

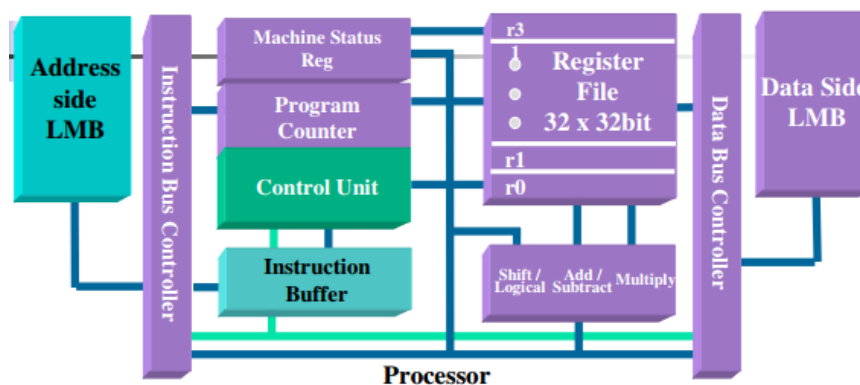
EE25266 – ASIC/FPGA Chip Design

Dr. Mahdi Shabany
Sharif University of Technology

Assignment #3 Introduction to MicroBlaze

Overview

This Lab is an introduction to Embedded System development with the Spartan-6 family FPGAs. In this session you will design a custom embedded hardware system and develop simple application software. A soft processor is a microprocessor core that can be wholly implemented using logic synthesis. It can be implemented via different semiconductor devices containing programmable logic like FPGAs. As a well-known case we will work with The MicroBlaze core. The MicroBlaze is a soft microprocessor core designed for Xilinx FPGAs from Xilinx. As a soft-core processor, MicroBlaze is implemented entirely in the general-purpose memory and logic fabric of Xilinx FPGAs. One sample microarchitecture is shown below:



The steps in this lab are as follows:

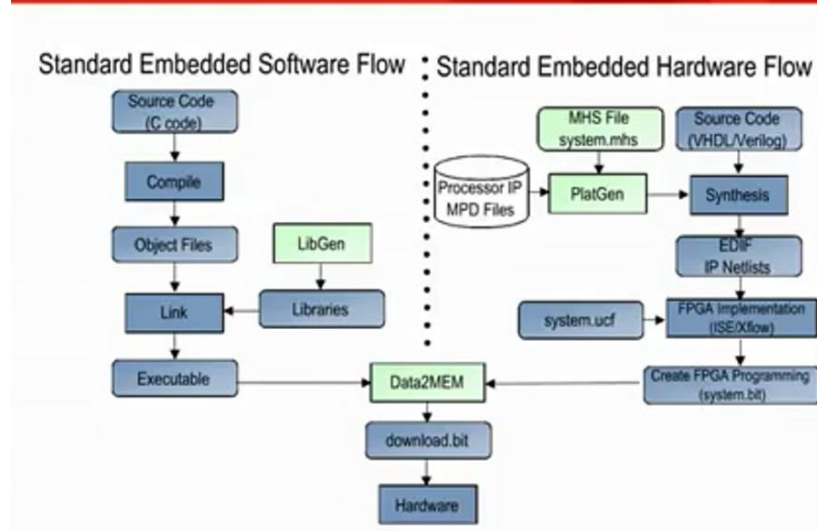
1. Create an Embedded system with the MicroBlaze processor for the Spartan-6 Embedded Kit (Atlys board) using Base System Builder (BSB) wizard.
2. Generate and Download the FPGA design bitstream on the board using Xilinx Platform Studio (XPS)
3. Create a simple software application using Xilinx Software Development Kit (SDK)

You may opt to watch this video if you are interested:

<https://www.youtube.com/watch?v=AHNKXriow90>

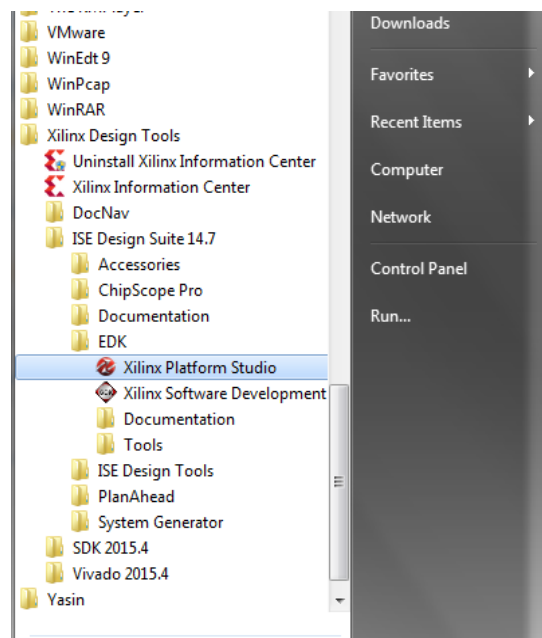
Design flow is depicted in following figure:

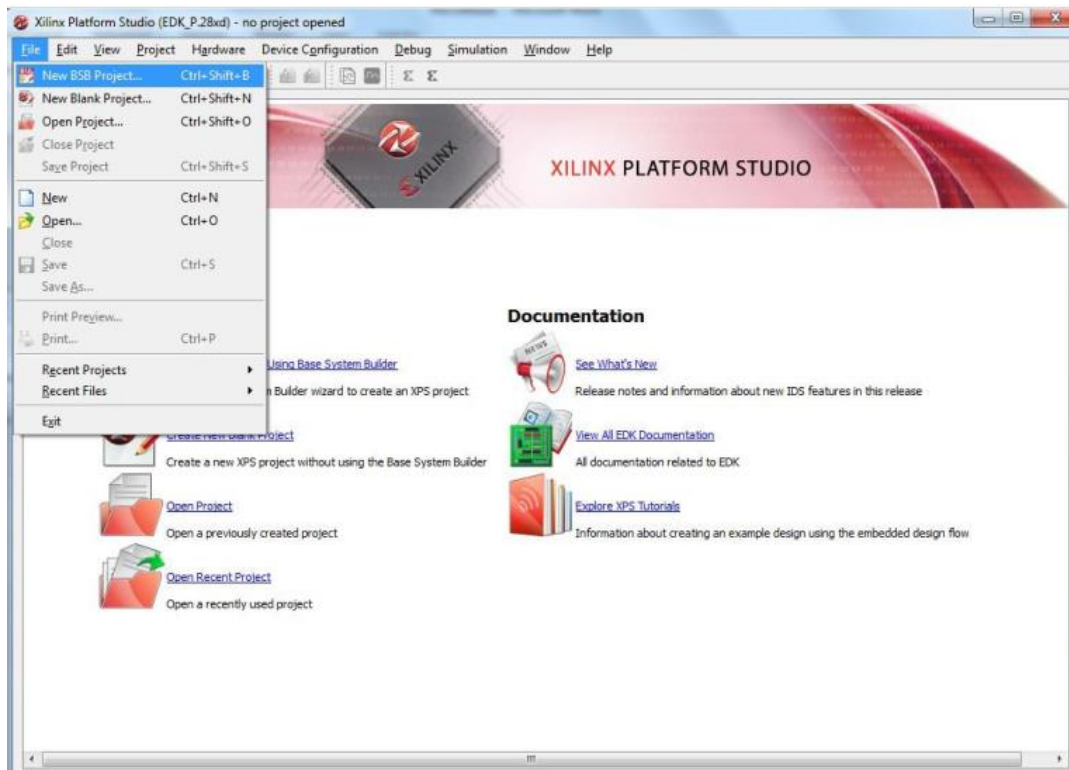
Detailed EDK Design Flow



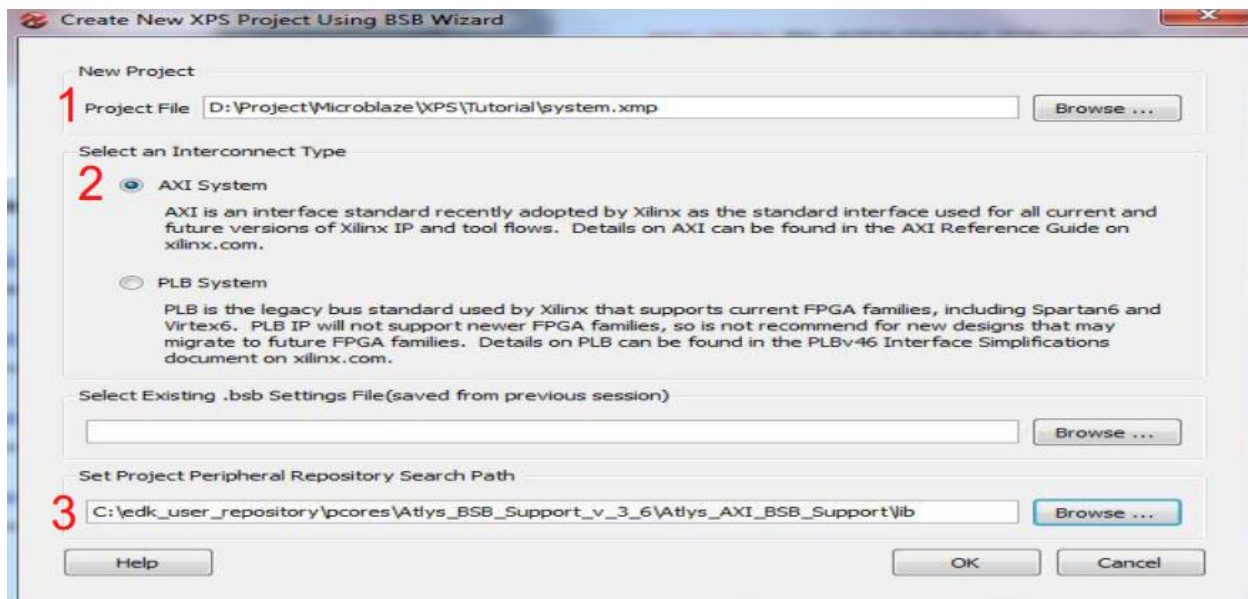
Part I - Create Custom Embedded System – Hardware

First of all, you should install and open Xilinx XPS to start designing our desirable hardware core. Xilinx XPS is a part of Xilinx EDK and is normally provided along with Xilinx ISE. (You should install Embedded or System edition of ISE design suite to have access to it and this is normally the case). If you have already installed the software, please find it in this menu:



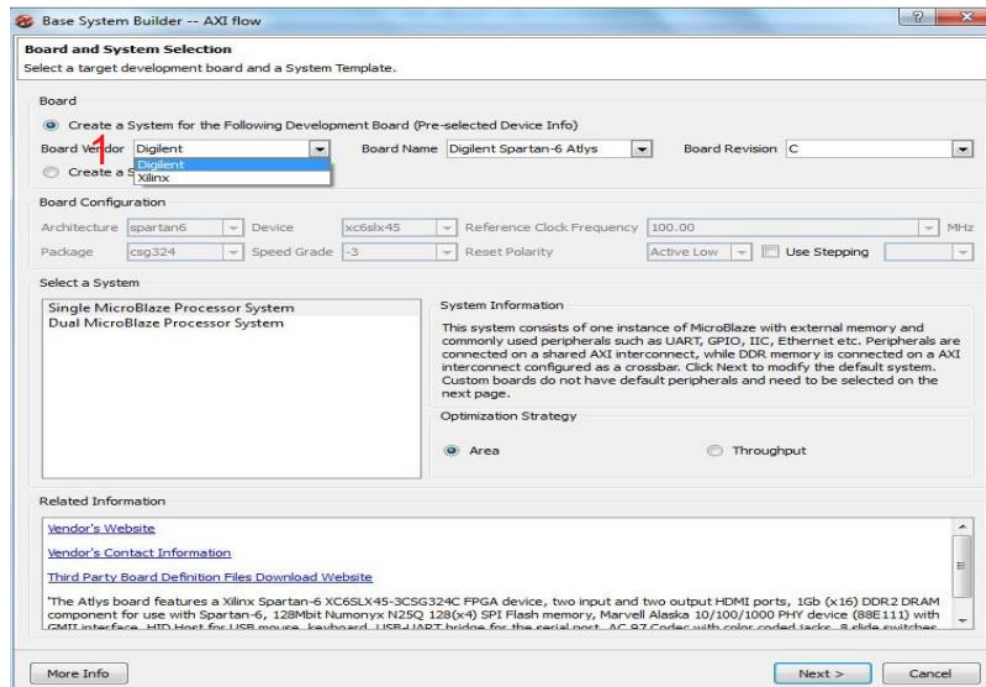


From file menu select New BSB project. Alternatively you can select [Create New Project Using Base System Builder](#) in the welcome pane. BSB or Base System Builder is a wizard procedure that helps us to build our base system along with other components/peripherals. Now set your project directory in section 1 of the appearing wizard:

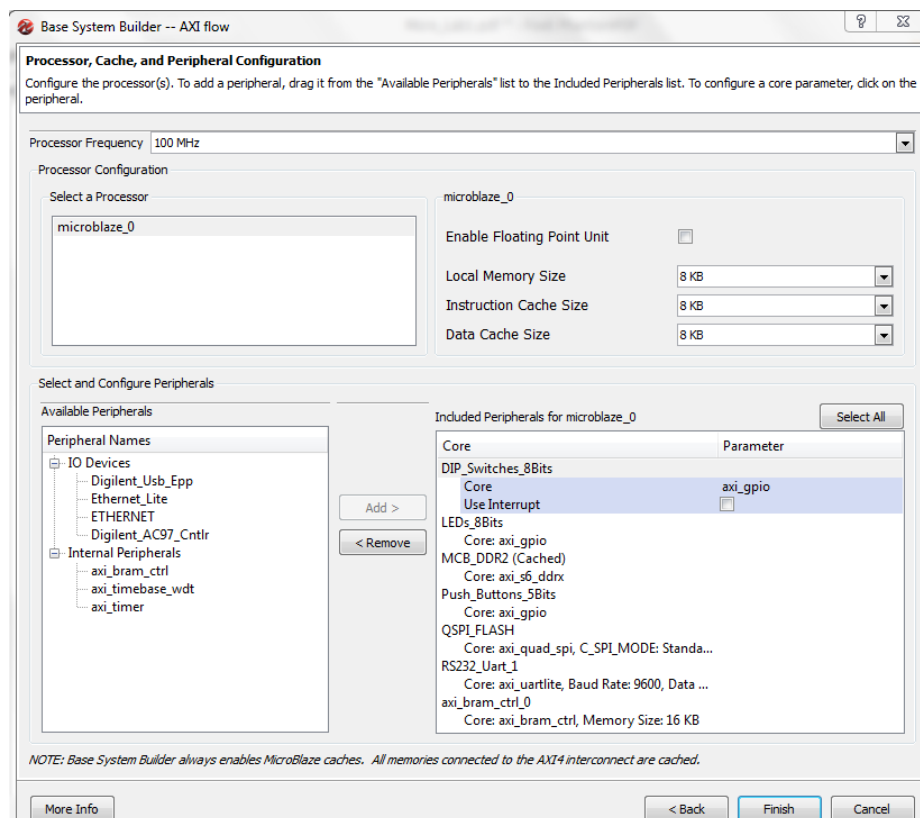


We will use AXI (Advanced Extensible Interface) bus which is defined under AMBA 3 standard which you might have heard it before.

At last, you should specify board support files which is provided for you along with this file. (It is not necessary to be in the exact path of figure!) Select OK to go to next wizard.



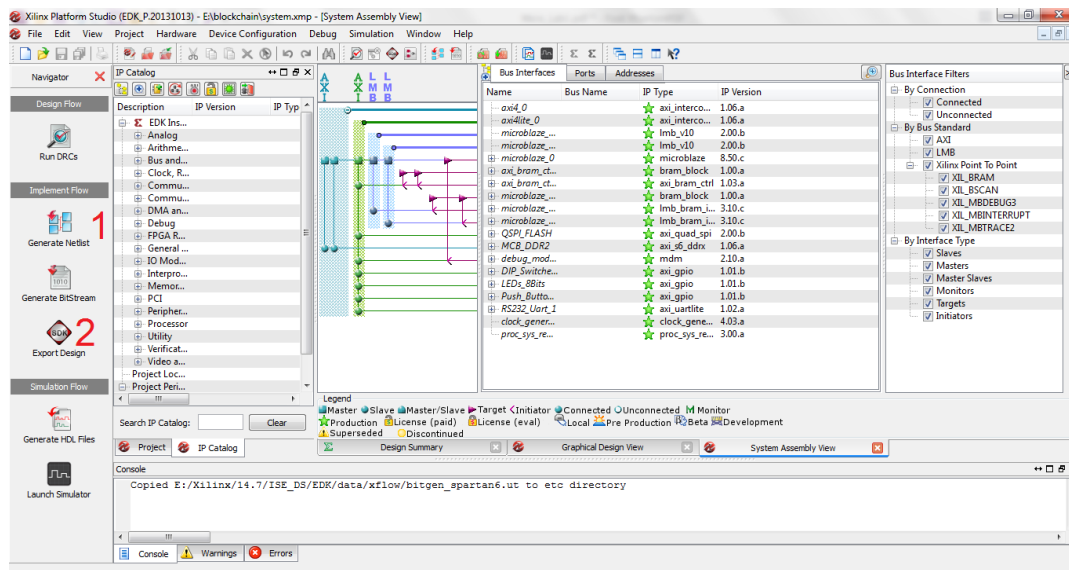
Select Digilent as board provider and continue to next wizard.



Select peripherals like the above figure. In this lab we shall not use RS232 (serial interface) but you may prefer adding it for future possible labs. Click on finish to start the next phase of adventure!

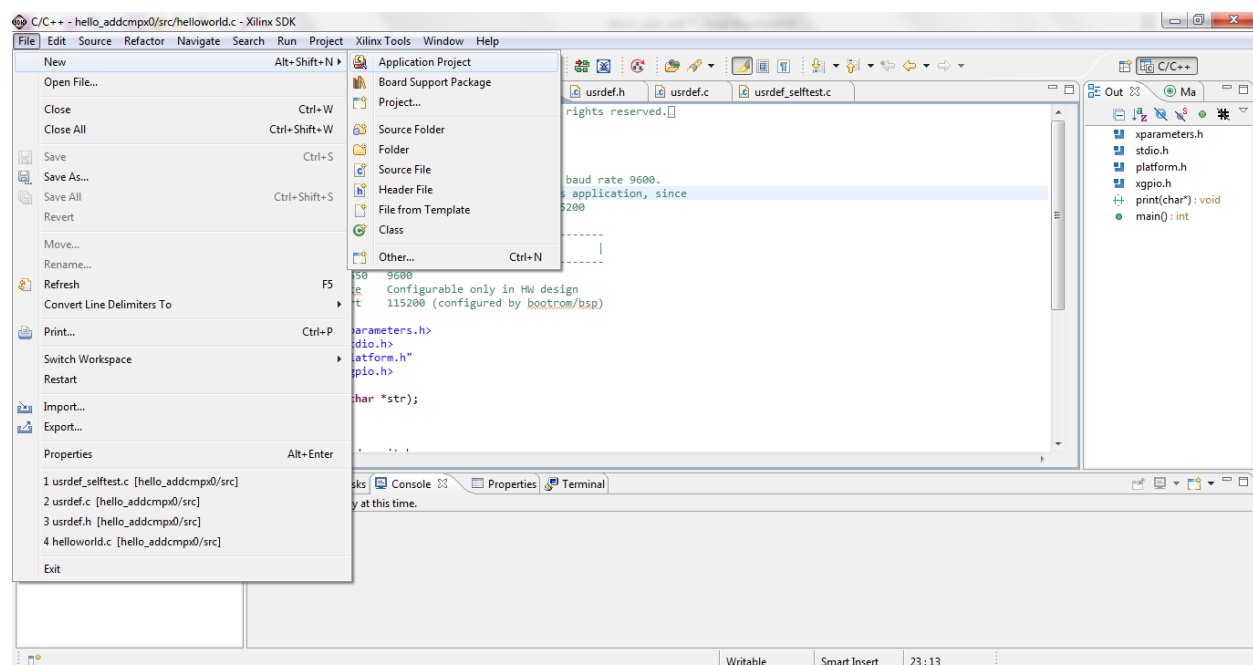
You can change peripheral options and settings in the new window. But leave them as are for now. Select Generate Netlist and wait until it finishes. Please be patient as this may take several minutes. Thereafter Select Export Design and select export & launch SDK in prompting window.

Note: If Xilinx SDK is already open, close it before selecting export & launch SDK



Part II – Developing The Software

Now we will be redirected to Xilinx SDK tool.



Select File->New-> Application project. Choose a name for your project and click on next. Select hello world as a template and finish the wizard.

Source codes can be found under src directory. Edit helloworld.c so as your program displays switch values on LEDs. Use the following guides to complete the code:

1. Add these two libraries:

```
#include <xparameters.h>
#include <xgpio.h>
```

2. Define two variables of type XGpio.

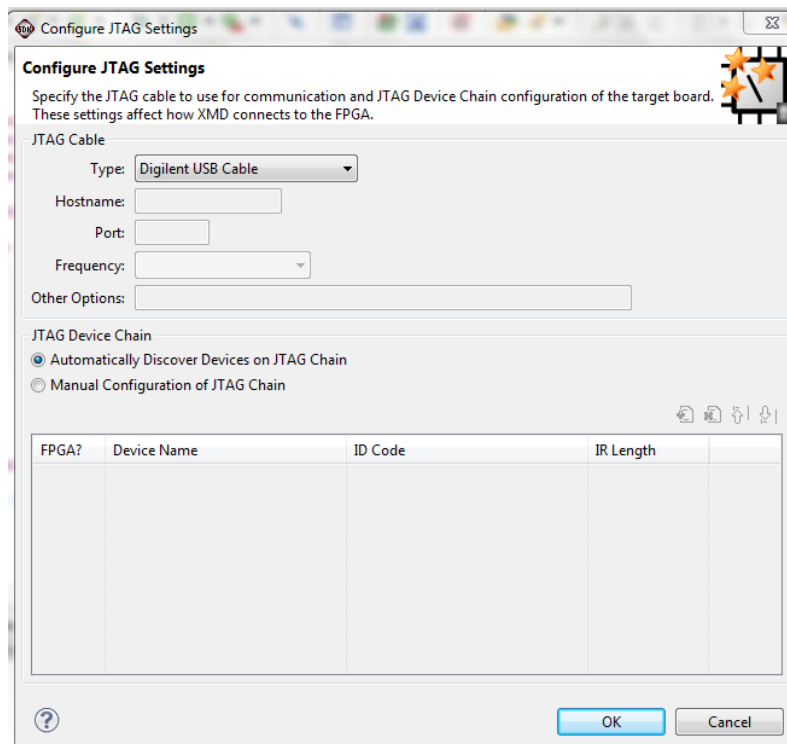
3. Assign these variables to switch and LED interface with the XGpio_Initialize function. (Search web to find out how to do this)

4. Set appropriate direction with XGpio_SetDataDirection function.

5. Read data from switches with XGpio_DiscreteRead. Similarly write to LEDs with XGpio_DiscreteWrite.

Now from project menu click on Build all to compile your codes. After that you are ready to download bitstream to FPGA and run the software. In SDK, from Xilinx tools select Program FPGA. Then from Run menu click on Run As->Launch on hardware.

Tip: You may encounter this error from SDK: **JTAG cable not found**. In the case, you should select Configure JTAG settings from Xilinx tools menu and specify the settings like this:



Part III – Add Two Variables

Change your C code to add right half of switches to left half and display the result on LEDs. The TA will ask you to change the code in lab session.