

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 בהירות.
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 מֶרב 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 12 עמודים ו-16 עמודי נספחים.

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

השאלות

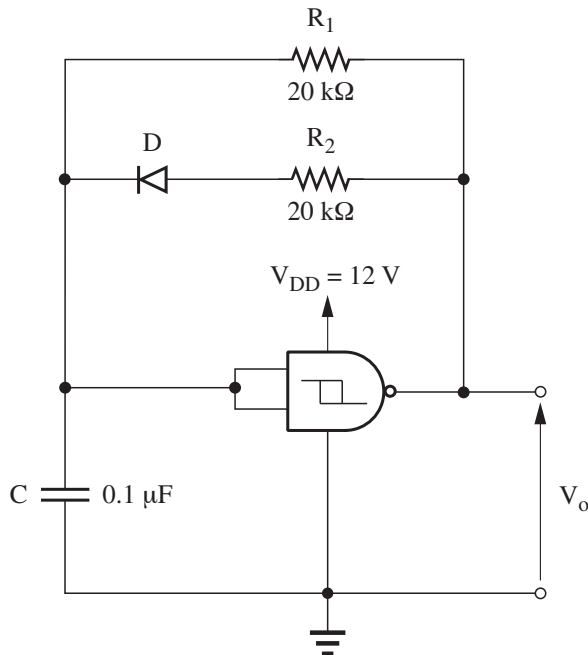
ענה על ארבע מבין השאלות 1–8. עליך לענות על שאלה אחת לפחות מכל פרק.

פרק ראשון: אלקטרוניקה ספרתית ב'

ענה על שאלה אחת לפחות מבין השאלות 1–4 (לכל שאלה – 25 נקודות).

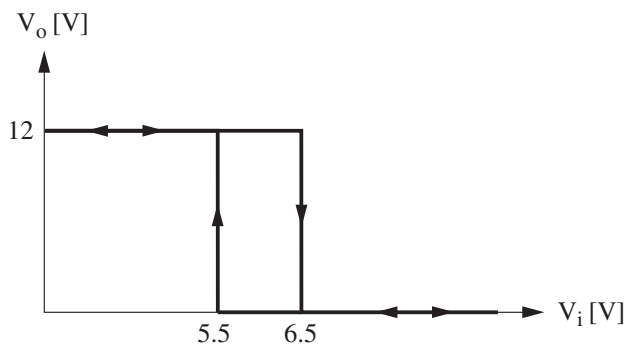
שאלה 1

באיור א' לשאלה 1 נתון מעגל חשמלי, הכולל שער NAND מסוג שמיט ממשפחת CMOS. הדיודה במעגל – אידיאלית.



איור א' לשאלה 1

באיור ב' לשאלה נתון אופיין המעבר של השער, $V_o = f(V_i)$.

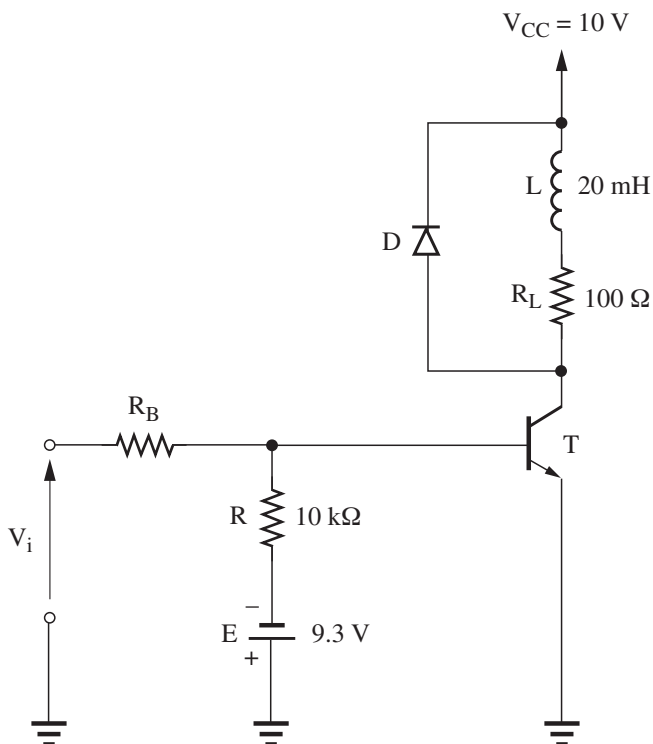


איור ב' לשאלה 1

- א. 1. סרטט, זו מתחת לזו בהתאמה, את צורת מתח-המוצא (V_o) ואת צורת המתח על הקבל (V_C) **במצב המתמיד**, כפונקציה של הזמן.
2. ציין בסרטוטך את הערך המרבי ואת הערך המזערי של מתח-המוצא (V_o) ושל המתח על הקבל (V_C).
- ב. חשב את תדר התנודות של מתח-המוצא (V_o).
- ג. חשב את מחזור הפעולה (Duty Cycle) של המעגל הזה.

שאלה 2

באיור א' לשאלה 2 נתון מעגל חשמלי המשמש למיתוג ממסר, המורכב מסליל L ונגד R_L .
הדיודה במעגל – אידיאלית.

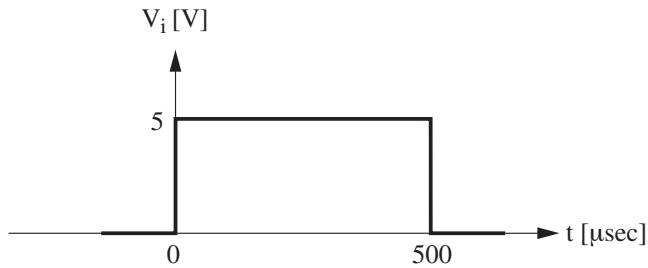


איור א' לשאלה 2

נתוני הממסר הם: זרם תפיסה – 30 mA , זרם התרה – 20 mA .

נתוני הטרנזיסטור T הם: $V_{BE} = 0.7 \text{ V}$, $V_{CE(\text{SAT})} = 0.2 \text{ V}$, $\beta = 100$.

למבוא המעגל מספקים את האות V_i , המתואר באיור ב' לשאלה.

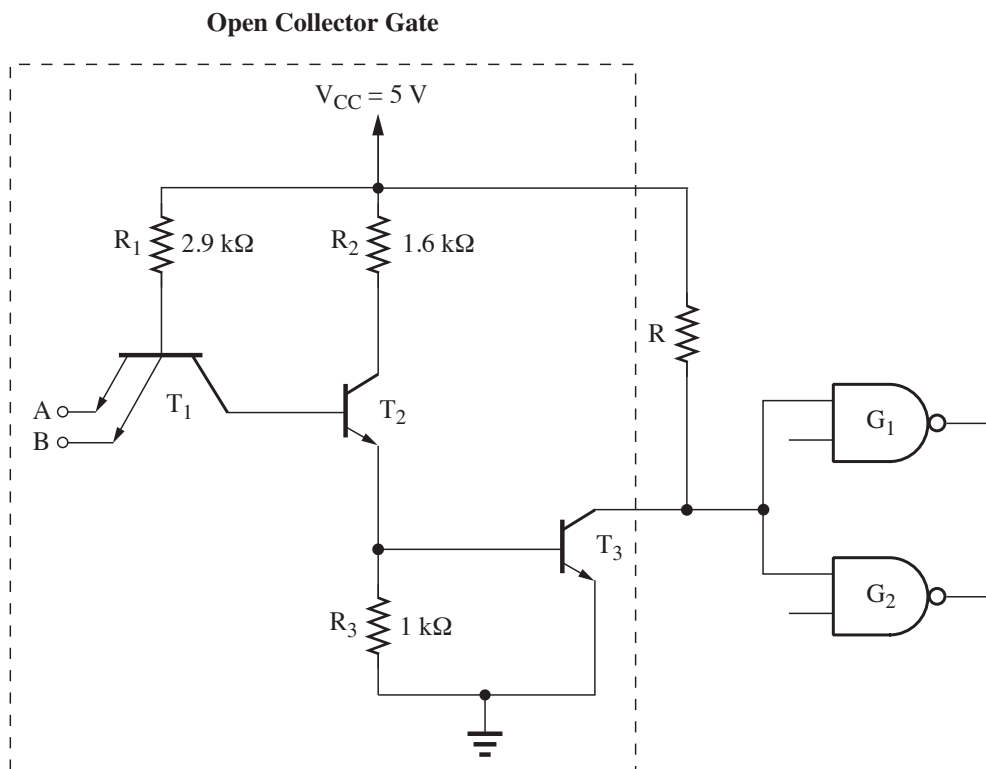


איור ב' לשאלה 2

- א. חשב את ערכו המרבי של הנגד R_B , שיבטיח שהטרנזיסטור T יימצא ברוויה עבור מתח־מבוא V_i קבוע של 5 V.
- ב. כעבור כמה זמן מרגע $t = 0$ הממסר ייתפס?
- ג. כעבור כמה זמן מרגע $t = 0$ הממסר ישתחרר?
- ד. 1. סרטט, זה מתחת לזה בהתאמה, את מתח־המבוא (V_i) ואת הזרם בסליל (I_L), כפונקציה של הזמן.
2. ציין בסרטוטך את הזמנים שחושבו בסעיפים ב' ו-ג'.

שאלה 3

באיור לשאלה 3 נתון המעגל החשמלי של שער לוגי מסוג קולט פתוח (Open Collector Gate) ממשפחת TTL. שער זה מחובר למבואות של שני שערים זהים, G_1 ו- G_2 , אף הם ממשפחת TTL.



איור לשאלה 3

נתוני הטרנזיסטורים $T_1 \div T_3$ הם:

$$V_{BC(ON)} = 0.7 \text{ V} ; V_{BE(ON)} = 0.7 \text{ V} ; V_{CE(SAT)} = 0.2 \text{ V} ; \beta_R = 0 ; \beta_{Fmin} = 10$$

כמו כן נתון: $V_A = V_B = 5 \text{ V}$

$$(I_{IL} \text{ זרם המבוא של שערי ה-TTL } G_1 \text{ ו-} G_2) = 1.34 \text{ mA}$$

- א. רשום שני מקרים שבהם משתמשים בשער לוגי מסוג קולט פתוח.
- ב. חשב את זרם הבסיס של כל אחד מן הטרנזיסטורים T_2 ו- T_3 , כאשר כל אחד מן המבואות A ו-B נמצא במצב '1'.
- ג. חשב את ההתנגדות המזערית של הנגד R, אם הטרנזיסטור T_3 נמצא על סף הרוויה והמתח המרבי בין הקולט ובין הפולט שלו הוא $V_{OL} = 0.4 \text{ V}$.

שאלה 4

- נדרש לתכנן מערכת עקיבה סינכרונית בעלת מבוא X ומוצא Z, ולממשה באמצעות רכיבי S.R.F.F. ושערים לוגיים.
- המערכת תפיק '1' במוצא Z בכל פעם שיתקבל רצף־הנתונים '1' '1' '1' '1' במבוא X (עבור כל '1' נוסף שיתקבל במבוא X – המערכת תוסיף להפיק '1' במוצא Z).
- רק כאשר יתקבל '0' במבוא X – המערכת לא תפיק '1' במוצא Z).
- בתהליך התכנון, בצע את השלבים שלהלן:

- א. סרטט דיאגרמת זרימה (דיאגרמת בועות), המתארת את הנדרש ממערכת העקיבה.
- ב. רשום את טבלת המצבים של המערכת הזו.
- ג. רשום את טבלת המעברים של המערכת.
- ד. רשום את טבלת העירור של המערכת.
- ה. פשט את פונקציות־המבוא ואת פונקציית־המוצא של המערכת.
- ו. ממש את המערכת שתכננת באמצעות S.R.F.F. ושערים לוגיים.

פרק שני: שפת תיאור חומרה VHDL

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

שאלה 5

לפניך התכנית TarA, הכתובה בשפת VHDL :

```
1  Library ieee;
2  Use ieee.std_logic_1164.all;
3  Entity TarA is
4  Port ( D : In Bit;
5         S : In Integer Range 0 to 3;
6         Y : Out Bit_Vector (3 DownTo 0) );
7  end;
8  Architecture Behave of TarA is
9  Begin
10 Y(3) <= D When S=3 Else '0';
11 Y(2) <= D When S=2 Else '0';
12 Y(1) <= D When S=1 Else '0';
13 Y(0) <= D When S=0 Else '0';
14 End Behave;
```

א. הסבר את הפעולה המתבצעת בשורה 13 בתכנית.

ב. הסבר את פעולת המערכת המתוארת בתכנית.

ג. כמה סיביות מוקצות למשתנה S בתכנית?

שאלה 6

לפניך התכנית TarB, הכתובה בשפת VHDL :

```

1  Library ieee;
2  Use ieee.std_logic_1164.all;
3  Entity TarB is
4  Port ( E, Rst, T : In Bit;
5         q          : Out Bit );
6  end;
7  Architecture Behave of TarB is
8  Signal sig : Bit;
9  Begin
10 Process (E, Rst)
11 Begin
12 If Rst = '1' then
13     sig <= '0';
14 Elsif E'event and E='1' then
15     if T = '0' then
16         sig <= sig;
17     Elsif T = '1' then
18         sig <= not sig;
19     End if;
20 End if;
21 End process;
22 q <= sig;
23 End Behave;
```

א. הסבר את הבדיקה המתבצעת בשורה 14 בתכנית.

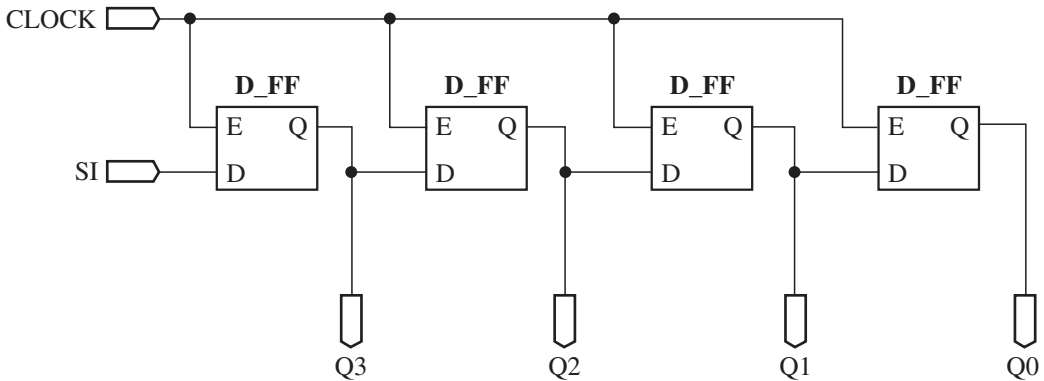
ב. הסבר את המשמעות של תוכן הסוגריים, המופיעות לאחר המילה Process בשורה 10 בתכנית.

ג. הסבר את פעולת המערכת המתוארת בתכנית. לווה את הסברך בטבלת פעולה של המערכת.

המשך בעמוד 10

שאלה 7

באיור לשאלה 7 נתונה מערכת, המבצעת פעולת אוגר הזזה ימינה. למערכת כניסת נתונים טורית (SI) ומוצא מקבילי בעל ארבע סיביות ($Q_3 \div Q_0$).



איור לשאלה 7

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

הערה: הסתמך על כך שתכניות ה-D_FF כתובות, מהודרות ומכילות את הישות הקיימת (ומודגשת) בתכנית ה-D_FF הבאה:

```
Library ieee;  
Use ieee.std_logic_1164.all;  
  
Entity D_FF is  
Port (E, D : In Bit ;  
       q      : Out Bit );  
end;  
  
Architecture Behave of D_FF is  
Begin  
Process (E)  
  
begin  
If E'event and E = '1' then  
  
    q <= D;  
  
End If;  
End Process;  
End Behave;
```

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

שאלה 8

לפניך התכנית TarC, הכתובה בשפת VHDL :

```

1  Library ieee;
2  Use ieee.std_logic_1164.all;
3  entity TarC is
4  generic ( bits : integer := 4);
5  port ( clk, si : in std_logic ;
6         q      : buffer std_logic_vector (bits-1 downto 0));
7  end;
8  Architecture behave of TarC is
9  Begin
10 Process (clk)
11 begin
12     if rising_edge(clk) then
13         q <= si & q(bits-1 downto 1);
14     end if;
15 End process;
16 End behave;

```

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

ב. העתק למחברתך את הטבלה הבאה, והשלם את ערכי המוצאים $q_3 \div q_0$ ואת הערך העשרוני במוצא, על-פי התכנית והנתונים בטבלה.

ערך עשרוני במוצא	q0	q1	q2	q3	si	clk
0	0	0	0	0	0	
8	0	0	0	1	1	
					0	
					1	
					1	

בהצלחה!

אין להעביר נוסחאון זה
מנבחן אחד למשנהו!

מקום לנחלקת נחן

נוסחאון באלקטרוניקה ספרתית ב' לכיתה י"ד

(4 עמודים)

משוואת הדפקים היסודית

מתח המוצא - $V(t)$ [V]

מתח סופי (עבור $t \rightarrow \infty$) - V_∞ [V]

מתח התחלתי - V_{0+} [V]

זמן - t [sec]

קבוע הזמן - τ [sec]

התנגדות שקולה ש"רואה" הרכיב - R_{eq} [Ω]

ההיגבי, מחושבת לפי תבנית

זרם מוצא - $I(t)$ [A]

זרם סופי (עבור $t \rightarrow \infty$) - I_∞ [A]

זרם התחלתי - I_{0+} [A]

$$V(t) = V_\infty - (V_\infty - V_{0+}) \cdot e^{-\frac{t}{\tau}}$$

$$\tau = R_{eq} \cdot C \quad ; \quad \tau = \frac{L}{R_{eq}}$$

$$I(t) = I_\infty - (I_\infty - I_{0+}) e^{-\frac{t}{\tau}}$$

תחום הרוויה בטרנזיסטור דו-נושאי

מתח רוויה בין קולט לפולט - V_{CEs} [V]

זרם בסיס - I_B [A]

זרם קולט - I_C [A]

הגבר זרם - β

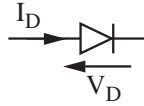
$$V_{CE} = V_{CEs} \\ \beta \cdot I_B > I_C$$

סף רוויה של התחום הפעיל בטרנזיסטור דו-נושאי

$$V_{CE} = V_{CEs} \\ \beta \cdot I_B = I_C$$

דיודת צומת

סימול:



א. דיודה אידיאלית:

ממתח קדמי - $V_D = 0$ (קָצֵר)

ממתח אחורני - $I_D = 0$ (נִתְק)

ב. קירוב באמצעות V_γ :

ממתח קדמי - $V_D = V_\gamma$

ממתח אחורני - $I_D = 0$ ($V_D < V_\gamma$)

ג. קירוב באמצעות V_γ ו- R_f :

ממתח קדמי - $V_D = I_D \cdot R_f + V_\gamma$ ($V_D > V_\gamma$)

ממתח אחורני - $I_D = 0$ ($V_D < V_\gamma$)

טעינה לינארית

טעינת קבל בזרם קבוע:

מתח הקבל	-	V_C	[V]
זרם הקבל	-	I_C	[A]
קיבול	-	C	[F]
זמן	-	t	[sec]

$$V_C = \frac{I_C}{C} \cdot t + V_C(0)$$

$$\Delta V_C = \frac{I_C}{C} \cdot \Delta t$$

זרם הסליל	-	I_L	[A]
מתח הסליל	-	V_L	[V]
השראות	-	L	[H]

טעינת סליל במתח קבוע:

$$I_L = \frac{V_L}{L} \cdot t + I_L(0)$$

$$\Delta I_L = \frac{V_L}{L} \cdot \Delta t$$

טבלת מצבים של JKFF

CLK	J	K	Q
1, 0	$\emptyset$	$\emptyset$	N.C
$\downarrow$	0	0	N.C
$\downarrow$	0	1	0
$\downarrow$	1	0	1
$\downarrow$	1	1	$\bar{Q}_{n-1}$ (שינוי מצב)

טבלת מצבים של SRFF (סינכרוני)

CLK	S	R	Q
0, 1	$\emptyset$	$\emptyset$	N.C
$\downarrow$	0	0	N.C
$\downarrow$	0	1	0
$\downarrow$	1	0	1
$\downarrow$	1	1	מצב אסור

טבלת מצבים של TFF

CLK	T	Q_n
0, 1	$\emptyset$	N.C
	0	N.C
	1	$\bar{Q}_{n-1}$ (שינוי מצב)

טבלת מצבים של DFF

CLK	D	Q
1, 0	$\emptyset$	N.C
	0	0
	1	1

טבלת עירור

PS			NS		JKFF		SRFF		TFF	DFF
q			→	Q	J	K	S	R	T	D
0	→	0		0	0	ϕ	0	ϕ	0	0
0	→	1		1	1	ϕ	1	0	1	1
1	→	0		0	ϕ	1	0	1	1	0
1	→	1		1	ϕ	0	ϕ	0	0	1

בהצלחה!

אין להעביר את הנוסחאון
לנבחן אחר

מקום למציאת נבחן

נוסחאון בשפת תיאור חומרה VHDL לכיתה י"ד

(12 עמודים)

בלוקים עיקריים בשפה	
<pre>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;</pre>	מבנה כללי של ישות תכנית

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

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 FOR בלולאת
__generate_label: IF __expression GENERATE __statement; __statement; END GENERATE __generate_label;	שרשור מותנה IF GENERATE

לולאות LOOPS	
__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

התניות בגוף התכנית – מחוץ ל-PROCESS	
__signal <= __expression WHEN __boolean_expression ELSE __expression WHEN __boolean_expression ELSE __expression _____ ; דוגמא 1 : מימוש שער AND באמצעות פקודת "WHEN" פשוטה. Y <= '1' when (a = '1') and (b = '1') else '0' ; דוגמא 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' ;	התנית 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; _____ דוגמא: למימוש 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 ;	התנית With ... Select

התניות בהליך טורי – בתוך PROCESS	
IF __boolean_expression THEN __signal <= __ expression; ELSIF __boolean_expression THEN __signal <= __expression; ELSE __signal <= __expression; END IF;	התניית If .. Then .. Else
CASE __expression IS WHEN __constant_value => __statement; __statement; WHEN __constant_value => __statement; __statement; WHEN OTHERS => __statement; __statement; END CASE;	התניית CASE
דוגמא למימוש מפענח בהתניית CASE : <pre> 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; </pre>	

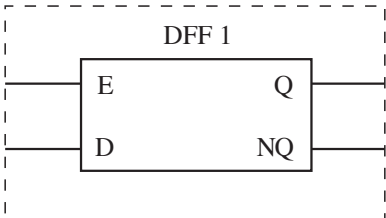
התניות בהליך טורי – בתוך PROCESS					
התוכנית תבצע את טבלת האמת הבאה:					
S : BIT_VECTOR (1 DOWNT0 0)		Y : BIT_VECTOR (3 DOWNT0 0)			
S1	S0	Y3	Y2	Y1	Y0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

```
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;
```

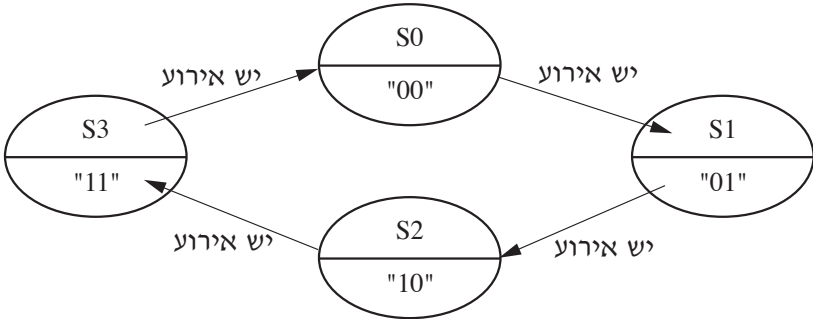
FIVE_K

```
graph LR
    D[D] --> MUX[FIVE_K]
    S[S[1...0]] --> MUX
    MUX --> Y[Y[3...0]]
```

דוגמא לתכנית שלמה המבצעת פעולת מפלג, תוך שימוש בהתניית CASE

התניות בהליך טורי – בתוך PROCESS		
wait; עצירה לעד אשר שימושית לעצירת סביבת בדיקה wait on רשימת משתנים ; wait until תנאי לוגי ; wait for מספר יחידות זמן ;		פקודת השהיה WAIT
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;		דוגמא למימוש DFF
התכנית יצרה סמל SYMBOL עפ"י הישות (ENTITY) בתוכנית.		
		

דוגמאות נוספות																																		
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: <table><tr><th>PRE</th><th>CLR</th><th>ENABLE</th><th>D</th><th>Q</th></tr><tr><td>1</td><td>0</td><td>Φ</td><td>Φ</td><td>1</td></tr><tr><td>0</td><td>1</td><td>Φ</td><td>Φ</td><td>0</td></tr><tr><td>0</td><td>0</td><td>אין עליה</td><td>Φ</td><td>נשמר מצב קודם</td></tr><tr><td>0</td><td>0</td><td>עליה</td><td>0</td><td>0</td></tr><tr><td>0</td><td>0</td><td>עליה</td><td>1</td><td>1</td></tr></table>				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	דוגמא למימוש DFF עם מבוואת ישירים
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																														
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;				דוגמאות ליצירה והצבה במערכים																														

דוגמאות נוספות	
 <pre> 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; </pre>	דוגמא לכתיבת מונה במכונה מצבים

דוגמאות נוספות	
<pre> 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; </pre>	דוגמא לתכנית המשתמשת בפונקציה המבצעת חיבור בין מספרים
<pre> 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; </pre>	דוגמא לתכנית המשתמשת בפרוצדורה המבצעת חיבור בין מספרים

דוגמאות נוספות	
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;	הגדרת PACKAGE

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

בהצלחה!