Spartan 3E Starter Kit   *Exercise 1*.

Problem Description;

Implement a 2-to-1 multiplexer. Use West and South buttons for two inputs. Use rightmost switch as the input selection signal. Connect output to the rightmost LED. Related buttons, switch and LED are shown in the Fig-1.

West   South   LED0   SW0

Figure 1. Locations of buttons, switch and LED used in the exercise.

Solution;

1. Create a new project using `File -> New Project` and entering appropriate values in the `New Project` dialog box as shown. Click *Next*. `Device Properties` dialog box will appear.
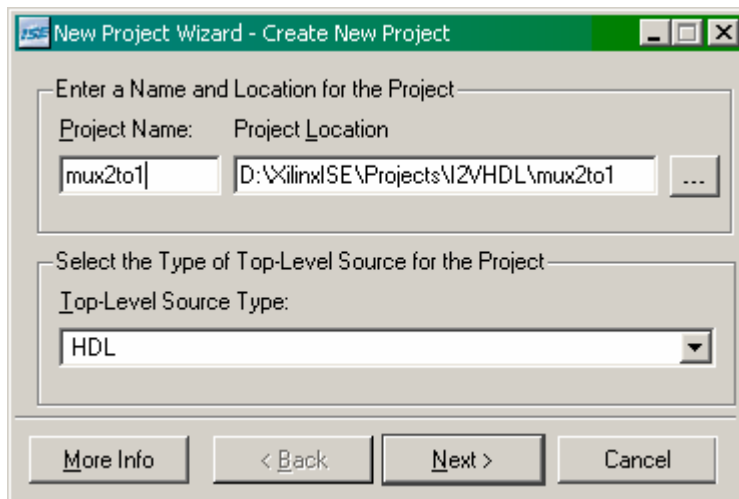

Figure 2. Creating a new project.

2. `Device Properties` dialog box is very similar to the `Project Properties` dialog box, which you can use later to change settings if you wish. For our SPARTAN 3E Starter Kits, however, just enter the following choices and you do not need to change later. After entering required values, click *Next*. `Create New Source` dialog box will appear.
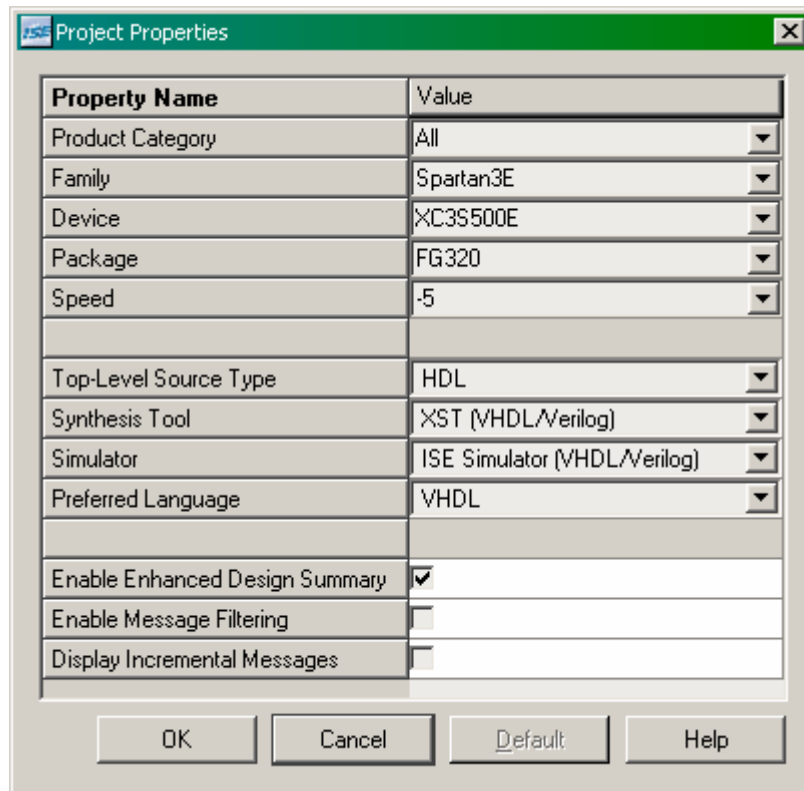
Figure 3. Enter the values seen here into the `Project Properties` dialog box.

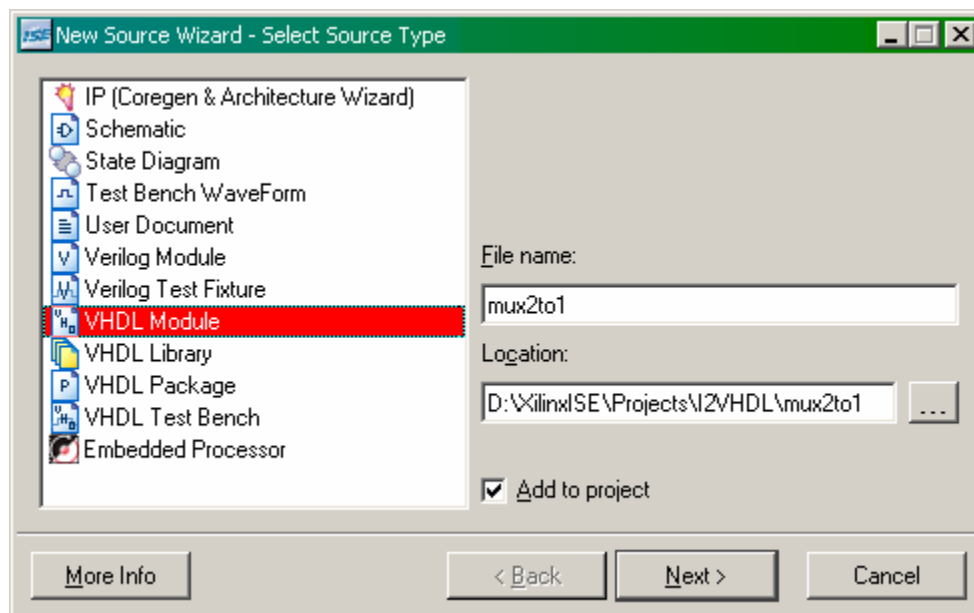**3.** Click *New Source* button and see `Select Source Type` dialog box.

Figure 4. Create a new (and first) source file as a VHDL module.

Select *VHDL Module* type and enter a name in the *File name* box, as shown.
Click *Next* afterwards. A dialog box named `Define Module` will appear.

4. For our simple example, we need two inputs and one output. So select three IO ports for the module as such as shown in Fig-5. Choose appropriate names for the IO ports, entity and architecture. You may enter the same name for entity and architecture too.
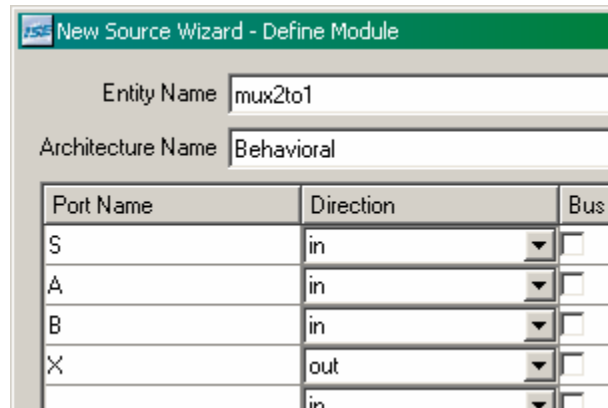

Figure 5. Ports of the designed entity.

5. Click *Next* and dismiss the `Summary` dialog box by clicking *Finish* button.

6. Click *Next* on `Create New Source` dialog box. Click *Next* on `Add Existing Sources` dialog box. Click *Finish* on `Project Summary` dialog box.

7. See `Sources` and `Processes` toolboxes on the left and double click on the `mux2to1.vhd` file name to display and edit the file.
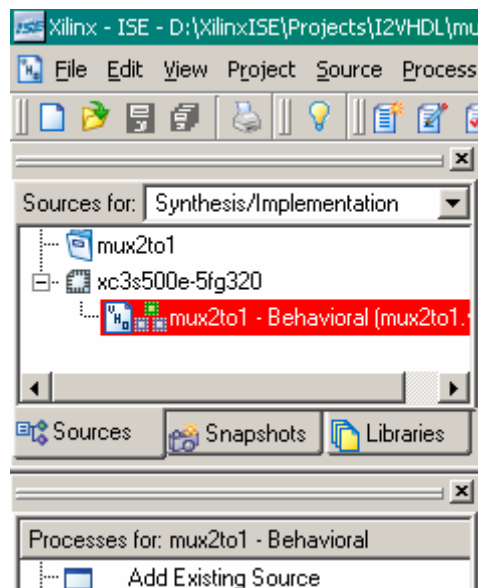

Figure 6. Sources Toolbox displayed on the left

8. The code in the `mux2to1.vhd` source is should be

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity mux2to1 is
    Port ( A : in BIT;
           B : in BIT;
           S : in BIT;
           X : out BIT);
end mux2to1;

architecture Behavioral of mux2to1 is
begin

end Behavioral;
```

excluding some informative comments. We do not currently need all `use` clauses for the library packages but it does not hurt to keep them as they are now. The `entity` block describes the IO ports of our little circuit. The `architecture` block is where we are going to put the behavioral description of the circuit. The section is currently empty.

9. Enter the following code between `begin` and `end` lines of the `architecture` block.

```
X <= (A and not S) or (B and S);
```

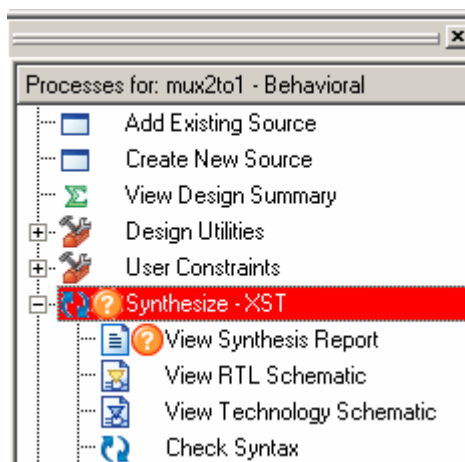**10.** Save the file and double click on the `Synthesize-XST` item in the processes toolbox.

Figure 7. Synthesize item on the Processes Toolbox.

This is equivalent to compiling a program. Synthesizer generates a circuit out of our HDL constructs. If you see an error at the synthesizing step, correct them.

11. Double click on the *View RTL Schematic* item below the *Synthesize-XST* item and see the circuit block diagram
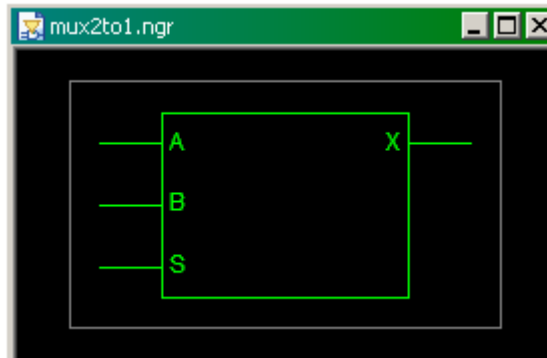

Figure 8. Circuit generated by the synthesizer.

Double click on the block to expand it and see that we have a circuit as expected as seen in Fig-9.
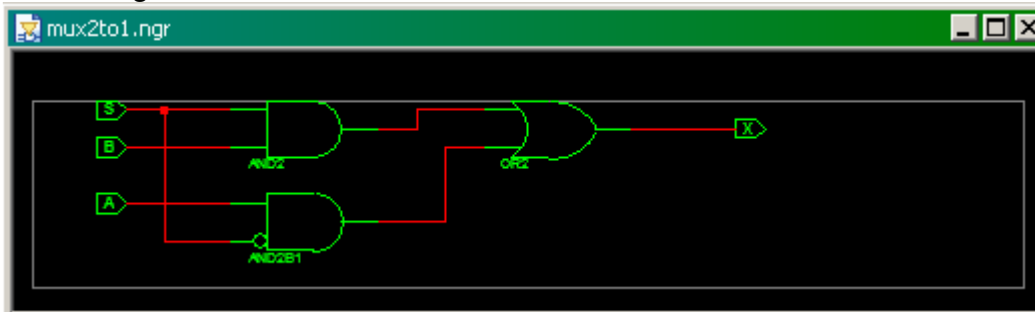

Figure 9. inside schematics of the circuit designed.

12. Open *PACE Package Pins Editor* by double clicking *Assign Package Pins* item in the *Processes* toolbox
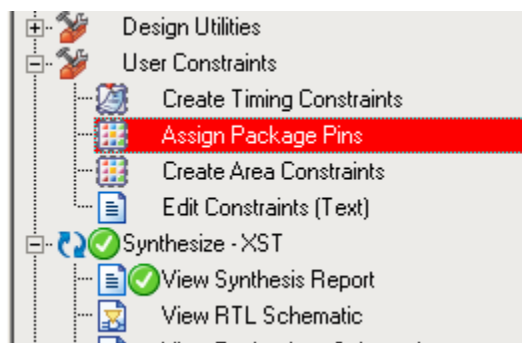

Figure 10. Click on the *Assign Package Pins*.

13. *PACE* displays the pin layout of the Spartan-3E FPGA chip on the Starter Kit in the right window. On the left, we will see the list of I/O Pins of our design by default. For different FPGA chips and packages, you may see different layout. Currently Spartan-3E Starter Kit has XC3S500E in FG320 package just as we selected at the startup of the project.
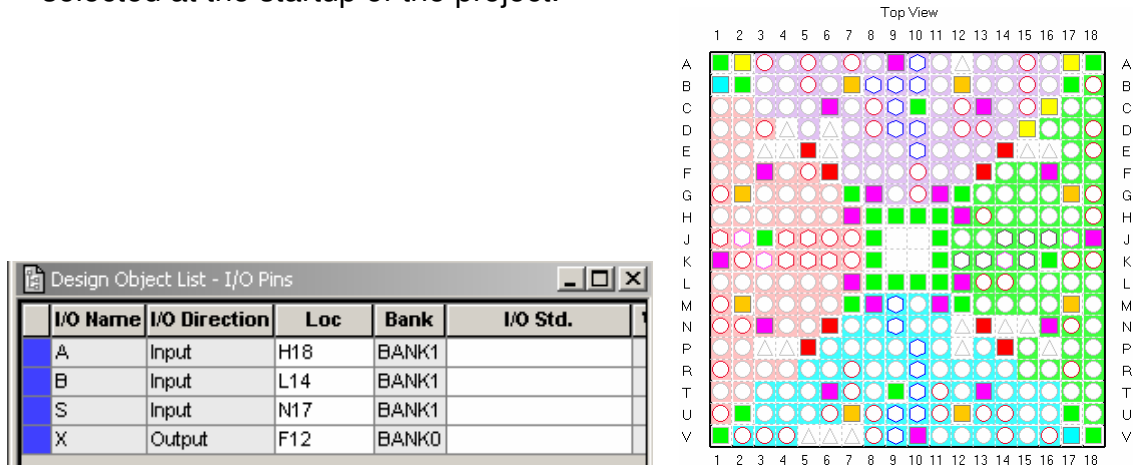

I/O Name	I/O Direction	Loc	Bank	I/O Std.
A	Input	H18	BANK1	
B	Input	L14	BANK1	
S	Input	N17	BANK1	
X	Output	F12	BANK0	

Figure 11. Important windows of the PACE.

14. Now open the Starter Kit Board User Guide and find LEDs section. See the rightmost LED is connected to F12 pin of the FPGA device.
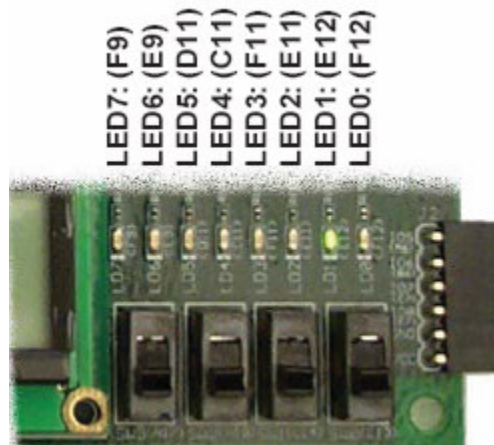

Figure 12. Portion of the Kit showing LEDs.

15. Repeat the same tracking process for the `West` and `South` buttons for A and B inputs and see that they are connected to K17 and D18 pins respectively. Also, verify that SW0 is connected to N17 pin.

16. In *PACE*, drag the LED I/O pin over the chip layout and drop it on the F12 pin. Repeat drag-drop action for other pins too. (You can enter the pin in Loc field too). Set all I/O Std. entries to LVTTL and input terminations to PULLDOWN. Leave other fields as they are. After pin assignments, I/O Pin window should look like shown in Fig-13.

Figure 13. I/O pins after assignments.

17. Save and close the PACE. If you see a formatting window like shown below, select *XST Default* and click *OK*.
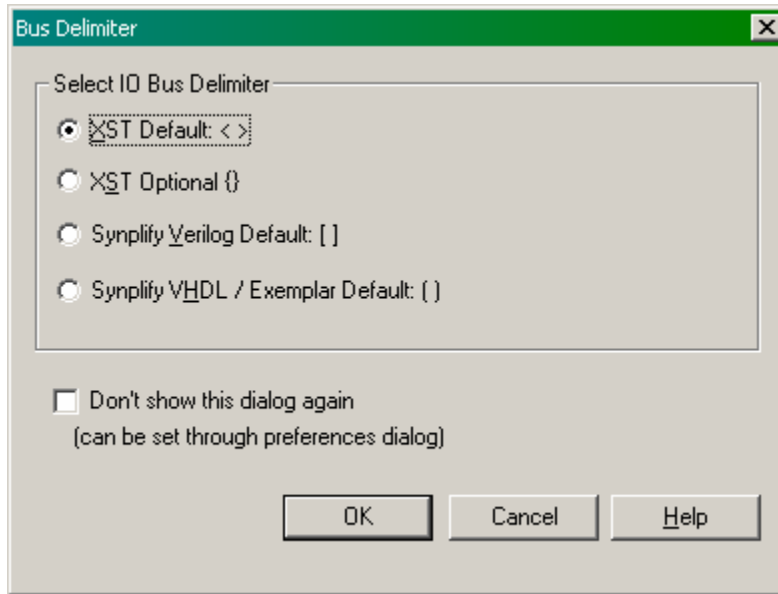

Figure 14. Bus delimiter dialog box.

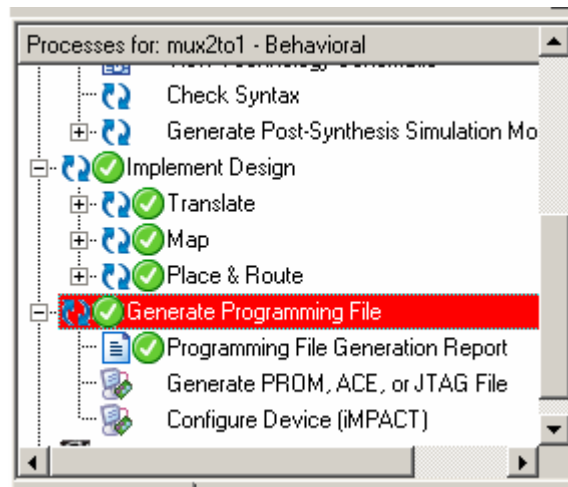**18.** Double click on *Generate Programming File* item in the *Processes* toolbox.

Figure 15. Click on the *Generate Programming File* selection.

19. Connect your "Starter Kit" to an available USB port of your computer and turn the kit's power on.

20. Double click on the *Configure Device* (iMPACT) item under *Generate Programming File* in the *Processes* Toolbox.
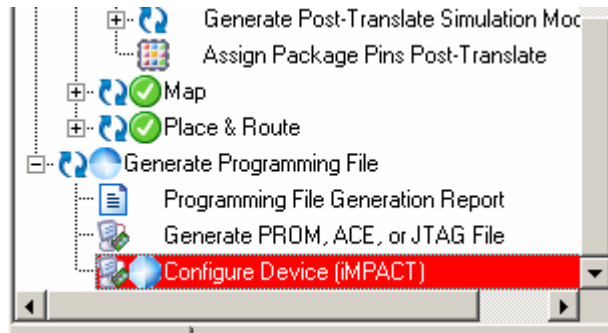

Figure 16. Double click on the *Configure Device* item to program FPGA.

Select *Configure devices using Boundary-Scan (JTAG)* selection when the IMPACT dialog box is displayed and click on the *Finish* button
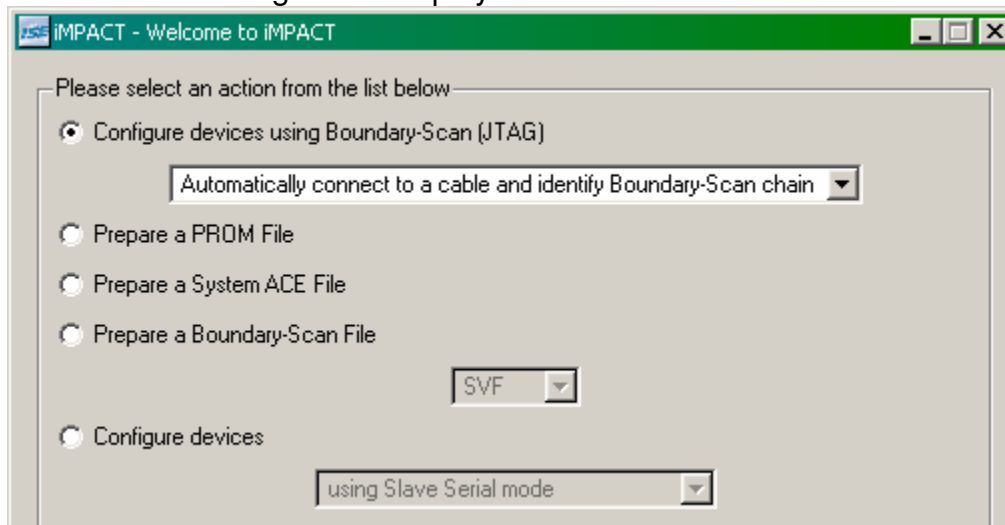

Figure 17. What to do in iMPACT.

21. Select *mux2to1.bit* file when JTAG devices are displayed and programming file for xc3s500e is asked.
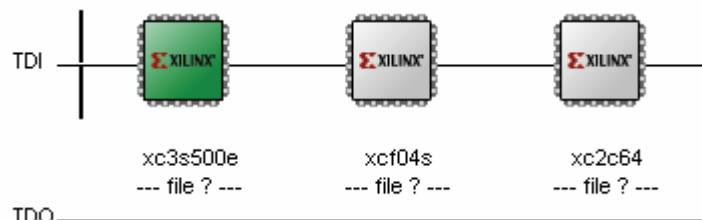

Figure 18. Found programmable devices on the Starter Kit.

You can right click on the xc3s500e device and select *Program* when design changes are occurred and you need to reprogram the device.

Figure 19. Loading design binary into FPGA.

22. Click *OK* on `Programming Properties` dialog box. You need not change the default properties. You should see a confirmation message when the programming file transfer is successfully completed

Figure 20. Message received after sending the program via USB link.

23. Test your design by pressing South and West buttons, changing position of the switch and see if the LED goes on/off as expected. This concludes the first FPGA exercise with ISE.