

סוג הבחינה: גמר לבתי-ספר לטכנאים ולהנדסאים

מועד הבחינה: אביב תשפ"ב, 2022

סמל השאלון: 711003

נספחים: א. נספח לשאלה 3

ב. מילון מונחים

ג. נוסחאון במעגלים

ד. אלקטרוניים לכיתה י"ד

נוסחאון בשפת תיאור חומרה

VHDL לכיתה י"ד

אלקטרוניקה ומחשבים ה'

מגמת הנדסת אלקטרוניקה ומחשבים

(כיתה י"ד)

הוראות לנבחנים

- א. משך הבחינה: ארבע שעות.
- ב. מבנה השאלון ומפתח ההערכה: בשאלון זה שמונה שאלות.
יש להשיב על ארבע שאלות בלבד.
לכל שאלה – 25 נקודות. סך-הכול – 100 נקודות.
- ג. חומר עזר מותר בשימוש: מחשבון, וכן כל חומר הכתוב (בכתב יד או בהדפסה) על שלושה דפי A4 הכתובים משני הצדדים.
- ד. הוראות מיוחדות:
 1. ענו על מספר השאלות הנדרש בשאלון. המעריך יקרא ויעריך את מספר התשובות הנדרש בלבד, לפי סדר כתיבתן במחברת, ולא יתייחס לתשובות נוספות.
 2. התחילו כל תשובה לשאלה חדשה בעמוד חדש.
 3. כתבו את כל התשובות אך ורק בעט.
 4. הקפידו לנסח את התשובות כהלכה ולסרטט את התרשימים בבהירות.
 5. כתבו את התשובות בכתב-יד ברור, כדי לאפשר הערכה נאותה שלהן.
 6. אם לדעתכם חסרים נתונים הדרושים לפתרון שאלה, אתם רשאים להוסיף אותם, בתנאי שתנמקו מדוע הוספתם אותם.
 7. בכתובת פתרונות חישוביים, קבלת מֶרֶב הנקודות מותנית בהשלמת כל המהלכים שלהלן, בסדר שבו הם כתובים:
 - * כתיבת הנוסחה המתאימה.
 - * הצבה של כל הערכים ביחידות המתאימות.
 - * חישוב (אפשר באמצעות מחשבון).
 - * כתיבת התוצאה המתקבלת, יחד עם יחידות המידה המתאימות.
 - * ליווי הפתרון החישובי בהסבר קצר.
 8. בנספח ב' לשאלון זה מובא מילון מונחים בשפות עברית, ערבית, אנגלית ורוסית. תוכלו להיעזר בו בעת הצורך.
 9. צרפו את שלושת דפי ה-A4 למחברת הבחינה.

בשאלון זה 12 עמודים ו-31 עמודי נספחים.

בהצלחה!

המשך מעבר לדף

השאלות

ענו על ארבע מבין השאלות 1–8 (לכל שאלה – 25 נקודות).

פרק ראשון: אלקטרוניקה תקבילית

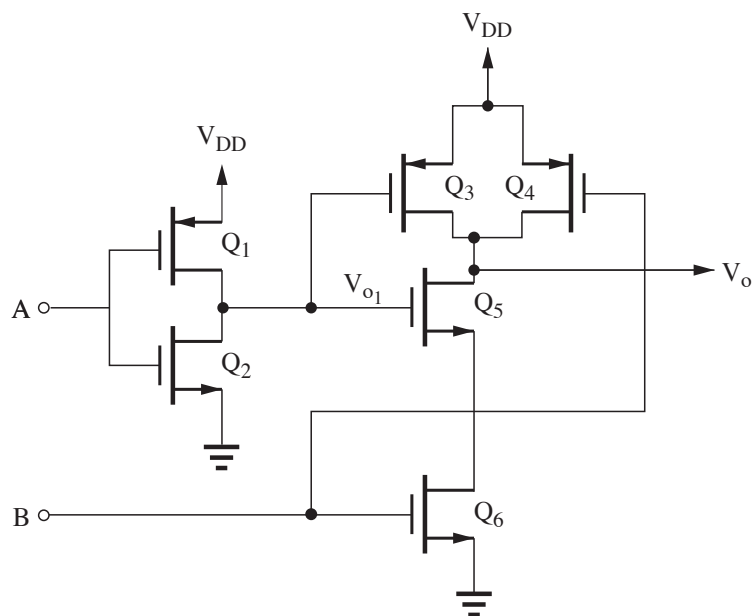
שאלה 1

תכננו מערכת עקיבה סינכרונית שתשמש כמונה מחזורי ותמומש באמצעות דלגלגי JK-FF ושערים לוגיים (על-פי הצורך). המונה יספור באופן מחזורי, בבסיס בינארי, באופן הזה (קראו משמאל לימין): 0, 1, 3, 5, 7. הניחו שהמצב ההתחלתי של המוצאים הוא $Q_0 = '0'$, $Q_1 = '0'$, $Q_2 = '0'$, כאשר המוצא Q_2 מייצג את סיבית ה-MSB.

- (5 נק') א. סרטטו את תרשים המצבים של המערכת.
- (5 נק') ב. כתבו את טבלת המצבים של המערכת.
- (5 נק') ג. כתבו את טבלת המעברים ואת טבלת העירור של המערכת.
- (5 נק') ד. פשטו את פונקציות המבוא של הדלגלגים.
- (5 נק') ה. ממשו את המערכת באמצעות דלגלגי JK-FF ושערים לוגיים, על-פי הצורך.

שאלה 2

באיור לשאלה 2 מתואר מעגל חשמלי של שער לוגי ממשפחת CMOS.



איור לשאלה 2

- א. (14 נק') העתיקו את הטבלה שלהלן למחברת. כתבו 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 . הניחו כי המעגל פועל בלוגיקה חיובית, כלומר: $V_{DD} = '1'$ ו- $0V = '0'$.

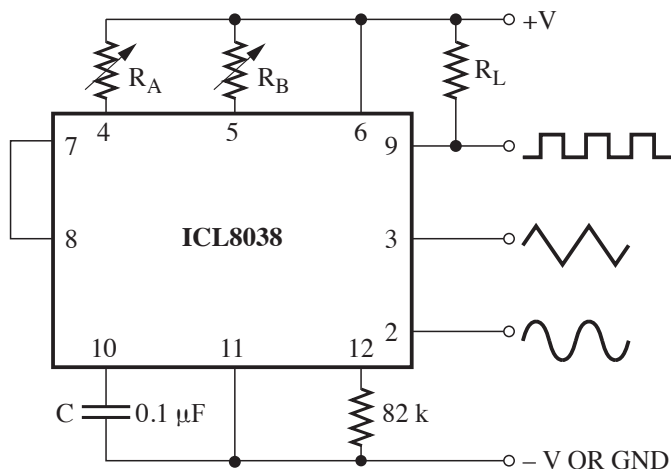
A	B	Q_1	Q_2	Q_3	Q_4	Q_5	Q_6	V_{o1}	V_o
0	0								
0	1								
1	0								
1	1								

- ב. (6 נק') היעזרו בטבלה שמילאתם בסעיף א', וכתבו את הפונקצייה הלוגית שמבצע המעגל החשמלי שבאיור.

- ג. (5 נק') ממשו את הפונקצייה הלוגית $F(A,B) = \overline{A} \cdot \overline{B}$ באמצעות טרנזיסטורים ממשפחת CMOS.

שאלה 3

באיור לשאלה 3 נתון סרטוט חשמלי של הרכיב ICL8038, ובנספח א' לשאלה 3 נתון דף נתונים של הרכיב.

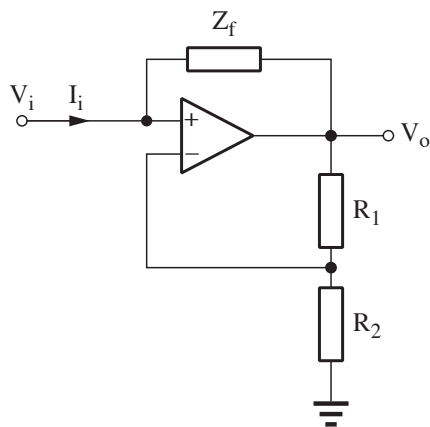


איור לשאלה 3

- א. (10 נק') חשבו את ערכי הנגדים R_A ו- R_B הנדרשים כדי לקבל בהדק 2 אות סינוסואידלי בתדירות של 1 kHz.
- ב. (15 נק') חשבו את ערכי הנגדים R_A ו- R_B כדי לקבל בהדק 9 אות ריבועי בתדירות של 1 kHz ומחזור פעולה (Duty Cycle) של 80%.

שאלה 4

באיור לשאלה 4 נתון מעגל חשמלי של ממיר עכבות.



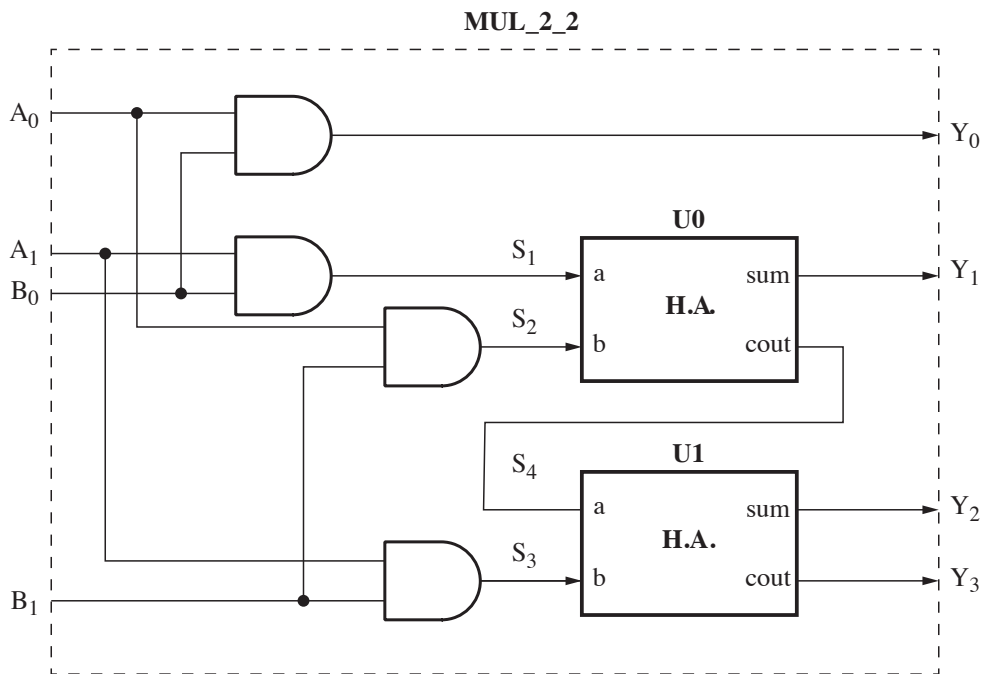
איור לשאלה 4

- א. (7 נק') כתבו ביטוי המתאר את זרם המבוא I_i כפונקצייה של הנגדים R_1 ו- R_2 , של העכבה Z_f ושל מתח המקור V_i .
- ב. (7 נק') כתבו ביטוי המתאר את מתח המוצא V_o כפונקצייה של רכיבי המעגל ושל מתח המקור V_i .
- ג. (6 נק') כתבו ביטוי המתאר את עכבת המבוא: $Z_i = \frac{V_i}{I_i}$.
- ד. (5 נק') הניחו שהעכבה Z_f בעלת אופי קיבולי טהור ($Z_f = -jX_c$) וש- $R_2 = R_1 = R$. הסבירו כיצד עכבת המבוא Z_i מייצגת סליל.

פרק שני: שפת תיאור חומרה VHDL

שאלה 5

באיור לשאלה 5 מתוארת מערכת המבצעת פעולת כפל בין שני מספרים בינאריים $(A_{0÷1}, B_{0÷1})$, שכל אחד מהם בעל שתי סיביות. הכפל מתבצע באמצעות ארבעה שערי AND ושני חצאי מסכמים (H.A.), כמתואר באיור.



איור לשאלה 5

להלן קוד בשפת VHDL המתאר חצי מסכם (H.A.).

```
ENTITY HA IS
    PORT (a,b:IN BIT;
          sum, cout: BIT);
END HA;

ARCHITECTURE behave OF HA IS
    BEGIN
        sum <= a XOR b;
        cout <= a AND b;
    END behave;
```

- א. (5 נק') היעזרו בקוד הנתון לעיל וסרטטו את המעגל הלוגי של החצי מסכם (H.A.).
- ב. (12 נק') כתבו תוכנית בשפת VHDL למימוש המערכת MUL2_2, המבצעת תיאור מבני לחיבור בין מרכיבי המערכת (COMPONENTS). יש להגדיר את המבואות, את המוצאים ואת הסיגנלים הפנימיים באמצעות BIT_VECTOR או באמצעות STD_LOGIC_VECTOR. המימוש יתבצע לפי עקרונות התכנון ההיררכי, על בסיס החצי מסכם (H.A.) שהקוד המתאר אותו נתון לעיל.
- ג. (8 נק') עבור הכניסות '10', A = '11', B = כתבו את ערכו של כל אחד מסיגנלים $S_{1÷4}$ ואת ערכו של כל אחד מהדקי המוצא, $Y_{0÷3}$.

שאלה 6

לפניכם תוכנית בשפת VHDL המתארת את המערכת הספרתית Ex6.

```
1.  library IEEE;
2.  use IEEE.std_logic_1164.all;
3.  use ieee.std_logic_arith.all;
4.  use ieee.std_logic_unsigned.all;
5.
6.  entity Ex6 is
7.      port (
8.          clk: in STD_LOGIC;
9.          nrst: in STD_LOGIC;
10.         y1, y2: out STD_LOGIC_VECTOR (1 downto 0)
11.     );
12. end Ex6;
13. -----
14. architecture arch of Ex6 is
15.     signal sig1, sig2 : STD_LOGIC_VECTOR (1 DOWNTO 0);
16. begin
17.     process (clk, nrst)
18.     BEGIN
19.         if nrst= '0' then
20.             sig1 <= (OTHERS => '0') ;
21.             sig2 <= (OTHERS => '0') ;
22.         elsif clk'event and clk ='1' then
23.             if sig1<3 then
24.                 sig1 <= sig1+1 ;
25.             else
26.                 sig1 <= (OTHERS => '0') ;
27.                 if sig2<3 then
28.                     sig2 <= sig2+1 ;
29.                 else
30.                     sig2 <= (OTHERS => '0') ;
31.                 end if;
```



```

32.           end if;
33.           end if;
34.   end process;
35.   y1  <= sig1 ;
36.   y2  <= sig2 ;
37.   end arch ;

```

- א. (6 נק') הסבירו את ההוראות שבשורות 10, 20, 35.
- ב. (4 נק') הסבירו את משמעות התוכן המופיע בסוגריים בשורה 17 (clk, nrst).
- ג. (10 נק')

1. (6 נק') העתיקו את הטבלה שלהלן למחברת והשלימו אותה.

nrst	clk	y1	y2
0			
1			
1			
1			
1			
0			

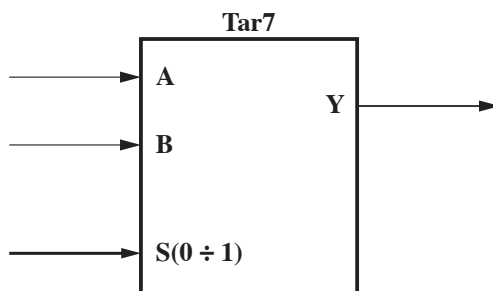
2. (4 נק') לאחר מספר מחזורי שעון מתקבלים הערכים $y_1 = (11)_2$ ו- $y_2 = (10)_2$. העתיקו את הטבלה שלהלן למחברת והשלימו אותה.

nrst	clk	y1	y2
1	כעבור כמה מחזורי שעון	$(11)_2$	$(10)_2$
1			
1			
1			

ד. (5 נק') הסבירו את פעולת המערכת המתוארת בתוכנית.

שאלה 7

באיור לשאלה 7 נתון הסמל של המערכת הספרתית Tar7.



איור לשאלה 7

אותות המבוא A, B ואות המוצא Y הם בעלי סיבית אחת, ואות המבוא S הוא בעל שתי סיביות. כתבו תוכנית בשפת VHDL, שתבצע את הפעולות הכתובות בטבלה שלפניכם ללא שימוש בתהליך (PROCESS).

S(1)	S(0)	Y
0	0	0
0	1	B
1	0	A
1	1	$Y = \overline{A + B}$

שאלה 8

לפניכם תוכנית בשפת VHDL המתארת מערכת ספרתית מסוימת.

```

1  library IEEE;
2  use IEEE.std_logic_1164.all;
3  entity FSM is
4      port (clk, Rst: in std_logic;
5            gB: in std_logic;
6            Y: out integer range 0 to 3);
7  end entity FSM;
8  architecture bv of FSM is
9      type StType is (ST0, ST1, ST2, ST3);
10     signal CurrentST, NextST: StType;
11 begin
12     p1: process (gB, CurrentST)
13     begin
14         case CurrentST is
15             when ST0 => Y <= 0;
16                     NextST <= ST1;
17             when ST1 => Y <= 1;
18                     if (gB = '1') then
19                         NextST <= ST2;
20                     else
21                         NextST <= ST3;
22                     end if;
23             when ST2 => Y <= 2;
24                     if (gB = '1') then
25                         NextST <= ST3;
26                     else
27                         NextST <= ST0;
28                     end if;
29             when ST3 => Y <= 3;
30                     if (gB = '1') then
31                         NextST <= ST0;
32                     else
33                         NextST: ST2;
34                     end if;
35             when others => Y <= 0;
36                     NextSt <= ST0;
37         end case;
38     end process;

```

```
39.      p2: process (clk, Rst)
40.      begin
41.          if (Rst = '1') then
42.              CurrentST <= ST0;
43.          elsif rising_edge(clk) then
44.              CurrentST <= NextST;
45.          end if;
46.      end process p2;
47. end architecture bv;
```

א. (6 נק') הסבירו את ההוראות שבשורות 9, 10, 14, 43.

ב. (6 נק') הסבירו מה מבצע קטע התוכנית שבשורות $22 \div 17$.

ג. (6 נק') סרטטו תרשים בועות המתאר את פעולת המערכת.

ד. (7 נק')

4) נק' 1. כתבו את ארבעת מצבי המערכת כאשר $gB = '0'$. המצב ההתחלתי: $Y = 0$.

3) נק' 2. כתבו את ארבעת מצבי המערכת כאשר $gB = '1'$. המצב ההתחלתי: $Y = 0$.

בהצלחה!

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



ICL8038

Precision Waveform Generator/ Voltage Controlled Oscillator

November 1996

Features

- Low Frequency Drift with Temperature . . . 250ppm/°C
- Low Distortion 1% (Sine Wave Output)
- High Linearity 0.1% (Triangle Wave Output)
- Wide Frequency Range 0.001Hz to 300kHz
- Variable Duty Cycle 2% to 98%
- High Level Outputs TTL to 28V
- Simultaneous Sine, Square, and Triangle Wave Outputs
- Easy to Use - Just a Handful of External Components Required

Description

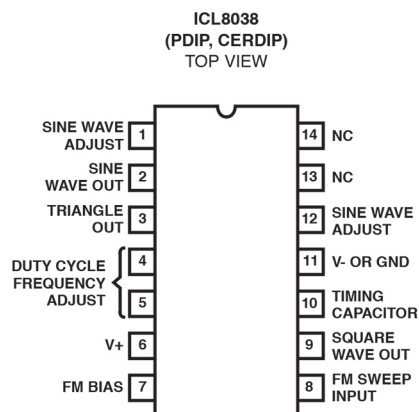
The ICL8038 waveform generator is a monolithic integrated circuit capable of producing high accuracy sine, square, triangular, sawtooth and pulse waveforms with a minimum of external components. The frequency (or repetition rate) can be selected externally from 0.001Hz to more than 300kHz using either resistors or capacitors, and frequency modulation and sweeping can be accomplished with an external voltage. The ICL8038 is fabricated with advanced monolithic technology, using Schottky barrier diodes and thin film resistors, and the output is stable over a wide range of temperature and supply variations. These devices may be interfaced with phase locked loop circuitry to reduce temperature drift to less than 250ppm/°C.

Ordering Information

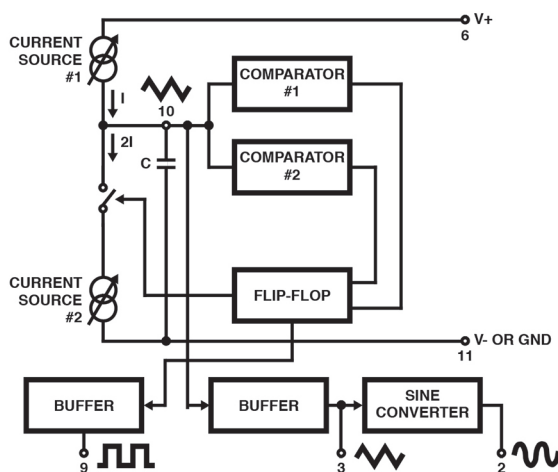
PART NUMBER	STABILITY	TEMP. RANGE (°C)	PACKAGE	PKG. NO.
ICL8038CCPD	250ppm/°C (Typ)	0 to 70	14 Ld PDIP	E14.3
ICL8038CCJD	250ppm/°C (Typ)	0 to 70	14 Ld CERDIP	F14.3
ICL8038BCJD	180ppm/°C (Typ)	0 to 70	14 Ld CERDIP	F14.3
ICL8038ACJD	120ppm/°C (Typ)	0 to 70	14 Ld CERDIP	F14.3
ICL8038BMJD (Note)	350ppm/°C (Max)	-55 to 125	14 Ld CERDIP	F14.3
ICL8038AMJD (Note)	250ppm/°C (Max)	-55 to 125	14 Ld CERDIP	F14.3

NOTE: Add /883B to part number if 883 processing is required.

Pinout



Functional Diagram



ICL8038

The levels of the current sources can, however, be selected over a wide range with two external resistors. Therefore, with the two currents set at values different from 1 and 2I, an asymmetrical sawtooth appears at Terminal 3 and pulses with a duty cycle from less than 1% to greater than 99% are available at Terminal 9.

The sine wave is created by feeding the triangle wave into a nonlinear network (sine converter). This network provides a decreasing shunt impedance as the potential of the triangle moves toward the two extremes.

Waveform Timing

The *symmetry* of all waveforms can be adjusted with the external timing resistors. Two possible ways to accomplish this are shown in Figure 3. Best results are obtained by keeping the timing resistors R_A and R_B separate (A). R_A controls the rising portion of the triangle and sine wave and the 1 state of the square wave.

The magnitude of the triangle waveform is set at $1/3 V_{SUPPLY}$; therefore the rising portion of the triangle is,

$$t_1 = \frac{C \times V}{I} = \frac{C \times 1/3 \times V_{SUPPLY} \times R_A}{0.22 \times V_{SUPPLY}} = \frac{R_A \times C}{0.66}$$

The falling portion of the triangle and sine wave and the 0 state of the square wave is:

$$t_2 = \frac{C \times V}{I} = \frac{C \times 1/3 V_{SUPPLY}}{2(0.22) \frac{V_{SUPPLY}}{R_B} - 0.22 \frac{V_{SUPPLY}}{R_A}} = \frac{R_A R_B C}{0.66(2R_A - R_B)}$$

Thus a 50% duty cycle is achieved when $R_A = R_B$.

If the duty cycle is to be varied over a small range about 50% only, the connection shown in Figure 3B is slightly more convenient. A $1k\Omega$ potentiometer may not allow the duty cycle to be adjusted through 50% on all devices. If a 50% duty cycle is required, a $2k\Omega$ or $5k\Omega$ potentiometer should be used.

With two separate timing resistors, the frequency is given by:

$$f = \frac{1}{t_1 + t_2} = \frac{1}{R_A C \left(1 + \frac{R_B}{2R_A - R_B} \right)}$$

or, if $R_A = R_B = R$

$$f = \frac{0.33}{RC} \text{ (for Figure 3A)}$$

Neither time nor frequency are dependent on supply voltage, even though none of the voltages are regulated inside the integrated circuit. This is due to the fact that both currents and thresholds are direct, linear functions of the supply voltage and thus their effects cancel.

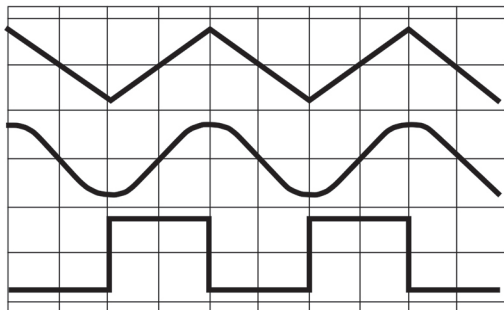


FIGURE 2A. SQUARE WAVE DUTY CYCLE - 50%

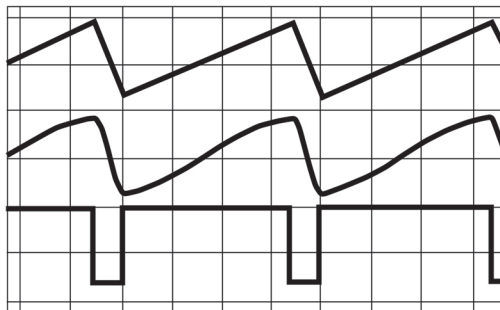


FIGURE 2B. SQUARE WAVE DUTY CYCLE - 80%

FIGURE 2. PHASE RELATIONSHIP OF WAVEFORMS

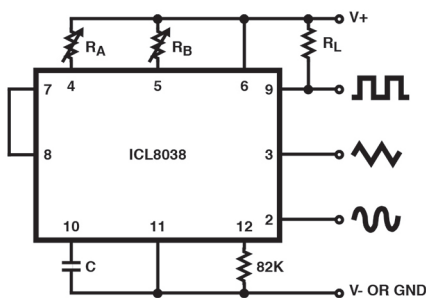


FIGURE 3A.

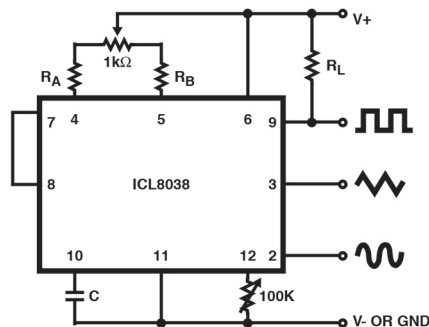


FIGURE 3B.

FIGURE 3. POSSIBLE CONNECTIONS FOR THE EXTERNAL TIMING RESISTORS

נספח ב': מילון מונחים
לשאלון 711003, אביב תשפ"ב

תרגום המונח			המונח
אנגלית	רוסית	ערבית	
input signal	Входной сигнал	إشارة دخل	אות מבוא
output signal	Выходной сигнал	إشارة خرج	אות מוצא
states diagram	Диаграмма состояний	مُخطَّط العلاقات	דיאגרמת מצבים
states table	Таблица состояний	جدول الحالة	טבלת מצבים
components	Составляющие	عَنَاصِر / مكوّنات	מבניות
amplifier	Усилитель	مكبر	מגבר
switching converter	Коммутируемый преобразователь	محوّل تبديل	ממיר ממותג
alternative circuit	Схема замещения	الدائرة البديلة	מעגל תמורה
array	Массив	مصفوفة	מערך
output port	Устройство вывода	مفتاح الإخراج	מפתח פלט
oscillator	Генератор колебаний	مذبذب	מתנד
bias point	Рабочая точка	نقطة الانحياز / نقطة التشغيل	נקודת העבודה
logic value	Логический сигнал	قيمة منطقيّة	ערך לוגי
output	Выходные данные	الإخراج	פלט
file	Файл	ملفّ	קובץ
vibrations frequency	Частота колебаний	تردد الذبذبات	תדר התנודות

אין להעביר את הנוסחאון
לנבחן אחר

נוסחאון במעגלים אלקטרוניים לכיתה י"ד

(15 עמודים)

דיודת צומת

משוואת זרם-מתח של דיודת מעשית:

- זרם הדיודה - I_D [A]
- זרם זליגה אחורי - I_S [A]
- מתח הדיודה - V_D [V]
- מתח התלוי בטמפרטורה - V_T [V]

$$I_D = I_S \left(\frac{V_D}{e \eta V_T} - 1 \right)$$

גרמניום - 1

מקדם - $\eta =$

סיליקון - 2

$$V_D = \eta V_T \ln \left(\frac{I_D}{I_S} + 1 \right)$$

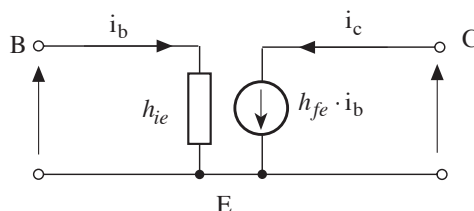
טרנזיסטור דו-נושאי (בתחום הפעיל)

- זרם קולט - I_C [A]
- זרם פולט - I_E [A]
- זרם בסיס - I_B [A]

$$I_C = \beta I_B, I_E = (\beta + 1) I_B, I_E = I_C + I_B$$

$$\alpha = \frac{I_C}{I_E} = \frac{\beta}{\beta + 1}, \beta = \frac{\alpha}{1 - \alpha}$$

תרשים תמורה מקורב מסוג h של טרנזיסטור דו-נושאי



תחום הרוויה בטרנזיסטור דו-נושאי

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

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

$$\beta \cdot I_B > I_C$$

סף רוויה של התחום הפעיל בטרנזיסטור דו-נושאי

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

$$\beta \cdot I_B = I_C$$

טרנזיסטור FET (אזור הרוויה)

זרם אפיק	-	I_D [A]
המתח בין השער למקור	-	V_{GS} [V]
מתח צביטה	-	V_p [V]
זרם האפיק עבור $V_{GS} = 0$	-	I_{DSS} [A]

$$I_D = I_{DSS} \left(1 - \frac{V_{GS}}{V_p} \right)^2$$

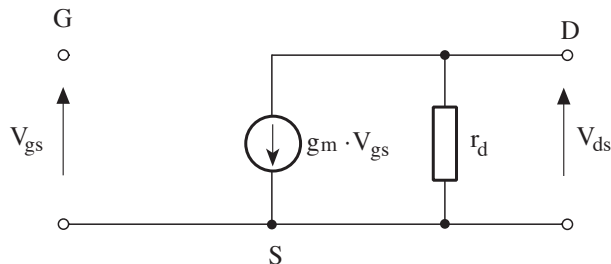
מוליכות הדדית	-	g_m [S]
---------------	---	-----------

$$g_m = \frac{2I_{DSS}}{|V_p|} \left(1 - \frac{V_{GS}}{V_p} \right)$$

מוליכות הדדית עבור $V_{GS} = 0$	-	g_{m0} [S]
---------------------------------	---	--------------

$$g_{m0} = \frac{2I_{DSS}}{|V_p|}$$

תרשים תמורה מקורב של FET



טרנזיסטור MOSFET (אזור הרוויה)

טרנזיסטור מסוג חיזוק הולכה בעל תעלה מסוג N

תנאים לרוויה: $V_{GS} - V_T > 0$ מתח הסף $V_T [V]$

$V_{DS} > V_{GS} - V_T > 0$ מקדם $K = \left[\frac{mA}{V^2} \right]$

הזרם: $I_D = K (V_{GS} - V_T)^2$

טרנזיסטור MOSFET באזור לאוהמי

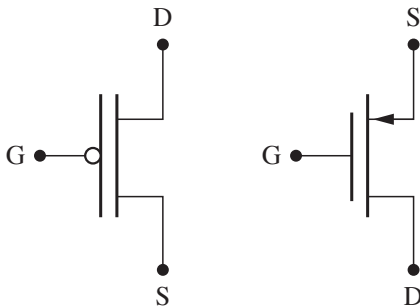
תנאים לאוהמי: $V_{DS} < V_{GS} - V_T$

$V_{GS} - V_T > 0$

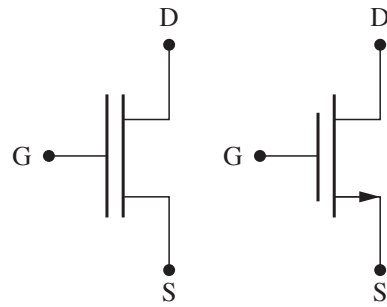
הזרם: $I_D = K \left[2 (V_{GS} - V_T) V_{DS} - V_{DS}^2 \right]$

סימול:

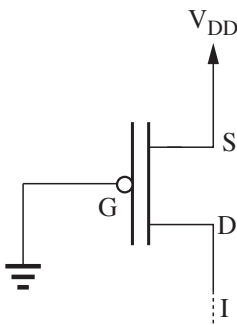
P – channel



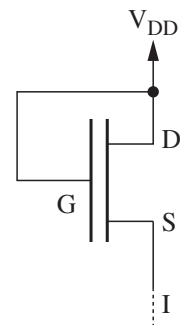
N – channel



P – channel כנגד מושך מעלה



N – channel כנגד מושך מעלה

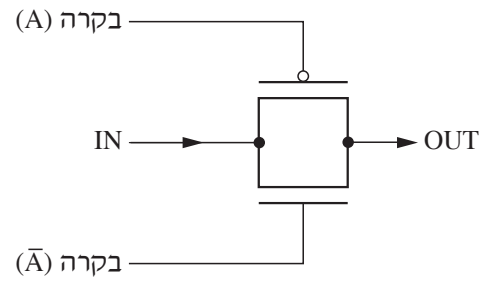


הערה: מעגל התמורה לאות חילופין של MOSFET זהה לזה של J-FET.

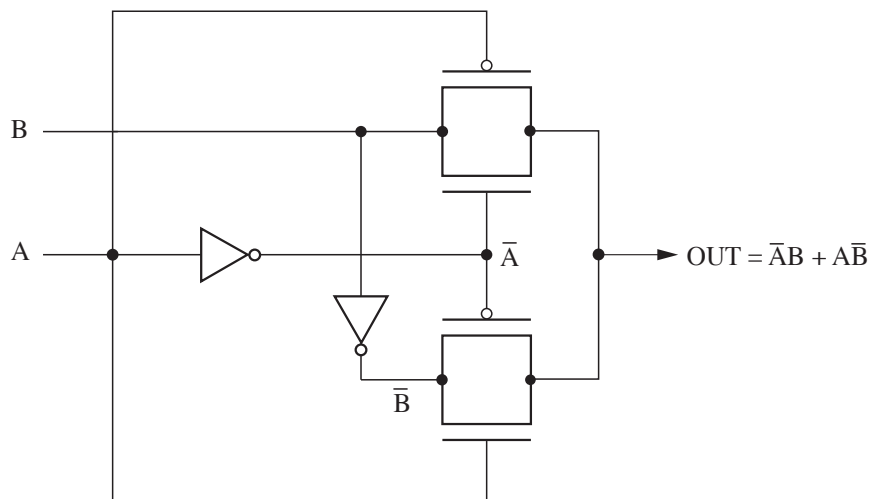
שערי תמסורת:

סימול:

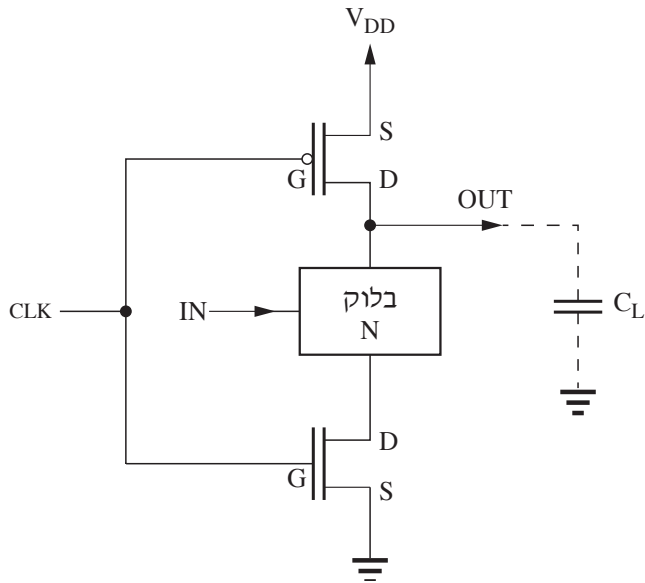
אם $A = 0$ $\leftarrow$ $OUT = IN$
אם $A = 1$ $\leftarrow$ OUT ללא שינוי



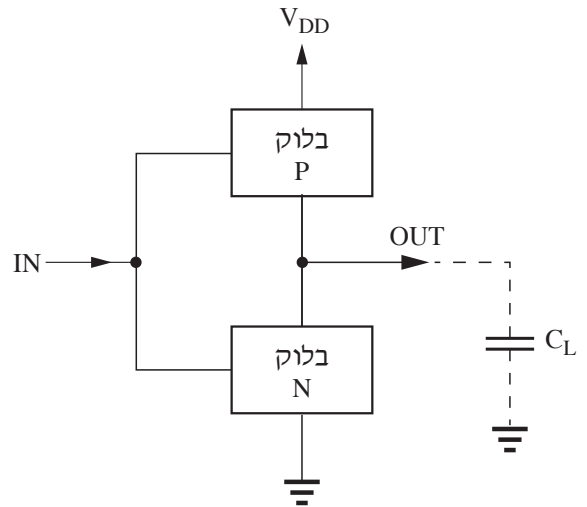
שער XOR באמצעות שערי תמסורת:



שער CMOS דינמי



שער CMOS סטטי



מגברי הספק

מאזן הספקים

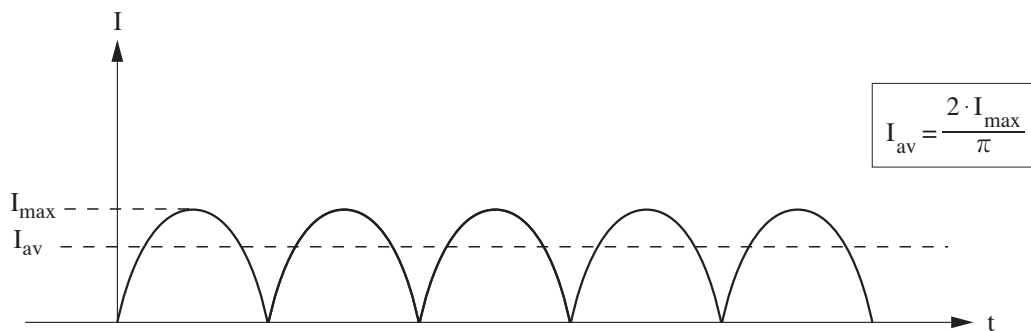
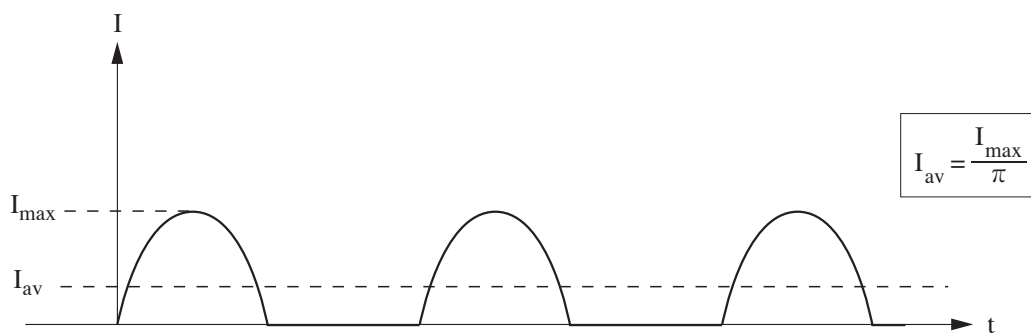
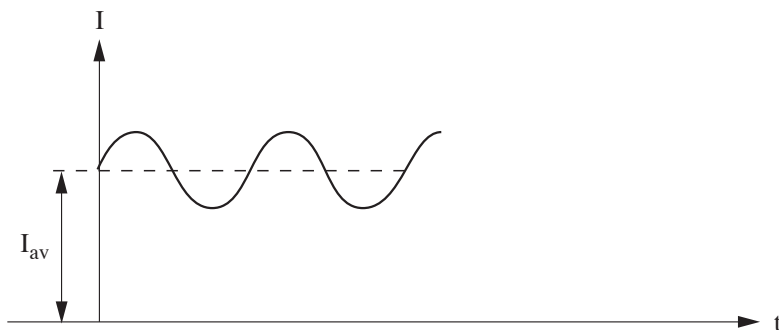
הספק מבוא	-	P_{in}	[W]
הספק נצרך מספק הכוח	-	P_{CC}	[W]
הספק העומס	-	P_L	[W]
הספק מבוזבז	-	P_{diss}	[W]
הספק המתפתח על העומס הנובע מאות חילופין	-	P_{LAC}	[W]
המתח היעיל על העומס	-	$V_{L_{eff}}$	[V]
הזרם היעיל על העומס	-	$I_{L_{eff}}$	[A]
נגד העומס	-	R_L	[Ω]
המתח הסינוסי המרבי על העומס	-	$V_{L_{max}}$	[V]
הזרם הסינוסי המרבי דרך העומס	-	$I_{L_{max}}$	[A]
הזרם הממוצע (DC) המסופק על-ידי ספק הכוח	-	I_{av}	[A]

$$P_{in} + P_{CC} = P_L + P_{diss}$$

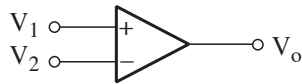
$$P_{LAC} = V_{L_{eff}} \cdot I_{L_{eff}} = \frac{V_{L_{eff}}^2}{R_L} = I_{L_{eff}}^2 \cdot R_L$$

$$V_{L_{eff}} = \frac{V_{L_{max}}}{\sqrt{2}}, \quad I_{L_{eff}} = \frac{I_{L_{max}}}{\sqrt{2}}$$

$$P_{CC} = V_{CC} \cdot I_{av}$$



מגבר הפרשי



מגברי הפרש

$$V_o = A_1 \cdot V_1 + A_2 \cdot V_2$$

$$A_1 = \frac{V_o}{V_1} \Big|_{V_2=0}$$

מתח מבוא V_1 [V] -

$$A_2 = \frac{V_o}{V_2} \Big|_{V_1=0}$$

מתח מבוא V_2 [V] -

$$V_d = V_1 - V_2$$

מתח הפרשי V_d [V] -

$$V_c = \frac{V_1 + V_2}{2}$$

מתח משותף V_c [V] -

$$A_d = \frac{V_o}{V_d} \Big|_{V_c=0} = \frac{V_o}{2V_i}$$

$$A_c = \frac{V_o}{V_c} \Big|_{V_d=0} = \frac{V_o}{V_i}$$

הגבר הפרשי A_d -

$$A_d = \frac{A_1 - A_2}{2}$$

הגבר האות המשותף A_c -

$$A_c = A_1 + A_2$$

CMRR - יחס דחיית האות המשותף

$$CMRR = \left| \frac{A_d}{A_c} \right|$$

$$V_o = A_d \cdot V_d + A_c \cdot V_c$$

ממירים ממותגים

א. משוואת מתח-זרם של סליל

(כאשר השינוי
בזרם הסליל
הוא לינארי)

$$v_L = L \frac{di_L}{dt}$$

$$V_L = L \frac{\Delta I_L}{\Delta T}$$

מתח רגעי על הסליל	-	v_L [V]
זרם רגעי בסליל	-	i_L [A]
זמן	-	t [sec]
השראות הסליל	-	L [H]
מתח על הסליל	-	V_L [V]
השינוי בזרם הסליל	-	ΔI_L [A]
השינוי בזמן	-	ΔT [sec]
זרם רגעי בקבל	-	i_C [A]
מתח רגעי על הקבל	-	v_C [V]
קיבול הקבל	-	C [F]
זרם הקבל	-	I_C [A]
השינוי במתח על הקבל	-	ΔV_C [V]

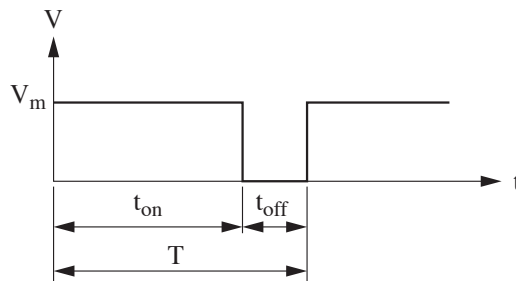
ב. משוואת זרם-מתח של קבל

(כאשר השינוי
במתח הקבל
הוא לינארי)

$$i_C = C \frac{dv_C}{dt}$$

$$I_C = C \frac{\Delta V_C}{\Delta T}$$

ג. מחזור פעולה



D - מחזור הפעולה (Duty cycle)

$$D = \frac{t_{on}}{t_{on} + t_{off}} = \frac{t_{on}}{T}$$

ד. ממיר (BUCK) STEP DOWN

מחזור הפעולה (Duty Cycle)	-	D	
זמן פעולה	-	ton	[sec]
זמן פוגה	-	toff	[sec]
זמן מחזור	-	T	[sec]
מתח מוצא	-	Vo	[V]
מתח מבוא	-	Vin	[V]
זרם מוצא	-	Io	[A]
זרם מבוא	-	Iin	[A]
אדוות־המתח הדרושה	-	ΔVo	[V]
הספק במוצא	-	Po	[W]
הספק במבוא	-	Pin	[W]
נצילות	-	η	

בממיר
אידיאלי

$$V_o = D \cdot V_{in}$$

$$I_o = \frac{I_{in}}{D}$$

$$L = \frac{(V_{in} - V_o)}{\Delta I_L} \cdot D \cdot T$$

$$C = \frac{V_o}{\Delta V_o} \cdot \frac{T^2 (1 - D)}{8 L}$$

או

$$\Delta V_o = \frac{\Delta I_L}{8 \cdot f \cdot c} = \frac{V_{in} \cdot D \cdot (1 - D)}{8 \cdot L \cdot C \cdot f^2}$$

$$P_o = \eta \cdot P_{in}$$

ה. (BOOST) STEP UP

מחזור הפעולה (Duty Cycle)	-	D		בממיר אידיאלי $V_o = \frac{V_{in}}{1 - D}$ $I_o = (1 - D) \cdot I_{in}$ $C = \frac{I_o}{\Delta V_o} \cdot D \cdot T$ $L = \frac{V_{in}}{\Delta I_L} \cdot D \cdot T$ $P_o = \eta \cdot P_{in}$
זמן פעולה	-	t _{on}	[sec]	
זמן פוגה	-	t _{off}	[sec]	
זמן מחזור	-	T	[sec]	
מתח מוצא	-	V _o	[V]	
מתח מבוא	-	V _{in}	[V]	
זרם מוצא	-	I _o	[A]	
זרם מבוא	-	I _{in}	[A]	
קיבול הקבל	-	C	[F]	
אדוות המתח הדרושה	-	ΔV _o	[V]	
השראות הסליל	-	L	[H]	
השינוי בזרם הסליל	-	ΔI _L	[A]	
הספק במוצא	-	P _o	[W]	
הספק במבוא	-	P _{in}	[W]	
נצילות	-	η		

מתנדים סינוסואידליים לתדר נמוך

קריטריון ברקהאוזן לקיום תנודות:

1. תנאי התנופה: $|\beta \cdot A| \geq 1$ - הגבר בחוג פתוח A

2. תנאי המופע: $\angle \beta \cdot A = 0 + 360^\circ \cdot k$ - β מקדם המשוב

מתנד גשר ווין

- f [Hz] - תדר התנודות
- R [Ω] - ערכו של כל אחד מהנגדים ברשת המשוב β
- C [F] - ערכו של כל אחד מהקבלים ברשת המשוב β
- R_f [Ω] - נגד המשוב ברשת המשוב השלילי
- R_1 [Ω] - נגד ברשת המשוב השלילי

$$f = \frac{1}{2\pi RC}$$

$$\frac{R_f}{R_1} \geq 2$$

תדר התנודות ברשת משוב LC

- L [H] - השראות
- C [F] - קיבול

$$f = \frac{1}{2\pi \sqrt{LC}}$$

מגבר משוב שלילי

- A_f - הגבר בחוג סגור
- A - הגבר בחוג פתוח
- β - רשת המשוב

$$A_f = \frac{A}{1 + A\beta}$$

משוואות הדפקים היסודית

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

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

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

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

טעינה לינארית

טעינת קבל בזרם קבוע:

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

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

טעינת סליל במתח קבוע:

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

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

מתח מוצא בזמן t	-	$v(t)$ [V]
מתח סופי (עבור $t \rightarrow \infty$) במצב המתמיד	-	V_{∞} [V]
מתח התחלתי	-	V_{0+} [V]
זמן	-	t [sec]
קבוע הזמן	-	τ [sec]
זרם מוצא בזמן t	-	$i(t)$ [A]
זרם סופי (עבור $t \rightarrow \infty$) במצב המתמיד	-	I_{∞} [A]
זרם התחלתי	-	I_{0+} [A]
התנגדות שקולה ש"רואה"	-	R_{eq} [Ω]
הרכיב ההיגבי, מחושבת לפי תבנית		

מתח הקבל	-	V_C [V]
זרם הקבל	-	I_C [A]
קיבול	-	C [F]
זמן	-	t [sec]
זרם הסליל	-	I_L [A]
מתח הסליל	-	V_L [V]
השראות	-	L [H]

טבלאות מצבים של דלגלים
א. טבלת המצבים של JKFF

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

ב. טבלת המצבים של SRFF (סינכרוני)

CLK	S	R	Q
0, 1	∅	∅	N.C
	0	0	N.C
	0	1	0
	1	0	1
	1	1	מצב אסור

ג. טבלת המצבים של TFF

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

ד. טבלת המצבים של DFF

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

ה. 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		

אין להעביר את הנוסחאון
לנבחן אחר

נוסחאון בשפת תיאור חומרה VHDL לכיתה י"ד

(13 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 תוכנית

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 תוכנית
__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

LOGICAL OPERATORS - פעולות לוגיות		
Operator	Operand Type פועל על	Result Type מחזיר
not, or, and, nor, nand, xor, xnor	boolean, bit_vector, std_logic_vector	same type

Relational Operators - פעולות יחס			
Operator	Operation	Operand Type פועל על	Result Type מחזיר
=	Equality שווה	any type	Boolean
/=	Inequality שונה	any type	Boolean
<	less than קטן מ	any scalar type	Boolean
<=	less than or equal to קטן או שווה	any scalar type	Boolean
>	greater than גדול מ	any scalar type	Boolean
>=	greater than or equal to גדול או שווה	any scalar type	Boolean

Arithmetic Operators - פעולות אריתמטיות			
Operator	Operation	Operand Type פועל על	Result Type מחזיר
**	Exponential חזקה	numeric**integer	numeric
abs	absolute value ערך מוחלט	numeric	numeric
*	Multiplication כפל	numeric * numeric	numeric
/	Division חילוק	numeric / numeric	integer
mod	Modulus החלק השלם	integer mod integer	integer
rem	remainder שארית	integer rem integer	numeric
+	unary plus חיבור	+ numeric	numeric
-	unary minus חיסור	- numeric	numeric

MISCELLANEOUS OPERATORS			
Operator	Operation	Operand Type פועל על	Result Type מחזיר
&	Concatenation שרשור	array or element & array or element	array

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 בתוכנית 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	
<pre>__loop_label: FOR __index_variable IN __range LOOP __statement; __statement; END LOOP __loop_label;</pre>	FOR לולאת
<pre>__loop_label: WHILE __boolean_expression LOOP __statement; __statement; END LOOP __loop_label;</pre>	WHILE לולאת

התניות בגוף התוכנית – 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

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

FIVE_K

D

Y[3....0]

S[1....0]

```

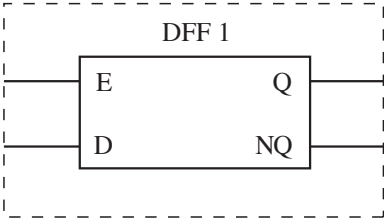
library ieee;
use ieee.std_logic_1164.all;

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

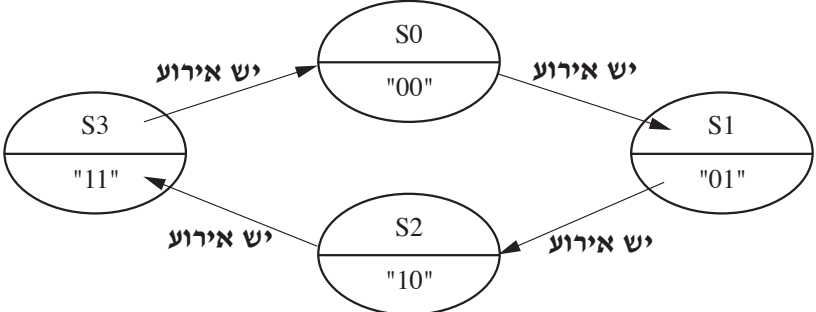
architecture behave of DEMUX is
begin
process (s,d)
begin
    case s is
        when "00" => y <= '0' & '0' & '0' & d;
        when "01" => y <= '0' & '0' & d & '0';
        when "10" => y <= '0' & d & '0' & '0';
        when "11" => y <= d & '0' & '0' & '0';
    end case;
end process;
end behave;

```

דוגמה לתוכנית שלמה המבצעת פעולת מפלג, תוך שימוש בהתניית CASE

התניות בהליך טורי – בתוך PROCESS	
פקודת השהיה WAIT wait; עצירה למשך הזמן שיוגדר ; wait on רשימת משתנים ; wait until תנאי לוגי ; wait for מספר יחידות זמן ;	
דוגמה למימוש DFF <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 (e) Begin Wait until e'event and e = '1' ; Q <= d; Nq <= not d; End process; End behave; </pre> התוכנית 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 בדוגמה 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 עלייה</td><td>F</td><td>נשמר מצב קודם</td></tr><tr><td>0</td><td>0</td><td>עלייה</td><td>0</td><td>0</td></tr><tr><td>0</td><td>0</td><td>עלייה</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 עלייה	F	נשמר מצב קודם	0	0	עלייה	0	0	0	0	עלייה	1	1	דוגמה למימוש DFF עם מבואות ישירים
PRE	CLR	ENABLE	D	Q																														
1	0	F	F	1																														
0	1	F	F	0																														
0	0	אין עלייה	F	נשמר מצב קודם																														
0	0	עלייה	0	0																														
0	0	עלייה	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 state_machine is port (clk : in bit; count_out : out bit_vector (1 downto 0)); end state_machine; architecture behave of 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>	דוגמה לכתובת מונה במכונת 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 לפונקצייה end behave; </pre>	דוגמה לתוכנית המשתמשת בפונקצייה המחברת 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 לתוכנית המשתמשת בפרוצדורה המחברת 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