Problem 1

library ieee;

use ieee.std\_logic\_1164.all;

entity Problem1 is

port (clk,x1,x2:in std\_logic;

z:OUT STD\_LOGIC);

end Problem1;

Architecture behave of Problem1 is

signal state,n\_s:integer range 1 to 2;

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

signal s::std\_logic;

begin

q <= x1 & x2;

z <= s;

process(clk)

begin

if (clk = '0' ) then state <= n\_s after 10 ns;

end if;

end process;

process (state,x1,x2)

begin

z<= '0'; --default values

case state is

when 1 => if (q = "00") then n\_s<= 1;

elsif (q = "01") then n\_s <= 2;

elsif (q = "10") then n\_s <= 2;

elsif (q = "11") then n\_s <= 2;

end if;

s <= '0' after 10 ns;

when 2 => if (q = "00") then n\_s<= 2;

elsif (q = "01") then n\_s <= 1;

elsif (q = "10") then n\_s <= 2;

elsif (q = "11") then n\_s <= 1;

end if;

s <= '1' after 10 ns;

when others => NULL;

end case;

end process;

end behave;

Problem 2:

Library ieee;

use ieee.std\_logic\_1164.all;

entity sele is

port (d1,d2:in std\_logic\_vector(7 downto 0);

s:in std\_logic;

y:out std\_logic\_vector(7 downto 0));

end entity sele;

Architecture behave of sele is

begin

y <= d1 when s = '0'

else d2;

end behave;

Library ieee;

use ieee.std\_logic\_1164.all;

entity ff is

port (d:in std\_logic\_vector(7 downto 0);

lda,ldb,clk: in std\_logic;

reg\_a,reg\_b: out std\_logic\_vector(7 downto 0));

end ff

architecture behave of ff is

signal qa,qb:std\_logic\_vector(7 downto 0);

begin

reg\_a <= qa;

reg\_b <= qb;

process(clk)

begin

if (clk'event and clk ='1') then

if lda = '1' then qa <= d;

else qa <= 0；

end if;

if ldb = '1' then qb <= d;

else qb <= 0；

end if;

end if;

end process;

end behave;

Library ieee;

use ieee.std\_logic\_1164.all;

entity problem2 is

port ( clk,se,ld1,ld2: in std\_logic;

da,db:in std\_logic\_vector(7 downto 0);

s1,s2:std\_logic\_vector(7 downto 0));

end problem2 ;

architecture structure of problem2 is

component sele

port (d1,d2:in std\_logic\_vector(7 downto 0);

s:in std\_logic;

y:std\_logic\_vector(7 downto 0));

end component;

component ff

port(d:in std\_logic\_vector(7 downto 0);

lda,ldb,clk: in std\_logic;

reg\_a,reg\_b: out std\_logic\_vector(7 downto 0));

end component;

signal out:std\_logic\_vector(7 downto 0);

begin

g1: sele port map (se, da,db,out);

g2: ff port map (clk1,ld1,out, s1);

g3: ff port map (clk1,ld2,out,s2);

end structure;





