Basys 2 Setup Tutorial

Paul Schow

March 26, 2012

Abstract

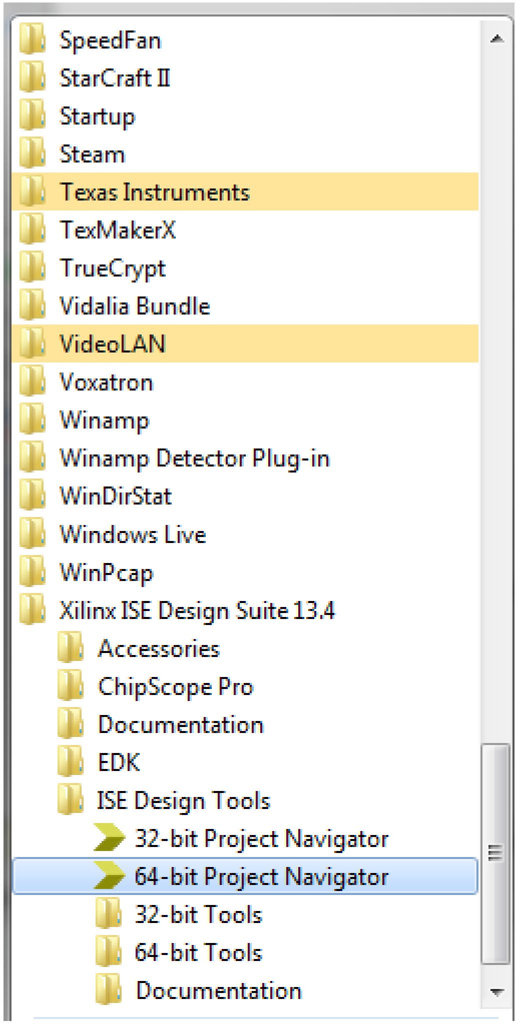
This is a tutorial that explains how to setup the Digilent Basys 2 FPGA board with Xilinx ISE Webpack.

1. Install Xilinx ISE Webpack and Digilent Adept.

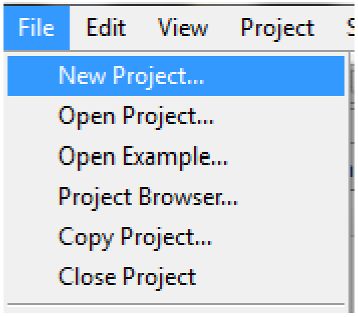
   You can get Xilinx ISE Webpack here:
   `http://www.xilinx.com/products/design-tools/ise-design-suite/ise-webpack.htm`

   You can get Digilent Adept here:
   `http://www.digilentinc.com/Products/Detail.cfm?Prod=ADEPT2`

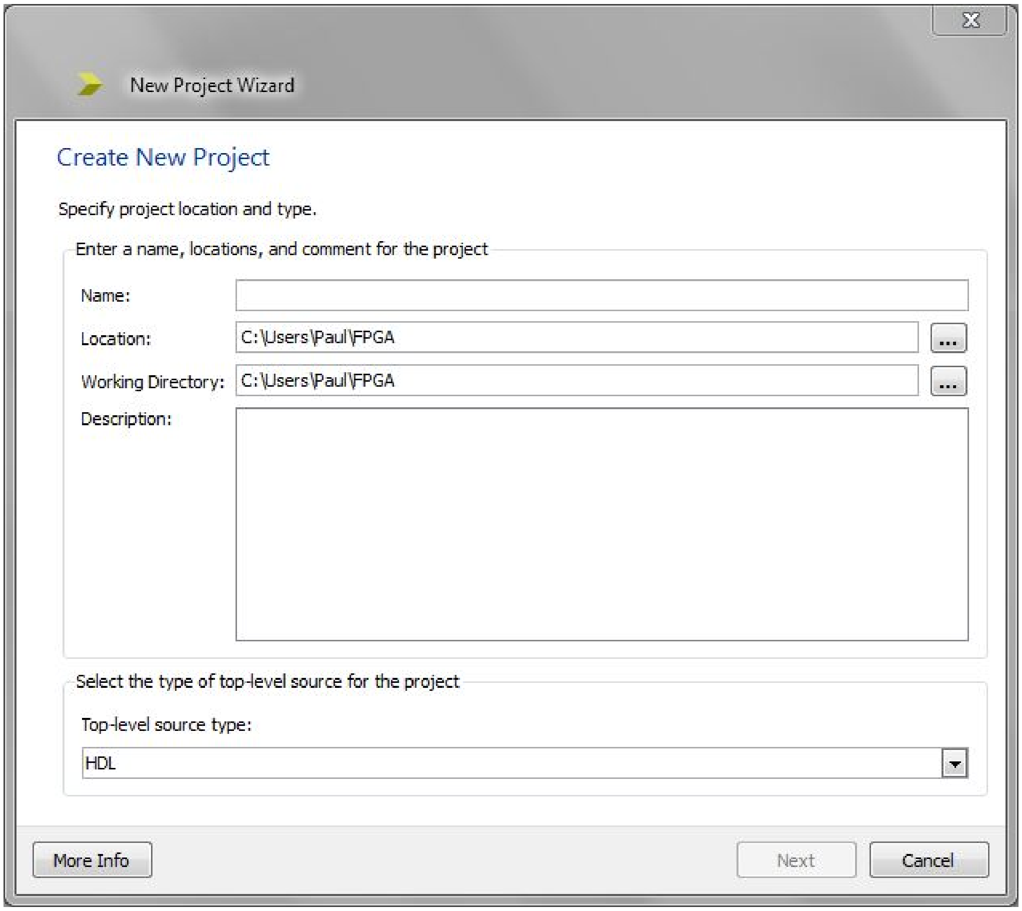
2. Launch Xilinx ISE Webpack from the Start menu.



3. Create a new project in ISE.



4. You should see this window.



New Project Wizard

Create New Project

Specify project location and type.

Enter a name, locations, and comment for the project

Name:

Location: C:\Users\Paul\FPGA

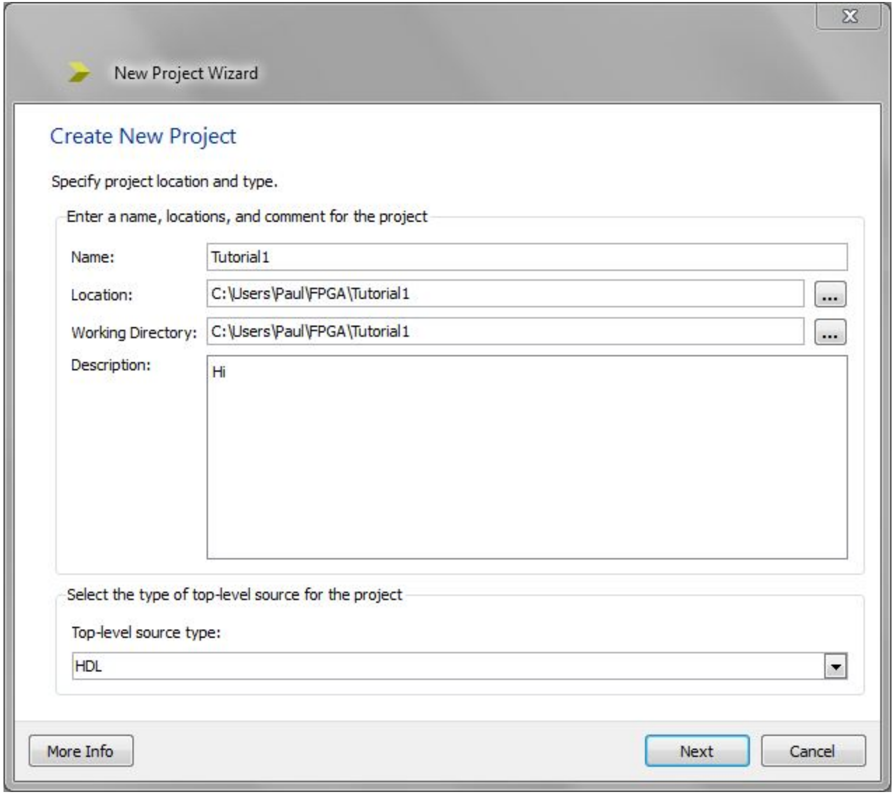
Working Directory: C:\Users\Paul\FPGA

Description:

Select the type of top-level source for the project

Top-level source type: HDL

5. Name the project “Tutorial1” and make sure the top-level source type is HDL.



New Project Wizard

Create New Project

Specify project location and type.

Enter a name, locations, and comment for the project

Name: Tutorial1

Location: C:\Users\Paul\FPGA\Tutorial1

Working Directory: C:\Users\Paul\FPGA\Tutorial1

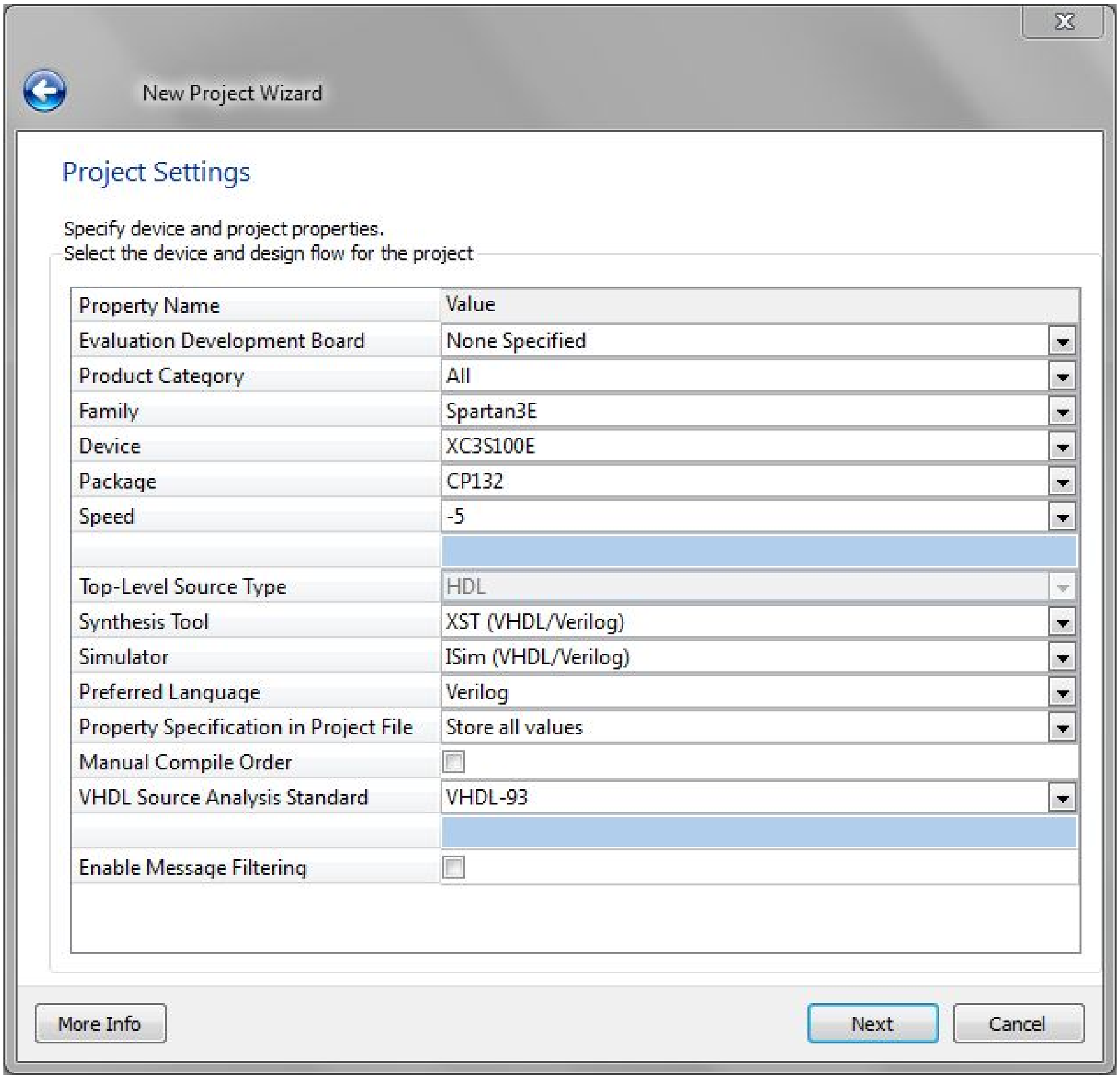
Description: Hi

Select the type of top-level source for the project

Top-level source type: HDL

Also take note of where you are saving the file, this will be important later.

6. All of these settings have to be exactly like they are shown here for the Basys 2 board.



New Project Wizard

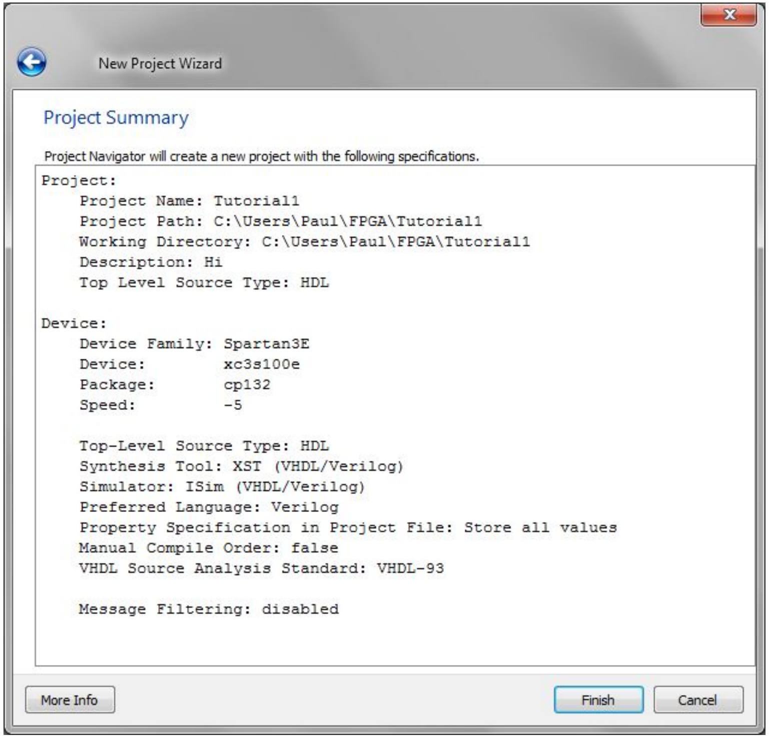
Project Settings

Specify device and project properties.
Select the device and design flow for the project

Property Name	Value
Evaluation Development Board	None Specified
Product Category	All
Family	Spartan3E
Device	XC3S100E
Package	CP132
Speed	-5
Top-Level Source Type	HDL
Synthesis Tool	XST (VHDL/Verilog)
Simulator	ISim (VHDL/Verilog)
Preferred Language	Verilog
Property Specification in Project File	Store all values
Manual Compile Order	☐
VHDL Source Analysis Standard	VHDL-93
Enable Message Filtering	☐

Then click to the next screen.

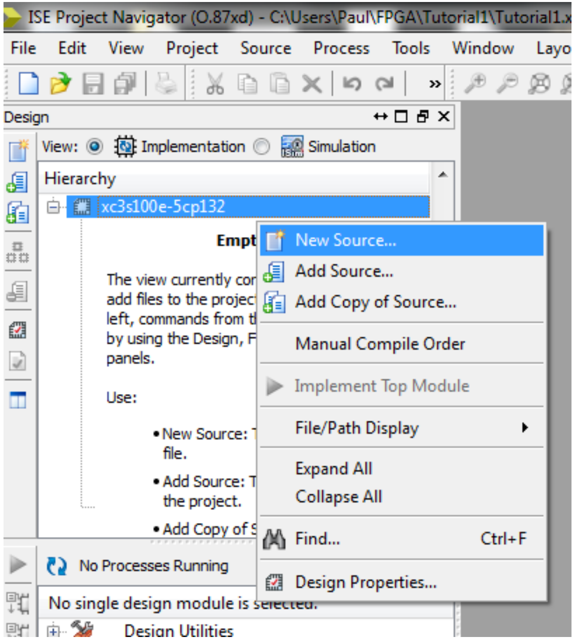
7. Now there is a confirmation screen, click next.



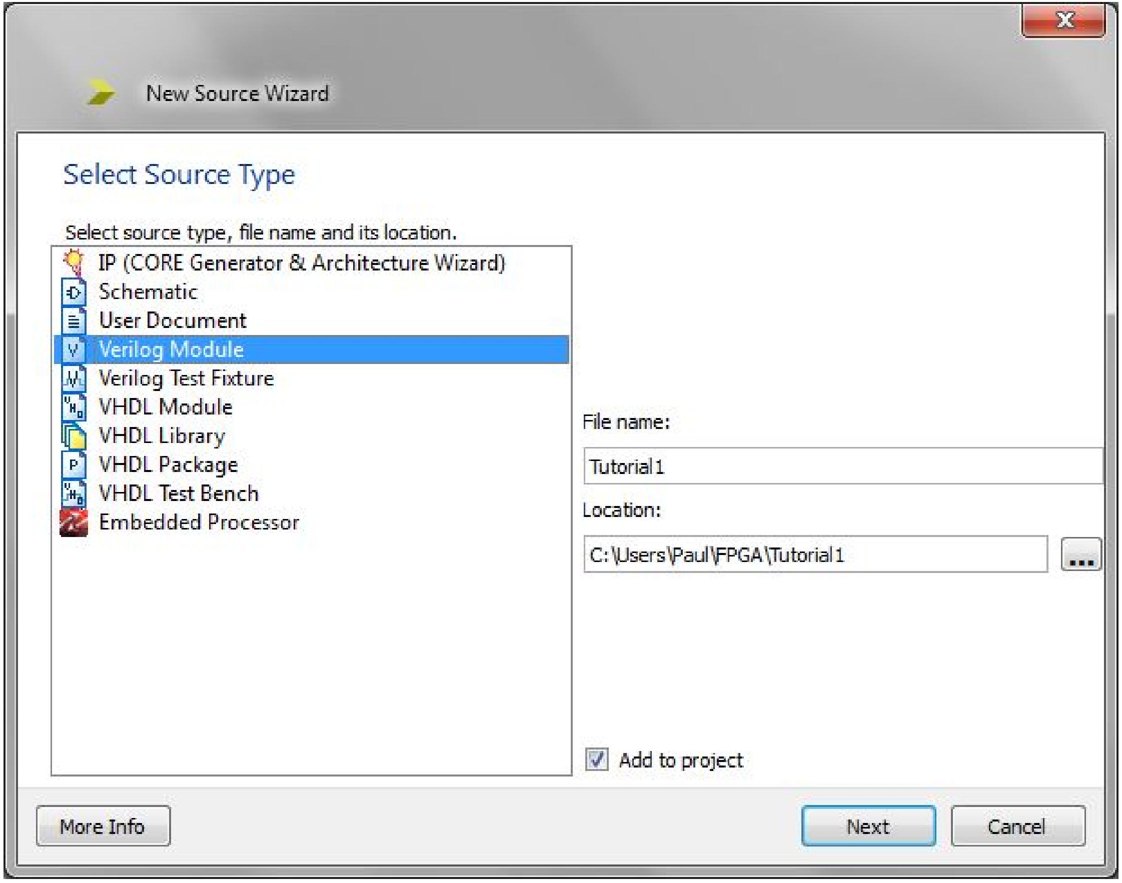
Project Navigator will create a new project with the following specifications.

```
Project:
    Project Name: Tutorial1
    Project Path: C:\Users\Paul\FPGA\Tutorial1
    Working Directory: C:\Users\Paul\FPGA\Tutorial1
    Description: Hi
    Top Level Source Type: HDL

Device:
    Device Family: Spartan3E
    Device:        xc3s100e
    Package:       cp132
    Speed:         -5

    Top-Level Source Type: HDL
    Synthesis Tool: XST (VHDL/Verilog)
    Simulator: ISim (VHDL/Verilog)
    Preferred Language: Verilog
    Property Specification in Project File: Store all values
    Manual Compile Order: false
    VHDL Source Analysis Standard: VHDL-93

    Message Filtering: disabled
```

8. Add a new source file to the project.



9. Make the new file a Verilog module and name it “Turorial1”



New Source Wizard

Select Source Type

Select source type, file name and its location.

Verilog Module

File name: Tutorial1

Location: C:\Users\Paul\FPGA\Tutorial1

☑ Add to project

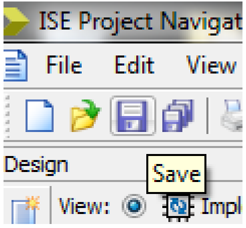
10. On the next page, just press Next to skip the screen.

11. Look at the summary, then click Finish to create the file.

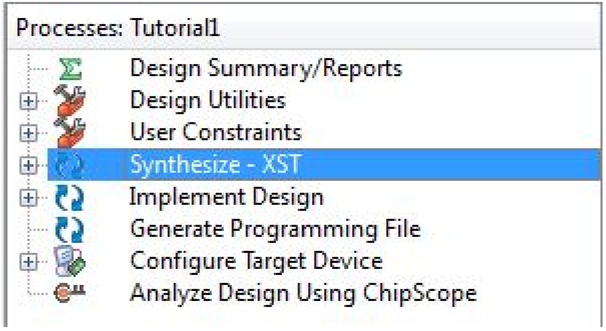
12. Copy this code into the new file.

```
/////////////////////////////////////////////////////////////////////////////////
// three bit counter
// Paul Schow 3-17-2012
// Enable: High = count up
// Enable: Low = count down
// Resets when reset goes low
/////////////////////////////////////////////////////////////////////////////////
module Tutorial1( counterValue, reset, clock, enable);

input enable; // Increment our counter on rising edge of 'count'
input reset; // Reset our counter on rising edge of 'reset'
input clock; // Synchronize outputs to clock
output counterValue; // Internally store the counter value

reg [2:0] counterValue;

// This takes care of synchronously driving the outputs

always @ ( posedge clock) begin //at the positive edge of the clock

  if( reset == 1'b1 ) begin //if reset is high
     counterValue <= 3'b000; //then set the counter to 0000
  end
  else if( enable == 1'b0) begin //if enable is low and reset is high
    counterValue <= counterValue + 1'b1; //count up
  end
  else if( enable == 1'b1) begin //if enable is high and reset is high
  counterValue <= counterValue - 1'b1; //count down
  end
end

endmodule
```

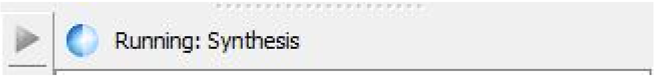
13. Save the file.



14. Double-click Synthesize in the lower left window.

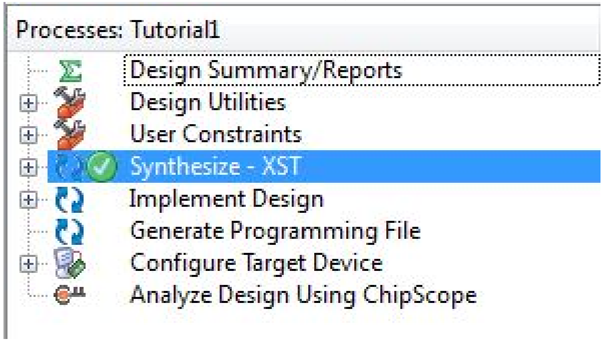


15. You should see it running for a bit.



Running: Synthesis

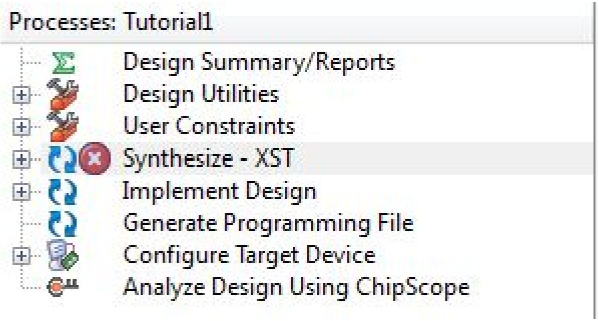
16. When it is done synthesizing, a green check will appear next to Synthesize.



Processes: Tutorial1

Design Summary/Reports
Design Utilities
User Constraints
Synthesize - XST ✓
Implement Design
Generate Programming File
Configure Target Device
Analyze Design Using ChipScope

If you have an error, a red X will appear instead.

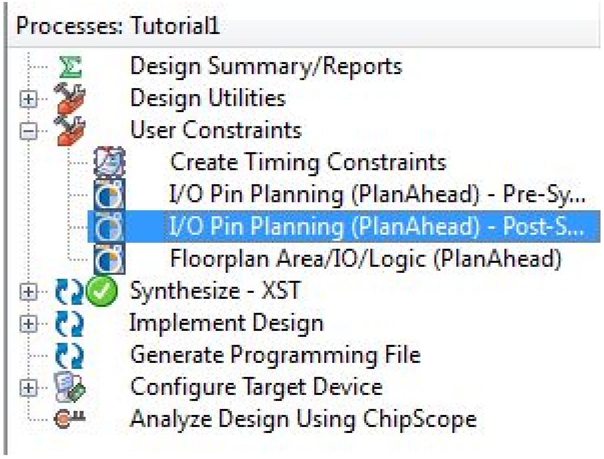


Processes: Tutorial1

Design Summary/Reports
Design Utilities
User Constraints
Synthesize - XST ☒
Implement Design
Generate Programming File
Configure Target Device
Analyze Design Using ChipScope

Go back and check the code if this happens.

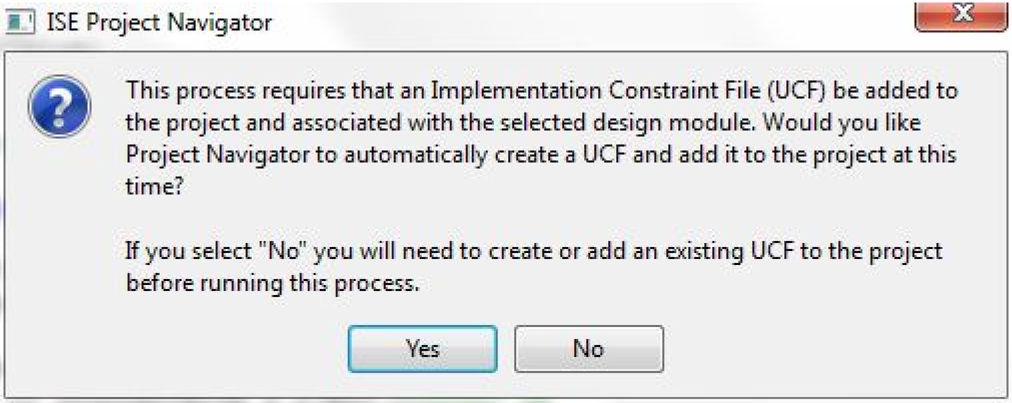
17. Open the User Constraints section and double click "I/O Pin Planning /(Planahead) Post-Synthesis"



Processes: Tutorial1

Design Summary/Reports
Design Utilities
User Constraints
Create Timing Constraints
I/O Pin Planning (PlanAhead) - Pre-Sy...
I/O Pin Planning (PlanAhead) - Post-S...
Floorplan Area/IO/Logic (PlanAhead)
Synthesize - XST ✓
Implement Design
Generate Programming File
Configure Target Device
Analyze Design Using ChipScope

18. Click “Yes” on the window that pops up to create the .UCF file.



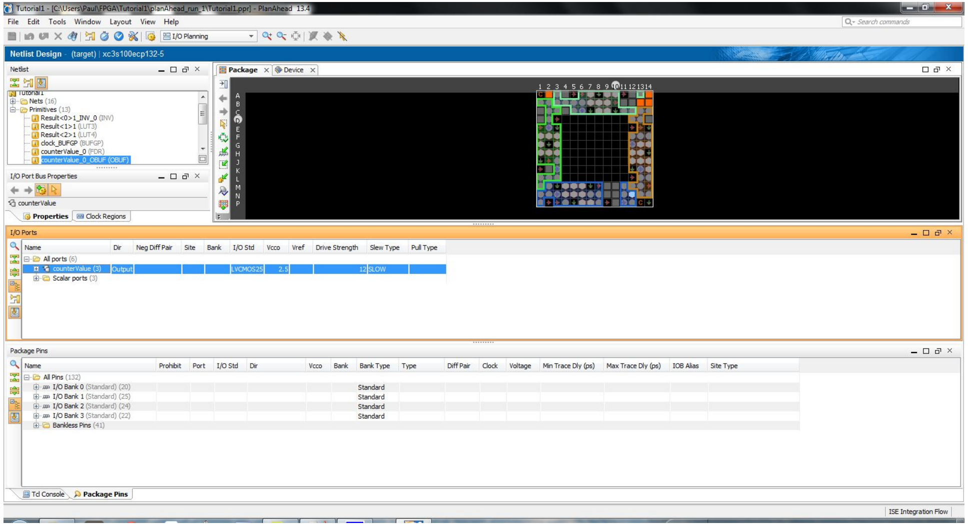
ISE Project Navigator

This process requires that an Implementation Constraint File (UCF) be added to the project and associated with the selected design module. Would you like Project Navigator to automatically create a UCF and add it to the project at this time?

If you select "No" you will need to create or add an existing UCF to the project before running this process.

19. Wait like 5 minutes for PlanAhead to launch.

20. The PlanAhead window looks like this



Tutorial1 - [C:\Users\Paul\FPGA\Tutorial1\planAhead_run_1\Tutorial1.ppr] - PlanAhead 13.4

Netlist Design - (target) | xc3s100ecp132-5

Netlist

- Tutorial1
  - Nets (16)
  - Primitives (13)
    - Result<0>1_INV_0 (INV)
    - Result<1>1 (LUT3)
    - Result<2>1 (LUT4)
    - clock_BUFGP (BUFGP)
    - counterValue_0 (FDR)
    - counterValue_0_OBUF (OBUF)

I/O Port Bus Properties

counterValue

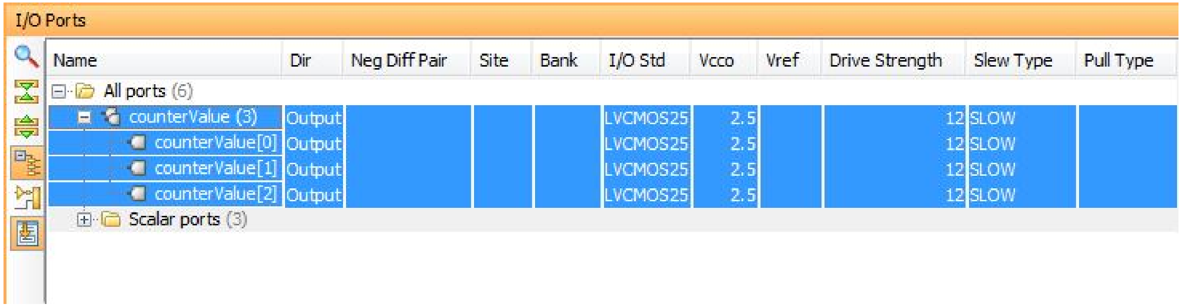
I/O Ports

Name	Dir	Neg Diff Pair	Site	Bank	I/O Std	Vcco	Vref	Drive Strength	Slew Type	Pull Type
All ports (6)										
counterValue (3)	Output				LVCMOS25	2.5		12	SLOW	
Scalar ports (3)										

Package Pins

Name	Prohibit	Port	I/O Std	Dir	Vcco	Bank	Bank Type	Type	Diff Pair	Clock	Voltage	Min Trace Dly (ps)	Max Trace Dly (ps)	IOB Alias	Site Type
All Pins (132)															
I/O Bank 0 (Standard) (20)							Standard								
I/O Bank 1 (Standard) (25)							Standard								
I/O Bank 2 (Standard) (24)							Standard								
I/O Bank 3 (Standard) (22)							Standard								
Bankless Pins (41)															

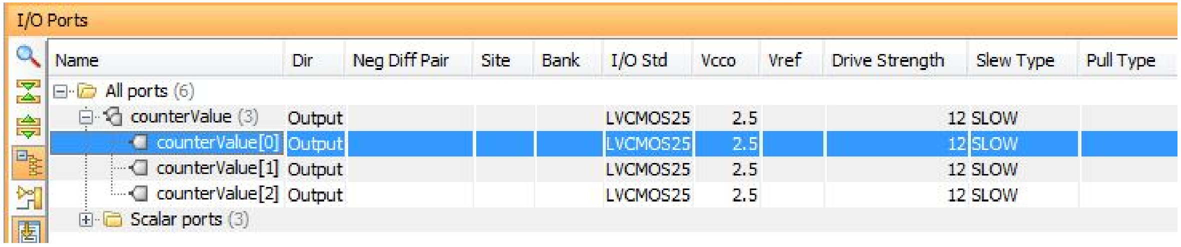
21. In the center window, expand the “Countervalue” list to see the output ports.



I/O Ports

Name	Dir	Neg Diff Pair	Site	Bank	I/O Std	Vcco	Vref	Drive Strength	Slew Type	Pull Type
All ports (6)										
counterValue (3)	Output				LVCMOS25	2.5		12	SLOW	
counterValue[0]	Output				LVCMOS25	2.5		12	SLOW	
counterValue[1]	Output				LVCMOS25	2.5		12	SLOW	
counterValue[2]	Output				LVCMOS25	2.5		12	SLOW	
Scalar ports (3)										

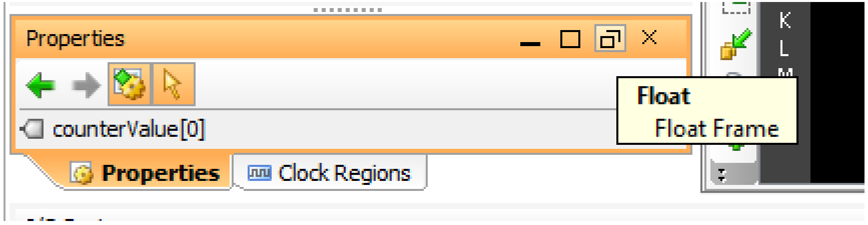
22. Highlight only one port. In this case we start with “Countervalue[0]”



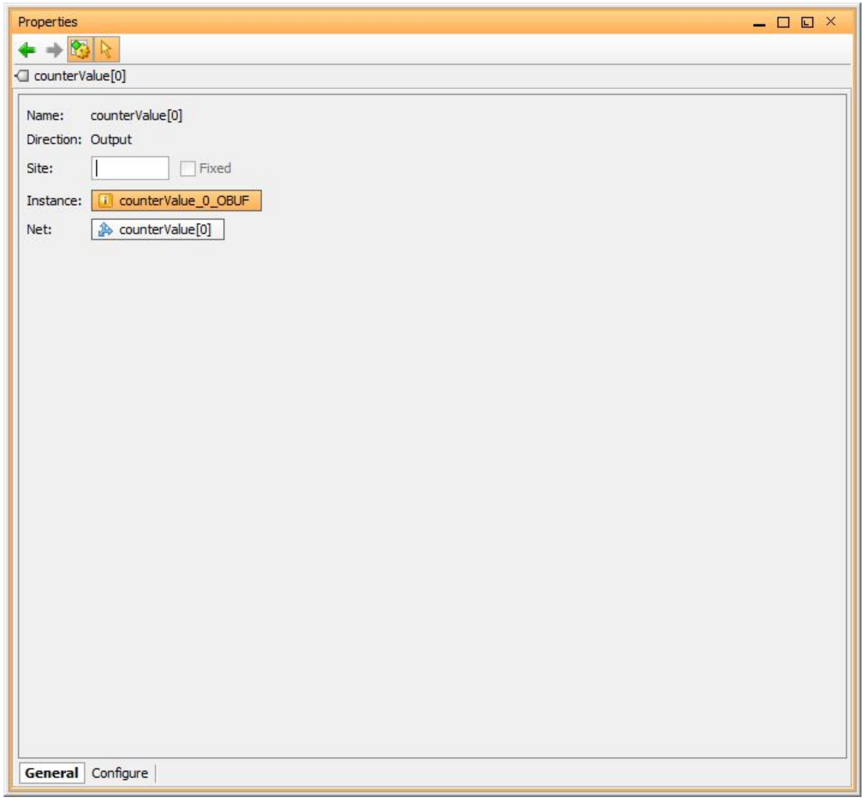
I/O Ports

Name	Dir	Neg Diff Pair	Site	Bank	I/O Std	Vcco	Vref	Drive Strength	Slew Type	Pull Type
All ports (6)										
counterValue (3)	Output				LVCMOS25	2.5		12	SLOW	
counterValue[0]	Output				LVCMOS25	2.5		12	SLOW	
counterValue[1]	Output				LVCMOS25	2.5		12	SLOW	
counterValue[2]	Output				LVCMOS25	2.5		12	SLOW	
Scalar ports (3)										

23. Go to the Properties frame and Float it out so it can be seen



24. You should have the properties window in full now.



Properties

counterValue[0]

Name: counterValue[0]

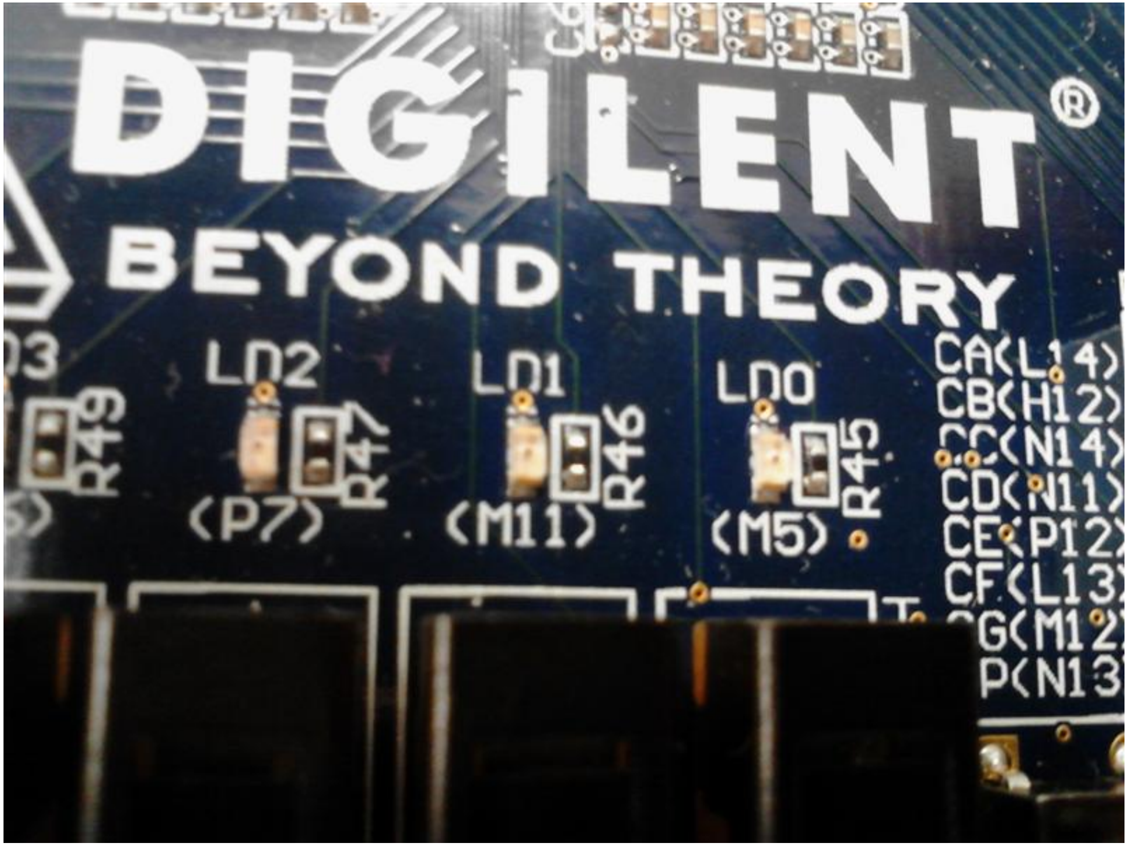
Direction: Output

Site: ☐ Fixed

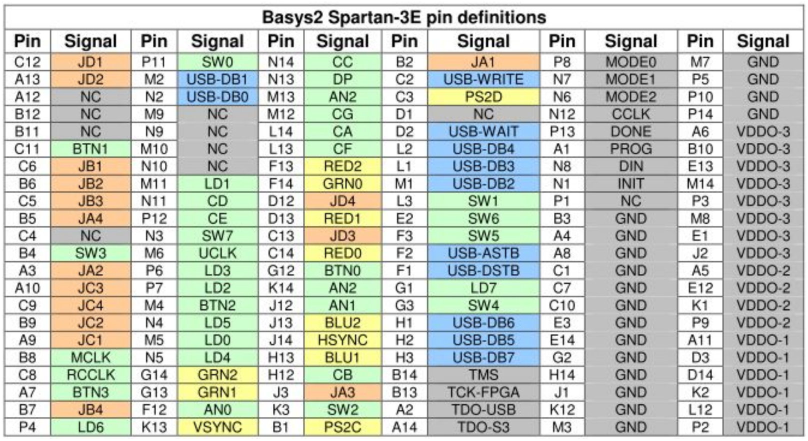
Instance: counterValue_0_OBUF

Net: counterValue[0]

25. Now, look at your Basys 2 board. “Countervalue[0]” is the least significant bit, so we want to map it to LD0.



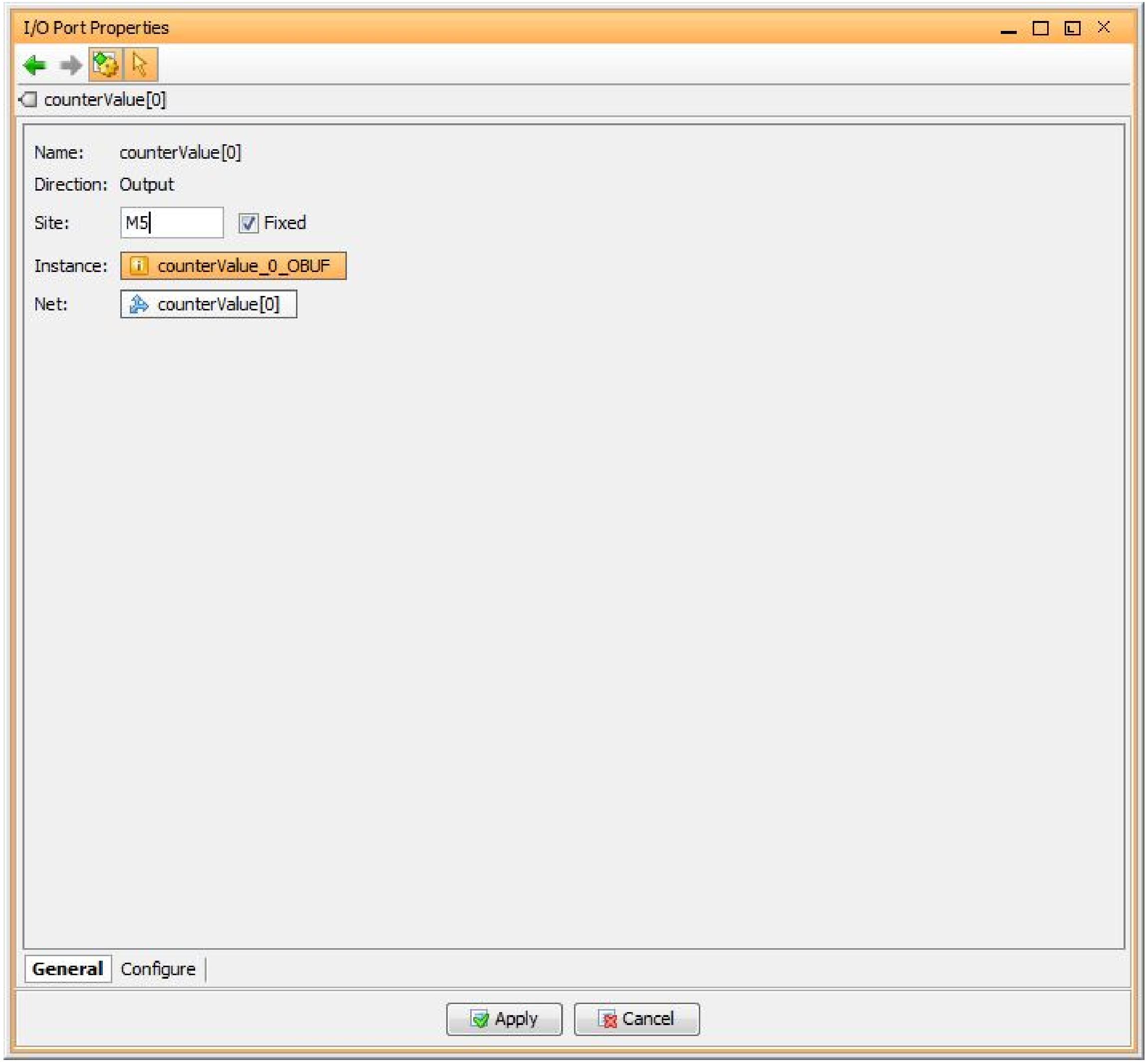
Look under the LED. See how it says “(M5)”? This is the pin that LD0 is on. This information can also be found in the manual for the Basys 2.



Basys2 Spartan-3E pin definitions

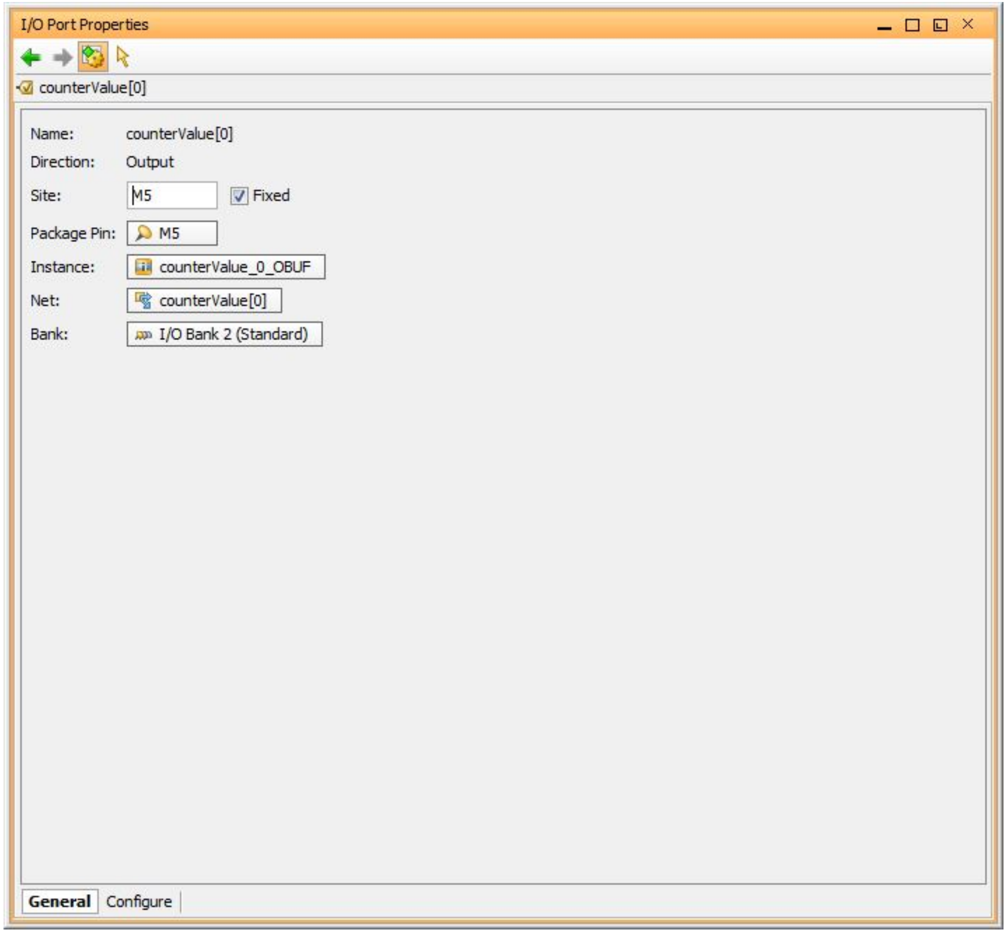
Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal
C12	JD1	P11	SW0	N14	CC	B2	JA1	P8	MODE0	M7	GND
A13	JD2	M2	USB-DB1	N13	DP	C2	USB-WRITE	N7	MODE1	P5	GND
A12	NC	N2	USB-DB0	M13	AN2	C3	PS2D	N6	MODE2	P10	GND
B12	NC	M9	NC	M12	CG	D1	NC	N12	CCLK	P14	GND
B11	NC	N9	NC	L14	CA	D2	USB-WAIT	P13	DONE	A6	VDDO-3
C11	BTN1	M10	NC	L13	CF	L2	USB-DB4	A1	PROG	B10	VDDO-3
C6	JB1	N10	NC	F13	RED2	L1	USB-DB3	N8	DIN	E13	VDDO-3
B6	JB2	M11	LD1	F14	GRN0	M1	USB-DB2	N1	INIT	M14	VDDO-3
C5	JB3	N11	CD	D12	JD4	L3	SW1	P1	NC	P3	VDDO-3
B5	JA4	P12	CE	D13	RED1	E2	SW6	B3	GND	M8	VDDO-3
C4	NC	N3	SW7	C13	JD3	F3	SW5	A4	GND	E1	VDDO-3
B4	SW3	M6	UCLK	C14	RED0	F2	USB-ASTB	A8	GND	J2	VDDO-3
A3	JA2	P6	LD3	G12	BTN0	F1	USB-DSTB	C1	GND	A5	VDDO-2
A10	JC3	P7	LD2	K14	AN2	G1	LD7	C7	GND	E12	VDDO-2
C9	JC4	M4	BTN2	J12	AN1	G3	SW4	C10	GND	K1	VDDO-2
B9	JC2	N4	LD5	J13	BLU2	H1	USB-DB6	E3	GND	P9	VDDO-2
A9	JC1	M5	LD0	J14	HSYNC	H2	USB-DB5	E14	GND	A11	VDDO-1
B8	MCLK	N5	LD4	H13	BLU1	H3	USB-DB7	G2	GND	D3	VDDO-1
C8	RCCLK	G14	GRN2	H12	CB	B14	TMS	H14	GND	D14	VDDO-1
A7	BTN3	G13	GRN1	J3	JA3	B13	TCK-FPGA	J1	GND	K2	VDDO-1
B7	JB4	F12	AN0	K3	SW2	A2	TDO-USB	K12	GND	L12	VDDO-1
P4	LD6	K13	VSYNC	B1	PS2C	A14	TDO-S3	M3	GND	P2	VDDO-1

26. Now that you know LD0 is mapped to M5 on the Basys 2, go back to PlanAhead and put in M5.



Name: counterValue[0]
Direction: Output
Site: M5 ☑ Fixed
Instance: counterValue_0_OBUF
Net: counterValue[0]

27. Click “Apply” and the window should change to this.



Name: counterValue[0]

Direction: Output

Site: M5 ☑ Fixed

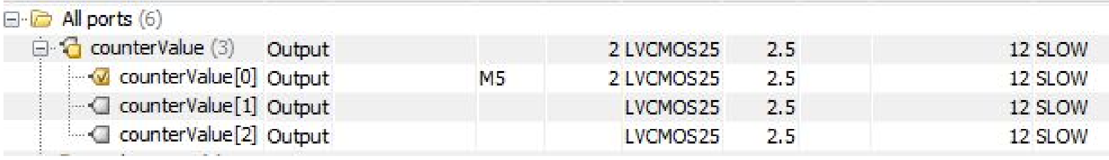
Package Pin: M5

Instance: counterValue_0_OBUF

Net: counterValue[0]

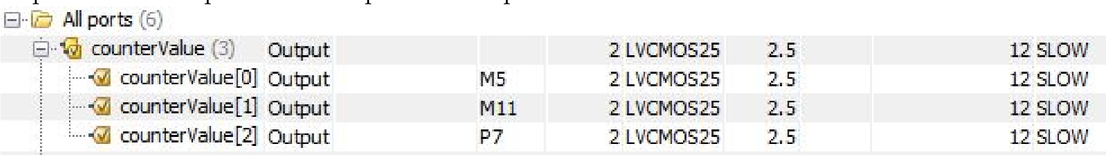
Bank: I/O Bank 2 (Standard)

It will also change on the main window.



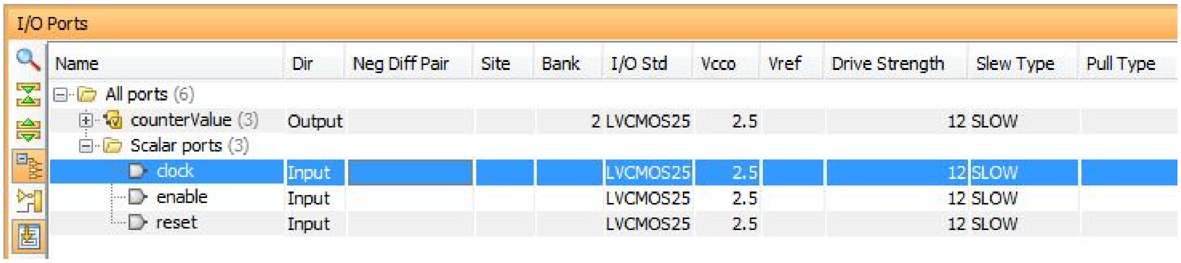
All ports (6)							
counterValue (3)	Output		2	LVCMOS25	2.5	12	SLOW
counterValue[0]	Output	M5	2	LVCMOS25	2.5	12	SLOW
counterValue[1]	Output			LVCMOS25	2.5	12	SLOW
counterValue[2]	Output			LVCMOS25	2.5	12	SLOW

28. Repeat the same process to map the other pins to the other LED’s.



All ports (6)							
counterValue (3)	Output		2	LVCMOS25	2.5	12	SLOW
counterValue[0]	Output	M5	2	LVCMOS25	2.5	12	SLOW
counterValue[1]	Output	M11	2	LVCMOS25	2.5	12	SLOW
counterValue[2]	Output	P7	2	LVCMOS25	2.5	12	SLOW

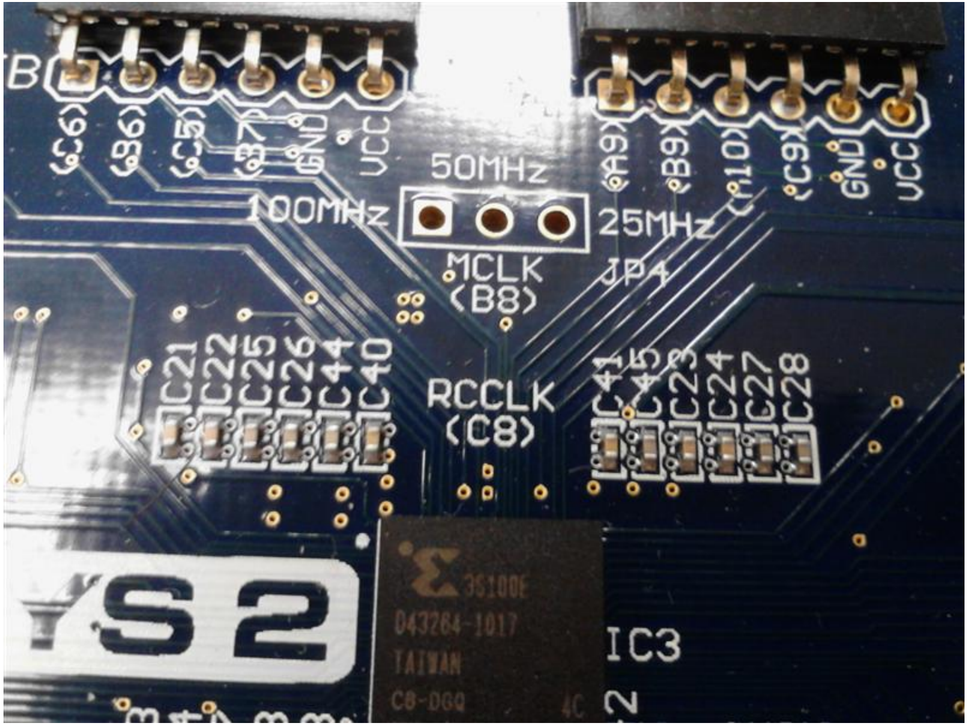
29. Now move to the “Scalar Ports” section. This is the inputs.



I/O Ports

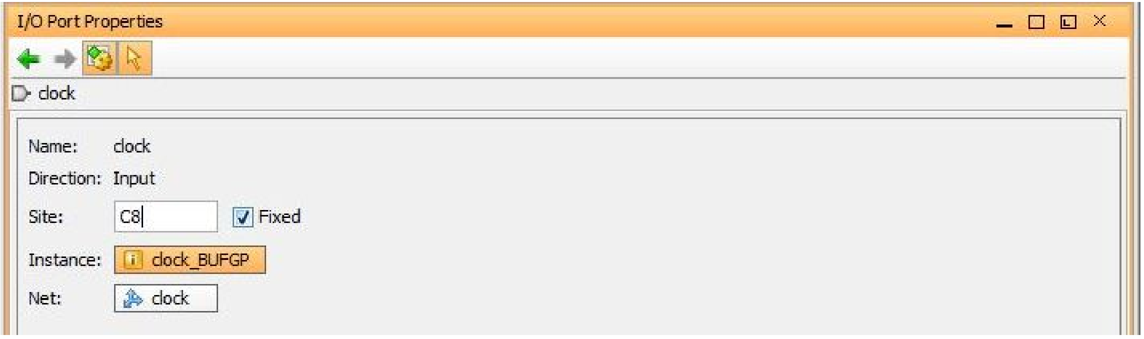
Name	Dir	Neg Diff Pair	Site	Bank	I/O Std	Vcco	Vref	Drive Strength	Slew Type	Pull Type
All ports (6)										
counterValue (3)	Output			2	LVCMOS25	2.5		12	SLOW	
Scalar ports (3)										
clock	Input				LVCMOS25	2.5		12	SLOW	
enable	Input				LVCMOS25	2.5		12	SLOW	
reset	Input				LVCMOS25	2.5		12	SLOW	

30. We need a clock. The Basys 2 has a 50mhz clock built in on pin C8.



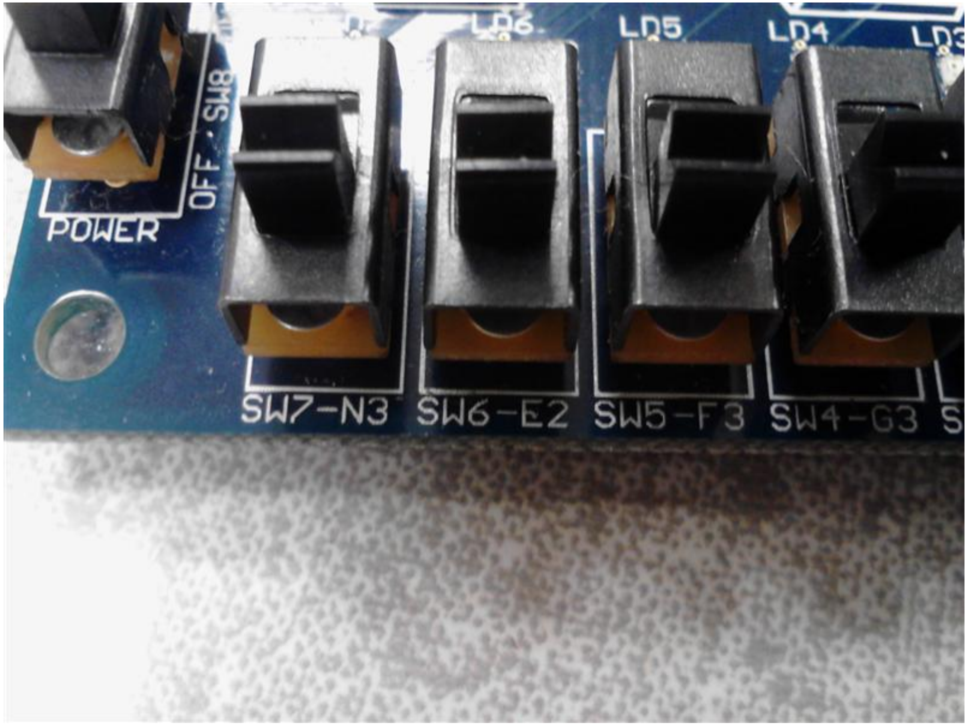
This will work fine.

31. Set the Clock to use C8.

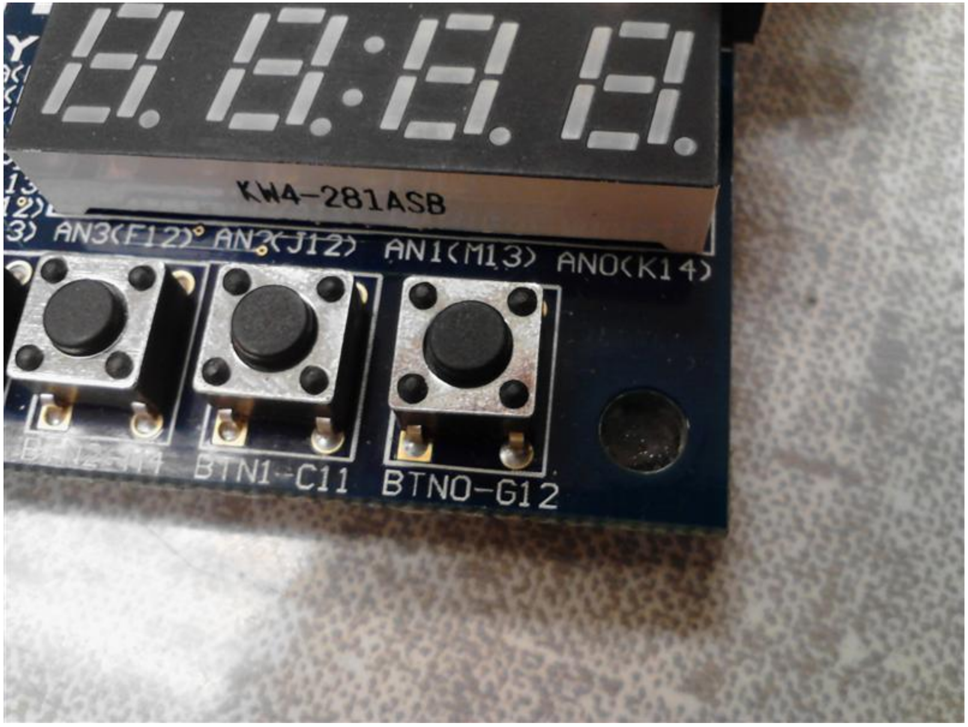


Name: clock
Direction: Input
Site: C8 ☑ Fixed
Instance: clock_BUFGP
Net: clock

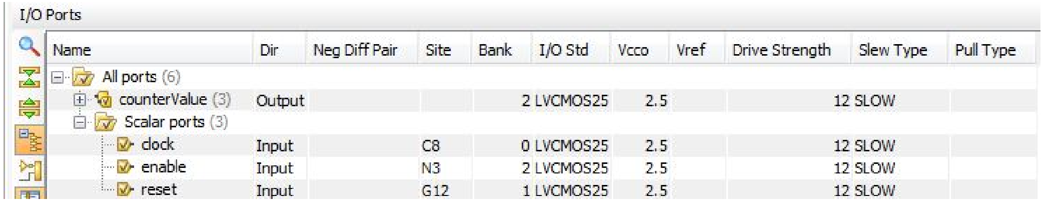
32. The Enable pin needs a switch. I use SW7, which is mapped to pin N3.



33. Reset uses a button. Use BTN0, mapped to pin G12.

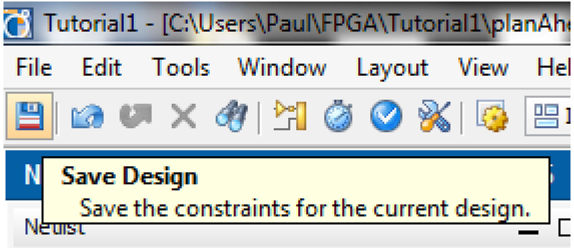


34. Now the main window should look like this.



Name	Dir	Neg Diff Pair	Site	Bank	I/O Std	Vcco	Vref	Drive Strength	Slew Type	Pull Type
All ports (6)										
counterValue (3)	Output			2	LVCMOS25	2.5		12	SLOW	
Scalar ports (3)										
clock	Input		C8	0	LVCMOS25	2.5		12	SLOW	
enable	Input		N3	2	LVCMOS25	2.5		12	SLOW	
reset	Input		G12	1	LVCMOS25	2.5		12	SLOW	

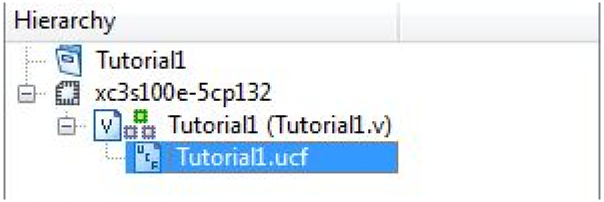
35. Save the file and exit PlanAhead.



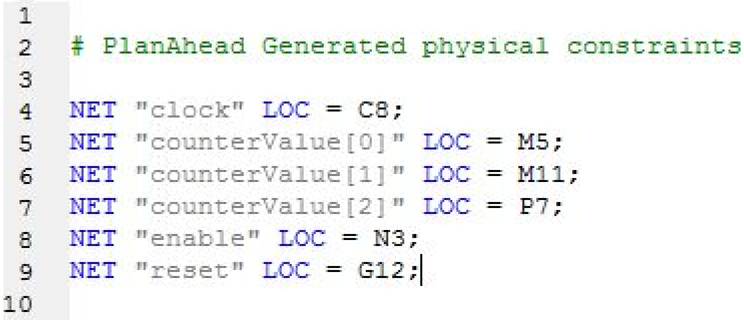
Save Design

Save the constraints for the current design.

36. PlanAhead created the .UCF, or User Constraints File for you. Double-click on it to open it.



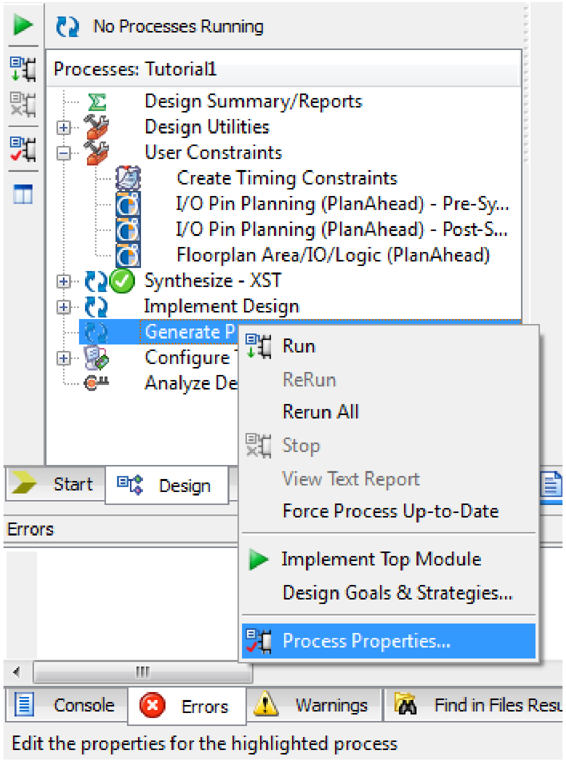
It looks like this.



```

# PlanAhead Generated physical constraints

NET "clock" LOC = C8;
NET "counterValue[0]" LOC = M5;
NET "counterValue[1]" LOC = M11;
NET "counterValue[2]" LOC = P7;
NET "enable" LOC = N3;
NET "reset" LOC = G12;

```

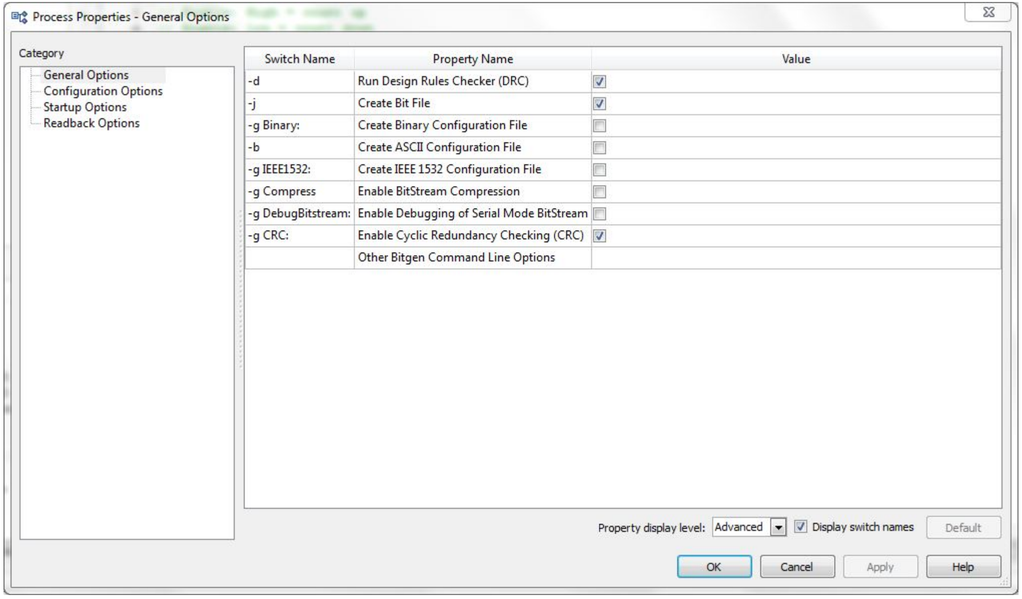
You can see that the .UCF is pretty simple when only working with a few pins. It can be edited directly from here, without using PlanAhead if you want to.

37. Go to the bottom left panel and right-click on “Generate Programming File”. Then click on “Process Properties”.



Edit the properties for the highlighted process

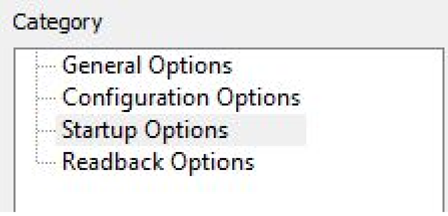
38. You are now in the Process Properties window.



Switch Name	Property Name	Value
-d	Run Design Rules Checker (DRC)	☑
-j	Create Bit File	☑
-g Binary:	Create Binary Configuration File	□
-b	Create ASCII Configuration File	□
-g IEEE1532:	Create IEEE 1532 Configuration File	□
-g Compress	Enable BitStream Compression	□
-g DebugBitstream:	Enable Debugging of Serial Mode BitStream	□
-g CRC:	Enable Cyclic Redundancy Checking (CRC)	☑
	Other Bitgen Command Line Options	

Property display level: Advanced ☑ Display switch names

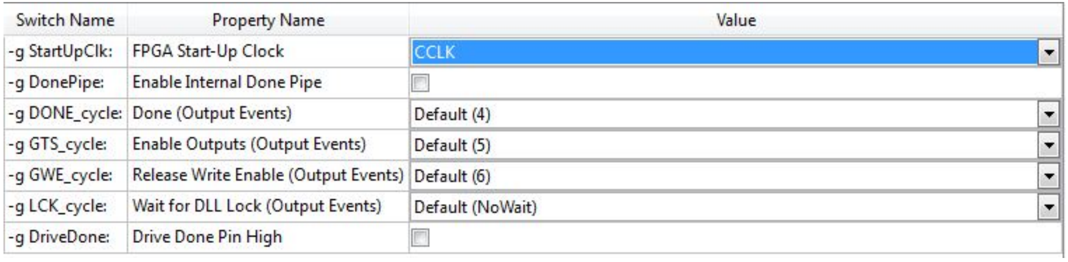
39. Click on “Startup Options” in the left pane.



Category

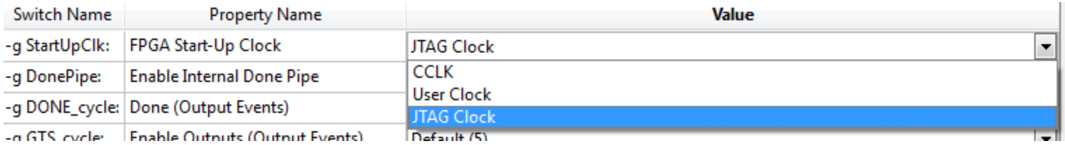
- General Options
- Configuration Options
- Startup Options
- Readback Options

40. Go to “FPGA Startup Clock”.



Switch Name	Property Name	Value
-g StartUpClk:	FPGA Start-Up Clock	CCLK
-g DonePipe:	Enable Internal Done Pipe	□
-g DONE_cycle:	Done (Output Events)	Default (4)
-g GTS_cycle:	Enable Outputs (Output Events)	Default (5)
-g GWE_cycle:	Release Write Enable (Output Events)	Default (6)
-g LCK_cycle:	Wait for DLL Lock (Output Events)	Default (NoWait)
-g DriveDone:	Drive Done Pin High	□

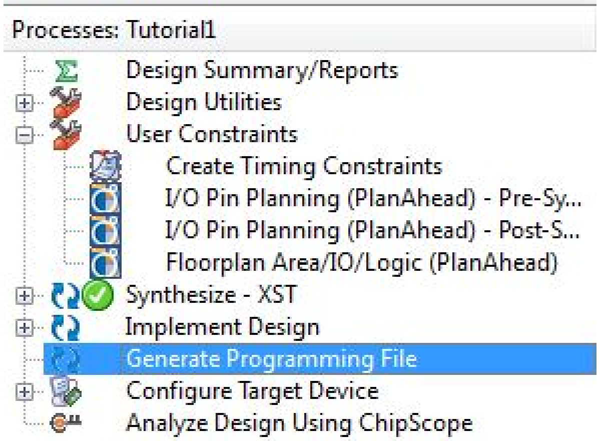
41. Change to clock the “JTAG clock”



Switch Name	Property Name	Value
-g StartUpClk:	FPGA Start-Up Clock	JTAG Clock
-g DonePipe:	Enable Internal Done Pipe	CCLK
-g DONE_cycle:	Done (Output Events)	User Clock JTAG Clock
-g GTS_cycle:	Enable Outputs (Output Events)	Default (5)

42. Click “OK” to save the properties.

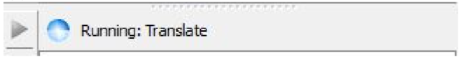
43. Now back in the main window, double-click on “Generate Programming File”



Processes: Tutorial1

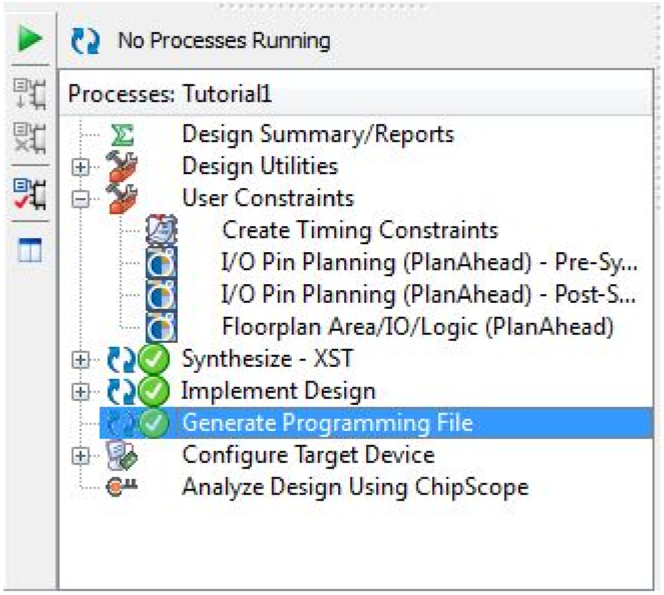
- Design Summary/Reports
- Design Utilities
- User Constraints
  - Create Timing Constraints
  - I/O Pin Planning (PlanAhead) - Pre-Sy...
  - I/O Pin Planning (PlanAhead) - Post-S...
  - Floorplan Area/IO/Logic (PlanAhead)
- Synthesize - XST
- Implement Design
- Generate Programming File
- Configure Target Device
- Analyze Design Using ChipScope

44. Wait for it to finish.

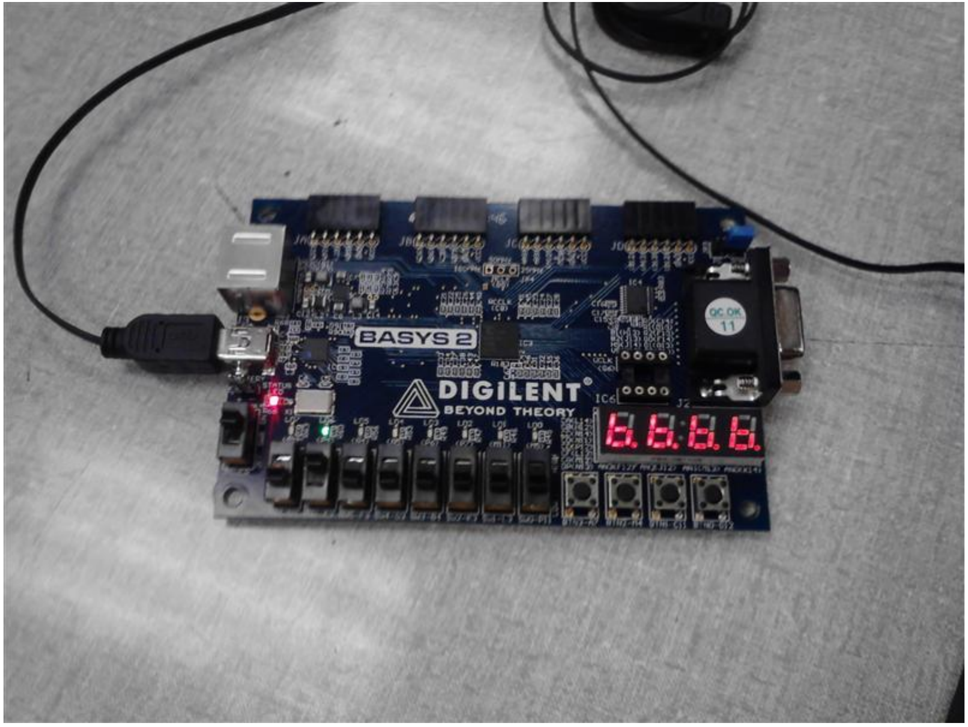


Running: Translate

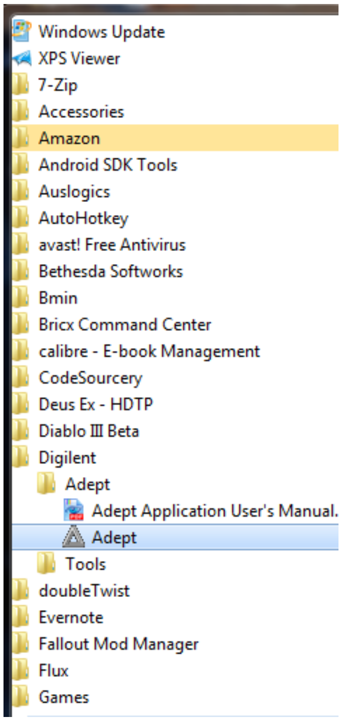
45. When it is done, there will be three green checks.


No Processes Running
Processes: Tutorial1
Design Summary/Reports
Design Utilities
User Constraints
Create Timing Constraints
I/O Pin Planning (PlanAhead) - Pre-Sy...
I/O Pin Planning (PlanAhead) - Post-S...
Floorplan Area/IO/Logic (PlanAhead)
Synthesize - XST
Implement Design
Generate Programming File
Configure Target Device
Analyze Design Using ChipScope

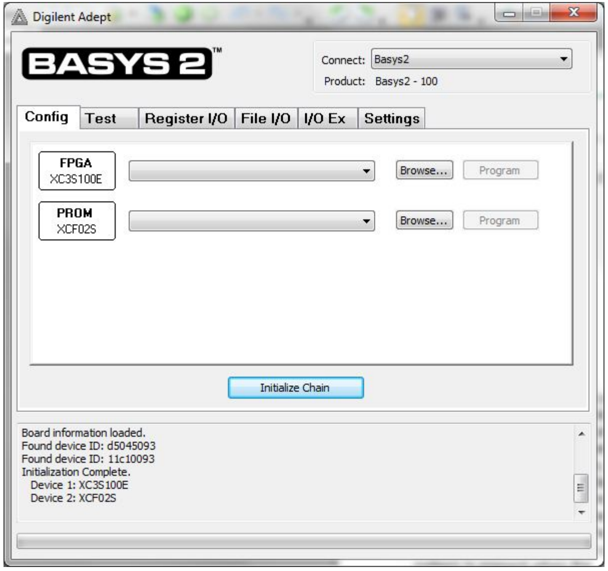
46. Now plug in the Basys 2 and power it on!



47. Go to the Start menu and run Adept.



48. Here is the Adept window.

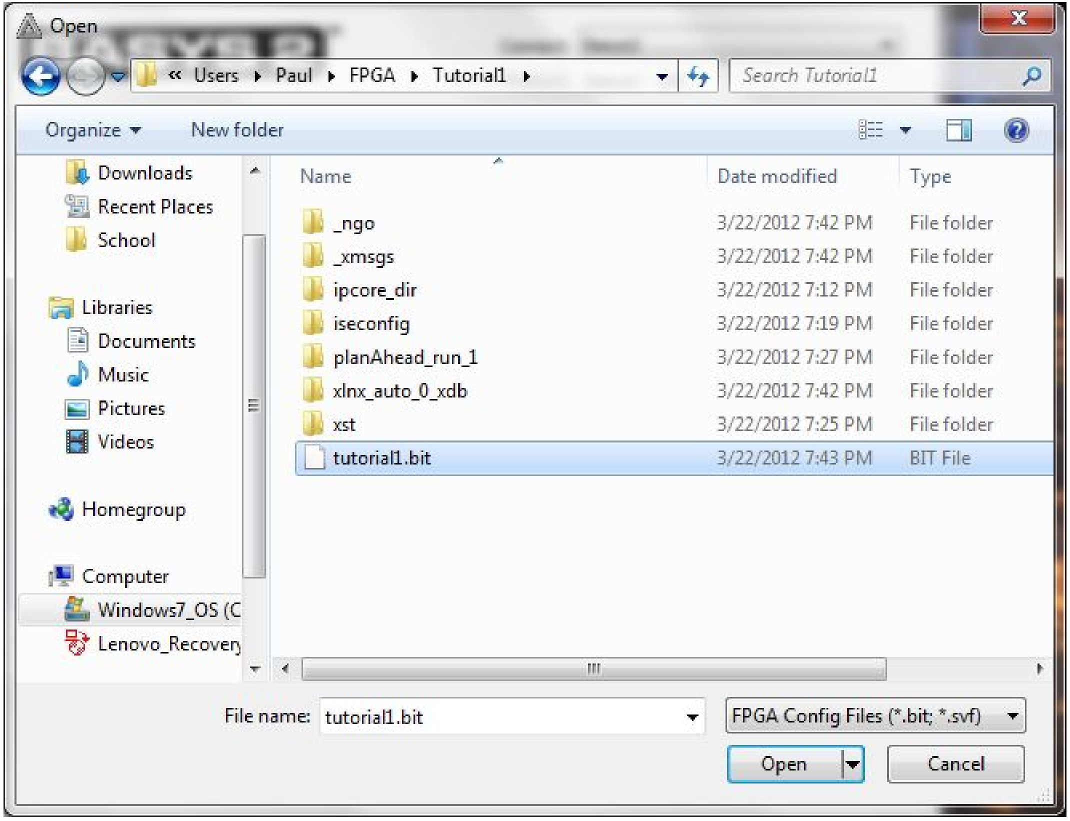


Connect: Basys2

Product: Basys2 - 100

```
Board information loaded.
Found device ID: d5045093
Found device ID: 11c10093
Initialization Complete.
  Device 1: XC3S100E
  Device 2: XCF02S
```

49. Next to FPGA click “Browse”. Browse your computer for the “Tutorial1.bit” file.

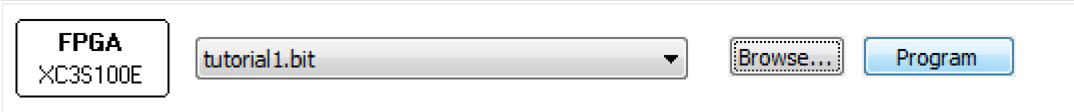


Name	Date modified	Type
_ngo	3/22/2012 7:42 PM	File folder
_xmsgs	3/22/2012 7:42 PM	File folder
ipcore_dir	3/22/2012 7:12 PM	File folder
iseconfig	3/22/2012 7:19 PM	File folder
planAhead_run_1	3/22/2012 7:27 PM	File folder
xlnx_auto_0_xdb	3/22/2012 7:42 PM	File folder
xst	3/22/2012 7:25 PM	File folder
tutorial1.bit	3/22/2012 7:43 PM	BIT File

File name: tutorial1.bit

FPGA Config Files (*.bit; *.svf)

50. Now click Program!



tutorial1.bit

51. If all went well you should now have a functioning 3-bit BCD counter!