BRUTUS BUCKEYE brutus.1 MB comp = mux2\_1

**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;

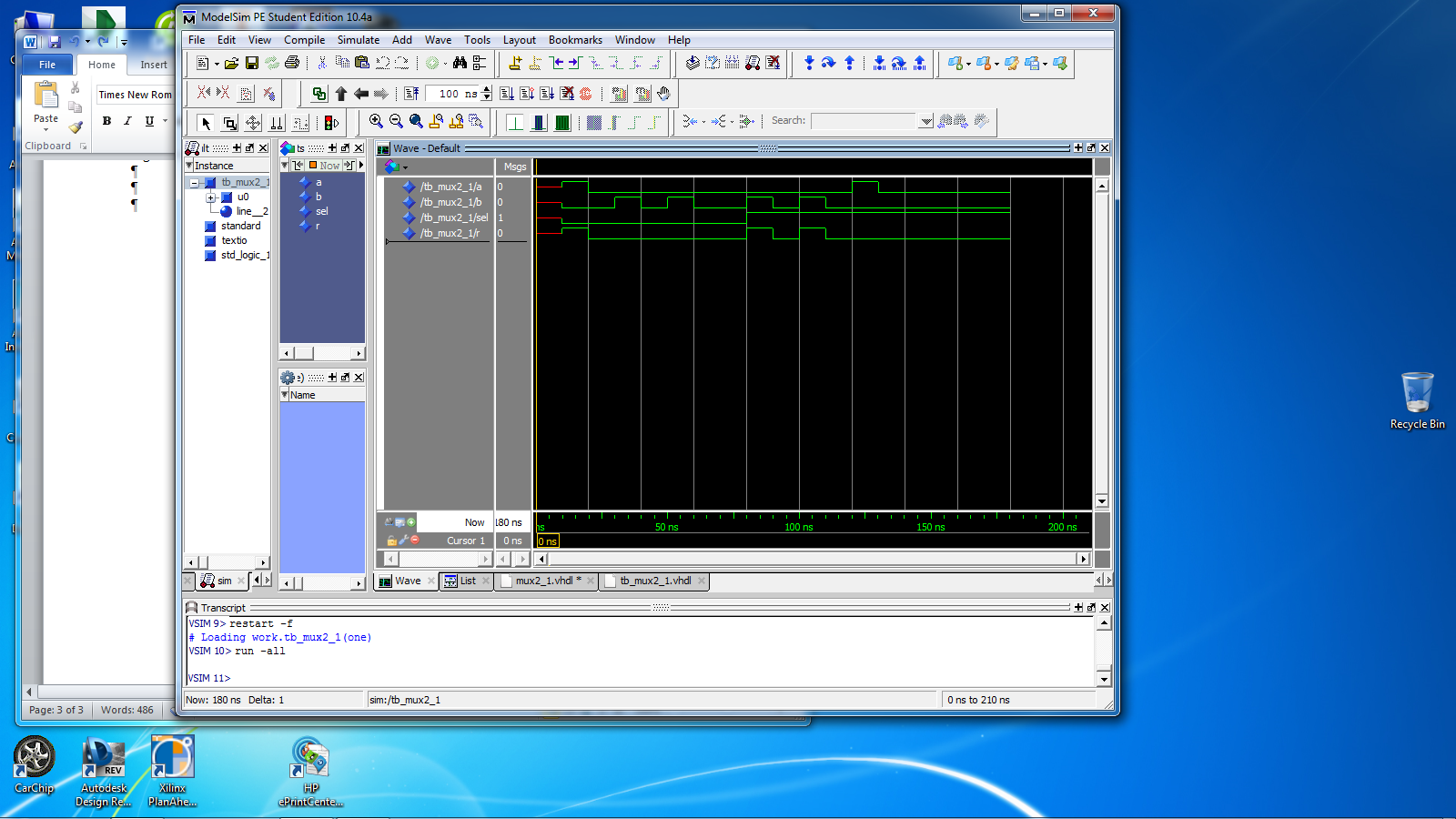
END one;

**SIMULATION RESULTS**

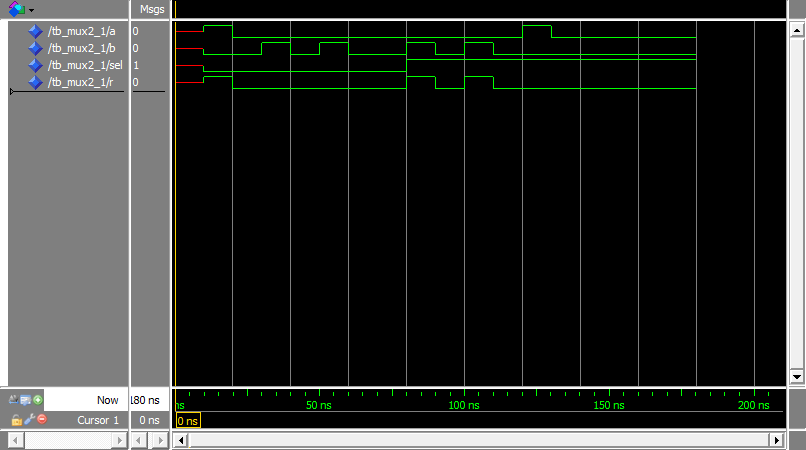
Simulation done on testbench with resolution set to ns.

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**