



Embedded System ISE Sub-Module Lab

Objectives

This lab demonstrates how to add a PowerPC embedded system as a sub-module into the ISE environment. The lab will show

- How to instantiate an existing XPS project as a sub-module into ISE
- How to modify the top level example instantiation file

Requirements

- ISE 9.2.04
- EDK 9.2.02
- ML403 Board
- USB Platform Cable
- RS232 Cable
- Completion of the Adding Custom IP to an Embedded System Lab

Reference

- Platform Studio Help
- ISE Help
- Embedded System Tools Reference Manual

Overview

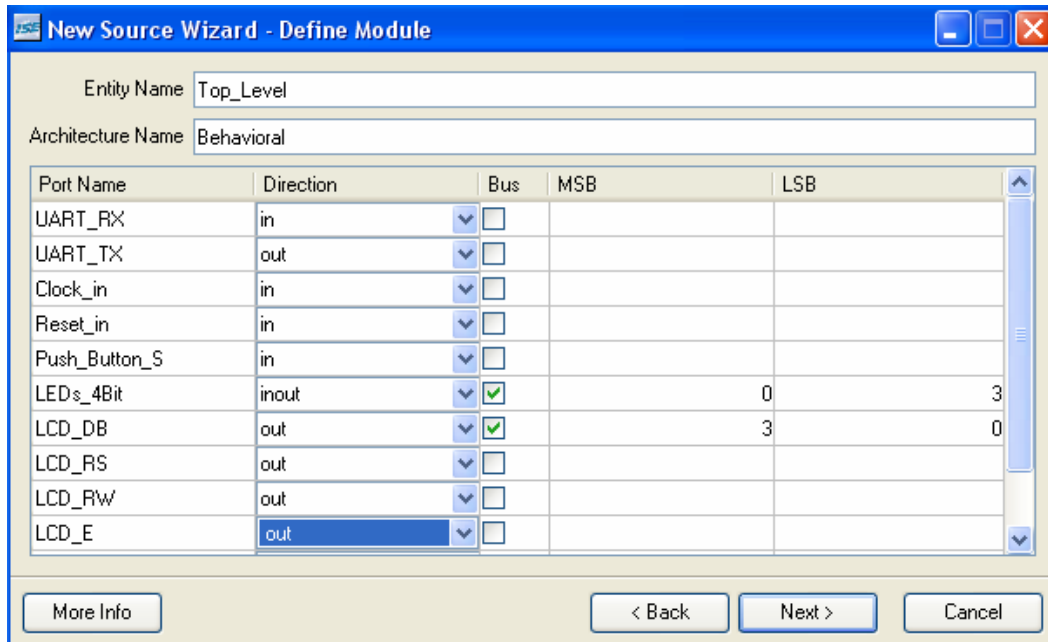
The embedded system is currently a top-level module inside ISE. In order to add user logic, it is often necessary to have the embedded system as one of the sub-module in the FPGA design. The lab is divided into two main steps: Modifying the embedded system in ISE to be a sub-module, and testing the system on the FPGA. The test application will reside in Block memory inside the FPGA.

Lab Steps

I. Embedded System sub-module in ISE

The embedded system is currently the top level module in ISE. However the majority of FPGA designs include some other non-embedded logic. We will make the transition from a top-module flow to a sub-module flow.

1. **Start** ISE Project Navigator and **open** the design from the *Adding Custom IP to an Embedded System Lab*.
2. In the Project Navigator *Processes* window **double-click** on *View HDL Instantiation Template*.
3. The file created provides the component definition for the embedded system as well as the instantiation template. We will create a top level file and add the embedded system.
4. Go to **Project > New Source...** **Select VHDL Module** and name it *Top_Level*. Click on **Next**.
5. We will need to add all the ports for the UART, clock, reset, LEDs, push button, and LCD. **Add** the ports needed for the board as shown below. Click **Next** then **Finish**.



- To match the embedded system, **change** the vector range for LEDs_4Bit from (0 downto 3) to (0 to 3)
- Copy** the *ppc_system* component declaration from the *ppc_sytem.vhi* file to the *Top_Level.vhd* file between the *architecture* and the *begin* line.
- Copy** the *ppc_system* component instantiation from the *ppc_sytem.vhi* file to the *Top_Level.vhd* file between the *begin* and the *end architecture* line.
- Add** a new internal signal which will map the *std_logic_vector* type for the push button before the *begin* line

```
signal Push_Button_S_GPIO_in_pin : std_logic_vector(0 to 0);
```

- Map** the top level ports to the embedded system ports. For the push button, **map** it to the internal signal first:

```
Push_Button_S_GPIO_in_pin(0) <= Push_Button_S;
```

```
Inst_ppc_system: ppc_system PORT MAP(
  fpga_0_RS232_Uart_RX_pin => UART_RX,
  fpga_0_RS232_Uart_TX_pin => UART_TX,
  fpga_0_LEDs_4Bit_GPIO_IO_pin => LEDs_4Bit,
  sys_clk_pin => Clock_in,
  sys_rst_pin => Reset_in,
  Push_Button_S_GPIO_in_pin => Push_Button_S_GPIO_in_pin,
  plb_lcd_0_LCD_DB_pin => LCD_DB,
  plb_lcd_0_LCD_RS_pin => LCD_RS,
  plb_lcd_0_LCD_RW_pin => LCD_RW,
  plb_lcd_0_LCD_E_pin => LCD_E
);
```

- Save** the *Top_Level.vhd* file.
- Since the name of the top level ports has changed, the constraint file also needs to be modified.
- Expand** the project files in the sources window. **Select** *ppc_system.ucf* and **select** *Edit Constraints (Text)* in the *User Constraints* selection.



14. **Name** the ports based on the new port names in the *Top_Level.vhd* file.

- sys_clk_pin becomes Clock_in
- sys_rst_pin becomes Reset_in
- fpga_0_RS232_Uart_RX_pin becomes UART_RX
- fpga_0_RS232_Uart_TX_pin becomes UART_TX
- Push_Button_S_GPIO_in_pin<0> becomes Push_Button_S
- fpga_0_LEDs_4Bit_GPIO_IO_pin becomes LEDs_4Bit
- plb_lcd_0_LCD_DB_pin becomes LCD_DB
- plb_lcd_0_LCD_RS_pin becomes LCD_RS
- plb_lcd_0_LCD_RW_pin becomes LCD_RW
- plb_lcd_0_LCD_E_pin becomes LCD_E

15. Some of the constraints need to be modified to account for the extra level of hierarchy. **Modify** the PowerPC reset signal constraints:

```
NET "Inst_ppc_system/ppc_reset_bus_Chip_Reset_Req" TPTHURU = "RST_GRP";  
NET "Inst_ppc_system/ppc_reset_bus_Core_Reset_Req" TPTHURU = "RST_GRP";  
NET "Inst_ppc_system/ppc_reset_bus_System_Reset_Req" TPTHURU = "RST_GRP";
```

16. **Save** and **close** the constraints file.

17. **Select** the top level module in the *Sources* window.

18. Cleanup the project files. Go to **Project > Cleanup Project Files**.

19. **Double-click** on **Update Bitstream with Processor Data** in the *Processes* window.

II. Test the Generated System with the Sample Application

1. Open XPS SDK.
2. Connect a serial cable to the RS232 port and connect the Platform USB cable.
3. Power-up the board.
4. Open Hyperterminal and connect at 9600, no parity.
5. **Select** the *Test_Lab* application to be loaded in BRAMs. Go to **Configuration > Program Hardware Settings...** **Select** *Test_Lab.elf* and **save**.
6. Download to the FPGA from SDK: **Device Configuration > Program Hardware**.
7. Verify that the print statement appears on the Hyperterminal window.
8. The LCD should show "Custom IP".
9. Using the Push Button (GPIO_SW_S) should turn on the LEDs while pressed.
10. Close SDK.