

סוג הבחינה: גמר לבתי-ספר לטכנאים ולהנדסאים

מועד הבחינה: אביב תשע"ט, 2019

סמל השאלון: 711003

נספחים: א. נוסחאון באלקטרוניקה

ספרתית ב' לכיתה י"ד

ב. נוסחאון בשפת תיאור חומרה

VHDL לכיתה י"ד

ג. מילון מונחים

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 שלהן.
 6. אם לדעתך חסרים נתונים הדרושים לפתרון שאלה, אתה רשאי להוסיף אותם, בתנאי שתנמק מדוע הוספת אותם.
 7. בכתבת פתרונות חישוביים, קבלת מֶרֶב 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.
 8. לנוחותך, לשאלון זה מצורף מילון מונחים בשפות עברית, ערבית, אנגלית ורוסית. תוכל להיעזר בו בעת הצורך.

בשאלון זה 10 עמודים ו-17 עמודי נספחים.

ההנחיות בשאלון זה מנוסחות בלשון זכר,

אך מכוונות הן לנבחנות והן לנבחנים.

השאלות

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

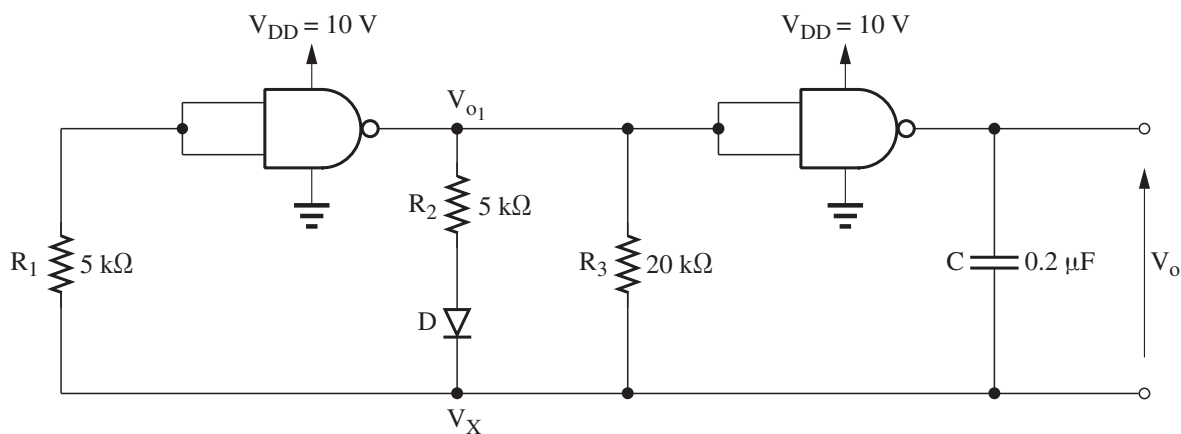
פרק ראשון: אלקטרוניקה ספרתית

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

שאלה 1

באיור לשאלה 1 נתון המעגל החשמלי של רב־רטט חופשי, הכולל שערים ממשפחת CMOS. הדיודה שבמעגל – אידיאלית.

נתוני שערי ה-CMOS: $V_{OH} = 10\text{ V}$; $V_{OL} = 0\text{ V}$; $V_{TH} = \frac{1}{2}V_{DD} = 5\text{ V}$.



איור לשאלה 1

1. א. (9 נק') סרטט, זו מתחת לזו בהתאמה, את צורות המתחים V_{O1} , V_O ו- V_X (יחסית לאדמה) במצב המתמיד, כפונקצייה של הזמן.

2. ציין בסרטוטך את הערך המרבי ואת הערך המזערי של כל אחד מן המתחים הללו.

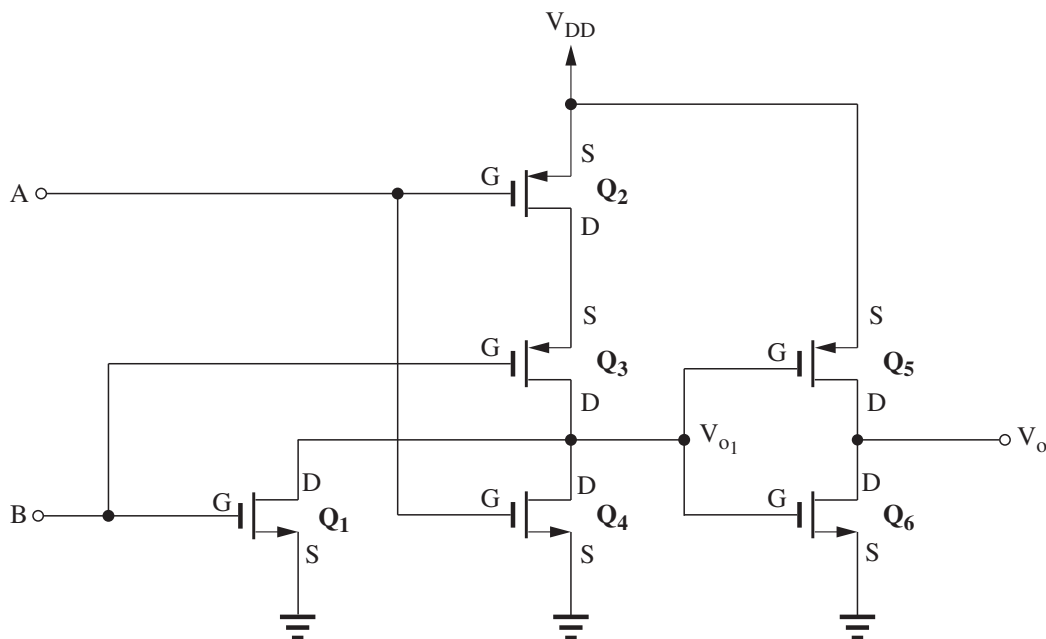
ב. (6 נק') חשב את התדר של מתח־המוצא V_O .

ג. (6 נק') חשב את מחזור הפעולה (Duty Cycle) של מתח־המוצא V_O .

ד. (4 נק') מהו תפקידה של הדיודה D במעגל הזה?

שאלה 2

באיור לשאלה 2 מתואר המעגל החשמלי של שער לוגי ממשפחת CMOS.



איור לשאלה 2

- 12 נק') א. העתק את הטבלה שלהלן למחברתך. רשום בה באיזה מצב (ON / OFF) נמצא כל אחד מן הטרנזיסטורים $Q_1 \div Q_6$, ואת הערך הלוגי של כל אחד מהמתחים V_{O1} ו- V_O . הנח כי המעגל פועל בלוגיקה חיובית, כלומר: $'0' = 0\text{ V}$, $'1' = V_{DD}$.

A	B	Q_1	Q_2	Q_3	Q_4	Q_5	Q_6	V_{O1}	V_O
0	0								
0	1								
1	0								
1	1								

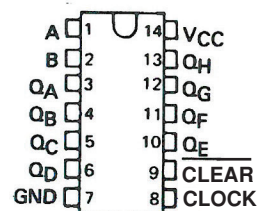
- 8 נק') ב. היעזר בטבלה שמילאת בסעיף א', ורשום את הפונקצייה הלוגית שמבצע המעגל החשמלי שבאיור.
5 נק') ג. ממש את הפונקצייה הלוגית $F(A,B,C) = \overline{A + B + C}$ באמצעות טרנזיסטורים ממשפחת CMOS.

שאלה 3

הרכיב SN74164 הוא אוגר הזזה, המשמש להמרת מידע בינארי טורי למידע בינארי מקבילי. להלן תצורת ההדקים של הרכיב וטבלת המצבים שלו, כפי שהם נתונים בדפיהנתונים שלו.

FUNCTION TABLE						
INPUTS				OUTPUTS		
CLEAR	CLOCK	A	B	Q_A	$Q_B \dots Q_H$	
L	X	X	X	L	L	L
H	L	X	X	Q_{A0}	Q_{B0}	Q_{H0}
H	↑	H	H	H	Q_{An}	Q_{Gn}
H	↑	L	X	L	Q_{An}	Q_{Gn}
H	↑	X	L	L	Q_{An}	Q_{Gn}

PIN CONFIGURATION
(TOP VIEW)



H = high level (steady state) , L = low level (steady state)

X = irrelevant (any input, including transitions)

↑ = transition from low to high level

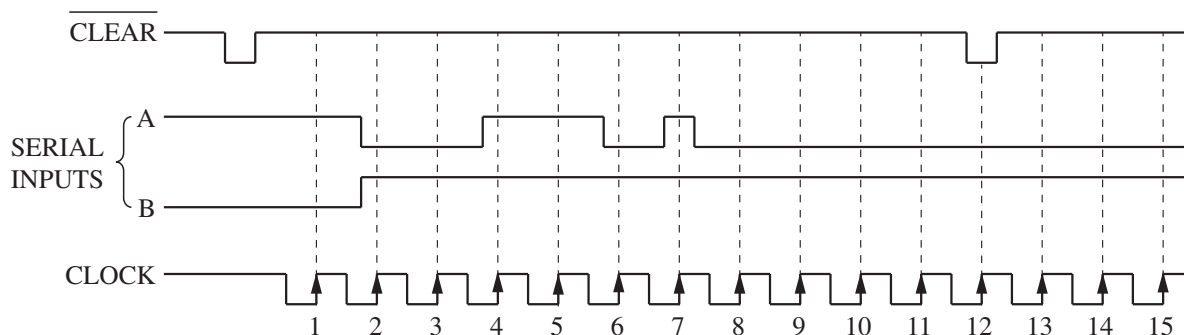
Q_{A0} , Q_{B0} , Q_{H0} = the level of Q_A , Q_B , or Q_H , respectively, before the indicated steady-state input conditions were established.

Q_{An} , Q_{Gn} = the level of Q_A or Q_G before the most-recent ↑ transition of the clock ; indicates a one-bit shift.

א. (8 נק') הסבר את תפקידו של כל אחד מן ההדקים 8 ו-9.

ב. (5 נק') רשום את הקשר הבוליאני בין הערך המתקבל בהדק Q_A לאחר הזזת המידע ובין ערכי האותות בהדקים A ו-B לפני ההזזה.

ג. (12 נק') באיור לשאלה 3 נתונים אותות המידע המתקבלים בהדקים A ו-B של הרכיב והאותות המתקבלים בהדקים 8 ו-9.



איור לשאלה 3

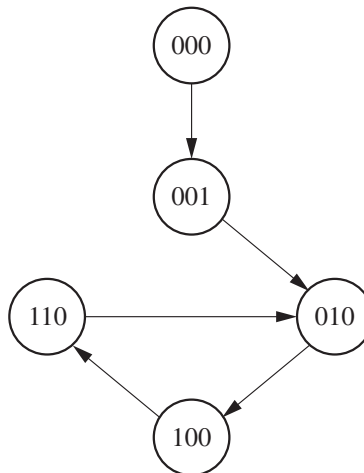
העתק את הטבלה שלהלן למחברתך. עקוב באיור אחר הזזת המידע על-ידי הרכיב, וציין בטבלה את מצבו של כל אחד מן ההדקים $Q_A \div Q_H$ בהתאם לעיתוי הנתון בה.

העיתוי	Q_A	Q_B	Q_C	Q_D	Q_E	Q_F	Q_G	Q_H
מיד לאחר דופק שעון מס' 4								
מיד לאחר דופק שעון מס' 8								
מיד לאחר דופק שעון מס' 10								
מיד לאחר דופק שעון מס' 12								

שאלה 4

תכנן מערכת עקיבה סינכרונית שתשמש כמונה (לא מחזורי), ותמומש באמצעות דלגלגי JK-FF ושערים לוגיים (על-פי הצורך). הנח שהמצב ההתחלתי של המוצאים הוא: $Q_2 = Q_1 = Q_0 = '0'$ (Q_2 – MSB, Q_0 – LSB).

דיאגרמת המצבים של המערכת נתונה באיור לשאלה 4.



איור לשאלה 4

בתהליך התכנון בצע את השלבים שלהלן:

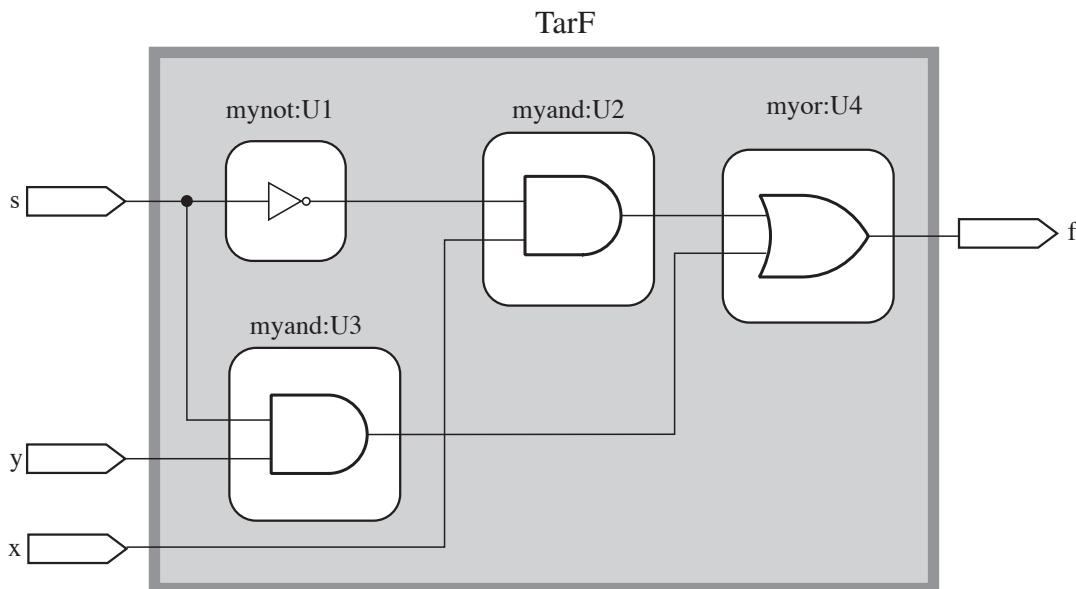
- 8 נק' א. רשום את טבלת המצבים של המערכת.
- 5 נק' ב. רשום את טבלת המעברים ואת טבלת העירור של המערכת.
- 6 נק' ג. פשט את פונקציות המבוא של הדלגלגים.
- 6 נק' ד. ממש את המערכת באמצעות דלגלגי JK-FF ושערים לוגיים (על-פי הצורך).

פרק שני: שפת תיאור חומרה VHDL

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

שאלה 5

באיור לשאלה 5 נתונה מערכת ספרתית בשם TarF.



איור לשאלה 5

כמור־כן נתון המימוש בשפת VHDL של כל אחת מהיחידות mynot, myor, myand.

```
ENTITY myand IS
PORT (a, b: IN BIT;
      y : OUT BIT);
END myand;

ARCHITECTURE behave OF myand IS
BEGIN
    y <= a AND b;
END behave;
```

```
ENTITY myor IS
    PORT (a, b: IN BIT;
          y : OUT BIT);
END myor;

ARCHITECTURE behave OF myor IS
BEGIN
    y <= a OR b;
END behave;
```

```
ENTITY mynot IS
    PORT (x: IN BIT;
          y : OUT BIT);
END mynot;

ARCHITECTURE behave OF mynot IS
BEGIN
    y <= NOT x;
END behave;
```

- 15 נק')** א. ממש את המערכת שבאיור לשאלה 5 בשפת VHDL על-פי עקרונות התכנון ההיררכי, תוך שימוש במבניות: myand , myor , mynot .
- 10 נק')** ב. העתק למחרתך את האיור לשאלה 5, וסמן עליו את האותות (Signals) שהגדרת בסעיף א'.

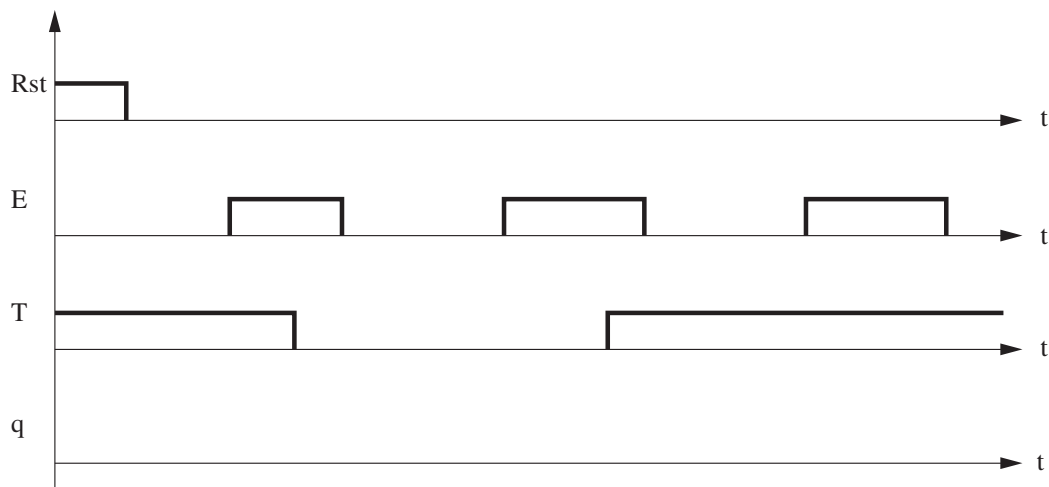
שאלה 6

לפניך התוכנית TarB , הכתובה בשפת VHDL :

```
1  Entity TarB is
2  Port ( E, Rst, T : In Bit;
3         q          : Out Bit );
4  End;
5  Architecture Behave of TarB is
6  Begin
7  Process (E, Rst)
8  Variable tmp  : Bit := 0;
9  Begin
10 If Rst = '1' then
11     tmp := '0';
12 Elsif E'event and E='1' then
13     If T = '0' then
14         tmp := tmp;
15     Else
16         tmp := not tmp;
17     End If;
18 End If;
19 q <= tmp;
20 End Process;
21 End Behave;
```

8 נק') א. הסבר את הבדיקה המתבצעת בשורה 12 בתוכנית.

10 נק') ב. באיור לשאלה 6 מתוארים, זה מתחת לזה בהתאמה, האותות E , Rst ו-T כפונקצייה של הזמן. העתק את האיור למחברתך, והשלם בו את אות המוצא q כפונקציה של הזמן.

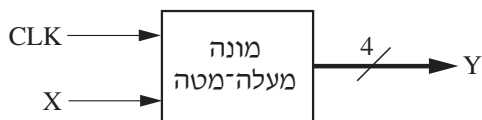


איור לשאלה 6

7 (נק') ג. מעבירים את הפקודות שבשורות 10÷11 לאחר שורה 12. כיצד ישפיע הצעד הזה על פעולת המערכת?

שאלה 7

באיור לשאלה 7 מתואר מונה מעלה-מטה.



איור לשאלה 7

כתוב תוכנית בשפת VHDL, המבצעת פעולת מונה מעלה-מטה. המונה יספור באופן מחזורי בכל עליית שעון, בהתאם לערך הלוגי של המבוא X. ערכו של המבוא X נבדק בכל עליית שעון:

אם ערכו הוא $X = '1'$, המונה יספור מ-0 עד 9 באופן מחזורי.

אם ערכו הוא $X = '0'$, המונה יספור מ-9 עד 0 באופן מחזורי.

הערה:

השימוש במכונת-מצבים בתוכנית אינו חובה.

שאלה 8

לפניך התוכנית 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. (5 נק') הסבר את הפעולה המתבצעת בשורה 10 בתוכנית.

ב. (5 נק') סרטט סמל למערכת TarA. ציין בסרטוט את שמות המבואות והמוצא, וציין את מספר הסיביות בכל אחד מהם.

ג. (10 נק') העתק למחברתך את טבלת המצבים שלהלן והשלם בה את החסר, על-פי פעולת המערכת המתוארת בתוכנית.

S	Y(3)	Y(2)	Y(1)	Y(0)
0				
1	'0'	'0'	D	'0'
2				
3				

ד. (5 נק') ממש את פעולת המערכת בצורה שונה מזו המתוארת בתוכנית TarA.

בהצלחה!

זכות היוצרים שמורה למדינת ישראל.

אין להעתיק או לפרסם אלא ברשות משרד החינוך.

אין להעביר נוסחאון זה
לנבחן אחר

נוסחאון באלקטרוניקה ספרתית ב' לכיתה י"ד

(4 עמודים)

משוואות הדפקים היסודית

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

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

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

$$t = \tau \cdot \ln \frac{V_{\infty} - V_{0+}}{V_{\infty} - V(t)} = \tau \cdot \ln \frac{I_{\infty} - I_{0+}}{I_{\infty} - I(t)}$$

תחום הרוויה בטרנזיסטור דו-נושאי

$$V_{CE} = V_{CE_S}$$

$$\beta \cdot I_B > I_C$$

סף רוויה של התחום הפעיל בטרנזיסטור דו-נושאי

$$V_{CE} = V_{CE_S}$$

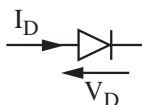
$$\beta \cdot I_B = I_C$$

מתח מוצא	-	$V(t) [V]$
מתח סופי (עבור $t \rightarrow \infty$)	-	$V_{\infty} [V]$
מתח התחלתי	-	$V_{0+} [V]$
זמן	-	$t [sec]$
קבוע הזמן	-	$\tau [sec]$
זרם מוצא	-	$I(t) [A]$
זרם סופי (עבור $t \rightarrow \infty$)	-	$I_{\infty} [A]$
זרם התחלתי	-	$I_{0+} [A]$
התנגדות שקולה ש"רואה" הרכיב ההיגבי, מחושבת לפי תבנית	-	$R_{eq} [\Omega]$

מתח רוויה בין קולט לפולט	-	$V_{CE_S} [V]$
זרם בסיס	-	$I_B [A]$
זרם קולט	-	$I_C [A]$
הגבר זרם	-	β

דיודת צומת

סימול:



א. דיודה אידיאלית:

ממתח קדמי - $V_D = 0$ (קָצֶר)

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
$\downarrow$	0	N.C
$\downarrow$	1	$\bar{Q}_{n-1}$ (שינוי מצב)

טבלת מצבים של DFF

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

טבלת עירור

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

בהצלחה!

אין להעביר את הנוסחאון
לנבחן אחר

נוסחאון בשפת תיאור חומרה VHDL לכיתה י"ד

(12 עמודים)

בלוקים עיקריים בשפה	
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;	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 _____ ; 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 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

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
דוגמה 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;

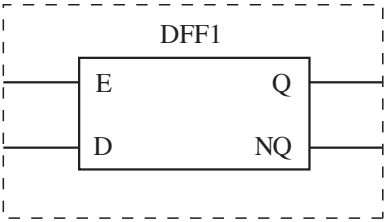
entity DEMUX is
port (d : in bit;
      s : in bit_vector (1 downto 0);
      y : out bit_vector (3 downto 0);
end;

architecture behave of DEMUX is
begin
process (s)
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;

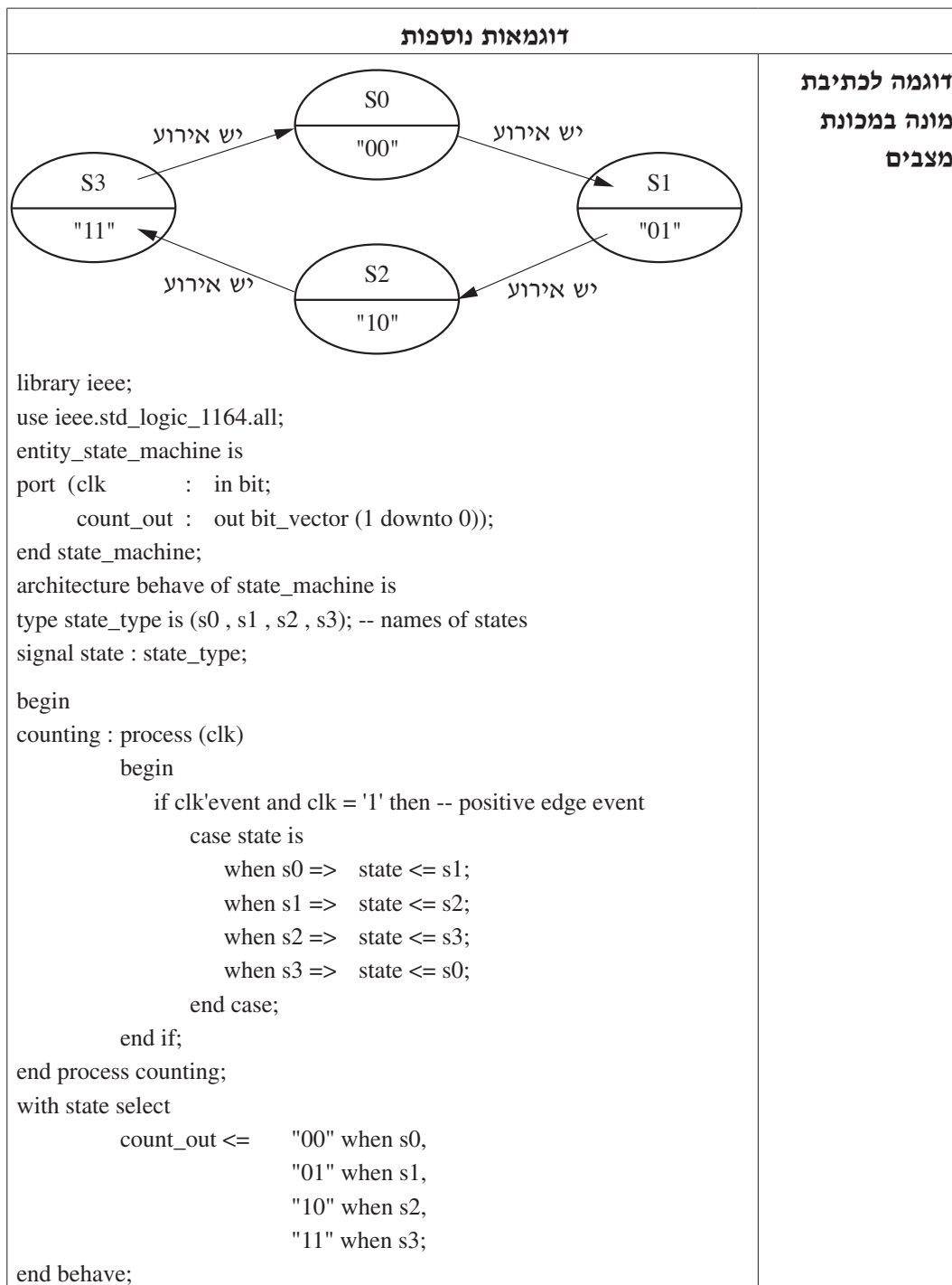
```

FIVE_K

דוגמה לתכנית שלמה המבצעת פעולת מפלג, תוך שימוש בהתניות 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
<pre> Library ieee; Use ieee.std_logic_1164.all; Entity DFF1 is Port (e, d : in std_logic; Q : inout std_logic; Nq : out std_logic); End DFF1; Architecture behave of DFF1 is Begin Process Begin Wait until e'event and e = '1' ; Q <= d; Nq <= not d; End process; End behave; </pre> התכנית יצרה סמל SYMBOL עפ"י הישות (ENTITY) בתכנית. 		דוגמה למימוש DFF

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>F</td><td>F</td><td>1</td></tr><tr><td>0</td><td>1</td><td>F</td><td>F</td><td>0</td></tr><tr><td>0</td><td>0</td><td>אין עליה</td><td>F</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	F	F	1	0	1	F	F	0	0	0	אין עליה	F	נשמר מצב קודם	0	0	עליה	0	0	0	0	עליה	1	1	דוגמה למימוש DFF עם מבוואת ישירים
PRE	CLR	ENABLE	D	Q																														
1	0	F	F	1																														
0	1	F	F	0																														
0	0	אין עליה	F	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> 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>	דוגמה 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>	דוגמה 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

נספח: מילון מונחים

לשאלון 711003, אביב תשע"ט

תרגום המונח			המונח
אנגלית	רוסית	ערבית	
states diagram	Диаграмма состояний	مُخَطَّطُ الحالات	דיאגרמת מצבים
flip-flop	Триггер	قَلَاب	דלגלג
transitions table	Таблица переходных состояний	جدول التنقّلات	טבלת מעברים
states table	Таблица состояний	جدول الحالات	טבלת מצבים
excitaion table	Таблица переходов	جدول الإثارة	טבלת עירור
components	составляющие	عَنَاصِر / مكوّنات	מבניות
counter	Счётчик	عدّاد	מונה
digital system	Синхронный автомат	نظام رقمي	מערכת ספרתית
synchronous tracking system	Синхронный автомат последовательных состояний	نظام تتبّع مُتزامن	מערכת עקיבה סינכרונית
bit	бит	رقم ثنائي (بت)	סיבית
symbol	символ	رمز	סמל
multi-vibrator	Мультивибратор	مُتعدّد الاهتزاز	רב־רטט
hierarchical design	Иерархическое программирование	تصميم مرتبي	תכנון היררכי