(10)

```
ENTITY counter IS
    PORT(
        rst    : IN std_logic;
        clk1   : IN std_logic;
        clk2   : IN std_logic;
        output : OUT std_logic_vector(7 DOWNTO 0)
    );
END counter;

ARCHITECTURE amir of counter IS
    SIGNAL con : std_logic_vector(7 DOWNTO 0);
    BEGIN
        PROCESS(clk1, clk2)
        BEGIN
            IF rst = '1' THEN
                con <= (others => '0')
            ELSIF (clk1 = '1' AND clk1'EVENT) OR
                  clk2 = '1' AND clk2'EVENT) )
                con <= con + '1';
            END IF;
            output <= con;
        END PROCESS;
END amir;
```

امیرحسین بیدگلی ۹۷۲۴۳۰۲۳

سوال ۲)

```
ENTITY mem2Port IS

PORT(
    clk     : IN  std_logic;
    din1    : IN  std_logic_vector(15 DOWNTO 0);
    addr1   : IN  std_logic_vector(7 DOWNTO 0);
    rw1     : IN  std_logic;
    din2    : IN  std_logic_vector(15 DOWNTO 0);
    addr2   : IN  std_logic_vector(7 DOWNTO 0);
    rw2     : IN  std_logic;
    dout1   : OUT std_logic_vector(15 DOWNTO 0);
    dout2   : OUT std_logic_vector(15 DOWNTO 0);
    OP_ok   : OUT std_logic
);

END mem2Port;
```

ادامه در صفحه بعد

```
ARCHITECTURE amir of mem2Port IS
    TYPE memoryType IS Array(0 TO 63) OF
                        std_logic_vector(15 DOWNTO 0);
    SIGNAL memory_singal : memoryType;
    BEGIN
    PROCESS(clk)
    BEGIN
        IF clk = '1' THEN
            IF (rw1 = '1' AND rw2 = '1') THEN
                dout1 <= memory_singal(conv_integer(addr1));
                dout2 <= memory_singal(conv_integer(addr2));
                op_ok <= '1';
            ELSIF (rw1 = '0' AND rw2 = '0') THEN
                memory_signal(conv_integer(addr1)) <= din1;
                memory_signal(conv_integer(addr2)) <= din2;
                op_ok <= '0';
            ELSIF (rw1 = '0' AND rw2 = '1') THEN
                memory_singal(conv_integer(addr1)) <= din1;
                dout2 <= memory_singal(conv_integer(addr2));
                op_ok <= '1';
            ELSE
                memory_signal(conv_integer(addr2)) <= din2;
                dout1 <= memory_signal(conv_integer(addr1));
                op_ok <= '1';
            END IF;

        END IF;
    END PROCESS;

END amir;
```

سؤال 3)

الف) زیرا هنگامی که وارد Process می‌شویم، ابتدا مقادیری که در پایین Process قرار دارند آپدیت می‌شوند و شبیه‌سازی نیز متوقف می‌شود.

ب) زیرا سخت‌افزار باید از تعداد سخت‌های سنتز شده مطلع باشد.

ج) WAIT ON و WAIT UNTIL می‌توانند سنتز شوند.
wait FOR و wait نمی‌توانند ← زیرا delay را به ما می‌گویند.

د) BUFFER نمی‌تواند بیش از یک منبع داشته باشد. اما INOUT می‌تواند.

(4 ج) السؤال

```
ENTITY MUX8TO1 IS
    PORT(
        input  : IN std_logic_vector (0 DOWNTO 7);
        select : IN std_logic_vector ( 0 DOWNTO 3);
        output : OUT std_logic
    );
END MUX8TO1;
ARCHITECTURE amir of MUX8TO1 IS
    BEGIN
        CASE select IS
            WHEN "000" => 0 <= input(0);
            WHEN "001" => 0 <= input(1);
            WHEN "010" => 0 <= input(2);
            WHEN "011" => 0 <= input(3);
            WHEN "100" => 0 <= input(4);
            WHEN "101" => 0 <= input(5);
            WHEN "110" => 0 <= input(6);
            WHEN "111" => 0 <= input(7);
        END CASE;
END amir;
```

سوال 5)

```
ENTITY 95 IS
    PORT(
        input    : IN  std_logic;
        clk      : IN  std_logic;
        output   : OUT std_logic
        );
END 95;
ARCHITECTURE amir of 95 IS
    PROCESS
      BEGIN
          WAIT ON clk;
          IF ck = '1' THEN
              output <= input;
          END IF;
    END PROCESS;
END amir;
```

```vhdl
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
USE ieee.std_logic_arith.ALL;
USE ieee.std_logic_unsigned.ALL;
ENTITY mult IS
    PORT (
        n1, n2 : IN STD_LOGIC_VECTOR(1 DOWNTO 0);
        pd     : OUT STD_LOGIC_VECTOR(3 DOWNTO 0)
        );
END mult;
ARCHITECTURE test OF mult IS
    SIGNAL n1_reg : STD_LOGIC_VECTOR(2 DOWNTO 0);
    SIGNAL pd_reg : STD_LOGIC_VECTOR(5 DOWNTO 0);
BEGIN
    PROCESS (n1, n2)
    BEGIN
        n1_reg <= '0' & n1;
        pd_reg <= "0000" & n2;
        FOR i IN 1 TO 2 LOOP --باید از 1 تا 2 باشد.
            IF pd_reg(0) = '1' THEN
                pd_reg(5 DOWNTO 3) <= pd_reg(5 DOWNTO 3) + n1_reg(2 DOWNTO 0);
            END IF;
            pd_reg(5 DOWNTO 0) <= '0' & pd_reg(5 DOWNTO 1); --  هنگامی که سیگنال داریم باید به صورت => مقدار دهی شود.  using & instead of AND
        END LOOP;
            pd <= pd_reg(5 DOWNTO 2); --چون نمیتوانیم 6 بیت را در 4 بیت بریزیم باید بخشی از آن را بریزیم.

    END PROCESS;
END test;
```