow \land $		XXX				MSEL1	
				P	10.00.00	A. VA	AV V V	VIVIN	0 0 00 00 00		3	MSEL2	
				N		D	ДХХХ	XAX	n •p •p •n R	E N	0	MSEL3	
Convert Daniel				P			TATA	/1/1//	P. O. V. (-P.	n P	0	CONF_DONE	
dioops Report					(n) (p) (n) ($\sqrt{n} \sqrt{n}$	PP TO LI	a =a =p = D =p	• n) R	(e)	nCE	
Tasks 🛛 🗍 🕄 🛞				riner,r T	LIA) *p /A /I	/1(#p)#n) \/	/1 = n L I	D D = p = n _	L T meno		nCONFIG	
* 🎦 Early Pin Planning				U		A	•o D •o D	D - n - p - p - 1	a) (A) = 0 = n = 0	• n) U		TDI	
Early Pin Planning				v	-n -p (D (0 *0 *p *0 *0		(p (p (n) C /	A 0 .	•n) V	8	тск	
Run I/O Assignment Analys				w		0.49.49	0 • p 0 0	-p -n D 0 -	P 0 _ 0 + P	• n W	1	TMS	
Export Pin Assignments				Y	n 10 10 1	D D		-n -n -n V-	. 0	•n) x	0	TDO	
Po Finder				AJ	V/D		-p -p 5 5	(a - a - a - a)	/ (D =p =p)	-n AA	0	nSTATUS	
* E Highlight Bing				At	VOOL				0 - 0 0	AB	A	VREF	
WO Reals					1 2 3	5 5 7 8	9 10 11 12	13 14 15 16 1	17 18 19 20 21	22	Δ.	VCCP/VCCR/	
						104.000.0	-	10001.0			A	VCCA	
4 VREF Groups						10 of grad / S and S	e ser	Real of the second	9		A	VCCINT	*
18 Named + + 20 Edit × of 5	LK12M											Eiter Dire	- II - T
												T MILL T MIL	
Direction	Location	I/O Bank	WREF Group	T E M (default)	Reserved	Current Strength	Siew Rate	Unrerential Pair	strict meservation	1.			
Get LED[7] Output	Pin_021	0	00_N1	2.5 V (default)		8mA (default)	2 (default)						
LED[6] Output				2.5 V (default)		8mA (default)	2 (default)						
LED[5] Output				2.5 V (default)		8mA (default)	2 (default)						
LED[4] Output				2.5 V (default)		8mA (default)	2 (default)						
LED[3] Output				2.5 V (default)		8mA (default)	2 (default)						
E CED[2] Output				2.5 V (default)		8mA (default)	2 (default)						
Cutput				2.5 V (default)		8mA (default)	2 (default)						

Note that the Location of the CLK12M is now set to Location PIN_G21 (as seen in blue colour in the top view of the FPGA).

5.2.12.3 The other pins need to be assigned as well. Just like previously set all the pins to their appropriate locations using the table below, by either drag and drop or writing manually the location.

wow

Node Name	Pin Location
LED[7]	PIN_U7
LED[6]	PIN_U8
LED[5]	PIN_U9
LED[4]	PIN_W10
LED[3]	PIN_Y10
LED[2]	PIN_AA9
LED[1]	PIN_AA10
LED[0]	PIN_AB10
USER_BTN	PIN_T11

On the board we have multiple user buttons, in this case, we will use S1, which is the central of joystick buttons.

5.2.12.4 Now the Pin Planner should look like this after assigning all the pin locations.

5.2.12.5 The specific pins are now selected, but the I/O standards now need to be set as well. The button, LEDS, and clock pins are the same I/O standard for C10LP RefKit since all banks and peripherals are powered by 3.3V. The USER_BTN, the LEDs and clock pins are 3.3-V LVTTL. These I/O standards can be set in the Pin Planner, by selecting the I/O Standard. Select the I/O standard either from the "All Pins" tab or the "Groups" tab and change the 2.5V (default) to the specific I/O standard.

The Pin Planner should now look like this:

5.2.12.6 Close the Pin Planner. The settings are automatically saved.

5.2.13 Compiling the Design

5.2.13.1 You can set the default I/O Standard which can eliminate some design warning and save you time from setting the standard for some pins manually.

Open Assignments \rightarrow Device \rightarrow Device and Pin Options \rightarrow Voltage and set Default I/O Standard to "3.3-V LVTTL" and press "OK" to all the windows.

🕥 Device and Pin Options - top			×
Category:			
General	Voltage		
Configuration Programming Files	Specify voltage options for	r the device.	
Unused Pins Dual-Purpose Pins	Default I/O standard:	3.3-V LVTTL	•
Capacitive Loading Board Trace Model	VCCIO I/O bank <u>1</u> voltage:	n/a in Cyclone 10 LP use Pin Planner to adjust VCCIO voltage	~
I/O Timing	VCCIO I/O bank <u>2</u> voltage:	n/a in Cyclone 10 LP use Pin Planner to adjust VCCIO voltage	v
Voltage Pin Placement Error Detection CRC CVP Settings	Core voltage: 1.2V		

The next step is to compile and complete the design. This step will verify that there are no errors, create internal databases, and create programming files that will be used in the next steps.

5.2.13.2 To compile the design, select **Processing** \rightarrow **Start Compilation** or push the \blacktriangleright button on the toolbar.

If there are errors, they will need to be resolved and re-compiled before the design can be programmed to the board. When Compiling finishes and there are no errors, there will be a message at the bottom of the window that states: Full Compilation was successful and a 100% indication along with the compile time in the right bottom corner.

Tasks		Compilation	- I 0 x
	Task		Time
 Image: A set of the set of the	🔹 🕨 Compile Design		00:00:41
 Image: A second s	🕨 🕨 Analysis & Synthesis		00:00:21
 Image: A second s	🕨 🕨 Fitter (Place & Route)		00:00:12
 Image: A second s	🕨 🕨 Assembler (Generate prog	gramming files)	00:00:04
 Image: A second s	🕨 🕨 Timing Analysis		00:00:04
	🕨 🕨 EDA Netlist Writer		
	Edit Settings		
	Program Device (Open Progra	mmer)	

5.2.14 Reading the Compilation Report

After successfully compiling the design, a Compilation Report should appear as shown above:

Compilation Report - top 🗙		
Table of Contents	Flow Summary	
Elow Summary	< <filter>></filter>	
Flow Settings	Flow Status	Successful - Fri Feb 11 16:16:04 2022
Flow Non-Default Global Settings	Quartus Prime Version	21.1.0 Build 842 10/21/2021 SJ Lite Edition
== Flow Elapsed Time	Revision Name	top
Flow OS Summary	Top-level Entity Name	top
Flow Log	Family	Cyclone 10 LP
Analysis & Synthesis	Device	10CL055YU484C8G
Fitter	Timing Models	Final
Elow Messages	Total logic elements	40 / 55,856 (< 1 %)
	Total registers	32
	Total pins	10/322(3%)
	Total memory bits	0 / 2 396 160 / 0 %)
	Embedded Multiplier 9-bit elements	0/312(0%)
	Total PLLs	1/4(25%)
4		
· ·		

This report is very useful with a lot of information about the design. Last message state that the design was fully constrained, Timing Analysis and compilation successful, but there is more to it:

- In the Flow Summary, it can be seen how many logic elements the whole design took, along with total PLLs, registers, pins, etc.
- In Analysis and Synthesis, more detailed information about the resources used can be seen in Resource Usage Summary, as well how many LEs were used for each component in Resource Utilization by Entity.
- In the Fitter, more detailed information about the pins and their banks can be seen.
- Timing Analyzer shows various timing information concerning the design, as well as if the design has met the timing requirements. In this case timing requirements were met, but in other cases that requirements might not be met, could be solved by going over the information provided in the reports inside this folder. Most notable reports in this folder are the maximum frequency the design can achieve, setup and hold slack, unconstrained paths in case they were missed, etc.

wow

Chapter 6 - Configuring the Cyclone 10 LP RefKit

After successfully compiling your project, there should new files be generated. In case of Cyclone 10 LP devices, only the .sof file is generated automatically.

6.1 Configure the FPGA in JTAG mode

6.1.1 Connect your C10LP RefKit board to a power supply and then to your PC using an USB cable. Since the Arrow USB Blaster should be already installed, the Window's Device Manager should display the following entries are highlighted in red (port number may differ depending on your PC). If the Arrow USB Blaster is not installed, please refer to <u>Chapter 4.2</u> for installing the drivers.

6.1.2 Open the Quartus Prime Programmer from **Tools** \rightarrow **Programmer** or double-click on Program Device (Open Programmer) from the Tasks pane.

Tasks		Compilation	• ₽ ⊘×
	Task		Time
-	🔹 🕨 Compile Design	00:0	0:41
-	Analysis & Synthesis	00:0	0:21
-	Fitter (Place & Route)	00:0	0:12
-	Assembler (Generate pro	ogramming files) 00:0	0:04
-	Timing Analysis	00:0	0:04
	🕨 🕨 EDA Netlist Writer		
	Edit Settings		
	Program Device (Open Prog	rammer)	

6.1.3 The programmer should add the programming file automatically. After opening the program this should be the view of the new window:

avraite betup	No Hardware				Mode	e: JTAG			Prog	ress:		
able real-time IS	SP to allow background p	programming when a	vailable									
la Start	File	Device	Checksum	Usercode	Program/ Configure	Verify	Blank- Check	Examine	Security Bit	Erase	Reac ISP ecks CLAM	'S FI (P
h Stop	output_files/top.sof	10CL055YU484	002B8A50	002B8A50	✓							
uto Detect												
Delete												
dd File												
ange File												
Save File												
d Device		····										
The second se												
Down												
Down	10CL055Y	U484										
	•											

6.1.4 Click **Hardware Setup...** and double click **Arrow-USB-Blaster** entry in the Hardware Setup tab. The Currently selected hardware should now show Arrow-USB-Blaster [USB0] (depending on your PC, the USB port number may variant).

Hardware Settings JTAG :	Settings		
Select a programming hardwa hardware setup applies only t	are setup to use wher o the current progran	n programming devic nmer window.	es. This programming
Currently selected hardware:	Arrow-USB-Blaster	[AR30FBPQ]	
Hardware frequency:	20000000		ŀ
Available hardware items	_		
Hardware	Server	Port	Add Hardware
Arrow-OSD-blaster	Locat	ARSUFDPQ	Remove Hardware

- 6.1.5 Click "Close".
- 6.1.6 Make sure the hardware setup is Arrow-USB-Blaster [USB0] and the mode is JTAG. If the Mode is not set to JTAG, click on it, and select JTAG from the drop-down menu.

🚖 Hardware Setup	Arrow-USB-Blaster [AR30FBPQ]		Mode:	JTAG	•	Progress:	
Enable real-time ISI	P to allow background programming wh	en available					

Cyclone 10 LP RefKit User Guide

WDW

www.arrow.com February 2022

MOW

- 6.1.7 If the configuration has been added by default, you can skip the following steps and continue with the 6.1.12 point.
- 6.1.8 Click "Auto Detect" on the left side of the Programmer.

6.1.9 Select **10CL055Y** device and click "OK" on the Select Device window.

6.1.10 Double click <none> to choose programming file.

File	Device	Checksum	Usercode	Program/	Verify	Blank-	Examine	Security	Erase	Read	ISP	'S Fi	(P F	
				Configure		Check		Bit		ecks	CLAMP			
<none></none>	10CL055Y	00000000	<none></none>											

6.1.11 Navigate to <project_directory>/output_files/ in your compilation directory. Select and open the top.sof file.

6.1.12 Make sure the Programmer shows the correct file and correct part in the JTAG chain and check the Program/Configure checkbox.

Ele Edit View Processing Tools Window Help Search Intel FPGA Arrow-USB-Blaster [AR30FBPO] Enable real-time ISP to allow background programming when available File Device Checksum Usercode Program/ Verify Blank- Examine Security Erase Reac ISP 5 Fi (P F Charles Stop Charles Tool 100CL055YU484 0028BA50 0028BA50 V Charles File Add File Charles File Add File Column 100CL055YU484 TOOL 055YU484	Programmer - C:/C1	10LPRefKit/C10LPRefKit_b	olinky/top - top - [top	p.cdf]*								-		×
Hardware Setup Arrow-USB-Blaster [AR30FBPO] Mode: JTAG Progress: Enable real-time ISP to allow background programming when available File Device Checksum Usercode Program/ Verify Blank- Examine Security Erase Sea ISP S FI (PF ids Stop output_files/top.sof 10CL055YU484 00288A50 00288A50 00288A50 @@Auto Detect Yoelete Madd File Produce TDI TD	<u>Eile Edit View Pro</u>	ocessing <u>T</u> ools <u>W</u> inde	ow <u>H</u> elp									Search Intel F	PGA	•
Enable real-time ISP to allow background programming when available	Aardware Setup	Arrow-USB-Blaster [AR	30FBPQ]			Mode	: JTAG		÷	Progr	ess:			
File Device Checksum Usercode Program/ Verify Blank- Examine Security Erase Real 15P 5Fi (PF) @bstop	Enable real-time ISF	P to allow background pr	ogramming when av	ailable										
Image Check Bit ICKS CLAMP Image Check Image Check Image Check Image Check Image Check Image Check <	▶ [™] Start	File	Device	Checksum	Usercode	Program/	Verify	Blank-	Examine	Security	Erase	Reac ISP	'S FI (P F	
Madd File Madd File Madd Device Mup Mup Imponent	Stop	output_files/top.sof	10CL055YU484	002B8A50	002B8A50	Contidure		Check		BIT		ECKS CLAME	,	
X Delete Madd File Y Change File Y Bown Y Bown 1 Down 1 DoL055YU484	💏 Auto Detect					_								
Madd File Change File Madd Device Thup Ub Down 10cL055YU484	× Delete													
Change File Image File <t< td=""><td>Add File</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></t<>	Add File													
Add Device 1 th Down 1 th Down 1 DCL055YU484	隆 Change File													
Mada Device 1 [™] Up 1 [™] Down 10CL055YU484	Save File													
1 [™] Down 10CL055YU484	Add Device	TDI												
J [™] Down 10CL055YU484 ↓ TDO	1 [™] Up	→ →												
	J™Down	10CL055YU	484											
		TDO												

6.1.13 Click Start to program the C10LP RefKit. When the configuration is complete, the Progress bar should reach 100% (Successful).

Progress. 100% (Successing)

The design is now programmed to the FPGA.

Note that turning off and then on the FPGA will result into losing its configuration.

6.2 Serial configuration flash memory programming

The configuration data to be written to EPCQ-A will be part of the JTAG indirect configuration file (.jic). This configuration data is automatically loaded from the serial configuration flash into the Cyclone 10 LP device when the board is powered up.

6.2.1 Programming File generation

6.2.1.1 In Quartus Prime, go to File → Convert Programming Files...

le Tools Window		e toer Kerkit_billiky/top -	top		_		
10000 <u>10</u> 0000	Search Intel F						
pecify the input files to ou can also import inpu ture use. onversion setup files	convert and the type of It file information from (programming file to gen other files and save the co	erate. onversion setup information	created here fo	pr		
Ор	en Conversion Setup D	ata	Save	Conversion Se	tup		
utput programming file	2						
Programming file type	Programmer Object F	ile (.pof)					
Options/Boot info	Configuration device:	EPCE16	• <u>M</u> ode:	1-bit P	assive Serial		
File <u>n</u> ame:	output_files/output_f	ile.pof					
Advanced	Remote/Local update difference file: NONE						
	Create CvP files (G	enerate output file neriol	and and an end of the same de	_			
put files to convert	Create config data	RPD (Generate output_fil	e_auto.rpd)	f)			
put files to convert File/Data	Create config data	RPD (Generate output_fil Properties	e_auto.rpd) Start Address	f) 		Add He <u>x</u> Da	
out files to convert File/Data Options	Create config data	RPD (Generate output_fil Properties	e_auto.rpd) Start Address 0x00010000	f)		Add He <u>x</u> Da	
out files to convert File/Data Options SOF Data	Create config data area Page	RPD (Generate output_fil Properties	e_auto.rpd) Start Address 0x00010000 <auto></auto>	f)		Add He <u>x</u> Da Add <u>S</u> of Pa Add <u>E</u> ile	
out files to convert File/Data Options SOF Data	Create config data area Page	Properties	e_auto.rpd) Start Address 0x00010000 <auto></auto>	f)		Add He <u>x</u> Da Add <u>S</u> of Pa Add <u>E</u> ile Remove	
put files to convert File/Data Options SOF Data	Create config data	Properties	e_auto.rpd) Start Address 0x00010000 <auto></auto>	f)		Add He <u>x</u> Da Add <u>S</u> of Pa Add <u>E</u> ile Remove Up	
put files to convert File/Data Options SOF Data	Create config data area Page	Properties	e_auto.rpd) Start Address 0x00010000 <auto></auto>			Add He <u>x</u> Da Add <u>S</u> of Pa Add <u>E</u> ile Remove Up Down	
put files to convert File/Data Options SOF Data	Create config data	Properties	e_auto.rpd) Start Address 0x00010000 <auto></auto>	f)		Add He <u>x</u> Da Add <u>S</u> of Pa Add <u>E</u> ile Remove Up Down Properties	

- 6.2.1.2 Set the programming file type to JTAG Indirect Configuration File (.jic).
- 6.2.1.3 Click on the button for configuration device.
- 6.2.1.4 Select Cyclone 10 LP for the Device family, choose **EPCQ16A** from the Configuration device tab, and make sure that the **Active Serial** is set to mode.

Device family: Cyclone 10 LP Configuration mode Active Serial Cutom database directory: D/Intel/Cuantur/2011te/devdata/ Configuration Device Instalization Program Erace Verify/Blank-Check//Examine Termination I <instalization< td=""> I Interver I Interver</instalization<></instalization<></instalization<></instalization<></instalization<></instalization<></instalization<></instalization<></instalization<></instalization<></instalization<></instalization<></instalization<></instalization<></instalization<></instalization<></instalization<></instalization<></instalization<></instalization<></instalization<></instalization<></instalization<></instalization<></instalization<>	Configuration Device					>
Configuration mode: Active Serial Program Custom database directory: [V]/Intel/Quartus/20 11te/devdata// Browse. Configuration Device Initialization Program Erase Verify/Blank-Check/Examile Termination Name I Screen devices> Device name: EPC016A Image: Configuration Device Device name: EPC016A 1 Screen devices> Device name: EPC016A Image: Configuration Device Image: Configuration Device name: EPC016A 2 EPC0178A EPC0178A Image: Configuration Device name: EPC016A Image: Configuration Device name: EPC016A 3 EPC0178A Image: Configuration Device name: EPC016A Image: Configuration Device name: EPC016A 5 EPC0178A Image: Configuration Device name: EPC016A Image: Configuration Device name: Image: Configuration Device	Device family:	Cyclone 10 LP				
Configuration Device Initialization Program Erse VerflyfBlahl-Check/Examine Termination Name I Initialization Program Erse VerflyfBlahl-Check/Examine Termination 1 <new device="">> I Device ID: I IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII</new>	Configuration mode:	Active Serial				•
Configuration Device Initialization Program Erase Verify/Blank-Check//Examine Termination Name filter:	Custom database direc	tory: D:/Intel/Quartus/20.1lite	/devdata/			Browse
Name Device name: EPCQ16A 1 senew device>> Device ID: Device ID: 2 EPCQ128A Device ID: Device ID: 5 EPCQ128A Total device de: Total device de: Total device de: 6 EPCQ32A Single (D mode dummy clock: Outroe dummy clock: Outroe dummy clock: Outroe dummy clock: 9 EPCQ32A Outroe dummy clock: Save as template Total device de: Save as template 10 EPCQ32A Save as template Save as template Save as template 11 EPCQ36A Save as template Save as template Save as template 10 MT2SQL03G Save as template Save as template Save as template 10 MT2SQL03G Save as template Save as template Save as template 10 MT2SQL03G Save as template Save as template Save as template Save as template 10 MT2SQL03G Save as template	Configuration Device	Initialization Program	Erase Verify/Blank-Check	/Examine Termination		
Name Device (D: 1 <code< td=""> 2 EPC0128 3 EPC0128A 4 5 EPC0128A 5 EPC0128A 6 EPC0258 7 EPC032 <</code<>	Name filter:			Device name:	EPCQ16A	
1 < <		Name	*			
2 PFC0128 PFC0128A PFC0128A PFC0256 PFC02512 PFC02512 PFC0264A PFC0264 <	1 < <new device="">></new>			Device ID:		
3 PPC0128A 4	2 EPCQ128			Device I/O voltage:		Ŧ
2 Total device de: 5 EPC032A 9 EPC032A 9 EPC032A 9 EPC032A 9 EPC032A 9 EPC04A 10 EPC0512 11 EPC064A 12 EPC064A 13 EPC516 16 EPC54 17 EPC54 18 MT250L016 19 MT250L026 20 MT250L128 21 MT250L256 23 MT250L26 24 MT250L26 25 MT250L28 26 MT250L26 26 <	3 EPCQ128A			Device density:		
s Crooted 6 EPC0256 7 EPC032 8 EPC032A 9 EPC04A 10 EPC0512 11 EPC054 12 EPC054 13 EPC51 14 EPC516 15 EPC516 16 EPC544 17 EPC544 18 MT250L026 20 MT250L026 20 MT250L128 21 MT250L256 23 MT250L256 24 MX25L28 25 MX25L512	4					
6 EPC2356 Guide Jumps clock: Ouad I/O mode dummy clock: Programming flow template: * Edit 10 EPCC312 12 EPC064A Save as template * Edit 12 EPC064A 13 EPC516 Save as template * Edit 12 EPC516 EPC544 13 MT250L026 Fordition Save as template * Save as template * Save as template * Edit Save as template * Edit Save as template * Save as template * <td< th=""><th>5 EPCQ16A</th><th></th><th></th><th>Total device die:</th><th></th><th></th></td<>	5 EPCQ16A			Total device die:		
7 ECC32 8 EPC32A 10 EPC912 11 EPC064 12 EPC312 13 EPC312 14 EPC312 15 EPC316 16 EPC34 17 EPC54 18 MT250L026 20 MT250L256 21 MT250L32 22 MT250L32 24 MT250L32 24 MT250L36 25 MT250L36 26 MT250L36 27 MT250L36 28 MT250L32 29 MT250L32 20 MT250L36 21 MT250L36 22 MT250L36 23 MT250L36 24 MT250L36 25 MT250L36 26 MT250L36 27 MT250L36 28 MT250L36 29 MT250L36 20 MT250L36 21 MT250L36 22 MT250L36 23 MT250L36 24 MT250L36 25 MT250L36	6 EPCQ256			Single I/O mode dummy c	lock:	
8 EPCQ32A 9 PPCQ4A 10 EPCQ512 11 EPCQ64 12 EPCQ64A 13 EPCS1 14 EPCS12 15 EPCS4 17 EPCS4 18 MT250L036 20 MT250L036 21 MT250L265 22 MT250L26 23 MT250L26 24 MT250L26 25 MT250L26 26 MT250L26 27 MT250L26 28 MT250L26 29 MT250L26 20 MT250L26 21 MT250L26 23 MT250L26 24 MT250L26 25 MT250L26	7 EPCQ32			Ouad I/O mode dummy cl	lock:	
9 EPCQ4A * Edit 10 EPCQ512 * Edit 11 EPCQ64 * Edit 12 EPCQ64A * Edit 13 EPCS1 * Save as template * 14 EPCS128 * * Edit 15 EPCS4 * * * 16 PPCS4 * * * 17 EPCS4 * * * 18 MT2SQL016 * * * 19 MT2SQL026 * * * 20 MT2SQL28 * * * 21 MT2SQL28 * * * 24 MX2SL28 * * * 24 MX2SL512 * * *	8 EPCQ32A			,,		
10 EPC0512 12 EPC064 13 EPC51 14 EPC5128 15 EPC516 16 EPC54 17 EPC544 18 MT250L026 20 MT250L128 21 MT250L256 23 MT250L256 24 MT250L256 24 MT250L26 24 MT250L26 25 MT250L2 26 MT250L2	9 EPCQ4A			Programming flow templa	ite:	* Edit
1 ECC084 13 EPCS1 14 EPCS18 15 EPCS16 16 EPCS4 17 EPCS64 18 MT2SQL026 20 MT2SQL256 21 MT2SQL256 22 MT2SQL256 23 MX2SL28 24 MX2SL28 25 MX2SL26 26 MX2SL26	10 EPCQ512			Save as template		
1 Dr. Constance 12 EPCS1 14 EPCS128 15 EPCS16 16 EPCS4 18 MT2SQL026 20 MT2SQL026 21 MT2SQL226 22 MT2SQL226 23 MT2SQL256 24 MT2SQL512 25 MT2SQL512 26 MT2SQL256 27 MT2SQL256 28 MT2SQL512 29 MT2SQL256 20 MT2SQL256 21 MT2SQL256 22 MT2SQL512	12 EPC064A					
14 EPCST28 15 EPCS16 16 EPCS16 17 EPCS64 18 MT250L03G 20 MT250L128 21 MT250L256 22 MT250L128 23 MT250L128 24 MT250L266 25 MT250L128 26 MT250L128 27 MT250L128 28 MT250L512 29 MT250L128 20 MT250L128 21 MT250L128 22 MT250L128 23 MT250L128 24 MT250L128 25 MT250L128	13 EPCS1					
10 EPC54 11 EPC54 12 EPC54 13 MT250L016 14 MT250L026 20 MT250L28 21 MT250L256 22 MT250L28 23 MT250L28 24 MT25128 25 MT2512 26 MT2512 27 MT250L26	14 EPC5128					
10 EPC54 17 EPC54 18 MT250L03G 20 MT250L128 21 MT250L256 22 MT250L512 23 MT251L28 24 MT25128 24 MT25128 25 MT25128 26 MT25128 27 MT250L512 28 MT25128 29 MT25128 20 MT250L512 20 MT250L	15 EPCS16					
17 EPC564 18 MT25QL016 19 MT25QL026 20 MT25QL28 21 MT25QL56 22 MT25QL526 23 MT25QL526 24 MT25QL526 25 MT25QL512 Delete	16 EPCS4					
10 MT250L01G 10 MT250L02G 20 MT250L128 21 MT250L256 22 MT250L512 23 MX25L128 24 MX25L256 25 MX25L512 Delete	17 EPCS64					
19 MT2SQL02G 20 MT2SQL12B 21 MT2SQL12B 22 MT2SQL256 23 MT2SQL512 23 MT2SQL512 24 MT2SQL512 25 MT2SQL512 26 MT2SQL512 27 MT2SQL2S6 28 MT2SQL512 29 MT2SQL2S6 29 MT2SQL2S6 20 MT2SQL2SG 20 MT	18 MT25QL01G					
20 MT25QL128 21 MT25QL256 22 MT25QL512 23 MX25L28 24 MX25L28 25 MX25L512 26 MX25L512 26 MX25L512 27 MX25L512 28 MX25L512 29 MX25L512 20 MX	19 MT25QL02G					
21 MT250L256 22 MT250L512 23 MX25L28 24 MX25L256 25 MX25L512 Delete	20 MT25QL128					
22 MT25QL512 23 MT25L28 24 MT25L256 25 MT25L250 26 MT25L512 26 Delete	21 MT25QL256					
23 MX25L128 24 MX25L256 25 MX25L512 Delete	22 MT25QL512					
24 MX25L256 25 MX25L512 Delete	23 MX25L128					
25 MX25L512 v Delete	24 MX25L256					
Delete OK Fansal Andri	25 MX25L512		•			
OK Careel Anniv			Delete			
					OK	Cancel Apply

6.2.1.5 Click OK. Now the output programming file settings should look like this:

Programming file type:	JTAG Indirect Configuration File (.jic)	FAG Indirect Configuration File (.jic)						
Options/Boot info	Configuration device: EPCQ16A	▼ <u>M</u> ode:	Active Serial 👻					
File <u>n</u> ame:	output_files/output_file.jic							
Advanced	Remote/Local update difference file:	note/Local update difference file: NONE						
	✔ Create Memory Map File (Generate	Create Memory Map File (Generate output_file.map)						
	Create CvP files (Generate output_f	Create CvP files (Generate output_file.periph.jic and output_file.core.rbf)						
	Create config data RPD (Generate o	utput_file_auto.rpd)						

6.2.1.6 Select **Flash Loader** under Input files to covert settings and click on **Add Device...** button.

File/Data area	Properties	Start Address
Flash Loader		
SOF Data	Page_0	<auto></auto>

6.2.1.7 On the new window select **Cyclone 10 LP** as Device family and **10CL055Y** as Device name.

vice family		Device name	
APEX20K		10CL006Y	New
Arria 10		10CL006Z	
Arria GX		10CL010Y	Import
Arria II GX		10CL010Z	Export
Arria II GZ		10CL016Y	
Arria V		10CL016Z	Edit
Arria V GZ		10CL025Y	Remove
Cyclone Cyclone		10CL025Z	Kentove
Cyclone 10 LP		10CL040Y	Uncheck All
Cyclone II		10CL040Z	
Cyclone III		10CL055Y	
Cyclone III LS		10CL055Z	
Cyclone IV E		10CL080Y	
Cyclone IV GX		10CL080Z	
Cyclone V		10CL120Y	
HardCopy	*	10011207	v

- 6.2.1.8 Click **OK** to add device to Flash Loader.
- 6.2.1.9 Select **SOF Data** under Input files to convert and click on **Add File...** button.
- 6.2.1.10 Go to <project_directory>/output_files/ and open top.sof.
- 6.2.1.11 Make sure that your settings are same as the picture below and if everything is correct.

File Tools Window Search altera.com Specify the input files to convert and the type of programming file to generate. You can also import input file information from other files and save the conversion setup information created here for future use. Conversion setup files	Convert Programming	g File - C:/CYC1000/CY	C1000_blinky/top - top				-		×
Specify the input files to convert and the type of programming file to generate. You can also import input file information from other files and save the conversion setup information created here for future use. Conversion setup files Output programming file Programming file type JTAG Indirect Configuration File (JiC) Options/Boot info Configuration device: EPCQ16A Mode: Active Serial Options/Boot info Configuration device: EPCQ16A Mode: Active Serial Context files/output_file.jiC Advanced Remote/Local update difference file: NONE Create CvP files (Generate output_file.map) Create CvP files (Generate output_file.map) Create CvP files (Generate output_file.map) Create CvP files (Generate output_file.map) Create CvP files (Generate output_file.map) Add Hex Data Add Sof Page Y Flash Loader 10CL025Y Y SOF Data V SOF Data Page_0 Context Con	File Tools Window						Search alter	ra.com	6
Open Conversion Setup Data Save Conversion Setup Output programming file Programming file type: Programming file type: Options/Boot info Configuration device: EPCQ16A Mode: Active Serial Imput files/output_file.jic Advanced Remote/Local update difference file: MONE Create Memory Map File (Generate output_file.map) Create CvP files (Generate output_file.periph.jic and output_file.core.rbf) Create config data RPD (Generate output_file_auto.rpd) Input files to convet File/Data area Properties Start Address Add Hex Data Add Siof Page Add File. Y Flash Loader 10CL025Y Y SOF Data Page_0 Vaporties Close Up Down Properties Close	Specify the input files to a You can also import inpu future use. Conversion setup files	convert and the type of t file information from	programming file to gen other files and save the co	erate. onversion setu	p information cre	ated here for			
Output programming file Programming file type JTAG Indirect Configuration File (jic) Options/Boot info Configuration device: EPCQ16A Mode: Active Serial File name: output_files/output_file.jic	Op	en Conversion Setup D	ata		Save Co	nversion Setu	ıp		
Programming file type JTAG Indirect Configuration File (jic) Options/Boot info Configuration device: EPCQ16A Mode: Active Serial Imput files/output_file.jic Advanced Remote/Local update difference file: NONE Create Memory Map File (Generate output_file.map) Create CVP files (Generate output_file.periph.jic and output_file.core.rbf) Create CVP files (Generate output_file_auto.rpd) Input files to convert File/Data area Properties Start Address Add Hex Data Add File_ Remove Up Down Properties Consof 10CL025YU256 Generate Close Help Entert Close Help Close Close Close Close Close Close Close Meduto Close Close Close Close Close Close Close Close<	Output programming fi	le							
Options/Boot info Configuration device: EPCQ16A Mode: Active Serial File name: output_files/output_file.jic Advanced Remote/Local update difference file: NONE Image: Create Memory Map File (Generate output_file.map) Image: Create CvP files (Generate output_file.periph.jic and output_file.core.rbf) Image: Create CvP files (Generate output_file_auto.rpd) Input files to convert File/Data area Properties Start Address Add Hex Data Y Flash Loader 10CL025Y Add File Remove Up Down Properties Sauto> Coreate Up Down Properties 10CL025YU256 Generate Up Down Properties Up Down Properties Close Help	Programming file type:	JTAG Indirect Config	uration File (.jic)						•
File name: output_file/jic Advanced Remote/Local update difference file: NONE Create Memory Map File (Generate output_file.map) Create CvP files (Generate output_file.auto.rpd) Input files to convert File/Data area Properties Start Address Add Hex Data Add Sof Page Add File Y SOF Data Page_0 top.sof 10CL025YU256 Remove Up Down Properties Cauto> top.sof 10CL025YU256 End Close Help	Options/Boot info	Configuration device:	EPCQ16A	•	Mode:	Active S	erial		-
Advanced Remote/Local update difference file: NONE Image: Comparison of the second s	File name:	output_files/output_f	ile.jic						
	Advanced	Remote/Local update	Remote/Local update difference file: NONE						
File/Data area Properties Start Address Add Hex Data Flash Loader 10CL025Y SOF Data Page_0 auto> Add File Remove Up Down Properties Generate Close Help		Create Memory Ma Create CvP files (G Create config data	p File (Generate output_fi enerate output_file.periph RPD (Generate output_file	ile.map) .jic and output a_auto.rpd)	_file.core.rbf)				
File/Data area Properties Start Address * Flash Loader 10CL025Y * SOF Data Page_0 top.sof 10CL025YU256 Up Down Properties Generate Close Help	Input files to convert		Desmation	Charles And June				Add Hay F)ata
Y SOF Data Page_0 <auto> Add File Remove Up Down Properties Down Properties Close Help</auto>	File/Data : Flash Loader 10CL025Y	area	Properties	Start Addre	:55			Add Hex L Add Sof P	ata age
top.sof 10CL025YU256 Remove Up Down Properties Generate Close Help	✓ SOF Data	Page	_0	<auto></auto>				Add File	
Up Down Properties Generate Close Help	top.sof	10CL	025YU256					Remove	8
Down Properties Generate Close Help								Up	
Properties Generate Close Help								Down	
Generate Close Help								Properti	es
Generate Close Help									
						Generate	Close	Hel	p

Cyclone 10 LP RefKit User Guide

6.2.1.12 Click Generate.

6.2.1.13 Click **OK** on the successful file generation notification and **close** Convert Programming File window.

- 6.2.2 Device Programming
- 6.2.2.1 Open Programmer.
- 6.2.2.2 Select output_files/top.sof and click Change File... button.

6.2.2.3 Go to <project_directory>/output_files/ and open output_file.jic.

When you add the .jic file, the Programmer will automatically update the JTAG chain and put EPCQ-A flash memory.

6.2.2.4 Make sure the Programmer shows the correct file and correct parts in the JTAG chain and check the Program/Configure checkbox.

6.2.2.5 Click **Start** to configure EPCQ-A. The programming could take a while.

6.2.2.6 When the programming is finished, the C10LP RefKit should be able to keep its configuration data even after powered off.

At this point our program is stored in the EPCQ-A flash memory, but the Cyclone 10 LP current configuration is the Serial Flash Loader which is responsible for programming configuration flash memory. We can simply reconfigure the FPGA with our program by pushing RESET button which will reset the FPGA and automatically loads the configuration from EPCQ-A.

6.3 Testing the Design

Does not matter which way the C10LP RefKit was configured, the results should be the same for both methods, with the only difference being if configuration is retained after power off.

On the board by default, the LEDS should now toggle in a slow counting sequence.

Push and hold the S1 USER_BTN (central joystick button) to see that the LEDs will now toggle in a very fast counting sequence. USER_BTN is on the side of the LEDs.

Releasing the USER_BTN, will make the LEDs toggle at a slower rate as before.

Chapter 7 - Common Issues and Fixes

1) **Issue:** In some rare cases when using Windows 10 operating system, the programmer DLL is not properly loaded/unloaded, causing the Quartus Programmer to not detect the Arrow USB Programmer2.

Solution: Restart the Altera JTAG Server using the Services application of Windows.

Cyclone 10 LP RefKit User Guide

WDW
WDW

Chapter 8 - Appendix

8.1 Revision History

Version	Change Log	Date of Change
V1.0	Initial Version	17/02/2022

Cyclone 10 LP RefKit User Guide



8.2 Legal Disclaimer

ARROW ELECTRONICS

EVALUATION BOARD LICENSE AGREEMENT

By using this evaluation board or kit (together with all related software, firmware, components, and documentation provided by Arrow, "Evaluation Board"), You ("You") are agreeing to be bound by the terms and conditions of this Evaluation Board License Agreement ("Agreement"). Do not use the Evaluation Board until You have read and agreed to this Agreement. Your use of the Evaluation Board constitutes Your acceptance of this Agreement.

PURPOSE

The purpose of this evaluation board is solely intended for evaluation purposes. Any use of the Board beyond these purposes is on your own risk. Furthermore, according the applicable law, the offering Arrow entity explicitly does not warrant, guarantee or provide any remedies to you with regard to the board.

LICENSE

Arrow grants You a non-exclusive, limited right to use the enclosed Evaluation Board offering limited features only for Your evaluation and testing purposes in a research and development setting. Usage in a live environment is prohibited. The Evaluation Board shall not be, in any case, directly or indirectly assembled as a part in any production of Yours as it is solely developed to serve evaluation purposes and has no direct function and is not a finished product.

EVALUATION BOARD STATUS

The Evaluation Board offers limited features allowing You only to evaluate and test purposes. The Evaluation Board is not intended for consumer or household use. You are not authorized to use the Evaluation Board in any production system, and it may not be offered for sale or lease, or sold, leased or otherwise distributed for commercial purposes.

OWNERSHIP AND COPYRIGHT

Title to the Evaluation Board remains with Arrow and/or its licensors. This Agreement does not involve any transfer of intellectual property rights ("IPR) for evaluation board. You may not remove any copyright or other proprietary rights notices without prior written authorization from Arrow or it licensors.

RESTRICTIONS AND WARNINGS

Before You handle or use the Evaluation Board, You shall comply with all such warnings and other instructions and employ reasonable safety precautions in using the Evaluation Board. Failure to do so may result in death, personal injury, or property damage.

You shall not use the Evaluation Board in any safety critical or functional safety testing, including but not limited to testing of life supporting, military or nuclear applications. Arrow expressly disclaims any responsibility for such usage which shall be made at Your sole risk.

WARRANTY

Arrow warrants that it has the right to provide the evaluation board to you. This warranty is provided by Arrow in lieu of all other warranties, written or oral, statutory, express or implied, including any warranty as to merchantability, non-infringement, fitness for any particular purpose, or uninterrupted or error-free operation, all of which are expressly disclaimed. The evaluation board is provided "as is" without any other rights or warranties, directly or indirectly.

You warrant to Arrow that the evaluation board is used only by electronics experts who understand the dangers of handling and using such items, you assume all responsibility and liability for any improper or unsafe handling or use of the evaluation board by you, your employees, affiliates, contractors, and designees.

Cyclone 10 LP RefKit User Guide



wow

LIMITATION OF LIABILITIES

In no event shall Arrow be liable to you, whether in contract, tort (including negligence), strict liability, or any other legal theory, for any direct, indirect, special, consequential, incidental, punitive, or exemplary damages with respect to any matters relating to this agreement. In no event shall arrow's liability arising out of this agreement in the aggregate exceed the amount paid by you under this agreement for the purchase of the evaluation board.

IDENTIFICATION

You shall, at Your expense, defend Arrow and its Affiliates and Licensors against a claim or action brought by a third party for infringement or misappropriation of any patent, copyright, trade secret or other intellectual property right of a third party to the extent resulting from (1) Your combination of the Evaluation Board with any other component, system, software, or firmware, (2) Your modification of the Evaluation Board, or (3) Your use of the Evaluation Board in a manner not permitted under this Agreement. You shall indemnify Arrow and its Affiliates and Licensors against and pay any resulting costs and damages finally awarded against Arrow and its Affiliates and Licensors or agreed to in any settlement, provided that You have sole control of the defense and settlement of the claim or action, and Arrow cooperates in the defense and furnishes all related evidence under its control at Your expense. Arrow will be entitled to participate in the defense of such claim or action and to employ counsel at its own expense.

RECYCLING

The Evaluation Board is not to be disposed as an urban waste. At the end of its life cycle, differentiated waste collection must be followed, as stated in the directive 2002/96/EC. In all the countries belonging to the European Union (EU Dir. 2002/96/EC) and those following differentiated recycling, the Evaluation Board is subject to differentiated recycling at the end of its life cycle, therefore: It is forbidden to dispose the Evaluation Board as an undifferentiated waste or with other domestic wastes. Consult the local authorities for more information on the proper disposal channels. An incorrect Evaluation Board disposal may cause damage to the environment and is punishable by the law.

Cyclone 10 LP RefKit User Guide