


# **XILINX FOUNDATION SIMULATION TUTORIAL**


## Exercise 1:

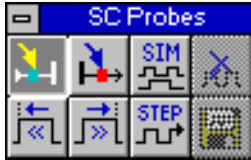
**Topics:** Selecting signals for simulation, assigning stimuli to signals, setting simulation parameters


**Software:** Xilinx Foundation 6.0.1 with CALC sample project


1. Start the *Project Manager* and open the CALC project (click , select CALC in the *Open Project* window, and click on **Open**).

2. Start the *Schematic Editor* (click  in *Design Flow* section).

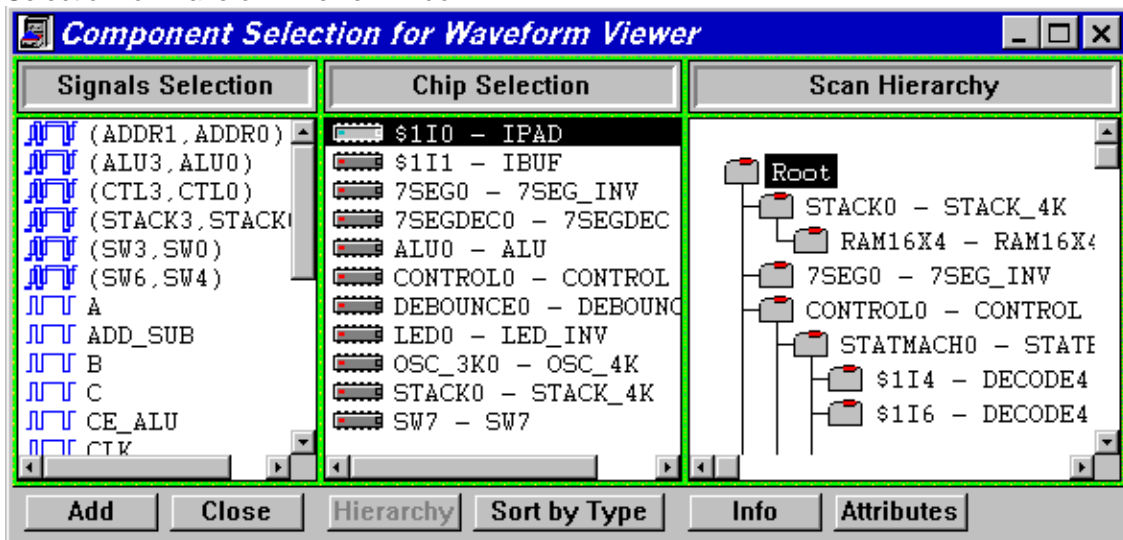
3. Click on the *Simulation Toolbox* toolbar button  - the *SC Probes* toolbox will appear.




4. Make sure that the *Select Probe* () button is pressed and put probes on the CLK, EXC\_P, ALU[3:0] and STACK[3:0] nets (click on the net label to add a probe).

5. Click on the  button (either in the toolbar or in the *SC Probes* toolbox) to start the simulator.

6. Click on the  button (or select **Signal | Add Signal**) to invoke the *Component Selection for Waveform Viewer* window.



Double-click on the **SW7** chip in the *Chip Selection* column; double-click on SW0 to SW6 pins in the right column to add these pins to the Waveform Viewer.


7. Click on the **Hierarchy** button to return to the hierarchy display in the right column.
8. Select LED0 - LED\_INV in the *Scan Hierarchy* column, then double click on the LED0\_P, LED1\_P, LED2\_P, and LED3\_P signals in *Signal Selection* window.
9. Close the *Component Selection...* window.
10. Click on the  button (or select **Utilities | View | Hex Buses**) to display the ALU and STACK buses in hexadecimal form.

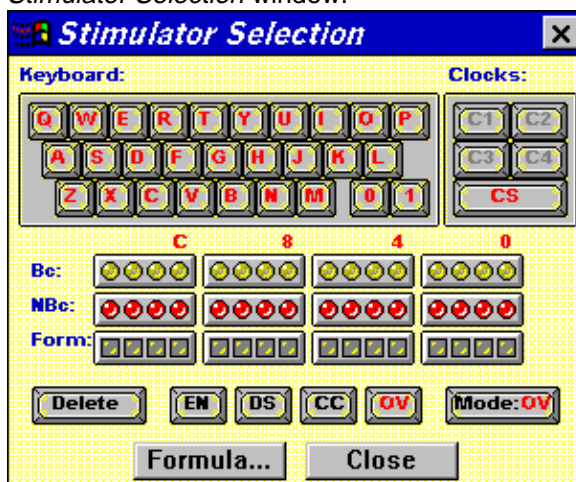
11. Rearrange the signals in the *Waveform Viewer* window by dragging the ALU and STACK buses to a position between SW.SW6 and LED0\_P. The final signal list should look like this:

```

1 CLK
1 EXC_P
0 SW7.SW0 -
0 SW7.SW1 -
0 SW7.SW2 -
0 SW7.SW3 -
0 SW7.SW4 -
0 SW7.SW5 -
0 SW7.SW6 -
B ALU3 (hex)#4
B STACK3 (hex)#4
1 LED0_P
1 LED1_P
1 LED2_P
1 LED3_P

```

12. Click on the CLK signal name in the *Waveform Viewer* window; the text background should change to cyan to indicate that this signal has been selected.
13. Click on the  button (or select **Stimulator | Add Stimulators**) to display the *Stimulator Selection* window.



Click on the third diode from the right in the **Bc:** row. This assigns the B2 stimulator to the selected signal.

#### NOTE:

- The **Bc:** row represents the non-inverted bits of the 16-bit binary counter visible at the top of simulator window.
  - The **NBc:** row represents the inverted bits of this counter.
  - The **Form:** row represents formula stimulators.
- Stimulators are referenced by row (**B**, **N**, or **F**) and column (**F** to **0**); hence the third diode from the right in the **Bc:** row is referenced as **B2**.

14. Drag the **E** key from the keyboard in the *Stimulator Selection* window onto the EXC\_P signal name in the *Waveform Viewer* window.
15. Drag the **Z,X,C,V,B,N,M** buttons from the keyboard in the *Stimulator Selection* window onto SW7.SW0 to SW7.SW6 signal names in the *Waveform Viewer* window.

16. Close the *Stimulator Selection* window. The signal list should look like this:

1	CLK		B2	
1	EXC_P		e	
<input type="checkbox"/>	SW7.SW0	-	z	
<input type="checkbox"/>	SW7.SW1	-	x	
<input type="checkbox"/>	SW7.SW2	-	c	
<input type="checkbox"/>	SW7.SW3	-	v	
<input type="checkbox"/>	SW7.SW4	-	b	
<input type="checkbox"/>	SW7.SW5	-	n	
<input type="checkbox"/>	SW7.SW6	-	m	
	BALU3 (hex)#4		X	
	BSTACK3 (hex)#		X	
1	LED0_P			
1	LED1_P			
1	LED2_P			
1	LED3_P			

17. By clicking on the red stimulus letter to the right of the signal name (or by typing the letter on the keyboard), set following logical states:

EXC\_P, SW7.SW1, SW7.SW2, SW7.SW3, SW7.SW5, SW7.SW6 - **High**,  
SW7.SW0, SW7.SW4 - **Low**

**NOTE:**

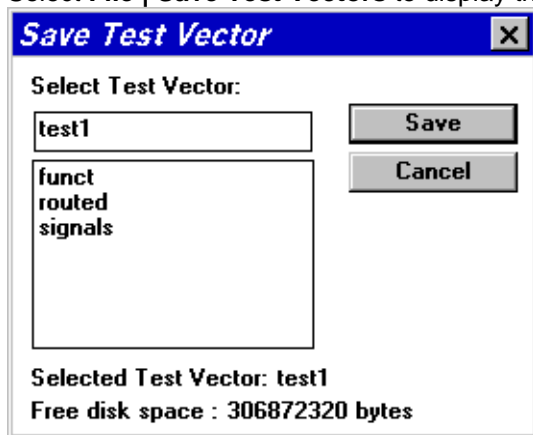
These settings prepare to load the hex value E into the calculator register.

18. Select **Utilities | Simulation Step** to display the *Step* window.



Type **40ns** in the **Short Step:** box, then click on **Set Step** (**Long Step:** should be 200ns).

19. Select **File | Save Test Vectors** to display the *Save Test Vector* window.

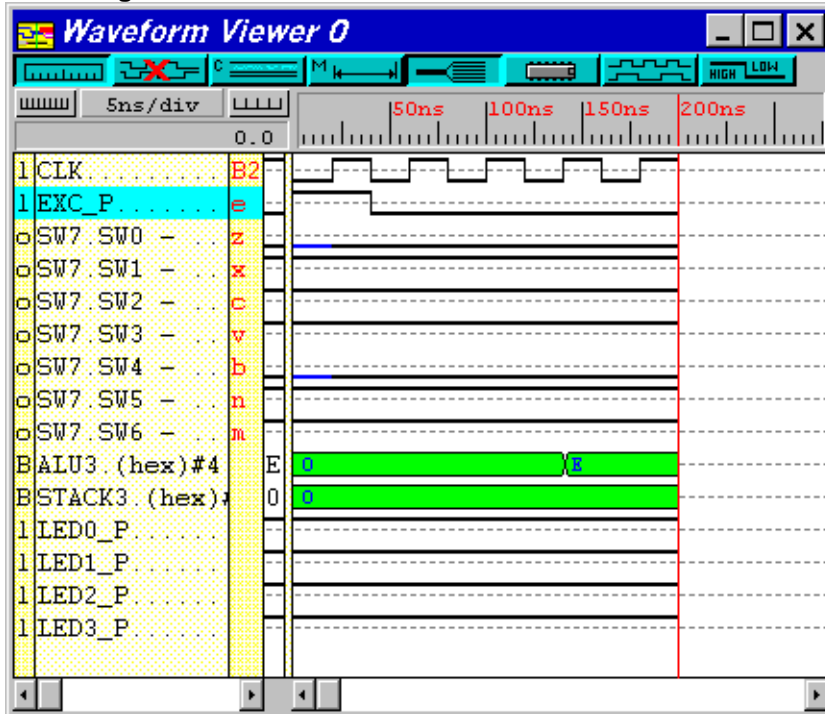


Type **test1** in the **Select Test Vector** box and click **Save**.

20. Click on the **Step** button in *Simulator* toolbox.




21. Click on the letter **e** stimulus displayed to the right of the EXC\_P signal name to switch it to **Low**.
22. Click **Long** in the *Simulator* toolbox. The *Waveform Viewer* window should look like this:



23. Set following logical states on the inputs:  
 EXC\_P, SW7.SW1, SW7.SW3, SW7.SW4, SW7.SW5, SW7.SW6 - **High**,  
 SW7.SW0, SW7.SW2 - **Low**



**NOTE:**

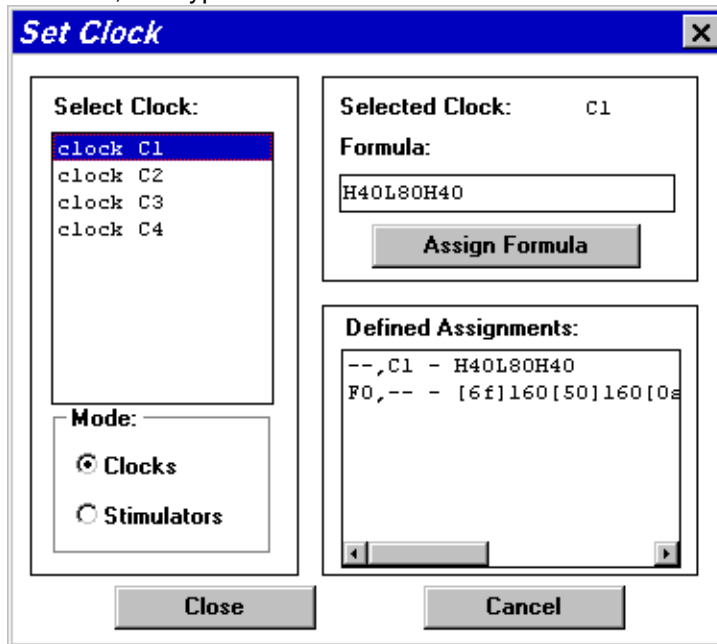
These settings prepare for the 'push calculator register to stack' operation

24. Repeat steps 20, 21, and 22.
25. Save your simulation results as *result1* (use **File | Save Test Vectors**).
26. Select all the output signals in the *Waveform Viewer* window (click on the ALU bus, then hold down the SHIFT key and click on LED3\_P)
27. Select **Signal | Delete | Waveforms of Selected Signal Names**.
28. Select the EXC\_P signal and SW.SW0 to SW7.SW6 signals.
29. Click on  to display the *Stimulator Selection* window and click **CS**.

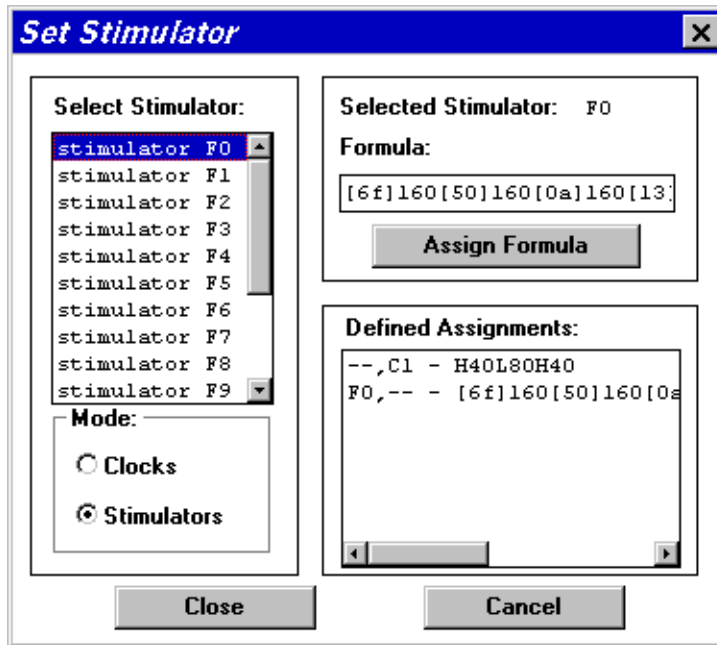
**NOTE:**

This operation assigns the waveforms that are currently displayed to the right of the selected signals as the stimuli for future simulations.

30. Save your test vectors as *test2* (use **File | Save Test Vectors**).
31. Close the *Simulator* and the *Schematic Editor*.
32. Click on the  button in the *Project Manager* to restart the simulator.
33. Load the *test2* test vectors (**File | Load Test Vectors**).
34. In the *Simulator* toolbox, click **Long** twice.
35. Select the SW.SW0 to SW7.SW6 signals and select **Signal | Bus | Create**.
36. Select **Signal | Delete | All Waveforms with Power On**.
37. Click  to display the *Stimulator Selection* window and click on the **Formula...** button.
38. In the *Set Stimulator* window select **Clocks** mode. Double-click on *clock C1* in the **Select clock** list, and type *H40L80H40* in the **Formula:** box. Click on the **Assign Formula** button.



39. Select **Stimulators** mode. Double-click on *stimulator F0* in the **Select Stimulator** list, and type *[6f]160[50]160[0a]160[13]160[24]160[3f]160[7a]160* in the **Formula:** box. Click on **Assign Formula**. Click on **Close**.


**NOTE:**

The microprogram listed below has just been defined:

```
[6f]  LOAD  #F
[50]  CLEAR
[0a]  ADD   #A
[13]  AND   #3
[24]  OR    #4
[3f]  XOR   #F
[7a]  PUSH
```

40. Drag the **C1** button from the **Clocks:** box in the *Stimulator Selection* window onto the EXC\_P signal name in the *Waveform Viewer* window.
41. Drag the **F0** diode from the **Form:** row onto the SW7.SW[0:6] bus.
42. Close the *Stimulator Selection* window.
43. Add the CTL[3:0] bus from the *Root* hierarchy level to the list of signals in the *Waveform Viewer* window (use procedure in point 8).
44. Select the CTL[3:0] bus in the *Waveform Viewer* window and select **Signal | Bus | Destroy**.
45. Simulate for 800ns (4 Long steps).
46. Holding down the CTRL key, select the CTL3, CTL2, CTL1, and CTL0 signals (order is important!) and create a bus.
47. Click on the CTL bus and select **Signal | Bus | Bus Direction** to achieve the correct display.

**NOTE:**

To verify which bus member is displayed as the LSB, expand the bus display to individual lines (deselect the  button ) and look for an asterisk (\*) displayed to the right of signal name. The asterisk marks the LSB, and a plus sign (+) marks other bus members


48. Simulate to 1.2μs (2 more Long steps).
49. Select different **Signal | Bus | Display xxx** options to view the CTL bus in binary, octal, decimal, or hexadecimal base.
50. Close the *Simulator* window.

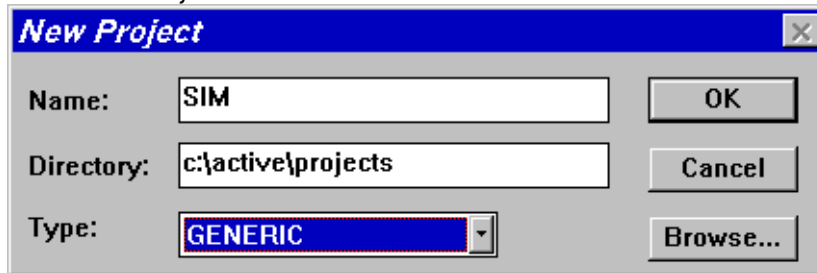
## Exercise 2:

**Topics:** Using the *Load Netlist* option, timing violations

**Software:** Xilinx Foundation 6.0.1 with CALC sample project

1. In the *Project Manager*, create a generic project called SIM:


- Click on the  button (or select **File | New Project**).
- In the *New Project* window

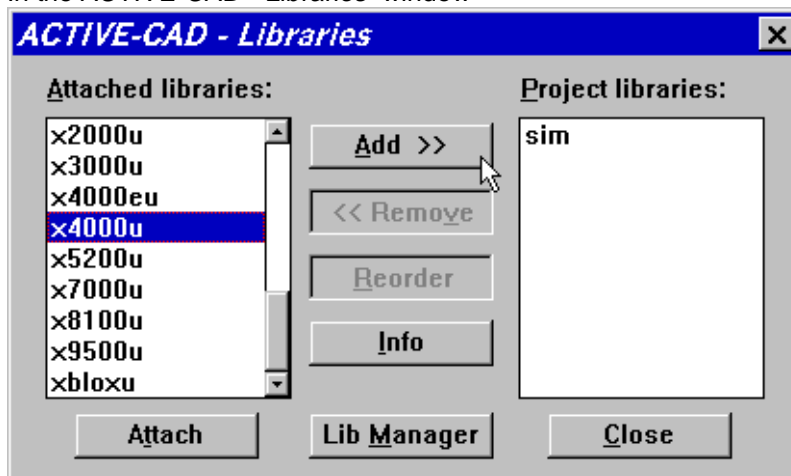


select *GENERIC* in the **Type:** listbox and type *SIM* in the **Name:** box.

- Click **OK**.


2. Add the X4000U library to the project resources:

- Click on the  button (or select **File | Libraries**).
- In the *ACTIVE-CAD - Libraries* window



select the X4000U library in the **Attached libraries:** listbox and click **Add >>**.


- Click **Close**.

3. Run the *Simulator* (select **Applications | Simulator**).
4. Load the *calc.bax* netlist from the CALC project (select **File | Load Netlist**, browse to the CALC project, and select *calc.bax*).
5. Click on the  button (or select **Signal | Add Signal**) to open the *Component Selection for Waveform Viewer* window.
6. Double-click on the following signal names:
  - CLK
  - EXC\_P
  - (ALU0,ALU3)





(STACK0,STACK3)

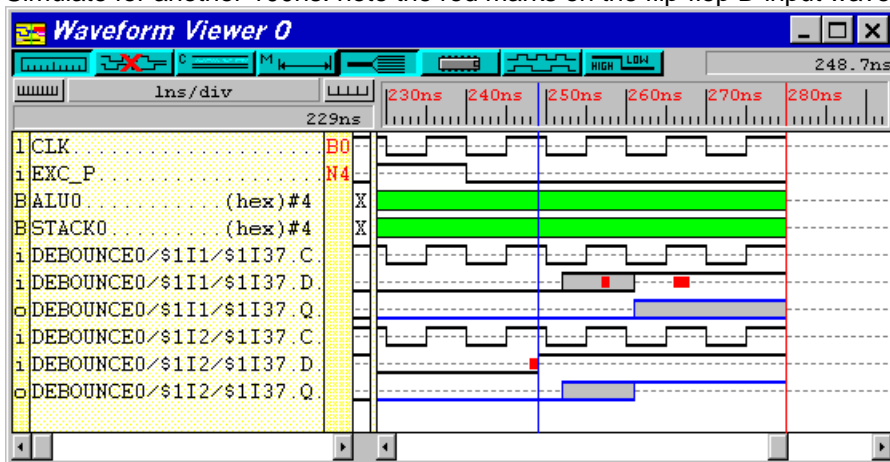
then close the *Component Selection for Waveform Viewer* window.

7. Click on the  button (or select **Stimulator | Add Stimulators**) to display the *Stimulator Selection* window and assign the **B0** stimulator to CLK and **N4** to EXC\_P.
8. Select Timing mode of simulation (click on the **Mode: FN** button in the *Simulator* toolbox

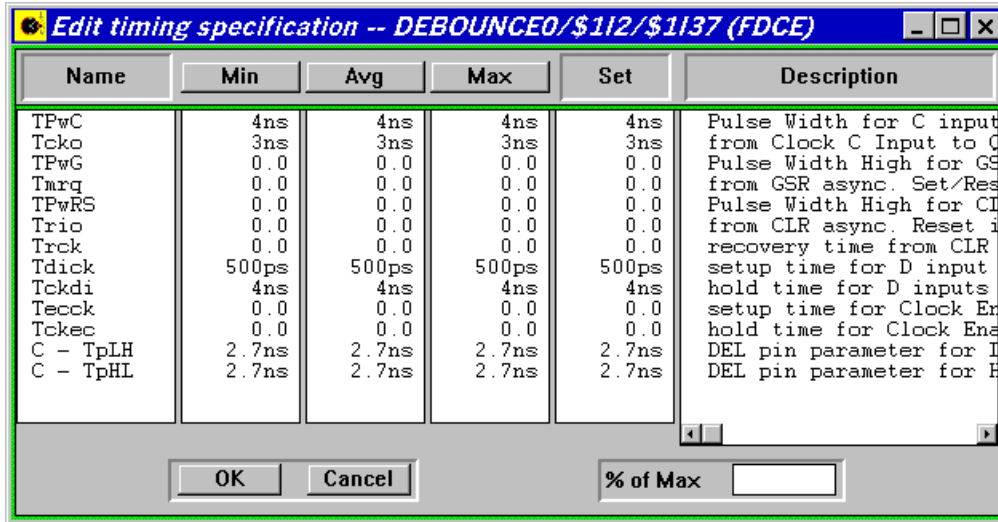


until **Mode: TM** is displayed)

9. Save test vectors as *signals.tve* (select **File | Save Test Vectors** and type *signals* in the **Select Test Vector** box).
10. Simulate for 200ns using the **Step** or **Long** buttons in the *Simulator* toolbox: note that setup/hold violations are reported.
11. Open the *Errors Viewer* window (select **Utilities | Error Viewer**).  
Note that there are recurring violations inside the DEBOUNCE0/\$1I1 and DEBOUNCE0/\$1I2 macros.
12. Select these pins for viewing: C, D and Q pins of DEBOUNCE0/\$1I1/S1I37 and C, D and Q pins of DEBOUNCE0/\$1I2/S1I37.
13. Change the waveform display scale to 1ns/div (use the  and  buttons placed above signal names and until 1ns/div is displayed between them).
14. Simulate for another 100ns: note the red marks on the flip-flop D input waveforms.



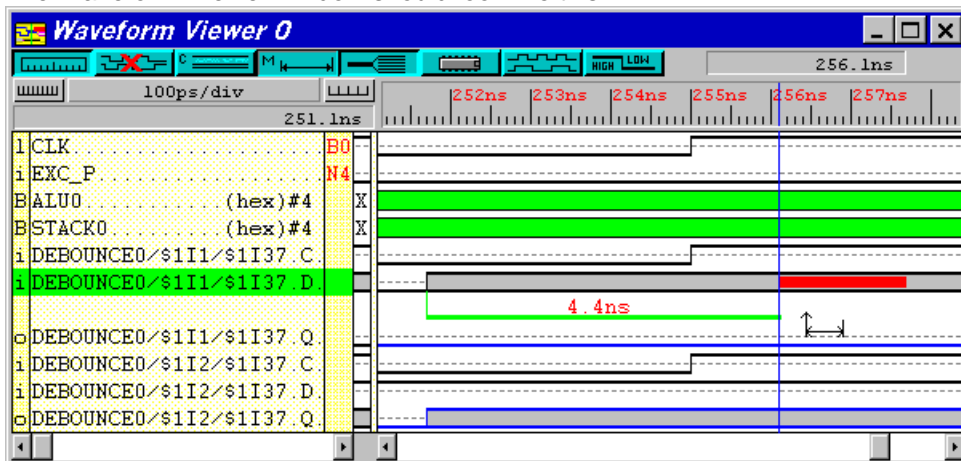
15. Click on the right side of the red rectangle marking the hold time violation on DEBOUNCE0/\$1I2/S1I37.D and change the waveform display scale to 100ps/div.
16. Select **Patching | Edit Timing Specification** and double-click on DEBOUNCE0/\$1I2/S1I37 to display the *Edit timing specification* window.

**NOTE:**

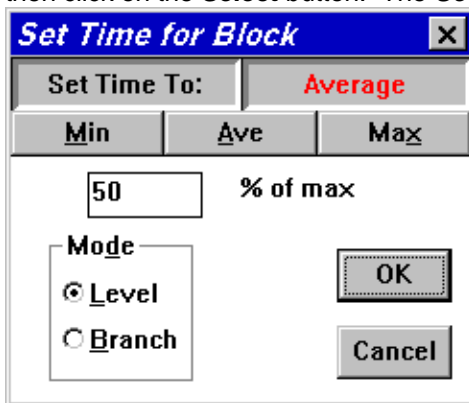
- Flip-flop model parameters, such as setup and hold times, are stored in the system libraries.
- Routing delays are added by the XACTstep software to the backannotated netlist as delay (DEL) parameters.
- During the netlist loading process, the simulator loads parameters from the libraries first, then adds the DEL parameters to the timing specification; both types of parameters are visible in the *Edit timing specification* window.
- Setup/hold time violations are marked by red rectangles anchored at the **delayed** clock edge and the **delayed** offending event on the data input.
- Due to those delays, the red rectangle is usually displaced to the right from the clock and data events that caused the violation.

17. Close the *Edit timing specification* window and switch to the *Waveform Viewer* window.
18. Scroll the Waveform Viewer display to show the nearest setup time violation on DEBOUNCE0/\$1I1/S1I37.D.
19. Verify the timing specification of DEBOUNCE0/\$1I1/S1I37 (see instructions in point 16).
20. Measure the time interval between the Low to Unkn\_X data transition and the left edge of the red rectangle:
  - Select **Waveform | Measurements | Measurements On ...**
  - Click on the Low to Unkn\_X waveform edge with the vertical arrow of the mouse pointer (note that pointer shape changes afterwards).
  - Click on the left edge of the red rectangle with the vertical arrow of the mouse pointer.

The *Waveform Viewer* window should look like this:



21. Measure the time interval between the rising clock edge and the right edge of the red rectangle.
22. Verify that the measured intervals are equal to the pin delays listed in the timing specification.
23. Change the *T<sub>dick</sub>* setup time to 0 in the timing specification of DEBOUNCE0/\$1I1/S1I37:
  - Double-click on *3ns* in the *T<sub>dick</sub>* row and *Set* column.
  - Type *0*.
  - Press **Enter**.
  - Click **OK**.
24. Simulate for another 100ns: note that the setup violations have disappeared.
25. Select **Patching | Edit Timing Specification**. Click on *Root* in the *Chip Selection* section, then click on the **Select** button. The *Set Time for Block* window is displayed:




26. In the *Set Time for Block* window, enter *50* in the *% of max* field and click **OK**.
27. Simulate for another 100ns. All timing violations should disappear.
28. Close the *Simulator* window.

### Exercise 3:


**Topics:**        **Resetting simulation**

**Software:**    **Xilinx Foundation 6.0.1 with FLASH and CALC7K sample projects**

1. Open the FLASH sample project (**File | Open Project** in *Project Manager*).
2. Start the *Simulator* (**SIM Funct** button).
3. Load the *Signals* test vectors (**File | Load Test Vectors**).
4. Switch on hexadecimal bus display (click on the  button or select **Utilities | View | Hex Buses**).
5. Simulate for 400ns.
6. Invoke a Global Reset (**Options | Global Reset**).


**NOTE:**

*Global Reset* is invoked automatically during the Power On process: to disable this feature, deselect **Options | Power On Settings | Global Reset** checkbox.

7. Simulate for 100ns.
8. Select **Signal | Delete | All Waveforms with Power On**.
9. Add the *SimGlobalReset* signal to the *Waveform Viewer* (click  or select **Signal | Add Signal**).

**NOTE:**


*SimGlobalReset* is added to the signal list automatically during netlist creation.

10. Assign the keyboard **Q** key stimulator to the *SimGlobalReset* signal:
  - Click  or select **Stimulator | Add Stimulator**.
  - Drag the **Q** key from the keyboard onto the *SimGlobalReset* signal.
11. Type *q* to set *SimGlobalReset* to the *Low* state.
12. Simulate for 150ns.
13. Type *q* to set *SimGlobalReset* to the *High* state. Click **Step**.
14. Load the *flash.bax* netlist (**File | Load Netlist**).
15. Load the *Signals* test vectors (**File | Load Test Vectors**).
16. Switch to timing simulation mode (**Mode:** button in *Simulator* toolbox).
17. Add the *GSR* signal to the *Waveform Viewer*.

**NOTE:**

The *GSR* signal (in XC4000 families only, *GR* in other families) is added to the signal list during place & route.

18. Assign the keyboard **Q** key stimulator to the *GSR* signal (use the procedure described in point 10).
19. Set *GSR* to the *Low* state (click on the letter **q** next to *GSR* or type *q*).
20. Simulate for 200ns.
21. Set *GSR* to the *High* state. Click **Step**.
22. Close the simulator.
23. Open the CALC7K sample project.
24. Start the *Simulator* in timing mode (**SIM Timing** button).

25. Add CLK, EXC\_P, SW[6:0], and ALU[3:0] to the *Waveform Viewer* ( button or **Signal | Add Signal**)
26. Assign the **B2** stimulator to the CLK signal and **N4** to the EXC\_P signal.
27. Assign the formula  
     $[7e]40[6f]160[50]160[0a]160$   
to the **F0** stimulator.

**NOTE:**

The microprogram listed below has just been defined:

```
[7e]  NOP
[6f]  LOAD   #F
[50]  CLEAR
[0a]  ADD    #A
```

28. Assign the **F0** stimulator to the SW[6:0] bus.
29. Simulate for 600ns. Note that the calculator is not working correctly.
30. Click on the **Power ON** button in the toolbox.
31. Add the *MRESET* signal to the *Waveform Viewer*.

**NOTE:**

The *MRESET* signal in CPLDs should always be driven during the simulation of a fitted netlist.  
If you do not want to use *MRESET* during the simulation, assign the constant 1 to this signal.

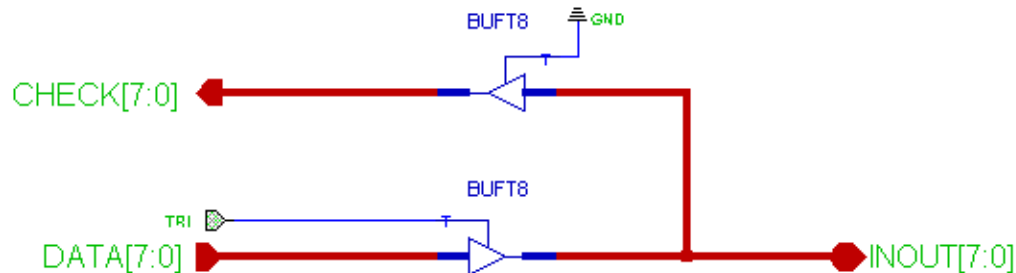
32. Assign the **M** keyboard stimulus to the *MRESET* signal and set it to the *High* state.
33. Simulate for 600ns.
34. Set *MRESET* to the *Low* state. Click **Step**.
35. Close the *Simulator*.




## Exercise 4:

**Topics:** Simulating tristate buses/pins, simulation macros

**Software:** Xilinx Foundation 6.0.1

1. Create a new project called TRI (XACTstep6 type, XC4000A family, 4002APC84 part).
2. Open the *Schematic Editor*.
3. Enter the schematic as pictured below:



4. Click on the *Simulation Toolbox* toolbar button (.
5. Place probes on TRI, DATA[7:0], INOUT[7:0], and CHECK[7:0] by clicking on their names.
6. Start the *Simulator* by clicking on the  button (on the toolbar or in the *SC Probes* toolbox).
7. Select **Utilities | View | Hex Buses** (or click on the  button).
8. Assign the **B3** stimulator to the *TRI* signal, the **D** stimulator to the *DATA[7:0]* signal, and the **I** stimulator to the *INOUT[7:0]* signal.
9. Set the **D** stimulus to *Low* (this puts a hex value of 00 onto the bus) and set **I** to *High* (this puts a hex value of FF on the bus).
10. Simulate for 250ns.
11. Select the *INOUT* bus in the *Waveform Viewer* and change its stimulator mode to *Chip Controlled* (select **Stimulator | Chip Controlled Mode**).

### NOTE:

- *Override* mode makes the logic state set by stimulator stronger than any other signal coming from the simulated circuit.
- *Chip Controlled* mode sets the strength of the stimulator to weak level.

By default all stimulators set in the Foundation Logic Simulator are in *Override* mode. To switch modes use the **Stimulator** menu or the **CC** and **OV** buttons in the *Stimulator Selection* window.

12. Simulate for 200ns.
13. Switch back to *Override* mode (select the bus, select **Stimulator | Override Mode**).
14. Disconnect the stimulator from the *INOUT* bus (select the bus, select **Stimulator | Disconnect**).
15. Simulate for 200ns.
16. Click **Power ON**, then select **Signal | Delete | All Signal Names & Waveforms**.

17. In any text editor, create an ASCII file with the following contents:

```
watch tri
vector dt data[7:0]
v io inout[7:0]
watch check7 check6 check5 check4 check3 check2 check1 check0
break io ? do (print > tri.ref)
h tri
wfm dt @0=0
wfm io @0=55\h 50ns=aa\h
sim 100ns
l tri
wfm dt @100ns=33\h 50ns=cc\h
release io
sim 100ns
h tri
assign io ff\h
sim 100ns
|chk_results tri.ref tri.out comp.log
```

**NOTES:**

- WATCH selects signals to display in the Waveform Viewer and for listing by the PRINT command.
- VECTOR (or V) defines an alias for a group of signals and displays the alias as a bus in the Waveform Viewer.
- BREAK sets a breakpoint on the IO signal and when an event occurs on that signal, it prints all watched signal states to the file TRI.REF.
- H and L force *High* and *Low* states on given signals.
- WFM creates a stimulus waveform according to the description.
- RELEASE removes any stimuli from the signal (and enables output data display on bi-directional pins).
- SIM runs simulation for the given period.
- ASSIGN assigns a value to a signal or bus.
- CHK\_RESULTS compares two stored printouts of simulation results.
- The vertical bar (|) indicates a comment line.

18. Save the file as TRI.CMD in the ACTIVE\PROJECTS\TRI directory.
19. Select **Utilities | Macro | Run** in the *Simulator* window.
20. Click the **Browse** button in the *Run Macro File* window and select TRI.
21. Click **OK**, then **Run**.
22. Select **Utilities | Macro | Edit**. Select TRI and click **Edit**.
23. Replace tri.ref with tri.out in the break command line.
24. Replace ff\h with fe\h in the assign command line
25. Remove the vertical bar (|) from the beginning of the chk\_results command line.
26. Save the macro and exit the editor (**File | Save, File | Exit**).
27. Click on the **Power\_ON** button in the *Simulator* toolbox, then select **Signal | Delete | All Signal Names & Waveforms**.
28. Select **Utilities | Macro | Run**. Select TRI and click **Run**.
29. Using any text editor review the following files:
- |          |                                     |
|----------|-------------------------------------|
| TRI.REF  | reference file from first macro run |
| TRI.OUT  | output file from second macro run   |
| COMP.LOG | comparison results                  |
30. Quit the *Simulator*.