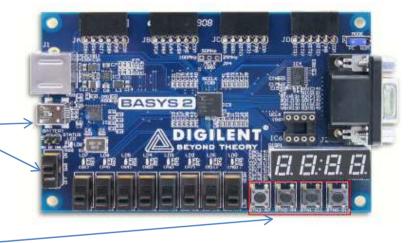
Week 5

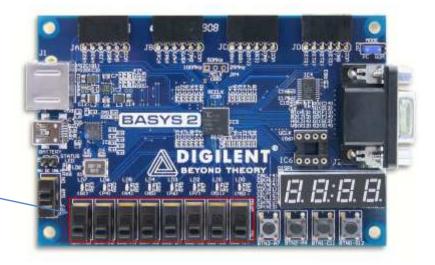
I/O on the BASYS 2 Board

- Power Supply
 - Switch Selectable (SW8)
 - Options
 - USB
 - External (3.5VDC 5.5VDC Power Supply)

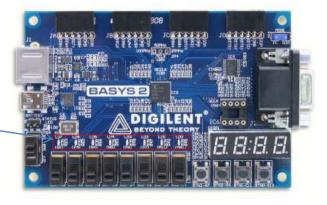
• Pushbuttons

- Four Pushbuttons
- Normally low
- Driven high when button is pressed
- Short Circuit Protection Provided
 - Connecting FPGA pin as output could result in short circuit
 - Remedied with resistor
- Refer to page 4 of BASYS 2 Reference Manual



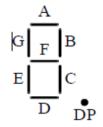


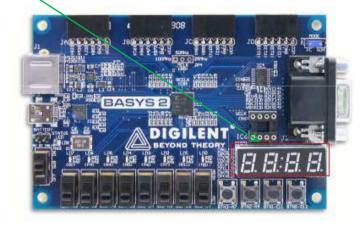
- Slider Switches
 - Eight Slider Switches
 - Constant high or low signal
 - Short Circuit Protection Provided
 - Connecting FPGA pin as output could result in short circuit
 - Remedied with resistor
 - Refer to page 4 of BASYS 2 Reference Manual
- LEDs
 - Eight LEDs –
 - Anodes driven from FPGA
 - Logic 1 illuminates LED
 - Cathodes connected to ground via resistor
 - Refer to page 4 of BASYS 2 Reference Manual



• Seven-Segment Displays

- Four Seven-Segment Displays
- Common Anode
- Enabling Digits
 - AN0...AN3
 - Active Low
 - When enabled, cathodes (CA, CB, ..., CF, CG, DP) can be driven
 - Low signal illuminates segment
- Illuminating multiple digits with different values
 - All digits can APPEAR continuously illuminated if driven once every 1 to 16 ms (1 KHz to 60 Hz)
- Refer to pages 4-5 of BASYS 2 Reference Manual



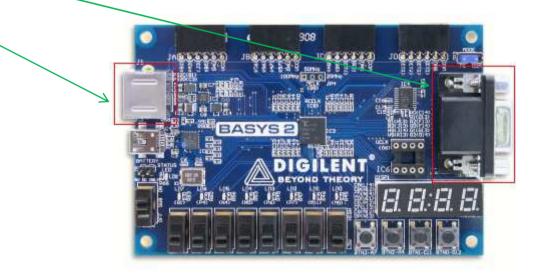


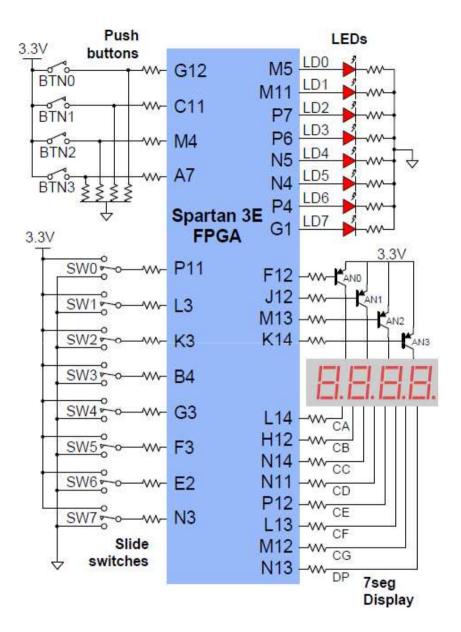
• Oscillators

- Primary Oscillator
 - User selectable (soldering required)
 - 25, 50, or 100 MHz
 - Refer to page 3 of BASYS Reference Manual
 - CLK1
 - Pin B8 of FPGA
- Secondary Oscillator
 - Socket provided
 - CLK2

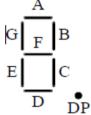
• PS/2 &VGA Port

– PS/2 & VGA Ports Available





To display Number "1" on First Segment , then we need to have: - AN1 to be activated which means "0" (active low) - CB, CC are activated which means "0" and every other pins to be inactivated which means "1"



Basys2 I/O circuits

Managing a Large Project

• Managing a Large Project

- Tasks
- Functions
- Modules
- Tasks
 - Can enable other tasks & functions
 - May execute in non-zero simulation time
 - May contain delay, event, or timing control statements
 - May have zero or more arguments
 - Туре
 - Input
 - Output
 - Inout
 - Can pass values, but do NOT return a value
 - Syntax
 - Definition
 - task Task_Name; input, Output, & Local Variable List begin body end endtask
 - Call

Task_Name(Argument_List);

• Functions

- Can enable another function
 - Not a task
- Execute in zero simulation time
- Can NOT contain
 - A delay event
 - Timing control statements
- Must have at least one input argument
- Always return a single value
 - Output & InOut arguments NOT allowed
- Syntax
 - Definition

function Function_Name; // Note name may represent a vector Input & Local Variable List

begin

Body - Note the output must be assigned a value end

endfunction

• Call

Function_Name(Argument_List);

Tasks & Functions

- Local to the module
- They can contain
 - Local variables
 - Registers
 - Time Variables
 - Integers
 - Reals
 - Events
- Cannot contain
 - Wires
 - Always Blocks
 - Initial Blocks
- Behavioral only!

Modules

• Instantiating modules can help make code easier to write, modify, read, and debug

Strobe Example

- Description
 - The number '12' is displayed on the middle two digits of the 7segment display
 - LED blinks at the frequency used to strobe the 7-segment display
 - Reset
 - Switch 0
 - When low, the 7-segment display is disabled
 - When high, '12' is displayed on the 7-segment display
 - Pin Assignments
 - Reset (P11)
 - Clock (B8)
 - LD0 (M5)
 - Anodes 1...4 (F12, J12, M13, K14)
 - Seg A...G (L14, H12, N14, N11, P12, L13, M12)

- In this example we are going to download Nonvolatile Designs to the FPGA PROM on the BASYS 2 Board
- If you are going to Print your design on PROM you **don't** need to change the clock to JTAG

- Open the ISE project Navigator

-

🗺 ISE Project Navigator (M.81d)	- 6 💌
<u>F</u> ile <u>E</u> dit <u>V</u> iew P <u>r</u> oject <u>S</u> ource <u>P</u> rocess <u>T</u> ools <u>W</u> indow Layout <u>H</u> elp	
Start ↔ □ ♂ ×	
Welcome to the ISE® Design Suite	
Project commands	
Open Project Project Browser	
New Project Open Example	
Recent projects	
Double click on a project in the list below to open	
safiwan777	
Safiwan123	
MainProject1 project4	
Additional resources	
ISE Design Suite InfoCenter	
Key New Features in Project Navigator Tutorials on the Web	
Design Resources	
Application Notes	
Warnings	⇔⊡₽×
✓ III	
😢 Errors 🔬 Warnings	

- Type the project name

		×	
	🚾 New Project Wi	zard	
C	Create New Pro	ject	
s	pecify project location	n and type.	
	-Enter a name, locati	ons, and comment for the project	
	N <u>a</u> me:	SevenSegments	
	Location:	C:\temp\SevenSegments	
	Working Directory:	C:\temp\SevenSegments	
	Description:		
	Select the tupe of to	p-level source for the project	
	<u>T</u> op-level source typ		
	Schematic		
	L		
M	ore Info	<u>N</u> ext Cancel	

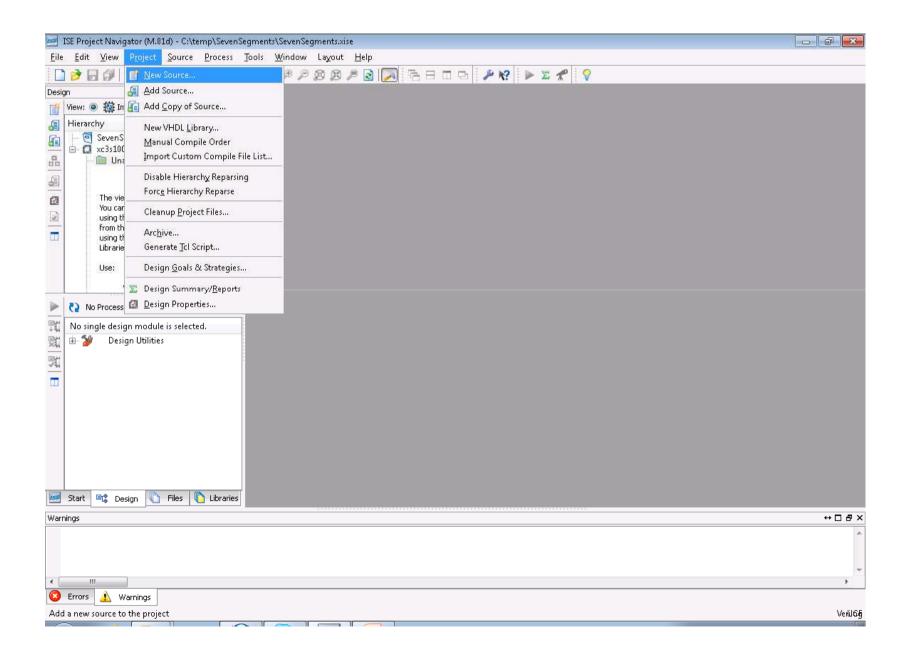
- Select these options then click next

Project Settings	
Specify device and project properties. Select the device and design flow for the pr	roject
Property Name	Value
Product Category	All
Family	Spartan 3E
Device	XC3S100E
Package	CP132 .
Speed	-5
Top-Level Source Type	Schematic
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	

- click finish

🚱 🔤 New Project Wizard
Project Summary
Project Summary
Project Navigator will create a new project with the following specifications.
Project:
Project Name: SevenSegments
Project Path: C:\temp\SevenSegments
Working Directory: C:\temp\SevenSegments
Description:
Top Level Source Type: Schematic
Device:
Device Family: Spartan3E
Device: xc3s100e
Package: cp132
Speed: -5
Synthesis Tool: XST (VHDL/Verilog)
Simulator: Modelsim-PE Mixed
Preferred Language: Verilog
Property Specification in Project File: Store all values
Manual Compile Order: false
VHDL Source Analysis Standard: VHDL-93
Message Filtering: disabled
More Info

- Select Project \rightarrow new source



- Select Verilog module and write the file name, then click next

🚾 New Source Wizard	
Select Source Type, file name and its location. Select source type, file name and its location. Select Source type, file name and its location. Schematic User Document User Document Verilog Module Verilog Test Fixture VHDL Module VHDL Library VHDL Package VHDL Test Bench Sembedded Processor	File name: SevenSegModule Location: C:\temp\SevenSegments
More Info	Next Cancel

- click next

Define Module					
Specify ports for module.					
Module name SevenSegModule					
Port Name	Directio	n	Bus	MSB	LSB
	input	-			
	input	-			
	input	-			
	input	-			
	input	-			
	input	-			
	input	-			
	input	-			
	input	-			
	input	-			
	input	-			
	input	-			

- Click Finish

	×
🚱 🔤 New Source Wizard	
Summary	
_ Project Navigator will create a new skeleton source with the following specifications.	
Add to Project: Yes Source Directory: C:\temp\SevenSegments Source Type: Verilog Module Source Name: SevenSegModule.v	
Module name: SevenSegModule Port Definitions:	
More Info	:el

- The file will be created and it will look like as below

ISE	ISE Project Navigator (M.81d) - C:\temp\SevenS	ments\SevenSegments.	xise - [SevenSegModule.v]	
	Eile <u>E</u> dit <u>V</u> iew P <u>r</u> oject <u>S</u> ource <u>P</u> roces	<u>T</u> ools <u>W</u> indow Lay	iout <u>H</u> elp	- 8 ×
	o al 🗶 🗇 🖧 👹 🖓 📢 📢 📢	» 🏓 🏓 👧 🙉	🏓 🗟 🔀 🖻 🗗 🖻 🥬 🛠? 🕨 🗶 🗶 💡	
	yn ↔ □ ♂ × View: Wiew: Yi	2 ///// 3 // Com 4 // Eng 5 // 6 // Cre 7 // Des 8 // Mod 9 // Pro 10 // Tar 11 // Too 12 // Des 13 // 14 // Dep 15 // 16 // Rev 17 // Rev	ate Date: 19:10:47 02/13/2011 ign Name: ule Name: SevenSegModule ject Name: get Devices: l versions: cription: endencies: ision: ision 0.01 - File Created	
► III	No Processes Running Processes: SevenSegModule	19 // 20 //////	itional Comments: ////////////////////////////////////	
	Processes: Seversegwindule Design Summary/Reports Design Utilities User Constraints Synthesize - XST Implement Design Generate Programming File Configure Target Device Analyze Design Using ChipScope	22); 23 24 25 endmod 26		
150	Start 💷 Design 🐚 Files 🐚 Libraries	SevenS	egModule.v 🔀 🔀 Design Summary 🔀	•
	nings		ognodaorr 🖬 🔤 borgin banning y	↔□ਰ×
••••	III (JS			
1	Errors 🔔 Warnings			n 21 Col 23 Verkil66j

- Add your code

 \sim

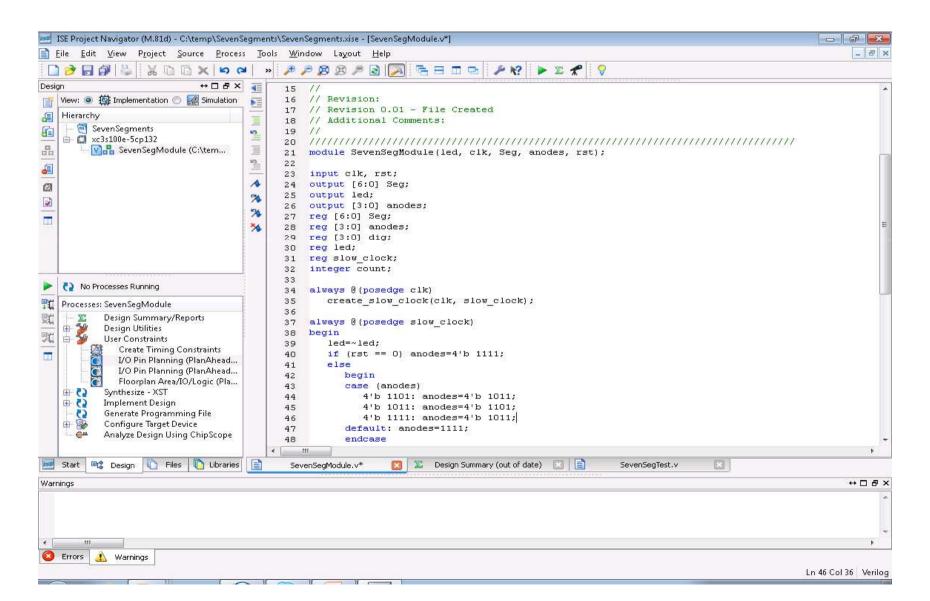
ISE	ISE Project Navigator (M.81d) - C:\temp\SevenS	Segments\SevenSegments.xise - [SevenSegModule.v]	- 6 .
B .	ile <u>E</u> dit <u>V</u> iew P <u>r</u> oject <u>S</u> ource <u>P</u> roces	ss <u>T</u> ools <u>W</u> indow Layout <u>H</u> elp	- 8 ×
	o 🕶 🗶 🚺 🖓 😓 🕷 🖬 🍯	🛤 » 🏓 🔊 🕼 🏓 💽 🦰 🗄 🗖 🖙 🥬 🛠? 🕨 🕱 🛠 💡	
	yn ↔ □ ₽ × View:	<pre>3 // Company: 4 // Engineer: 5 // 6 // Create Date: 15:44:31 02/13/2011 7 // Design Name: 8 // Module Name: sfaaa 9 // Project Name: 10 // Target Devices: 11 // Tool versions: 12 // Description: 13 // 14 // Dependencies: 15 // 16 // Revision: 17 // Revision 0.01 - File Created 18 // Additional Comments:</pre>	E
	No Processes Running		
- PC	Processes: SevenSegModule	21 22	
	 Design Summary/Reports Design Utilities User Constraints Synthesize - XST Implement Design Generate Programming File Configure Target Device Analyze Design Using ChipScope 	<pre>23 24 module SevenSegModule(led, clk, Seg, anodes, rst); 25 26 input clk, rst; 27 output [6:0] Seg; 28 output led; 29 output [3:0] anodes; 30 reg [6:0] Seg; 31 reg [3:0] anodes; 32 reg [3:0] dig; 33 reg led; 34 reg slow_clock; 35 integer count; 4</pre>	•
ISE	Start 🔍 Design 🚺 Files 陷 Libraries	SevenSegModule.v 🙁 Design Summary	
Erro	s		⇔⊡∄×
			*
•	III		¥.
8	Errors 🔬 Warnings	Ln	80 Col 10 Verilog

module SevenSegModule(led, clk, Seg, anodes, rst);

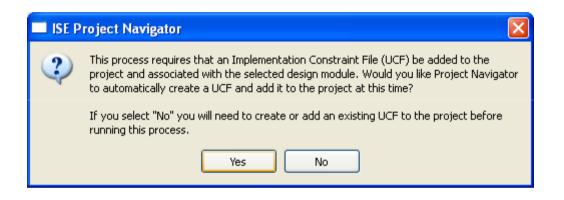
```
input clk, rst;
output [6:0] Seg;
output led;
output [3:0] anodes;
reg [6:0] Seg;
reg [3:0] anodes;
reg [3:0] dig;
reg led;
reg slow_clock;
integer count;
always @(posedge clk)
           create slow clock(clk, slow clock);
always @(posedge slow clock)
begin
led=~led;
if (rst == 0) anodes=4'b 1111;
else
           begin
           case (anodes)
                      4'b 1101: anodes=4'b 1011;
                      4'b 1011: anodes=4'b 1101;
```

```
4'b 1111: anodes=4'b 1011;
            default: anodes=1111;
            endcase
            case (anodes)
                        4'b 1011: dig=1;
                        4'b 1101: dig=2;
            endcase
                        case (dig)
                                    1: Seg = 7'b 1111001;
                                    2: Seg = 7'b 0100100;
                        endcase
end
end
task create_slow_clock;
            input clock;
            inout slow_clock;
            integer count;
            begin
            if (count > 250000)
            begin
                        count=0;
                        slow_clock = ~slow_clock;
            end
                        count = count+1;
            end
endtask
endmodule
```

-To implement the code on the fpga board go to implementation mode -- Select circuit.v file and select I/O planning (Planhead-Presynthesis) to open Planahead application to assign the fpga I/O ports to the code I/O



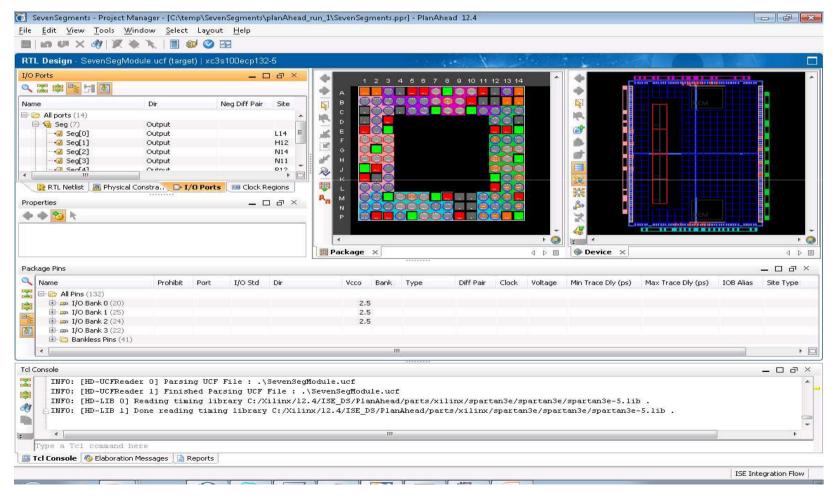
- Click yes to create UCF file and open Planahead application



-Planahead Application after its opened

-Select I/O ports, drag and drop each of them to Pin in the Package figure, refer to <u>http://www.digilentinc.com/Data/Products/BASYS2/Basys2 rm.pdf</u> to find the pin definition

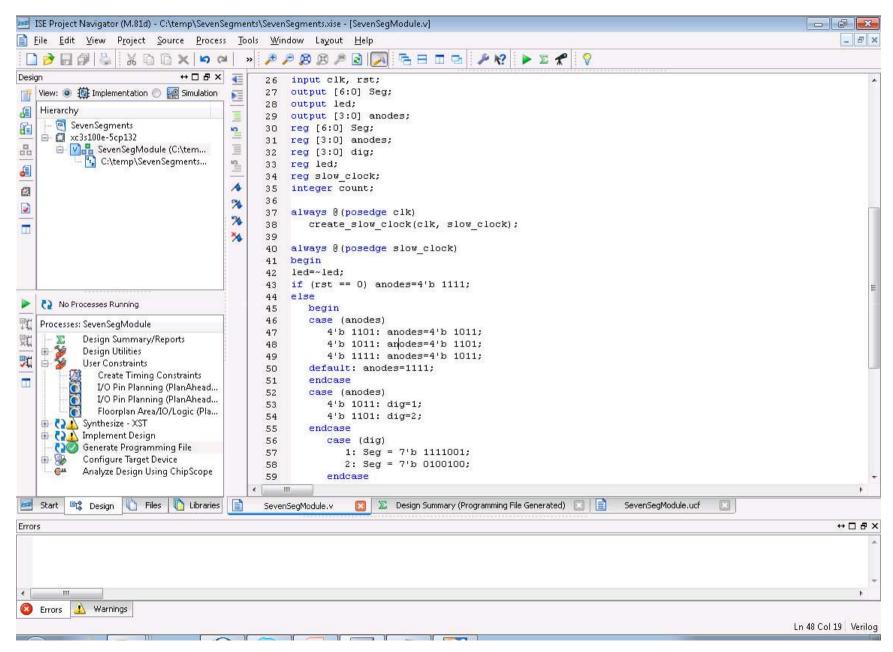
-- Click save design and exit



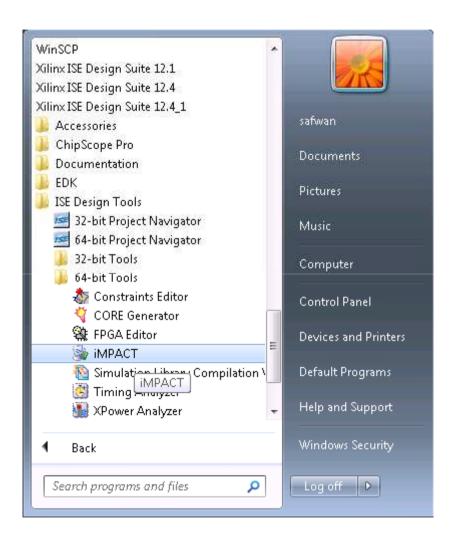
Create the *.bit file,* Double click on *Generate Programming File in the Processes* Window

🐖 ISE Project Navigator (M.81d) - C:\temp\SevenS	gments\SevenSegments.xise - [SevenSegModule.v]	
📄 <u>F</u> ile <u>E</u> dit <u>V</u> iew P <u>r</u> oject <u>S</u> ource <u>P</u> rocess	<u>T</u> ools <u>W</u> indow Layout <u>H</u> elp	_ 8 ×
○ ○ ○ ○ × ○ ○ × ○ ○	» 🔑 🔑 🛞 🔎 🗟 🏹 🖙 🗄 🗉 🕞 🤌 🛠? 🕨 🗵 🛠 💡	
Design ↔ □ ♂ × View: ● ☆ Implementation ● Simulation Hierarchy □ ☆ xc3s100e-5cp132 ● ☆ xc3s100e-5cp132 ● ☆ SevenSegModule (C:\tem ○ ☆ C:\temp\SevenSegments	<pre>24 module SevenSegModule(led, clk, Seg, anodes, rst); 25 26 input clk, rst; 27 output [6:0] Seg; 28 output led; 29 output [3:0] anodes; 30 reg [6:0] Seg; 31 reg [3:0] anodes; 32 reg [3:0] dig; 33 reg led; 34 reg slow_clock; 35 integer count; 36 37 always @(posedge clk) 38 create_slow_clock); 39 40 always @(posedge slow_clock)</pre>	
 Running: Generate Bitstream Processes: SevenSegModule Design Summary/Reports Design Utilities User Constraints U/O Pin Planning (PlanAhead I/O Pin Planning (PlanAhead I/O Pin Planning (PlanAhead Floorplan Area/IO/Logic (Pla Synthesize - XST Synthesize - XST Generate Programming File Configure Target Device Analyze Design Using ChipScope 	<pre>41 begin 42 led=~led; 43 if (rst == 0) anodes=4'b 1111; 44 else 45 begin 46 case (anodes) 47 4'b 1101: anodes=4'b 1011; 48 4'b 1011: anodes=4'b 1101; 49 4'b 1111: anodes=4'b 1011; 50 default: anodes=1111; 51 endcase 52 case (anodes) 53 4'b 1011: dig=1; 54 4'b 1101: dig=2; 55 endcase 56 case (dig) 57 1: Seg = 7'b 1111001; 4 10 11 10 11 11 11 11 11 11 11 11 11 11</pre>	
Start 🕫 Design 🚺 Files 🚺 Libraries	SevenSegModule.v 🛛 🖾 Design Summary (Implemented) 🛛 📄 SevenSegModule.ucf 🔍	
Errors	and a second and a second a se	↔□₽×
< III		
🙆 Errors 🔔 Warnings		Ln 42 Col 10 Veriloo

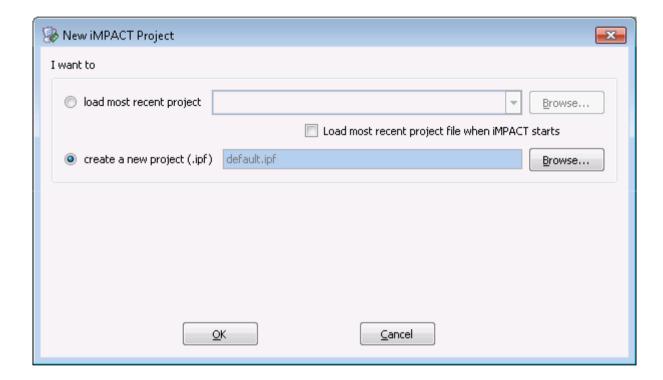
Make sure that every thing is compiling and building



Open iMPACT application from Xillinx ISE Deisgn Suite 12.4 tools



Create New Project



In the Welcome to iMPACT popup window, select Prepare a PROM File and click on OK

🛞 Welcome to iMPACT	×				
Please select an action from the list below					
Configure devices using Boundary-Scan (JTAG)					
Automatically connect to a cable and identify Boundary-Scan chain 📼					
Prepare a PROM File					
Prepare a System ACE File					
Prepare a Boundary-Scan File					
SVF 👻					
OK					

- Select Xilinx Flash/PROM

- Click on the green arrow

🛞 PROM File Formatter			×		
Step 1. Select Storage Target	Step 2, Add Storage Device(s)	Step 3,	Enter Data		
Storage Device Type : Xlinx Flash/PROM Spartan3AN Spartan3AN Configure Single FPGA Configure MultiBoot FPGA Configure Single FPGA Configure MultiBoot FPGA Configure MultiBoot FPGA Configure from Paralleled PROMs Generic Parallel PROM	Target FPGA Spartan3E Storage Device (bits): 512K Add Storage Device Remove Storage Device	General File Detail Checksum Fill FF Output File Name Untitled Output File Name C:\Xilinx\12.4 Output File Format C:\Xilinx\12.4 Flash/PROM File Property File Format Use Power-of-2 for Start Addr Number of Bitstream Bitstream 0 Start Address Bitstream 1 Start Address Add Non-Configuration Data File Number of Data File	Value BIN No 2 0 675840		
Description: The PROM File Formatter will guide you through the steps to format bitstream BIT files into a PROM file that is compatible with Xilinx® and third-party PROM programmers. The programmed PROM device can then be used to configure the target FPGA. Additional capabilities of the PROM File Formatter include: • Generating PROM files containing specific EPG0 configuration instructions required to support daisy-chained EPG0 bitstream BIT files OK Cancel					

-Select *Platform Flash under PROM Family* -Select *xcf02s under Device* -Click on *Add Storage Device* -Click on the green arrow

😵 PROM File Formatter			×
Step 1. Select Storage Targe	t Step 2. Add Storage Device(s)	Step 3,	Enter Data
Storage Device Type : 	PROM Family Platform Flash Device (bits) xcf02s Add Storage Device Remove Storage Device	General File Detail Checksum Fill Value Output File Name Output File	Value d Ix\12.4\ P BIN
	Auto Select PROM	Number of Bitstream Bitstream 0 Start Address Bitstream 1 Start Address Add Non-Configuration Da Number of Data File	2 0 675840 ta Files Yes
	bose the specific family you are targeting.		cremove it from the list

-Checksum Fill Value should be FF -Enter a filename & location -Select MCS under File Format -Select No under Add Data Files

🛞 PROM File Forma	otter								×
Step 1.	Select Storage Target		Step 2.	Add Si	torage Device(s)	I	Step 3,	Enter D	Data
BPI Flash Configure Sir Configure Mu	M A JultiBoot FPGA Ingle FPGA JultiBoot FPGA JultiBoot FPGA Jom Paralleled PROMs	•	PROM Family Device (bits) Add Storage Devi xcf02s [2 M]	ice R	atform Flash	•	General File Detail Checksum Fill Value Output File Name Output File Location Flash/PROM Fil File Format Add Non-Configura	nSegments Value MCS	
Description:									
In this step, you will • Checksum • Output File • Output File	enter information to assist in setting Fill Value: When data is insufficient Name: This allows you to specify the Location: This allows you to specif PROM files can be generated in an	to fill th ne base y the dir	e entire memory of a P name of the file to whic ectory in which the file	ROM, the ch your Pl e named a	e value specified here is u ROM data will be written ibove will be created	sed to ca	lculate the checksun		

An Add Device window will appear indicated that Xilinx will start adding the device

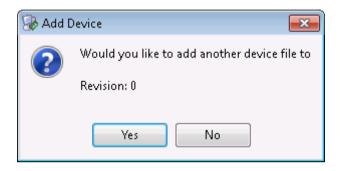
file to data stream 0, click ok



A popup window will appear Select your design (bit file) click Open

🖉 🖉 🖉 🚺 🦉 temp)	• SevenSegments • •	← Search SevenSegments	\$
Organize 👻 New fol	der	8== ┯	1 0
🗐 Recent Places 🦯	Name	Date modified	Туре
	📔 🛄 _ngo	2/13/2011 7:50 PM	File fol
	\mu _xmsgs	2/13/2011 7:50 PM	File fol
🕌 LineBased 👘	🔋 🔋 ipcore_dir	2/13/2011 7:44 PM	File fol
-	🔋 🔋 iseconfig	2/13/2011 7:44 PM	File fo
词 Libraries	🌗 planAhead_run_1	2/13/2011 7:46 PM	File fo
Documents	🎳 xinx_auto_0_xdb	2/13/2011 7:50 PM	File fol
🎝 Music	\mu xst	2/13/2011 7:50 PM	File fol
📔 Pictures 🛃 Videos	sevensegmodule.bit	2/13/2011 7:50 PM	BIT Fil
Computer Solution OS (C:) RECOVERY (D:)	4		
File	<u>n</u> ame: sevensegmodule.bit	▼ FPGA Bit Files (*.bit)	-

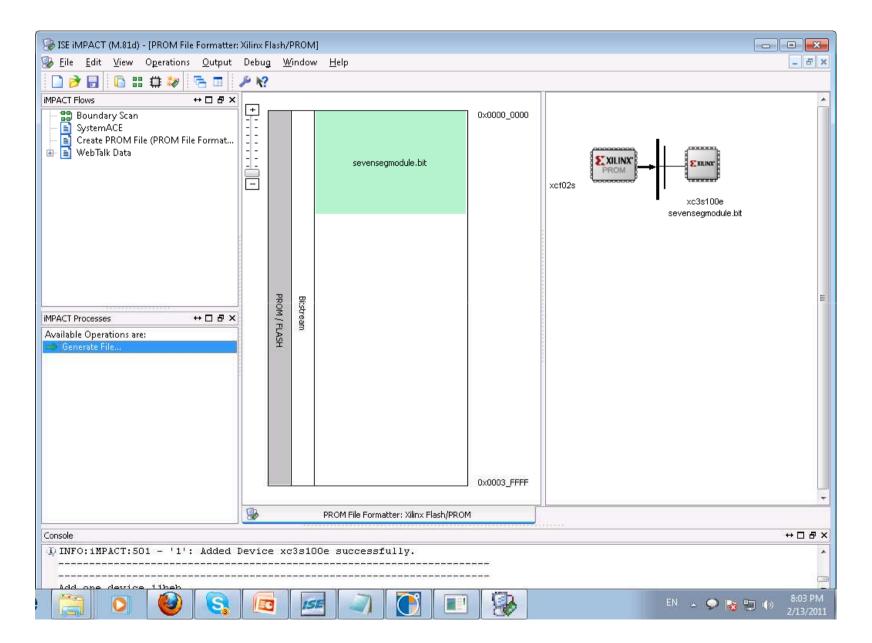
You'll be asked, "Would you like to add another design file to Data Stream:0?", click **NO**



You'll be informed that you've completed the device file entry, click OK



In the Processes window, double click on Generate File



If all goes well, a blue success message will be displayed

	Vit EL	1 mpot	0		1	
ISE iMPACT (M.81d) - [PROM File Formatter:						
Eile Edit View Operations Output	Debug	windov	w <u>H</u> eip			- 8 ×
🗋 🌶 🗐 🐲 📑 🖬 🥕 K?						
IMPACT Flows ↔ □	E C					<u>^</u>
Boundary Scan SystemACE Create PROM File (PROM File Format WebTalk Data MPACT Processes ↔ □ 중 × Available Operations are: Generate File		Bitstream PROM / FLASH	sevensegmodule.bit	0x0000_0000	xcf02s	H
				0x0003_FFFF	Generate Succeeded	
			PROM File Formatter: Xilinx Flash/PRO	м		
Console	-	(***			*****	↔□₫×
Writing file "C:\temp\SevenSe	aments	\Untit	led.sia".			A
Writing file "C:\temp\SevenSe // *** BATCH CMD : setCurrent	gments	\Untit.	led.cfi".			
- 📋 💽 🔮 💽		15	z 🤍 🦳 💽		en 💊 🗭 🎦	() 8:03 PM 2/13/2011

- Open Adept2.1 (Downloaded from Digilent's Website)
- -Click on the Config tab
- -Select the .mcs file for the PROM (XCF02S) by clicking on the Browse button
- Click Program
- -Wait until the Program successfully loaded
- -Turn the board on and off, then run the program

	🛆 Digilent Adept
	BASYS2 Connect: Basys2 Product: Basys2 - 100
	Config Test Register I/O File I/O I/O Ex Settings
1	FPGA Circuit.bit Browse Program
	PROM SevenSegments.mcs Browse Program
	 Initialize Chain
	Set Config file for XCF025: "C:\Documents and Settings\Administrator\Desktop\SevenSegments.mcs" Preparing to program XCF025
	Erasing device Programming Successful.
	Set Config file for XC35100E: "C:\Documents and Settings\circuit\circuit.bit"

Example 2:

- A 0, 1, 2, or 3 is displayed on the seven-segment display, depending upon whether button #0, #1, #2, or #3 is pressed
- Enable
 - Switch #0

module ckt(btn, clk, a, b, c, d, e, f, g, an, rst); input [3:0] btn; input clk, rst; output a, b, c, d, e, f, g; output [3:0] an; reg a, b, c, d, e, f, g; reg [2:0] cstate, nstate; reg [3:0] an; always @(posedge clk or negedge rst) begin if (~rst) cstate<=7; else cstate<=nstate;</pre> an=14; end always @(btn or cstate) case (btn) 4'b1000: nstate=3; // Button 3 pressed 4'b0100: nstate=2; // Button 2 pressed 4'b0010: nstate=1; // Button 1 pressed 4'b0001: nstate=0; // Button 0 pressed 4'b0000: nstate=cstate; // No button pressed default: nstate=7; // No button pressed yet or multiple // buttons pressed endcase

```
always @(posedge clk)
case (cstate)
3: begin // Button 3 pressed
a=0; b=0; c=0; d=0; e=1; f=1; g=0;
end
2: begin // Button 2 pressed
a=0; b=0; c=1; d=0; e=0; f=1; g=0;
end
1: begin // Button 1 pressed
a=1; b=0; c=0; d=1; e=1; f=1; g=1;
end
0: begin // Button 0 pressed
a=0; b=0; c=0; d=0; e=0; f=0; g=1;
end
7: begin // No button pressed yet or multiple
buttons pressed
a=1; b=1; c=1; d=1; e=1; f=1; g=1;
end
endcase
endmodule
```

References

- Michael D. Ciletti, Advanced Digital Design with the Verilog HDL, Pearson Education, Inc.
- (Prentice Hall), 2003
- Donald E. Thomas and Philip R. Moorby, *The Verilog Hardware Description Language*,
- Kluwer Academic Publishers, 1998
- Samir Palnitkar, Verilog HDL A Guide to Digital Design and Synthesis, Prentice Hall, Inc., 4th
- Edition, 1996
- David R. Smith and Paul D. Franzon, Verilog Styles of Digital Systems, Prentice Hall, Inc.,
- 2000
- Digilent Basys 2 Board Reference Manual, Digilent, Inc., May 25, 2009
- Digilent BASYS 2 System Board Schematics, Digilent, Inc., December 12, 2008