

## שאלה 5

להלן תת-שגרה הכתובה בשפת-הסף של המיקרו-בקר 8051 :

```

1.  START:  MOV    DPTR, #1000H
2.          MOV    R7, #0AH
3.          MOVX   A, @DPTR
4.          MOV    40H, A
5.          MOV    41H, #01H
6.  AGAIN:  INC    DPTR
7.          MOVX   A, @DPTR
8.          CJNE  A, 40H, NOTE
9.          INC    41H
10. NOTE:   DJNZ  R7, AGAIN
11.          RET

```

א. הסבר את ההוראות שבשורות 4, 5, 8 ו-9.

ב. בטבלה שלהלן נתונים תכני התאים שכתובותיהם  $1000H \div 100AH$  בזיכרון החיצוני של המיקרו-בקר, לפני ביצוע התת-שגרה.

100AH	1009H	1008H	1007H	1006H	1005H	1004H	1003H	1002H	1001H	1000H	כתובת התא
A3H	FFH	A3H	A4H	FFH	A5H	A3H	A6H	A3H	A3H	A4H	תוכן התא

מה יהיו תכני התאים, שכתובותיהם בזיכרון הפנימי של המיקרו-בקר הן  $40H$  ו- $41H$ , לאחר ביצוע התת-שגרה? נמק את תשובתך במילים או הצג טבלת-מעקב אחר כל האוגרים הלוקחים חלק בתת-השגרה.

ג. מה צריכים להיות תכני התאים שכתובותיהם  $1000H \div 100AH$  בזיכרון החיצוני של המיקרו-בקר, כדי שתוכן התאים  $40H$  ו- $41H$  בזיכרון הפנימי של המיקרו-בקר יהיה  $0BH$  לאחר ביצוע התת-שגרה? נמק את תשובתך.

## תשובה 5

א.

Mov 40h,a

העבר את הנתון שבאקומולטור אל כתובת 40h ב RAM הנתונים הפנימי.

Mov 41h,#01h

העבר מידית לכתובת 41h ב RAM הנתונים הפנימי את הנתון 1 הקסה (00000001 בינארי).

Cjne a,40h,note

בצע השוואה בין הנתון באקומולטור והנתון שבכתובת 40h ב RAM הנתונים הפנימי. אם הם לא שווים קפוץ לתווית note. אם הם שווים ממשיכים לפקודה הבאה שבשורה 9.

Inc 41h

הגדל ב 1 את ערך הנתון שבכתובת 41h ב RAM הנתונים הפנימי.

ב.

בתחילת התוכנית מעבירים את הנתון שבכתובת 1000h, בזיכרון נתונים חיצוני, אל האקומולטור. מהאקומולטור מעבירים את הנתון לתא מספר 40h ב RAM הנתונים הפנימי. הנתון הוא A4H. בהמשך התוכנית לא משנים את הנתון שבכתובת זו ולכן גם בסיום תת השגרה הנתון יהיה A4H. לאחר מכן שמים בכתובת 41H את הנתון 1.

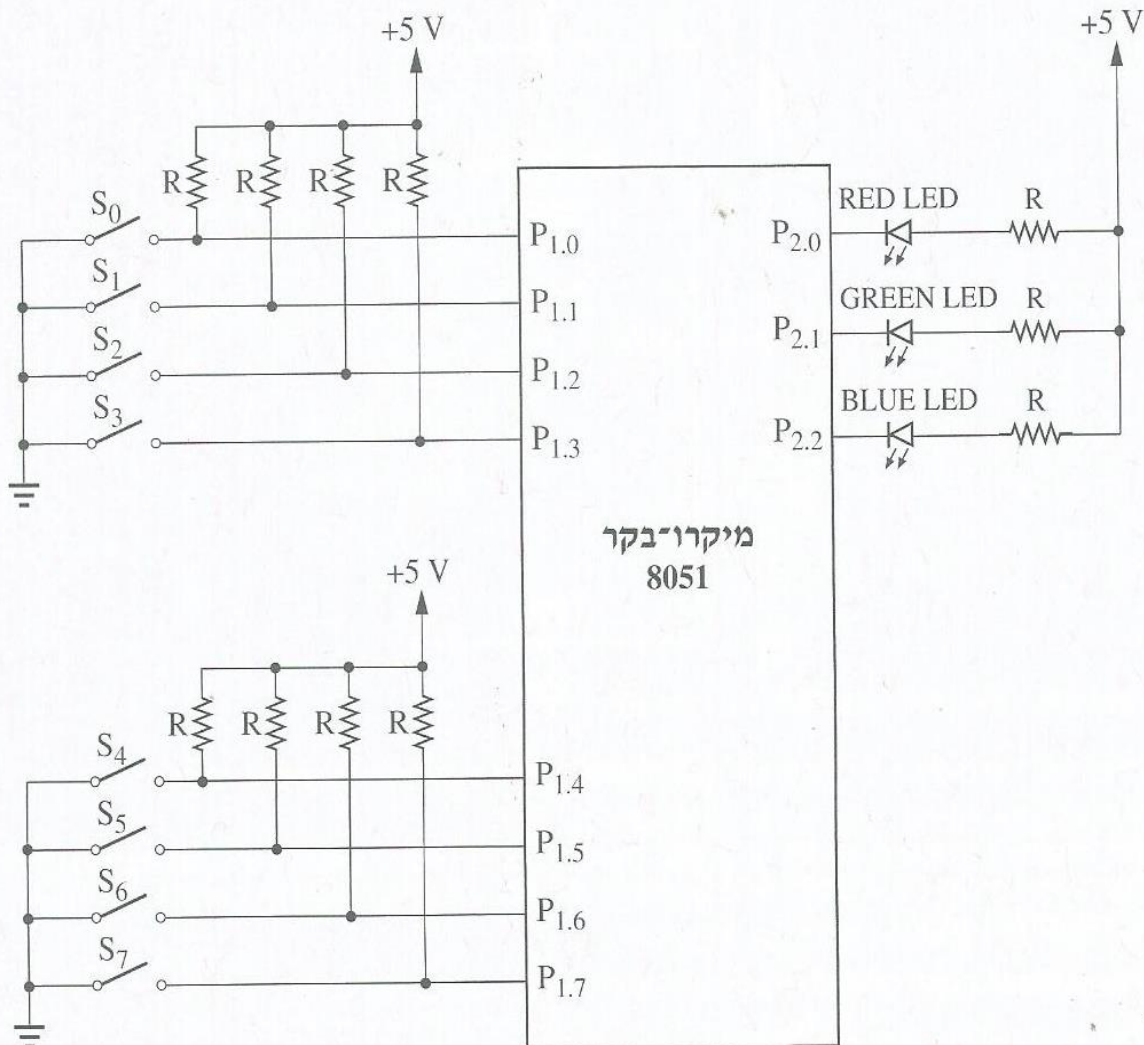
בפקודות הבאות (שורות 6 עד 10) מבצעים לולאה המתבצעת 0AH (10 פעמים). בלולאה מביאים את הנתון שבכתובת הבאה (הנתון בכתובת 1001H הוא A3H) ומשווים אותו לנתון ששמרנו בכתובת 40H (A4H). אם הם שווים מגדילים ב 1 את הערך שבכתובת 41H ואם הם לא שווים (כמו כאן שבכתובת 40H יש A4H ובאקומולטור יש A3) קופצים לתווית note מחסרים 1 מרגיסטר R7 והיות והוא לא אפס חוזרים לתווית again (שורה 6).

התוכנית בודקת כמה פעמים מופיע הנתון שבכתובת 1000H בזיכרון החיצוני בבלוק הכתובת מ 1001H עד 100AH. בדוגמה כאן הנתון A4H מופיע 2 פעמים בלבד ולכן בכתובת 41H יהיה הערך 2.

ג. כדי שגם תוכן 40H וגם תוכן 41H יהיה 0BH צריך שכל הכתובות בזיכרון החיצוני מ 1000H ועד 100AH יהיה את הנתון 0BH.

## שאלה 6

באיור לשאלה 6 נתון מיקרו־בקר 8051. ההדקים  $P_{1.0} \div P_{1.3}$  שלו חוברו, בהתאמה, למפסקים  $S_0 \div S_3$ , וההדקים  $P_{1.4} \div P_{1.7}$  שלו חוברו, בהתאמה, למפסקים  $S_4 \div S_7$ . כמו־כן חוברו נורית LED אדומה להדק  $P_{2.0}$ , נורית LED ירוקה להדק  $P_{2.1}$ , ונורית LED כחולה להדק  $P_{2.2}$ .



### איור לשאלה 6

כתוב תכנית בשפת C של המיקרו־בקר 8051, שתבצע את הפעולות שלהלן:

1. תקלוט את מצב המפסקים  $S_0 \div S_7$ .
2. אם כל המפסקים פתוחים – היא תדליק את נורית ה־LED הכחולה בלבד.
3. אם לפחות אחד מהמפסקים סגור – התכנית תבדוק אם מצב המפסקים  $S_0 \div S_3$  זהה למצב המפסקים  $S_4 \div S_7$ , בהתאמה. אם הוא זהה – התכנית תדליק את נורית ה־LED הירוקה בלבד. אם הוא אינו זהה – התכנית תדליק את נורית ה־LED האדומה בלבד.

## תשובה 6

.1

בהנחה שהוגדרו קבצי הכותר המתאימים :

```
unsigned char switches;
```

```
switches = P1;
```

.2 + .3 נפתור את שני הסעיפים יחד

נרשום פונקציה :

```
void leds()
{
P2_0=1;          // כיבוי לד אדומה
P2_1=1;          // כיבוי לד ירוקה
P2_2=1;          // כיבוי הLED הכחולה

If(switches == 0xff)
    P2_2=0;      // הדלקת LED כחולה
else if(switches != 0xff) // האם לא כל המפסקים פתוחים ( לפחות אחד לחוץ )
    (ומכניס 0)
    if(switches/16 == switches%16) // בדיקה האם 4 הביטים הגבוהים שווים ל 4 הנמוכים
        P2_1=0; // הדלקת LED ירוקה
    else // 4 הביטים הגבוהים לא שווים ל 4 הביטים הנמוכים
        P2_0=0; // הדלקת הLED האדומה
}
```

```

2. {
3.     CH='A';
4.     I=0;
5.     TMOD=0x20;
6.     SCON=0x40;
7.     TH1=0xFD;
8.     TL1=0xFD;
9.     TR1=1;
10.    TI=0;
11.    IE=0x81;
12.    IT0=1;
13.    while(1)
14.    {
15.        SBUF=CH;
16.        while(!TI);
17.        TI=0;
18.    }
19. }
    
```

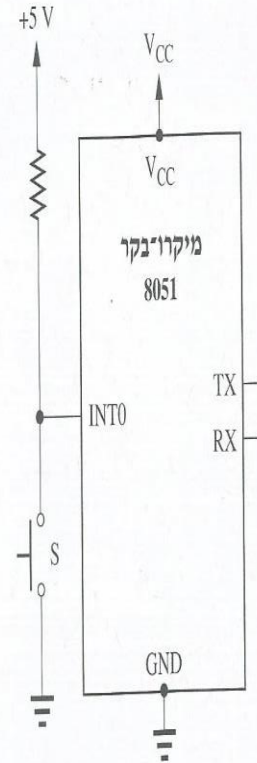
א. הסבר את ההוראות שבשורות 1, 3, 6 ו-26.

ב. הסבר את המשמעות של השורות 22 ÷ 15 בקוד התכנית. היעזר בדף הנוסחאות של המיקרו־בקר לצורך ניתוח מילות הבקרה.

ג. העתק למחברתך את הטבלה שלהלן, השלם בה את הנתונים שישודרו במוצא ה-UART בכל אחד מן המצבים, ונמק את תשובותיך. התעלם מבעיית הריטוטים העשויים להתרחש בעת לחיצה על הלחצן S.

נימוק	הנתונים שישודרו במוצא ה-UART
	לפני הלחיצה הראשונה
	לאחר הלחיצה הראשונה
	לאחר 2 לחיצות
	לאחר 3 לחיצות
	לאחר 4 לחיצות
	לאחר 5 לחיצות
	לאחר 6 לחיצות
	לאחר 7 לחיצות

איור לשאלה 7 נתון לחצן S המחובר להדק המבוא INTO במיקרו־בקר 8051. מצבו הרגיל של הלחצן הוא פתוח (N.O.).



איור לשאלה 7

ז"ל תכנית בשפת C של המיקרו־בקר 8051, העושה שימוש ב-UART ובפסיקות חומרה חיצוניות:

```

1. #include <8051.H>
2. unsigned char CH,I;
3. void int0() interrupt0
4. {
5.     I++;
6.     if(I%3==0)
7.         CH = 'B';
8.     else
9.         CH = 'C';
10. }
11. void main(void)
    
```

תשובה 7

.א.

#include &lt;8051.h&gt;

הנחיה לקומפיילר לכלול קובץ כותר בשם 8051.h . בקובץ הזה יש הצהרות על כל הרגיסטרים והביטים של המיקרו בקר 8051 . בעזרת קובץ זה הקומפיילר יודע מה כתובת פורט 1 ומה כתובת כל ביט בפורט וכו'.

void int0 ( ) interrupt 0

הגדרה של פונקציה בשם int0 שאיננה מחזירה ערך (המילה void) ואיננה מקבלת ערכים (הסוגריים הקטנים ריקים). המילה interrupt 0 אומרת לקומפיילר שזו לא פונקציה רגילה אלא פונקציה של פסיקה מספר 0 (פסיקה חיצונית 0) ויש להכניס אותה לזיכרון התכנית החל מכתובת 3 (וקטור הפסיקה של פסיקה חיצונית 0).

if ( I%3==0)

אם השארית אחרי החלוקה של המשתנה I ב 3 שווה 0 אז עבור לפקודה הבאה (ששורה 7) . אם השארית איננה 0 עוברים לשורה 8 .

While ( !TI) ;

לולאת while הבודקת האם מה שנמצא בסוגריים הוא TRUE . היות ובתוך הסוגריים יש !TI (המצב ההפוך ממצב TI) אז כאשר TI הוא 0 אז !TI הוא 1 ושוב מבצעים את לולאת ה while פעם נוספת. כאשר TI יהיה 1 אז !TI יהיה 0 ואז נעבור לפקודה הבאה. באופן מעשי הלולאה בודקת האם הסתיים שידור התו שנשלח ל SBUF לשידור . כל עוד לא הסתיים שידור התו אז TI=0 ו !TI הוא TRUE וחוזרים לשורת ה while לבדיקה נוספת. כאשר הסתיים שידור התו, החומרה של ה UART (התקשורת הטורית) מעלה את הדק TI ל 1 ואז מסתיימת לולאת ה while כי אז TI שווה 1 ו !TI = 0 .



ב.

TMOD=0x20

שורה 15 :

מעבירים לרגיסטר TMOD את הערך 20H .

רגיסטר TMOD ייראה כך :

GATE	C/T'	M1	M0	GATE	C/T'	M1	M0
0	0	1	0	0	0	0	0
טיימר 1				טיימר 0			

זה אומר שעובדים עם טיימר 1 באופן 2, כטיימר (לא קאונטר) וב  $GATE=0$  זה אומר שמאפשרים ספירה ללא תלות בהדק  $\overline{int1}$ . טיימר 1 הפועל באופן 2 (טעינה חוזרת אוטומטית -  $\overline{AUTO}$  (RELOAD) הוא זה שמוציא גל מרובע - פולסי שעון - אל ה UART והוא קובע את קצב התקשורת באופן 1 ו 3 של ה UART.

את טיימר 0 מפעילים כטיימר באופן 0 (טיימר של 13 ביטים) ללא תלות בהדק  $\overline{int0}$ .

SCON = 0x40 ;

שורה 16 :

העבר 40H לרגיסטר SCON (רגיסטר בקרת התקשורת הטורית). הרגיסטר נראה כך:

SM0	SM1	SM2	REN	TB8	RB8	TI	RI
0	1	0	0	0	0	0	0

2 הסיביות הגבוהות M0 M1 קובעות את אופן העבודה עם ה UART. היות ויש בהן 01 זה אומר שעובדים באופן 1 שבו ה UART עובד עם 8 ביט בקצב תקשורת משתנה, כלומר ניתן לשנות אותו בעזרת טיימר 1.

SM2=0 - עובדים עם מיקרו בקר אחד בלבד.

REN=0 - לא מאפשרים קליטה של תקשורת טורית.

TB8=0 - Transmit Bit 8 - כאן המתכנת ישים את הביט התשיעי המשודר כשעובדים עם זוגיות. לא שימושי בשאלה כאן.

**Receive bit 8 - RB8=0** - כאן יימצא הביט התשיעי הנקלט (כי מתחילים מביט מספר 0) בתקשורת טורית, כאשר עובדים עם זוגיות. הביט נשלט על ידי החומרה ומציין מהו הביט התשיעי שנקלט. אין לו משמעות בתרגיל זה כי עובדים עם UART של 8 ביט.

**TI=0** - הביט עולה ל 1 על ידי החומרה של ה UART כאשר הסתיים שידור התו שנמצא ב SBUF. המתכנת בודק את הביט ואם יש בו 1 הוא יודע שהסתיים שידור הבייט ואז הוא שם בביט אפס בעזרת הפקודה CLR TI ואז יכול לשלוח ביית חדש לשידור. אנחנו מאפסים אותו כדי שנוכל להתחיל לחדש ואז לבדוק האם הסתיים השידור.

**RI=0** הביט עולה ל 1 על ידי החומרה של ה UART כאשר הסתיימה קליטה טורית של ביית. המתכנת יבדוק את הביט. אם יש בו 1 יודעים שנקלט בית חדש. יש למשוך את הבייט מה SBUF ולאפס את הביט בתוכנה כדי להתכונן לקליטת ביית חדש. היות ובתרגיל כאן לא מאפשרים קליטה אז הביט חסר משמעות.

**שורה 17 TH1=0xFD;**

העבר לחלק הגבוה של טיימר 1 את הערך FDH (253 עשרוני). טיימר 1 עובד באופן Auto – Reload כאשר החלק הנמוך של הטיימר TL1 הוא הסופר ובסיום ספירה הוא נטען לערך שיש ב TH1. ערך זה קובע את קצב התקשורת הטורית.

**שורה 18 : TL1=0xFD;**

העבר ל 8 הביטים הנמוכים של טיימר 1 את הערך FDH. כדי שהוא יתחיל לספור מערך זה.

**שורה 19 : TR1=1;**

שים '1' בביט TR1 שברגיסטר ה TCON. כך אומרים לטיימר 1 רוץ והוא מתחיל לספור את הפולסים המגיעים אליו.

**שורה 20 : TI = 0**

שים 0 בביט TI ברגיסטר ה SCON. ראה הסבר בעמוד קודם על הביט.

**שורה 21 : IE = 0x81;**

העבר את הערך 81H לרגיסטר IE שהוא רגיסטר אפשור הפסיקות. בפקודה זו מאפשרים קבלת פסיקה חיצונית 0 וחוסמים קבלת שאר הפסיקות.

**שורה 22 : IT0 = 1;**

שים 1 בביט IT0 שברגיסטר ה TCON. כך קובעים שפסיקה חיצונית 0 תעבוד על ירידות מ 1 ל 0 - EDGE - ולא על רמה - LEVEL.

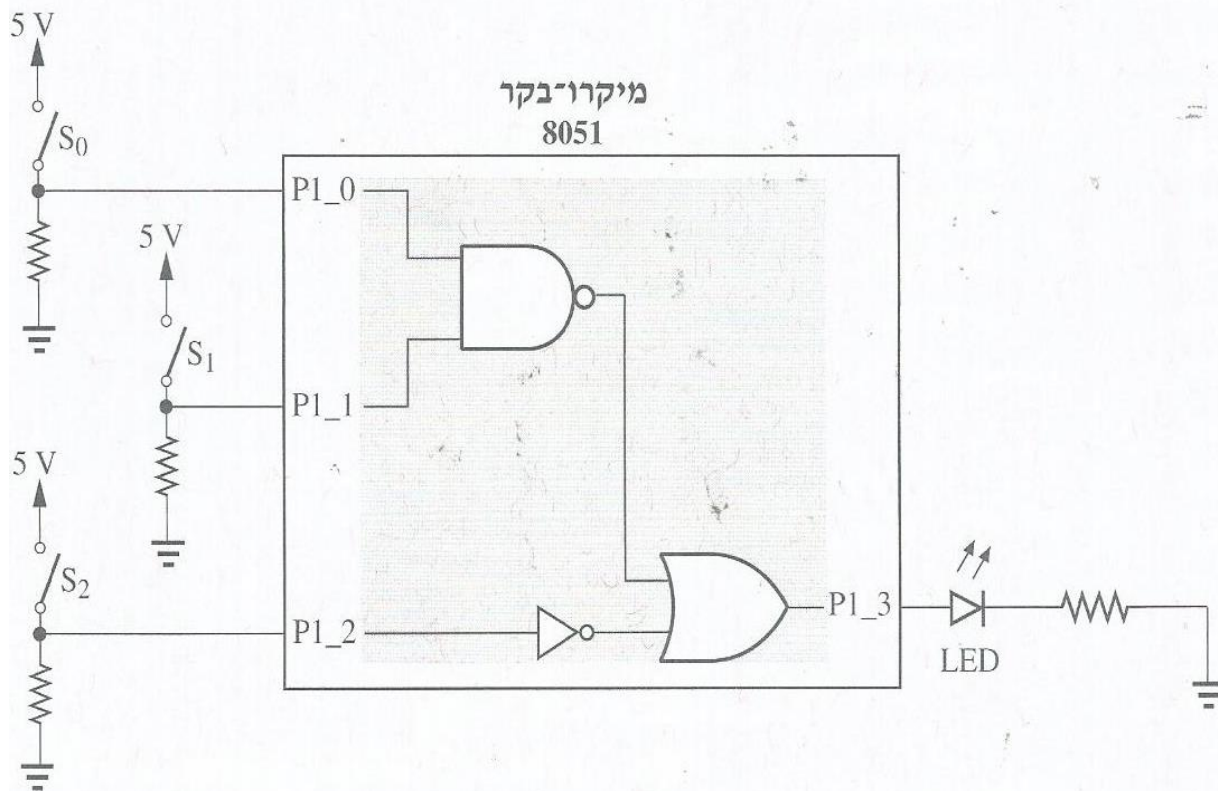


המשדר משדר את התו שבמשתנה CH. בשורה 13 מעבירים ל CH את הערך 'A' (ערך האסקי של התו A). בכל פעם שהמשתמש לוחץ על המפסק עוברים לתכנית הפסיקה ושם מקדמים את הערך שבמשתנה I הסופר כמה לחצות ביצענו. בודקים האם השארית לאחר חלוקה של I ב 3 היא 0 (כאשר I הוא 0, 3, 6, 9 וכו' השארית לאחר חלוקה ב 3 היא 0!). אם השארית 0 שמים במשתנה CH את הערך 'B' ולכן משדרים את התו 'B' ואם השארית איננה 0 (כלומר השארית היא 1 או 2) מעבירים למשתנה CH את הערך 'C'. במילים אחרות: בהתחלה, לפני הלחיצה הראשונה נשדר 'A'. בלחיצה הראשונה והשנייה משודר התו 'C' בלחיצה השלישית משודר התו 'B'. הדבר מתבצע בצורה מחזורית כאשר 2 לחיצות משדרות את התו 'C' והלחיצה השלישית את התו 'B'.

נימוק	הנתונים שישודרו במוצא ה UART	
הערך במשתנה CH='A' (שורה 13).	'A'	לפני הלחיצה הראשונה
I=1 לא נותן שארית 0 לאחר חלוקה ב 3	'C'	לאחר הלחיצה הראשונה
I=2 לא נותן שארית 0 לאחר חלוקה ב 3	'C'	לאחר 2 לחיצות
I=3 נותן שארית 0 לאחר חלוקה ב 3	'B'	לאחר 3 לחיצות
I=4 לא נותן שארית 0 לאחר חלוקה ב 3	'C'	לאחר 4 לחיצות
I=5 לא נותן שארית 0 לאחר חלוקה ב 3	'C'	לאחר 5 לחיצות
I=3 נותן שארית 0 לאחר חלוקה ב 3	'B'	לאחר 6 לחיצות
I=7 לא נותן שארית 0 לאחר חלוקה ב 3	'C'	לאחר 7 לחיצות

שאלה 8

באיור לשאלה 8 נתון מיקרו־בקר 8051. מעוניינים לממש בתוכנה את המעגל הלוגי המתואר בתוכו.



איור לשאלה 8

א. כתוב תכנית בשפת־הסף של המיקרו־בקר 8051 שתממש את המעגל הלוגי המתואר באיור. על התכנית:

1. לקלוט את הרמה הלוגית של המפסקים  $S_0 \div S_2$  דרך שלושת ההדקים  $P1_0 \div P1_2$ , בהתאמה.

2. לחשב את הרמה הלוגית במוצא המעגל הלוגי, ולהציג אותה על־ידי נורית ה־LED במוצא ההדק  $P1_3$ .

ב. תדר הגביש של המיקרו־בקר הוא 12 MHz. חשב את הזמן שיעבור מרגע שינוי מצב המפסקים במבוא הבקר ועד לרגע שינוי מצב ה־LED, בהתאם לקוד התכנית שכתבת בסעיף א'.

**תשובה 8**

.א.

נבדוק מהי הפונקציה המתקבלת ב P1\_3

$$P1\_3 = \overline{P1\_0} \& \overline{P1\_1} + \overline{P1\_2} = \overline{P1\_0} + \overline{P1\_1} + \overline{P1\_2}$$

הערה: הסימן & מציין פעולת AND

נרשום את התוכנית כפרוצדורה ( בסוגריים בצד של פקודה רשום מספר מחזורי מכונה של כל פקודה – נלקח מהנוסחהון):

LED:

MOV C,P1.0 (1)

CPL C (1)

ORL C, /P1.1 (2)

ORL C, /P1\_2 (2)

MOV P1.3,C (2)

RET

ב. תדר הגביש הוא 12MHz. זמן פולס שעון הוא:  $1/12 * 10^6 = 1/12 \mu\text{sec}$  (1 חלקי 12 מליונית שנייה).

12 מחזורי שעון הם מחזור מכונה אחד ולכן:  $T_{\text{machine cycle}} = 12 * 1/12 \text{Usec} = 1 \mu\text{sec}$ .

נסכם את כל מחזורי המכונה הרשומים בסוגריים בסעיף הקודם:  $1 + 1+2+2+2 = 8$

מכאן שהזמן העובר מהפקודה הראשונה עד RET הוא:  $T_{\text{delay}} = 8 * 1 \text{Usec} = 8 \mu\text{sec}$

