BEE 271 Spring 2017
How to use the SignalTap II logic analyzer

Nicole Hamilton

SignalTap II is the builtin Quartus logic analyzer for debugging clocked sequential circuits running on the FPGA. It's comprised of a user interface running on the PC for setting it up and viewing captured data plus the logic analyzer itself, which is compiled onto the FPGA along with your design. The logic analyzer captures data on every positive edge of the clock and then, when a trigger condition is met, transfers it to the PC for display.

The example used here is the simple a 32-bit counter included in Simulation.zip, posted to the files area in Canvas. Download and unzip it and open the SimpleCounter.qpf file.

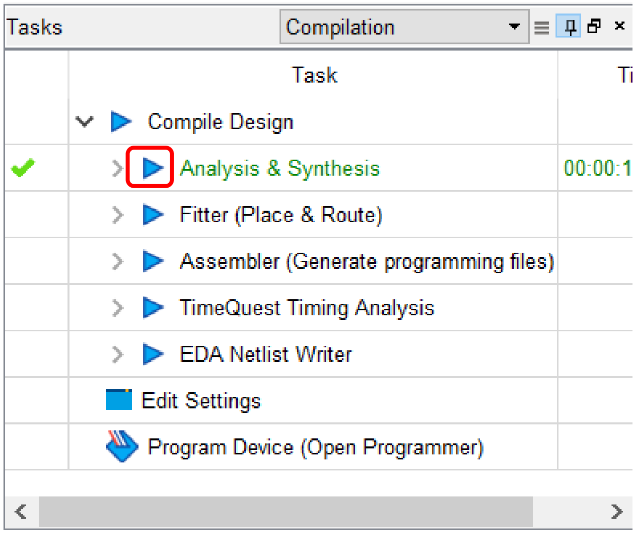
```
module CounterA(
      input clock, reset,
      input [ 31:0 ] resetValue,
      output reg [ 31:0 ] count = 0 );

   // Synchronous reset (synchronized to the clock)
   always @( posedge clock )
      count <= reset ? resetValue : count + 1;

endmodule
```

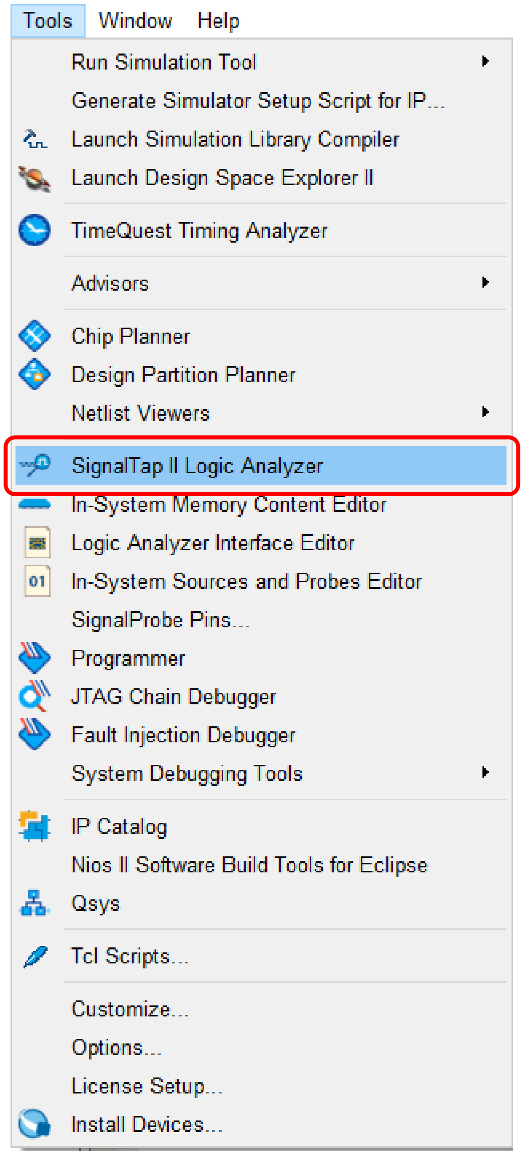
In this simple wrapper for the DE1-SoC board, the high-order 10 bits of the counter are tied to the LEDs, the reset value is tied to the switches and the reset button to KEY[3].

```
module SimpleCounter(
      input  CLOCK_50,
      input  [ 3:0 ] KEY,
      output [ 9:0 ] LEDR,
      input  [ 9:0 ] SW );

   wire reset = ~KEY[ 3 ];
   wire [ 31:0 ] count;

   assign LEDR = count[ 31:22 ];
   CounterA c ( CLOCK_50, reset, { SW, 22'b0 }, count );

endmodule
```

To add SignalTap II to the project, you must first compile it in Quartus or at least run Analysis & Synthesis. (You can do just that single step by double-clicking the blue triangle.)

You must also plug in and power up a DE1-SoC board.

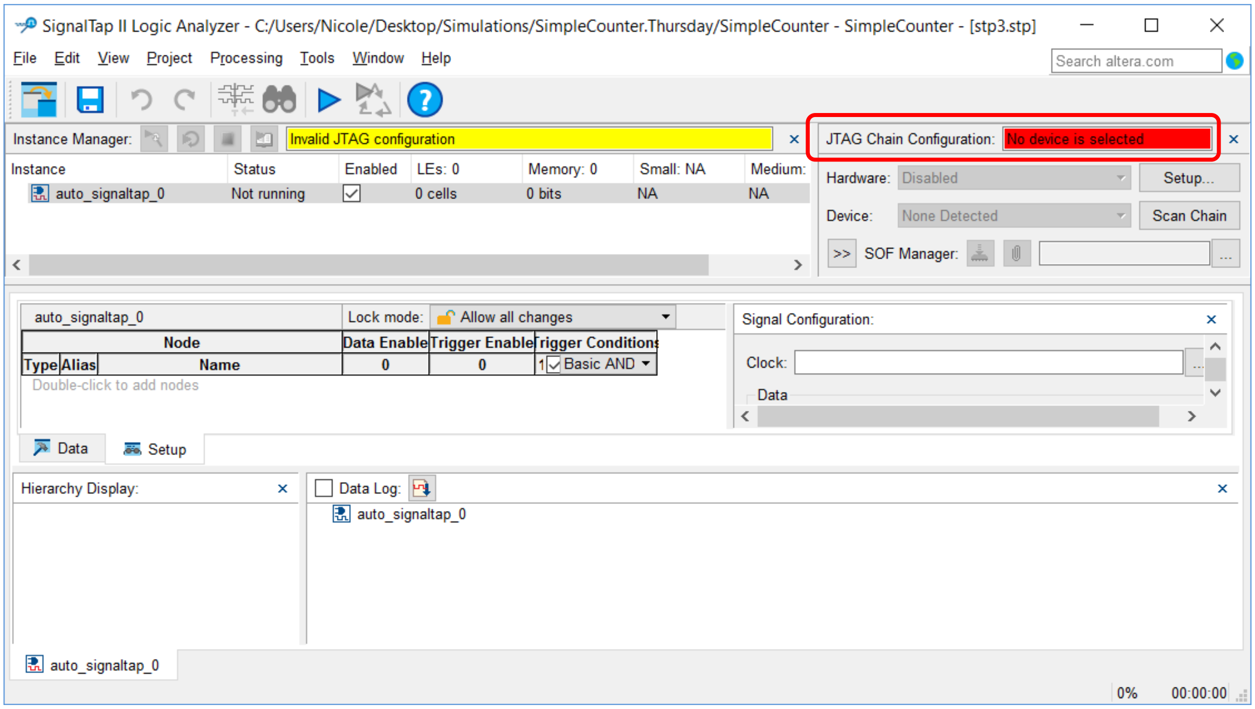



Tools
Window
Help
Run Simulation Tool
Generate Simulator Setup Script for IP...
Launch Simulation Library Compiler
Launch Design Space Explorer II
TimeQuest Timing Analyzer
Advisors
Chip Planner
Design Partition Planner
Netlist Viewers
SignalTap II Logic Analyzer
In-System Memory Content Editor
Logic Analyzer Interface Editor
In-System Sources and Probes Editor
SignalProbe Pins...
Programmer
JTAG Chain Debugger
Fault Injection Debugger
System Debugging Tools
IP Catalog
Nios II Software Build Tools for Eclipse
Qsys
Tcl Scripts...
Customize...
Options...
License Setup...
Install Devices...

Then, from the Quartus main menu bar, select Tools → SignalTap II Logic Analyzer to start SignalTap.

This will bring up the SignalTap II logic analyzer screen.

If you see this, “No device is selected”, it’s because you haven’t yet plugged in and powered up a DE1-SoC board. Close the SignalTap window, plug in and power up the DE1-SoC and then restart SignalTap.



SignalTap II Logic Analyzer - C:/Users/Nicole/Desktop/Simulations/SimpleCounter.Thursday/SimpleCounter - SimpleCounter - [stp3.stp]

Instance Manager: Invalid JTAG configuration

JTAG Chain Configuration: No device is selected

Instance	Status	Enabled	LEs: 0	Memory: 0	Small: NA	Medium:
auto_signaltap_0	Not running	☑	0 cells	0 bits	NA	NA

Hardware: Disabled

Device: None Detected

auto_signaltap_0 Lock mode: Allow all changes

Node			Data Enable	Trigger Enable	Trigger Conditions
Type	Alias	Name	0	0	1☑ Basic AND

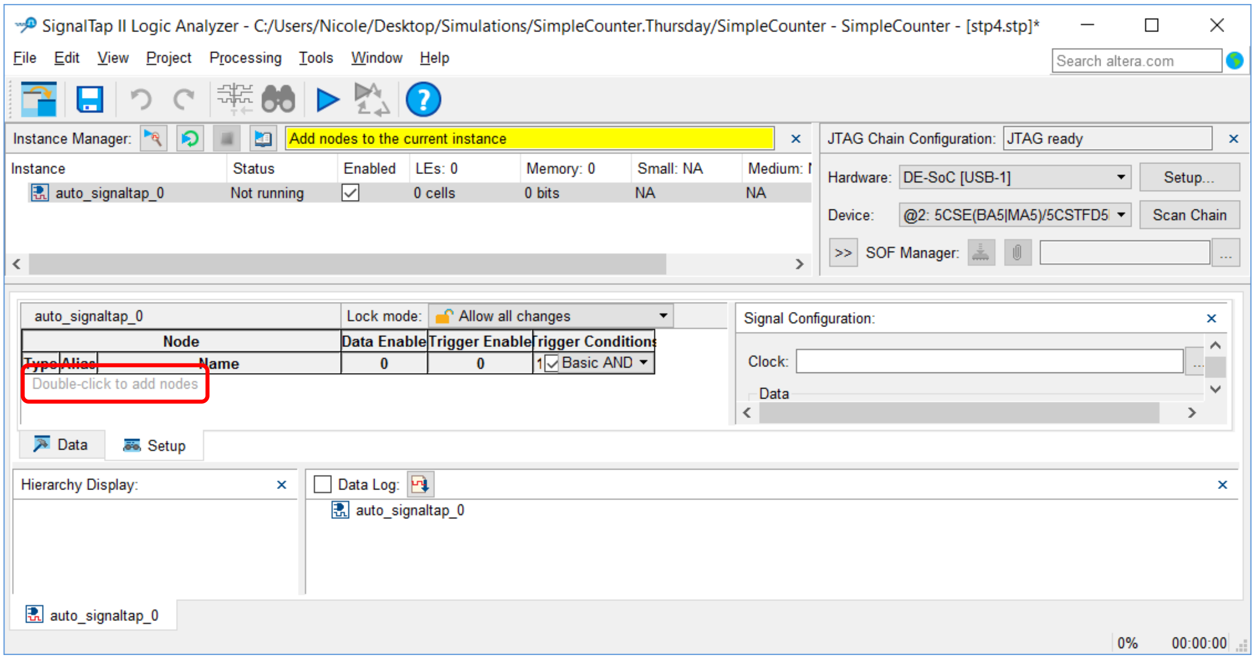
Double-click to add nodes

Signal Configuration:

Clock:

Data

This is what you should see.



SignalTap II Logic Analyzer - C:/Users/Nicole/Desktop/Simulations/SimpleCounter.Thursday/SimpleCounter - SimpleCounter - [stp4.stp]*

Instance Manager: Add nodes to the current instance

Instance	Status	Enabled	LEs: 0	Memory: 0	Small: NA	Medium: I
auto_signaltap_0	Not running	☑	0 cells	0 bits	NA	NA

JTAG Chain Configuration: JTAG ready

Hardware: DE-SoC [USB-1]

Device: @2: 5CSE(BA5|MA5)/5CSTFD5

auto_signaltap_0 Lock mode: Allow all changes

Node			Data Enable	Trigger Enable	Trigger Conditions
Type	Alias	Name	0	0	1☑ Basic AND

Double-click to add nodes

Signal Configuration:

Clock:

Data

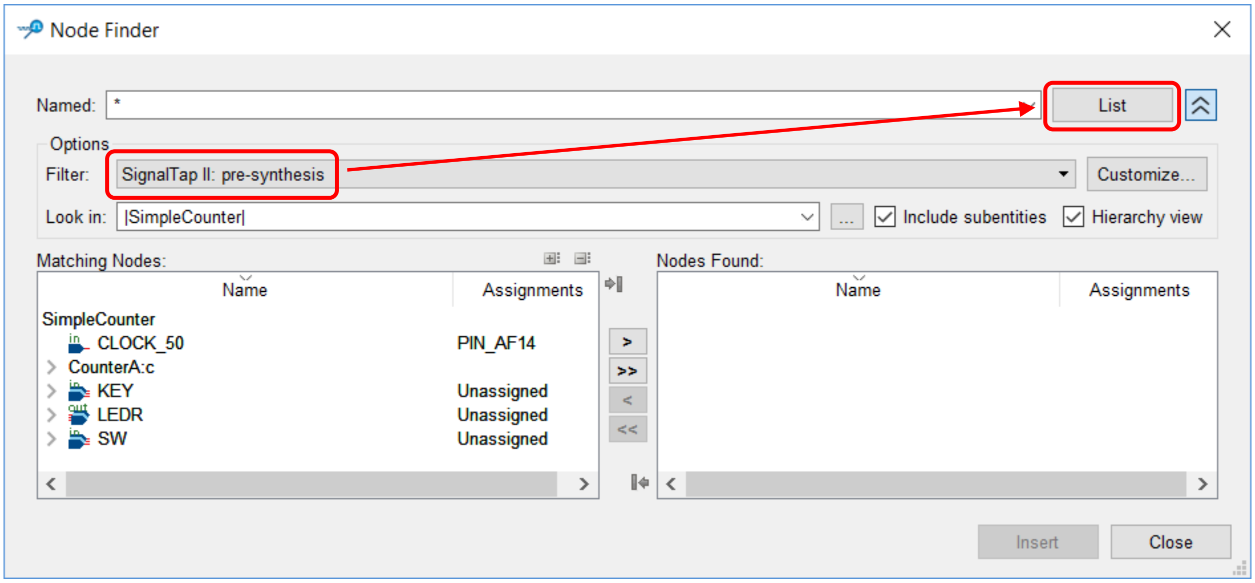
Hierarchy Display:

Data Log: auto_signaltap_0

auto_signaltap_0

0% 00:00:00

Double clicking to add nodes opens the Node Finder. Filter on "SignalTap II: pre-synthesis" nodes (i.e., the names used in your source, before optimization by the compiler), then click List. You should see something like this.



Node Finder

Named: *

Options

Filter: SignalTap II: pre-synthesis

Look in: |SimpleCounter| ☑ Include subentities ☑ Hierarchy view

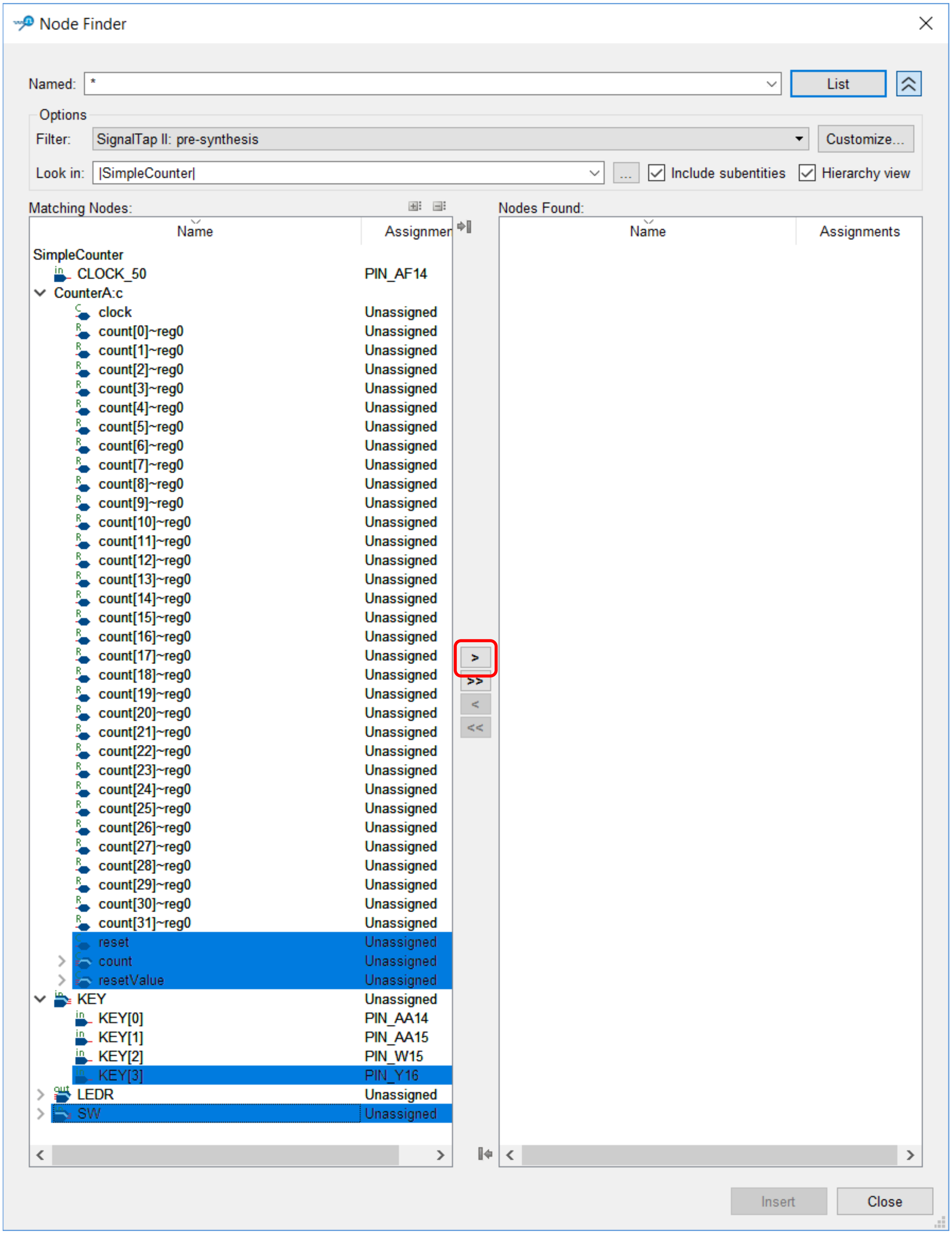
Matching Nodes:

Name	Assignments
SimpleCounter	
CLOCK_50	PIN_AF14
CounterA:c	
KEY	Unassigned
LEDR	Unassigned
SW	Unassigned

Nodes Found:

Name	Assignments

Expanding the CounterA and the KEY nodes, Ctrl-left-click to select reset, count, resetValue, KEY[3] and SW, then press > to copy them to right hand panel.



Node Finder

Named: *

Options

Filter: SignalTap II: pre-synthesis

Look in: |SimpleCounter|

☑ Include subentities ☑ Hierarchy view

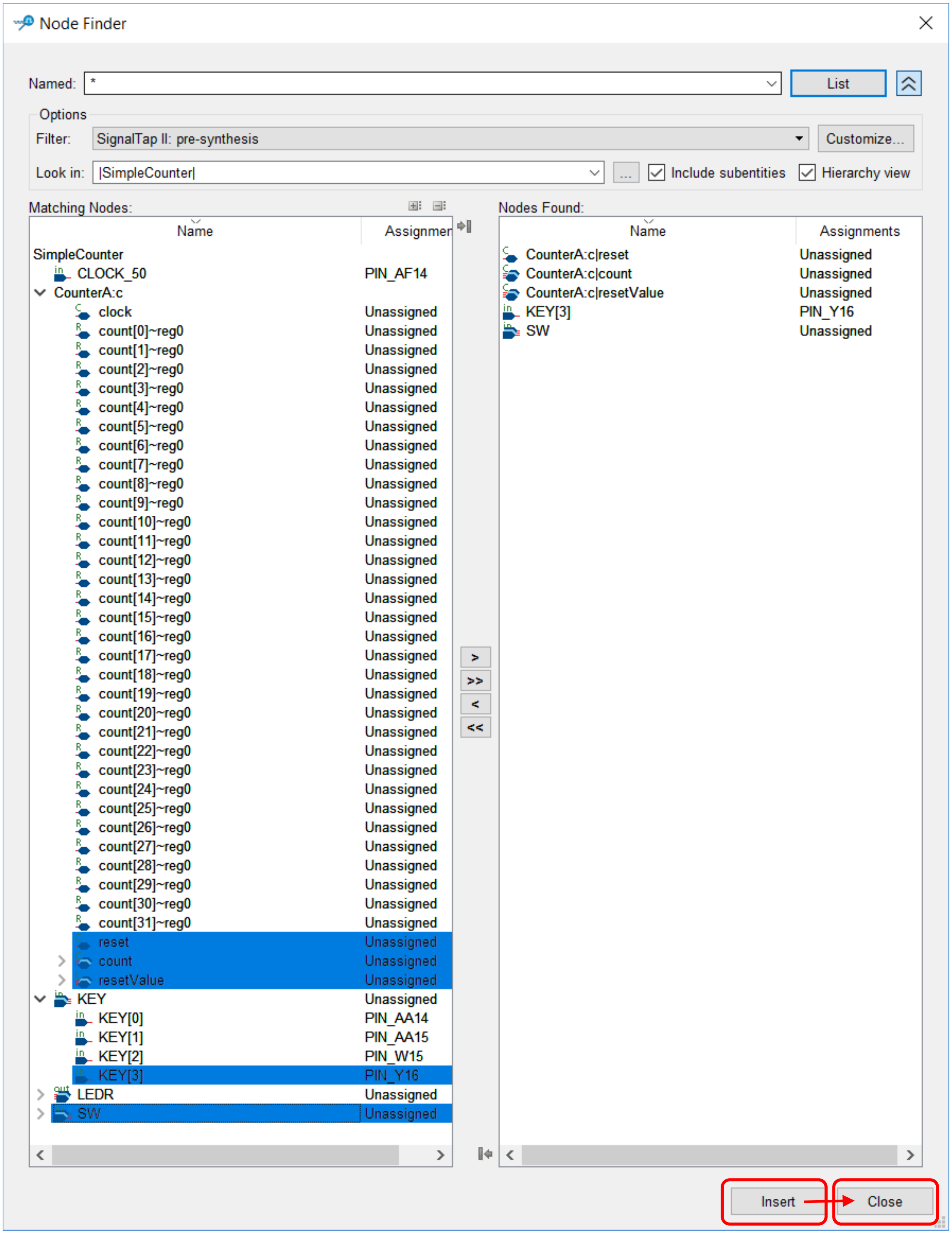
Matching Nodes:

Name	Assignmen
SimpleCounter	
CLOCK_50	PIN_AF14
CounterA:c	
clock	Unassigned
count[0]~reg0	Unassigned
count[1]~reg0	Unassigned
count[2]~reg0	Unassigned
count[3]~reg0	Unassigned
count[4]~reg0	Unassigned
count[5]~reg0	Unassigned
count[6]~reg0	Unassigned
count[7]~reg0	Unassigned
count[8]~reg0	Unassigned
count[9]~reg0	Unassigned
count[10]~reg0	Unassigned
count[11]~reg0	Unassigned
count[12]~reg0	Unassigned
count[13]~reg0	Unassigned
count[14]~reg0	Unassigned
count[15]~reg0	Unassigned
count[16]~reg0	Unassigned
count[17]~reg0	Unassigned
count[18]~reg0	Unassigned
count[19]~reg0	Unassigned
count[20]~reg0	Unassigned
count[21]~reg0	Unassigned
count[22]~reg0	Unassigned
count[23]~reg0	Unassigned
count[24]~reg0	Unassigned
count[25]~reg0	Unassigned
count[26]~reg0	Unassigned
count[27]~reg0	Unassigned
count[28]~reg0	Unassigned
count[29]~reg0	Unassigned
count[30]~reg0	Unassigned
count[31]~reg0	Unassigned
reset	Unassigned
count	Unassigned
resetValue	Unassigned
KEY	Unassigned
KEY[0]	PIN_AA14
KEY[1]	PIN_AA15
KEY[2]	PIN_W15
KEY[3]	PIN_Y16
LEDR	Unassigned
SW	Unassigned

Nodes Found:

Name	Assignments

You should see this result. Press Insert, then Close.



Node Finder

Named: *

Options

Filter: SignalTap II: pre-synthesis

Look in: |SimpleCounter|

☑ Include subentities ☑ Hierarchy view

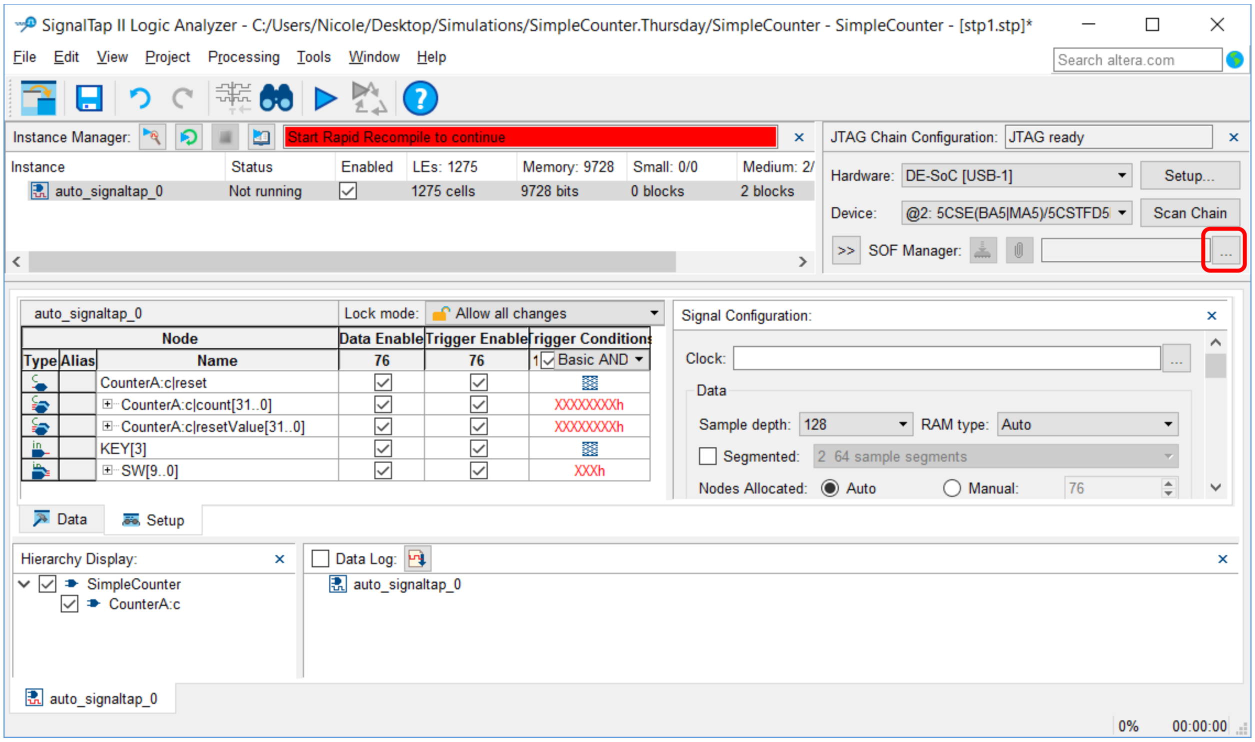
Matching Nodes:

Name	Assignments
SimpleCounter	
CLOCK_50	PIN_AF14
CounterA:c	
clock	Unassigned
count[0]~reg0	Unassigned
count[1]~reg0	Unassigned
count[2]~reg0	Unassigned
count[3]~reg0	Unassigned
count[4]~reg0	Unassigned
count[5]~reg0	Unassigned
count[6]~reg0	Unassigned
count[7]~reg0	Unassigned
count[8]~reg0	Unassigned
count[9]~reg0	Unassigned
count[10]~reg0	Unassigned
count[11]~reg0	Unassigned
count[12]~reg0	Unassigned
count[13]~reg0	Unassigned
count[14]~reg0	Unassigned
count[15]~reg0	Unassigned
count[16]~reg0	Unassigned
count[17]~reg0	Unassigned
count[18]~reg0	Unassigned
count[19]~reg0	Unassigned
count[20]~reg0	Unassigned
count[21]~reg0	Unassigned
count[22]~reg0	Unassigned
count[23]~reg0	Unassigned
count[24]~reg0	Unassigned
count[25]~reg0	Unassigned
count[26]~reg0	Unassigned
count[27]~reg0	Unassigned
count[28]~reg0	Unassigned
count[29]~reg0	Unassigned
count[30]~reg0	Unassigned
count[31]~reg0	Unassigned
reset	Unassigned
count	Unassigned
resetValue	Unassigned
KEY	Unassigned
KEY[0]	PIN_AA14
KEY[1]	PIN_AA15
KEY[2]	PIN_W15
KEY[3]	PIN_Y16
LEDR	Unassigned
SW	Unassigned

Nodes Found:

Name	Assignments
CounterA:c|reset	Unassigned
CounterA:c|count	Unassigned
CounterA:c|resetValue	Unassigned
KEY[3]	PIN_Y16
SW	Unassigned

The SignalTap II window should now look like this. Click "…" to add a clock.


SignalTap II Logic Analyzer - C:/Users/Nicole/Desktop/Simulations/SimpleCounter.Thursday/SimpleCounter - SimpleCounter - [stp1.stp]*

Instance Manager: Start Rapid Recompile to continue

Instance	Status	Enabled	LEs: 1275	Memory: 9728	Small: 0/0	Medium: 2/
auto_signaltap_0	Not running	☑	1275 cells	9728 bits	0 blocks	2 blocks

JTAG Chain Configuration: JTAG ready

Hardware: DE-SoC [USB-1]

Device: @2: 5CSE(BA5\|MA5)/5CSTFD5

SOF Manager: …

auto_signaltap_0 — Lock mode: Allow all changes

Type	Alias	Name	Data Enable 76	Trigger Enable 76	Trigger Conditions 1 ☑ Basic AND
		CounterA:c|reset	☑	☑	▨
		⊞ CounterA:c|count[31..0]	☑	☑	XXXXXXXXh
		⊞ CounterA:c|resetValue[31..0]	☑	☑	XXXXXXXXh
		KEY[3]	☑	☑	▨
		⊞ SW[9..0]	☑	☑	XXXh

Signal Configuration:

Clock: …

Data

Sample depth: 128 RAM type: Auto

□ Segmented: 2 64 sample segments

Nodes Allocated: ● Auto ○ Manual: 76

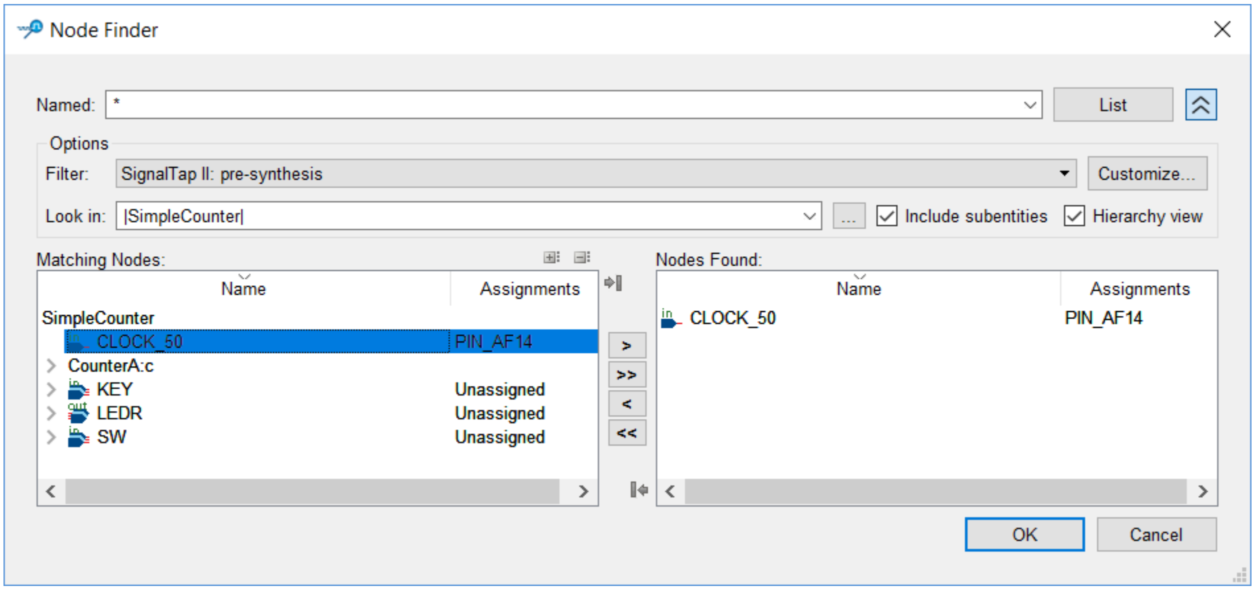
Hierarchy Display:

☑ SimpleCounter
☑ CounterA:c

□ Data Log:

auto_signaltap_0

In Node Finder, select CLOCK_50. Signals will be sampled on the rising edge of this clock.


Node Finder

Named: *

Options

Filter: SignalTap II: pre-synthesis

Look in: |SimpleCounter| … ☑ Include subentities ☑ Hierarchy view

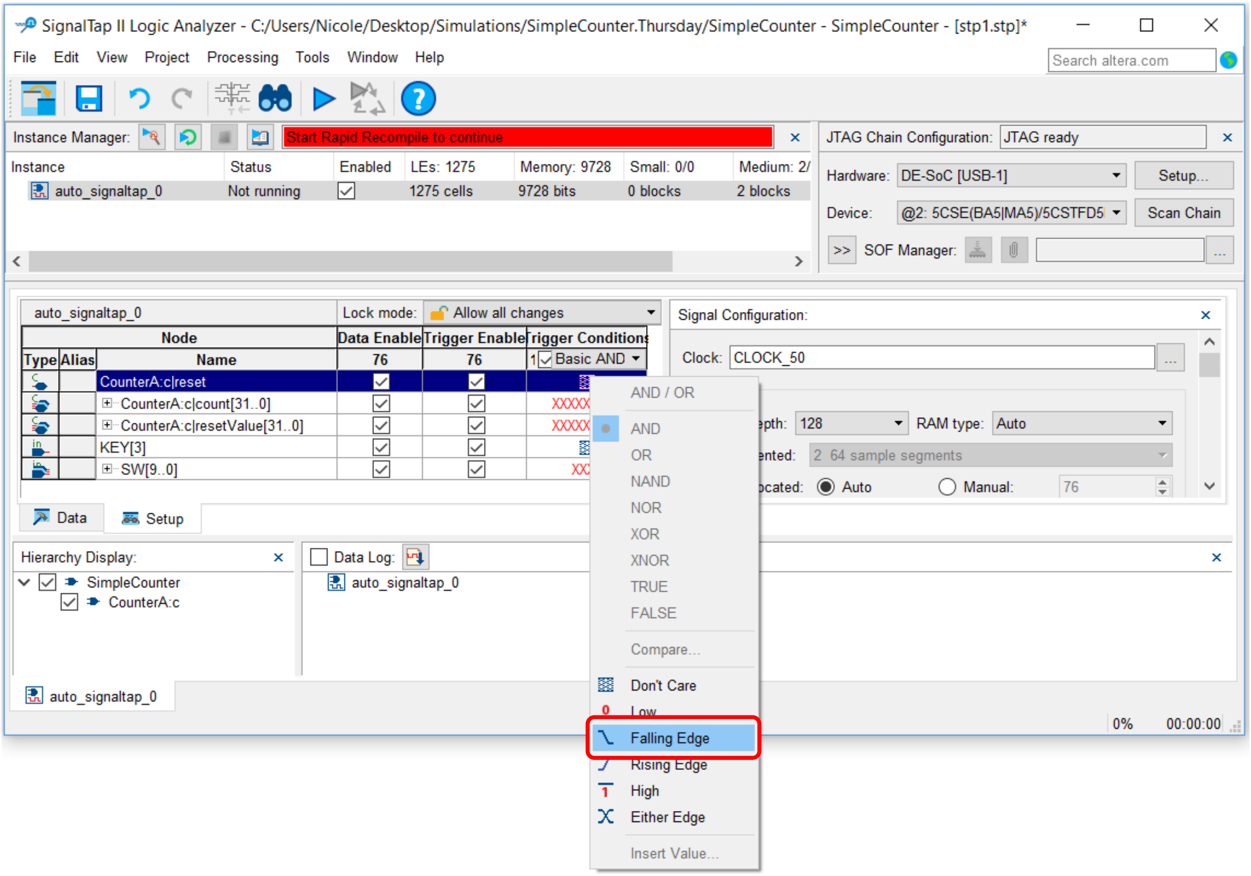
Matching Nodes:

Name	Assignments
SimpleCounter	
CLOCK_50	PIN_AF14
CounterA:c	
KEY	Unassigned
LEDR	Unassigned
SW	Unassigned

Nodes Found:

Name	Assignments
CLOCK_50	PIN_AF14

In the SignalTap II window, right-click and select “Falling Edge” as the trigger condition.



SignalTap II Logic Analyzer - C:/Users/Nicole/Desktop/Simulations/SimpleCounter.Thursday/SimpleCounter - SimpleCounter - [stp1.stp]*

Instance Manager: Start Rapid Recompile to continue

JTAG Chain Configuration: JTAG ready

Instance	Status	Enabled	LEs: 1275	Memory: 9728	Small: 0/0	Medium: 2/
auto_signaltap_0	Not running	☑	1275 cells	9728 bits	0 blocks	2 blocks

Hardware: DE-SoC [USB-1]

Device: @2: 5CSE(BA5|MA5)/5CSTFD5

auto_signaltap_0 Lock mode: Allow all changes

Type	Alias	Name	Data Enable	Trigger Enable	Trigger Conditions
			76	76	1 ☑ Basic AND
		CounterA:c|reset	☑	☑	▨
		⊞ CounterA:c|count[31..0]	☑	☑	XXXXX
		⊞ CounterA:c|resetValue[31..0]	☑	☑	XXXXX
		KEY[3]	☑	☑	▨
		⊞ SW[9..0]	☑	☑	XX

Signal Configuration:

Clock: CLOCK_50

Sample depth: 128 RAM type: Auto

Segmented: 2 64 sample segments

Nodes Allocated: ● Auto ○ Manual: 76

Hierarchy Display:

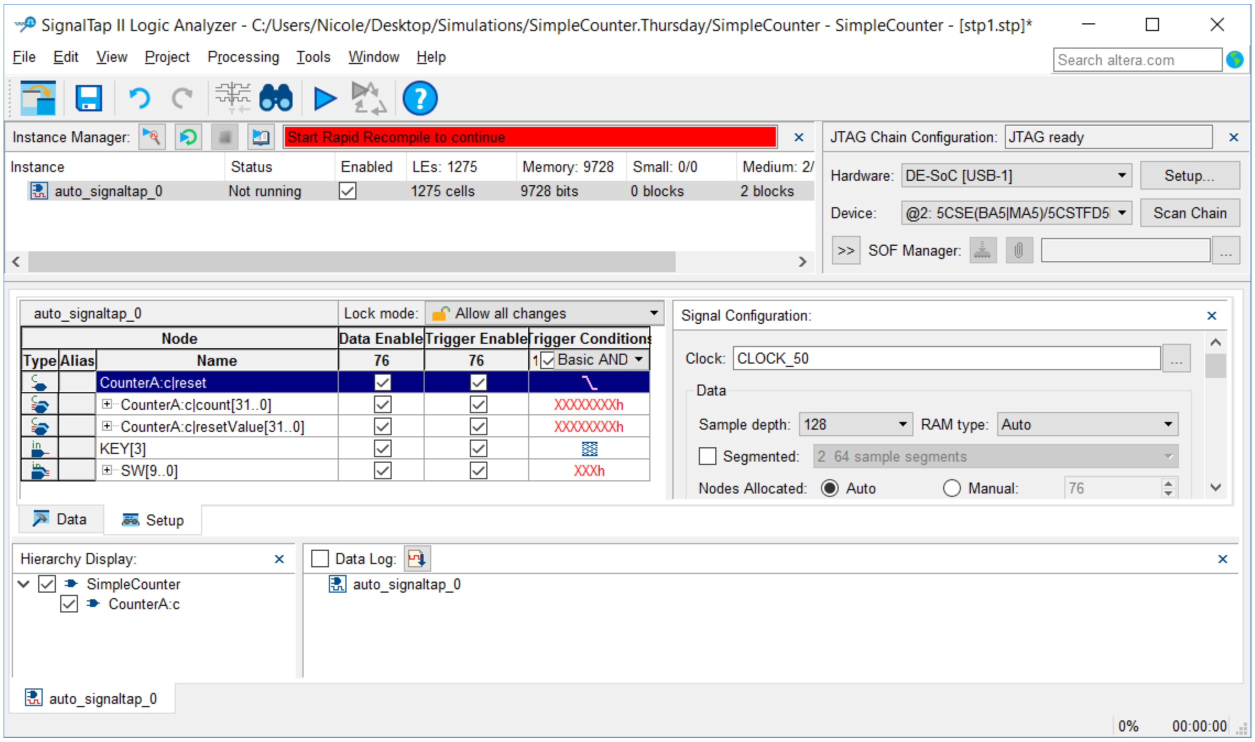
☑ SimpleCounter
☑ CounterA:c

☐ Data Log:

auto_signaltap_0

0% 00:00:00

Your result should look like this.



SignalTap II Logic Analyzer - C:/Users/Nicole/Desktop/Simulations/SimpleCounter.Thursday/SimpleCounter - SimpleCounter - [stp1.stp]*

Instance Manager: Start Rapid Recompile to continue

Instance	Status	Enabled	LEs: 1275	Memory: 9728	Small: 0/0	Medium: 2/
auto_signaltap_0	Not running	☑	1275 cells	9728 bits	0 blocks	2 blocks

JTAG Chain Configuration: JTAG ready

Hardware: DE-SoC [USB-1]

Device: @2: 5CSE(BA5|MA5)/5CSTFD5

SOF Manager:

auto_signaltap_0 Lock mode: Allow all changes

Type	Alias	Name	Data Enable 76	Trigger Enable 76	Trigger Conditions 1☑ Basic AND
		CounterA:c|reset	☑	☑	\
		⊞ CounterA:c|count[31..0]	☑	☑	XXXXXXXXh
		⊞ CounterA:c|resetValue[31..0]	☑	☑	XXXXXXXXh
		KEY[3]	☑	☑	▨
		⊞ SW[9..0]	☑	☑	XXXh

Signal Configuration:

Clock: CLOCK_50

Data

Sample depth: 128 RAM type: Auto

□ Segmented: 2 64 sample segments

Nodes Allocated: ● Auto ○ Manual: 76

Hierarchy Display:

☑ SimpleCounter

☑ CounterA:c

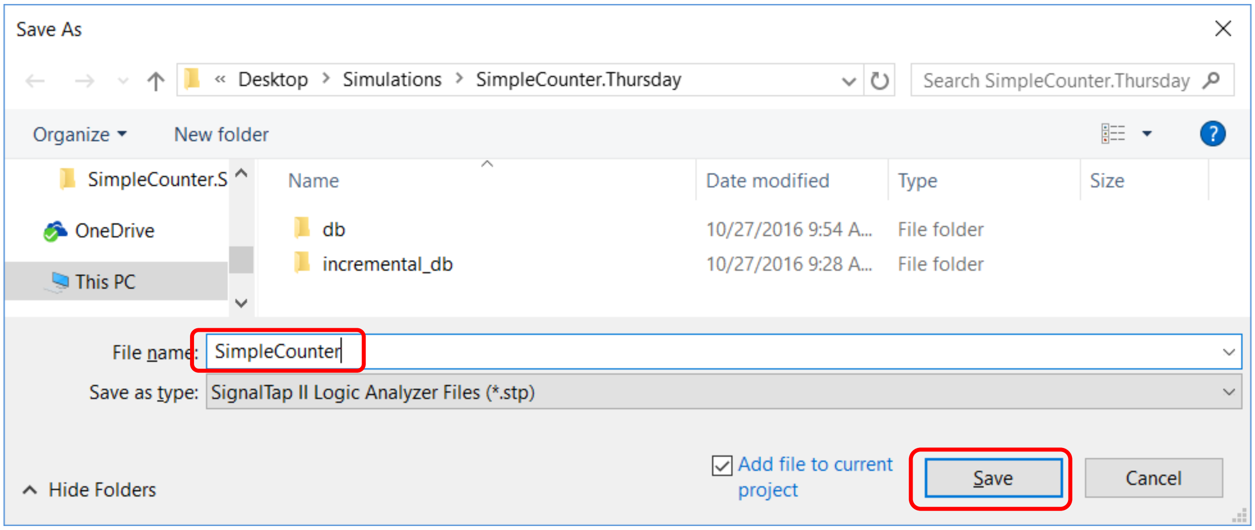
□ Data Log:

auto_signaltap_0

auto_signaltap_0

0% 00:00:00

File → Save as “SimpleCounter.stp”.



Save As

« Desktop › Simulations › SimpleCounter.Thursday

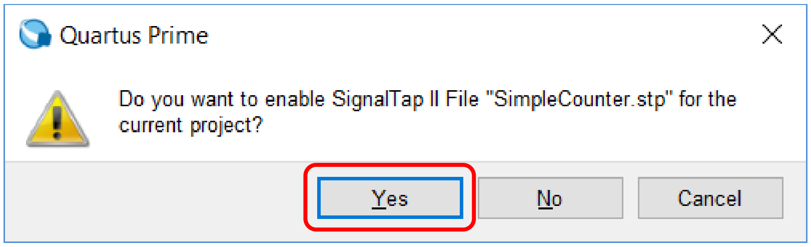
Name	Date modified	Type	Size
db	10/27/2016 9:54 A...	File folder	
incremental_db	10/27/2016 9:28 A...	File folder	

File name: SimpleCounter

Save as type: SignalTap II Logic Analyzer Files (*.stp)

☑ Add file to current project

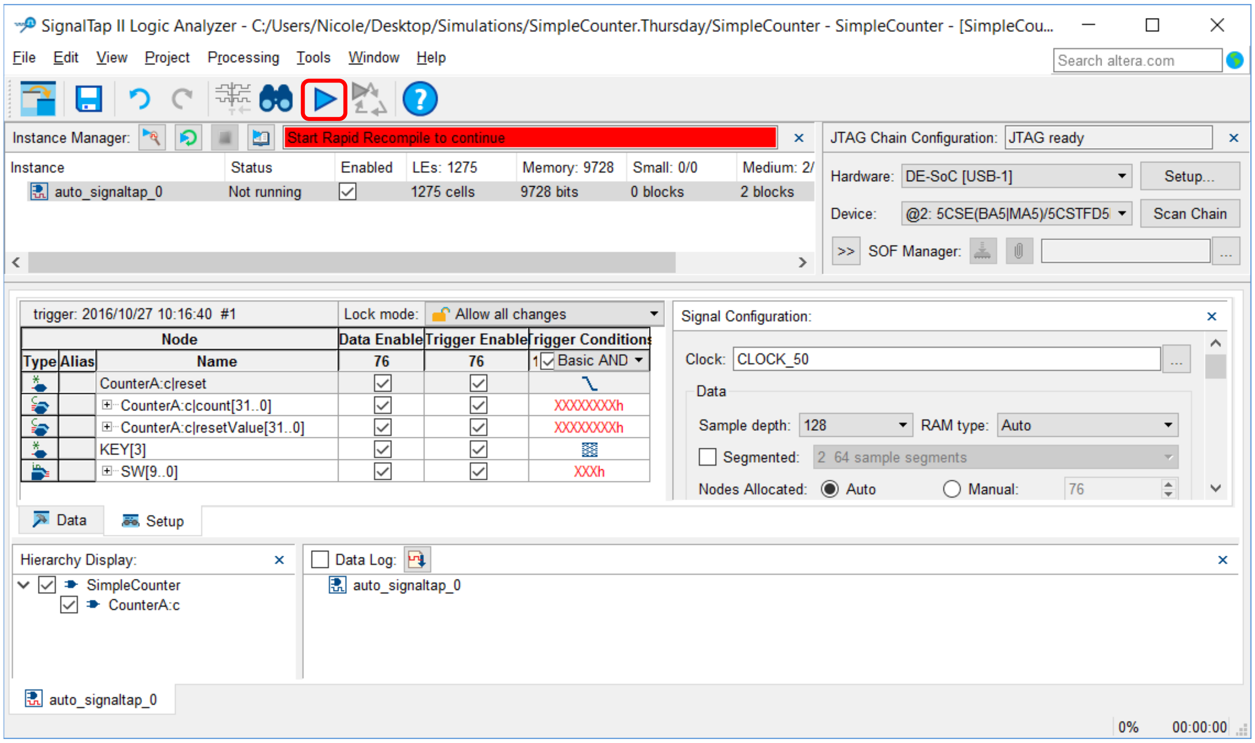
Yes, you do want to add this to your project.



Quartus Prime

Do you want to enable SignalTap II File "SimpleCounter.stp" for the current project?

Press the Compile button. This will trigger a complete recompile in Quartus.



SignalTap II Logic Analyzer - C:/Users/Nicole/Desktop/Simulations/SimpleCounter.Thursday/SimpleCounter - SimpleCounter - [SimpleCou...

Instance Manager: Start Rapid Recompile to continue

Instance	Status	Enabled	LEs: 1275	Memory: 9728	Small: 0/0	Medium: 2/
auto_signaltap_0	Not running	☑	1275 cells	9728 bits	0 blocks	2 blocks

JTAG Chain Configuration: JTAG ready

Hardware: DE-SoC [USB-1]

Device: @2: 5CSE(BA5|MA5)/5CSTFD5

trigger: 2016/10/27 10:16:40 #1

Lock mode: Allow all changes

Node			Data Enable	Trigger Enable	Trigger Conditions
Type	Alias	Name	76	76	1 ☑ Basic AND
		CounterA:c|reset	☑	☑	↘
		⊞ CounterA:c|count[31..0]	☑	☑	XXXXXXXXh
		⊞ CounterA:c|resetValue[31..0]	☑	☑	XXXXXXXXh
		KEY[3]	☑	☑	▨
		⊞ SW[9..0]	☑	☑	XXXh

Signal Configuration:

Clock: CLOCK_50

Data

Sample depth: 128 RAM type: Auto

☐ Segmented: 2 64 sample segments

Nodes Allocated: ● Auto ○ Manual: 76

Hierarchy Display:

☑ SimpleCounter
☑ CounterA:c

☐ Data Log:

auto_signaltap_0

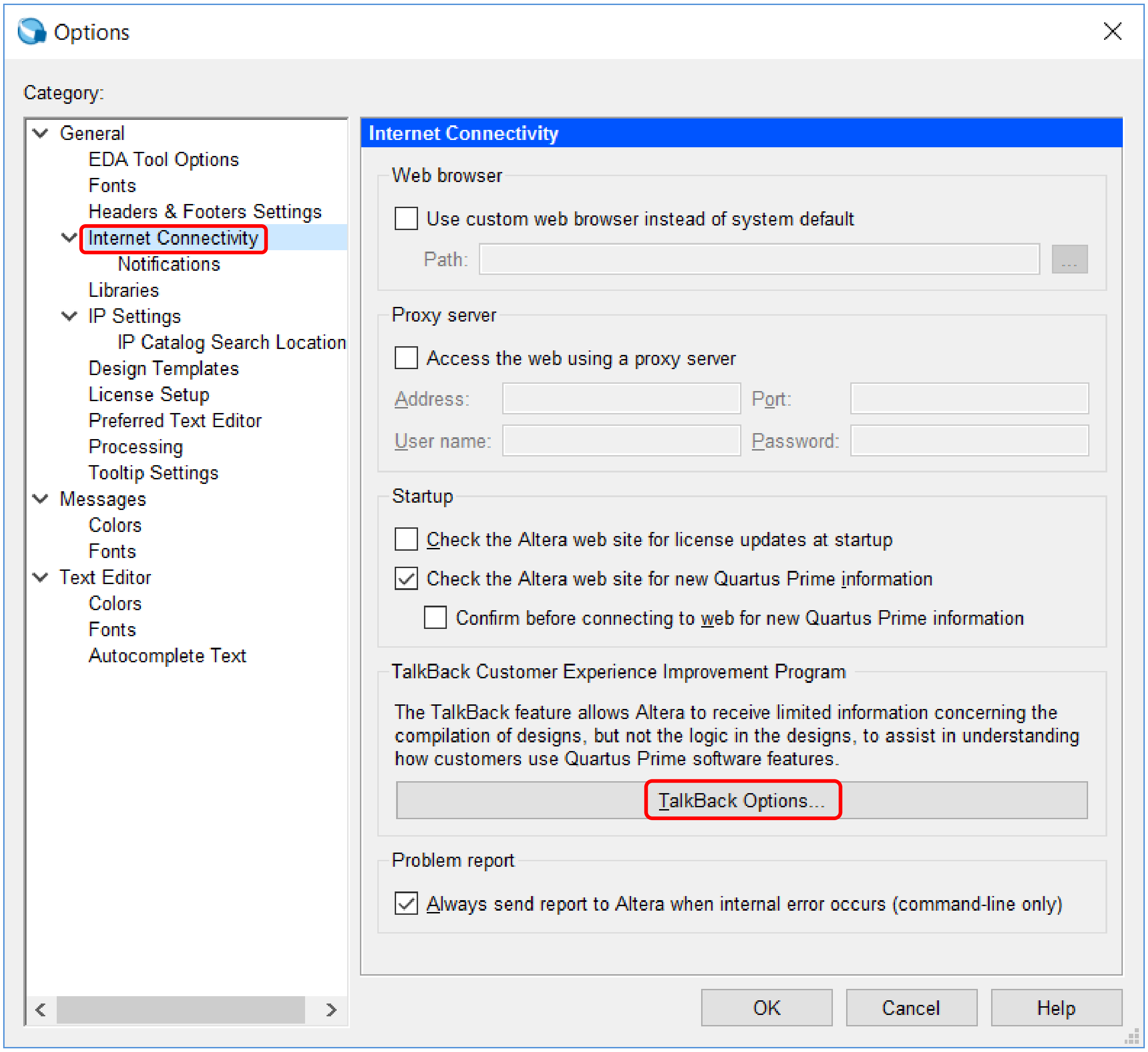
auto_signaltap_0

0% 00:00:00

If you get this error complaining about compiling with SignalTap in your project, you will need to enable TalkBack in Quartus.

```
265013 Can't open SignalTap II Logic Analyzer. Verify that the
license file exists and is stored in the correct location. If you are
using the Quartus Prime Lite Edition software, you must turn on the
TalkBack feature to use the SignalTap II Logic Analyzer.
```

This is buried in Tools → Options → Internet Connectivity from the Quartus main menu bar. Click “TalkBack Options…”.



Options

Category:

- General
  - EDA Tool Options
  - Fonts
  - Headers & Footers Settings
  - Internet Connectivity
    - Notifications
  - Libraries
  - IP Settings
    - IP Catalog Search Location
  - Design Templates
  - License Setup
  - Preferred Text Editor
  - Processing
  - Tooltip Settings
- Messages
  - Colors
  - Fonts
- Text Editor
  - Colors
  - Fonts
  - Autocomplete Text

Internet Connectivity

Web browser

☐ Use custom web browser instead of system default

Path:

Proxy server

☐ Access the web using a proxy server

Address: Port:

User name: Password:

Startup

☐ Check the Altera web site for license updates at startup

☑ Check the Altera web site for new Quartus Prime information

☐ Confirm before connecting to web for new Quartus Prime information

TalkBack Customer Experience Improvement Program

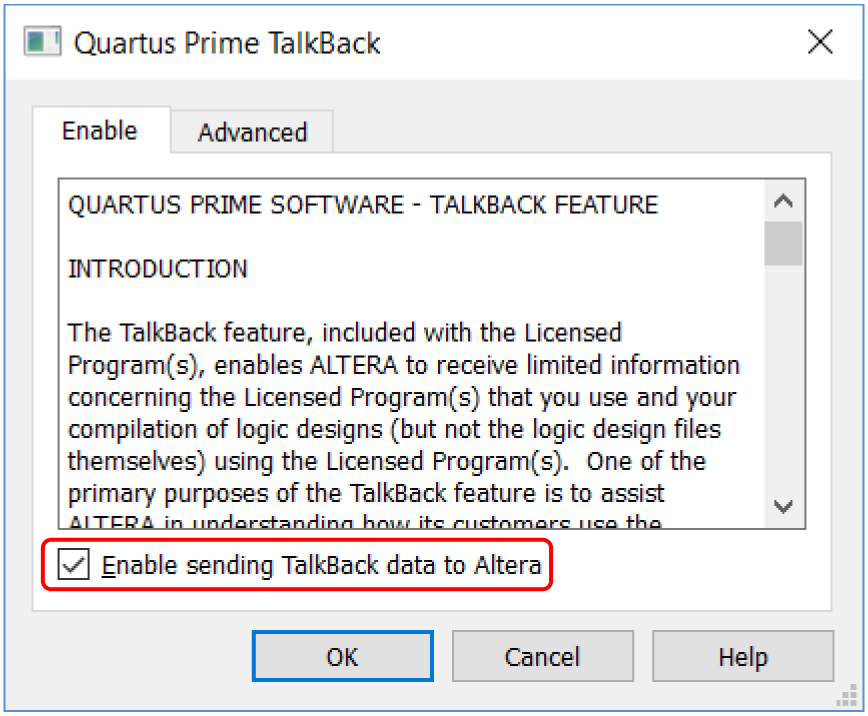
The TalkBack feature allows Altera to receive limited information concerning the compilation of designs, but not the logic in the designs, to assist in understanding how customers use Quartus Prime software features.

TalkBack Options...

Problem report

☑ Always send report to Altera when internal error occurs (command-line only)

Enable TalkBack, click OK twice to get back to Quartus, then retry the compile.



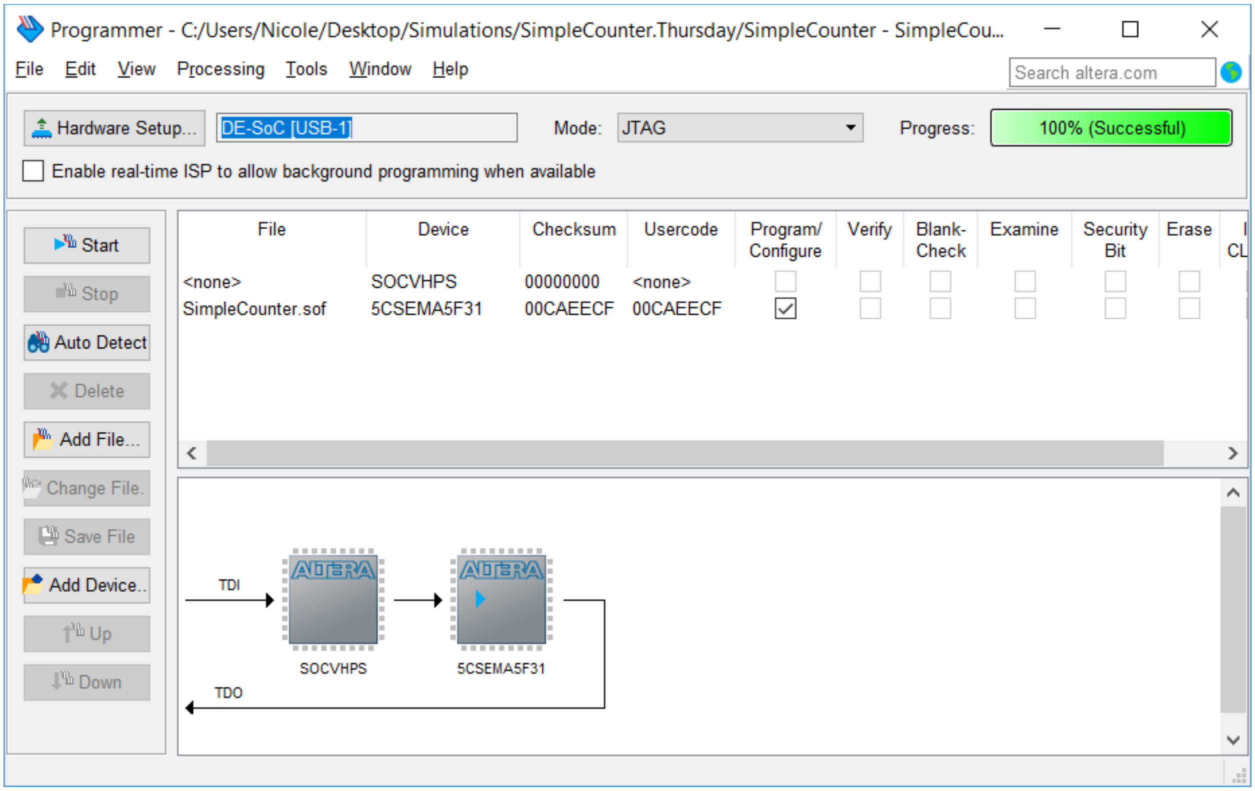
QUARTUS PRIME SOFTWARE - TALKBACK FEATURE

INTRODUCTION

The TalkBack feature, included with the Licensed Program(s), enables ALTERA to receive limited information concerning the Licensed Program(s) that you use and your compilation of logic designs (but not the logic design files themselves) using the Licensed Program(s). One of the primary purposes of the TalkBack feature is to assist ALTERA in understanding how its customers use the

☑ Enable sending TalkBack data to Altera

After the compile successfully finishes in Quartus, program the DE1-SoC.



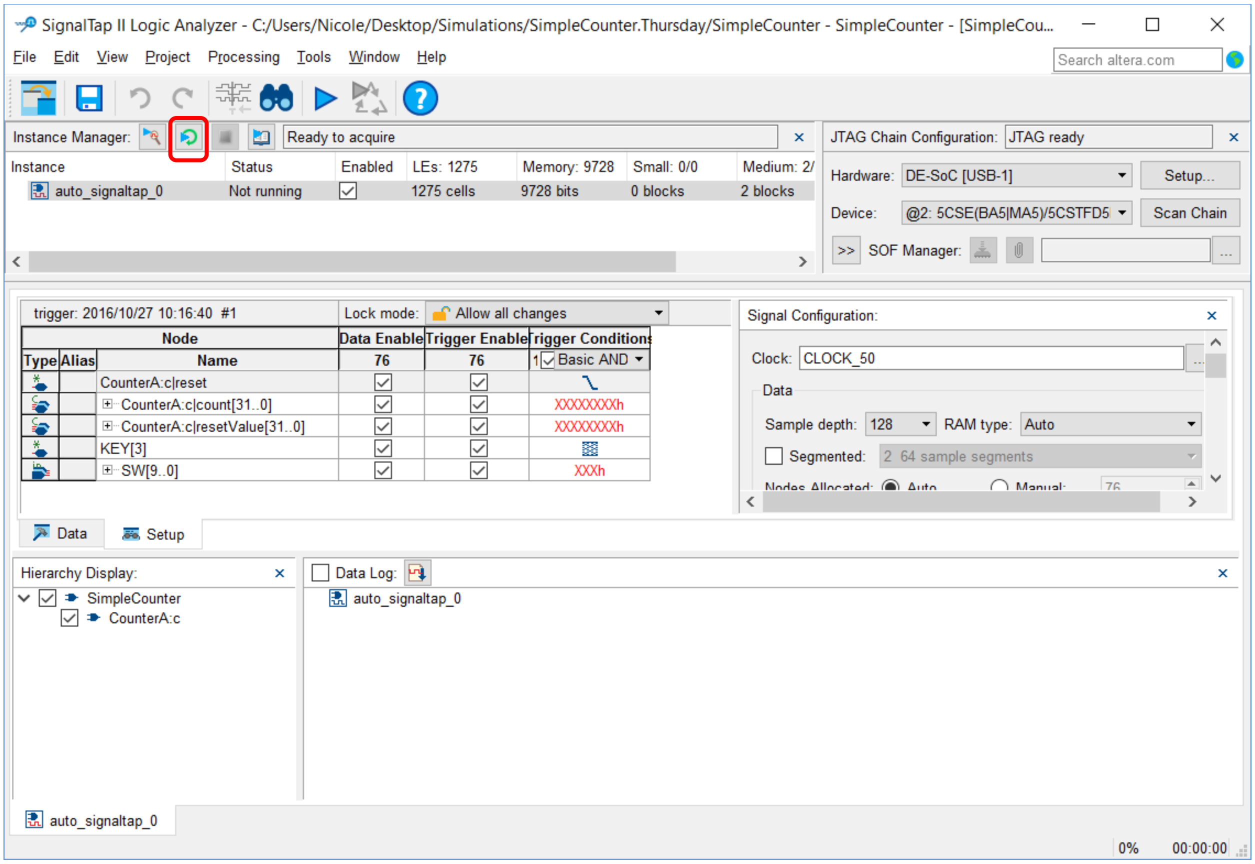
Programmer - C:/Users/Nicole/Desktop/Simulations/SimpleCounter.Thursday/SimpleCounter - SimpleCou...

Hardware Setup... DE-SoC [USB-1]   Mode: JTAG   Progress: 100% (Successful)

□ Enable real-time ISP to allow background programming when available

File	Device	Checksum	Usercode	Program/ Configure	Verify	Blank- Check	Examine	Security Bit	Erase	I CL
<none>	SOCVHPS	00000000	<none>	□	□	□	□	□	□	
SimpleCounter.sof	5CSEMA5F31	00CAEECF	00CAEECF	☑	□	□	□	□	□	

Close and then re-open SignalTap II and click the continuous button.



SignalTap II Logic Analyzer - C:/Users/Nicole/Desktop/Simulations/SimpleCounter.Thursday/SimpleCounter - SimpleCounter - [SimpleCou...

Instance Manager: Ready to acquire

Instance	Status	Enabled	LEs: 1275	Memory: 9728	Small: 0/0	Medium: 2/
auto_signaltap_0	Not running	☑	1275 cells	9728 bits	0 blocks	2 blocks

JTAG Chain Configuration: JTAG ready

Hardware: DE-SoC [USB-1]

Device: @2: 5CSE(BA5|MA5)/5CSTFD5

trigger: 2016/10/27 10:16:40 #1

Lock mode: Allow all changes

Node			Data Enable	Trigger Enable	Trigger Conditions
Type	Alias	Name	76	76	1 ☑ Basic AND
		CounterA:c|reset	☑	☑	⤥
		⊞ CounterA:c|count[31..0]	☑	☑	XXXXXXXXh
		⊞ CounterA:c|resetValue[31..0]	☑	☑	XXXXXXXXh
		KEY[3]	☑	☑	▨
		⊞ SW[9..0]	☑	☑	XXXh

Signal Configuration:

Clock: CLOCK_50

Data

Sample depth: 128 RAM type: Auto

□ Segmented: 2 64 sample segments

Nodes Allocated: ● Auto ○ Manual: 76

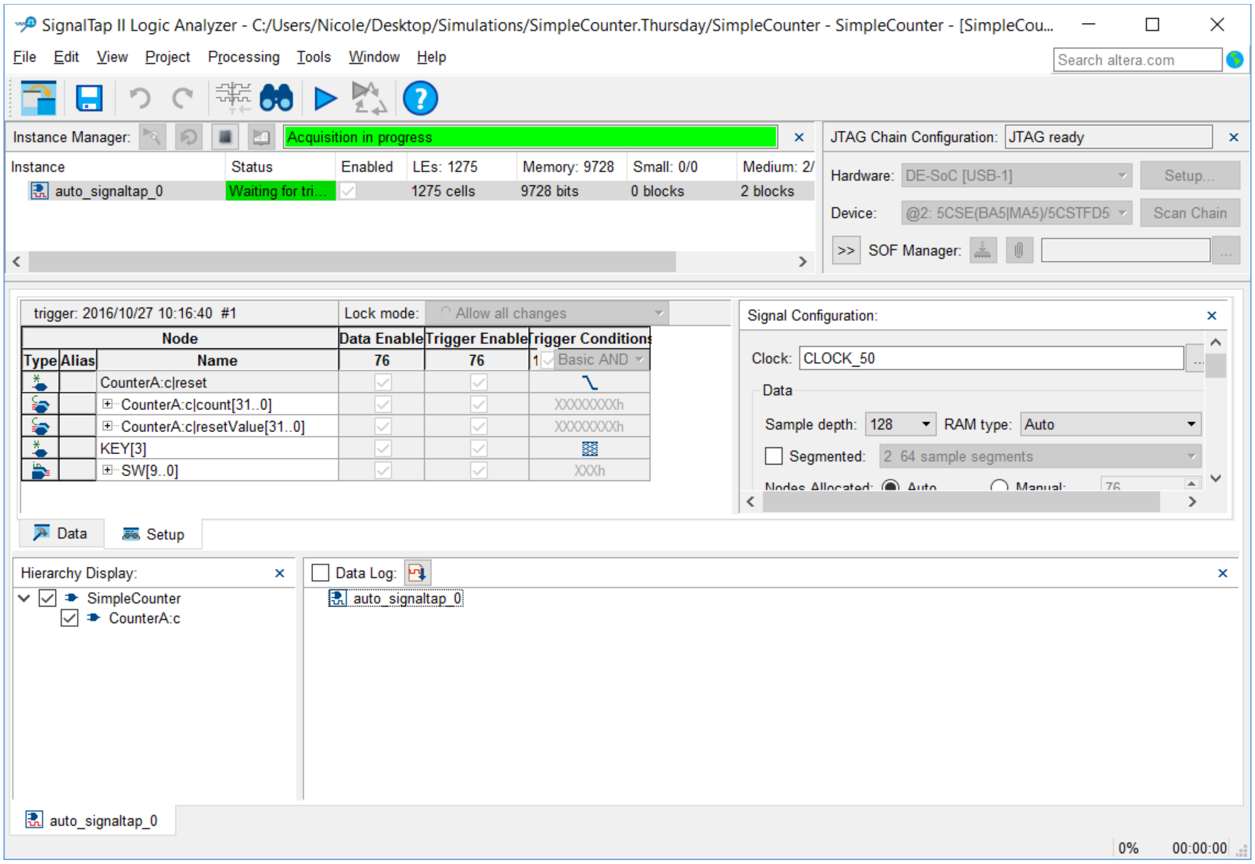
Hierarchy Display:

☑ SimpleCounter
☑ CounterA:c

□ Data Log:

auto_signaltap_0

It will then switch to acquisition mode.



SignalTap II Logic Analyzer - C:/Users/Nicole/Desktop/Simulations/SimpleCounter.Thursday/SimpleCounter - SimpleCounter - [SimpleCou...

Instance Manager: Acquisition in progress

JTAG Chain Configuration: JTAG ready

Instance	Status	Enabled	LEs: 1275	Memory: 9728	Small: 0/0	Medium: 2/
auto_signaltap_0	Waiting for tri...	✓	1275 cells	9728 bits	0 blocks	2 blocks

Hardware: DE-SoC [USB-1]

Device: @2: 5CSE(BA5\|MA5)/5CSTFD5

SOF Manager:

trigger: 2016/10/27 10:16:40 #1

Lock mode: Allow all changes

Node			Data Enable	Trigger Enable	Trigger Conditions
Type	Alias	Name	76	76	1 ✓ Basic AND
		CounterA:c|reset	✓	✓	⌄
		⊞ CounterA:c|count[31..0]	✓	✓	XXXXXXXXh
		⊞ CounterA:c|resetValue[31..0]	✓	✓	XXXXXXXXh
		KEY[3]	✓	✓	▨
		⊞ SW[9..0]	✓	✓	XXXh

Signal Configuration:

Clock: CLOCK_50

Data

Sample depth: 128 RAM type: Auto

□ Segmented: 2 64 sample segments

Nodes Allocated: ● Auto ○ Manual: 76

Hierarchy Display:

☑ SimpleCounter
☑ CounterA:c

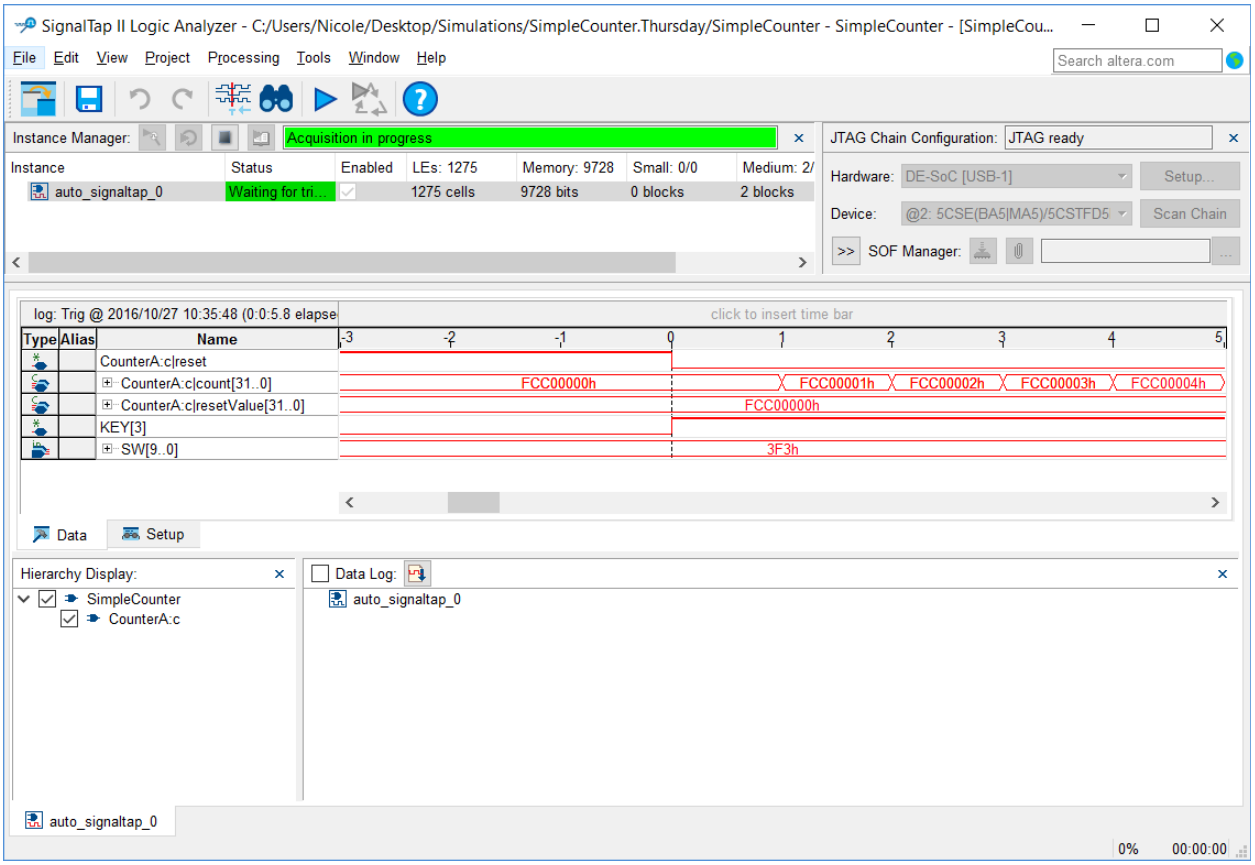
□ Data Log:

auto_signaltap_0

auto_signaltap_0

0% 00:00:00

Each time the reset key is pressed, the captured data is displayed. (The hex 3F3 is what happened to be entered on the switches when I pressed reset.)


SignalTap II Logic Analyzer - C:/Users/Nicole/Desktop/Simulations/SimpleCounter.Thursday/SimpleCounter - SimpleCounter - [SimpleCou...
File
Edit
View
Project
Processing
Tools
Window
Help
Search altera.com
Data
Setup
Setup...
Scan Chain
>>
-3
-2
-1
0
1
2
3
4
5

Instance Manager: Acquisition in progress

Instance	Status	Enabled	LEs: 1275	Memory: 9728	Small: 0/0	Medium: 2/
auto_signaltap_0	Waiting for tri...	✓	1275 cells	9728 bits	0 blocks	2 blocks

JTAG Chain Configuration: JTAG ready

Hardware: DE-SoC [USB-1]

Device: @2: 5CSE(BA5|MA5)/5CSTFD5

SOF Manager:

log: Trig @ 2016/10/27 10:35:48 (0:0:5.8 elapse

click to insert time bar

Type	Alias	Name	
		CounterA:c|reset	
		⊞ CounterA:c|count[31..0]	FCC00000h, FCC00001h, FCC00002h, FCC00003h, FCC00004h
		⊞ CounterA:c|resetValue[31..0]	FCC00000h
		KEY[3]	
		⊞ SW[9..0]	3F3h

Hierarchy Display:

☑ SimpleCounter

☑ CounterA:c

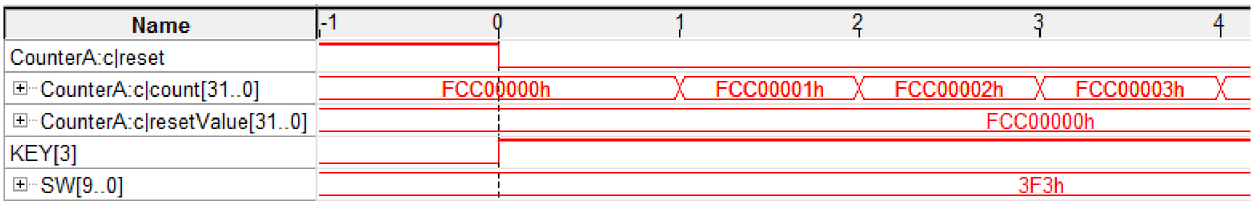
□ Data Log:

auto_signaltap_0

auto_signaltap_0

0% 00:00:00

Zooming in:



Name	
CounterA:c|reset	
⊞ CounterA:c|count[31..0]	FCC00000h, FCC00001h, FCC00002h, FCC00003h
⊞ CounterA:c|resetValue[31..0]	FCC00000h
KEY[3]	
⊞ SW[9..0]	3F3h

On the falling edge of reset, as you release the button, it starts counting from the value entered via the switches.

To end the session, press Esc. Occasionally, I find that SignalTap hangs and will not exit. Unplugging the DE1-SoC will get it unstuck.