Getting Started with Xilinx ISE

- Start the Xilinx ISE
- Create a new Project by clicking on 'New Project' (on the top left window panel)

Lincer a manne, rocat	ions, and comment for the project
N <u>a</u> me:	SampleProject
<u>L</u> ocation:	ome/suyash/Xilinx_ISE_DS_Lin_14.7_1015_1/SampleProject
<u>W</u> orking Directory: <u>D</u> escription:	ome/suyash/Xilinx_ISE_DS_Lin_14.7_1015_1/SampleProject
	op-level source for the project

• Enter the project details as shown here

ISE Project Navigator (P.20131013) - F:\acads\Sem	ester_4\cs226_lab\fpga\fpga.	xise - [dla.ucf]		- 0 ×
File Edit View Project Source Process	Tools Window Layout	Help		- 8 ×
う x a a X a a X a a X a a	» / / / 8 8 /	2 🔊 🗟 🗆 🖉 🗡	k? 🕨 🕱 🛠 💡	
Design ← □ ♂ × Vew: ◎ ∰ Implementation ○ ∰ Simulation Hierarchy e → cxctrk45-3csg324 a → cxctrk45-3csg324	> >	Coc="A10"; coc="A10"; coc="A10"; coc="A10"; coc="A10"; Design Properties Name: Location: Fluce Working directory: Project Settings	Af Y X X V ads/Semester_4/xx226_Jab/fpga ads/Semester_4/xx226_Jab/fpga	
 × No Processes Running No single design module is selected. ₩ ₩ Design Utilities ₩ <td></td><td>Property Name Top-Level Source Type Evaluation Development Board Product Category Family Device Package Speed Synthesis Tool Simulator</td><td>Value A HDL V Inne Specified V Spartan6 V XC6SLX45 V CS0224 V -3 V XST (VHDL/Verilog) V</td><td></td>		Property Name Top-Level Source Type Evaluation Development Board Product Category Family Device Package Speed Synthesis Tool Simulator	Value A HDL V Inne Specified V Spartan6 V XC6SLX45 V CS0224 V -3 V XST (VHDL/Verilog) V	
> Start 🔍 Design 🚺 Files 🚺 Libraries	and xor.vhd 🛛 🔀	Preferred Language	VHDL	rt Search Results: "MapLib:30" 🔟 📄 dla.ucf 🕅
Errors		Dronerty Specification in Droiect File	Store all valuer OK Cancel Help	+
<	d in Files Results			Ln1Col1 UCF
Search the web and Windows	Ø) 🗎 🧕 🗧	S > A	∧ 💫 🗈 ⊄× 🦟 📮 ENG 13:01 US 21-01-2017

And thus finish the creation of the project.

• Add a new source, VHDL module to your project and enter details of the input/output variables. Also note how a vector can be declared here itself.

File Edit View Project Source	rocess <u>T</u> ools <u>W</u> indow La <u>v</u> out <u>H</u> elp	
🗋 ờ 🗐 🎒 🖏 🕹 🕹	×ν×××××××××××××××××××××××××××××××××××	
Jesign 💮 💮 View 🖲 🔯 Implemental 🔿 📾	0 X India	
Hierarchy		
SampleProject		
* * xcosix45t-5igg484	New Source	
in .	Add Source	
8	Add Copy of Source	
*	Manual Compile Order	
No Processes Running	Implement Top Module	
No single design module is s	File/Path Display	
	Expand All	
71	Collapse All	
* Start R* Dosign C Files	Eind Ctrl+F	
Frrors	Design Properties	-
		19 00 00 0 [-
		-
Console 🔕 Errors 🔬 Wa	ings 🙀 Find in Files Results	
Add a new source to the project		
W I New Source Wizard		
Select Source Type		
Select source type, file har	and its location.	
	itochura Wiza	
Schematic	incertaine witza	
User Document		
Verilog Module	Ele name:	
VHDL Module		
VHDL Library	Main	
VHDL Test Bench	Location:	
Embedded Processor	E_DS_Lin_14.7_1015_1/SampleProject	
•	Add to project	
Neck.		
More Info	Next > Cancel	
Rew Source Wizard		
Specify ports for module		
specify ports for module.		
Entity name Main		
Architecture name Behavio	ıl	
Port Name	Direction Bus MSB LSB	
а	in 🗘 🗹 3 0	
b	in 🗢	
c	out 🗘	
d	inout 🗢 🗆	
	in 🗘	
	in 🗘	
	in 🗘	
1		
More Info	< <u>Back</u> Next > Cancel	
and the second se		

Thus finish the creation of this module.

• Then you can write the code you need in the architectural section as shown below:



- Save your code. Synthesize, which should then change to a green tick if all goes right. If not, go through the errors and warnings dumped on the console and debug.
- Then add a testbench for testing and simulation of your program by right clicking on the project and then 'New Source' as we did in step1 while adding VHDL module

elect Source Type Select source type, file name and its locatio	on.		
BMM File ChipScope Definition and Connection File Implementation Constraints File IP (CORE Generator & Architecture Wizar MEM File Concentration	<u>File name:</u>		
User Document	TestBench		
Verilog Module	Location:		
Venlog lest Hxture	5E DS Lin 14.7 1015 1/SampleProject		
NHDL Library VHDL Package			
When the second seco			
	✓ Add to project		

Then click on Next -> Next -> Finish to see the testbench added to your project.

• If you are don't need clock in your simulation, you can remove it from the testbench. Comment the clock_process entirely out of the code as shown below (shortcut: Alt+C)

```
-- constant <clock>_period = 2*10ns
```

```
-- <clock>_process :process
```

- -- begin
- --- <clock> <= '0';
- -- wait for <clock>_period/2;
- -- <clock> <= '1';
- -- wait for <clock>_period/2;
- -- end process;

Else use this clock to give waits.

• Visualize the testbench as a script to give various inputs to the program with timed waits. Consider the example given below:

File Edit View Project Source Process	i <u>T</u> ools <u>W</u> indow La <u>x</u> out <u>H</u> elp	
1 🗋 🏓 🖬 🎒 😓 1% 🗅 🗅 🗙 🕨	n » /# /# /# /# /# = = /# /# = x x ! ?	
Obsign Control (Control (Contro) (Control (Control (Control (Contro) (Control (Contro) (Co	<pre>9 *** ** *** *************************</pre>	
Processes: TestBench - behavior 3 Image: Sim Simulator 3 Consultation: 3 Simulate Behavioral 3 Simulate Behavioral 3 Simulate Behavioral 3 Start 82 Design Files €	<pre>9 96 97 insert stimulus here 98 100 end process; 101 101 102 103 103 104 104 105 105 105 106 107 107 107 107 107 107 107 107 107 107</pre>	
Errors		+ D Ø X
Console O Errors J. Warnings 💰	Find in Files Results	e D
		Ln 95 Col 13 VHDL

• Click on 'Behavioral Check Syntax'. It should change to a green tick if all goes right. If not, go through the errors and warnings dumped on the console and debug.

Elle Edit View Simulation Window Layout Help		
1 🔊 🖬 🕹 💥 🖻 🖻 🗙 😕 🛤 😹 🗍 1	0 5 8 8 9 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	🚦 👘 🐴 🖬 🕨 🗚 1.00us 🕶 🔩 II 🗔 Re-launch
Instances and Processes () () Objects () () () () () () () () () () () () ()	0 0	1000 000 ps ↑ 1000 000 ps ↑
	X1: 1,000,000 ps	
AInstan 🚡 Memory 📔 🕩 🕢 🔹	Default.wcfg 🗶	
Console		e () Ø 🛛
WARNING: ISim will run in Lite mode. Please refer to the ISim documen This is a Lite version of ISim. Time resolution is 1 ps Simulator is doing circuit initialization process. Finished circuit initialization process. ISim>	tation for more information on the differences between the Lite and t	ne Full version.
Console Compilation Log Breakpoints Rid in Files	Results 🔚 Search Results	Sim Time: 1,000,000 ps

Playing around with inputs

• You can also add signals and variables to waveform viewer window as shown Click on the main instance on left side of screen. Expand it to be able to see the objects



Right click on the variable/signal you wish to see in waveform. Click on 'Add to Wave Window' option



😣 🚍 💿 🛛 ISim (P.20131013) – [Default.wcf	g*]			¥ 🕲 🛆 年	. En 🖂 🗉	(0:12, 38%) 🖪	× Wed Jan 27	11:51 PM 🔱
File Edit View Simulation Window Layo	out <u>H</u> elp							-0x
🗋 🆻 🖶 🖕 💥 🖻 🖺 🗙 🕲 ピ) ∩ (A (A ⊨) (0 5 8 0 9 <i>P</i> K?	P P B P	🗟 🗠 🛨 肯	1° 1 🖸	▶ ▶ 1.00u	s 🕶 归 🛛	Re-launch
Instances and Processes Instances and Processes Instance and Process Name Instance and Process Name Design Unit V main main(behav V std_logic_1164 std_logic_1 V	Chi (A) (A) Chi (A) Objects Image: Chi (A) Simulation Objects f Image: Chi (A) Object Name Va Image: Chi (A) Image: Chi (A) <th>Image: Second second</th> <th>20 20 20 20 20 20 20 20 20 20 20 20 20 2</th> <th>3 12 2 1 999,996 ps</th> <th></th> <th>999,998 ps</th> <th>999,999 ps</th> <th></th>	Image: Second	20 20 20 20 20 20 20 20 20 20 20 20 20 2	3 12 2 1 999,996 ps		999,998 ps	999,999 ps	
		•	(▲)					
👗 Instances and Processes 📔 Source Files		AX Default	.wctg*	×				
Compilation Log								
Console Compilation Log Break	points 🛛 🕅 Find in Files Re	esults Search Results					Sim Tim	e: 1,000,000 ps
-								

You can see your variable/signal added to the waveform.

You can also force values to the input variables directly in the simulation as shown below



See Sim (P.20131013)	🗱 🕞 🏝 🕇 🖬 🖂 🗩 (1:	07, 83%) 🜒 Wed Jan 27 11:42 PM 🔱
		x 1.00us y k≡ II □ Re-launch
Instances and Processes O a Collects		1,000,000 ps
🔋 😮 📑 📢 📑 🐘 Simulation Objects for main		
Instance and Process Name C	Name Value 999,995 ps 999,996 ps 999,997 ps 999,997 ps	38 ps 999,999 ps 1,000,000 ps 1
▶ main n bject Name Value		
Std_logic_1164 51 6 a[3:0] 0000	🧖 🛞 🗉 Force Selected Signal	
La c U	Enter parameters below to force the signal to a	
d U	Constant value. Assignments made from within HDL code or any previously applied constant or	
	clock force will be overridden.	
	Signal Name: /main/a	
	Value Radix Binary 🗢	
	Force to Value: 0001	
	I ^{de} Starting at Time Offset: 0	
	Cancel after Time Offset:	
	<u>Ok</u> Cancel <u>Apply</u> <u>H</u> elp	
	X1: 1,000,000 ps	
AInstan 🗟 Memory 📔 🕩 🕢 👘	Default.wcfg	
Console		+
WARNING: ISim will run in Lite mode. Please refer to the ISim document	ation for more information on the differences between the Lite and the Full version.	
Time resolution is 1 ps		
Simulator is doing circuit initialization process.		=
Isim>		-
🔳 Console 📄 Compilation Log 🗕 Breakpoints 🕅 Kind in Files	Results 🙀 Search Results	
Force the selected signal to a constant value		Sim Time: 1,000,000 ps
		01 00% W U 2711 42 DU M
⊗ ⊜ © ISim (P.20131013) - [Default.wcfg] ■ Ele Edit View Simulation Window Lavout Help	🗱 🕞 🍐 🗤 🖬 🖾 📼 (ki	01, 80%) 🜒 Wed Jan 27 11:43 PM 🛟
See Sim (P.20131013) - [Default.wcfg] ☐ File Edit Yiew Simulation Window Layout Help D → D P V D D V O D A O D A O D C V C D C V C D C V C D C V C C V C C V C C V C C V C C V C C V C C V		01, 80%) (1)) Wed Jan 27 11:43 PM 🛟
See Sim (P.20131013) - [Default.wcfg] Ele Edit View Simulation Window Layout Help Compared Barrow See Some Simulation Objects Default See Some Simulation See Some Simulation See Some Simulation Sector S	 ★ ⊕ ▲ 1; h ∞ ■)(: 0 5 8 0 0 / K? > 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	01, 80%) ◀)) Wed Jan 27 11:43 PM 🛟 □@@ ↓X 1.00us ♥ ♀ ♀ Re-launch 1.0000000005
Isim (P.20131013) - [Default.wcfg] Image: File Edit View Simulation Window Layout Help Image: File Edit View Simulation Window Cayout Help Image: File Edit View Simulation View Cayout Help Imag		01, 80%) ◀)) Wed Jan 27 11:43 PM 🛟 ⓒ 🕅 🖉 , X 1.00us ♥ ५ II 🔽 Re-launch 1.000.000 ps
Isim (P.20131013) - [Default.wcfg] Image: Simulation Window Layout Help Image: Simulation Help Image: Simulation Help Image: Simulation Help Image: Simulatin Help Image: Simulati	Image: Second system Image: Second system <t< th=""><th>01, 80%) 4)) Wed Jan 27 11:43 PM</th></t<>	01, 80%) 4)) Wed Jan 27 11:43 PM
Isim (P.20131013) - [Default.wcfg] Instance and Processes Instance and Process Name (Instance and Process Name (Insta	Image: Second secon	01, 80%) 4)) Wed Jan 27 11:43 PM
Isim (P.20131013) - [Default.wcfg] Instances and Processes Isim (P.20131013) - [Default.wcfg] Instance and Processes Instance and Process Name (Instance and		01, 80%) 4)) Wed Jan 27 11:43 PM
Isim (P.20131013) - [Default.wcfg] Instances and Processes Instance and Process Name (Instance and Instance Instance (Instance Instance Instance (Instance Instance Instance (Instance Instance Instance (Instance Instance (Instance Instance (Instance (Instance Instance (Instance		01, 80%) 4 ()) Wed Jan 27 11:43 PM
Isim (P.20131013) - [Default.wcfg] Isim (P.20131013) - [Default.wcfg] Isim (P.20131013) - [Default.wcfg] Isim (P.20131013) - [Default.wcfg] Instances and Processes Instance and Processes Instance and Process Name Instance and Procese Name Inst		01, 80%) 4 ()) Wed Jan 27 11:43 PM ↓ ©@ ⊠ pX 1.00us ♥ ↓ II
Isim (P.20131013) - [Default.wcfg] File Edit View Simulation Window Layout Help Instances and Processes Instance and Processes Instance and Process Name Instance of the proceseeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee	 ♣ 1,	01, 80%) ◀0) Wed Jan 27 11:43 PM ↓ □@® pX 1.00us ♥ € II @Re-launch 1.000.000 ps 1 1.000.000 ps 1
Isim (P.20131013) - [Default.wcfg] Image: State of the st	 ♣ 1,	01, 80%) ◀)) Wed Jan 27 11:43 PM ↔ □@@ pX 1.00us ♥ ← II @Re-launch 1.000.000 ps 1 38 ps 1999.999 ps 1.000.000 ps 1
Isim (P.20131013) - [Default.wcfg] Image: State of the st	★ ① ▲ 1↓ 函 ⊠ ■ (1: ③ ● 日 □ ○ ▶ 12 ▶ 2 ▶ 2 ▶ 2 ▶ 2 ■ 12 1 ↑ 1 1 ↓ Ø ● □ □ ○ ▶ 12 ▶ 12 1 ↑ 1 1 ↓ Ø ● □ □ ○ ▶ 12 ▶ 12 1 ↑ 1 1 ↓ Ø ● □ □ ○ ▶ 12 ▶ 12 1 ↑ 1 1 ↓ Ø ● □ □ ○ □ □ □ □ 0001 0001 0001 Ø ● □ □ □ □ □ □ 0001 □ □ 0001 □ □ Ø ● □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ Ø ● □ □ □ □ □ □ □ □ □ □ □ □ □ □ Ø ■ □ □ □ □ □ □ □ □ □ □ □ □ □ Ø ■ □ □ □ □ □ □ □ □ □ □ □ □ □ □ Ø □	01, 80%) ◀)) Wed Jan 27 11:43 PM () □@@ yX 1.00us ♥ €2 II C Re-launch 1.000.000 ps 1 1.000.000 ps 1 0.000.000 ps 1 0.0000.0000 ps 1 0.000.0000 ps 1 0.000.0000 ps 1 0.000.0000 ps 1
Simulation Dijects minor state and Process Name state and Process Name stat	Image: Second state	01, 80%) ◀)) Wed Jan 27 11:43 PM ↓ □@® J ^X 1.00us ♥ ५ II CR-launch 1.000.000 ps 1 1.000.000 ps 1 1.000.000 ps 1
Isim (P.20131013) - [Default.wcfg] Image: State of the st	Image: Second secon	01, 80%) ◀)) Wed Jan 27 11:43 PM
Isim (P.20131013) - [Default.wcfg] Image: State of the st	Image: Second secon	01, 80%) ◀)) Wed Jan 27 11:43 PM
Isim (P.20131013) - [Default.wcfg] Image: State of the st	★ ① ▲ 1 ⊆ № (1) Image: Section of the sec	01, 80%) ◀1)) Wed Jan 27 11:43 PM
Isim (P.20131013) - [Default.wcfg] Image: State of the st	Image: Second secon	01, 80%) ◀1)) Wed Jan 27 11:43 PM
Isim (P.20131013) - [Default.wcfg] Image: State of the st	Image: Section of the section of t	01, 80%) ◀)) Wed Jan 27 11:43 PM
Isim (P.20131013) - [Default.wcfg] Isim (P.20131013) - [Default.wcfg] Isim (P.20131013) - [Default.wcfg] Instances and Processes Instance and Processes Isimulation Dijects for main Instance and Process Name Instance and Process Name Instance and Process Name Isimulation Dijects for main Isimulation Diject Name Value Isimulation Diject Name Isimulation D	Image: Second secon	01, 80%) ◀1)) Wed Jan 27 11:43 PM
Isim (P.20131013) - [Default.wcfg] File Edit View Simulation Window Layout Help Instances and Processes Instance and Processes Instance and Process Name Instan Image: Instan Image: Instan	Image: Second secon	O1, 80%) ◀0) Wed Jan 27 11:43 PM ↓ □@@ pX 1.00us ♥ € II @Re-launch 1.000.000 ps 1.000.000 ps 1.000
Isim (P.20131013) - [Default.wcfg] File Edit View Simulation Window Layout Help Instances and Processes Instance and Processes Instance and Process Name Instan Instance and Process Name Instan Instance and Process Name <t< th=""><td>A the End Control of the con</td><td>01, 80%) ◀)) Wed Jan 27 11:43 PM ↔ □@@ p^X 1.00us ♥ ♀ II @Re-launch 1.000.000 ps 1.000.000 ps 1.000.000 ps III ♥ ♥</td></t<>	A the End Control of the con	01, 80%) ◀)) Wed Jan 27 11:43 PM ↔ □@@ p ^X 1.00us ♥ ♀ II @Re-launch 1.000.000 ps 1.000.000 ps 1.000.000 ps III ♥ ♥
File Edit View Şimulation Window Layout Help Instances and Processes Objects Instance and Process Name Simulation Objects for main Instance and Process Name Instance and Process Name Instance and Process Name Instance Tr Instance and Process Name Instance Tr Instance and Process Name Instance Tr Instance Instance Instance Instance Tr Instance Instance Instance Instance Tr Instance Instance Instance Instance Tr	A 1	O1, 80%) ◀)) Wed Jan 27 11:43 PM ↔ □@@ p ^X 1.00us ♥ ♀ II
Simulation Window Layout Help File Edit View Simulation Window Layout Help Instances and Processes Instance and Process Name Instance and Process Name Instance and Process Name Instance in the process Name Instan Instance in the process Name Instan Instan Instan Inst	A 1	01, 80%) ◀)) Wed Jan 27 11:43 PM () □@@ pX 1.00us ♥ €2 II @Re-launch 1.000.000 ps 1.000.000 ps 1.000.0000 ps 1.000.000 ps 1.0000.000 ps 1.000.0000
Simulation (P.20131013) - [Default.wcfg] File Edit View Simulation Window Layout Help Instances and Processes Objects Image: Simulation Objects for main Instance and Process Name Simulation Objects for main Instance and Process Name Simulation Object Name Value Value	A 1	01, 80%) ◀)) Wed Jan 27 11:43 PM ↓ □@@ p ^X 1.00us ♥ ♀ II
Simulation (P.20131013) - [Default.wcfg] File Edit View Simulation Window Layout Help Instances and Processes Instance and Process Name Instance and Process Insta	() (1) (1) (1) (1) (1) (1) (1) (1) (01, 80%) (1)) Wed Jan 27 11:43 PM (1) PX 1.00us (2) (2) (1) (2) Re-launch 18 ps 1999,999 ps 1.000.000 ps 1 1.000.000 ps 1 (1) (2) (2) (2) (2) (2) (2) (2) (2) (2) (2
Simulation (P.20131013) - [Default.wcfg] File Edit View Simulation Window Layout Help Instances and Processes Instance and Processes Instance and Process Name Instan	Image: Search Results	01, 80%) (1)) Wed Jan 27 11:43 PM (1) P (2) (2) (2) (2) (2) (2) (2) (2) (2) (2)

You can see the value '0001' been forced on to the variable a.

Modularity: Multiple files in same project

Often you would feel the need to modularize your code by putting different functional entities into different files. These files can be added the same way you add vhdl codes and testbenches in your project. To link them however, you would need to link their ports via what is called port mapping. Below is an example where the entity *arthur* is composed of internal VHDL modules *barney* and *elmo*. We define the components first with their port types and then map the ports to those *arthur*.

```
LIBRARY ieee;
USE ieee.std logic 1164.ALL;
ENTITY arthur IS
      PORT ( clk : IN STD LOGIC; --clock input
            enbl : IN STD LOGIC; --enable for arthur
            adv : IN STD LOGIC; --advance the state
            pulse : OUT STD LOGIC; --pulse the dog collar
            open door : OUT STD LOGIC -- output to door driver
            );
END arthur;
ARCHITECTURE struct OF arthur IS
      SIGNAL ask me : STD LOGIC; --barney asks elmo
      SIGNAL clear : STD LOGIC; --elmo clears request to ask
      COMPONENT barney PORT (
                  clk : IN STD LOGIC;
                  enable : IN STD LOGIC;
                  adv : IN STD LOGIC;
                  clear : IN STD LOGIC;
                  outa : OUT STD LOGIC;
                  ask me : OUT STD LOGIC
                  );
      END COMPONENT;
      COMPONENT elmo PORT(
                  ask me : IN STD LOGIC;
                  go : OUT STD LOGIC;
                  clear : OUT STD LOGIC
                  );
      END COMPONENT;
      BEGIN
      U1 : barney PORT MAP(
                  clk => clk,
                  enable => enbl,
```

```
adv => adv,
clear => clear,
outa => pulse,
ask_me => ask_me
);
U2 : elmo PORT MAP(
ask_me => ask_me,
go => open_door,
clear => clear
);
END struct;
```

Forcing Signal via ISim

Right click on the variable you want to change -> select Force Clock/Force Constant option -> Fill the required details -> click OK -> Refresh.

Force to Value : 1 Starting at Time Offset : 10ns

Cancel after Time Offset : 30ns

	ISim (P.20131013) - [Default.wcf]] Iou Levout Holp				🗱 🔿 En 🜒 Jan 27 11:40:31 F	PM (C):
0		S CH [AS [] [] [] [] [] [] [] [] [] [8 7 8 2 2 3	🗠 🐴 🖸 🕨 🗚 1.00us 💌 🤇	II 🗔 Re-launch	
	Instances and Processes ↔ □ ₽ ×	Objects	↔□∂× 🍠		854.485 ns		*
		Simulation Objects for test_b	pench	alue (820 ns	1840 ns 1860 ns	1880 ns 1900 ns	920
	Instance and Process Name E	hiert Name Value					
9	std_logic_1164 s	a O	C L clock_peric 1	0000 ps	10000	ps	
		lie b 0 Le clock_period 10000 pr	s 🎼 Add To Wave Window Ctrl	•₩			
			Ctrl	+C			
			Select All Ctrl	+A			
			Collapse				
			🕞 Search				
			Show All Elements				
>_			Badix	+			
			Select in Wave Window				
2			Show Drivers Force Constant				
			Force <u>C</u> lock	×1: 854 485 ns			
	<u> </u>		Remove Force				•
	🔒 Instan 🗎 Memory 📔 S 🕧 🕨	•	Go To Source Code	ult.wcfg			
	Console	conigatorion managento and				•	+ □ ₽ ×
	WARNING: ISim will run in Lite mode. F This is a Lite version of ISim.	lease refer to the ISim docum	nentation for more information on the diffe	rences between the Lite and	the Full version.		-
	Time resolution is 1 ps Simulator is doing circuit initialization pr	ocess.					
	Finished circuit initialization process.						_
	🔳 Console 🔲 Compilation Log	🔹 Breakpoints 🕅 Find	in Files Results 🛛 🖬 Search Results				
	Force the selected signal to an alternation	ing pattern (clock)				Sim Time: 1,00	00,000 ps
000	Icim (000121012) [Dofaultaural	-1				💓 👝 🗖 dul upo 07.11/41/05/5	74 11
	Eile Edit View Simulation Wind	low La <u>v</u> out <u>H</u> elp					
O		1 A& Co. 11 (2)					
	The second)∐≒⊟⊡≏∐ <i>₽№</i> ∐҂҂	873121	rh 🐴 🖸 🕨 🗚 1.00us 💌 🤇	II GRe-launch	
	Instances and Processes ↔ □ 5 ×	Objects	+= = = = > R? > >	8721144111	i ^t r → C ► ► ► T.00us ► ←	II Re-launch	×
	Instances and Processes ↔ □ 5 ×	Objects Simulation Objects for test_b	+ □ 5 × 5 pench 8 Name V	8 기 로 1 년 20 1 1 alue	th → I 1.00us y 4 854.485 ns 1840 ns 1860 ns	E II Re-launch	≥ 1920
	Instances and Processes +	Objects Simulation Objects for test_b	+ □ # × # pench	8 가 오 말 같 : alue	A → C → X 1.00us → C 854.485 ns 1840 ns 1840 ns	II CR Re-launch 11 CR Re-launch 100 ns 1900 ns	920
	Instances and Processes $\leftrightarrow \square \emptyset \times$	Objects Simulation Objects for test_b bject Name Value	+	alue 1820 ns	840 ns 10000 10000 10000 10000 10000 10000 10000 10000 10000 10000 10000 10000 1000000	B80 ns 900 ns 1000 ns 900 ns 1000 ns 1000 ns	920
	Instances and Processes ↔ □ ₹ X Instance and Process Name □ Mathematical Structure State Stat	Objects Simulation Objects for test_t bject Name Value C b 0 b clock_period 1000C	→ □ ∂ × → □ ∂ × □ ∂	alue	2 1 00us ↓ 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	B80 ns 900 ns 1000 ns 900 ns 1000 ns 1000 ns 1000 ns 1000 ns	920
	Instances and Processes ↔ □ ♥ × Instance and Process Name □ Instance and Process Name □ ■ ■ tes_bench te ⓒ std_logic_1164 s	Objects Simulation Objects for test L bject Name Value Cock_period 1000C	Define Clock	alue 1820 ns.	1 00us y 654485 ns 1840 ns 1860 ns 10000 10000	Re-launch (880 ns	1920
	Instances and Processes ↔ □ ♥ × Instance and Process Name □ Instance and Process Name □ ■ ■ test_bench te	Objects Simulation Objects for test L bject Name Value Cock_period 1000C	Define Clock Define	alue	2 1 00us ↓ 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	Re-launch (880 ns	920
	Instances and Processes ↔ □ ♥ ×	Objects Simulation Objects for test E bject Name Value b Object Name b Object Name b Object Name b Object Name b Object Name b Object Name b Objects for test E b Objects for tes	Define Clock Define	A A Constant of the second sec	2 1 00us ↓ 1 0 00us ↓ 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	II Image: Re-launch 680 ns 900 ns Image: Re-launch 900 ns Image: Re-launch Image: Re-launch	
	Instances and Processes ++ □ \$ × Instance and Process Name Instance And P	Objects Simulation Objects for test E	Define Clock Define	alue 620 ns alue 620 ns e signal to noments oreviously e	2 1 00us ↓ 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	Re-launch P00 ns P00 ns ps	920
🔶 🟹 😨 📀 🗊	Instances and Processes ++ □ \$ X Instance and Process Name Instance Instance Insta	Objects Simulation Objects for test E	Define Clock Define	alue	2 1 00us y 3 40 ns 8 40 ns 1 0000 1 0000 1 0000	Re-launch P00 ns ps	1920
i 🔶 💌 🧒 📀 💼	Instances and Processes ++ □ 0 × Instance and Process Name Instance Instance Insta	Objects Simulation Objects for test E bject Name Value Clock_period 1000ct	Define Clock Name Define Clock Define Cl	alue	2 1 00us y 3 40 ns 8 40 ns 1 0000 1 0000 1 0000	Re-launch P00 ns ps	920
i 🗹 🔶 📰 🙋 📀 💼	Instances and Processes ++ □ 6 × Instance and Process Name Instance and Process Name Unstance and P	Objects Simulation Objects for test to bject Name Value Cock_period 1000ct	Define Clock Name Define Clock Define Cl	alue	A 1 00us ↓ X 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	II Image: Re-launch Image: Re-launch 900 ns Image: Re-launch Image: Re-launch Image: Re-launch Image: Re-la	1920
0 🖸 🐳 🗖 🧿 💼	Instances and Processes ++ □ 6 × Instance and Process Name Instance and Instance and Ins	Objects Simulation Objects for test to bject Name Value bject Name Cock_period 1000ct	Define Clock Name Define Clock Define Cl	alue	A 1 00us y 0 (354.405 ns) (840 ns)	II Image: Re-launch Image: Re-launch 900 ns Image: Re-launch Image: Re-launch Image: Re-launch Image: Re-la	920
	Instances and Processes ++ □ 6 × Instance and Process Name Instance and Instance and Ins	Objects Simulation Objects for test to bject Name Value bject Name Cock_period 1000ct	Define Clock Name Name Name Name Name Search Solution Solut	alue		II Image: Re-launch Image: Re-launch 900 ns Image: Re-launch Image: Re-launch Image: Re-launch Image: Re-la	<u>1920</u>
🗾 🕐 🗹 🐳 📢 📀 📑	Instances and Processes ++ □ 6 × Instance and Process Name Instance and Process Name 0 test_bench 0 std_logic_1164 164 164 164 164 164 164 164	Objects Simulation Objects for test to bject Name Value bject Name Cock_period 1000ct	Define Clock Name	alue alue e signal to nments bench/b e stat485 ns	A 1 00us y 4 (354405 ns) (840 ns) (II Re-launch 900 ns 900 ns ps 900 ns	<u>, 920</u>
谢 🕐 🗹 🔶 🔻 🙋 📀 📑	Instances and Processes ++ □ 6 × Instance and Process Name Instance and Instance and Instan	Objects Simulation Objects for test to bject Name Value bject Name Value V	Define Clock Name Name Name Name Name Name Source S	alue	1 00us y 640 ns 1840 ns 1840 ns 1840 ns 100us y 10000	II Re-launch 680 ns 900 ns ps III	
	Instances and Processes ++ Instances and Processes ++ Instance and Process Name Instance and Process Name	Objects Simulation Objects for test to bject Name Value bject Name Value Value Value Value Value Value Value Value Value Value Value Value Value Value Value	Define Clock Name	alue 820 ns alue 820 ns e signal to nments oreviously teench/b T	1 100us y 6 1 40 ns 1660 ns	II Image: Re-launch 880 ns 900 ns ps Image: Re-launch	2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
	Instances and Processes ++ 0 * X Instance and Processes ++ 0 * X Instance and Process Name test_bench te std_logic_1164 * Console WARNING: Figure 4 the model of the model WARNING: Figure 4 the model of the model of the model WARNING: Figure 4 the model of t	Objects Simulation Objects for test to bject Name Value bject Name Value V	Define Clock Name Define Clock Name Define Clock Define Clock There parameters below to force the n alternating pattern (clock). Assig ade from within HDL code or any pplied constant or clock force will t verriden alue Radix Binary eading Edge Value: 1 ancel after Time Offset: 0 ancel after Time Offset: 0 0 Cancel Apply Cancel Apply	alue 820 ns alue 820 ns e signal to nments oreviously the thench/b s thelp ences between the Lite and	1 100us Y 100us Y 100us Y 1054485 ns 1840 ns 1840 ns 1860 ns 1840 ns <	II Image: Re-launch 880 ns 900 ns ps Image: Re-launch	220 220 → → → → → → → → → → → → → → → → → → →
	Instances and Processes ++ 0 * X Instance and Processes ++ 0 * X Instance and Process Name test_bench te std_logic_1164 * Console WARNING: Sim will run in Life mode. For This is a Life version of Sim. Time resolution is 1 ps	Objects Simulation Objects for test to bject Name Value bject Name Value Value Value Value Value Value Value Value Value Valu	Define Clock Name Name Define Clock Name Define Clock naternating pattern (clock). Assigned naternating pattern (clock). Assigned naternating pattern (clock). Assigned naternating pattern (clock). Assigned naternating attern (clock). Apply	alue 820 ns alue 820 ns e signal to noments oreviously te thench/b thench/b thench/b	1 100us y 6	II Re-launch 880 ns 900 ns ps III	220 220 ► ► ► ► ► ► ► ► ► ► ► ► ►
	Instances and Processes ++ 0 * X Instance and Processes ++ 0 * X Instance and Process Name test_bench te std_logic_1164 ** Console WARNING: Sim will run in Life mode. This is a Life version of Sim. Time resolution is 1 ps Simulator is doing circuit initialization process.	Objects Simulation Objects for test to bject Name Value bject Name Value Value Value Value Value Value Value Value Value Valu	Define Clock Name Name Define Clock Define Clock	alue 820 ns alue 820 ns e signal to nments previously the bench/b s Help s Help	* * 1 100us * 1 (556485 ns) (840 ns)	II Image: Re-launch 880 ns 900 ns ps Image: Re-launch	
	Instances and Processes ++ 0 × Instance and Processes ++ 0 × Instance and Process Name test_bench	Objects Simulation Objects for test to bject Name Value bject Name Value Value Value Value Value Value Value Value Value Value Value Value Value Value Value Va	Define Clock Name Nam Name Name Name Name Name Name Name	alue 820 ns alue 820 ns e signal to nments previously the thelp ss Help ss Help	* * 1 100us * 1 (556485 ns) (840 ns)	II Image: Re-launch 880 ns 900 ns ps Image: Re-launch	
	Instances and Processes ++ 0 * X Instance and Processes ++ 0 * X Instance and Process Name test_bench test_ben	Objects Simulation Objects for test L bject Name Value Cock_period 1000C Cock_period	A arrest and a second sec	alue 620 ns alue 620 ns e signal to nments oreviously re Leench/b s Help ences between the Lite and	* 1 00us 656485 ns 840 ns 840 ns 1 00us 100000 10000 10000 10000 1000	B80 ns 900 ns B80 ns 900 ns PS PS	

For more details view this video : <u>https://www.youtube.com/watch?v=EheplQla9hg</u>

Common Errors

In Entities and Architectures:

reference: VHDL for Programmable_Logic by Kevin Skahill

There are several common errors that are worth mentioning. Identifying them early may prevent misconceptions. Some of the errors are with syntax, others with semantics. Following is a code example with several errors. See if you can identify the errors:

```
entity many errors is port --line 1
      a: bit vector (3 to 0) ; --line 2
     b: out std logic vector(O to 3) ; --line 3
      c: in bit vector (6 downto 0) ;) --line 4 \setminus
end many errors --line 5
                 line 6
architecture not so good of many errors --line 7
begin --line 8
my label: process --line 9
     begin --line 10
     if c x"F" then --line 11
     b <= a --line 12
     else --line 13
     b <= ' 0101'; --line 14
     end if --line 15
end process; --line 16
end not so good --line 17
```

We'll take this design one line at a time because there are so many errors. The port declaration requires an "(" at the end of line I or beginning of line 2. In line 2, "to" should read "downto." The lack of the keyword IN to identify the mode is acceptable. If the mode is not explicitly declared, then the default of IN is assumed. Line 3 is ok. The semicolon on line 4 should appear after the second ")". The omission of a semicolon is one of the most common syntax errors. A semicolon is required at the end of line 5. There's even an error in line 6: The comment character " " is required. Line 7 is missing the keyword "is" after the name of the entity. Line 8 is ok. The process sensitivity list is missing from line 9. Line 10 is ok. The comparison of line 11 will always evaluate to FALSE because x"F" represents "1111" not "001111 "-six bits must be compared with six bits (this is not error, but is probably not what the designer wanted). In line 12, a signal of one type may be assigned to a signal only of the same base type, so we will have to change a or b to be of the same type. Line 13 is ok. The single quote marks (') of line 14 should be replaced with double quote marks ("). Line 15 requires a semicolon. Line 16 is ok. Line 17 requires a semicolon. The design is corrected and listed below.

Corrected code:

```
entity many errors is port(
      a: std logic vector (3 downto 0);
      b: out std logic vector(0 to 3);
      c: in bit vector(6 downto 0));
end many errors;
architecture not so good of many errors is
begin
my label: process(c, a)
     begin
     if c "001111" then
     b <= a;
     else
     b <= "0101";
     end if;
end process;
end not so good;
```

Initialisation of signals in architecture:

```
Credits: Team techNologic
```

```
architecture Behavioral of find_smallest_i is
    signal b_new: STD_LOGIC_VECTOR(N downto 0);
    signal i current : STD LOGIC VECTOR(N-2 downto 0);
```

begin

```
process(clk, reset)
variable flag : boolean := false;
b_new <= b; -- b is an input signal from entity definition</pre>
```

This snippet gives an error about initialising of b_new. Stating both b_new<3> and b<3> are active.

This can be simply fixed by initialising b_new in a proper fashion.

```
architecture Behavioral of find_smallest_i is
    signal b_new: STD_LOGIC_VECTOR(N downto 0) := b; -- changed line
    signal i_current : STD_LOGIC_VECTOR(N-2 downto 0);
begin
    process(clk, reset)
    variable flag : boolean := false;
```

In Creating Combinational and Synchronous Logic

reference: VHDL for Programmable_Logic by Kevin Skahill

Usage of variables:

In terms of processing VHDL for simulation, variables are only meaningful in a process and do not retain their values between periods of time when the processes are inactive. Variables must be reinitialized each time the process is activated. Synthesis creates logic based on these assumptions. Thus, the following code does not describe a counter:

```
architecture incorrect of bad_counter
begin
count: process (rst, clk)
    variable cnt: integer;
begin
    if rst = '1' then
        cnt := 0;
    elsif (clk'event and clk='1') then
        cnt := cnt + 1;
    end if;
end process;
end incorrect;
```

The variable ent does not retain its value for the time during which the process is inactive.

In using Arithmetic and Comparison operations

These functions are defined in the library 'IEEE.Std_logic_unsigned.ALL'. reference: <u>https://www.cs.sfu.ca/~ggbaker/reference/std_logic/unsigned/</u> The functions in this library take STD_LOGIC_VECTOR arguments and treat them as unsigned integers when performing the operations. Hence, this library must be included whenever any arithmetic operations are to be performed.

In Synthesis :

Signal is connected to multiple drivers

Consider the following code:

```
signal out2: std_logic_vector(7 downto 0);
begin
out1 <= out2;
process(clk1)
begin
if(clk1'event and clk1='1') then
out2 <= out2 + "00000001"; --increment by '1'
end if;
end process;
process(clk2)
begin
if(clk2'event and clk2='1') then
out2 <= out2 + "00000011"; -- --increment by '3'
end if;
end process;
```

end Behavioral;

The above code can be successfully compiled without any errors or warnings.But when you try to synthesis it you will get the following errors:

ERROR:Xst:528 - Multi-source in Unit on signal <0>>; this signal is connected to multiple drivers.

ERROR:Xst:528 - Multi-source in Unit on signal ; this signal is connected to multiple drivers.

•••

ERROR:Xst:528 - Multi-source in Unit on signal ; this signal is connected to multiple drivers.

This error occurs when we try to change a signal in two different processes. As you can see from the above code the signal 'out2' is driven by two clocks, named clk1 and clk2. A multi-driven signal cannot be realized in hardware. If the clk1'event and clk2'event occurs at the same time then, the hardware doesn't know which statement to execute. That is why this kind of code is not synthesizable. And there is no way to solve this error. Only option is to change your logic in some way that you can get your things done without using a multi-driven signal.

For more details visit :

http://vhdlguru.blogspot.in/2010/03/synthesis-error-signal-is-connected-to.html

The Golden Rules of Debugging

reference : https://www.doulos.com/knowhow/fpga/debugging/

1) Always fix the first error first:

Error: COMP96_0078: mux4_tb.vhd : (8, 16): Unknown identifier "STD_LOGIC_VECTOR".

This error generally pops on the top, the fix is most of the times

library IEEE; use IEEE.std logic 1164.all;

the error comes as the compiler can't resolve logic vector, so similarly when ever you get an unknown identifier try to find if you are including the right libraries

2) Don't ignore warnings

```
Warning (10492): VHDL Process Statement warning at alu.vhd(32):
signal "Op" is read inside the Process Statement but isn't in the Process Statement's
sensitivity list
```

This warning is interesting because most synthesis tools will actually automatically "fix" your sensitivity list, that is they will "pretend" that the missing signal is actually there. However simulation tools are not allowed to "pretend", so you will get a mis-match between simulation and synthesis. Your simulation may appear to be doing what you expect, but your hardware will behave differently. Or perhaps even more confusing, your hardware may behave correctly but your simulation will fail!

In that case the fix is easy, and the consequences are sufficiently bad you should fix the problem.

In synthesis you may be warned about latches - again this is an important warning that you should take notice of it.

3) Try rebuilding

Sometimes tools go wrong for no apparent reason.

4) Don't Panic!

Additional reference:

For syntax: <u>http://www.ics.uci.edu/~jmoorkan/vhdlref/vhdl_golden_reference_guide.pdf</u> For further more errors: http://class.ece.iastate.edu/cpre583/ref/VHDL/Common_VHDL_mistakes.pdf

Compiled By:

Utkarsh Kumar Chandra Maloo Aditya Kusupati Anand Dhoot Jash Dave