## **FPGA Verification of ALU**

Click the tab at the bottom of the Design Manager to switch to the *CSELib* library view.

Highlight the *ALU\_Test* component, then right click the highlighted black area and select *copy* from the pop-up menu.



Switch back to the *ALU* library view and right click the *ALU* Design Unit and select *paste* in the pop-up menu.

Image: Construct State of the state of	Image: Comparison     Image: Comparison       Image: Comparison     Comparison	Los ner	The first join grant minute i	24				
Name     Carting Legions (Layor yenepoid)     Carting Legions (Layor yenepoid)     Carting Legions (Layor yenepoid)       All     All     Composed     All     All     All       All     All     Composed     All     All     All       All     Composed     All     All     All     All       All     Composed     Block     All     All     All       All     Composed     Block     Block     Block     Block       All     Composed     Glock     Block     Block     Block	Name     Certip Legical University Response Control Vector Factor Vector Factor Vector     This       Image: Control International Control Internatione Control Internatin Control International Control Intern	6 8 3		3 M 3 🗗 1 🐜 -	12 · 10 1 · 10 · 10 ·	M - 🖤 - 🛃 -		
Composite     Type     Decision     Decision       In the second of the seco	Composite     Type     Decision     Decision       Image: Second biology     Composited     Composited     Composited       Image: Second biology     Second biology     ALU     ALU       Image: Second biology     Second biology     Second biology     ALU       Image: Second biology     Second biology     Second biology     Second biology       Image: Second biology     Second biology     Second biology     Second biology       Image: Second biology     Second biology     Second biology     Second biology       Image: Second biology     Second biology     Second biology     Second biology       Image: Second biology     Second biology     Second biology     Second biology       Image: Second biology     Second biology     Second biology     Second biology       Image: Second biology     Second biology     Second biology     Second biology       Image: Second biology     Second biology     Second biology     Second biology       Image: Second biology     Second biology     Second biology     Second biology       Image: Second biology	tain	Design Explorer   Using viewpoint - Default View	soint (Fithred) - Dont Touch I	Hisden 1		Tasks	
Carrier     Type       The Charge Arrivation     Type       The Arrivation     Type <th>Charles     Type       The Charge-Checker     The Charge-Checker       The Charge-Checker     Type       The Charge-Checker     Charge-Checker       The Checker     Charge-Checker       The Checker     Checker       The Checker     Checker    &lt;</th> <th>vi Add Vi Add Nata Nata Nata Nata</th> <th>Chephold Type Charles ALU Chephonet ALU Chephonet ALU Chephonet Chepho</th> <th>Estands</th> <th>Design Nearchy = ★AU = ★AU = ₹ AU = ₹ anot</th> <th>ALU III</th> <th>y My Tatas         (g) Wy Tratas         (g) Wy Tratas         (g) Wy Checker         (g) Wy Checker Flow         (g) Wy Checker         (g) Wy Chechecker Flow         (g) Wy Checker Flow</th> <th></th>	Charles     Type       The Charge-Checker     The Charge-Checker       The Charge-Checker     Type       The Charge-Checker     Charge-Checker       The Checker     Charge-Checker       The Checker     Checker       The Checker     Checker    <	vi Add Vi Add Nata Nata Nata Nata	Chephold Type Charles ALU Chephonet ALU Chephonet ALU Chephonet Chepho	Estands	Design Nearchy = ★AU = ★AU = ₹ AU = ₹ anot	ALU III	y My Tatas         (g) Wy Tratas         (g) Wy Tratas         (g) Wy Checker         (g) Wy Checker Flow         (g) Wy Checker         (g) Wy Chechecker Flow         (g) Wy Checker Flow	
s S MaxButt2_bhrav.htd Source File II s S MaxButt2_bhrav.htd Source File II store	Comparison, Mit Mit Greenstad File     Sport Lynemiddentificats what Greenstad File     Dig Call Lynemiddentificats what Greenstad File     Dig Call Lynemiddentificates what Greenstad File     Dig Call Lynemiddentes     Dig Call Lynemidden	plore	Construction     C	Type Generated File Generated File Source File Generated File	7		A fairs (Systema) Tool     Nore, Inpot     Nore, Inpot     Nore, Inpot     Nore, Inpot     Outrus I Systema     Outrus I Systema     Outrus I Proce and Route     Outrus I Processor     Outrus I Processor	

Double click the ALU\_Test component to view the VHDL codes for the component. Find the statement "LIBRARY CSELIB;" and replace all the USE statements with "USE CSELIB.ALL;". Be sure to save the file after editing it.



In Design Manager click the *Tasks/Templates* tab at the top right part of the window and double click the task *Quartus II Synthesis* from the *My Tasks* list.



After a moment, both a dialog box named *FPGA Synthesis Setup* and another one named *Log Window* pop up. In *FPGA Synthesis Setup* change the settings for the Altera EP2C35F672C6 FPGA. Click the **OK** button. Next change *Design Frequency* to 50 MHz, which is the default clock frequency for the Altera DE2 board. Click the **OK** button again.

D FPGA	Technolog	y Setup 📃 🗙				
-FPGA Flow-						
FPGA Vendor:	altera	•				
Family:	cyclone ii	•				
Device:	ep2c35	<b>_</b>				
Package:	f672c	<b>-</b>				
Speed:	6	<b>-</b>				
FPGA Libraries This can be used to compile the necessary technology files for simulation						
Compile/Update						
	ок	Cancel				



Now the *Log Window* shows the various synthesis steps being performed by Quartus II. The error messages, if any, would be reported at the end of the synthesis process. Please exam the error messages carefully.

K IN IN A WALL AND ALLU, Letti Wall (L. Edit Wall) selected name in use clause is not an expanded name File Acctivity (intro-Domy, S1), project/UndrikU), Testi Wall Line 12 Enor (1000), VHD, enor al ALU, Testi Wall (S1), selected name in use clause is not an expanded name File Acctivity (intro-Domy, S1), project/UndrikU), Testi Wall Line 13 Enor (1000), VHD, enor al ALU, Testi Wall (S1), selected name in use clause is not an expanded name File Acctivity (intro-Domy, S1), project/UndrikU), Testi Wall Line 13 Enor (1000), VHD, enor al ALU, Testi Wall (S1), selected name in use clause is not an expanded name File Acctivity (intro-Domy, S1), project/ALUbrikU, Testi Wall Line 13 Enor (1000), VHD, enor al ALU, Testi Wall, Testi Wall Line 13 Enor (1000), VHD, enor al ALU, Testi Wall, Testi Wall Line 13 Enor (1000), VHD, enor al ALU, Testi Wall, Testi Wall Line 15 Enor (1000), VHD, enor al ALU, Testi Wall, Line 15 Enor (1000), VHD, enor al ALU, Testi Wall, Line 15 Enor (1000), VHD, enor al ALU, Testi Wall, Testi Wall, Line 15 Enor (1000), VHD, enor al ALU, Testi Wall, Line 15 Enor File (PUIme (Intro-Dolm, Enormatic Status), Selected name in use clause is not an expanded name File Acctivity (intro-Dolm, Enormatic Status), Selected name in use clause is not an expanded name File Acctivity (intro-Dolm, Enormatic Status), Selected name in use clause is not an expanded name File Acctivity (intro-Dolm, Enormatic Status), Selected name in use clause is not an expanded name File Acctivity (intro-Dolm, Enormatic Status), Selected name in use clause is not an expanded name File Acctivity (intro-Dolm, Enormatic Status), Selected name in use clause is not an expanded name File Acctivity (intro-Dolm, Enormatic Status), Selected name in use clause is not an expanded name File Acctivity (intro-Dolm, Enormatic Status), Selected name in use clause is not an expanded name File Acctivity (intro-Dolm, Enormatic Status),		Log Window	
Enco (1000), VEOL encord at ALU, Test VeoL(2), selected name in use clause is not an expanded name File: AccV13/ign/SDS/m_S13_mpre/LMM/ALU_Test VeoL(2), selected name in use clause is not an expanded name File: SecV13/ign/SDS/m_S13_mpre/LMM/ALU_Test VeoL(2), selected name in use clause is not an expanded name File: SecV13/ign/SDS/m_S13_mpre/LAU_MA/ALU_Test VeoL(2), selected name in use clause is not an expanded name File: SecV13/ign/SDS/m_S13_mpre/LAU_MA/ALU_Test VeoL(2), selected name in use clause is not an expanded name File: SecV13/ign/SDS/m_S13_mpre/LAU_MA/ALU_Test VeoL(2), selected name in use clause is not an expanded name File: SecV13/ign/SDS/m_S13_mpre/LAU_MA/ALU_Test VeoL(2), selected name in use clause is not an expanded name File: SecV13/ign/SDS/m_S13_mpre/LAU_MA/ALU_Test VeoL(2), selected name in use clause is not an expanded name File: SecV13/ign/SDS/m_S13_mpre/LAU_MA/ALU_Test VeoL(2), selected name in use clause is not an expanded name File: SecV13/ign/SDS/m_S13_mpre/LAU_MA/ALU_Test VeoL(2), selected name in use clause is not an expanded name File: SecV13/ign/SDS/m_S13_mpre/LAU_MA/ALU_Test VeoL(2), selected name in use clause is not an expanded name File: SecV13/ign/SDS/m_S13_mpre/LAU_MA/ALU_Test VeoL(2), selected name in use clause is not an expanded name File: SecV13/ign/SDS/m_S13_mpre/LAU_MA/ALU_Test VeoL(2), selected name in use clause is not an expanded name File: SecV13/ign/SDS/m_S13_mpre/LAU_MA/ALU_Test VeoL(2), selected name in use clause is not an expanded name File: SecV13/ign/SDS/m_S13_mpre/LAU_MA/ALU_Test VeoL(2), selected name interviewed name File: S		0000 <u>00</u>	
	Erick (10800) VHOL error AccV13/10.nd/Shing_111 AccV13/10.nd/Shing_111 AccV13/10.nd/Shing_111 Erick (10800), VHOL error AccV13/10.nd/Shing_11 Error (10800) VHOL error AccV13/10.nd/Shing_11 Error (10800) VHOL error AccV13/10.nd/Shing_11 Error (10800) VHOL error Error Flag AccV13 for an Error Erick Shing AccV13 Error Erick Shing AccV13 Error Flag AccV13	ral ALU, Trett vhol(15): elected name in use clause is not an expanded name File: 	
Control         Control           Control         Cont			V
Kon Bio C Marco Register     State Date (R) Fig. 2019 (R) R R R R R R R R R R R R R R R R R R		Log Window	- 0
Into: Register ISSP_Done_[jattource_probe_attource_probe_componentfattource_probe_body_attource_probe_body_intgattource_probe_ingl\equal_width_ge equal_width_inglid_jom_srive_instance_id_genrom_info_insteverd_counter[]? I lost all to fanouts during netitid optimizations. ISSP_Done_istance_id_extrem_info_insteverd_during_insteverd_counter[]? I lost all to fanouts during netitid optimizations. ISSP_Done_istance_id_genrom_info_insteverd_counter[]? I lost all to fanouts during netitid optimizations. Info (TVAR) togic cell forme info (TV	13 10 10 0 15	8688 <u>8</u>	
2IS Synthesis completed successfully	Info: Register 15SP_Done J.S.P. Done ISSP_Done J. 15SP_Done ISSP_One J. 15SP_Done:SSP_Done J. 15SP_Done:SSP_Done J. 15SP_Done:SSP_Done Info: Found the following in this for Found the following in this (for Generating had, blic froi implemented 3 log) this: Implemented 3 log) this: Implemented 1208 this: Implemented 1208	Instructe_probe_sitiource_probe_component[attource_probe_body_attource_probe_body_inttalitource_probe_insplt _sr/no_pristance_id_genorum_info_intitource_probe_body_attource_probe_body_inst[attource_probe_insplt _sr/no_pristance_id_genorum_info_intitource_probe_body_attource_probe_body_inst[attource_probe_insplt]; _sr/so_pristance_id_genorum_info_intitource_probe_body_attource_probe_body_inst[attource_probe_insplt]; _sr/so_pristance_id_genorum_info_intitource_probe_body_attource_probe_insplt]; Tooref _spandon_fraud_block_aub_genorum_info_intit _vice resources after synthesis - the final resource count might be different proprint _spitterist successful 0 errors, 0 warnings pr_272_majb/bft	rednal-wiqti-Be

Next double click the task *Quartus Place and Route* from the *My Tasks* list. In the popup dialogue box named *Quartus Integrated Place & Route*, select *Run Place & Route in Batch* Mode under *Place & Route Run Mode*. Then click the **OK** button. After a moment, the Quartus II IDE will be launched.

D Quartus Ir	itegrated Place	e & Route 🛛 🗕 🗙					
Synthesis Source							
🔶 Quartus QIS Synthesis	results						
Other tools Synthesis results							
Constraints File							
Don't Use a Constraints File							
♦ Use QSF Constraints File in SideData/PAR/Constraints							
Specify a Custom Constraints File							
Browse							
Use HDS gui settings if they exist in external constraint files							
Place & Route Run Mode							
📕 Run Place & Route in B	atch Mode						
Produce Script File For Batch Run							
Launch Quartus II Inter	active ints File to Side Data	on Exit					
ок	Cancel	Advanced					

During the process of Place & Route Log Window shows the timing requirements were not met. However, we can ignore the warning messages for the ALU design.

Double click the task *Quartus Programmer* from the *My Tasks* list. In the pop-up dialogue box named *FPGA Technology Setup*, click the **OK** button. After a while Quartus Programmer dialogue box pops up and click the *Run Programmer* button at the bottom of the dialogue box.

Hardware programming file
<ul> <li>Use Quartus II Programmer GUI</li> </ul>
◇ Choose programming file from list
qis/@a@I@u_@test_struct/ALU_Test.s
Browse for programming file
Browse
Advanced
Software programming file
Open Nios II IDE
Run Programmer Cancel

LHardware Set	qı	No Hardware	: 1	Mode	JTAG			Prog	ress:			
Enable real-tim	ISP	to allow backgr	ound program	mming (f	or MAX II d	evices)			1			
₩ <sup>b</sup> Start		File	Devic		Dhecksum	Usercode	Program/ Configure	Verify	Blank- Check	Examine	Security Dit	C
<b>#</b> №Stop												
Auto Detect												
XDefete												
Add File	4					nik.						
Change File.												
Save File												
Add Device.												
<b>1</b> <sup>th</sup> ∪p												
#Down												
a i												
7ype Hessage	÷.											
												-

Then in the window named Quartus II Programmer – [Chain1.cdf], click the *Hardware Setup...* button at the top left of the window and select the USB-Blaster in the window that pops up. Click the *Close* button after you select the Hardware.

	Har	dware Setup	
Hardware Settings JTA	G Settings		
Select a programming har hardware setup applies on	rdware setup to u Ily to the current	se when programmin programmer window.	g devices. This programming
Currently selected hardwa	re: USB-Blaste	r [USB 2-1.1]	\$
Available hardware items	;		
Hardware	Server	Port	Add Hardware
Hardware USB-Blaster	Server Local	Port USB 2-1.1	Add Hardware
Hardware USB-Blaster	Server Local	Port USB 2-1.1	Add Hardware Remove Hardware
Hardware USB-Blaster	Server Local	Port USB 2-1.1	Add Hardware Remove Hardware
Hardware USB-Blaster	Server Local	Port USB 2-1.1	Add Hardware Remove Hardware

Observe that the configuration file *ALU\_Test.sof* is listed in the window in Figure 9-3. If the file is not already listed, then click the **Add File...** button and select it. This is a binary file produced by the Compiler's Assembler module, which contains the data needed to configure the FPGA device. Note also that the device selected is **EP2C35F672**, which is the FPGA device used on the Altera DE2 board. Press the **Start** button in the window in Figure 9-3. A LED on the board will light up when the configuration data has been downloaded successfully. If you see an error reported by Quartus II software indicating that programming failed, check to ensure that the board is properly powered on.

0	Programmer (Beta	i) - /acct/s1/jinz	/altera_wor	k/ALU/ALU	_Test - ALI	J_Test	- [Chair	n1.cdf]*		-	• ×
<u>F</u> ile <u>E</u> dit <u>∨</u> iew <u>F</u>	Project <u>A</u> ssignments	P <u>r</u> ocessing <u>O</u> pti	ons <u>T</u> ools <u>W</u>	/indow <u>H</u> elp							
🗈 🗳 🖬 🎒	🕺 😢 🖌 🏈	🥺 🚸 👘	🕨 🦻 🏍	🐚 🔿	ا 🍕	2 0	)				
1 Hardwara Sa		100 2 1 11	Mode	ITAG			Bro	aross:			0%
- Hardware Se	USB-Blaster [t					•	- FIO	gress.			070
Enable real-tim	ne ISP to allow backgro	und programming	g (for MAX II d	evices)							
🔊 Start	File	Device	Checksum	Usercode	Program/ Configure	∨erify	Blank- Check	Examine	Security Bit	Erase	ISP CLAMP
Stop	ALU_Test.sof	EP2C35F672	003BCAC9	FFFFFFF	~						
Auto Detect											
🗙 Delete											
🗳 Add File											
				1000							
Change File.											
Save File											
😕 Add Device											
	EP2C35F6	572									
	<b>←</b>										
- Down											
© Type Message	-										
S0 (											•
System / Pr	rocessing /\ Extra Info	/ Info / Warni	ing /\ Critical	Warning /\	Error /\ Su	ppresse	d / Flag	1/			
Message:	<b>1</b>									¢ []	ocate

## Verification of ALU Design

After successful programming the FPGA device, we can verify our ALU design with a GUI test program.

- 1. Open a terminal and copy the file *ALU\_Test.tcl* from /*usr/local/3rdparty/csce611/ALU* directory to your project directory /*acct/s1/username/altera\_work/ALU* directory.
- 2. Change your current directory to /acct/s1/username/altera\_work/ALU. Then type in it the command *quartus\_stp -t ALU\_Test.tcl*. After a while, a window in Figure 10-1 pops up showing a green ALU symbol with several text entries and buns.



The usage of the test program is described as follows:

	Table 11-1.	Description	of the ALU	GUI test program
--	-------------	-------------	------------	------------------

	I	1 0	
Name	Туре	Format	Description
A	Text entry	Hexadecimal number	32-bit ALU input A
В	Text entry	Hexadecimal number	32-bit ALU input B
ALUOp	Text entry	Binary number	4-bit ALU Operation encoding
SHAMT	Text entry	Decimal number	5-bit ALU Operation encoding
Zero	Text entry	Binary number	ALU Zero flag
<b>Overflow</b>	Text entry	Binary number	ALU Overflow flag
R	Text entry	Hexadecimal number	32-bit ALU Result
			Click the Test button and your
Test	Button		ALU results are shown in Zero,
			Overflow and R text entries
Check	Button		Click the Check button to see if the
Check	Dutton		ALU design results are correct

Make sure you follow the formats of the text entries. Otherwise, some ALU inputs may be set to 0 and a warning message is prompted. If you have problem with testing the ALU design, ask the instructors for help.

## Use SignalTap II to Debug ALU Designs

This tutorial explains how to use the SignalTap II feature within Altera's Quartus II software. The Signal-Tap II Embedded Logic Analyzer is a system-level debugging tool that captures and displays signals in circuits designed for implementation in Altera's FPGAs.

Before you open the SignalTap II window, ensure the JTAG programmer (ByteBlaster) is connected between the Altera DE2 board and the computer, and that the board has power.

After successfully compiling a completed project, select SignalTap II Logic Analyzer from **Tools > SignalTap II Logic Analyzer** within *Quartus II*, which brings up the window in Figure 11-1.

	SignalTap II Logic Analy:	zer (Beta) - /acct/s1/jir	nz/altera_work/ALU/ALU_Test	t - ALU_Test
<u>F</u> ile <u>E</u> dit <u>V</u> iew <u>P</u> roject <u>A</u> ssignments P <u>r</u>	ocessing <u>T</u> ools <u>W</u> indow <u>H</u>	elp		
	o 🖓 🕺 🖌 🏈 🤻	🦻 🚸 i 🌐 i 🕨 😻 🕨	» 🏷 🗿 🧶 💆 🤅	0
Instance Manager: 🍡 🐶 🔳 🔛	Add nodes to the current in	stance	🕘 x	JTAG Chain Configuration: JTAG ready 🛛 🖉 🗙
Instance Status LE usage	Memory usage M512/LU	TRAM M4K/M9K usag M-	RAM/M144K usage	Hardware: USB-Blaster [USB 2-1.1]
🗏 🛃 auto Not running				
				Device: (@1: EP2C35 (0x020B40DD)
				File:
auto_signaltap_0	Lock mode: 🔓 Allow all c	hanges 🔷		Signal Configuration: ×
Node Type Alias Name	Data Enable	Trigger Enable 0	Trigger Conditions	
Type Times Hame	ů	Ŭ		Data E
				Sample depth: 128 ¢ RAM type: M4K/M9K ¢
				Segmented: 2 64 sample segments
				Storage qualifier:
				Type: 🗱 Continuous 💠
				Input port:
Data Setup				
Hierarchy Display:		×	🗆 Data Log: 🛃	x
auto_signaltap_0				
				0% 00:00:00

Figure 11-1. The SignalTap II window

Once SignalTap has opened, look under the **JTAG Chain Configuration** in Figure 11-1 and select the programming hardware *USB-Blaster* in the **Hardware:** entry.

You can also open the SignalTap II Logic Analyzer by selecting **File > New**, which gives the window shown in Figure 11-2. Select **SignalTap II Logic Analyzer File** in the **Verification /Debugging Files** category and click the OK button to reach the window displayed in Figure 11-1.

To open an existing **.stp** file already associated with your project, on the Tools menu, click **SignalTap II Logic Analyzer**. You can also use this method to create a new **.stp** file if no **.stp** file exists for the current project.

To open an existing file, on the File menu, click Open and select a .stp file

о Ne	W	×
SOPC Builder System     Design Files     AHDL File     Block Diagram/Schematic File     EDIF File     IP Variation File     SystemVerilog HDL File     Tol Script File     VHDL File     Verlog HDL File     Hexadecimal (Intel-Format) File     Memory Initialization File     Verification/Debugging Files     In-System Sources and Probes File     Logic Analyzer Interface File     SignalTap II Logic Analyzer File     Other Files     AHDL Include File		
– AHDL Include File – Block Symbol File – Chain Description File – Synopsys Design Constraints File – Text File		•
	OK Cancel Help	

Figure 11-2. Choose to prepare a SignalTap II File.

We now need to add the nodes in the project that we wish to probe. In the Setup tab of the SignalTap II window, right-click in the window shown in Figure 11-3., bringing up the Node Finder window in Figure 11-4. If there are no nodes for addition, make sure a Quartus project (e.g. ALU Test) has been opened within *Quartus II*.

Signaffap II Logic	Select <u>A</u> ll	Ctrl+A	U_Test	t - ALU_Test	- • ×			
Eile Edit View Project Assignments Processing Tools Win D 🕼 💭 🎒 🎒 X h 🕅 🗠 ా 🖓 🎾 🖊	End			Ø				
Instance Manager: 🍳 🎝 🔳 🔠 Add nodes to the cu			×	JTAG Chain Configuration: JTAG ready	🛛 🕖 🗙			
Instance Status LE usage Memory usage M	Find Next Bus Value Find Brevious Bus Value			Hardware: USB-Blaster [USB 2-1.1]	Setup     Scan Chain			
	Add facdes with Plug in Add Nodes Plug in Optic			File:	· ) (2000 (2000)			
auto_signaltap_0 Lock mode: A	Add State Machine Nodes. Recreate State Machine Mneme Locate	onicx	15	Signal Configuration:	×			
_ ypr _ come	Group Ungroup Bename			Clock:				
	Mnemonic Table Setup <sub>2</sub> : <u>C</u> reate SignalTap II Ust File			Storage qualifier: Type: IB Continuous				
	Invert Signal			Input port:				
Data Setup	Align <u>L</u> eft Align <u>B</u> ight							
Hierarchy Display:	MSB on Top, LSB on Bottom LSB on Top, MSB on Bottom		,		×			
	Bus Display Format							
Hierarchy Display:	MSB on Top, LSB on Bottom LSB on Top, MSB on Bottom Bus Display Format	,	0					

Figure 11-3 Open Node Finder window

	Node Find	er		×
Named: \star 🖌 Filter: Pins	is: assigned	Customize	List	ок
Look in: [ALU_Test]	🖌 🛄 🖬	clude subentities		Cancel
Nodes Found:	S	elected Nodes:		
Name 🔺 Assignments T	Гуре Сг	Name 🔺 Assignr	nents Type	۲Cr
	>			
	>>			
	<			
	<<			
		•		

Figure 11-4 Node Finder window

In the **Filter** box, select **SignalTap II: pre-synthesis** and check the **Include subentities** checkbox. Click the List button and this will now display all the nodes that can be probed in the project. You can resize the window to see the names of the nodes. Since we want to tap the ALU input and output signals, highlight *A*, *B*, *SHAMT*, *ALUOp*, *R*, *Overflow* and *Zero*, and then click the > button in the middle of the window to add these signals to be probed. Then click the **OK** button. Figure 11-5 shows the selected nodes in the Node Finder window.

			No	de Fin	der			×
Named: *		~	Filter: SignalTa	p II: pre	-synthesis	¢ Cu	stomize	List OK
Look in: [ALU_Test]			<b>~</b> ][	🗸	Include subentities			Cancel
Nodes Found:					Selected Nodes:			
Name	Assignments	Туре	Creator		Name	<ul> <li>Assignments</li> </ul>	Туре	Creator
<ul> <li>ALU:ALU_i B[29]</li> <li>ALU:ALU_i B[30]</li> <li>ALU:ALU_i B[31]</li> <li>ALU:ALU_i R[31]</li> <li>ALU:ALU_i R[31]</li> <li>ALU:ALU_i R[4]</li> <li>ALU:ALU_i R[1]</li> <li>ALU:ALU_i R[1]</li> <li>ALU:ALU_i R[2]</li> <li>ALU:ALU_i R[4]</li> <li>ALU:ALU_i R[5]</li> <li>ALU:ALU_i R[5]</li> <li>ALU:ALU_i R[6]</li> <li>ALU:ALU_i R[6]</li> <li>ALU:ALU_i R[8]</li> <li>ALU:ALU_i R[9]</li> <li>ALU:ALU_i R[1]</li> <li>ALU:ALU_i R[</li></ul>	Unassigned Unassigned	Combinational Combinational	User entered User entered	> > V	ALU:ALU_i A ALU:ALU_i B ALU:ALU_i Overflow ALU:ALU_i Zero ALU:ALU_i Zero ALU:ALU_i SHAMT ALU:ALU_i SHAMT ALU:ALU_i ALUOp	Unassigned Unassigned Unassigned Unassigned Unassigned Unassigned	Combinatio Combinatio Combinatio Combinational Combinatio Combinatio	User entered User entered User entered User entered User entered User entered User entered

Figure 11-5. Add nodes in the Node Finder window.

Before the SignalTap analyzer can work, we need to set up what clock is going to run the SignalTap module that will be instantiated within our design. To do this, in the **Clock** box of the **Signal Configuration** pane of the SignalTap window, click ..., which will again bring up the Node Finder window. Click the **List** button to add all the nodes that can be added as the clock, and set *ISSP\_ALUOp:ISSP\_ALUOp\_i/altsource\_probe: altsource\_probe\_component/raw\_tck* as the clock for this SignalTap instance. Click the OK button to return to the SignalTap window in Figure 11-6.

0		SignalTap II Logic	: Analyzer (Beta) - /ac	ct/s1/jinz/altera_work/ALU	J/ALU	_Test - ALU_Test	- • ×
<u>F</u> ile <u>E</u> dit <u>V</u> iew <u>P</u> roj	ect <u>A</u> ssignments P <u>r</u> oc	essing <u>T</u> ools <u>W</u> indow <u>H</u>	elp				
🗅 🛥 🖬 🌒 🤞		🗠 💢 😢 🖌 🏈 🤻	🦻 🛞 👘 🕨 💌 🕨	5 10 O 🕘 💩 👱	0		
Instance Manager:	< P . C	compile the project to conti	nue		×	JTAG Chain Configuration: JTAG ready	
Instance Statu	LE usage	Memory usage M512/LU	TRAM M4K/M9K usag M-	RAM/M144K usage		Hardwara USB Plaster [USB 2.1.1]	▲ Cotup
🗏 🛃 auto Not ru	nning				_	Hardware. OSB-blaster [OSB 2-1.1]	- Setup
						Device: @1: EP2C35 (0x020B40DD)	🗢 Scan Chain
					-	File:	
auto_signaltap_0		Lock mode: 📑 Allow all c	hanges 🗢		_	Signal Configuration:	×
	lode	Data Enable	Trigger Enable	Trigger Conditions	_		
Type Allas	Name	107	107	J Basic	Ŧ	Clock: ALUA_Ialtsource_probe:altsource_probe_component	raw_tck
	I⊞-ALU:ALU_IA			xxxxxxxxh	-	Data	
	E-ALU:ALU i B			xxxxxxxxx		Sample depth: 128	
<b>a</b>				XXh	-		
•	ALU:ALU_i Zero			X		Segmented: 2 64 sample segments	÷
•	ALU:ALU_i Overflow			X	_	Storage qualifier:	
	. E- ALU:ALU_i R			xxxxxxxx		Type: I Continuous	÷ -
Data Setup							
Hierarchy Display:				🗙 🗌 Data Log: 🔂			×
				🔤 🕄 auto_signalta	ap_O		
auto signaltap 0							
						0	% 00:00:00

Figure 11-6. Select the ALU signals and clock for SignalTap analyzer to work

How to tap or capture the signals when the trigger condition is met? An additional signal "Done" is added to the ALU Top-level file as the trigger condition. When the signal is asserted high, the trigger condition is met and we are finished with one ALU operation and able to probe the current values of ALU input/output signals.

With the Setup tab of the SignalTap window selected, add the *Done* signal in the Node Finder window, and then find the **Trigger Conditions** column. In the dropdown menu at the top of this column, select **Basic**. Right-click on the Trigger Level cell corresponding to the node *Done* and select **High**. Now, the trigger for running the Logic Analyzer will be when the signal, *Done*, is set to high, as shown in Figure 11-7. Note that you can right click on the Trigger Levels cell of any of the nodes being probed and set the trigger condition to a number of choices.

After instantiating SignalTap II logic analyzer file in your design, make sure you save the file by selecting **File > Save as.** The default file name is "**stp1.stp**". Make sure you save the SignalTap file with the same file name when you make modifications to it.

ð		SignalTa	ap II Logic Ana	alyzer (Beta)			-	•		
ile <u>E</u> dit ⊻iew <u>P</u> roject <u>A</u> ssig	nments Processin	g Tools Window H	telp							
	n a a	× % / @ 8	9 () () ()	- 15° ≥0 *00	0 0 0	🖻 🔬 🐠				
nstance Manager: 🔤 😡	E Progra	m the device to cont	inue	💓 🗙	JTAG Chain	Configuration: JTAG ready	10	a) y		
stance Status	LE usage Mer	nory usage M512/LU	TRAM M4K/M9K	usag M-RAM/M1	Trank and	uco olama fuco o 1.11				
auto Not running					Maroware.	USB-Blaster [USB 2-1.1]	•	tup		
					Device:	@1: EP2C35 (0x020B40DD)	\$ Scar	n Chair		
					File:	/acct/s]/iinz/altara_work/ALLUALL	Test sof	-		
				•						
		allow all changes	•							
Node	Data Enable	Trigger Enable	Trigger Con	ditions	Signal Conf	iguration:		×		
TypeAlias Name	108	104	1 Basic	=	Clock: uree	proheialtsource prohe compon	entiraw tok	-		
🥹 🗄 ALU:ALU İALU	)p	0.000	Contraction of the second		Contract price		entren_tex [	<u> </u>		
😔 🗄 ALU:ALU_İA	·		2000000	oth	Data					
😔 🗉 ALU:ALU_İ B			>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>	oth	Sample o	lepth: 64 🗘 RAM type: M4	к/мэк 😂	1		
😻 🗄 ALU:ALU_i SHA	MT		XXh				1.4	\$ L.		
ALU:ALU_i Overflo	v		×		Segm	ented: 2 32 sample segments	•			
ALU:ALU_i Zero			×		Storage qualifier:					
ALU:ALU_iR			0000000	Oth		IIII Continues				
I Done			1		type:	88 Continuous	÷ ]			
					Input po	ort: Lauto etn enternal storiane o	anlówe I			
Data Setup										
Hierarchy Display:			×	Data Log:				×		
				auto sign	alten 0					
ALU:ALU_i										

Figure 11-7. Setting the trigger conditions.

The SignalTap II Analyzer may not be automatically enabled after creating the file. To enable it, type in the terminal the command "*quartus\_stp ALU\_Test -e -- stp\_file=stp1.stp*" in your ALU directory. After a while, information of running Quartus II SignalTap II in the command line is produced, as shown in Figure 11-8.

Co Terminal	- 🗆 🗙
Eile Edit View Terminal Help	
jinz@gluon:~/altera_work/ALU\$ quartus_stp ALU_Teststp_file=stp1.stp -e Info: ************************************	^
Info: Running Quartus II SignalTap II	
Info: Version 9.1 Build 222 10/21/2009 SJ Web Edition	
Info: Copyright (C) 1991-2009 Altera Corporation. All rights reserved.	
Info: and other software and tools, and its AMPP partner logic	
Info: functions, and any output files from any of the foregoing	
Info: (including device programming or simulation files), and any	
Info: associated documentation or information are expressly subject	
Info: to the terms and conditions of the Altera Program License	
Into: Subscription Agreement, Altera MegaCore Function License	
Info: Agreement, or other applicable license agreement, including,	
Info: without timitation, that your use is for the sole purpose of Info: programming logic devices manufactured by Altera and sold by	
Info: Altera or its authorized distributors. Please refer to the	
Info: applicable agreement for further details.	_
Info: Processing started: Sat Jan 16 12:10:57 2010	
Info: Command: quartus_stp ALU_Teststp_file=stp1.stp -e	
Info: Quartus II SignalTap II was successful. 0 errors, 0 warnings	
Info: Peak virtual memory: 71 megabytes	
Into: Processing ended: Sat Jan 16 12:10:58 2010	
Info: Elapsed Lime: 00:00:01 Info: Total (PU time (on all processors): 00:00:00	
linz@gluon:~/altera work/ALU\$	

Figure 11-8. Enable SignalTap II in the command line

The last step is to compile the design. Click on the toolbar icon  $\blacktriangleright$  and start compilation. After compilation, use the toolbar icon to load the project onto the DE2 board. The compiler may display some warning messages, but make sure *Quartus II Analysis & Synthesis* was successful with 0 errors and 0 warnings. If you have trouble with the compilation of the project, please ask the instructors for help.

## 12. Probing the Design Using SignalTap II

Now that the project with SignalTap II instantiated has been loaded onto the DE2 board, we can now probe the nodes as we would with an external logic analyzer.

1. Select **Processing > Autorun Analysis** or click the icon in the SignalTap window. Then, click on the **Data** tab of the SignalTap II Window. You should get a screen similar to Figure 12-1. Note that the status column of the SignalTap II Instance window says "Waiting for trigger." This is because the trigger condition has not yet been met.

		× 🕱 % 🤉	0000		12	P0 0	0	00	盘	10					
nstance Manager: 🏨 🎭	■ [£] Acq	uisition in prog	ess						×	JTAG Chai	n Configur	ation: JT	AG ready		
nstance Sta	us	LE usage	Memory u	age MS12/L	UTRAM	M4K/M9	K usag	M-RAM/N	1144K	Hardware				÷	Setup
auto_signaltap_0 Wai	ing for trigger								÷.						International Action
										Device:				:	Scan Cha
										File: 🛓	/acct/s1	/jinz/alter	ra_work/AL	U/ALU_	Test.sof
									•						
laa: 2010/01/16 12:45:40															
TypeAlias Name	0 04. Op Acquisition	8 1 in progress	ļ21€.	20	. 2,4 ,	click	to mse	rt time bi 2 , , , 3/	r •	. 4p 4	4 4	B.,,1	5,2 5	<sup>6</sup>	60 6
OpenAlia         Name           Image: State and	0 4 . Op Acquisition	ę 1	121¢.	. 20	. 24 .	click	to inse	rt time bi 2 , 3/	\$	.40 4	a 4j	8	5,2 5	5	60 6
Deta Setup	ro O, , 4 , COp Acquisition MT W W	, ę , 1	12., <u>1</u> 6.	20	. 24 .	dd: 28 .	to inse	rt time his 2 34		4p 4	4 4	8	5,2 , , , 5	¢	6p <u>6</u>
TypeAlias Name TypeAlias Name Alias Aliastu jiku HALIALU jiku ALIastu jika ALIastu jika ALIastu jika ALIastu jika ALIastu jika Done Data Setup Hierarchy Display:	ro Op Acquisition IMT W	, ţ , 1	,2 1,6 .	20	. 2,4 . ×	cick 	Log:	nt time hi 2 34	·	4ρ 4	4 4	8	5,2 , , , 5	6	60 <u>,</u> 6
TypeAlias Name TypeAlias Name ALUALU jALU jALU ALUALU jALU jB ALUALU jALU jB ALUALU jALU jB ALUALU jCero ALUALU jCero ALUALU jCero ALUALU jCero ALUALU jCero ALUALU jCero ALUALU jCero ALUALU jB ALUALU JB A	o o Acquisition MT w	, . ए 1 in progress	12.,.1¢.	. 20	. 2,4 . ×	Cick 28 . Deta	Log: {	rt time ba 2 3/ 2 3/ 2 3/	\$	4p 4	4 4	8 5	52 5	5	60 e

Figure 12-1. SignalTap II window after Run Analysis has been clicked.

2. Now, to observe the trigger feature of the Logic Analyzer, launch the ALU GUI test program by typing the command "quartus\_stp -t ALU\_Test.tcl" in your ALU directory. Enter ALU input data and operation encodings in the text boxes. After you click on the **Test** button, the data tab of the SignalTap II window should display the similar image in Figure 12-2. And the ALU input/output signals should display the same values as those shown in the GUI test program.

nstance Manager: 🍬 🕸 🔳 🖹	Acquisition in progress		×	JTAG Chai	n Configuration: JTAG ready		
istance Status           stance         Status           auto_signaltap_0         Waiting for the	LE usage Memory usagi MS12/LUTRAM. M	4K/M9K usag M-RAM/M144K	usage	Hardware Device: File: 🛓	USB-Blaner (USB 2-1-1) (01: EP2C35 (0x02dB4000) (/acct/s1/jinz/altera_work/ALB	¢ ¢ J/ALU_	Setup Scan Cha Test.sof
log: 2010/01/16 12:45:49 #0 TypeAilas Name 9, # ALU3ALU_jALU50 # ALU3ALU_jAL # ALU3ALU_jA # ALU3ALU_jB # ALU3ALU_jB	. <del>.4</del>	скіск to meant time bar , 20, , 24, , 29 ок вовововолі вовововолі		32 <u>3</u> 6	4ρ4β	4 + 4	5,2 , , ,5
ALU:ALU_i Overflow     ALU:ALU_i Zero     ALU:ALU_i Zero     ·· ALU:ALU_i R     Done Data Setup		0000000Ah					
Hierarchy Display:	×	Data Log: 💽					,

Figure 12-2. Graphical display of values after trigger condition is met.

Whenever you enter new values into the ALU input boxes and click on the Test button, the waveforms in the **Data** tab of the SignalTap II window will be updated with the values you have just entered in the GUI test program. So you can add more ALU internal

signals for debugging purpose. To stop the *Autorun Analysis* function, click the icon If you do not want to continuously probe the signals, instead of clicking *Autorun Analysis*, click **Run Analysis** which is the icon left next to the *Autorun Analysis* icon. If you do this, the value in the display will be updated only once when the trigger condition is met.