

Getting Started With the NEXYS2 Spartan 3E Kit and Xilinx ISE Webpack Tutorial



OVER VIEW

The Nexys2 circuit board is a complete, ready-to-use circuit development platform based on a Xilinx Spartan 3E FPGA. The Nexys2 board input power input bus can be driven from a USB cable, from a 5VDC-15VDC, center positive, 2.1mm wall-plug supply, or from a battery pack. A shorting block loaded on the “power select” jumper selects the power source. The USB circuitry is always powered from the USB cable – if no USB cable is attached, the USB circuitry is left unpowered.

Features:

- 500K-gate Xilinx Spartan 3E FPGA
- USB2-based FPGA configuration and high-speed data transfers (using the free Adept Suite Software)
- USB-powered (batteries and/or wall-plug can also be used)
- 16MB of Micron PSDRAM & 16MB of Intel StrataFlash ROM
- Xilinx Platform Flash for nonvolatile FPGA configurations
- Efficient switch-mode power supplies (good for battery powered applications)
- 50MHz oscillator plus socket for second oscillator
- 60 FPGA I/O's routed to expansion connectors (one highspeed Hirose FX2 connector and four 6-pin headers)
- 8 LEDs, 4-digit 7-seg display, 4 buttons, 8 slide switches
- Ships in a plastic carry case with USB cable

FPGA and Platform Flash Configuration

The FPGA on the Nexys2 board must be configured (or programmed) by the user before it can perform any functions. During configuration, a “bit” file is transferred into memory cells within the FPGA to define the logical functions and circuit interconnects. The free ISE/ Web Pack CAD software from Xilinx can be used to create bit files from VHDL, Verilog, or schematic-based source files.

The FPGA can be programmed in two ways: directly from a PC using the on-board USB port, and from an on-board Platform Flash ROM (the Flash ROM is also user-programmable via the USB port). A jumper on the Nexys2 board determines which source (PC or ROM) the FPGA will use to load its configuration. The FPGA will automatically load a configuration from the Platform Flash ROM at power-on if the configuration Mode jumper is set to “Master serial”. If

the Mode jumper is set to “JTAG”, the FPGA will await programming from the PC (via the USB cable).

Digilent’s freely available PC-based Adept software can be used to configure the FPGA and Platform Flash with any suitable file stored on the computer. Adept uses the USB cable to transfer a selected bit file from the PC to the FPGA or Platform Flash ROM. After the FPGA is configured, it will remain so until it is reset by a power-cycle event or by the FPGA reset button (BTNR) being pressed. The Platform Flash ROM will retain a bit file until it is reprogrammed, regardless of power-cycle events.

To install Drivers of the software:

Download and install Digilent Adept suite (ADEPT 1.10, 32 bit windows) from the following link <http://www.digilentinc.com/Products/Detail.cfm?NavTop=2&NavSub=451&Prod=NEXYS2>

User I/O

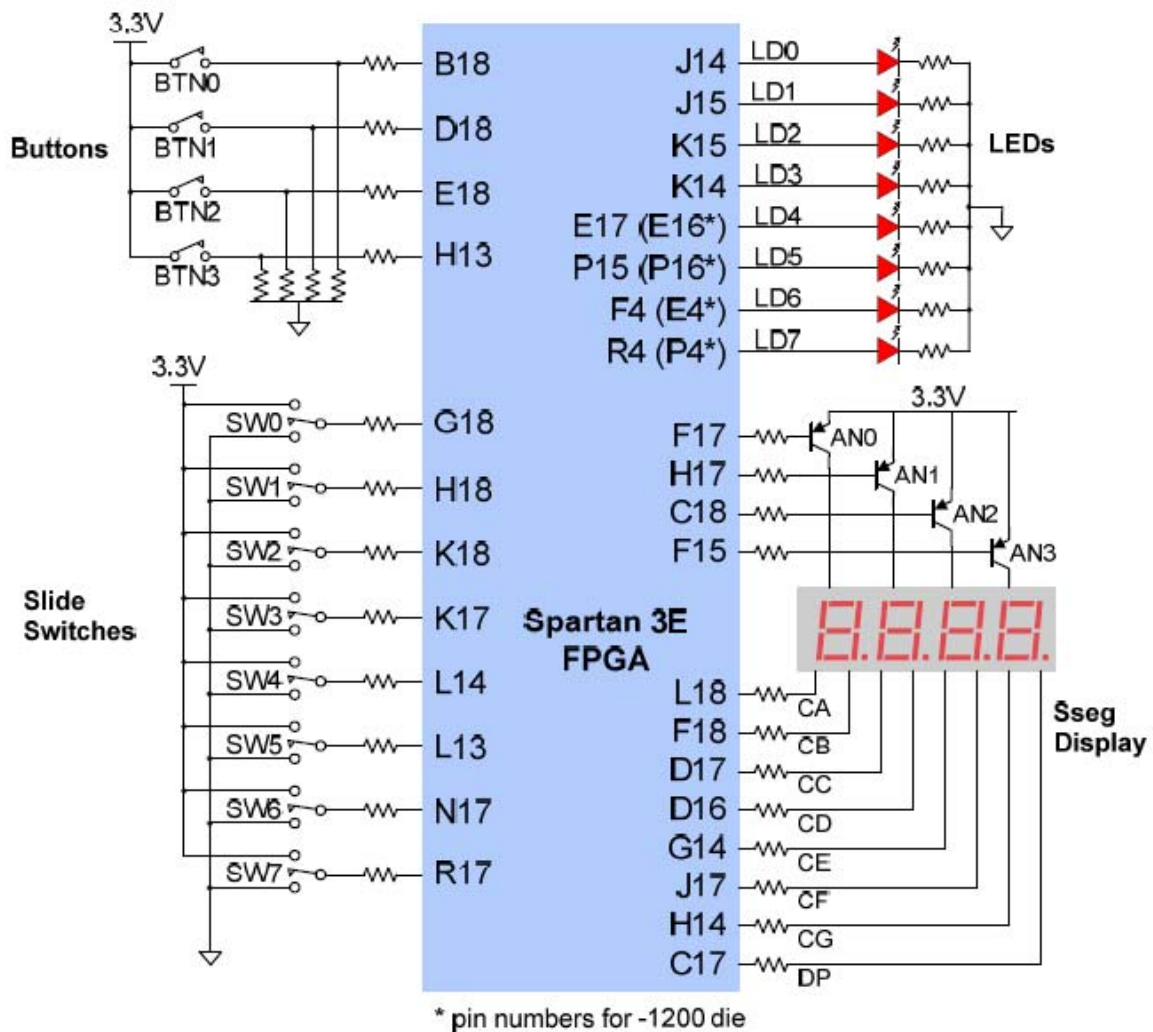
The Nexys2 board includes several input devices, output devices, and data ports, allowing many designs to be implemented without the need for any other components.



Four pushbuttons and eight slide switches are provided for circuit inputs. Pushbutton inputs are normally low, and they are driven high only when the pushbutton is pressed. Slide switches generate constant high or low inputs depending on their position. Pushbutton and slide switch inputs use a series resistor for protection against short circuits (a short circuit would occur if an FPGA pin assigned to a pushbutton or slide switch was inadvertently defined as an output).

The Nexys2 board contains a four-digit common anode seven-segment LED display. Each of the four digits is composed of seven segments arranged in a “figure 8” pattern, with an LED embedded in each segment.

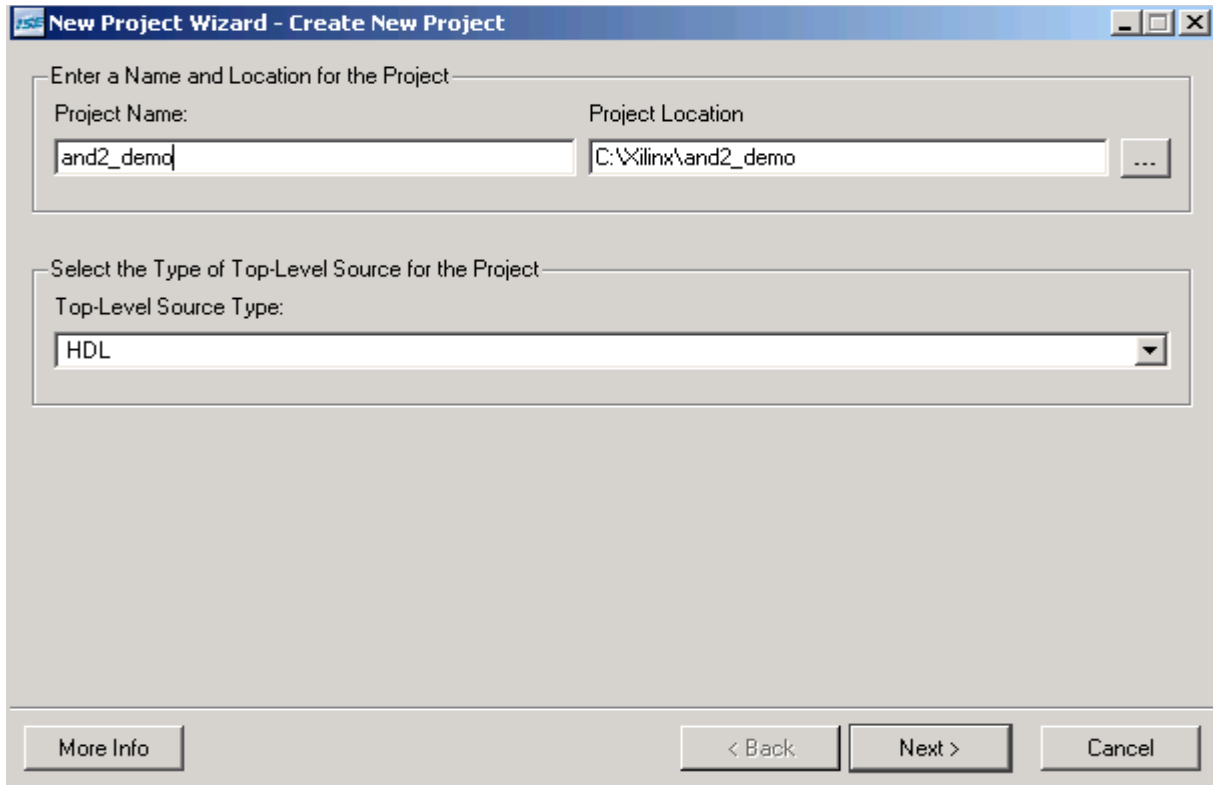
PIN ASSIGNMENT:



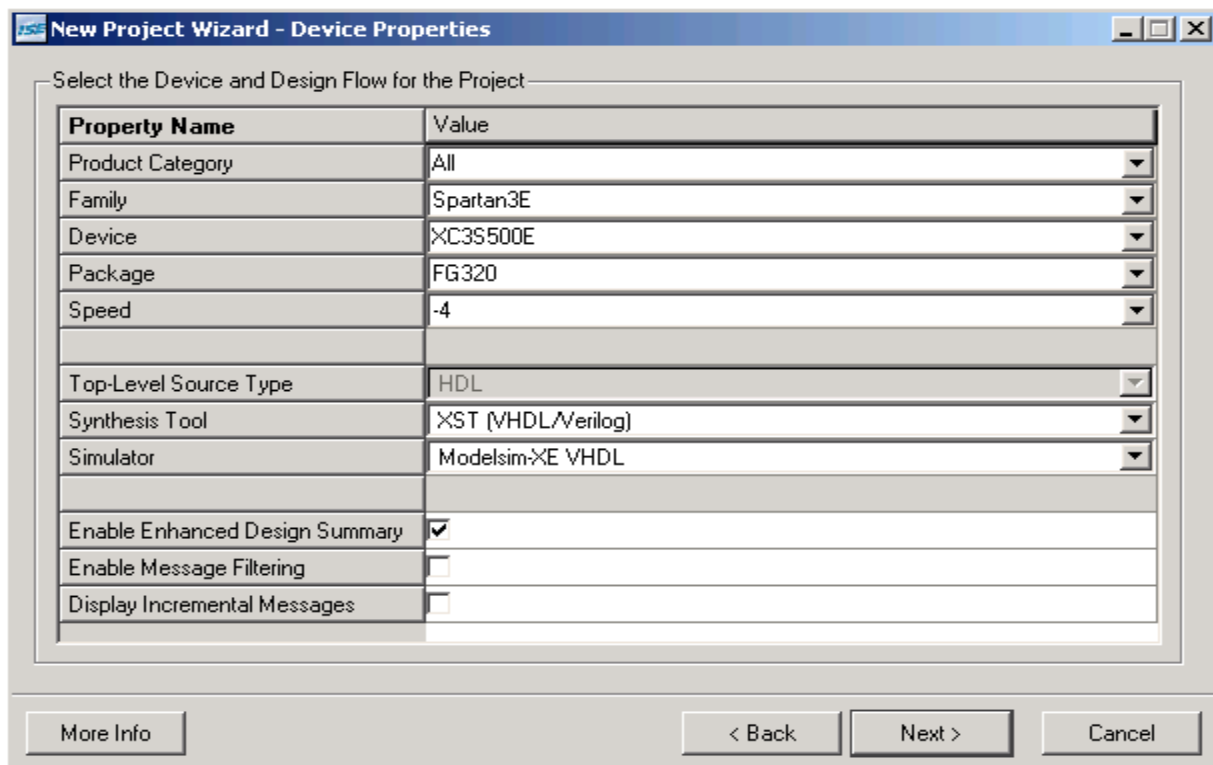
The Project

Let us write a code for AND gate and implement it on nexys2 board. We would like to place the AND value of two inputs on the rightmost 7-segment display, and leave the rest blank.

To start the design, first start Xilinx ISE, and create a new Project with File New Project. When specifying the project name and directory, pick a location without any spaces in the path (i.e. not the desktop) You will often find the Xilinx tools behaving very strangely with difficult to diagnose errors if you have spaces in the project path.



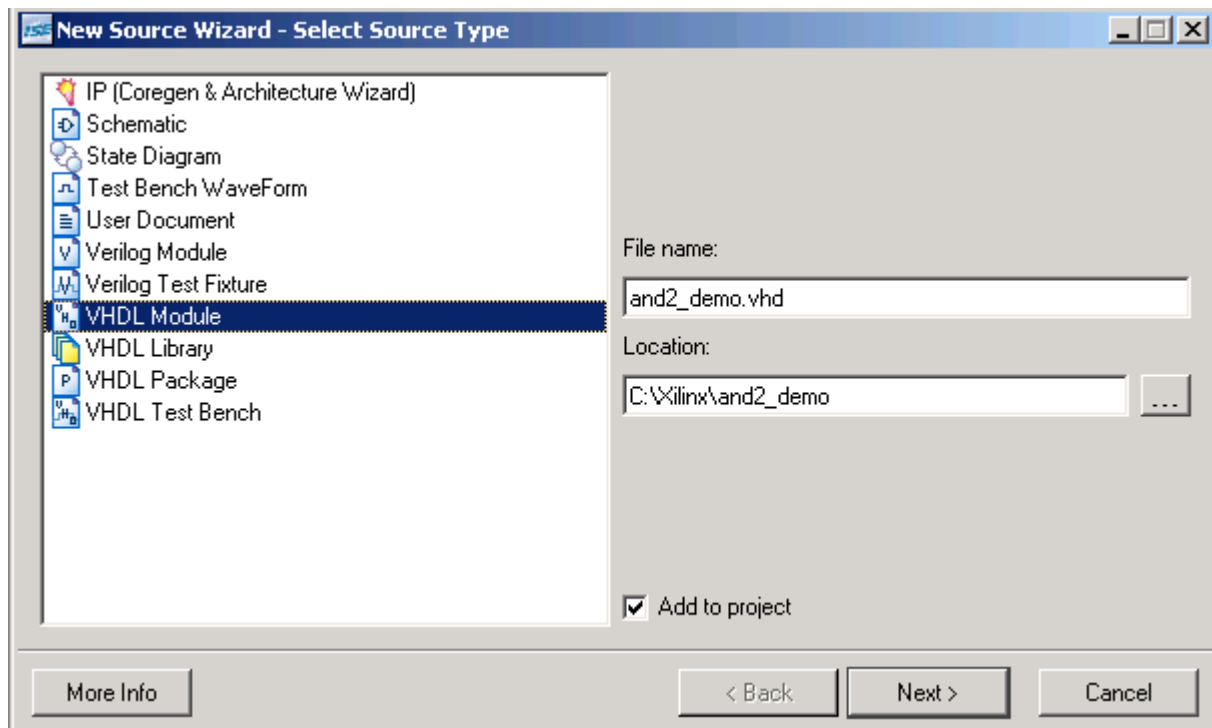
The next step will be to specify the FPGA you are targeting. This is obviously the one that is on the NEXYS2 board.



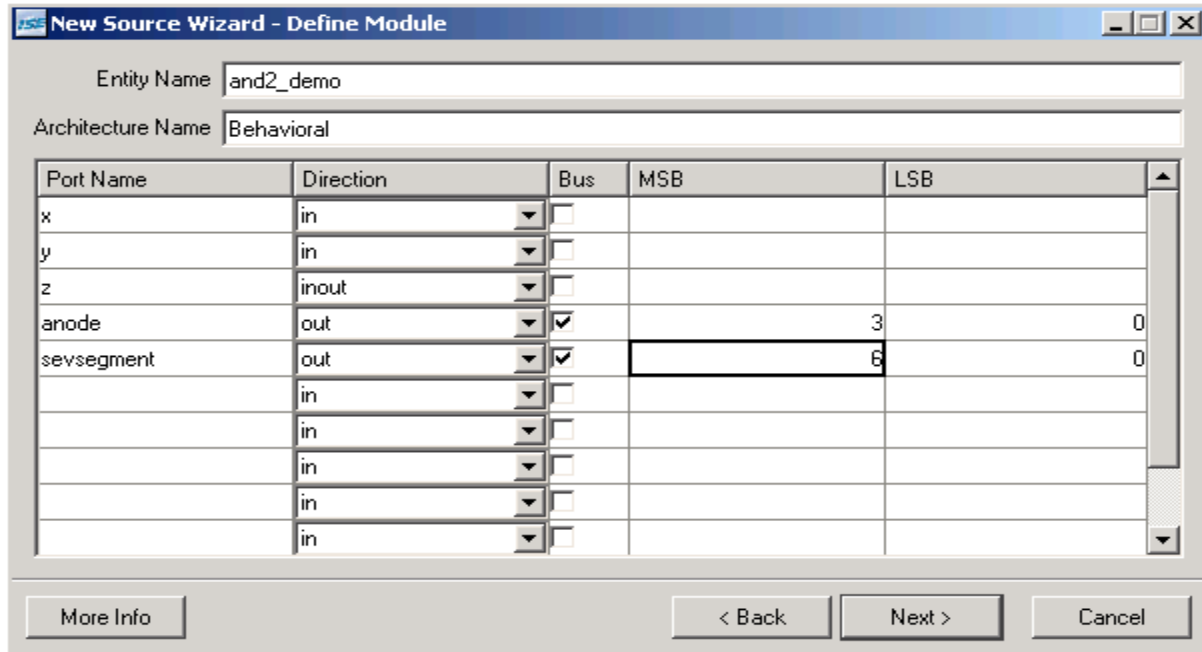
Skip over the next two pages, and click Finish to complete the project creation and begin the design process.

Entering VHDL

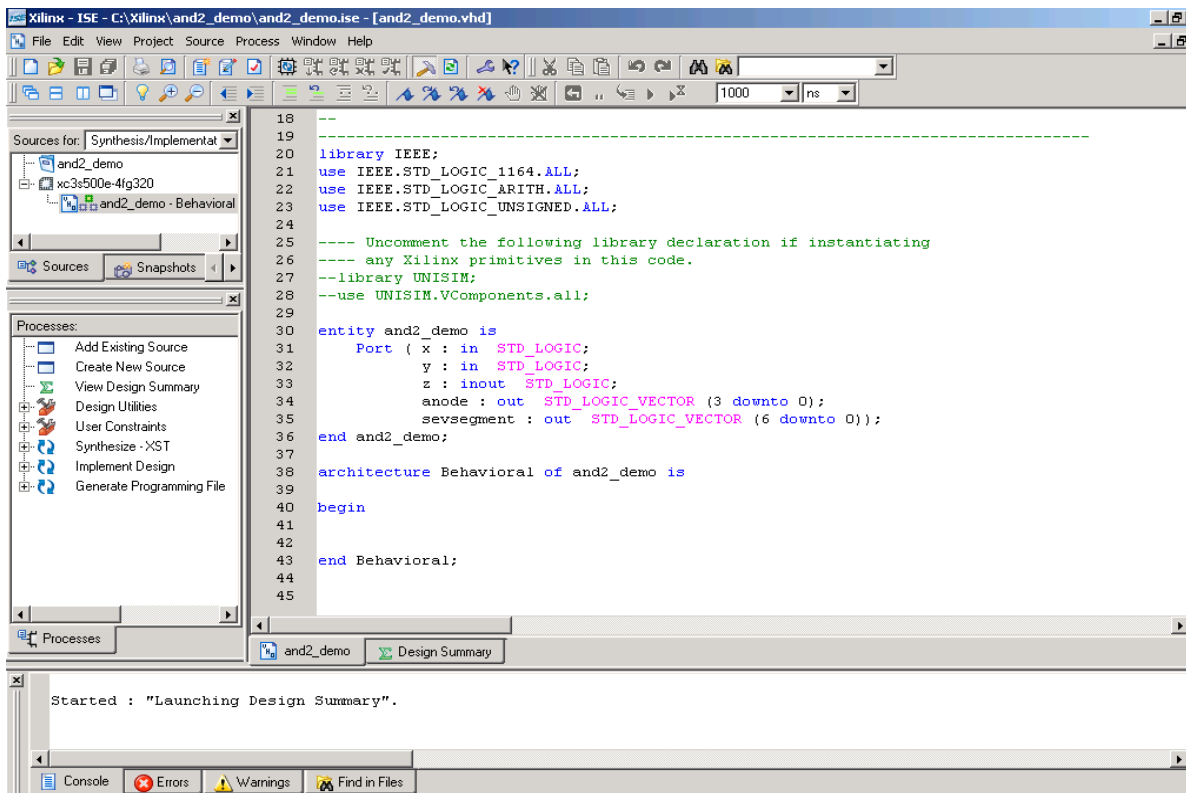
To describe the design, we will add one simple VHDL file to our project. Though this can be done with any text editor, it is convenient to allow the ISE tools to create the framework for us since we are starting the file from scratch. Adding an existing file is similarly easy, and can be done with “Project → Add Copy of Source”. Select “Project → New Source”, and fill out the dialog telling the tools to create a new VHDL module with the name “and2_demo.vhd”



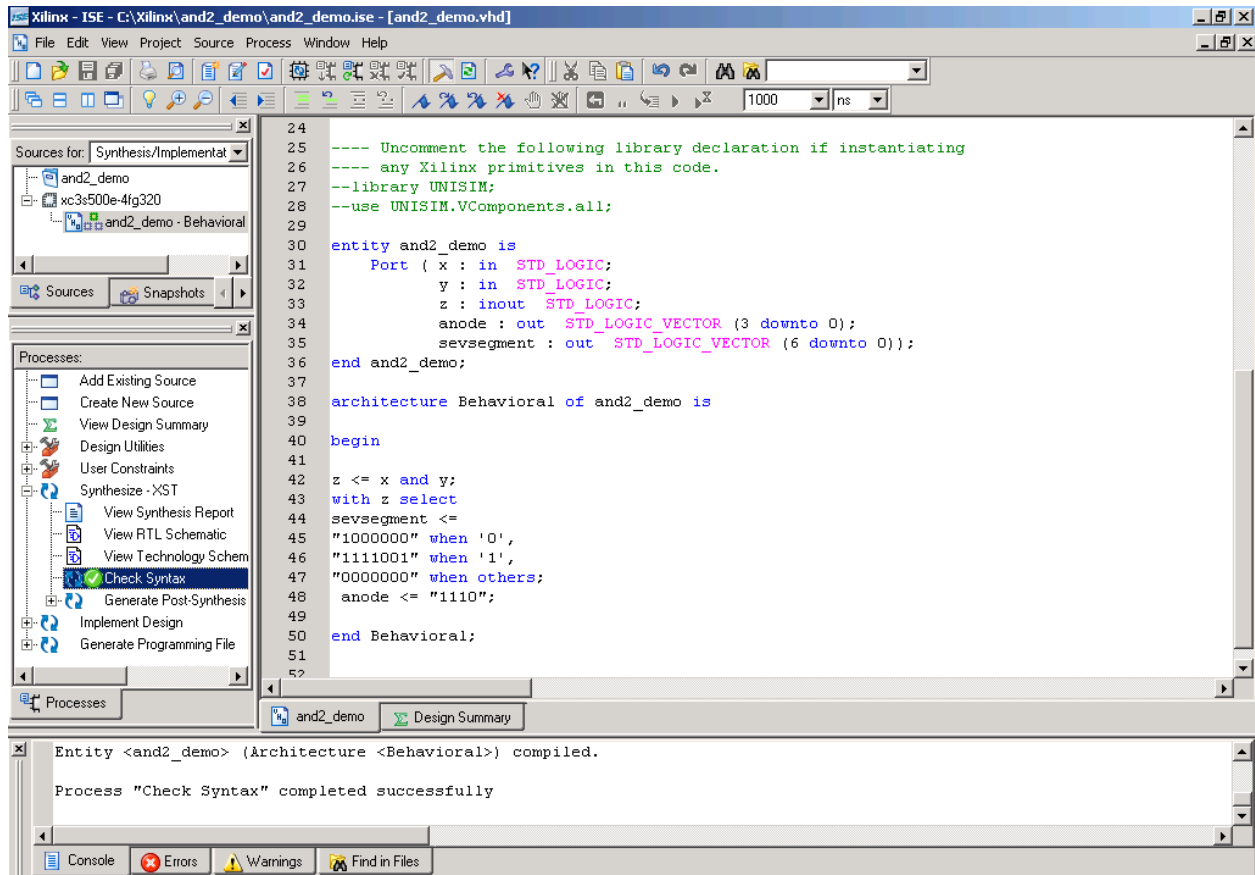
On the next page, you will be prompted to enter the ports of your new module. This is strictly optional, and saves you the typing of creating the VHDL entity manually. Whether you fill this out or not, the resultant text file will be totally under your control for later editing.



We have entered the ports which will be placed into our entity in the VHDL file : the switches as our inputs, and the seven segments to drive on the display as our outputs. Furthermore, we have included the port “anodes” which will be used to enable / disable each of the 4 displays individually. The resultant VHDL file looks essentially like this :



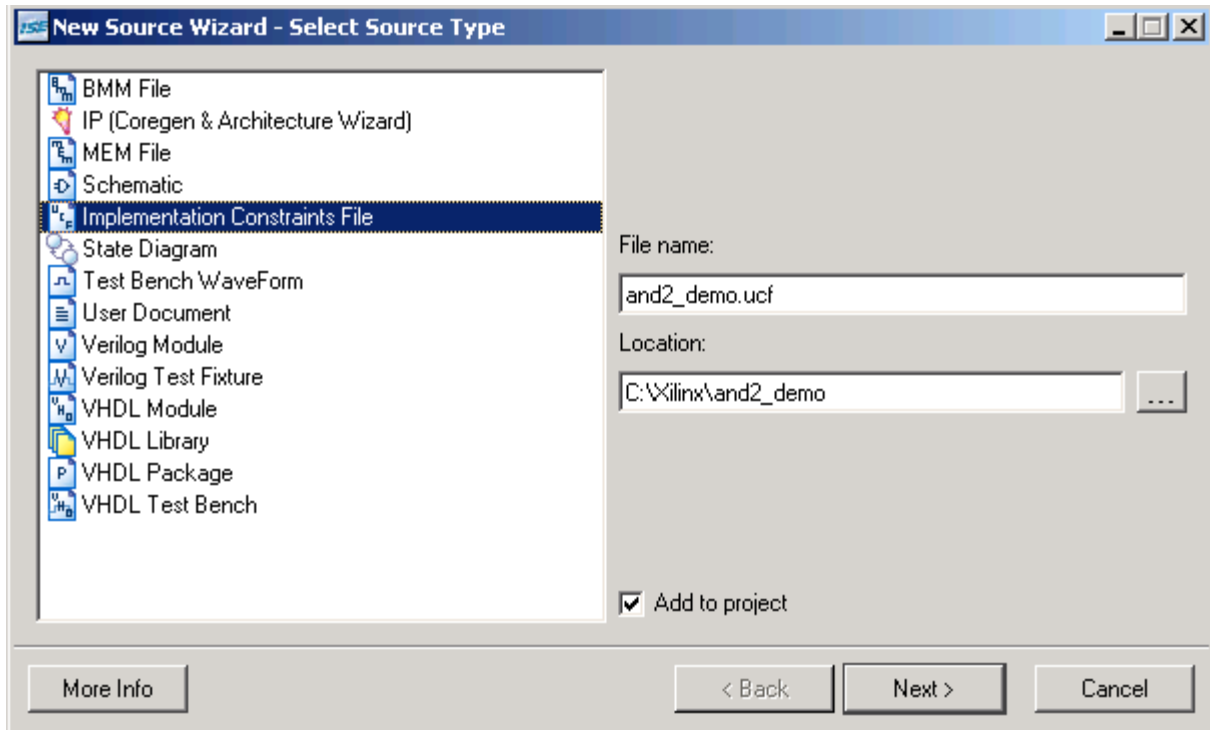
Xilinx has conveniently left a place for us to insert the architecture of our design, which will be two very simple statements:



After entering the VHDL, save the file, and Check the Syntax for errors by clicking on “Check Syntax” under “Synthesize” in the Processes Pane.

Assigning Pins

The next task is to assign our input and output ports to the correct locations. If we neglected this step, the pins would be assigned by the Xilinx software to random locations, which would not be good. To do this, we must first create one more new file, the “Universal Constraints File” (.ucf).



Now that we have a constraints file for the project, we can easily assign pins graphically using "Assign Package Pins" under the "User Constraints" section in the Process Pane. This can also be entered by double clicking on the .ucf file that you just created from within Project Navigator. For this exercise though, we are going to edit the .ucf file in text, to get a good feel for how simple it is.

Single Click on your ucf file from within Project Navigator, and then Select "Edit Constraints (Text)" from the Process Pane as below. Then you can edit the constraints as shown below.

```
NET "anode<0>" LOC = "F17" ;  
NET "anode<1>" LOC = "H17" ;  
NET "anode<2>" LOC = "C18" ;  
NET "anode<3>" LOC = "F15" ;  
NET "sevensseg<0>" LOC = "L18" ;  
NET "sevensseg<1>" LOC = "F18" ;  
NET "sevensseg<2>" LOC = "D17" ;  
NET "sevensseg<3>" LOC = "D16" ;  
NET "sevensseg<4>" LOC = "G14" ;
```

NET "sevensseg<5>" LOC = "J17" ;

NET "sevensseg<6>" LOC = "H14" ;

NET "x" LOC = "G18" ;

NET "y" LOC = "H18" ;

NET "z" LOC = "J14" ;

Save the file, now we are ready to try the design on the FPGA.

Synthesis, Place+Route, Programming File

The next step in the process is to have the Xilinx ISE create a programming file for you. The whole process can be completed by clicking once on your top-level design in the Sources Pane, followed by a double-click on "Generate Programming File". This will run the whole process, and should complete as in the screenshot below.

The screenshot shows the Xilinx ISE Design Summary window for a project named 'AND2_DEMO'. The window is divided into several panes:

- Sources for:** Synthesis/Implementat... (and2_demo, xc3s500e-4fg320, and2_demo - Behavioral, and2_demo.ucf)
- Processes:** Add Existing Source, Create New Source, View Design Summary, Design Utilities, User Constraints, Synthesize - XST, View Synthesis Report, View RTL Schematic, View Technology Schem, Check Syntax, Generate Post-Synthesis, Implement Design, Generate Programming File.
- FPGA Design Summary:** Design Overview (Summary, IOB Properties, Timing Constraints, Pinout Report, Clock Report), Errors and Warnings (Synthesis Messages, Translation Messages, Map Messages, Place and Route Messages, Timing Messages, Bitgen Messages, All Current Messages), Detailed Reports (Synthesis Report), Project Properties (Enable Enhanced Design Summary, Enable Message Filtering, Display Incremental Messages), Enhanced Design Summary Contents (Show Errors, Show Warnings, Show Failing Constraints, Show Clock Report).
- AND2_DEMO Project Status:**

Project File:	and2_demo.isc	Current State:	Programming File Generated
Module Name:	and2_demo	Errors:	No Errors
Target Device:	xc3s500e-4fg320	Warnings:	No Warnings
Product Version:	ISE, 8.1i	Updated:	Mon Aug 10 11:12:34 2009
- Device Utilization Summary:**

Logic Utilization	Used	Available	Utilization	Note(s)
Number of 4 input LUTs	1	9,312	1%	
Number of occupied Slices	1	4,656	1%	
Number of Slices containing only related logic	1	1	100%	
Number of Slices containing unrelated logic	0	1	0%	
Total Number of 4 input LUTs	1	9,312	1%	
Number of bonded IOBs	14	232	6%	
Total equivalent gate count for design	6			
Additional JTAG gate count for IOBs	672			
- Performance Summary:**

Final Timing Score:	0	Pinout Data:	Pinout Report
Routing Results:	All Signals Completely Routed	Clock Data:	Clock Report
Timing Constraints:	All Constraints		
- Detailed Reports:**

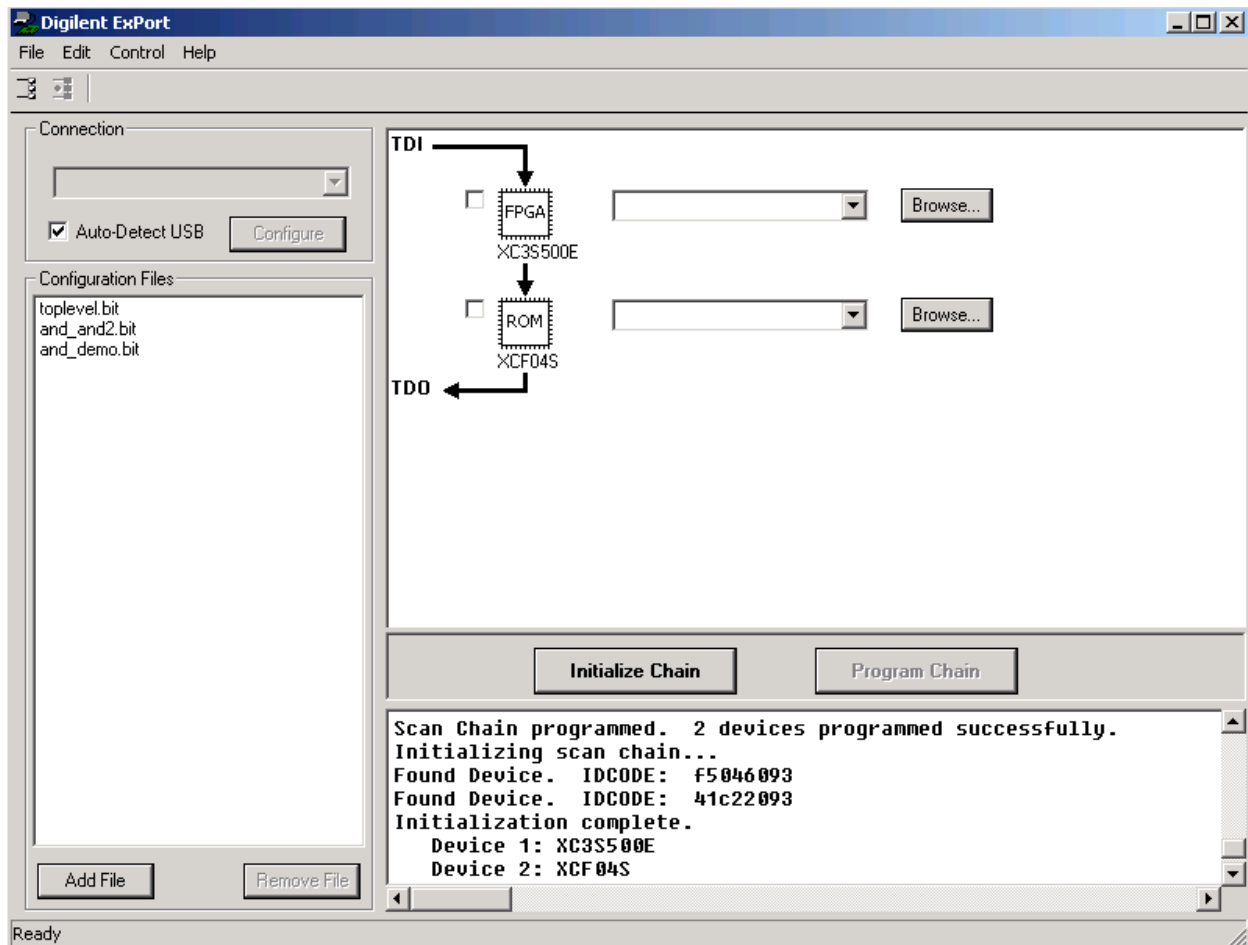
The console window at the bottom shows the message: "Started : 'Generate Programming File'." and "Process 'Generate Programming File' completed successfully".

Now that the programming file has been generated, it is time to load your design into the FPGA, and test it out.

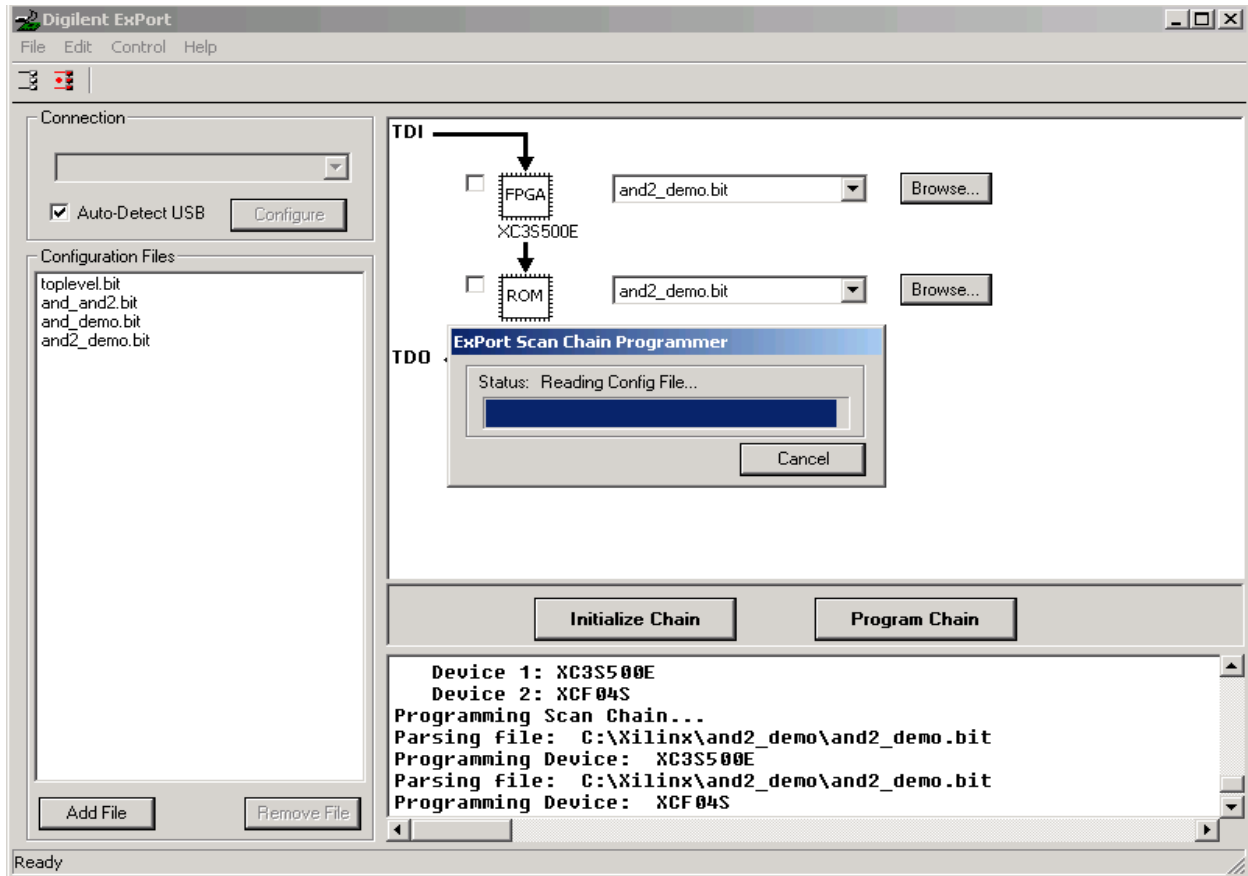
Configuration Using the USB port and Digilent ADEPT Software Suite

Open the Export.exe file from the digilent adept folder.

While the USB is attached, click on "Initialize Chain" to identify the devices on the board.



To program the FPGA, browse to the configuration file and select your and2_demo.bit. Similarly, if you would like to program the board to start on power-up program the PROM (XCF04S) with the file as well.



Click "Program Chain" and wait. When configuration is complete, your FPGA should be operating as your VHDL specified.