



#### SignalTap II





#### Agenda

- SignalTap II Interface in Quartus II S/W
- SignalTap II Demonstration
  - Initial Compilation
  - Second Compilation:
     Using SignalTap II Logic Analyzer
- Conclusions









### SignalTap II Interface in Quartus II Software



## SignalTap II File (.stp)

- Creating SignalTap<sup>®</sup> II Logic Analyzer File
  - Choose New (File Menu)
  - Click the Other Files Tab & Select SignalTap II File
  - Click OK

🗗 Stol	
Instance Manager:	JTAG Chain Configuration × Hardware: Please Select ▼ Setup Device: None Detected ▼ Scan Chain File:
auto_signaltap_0       Type     Alias       Name     Out       Data     Trigger	Signal Configuration:       ×         Clock:       Trigger In:         Sample Depth:       Input Pattern:         128 samples       ✓         Trigger Levels:       Trigger Out:         Trigger Position:       ✓         \$\$\vee Pre       ✓
A Data Setup	□ Data Log: 🔄 × 🕄 auto_signaltap_0





### SignalTap II File

**Hierarchy Display** 

	<mark>n</mark> Stp1		
	Instance Manager: 🕨 🖓 🗆 🔛 No Device Selected 🛛 🗙 🗙	JTAG Chain Configuration X	
	Instance Status	Hardware: Please Select 💌 Setup	
1	S. auto_signaltap_0 Ready to Hun		
Instance -			JTAG Chain
Manager		File: 📥 🛄	Configuratio
	auto_signaltap_0	Signal Configuration: ×	
	Type Alias Name Out Data Trigger VL1	Clock:	
		Sample Depth: Input Pattern:	
Signal		128 samples 🔽 🛛 🔛 Don't Care	Signal
Viewer		Trigger Levels:	Configuratio
VICWCI		1 level	Configuratio
		Trigger Position: Output Level:	
		₩ Pre 🗾	
	🔊 Data 👷 Setup		
	Hierarchy Display: ×	🗖 Data Log: 🔂 🗙	
	auto_signaltap_0		
	•		

**Data Log** 

10



#### SignalTap II File: Acquisition Clock

#### Acquisition Clock

- For Best Results, Assign Only Global Clock as the SignalTap II Clock Signal
- Without Assigning Clock Signal, Quartus II Software Creates Clock Pin auto\_stp\_external\_clock

		Signal Configuration:		×	
Acquisition Clock	-	Clock:	- Trigger In:		
Sample Depth	-	Sample Depth: 128 samples	Input Pattern: Don't Care		Trigger In/Out
Trigger Levels	-	Trigger Levels:	Trigger Out:		Thgger in/Out
Trigger Position	-	Trigger Position:	Output Level:		



# SignalTap II File: Trigger Pattern

#### Sample Depth

- Set Number of Samples Stored for Each Input Signal
  - 0 to 128K Sample Depth
- Trigger Levels
  - Configure Analyzer with up to 10 Trigger Levels
- Trigger Position
  - Specify Amount of Data Captured by SignalTap II
     Logic Analyzer that Should be Acquired before the
     Trigger as well as Amount that Should be Acquired
     after the Trigger





# **Trigger Position**

#### Pre-Trigger

 Captures Signals that Occur Immediately after Triggering (12% Pre-Trigger, 88% Post-Trigger)



#### Center Trigger

 Captures Signals before & after Triggering (50% Pre-Trigger, 50% Post-Trigger)



# **Trigger Position**

#### Post-Trigger

 Captures Signal that Occur Immediately before Triggering (88% Pre-Trigger, 12% Post-Trigger)



#### Continuous Trigger

 Captures Signals Indefinitely Until Stopped Manually (Useful When Using the Trigger Out Feature)



# SignalTap II File:Trigger In/Out

#### Trigger In

- Any I/O Pin Can Trigger the SignalTap II Analyzer
- Pin auto\_stp\_trigger\_in\_0 Generated in Device
- Trigger Input Can be Set to Recognize High, Low, Rising Edge, Falling Edge, Either Edge, or Don't Care Condition

#### Trigger Out

- Spare I/O Pin that Is Set as Trigger Output Signal That Indicates When Trigger Pattern Occurs
- Pin auto\_stp\_trigger\_out\_0 Generated in Device
- Output Pulse Polarity Is Programmable





### SignalTap II File: Debug Ports

- Routing SignalTap II Signal to Spare I/O Pin for Capture by Logic Analyzer
- Quartus II Software Automatically Generates a Pin
  - Debug Port Pin Name Is <a href="stp\_debug\_out\_1\_n">stp\_debug\_out\_1\_n</a>
    - n Is a Number Representing the Order in Which the Debug Port Pin Occurs in the Signal List

Туре	Alias	Name	Out	Data	Trigger	₩L1
		CNT_ONE_ENABLE		~	•	
		CNT_ONE0		V		
•3		CNT_ONE1		Debug P	ort On	
		CNT_ONE2		Debug P	ort: Off	
•3		CNT_ONE3		~	•	





### SignalTap II File: Control Settings



![](_page_12_Picture_2.jpeg)

![](_page_12_Picture_3.jpeg)

### **Specify SignalTap II File**

- Choose Compiler Settings (Processing Menu)
- Click the SignalTap II Logic Analyzer Tab
- Turn on Enable SignalTap II Logic Analyzer
- In the SignalTap II File Name box:
  - Type the Name of the SignalTap II File (.stp) for Compilation
  - Or Select a File Name with Browse Button

Compiler Setti	ings				×		
					and the second the second the second		
General	Chips & Devices	Mode	Synthesi	s & Fitting	Verification		
SignalTap	) II Logic Analyzer	Design A	ssistant	Netlist	Optimizations		
Specify which SignalTap II File to compile with the selected project. Note: the availability of these options depends on the current device family.							
Enable SignalTap II Logic Analyzer							
SignalTap II File name: C:\Training\Fixed\Stp1.stp							

![](_page_13_Picture_8.jpeg)

![](_page_13_Picture_9.jpeg)

### **Project Compilation**

- Design Recompilation Required When Any of These Parameters Change
  - Acquisition Clock
  - Number of Channels
  - Sample Depth
  - Debug Ports
  - Trigger in/out Ports
- Design Recompilation Not Required When
  - Changing Trigger Pattern or Position
  - Modifying Trigger Levels
  - Starting or Stopping SignalTap II Logic Analyzer
- Internal Nodes to Be Analyzed Should be Inputs, Outputs, Register Outputs, or Memory Outputs

![](_page_14_Picture_12.jpeg)

![](_page_14_Picture_13.jpeg)

#### SignalTap II Demonstration

- Description of the Test Case
- Initial Compilation
  - Observing the Malfunction
- Second Compilation
  - Creating the Embedded Logic Analyzer
  - Debugging Design
- Third Compilation
  - Fixing the Design Issue

![](_page_15_Picture_9.jpeg)

![](_page_15_Picture_10.jpeg)

### **Design: Counter from 00 to 99**

- Function: Two 4-Bit Counters Cascaded to Drive a Pair of Seven-Segment LEDs that Count from 0 to 99
- Top Level Name: Counter
- Target Device : EP20K200EFC484-2X
- Tools
  - Synthesis: LeonardoSpectrum
  - Fitter: Quartus II Version. 2.1
- Place Source Design Files in C:\Training\Synthesis Directory
- Place pins.tcl File in C:\Training Directory

![](_page_16_Picture_9.jpeg)

![](_page_16_Picture_10.jpeg)

#### **Counter Block Diagram**

![](_page_17_Figure_1.jpeg)

\* : The Prescaler Generates a Counter Enable with a Frequency of about 1 Hz

![](_page_17_Picture_3.jpeg)

![](_page_17_Picture_4.jpeg)

# Synthesize Design (1/2)

- Launch LeonardoSpectrum Tool
- Select Mode Quick Setup

![](_page_18_Picture_3.jpeg)

- Working Directory: C:\Training\Synthesis
- Open Files: counter.vhd
- Technology: APEX 20KE
- Device: EP20K200EFC484
- Speed Grade: -2X
- Output File: C:\Training\Synthesis\counter.edf

Click on Run Flow

![](_page_18_Picture_11.jpeg)

![](_page_18_Picture_12.jpeg)

### Synthesize Design (2/2)

![](_page_19_Figure_1.jpeg)

### **Create Quartus II Project**

- Launch the Quartus II Software
- Choose New Project
   Wizard (File menu) to
   Create a New Quartus II
   Project
  - Directory Name:
     C:\Training\Fitting
  - Project Name: counter
  - Top Level Name: counter
- Click Next

#### New Project Wizard: Directory, Name, and Top-Level Entity [page 1 of 6]

What is the working directory for this project? This directory will contain design files and other related files associated with this project. If you type a directory name that does not exist, Quartus II can create it for you.

C:\Training\Fitting

What is the name of this project? If you wish, you can use the name of the project's top-level design entity.

counter

What is the name of the top-level design entity in your project? The Quartus II software will automatically create Compiler and Simulator settings for the top-level entity you specify in this wizard. After you create a project, you can add more top-level entities and create Compiler and Simulator settings for them with commands on the Processing menu.


![](_page_20_Picture_14.jpeg)

![](_page_20_Picture_15.jpeg)

...

...

#### **Create Quartus II Project**

#### Add Following Files in Project

- C:\Training\Synthesis\counter.edf
- Click Next
- Click Finish

ew Project Wizard: Ac	ld Files [page 2 of	6]		
Select the design files a Add All to add all design Note: it is optional to ad directory, or files in whic	nd software source file files and software sou d files here unless you h the file name is not th	s you want to inclu rce files in the pro have design files un ne same as the en	ude in your ject directo not contain tity name.	project. Click ny. red in the project
<u>F</u> ile name:				<u>A</u> dd
File name		Туре		Add All
\Synthesis\COUNTE	R.edf	EDIFI	File	<u>R</u> emove
				<u>P</u> roperties
				<u>Ш</u> р
				Down
If your project includes li	braries of custom func	tions, specify their		
parinanies.	J <u>s</u> er Library Pathname:	s		
	Back	Next	Finish	Cancel

![](_page_21_Picture_6.jpeg)

![](_page_21_Picture_7.jpeg)

#### **Create Project: Set EDA Tool**

#### Select LeonardoSpectrum Tool (Level 1) as Design Entry/Synthesis Tool

Click Next

DA tools		T			
Toortype Design entry/synthe Simulation Timing analysis Board-level Formal verification Resynthesis	esis	<ul> <li>Leonardo</li> <li><none></none></li> <li><none></none></li> <li><none></none></li> <li><none></none></li> <li><none></none></li> <li><none></none></li> <li><none></none></li> </ul>	Spectrum(Lev	vel 1)	
Tool settings Tool type: D Tool name: 1	esign entry .eonardo 9	y/synthesis Spectrum( <u>Lev</u>	el 1)		<b>-</b>
Generate a cor files when they	npilable ne change	etlist automati	cally from the	source	Settings Advanced

![](_page_22_Picture_4.jpeg)

![](_page_22_Picture_5.jpeg)

#### **Create Project: Select Device**

- Select APEX & Yes to Assign Specific Device
- Click Next
- Select EP20K200EFC484-2X
- Click Finish

New Project Wizard: Device Family [page 4 of 6]	New Project Wizard: Select a Target Device [page 5 of 6]
<ul> <li>Which device ramily do you wish to target?</li> <li>Eamily: APEX20KE</li> <li>Do you want to assign a specific device?</li> <li>⊙ Yes</li> <li>⊙ No, I want to allow the Compiler to choose a device</li> </ul>	devices:       Ist. Select a device in the list, and click Next to continue.         Available devices:       EP20K160EFC484-1         EP20K160EFC484-1X       EP20K160EFC484-2         EP20K160EFC484-2X       EP20K160EFC484-3         EP20K160EFC484-3       EP20K200EFC484-3         EP20K200EFC484-1X       EP20K200EFC484-1         EP20K200EFC484-2X       EP20K200EFC484-2         EP20K200EFC484-3       EP20K200EFC484-2         EP20K200EFC484-3       EP20K200EFC484-3         EP20K200EFC484-2X       EP20K200EFC484-2         EP20K200EFC484-3       EP20K200EFC484-3
Back Next Finish Cancel	Back Next Finish Cancel

![](_page_23_Picture_6.jpeg)

![](_page_23_Picture_7.jpeg)

### Set All Unused I/Os as Tri-Stated

- Choose Compiler Settings (Processing Menu)
- Click the Chips & Devices Tab
- Click Device & Pin
   Options Button
- Click the Unused Pins Tab
  - Select As inputs, tri-stated

Device & Pin Options 🛛 🗙
General Configuration Programming Files Unused Pins Dual-Purpose Pins Voltage
Specify device-wide options for reserving all unused pins on the device. To reserve individual dual-purpose configuration pins, go to the Dual-Purpose Pins tab. To reserve other pins individually, click the 'Assign Pins' button in the Chips & Devices tab of the Compiler Settings dialog box.
Changes apply to Compiler settings 'counter'
Reserve all unused pins
As inputs, tri-stated
C As outputs, driving ground
As outputs, griving an unspecified signal
Description:
are tri-stated, as outputs that drive ground, or as outputs that drive an unspecified
signal.
<b>v</b>
<u>R</u> eset
OK Cancel

![](_page_24_Picture_7.jpeg)

![](_page_24_Picture_8.jpeg)

#### **Hardware Setup**

- Nios Demo Board with APEX EP20K200EFC484-2X Device
- DC Power Supply
- Device Download Cable (MasterBlaster or ByteBlasterMV Cable)

![](_page_25_Figure_4.jpeg)

![](_page_25_Picture_5.jpeg)

![](_page_25_Picture_6.jpeg)

#### **Compile Project & Configuration**

- Choose Start Compilation (Processing menu)
- Choose Open Programmer (Processing menu)
  - Click Add File
    - Select the File counter.sof
  - Programming Hardware : ByteBlasterMV
  - Mode : JTAG
  - Check Option Program/Configure
- Click Start

[	🖥 Chain1*										_ 🗆 ×
	<u>S</u> tart Stop	<u>M</u> ode:	JTAG	Progress:	0%	- Programming H Type: ByteBla	ardware asterMV				Setup
	Add File	File		Device	Checksum	Usercode	Program/ Configure	Blank- Check	Examine	Security Bit	
	Remo <u>v</u> e	1\	Training\counter.sof	EP20K200EF484	0001C36B	FFFFFFF					
	Up Down										
	Properties										
	Auto D <u>e</u> tect Save File										

![](_page_26_Picture_10.jpeg)

![](_page_26_Picture_11.jpeg)

#### **Observing the Malfunction**

Number 9 Never Appears on the Least & Most Significant Digit

![](_page_27_Figure_2.jpeg)

#### SignalTap II Analyzer Used to Monitor Control Signals of LEDs

![](_page_27_Picture_4.jpeg)

![](_page_27_Picture_5.jpeg)

#### **Essential Steps**

Configure SignalTap II Logic Analyzer

- Select Nodes for Analysis
- Select Acquisition Clock
- Set Sample Depth & Trigger Options
- Enable the Logic Analyzer & Compile
- Configure Device
- Set the Trigger Pattern
- Run the SignalTap II Logic Analyzer

![](_page_28_Picture_9.jpeg)

![](_page_28_Picture_10.jpeg)

# **Configure Logic Analyzer**

- Choose New (File Menu)
- Click the Other Files Tab & Select SignalTap II File

Ti Sto1

Click OK

	Instance Manager: 🕨 🌳 🗆 🔛 No Device Selected 🛛 🗙 🗙	JTAG Chain Configuration ×
New X	Instance Status auto_signaltap_0 Ready to Run	Hardware: Please Select 💽 Setup Device: None Detected 💌 Scan Chain File: 📥
Device Design Files Software Files Other Files AHDL Include File Block Symbol File Chain Description File Hexadecimal (Intel+Formal) File Memory Intelization File Cignat Ling Life Tcl Script File Tcl Script File Vector Waveform File OK Cancel	auto_signatap_0         Type       Alias         Name       Out       Data         Trigger       Image: Contract of the second sec	Signal Configuration:  Clock: Trigger In: Sample Depth: I28 samples Trigger Levels: Trigger Position: Signal Configuration: Signal C
Or Select	Hierarchy Display: X	Data Log      Auto_signaltap_0

Save the SignalTap II File as analysis.stp

![](_page_29_Picture_6.jpeg)

![](_page_29_Picture_7.jpeg)

\_ 🗆 ×

#### **Analyzer 1: Select Nodes**

- Open Node Finder Window
  - Double Click on the Signal Viewer
- Click Start to List All Pins & Internal Nodes
- Add Signal CNT\_ONE\_0 to Selected Nodes List
- Repeat
  - CNT\_ONE\_1
  - CNT\_ONE\_2
  - CNT\_ONE\_3
  - CNT\_ONE\_ENABLE
- Click OK

Node Finder		
Named: Filte	SignalTap II	Sta <u>r</u> t OK
Look in: ICOUNTERI	☑ Include subenti	ties Stop Cancel
Nodes Foun <u>d</u> :	Selected Nodes:	
Name	Assign 🔺 🛛 Name	Assignmer
■ CLOCK	Unassi CNT_ONE0	Unassigne
CLOCK_ibuf~REGOUT	Incost and a second sec	-
CLOCK_int	Unassi	
CNT_ONE0	<u>Unassi</u>	
	Unassi	
CNT_ONE2	Unassi	
CNT_ONE3	Unassi <u>&lt;</u>	
CNT_ONE_ENABLE	Unassi	
CNT_ONE_ix6~COMBOUT	Unassi <u> </u>	
CNT_ONE_ix9~COMBOUT	Unassi	
CNT_ONE_ix12~COMBOUT	Unassi	
CNT_ONE_ix15~COMBOUT		

![](_page_30_Picture_12.jpeg)

![](_page_30_Picture_13.jpeg)

#### **Analyzer 1: Acquisition Clock**

#### Add Clock Input Pin from Node Finder

Node Finder									
<u>N</u> amed:	<b>▼</b> !	Eilter: Design Enl	try (all na	mes) 💌	<u></u> u:	tomize	Sta <u>r</u> t	ОК	
Look in: COUNTER				200	🔽 Includ	le subentities	Stop	Cancel	
Nodes Foun <u>d</u> :				Selected Nodes:					
Name	Assignments	Туре 🔺		Name		Assignments	Туре		
CLOCK	counter	mpai	$\rightarrow$	🗈 CLOCK		counter	Input		
CLOCK_ibuf	Unassigned (	Combinatori							
CLOCK~out0	Unassigned (	Combinatori							
CNT_ONE_ENABLE	Unassigned I	Registered							
CNT_ONE_ix6	Unassigned I	Combinatori	>>						
CNT_UNE_IX9	Unassigned I	Combinatori			Circu.	10			×
CNT_ONE_X12	Unassigned (	Combinatori	<u> </u>		Signa	a configuration:			^
CNT_ONE_px6	Unassigned I	Combinatori	<<		Clock	c	- Triager	In:	
CNT ONE nx12	Unassigned (	Combinatori							
CNT_ONE_nx17	Unassigned I	Combinatori			1				
CNT ONE O	Unassigned	Reaistered 💌		•	Sam	ble Depth:	Input Patte	m:	
					128	samples 💌	Don't	Care	-
	C	nen Nod	. /		Trigg	er Levels:	Trigger	Out:	
	Find	der for Cl	ock		1 le	vel 🔻	1		-
	1				1		- I		
					Trigg	er Position:	Output Lev	el:	
					ŧ≈:	Pre 🔻	]		-
							·		

![](_page_31_Picture_3.jpeg)

![](_page_31_Picture_4.jpeg)

### **Configure Analyzer 1**

### Configure Logic Analyzer 1 (auto\_signaltap\_0)

	Signal Configuration:	×
	Clock: CLOCK	Trigger In:
	Sample Depth:	Input Pattern:
128 Samples	128 samples 💽	🖾 Don't Care 💌
	Trigger Levels:	Trigger Out:
Single Level	1 level	<b>_</b>
	Trigger Position:	Output Level:
Center	Senter	

#### Choose Save (File menu)

![](_page_32_Picture_4.jpeg)

![](_page_32_Picture_5.jpeg)

#### **Analyzer 2: Select Nodes**

- Right Click in Instance Manager & Select Create Instance
- Open the Node Finder Window
- Click Start to List All Pins & Internal Nodes
- Add Following Signals
  - CNT\_ONE\_0, CNT\_ONE\_1, CNT\_ONE\_2, CNT\_ONE\_3
  - CNT\_ONE\_ENABLE
  - CNT\_TEN\_0, CNT\_TEN\_1, CNT\_TEN\_2, CNT\_TEN\_3

Click OK

![](_page_33_Picture_9.jpeg)

![](_page_33_Picture_10.jpeg)

#### **Analyzer 2: Acquisition Clock**

#### Add Clock Input Pin from Node Finder

Node Finder									
Named:	•	Eilter: Design E	ntry (all na	mes)	<u>C</u> usto	omize	Sta <u>r</u> t 🔾	ОК	
Look in: COUNTER					🔄 🔽 Include	subentitie	s Stop	Cancel	
Nodes Foun <u>d</u> :				Selected No	des:				
Name	Assignments	Туре 🔺		Name		Assignm	ents Type		
CLOCK	counter	Input	$\rightarrow$	🗈 СLОСК		counter	Input		
CLOCK_ibuf	Unassigned	Combinatori							
CLOCK~out0	Unassigned	Combinatori							
CNT_ONE_ENABLE	Unassigned	Registered	<u> </u>						
	Unassigned	Combinatori	>>	l 1					<u> </u>
CNT_ONE_ix12	Unassigned	Combinatori			Signal Configura	ation:			×
CNT ONE ix15	Unassigned	Combinatori			Clock:		Trigger In:		
CNT_ONE_nx6	Unassigned	Combinatori	<<			- 🗖			
CNT_ONE_nx12	Unassigned	Combinatori							
CNT_ONE_nx17	Unassigned	Combinatori			Caral Date		Local Dellama		
CNT ONE O	Unassigned	Reaistered			Sample Depth:		Input Pattern:		
					128 samples	-	📰 Don't Care	<b>v</b>	
					Trigger Levels:	ſ	— 🗖 Trigger Out: —		
Onen Node	Finder fo	r Clock			1 level	•		7	
open node			-		Trigger Position	r:	Output Level:		
					₽= ₽= Pre	<b>•</b>		<b>v</b>	
					Ţ.				

![](_page_34_Picture_3.jpeg)

![](_page_34_Picture_4.jpeg)

# **Configure Analyzer 2**

Configure Logic Analyzer 2 (auto\_signaltap\_1) as Follows

	Signal Configuration:	×
	Clock: CLOCK	Trigger In:
	Sample Depth:	Input Pattern:
128 Samples ———	128 samples	📰 Don't Care 🔽
	Trigger Levels:	Trigger Out:
2 Levels	2 levels 🔻	~
	Trigger Position:	Output Level:
Center ———	🛱 Center 🔽	

Choose Save (File menu)

![](_page_35_Picture_4.jpeg)

![](_page_35_Picture_5.jpeg)

### Compile Design with SignalTap II Logic Analyzer

- Choose Compiler Settings (Processing Menu)
- Click the SignalTap II Logic Analyzer Tab
- Turn on Enable SignalTap II Logic Analyzer
- Browse to analysis.stp File
- Click OK
- Choose Start Compilation
- (Processing menu)

![](_page_36_Picture_8.jpeg)

![](_page_36_Picture_9.jpeg)

![](_page_36_Picture_10.jpeg)

#### **Running Analysis**

#### Select Hardware & Configuration File

![](_page_37_Figure_2.jpeg)

![](_page_37_Picture_3.jpeg)

![](_page_37_Picture_4.jpeg)

### **Running Analyzer 1**

- Set Trigger Pattern When CNT\_ONE Reaches 9
- Configure Device
- Run SignalTap II Analyzer

2. Configure Device

3. Run SignalTap II Analyzer	Instance Mana	▶     □     Image: Base of the second secon	cquire		×	JTAG Chai     Configuration       Hardware:     1: ByteBlasterMV        Device:     @1: EP20K200C        File:     Counter.sof
1. CNT_ONE  - Value 9	trigger: 2002/05/23         Type       Alias         Image: Comparison of the second	13:26:04 #1  NT_ONE0 NT_ONE1 NT_ONE2 NT_ONE3 NT_ONE_ENABLE	Out     Data       Image: Constraint of the second o	Trigger     Image: state sta		Signal Configuration:       ×         Clock:       Trigger In:         CLOCK       Input Pattern:         Sample Depth:       Input Pattern:         128 samples       Son't Care         Trigger Levels:       Trigger Out:         1 level       ✓         Trigger Position:       Output Level:         Stringer Position:       ✓

![](_page_38_Picture_6.jpeg)

![](_page_38_Picture_7.jpeg)

#### **View Results**

	<mark>1 ni</mark> analysis	s. stp					×
	Instance M	anager:	🕨 🖓 🗆 🔛 🛛 Ready to /	Acquire	×	JTAG Chain Configuration	×
	Instance auto_sig	gnaltap_0 gnaltap_1	Status Ready to Run Ready to Run			Hardware: 1: ByteBlasterMV ▼ Setup Device: @1: EP20K200C ▼ Scan Chain	
	log: 200:	2/06/03 1	5:35:59 #0			click to add timebar	-1
	Туре	Alias	Name	-16 0	16	32 48 64 80 96 1 <sup>-</sup>	12
	•5>		CNT_ONE0	l1			
			CNT_ONE1				
	•		CNT_ONE2				
	• 🔊		CNT_ONE3				_
			CNT_ONE_ENABLE				-11
Descrite Discriminant							
Results Displayed							
in Data Tab	Data	Se المظل					
of Signal Viewer 🛛 🛏	Hierarchy	Display:		>	<   r	Data Log: 🔂	×
	→	counter				<mark>≹,</mark> auto_signaltap_0	
	🛃 auto_s	ignaltap_	0 🚴 auto_signaltap_1				

![](_page_39_Picture_2.jpeg)

![](_page_39_Picture_3.jpeg)

### **Isolating the Problem**

The CNT\_ONE Counts to 9 but its Synchronous Reset Signal Occurs on the Next Clock Immediately Resetting the Counter to Zero

![](_page_40_Figure_2.jpeg)

![](_page_40_Picture_3.jpeg)

#### **Analyzer 2 - Further Analysis**

#### Select Analyzer 2 - auto\_signaltap\_1

	<mark>uri</mark> analys	sis.stp							
	Instance	Manager:	🕨 🏷 🗆 🔝 🛛 Ready to /	Acquire		[		×	JTAG Chain Configuration ×
Double Click ——	Instance	<del>signaltap_1</del> signaltap_1	Status Ready to Run Ready to Run						Hardware: 1: ByteBlasterMV ▼ Setup Device: @1: EP20K200C ▼ Scan Chain File: ♣ counter.sof
	trigger	: 2002/05/	24 09:34:45 <b>#</b> 0						Signal Configuration:
	Туре	Alias	Name	Out	Data	Trigger	<b>⊽L1</b>	IVL2	Clock:
	•©>		CNT_ONE0		V		_	-	
	•©		CNT_ONE1	-***			_	<u> </u>	Sample Depthy Input Patterny
	•③		CNT_ONE2	-***			_		100 secolar
			CNT_ONE3					22	128 samples
	- C		CNT_UNE_ENABLE				82		Trigger Levels:
			CNT_TEN1	-223			=		2 levels
			CNT_TEN2	-353		<u>,</u>			Trigger Position: Output Level:
	•		CNT_TEN3	-*:3	V		_	_	Center Active High
	Dat	a 🙀 Si	etup						
	Hierarch	y Display:					×	🗌 Data Log	r 🕅 🗙
	•••••• •••	counter			signaltap_1				
Select Tab	🔹 auto	signal	🔝 auto_signaltap_1						

![](_page_41_Picture_3.jpeg)

![](_page_41_Picture_4.jpeg)

### **Running Analyzer 2**

- Set Trigger Pattern As Shown
- Configure Device
- Run SignalTap II Analyzer

Run SignalTap II Analyzer

Instance Instance & auto auto	iviana,	▶ २ ि ि Ready t Status 0 Ready to Run 1 Ready to Run	o Acquire					×	JTAG Chan Configuration Hardware 1: ByteBlasterMV Setup Device: @1: EP20K200C Scan Chain File: Locunter.sof	×
trigge	r: 2002/05/	24 09:34:45 #0							Signal Configuration:	×
Туре	Alias	Name	Out	Data	Trigger	<b>⊡</b> L1	<b>   </b>  2		Clock:	
•3		CNT_ONE0	-***			_	—			
•3		CNT_ONE1		V		_				
•		CNT_ONE2		•		_			Sample Depth: Input Pattern:	
•3		CNT_ONE3		V		_	—		128 samples 💌 Don't Care	-
•3		CNT_ONE_ENABLE		V			<b>33</b>	- 1	Trigger Levels:	
•3		CNT_TEN0	-***	V	V	_	—			
•6>		CNT_TEN1	-83			-	_	- 1		
•3		CNT_TEN2	-***			_	_		Trigger Position: Output Level:	
•3		CNT_TEN3	-#33			—			🗱 Center 🔽	

Configure

**Device** 

![](_page_42_Picture_6.jpeg)

Trigger on 19 after Encountering 20 – Notice that Analyzer Does Trigger on First Instance of 19

![](_page_42_Picture_8.jpeg)

#### **View Results**

	ri analy:	sis.stp				-					
	Instance	Manager:	🕨 🍫 🗆 🔛 🛛 Ready to /	Acquire	×	JTAG Chain Configuration	×				
	Instance auto_ auto_	signaltap_( signaltap_1	Status D Ready to Run I Ready to Run		Hardware: 1: ByteBlasterMV ▼ Setup Device: @1: EP20K200C ▼ Scan Chain	en. Ne					
					File: 📥 counter.sof						
	log: 20	02/06/05 0	)9:58:15 <b>#</b> 0			click to add timebar					
	Туре	Alias	Name	-64 -48 -32	-16	0 16 32 48	64				
	•5>		CNT_ONE0			l					
	•3>		CNT_ONE1								
	<b>(B)</b>	CNT_ONE2									
	<b></b>		CNT_ONE3								
	•3		CNT_ONE_ENABLE								
	•3>		CNT_TEN0								
	•3>		CNT_TEN1								
	•3>		CNT_TEN2								
Results Displayed	•3		CNT_TEN3								
in Data Tab	Dat	a 😹 Se	etup								
of Signal Viewer	Hierarch	y Display:		×	🗖 Data Log	r 💁	×				
	2 3	<ul> <li>counter</li> </ul>				signaltap_1					
	🔹 autoj	_signaltap_	0 💦 auto_signaltap_1								

![](_page_43_Picture_2.jpeg)

![](_page_43_Picture_3.jpeg)

![](_page_44_Picture_0.jpeg)

#### Conclusions

![](_page_44_Picture_2.jpeg)

#### SignalTap II Benefits

- Access Internal Signals within Design
- Easy Configuration through Quartus II Interface
- Using SignalTap II Analyzer Does Not Require Making Modifications to Design Files
- Available with Quartus II Software

![](_page_45_Picture_5.jpeg)

![](_page_45_Picture_6.jpeg)