# LAB 1

# INTRODUCTION TO XILINX ISE AND S3BOARD

**Objectives:**

Introduction to FPGA and Xilinx

**Software used:**

## XILINX

**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 our labs are SPARTAN-3 STARTER KIT BOARD.

**SPARTAN-3 STARTER KIT BOARD:**

The features of this kit are

200,000-gate Xilinx Spartan-3 XC3S200 FPGA in a 256-ball thin Ball Grid Array package (XC3S200FT256)

4,320 logic cell equivalents

Twelve 18K-bit block RAMs (216K bits)

Twelve 18x18 hardware multipliers

Four Digital Clock Managers (DCMs)

Up to 173 user-defined I/O signals

2Mbit Xilinx XCF02S Platform Flash, in-system programmable configuration PROM

1Mbit non-volatile data or application code storage available after FPGA configuration

Jumper options allow FPGA application to read PROM data or FPGA configuration from other sources

1M-byte of Fast Asynchronous SRAM (bottom side of board, see Figure 1-3)

Two 256Kx16 ISSI IS61LV25616AL-10T 10 ns SRAMs

Configurable memory architecture

Single 256Kx32 SRAM array, ideal for MicroBlaze code images

Two independent 256Kx16 SRAM arrays

Individual chip select per device

Individual byte enables

3-bit, 8-color VGA display port

9-pin RS-232 Serial Port

DB9 9-pin female connector (DCE connector)

Maxim MAX3232 RS-232 transceiver/translator

Uses straight-through serial cable to connect to computer or workstation serial port

Second RS-232 transmit and receive channel available on board test points

PS/2-style mouse/keyboard port

Four-character, seven-segment LED display

Eight slide switches

Eight individual LED outputs

Four momentary-contact push button switches

50 MHz crystal oscillator clock source (bottom side of board, see Figure 1-3)

Socket for an auxiliary crystal oscillator clock source

FPGA configuration mode selected via jumper settings

Push button switch to force FPGA reconfiguration (FPGA configuration happens automatically at power-on)

LED indicates when FPGA is successfully configured

Three 40-pin expansion connection ports to extend and enhance the Spartan-3 Starter Kit Board

FPGA serial configuration interface signals available on the A2 and B1 connectors PROG\_B, DONE, INIT\_B, CCLK, DONE

JTAG port for low-cost download cable

Digilent JTAG download/debugging cable connects to PC parallel port

JTAG download/debug port compatible with the Xilinx Parallel Cable IV and MultiPRO Desktop Tool

AC power adapter input for included international unregulated +5V power supply

Power-on indicator LED

On-board 3.3V , 2.5V , and 1.2V regulators





**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).

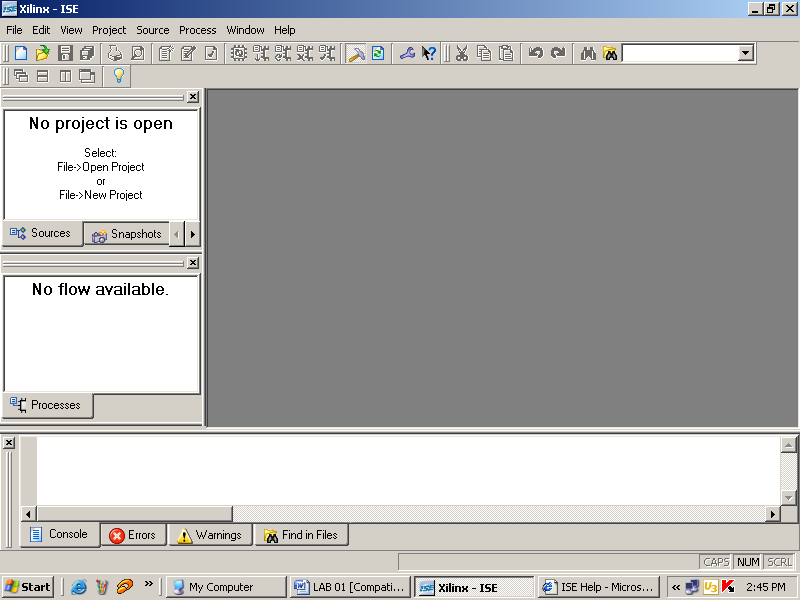
## Verification

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

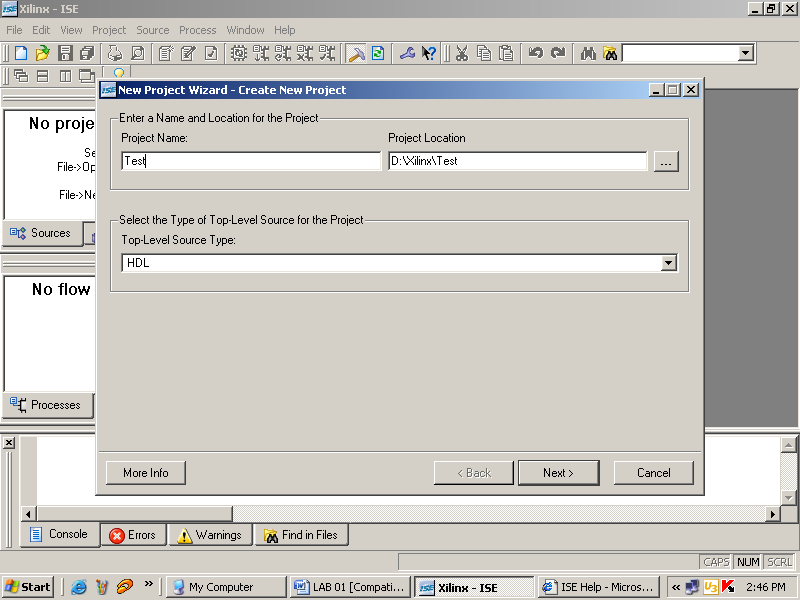
After generating a programming file, you configure your device. During configuration, you generate configuration files and download the programming files from a host computer to a Xilinx device.

## How to make new Project:

Steps for new project: 

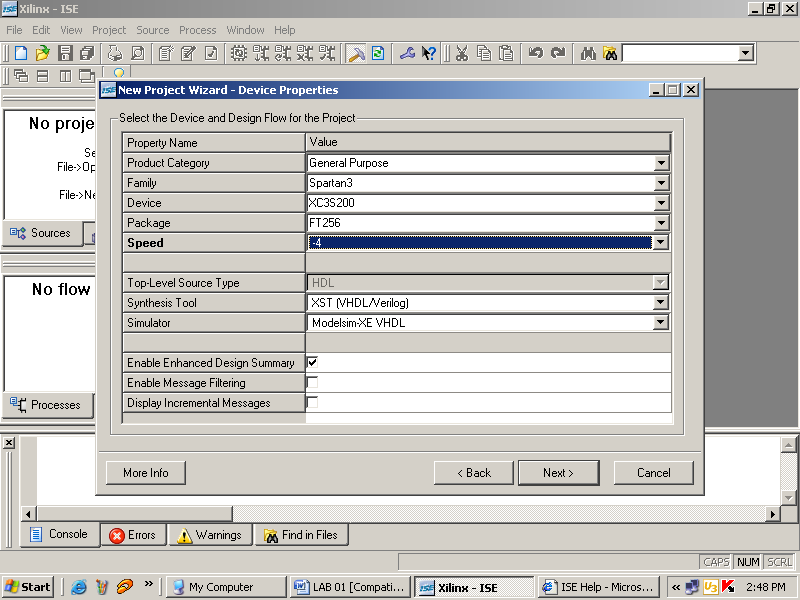
File🡪New Project

The following window will appear



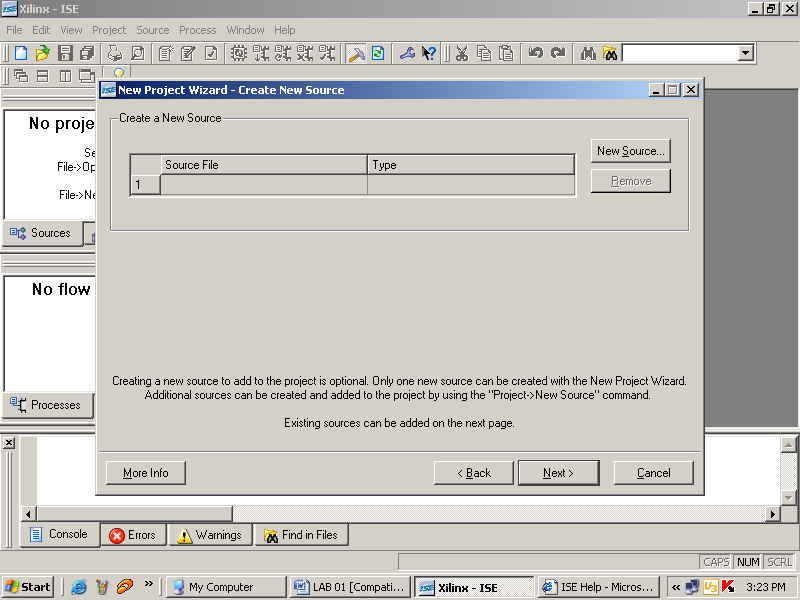
Enter the Project Name and Project Location and Press Next>

The following window will appear



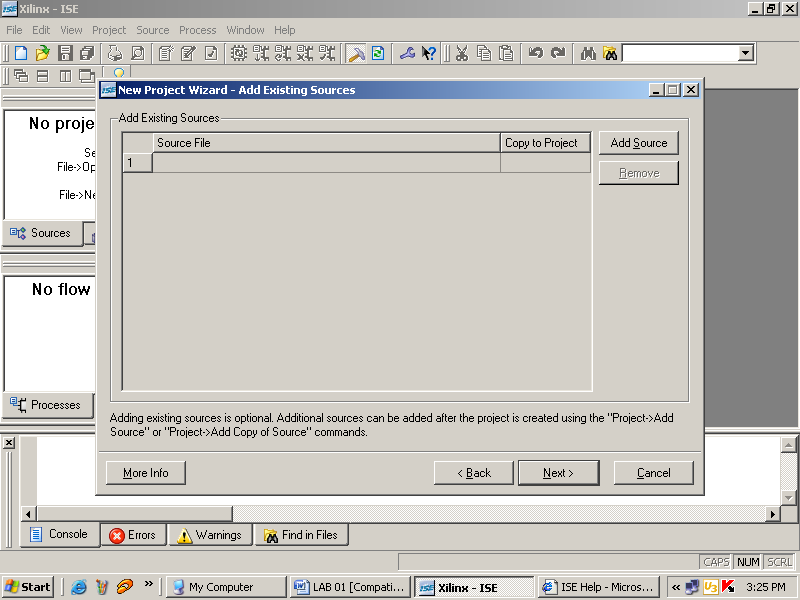
Select the options from the PULL DOWN lists as show above, and Press Next>

The following window will appear



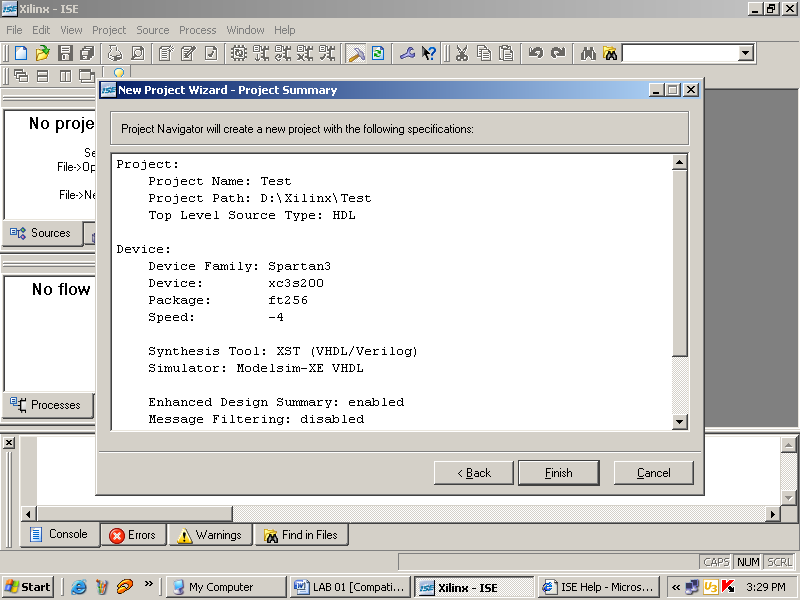
Now if you want to create a source file, then click New Source otherwise click Next> to add the existing Source file.

The following window will appear after pressing the Next>

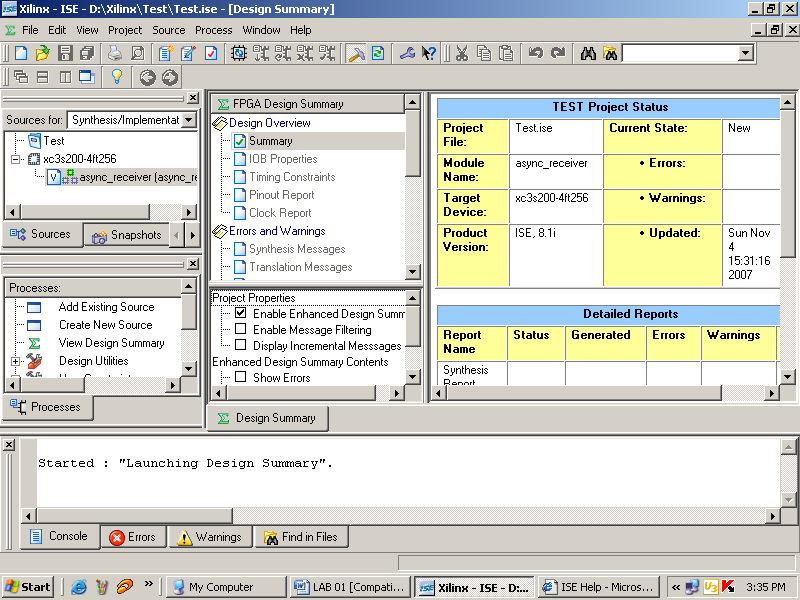


Click Add Source to add the source files and then press Next>

The next window will show the project summary as shown below.



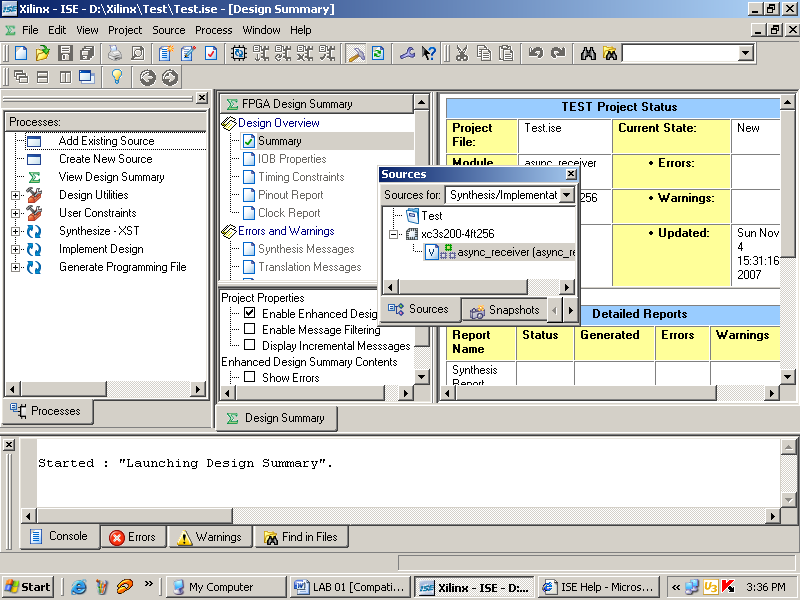
And press Finish.



The code can be synthesized by the following steps.

1. First of all the USER DEFINED CONSTRAINT file is created and added in the project
2. Code is then synthesized
3. Design is implemented
4. Programming file is then generated which can be downloaded into the FPGA

The Process windows has all the above options



**Lab Task:**

1. Develop a program to control on Board LED using On board available switch.
2. Develop a program that implements a 2x1 multiplexer using gate level modeling.
   1. Implement 2x1 mux on the board. Connect the inputs to switches and output to led.
   2. Implement 4x1 mux using modelsim.
   3. Simulate the multiplexer with a test bench.