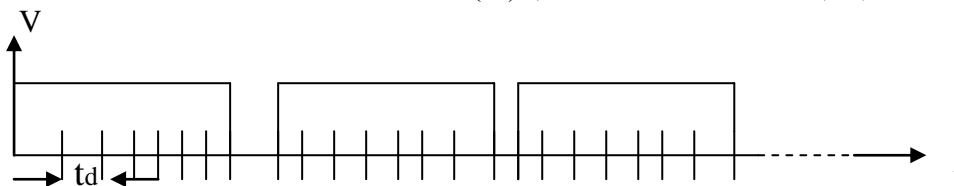


שאלה 5

להלן תת שגרה, הכתובה בשפת הסף של המיקרו בקר 8051.

1. MOV A,#0FH
2. CLR C
3. MOV R5,#8
4. SETB P2.1
5. NEXT: RRC A
6. MOV P2.1,C
7. DJNZ R5,NEXT
8. SETB P2.1
9. RET

- א. הסבר את הפקודות שבשורות 4, 6, 7.
- ב. סרטט את האות המופק בהדק P2.1 בעקבות ביצוע תת השגרה. לווה שרטוטך בהסבר מתאים.
- ג. שנה את תת השגרה כך שהיא תפיק במוצא הדק P2.1 את האות המחזורי, המתואר באיור הבא. כל מחזור מורכב משבעה פרקי זמן זהים שמתח המוצא בהם גבוה ('1') ומפרק זמן אחד שמתח המוצא בו נמוך ('0').



הערה: t_d - זמן הביצוע של מחזור אחד של הלולאה שבשורות 5 עד 7 בתת השגרה.

פתרון שאלה 5

א.

SETB P2.1

שים 1 בהדק P2.1.

MOV P2.1,C

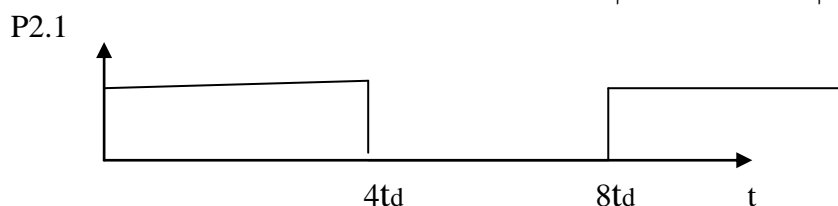
העבר את תוכן דגל הנשא - CARRY - אל הדק P2.1.

DJNZ R5,NEXT

חסר 1 מתוכן רגיסטר R5 ואם התוכן שלו איננו 0 חזור אל התווית (כתובת) NEXT.

ב.

היות והנתון ב A הוא 0FH שזה 00001111 בינארי, אז 4 סיבובים של הפקודות 5 עד 7 ההדק יהיה ב 1 ו 4 מחזורים יהיה ב 0. בסיום מעלים את הרגל ל 1. אם למחזור הזמן של הפקודות שבשורות 5 עד 7 נקרא t_d הגל המופק יראה:



ג.

כדי לקבל את הגל המחזורי המשורטט יש להעביר לרגיסטר A נתון שבו 7 "אחדים" ו '0' בודד. לדוגמא נשלח את הנתון 7FH (01111111).

אם רוצים צורת גל מחזורית אין סופית יש לבצע השינויים הבאים:

1. יש לשנות את שורה 1 ל : MOV A,#7FH (או כל נתון אחר שהזכרנו)
2. במקום שורה 7 יש לרשום את הפקודה SJMP NEXT
3. בשורה 9 במקום RET לרשום פקודת קפיצה לשורה 1 .

שאלה 6

כתוב תת שגרה בשפת הסף של המיקרו בקר 8051 או תוכנית בשפת C שלו , המאפסת את כל המספרים השליליים בבלוק נתונים המתחיל בכתובת 40H בזיכרון הפנימי של המיקרו בקר. גודל הבלוק 10 בתים. כל נתון בבלוק הנתונים הוא בן 8 ביט והנתונים מיוצגים בשיטת המשלים ל 2 .

פתרון שאלה 6

נפתור גם בשפת סף וגם בשפת C51 .

נתחיל עם שפת סף. הרעיון הוא לבדוק את סיבית ה MSB . אם יש בה '1' אז המספר שלילי ואם יש '0' אז המספר חיובי.

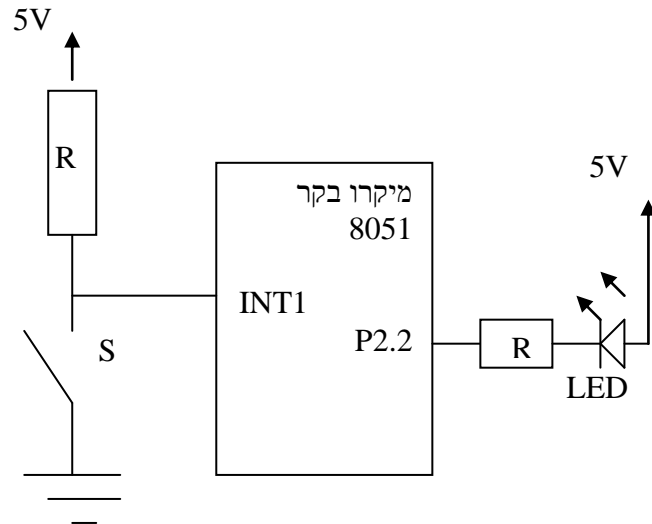
```
NEG_CHK:
    MOV R0,#40H      ; כתובת התחלה
AG:    MOV A,@R0
        ANL A,#80H   ; איפוס 7 הביטים הנמוכים
        JNZ ZERO_NATUN ; אם המספר שלילי קפוץ לכתובת המאפסת את הנתון
CONT:  INC R0
        CJNE R0,#4AH,AG // האם הגענו לסוף בלוק הנתונים
        RET
ZERO_NATUN:
        MOV @R0,#0   // איפוס הנתון השלילי
        SJMP CONT
```

בשפת C51

```
void chk_neg(void)
{
    data at 0x40 char block[10],x;
    for(x=0;x<10;x++)
        if(block[x]<0)
            block[x]=0;
}
```

שאלה 7

באיור לשאלה נתונה נורית LED המחוברת להדק P2.2 של המיקרו בקר 8051, ומפסק המחובר להדק הפסיקה INT1 שלו.



להלן תוכנית בשפת C של המיקרו 8051 .

```
1. #include <8051.h>
2. void init(void)
3. {
4.     TCON=0x04;
5.     IE = 0x84;
6. }
7. void INT1_ISR (void) interrupt 2
8. {
9.     unsigned int I;
10.    P2_2 =0 ;
11.    for(i=0;i<64000;i++);
12.    P2_2=1;
13. }
14. void main (void)
15. {
16.    init();
17.    P2_2=1;
18.    while(1);
19. }
```

א. הסבר את הפקודות שבשורות 7, 9, 12, 16, 18. ב. הסבר את משמעות השורות 4 ו 5 בתוכנית. ג. הסבר מה מבצעת התוכנית.

פתרון שאלה 7

א.

void INT1_ISR (void) interrupt 2
הגדרה של פונקציה בשם INT1_ISR אשר איננה מחזירה ערך – ה void - השמאלי, שאיננה מקבלת פרמטרים - ה void שבסוגריים. המילה interrupt אומרת לקומפיילר שהפונקציה

איננה פונקציה "רגילה" אלא פסיקה והמספר 2 אומר שזוהי פסיקה מספר 2 (פסיקת int1) ויש להכניס את הפקודות של הפונקציה החל מכתובת 13H בזיכרון התוכנית.

```
unsigned int i ;
```

הגדרה של משתנה בשם i מטיפוס שלם לא מסומן (חיובי בלבד).

```
P2_2=1;
```

בצע השמה להדק P2_2 ושים בו '1'.

```
init( );
```

קריאה (זימון) פונקציה בשם init .

```
while(1);
```

לולאת while היא לולאה המבצעת משפט או מספר משפטים כל עוד התנאי הנרשם בסוגריים מיד אחרי המילה while מתקיים. במקרה שלנו התנאי הוא כל עוד (1==TRUE). היות והתנאי תמיד מתקיים אז הלולאה היא אין סופית. בגלל שבלולאה אין משפט או משפטים לבצע (כי בסיום התנאי יש נקודה פסיק) אז נמצאים בלולאה אין סופית שרק בודקת האם התנאי מתקיים, והיות והוא מתקיים, שוב חוזרת לבדוק האם התנאי מתקיים וחוזר חלילה. מלולאה זו נצא רק כאשר יש פסיקה ובסיום הפסיקה נחזור ללולאת ה while .

ב.

```
TCON=0x04;
```

עבוד עם רמה - LEVEL - עם פסיקה חיצונית 0 (IT0=0), ועם שפה - EDGE - בפסיקה חיצונית 1 (IT1=1).

IE0=IE1=0 איפוס בקשות פסיקה חיצונית 0 ו 1 (אם היו) .

TR1=TR0=0 אומר שלא מריצים את טיימר 0 וטיימר 1 .

TF0=TF1=0 אומר איפוס בקשות פסיקה של הטיימרים (במידה והיו).

```
IE=0X84;
```

אפשר פסיקה חיצונית 1 בלבד.

שתי השורות ביחד מאפשרות את פסיקה חיצונית 1 ועבודה איתה ב EDGE (על "ירידות").

ג.

בתוכנית הראשית קוראים לפונקציה איתחול init שבה מאפשרים את פסיקה חיצונית 1 (אליה מחובר המפסק), ומפעילים את הפסיקה על EDGE.

לאחר האתחול שמים בהדק P2_2 של המיקרו '1' (כ 5V - בהנחה שזהו מתח הספק של ה 8051) דבר הגורם לכיבוי ה LED ונכנסים ללולאה אין סופית.

כאשר יש לחיצה על מפסק S עוברים לפונקציה הפסיקה שבה מורידים את הדק P2_2 ל '0' (מדליקים את ה LED), מבצעים שהייה לזמן מסוים (לפי המספר שבמשפט ה for - 64000 במקרה שלנו) ומחזירים את ההדק ל '1' .

לסיכום: בכל לחיצה על המפסק נקבל הדלקה של ה LED לזמן מסוים .

שימושים נוספים למעגל כזה הם כמעגל חד יציב - ONE SHOT או כמערכת להתגברות על ניתוח המגעות בזמן לחיצה על מפסק.

שאלה 8

נתונה תת שגרה, הכתובה בשפת הסף של המיקרו בקר 8051 .

```
1. SUB:      MOV R0,#40H
```

- 2. MOV R5,#5
- 3. CLR A
- 4. NEXT: ADD A,@R0
- 5. DA A
- 6. INC R0
- 7. DJNZ R5,NEXT
- 8. MOV @R0,A
- 9. RET

א. הסבר את הפקודות שבשורות 1, 4, 5, 7.
ב. הסבר מה מבצעת תת השגרה.
ג. בטבלה שלהלן נתונים התכנים של תאי הזיכרון שכתובותיהם 40H עד 45H בזיכרון הפנימי של המיקרו בקר, לפני ביצוע תת השגרה.

45H	44H	43H	42H	41H	40H	כתובת התא
59H	11H	10H	2H	28H	12H	תוכן התא

מה יהיו התכנים של התאים לאחר ביצוע תת השגרה?
ד. מה יהיו התכנים של התאים לאחר ביצוע תת השגרה אם נבטל את הפקודה DA A שבשורה 5?

פתרון שאלה 8

א.

SUB: MOV R0,#40H

העבר לרגיסטר R0 את הנתון 40H. SUB היא שם הפרוצדורה וגם תווית – כתובת.

NEXT: ADD A,@R0

חבר בין הערך שבאקומולאטור עם הערך הנמצא בכתובת שרגיסטר R0 מצביע עליו. תוצאת החיבור תהיה באקומולאטור והדגלים מושפעים.

DA A

בצע תיקון עשרוני לאקומולאטור. הפקודה עובדת אחרי פעולת חיבור בין 2 מספרי BCD.

DJNZ R5,NEXT

חסר 1 מהתוכן של רגיסטר R5 ואם הוא לא אפס קפוץ לכתובת המסומלת על ידי התווית NEXT. אם R5 שווה אפס המשך הלאה בתוכנית.

ב.

התוכנית מחברת בלוק של 5 כתובות, החל מכתובת 40H ועד 44H בזיכרון הנתונים הפנימי. את תוצאת החיבור היא מכניסה לכתובת 45H. המספרים בבלוק הם מספרים המיוצגים ב BCD מצופף. החיבור המתבצע הוא חיבור עם תיקון התוצאה ל BCD.

ג.

התאים 40H עד 44H לא משנים את ערכם. תא 45H יהיה שווה 63H הסבר:

- 40H → אחרי DA A → 12H+28H=3AH
- 42H → DA A → 40H+2H=42H נשאר
- 52H → DA A → 42H+10H=52H נשאר
- 63H → DA A → 52H+11H=63H נשאר

7. אם נבטל את השורה DA A אז החיבור הוא בינארי רגיל ולכן בכתובת 45H נקבל 5DH לפי :

$$12H+28H = 3AH$$

$$3AH+2H = 3CH$$

$$3CH+10H=4CH$$

$$4CH+11H=5DH$$