

אלקטרוניקה ומחשבים – מוגבר

על-פי תכנית הרפורמה ללמידה משמעותית

הוראות לנבחן

- א. משך הבחינה: שעתיים.
- ב. מבנה השאלון ומפתח ההערכה: בשאלון זה חמש שאלות בשני פרקים. יש לענות על שלוש שאלות, שאלה אחת לפחות מכל פרק.
כל שאלה – $33\frac{1}{3}$ 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 התשובות הנדרש בלבד, לפי סדר כתיבתן במחברתך, ולא יתייחס לתשובות נוספות.
 2. התחל כל תשובה לשאלה 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, אך עליך להסביר מדוע הוספת אותם.
 7. בכתיבת פתרונות חישוביים, קבלת מֶרֶב הנקודות מותנית בהשלמת כל המהלכים שלהלן, בסדר שהם רשומים בו:
 - * רישום הנוסחה המתאימה.
 - * הצבה של כל הערכים ביחידות המתאימות.
 - * חישוב (אפשר באמצעות מחשבון).
 - * רישום התוצאה המתקבלת, ולצדה יחידות המידה המתאימות.
 - * ליווי הפתרון החישובי בהסבר קצר.

כתוב במחברת הבחינה בלבד, בעמודים נפרדים, כל מה שברצונך לכתוב כטייטה (ראשי פרקים, חישובים וכדומה).
רשום "טייטה" בראש כל עמוד טייטה. רישום טייטות כלשהן על דפים שמחוץ למחברת הבחינה עלול לגרום לפסילת הבחינה!

בשאלון זה 6 עמודים ו-12 עמודי נוסחאון.

ההנחיות בשאלון זה מנוסחות בלשון זכר, אך מכוונות הן לנבחנות והן לנבחנים כאחד.

השאלות

בשאלון זה חמש שאלות בשני פרקים. יש לענות על שלוש שאלות, שאלה אחת לפחות מכל פרק.

פרק ראשון: מערכות ספרתיות

ענה על שאלה אחת לפחות מבין השאלות 1–3 (לכל שאלה – $\frac{1}{3}$ נקודות).

שאלה 1

נתונה הפונקציה: $F(A, B, C, D) = \sum (0, 2, 8, 12, 15) + \sum_{\phi} (10, 11, 13, 14)$

- רשום את טבלת האמת של הפונקציה $F(A, B, C, D)$.
- הצג את הפונקציה $F(A, B, C, D)$ באמצעות מפת קרנו.
- בטא את הפונקציה F כסכום של מכפלות במינימום ליטרלים.
- ממש את הפונקציה F המצומצמת באמצעות שערים לוגיים.

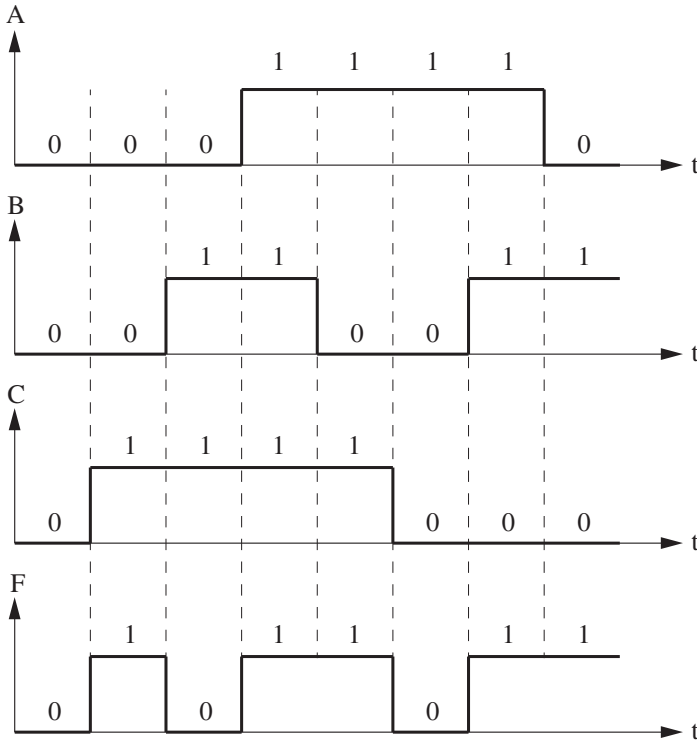
שאלה 2

באיור א' לשאלה 2 נתונה מערכת צירופים בעלת שלושה מבואות A, B ו- C ומוצא F .



איור א' לשאלה 2

באיור ב' לשאלה מתוארים האותות במבואות A, B ו-C והאות המתקבל במוצא F, כפונקציה של הזמן.



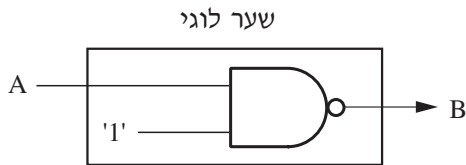
איור ב' לשאלה 2

- א. היעזר באותות המתוארים באיור ב', ורשום את טבלת-האמת של הפונקציה $F(A,B,C)$.
- ב. צמצם את הפונקציה F באמצעות מפת קרנו, ובטא אותה כסכום של מכפלות במינימום ליטרלים.
- ג. ממש את הפונקציה F באמצעות שערים לוגיים.

שאלה 3

א. נתונה הפונקציה: $F(A,B,C) = B(\bar{B} + A) + \bar{B}C + \bar{A} + \bar{C} + ABC$.
פשט את הפונקציה $F(A,B,C)$ באמצעות חוקי האלגברה הבוליאנית.

ב. נתונה הפונקציה: $F(X,Y,Z) = \bar{Z} + \bar{Y} + ZY\bar{X}$.
1. פשט את הפונקציה $F(X,Y,Z)$ באמצעות חוקי האלגברה הבוליאנית.
2. באיור לשאלה 3 מתואר שער לוגי.



איור לשאלה 3

ממש את הפונקציה המפושטת $F(X,Y,Z)$ באמצעות שערים לוגיים מהסוג הזה ושערי OR בלבד.

פרק שני: מבוא להנדסת מחשבים

ענה על שאלה אחת לפחות מבין השאלות 4-5 (לכל שאלה – $\frac{1}{3}$ נקודות).

שאלה 4

להלן תת־שגרה הכתובה בשפת־הסף של המיקרו־מעבד 8086/88.

```
1.  MAIN:  MOV  SI, 200H
2.      MOV  CL, 0
3.  NEXT:  MOV  AL, [SI]
4.      CMP  AL, [SI+1]
5.      JNE  NOTE
6.      CMP  AL, [SI+2]
7.      JNE  NOTE
8.      INC  CL
9.  NOTE:  ADD  SI, 3
10.     CMP  SI, 209H
11.     JNE  NEXT
12.     MOV  AL, CL
13.     MOV  [209H], AL
14.     RET
```

בטבלה שלהלן נתונים תוכני התאים $200H + 209H$ לפני ביצוע התת־שגרה.

200H	201H	202H	203H	204H	205H	206H	207H	208H	209H	כתובת התא
AAH	AAH	AAH	80H	80H	8H	A0H	8H	A0H	0H	תוכן התא

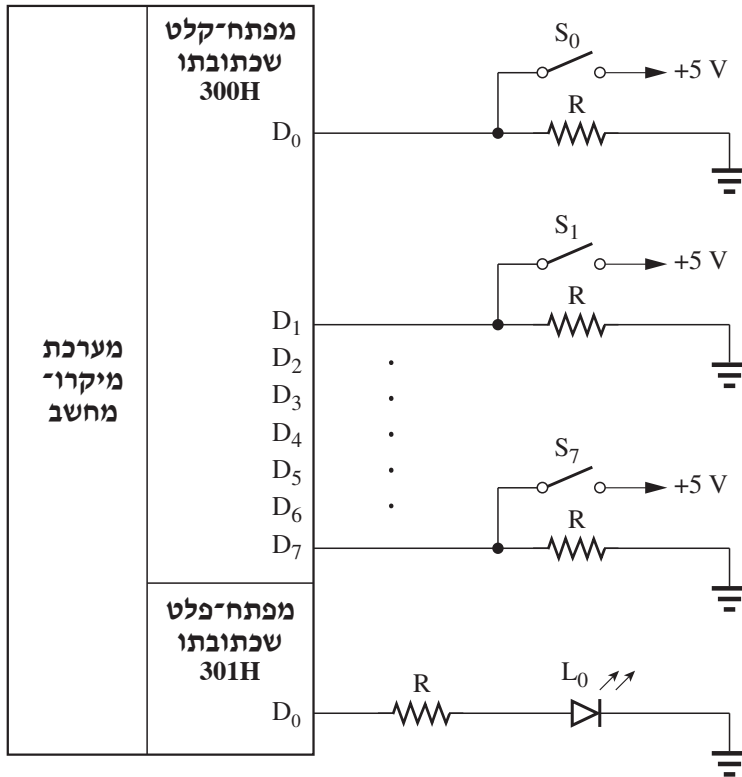
א. הסבר את ההוראות שבשורות 1, 3, 4 ו־5.

ב. רשום את תוכן התא $209H$ לאחר ביצוע התת־שגרה. היעזר בטבלת מעקב אחר תוכני האוגרים AL, CL, ו־SI.

ג. קבע ערכים אפשריים לתוכני התאים $200H + 208H$, כך שלאחר ביצוע התת־שגרה תוכן התא $209H$ יישאר 0H.

שאלה 5

- באיור לשאלה 5 נתון תרשים של מפתח-קלט שכתובתו 300H ומפתח-פלט שכתובתו 301H. להדק D_0 של מפתח-הפלט מחוברת נורית LED, ולהדקים $D_0 \div D_7$ של מפתח-הקלט מחוברים המפסקים $S_0 \div S_7$, בהתאמה.



איור לשאלה 5

- כתוב תת-שגרה בשפת-הסף של המיקרו-מעבד 8086/88, שתבצע את הפעולות שלהלן:
1. תקלוט את מצב שמונת המפסקים $S_0 \div S_7$ לתוך אוגר.
 2. תבדוק אם מצב המפסקים $S_0 \div S_3$ שווה, בהתאמה, למצב המפסקים $S_4 \div S_7$.
אם כן – היא תדליק את נורית ה-LED. אם לא – היא תכבה את נורית ה-LED.

בהצלחה!

נוסחאון במיקרו-מעבד 8086/88 (12 עמודים)

מקרא לאוגר הדגלים:

- X – מושפע מהפעולה (Modified)
 U – לא מוגדר אחרי הפעולה (Undefined)
 R – מוחזר מהמחשנית (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, 0101000B

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 X
Operands	Clocks	Transfers*	Bytes	Coding Example
dest-string,	22(22)	2	1	CMPS BUFF1, BUFF2
source-string (repeat)				
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 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 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 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 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, 0FEH
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 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)	1	3-6	MOV MASK [BX] [SI], 2CH
	+EA			
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/MOVSX	MOVSB/MOVSX (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 MOVSX

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 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, 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
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
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
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