## ECE 4250/ 7250: VHDL and Programmable Logic Devices Laboratory

**Lab # 4**

**Lab Title: Simulation the Design Using Test Bench**

**Group# 8**

**Group Names: Phanindra Sai, Karanam**

**Zhengyu Li**

**Teaching Assistant Use Only:**

**Points Earned Reasons for Deduction**

**Pre-lab:**

**Post Lab report**:

**Demonstration:**

**Final Lab Grade:**

**Comments to students:**

# Objective

To learn writing a test bench for simulating the design, we take a soda vending machine as an example and write a test bench to get the simulation results by using ModelSim.

# Lab Work

### **VMBehave.vhd**

library IEEE;

use IEEE.std\_logic\_1164.all;

entity VM1 is

port(N,D,clk: in std\_logic;

S,N\_ret: out std\_logic);

end VM1;

architecture Behave of VM1 is

type state\_type is (s0,s1,s2,s3); --type of state machine.

signal state: state\_type;

signal n\_state: state\_type;

signal So: std\_logic;

signal inp: std\_logic\_vector(1 downto 0);

begin

inp <= N & D ;

s <= So;

process(clk)

begin

if (clk = '1' and clk 'event) then state <= n\_state;

end if;

end process;

process(inp,state)

begin

N\_ret <= '0';

case state is

when s0 => if(inp = "00") then So <= '0'; n\_state <= s0;

elsif(inp = "10") then So <= '0'; n\_state <= s1;

elsif(inp = "01") then So <= '0'; n\_state <= s2;

end if;

when s1 => if(inp = "00") then So <= '0'; n\_state <= s1;

elsif(inp = "10") then So <= '0'; n\_state <= s2;

elsif(inp = "01") then So <= '0'; n\_state <= s3;

end if;

when s2 => if(inp = "00") then So <= '0'; n\_state <= s2;

elsif(inp = "10") then So <= '0'; n\_state <= s3;

elsif(inp = "01") then So <= '1'; n\_state <= s0;

end if;

when s3 => if(inp = "00") then So <= '0'; n\_state <= s3;

elsif(inp = "10") then So <= '1'; n\_state <= s0;

elsif(inp = "01") then So <= '1'; n\_state <= s0; N\_ret <= '1';

end if;

end case;

end process;

end behave;

### **VMROM.vhd**

library IEEE;

use IEEE.std\_logic\_1164.all;

use IEEE.std\_logic\_arith.all;

use IEEE.std\_logic\_unsigned.all;

entity vm is

port( clk,N, D: in std\_logic;

S,N\_ret: out std\_logic);

end vm;

architecture rom of vm is

type ROM is array (0 to 11) of std\_logic\_vector(3 downto 0);

constant FSM\_rom: ROM := ("0000","0001","0010","0011","0010","0011","1000","1100","0001","0010","0011","1000");

signal q,qp: std\_logic\_vector(1 downto 0):="00";

signal inp: std\_logic\_vector(1 downto 0);

signal ROM\_out: std\_logic\_vector(3 downto 0);

begin

inp <= N & D;

Rom\_out <= FSM\_rom(conv\_integer (inp & q));

qp<= Rom\_out(1 downto 0);

S<= Rom\_out(3);

N\_ret <= Rom\_out(2);

process(clk)

begin

if(clk = '1' and clk'event) then q<=qp; end if;

end process;

end rom;

### **VMTestBench.vhd**

library IEEE;

use IEEE.std\_logic\_1164.all;

entity testvm is

end testvm;

architecture test2 of testvm is

component VM1 is

port( N, D,clk: in std\_logic;

S,N\_ret: out std\_logic);

end component;

component vm is

port( clk,N, D: in std\_logic;

S,N\_ret: out std\_logic);

end component;

constant Nu: integer := 12;

type arr is array(1 to Nu) of std\_logic;

constant n\_arr: arr := ('0','1','1','1','1','0','0','0','0','0','1','0');

constant d\_arr: arr := ('0','0','0','0','0','0','1','1','0','1','0','1');

constant s\_arr: arr := ('0','0','0','1','0','0','0','1','0','0','1','0');

constant nr\_arr: arr := ('0','0','0','0','0','0','0','0','0','0','0','1');

signal n,d,s,n\_r,clk: std\_logic;

begin

clk <= not clk after 50ns;

process

for i in 1 to Nu loop

n <= n\_arr(i);

d <= d\_arr(i);

wait for 100 ns;

assert ( s = s\_arr(i) and n\_r = n\_arr(i))

report "wrong"

severity error;

end loop;

report "test Finished";

end process;

c1: VM1 port map (n,d,clk,s,n\_r);

c2: vm port map (clk,n,d,s1,n\_r1);

end test2;

**State Transition Diagram**

Input (N,D)/Output (ret,disp)

00/00 00/00

10/00

10/01 01/00

01/11 10/00

01/01

10/00

10/00

00/00 00/00

**Waveforms**

# 

# 

# 

# Conclusion

# 

We had learnt how to write a test bench for simulating the design. And we simulate the soda vending machine successfully.