

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, ובהם תשע שאלות. יש לענות על חמש שאלות, שאלה אחת לפחות מכל פרק. לכל שאלה – 20 נקודות. סך הכול – 100 נקודות.
 - ג. חומר עזר מותר לשימוש: נוסחאון בית-ספרי מודפס לשפת C או VB, ומחשבון.
 - 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. 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, ולצידה יחידות המידה המתאימות.
 - * ליווי הפתרון החישובי בהסבר קצר.
- בשאלון זה 10 עמודים ו-16 עמודי נוסחאון.

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

בהצלחה!

השאלות

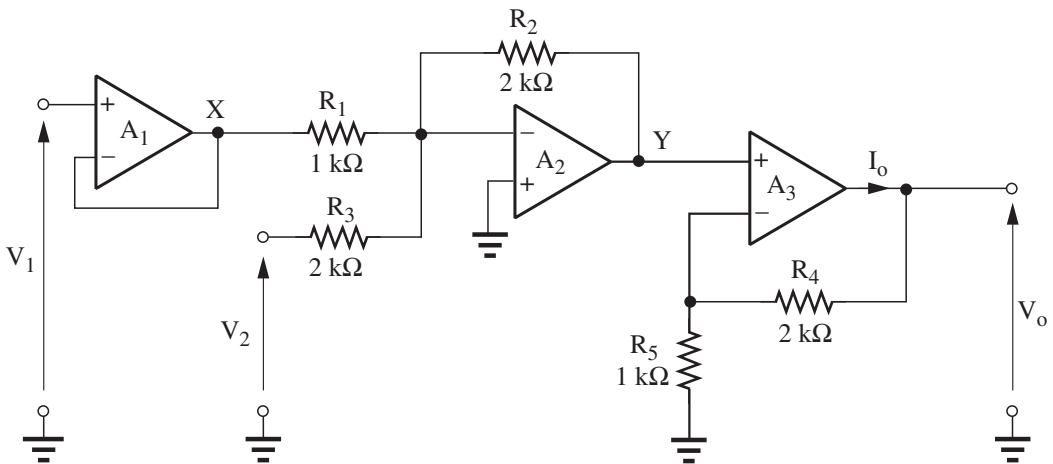
בשאלון זה שני פרקים ובהם תשע שאלות. יש לענות על חמש שאלות בלבד, שאלה אחת לפחות מכל פרק.

פרק ראשון: מבוא להנדסת אלקטרוניקה

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

שאלה 1

המעגל החשמלי המתואר באיור לשאלה 1 כולל מגברי שרת אידיאליים.



איור לשאלה 1

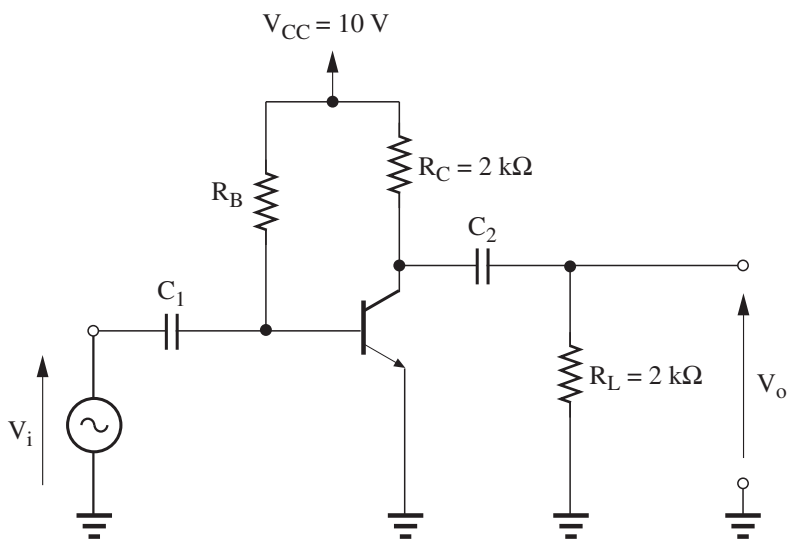
- א. הסבר את תפקידו של המגבר A_1 ומנה את תכונותיו.
- ב. עבור $V_1 = -1\text{ V}$, $V_2 = -2\text{ V}$, חשב את המתח בנקודות X ו-Y ואת מתח המוצא, V_0 .
- ג. חשב את הזרם I_0 .

שאלה 2

באיור לשאלה 2 מתואר מגבר טרנזיסטורי.

נתוני הטרנזיסטור: $h_{ie} = 1 \text{ k}\Omega$, $V_{BE} = 0.7 \text{ V}$, $\beta = 100$, $h_{fe} = 100$

היגבי הקבלים במעגל זניחים.

**איור לשאלה 2**

א. חשב את ערך ההתנגדות של הנגד R_B הדרוש לקבלת $V_{CE} = 5 \text{ V}$.

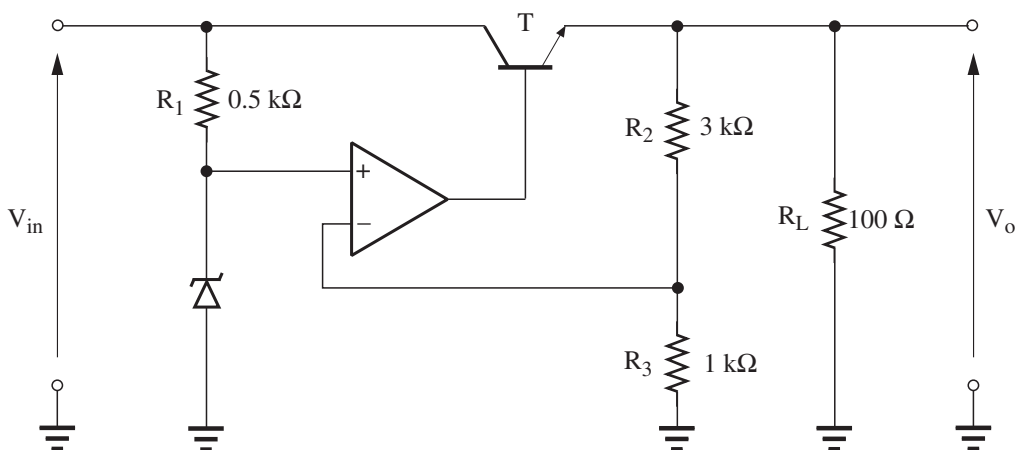
ב. סרטט מעגל תמורה לאות חילופין של המעגל הנתון.

ג. חשב את הגבר המתח $\frac{V_o}{V_i}$.

שאלה 3

המעגל המתואר באיור לשאלה 3 משמש כמייצב מתח. מגבר השרת שבמעגל הוא אידיאלי.

נתונים: $V_z = 2\text{ V}$; $V_{in} = 12\text{ V}$



איור לשאלה 3

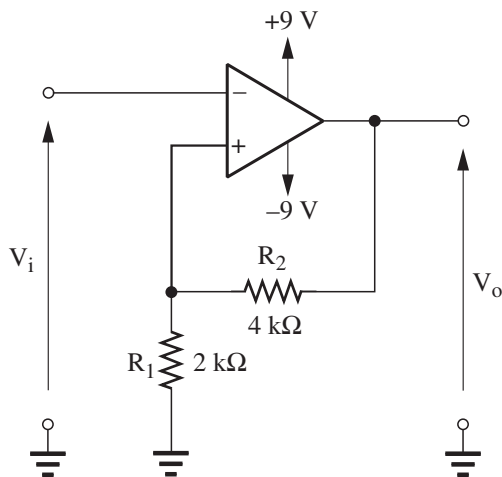
א. מה תפקידו של הטרנזיסטור T במעגל?

ב. חשב את הזרם דרך דיודת הזנר.

ג. חשב את מתח המוצא V_o .

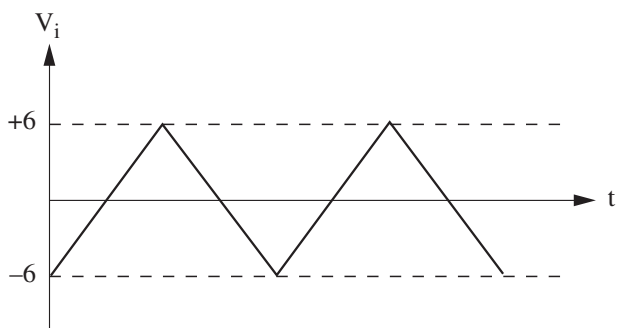
שאלה 4

המעגל החשמלי המתואר באיור א' לשאלה 4 משמש כמעגל שמיט. מתחי הרוויה של מגבר השרת הם $\pm 9\text{ V}$.



איור א' לשאלה 4

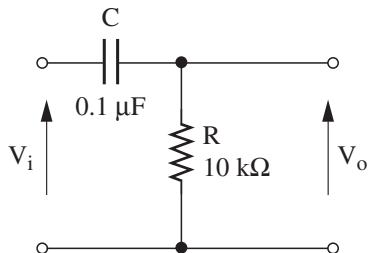
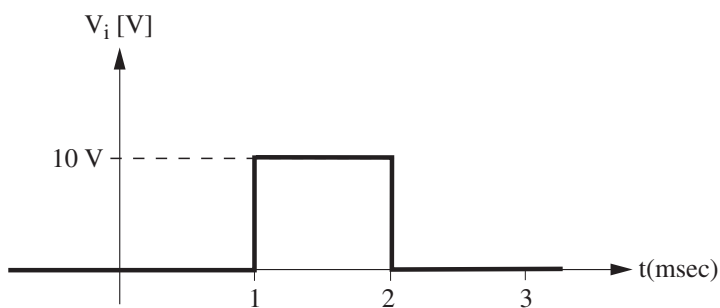
- א. חשב את מתחי הסף של מעגל השמיט.
- ב. סרטט את אופיין המעבר $V_o = f(V_i)$
- ג. באיור ב' לשאלה מתואר גל מתח משולש המסופק למבוא המעגל V_i . העתק למחברתך את מתח המבוא V_i וסרטט מתחתיו, בהתאמה, את מתח המוצא V_o כתלות בזמן.



איור ב' לשאלה 4

שאלה 5

לרשת המתוארת באיור א' לשאלה 5 מספקים את הדופק המתואר באיור ב' לשאלה.

**איור א' לשאלה 5****איור ב' לשאלה 5**

א. העתק למחברתך את מתח המבוא V_i וסרטט מתחתיו, בהתאמה, את מתח המוצא V_o כתלות בזמן.

ב. חשב את מתח המוצא ברגע $t = 3 \text{ msec}$.

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. SHIGRA:  MOV    SI, 10H
2.           MOV    AL, 0H
3.           MOV    CH, 0H
4.           MOV    CL, 4H
5.           MOV    BH, [SI]
6.           MOV    BL, BH
7.           AND    BL, 0F0H
8.           ROR    BL, CL
9.           AND    BH, 0FH
10. ABC:     ADD    AL, BL
11.          DEC    BH
12.          JNZ    ABC
13.          MOV    [SI+1], AL
14.          RET
```

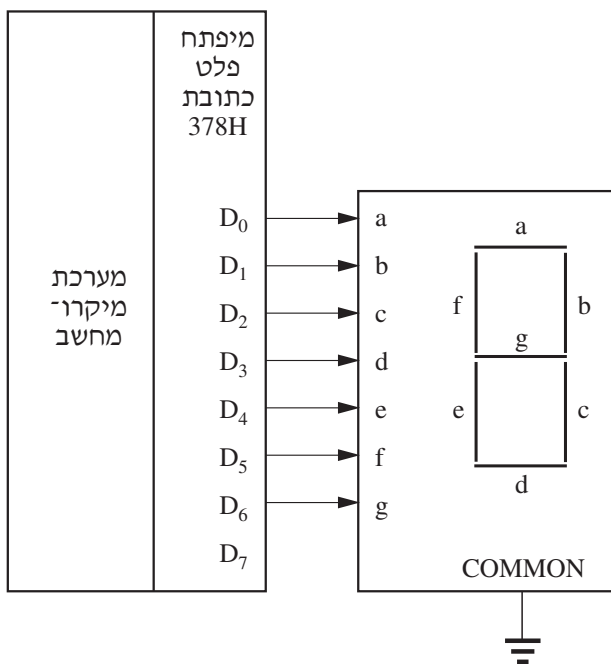
א. הסבר את ההוראות שמספריהן הם: 5, 7, 8, 12.

ב. הסבר מה מבצעת תת-השגרה הנתונה.

ג. הנח כי לפני ביצוע תת-השגרה תוכן תא הזיכרון שכתובתו 10 H הוא 42 H. מה יהיה ערכו של תא הזיכרון שכתובתו 11 H לאחר ביצוע תת-השגרה?

שאלה 7

באיור לשאלה 7 מתוארת מערכת מיקרו-מחשב המבוססת על המיקרו-מעבד 8086/88. המערכת מיועדת להצגת ספרות מ-0 עד 9 כמתואר באיור.

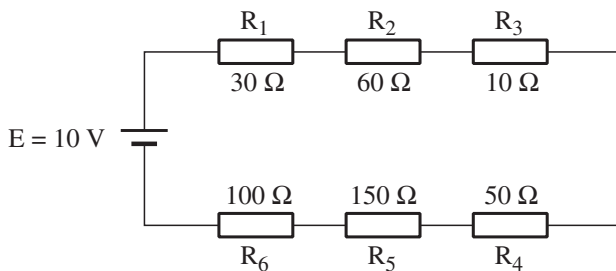


איור לשאלה 7

- א. רשום את הערכים הנדרשים במוצא המיפתח כדי להציג את הספרות 1 ו-0.
- ב. כתוב תת-שגרה הבודקת את ערכו של הנתון שבתא H 10 ופועלת בהתאם:
1. אם ערכו זוגי, תוצג הספרה 0.
 2. אם ערכו אי-זוגי, תוצג הספרה 1.

שאלה 8

באיור לשאלה 8 מתואר מעגל חשמלי.



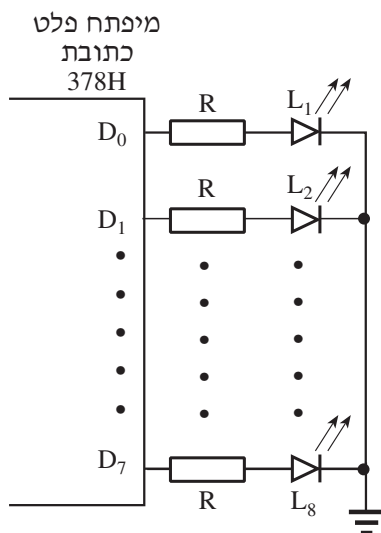
איור לשאלה 8

כתוב תכנית בשפת C או VB שתבצע את הפעולות הבאות:

- תגדיר מערך ותציב את ערכי ההתנגדויות של הנגדים במערך.
 - תדפיס את ערכי ההתנגדויות של הנגדים ואת ערך ההתנגדות השקולה.
 - תחשב את מתחי הנגדים ותדפיס את התוצאות.
- הערה:** יש לכתוב את התכנית תוך שימוש בלולאות.

שאלה 9

באיור לשאלה 9 מופיע מיפתח פלט שמחוברות אליו נוריות LED.



איור לשאלה 9

כתוב תכנית בשפת C או VB שתבצע את הפעולות הבאות:

- א. תגדיר משתנה שלם N .
- ב. תקלוט ערך למשתנה N .
- ג. אם $N \leq 0$ או $N \geq 9$, התכנית תסתיים.
- ד. אם $0 < N < 9$, תידלק הנורית המתאימה למשך N שניות (לדוגמה, עבור $N = 4$, תידלק L_4 למשך 4 שניות), ולאחר מכן התכנית תסתיים.

בהצלחה!

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

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 [Ω]

התנגדות מבוא — R_i [Ω]

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

הגבר הספק בדציבלים — A_P [dB]

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

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

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

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

הגבר כולל בדציבלים של
N דרגות המחוברות בשרשרת
(קסקדה) — $A_{VT} \text{ [dB]}$

מאזן הספקים

הספק מבוא — $P_I \text{ [W]}$
הספק נצרך מהספקים — $P_{CC} \text{ [W]}$
הספק העומס — $P_L \text{ [W]}$
הספק מבוזבז — $P_{diss} \text{ [W]}$

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

משוב שלילי

הגבר עם משוב (בחוג סגור) — $A_f \text{ [A]}$
הגבר ללא משוב — A
(הגבר חוג פתוח)
מקדם משוב — β

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

משוב מתח טורי

התנגדות מבוא עם משוב — $R_{if} \text{ } [\Omega]$
התנגדות מבוא ללא משוב — $R_i \text{ } [\Omega]$
התנגדות מוצא עם משוב — $R_{of} \text{ } [\Omega]$
התנגדות מוצא ללא משוב — $R_o \text{ } [\Omega]$

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

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

טרנזיסטור דו-נושאי (בתחום הפעיל)

בהזנחת זרם הזליגה I_{CBO} :

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

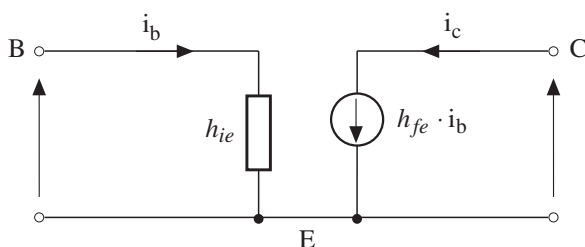
זרם הקולט — I_C [A]

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	∞	∞

מגברי שרת

מגבר מהפך

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

$$\begin{array}{ll} \text{נגד המשוב} & - R_f \quad [\Omega] \\ \text{הנגד המחובר לכניסה המהפכת} & - R_l \quad [\Omega] \end{array}$$

מגבר עוקב

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

$$\begin{array}{ll} \text{נגד המשוב} & - R_f \quad [\Omega] \\ \text{הנגד היוצא מהכניסה המהפכת} & - R_l \quad [\Omega] \\ \text{לאדמה} & \end{array}$$

2. נוסחאות באלקטרוניקה ספרתית

$$\begin{array}{ll} \text{הערך הרגעי של המתח} & - v(t) \quad [V] \\ \text{הערך שאליו המתח שואף} & - V_\infty \quad [V] \\ \text{להגיע כאשר } t \rightarrow \infty & \\ \text{הערך ההתחלתי של המתח} & - V_{0+} \quad [V] \end{array}$$

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

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

$$\text{קבוע זמן} \quad - \tau \quad [\text{sec}]$$

$$\tau = RC$$

$$\text{התנגדות} \quad - R \quad [\Omega]$$

$$\text{קיבול} \quad - C \quad [F]$$

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

$$\text{תדר חצי הספק עליון של רשת} \\ \text{מעבירת נמוכים} \quad - f_H \quad [Hz]$$

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

$$\text{תדר חצי הספק תחתון של רשת} \\ \text{מעבירת גבוהים} \quad - f_L \quad [Hz]$$

$$\text{זמן עלייה של רשת} \\ \text{מעבירת נמוכים} \quad - t_r \quad [\text{sec}]$$

$$t_r = 2.2\tau$$

3. אוסף פקודות למיקרו-מעבדים 8086/88

מקרא לאוגר הדגלים:

0 — מתאפס (Modified) X — מושפע מהפעולה

1 — מקבל '1' (Undefined) U — לא מוגדר אחרי הפעולה

R — מוחזר מהמחשנית (Returned)

ADC	ADC destination, source Add with carry			Flags
				O D I T S Z A P C X 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 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

CALL	CALL target Call a procedure			Flags O D I T S Z A P C
Operands	Clocks	Transfers*	Bytes	Coding Example
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]

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

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

CMP	CMP destination, source Compare destination to source			Flags O D I T S Z A P C X X X X X X
Operands	Clocks	Transfers*	Bytes	Coding Example
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

CMPS	CMPS des-string, source-string Compare string			Flags O D I T S Z A P C X X X X X
Operands	Clocks	Transfers*	Bytes	Coding Example
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

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

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

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

DIV	DIV source Division, unsigned			Flags O D I T S Z A P C U U U U U
Operands	Clocks	Transfers*	Bytes	Coding Example
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]

IN	IN accumulator, port Input byte or word			Flags O D I T S Z A P C
Operands	Clocks	Transfers*	Bytes	Coding Example
accumulator, immed 8	10(10)	1	2	IN AL, 0FEH / IN AX, OFEH
accumulator, DX	8(8)	1	1	IN AL, DX / IN AX, DX

INC	INC destination Increment by 1			Flags O D I T S Z A P C X X X X X
Operands	Clocks	Transfers*	Bytes	Coding Example
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]

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

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

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

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

JMP	JMP target Jump			Flags O D I T S Z A P C
Operands	Clocks	Transfers*	Bytes	Coding Example
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

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

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

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

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

MOV	MOV destination, source Move			Flags O D I T S Z A P C
Operands	Clocks	Transfers*	Bytes	Coding Example
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

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

MUL	MUL source Multiplication, unsigned			Flags O D I T S Z A P C X U U U X
Operands	Clocks	Transfers*	Bytes	Coding Example
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]

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

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

OR	OR destination, source Logical inclusive or			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	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

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

POP	POP destination Pop word off stack			Flags O D I T S Z A P C
Operands	Clocks	Transfers*	Bytes	Coding Example
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]

PUSH	PUSH source Push word onto stack			Flags O D I T S Z A P C
Operands	Clocks	Transfers*	Bytes	Coding Example
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]

RCL	RCL destination, count Rotate left through carry			Flags O D I T S Z A P C X X
Operands	Clocks	Transfers*	Bytes	Coding Example
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

RCR	RCR destination, count Rotate right through carry			Flags O D I T S Z A P C X
Operands	Clocks	Transfers*	Bytes	Coding Example
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

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

RET	RET optional-pop-value Return from procedure			Flags O D I T S Z A P C
Operands	Clocks	Transfers*	Bytes	Coding Example
(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

ROL	ROL destination, count Rotate left			Flags O D I T S Z A P C X X X X X X X
Operands	Clocks	Transfers*	Bytes	Coding Example
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

ROR	ROR destination, count Rotate right			Flags O D I T S Z A P C X X X X X X X
Operands	Clocks	Transfers*	Bytes	Coding Example
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

SBB	SBB destination, source Subtract with borrow			Flags O D I T S Z A P C X X X X X X X
Operands	Clocks	Transfers*	Bytes	Coding Example
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

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

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

SUB	SUB destination, source Subtraction			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	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

TEST	TEST destination, source Non-destructive 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	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

XCHG	XCHG destination, source Exchange registers			Flags O D I T S Z A P C
Operands	Clocks	Transfers*	Bytes	Coding Example
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

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

XOR	XOR destination, source Logical exclusive or			Flags O D I T S Z A P C 0 X X U X X
Operands	Clocks	Transfers*	Bytes	Coding Example
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!