Assignment : *assignment name* Example follows for half adder

Name

The VHDL code for this assignment follows:

**VHDL code for the half**

-- ENTITY and ARCHITECTURE for a half adder

ENTITY ha IS

PORT (a,b : IN bit;

sum,cout : OUT bit);

END ha;

ARCHITECTURE one OF ha IS

BEGIN

sum <= a XOR b;

cout <= (a AND b);

END one;

**The testbench for the half adder was provided.**

As it so rudimentary the testbench is included here.

-- Testbench for a half adder

-- Adapted from a full adder so a cin signal is available also but not used

ENTITY testha IS

END testha;

ARCHITECTURE one OF testha IS

--Declare component and configure

COMPONENT ha IS

PORT (a,b : IN bit;

sum,cout : OUT bit);

END COMPONENT;

FOR all : ha USE ENTITY work.ha(one);

-- declare signals for stimulus and hookup

SIGNAL a,b,cin,sum,cout : BIT;

BEGIN

-- instantiate component

a1 : ha PORT MAP (a,b,sum,cout);

PROCESS

BEGIN

--WAIT FOR 20 ns;

a <= '0'; b <= '0'; cin<= '0';

WAIT FOR 20 ns;

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

WAIT FOR 20 ns;

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

WAIT FOR 20 ns;

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

WAIT FOR 20 ns;

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

WAIT FOR 20 ns;

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

WAIT FOR 20 ns;

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

WAIT FOR 20 ns;

a <= '1'; b <= '1'; cin<= '1';

WAIT FOR 20 ns;

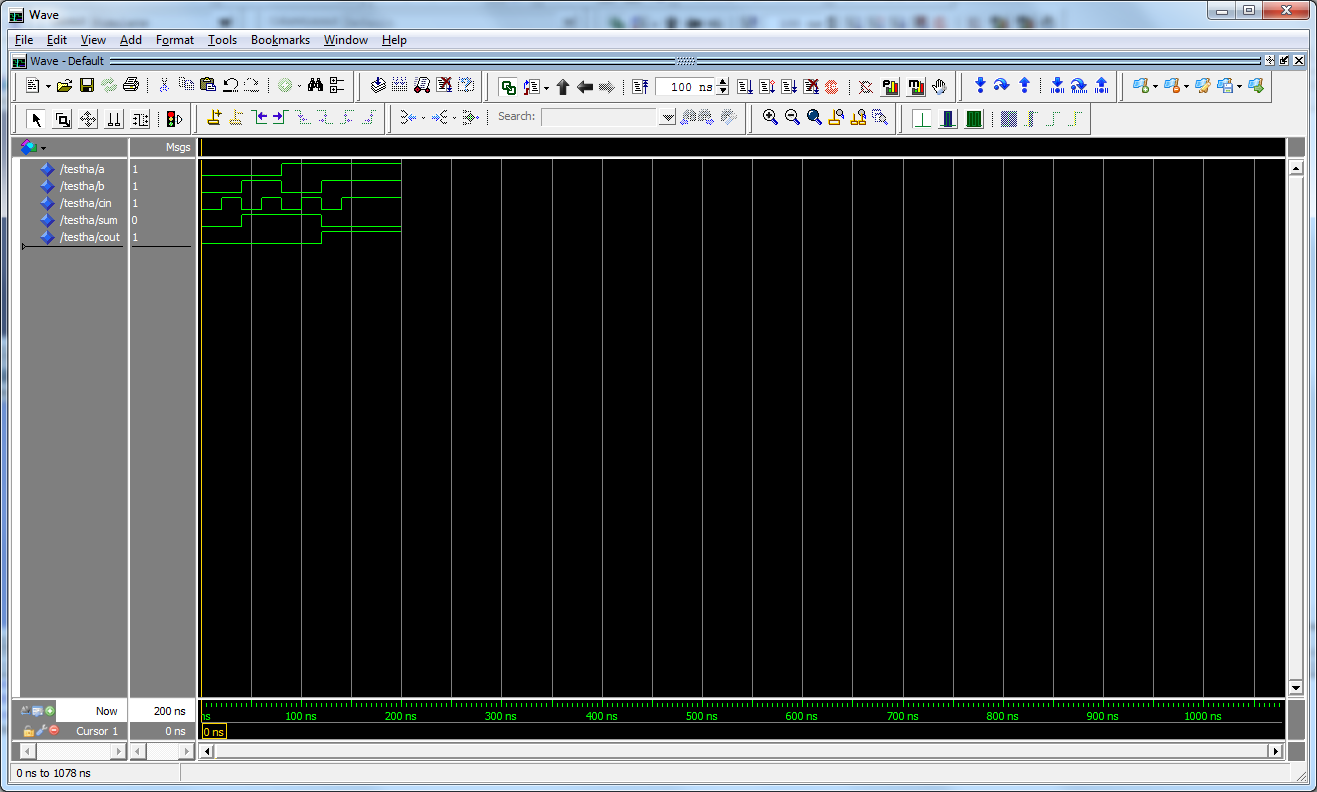
WAIT;

END PROCESS;

END one;

Files were compiles in ModelSim and then simulated.

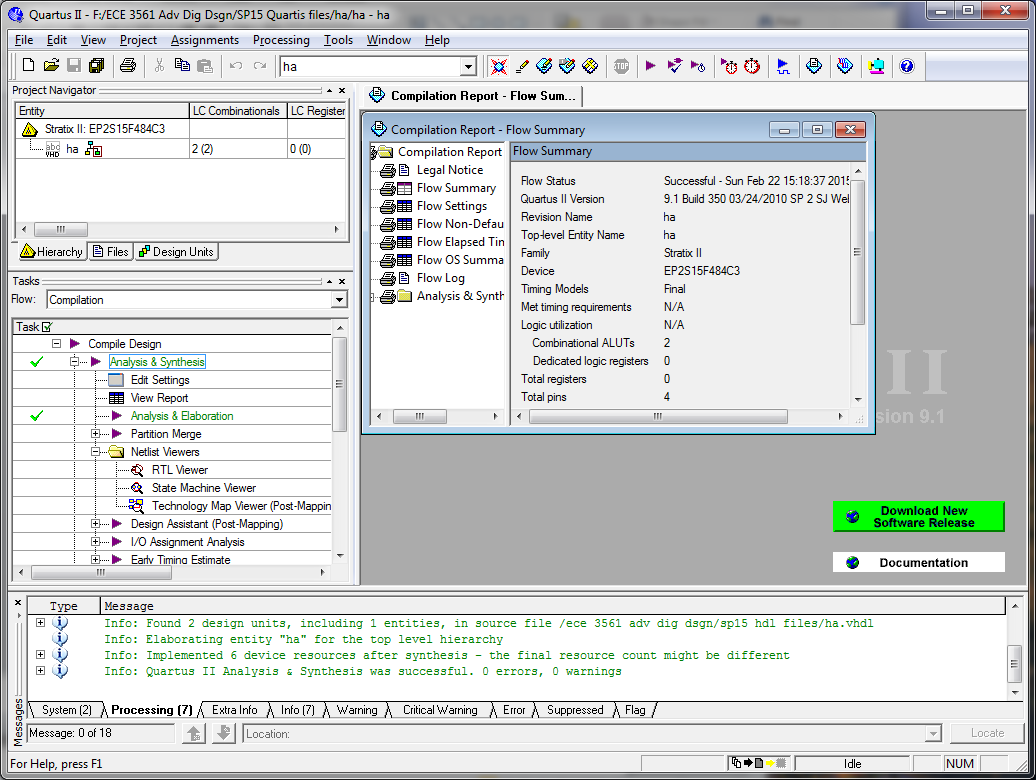
The waveform window was undocked and using ctl-alt-PrintScreen on a PC captured and pasted below.



Quartis work:

Quartis was used to synthesize the design. The wizard was used to link the VHDL files.

Synthesis was run and was successful.



The RTL viewer shows the circuit for the half adder is an OR gate and an AND gate.

