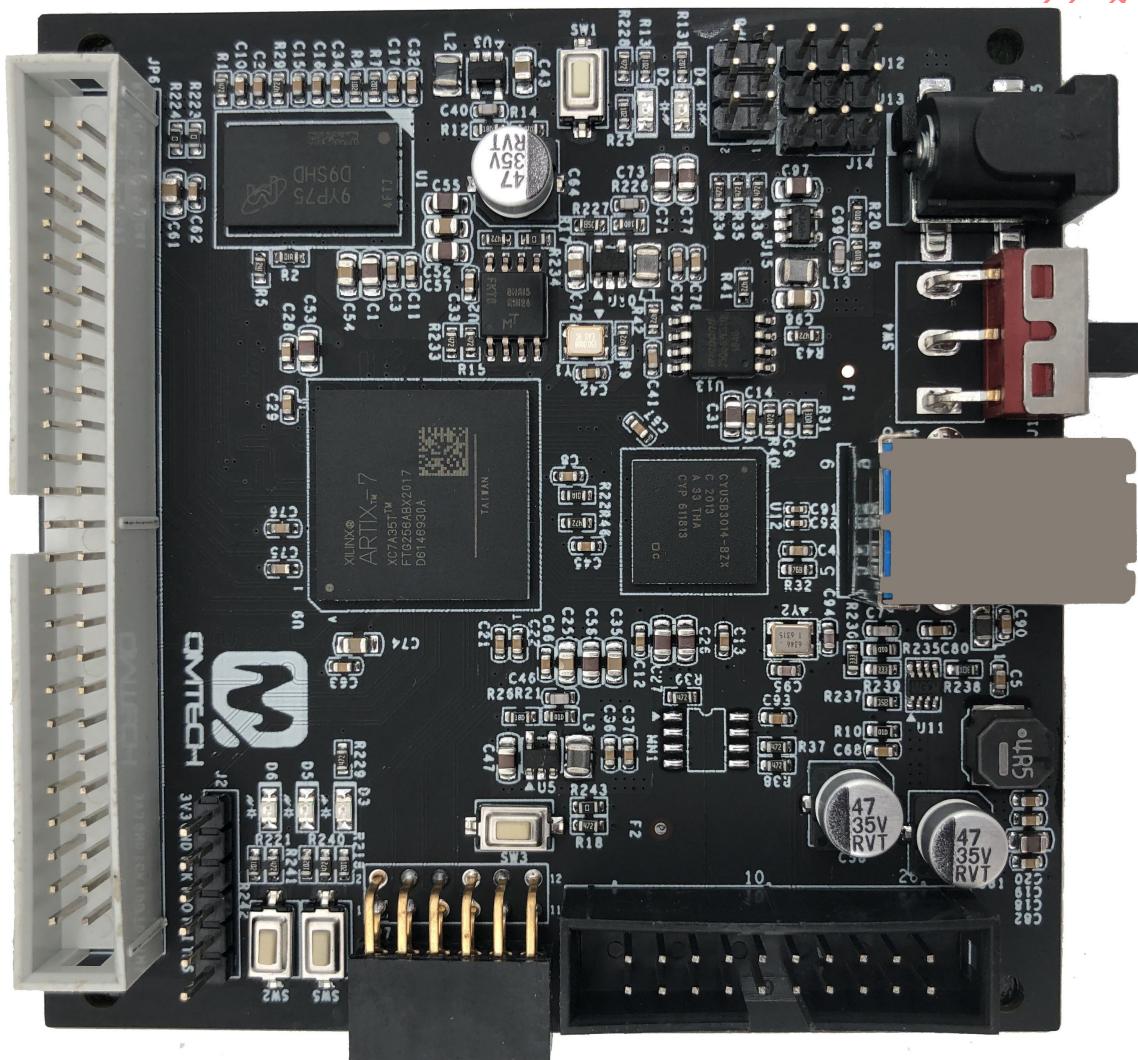


ARTIX7 USB3.0 DEVELOPMENT BOARD

USER MANUAL(VIVADO 2018.3)



Preface

The QMTECH® Artix-7 USB 3.0 development board uses Cypress's EZ-USB® FX3™ high-bandwidth USB 3.0 peripheral controller and Xilinx Artix®-7 device to demonstrate USB 3.0 SuperSpeed technology's high usable bandwidth and great power delivery. This user manual introduces the usage of Xilinx Vivado 2018.3 including how to download the FPGA and program the SPI flash. It also lists the key configurations of the DDR3 memory test project.



QMTECH

Artix-7 USB3.0 Development Board - User Manual(Vivado 2018.3)

Table of Contents

| | |
|---|-----------|
| 1. VIVADO 2018.3 INTRODUCTION..... | 3 |
| 1.1 OVERVIEW..... | 3 |
| 1.2 VIVADO 2018.3 ENVIRONMENT..... | 3 |
| 2. FPGA DOWNLOAD..... | 4 |
| 3. SPI FLASH PROGRAM..... | 13 |
| 4. DDR3 MEMORY TEST..... | 17 |
| 5. REFERENCE..... | 25 |
| 6. REVISION..... | 26 |

上海勸謀電子科技有限公司



1. Vivado 2018.3 Introduction

1.1 Overview

QMTECH Artix-7 USB3.0 Development Board has mounted an 16MB SPI Flash provided by Micron, chip part number is MT25QL128. And the FPGA hardware design allows the FPGA boots from external SPI Flash after power up. The following chapters describe the FPGA download and SPI flash program by using Vivado 2018.3:

- (1) *.bit file downloaded into FPGA, RAM based content will lost during power up stage;
- (2) *.mcs file programmed into external SPI Flash, Flash based content is non-volatile and retained during power up stage.

1.2 Vivado 2018.3 Environment

The test examples contained in the release package which are all developed with Xilinx Vivado 2018.3. Users could download the Vivado 2018.3 from Xilinx official website.

<https://www.xilinx.com/support/download.html>

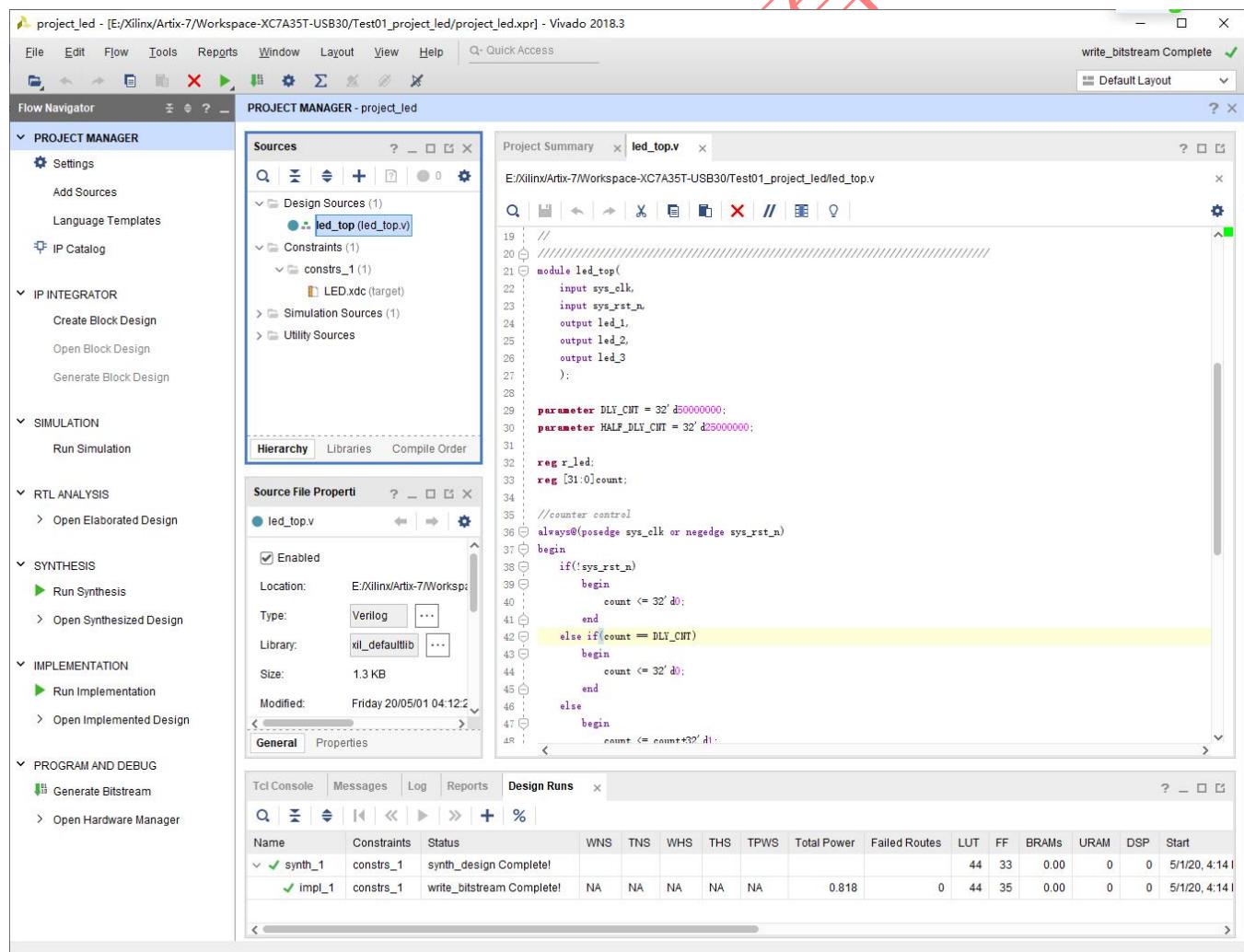


Figure 1-1. Vivado 2018.3 GUI



QMTECH

Artix-7 USB3.0 Development Board

User Manual-V01

2. FPGA Download

Users could download *.bit file directly into FPGA to verify the RTL behavior performs correctly or not. In this section we use the example project Test01_project_led to demonstrate the procedure of downloading led_top.bit into FPGA.

First step is to make the example project Test01_project_led compiled without any error generated. Since this test example is already verified, users could click "Flow Navigator" → "Project MANAGER" → "PROGRAM AND DEBUG" → "Generate Bitstream" to get the led_top.bit generated directly. If users want to test with some customized project, please follow bellow steps to generate the *.bit file.

Users could click the 【Run Synthesis】 button shown in below image highlighted with red rectangle to SYNTHESIS the example project. The SYNTHESIS progress is displayed in the tab of "Design Runs" which is also highlighted in below image. Users could get the compile info in the tabs like "Log", "Message":

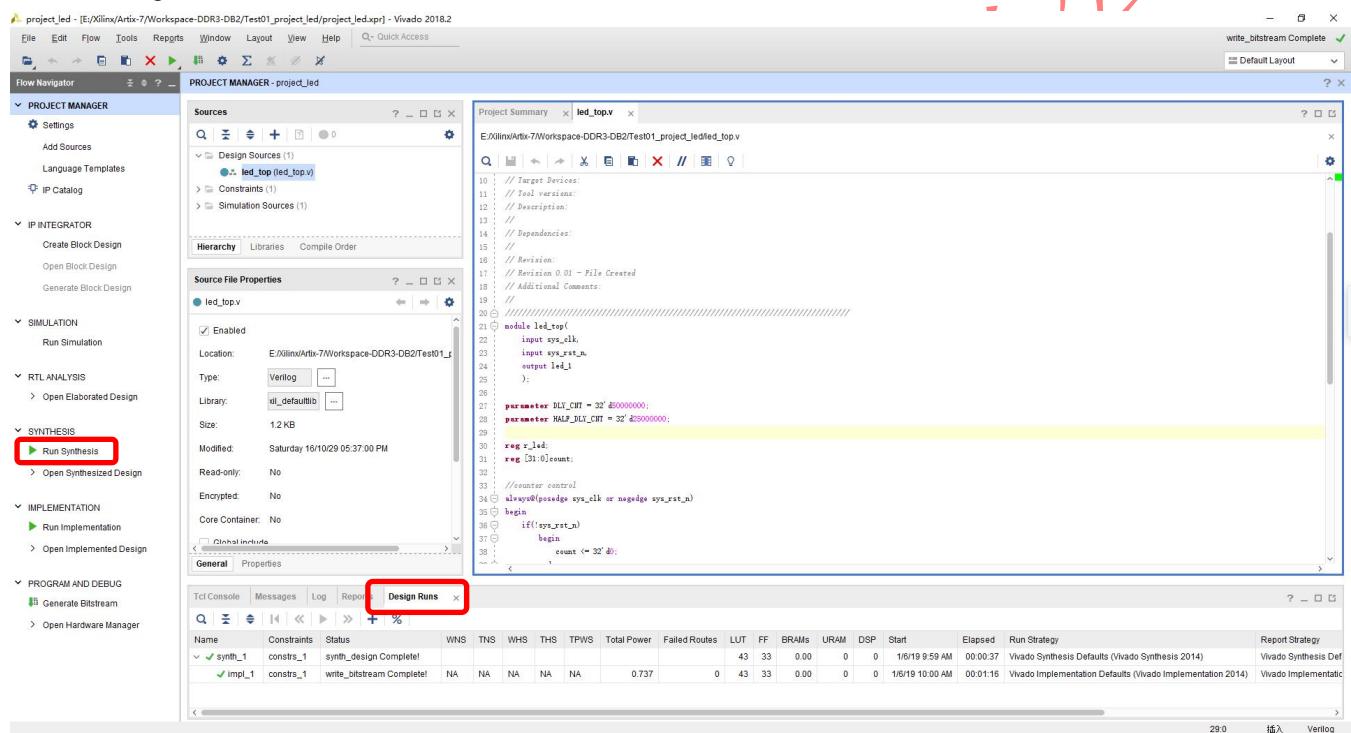


Figure 2-1. SYNTHESIS

There will be a "Synthesis Completed" window popup once the example project successfully synthesized.



Figure 2-2. Synthesis Completed

Users could press the 【ok】 button shown in the previous image to start the project “Implementation”. Or press the 【Cancel】 button shown in the previous image and then click “Run Implementation” button highlighted with red rectangle in below image to start a new implementation.

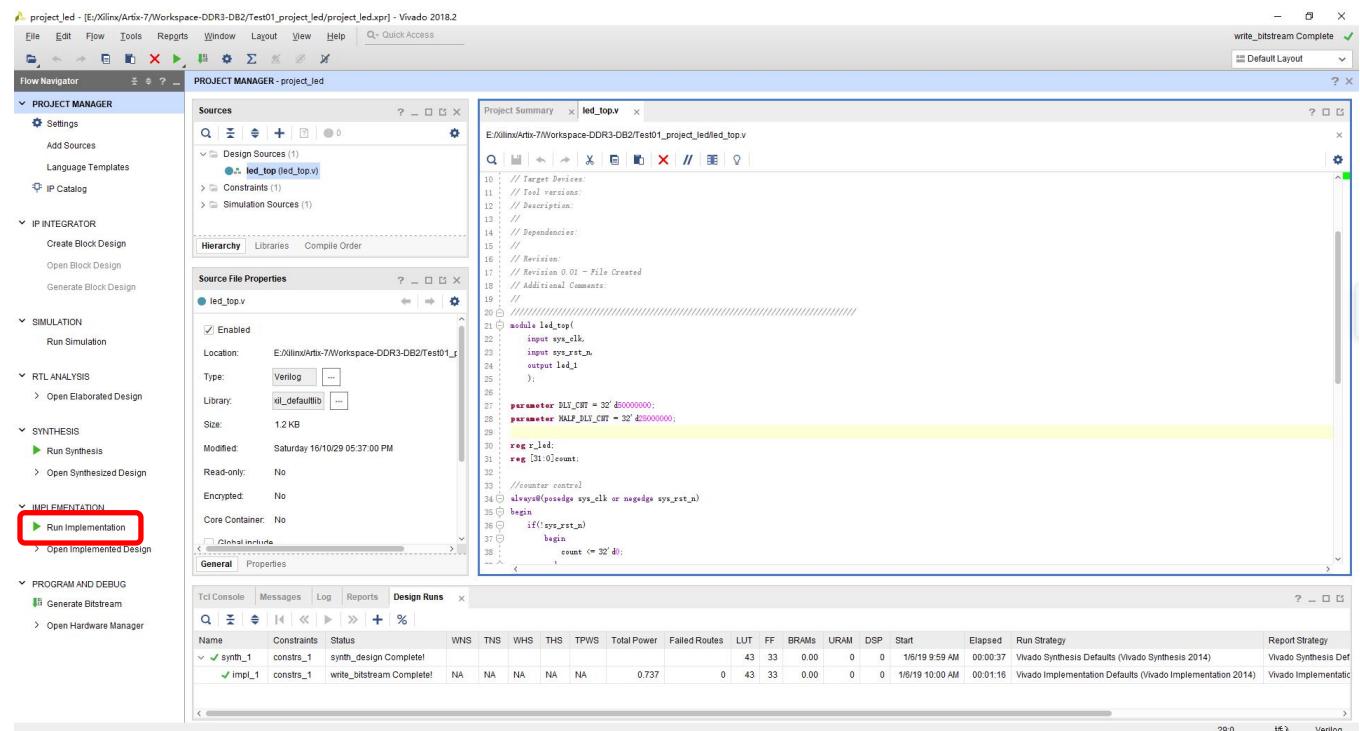


Figure 2-3. Run Implementation

The implementation progress info could be retrieved from the “Design Runs” window. Users could get the implementation info in the tabs like “Log”, “Message” and “Reports”:

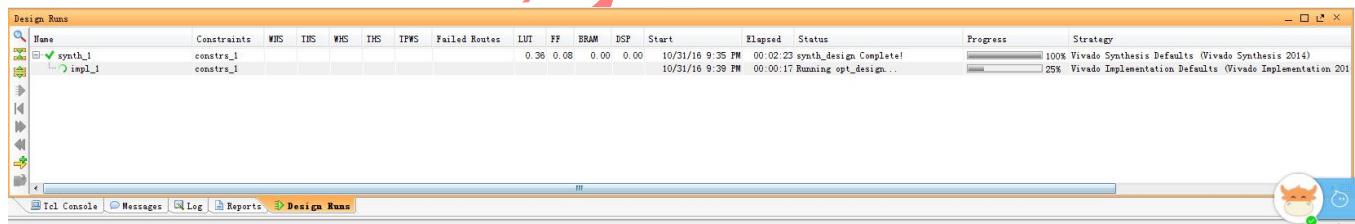


Figure 2-4. Implementation



If there's no error generated during Implementation process, users could start the *.bit file generation stage. Please make sure the constraint file LED.xdc is already contained in the project.

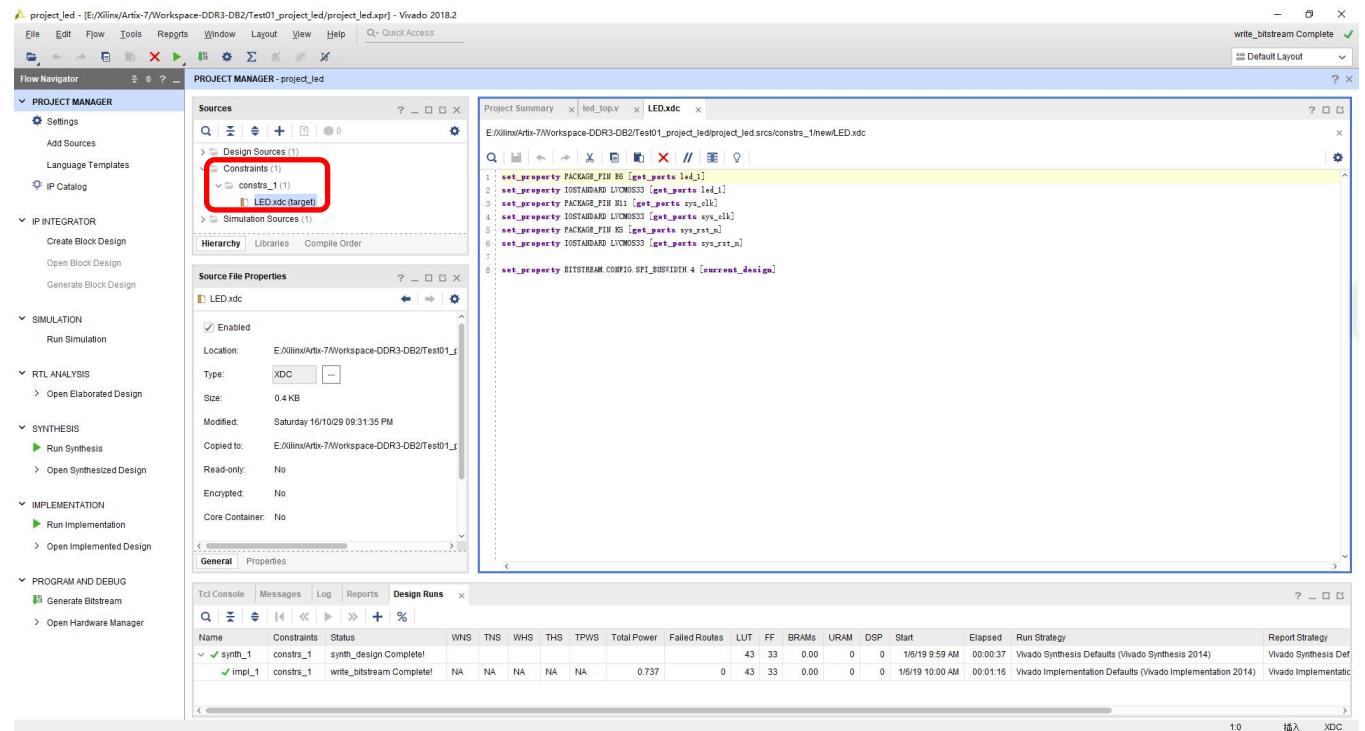


Figure 2-5. LED.xdc Constraint File

There will be a Implemented Completed window popup once the Implementation process finished. Users could click 【OK】 to start the bitstream generation.

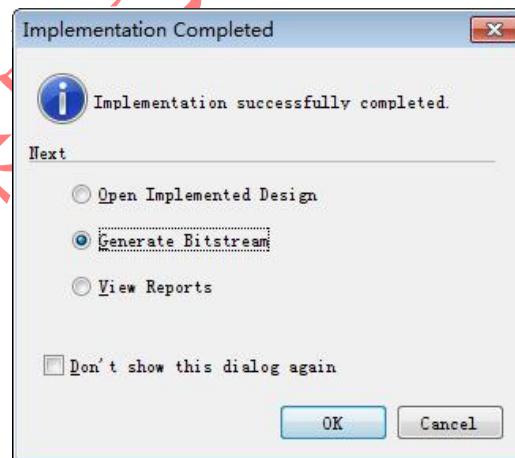


Figure 2-6. Generate Bitstream

Users could also click the 【Cancel】 button shown in the previous image and start to generate the bitstream manually. And then users could click 【Generate Bitstream】 shown in below image to start the bitstream generation.

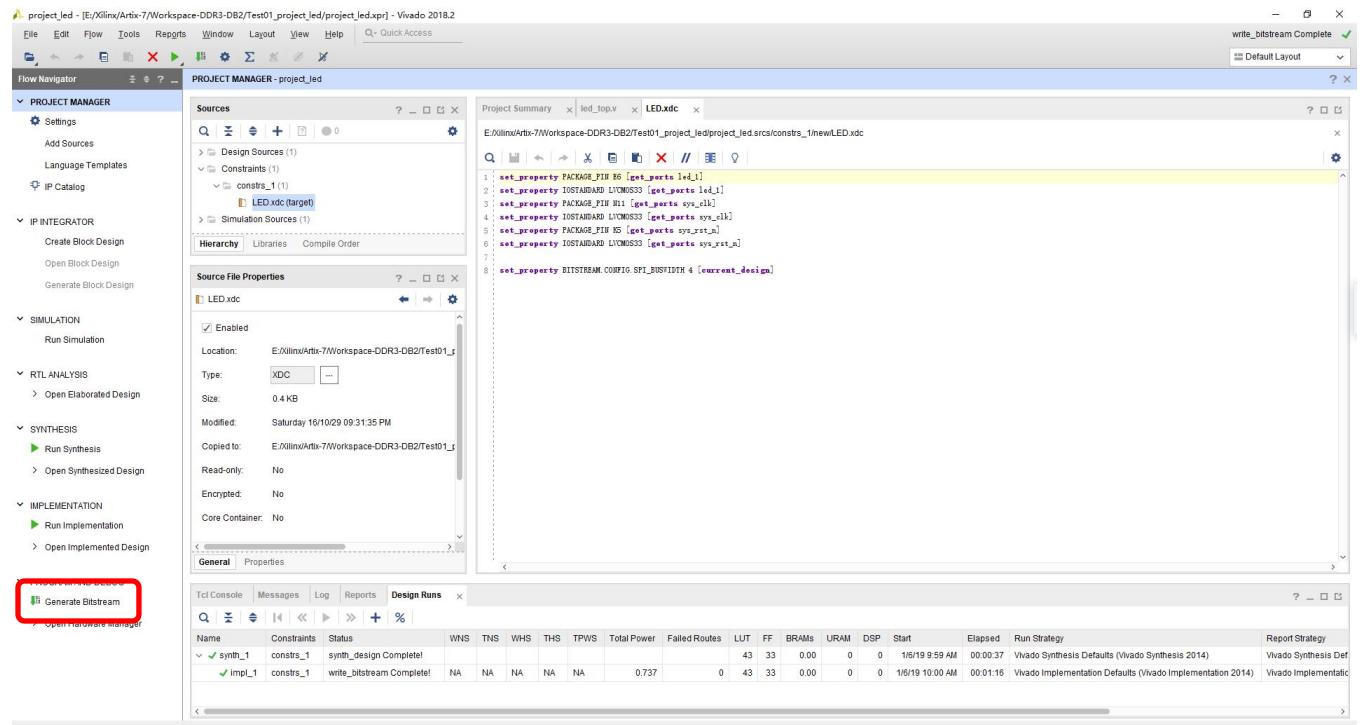


Figure 2-7. Generate Bitstream

Once the *.bit correctly generated without any error, users could download this *.bit into FPGA directly. Make sure the Xilinx USB platform cable is correctly connected to the FPGA board's JTAG interface. And then click the 【Open Target】 button shown in below image:

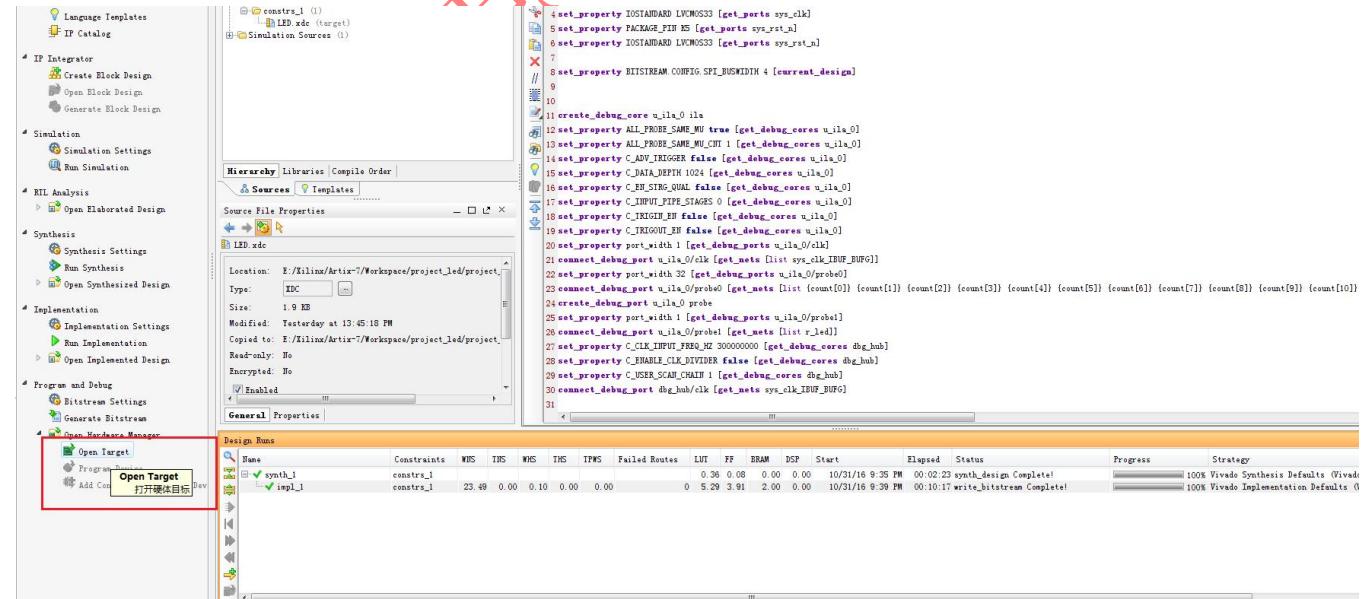


Figure 2-8. Open Target



Click 【Next】 button:

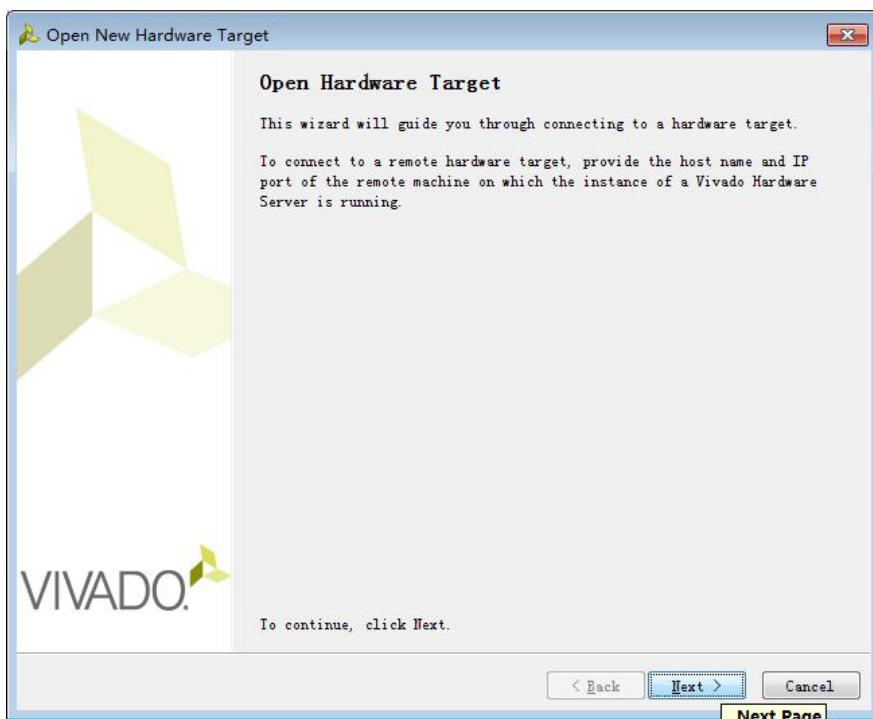


Figure 2-9. "Next"

Click 【Next】 button:

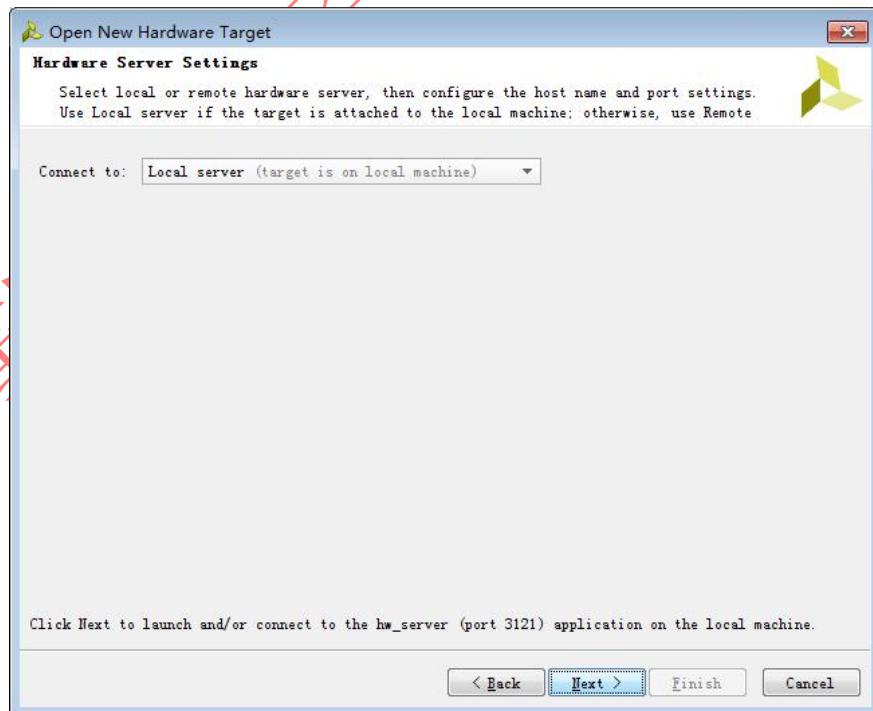


Figure 2-10. "Next"

Below image shows that the Hardware Target "xc7a35t_0" is successfully detected, and then click 【Next】:

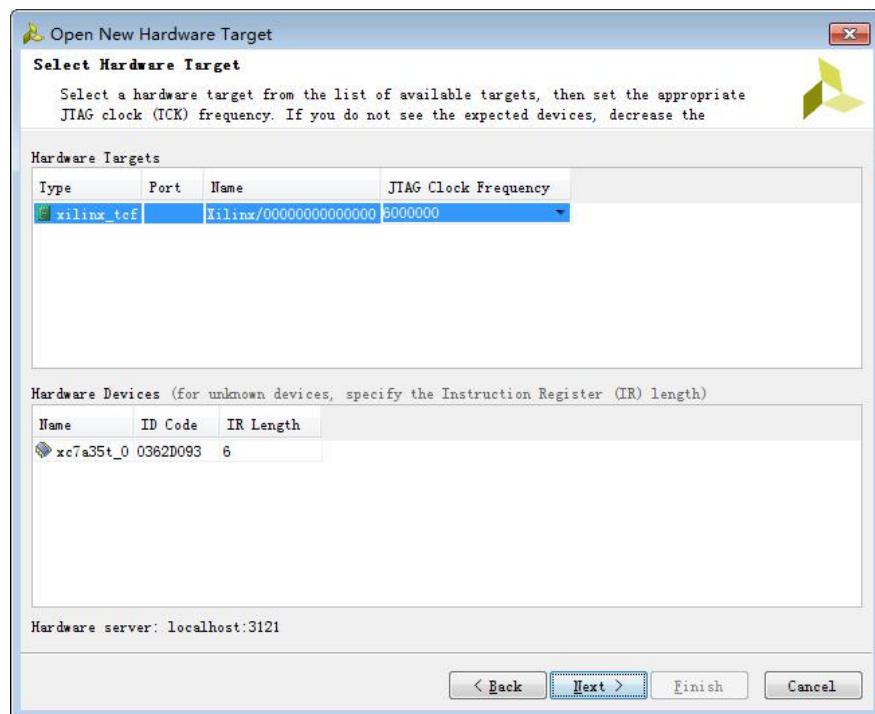


Figure 2-11. Target Detected

Click 【Finish】button:

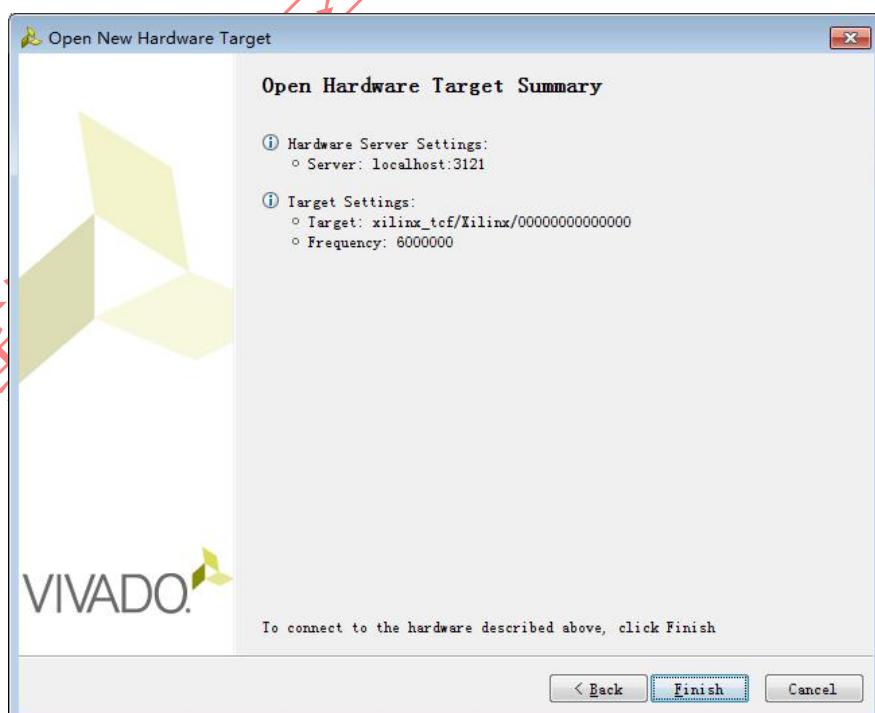


Figure 2-12. “Finish”

Below image shows the main page of the “Hardware Manager”:

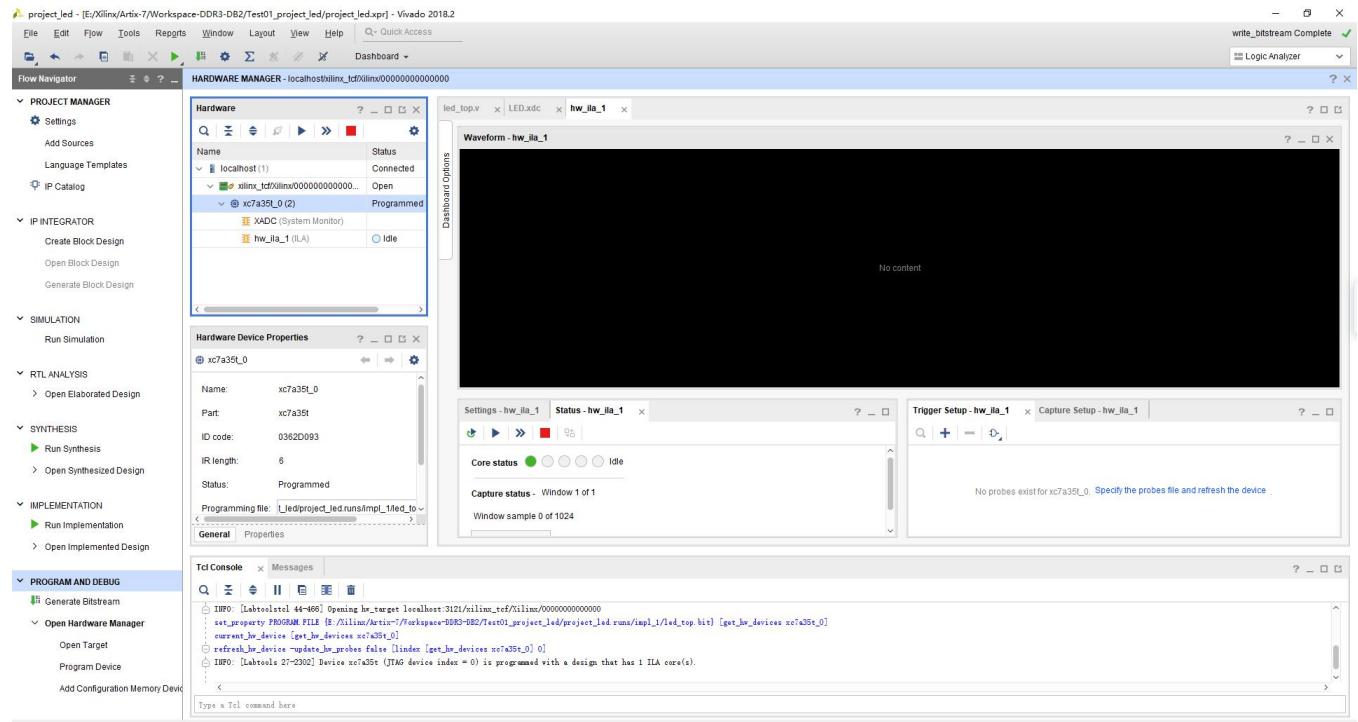


Figure 2-13. Hardware Manager

Right click the detected chip “xc7a35t_0 and choose 【Program Device】 to start the *.bit file download:

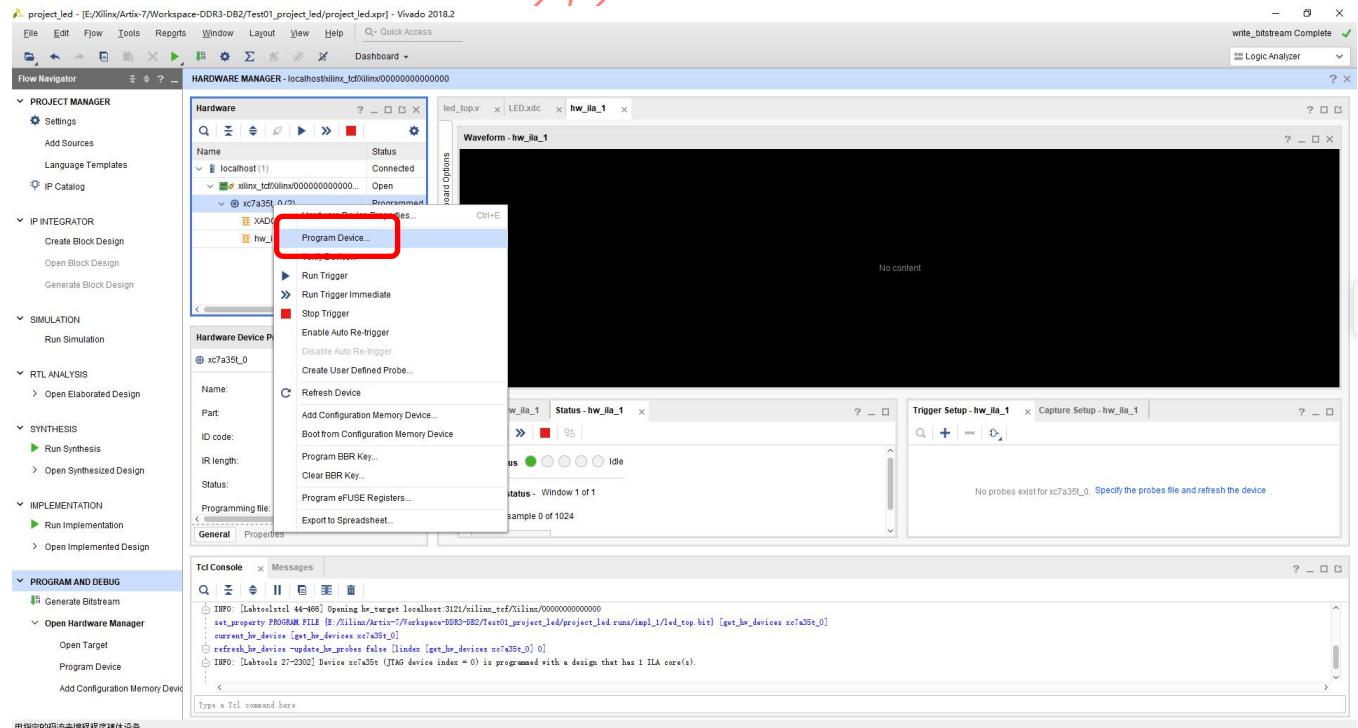


Figure 2-14. Program Device



QMTECH

Artix-7 USB3.0 Development Board

User Manual-V01

Choose the previously generated led_top.bit file and click the 【Program】:

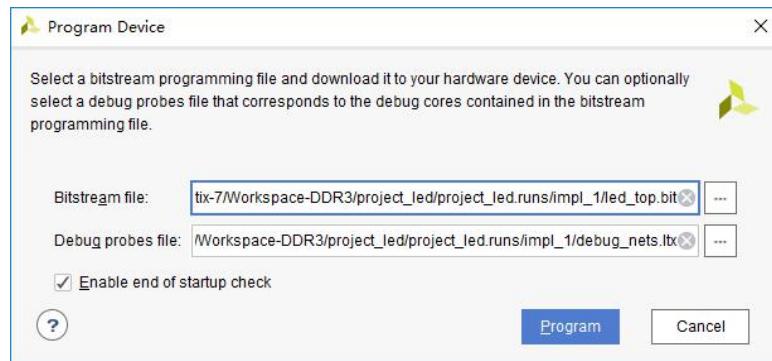


Figure 2-15. Choose *.bit file

Below image shows the progress of the FPGA downloading:

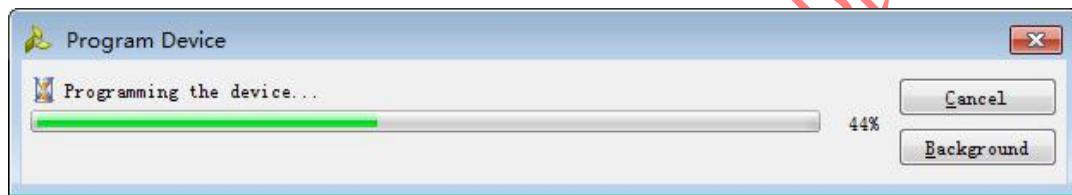


Figure 2-16. Progress Bar

Users could monitor the FPGA internal XADC, Core Voltage supply status, and die temperature by clicking 【XADC (System Monitor)】 shown in below image:

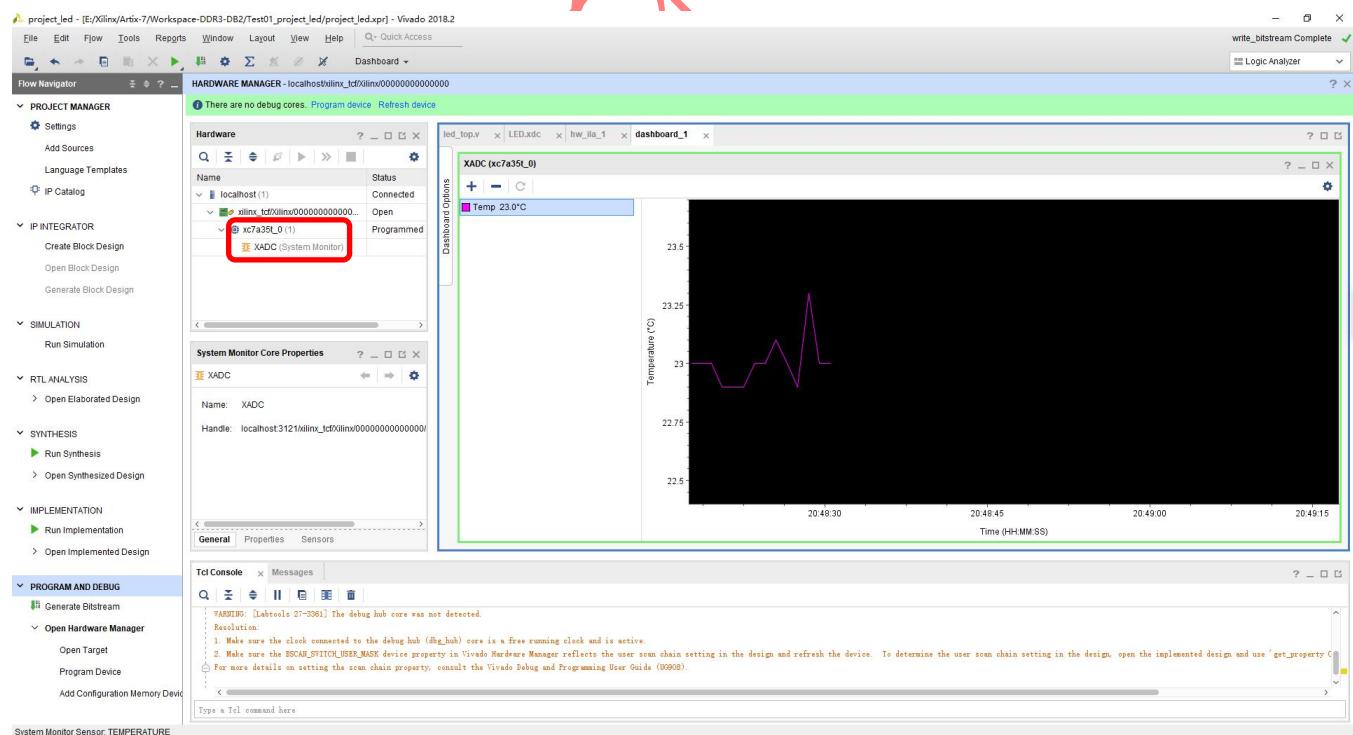


Figure 2-17. XADC

If there's ILA debug core embedded in the example project, users could use this tool to Monitor the waveform and Debug the RTL code. Users could double click the 【hw_il_1 (ILA)】 button and then the



QMTECH

Artix-7 USB3.0 Development Board

User Manual-V01

waveform monitor window "ILA – hw_il_1" will be displayed. The sampling signals and sampling buffer depth could be configured in the Properties tab. Users could click the 【Run Trigger Immediate】 to start waveform capture.

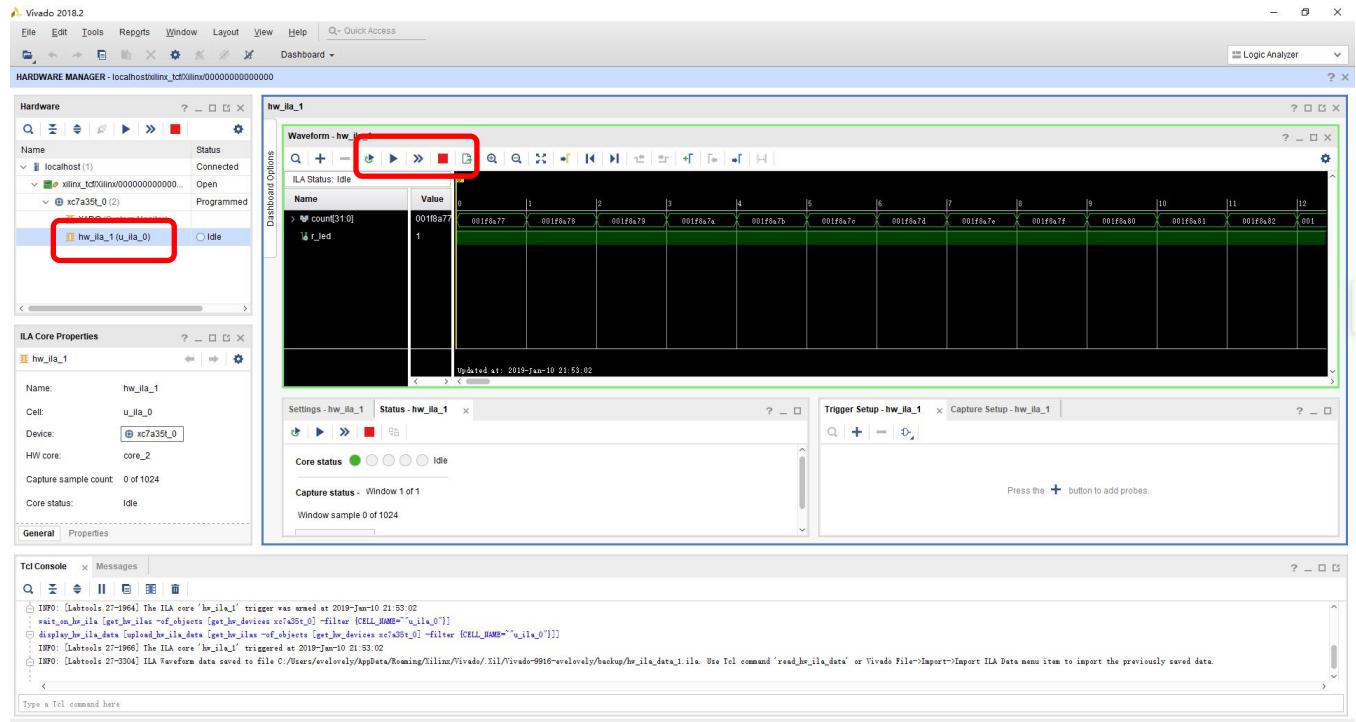


Figure 2-18. ILA Debug

Waveform in hw_il_1:

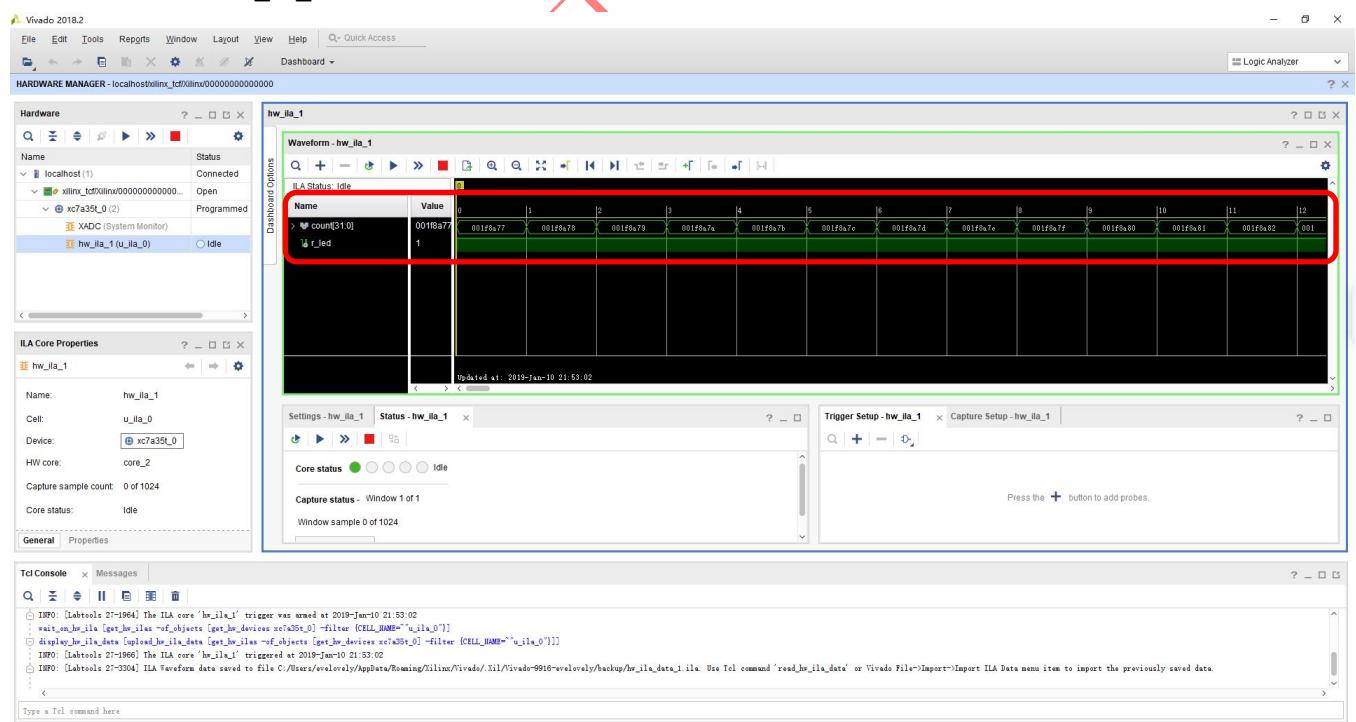


Figure 2-19. Waveform



QMTECH

Artix-7 USB3.0 Development Board

User Manual-V01

3. SPI Flash Program

In the previous chapter, we described how to use Vivado Program tool to download *.bit file into FPGA. But the storage memory embedded in FPGA is SRAM based which means all the content will be flushed during the power on stage. On the development board, there's a non-volatile SPI flash mounted. And the XC7A35T FPGA supports to load bitstream from external SPI flash during power on. In this chapter we will introduce the way to program the bitstream into SPI flash.

Since the *.bit file could not be programmed into SPI flash directly, the file format conversion needs to be done at the very beginning stage.

Users could use Vivado2018.3 【Generate Memory Configuration File】 to convert the led_top.bit into led_top.mcs file. Below image shows where this file format convert tool locates:

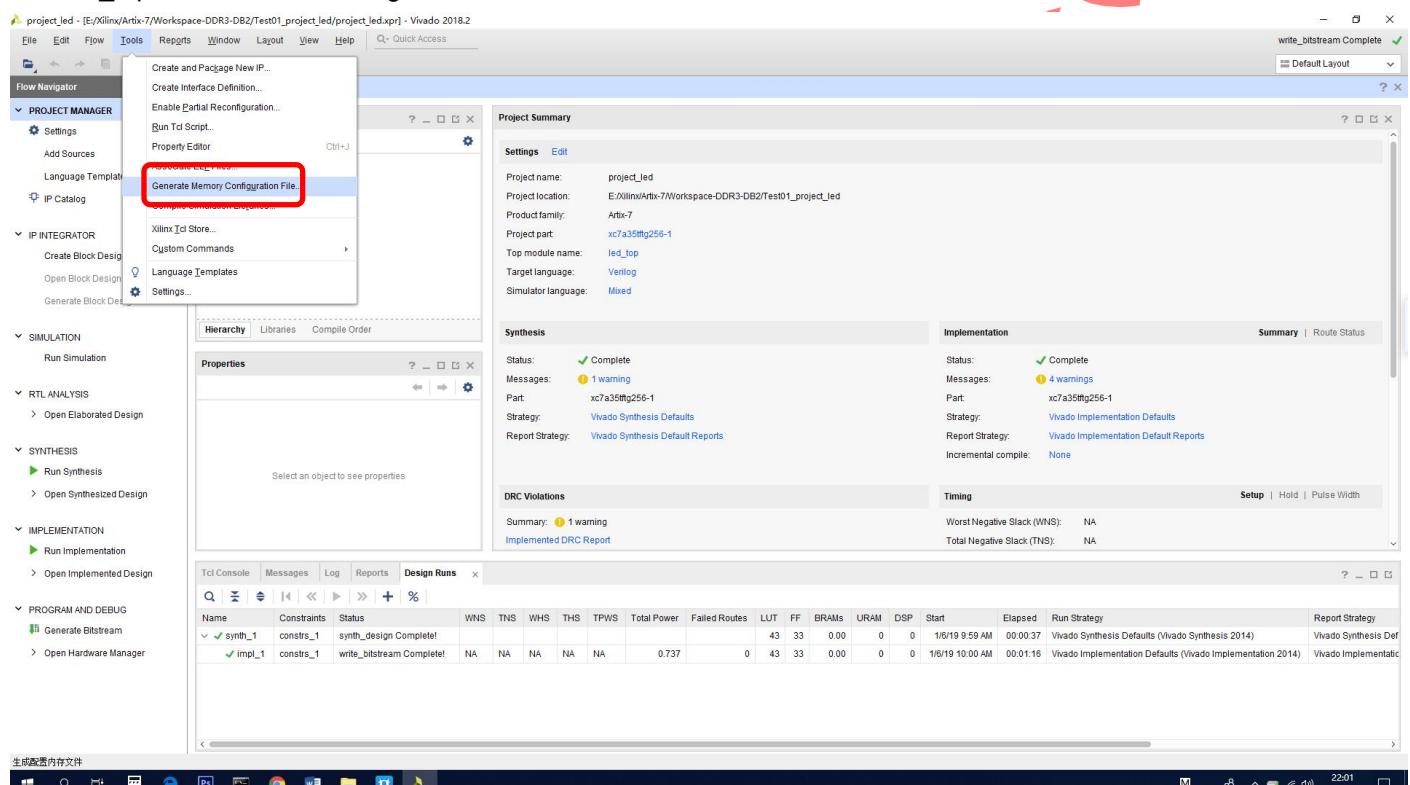


Figure 3-1. Generate Memory Configuration File Tool

Below image shows the example configuration for the file format conversion:

- SPI Flash Part Number: MT25QL128
- SPI Flash bus width: SPIx4
- Input File: led_top.bit
- Output File: led_top.mcs
- Start Address: 0x0000000

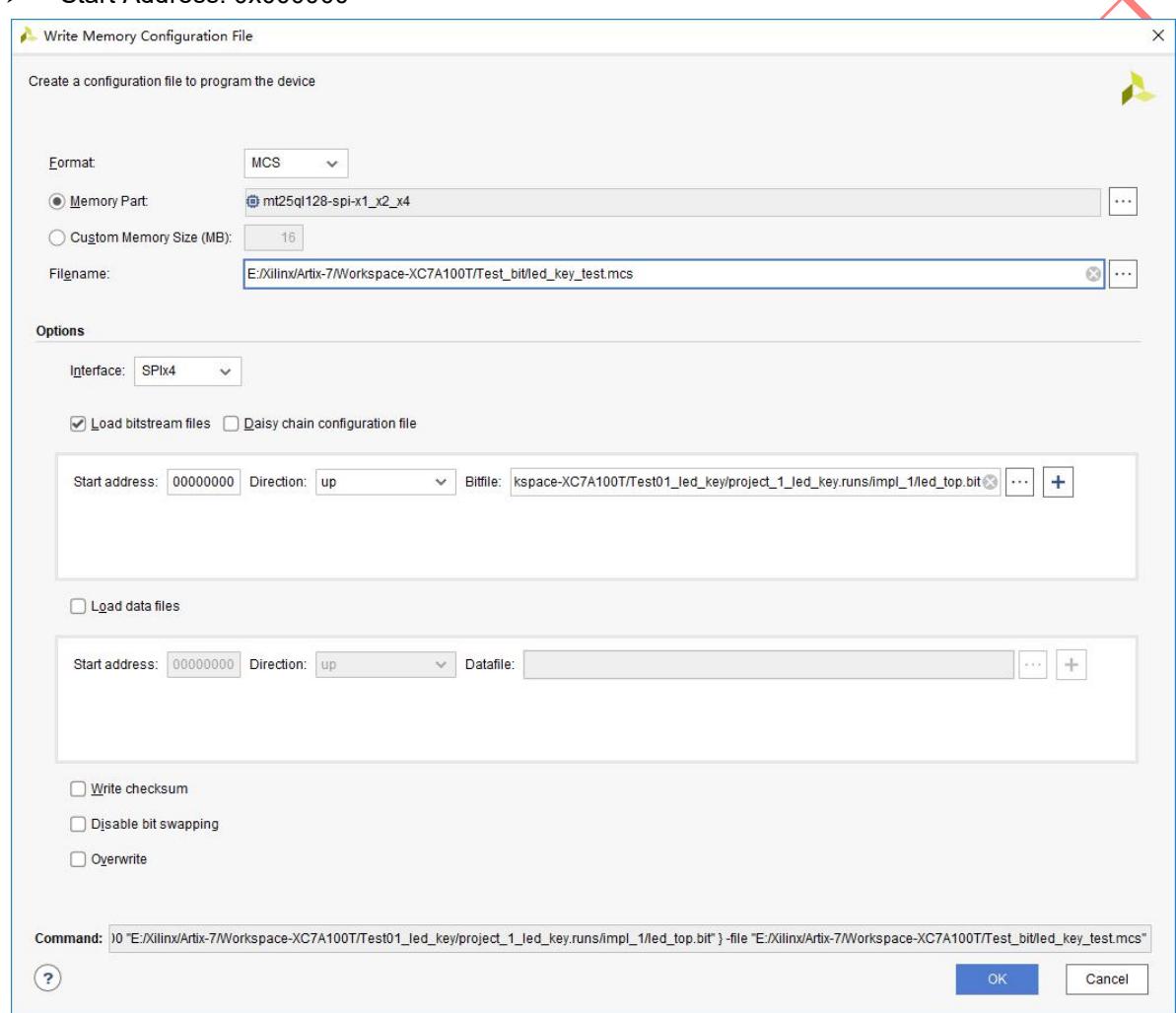


Figure 3-2. Configuration

Once the led_top.mcs file successfully generated, users could program this file into the SPI Flash. Make sure the Xilinx USB Platform Cable is correctly connected to FPGA's JTAG interface. And use Hardware Manager to connect the device “xc7a35t_0 (1)”. Then right click on the detected FPGA device and choose “Add Configuration Memory Device..”. Below image shows the program procedure.

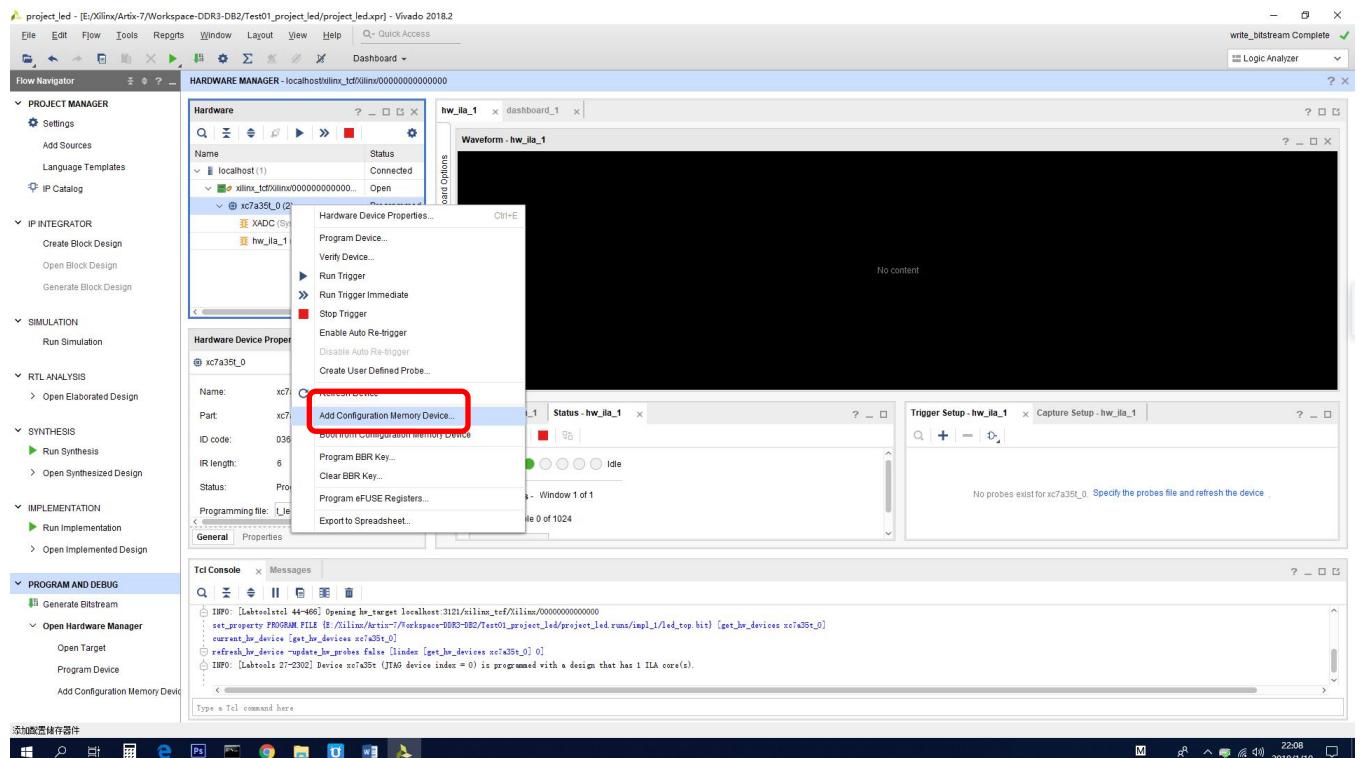


Figure 3-3. Add Memory Device

Choose the SPI Flash chip part: MT25QL128 provided by Micron. And then click 【OK】 button:

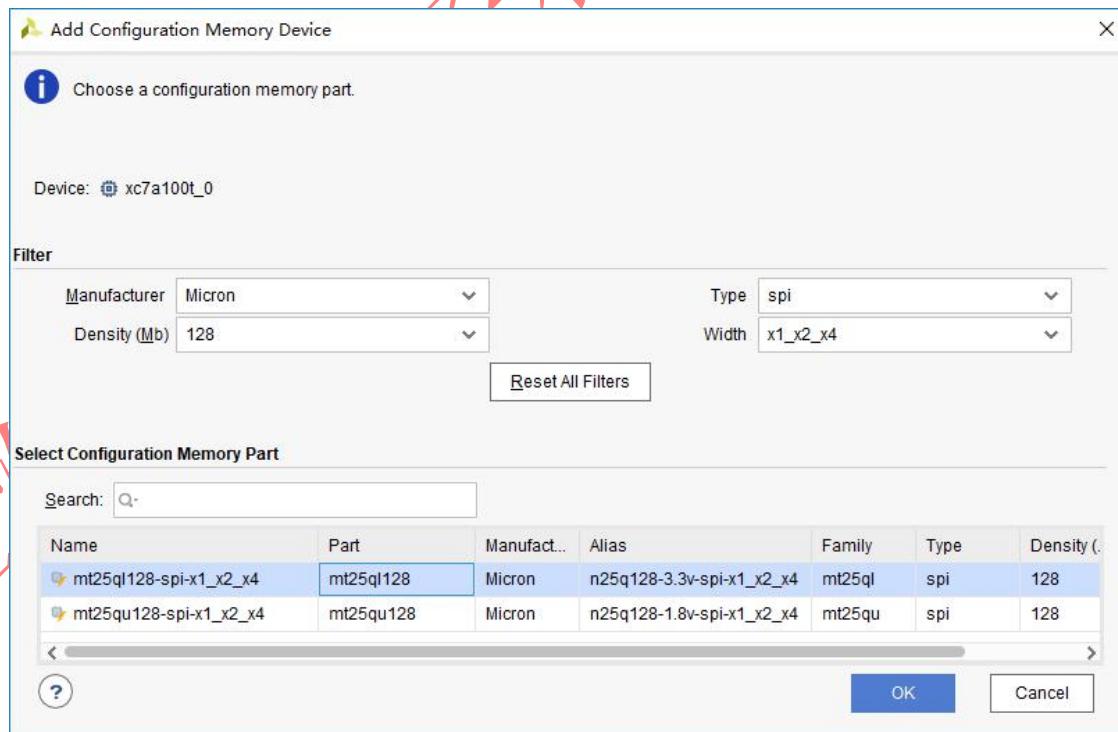


Figure 3-4. Choose SPI Flash

Click 【OK】 button shown in below image to start the Program:





Figure 3-5. Start to Program

Choose the led_top.mcs file and click 【OK】button.

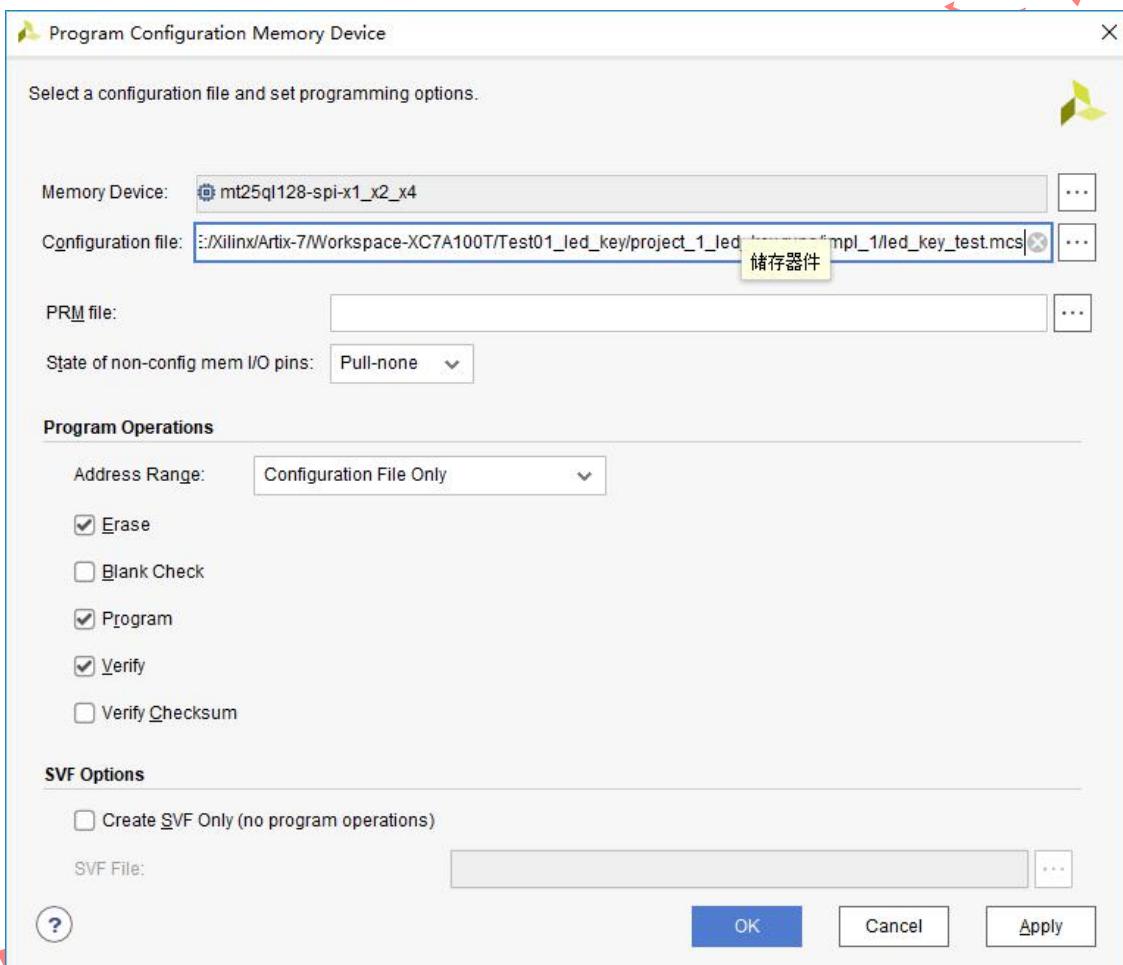


Figure 3-6. Start to Program

Below image shows the progress of the SPI flash programming.

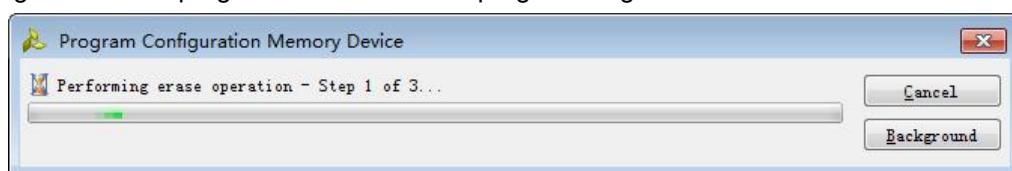


Figure 3-7. Progress Bar

Once the program is successfully finished, users could repower on the board to check whether the FPGA correctly loads the content from SPI flash and implemented functionality is correctly running on FPGA.



4. DDR3 Memory Test

Use Vivado2018.3 to open the DDR3 memory test project located in this release folder: /Software/Test04_DDR3_mig_7series_0.zip. Below figure shows the example project of **example_top**:

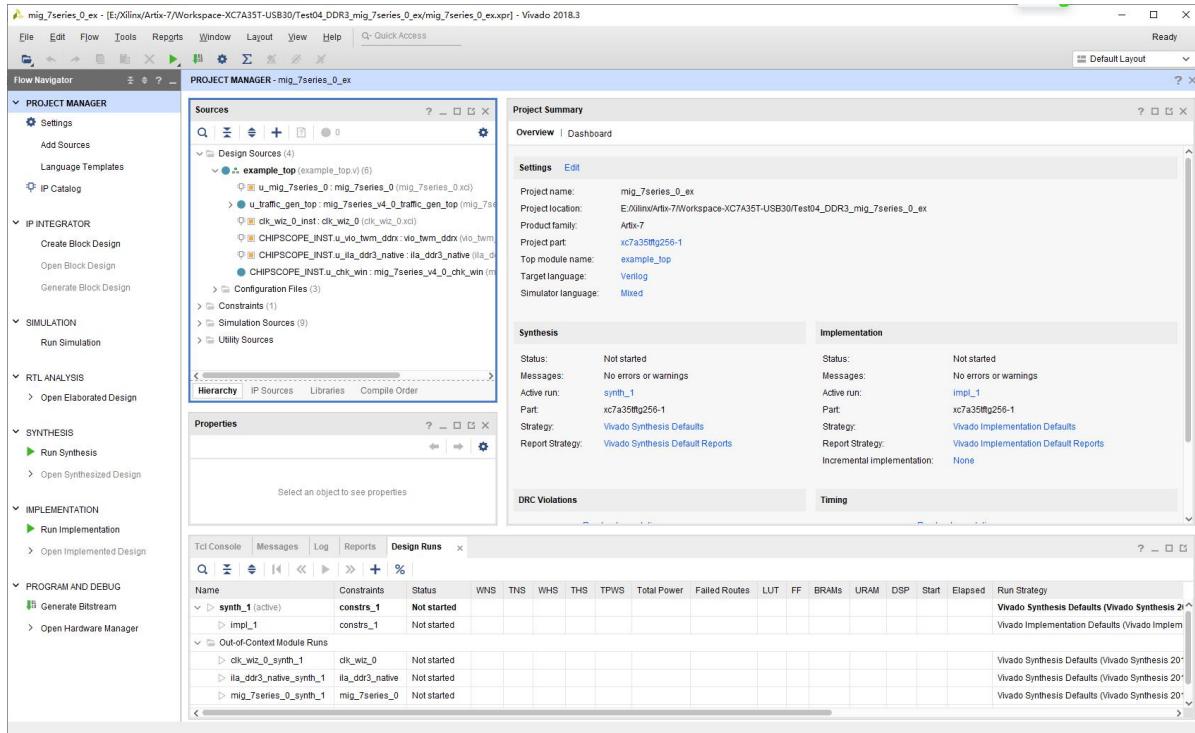
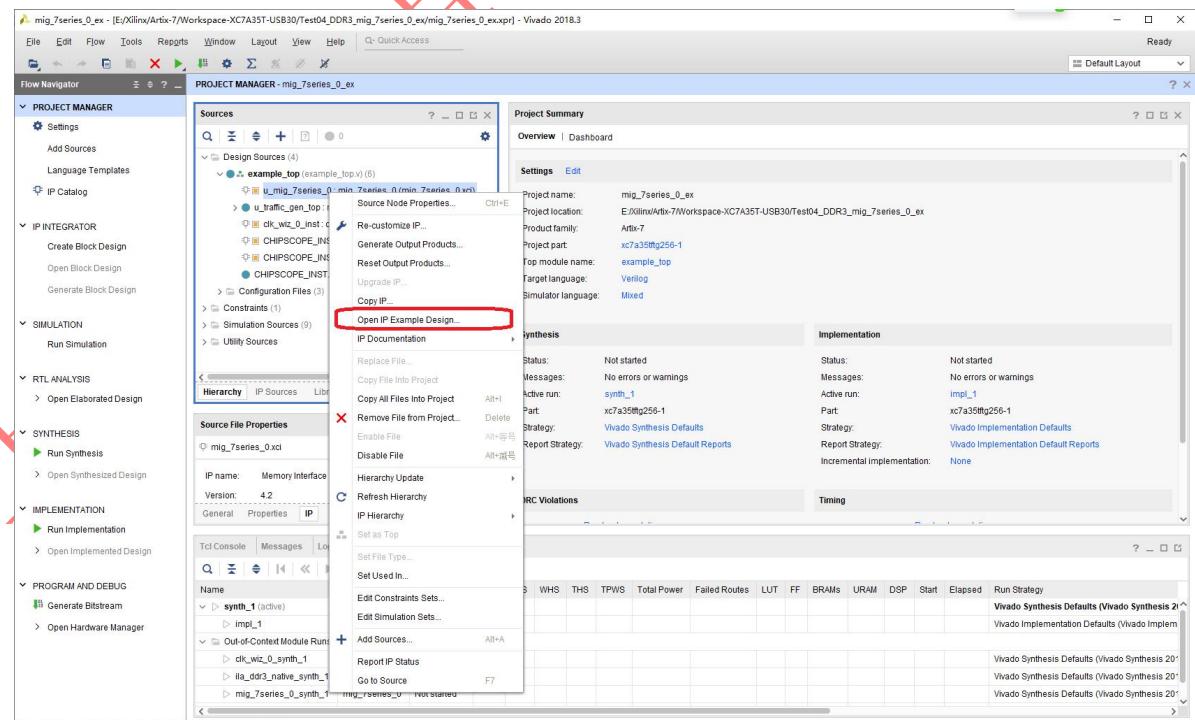


Figure 4-1. FPGA Program

This DDR3 test project is originally created by the Xilinx MIG IP tools. Users could create the test project by right click the **mig_7series_0** IP and select **【Open IP Example Design】**.



After the DDR3 memory test project correctly synthesized, implemented and generated *.bit file, users could use Vivado 2018.3 program tool to program the generated *.bit file into FPGA. Below image shows the FPGA program status with program tool.

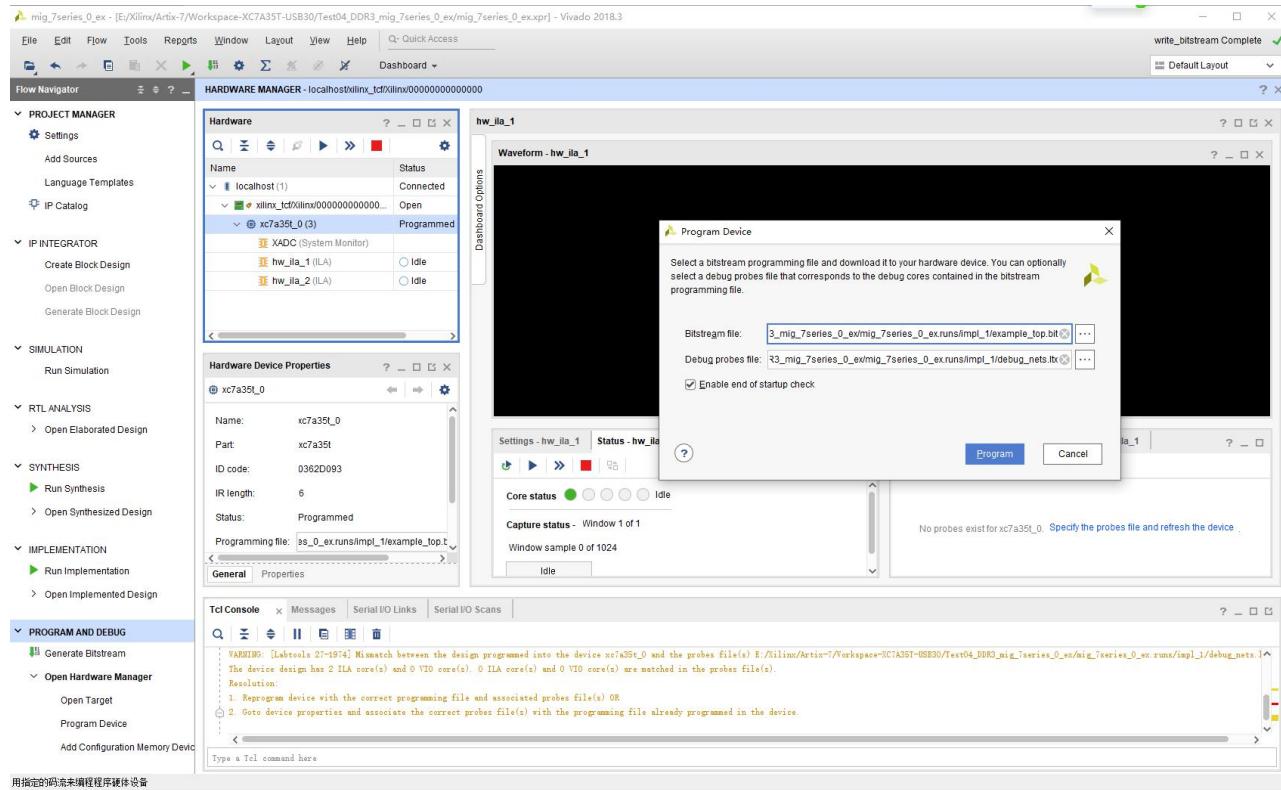
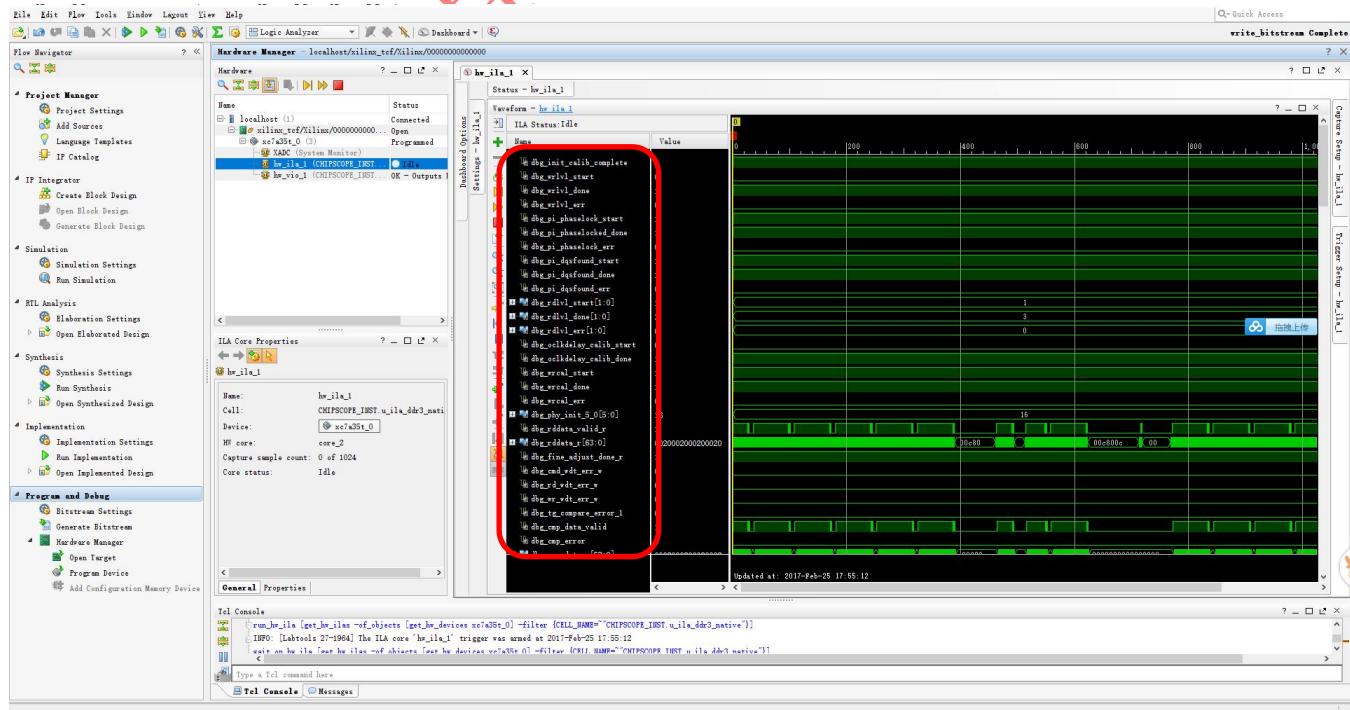
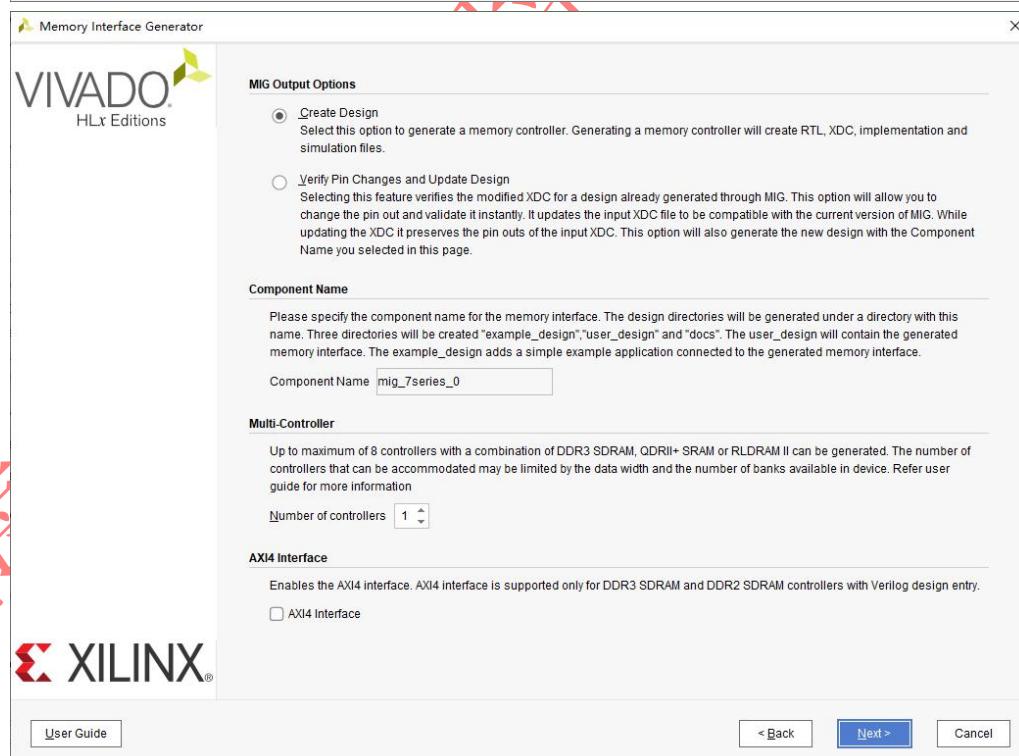
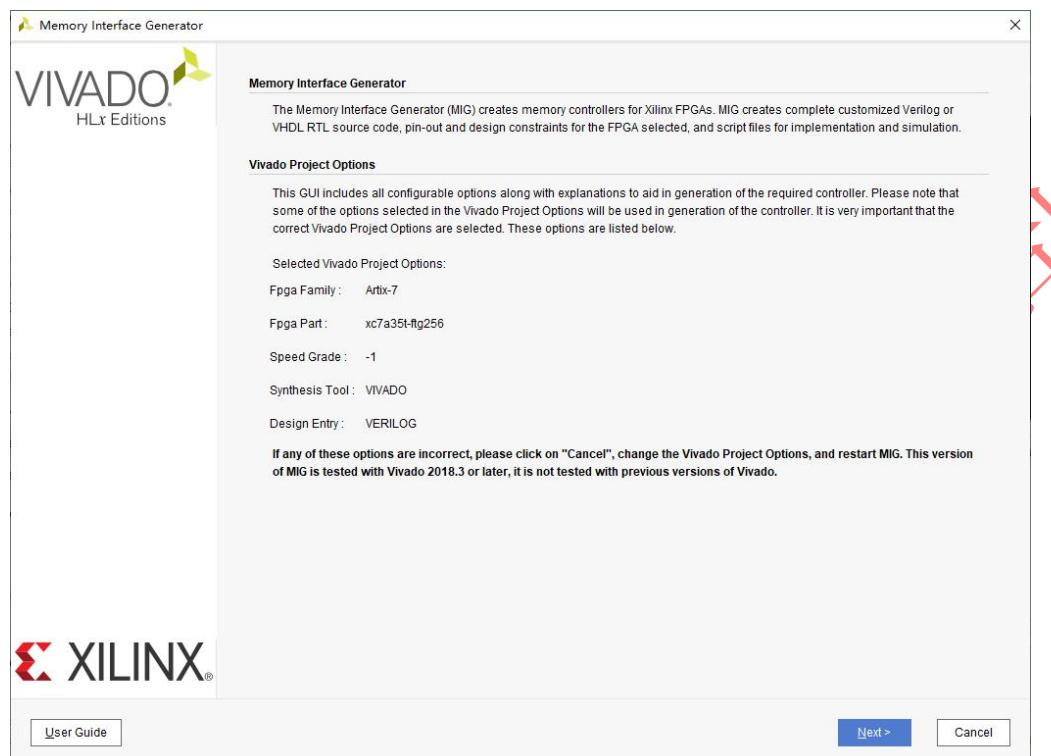


Figure 4-2. FPGA Program

Users could check the signals like init_calib_complete, cmp_error etc in the ILA waveform window.



Below images list the example configurations for this MIG IP. Users may customize it if needed.



Memory Interface Generator

VIVADO
HLx Editions

Pin Compatible FPGAs

- Memory Selection
- Controller Options
- AXI Parameter
- Memory Options
- FPGA Options
- Extended FPGA Options
- IO Planning Options
- Pin Selection
- System Signals Selection
- Summary
- Simulation Options
- PCB information
- Design Notes

XILINX

User Guide

< Back **Next >** **Cancel**

Memory Interface Generator

VIVADO
HLx Editions

Pin Compatible FPGAs

Memory Selection

Select the type of memory interface. Please refer to the User Guide for a detailed list of supported controllers for each FPGA family. The list below shows currently available interface(s) for the specific FPGA, speed grade and design entry chosen.

Select the controller type:

DDR3 SDRAM

DDR2 SDRAM

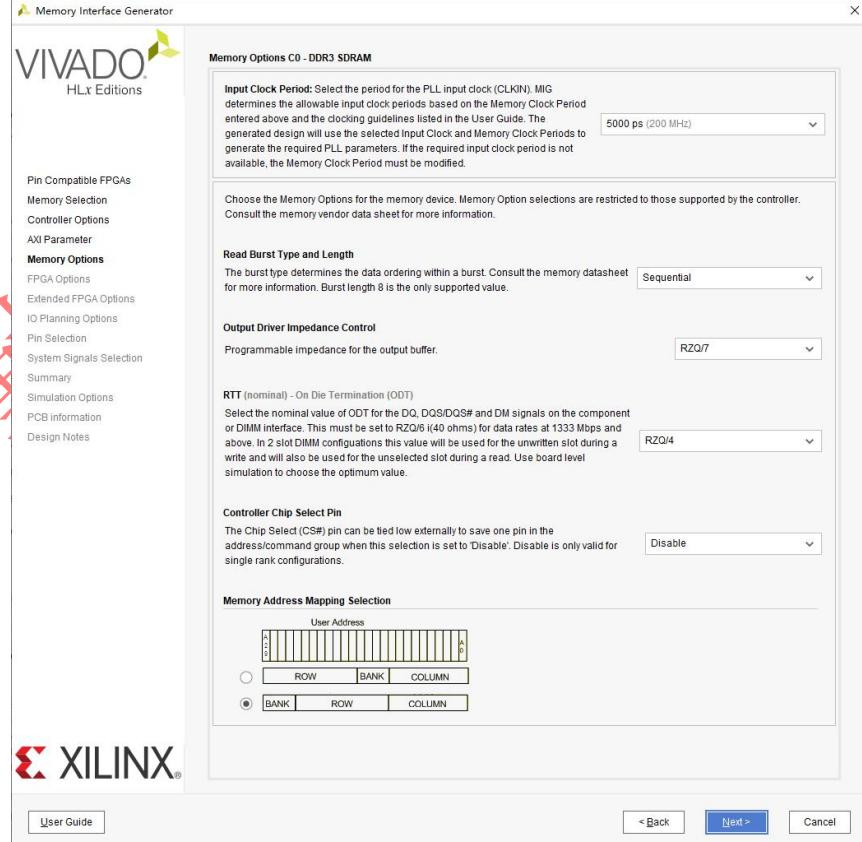
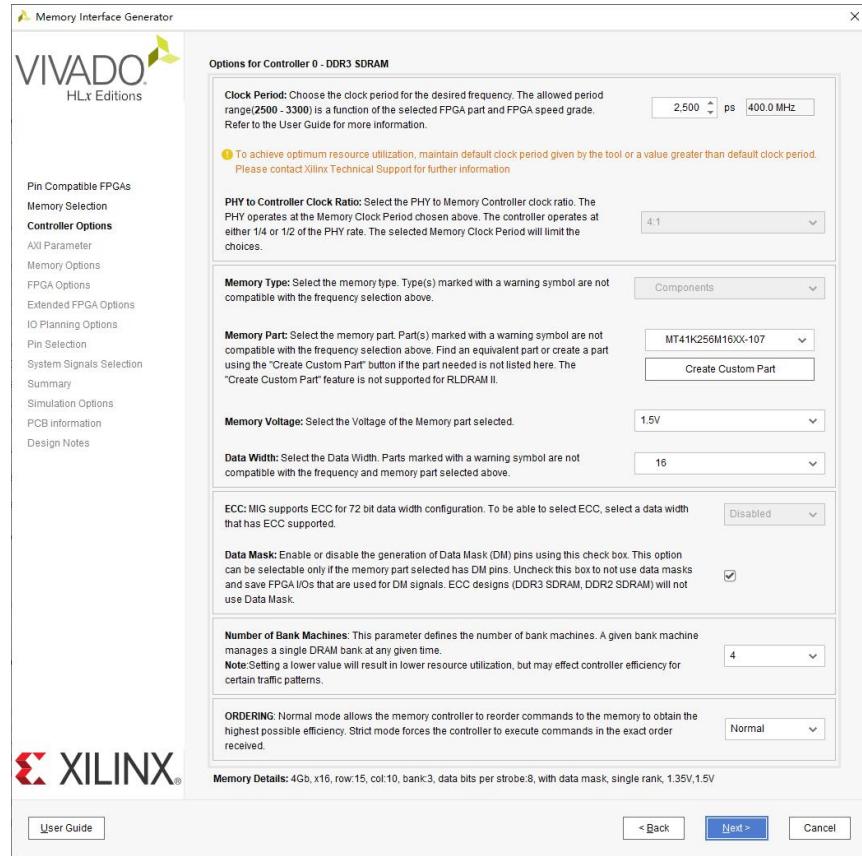
LPDDR2 SDRAM

XILINX

User Guide

< Back **Next >** **Cancel**

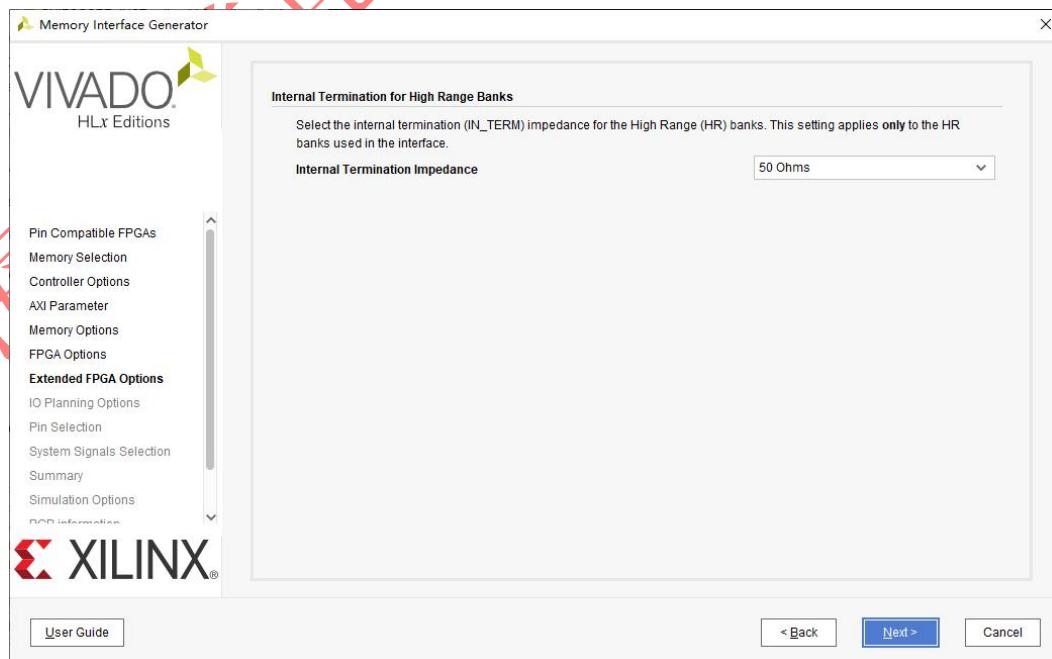
Figure 4-3. MIG Configuration

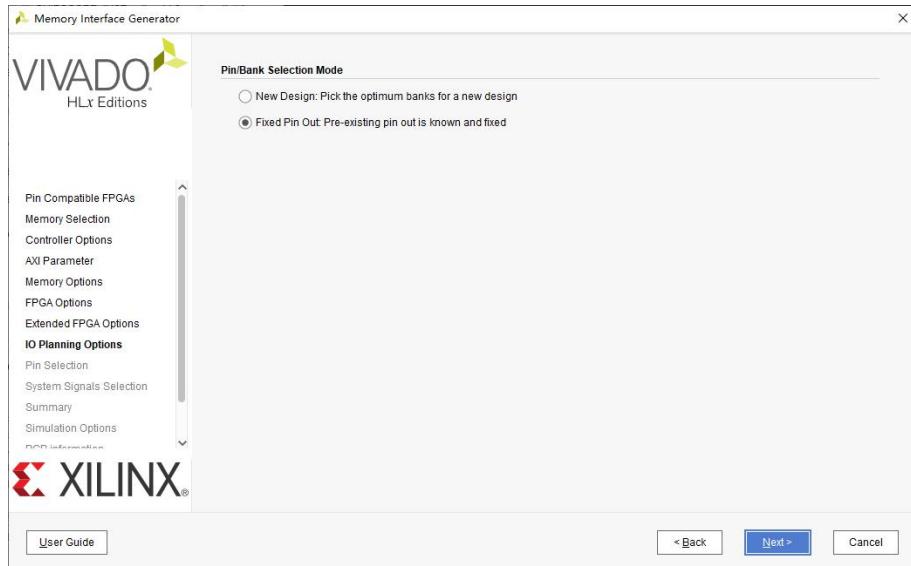


QMTECH

Artix-7 USB3.0 Development Board

User Manual-V01





Pin Selection For Controller 0 - DDR3 SDRAM

| Signal Name | Bank Number | Byte Number | Pin Number | IO Standard |
|------------------|-------------|-------------|------------|---------------|
| 1 ddr3_dq[0] | 15 | ▼ T2 | ▼ F15 | ▼ SSTL15 |
| 2 ddr3_dq[1] | 15 | ▼ T2 | ▼ F13 | ▼ SSTL15 |
| 3 ddr3_dq[2] | 15 | ▼ T2 | ▼ E16 | ▼ SSTL15 |
| 4 ddr3_dq[3] | 15 | ▼ T2 | ▼ D11 | ▼ SSTL15 |
| 5 ddr3_dq[4] | 15 | ▼ T2 | ▼ E12 | ▼ SSTL15 |
| 6 ddr3_dq[5] | 15 | ▼ T2 | ▼ E13 | ▼ SSTL15 |
| 7 ddr3_dq[6] | 15 | ▼ T2 | ▼ D16 | ▼ SSTL15 |
| 8 ddr3_dq[7] | 15 | ▼ T2 | ▼ E11 | ▼ SSTL15 |
| 9 ddr3_dq[8] | 15 | ▼ T3 | ▼ G12 | ▼ SSTL15 |
| 10 ddr3_dq[9] | 15 | ▼ T3 | ▼ J16 | ▼ SSTL15 |
| 11 ddr3_dq[10] | 15 | ▼ T3 | ▼ G16 | ▼ SSTL15 |
| 12 ddr3_dq[11] | 15 | ▼ T3 | ▼ J15 | ▼ SSTL15 |
| 13 ddr3_dq[12] | 15 | ▼ T3 | ▼ H14 | ▼ SSTL15 |
| 14 ddr3_dq[13] | 15 | ▼ T3 | ▼ H12 | ▼ SSTL15 |
| 15 ddr3_dq[14] | 15 | ▼ T3 | ▼ H16 | ▼ SSTL15 |
| 16 ddr3_dq[15] | 15 | ▼ T3 | ▼ H13 | ▼ SSTL15 |
| 17 ddr3_dm[0] | 15 | ▼ T2 | ▼ F12 | ▼ SSTL15 |
| 18 ddr3_dm[1] | 15 | ▼ T3 | ▼ H11 | ▼ SSTL15 |
| 19 ddr3_dqs_p[0] | 15 | ▼ T2 | ▼ D14 | ▼ DIFF_SSTL15 |
| 20 ddr3_dqs_n[0] | 15 | ▼ T2 | ▼ D15 | ▼ DIFF_SSTL15 |
| 21 ddr3_dqs_p[1] | 15 | ▼ T3 | ▼ G14 | ▼ DIFF_SSTL15 |
| 22 ddr3_dqs_n[1] | 15 | ▼ T3 | ▼ F14 | ▼ DIFF_SSTL15 |
| 23 ddr3_addr[14] | 15 | ▼ T0 | ▼ A9 | ▼ SSTL15 |
| 24 ddr3_addr[13] | 15 | ▼ T0 | ▼ B11 | ▼ SSTL15 |
| 25 ddr3_addr[12] | 15 | ▼ T0 | ▼ A8 | ▼ SSTL15 |
| 26 ddr3_addr[11] | 15 | ▼ T4 | ▼ D10 | ▼ SSTL15 |
| 27 ddr3_addr[10] | 15 | ▼ T0 | ▼ B12 | ▼ SSTL15 |
| 28 ddr3_addr[9] | 15 | ▼ T1 | ▼ A13 | ▼ SSTL15 |
| 29 ddr3_addr[8] | 15 | ▼ T0 | ▼ D8 | ▼ SSTL15 |
| 30 ddr3_addr[7] | 15 | ▼ T0 | ▼ A12 | ▼ SSTL15 |
| 31 ddr3_addr[6] | 15 | ▼ T0 | ▼ D9 | ▼ SSTL15 |
| 32 ddr3_addr[5] | 15 | ▼ T0 | ▼ B10 | ▼ SSTL15 |

Validation successful. Press Next to proceed.

Buttons at the bottom include 'Validate', 'Read XDC/UCF', 'Save Pin Out', '< Back', 'Next >', and 'Cancel'.



System Signals Selection

Select the system pins below appropriately for the interface. Customization of these pins can also be made in the XDC after the design is generated. For more information see [UG586 Bank and Pin rules](#).

System Clock and Reference Clock pin selections will not be visible if the 'No Buffer' option was selected in the FPGA Options page.

System Signals

These signals may be connected internally to other logic or brought out to a pin.

- **sys_rst:** This input signal is used to reset the interface.
- **init_calib_complete:** This signal indicates that the interface has completed calibration and memory initialization and is ready for commands. LOC constraint will be generated in XDC for Example design only based on "Pin Number" selection below.
- **error:** This output signal indicates that the traffic generator in the Example Design has detected a data mismatch. This signal does not exist in the User Design.

| Signal Name | Bank Number | Pin Number |
|---------------------|-------------|------------|
| sys_rst | 35 | J3 |
| init_calib_complete | 34 | N3 |
| tg_compare_error | 34 | M4 |

All pins must be constrained to specific locations in order to generate a bit file in the implementation phase (this is not required for simulation).

User Guide **< Back** **Next >** **Cancel**

Vivado Project Options:

```

Target Device : xc7a35t-ftg256
Speed Grade : -1
HDL : verilog
Synthesis Tool : VIVADO

```

If any of the above options are incorrect, please click on "Cancel", change the CORE Generation Options and try again.

MIG Output Options:

```

Module Name : mig_7series_0
No of Controllers : 1
Selected Compatible Device(s) : xc7a50t-ftg256, xc7a75t-ftg256, xc7a100t-ftg256, xc7a15t-ftg256

```

FPGA Options:

```

System Clock Type : No Buffer
Reference Clock Type : Use System Clock
Debug Port : ON
Internal Vref : enabled
IO Power Reduction : OFF
XADC instantiation in MIG : Enabled

```

Extended FPGA Options:

```

DCI for DQ,DQS/DQS#,DM : enabled
Internal Termination (HR Banks) : 50 Ohms

```

Controller Options :

```

Memory : DDR3_SDRAM
Interface : NATIVE
Design Clock Frequency : 2500 ps (400.00 MHz)
Phy to Controller Clock Ratio : 4:1
Input Clock Period : 4999 ps
CLKFBOUT_MULT (PLL) : 4
DIVCLK_DIVIDE (PLL) : 1
VCC_AUX IO : 1.8V
Memory Type : Components
Memory Part : MT41K256M16XX-107
Equivalent Part(s) : --
Data Width : 16
ECC : Disabled
Data Mask : enabled
ORDERING : Normal

```

User Guide **< Back** **Next >** **Cancel**

Figure 4-4. MIG Configuration

5. Reference

- [1] ug470_7Series_Config.pdf
- [2] ds181_Artix_7_Data_Sheet.pdf
- [3] ug475_7Series_Pkg_Pinout.pdf
- [4] MT25QL128A.pdf
- [5] MT41K256M16.pdf
- [6] MP2315.pdf
- [7] NCP1529-D.PDF

上海勸謀電子科技有限公司



6. Revision

| Doc. Rev. | Date | Comments |
|-----------|-----------|----------------------|
| 0.1 | 7/1/2020 | Initial Version. |
| 1.0 | 7/11/2020 | V1.0 Formal Release. |
| | | |
| | | |
| | | |

上海勸謀電子科技有限公司

