RIPHAH INTERNATIONAL UNIVERSITY

Lab 15
Adding Custom IP Lab: MicroBlaze

RIPHAH INTERNATIONAL UNIVERSITY

Adding Custom IP Lab: MicroBlaze

Introduction

This lab guides you through the process of adding a custom OPB peripheral to a processor system by using the Import Peripheral Wizard.

Objectives

After completing this lab, you will be able to:

- Add a custom IP to your design
- Modify the UCF file
- Implement the design

Procedure

The purpose of this lab exercise is to complete the hardware design started in Lab 13 and extended in Lab 14. Lab 13 included the MicroBlaze™ processor, mb_opb, debug_module, OPB UART, two GPIOs, DLMB controller, ILMB controller, and LMB BRAM. Lab 14 added the remaining IP, except for a GPIO instance for the 7-segment LEDs, to extend the hardware design.

In this lab, you will use the Import Peripheral Wizard of Xilinx Platform Studio (XPS) to create a user peripheral from an HDL module, add an instance of the imported peripheral, and modify the system.ucf file to provide an interface to the on-board 7-segment LED module.

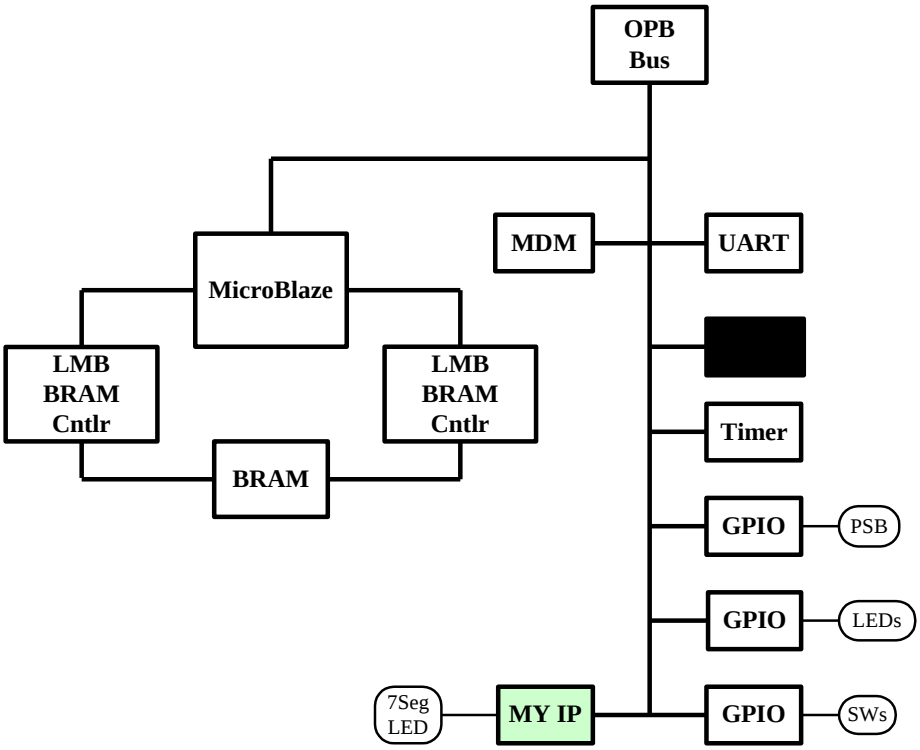


Figure 7c-1. Completed Design

This lab comprises several steps involving the addition of a custom OPB peripheral. Although the change to the hardware is simple, the lab illustrates the integration of a user peripheral through the importing peripheral wizard. The lab also illustrates the use of an existing peripheral to provide the OPB bus interface.

Below each general instruction for a given procedure, you will find accompanying step-by-step directions and illustrated figures providing more detail for performing the general instruction. If you feel confident about a specific instruction, feel free to skip the step-by-step directions and move on to the next general instruction in the procedure.

Opening the Project — Step 1

Create a **lab15** folder under c**:\xup\embedded\labs**. If you wish to continue with your completed design from lab2 then copy the contents of the **lab13** folder into the **lab15** folder.

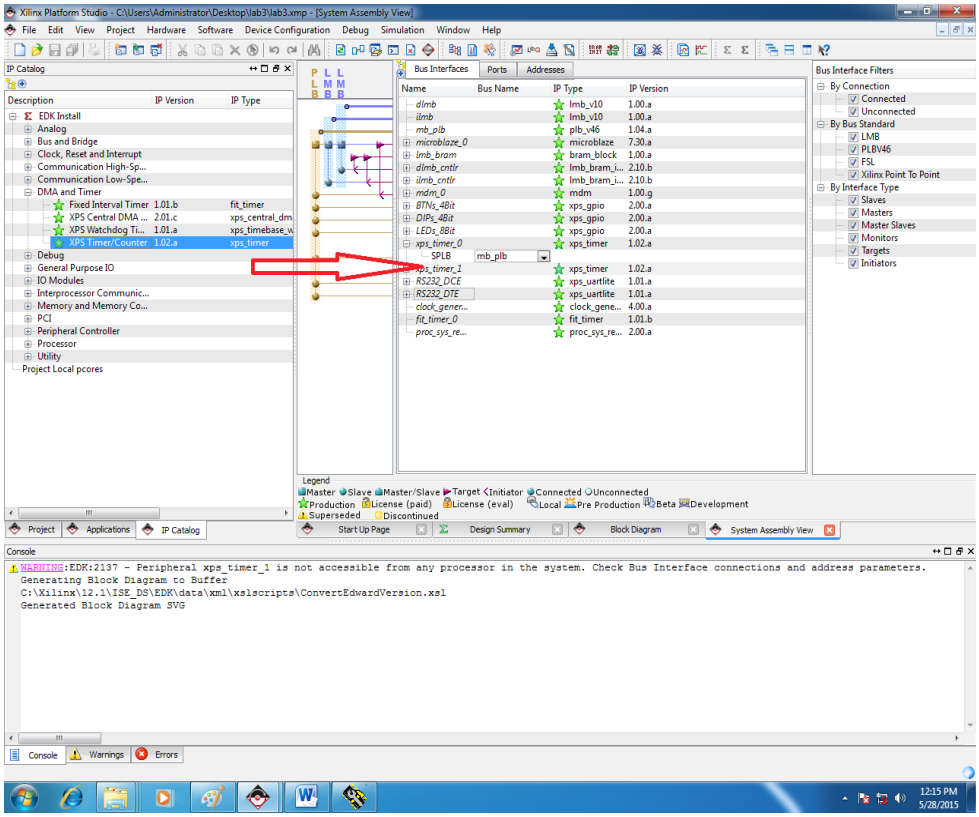
❶

❷ Open XPS by clicking **Start → Programs → Xilinx Platform Studio → Xilinx Platform Studio**

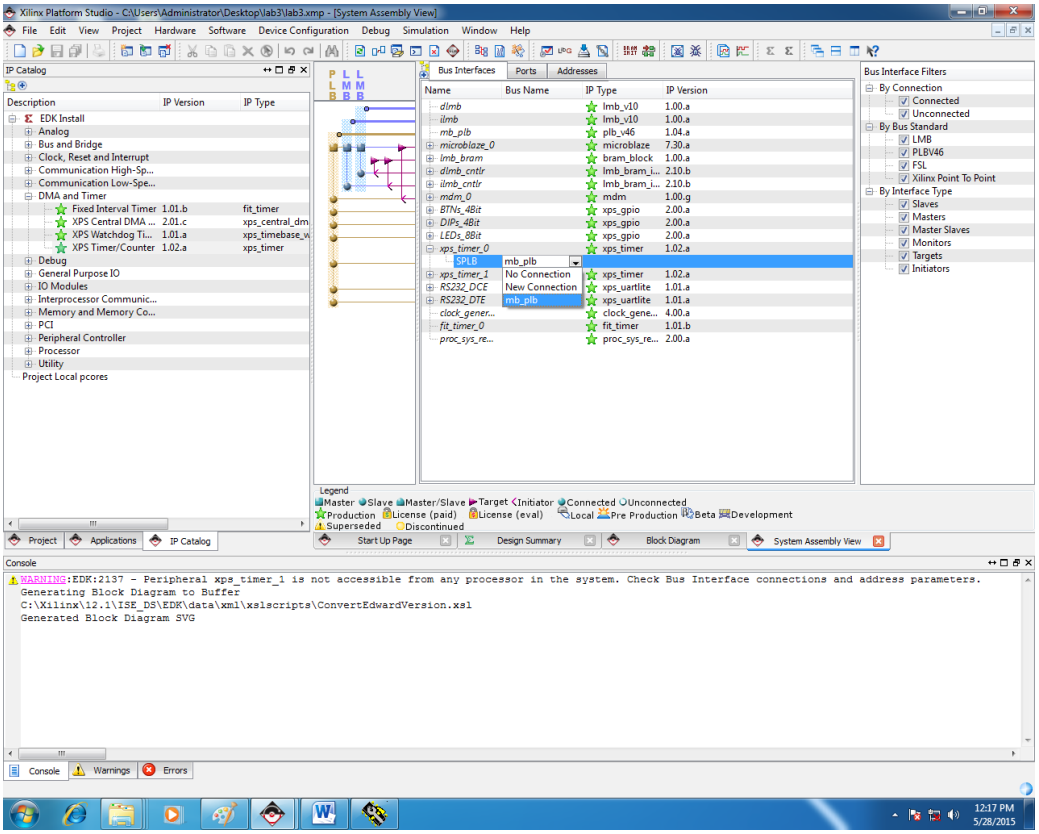
❸ Click **File → Open Project** and browse to the project which in the directory: c**:\xup\embedded\labs\lab15**

❹ Select **system.xmp** in the **lab3mb** directory to open the project

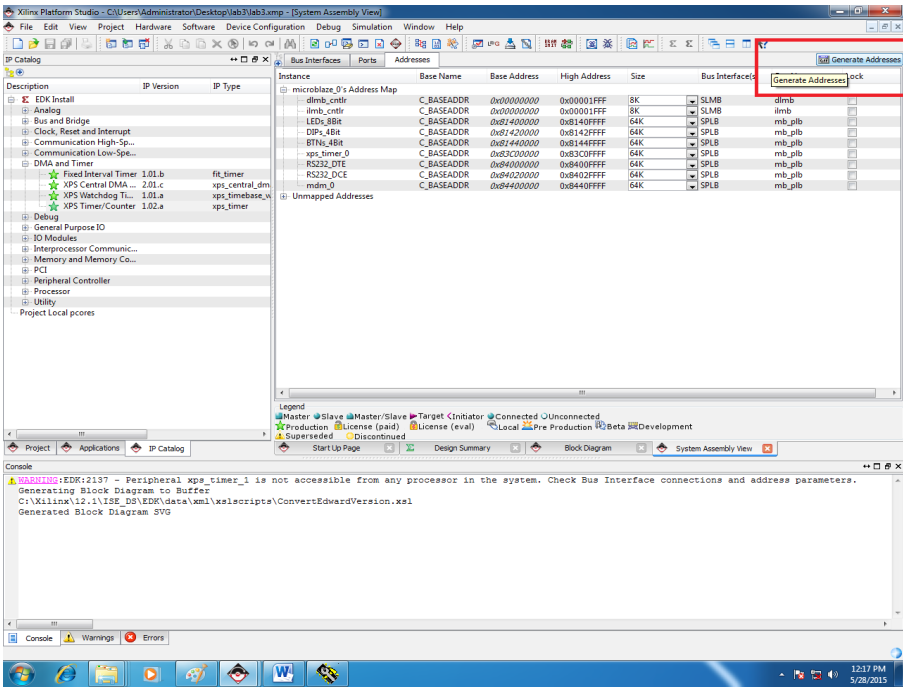
Adding Timer into existing Design — Step 2



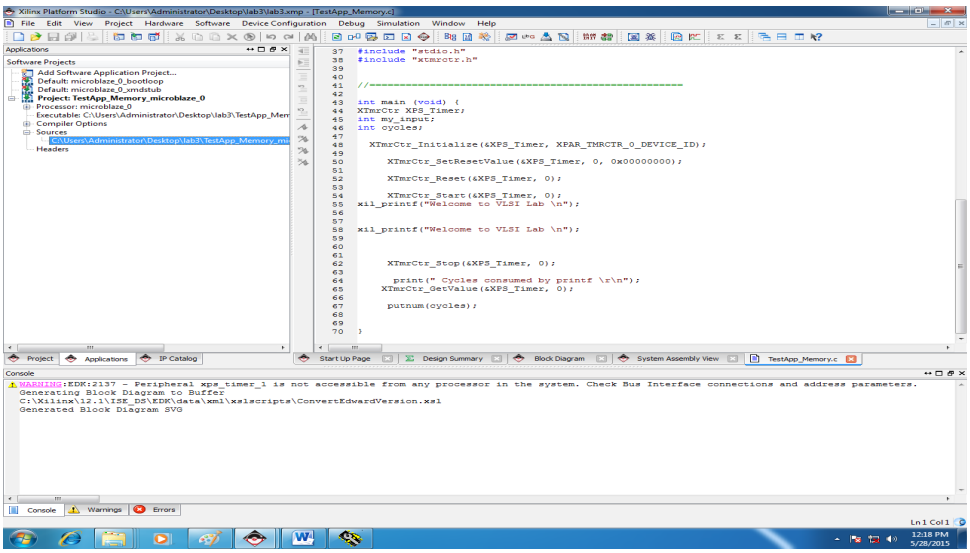
Connect timer with PLB bus



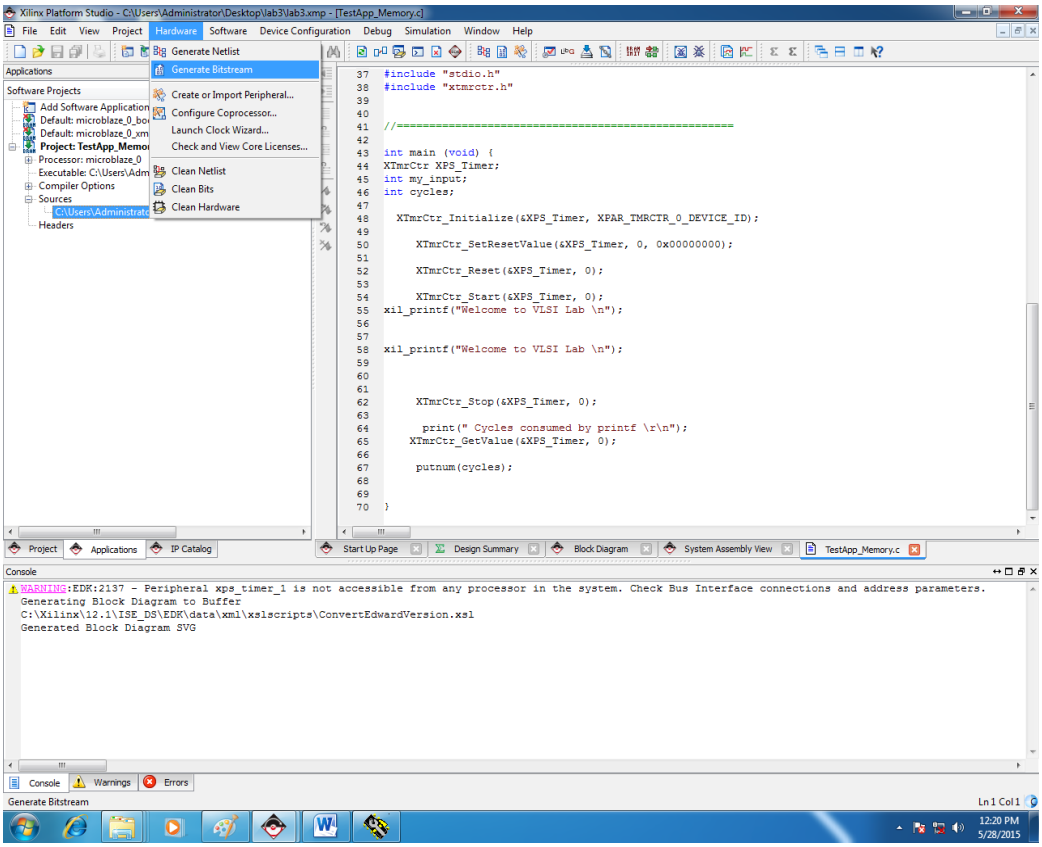
Generate Base Addresses

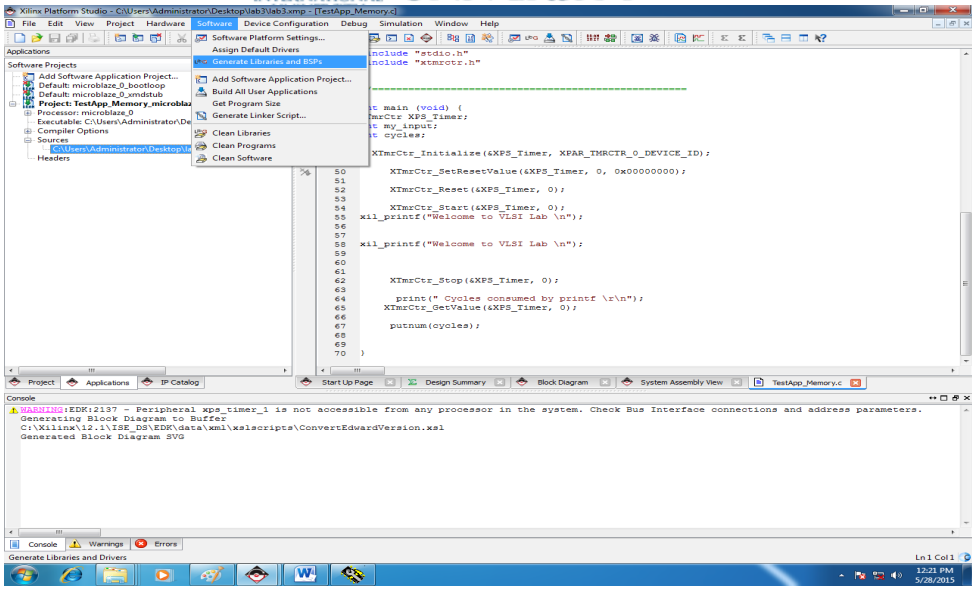


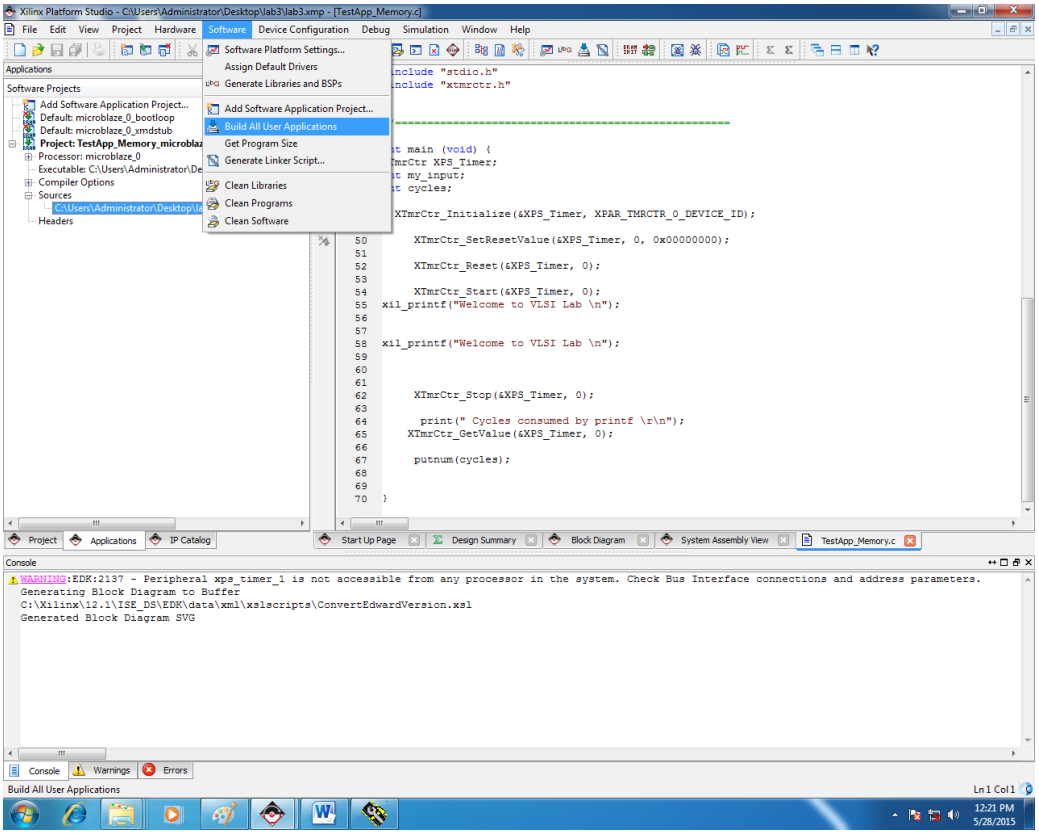
Writing Software for the added component



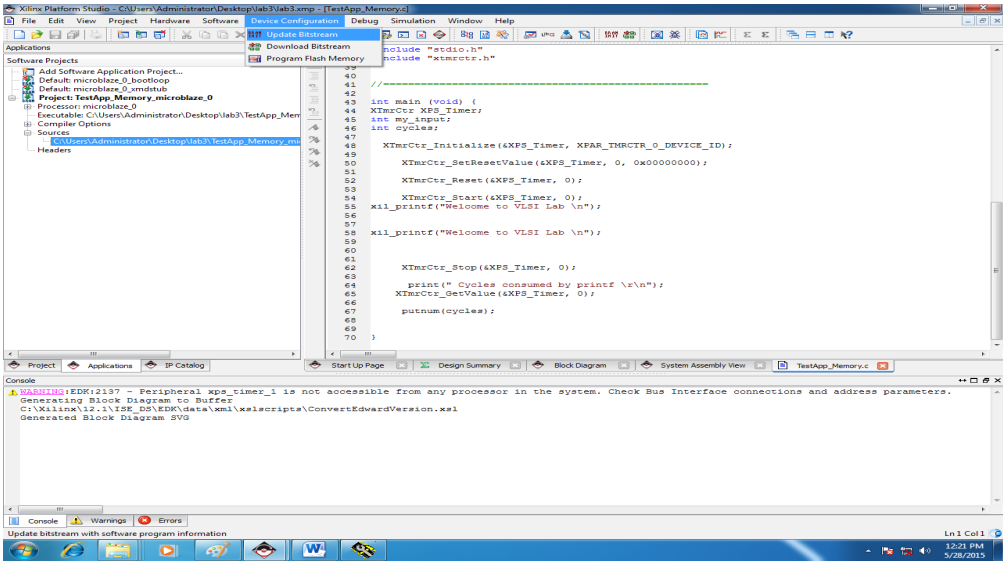
Generate Libraries and Linker Script for the design.

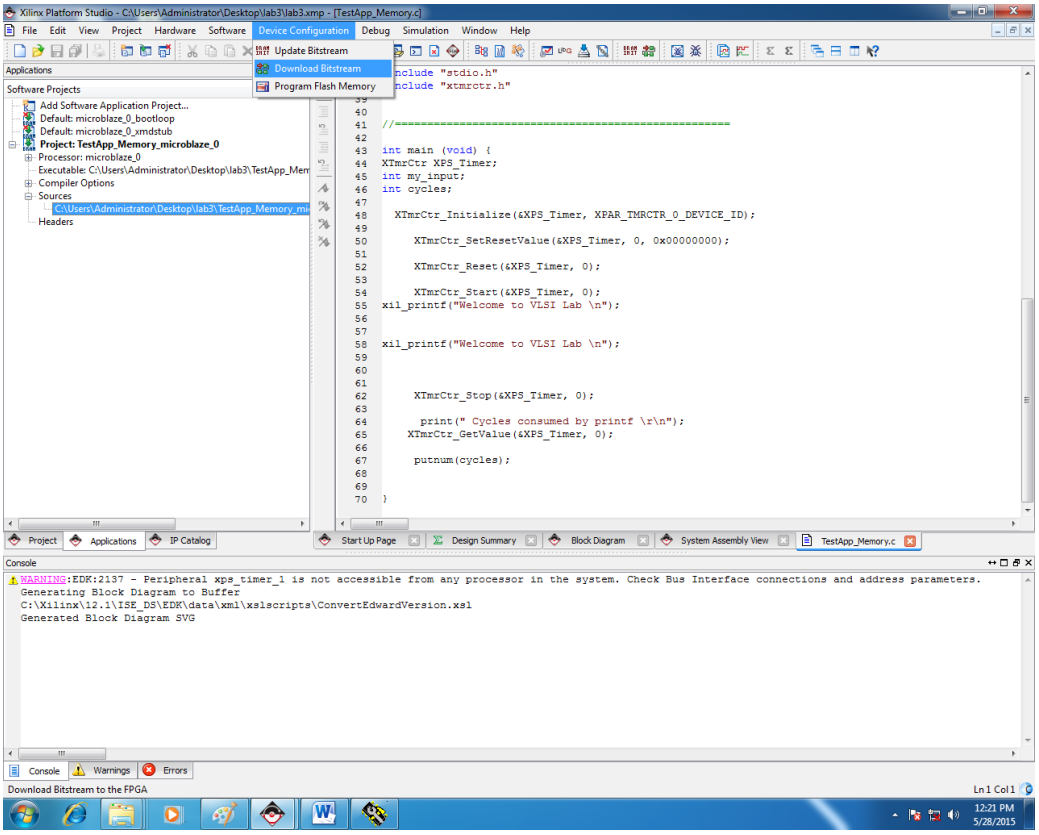



Xilinx Platform Studio - C:\Users\Administrator\Desktop\lab3\lab3.xmp - [TestApp_Memory.c]
File Edit View Project Hardware Software Device Configuration Debug Simulation Window Help
Software Platform Settings...
Assign Default Drivers
Generate Libraries and BSPs
Add Software Application Project...
Build All User Applications
Get Program Size
Generate Linker Script...
Clean Libraries
Clean Programs
Clean Software
Software Projects
Add Software Application Project...
Default: microblaze_0_bootloop
Default: microblaze_0_xmdstub
Project: TestApp_Memory_microblaz
Processor: microblaze_0
Executable: C:\Users\Administrator\De
Compiler Options
Sources
C:\Users\Administrator\Desktop\la
Headers
#include "stdio.h"
#include "xtmrctr.h"
int main (void) {
XTmrCtr XPS_Timer;
int my_input;
int cycles;
XTmrCtr_Initialize(&XPS_Timer, XPAR_TMRCTR_0_DEVICE_ID);
XTmrCtr_SetResetValue(&XPS_Timer, 0, 0x00000000);
XTmrCtr_Reset(&XPS_Timer, 0);
XTmrCtr_Start(&XPS_Timer, 0);
xil_printf("Welcome to VLSI Lab \n");
xil_printf("Welcome to VLSI Lab \n");
XTmrCtr_Stop(&XPS_Timer, 0);
print(" Cycles consumed by printf \r\n");
XTmrCtr_GetValue(&XPS_Timer, 0);
putnum(cycles);
}
Project Applications IP Catalog
Start Up Page Design Summary Block Diagram System Assembly View TestApp_Memory.c
Console
WARNING:EDK:2137 - Peripheral xps_timer_1 is not accessible from any processor in the system. Check Bus Interface connections and address parameters.
Generating Block Diagram to Buffer
C:\Xilinx\12.1\ISE_DS\EDK\data\xml\xslscripts\ConvertEdwardVersion.xsl
Generated Block Diagram SVG
Console Warnings Errors
Generate Libraries and Drivers
Ln 1 Col 1
12:21 PM
5/28/2015


Xilinx Platform Studio - C:\Users\Administrator\Desktop\lab3\lab3.xmp - [TestApp_Memory.c]
File Edit View Project Hardware Software Device Configuration Debug Simulation Window Help
Software Platform Settings...
Assign Default Drivers
Generate Libraries and BSPs
Add Software Application Project...
Build All User Applications
Get Program Size
Generate Linker Script...
Clean Libraries
Clean Programs
Clean Software
Software Projects
Add Software Application Project...
Default: microblaze_0_bootloop
Default: microblaze_0_xmdstub
Project: TestApp_Memory_microblaz
Processor: microblaze_0
Executable: C:\Users\Administrator\De
Compiler Options
Sources
C:\Users\Administrator\Desktop\la
Headers
#include "stdio.h"
#include "xtmrctr.h"
int main (void) {
XTmrCtr XPS_Timer;
int my_input;
int cycles;
XTmrCtr_Initialize(&XPS_Timer, XPAR_TMRCTR_0_DEVICE_ID);
XTmrCtr_SetResetValue(&XPS_Timer, 0, 0x00000000);
XTmrCtr_Reset(&XPS_Timer, 0);
XTmrCtr_Start(&XPS_Timer, 0);
xil_printf("Welcome to VLSI Lab \n");
xil_printf("Welcome to VLSI Lab \n");
XTmrCtr_Stop(&XPS_Timer, 0);
print(" Cycles consumed by printf \r\n");
XTmrCtr_GetValue(&XPS_Timer, 0);
putnum(cycles);
}
Project Applications IP Catalog
Start Up Page Design Summary Block Diagram System Assembly View TestApp_Memory.c
Console
WARNING:EDK:2137 - Peripheral xps_timer_1 is not accessible from any processor in the system. Check Bus Interface connections and address parameters.
Generating Block Diagram to Buffer
C:\Xilinx\12.1\ISE_DS\EDK\data\xml\xslscripts\ConvertEdwardVersion.xsl
Generated Block Diagram SVG
Console Warnings Errors
Build All User Applications
Ln 1 Col 1
12:21 PM
5/28/2015

Port design to FPGA





Source code

```
// Located in: microblaze_0/include/xparameters.h
#include "xparameters.h"
#include "xgpio.h"
#include "stdio.h"
#include "xtmrctr.h"


//====================================================

int main (void) {
XTmrCtr XPS_Timer;
int my_input;
int cycles;
  XTmrCtr_Initialize(&XPS_Timer, XPAR_TMRCTR_0_DEVICE_ID);
  XTmrCtr_SetResetValue(&XPS_Timer, 0, 0x00000000);
  XTmrCtr_Reset(&XPS_Timer, 0);
 XTmrCtr_Start(&XPS_Timer, 0);

 xil_printf("Welcome to VLSI Lab \n");
 xil_printf("Welcome to VLSI Lab \n");

XTmrCtr_Stop(&XPS_Timer, 0);
        print(" Cycles consumed by printf \r\n");
XTmrCtr_GetValue(&XPS_Timer, 0);
 putnum(cycles);

}
```