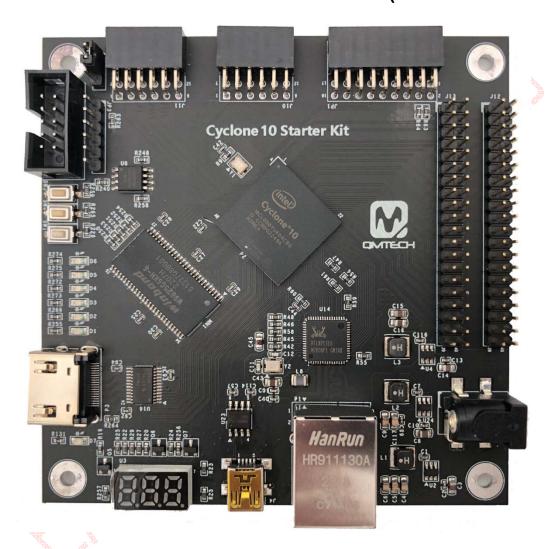
QMTECH CYCLONE10 STARTER KIT

USER MANUAL(QUARTUS 18.1)



Preface

The QMTECH® Cyclone 10 Starter Kit uses Altera 10CL080 device to demonstrate the industry's lowest system cost and power, along with performance levels that make the device family ideal for differentiating your high-volume applications. All Altera Cyclone® 10 LP FPGAs require only two core power supplies for operation, simplifying your power distribution network and saving you board costs, board space, and design time. The flexibility of the Altera Cyclone® 10 LP FPGA enables you to design in a smaller, lower cost device, lowering your total system costs.

Table of Contents

| 1. QUARTUS PRIME 18.1 INSTALLATION | | |
|--------------------------------------|-----------------------|----|
| 2. FPGA PROJECT COMPILE, *.SOF DOWNL | OAD AND *.JIC PROGRAM | 7 |
| 2.1 Create New Project | | |
| 2.2 COMPILE THE PROJECT | | 12 |
| 2.3 PIN ASSIGNMENT | | 13 |
| 2.4 DOWNLOAD *.SOF INTO FPGA | | 15 |
| 2.5 PROGRAM *.JIC INTO SPI FLASH | | |
| 3. SIGNALTAP II LOGIC ANALYZER | | |
| 4. REFERENCE | | |
| F DEVISION | | 2- |



1. Quartus Prime 18.1 Installation

The revolutionary Altera Quartus® Prime Design Software includes everything you need to design for Altera FPGAs, SoCs, and CPLDs from design entry and synthesis to optimization, verification, and simulation. Dramatically increased capabilities on devices with multi-million logic elements are providing designers with the ideal platform to meet next-generation design opportunities.

The Altera Quartus® Prime Software design flow comprises of the following high-level steps:



The Quartus Prime software version 18.1 supports the following device families: Stratix IV, Stratix V, Arria II, Arria V, Arria V GZ, Arria 10, Cyclone 10 LP, Cyclone IV, Cyclone V, MAX II, MAX V, and MAX 10 FPGA. Below image shows the startup UI of Quartus II Prime 18.1:

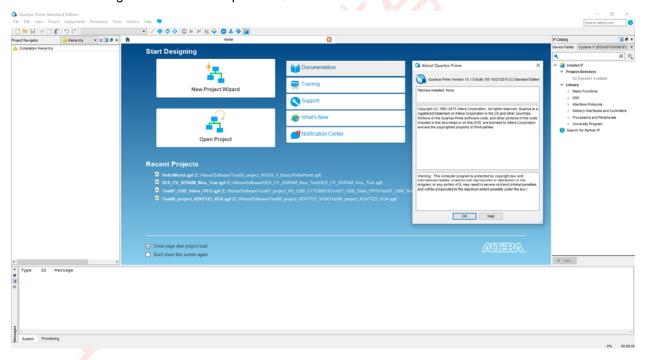


Figure 1-1. Quartus II Prime 18.1

After the Quartus II Prime 18.1 is correctly installed, users still need to install the device package from Altera official website. Below lists the download center address:

https://www.intel.com/content/www/us/en/programmable/downloads/download-center.html

In the Intel Download Center website, select the tab of 'Select by Device' and then all the available device packages will be listed as below image. The device used in this user manual is Cyclone10 LP series and the detailed chip part number is 10CL080YU484C8G, so please download the device package for Quartus II 18.1: cyclone10lp-18.1.0.625.qdz.



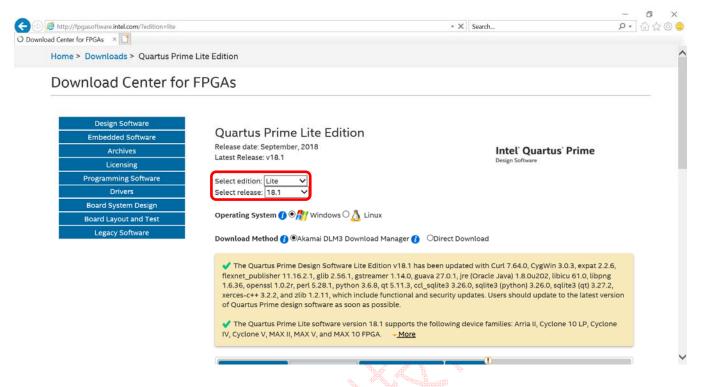


Figure 1-2. Download Device Package

Open Quartus II 18.1, Click Tools → Install Device and then select the downloaded device package:

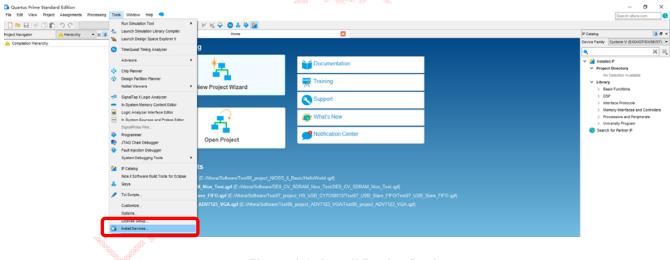


Figure 1-3. Install Device Package



Below window will pop up and click Next:

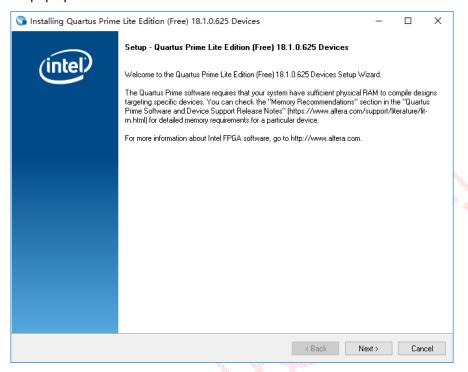


Figure 1-4. Install Device Package

Choose the Download Directory where contains the cyclone10lp-18.1.0.625.qdz file:

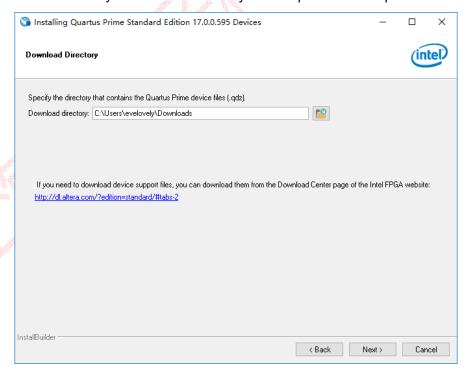


Figure 1-5. Choose Device Package



Choose the device package needs to be installed and click Next:

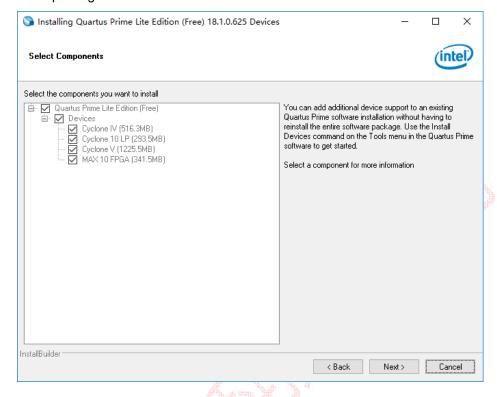


Figure 1-6. Install the Device Package

User could also install the device package by using Quartus II Prime 18.1 Device Installer directly:

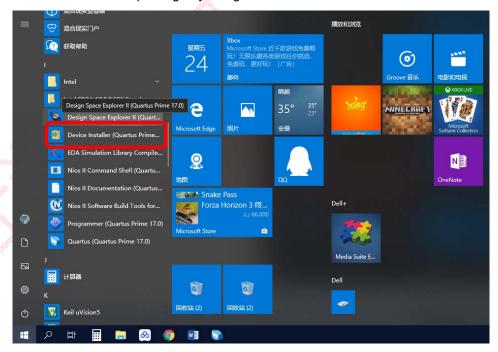


Figure 1-7. Device Installer



2. FPGA Project Compile, *.sof Download and *.jic Program

2.1 Create New Project

Click 【File】 → 【New Project Wizard…】 to create a new project:

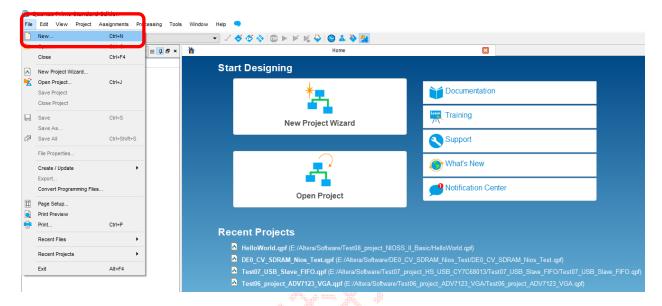


Figure 2-1. Create New Project

Choose [New Quartus Prime Project]:

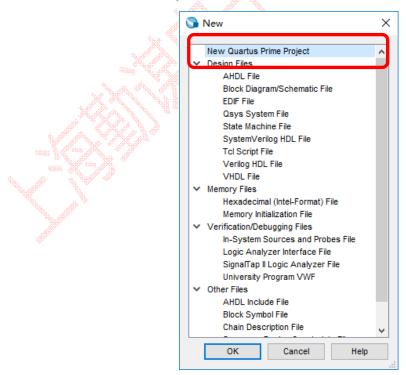


Figure 2-2. New Quartus Prime Project



In below [New Project Wizard] page, choose Next:

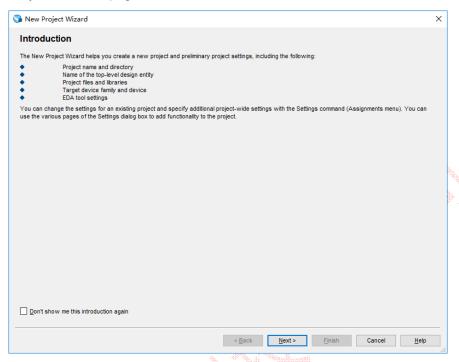


Figure 2-3. New Project Wizard

Set the target working folder below 【What is the working directory for this project?】. Set the new project name below 【What is the name of this project?】. And finally set the example project name: Test01_Project_LED shown as below.

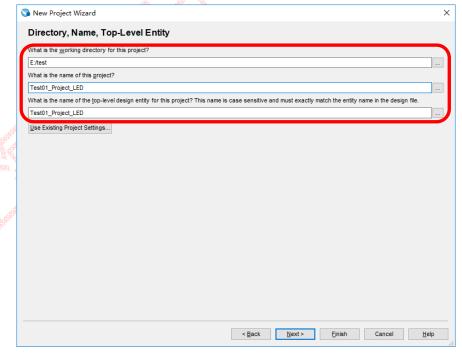


Figure 2-4. Set Working Directory and Project Name



Select 【Empty Project】 and then click Next:

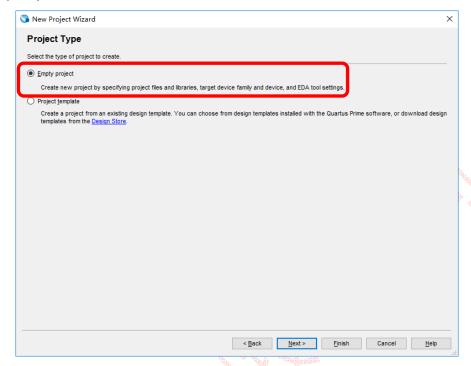


Figure 2-5. Create Empty Project

If user already has some source code, please add all these necessary files in this step:

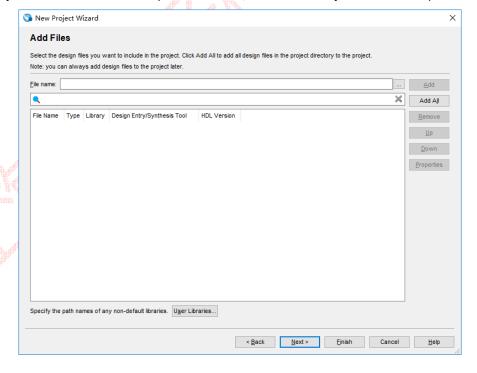


Figure 2-6. Add Source Code



Choose the FPGA Chip number: 10CL080YU484C8G

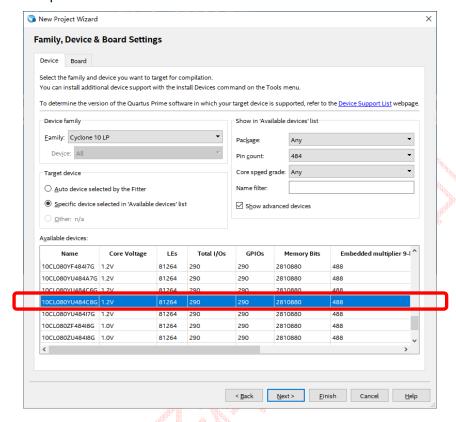


Figure 2-7. Select Device

Summary page will be shown and click [Finish] if there's nothing needs to be changed:

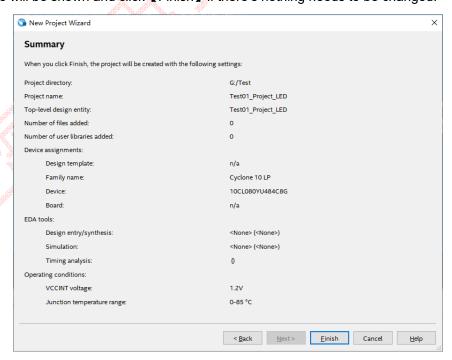


Figure 2-8. Project Summary Page



After the Empty Project created, below image will be shown:

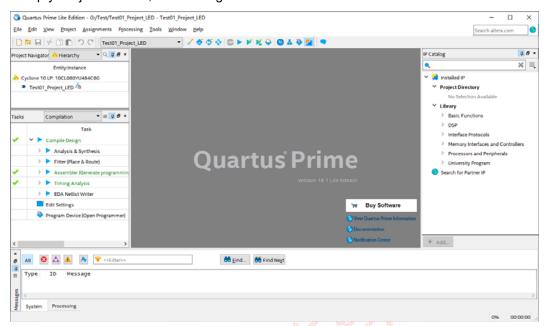


Table 2-1. Empty Project

Users may add example source file Test01_Project_LED.v into this Empty Project shown as below:

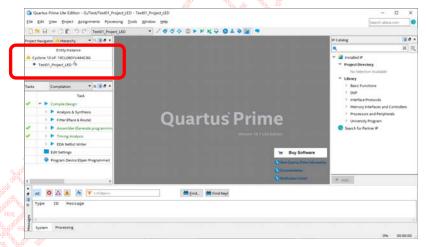


Table 2-2. Add Source File

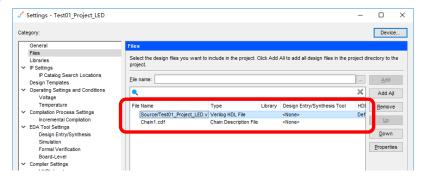


Table 2-3. Add Source File



🕥 Quartus Prime Lite Edition - E:/Altera/Cyclone_10_10CL016/SW_080/Test01_LED/Test01_LED - Test01_LED <u>F</u>ile <u>E</u>dit <u>V</u>iew <u>Project Assignments Processing <u>T</u>ools <u>W</u>indow <u>H</u>elp</u> □ 🛗 🚽 🖺 🗈 🤊 ் Test01_LED 🔻 🗹 🎸 🌣 🕲 ▶ 🚩 🐇 🗳 🎎 🥌 rroject Navigator hierarchy Q I 5 × IP Catalog Test01 LED v Entity.Instance

Cyclone 10 LP. 10CL080YU484C8G

Testol_LED Tb

Tasks

Compilation

Task

Task | 📳 | 🍪 (7) | 蒜 蒜 | 🗈 🗗 怆 | 🛈 🖫 | 🐉 🗏 ×≡ • Entity:Instance Installed IP → Project Directory // // Revision: // Revision 0.01 - File Created // Additional Comments: No Selection Available Library > Basic Functions input sys_clk,
input sys_rst_n,
output[5:0] led_1
); > DSP > Interface Protocols > Memory Interfaces and Controllers parameter DLY_CNT = 32'd50000000; parameter HALF_DLY_CNT = 32'd25000000; Processors and Peripherals > University Program Search for Partner IP //counter control
always@(posedge sys_clk or negedge sys_rst_n)
Blegin
if(lsys_rst_n)
begin begin st_n)
begin st_n)
count <= 32'd0;
end
else if(count == DLY_CNT)
begin
count <= 32'd0;
end
else € Find... € Find Next Type ID Message

After the newly added source file loaded into project, user can view the source code shown as below:

Figure 2-9. View of Source Code

2.2 Compile the Project

System Processing

Users could use the button [Start Compilation – Ctrl + L] shown in below image to compile the project:

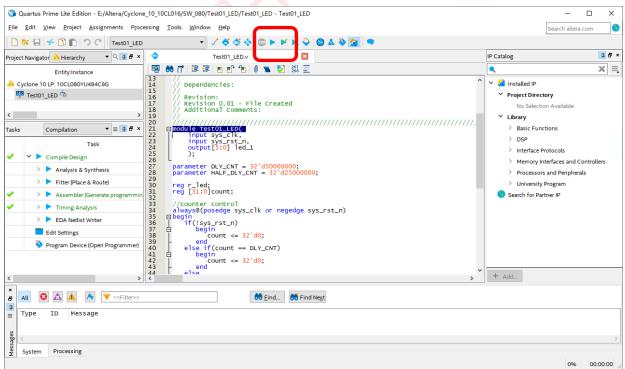


Figure 2-10. Compilation



0% 00:00:00

There will be compilation report after the compile finished, in which shows the info like logical element resource usage, how many PLLs are used, etc. Below image shows an example Compilation Report:

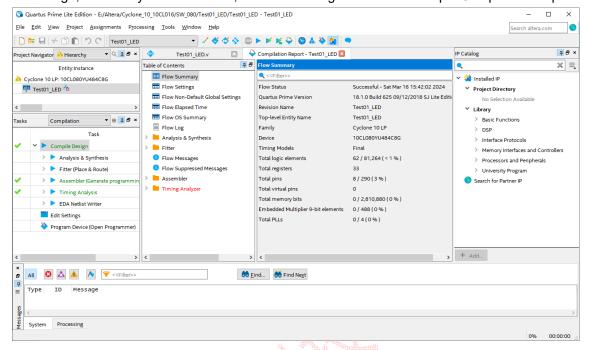


Figure 2-11. Compilation Report

2.3 PIN Assignment

There are several ways to assign the Pins for the example project. Method 1: Choose 【Assignment】→
【Pin Planner】:

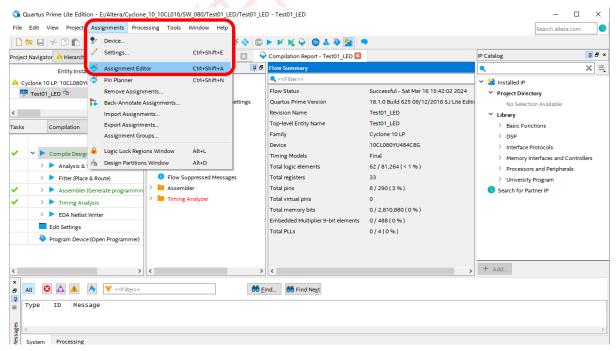


Figure 2-12. Pin Planner



Fin Planner - E:/Altera/Cyclone_10_10CL016/SW_080/Test01_LED/Test01_LED - Test01_LED File Edit View Processing Tools Window Help ДÐ× ДÐХ Cyclone 10 LP - 10CL080YU484C8G Report not available Symbol Pin Type User I/O User assigned I... Fitter assigned I... Unbonded pad Reserved pin Other configura. 6 DEV_OE Tasks ηÐ× R DEV CLR Œ Early Pin Planning $\langle n \rangle$ DIFF n 3 Early Pin Planning... DIFF_p Run I/O Assignment Analysi CLK_n Export Pin Assignments... CLK p t 🚰 ∨ 🦥 Edit: X Filter: Pins: all ä Differentia out led_1[5]
out led_1[4]
out led_1[3]
out led_1[2] ø Output B3_N1 PIN_V10 3.3-V LVTTL 8mA (default) 2 (default) Output PIN_U9 B3_N2 PIN_U9 3.3-V LVTTL 8mA (default) 2 (default) PIN_V9 3.3-V LVTTL 8mA (default) 2 (default) Output PIN_V9 B3_N1 8mA (default) Output ed_1[1] Output PIN_V5 B3_N2 PIN_V5 3.3-V LVTTL 8mA (default) 2 (default) 3.3-V LVTTL 8mA (default) " led_1[0] Output PIN Y4 B3 N2 PIN Y4 2 (default) sys_clk PIN_G1 PIN_G1 8mA (default) Input sys_rst_r Input PIN_P3 B2_N1 PIN_P3 3.3-V LVTTL 8mA (default)

Below image shows PIN settings for this test example:

Figure 2-13. PIN Assignment

Method 2: Prepare a *.csv file from other project, then use 【Assignment】 → 【Import Assignment】 to import the existing *.csv file to allocate the Pin assignment:

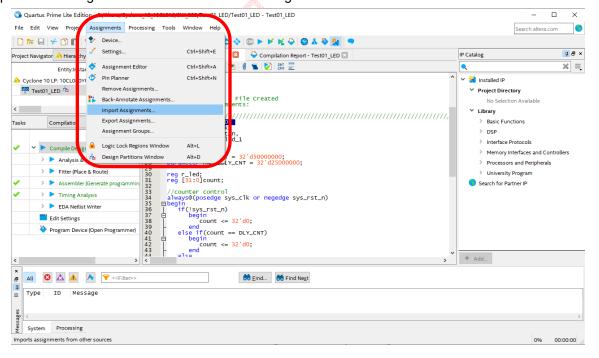


Figure 2-14. Import Assignment



00:00:00

2.4 Download *.sof into FPGA

After the test example correctly compiled, the Quartus will generate a *.sof file which could be directly loaded into FPGA to check whether implemented functions perform as expected. User could use 【Tools】 → 【Programmer】 to start a new download:

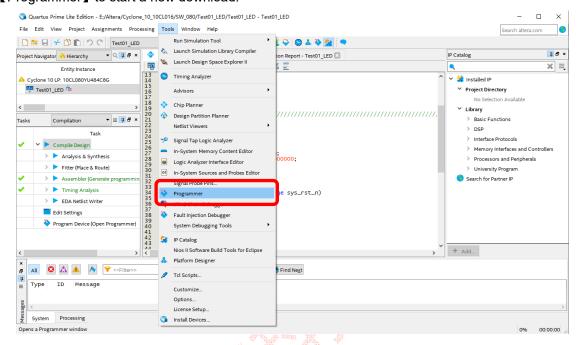


Figure 2-15. Programmer

Make sure the USB Blaster's cable are correctly connected to FPGA's JTAG port before using Programmer to download *.sof file. Then click [Auto Detect] to check the hardware setup is okay or not:

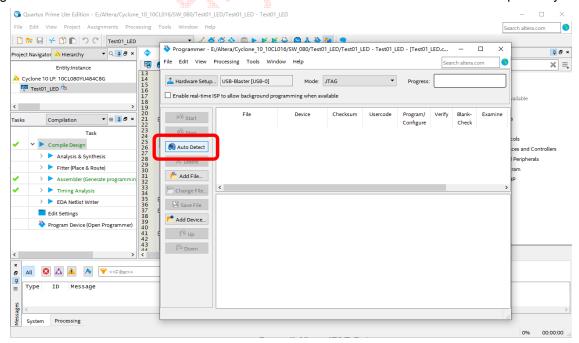


Figure 2-16. JTAG Setup



Programmer - E:/Altera/Cyclone_10_10CL016/SW_080/Test01_LED/Test01_LED - Test01_LED - [Test01_LED.c... <u>F</u>ile <u>E</u>dit <u>V</u>iew <u>Pr</u>ocessing <u>T</u>ools <u>W</u>indow <u>H</u>elp Search altera.com A Hardware Setup... USB-Blaster [USB-0] Mode: JTAG Enable real-time ISP to allow background programming when available Program/ ▶^V Start Configure Stop 10CL080Y 00000000 Auto Detect × Delete Add File... Change File... Save File Add Device... 1[™] Up ↓[™] Down 10CL080Y TDO

Below image shows the FPGA has been detected by the Programmer:

Figure 2-17. Detect FPGA

Users click [None] column to choose the *.sof file to be loaded into FPGA.

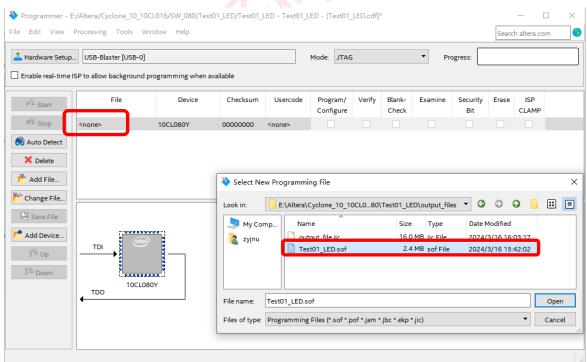
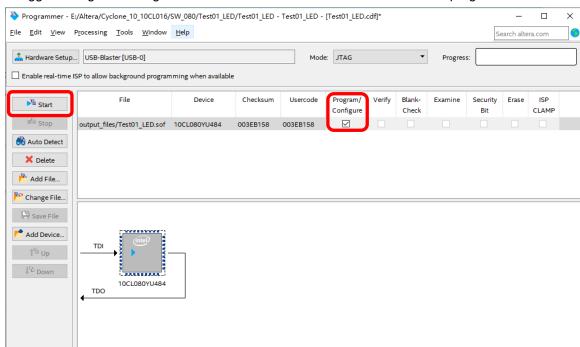


Figure 2-18. Choose *.sof File





Then toggle [Program/Configure] and click the [Start] button to start a new program:

Figure 2-19. Program *.sof

If the *.sof file is correctly programed, the Progress bar will show info like: 100%(Successful). Then users could check whether the LEDs on FPGA board blinking or not.

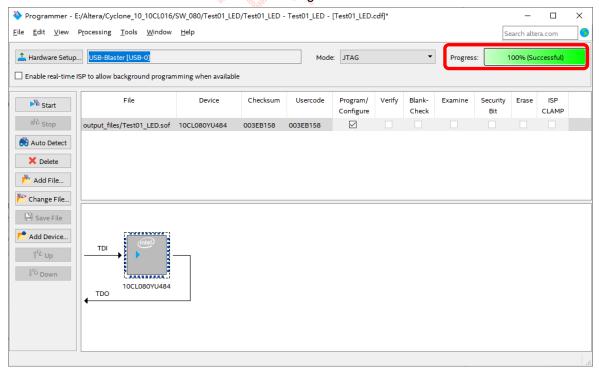


Figure 2-20. Program Successful



2.5 Program *.jic into SPI Flash

QMTECH Cyclone10 Starter Kit has mounted an external SPI Flash with 16MB capacity. The hardware design chooses Active Serial x 1 method to make the FPGA could boot up from external SPI Flash after power on. In this section, it describes how to program eternal SPI Flash through JTAG port. The SPI Flash is non-volatile device which means the programmed *.jic file will never lose its content after power down.

The SPI Flash programing file *.jic is converted by *.sof file described in previous chapter. So make sure *.sof could be correctly running on FPGA before performing below steps. Step1: choose the Quartus II 18.1file convert tool by click 【File】 → 【Convert Programming File】:

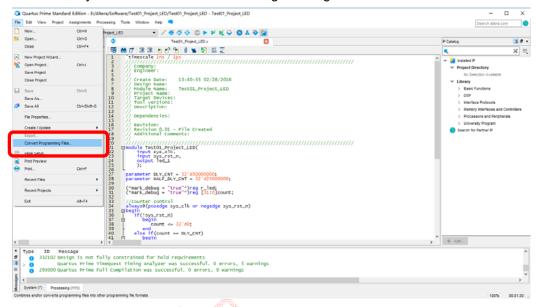


Figure 2-21. Convert Programming File Tool

Change the settings following below figure: choose EPCQ128, generated file name output_file.jic, etc.

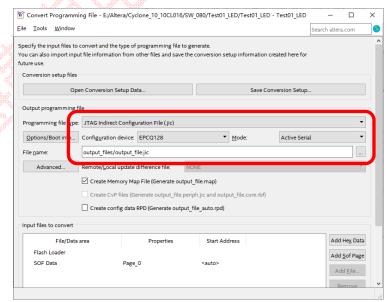


Figure 2-22. Configure Convert Programming File Tool



Convert Programming File - E:/Altera/Cyclone_10_10CL016/SW_080/Test01_LED/Test01_LED - Test01_LED 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 Advanced Options Conversion setup files ☑ Disable EPCS/EPCQ ID check Open Conversion Setup Data... ☑ Disable AS mode CONF DONE error check Output programming file Program length count adjustment: Programming file type: JTAG Indirect Configuration File (.jic) Post-chain bitstream pad bytes: default Options/Boot info... Configuration device: EPCQ128 Post-device bitstream pad bytes: default output_files/output_file.jic 1 🔻 Bitslice padding value: Advanced... Remote/Local update difference file: NONE QSPI Flash single IO mode dummy clock: Unchangeable ☑ Create Memory Map File (Generate output_file QSPI Flash quad IO mode dummy clock: Unchangeable Create CvP files (Generate output_file.periph.j ОК Cancel Create config data RPD (Generate output_file Input files to convert File/Data area Start Address Add He<u>x</u> Data Flash Loader Add <u>S</u>of Page SOF Data Page_0 <auto> Add File...

Click the 【Advanced…】 option, and set these below two options in the red rectangle in Disable status:

Figure 2-23. Advanced Options

Select [Flash Loader] and then click [Add Device] button:

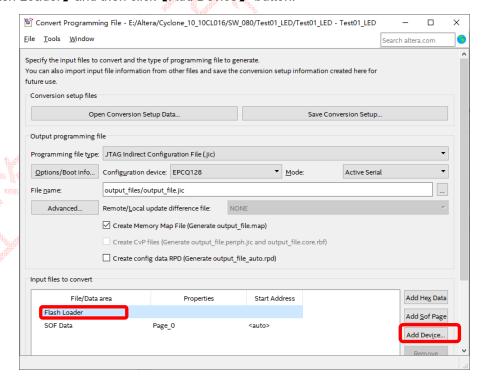


Figure 2-24. Flash Loader



Choose the target Flash Loader device: 10CL080Y:

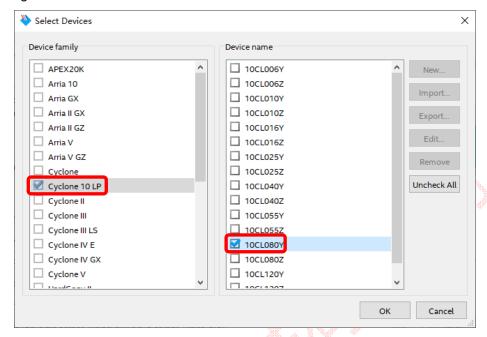


Figure 2-25. Flash Loader for 10CL080Y

Select 【SOF Data】 and then choose 【Add File...】 to add the verified *.sof file. And then click 【Generate】 to generate the output_file.jic file:

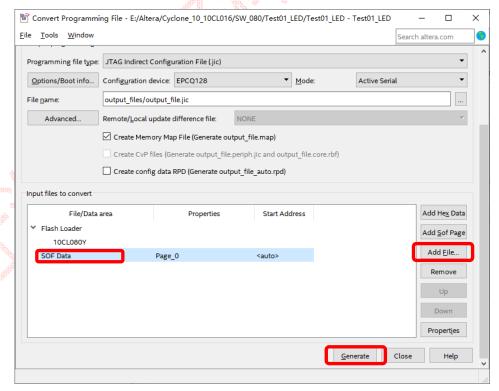


Figure 2-26. Generate *.jic File



After the output_file.jic correctly generated, run the $[Tools] \rightarrow [Programmer]$. And then click [Add File...] to choose the output file.jic.

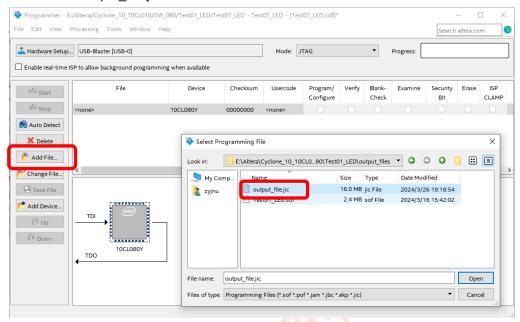


Figure 2-27. Choose *jic File

Toggle 【Program/Configure】 and then click 【Start】 button to program the external SPI Flash. Program status will be shown in the 【Progress】 bar. After the *.jic correctly programmed, user may repower on the board to check whether the FPGA could boot from external SPI Flash.

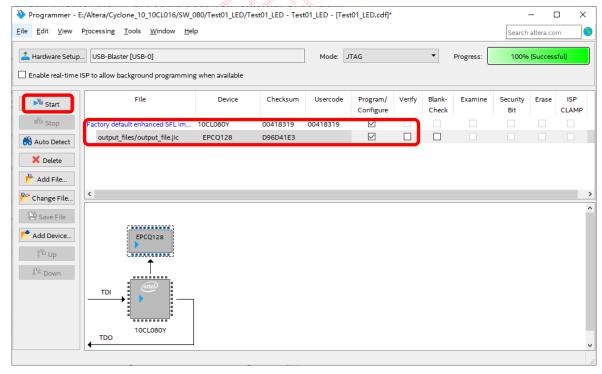


Figure 2-28. Program *.jic



3. SignalTap II Logic Analyzer

The SignalTap II Embedded Logic Analyzer is a system-level debugging tool that captures and displays signals in circuits designed for implementation in Altera's FPGAs. The user is expected to have access to a computer that has Quartus II 18.1 software installed. The detailed example in this chapter was obtained using Quartus II version 18.1, but newer versions of the software can also be used.

After successfully compiling the Test04_SDRAM project and setting pin assignments, select SignalTap II Logic Analyzer from the tools dropdown menu (as shown below). Ensure the JTAG programmer (USB Byte Blaster) is connected between the board and the computer.

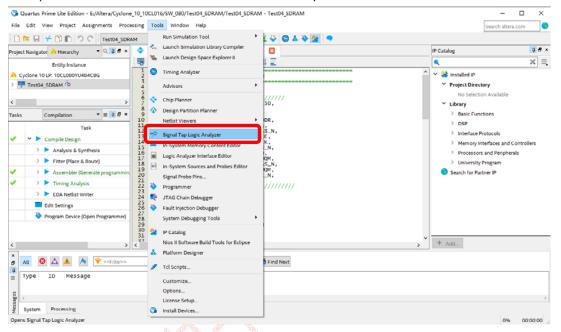


Figure 3-1. Open SignalTap II Logic Analyzer

Below image shows the UI of the SignalTap II:

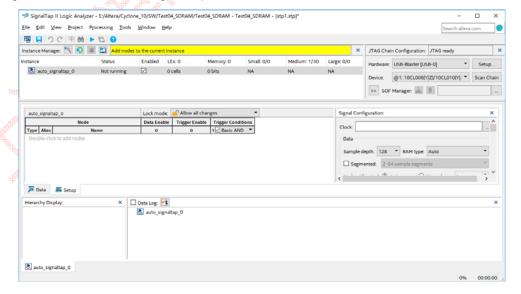


Figure 3-2. SignalTap II Logic Analyzer UI



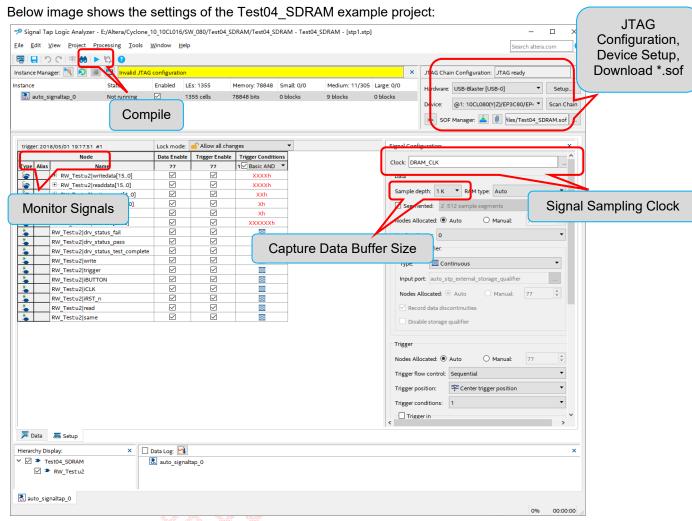


Figure 3-3. SignalTap II Logic Analyzer Main Window

Double click the 【Node】column shown in the above image. Below window will pop up and user clicks the 【List】button to add the signals need to be monitored:

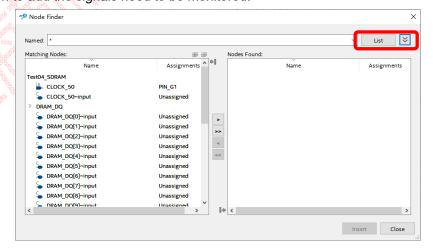


Figure 3-4. Set Capturing Signals



All the signals added in the above step will be displayed in the 【setup 】 page. Please also select the signal sampling clock 【DRAM CLK】:

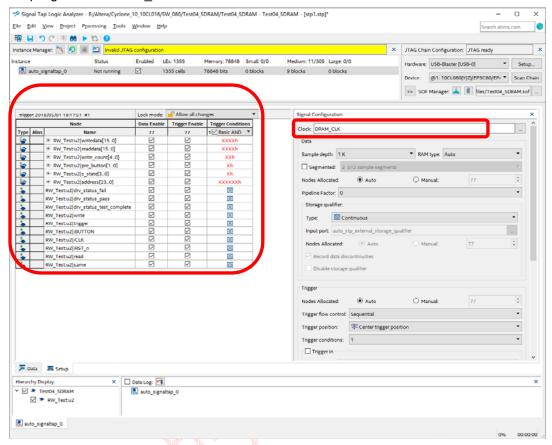


Figure 3-5. Sampling Signals and Clocks

User needs to click the 【Compile】 button shown in the below image to recompile the whole project. Then user may download the newly compiled *.sof into FPGA.

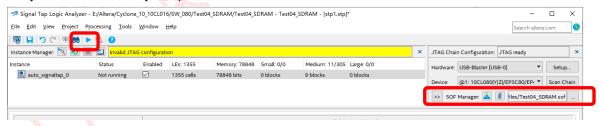


Figure 3-6. Compile the SignalTap II Project

User could click the button 【AutoRun Analysis】 or button 【Run Analysis】 to start the waveform capture:

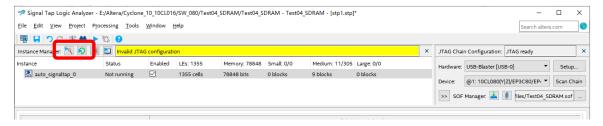


Figure 3-7. Start Capture



Below two images show the SDRAM data write and SDRAM data read:



Figure 3-8. Waveform for Writing Data into SDRAM



Figure 3-9. Waveform for Reading Data from SDRAM



Reference 4.

- [1] QMTECH-Cyclone10-Starter-Kit-V03.pdf
 [2] c10lp-51002.pdf
 [3] c10lp-51003.pdf
 [4] pcg-01021.pdf
 [5] cyclone-10-lp-product-table.pdf
 [6] an800.pdf
 [7] aib-01029.pdf





5. Revision

| Doc. Rev. | Date | Comments |
|-----------|------------|----------------------------|
| 0.1 | 03/07/2019 | Initial Version. |
| 1.0 | 11/07/2019 | Formal Release. |
| 2.0 | 17/01/2021 | Upgraded to V2.0 Hardware. |
| 3.0 | 26/03/2024 | Upgraded to V3.0 Hardware. |



