

פתרון מבחן במיקרו תשע"אשאלה 5

- כתוב תת שגרה בשפת הסף של המיקרו בקר 8051 או תוכנית בשפת C שלו שתבצע את הפעולות שלהלן:
1. תחפש את המספר הגדול ביותר בבלוק נתונים, הכולל מספרים לא מסומנים (חיוביים) בלבד. הבלוק מתחיל בכתובת 40H בזיכרון הפנימי של המיקרו בקר וגודלו 10 בתים.
 2. תעביר את המספר הגדול ביותר בבלוק הנתונים להדקי המוצא של המיפתח P1 במיקרו-בקר ותציג אותו כמספר בינארי.

תשובה 5

- א. נפתור את התרגיל גם בשפת סף וגם בשפת C51. נתחיל עם שפת סף.

Find_max:

```
Mov r0,#40h ; מצביע כתובות
Mov a,#0 ; המספר החיובי הקטן ביותר האפשרי
Mov r7,#10 ; מונה לולאה
```

Again: clr c

```
Subb a,@r0
Jnc next ; אם המספר באקומולטור גדול או שווה למספר בזיכרון
Mov a,@r0 ; במידה ויש מספר גדול יותר בזיכרון מעבירים אותו לאקומולטור
```

Next: inc r0

```
Djnz r7,again
Ret
```

פתרון בשפת C51

```
#include <8051.h>
void find_max()
{
    unsigned char at 0x40 block[10];
    unsigned char big,i;
    for(big=i=0;i<10;i++)
        if(big<block[i])
            big=block[i];
}
```

ב.

בשפת הסף: לפני הפקודה האחרונה (ret) יש לרשום
 בשפת C51: c51 : נרשום בשורה האחרונה של הפונקציה : P1=big;

שאלה 6

נתונה תוכנית בשפת C של המיקרו בקר 8051 (תדר הגביש של המיקרו בקר הוא 12MHz):

1. #include <8051.h>
2. unsigned char c;
3. sbit P1_7 = 0x97;
4. void t0 () interrupt 1
5. {
6. c++;

```

7.      if(c==15)
8.      {
9.          c = 0;
10.         P1_7 = ~P1_7;
11.     }
12. }
13. void main ( )
14. {
15.     c=0;
16.     ET0 = 1;
17.     EA = 1;
18.     TMOD = 0x01;
19.     TH0 = 0;
20.     TL0 = 0;
21.     TR0 = 1;
22.     while(1);
23. }
```

- א. הסבר את הפקודות שבשורות : 3 , 4 , 10 , 22
 ב. הסבר את משמעות קטע תת השגרה שבשורות 16 עד 21 בתוכנית. היעזר בנוסחאון של המיקרו בקר לצורך ניתוח מילות הבקרה.
 ג. 1. חשב את זמן המחזור של האות המופק בהדק P1_7 .
 2. סרטט את צורת האות המופק בהדק P1_7 כפונקציה של הזמן. בסרטוטך ציין את הרמות הלוגיות של האות ואת זמן המחזור שלו.

תשובה 6

א.

```
sbit P1_7 = 0x97 ;
```

הפקודה מגדירה ביט באחד הרגיסטרים שבאזור הרגיסטרים המיוחדים (SFR). הביט נמצא בכתובת 97H (שייך לפורט 1 שהביטים שלו מ 90H ל P1.0 ועד ביט 97H לביט P1.7). הפקודה אומרת לקומפיילר שבכל מקום בתוכנית שנרשום P1_7 מדובר בביט מספר 97H.

```
void t0 ( ) interrupt 1
```

הגדרה של פונקציה בשם t0. הפונקציה איננה מחזירה ערך ולא מקבלת ערכים. המילה interrupt היא מילה שמורה שאומרת לקומפיילר שמדובר בפסיקה מספר 1 שהיא פסיקה של טיימר 0 ויש לשים את הפקודה הראשונה של הפונקציה החל מכתובת 0BH שהיא הכתובת המתאימה לפסיקה של טיימר 0.

```
P1_7 = ~ P1_7;
```

הסימן ~ (טילדה) אומר להפוך את מצב הדק P1_7. אם בהדק זה היה 0 אז ההדק יעבור ל 1 ולהפך.

```
while(1);
```

לולאת while שבה התנאי הוא האם 1 הוא TRUE. היות והתשובה לתנאי היא חיובית ובסוף הלולאה יש נקודה פסיק (;) אז השורה 22 מתבצעת אין סוף פעמים. במקרה שלנו, יציאה מהלולאה תהיה רק כאשר תגיע פסיקת טיימר 0 ואז נעזוב את הלולאה, נבצע את הפסיקה ונחזור ללולאה.

ב.

נתבסס על רגיסטרים IE (אפשר הפסיקות) ו TMOD (אופני הטיימר) הנראים כך :

רגיסטר IE

EA	X	ET2	ES	ET1	EX1	ET0	EX0
----	---	-----	----	-----	-----	-----	-----

16. $ET0 = 1;$ // Enable Timer 0 interrupt - 0 שים 1 בביט אפשר פסיקת טיימר 0
 // (תלוי גם בביט EA שבו יש לשים 1)
17. $EA = 1;$
 שים 1 בביט EA ואפשר את כל הביטים של הבקרה שבהמשך הרגיסטר, כלומר בביטים הבאים בהמשך, מימין לביט EA, אם נשים 1 בביט כלשהו אז הפסיקה הספציפית תאופשר.

רגיסטר TMOD

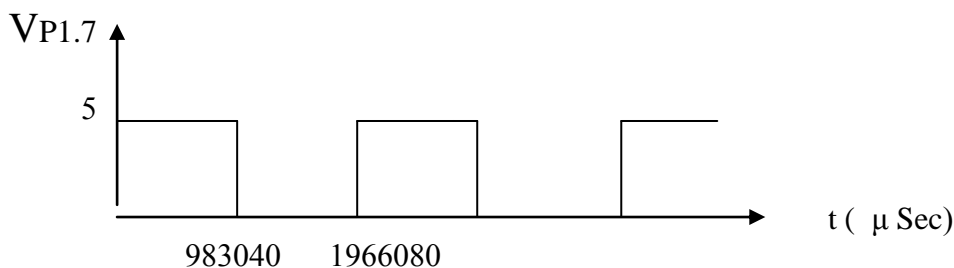
GATE	C / T'	M1	M0	GATE	C / T'	M1	M0
← טיימר 1				← טיימר 0 →			

18. $TMOD = 0x01;$
 הפעל את טיימר 0 כטיימר (ספירת פולסים בתדר הגביש לאחר שחולקו ב 12) באופן עבודה מספר 1 (טיימר של 16 ביטים) ואפשר ספירה שלו (GATE=0). את טיימר 1 הפעל כטיימר באופן 0 וגם כאן מאפשרים את הספירה שלו.
19. $TH0 = 0;$
 שים בחלק הגבוה של טיימר 0 (8 ביטים) את המספר 0
20. $TL0 = 0;$
 שים בחלק הנמוך של טיימר 0 (8 ביטים) את המספר 0
21. $TR0 = 1;$ // TIMER 0 RUN
 העבר לביט TR0 (שברגיסטר ה TCON) אחזק לוגי ועל ידי כך אומרים לטיימר 0 "רוץ", כלומר לספור את הפולסים המגיעים אליו.
 לסיכום: מאפשרים פסיקת טיימר 0, מפעילים את טיימר 0 באופן 1 שהוא ומאשרים את הספירה שלו. טיימר 0 מקבל תדר קבוע שהוא תדר הגביש חלקי 12 וכל פעם שהוא יסיים את הספירה שלו הוא יתאפס ויבקש פסיקה.

ג.
1.

תדר הגביש $F_{crystal} = 12MHz$. התדר המגיע לספירה לטיימר הוא:
 $F_{crystal} / 12 = 12MHz / 12 = 1MHz$

זמן מחזור של תדר 1 מגה הרץ הוא: 1 מיקרו שנייה.
 היות והטיימר הוא של 16 ביטים הוא סופר 2 בחזקת 16 פולסי שעון בין פסיקה לפסיקה. כלומר 65536 מיקרו שניות בין פסיקה לפסיקה.
 בתוכנית הפסיקה מקדמים משתנה בשם c ואם הוא שווה 15 מחליפים את מצב הדק P1.7. לכן יש להכפיל את המספר מיקרו שניות 65536 ב 15 ונקבל 983040 מיקרו שניות. את זה נכפיל ב 2 כי זמן מחזור מורכב מזמן שבו ההדק ב 0 וגם מהזמן שההדק ב 1. מכאן נקבל 1966080 מיקרו שניות.
 $T = 1.966080$



2.

שאלה 7

נתונה תת השגרה בשפת הסף של המיקרו בקר 8051 .

```

1. SUB: MOV R1,#50H
2.      MOV R6,#0
3.      MOV R7,#10
4. AGAIN:MOV A,@R1
5.      MOV B,#2
6.      DIV AB
7.      MOV A,B
8.      CJNE A,#0,NEXT
9.      INC R6
10. NEXT: INC R1
11.     DJNZ R7,AGAIN
12.     RET
    
```

א. הסבר את הפקודות שבשורות : 2 , 4 , 6 , 8 , 11

ב. מה מבצעת תת השגרה הזו ?

ג. בטבלה שלהלן נתונים התכנים של תאי הזיכרון שכתובותיהם $50H \div 59H$ בזיכרון הפנימי של המיקרו בקר , לפני ביצוע תת השגרה.

50H	51H	52H	53H	54H	55H	56H	57H	58H	59H
11H	44H	33H	22H	12H	55H	67H	67H	88H	14H

מה יהיה תוכן אוגר R6 לאחר ביצוע תת השגרה הזו ?

תשובה 7

א.

```
MOV R6,#0
```

העבר לרגיסטר R6 מיידית את הנתון 0 .

```
AGAIN: MOV A,@R1
```

העבר אל האקומולאטור נתון מזיכרון הנתונים הפנימי, מהכתובת שעליה מצביע הרגיסטר R1 .
AGAIN היא תווית – כתובת .

```
DIV AB
```

בצע חלוקה בין האקומולאטור לרגיסטר B . התוצאה עוברת לאקומולאטור והשאריית תהיה ברגיסטר B .

```
CJNE A,#0,NEXT
```

בצע השוואה בין תוכן האקומולאטור לנתון 0 . אם הם לא שווים – קפוץ לתווית NEXT . פעולת ההשוואה מתבצעת ע"י חיסור של האקומולאטור בנתון 0 ובדיקת תוכן האקומולאטור האם הוא אפס.

```
DJNZ R7,AGAIN
```

חסר 1 מתוכן רגיסטר R7 ואם תוכן הרגיסטר R7 לא שווה 0 – קפוץ לתווית AGAIN .

ב.

תת השגרה פועלת על בלוק נתונים בגודל 10 כתובות (בזיכרון הנתונים הפנימי – DIRECT) , החל מכתובת 40H ועד 49H (כולל). התוכנית בודקת האם הנתון בכל כתובת הוא זוגי (בחלוקת כל מספר ב 2 השאריית יכולה להיות או 0 – אם המספר זוגי, או 1 אם המספר איננו זוגי). אם הנתון זוגי אז מגדילים מונה ב 1 . בסיום תת השגרה יהיה ברגיסטר R6 – המשמש כמונה כמות הנתונים הזוגיים - את כמות המספרים הזוגיים שבבלוק. R1 הוא מצביע בעזרתו "רצים" כתובת אחרי כתובת בזיכרון. R7

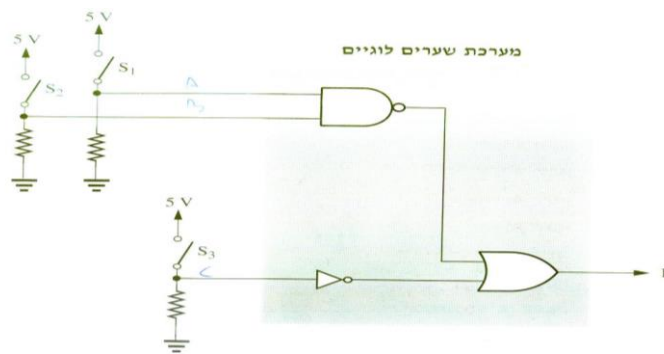
הוא מונה לולאה שמראה על כמה כתובות יש לעבור בזיכרון , או במילים אחרות, הוא מציין את גודל הבלוק.

ג.

עבור הטבלה הנתונה יהיה ברגיסטר R6 את המספר 5 כי הנתונים הזוגיים נמצאים בכתובות : 51H , 53H , 54H, 58H, 59H

שאלה 8

באיור לשאלה נתונה מערכת שערים לוגיים, המופעלת באמצעות שלושה מתגים S1 ÷ S3 .

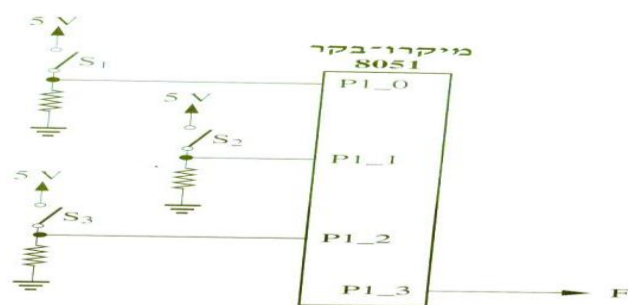


איור א' לשאלה 8

כתוב תוכנית בשפת הסף של המיקרו בקר 9051 למימוש בתוכנה של מערכת השערים הלוגיים במעגל המתואר באיור א'.

על התוכנית :

1. לקלוט את הרמות הלוגיות של המתגים דרך שלושת ההדקים P1.0 ÷ P1.2 , כמתואר באיור ב' לשאלה.
2. להפיק את הרמה הלוגית במוצא ההדק P1.3 , בהתאם לפונקציה המתקבלת ממערכת השערים הלוגיים.



איור ב' לשאלה 8

תשובה 8

נפתור את התרגיל ב 2 דרכים.

דרך א' :

העבר לדגל הנשא את מצב P1.0 ; `MOV C,P1.0`
 בצע פעולת AND בין דגל הנשא ומצב הדק P1.1 ; `ANL C,P1.1`

```
CPL C ; הפוך את מצב דגל הנשא
ORL C,/P1.2
MOV P1.3,C
RET
```

זרד ב'

מתוך המעגל עם השערים ניתן לראות ש: $F = \overline{P1.0 \text{ and } P1.1} + \overline{P1.2}$
או ניתן לרשום גם:

$F = \overline{P1.1 \text{ and } P1.2 \text{ and } P1.3}$

במילים אחרות: אם ב 3 הסיביות יש 1 או F יהיה 0. כאשר אחת או יותר מהכניסות ב 0 – היציאה ב 1.

```
MOV A,P1
```

```
ANL A,#00000111B ; בדיקת 3 הסיביות הנמוכות המתאימות ל 3 הכניסות של המפסקים
```

```
CJNE A,#7,ZERO ; האם כל 3 המפסקים ב 1 ?
```

```
SETB P1.3
```

```
RET
```

```
ZERO: CLR P1.3
```

```
RET
```