This assignment is to create the microbaby components. Each component will be written in VHDL and tested with a testbench using ModelSIM. After this each component will be synthesized in Quartis. Then a report for each component will be done which gives the HDL code, the simulation results, and the synthesis results. This report will be submitted via a dropbox. As these component assignments will be used for Midterm 2 grade, you must work each by yourself will no consultation with your classmates.

The components are:

2-to-1 mux – will be done for you here as an example of what is to be done

2-to-1x8 mux - will be done for you here as an example of what is to be done

1-bit register

* 4-to-1 mux (model it with a single concurrent signal assignment statement similar to the one used for the 2-to-1 mux.
* 4-to-1x8 mux (by multiplexing bit vectors via a selected or conditional signal assignment, not by use a single bit mux)
* All 0’s detector for testing an 8-bit byte to see if it is all 0’s
* full\_adder
* 8-bit adder constructed from 8 instantiations of the full adder
* 8-bit register constructed from 8 instantiations of a 1-bit register
* 8-bit Bus Driver

There are a total of 7 file to write and simulate. The mux2\_1 and mux2\_1x8 and their testbench setups can be accessed via the webpage. There is the core of a testbench for a full adder and an 8-bit adder .

The complete assignments are to be submitted to the corresponding dropbox on CARMEN by Monday 4/11/16 at noon.

For each unit(the 7 listed above) create a word file which looks like the following. Failure to follow this format will result in a loss of points. You will submit it to the correct drop box. The design will be done using std\_logic as the type for signals.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Your name and name.1 Component name

**HDL CODE of Component**

LIBRARY ieee;

USE ieee.std\_logic\_1164.all;

ENTITY mux2\_1 IS

PORT (a,b : IN std\_logic;

sel : IN std\_logic;

r : OUT std\_logic);

END mux2\_1;

ARCHITECTURE one OF mux2\_1 IS

BEGIN

r <= (a AND NOT sel) OR (b AND sel);

END one;

**HDL CODE of Testbench**

LIBRARY ieee;

USE ieee.std\_logic\_1164.all;

ENTITY tb\_mux2\_1 IS

END tb\_mux2\_1;

ARCHITECTURE one OF tb\_mux2\_1 IS

--Declare and Configure DUT

COMPONENT mux2\_1 IS

PORT (a,b : IN std\_logic;

sel : IN std\_logic;

r : OUT std\_logic);

END COMPONENT;

FOR all : mux2\_1 USE ENTITY work.mux2\_1(one);

--Declare signals to hook up DUT

SIGNAL a,b,sel,r : std\_logic;

BEGIN

u0 : mux2\_1 PORT MAP (a,b,sel,r);

--apply stimulus

PROCESS

BEGIN

WAIT FOR 10 ns;

a <= '1'; b <= '0'; sel <= '0';

WAIT FOR 10 ns;

a <= '0'; b <= '0'; sel <= '0';

WAIT FOR 10 ns;

a <= '0'; b <= '1'; sel <= '0';

WAIT FOR 10 ns;

a <= '0'; b <= '0'; sel <= '0';

WAIT FOR 10 ns;

a <= '0'; b <= '1'; sel <= '0';

WAIT FOR 10 ns;

a <= '0'; b <= '0'; sel <= '0';

WAIT FOR 20 ns;

a <= '0'; b <= '1'; sel <= '1';

WAIT FOR 10 ns;

a <= '0'; b <= '0'; sel <= '1';

WAIT FOR 10 ns;

a <= '0'; b <= '1'; sel <= '1';

WAIT FOR 10 ns;

a <= '0'; b <= '0'; sel <= '1';

WAIT FOR 10 ns;

a <= '1'; b <= '0'; sel <= '1';

WAIT FOR 10 ns;

a <= '0'; b <= '0'; sel <= '1';

WAIT FOR 50 ns;

WAIT;

END PROCESS;

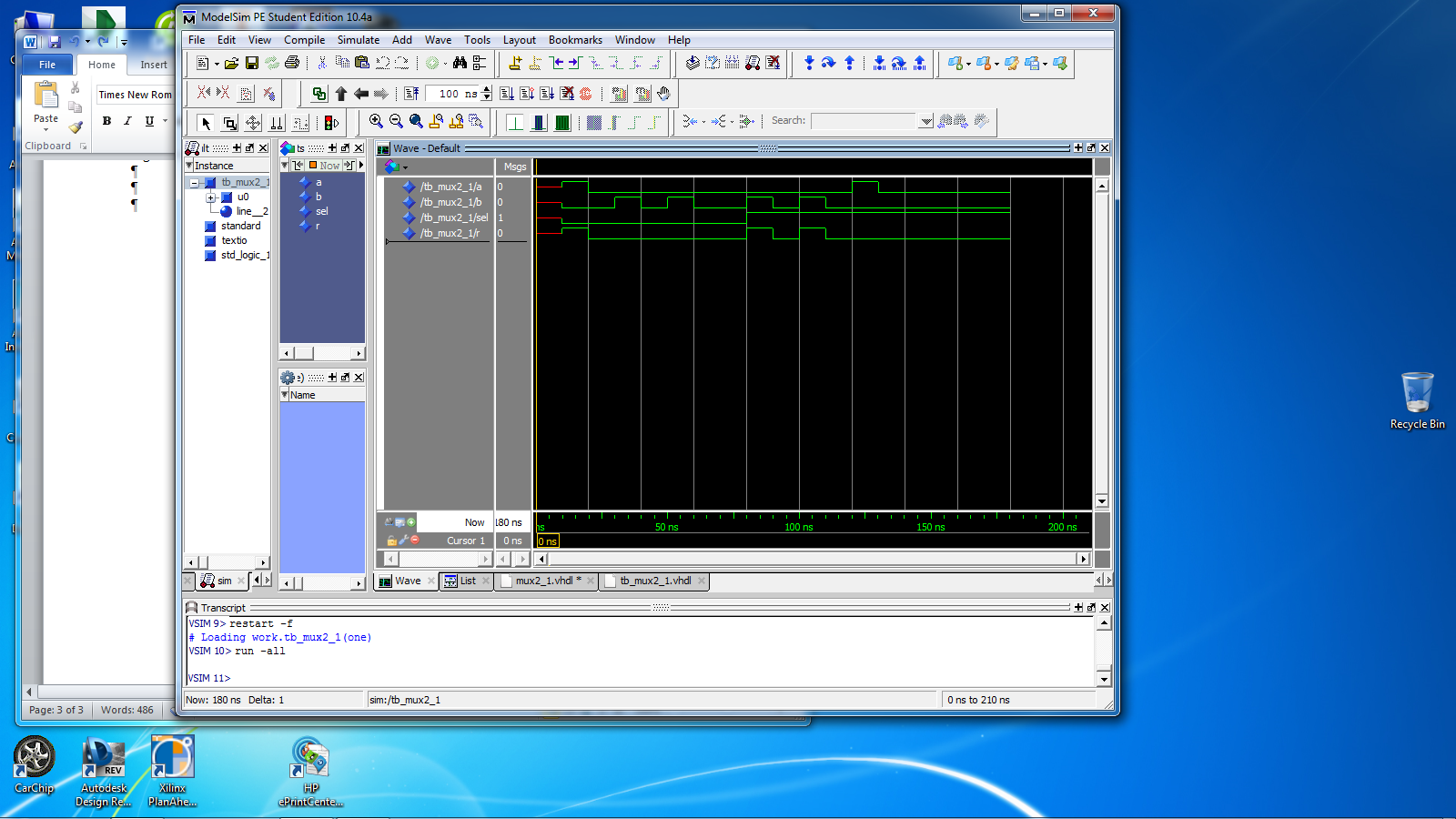
END one;

**SIMULATION RESULTS**

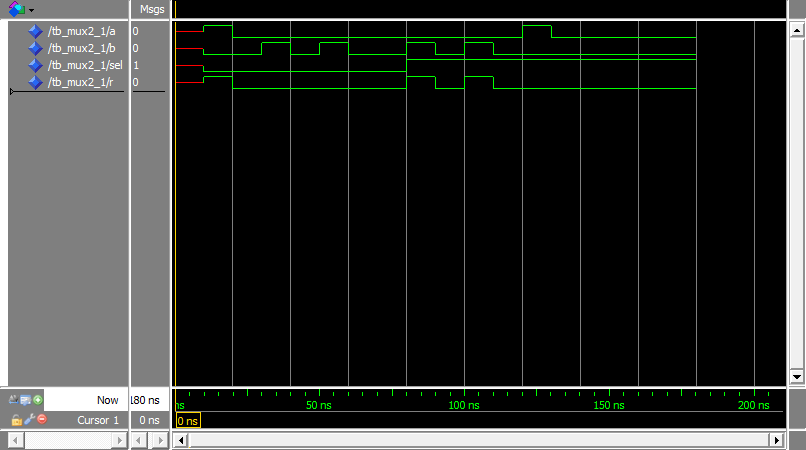
Simulation done on testbench with resolution set to ns.

Signals display are all object in testbench region

Could be done by Fn-Print Screen capture



Or highlighting the simulation window and then using export and getting a bitmap image (preferred)



**The Quartis Synthesis**

The code is put into Quartis and the following results are obtained:

Basic information

Combination ALUTs 1

Registers 0

Pins 4

RTL viewer



**END OF REPORT**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_