

ALTERA INSTALLATION

CSE 141L – UC San Diego
January 22, 2013

Introduction

- Fortunately most of you have already used Altera in CSE 140L, so things will largely be the same for you.
- If you haven't, its basically an IDE for hardware design
- At least B250 and B260 have Altera installed, but the labs are very crowded this quarter so you may want to install it yourself as well.

Altera Installation

- Step 1: Go to <https://www.altera.com/download/software/quartus-ii-we/11.0>
- Step 2: Select whatever version you choose (We will support the Windows version since that is what is on the machines in the lab, but there are Linux installs available as well.
- Step 3: Option 1 requires Java, if this is not to your liking (I wouldn't blame you given recent events), you can download the file directly.

Altera Installation

The screenshot shows the Altera website's download center for Quartus II Web Edition v11.0. The browser address bar (1) contains the URL <https://www.altera.com/download/software/quartus-ii-we/11.0>. The left sidebar (2) lists various software categories. The main content area features the 'Quartus II Web Edition Software' heading, a breadcrumb trail, release date, version, and two download options. 'Download Option 1: Altera Installer' includes buttons for Windows (13 MB) and Linux (20 MB) versions, with a note that the installer reduces file size by up to 50 percent. 'Download Option 2: Individual Files' includes a table of files with their names, MD5 sums, and sizes. A red box highlights the Linux file row (3), and an arrow points to the '11.0 quartus_free_linux.sh' file name. Below the table, a link explains MD5 sum values, and a list of other individual download files is provided.

Altera

Download Center Documentation myAltera Account

Devices Design Tools & Services End Markets Technology Training Support About Buy

Design Software

- Quartus II Subscription Edition
- Quartus II Web Edition
- MegaCore IP Library
- ModelSim-Altera
- ModelSim-Altera Starter
- Nios II EDS Legacy Tools
- DSP Builder
- OS Support

Archives

- Service Packs
- Design Software

Licensing

- Get and Manage Licenses
- Licensing FAQ
- License Daemon Software

Programming Software

- Quartus II Programmer
- Jam™ STAPL Software

Drivers

- Cable & Adapter Drivers

Board Layout and Test

- BSDL Models
- SPICE Models
- IBIS Models
- Schematic Review Worksheets
- Layout Review Worksheets
- Cadence PCB Libraries
- Mentor Graphics PCB Libraries
- Gerber Files

Legacy Software

- MAX+PLUS II
- Other Legacy Software

Quartus II Web Edition Software

Home > Support > Downloads > Quartus II Web Edition

Release date: May, 2011

Quartus II Web Edition v11.0

Download Option 1: Altera Installer

Use Altera® Installer to select your choice of design software with non-NTLM proxies. You must first download this small Altera Installer, which then allows you to choose the software and device you want to install. The Altera Installer reduces file download size by up to 50 percent.

Download Windows Version (13 MB) **Download Linux Version (20 MB)**

Free and No License Required!

Download, Install, and Go.

Download Option 2: Individual Files

Quartus II Web Edition	Platform	File Name	Size
Quartus II Web Edition	Windows	11.0 quartus_free_windows.exe MD5: b4b598f56522e47102b260b65649566e	2.7 GB
Quartus II Web Edition	Linux	11.0 quartus_free_linux.sh MD5: c2367f139845a3f2a351a9fcc014e940	3.2 GB

[What the MD5 sum value is and what it is for](#)

Other Individual Download Files:

- Quartus II Subscription Edition
- ModelSim-Altera
- ModelSim-Altera Starter
- Nios II EDS
- DSP Builder
- Programming Software

Altera Installation

- Step 4: When asked to create an account, you can just choose the “One-Time Access” option, using an email and “UCSD” for the business.
- Step 5: IF you did step 3 and downloaded the file directly, you may also need to download Modelsim separately. You can find Modelsim here:
<https://www.altera.com/download/software/modelsim-starter/11.0>
 - ▣ Download the stock 11.0 version first, use sp1 only if there are problems.

Altera Installation

- Step 6: Once the process to create your account is done, you can sign in to install the file.
- Step 7: Downloading a 2.7 GB takes time...try downloading on campus. Grab a coffee.
 - ▣ Kidding aside, this highlights another issue with Altera, installation takes time, do it asap.
- Step 8: If you used the download manager you should have Altera auto-install for Quartus II **Web Edition** and Modelsim-Altera **Starter Edition**
 - ▣ There are paid versions of this, make sure to get the free ones.

Altera Installation

- Step 8b: If you downloaded manually make sure you have these two programs and ensure that the Cyclone and Stratix FPGA libraries are installed as well.
- Step 9: Once you begin the installation...you can safely step away.
 - ▣ You can probably cook a nice dinner, this can take well more than an hour to install fully.
 - ▣ As such this is another reminder to install this ASAP, the basement labs appear swamped this quarter so space is at a premium.
- Step 10: You do not need a license, so if something pops up asking for one ignore it.
- Step 11: Fin!

ALTERA/VERILOG TUTORIAL

CSE 141L – UC San Diego
January 22, 2013

Altera Tutorial

- This tutorial is adapted from the CSE 141L Lab 1 from Summer 2012 (available here: <http://cseweb.ucsd.edu/classes/su12/cse141L-a/lab1.html>).
- It will show how to set up a project, write a couple Verilog files, then test them.
- It is NOT designed to show you how to do your lab, as such the ALU is very simple, but it may give you ideas on how to implement your project.

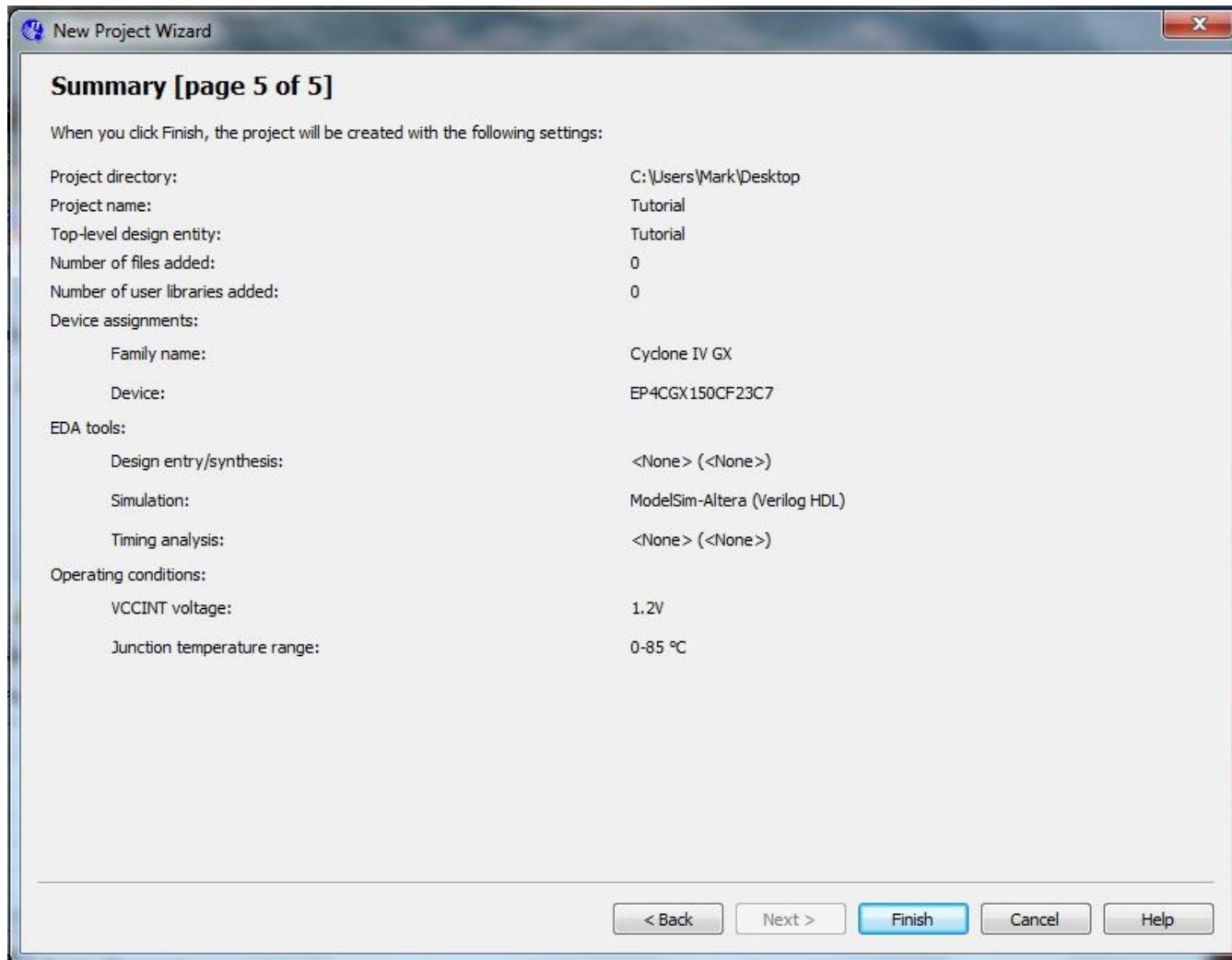
Altera Tutorial

- Some additional (and more detailed) tutorials can be found on Piazza.
 - ▣ One covers how to set up a project in Altera
 - ▣ The other covers how to do simulations using ModelSim
- Even further help can be found here:
<https://sites.google.com/a/eng.ucsd.edu/using-the-altera-tools/>

Altera Tutorial

- Step 1: With Altera open, go to File -> New Project Wizard...
- Step 2: Type the name of the project and the directory you want to place it in. Click next to pass step 2.
- Step 3: When you reach Step 3: Select the Cyclone II family and the EP2C35F672C6 chip.
- Step 4: For “Simulation”: Click the tool name drop box and select “Modelsim-Altera”, for the format: “Verilog HDL”
- Step 5: Verify that you did the above on the Summary screen.

Altera Tutorial



□ Sidebar: What the is a Cyclone II EP2C35F672C6?

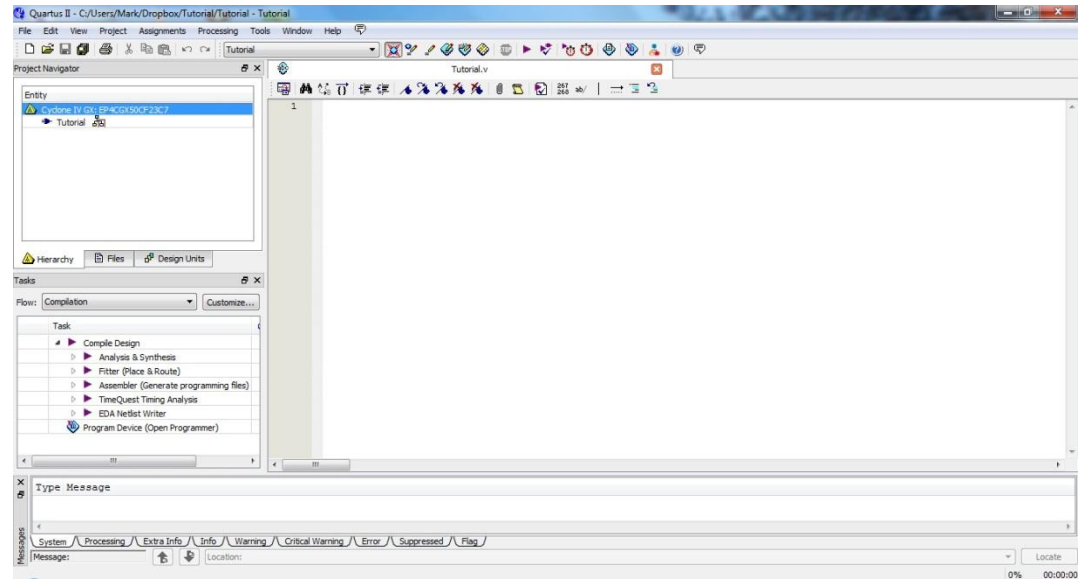
- [illegible]

Altera Tutorial

- Sidebar: What is a Cyclone II EP2C35F672C6?
 - We will not be actually doing putting your design on-chip, but other classes have attempted it, and some of you have experience in CSE 140L. We would like you to use the package denoted in Lab 2 for your testing.
 - The reason we want this is because, if you are arguing area or speed of the processor, the chip matters. If you are arguing something like cycle count, then this is less of a concern.

Altera Tutorial

- Step 6: Go to File->New... and select Verilog HDL File to start coding up components. Be sure to save your files in the same directory as your project.



Altera Tutorial

□ Step 7: Insert the following code:

```
`timescale 1ns / 1ps
```

```
module Tutorial#(parameter W = 8)
(
    input clk,
    input [W-1:0] inA, inB,
    output [W:0] out,
    output isOdd
);
```

```
reg [W-1:0] regA, regB;
reg [W:0] regOut;
reg regOdd;
wire [W:0] wireOut;
```

```
assign wireOut = regA + regB;
assign out = regOut;
assign isOdd = regOdd;
```

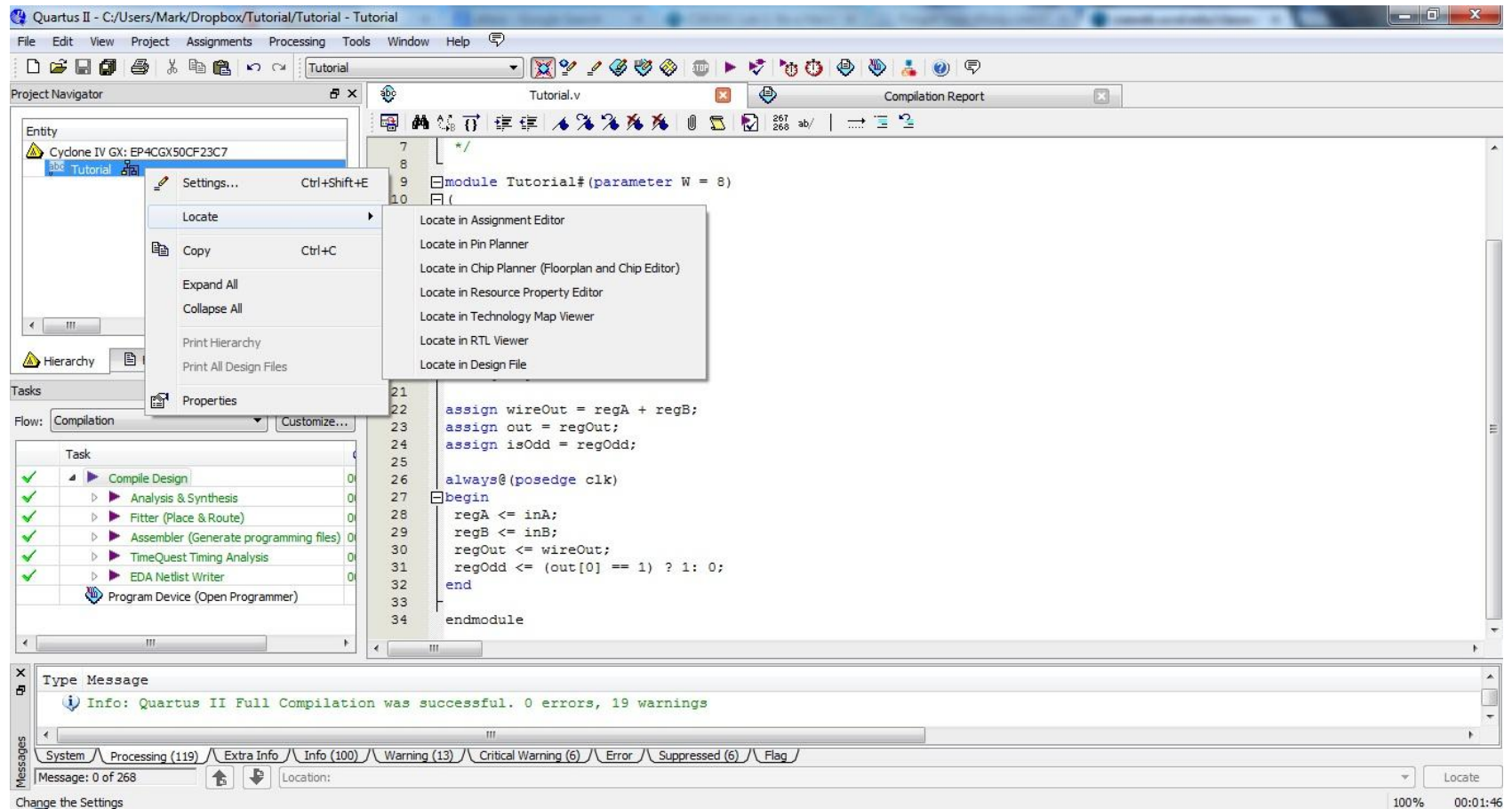
```
always@(posedge clk)
begin
    regA <= inA;
    regB <= inB;
    regOut <= wireOut;
    regOdd <= (out[0] == 1) ? 1: 0;
end
```

```
endmodule
```


Altera Tutorial

- Step 8: Now its time to check to make sure everything is syntactically correct. Go to the tasks tab, and double click on “Analysis & Synthesis”. This will compile the code and show you any errors that may crop up in compilation.
- Step 9: If you want to see how things look at the RTL level, go to the hierarchy tab, right click your base module (in this case: Tutorial), go to Locate, and select “Locate in RTL Viewer”

Altera Tutorial



The screenshot displays the RTL Viewer interface for a Verilog project. The main window shows a circuit diagram with the following components and connections:

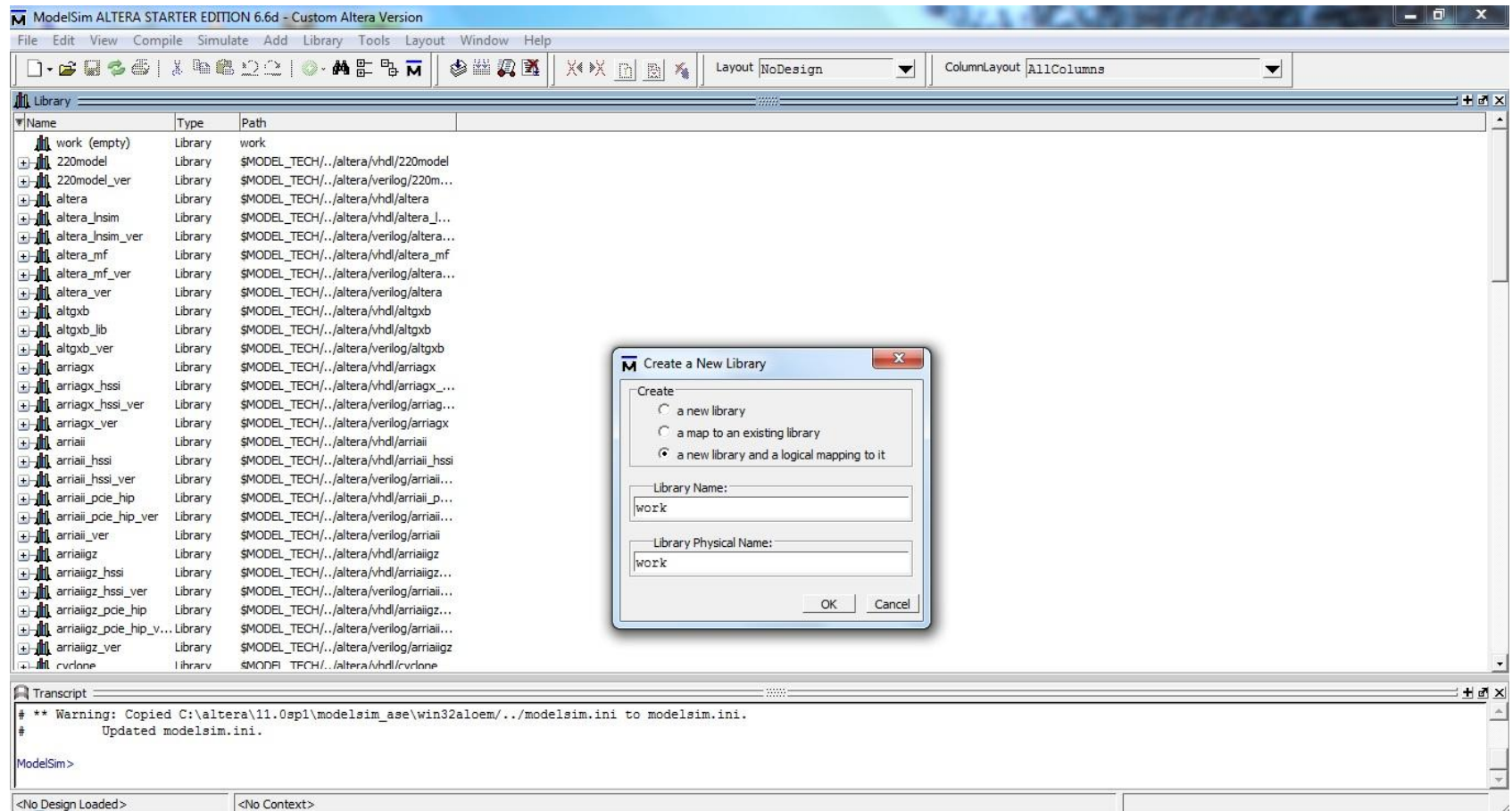
- Registers:**
 - `regA[7..0]`: A D-type flip-flop with `PRE` (preset), `D` (data), `Q` (output), `ENA` (enable), and `CLRn` (clear) inputs.
 - `regB[7..0]`: A D-type flip-flop with `PRE`, `D`, `Q`, `ENA`, and `CLRn` inputs.
 - `regOut[8..0]`: A D-type flip-flop with `PRE`, `D`, `Q`, `ENA`, and `CLRn` inputs.
 - `regOdd`: A D-type flip-flop with `PRE`, `D`, `Q`, `ENA`, and `CLRn` inputs.
- Combinational Logic:**
 - Adder:** A circular block labeled `Add0` and `ADDER` with a plus sign. It takes two 8-bit inputs, `A[8..0]` and `B[8..0]`, and produces an 8-bit output `out[8..0]`.
- Inputs and Outputs:**
 - Inputs:** `inA[7..0]`, `inB[7..0]`, and `clk`.
 - Outputs:** `isOdd` and `out[8..0]`.
- Connections:**
 - `inA[7..0]` and `inB[7..0]` are connected to the `D` inputs of `regA` and `regB` respectively.
 - `clk` is connected to the `CLK` inputs of all registers.
 - The `Q` outputs of `regA` and `regB` are connected to the `A[8..0]` and `B[8..0]` inputs of the adder.
 - The output of the adder (`out[8..0]`) is connected to the `D` input of `regOut`.
 - The `Q` output of `regOut` is connected to the `D` input of `regOdd`.
 - The `Q` output of `regOdd` is connected to the `isOdd` output.

The interface includes a menu bar (File, Edit, View, Tools, Window, Help), a toolbar with various icons, and a sidebar with a 'Find' panel and a 'Netlist Navigator' panel showing the current file 'Tutorial'.

Altera Tutorial

- Step 10: Repeat steps 6-8 for any other modules you wish to create. While we don't show it here, eventually you will have to wire up all your modules either using a schematic viewer or in verilog.
- Step 11: To see how it will work in simulation, we need to use modelsim. Open Modelsim up.
- Step 12: Go to File -> New... -> Library. Create a new library with the radio button "a new library and a logical mapping to it" selected.

Altera Tutorial



Altera Tutorial

- Step 13: Create a file called “test_tutorial.v”, to set up the testbench (see next slide)
- Step 14: Go to compile->compile. Select BOTH the Tutorial.v and test_tutorial.v files to compute.
- Step 15: Go to your work library and double click “test_tutorial.v”
- Step 16: Go to the top bar, change 100ps to 100ns, DO NOT hit run yet.

Altera Tutorial

```
`timescale 1ns / 1ps

module test_tutorial#(parameter W = 8);

    reg          clk;
    reg [W-1:0] a_r;
    reg [W-1:0] b_r;
    wire [W:0] sum;
    wire          is_odd;

    // The design under test is our adder
    Tutorial dut (    .clk(clk)
                    ,.inA(a_r)
                    ,.inB(b_r)
                    ,.out(sum)
                    ,.isOdd(is_odd)
                    );

    // Toggle the clock every 10 ns

    initial
    begin
        clk = 0;
        forever #10 clk = !clk;
    end
```

```
// Test with a variety of inputs.
// Introduce new stimulus on the falling clock edge so that
// values
// will be on the input wires in plenty of time to be read by
// registers on the subsequent rising clock edge.
initial
begin
    a_r = 0;
    b_r = 0;
    @(negedge clk);
    a_r = 1;
    b_r = 1;
    @(negedge clk);
    a_r = 5;
    b_r = 6;
    @(negedge clk);
    a_r = 2;
    b_r = 2;
    @(negedge clk);
    a_r = 3;
    b_r = 3;
    @(negedge clk);
    a_r = 1;
    b_r = 8;
end // initial begin

endmodule // test_adder
```

Altera Tutorial

The screenshot displays the ModelSim ALTERA STARTER EDITION 6.6d interface. The main window is divided into several panes:

- sim** (Left): A tree view showing the design hierarchy. The selected item is `test_tutorial`, which is a Module. Below it, the `dut` (Design Under Test) is shown as a Tutorial Module. The `#INITIAL#36` and `#INITIAL#46` are listed as Processes, and `#vsim_capacity#` is a Foreign entity.
- Objects** (Right): A table showing the current state of the simulation objects.
- Processes (Active)** (Right): A table showing the active processes during the simulation.
- Transcript** (Bottom): A log of the simulation commands and output.

The **Objects** table contains the following data:

Name	Value	Kind	Mode
W	8	Parameter	Internal
clk	0	Register	Internal
a_r	00000001	Packed Array	Internal
b_r	00001000	Packed Array	Internal
sum	000000100	Net	Internal
is_odd	St1	Net	Internal

The **Processes (Active)** table is currently empty.

The **Transcript** shows the following commands and output:

```
# Loading work.Tutorial
VSIM 6> run
VSIM 7>
```

The status bar at the bottom indicates: Now: 100 ns Delta: 3 sim:/test_tutorial

Altera Tutorial

- Step 17: To see the waveform, right click “dut”, go to Add -> To Wave -> All items in region
- Step 18: Now hit run. Right click the waveform and select “Zoom Full” (or press F).
- Step 19: If you did everything properly, the waveform should be displayed as shown here.
- Step 20: Note that the adder has a delay between the input and the correct output. This was by design. Be careful when coding in verilog not to have too many registers in your modules as this can slow down your processor.

Altera Tutorial

ModelSim ALTERA STARTER EDITION 6.6d

File Edit View Compile Simulate Add Structure Tools Layout Window Help

ColumnLayout AllColumns

sim

test_tutorial test_tutorial Module

#INITIA

#INITIA

#vsim_capa

Add

Copy

Find...

Expand Selected

Collapse Selected

Expand All

Collapse All

Code Coverage

Test Analysis

XML Import Hint

Reload sim

End Simulation

Objects

Name	Value
W	8
clk	St0
inA	00000001
inB	00001000
gOut	000000100
regOdd	1
wireOut	000000110

Processes (Active)

Name	Type (filtered)	St
------	-----------------	----

Wave

Msgs	St0	St1
/test_tutorial/dut/clk	00000001	00000010
/test_tutorial/dut/inA	00001000	00000010
/test_tutorial/dut/inB	00000000	00000010
/test_tutorial/dut/out	00000000	00000010
/test_tutorial/dut/is...	00000000	00000010
/test_tutorial/dut/r...	00000011	00000010
/test_tutorial/dut/r...	00000011	00000010
/test_tutorial/dut/r...	00000010	00000010
/test_tutorial/dut/r...	00000010	00000010
/test_tutorial/dut/r...	00000010	00000010
/test_tutorial/dut/w...	000000110	00000010

Now 100000 ps

Cursor 1 0 ps

40000 ps 80000 ps

Transcript

```
VSIM 9> restart -f
VSIM 10> run
VSIM 11>
```

Now: 100 ns Delta: 3

sim:/test_tutorial/dut

QUESTIONS?