Union College
ECE318/CSC318
Assignment 4

Due Date: February 21st at 10:55a.m.

Note: submit both the code and the simulation output for each problem requiring simulation

1. Read sections 5.4 and 5.5.3 of the text which details the operation of a simple vending machine. Enter the state machine design into Quartus. Quartus will by default adopt a one-hot encoding of states. This can be changed under Settings – Analysis & Synthesis Settings – State Machine Processing. Perform the synthesis using one-hot encoding and using Minimal Bits. Simulate the performance in both cases and note the difference in state transitions. Include comments on the differences you see in the floorplan editor and in the timing analyzer.

2. Come up with the design of the state machine for a sequential parity detector. The operation and code for this are detailed in section 5.5.2 of the text. Your task is to draw the state machine in two formats: (i) state diagram (which you are used to) and (ii) algorithmic state machine.

3. Question 5.5, pg. 113 from the book (note: there is a solution to this problem given in the back of the book but it would be a good test to try and do it yourself first.)

4. Question 5.6, pg. 113 from the book. (note: you will need to use Modelsim for the testbench.)

5. Question 5.7, pg. 113 from the book. Note that this is asking you to come up with equations in a similar manner to what is done on pages 83-85 and 93-95 for two different examples. This is material which was covered in ECE118 so it should be familiar to you. If you run into problems, you are welcome to come and ask me.

6. The following VHDL code describes a 2 to 4 decoder. When synthesized, it produces sequential logic. Simulate the performance using Quartus. Explain the problem and correct the model so that it is combinational. Once the problem has been rectified, redo the simulation to verify that the decoder works correctly. Submit both sets of simulation results.

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
```

```
ENTITY decoder_wrong IS
PORT ( D : IN STD_LOGIC_VECTOR(1 downto 0);
       S : OUT STD_LOGIC_VECTOR(3 downto 0));
END decoder_wrong;

ARCHITECTURE demo OF decoder_wrong IS
BEGIN
PROCESS(D)
        BEGIN
                CASE D IS
                   WHEN "00" =>
                     S(0) <= '1';
                   WHEN "01" =>
                     S(1) <= '1';
                   WHEN "10" =>
                     S(2) <= '1';
                   WHEN "11" =>
                     S(3) <= '1';
                   WHEN OTHERS => null;
                END CASE;
     END PROCESS;

END demo;
```

7. The following VHDL model has both a process and a concurrent signal assignment.

```
library ieee;
use ieee.std_logic_1164.all;

entity test is
port (a, b, c, d: in std_logic;
   w, x, y, z: out std_logic);
end test;

architecture seq of test is
begin

process(c, a, b)
 begin
  if d = '0' then
   w <= a; x <= b; y <= c;
  else
   w <= c; x <= b; y <= a;
  end if;
end process;

z <= a OR b OR c OR d after 2 ns;

end seq;
```

These input transitions occur during the simulation of this model. Show, on a simulation list (similar to what we used in class and that includes the outputs) all the transitions that will occur during the simulation. Remember that when an "after" clause is not included, the default delay is 1 "delta" time.

time	a	b	c	d
0	1	0	0	1
10ns	1	0	0	0
15ns	0	0	0	0