

## אלקטרוניקה ומחשבים ג'

שתי יחידות לימוד (השלמה לחמש יחידות לימוד)  
(כיתה י"א)

### הוראות לנבחן

א. משך הבחינה: שלוש שעות.

ב. מבנה השאלון ומפתח ההערכה: בשאלון זה שני פרקים. יש לענות על חמש שאלות, שאלה אחת לפחות מכל פרק. לכל שאלה – 20 נקודות. סך הכול – 100 נקודות.

ג. חומר עזר מותר לשימוש: מחשבון.

ד. הוראות מיוחדות:

1. 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 שלהן.
6. אם לדעתך חסרים נתונים הדרושים לפתרון שאלה, אתה רשאי להוסיף אותם, בתנאי שתנמק מדוע הוספת אותם.
7. בכתיבת פתרונות חישוביים, קבלת מִרְב 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, ביחד עם יחידות המידה המתאימות.
  - \* ליווי הפתרון החישובי בהסבר קצר.

בשאלון זה 10 עמודים ונוסחאון.

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

בהצלחה!

## השאלות

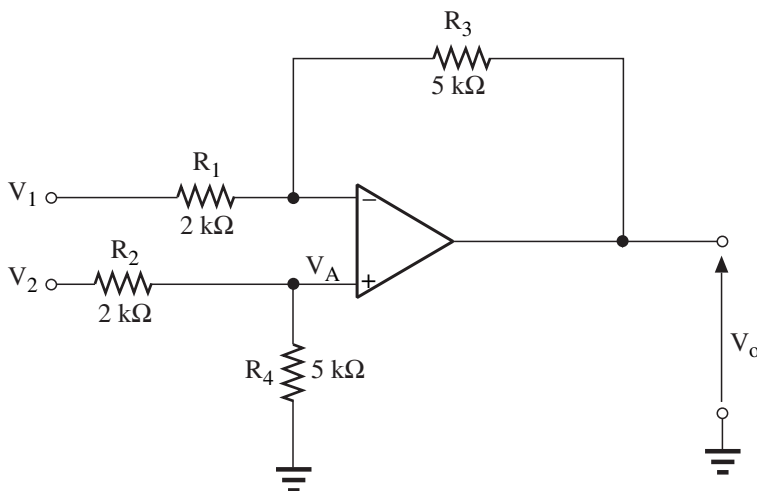
בשאלון זה שני פרקים ובהם תשע שאלות. יש לענות על חמש שאלות בלבד, שאלה אחת לפחות מכל פרק.

### פרק ראשון: מבוא להנדסת אלקטרוניקה

ענה על שאלה אחת לפחות מבין השאלות 1-5 (לכל שאלה – 20 נקודות).

#### שאלה 1

המעגל שבאיור לשאלה 1 כולל מגבר שרת אידיאלי. נתון:  $V_1 = -2\text{ V}$ ,  $V_2 = 1.4\text{ V}$ .

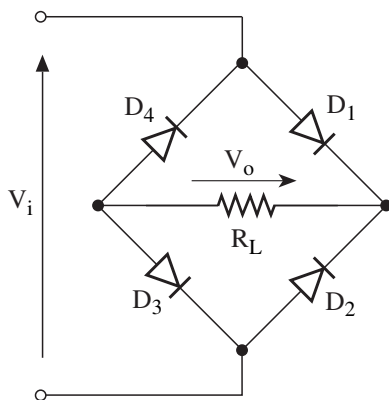


#### איור לשאלה 1

- חשב את המתח  $V_A$  בכניסת ה- (+) של מגבר השרת.
- חשב את מתח המוצא  $V_o$ .
- חשב את הזרם בנגד  $R_1$ .

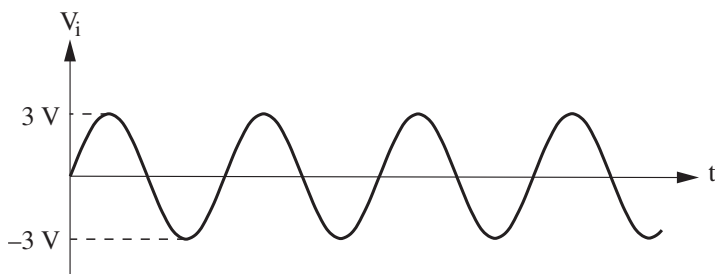
## שאלה 2

הדיודות במעגל שבאיור א' לשאלה 2 – אידיאליות.



איור א' לשאלה 2

- א. רשום את ייעודו של המעגל.  
 ב. נתון:  $V_i = 2\text{ V}$  (מתח DC).  
 1. ציין באיזה מצב (ON/OFF) נמצאת כל אחת מן הדיודות.  
 2. מה יהיה ערכו של מתח המוצא  $V_o$  ?  
 ג. מספקים למעגל את אות המבוא  $V_i$ , המתואר באיור ב' לשאלה.

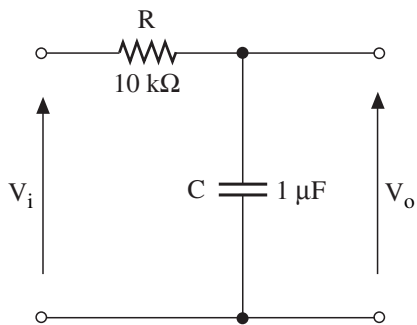
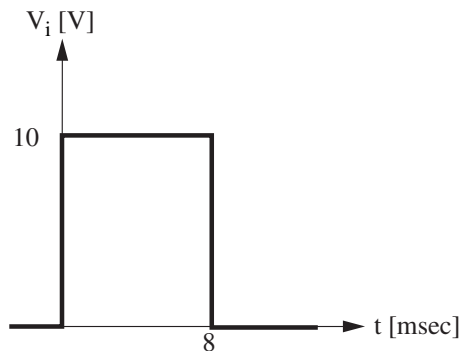


איור ב' לשאלה 2

העתק למחברתך את אות המבוא  $V_i$ , וסרטט מתחתיו, בהתאמה, את אות המוצא  $V_o$ , בתלות בזמן.

### שאלה 3

לרשת שבאיור לשאלה 3 מספקים דופק שעוצמתו 10 V ורוחבו 8 msec .

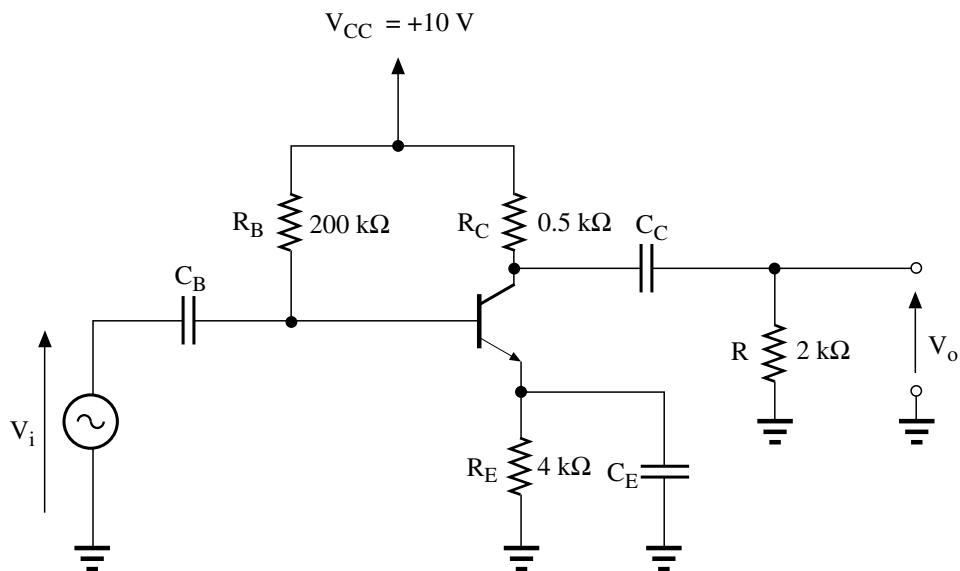


### איור לשאלה 3

- א. העתק למחברתך את מתח הכניסה  $V_i$  וסרטט מתחתיו, בהתאמה, את מתח המוצא  $V_o$ , בתלות בזמן.
- ב. חשב את המתח המרבי שהקבל C נטען אליו.
- ג. חשב את מתח המוצא  $V_o$  כאשר  $t = 10 \text{ msec}$ .

### שאלה 4

להלן נתוניו של המגבר הטרנזיסטורי שבאיור לשאלה 4 :  
 $\beta = 50$  ,  $V_{BE} = 0.7 \text{ V}$  ,  $h_{ie} = 1.1 \text{ k}\Omega$  ,  $h_{fe} = 50$   
 היגבי הקבלים במעגל – זניחים.

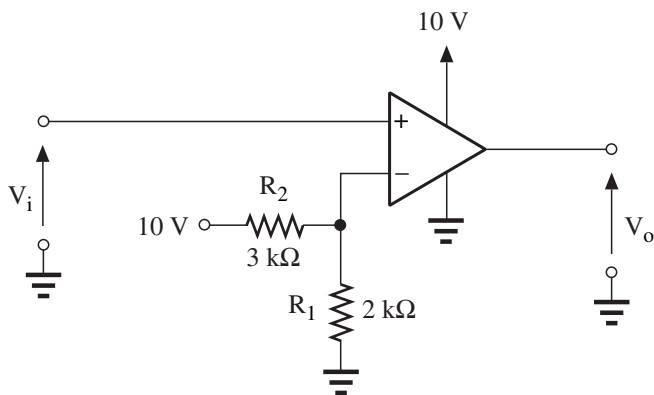


### איור לשאלה 4

- א. חשב את נקודת העבודה של הטרנזיסטור ( $I_C$ ,  $V_{CE}$ ).
- ב. סרטט מעגל תמורה של המגבר לאות חילופין.
- ג. חשב את הגבר המתח  $A_V = \frac{V_o}{V_i}$ .

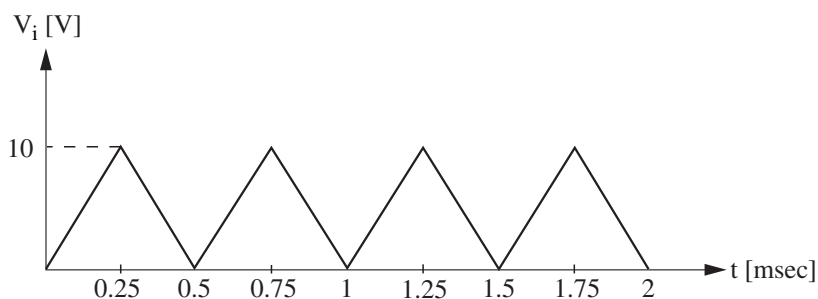
### שאלה 5

המעגל שבאיור א' לשאלה 5 כולל מגבר שרת אידיאלי.  
 מתחי הרוויה של מגבר השרת הם 10 V ו-0 V .



איור א' לשאלה 5

- א. חשב את המתח על הנגד  $R_1$  .
- ב. סרטט את אופיין המעבר  $V_o = f(V_i)$  של המעגל הנתון.
- ג. למבוא המעגל מספקים גל משולש, המתואר באיור ב' לשאלה.



איור ב' לשאלה 5

העתק למחברתך את מתח הכניסה  $V_i$  וסרטט מתחתיו, בהתאמה, את מתח המוצא  $V_o$ .

### פרק שני: מבוא להנדסת מחשבים

ענה על שאלה אחת לפחות מבין השאלות 6-9 (לכל שאלה – 20 נקודות).

#### שאלה 6

לפניך תת-שגרה, הכתובה בשפת הסף של המיקרו מעבד 8086/88.

1. CHANGE: MOV SI, 20 H
2. MOV CL, 4 H
3. MOV BL, 6 H
4. CHG: MOV AL, [SI]
5. ROL AL, CL
6. MOV [SI], AL
7. INC SI
8. DEC BL
9. JNZ CHG
10. RET

א. הסבר את ההוראות שמספריהן 4, 5, 9, 10.

ב. הסבר מה מבצעת תת-השגרה.

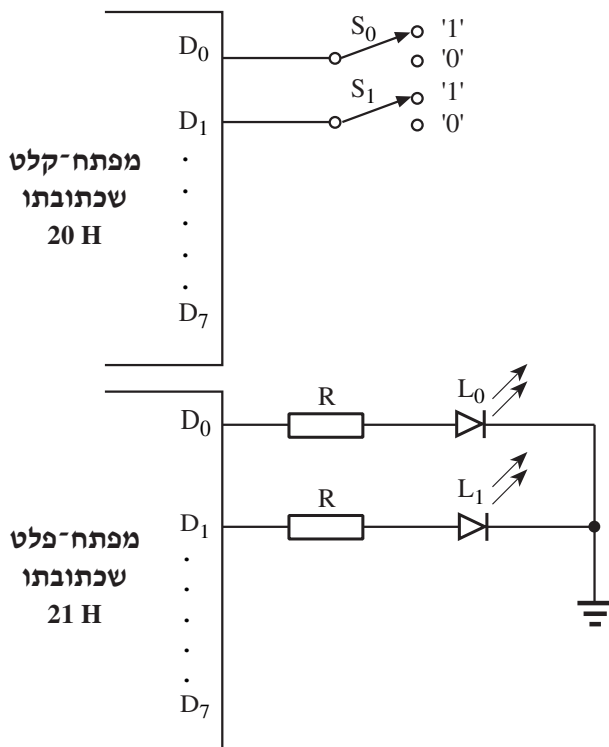
ג. להלן התכנים של תאי הזיכרון  $20\text{ H} \div 25\text{ H}$  לפני ביצוע תת-השגרה:

25 H	24 H	23 H	22 H	21 H	20 H	כתובת התא
37 H	21 H	87 H	56 H	48 H	35 H	תוכן התא

מה יהיו תכני התאים  $20\text{ H} \div 25\text{ H}$  לאחר ביצוע תת-השגרה?

### שאלה 7

באיור לשאלה 7 נתונים מפתח-קלט, הנמצא בכתובת 20 H, ומפתח-פלט, הנמצא בכתובת 21 H.



איור לשאלה 7

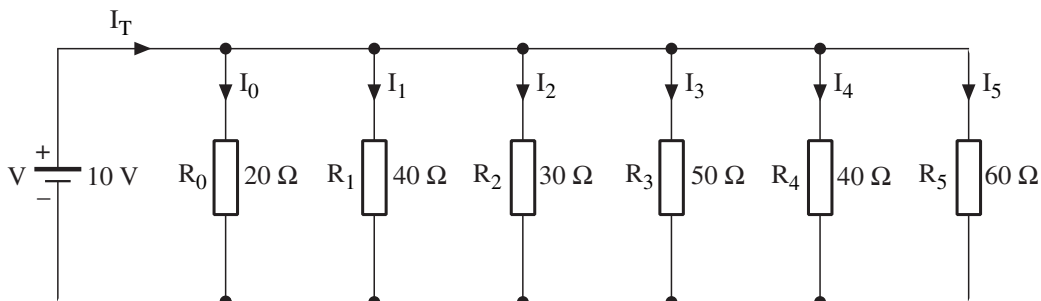
כתוב תכנית בשפת ASM-86 שתבדוק את המצב של כל אחד מן המתגים  $S_0$  ו- $S_1$ , ובהתאם לו תקבע את מצב הנוריות  $L_0$  ו- $L_1$  כמפורט בטבלה שלהלן:

$S_1$	$S_0$	$L_1$	$L_0$
0	0	ON	ON
0	1	ON	OFF
1	0	OFF	ON
1	1	OFF	OFF



### שאלה 8

באיור לשאלה 8 נתון מעגל חשמלי.



### איור לשאלה 8

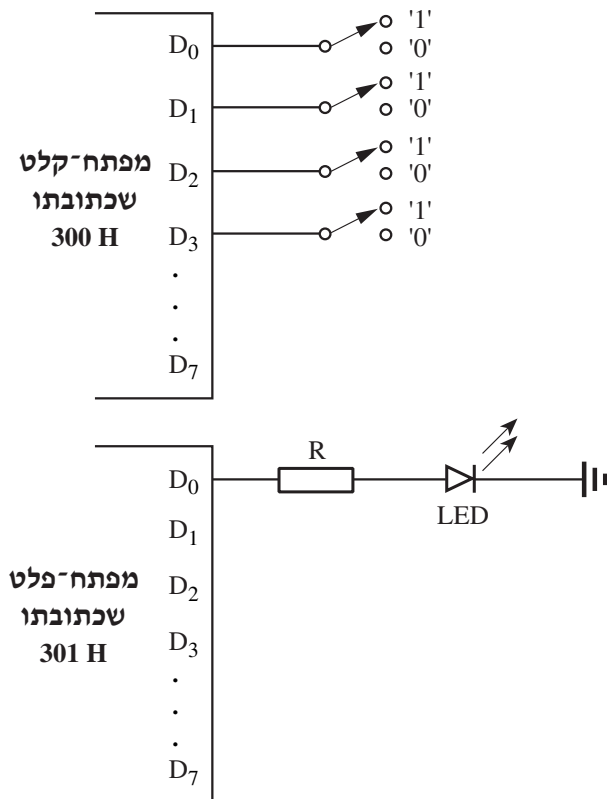
כתוב תכנית (בשפת C או בשפת VB) שתבצע את הפעולות האלה:

1. תגדיר מערך, ותציב את התנגדויות הנגדים שבמעגל במערך הזה.
2. תחשב את הזרם בכל אחד מן הנגדים ותדפיס את ערכו.
3. תחשב את הזרם  $I_T$  ואת ההתנגדות השקולה  $R_T$  של הנגדים במעגל, ותדפיס את ערכיהם.

**הערה:** יש להשתמש בלולאה לביצוע הנדרש בתכנית.

## שאלה 9

באיור לשאלה 9 נתונים מפתח-קלט שכתובתו 300 H, ומפתח-פלט שכתובתו 301 H.



### איור לשאלה 9

כתוב תכנית (בשפת C או בשפת VB) המבצעת את הפעולות האלה:

1. קוראת נתון מספרי N ממפתח-הקלט.
2. גורמת להבהוב של נורית ה-LED שבמוצא מפתח-הפלט N פעמים.

**הערה:** יש לבצע כל הבהוב של נורית ה-LED באופן הבא:

הדלקת הנורית, השהיית התכנית למשך 0.2 sec, כיבוי הנורית והשהיית התכנית למשך 0.1 sec.

### בהצלחה!

אין להעביר את הנוסחאון  
לנבחן אחר

## נוסחאון באלקטרוניקה ומחשבים

(16 עמודים)

### 1. נוסחאות באלקטרוניקה תקבילית

#### חישובי הגבר

הגבר מתח —  $A_V$

מתח מוצא —  $V_o$  [V]

מתח מבוא —  $V_i$  [V]

$$A_V = \frac{V_o}{V_i}$$

הגבר מתח בדציבלים —  $A_V$  [dB]

$$A_V = 20 \log \frac{V_o}{V_i}$$

הגבר זרם —  $A_I$

זרם מוצא —  $I_o$  [A]

זרם מבוא —  $I_i$  [A]

$$A_I = \frac{I_o}{I_i}$$

הגבר זרם בדציבלים —  $A_I$  [dB]

$$A_I = 20 \log \frac{I_o}{I_i}$$

הגבר הספק —  $A_P$

הספק מוצא —  $P_o$  [W]

הספק מבוא —  $P_i$  [W]

התנגדות נגד העומס —  $R_L$  [ $\Omega$ ]

התנגדות מבוא —  $R_i$  [ $\Omega$ ]

$$A_P = \frac{P_o}{P_i} = A_V \cdot A_I = A_I^2 \cdot \frac{R_L}{R_i} = A_V^2 \cdot \frac{R_i}{R_L}$$

הגבר הספק בדציבלים —  $A_P$  [dB]

$$A_P = 10 \log \frac{P_o}{P_i}$$

הגבר כולל של N דרגות —  $A_{VT}$  — המחבורות בשרשרת (קסקדה)

$$A_{VT} = A_{V1} \cdot A_{V2} \cdot A_{V3} \dots A_{VN}$$

$$A_{VT} \text{ (dB)} = A_{V1} \text{ (dB)} + A_{V2} \text{ (dB)} + A_{V3} \text{ (dB)} + \dots + A_{VN} \text{ (dB)}$$

הגבר כולל בדציבלים של —  $A_{VT}$  [dB] — דרגות המחבורות בשרשרת (קסקדה)

**מאזן הספקים**

הספק מבוא —  $P_I$  [W]

הספק נצרך מהספקים —  $P_{CC}$  [W]

הספק העומס —  $P_L$  [W]

הספק מבוזבז —  $P_{diss}$  [W]

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

**משוב שלילי**

הגבר עם משוב (בחוג סגור) —  $A_f$  [A]

הגבר ללא משוב —  $A$

(הגבר חוג פתוח)

מקדם משוב —  $\beta$

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

**משוב מתח טורי**

התנגדות מבוא עם משוב —  $R_{if}$  [ $\Omega$ ]

התנגדות מבוא ללא משוב —  $R_i$  [ $\Omega$ ]

התנגדות מוצא עם משוב —  $R_{of}$  [ $\Omega$ ]

התנגדות מוצא ללא משוב —  $R_o$  [ $\Omega$ ]

$$R_{if} = R_i (1 + \beta A)$$

$$R_{of} = \frac{R_o}{1 + \beta A}$$

**טרנזיסטור דו-נושאי (בתחום הפעיל)**

בהזנחת זרם הזליגה  $I_{CBO}$  :

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

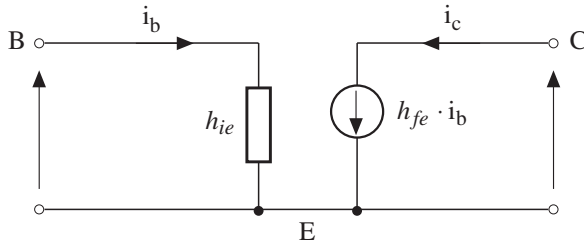
זרם הקולט —  $I_C$  [A]

זרם הפולט —  $I_E$  [A]

זרם הבסיס —  $I_B$  [A]

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

**תרשים תמורה מקורב מסוג h של טרנזיסטור דו-נושאי**



**טרנזיסטור בחיבור פולט (אמיטר) משותף**

	ללא נגד $R_E$	עם נגד $R_E$
$A_I$	$h_{fe}$	$h_{fe}$
$R_i$	$h_{ie}$	$h_{ie} + (1 + h_{fe})R_E$
$A_V$	$-\frac{h_{fe} \cdot R_L}{h_{ie}}$	$-\frac{h_{fe} \cdot R_L}{R_i}$
$R_o$	$\infty$	$\infty$

**מגברי שרת**

מגבר מהפך

$$A_V = - \frac{R_f}{R_1}$$

- נגד המשוב —  $R_f$  [ $\Omega$ ]
- הנגד המחובר לכניסה המהפכת —  $R_1$  [ $\Omega$ ]

מגבר עוקב

$$A_V = 1 + \frac{R_f}{R_1}$$

- נגד המשוב —  $R_f$  [ $\Omega$ ]
- הנגד היוצא מהכניסה המהפכת לאדמה —  $R_1$  [ $\Omega$ ]

**2. נוסחאות באלקטרוניקה ספרתית**

- הערך הרגעי של המתח —  $v(t)$  [V]
- הערך שאליו המתח שואף להגיע כאשר  $t \rightarrow \infty$  —  $V_\infty$  [V]
- הערך ההתחלתי של המתח —  $V_{0+}$  [V]

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

$$t = -\tau \cdot \ln \left( \frac{V_\infty - v(t)}{V_\infty - V_{0+}} \right)$$

- קבוע זמן —  $\tau$  [sec]
- התנגדות —  $R$  [ $\Omega$ ]
- קיבול —  $C$  [F]

$$\tau = RC$$

- תדר חצי הספק עליון של רשת מעבירת נמוכים —  $f_H$  [Hz]
- תדר חצי הספק תחתון של רשת מעבירת גבוהים —  $f_L$  [Hz]
- זמן עלייה של רשת מעבירת נמוכים —  $t_r$  [sec]

$$f_H = \frac{1}{2\pi\tau}$$

$$f_L = \frac{1}{2\pi\tau}$$

$$t_r = 2.2\tau$$

### 3. אוסף פקודות למיקרו-מעבדים 8086/88

מקרא לאוגר הדגלים:

- X מושפע מהפעולה (Modified)
- U לא מוגדר אחרי הפעולה (Undefined)
- R מוחזר מהמחסינית (Returned)
- 0 — מתאפס
- 1 — מקבל '1'

ADC	ADC destination, source Add with carry			Flags
				O D I T S Z A P C X X X X X
Operands	Clocks	Transfers*	Bytes	Coding Example
register, register	3(3)	—	2	ADC AX, SI
register, memory	9(10)+EA	1	2-4	ADC CX, BETA [SI]
memory, register	16(10)+EA	2	2-4	ADC ALPHA [BX] [SI], DI
register, immediate	4(4)	—	3-4	ADC BX, 256
memory, immediate	17(16)+EA	2	3-6	ADC GAMMA, 30H
accumulator, immediate	4(3-4)	—	2-3	ADC AL, 5

ADD	ADD destination, source Addition			Flags
				O D I T S Z A P C X X X X X
Operands	Clocks	Transfers*	Bytes	Coding Example
register, register	3(3)	—	2	ADD CX, DX
register, memory	9(10)+EA	1	2-4	ADD DI, [BX].ALPHA
memory, register	16(10)+EA	2	2-4	ADD TEMP, CL
register, immediate	4(4)	—	3-4	ADD CL, 2
memory, immediate	17(16)+EA	2	3-6	ADD ALPHA, 2
accumulator, immediate	4(3-4)	—	2-3	ADD AX, 200

AND	AND destination, source Logical and			Flags
				O D I T S Z A P C X X X U X X
Operands	Clocks	Transfers*	Bytes	Coding Example
register, register	3(3)	—	2	AND AL, BL
register, memory	9(10)+EA	1	2-4	AND CX, FLAG_WORD
memory, register	16(10)+EA	2	2-4	AND ASCII [DI], AL
register, immediate	4(4)	—	3-4	AND CX, 0F0H
memory, immediate	17(16)+EA	2	3-6	AND BETA, 01H
accumulator, immediate	4(3-4)	—	2-3	AND AX, 01010000B

<b>CALL</b>	CALL target Call a procedure			Flags O D I T S Z A P C
<b>Operands</b>	<b>Clocks</b>	<b>Transfers*</b>	<b>Bytes</b>	<b>Coding Example</b>
near-proc	19(14)	1	3	CALL NEAR_PROC
far-proc	28(23)	2	5	CALL FAR_PROC
memptr 16	21(19)+EA	2	2-4	CALL PROC_TABLE [SI]
regptr 16	16(13)	1	2	CALL AX
memptr 32	37(38)+EA	4	2-4	CALL [BX], TASK [SI]

<b>CLC</b>	CLC (no operands) Clear carry flag			Flags O D I T S Z A P C 0
<b>Operands</b>	<b>Clocks</b>	<b>Transfers*</b>	<b>Bytes</b>	<b>Coding Example</b>
(no operands)	2(2)	—	1	CLC

<b>CLI</b>	CLI (no operands) Clear interrupt flag			Flags O D I T S Z A P C 0
<b>Operands</b>	<b>Clocks</b>	<b>Transfers*</b>	<b>Bytes</b>	<b>Coding Example</b>
(no operands)	2(2)	—	1	CLI

<b>CMP</b>	CMP destination, source Compare destination to source			Flags O D I T S Z A P C X X X X X
<b>Operands</b>	<b>Clocks</b>	<b>Transfers*</b>	<b>Bytes</b>	<b>Coding Example</b>
register, register	3(3)	—	2	CMP BX, CX
register, memory	9(10)+EA	1	2-4	CMP DH, [ALPHA]
memory, register	9(10)+EA	1	2-4	CMP [BP+2], SI
register, immediate	4(3)+EA	—	3-4	CMP BL, 02H
memory, immediate	10(10)+EA	1	3-6	CMP RADAR [BX], [DI],3420H
accumulator, immediate	4(3-4)	—	2-3	CMP AL, 00010000B



<b>CMPS</b>	CMPS des-string, source-string Compare string			Flags O D I T S Z A P C X           X X X X X
<b>Operands</b>	<b>Clocks</b>	<b>Transfers*</b>	<b>Bytes</b>	<b>Coding Example</b>
dest-string, source-string (repeat)	22(22)	2	1	CMPS BUFF1, BUFF2
dest-string, source-string	9+22/rep (5+22/rep)	2/rep	1	REPE CMPS ID, KEY

<b>DAA</b>	DAA (no operands) Decimal adjust for addition			Flags O D I T S Z A P C U           X X X X X
<b>Operands</b>	<b>Clocks</b>	<b>Transfers*</b>	<b>Bytes</b>	<b>Coding Example</b>
(no operands)	4(4)	—	1	DAA

<b>DAS</b>	DAS (no operands) Decimal adjust for subtraction			Flags O D I T S Z A P C U           X X X X X
<b>Operands</b>	<b>Clocks</b>	<b>Transfers*</b>	<b>Bytes</b>	<b>Coding Example</b>
(no operands)	4(4)	—	1	DAS

<b>DEC</b>	DEC destination Decrement by 1			Flags O D I T S Z A P C X           X X X X
<b>Operands</b>	<b>Clocks</b>	<b>Transfers*</b>	<b>Bytes</b>	<b>Coding Example</b>
reg 16	3(3)	—	1	DEC AX
reg 8	3(3)	—	2	DEC AL
memory	15(15)+EA	2	2-4	DEC ARRAY [SI]

<b>DIV</b>	DIV source Division, unsigned			Flags O D I T S Z A P C U U U U U
<b>Operands</b>	<b>Clocks</b>	<b>Transfers*</b>	<b>Bytes</b>	<b>Coding Example</b>
reg 8	80-90(29)	—	2	DIV CL
reg 16	144-162(38)	—	2	DIV BX
mem 8	86-96+EA (35)	1	2-4	DIV [ALPHA]
mem 16	150-168+ EA(94)	1	2-4	DIV TABLE [SI] / DIV [TABLE + SI]

<b>IN</b>	IN accumulator, port Input byte or word			Flags O D I T S Z A P C
<b>Operands</b>	<b>Clocks</b>	<b>Transfers*</b>	<b>Bytes</b>	<b>Coding Example</b>
accumulator, immed 8	10(10)	1	2	IN AL, OFEH / IN AX, OFEH
accumulator, DX	8(8)	1	1	IN AL, DX / IN AX, DX

<b>INC</b>	INC destination Increment by 1			Flags O D I T S Z A P C X X X X X
<b>Operands</b>	<b>Clocks</b>	<b>Transfers*</b>	<b>Bytes</b>	<b>Coding Example</b>
reg 16	3(3)	—	1	INC CX
reg 8	3(3)	—	2	INC BL
memory 8	15(15)+EA	2	2-4	INC ALPHA [DI] [BX]

<b>INT</b>	INT interrupt-type Interrupt			Flags O D I T S Z A P C X X
<b>Operands</b>	<b>Clocks</b>	<b>Transfers*</b>	<b>Bytes</b>	<b>Coding Example</b>
immed 8 (type=3)	52(45)	5	1	INT 3
immed 8 (type≠3)	52(47)	5	2	INT 67

<b>IRET</b>	IRET (no operands) Interrupt Return			Flags O D I T S Z A P C R R R R R R R R R
<b>Operands</b>	<b>Clocks</b>	<b>Transfers*</b>	<b>Bytes</b>	<b>Coding Example</b>
(no operands)	32(28)	3	1	IRET

<b>JC</b>	JC short-label Jump if carry			Flags O D I T S Z A P C
<b>Operands</b>	<b>Clocks</b>	<b>Transfers*</b>	<b>Bytes</b>	<b>Coding Example</b>
short-label	16 or 4 (13 or 4)	—	2	JC CARRY_SET

<b>JE/JZ</b>	JE/JZ short-label Jump if equal/Jump if zero			Flags O D I T S Z A P C
<b>Operands</b>	<b>Clocks</b>	<b>Transfers*</b>	<b>Bytes</b>	<b>Coding Example</b>
short-label	16 or 4 (13 or 4)	—	2	JZ ZERO

<b>JMP</b>	JMP target Jump			Flags O D I T S Z A P C
<b>Operands</b>	<b>Clocks</b>	<b>Transfers*</b>	<b>Bytes</b>	<b>Coding Example</b>
short-label	15(13)	—	2	JMP SHORT
near-label	15(13)	—	3	JMP WITHIN_SEGMENT
far-label	15(13)	—	5	JMP FAR_LABEL
memptr 16	18(17)+EA	1	2-4	JMP [BX] TARGET
regptr 16	11(11)	—	2	JMP CX
memptr 32	24(26)+EA	2	2-4	JMP OTHER_SEG

<b>JNC</b>	JNC short-label Jump if not carry			Flags O D I T S Z A P C
<b>Operands</b>	<b>Clocks</b>	<b>Transfers*</b>	<b>Bytes</b>	<b>Coding Example</b>
short-label	16 or 4 (13 or 4)	—	2	JNC NOT_CARRY

<b>JNE/JNZ</b>	JNE/JNZ short-label Jump if not equal/Jump if not zero			Flags O D I T S Z A P C
<b>Operands</b>	<b>Clocks</b>	<b>Transfers*</b>	<b>Bytes</b>	<b>Coding Example</b>
short-label	16 or 4 (13 or 4)	—	2	JNE NOT_EQUAL

<b>LEA</b>	LEA destination, source Load effective address			Flags O D I T S Z A P C
<b>Operands</b>	<b>Clocks</b>	<b>Transfers*</b>	<b>Bytes</b>	<b>Coding Example</b>
reg 16, mem 16	2(6)+EA	—	2-4	LEA BX, [BP] [DI]

<b>LOOP</b>	LOOP short – label Decrement cx and jump if not zero			Flags O D I T S Z A P C
<b>Operands</b>	<b>Clocks</b>	<b>Transfers*</b>	<b>Bytes</b>	<b>Coding Example</b>
short label	17/5(15/5)	—	2	LOOP AGAIN

<b>MOV</b>	MOV destination, source Move			Flags O D I T S Z A P C
<b>Operands</b>	<b>Clocks</b>	<b>Transfers*</b>	<b>Bytes</b>	<b>Coding Example</b>
memory, accumulator	10(9)	1	3	MOV ARRAY [SI], AL
accumulator, memory	10(8)	1	3	MOV AX, TEMP_RESULT
register, register	2(2)	—	2	MOV AX, CX
register, memory	8(12)+EA	1	2-4	MOV BP, STACK_TOP
memory, register	9(9)+EA	1	2-4	MOV COUNT [DI], CX
register, immediate	4(3-4)	—	2-3	MOV CL, 2
memory, immediate	10(12-13) +EA	1	3-6	MOV MASK [BX] [SI], 2CH
seg-reg, reg 16	2(2)	—	2	MOV ES, CX
seg-reg, mem 16	8(9)+EA	1	2-4	MOV DS, SEGMENT_BASE
reg 16, seg-reg	2(2)	—	2	MOV BP, SS
memory, seg-reg	9(11)+EA	1	2-4	MOV [BX] SEG_SAVE, CS

<b>MOVSB/MOVSW</b>	MOVSB/MOVSW (no operands) Move string (byte/word)			Flags O D I T S Z A P C
<b>Operands</b>	<b>Clocks</b>	<b>Transfers*</b>	<b>Bytes</b>	<b>Coding Example</b>
(no operands)	18(9)	2	1	MOVSB
(repeat) (no operands)	9+17/rep (8+8/rep)	2/rep	1	REP MOVSW

<b>MUL</b>	MUL source Multiplication, unsigned			Flags O D I T S Z A P C X U U U X
<b>Operands</b>	<b>Clocks</b>	<b>Transfers*</b>	<b>Bytes</b>	<b>Coding Example</b>
reg 8	70-77 (26-28)	—	2	MUL BL
reg 16	118-133 (35-37)	—	2	MUL CX
mem 8	76-83+ EA(32-34)	1	2-4	MUL MONTH [SI]
mem 16	124-139+ EA(41-43)	1	2-4	MUL [BAUD_RATE]

<b>NOP</b>	NOP (no operands) No Operation			Flags O D I T S Z A P C
<b>Operands</b>	<b>Clocks</b>	<b>Transfers*</b>	<b>Bytes</b>	<b>Coding Example</b>
(no operands)	3(3)	—	1	NOP

<b>NOT</b>	NOT destination Logical not			Flags O D I T S Z A P C
<b>Operands</b>	<b>Clocks</b>	<b>Transfers*</b>	<b>Bytes</b>	<b>Coding Example</b>
register	3(3)	—	2	NOT AX
memory	16(3)+EA	2	2-4	NOT [CHARACTER]

<b>OR</b>	OR destination, source Logical inclusive or			Flags O D I T S Z A P C X X X U X X
<b>Operands</b>	<b>Clocks</b>	<b>Transfers*</b>	<b>Bytes</b>	<b>Coding Example</b>
register, register	3(3)	—	2	OR AL, BL
register, memory	9(10)+EA	1	2-4	OR DX, PORT_ID [DI]
memory, register	16(10)+EA	2	2-4	OR FLAG_BYTE, CL
accumulator, immediate	4(3-4)	—	2-3	OR AL, 01101100B
register, immediate	4(4)	—	3-4	OR CX, 01H
memory, immediate	17(16)+EA	2	3-6	OR [BX], CMD_WORD

<b>OUT</b>	OUT port, accumulator Output byte or word			Flags O D I T S Z A P C
<b>Operands</b>	<b>Clocks</b>	<b>Transfers*</b>	<b>Bytes</b>	<b>Coding Example</b>
immed 8, accumulator	10(9)	1	2	OUT 44, AX
DX, accumulator	8(7)	1	1	OUT DX, AL

<b>POP</b>	POP destination Pop word off stack			Flags O D I T S Z A P C
<b>Operands</b>	<b>Clocks</b>	<b>Transfers*</b>	<b>Bytes</b>	<b>Coding Example</b>
register	8(10)	1	1	POP DX
seg-reg (CS illegal)	8(8)	1	1	POP DS
memory	17(20)+EA	2	2-4	POP [PARAMETER]

<b>PUSH</b>	PUSH source Push word onto stack			Flags O D I T S Z A P C
<b>Operands</b>	<b>Clocks</b>	<b>Transfers*</b>	<b>Bytes</b>	<b>Coding Example</b>
register	11(10)	1	1	PUSH SI
seg-reg (CS legal)	10(9)	1	1	PUSH ES
memory	16(16)+EA	2	2-4	PUSH RETURN_CODE [SI]

<b>RCL</b>	RCL destination, count Rotate left through carry			Flags O D I T S Z A P C X X
<b>Operands</b>	<b>Clocks</b>	<b>Transfers*</b>	<b>Bytes</b>	<b>Coding Example</b>
register, 1	2(2)	—	2	RCL CX, 1
register, CL	8+4/bit (5+1/bit)	—	2	RCL AL, CL
memory, 1	15(15)+EA	2	2-4	RCL ALPHA, 1
memory, CL	20+4/bit (17+1/bit) +EA	2	2-4	RCL [BP].PARAM, CL

<b>RCR</b>	RCR destination, count Rotate right through carry			Flags O D I T S Z A P C X X X X X X X X
<b>Operands</b>	<b>Clocks</b>	<b>Transfers*</b>	<b>Bytes</b>	<b>Coding Example</b>
register, 1	2(2)	—	2	RCR BX, 1
register, CL	8+4/bit (5+1/bit)	—	2	RCR BL, CL
memory, 1	15(15)+EA	2	2-4	RCR [BX].STATUS, 1
memory, CL	20+4/bit (17+1/bit) +EA	2	2-4	RCR ARRAY [DI], CL

<b>REP</b>	REP (no operands) Repeat string operation			Flags O D I T S Z A P C
<b>Operands</b>	<b>Clocks</b>	<b>Transfers*</b>	<b>Bytes</b>	<b>Coding Example</b>
(no operands)	2(2)	—	1	REP MOVSB, SRCE

<b>RET</b>	RET optional-pop-value Return from procedure			Flags O D I T S Z A P C
<b>Operands</b>	<b>Clocks</b>	<b>Transfers*</b>	<b>Bytes</b>	<b>Coding Example</b>
(intra-segment, no pop)	16(16)	1	1	RET
(intra-segment, pop)	20(18)	1	3	RET 4
(inter-segment, no pop)	26(22)	2	1	RET
(inter-segment, pop)	25(25)	2	3	RET 2

<b>ROL</b>	ROL destination, count Rotate left			Flags
				O D I T S Z A P C X X X X X X X
<b>Operands</b>	<b>Clocks</b>	<b>Transfers*</b>	<b>Bytes</b>	<b>Coding Example</b>
register, 1	2(2)	—	2	ROL BX, 1
register, CL	8+4/bit (5+1/bit)	—	2	ROL DI, CL
memory, 1	15(15)+EA	2	2-4	ROL FLAG_BYTE [DI], 1
memory, CL	20+4/bit (17+1/bit) +EA	2	2-4	ROL ALPHA, CL

<b>ROR</b>	ROR destination, count Rotate right			Flags
				O D I T S Z A P C X X X X X X X
<b>Operands</b>	<b>Clocks</b>	<b>Transfers*</b>	<b>Bytes</b>	<b>Coding Example</b>
register, 1	2(2)	—	2	ROR BX, 1
register, CL	8+4/bit (5+1/bit)	—	2	ROR BX, CL
memory, 1	15(15)+EA	2	2-4	ROR PORT_STATUS, 1
memory, CL	20+4/bit (17+1/bit) +EA	2	2-4	ROR CMD_WORD, CL

<b>SBB</b>	SBB destination, source Subtract with borrow			Flags
				O D I T S Z A P C X X X X X X X
<b>Operands</b>	<b>Clocks</b>	<b>Transfers*</b>	<b>Bytes</b>	<b>Coding Example</b>
register, register	3(3)	—	2	SBB BX, CX
register, memory	9(10)+EA	1	2-4	SBB DI, [BX].PAYMENT
memory, register	16(10)+EA	2	2-4	SBB BALANCE, AX
accumulator, immediate	4(3-4)	—	2-3	SBB AX, 2
register, immediate	4(4)	—	3-4	SBB CL, 1
memory, immediate	17(16)+EA	2	3-6	SBB COUNT [SI], 10



<b>STC</b>	STC (no operands) Set carry flag			Flags O D I T S Z A P C 1
<b>Operands</b>	<b>Clocks</b>	<b>Transfers*</b>	<b>Bytes</b>	<b>Coding Example</b>
(no operands)	2(2)	—	1	STC

<b>STI</b>	STI (no operands) Set interrupt enable flag			Flags O D I T S Z A P C 1
<b>Operands</b>	<b>Clocks</b>	<b>Transfers*</b>	<b>Bytes</b>	<b>Coding Example</b>
(no operands)	2(2)	—	1	STI

<b>SUB</b>	SUB destination, source Subtraction			Flags O D I T S Z A P C X X X X X
<b>Operands</b>	<b>Clocks</b>	<b>Transfers*</b>	<b>Bytes</b>	<b>Coding Example</b>
register, register	3(3)	—	2	SUB CX, BX
register, memory	9(10)+EA	1	2-4	SUB DX, MATH_TOTAL [SI]
memory, register	16(10)+EA	2	2-4	SUB [BP+2], CL
accumulator, immediate	4(3-4)	—	2-3	SUB AL, 10
register, immediate	4(4)	—	3-4	SUB SI, 5280

<b>TEST</b>	TEST destination, source Non-destructive logical and			Flags O D I T S Z A P C X X X U X X
<b>Operands</b>	<b>Clocks</b>	<b>Transfers*</b>	<b>Bytes</b>	<b>Coding Example</b>
register, register	3(3)	—	2	TEST SI, DI
register, memory	9(10)+EA	1	2-4	TEST SI, END_COUNT
accumulator, immediate	4(3-4)	—	2-3	TEST AL, 00100000B
register, immediate	5(4)	—	3-4	TEST BX, 0CC4H
memory, immediate	11(10)+EA	—	3-6	TEST [RETURN_COUNT], 01H

<b>XCHG</b>	XCHG destination, source Exchange registers			Flags O D I T S Z A P C
<b>Operands</b>	<b>Clocks</b>	<b>Transfers*</b>	<b>Bytes</b>	<b>Coding Example</b>
accumulator, reg 16	3(3)	—	1	XCHG AX, BX
memory, register	17(17)+EA	2	2-4	XCHG SEMAPHORE, AX
register, register	4(4)	—	2	XCHG AL, BL

<b>XLAT</b>	XLAT source-table Translate			Flags O D I T S Z A P C
<b>Operands</b>	<b>Clocks</b>	<b>Transfers*</b>	<b>Bytes</b>	<b>Coding Example</b>
source-table	11(11)	1	1	XLAT ASCII_TAB

<b>XOR</b>	XOR destination, source Logical exclusive or			Flags O D I T S Z A P C 0 X X U X X
<b>Operands</b>	<b>Clocks</b>	<b>Transfers*</b>	<b>Bytes</b>	<b>Coding Example</b>
register, register	3(3)	—	2	XOR CX, BX
register, memory	9(10)+EA	1	2-4	XOR CL, MASK_BYTE
memory, register	16(10)+EA	2	2-4	XOR ALPHA [SI], DX
accumulator, immediate	4(3-4)	—	2-3	XOR AL, 01000010B
register, immediate	4(4)	—	3-4	XOR SI, 00C2H
memory, immediate	17(16)+EA	2	3-6	XOR RETURN_CODE, 0D2H

**בהצלחה!**