طراحی سیستم های دیجیتال
تمرین سوم
مهندسی کامپیوتر، دانشگاه شهید بهشتی

عباس یزدان مهر
۹۹۲۴۳۰۷۷

پدرام رمضان زاده
۹۹۲۴۳۰۸۵

۱۸ آذر ۱۴۰۱

	assignment	true or false
a	a <= x(2);	false
b	b <= x(2);	true
c	b <= y(3,5);	true
d	b <= w(5)(3);	false
e	y(1)(0) <= z(7);	false
f	x(0) <= y(0,0);	true
g	x <= ”1110000”;	true
h	a <= ”0000000”;	false
i	y(1) <= x;	false
j	w(0) <= y;	false
k	w(1) <= (7=>’1’, OTHERS=>’0’);	true
l	y(1) <= (0=>’0’, OTHERS=>’1’);	false
m	w(2)(7 DOWNTO 0) <= x;	true
n	w(0)(7 DOWNTO 6) <= z(5 DOWNTO 4);	false
o	x(3) <= x(5 DOWNTO 5);	true
p	b <= x(5 DOWNTO 5);	true
q	y <= ((OTHERS=>’0’), (OTHERS=>’0’), (OTHERS=>’0’), ”10000001”);	true
r	z(6) <= x(5);	true
s	z(6 DOWNTO 4) <= x(5 DOWNTO 3);	false
t	z(6 DOWNTO 4) <= y(5 DOWNTO 3);	false
u	y(6 DOWNTO 4) <= z(3 TO 5);	false
v	y(0, 7 DOWNTO 0) <= z;	true
w	w(2,2) <= ’1’;	false

ROM module code:

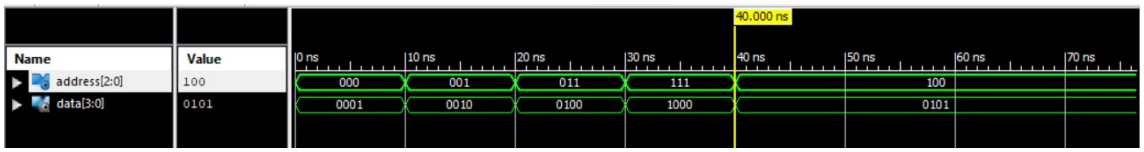
```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.NUMERIC_STD.ALL;

entity rom_8_x_4 is
   generic (
      address_width: integer := 8;
      address_bits: integer := 3;
      data_width: integer := 4
   );
   port (
      address: in std_logic_vector(address_bits-1 downto 0);
      data: out std_logic_vector(data_width-1 downto 0)
   );
end rom_8_x_4;

architecture Behavioral of rom_8_x_4 is
   type rom_type is array (0 to 8-1) of std_logic_vector(4-1 downto 0);
   signal my_rom : rom_type := (
                  "0001",
                  "0010",
                  "0011",
                  "0100",
                  "0101",
                  "0110",
                  "0111",
                  "1000"
     );
begin
   data <= my_rom (to_integer(unsigned(address)));
end Behavioral;
```

Testbench code:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity tb is
end tb;

architecture Behavioral of tb is
   signal address: std_logic_vector(3-1 downto 0);
   signal data: std_logic_vector(4-1 downto 0);
begin
   my_rom8x4: entity work.rom_8_x_4 port map (address => address, data => data);
   address <=  "000",
             "001" after 10 ns,
             "011" after 20 ns,
             "111" after 30 ns,
```

```
            "100" after 40 ns;
end Behavioral;
```

Output:

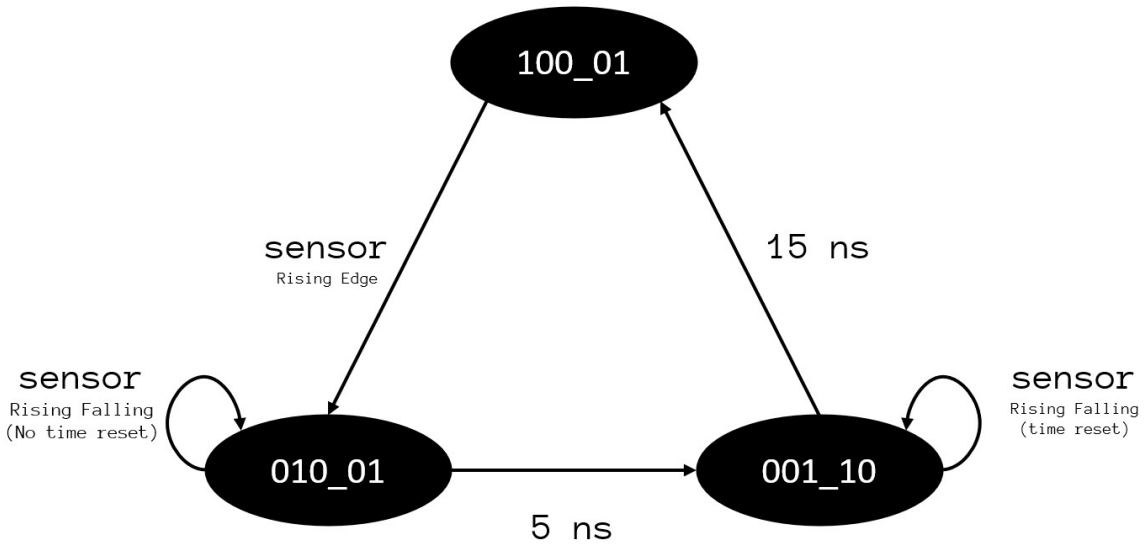


۳

	assignment	result
a	x1 <= a & c;	$x1 <= ”10010”$
b	x2 <= c & b;	$x2 <= ”00101100”$
c	x3 <= b XOR c;	$x3 <= ”1110”$
d	x4 <= a NOR b(3);	$x4 <=' 0'$
e	x5 <= b sll 2;	$x5 <= ”0000”$
f	x6 <= b sla 2;	$x6 <= ”0000”$
g	x7 <= b rol 2;	$x7 <= ”0011”$
h	x8 <= a AND NOT b(0) AND NOT c(1);	$x8 <=' 0'$
i	d <= (5=>’0’, OTHERS=>’1’);	$d <= ”11011111”$

۴

ابتدا باید ماشین حالات مورد نظر سوال را طراحی کنیم که به شکل زیر خواهد بود:



System module code:

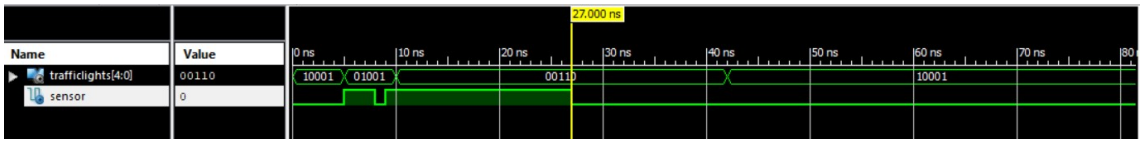
```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity my_system is
   port (
      sensor: in std_logic;
      trafficlights: out std_logic_vector (4 downto 0)
   );
end my_system;

architecture Behavioral of my_system is
   type state_type is (s_g_r, s_y_r, s_r_g); -- s = state, green = g, red = r, yellow = y

   signal state: state_type;


begin

sensor_event:
   process (sensor)
   begin
      -- rising edge
      if sensor'event and sensor = '1' then
         if state = s_g_r then
            state <= s_y_r, s_r_g after 5 ns;
         end if;
      end if;

      -- falling edge
      if sensor'event and sensor = '0' then
         if state = s_r_g then
            state <= s_g_r after 15 ns;
         end if;
      end if;
   end process;

change_state:
   process (state)
   begin
      case state is
         when s_g_r =>   trafficlights <= "10001";
         when s_y_r =>    trafficlights <= "01001";
         when s_r_g =>   trafficlights <= "00110";
      end case;
   end process;

end Behavioral;
```

Testbench code:

```
library IEEE;
```

```
use IEEE.STD_LOGIC_1164.ALL;

entity my_system_tb is
end my_system_tb;

architecture Behavioral of my_system_tb is
   signal trafficlights : std_logic_vector(4 downto 0);
   signal sensor : std_logic;
begin
   mytlc: entity work.my_system port map (trafficlights => trafficlights, sensor => sensor);


   -- sensor <=  '0', '1' after 5 ns,'0' after 20 ns, '1' after 23 ns, '0' after 27 ns;
   -- sensor <= '0', '1' after 5 ns,'0' after 8 ns, '1' after 23 ns, '0' after 27 ns;
   sensor <= '0', '1' after 5 ns,'0' after 8 ns, '1' after 9 ns, '0' after 27 ns;


end Behavioral;
```

Output:



پایان