COSE222 Computer Architecture

Newly updated for RISC-V on July 15, 2020

by Prof. Taeweon Suh

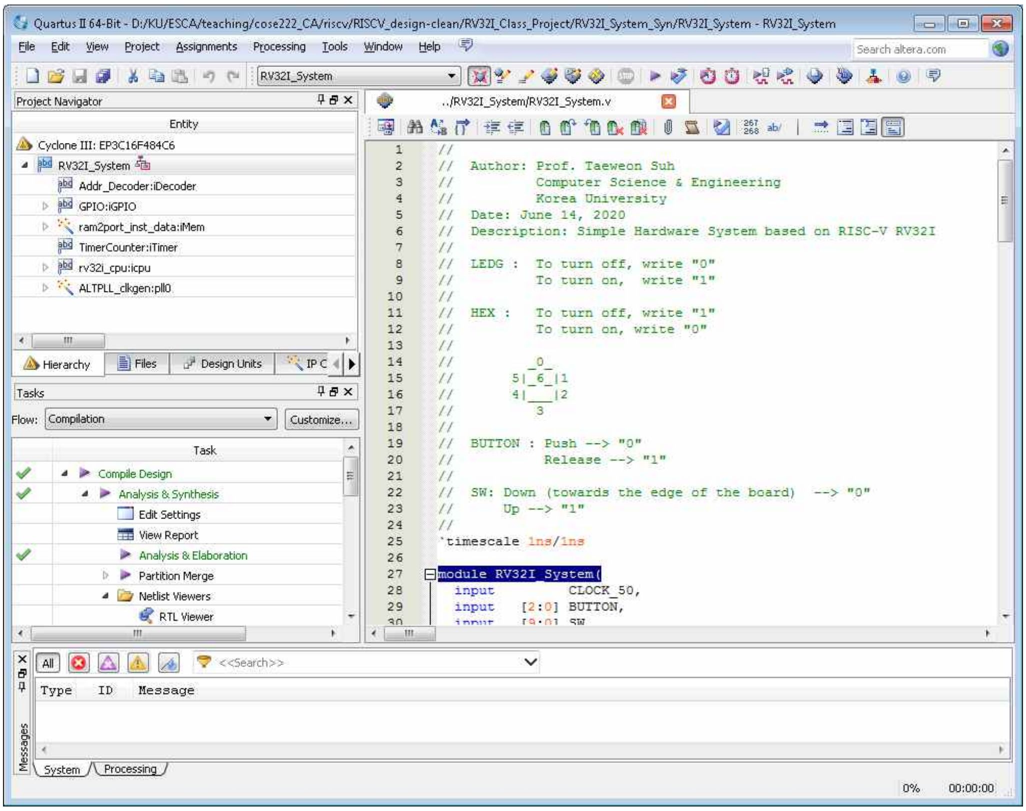
RV32I (RISC-V) System Synthesis and Simulation

This note provides the instructions on how to do the hardware synthesis & simulation. You are provided with a simple RV32I CPU design in Verilog. The CPU design has capability of executing only a few RV32I instructions. Its implementation is based on single-cycle execution, meaning that each machine code (instruction) takes 1 clock cycle for execution. The design also comes with a simple testbench. The test (assembly) program is loaded in memory after compilation. For the hardware simulation, we use a CAD (Computer Aided Design) tool called `ModelSim`, which is widely used in industry. Follow the steps below to run the RV32I CPU design with `ModelSim`. Prior to following the instructions below, install the EDA (Electronic Design Automation) tools linked on the class web: **ModelSim (v10.1d)** and **Quartus-II (v13.1)**. Download the RV32I System design on the class web, unzip `RV32I_Class_Project.7z` and you will find 3 directories:

- **`RV32I_System`** contains the Verilog source code for RV32I CPU, I/O devices (Timer and GPIO), etc. Traverse the directories and Verilog code yourself…
- **`RV32I_System_Sim`** contains the compiled binaries of the 'RV32I_system' for hardware simulation using ModelSim.
- **`RV32I_System_Syn`** contains the synthesized version of the 'RV32I_System' using Quartus-II.

• Synthesis using Quartus-II and Download to DE0 Board

1. Double-click the Quartus project file (`RV32I_System.qpf`) at the `RV32I_System_Syn` directory
   - I have created a Quartus-II project for you. The project file contains all the necessary files for test-driving.



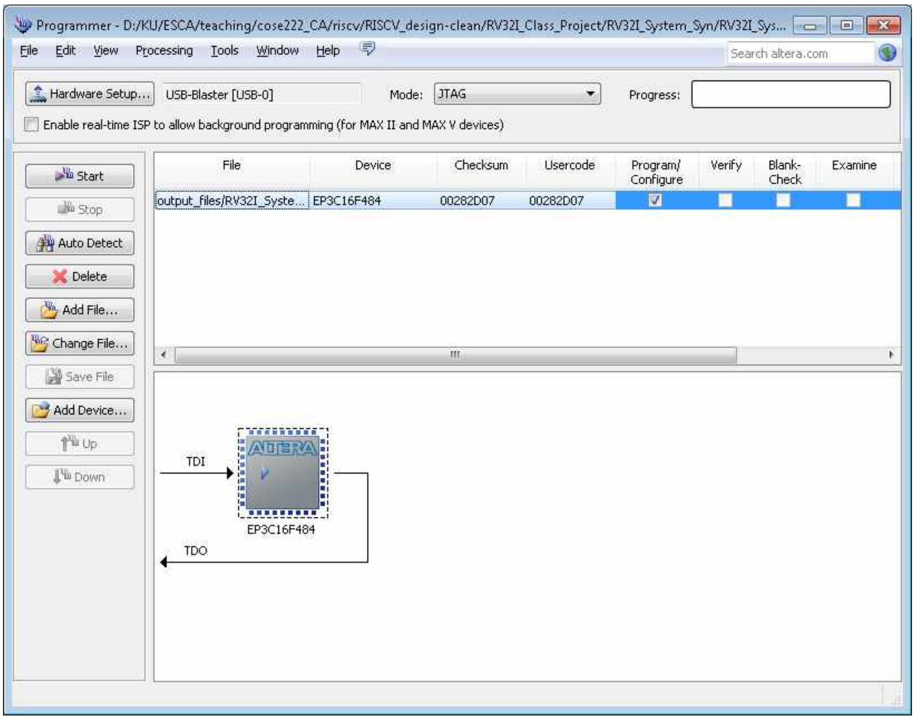
```
//
//  Author: Prof. Taeweon Suh
//          Computer Science & Engineering
//          Korea University
//  Date: June 14, 2020
//  Description: Simple Hardware System based on RISC-V RV32I
//
//  LEDG :  To turn off, write "0"
//          To turn on,  write "1"
//
//  HEX :   To turn off, write "1"
//          To turn on, write "0"
//
//         _0_
//       5|_6_|1
//       4|___|2
//          3
//
//  BUTTON : Push --> "0"
//           Release --> "1"
//
//  SW: Down (towards the edge of the board)  --> "0"
//      Up --> "1"
//
`timescale 1ns/1ns

module RV32I_System(
  input          CLOCK_50,
  input    [2:0] BUTTON,
  input    [9:0] SW
```

2. Synthesize the whole design again
   - Select `RV32I_System` in the **Project Navigator** pane
   - **Processing** → **Start Compilation.** It will take a little bit of time to compile the whole design

3. Download the design to the Cyclone-III FPGA on DE0 board
   - **Tools** → **Programmer,** Then click on **Start**

     • Make sure that you have USB-Blaster set up correctly, which is shown on the upper left corner of the window. If it is not set up correctly, click on **Hardware Setup** button and select USB blaster. If you don't have the USB blaster listed with the Hardware Setup, you should install the USB device driver. The driver is located at the Quartus-II installation directory. In my case, the driver is located at **C:\altera\13.1\quartus\drivers\usb-blaster**
   - **Number '5' should be displayed on the HEX0 7 Segment**

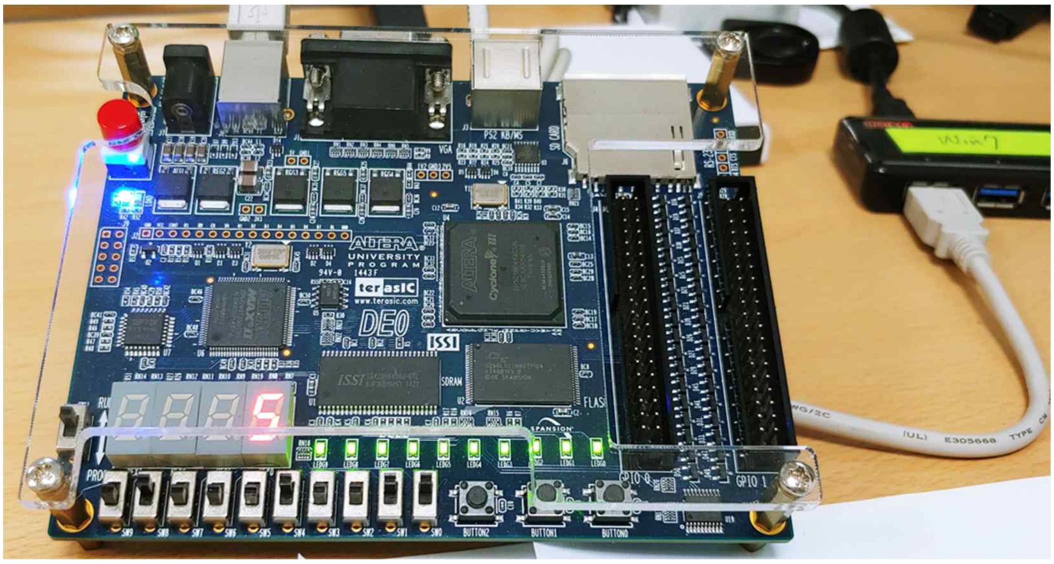


Programmer - D:/KU/ESCA/teaching/cose222_CA/riscv/RISCV_design-clean/RV32I_Class_Project/RV32I_System_Syn/RV32I_Sys...

Hardware Setup... USB-Blaster [USB-0] Mode: JTAG Progress:

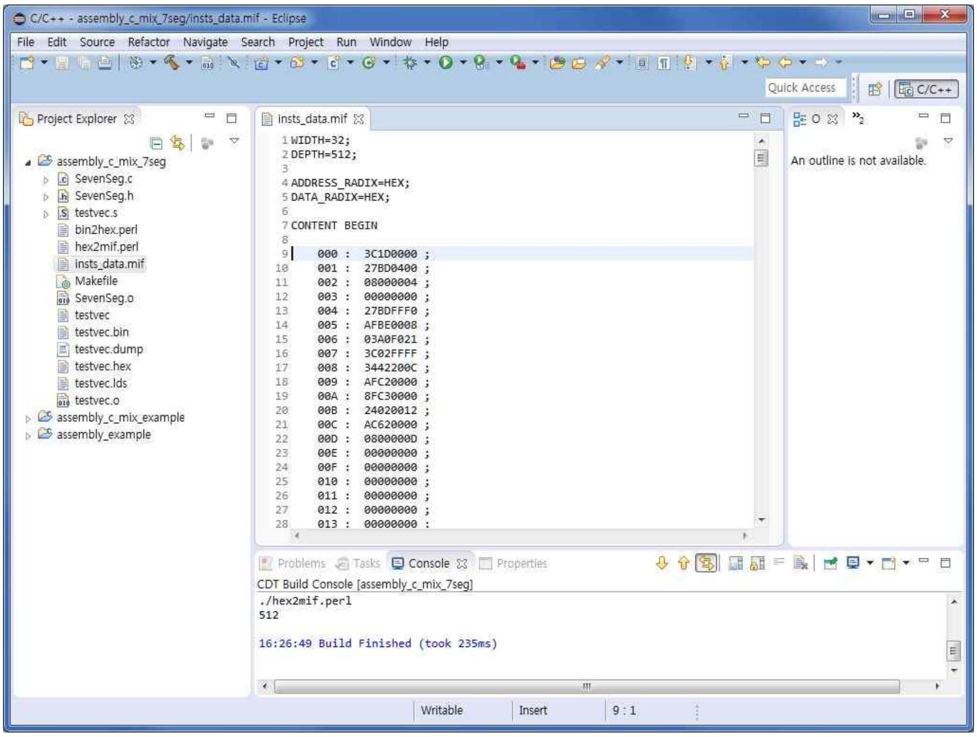
☐ Enable real-time ISP to allow background programming (for MAX II and MAX V devices)

File	Device	Checksum	Usercode	Program/ Configure	Verify	Blank- Check	Examine
output_files/RV32I_Syste...	EP3C16F484	00282D07	00282D07	☑	☐	☐	☐



4. If you want to run your own program on RV32I, compile your C and/or assembly code with Eclipse.

- It will generate a file called `insts_data.mif`
- mif (memory initialization file) is for initializing the internal memory inside the Altera FPGA
- I have created the internal memory via **Tools** → **MegaWizard Plugin Manager** on Quartus-II



```
WIDTH=32;
DEPTH=512;

ADDRESS_RADIX=HEX;
DATA_RADIX=HEX;

CONTENT BEGIN

    000 :  3C1D0000 ;
    001 :  27BD0400 ;
    002 :  08000004 ;
    003 :  00000000 ;
    004 :  27BDFFF0 ;
    005 :  AFBE0008 ;
    006 :  03A0F021 ;
    007 :  3C02FFFF ;
    008 :  3442200C ;
    009 :  AFC20000 ;
    00A :  8FC30000 ;
    00B :  24020012 ;
    00C :  AC620000 ;
    00D :  0800000D ;
    00E :  00000000 ;
    00F :  00000000 ;
    010 :  00000000 ;
    011 :  00000000 ;
    012 :  00000000 ;
    013 :  00000000 ;
```

CDT Build Console [assembly_c_mix_7seg]

```
./hex2mif.perl
512

16:26:49 Build Finished (took 235ms)
```

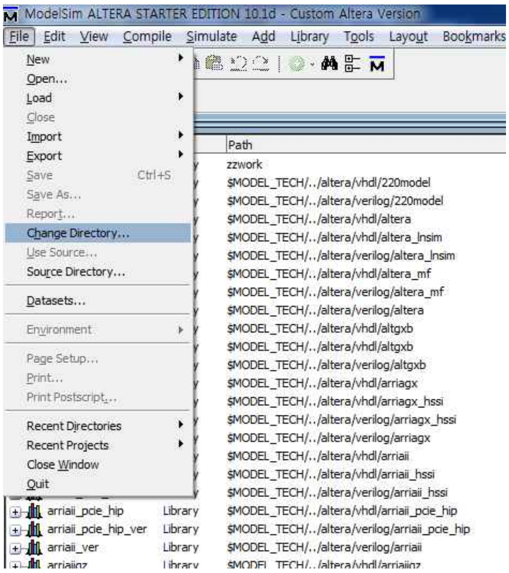
5. For testing your own assembly and/or C,

- Download an example program (Assembly and C mixed program) at http://esca.korea.ac.kr/teaching/cose222_CA/hw-sw-tools/RISCV/lab0_c_asm_mix.7z The program will display a number 5 on HEX0 7 Segment.
  - ✓ Create an Eclipse project to compile the code
- Copy **insts_data.mif** to the `RV32I_System_Syn` directory
- Then, you have 2 options for downloading
  - ✓ 1st option: follow from step #1 to step #3
  - ✓ 2nd option: Execute the `quartus_mem_update_download.bat` by double-clicking the file (you don't have to start Quartus-II). The file is located under the `RV32I_System_Syn` directory
- Change the example assembly and/or C code, generate mif, copy it to `RV32I_System_Syn` and download to DE0…

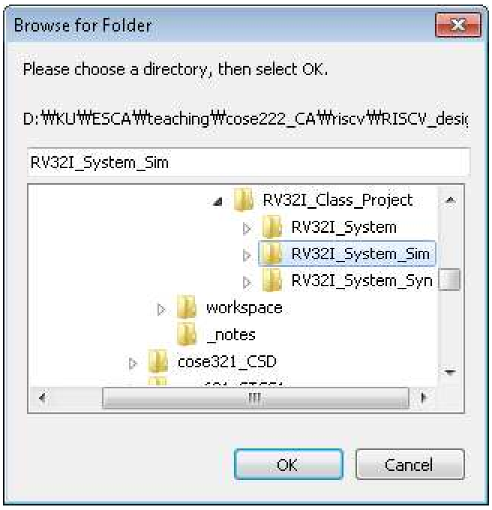
- ## Hardware Simulation with ModelSim

1. Invoke ModelSim

2. **File → Change Directory**
   - Select `RV32I_System_Sim`



		Path
		zzwork
		$MODEL_TECH/../altera/vhdl/220model
		$MODEL_TECH/../altera/verilog/220model
		$MODEL_TECH/../altera/vhdl/altera
		$MODEL_TECH/../altera/vhdl/altera_lnsim
		$MODEL_TECH/../altera/verilog/altera_lnsim
		$MODEL_TECH/../altera/vhdl/altera_mf
		$MODEL_TECH/../altera/verilog/altera_mf
		$MODEL_TECH/../altera/verilog/altera
		$MODEL_TECH/../altera/vhdl/altgxb
		$MODEL_TECH/../altera/vhdl/altgxb
		$MODEL_TECH/../altera/verilog/altgxb
		$MODEL_TECH/../altera/vhdl/arriagx
		$MODEL_TECH/../altera/vhdl/arriagx_hssi
		$MODEL_TECH/../altera/verilog/arriagx_hssi
		$MODEL_TECH/../altera/verilog/arriagx
		$MODEL_TECH/../altera/vhdl/arriaii
		$MODEL_TECH/../altera/vhdl/arriaii_hssi
		$MODEL_TECH/../altera/verilog/arriaii_hssi
arriaii_pcie_hip	Library	$MODEL_TECH/../altera/vhdl/arriaii_pcie_hip
arriaii_pcie_hip_ver	Library	$MODEL_TECH/../altera/verilog/arriaii_pcie_hip
arriaii_ver	Library	$MODEL_TECH/../altera/verilog/arriaii
arriaiigz	Library	$MODEL_TECH/../altera/vhdl/arriaiigz

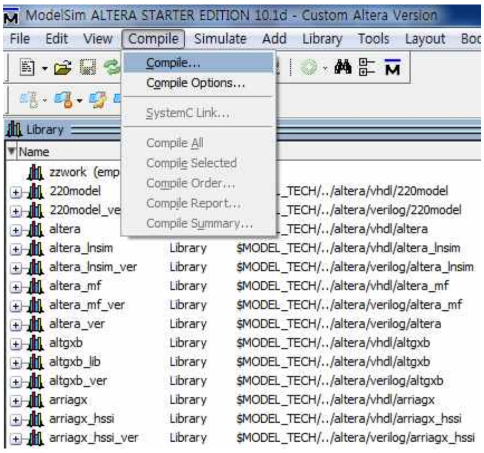


Please choose a directory, then select OK.

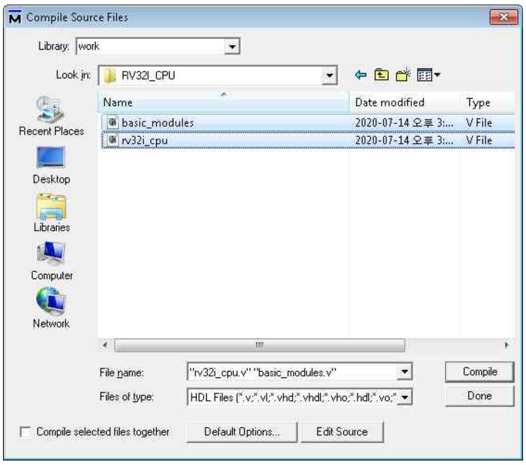
D:₩KU₩ESCA₩teaching₩cose222_CA₩riscv₩RISCV_desi

RV32I_System_Sim

3. '`RV32I_System_Sim`' folder has a directory (called **work**) where the RV32I System Verilog design was compiled to
   - So, the **work** directory contains the compiled version of the Verilog code.
   - Remember that every time you change and/or add the Verilog code, you have to compile it before simulation.
   - Let's compile the CPU Verilog code just for showing the steps you have to take.
     - ✓ **Compile → Compile**
     - ✓ In the **Look in**, change to the directory where the CPU source code is located (`RV32I_System/RV32I_CPU`)
     - ✓Select all the files and click on **Compile**



Name	Type	Path
zzwork (emp		
220model		_TECH/../altera/vhdl/220model
220model_ve		_TECH/../altera/verilog/220model
altera		_TECH/../altera/vhdl/altera
altera_lnsim	Library	$MODEL_TECH/../altera/vhdl/altera_lnsim
altera_lnsim_ver	Library	$MODEL_TECH/../altera/verilog/altera_lnsim
altera_mf	Library	$MODEL_TECH/../altera/vhdl/altera_mf
altera_mf_ver	Library	$MODEL_TECH/../altera/verilog/altera_mf
altera_ver	Library	$MODEL_TECH/../altera/verilog/altera
altgxb	Library	$MODEL_TECH/../altera/vhdl/altgxb
altgxb_lib	Library	$MODEL_TECH/../altera/vhdl/altgxb
altgxb_ver	Library	$MODEL_TECH/../altera/verilog/altgxb
arriagx	Library	$MODEL_TECH/../altera/vhdl/arriagx
arriagx_hssi	Library	$MODEL_TECH/../altera/vhdl/arriagx_hssi
arriagx_hssi_ver	Library	$MODEL_TECH/../altera/verilog/arriagx_hssi



Library: work

Look in: RV32I_CPU

Name	Date modified	Type
basic_modules	2020-07-14 오후 3:...	V File
rv32i_cpu	2020-07-14 오후 3:...	V File

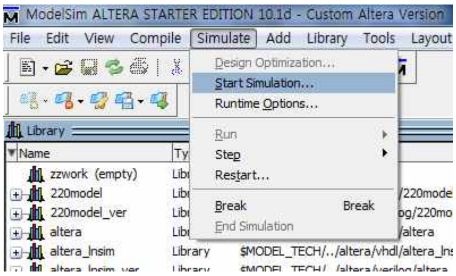
File name: "rv32i_cpu.v" "basic_modules.v"

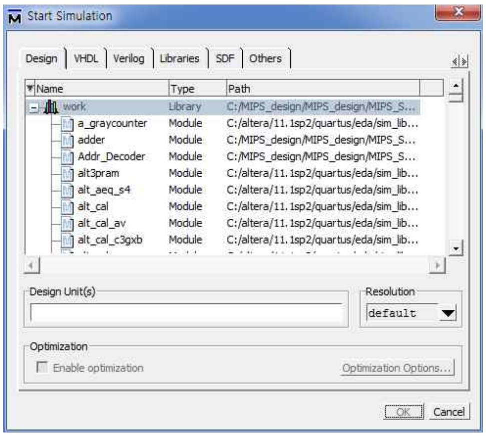
Files of type: HDL Files (*.v;*.vl;*.vhd;*.vhdl;*.vho;*.hdl;*.vo;*

□ Compile selected files together

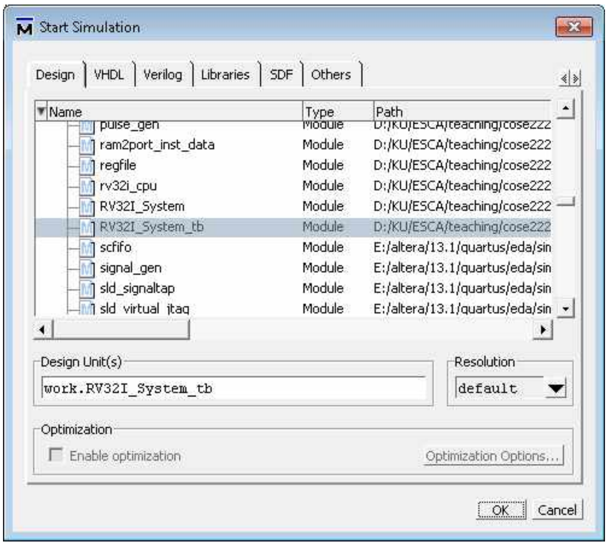
4. Start Simulation

- **Simulate → Start Simulation**
- Select `RV32I_System_tb` (testbench) under **work** and click on **OK**
  - ✓ Check out the testbench source code (`RV32I_System_tb.v`) which is located at `RV32I_System`





Name	Type	Path
work	Library	C:/MIPS_design/MIPS_design/MIPS_S...
a_graycounter	Module	C:/altera/11.1sp2/quartus/eda/sim_lib...
adder	Module	C:/MIPS_design/MIPS_design/MIPS_S...
Addr_Decoder	Module	C:/MIPS_design/MIPS_design/MIPS_S...
alt3pram	Module	C:/altera/11.1sp2/quartus/eda/sim_lib...
alt_aeq_s4	Module	C:/altera/11.1sp2/quartus/eda/sim_lib...
alt_cal	Module	C:/altera/11.1sp2/quartus/eda/sim_lib...
alt_cal_av	Module	C:/altera/11.1sp2/quartus/eda/sim_lib...
alt_cal_c3gxb	Module	C:/altera/11.1sp2/quartus/eda/sim_lib...

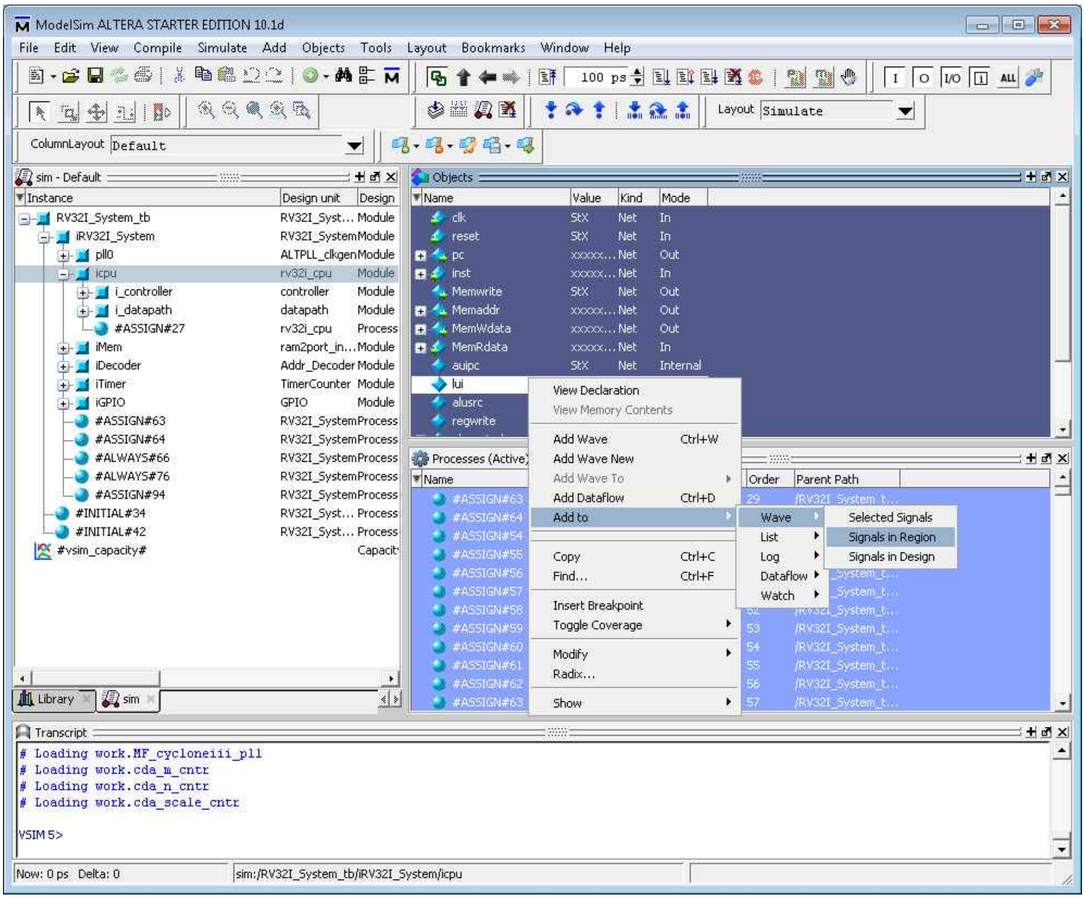


Name	Type	Path
pulse_gen	Module	D:/KU/ESCA/teaching/cose222
ram2port_inst_data	Module	D:/KU/ESCA/teaching/cose222
regfile	Module	D:/KU/ESCA/teaching/cose222
rv32i_cpu	Module	D:/KU/ESCA/teaching/cose222
RV32I_System	Module	D:/KU/ESCA/teaching/cose222
RV32I_System_tb	Module	D:/KU/ESCA/teaching/cose222
scfifo	Module	E:/altera/13.1/quartus/eda/sin
signal_gen	Module	E:/altera/13.1/quartus/eda/sin
sld_signaltap	Module	E:/altera/13.1/quartus/eda/sin
sld_virtual_jtag	Module	E:/altera/13.1/quartus/eda/sin

Design Unit(s): work.RV32I_System_tb

Resolution: default

5. Add the signals you want to watch in the waveform
    - Select **icpu** on the left pane
    - Right-click on the **Objects** pane
        ✓ **Add To → Wave → Signals in Region**



Instance	Design unit	Design
RV32I_System_tb	RV32I_Syst...	Module
iRV32I_System	RV32I_System	Module
pll0	ALTPLL_clkgen	Module
icpu	rv32i_cpu	Module
i_controller	controller	Module
i_datapath	datapath	Module
#ASSIGN#27	rv32i_cpu	Process
iMem	ram2port_in...	Module
iDecoder	Addr_Decoder	Module
iTimer	TimerCounter	Module
iGPIO	GPIO	Module
#ASSIGN#63	RV32I_System	Process
#ASSIGN#64	RV32I_System	Process
#ALWAYS#66	RV32I_System	Process
#ALWAYS#76	RV32I_System	Process
#ASSIGN#94	RV32I_System	Process
#INITIAL#34	RV32I_Syst...	Process
#INITIAL#42	RV32I_Syst...	Process
#vsim_capacity#		Capacit

Name	Value	Kind	Mode
clk	StX	Net	In
reset	StX	Net	In
pc	xxxxx...	Net	Out
inst	xxxxx...	Net	In
Memwrite	StX	Net	Out
Memaddr	xxxxx...	Net	Out
MemWdata	xxxxx...	Net	Out
MemRdata	xxxxx...	Net	In
auipc	StX	Net	Internal
lui			
alusrc			
regwrite			

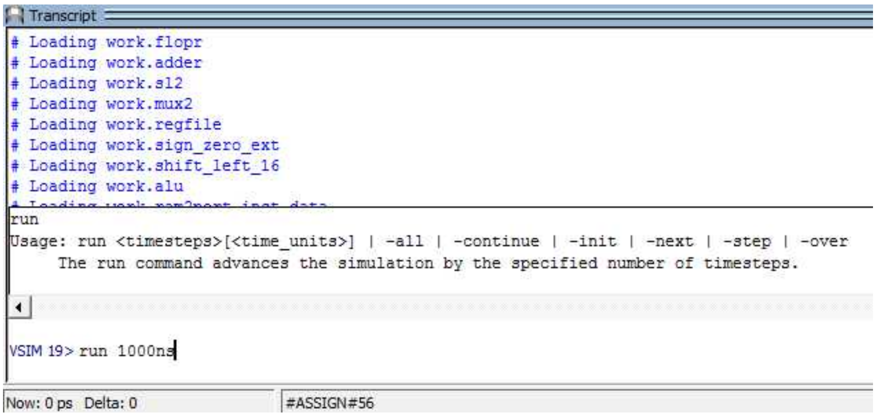
Name	Order	Parent Path
#ASSIGN#63	29	/RV32I_System_t...
#ASSIGN#64		
#ASSIGN#54		
#ASSIGN#55		_System_t...
#ASSIGN#56		_System_t...
#ASSIGN#57		_System_t...
#ASSIGN#58	52	/RV32I_System_t...
#ASSIGN#59	53	/RV32I_System_t...
#ASSIGN#60	54	/RV32I_System_t...
#ASSIGN#61	55	/RV32I_System_t...
#ASSIGN#62	56	/RV32I_System_t...
#ASSIGN#63	57	/RV32I_System_t...

```
# Loading work.MF_cycloneiii_pll
# Loading work.cda_m_cntr
# Loading work.cda_n_cntr
# Loading work.cda_scale_cntr

VSIM 5>
```

Now: 0 ps Delta: 0 sim:/RV32I_System_tb/iRV32I_System/icpu

6. Run simulation
    - Type **run 1000ns** in the Transcript pane
        ✓ It means that the simulation will run for 1000ns

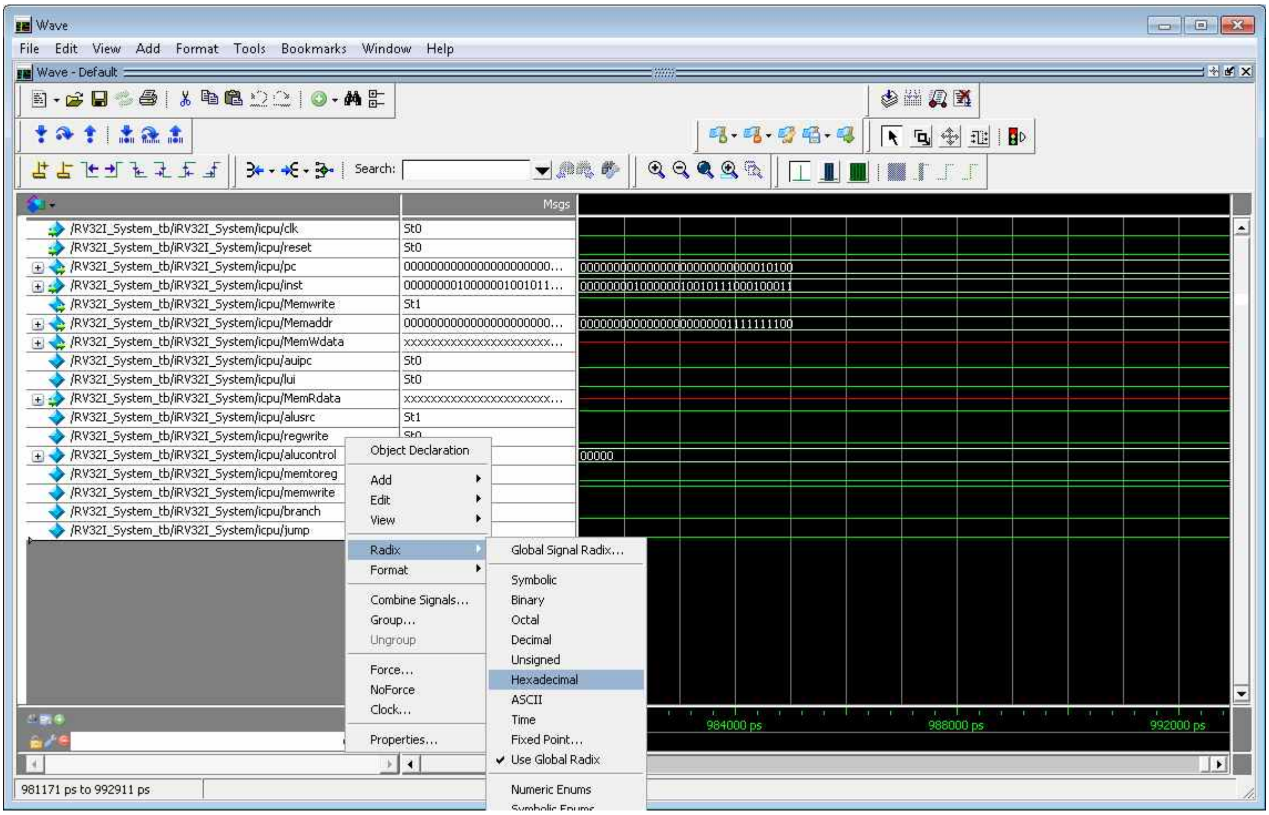


```
# Loading work.flopr
# Loading work.adder
# Loading work.sl2
# Loading work.mux2
# Loading work.regfile
# Loading work.sign_zero_ext
# Loading work.shift_left_16
# Loading work.alu
# Loading work.ram2port_inst_data
run
Usage: run <timesteps>[<time_units>] | -all | -continue | -init | -next | -step | -over
    The run command advances the simulation by the specified number of timesteps.

VSIM 19> run 1000ns
```

Now: 0 ps Delta: 0 #ASSIGN#56

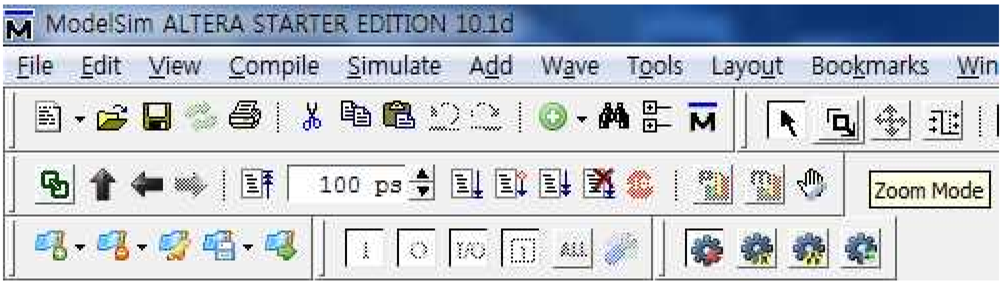
- Click on the Wave window and observe the signals (such as pc, fetched instructions, control signals, ALU output etc) you want to watch
  - ✓ **View → Zoom → Zoom Full**
  - ✓ Change the radix to Hexadecimal for buses
  - ✓ Run simulation for 1000ns more (type **run 1000ns** in the Transcript pane)



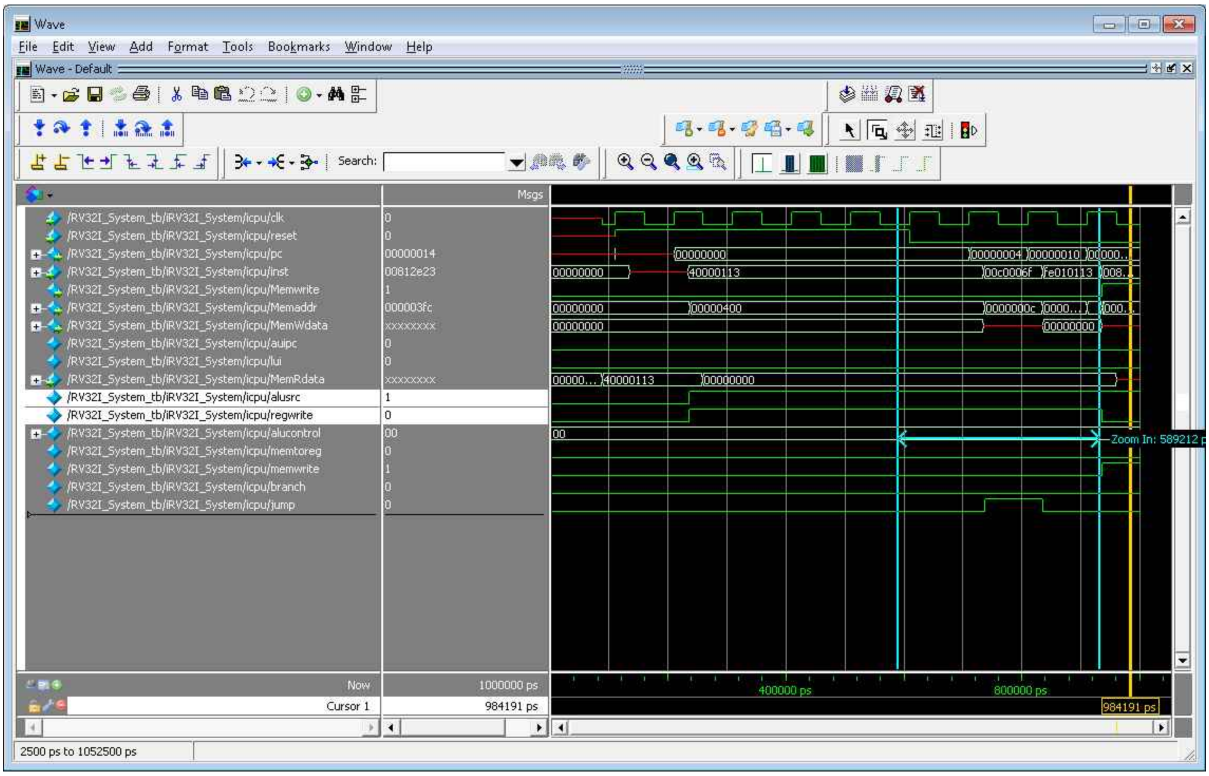
	Msgs	
/RV32I_System_tb/iRV32I_System/icpu/clk	St0	
/RV32I_System_tb/iRV32I_System/icpu/reset	St0	
/RV32I_System_tb/iRV32I_System/icpu/pc	0000000000000000000000...	00000000000000000000000000010100
/RV32I_System_tb/iRV32I_System/icpu/inst	0000000010000001001011...	00000000100000001001011100010011
/RV32I_System_tb/iRV32I_System/icpu/Memwrite	St1	
/RV32I_System_tb/iRV32I_System/icpu/Memaddr	0000000000000000000000...	00000000000000000000001111111100
/RV32I_System_tb/iRV32I_System/icpu/MemWdata	xxxxxxxxxxxxxxxxxxxxxx...	
/RV32I_System_tb/iRV32I_System/icpu/auipc	St0	
/RV32I_System_tb/iRV32I_System/icpu/lui	St0	
/RV32I_System_tb/iRV32I_System/icpu/MemRdata	xxxxxxxxxxxxxxxxxxxxxx...	
/RV32I_System_tb/iRV32I_System/icpu/alusrc	St1	
/RV32I_System_tb/iRV32I_System/icpu/regwrite	St0	
/RV32I_System_tb/iRV32I_System/icpu/alucontrol		00000
/RV32I_System_tb/iRV32I_System/icpu/memtoreg		
/RV32I_System_tb/iRV32I_System/icpu/memwrite		
/RV32I_System_tb/iRV32I_System/icpu/branch		
/RV32I_System_tb/iRV32I_System/icpu/jump		

981171 ps to 992911 ps

✓Use the Zoom Mode to zoom in the area you are interested in



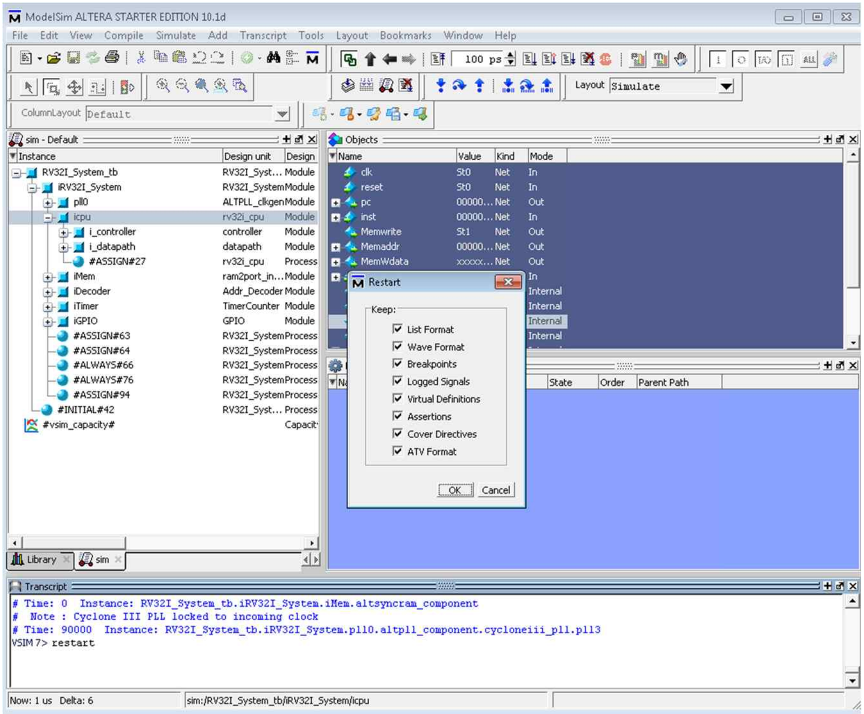
ModelSim ALTERA STARTER EDITION 10.1d
File Edit View Compile Simulate Add Wave Tools Layout Bookmarks Win
100 ps
Zoom Mode



Signal	Msgs
/RV32I_System_tb/iRV32I_System/icpu/clk	0
/RV32I_System_tb/iRV32I_System/icpu/reset	0
/RV32I_System_tb/iRV32I_System/icpu/pc	00000014
/RV32I_System_tb/iRV32I_System/icpu/inst	00812e23
/RV32I_System_tb/iRV32I_System/icpu/Memwrite	1
/RV32I_System_tb/iRV32I_System/icpu/Memaddr	000003fc
/RV32I_System_tb/iRV32I_System/icpu/MemWdata	xxxxxxxx
/RV32I_System_tb/iRV32I_System/icpu/auipc	0
/RV32I_System_tb/iRV32I_System/icpu/lui	0
/RV32I_System_tb/iRV32I_System/icpu/MemRdata	xxxxxxxx
/RV32I_System_tb/iRV32I_System/icpu/alusrc	1
/RV32I_System_tb/iRV32I_System/icpu/regwrite	0
/RV32I_System_tb/iRV32I_System/icpu/alucontrol	00
/RV32I_System_tb/iRV32I_System/icpu/memtoreg	0
/RV32I_System_tb/iRV32I_System/icpu/memwrite	1
/RV32I_System_tb/iRV32I_System/icpu/branch	0
/RV32I_System_tb/iRV32I_System/icpu/jump	0

Now 1000000 ps
Cursor 1 984191 ps
2500 ps to 1052500 ps

7. To restart the simulation
   - Type **restart** in the Transcript pane and click on **OK**
   - Add more signals to the wave if you want to add more signals
     - ✓Go back to step #6 to see how to add signals to the wave
   - Run simulation
     - ✓For example, type **run 2000ns** in the Transcript pane as you did in the step #7



ModelSim ALTERA STARTER EDITION 10.1d
File Edit View Compile Simulate Add Transcript Tools Layout Bookmarks Window Help
100 ps
Layout Simulate
ColumnLayout Default
sim - Default
Objects
Library
sim

Instance	Design unit	Design
RV32I_System_tb	RV32I_Syst…	Module
iRV32I_System	RV32I_System	Module
pll0	ALTPLL_clkgen	Module
icpu	rv32i_cpu	Module
i_controller	controller	Module
i_datapath	datapath	Module
#ASSIGN#27	rv32i_cpu	Process
iMem	ram2port_in…	Module
iDecoder	Addr_Decoder	Module
iTimer	TimerCounter	Module
iGPIO	GPIO	Module
#ASSIGN#63	RV32I_System	Process
#ASSIGN#64	RV32I_System	Process
#ALWAYS#66	RV32I_System	Process
#ALWAYS#76	RV32I_System	Process
#ASSIGN#94	RV32I_System	Process
#INITIAL#42	RV32I_Syst…	Process
#vsim_capacity#		Capacit

Name	Value	Kind	Mode
clk	St0	Net	In
reset	St0	Net	In
pc	00000…	Net	Out
inst	00000…	Net	In
Memwrite	St1	Net	Out
Memaddr	00000…	Net	Out
MemWdata	xxxxx…	Net	Out
			In
			Internal
			Internal
			Internal
			Internal

Restart

Keep:

- ☑ List Format
- ☑ Wave Format
- ☑ Breakpoints
- ☑ Logged Signals
- ☑ Virtual Definitions
- ☑ Assertions
- ☑ Cover Directives
- ☑ ATV Format

OK Cancel

State Order Parent Path

Transcript

```
# Time: 0  Instance: RV32I_System_tb.iRV32I_System.iMem.altsyncram_component
#  Note : Cyclone III PLL locked to incoming clock
# Time: 90000  Instance: RV32I_System_tb.iRV32I_System.pll0.altpll_component.cycloneiii_pll.pll3
VSIM 7> restart
```

Now: 1 us Delta: 6
sim:/RV32I_System_tb/iRV32I_System/icpu

8. To run the simulation with a different program
    - Copy **`insts_data.mif`** you generated with Eclipse to the `RV32I_System_Sim` directory
    - Run the simulation again: Probably you only need to restart the simulation without exiting the ModelSim.