University of Engineering and Technology, Peshawar
Department of Computer Systems Engineering.
Course Lab: CSE-308 Digital System Design

Zubair Khan **19** PWCSE **1797**
Section A
Batch 21 (Spring_2022)
Submitted to Engr. Madiha Sher

LAB # 3 TITLE

INTRODUCTION TO XILINX ISE AND SPARTAN 6 BOARD

Objectives(s):

- Introduction to FPGA
- Introduction to Xilinx ISE

Software used:

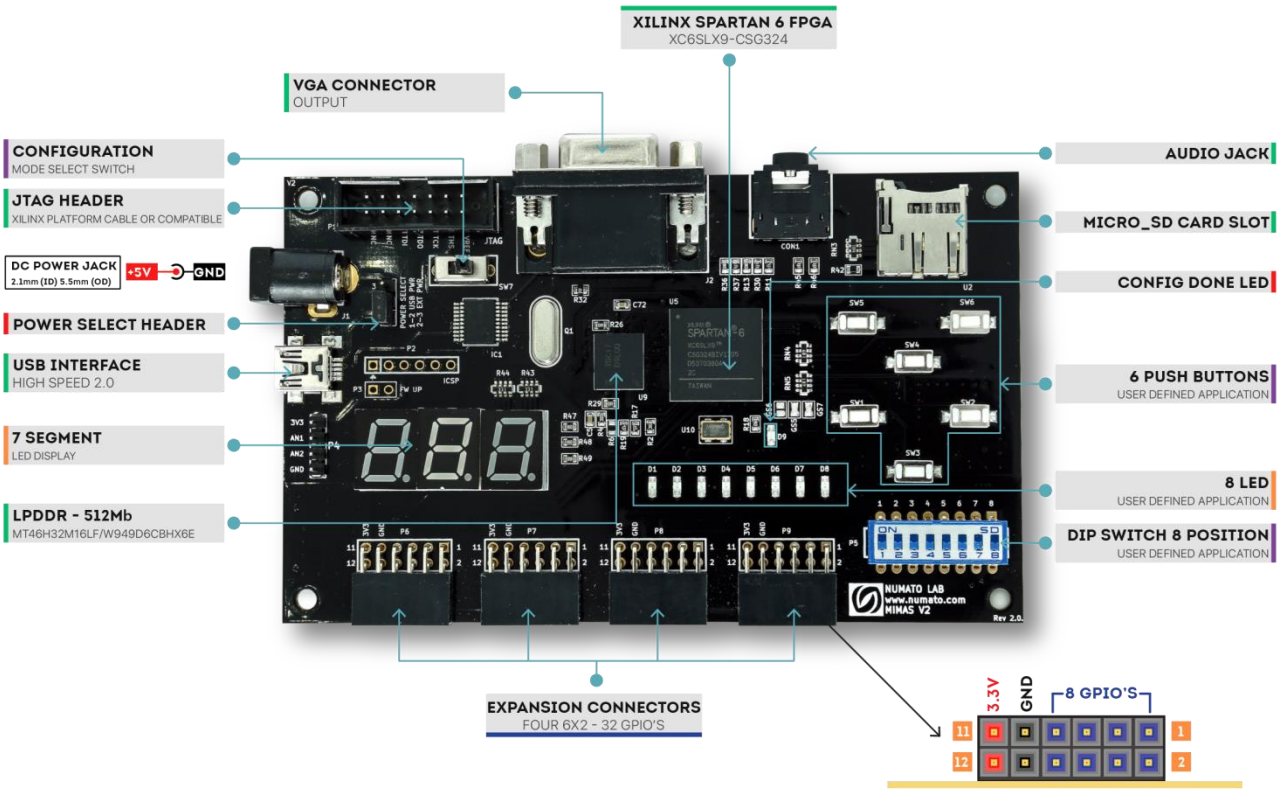
- Xilinx ISE

FPGA:

FPGAs are programmable digital logic circuits. It can be programmed to do almost any digital function. There are at least 5 companies making FPGAs in the world. Xilinx is the biggest name in the FPGA world.
The FPGA kits available in the lab are Mimas V2 Spartan 6 FPGA Development Board.

Mimas V2 Spartan 6 FPGA Development Board:

MIMAS V2 is a feature-packed yet low-cost FPGA Development board featuring Xilinx Spartan 6 FPGA. MIMAS V2 is specially designed for experimenting and learning system design with FPGAs. This development board features SPARTAN XC6SLX9 CSG324 FPGA with onboard 512Mb DDR SDRAM. The USB 2.0 interface provides fast and easy configuration download to the onboard SPI flash. No need to buy an expensive programmer or special downloader cable to download the bitstream to the board.



Features:

- FPGA: Spartan XC6SLX9 in CSG324 package
- DDR Memory: 166MHz 512Mb LPDDR (MT46H32M16LF/W949D6CBHX6E)
- Flash memory: 16 Mb SPI flash memory (M25P16)
- USB 2.0 interface for On-board flash programming
- FPGA configuration via JTAG and USB
- 8 LEDs, Six Push Buttons and 8 way DIP switch for user-defined purposes
- VGA Connector
- Stereo Jack
- Micro SD Card Adapter
- Three-Digit Seven Segment Displays
- 32 IOs for user-defined purposes
- Four 6×2 Expansion Connectors
- Onboard voltage regulators for single power rail operation

Applications:

- Product Prototype Development
- Signal Processing
- Learning Digital Electronics
- Educational tool for schools and universities

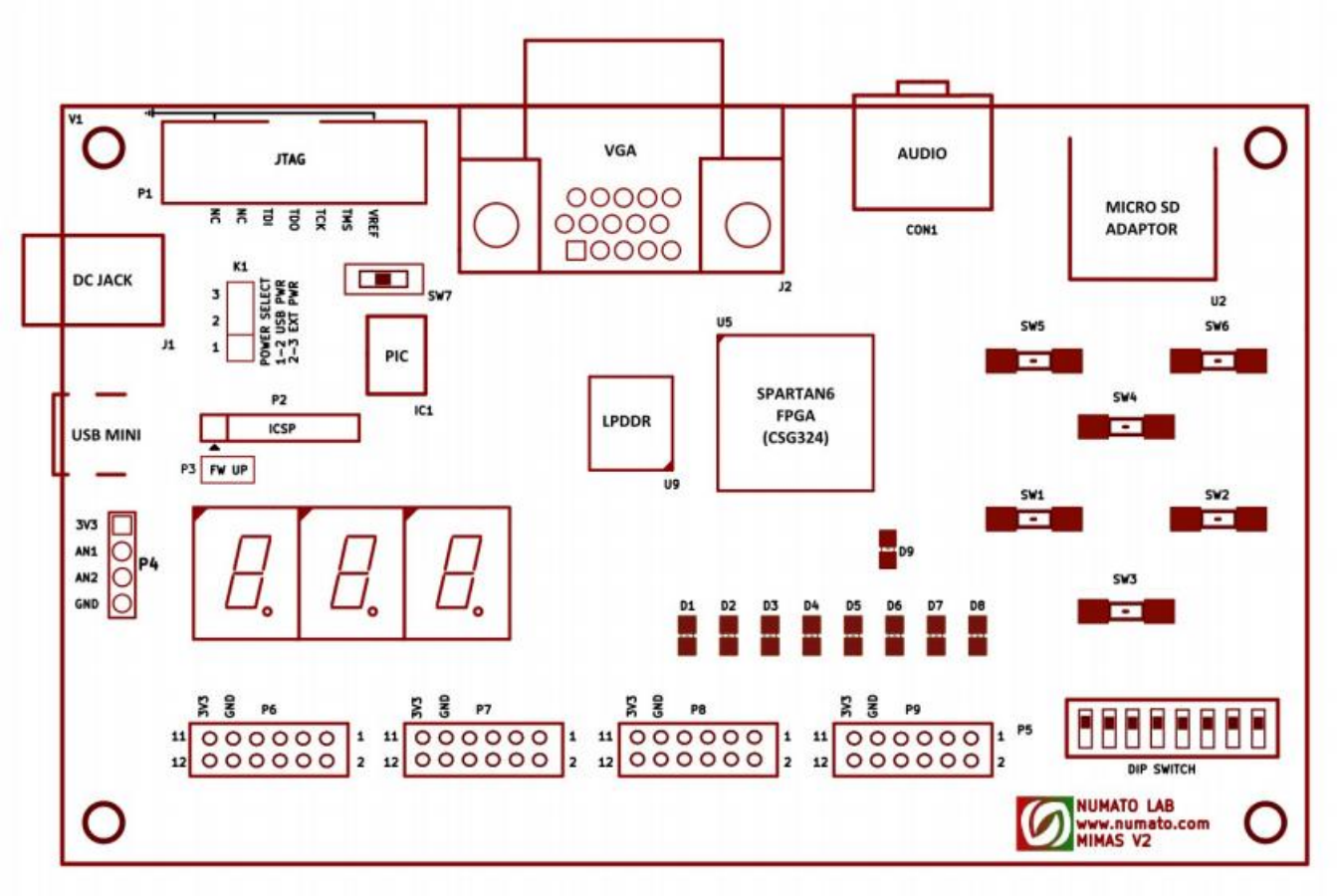
2. How to Use Mimas V2 Spartan 6 FPGA Development Board?

2.1. Components/Tools Required

Along with the module, you may need the items in the list below for easy and fast installation.

1. USB A to Mini B cable.
2. DC Power supply (Optional).

2.2.Connection Diagram



2.3 USB Interface

The onboard full-speed USB controller helps a computer to communicate with this module. Use a USB A to Mini B cable to connect with a PC. By default, the module is powered from USB so make sure not to overcrowd unpowered USB hubs.

2.4 DC Power Supply

This module uses a +5V power supply to function properly. By default, the board is configured to use the +5V supply from USB. So an external +5V power is not required unless the USB port is unable to supply enough current. In most cases, USB ports are capable of providing enough current for the module. The current requirement for this board largely depends on your application. Please consult the FPGA datasheet for more details on power requirements. If for any reason, an external 5V power supply needs to be used for the module, the Power select jumper should be configured properly before connecting the

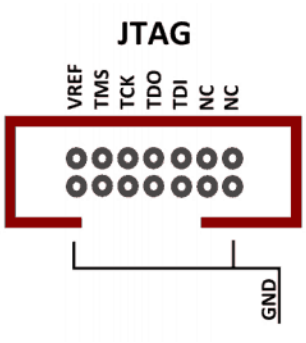
power supply. Please refer to the marking on the board for more details.

2.5 Power Select

The Power Select header K1 is used to configure the power source for the board. The jumper in pin 1 and 2 is shorted to switch the power source to the onboard USB port and pin 2 and 3 to use the external DC power.

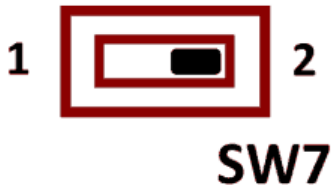
2.6.JTAG Connector

A JTAG connector provides access to FPGA's JTAG pins. A XILINX platform cable can be used for JTAG programming.



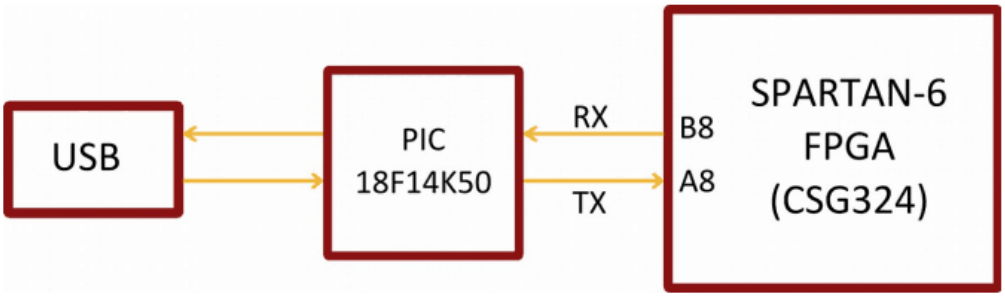
2.7.Configuration Mode Selection

Slide switch SW7 is used to switch between the USB configuration mode and UART. Slide the switch to Position 1 to download bitstream through USB configuration tool and Position 2 to use the interface as a UART in order to communicate from your code in FPGA with the PC. By default, the board is shipped with a slide switch position in the USB configuration tool mode.



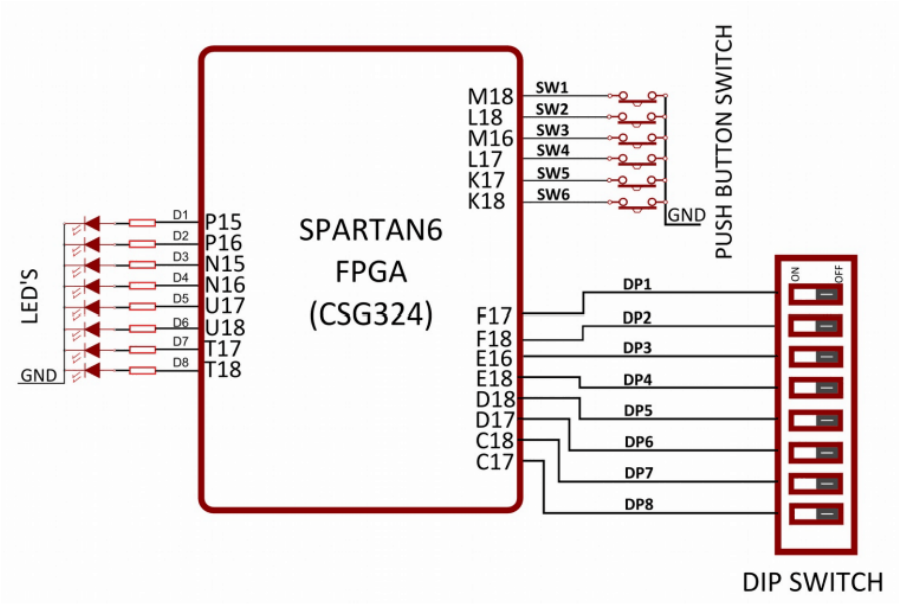
2.8 UART

The MIMAS V2 includes USB-UART, which helps to establish the communication between the code in the FPGA and any application running on the PC. Data can be sent and received from the FPGA by using Serial Terminal at baud rate 19200.



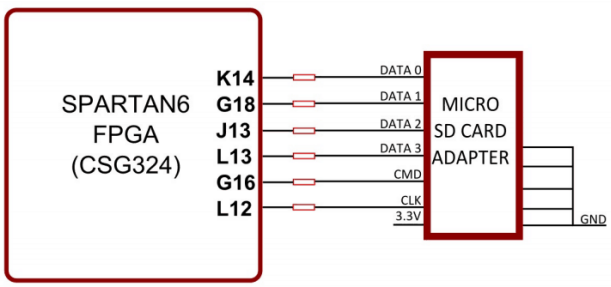
2.9 LED, Push Button and Dip Switch

MIMAS V2 has six push-button switches, an eight-position DIP switch and eight LEDs for human interaction. All switches are directly connected to Spartan 6 FPGA and can be used in your design with minimal effort.



2.10 Micro SD

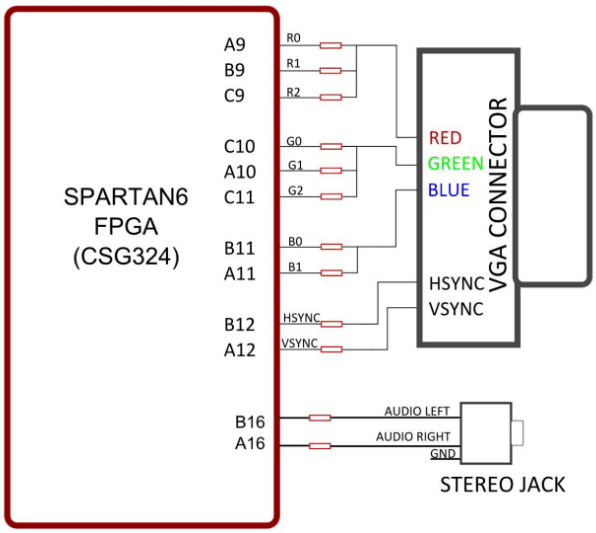
MIMAS V2 features a Micro SD adapter on-board. By installing a Micro SD card, you can add data logging, media storage and other file storage to your design.


SPARTAN6
FPGA
(CSG324)
K14
G18
J13
L13
G16
L12
DATA 0
DATA 1
DATA 2
DATA 3
CMD
CLK
3.3V
MICRO
SD CARD
ADAPTER
GND

2.11 VGA and Audio

The VGA interface provides this board the ability to generate VGA signals from FPGA and display information on any Display/monitor that supports standard VGA connector. This VGA interface uses resistor network-based DAC for easy code implementation. This 8 bit VGA interface can display up to 256 colors.

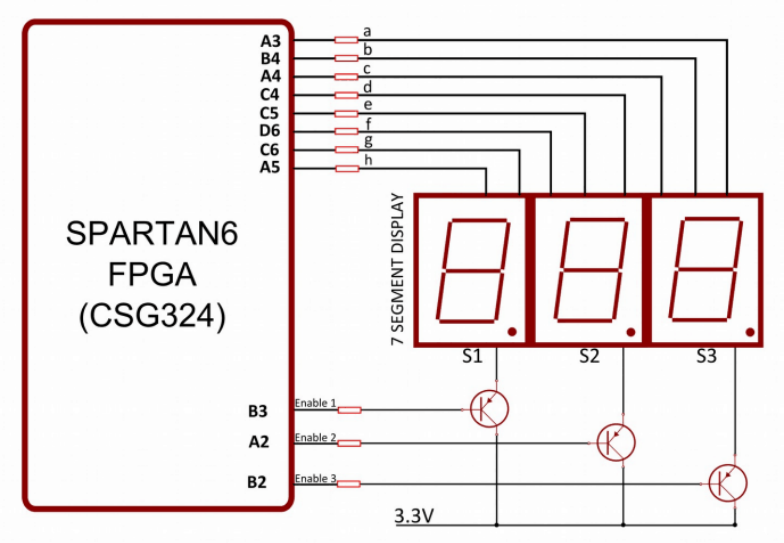
Two IOs on the FPGA is dedicated to generating two channels of audio. Different audio tones can be generated by using PWM and Frequency synthesis.



2.12 7 Segment LED Display

This board features three 7-segment LED display multiplexed for low pin count operation. Each module can be separately turned on and off with the three switching transistors.

Note: All signals (a, b, c, d, e, f, g, dot, enable 1, enable 2, enable 3) used for controlling 7-Segment display are active-low signals. So, for example, for displaying “8” in display-2, users need to drive Enable 2 to 0 as well as drive signals a, b, c, d, e, f to 0. All other signals need to be driven to 1.



XILINX:

The Integrated Software Environment (ISE™) is the Xilinx® design software suite that allows you to take your design from design entry through Xilinx device programming. The ISE Project Navigator manages and processes your design through the following steps in the ISE design flow.

Design Entry

Design entry is the first step in the ISE design flow. During design entry, you create your source files based on your design objectives. You can create your top-level design file using a Hardware Description Language (HDL), such as VHDL, Verilog, or ABEL, or using a schematic. You can use multiple formats for the lower-level source files in your design.

Note

If you are working with a synthesized EDIF or NGC/NGO file, you can skip design entry and synthesis and start with the implementation process.

Synthesis

After design entry and optional simulation, you run synthesis. During this step, VHDL, Verilog, or mixed language designs become netlist files that are accepted as input to the implementation step.

Implementation

After synthesis, you run design implementation, which converts the logical design into a physical file format that can be downloaded to the selected target device. From Project Navigator, you can run the implementation process in one step, or you can run each of the implementation processes separately. Implementation processes vary depending on whether you are targeting a Field Programmable Gate Array (FPGA) or a Complex Programmable Logic Device (CPLD).

Verification

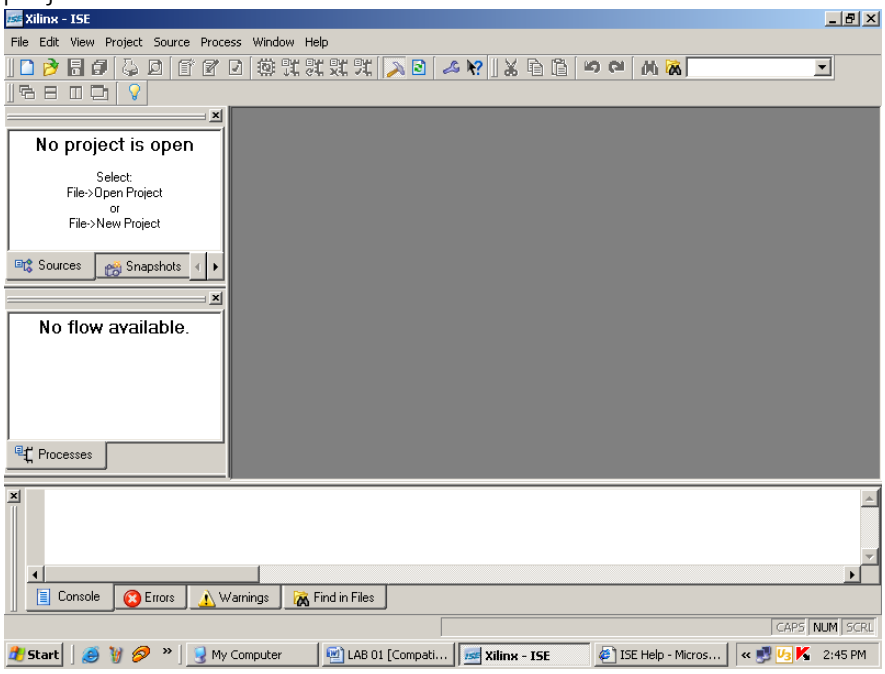
You can verify the functionality of your design at several points in the design flow. You can use simulator software to verify the functionality and timing of your design or a portion of your design. The simulator interprets VHDL or Verilog code into circuit functionality and displays logical results of the described HDL to determine correct circuit operation. Simulation allows you to create and verify complex functions in a relatively small amount of time. You can also run in-circuit verification after programming your device.

Device Configuration

After generating a programming file, you configure your device. During configuration, you generate configuration files and download the programming files from a host computer to a Xilinx device.

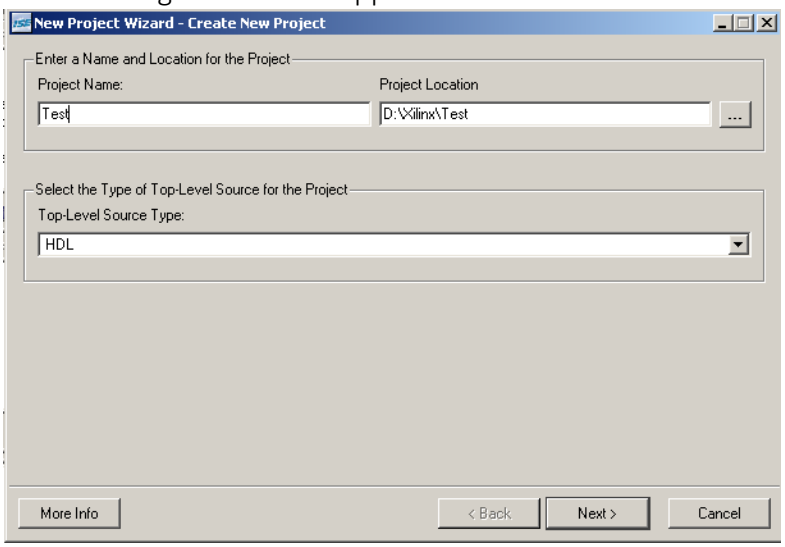
How to make new Project:

Steps for new project:

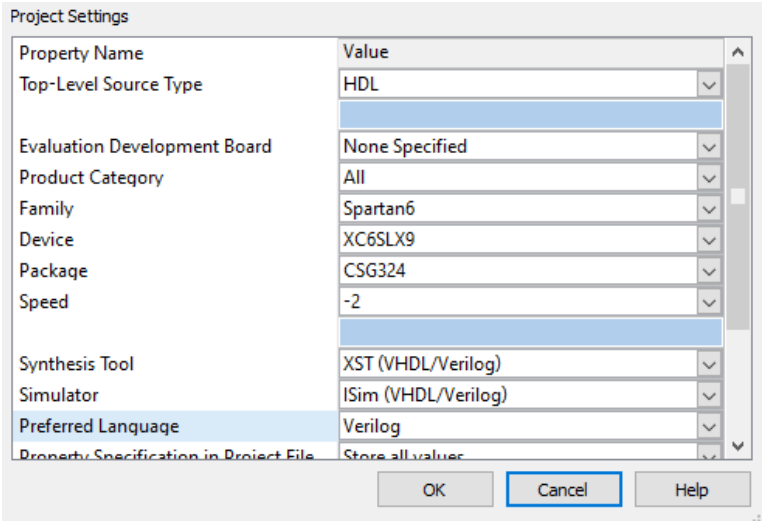


File New Project

The following window will appear

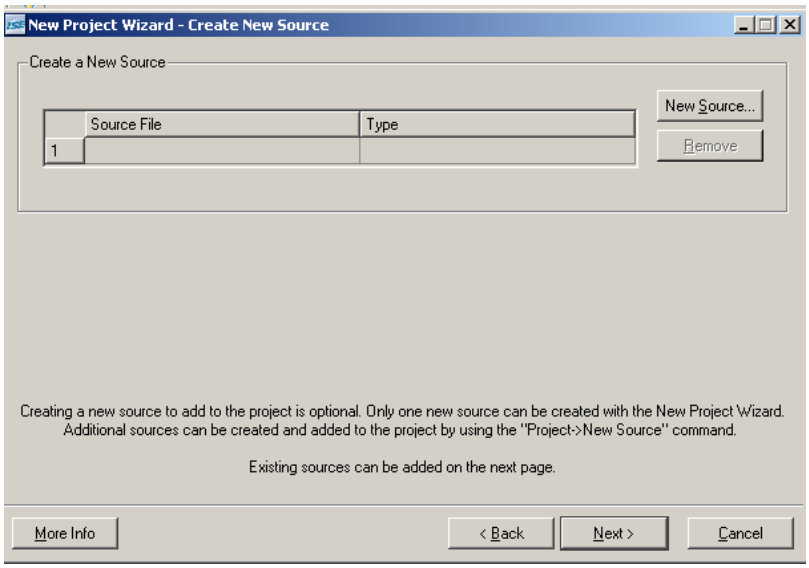


Enter the Project Name and Project Location and Press Next>
The following window will appear

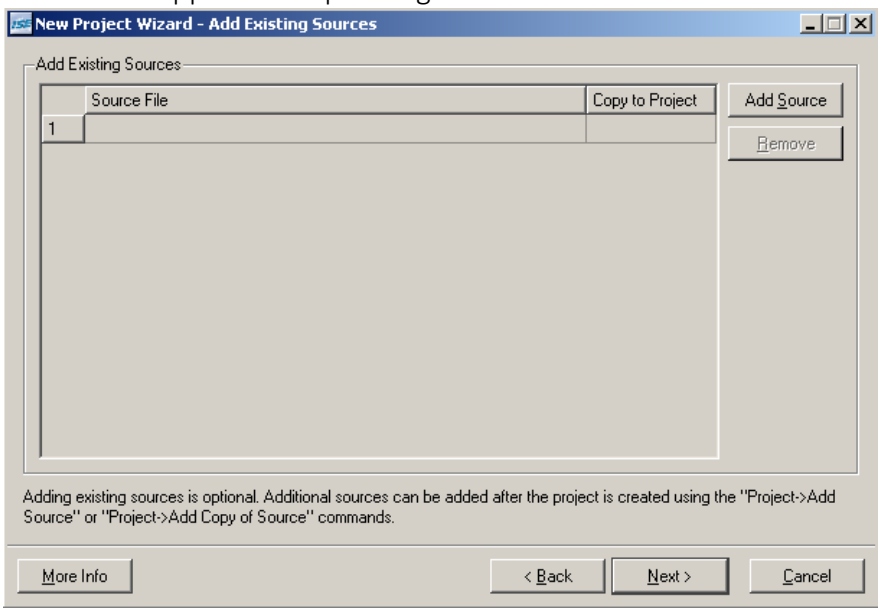


Select the options from the PULL DOWN lists as show above, and Press Next.

The following window will appear.

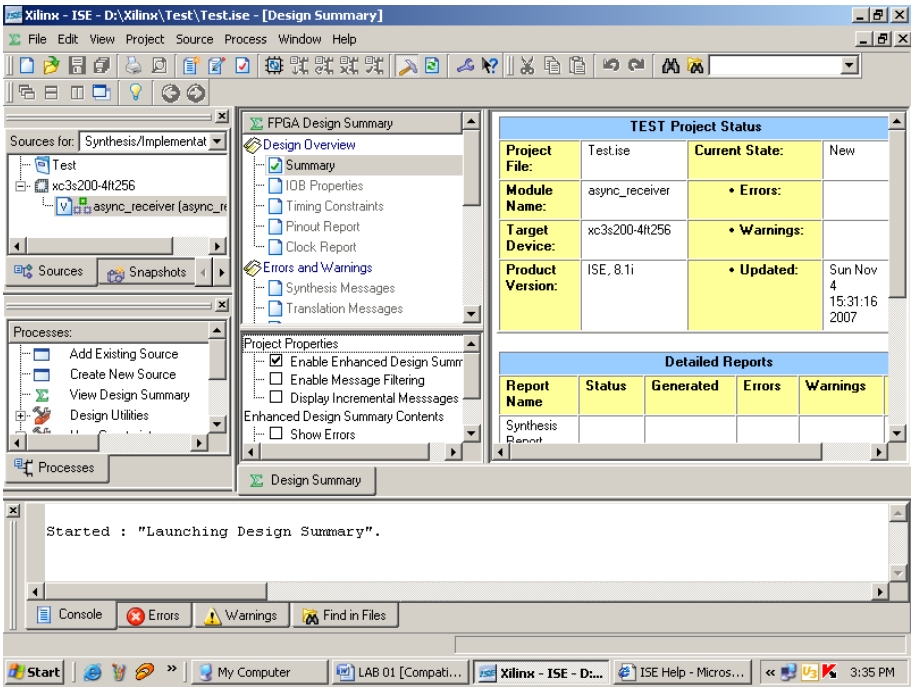


Now if you want to create a source file, then click New Source otherwise click Next> to add the existing Source file.
The following window will appear after pressing the Next>



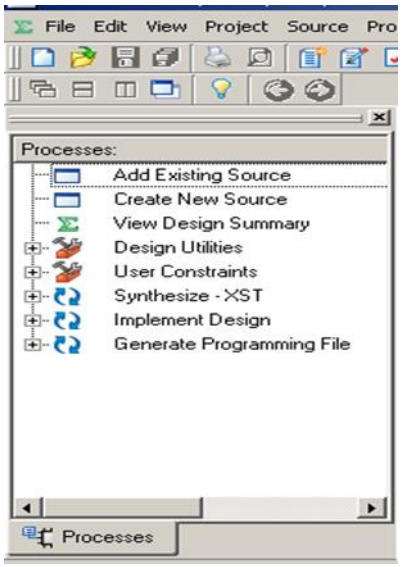
Click Add Source to add the source files and then press Next.

The next window will show the project summary, press Finish.



The code can be synthesized by the following steps.

1. First of all the Verilog module is created.
2. USER DEFINED CONSTRAINT file is created and added in the project
3. Verilog code is then synthesized
4. Design is implemented
5. Programming file is then generated as binary file which can be downloaded into the FPGA
   The Process windows has all the above options
6. Programming file is downloaded to FPGA using Mimas V2 Configuration tool for windows.



Lab Tasks:

Task1 :

Develop a program to control on Board LEDs using on board available switches.

Code:

```
//////////////////////////////////////
module task1(
    input [7:0] in,
    output [7:0] out
    );
assign out=in;
endmodule
```

Ucf File:

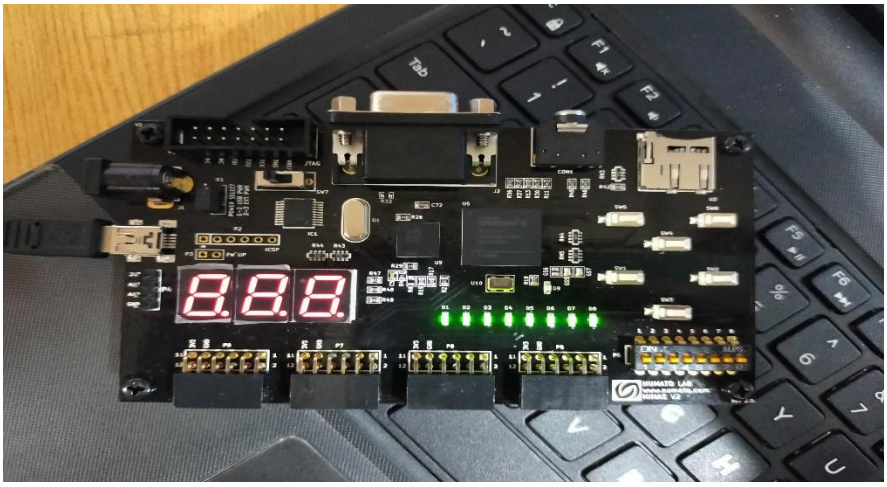
```
# PlanAhead Generated physical constraints |
NET "in[7]" LOC = F17;
NET "in[6]" LOC = F18;
NET "in[5]" LOC = E16;
NET "in[4]" LOC = E18;
NET "in[3]" LOC = D18;
NET "in[2]" LOC = D17;
NET "in[1]" LOC = C18;
NET "in[0]" LOC = C17;

# PlanAhead Generated IO constraints

NET "in[7]" PULLUP;
NET "in[6]" PULLUP;
NET "in[5]" PULLUP;
NET "in[4]" PULLUP;
NET "in[3]" PULLUP;
NET "in[2]" PULLUP;
NET "in[1]" PULLUP;
NET "in[0]" PULLUP;
NET "out[7]" SLEW = FAST;
NET "out[6]" SLEW = FAST;
NET "out[5]" SLEW = FAST;
NET "out[4]" SLEW = FAST;
NET "out[3]" SLEW = FAST;
NET "out[2]" SLEW = FAST;
NET "out[1]" SLEW = FAST;
NET "out[0]" SLEW = FAST;
```

I/O Pin Planning:

I/O Ports

Name	Direction	Neg Diff Pair	Site	Fixed	Bank	I/O Std	Vcco	Vref	Drive Stre...	Slew Type	Pull Type	Off-Chip T...	IN_TERM	OUT_TERM
All ports (16)														
in (8)	Input					1 default (LVCMOS25)					PULLUP*	NONE	NONE	
in[7]	Input		F17	☑		1 default (LVCMOS25)					PULLUP*	NONE	NONE	
in[6]	Input		F18	☑		1 default (LVCMOS25)					PULLUP*	NONE	NONE	
in[5]	Input		E16	☑		1 default (LVCMOS25)					PULLUP*	NONE	NONE	
in[4]	Input		E18	☑		1 default (LVCMOS25)					PULLUP*	NONE	NONE	
in[3]	Input		D18	☑		1 default (LVCMOS25)					PULLUP*	NONE	NONE	
in[2]	Input		D17	☑		1 default (LVCMOS25)					PULLUP*	NONE	NONE	
in[1]	Input		C18	☑		1 default (LVCMOS25)					PULLUP*	NONE	NONE	
in[0]	Input		C17	☑		1 default (LVCMOS25)					PULLUP*	NONE	NONE	
out (8)	Output					1 default (LVCMOS25)	2.500			12 FAST*	NONE	FP_VTT_50		NONE
out[7]	Output		P15	☑		1 default (LVCMOS25)	2.500			12 FAST*	NONE	FP_VTT_50		NONE
out[6]	Output		P16	☑		1 default (LVCMOS25)	2.500			12 FAST*	NONE	FP_VTT_50		NONE
out[5]	Output		N15	☑		1 default (LVCMOS25)	2.500			12 FAST*	NONE	FP_VTT_50		NONE
out[4]	Output		N16	☑		1 default (LVCMOS25)	2.500			12 FAST*	NONE	FP_VTT_50		NONE
out[3]	Output		U17	☑		1 default (LVCMOS25)	2.500			12 FAST*	NONE	FP_VTT_50		NONE
out[2]	Output		U18	☑		1 default (LVCMOS25)	2.500			12 FAST*	NONE	FP_VTT_50		NONE
out[1]	Output		T17	☑		1 default (LVCMOS25)	2.500			12 FAST*	NONE	FP_VTT_50		NONE

Output:



Task2 :

Develop a program that implements a 2x1 multiplexer on the board. Connect the inputs to switches and output to LEDs.

Code:

```
//////////////////////////////////////////////////////////////////////////////////
module task2(
    input I0,
    input I1,
    input sel,
    output out
    );

assign out=sel?I0:I1;
endmodule
```

Ucf File:

```
# PlanAhead Generated physical constraints

NET "I0" LOC = C17;
NET "I1" LOC = C18;
NET "sel" LOC = F17;
NET "out" LOC = T18;

# PlanAhead Generated IO constraints

NET "out" SLEW = FAST;
NET "I0" PULLUP;
NET "I1" PULLUP;
NET "sel" PULLUP;
```

I/O Pin Planning:

I/O Ports

Name	Direction	Neg Diff Pair	Site	Fixed	Bank	I/O Std	Vcco	Vref	Drive Stre...	Slew Type	Pull Type	Off-Chip T...	IN_TERM	OUT_TERM
All ports (4)														
Scalar ports (4)														
I0	Input		C17	☑		1 default (LVCMOS25)					PULLUP*	NONE	NONE	
I1	Input		C18	☑		1 default (LVCMOS25)					PULLUP*	NONE	NONE	
out	Output		T18	☑		1 default (LVCMOS25)	2.500		12	FAST*	NONE	FP_VTT_50		NONE
sel	Input		F17	☑		1 default (LVCMOS25)					PULLUP*	NONE	NONE	

Output: