מדינת ישראל
משרד החינוך

סוג הבחינה: גמר לבתי־ספר לטכנאים ולהנדסאים
מועד הבחינה: אביב תשע"א, 2011
סמל השאלון: 711003
נספחים: א. נוסחאון באלקטרוניקה תקבילית ב' לכיתה י"ד
ב. נוסחאון בשפת תיאור חומרה VHDL לכיתה י"ד

אלקטרוניקה ומחשבים ה'

מגמת הנדסת אלקטרוניקה ומחשבים
(כיתה י"ד)

הוראות לנבחן

א. משך הבחינה: ארבע שעות.

ב. מבנה השאלון ומפתח ההערכה: בשאלון זה שני פרקים, ובהם שמונה שאלות.
יש להשיב על **ארבע שאלות בלבד, שאלה אחת לפחות מכל פרק**.
לכל שאלה – 25 נקודות. סך־הכול – 100 נקודות.

ג. חומר עזר מותר לשימוש: מחשבון.

ד. הוראות מיוחדות:

1. ענה על מספר השאלות הנדרש בשאלון. המעריך יקרא ויעריך את מספר השאלות הנדרש בלבד, לפי סדר כתיבתן במחברתך, ולא יתייחס לתשובות נוספות.
2. התחל כל תשובה לשאלה חדשה בעמוד חדש.
3. רשום את כל תשובותיך **אך ורק בעט**.
4. הקפד לנסח את תשובותיך כהלכה ולסרטט את תרשימיך בבהירות.
5. כתוב את תשובותיך בכתב־יד ברור, כדי לאפשר הערכה נאותה של תשובותיך.
6. אם לדעתך חסרים נתונים הדרושים לפתרון שאלה, אתה רשאי להוסיף אותם, בתנאי שתנמק מדוע הוספת אותם.
7. בכתיבת פתרונות חישוביים, קבלת מֵרַב הנקודות מותנית בהשלמת כל המהלכים שלהלן, בסדר שבו הם רשומים:
   * רישום הנוסחה המתאימה.
   * הצבה של כל הערכים ביחידות המתאימות.
   * חישוב (אפשר באמצעות מחשבון).
   * רישום התוצאה המתקבלת, יחד עם יחידות המידה המתאימות.
   * ליווי הפתרון החישובי בהסבר קצר.

בשאלון זה 10 עמודים ו־24 עמודי נספחים.

ההנחיות בשאלון זה מנוסחות בלשון זכר,
אך מכוונות לנבחנות ולנבחנים כאחד.

בהצלחה!

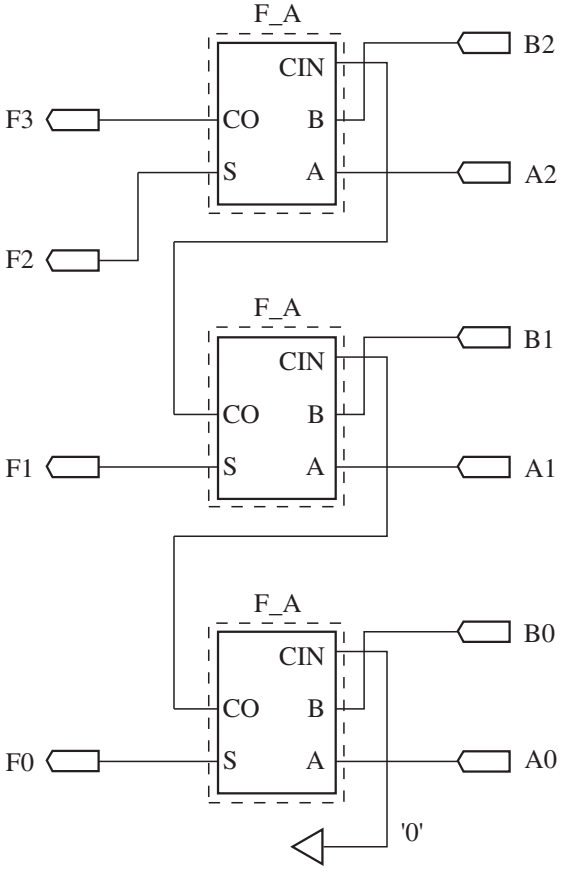
פרק שני: שפת תיאור חומרה VHDL

ענה על שאלה אחת לפחות מבין השאלות 5–8 (לכל שאלה – 25 נקודות).

שאלה 5

באיור לשאלה 5 מתוארת מערכת המבצעת פעולת חיבור בין שני מספרים (A , B) , שכל אחד מהם בעל שלוש סיביות.

החיבור מתבצע באמצעות שְׁרשׁוּר מחברים מלאים (FULL ADDERS).



איור לשאלה 5

המשך בעמוד 7

א. כתוב תכנית בשפת VHDL המבצעת תיאור מבני לחיבור בין מרכיבי המערכת (COMPONENTS) . השתמש בפקודת PORT MAP לצורך החיבור בין מרכיבי המערכת.

<u>הערות:</u>

* הסתמך על כך שתכניות ה־F_A כתובות, מהודרות ומכילות את הישות הבאה:

```
entity F_A is
port (A,B,Cin : in bit;
      S,Co : out bit );
end;
```

* השימוש בפקודה GENERATE מומלץ – אך אינו חובה.

ב. העתק למחברתך את סרטוט המערכת, והצג בו את מיקומו ושמו של כל אחד מן האותות (SIGNALS) שבחרת לצורך החיבור המבני שבסעיף א'.

המשך בעמוד 8

שאלה 6

לפניך התכנית TAR6 , הכתובה בשפת VHDL :

```
1.    library ieee;
2.    use ieee. std_logic_1164.all;
3.    entity TAR6 is
4.    Port ( A , B , Cin : in bit );
5.           S , Co   : out bit );
6.    end;
7.    architecture behave of TAR6 is
8.    signal T : bit_vector (2 downto 0);
9.    signal Y : bit_vector (1 downto 0);
10.   begin
11.   T  <= A & B & Cin;
12.   Co <= Y (1) ;
13.   S  <= Y (0) ;
14.   With T Select
15.         Y<=   "00" when "000",
16.         "01" when "001" | "010" | "100" ,
17.         "11" when "111",
18.         "10" when others;
19.   end behave;
```

<u>הערה:</u> בשורה 16 בתכנית מופיע האופרטור | , שמשמעותו פעולה לוגית OR בתוך ההתניה.

א. הסבר את הפעולה המתבצעת בשורה 11 בתכנית.

ב. העתק למחברתך את הטבלה הבאה, והשלם בה את מצב מוצאי המערכת בהתאם למצב המבואות שלה.

המשך בעמוד 9

A	B	Cin	Co	S
0	0	0		
0	0	1		
0	1	0	0	1
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

שאלה 7

באיור לשאלה 7 מתוארת מערכת לזיהוי הקוד "110" , המגיע אליה באופן טורי משמאל לימין (סיבית אחת בכל עליית שעון).

איור לשאלה 7

הסיביות נכנסות למערכת דרך המבוא הטורי SI באופן אקראי .

ערכו של המוצא Z יהיה '1' רק לאחר קליטת הקוד, כמתואר בדוגמה הבאה:

מבוא SI	0	**1**	**1**	**0**	1	0	1	**1**	**1**	**0**
מוצא Z	0	0	0	**1**	0	0	0	0	0	**1**

כתוב תכנית בשפת VHDL המבצעת את פעולת המערכת הזו.

<u>הערה:</u> השימוש במכונת מצבים מומלץ – אך אינו חובה.

המשך בעמוד 10

שאלה 8

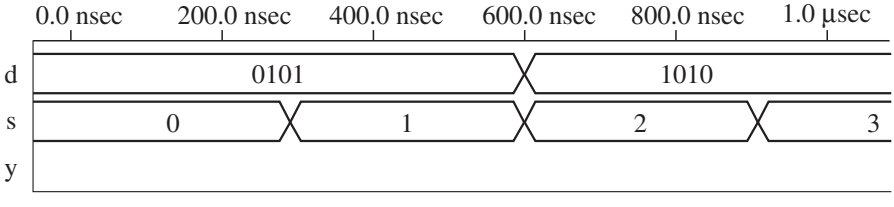
לפניך התכנית TAR8 , הכתובה בשפת VHDL :

```
1.  library ieee;
2.  use ieee.std_logic_1164.all;
3.  entity TAR8 is
4.  generic ( m : integer :=4 );
5.  port ( d : in bit_vector(m-1 downto 0) ;
6.         s : in integer range 0 to m-1 ;
7.         y : out bit) ;
8.  end;
9.  architecture behave of TAR8 is
10. begin
11. y <= d(s);
12. end behave;
```

א. סרטט את צורת המבנית (SYMBOL) הנוצרת כתוצאה מהישות (ENTITY) בתכנית TAR8 . ציין בתשובתך את מספר הסיביות בכל מבוא ובכל מוצא של המערכת.

ב. הסבר את ההוראה שבשורה 4 בתכנית, ואת משמעותה לגבי התכנית כולה.

ג. באיור לשאלה 8 מתוארת דיאגרמת זמן. העתק אותה למחברתך, והשלם בה את מצב המוצא y בהתאם למצב המבואות d ו־s .



איור לשאלה 8

בהצלחה!

זכות היוצרים שמורה למדינת ישראל.
אין להעתיק או לפרסם אלא ברשות משרד החינוך.

מדינת ישראל
משרד החינוך

סוג הבחינה: גמר לבתי־ספר לטכנאים ולהנדסאים
מועד הבחינה: אביב תשע"א, 2011
נספח לשאלון: 711003

אין להעביר את הנוסחאון לנבחן אחר

מקום למדבקת נבחן

נוסחאון בשפת תיאור חומרה VHDL לכיתה י"ד

(12 עמודים)

בלוקים עיקריים בשפה	
מבנה כללי של ישות תכנית	**ENTITY**__entity_name **IS** **GENERIC** (parameter_name : string := default_value; parameter_name : integer := default_value**);** **PORT** (input_name : **IN STD_LOGIC**; input_vector_name : **IN BIT_VECTOR** (high downto low); bidir_name : **INOUT STD_LOGIC**; output_name : **OUT STD_LOGIC** **);** **END**__entity_name;

המשך בעמוד 2

בלוקים עיקריים בשפה	
ARCHITECTURE a **OF** __entity_name **IS** הצהרה על משאבי התוכנית **BEGIN** -- Process Statement -- Concurrent Procedure Call -- Concurrent Signal Assignment -- Conditional Signal Assignment -- Selected Signal Assignment -- Component Instantiation Statement -- Generate Statement **END** a;	**מבנה כללי של גוף תכנית**
__process_label: **PROCESS** (__signal_name, __signal_name) **VARIABLE** __variable_name : **STD_LOGIC;** **VARIABLE** __variable_name : **STD_LOGIC;** **BEGIN** -- Signal Assignment Statement -- Variable Assignment Statement -- Procedure Call Statement -- If Statement -- Case Statement -- Loop Statement **END PROCESS** __process_label;	**מבנה כללי של הליך טורי**

המשך בעמוד 3

שימוש במבניות בתכנון היררכי	
COMPONENT __component_name **GENERIC (**__parameter_name : string := __default_value; __parameter_name : integer := __default_value **);** **PORT (** input_name, input_name **: IN STD_LOGIC;** bidir_name, bidir_name **: INOUT STD_LOGIC;** output_name, output_name **: OUT STD_LOGIC);** **END COMPONENT;**	**הצהרה על מבנית שבשימוש בתכנית האב**
__instance_name: __component_name **GENERIC MAP** (parameter_name => parameter_value, parameter_name => parameter_value) **PORT MAP** (component_port => connect_port, component_port => connect_port);	**שימוש במבנית בתכנון היררכי**
__generate_label: **FOR** __index_variable **IN** __range **GENERATE** __statement; __statement; **END GENERATE** __generate_label;	**שרשור מבניות GENERATE בלולאת FOR**
__generate_label: **IF** __expression **GENERATE** __statement; __statement; **END GENERATE** __generate_label;	**שרשור מותנה IF GENERATE**

המשך בעמוד 4

LOOPS לולאות	
__loop_label: **FOR** __index_variable **IN** __range **LOOP** __statement; __statement; **END LOOP** __loop_label;	**לולאת FOR**
__loop_label: **WHILE** __boolean_expression **LOOP** __statement; __statement; **END LOOP** __loop_label;	**לולאת WHILE**

המשך בעמוד 5

התניות בגוף התכנית – מחוץ ל־PROCESS	
__signal <= __expression **WHEN** __boolean_expression **ELSE** __expression **WHEN** __boolean_expression **ELSE** __expression _____ ; **דוגמה 1 :** מימוש שער AND באמצעות פקודת "WHEN" פשוטה. Y <= '1' when (a = '1') and (b = '1') else '0' ; **דוגמה 2 :** מימוש שער AND באמצעות פקודת "WHEN" המשכית. Y <= '0' when (a = '0') and (b = '0') else '0' when (a = '0') and (b = '1') else '0' when (a = '1') and (b = '0') else '1' ;	**התניית** **When ... Else**

WITH __ expression **SELECT**
__signal <= __expression **WHEN** __constant_value,
__expression **WHEN** __constant_value,
__expression **WHEN** __constant_value,
__expression **WHEN** __constant_value,
__expression **WHEN** others;
_____
דוגמה: מימוש שער AND באמצעות פקודת התניה WITH .. SELECT
Signal ab : bit_vector (1 downto 0);
Begin
ab <= a & b;
 	
with ab select
y <= '0' when "00" ,
'0' when "01" ,
'0' when "10",
'1' when others ;
	התניית
	With ... Select

המשך בעמוד 6

התניות בהליך טורי – בתוך PROCESS

התניית

If .. Then .. Else

```
IF      __boolean_expression THEN
        __signal <= __ expression;
ELSIF   __boolean_expression THEN
        __signal <= __expression;
ELSE
        __signal <= __expression;
END IF;
```

התניית CASE

```
CASE __expression IS
    WHEN __constant_value =>
        __statement;
        __statement;
    WHEN __constant_value =>
        __statement;
        __statement;
    WHEN OTHERS =>
        __statement;
        __statement;
END CASE;
```

דוגמה למימוש מפענח בהתניית CASE :

```
Process(s)
Begin
    Case  s  is
            When "00" =>   y <= "0001" ;
            When "01" =>   y <= "0010" ;
            When "10" =>   y <= "0100" ;
            When "11" =>   y <= "1000" ;
    End case;
End process;
```

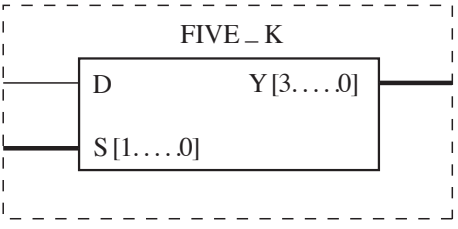
המשך בעמוד 7

התניות בהליך טורי – בתוך PROCESS

התוכנית תבצע את טבלת האמת הבאה:

S : BIT_VECTOR (1 DOWNTO 0)		Y : BIT_VECTOR (3 DOWNTO 0)			
S1	S0	Y3	Y2	Y1	Y0
0	0	0	0	0	**1**
0	1	0	0	**1**	0
1	0	0	**1**	0	0
1	1	**1**	0	0	0

דוגמה לתכנית שלמה המבצעת פעולת מפלג, תוך שימוש בהתניית CASE




```
library ieee;
use ieee.std_logic_1164.all;
entity DEMUX is
port  (d  : in bit;
       s  : in bit_vector (1 downto 0);
       y  : out bit_vector (3 downto 0);
end;
architecture behave of DEMUX is
begin
process
begin
       case s is
              when "00" =>    y <= '0' & '0' & '0' & d;
              when "01" =>    y <= '0' & '0' & d & '0';
              when "10" =>    y <= '0' & d & '0' & '0';
              when "11" =>    y <= d & '0' & '0' & '0';
       end case;
end process;
end behave;
```

המשך בעמוד 8

התניות בהליך טורי – בתוך PROCESS

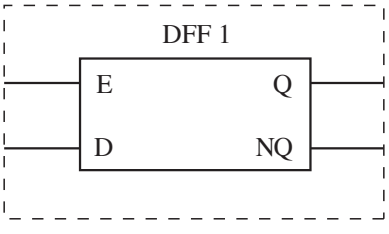
פקודת השהיה WAIT

wait;	עצירה לעד אשר שימושית לעצירת סביבת בדיקה
wait on	רשימת משתנים ;
wait until	תנאי לוגי ;
wait for	מספר יחידות זמן ;

דוגמה למימוש DFF

```
Library ieee;
Use ieee.std_logic_1164.all;
Entity DFF is
    Port ( e, d  : in std_logic;
           Q     : inout std_logic;
           Nq    : out std_logic );
End;
Architecture behave of DFF is
Begin
    Process ( e )
    Begin
        Wait until e'event and e = '1' ;
        Q <= d;
        Nq <= not d;
    End process;
End behave;
```

התכנית יצרה סמל SYMBOL עפ"י הישות (ENTITY) בתכנית.



המשך בעמוד 9

דוגמאות נוספות

דוגמה למימוש DFF עם מבואות ישירים

```
Process (enable)
Begin
        If      pre = '1'   then      q <= '1' ;
        Elsif   clr = '1'   then      q <= '0' ;
        Elsif   enable 'event and enable = '1'    then    q <= d;
End process;
```

ההליך בדוגמא יבצע את הפעולות המתוארות בטבלת האמת הבאה:

PRE	CLR	ENABLE	D	Q
1	0	Φ	Φ	1
0	1	Φ	Φ	0
0	0	אין עליה	Φ	נשמר מצב קודם
0	0	עליה	0	0
0	0	עליה	1	1

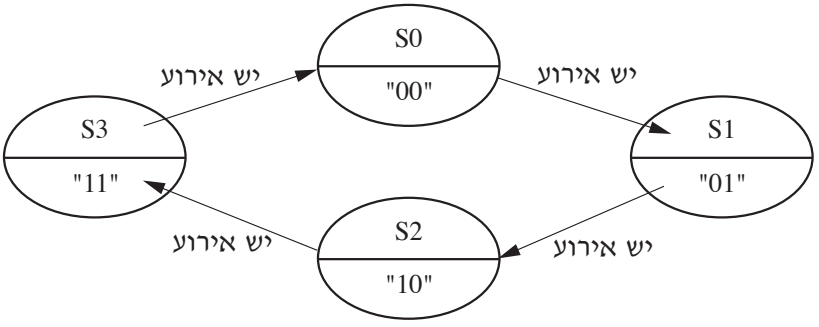
דוגמאות ליצירה והצבה במערכים

```
architecture EXAMPLE1 of ARRAY is

    type INTEGER_VECTOR is array (1 to 8) of integer;
    -- 1 --
    type MATRIX_A is array (1 to 3) of INTEGER_VECTOR;
    -- 2 --
    type MATRIX_B is array (1 to 4 , 1 to 8) of integer;

    signal MATRIX_3x8 : MATRIX_A;

    signal MATRIX_4x8 : MATRIX_B;

begin

    MATRIX_3x8 (3) (5)  <= 10; -- מערך בתוך מערך

    MATRIX_4x8 (4 , 5)  <= 17; -- מערך דו־ממדי

end EXAMPLE1;
```

המשך בעמוד 10

דוגמאות נוספות

דוגמה לכתיבת מונה במכונת מצבים




```
library ieee;
use ieee.std_logic_1164.all;
entity mealy_state_machine is
port  (clk          :   in bit;
       count_out  :   out bit_vector (1 downto 0));
end;
architecture behave of mealy_state_machine is
type state_type is (s0 , s1 , s2 , s3); -- names of states
signal state : state_type;

begin
counting : process (clk)
           begin
               if clk'event and clk = '1' then -- positive edge event
                   case state is
                       when s0 =>   state <= s1;
                       when s1 =>   state <= s2;
                       when s2 =>   state <= s3;
                       when s3 =>   state <= s0;
                   end case;
           end if;
end process counting;
with state select
           count_out <=      "00" when s0,
                             "01" when s1,
                             "10" when s2,
                             "11" when s3;
end behave;
```

המשך בעמוד 11

דוגמאות נוספות

דוגמה לתכנית המשתמשת בפונקציה המבצעת חיבור בין מספרים

```
entity ADDER_FUNC is
generic (max : integer := 15);
port (
            A , B  :  in integer range 0 to max;
            SUM :  out integer range 0 to MAX + MAX);
end;
architecture behave of ADDER_FUNC is
    function ADD (x , y : integer) return integer is
    variable s : integer;
    begin
        s := x + y;
        return s;
    end function ADD;
begin
sum <= ADD (a , b); -- הקריאה לפונקציה
end behave;
```

דוגמה לתכנית המשתמשת בפרוצדורה המבצעת חיבור בין מספרים

```
library ieee;
use ieee.std_logic_1164.all;
entity adder_proc is
port (      A , B : in integer;
            SUM : out integer);
end;

architecture behave of adder_proc is
    procedure ADD (signal x , y : in integer;
      signal s : out integer) is
    begin
      s <= x + y;
    end procedure ADD;

begin

ADD (a , b , sum);

end behave;
```

המשך בעמוד 12

דוגמאות נוספות	
הגדרת PACKAGE	**PACKAGE** __package_name **IS** -- Type Declaration -- Subtype Declaration (FUNCTIONS , PROCEDURES) -- Constant Declaration -- Signal Declaration -- Component Declaration **END** __package_name; **package body** package_name **is** declarations deferred constant declaration subprogram bodies (תיאור פעולות הפונקציות והפרוצדורות) **end package body** package_name;

Mode	קריאה	כתיבה
in	כן	לא
out	לא	כן
inout	כן	כן
buffer	כן	כן

בהצלחה!