

## דוגמא לשאלות במבחן + פיתרון מלא

### שאלה 1 (50%):

תכנן בשפת VHDL, מכונת מצבים המקבלת בכניסתה רצף של ביטים, אוספת כל ארבעה ביטים ומקודדת אותם מ BCD ל Seven Segment. ההמרה תעשה כל ביט רביעי ותוציא 7 ביטים במוצא.  
א (20). שרטט דיאגרמת מצבים של ASM.  
ב (30). כתוב קוד מלא ב VHDL ב Process יחיד.  
השתמש רק בספרייה הנומרית הסטנדרטית לסינתזה (ieee.numeric\_std).

### שאלה 2 (50%):

א (30). תכנן בשפת VHDL, במודל התנהגותי, מעגל צירופי המקבל בכניסתו שני מספרים בני 4 ביטים, בייצוג בינארי ללא סימן, ומוציא את ריבוע הנורמה שלהם ברוחב 8 ביטים. במילים אחרות המעגל מקבל בכניסה שני מספרים: A ו B ומוציא  $Z = A^2 + B^2$ .  
השתמש רק בספרייה הנומרית הסטנדרטית לסינתזה (ieee.numeric\_std).

א (10). כתוב פיתרון לסימולציה בלבד בשורה אחת.

ב (30). כתוב מודל יעיל ומקוצר הניתן לסנתוז כאשר אופרטור הכפל לא ניתן לסנתוז. פתור את השאלה המבוקשת פתרונות אחרים לא יבדקו.

ג (10). האם יש קושי בסנתוז מעגל כזה? אם כן מהו הקושי?

```

1 library ieee;
2 use ieee.std_logic_1164.all;
3 use ieee.numeric_std.all;
4
5 entity XControl is port (
6     clk      : in std_logic;
7     reset    : in std_logic;
8     Z        : out std_logic_vector(6 downto 0);
9     Y        : in std_logic);
10 end XControl;
11
12 architecture Flaxer of XControl is
13     TYPE StatusType IS (X0, X1, X2, X3);
14     -----
15     signal XState :StatusType;
16     begin
17     -----
18     Status: process(reset, clk, Y)
19     variable Temp :std_logic_vector(3 downto 0);
20     begin
21         IF reset = '1' THEN
22             XState <= X0;
23             Z <= (OTHERS => '0');
24         ELSIF (clk'event and clk = '1') THEN
25             CASE XState IS
26                 WHEN X0 =>
27                     XState <= X1;
28                     Temp(0) := Y;
29                 WHEN X1 =>
30                     XState <= X2;
31                     Temp(1) := Y;
32                 WHEN X2 =>
33                     XState <= X3;
34                     Temp(2) := Y;
35                 WHEN X3 =>
36                     XState <= X0;
37                     Temp(3) := Y;
38                 CASE Temp IS
39                     WHEN "0000" => Z <= "0111111";
40                     WHEN "0001" => Z <= "0000110";
41                     WHEN "0010" => Z <= "1011011";
42                     WHEN "0011" => Z <= "1001111";
43                     WHEN "0100" => Z <= "1100110";
44                     WHEN "0101" => Z <= "1101101";
45                     WHEN "0110" => Z <= "1111101";
46                     WHEN "0111" => Z <= "0000111";
47                     WHEN "1000" => Z <= "1111111";
48                     WHEN "1001" => Z <= "1101111";
49                     WHEN others => Z <= "1000000"; -- Out of rang
50                 END CASE;
51             END CASE;
52         END IF;
53     end process;
54
55
56     -----
57 end Flaxer;
58
59

```

```

1 library ieee;
2 use ieee.std_logic_1164.all;
3 use ieee.numeric_std.all;
4
5 ENTITY Mul IS
6 GENERIC (N: integer := 4);
7 PORT (
8   X1, X2 : IN  unsigned(N-1 downto 0);
9   Y      : OUT unsigned(2*N-1 downto 0)
10  );
11 END Mul;
12
13 ARCHITECTURE Multiplier OF Mul IS
14   SIGNAL A1, A2 : unsigned(N-1 downto 0);
15 BEGIN
16   A1 <= X1;
17   A2 <= X2;
18
19   PROCESS (A1, A2)
20     VARIABLE SUM1: unsigned(2*N-1 downto 0);
21     VARIABLE SUM2: unsigned(2*N-1 downto 0);
22     VARIABLE TMP:  unsigned(2*N-1 downto 0);
23
24     BEGIN
25       SUM1 := (others => '0');
26       FOR i IN 0 to N-1 LOOP
27         IF A1(i) = '1' THEN
28           TMP := (others => '0');
29           TMP(N-1+i downto i) := A1;
30           SUM1 := SUM1 + TMP;
31         END IF;
32       END LOOP;
33
34       SUM2 := (others => '0');
35       FOR i IN 0 to N-1 LOOP
36         IF A2(i) = '1' THEN
37           TMP := (others => '0');
38           TMP(N-1+i downto i) := A2;
39           SUM2 := SUM2 + TMP;
40         END IF;
41       END LOOP;
42
43       Y <= SUM1 + SUM2;
44
45     END PROCESS;
46 END ARCHITECTURE Multiplier;
47
48 END ARCHITECTURE Multiplier;
49
50

```