

## פתרון מבחן אלקטרוניקה מחשבים 711003 תשע"ח מרץ 2018

### שאלה 5

להלן תוכנית הכתובה בשפת C :

```
1. void chk(int num)
2. {
3.     int a,b,c;
4.     a = num / 100;
5.     b = (num / 10) % 10;
6.     c = num % 10;
7.     printf("%d, %d, %d\n", a, b, c);
8. }
9. void main(void)
10. {
11.     int arr[] = {9,99,1000,999,100};
12.     for (int i=0;i<5;i++)
13.     {
14.         chk(arr[i]);
15.     }
16. }
```

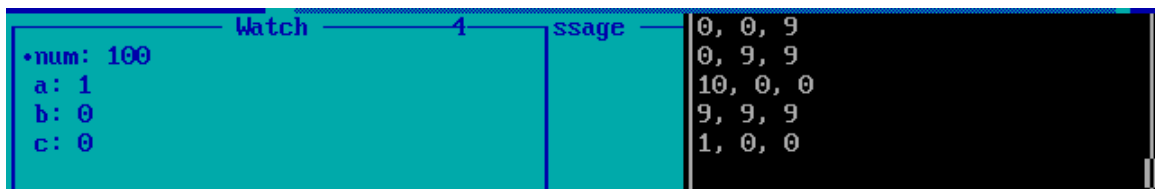
- א. הסבר מה מבצעת הפונקצייה chk . לווה את הסברך בטבלת מעקב אחר כל אחד מהמשתנים בתוכנית (num , a , b , c) .
- ב. רשום את פלט התוכנית, והסבר מה מבצעת התוכנית.
- ג. הוסף לתוכנית קטע־קוד שבו פונקצייה נוספת בשם chk2 , המקבלת מהתוכנית את כל איברי המערך arr כפרמטר, ומחזירה לתוכנית את מספר האיברים במערך המכילים שלוש ספרות.

## פתרון שאלה 5

הערה : בשאלה היה חסר את הפקודה `#include <stdio.h>` בראש התכנית.

א. הפונקציה `chk` מקבלת אחד אחרי השני 5 מספרים הנמצאים במערך `arr`. כל מספר שהיא קיבלה היא מפרקת לספרות המרכיבות את המספר ומציגה את כמות המאות שיש במספר, את ספרות העשרות ואת ספרות היחידות. במשתנה `a` יהיה את כמות המאות, במשתנה `b` את ספרות העשרות ובמשתנה `c` את היחידות  
 לדוגמה : אם המספר בן ספרה אחת לדוגמה 9 נקבל 0, 0, 9  
 עבור המספר 12 נקבל 0, 1, 2  
 עבור מספר בן 3 ספרות כמו 123 נקבל 1, 2, 3  
 עבור המספר בן 4 ספרות כמו 1234 נקבל 1, 2, 3, 4 .  
 עבור המספר בן 5 ספרות כמו 12345 נקבל 1, 2, 3, 4, 5 וכך הלאה.

באיור הבא מתוארת ההדפסה במסך עם דוגמת מעקב אחרי המספר האחרון 100. מצד ימין במסך השחור מקבלים את הספרות עבור כל מספר ובצד שמאל מתואר הפרוק בכל משתנה עבור המספר 100.



ב.

עבור המספר הראשון 9 :

נקבל : 0, 0, 9

עבור המספר 99 : 0, 9, 9

עבור המספר 1000 : 10, 0, 0

עבור המספר 999 : 9, 9, 9

עבור המספר 100 : 1, 0, 0

ג.

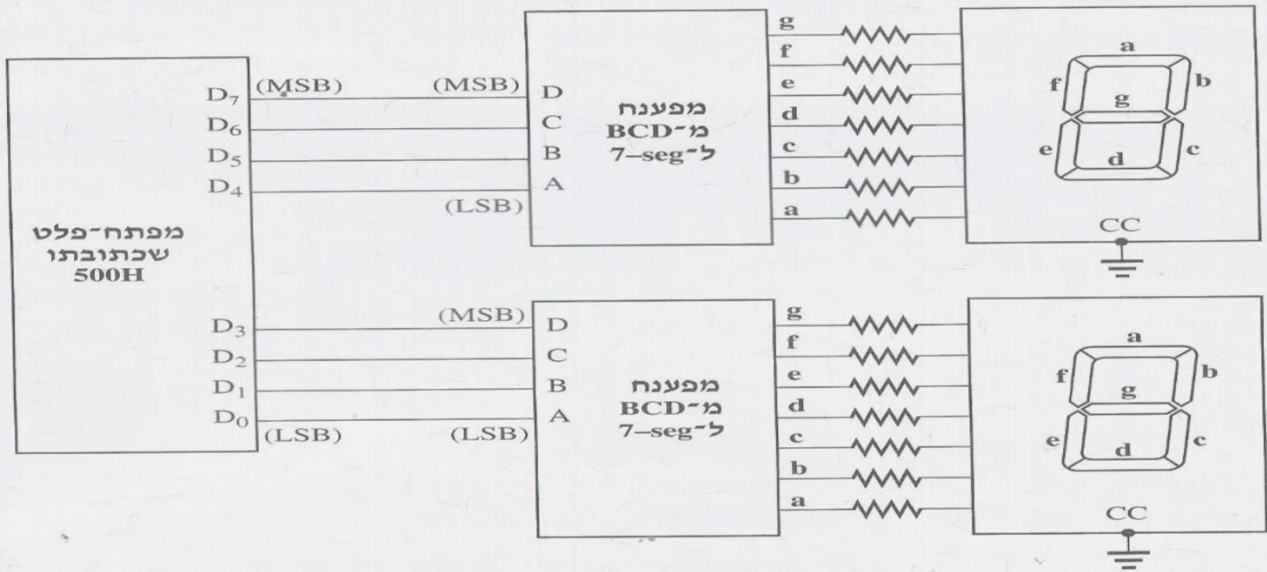
הפקודה שנוסוף בתוכנית ה `main` תהיה `int threeDigits=chk2(arr);` (שולחים את שם המערך שהוא הכתובת של האיבר הראשון של המערך) הפונקציה עצמה תהיה :

```

int chk2(int a[ ])
{
    for( int number=0,int i=0 ; i<5; i++)
        if (a[i]>=100 && a[i]<=999)
            number++;
    return(number);
}
    
```

שאלה 6

למפתח-פלט שכתובתו 500H חיברו שתי תצוגות שבעה מקטעים (7-seg) בחיבור קתודה משותפת (CC), דרך שני מפענחים מ-BCD ל-7-seg, כמתואר באיור לשאלה 6.



איור לשאלה 6

להלן תוכנית הכתובה בשפת C :

```

1. #include <stdio.h>
2. #include <windows.h>
3. void _stdcall Out32(short PortAddress, short data);
4. void main ()
5. {
6.     int lsb = 0, msb = 0, out;
7.     int *p_lsb, *p_msb;
8.     p_lsb = &lsb;
9.     p_msb = &msb;
10.    for(int i=1;i<100;i++)
11.    {
12.        if (*p_lsb == 9)
13.        {
14.            *p_lsb = 0;
15.            *p_msb = *p_msb + 1;
16.        }
17.        else
18.        {
19.            *p_lsb = *p_lsb + 1;
20.        }
21.        out = (*p_msb * 16) | *p_lsb;
22.        Out32(0x500, out);
23.        Sleep(100);
24.    }
25. }

```

- א. הסבר את ההוראות שבשורות 7, 8, 15 ו-22.
- ב. מה יהיה פלט התוכנית שיתקבל על-גבי תצוגות ה-7-seg?
- ג. מחליפים את ההוראה: `out = (*p_msb * 16) | *p_lsb;` בהוראה: `out = (*p_msb << 4) | *p_lsb;` כיצד ישפיע שינוי זה על אופן ביצוע התוכנית? נמק את תשובתך.
- ד. נדרש להציג בתצוגות שבעת המקטעים את המספר 48. כתוב קטע קוד שיבצע זאת.

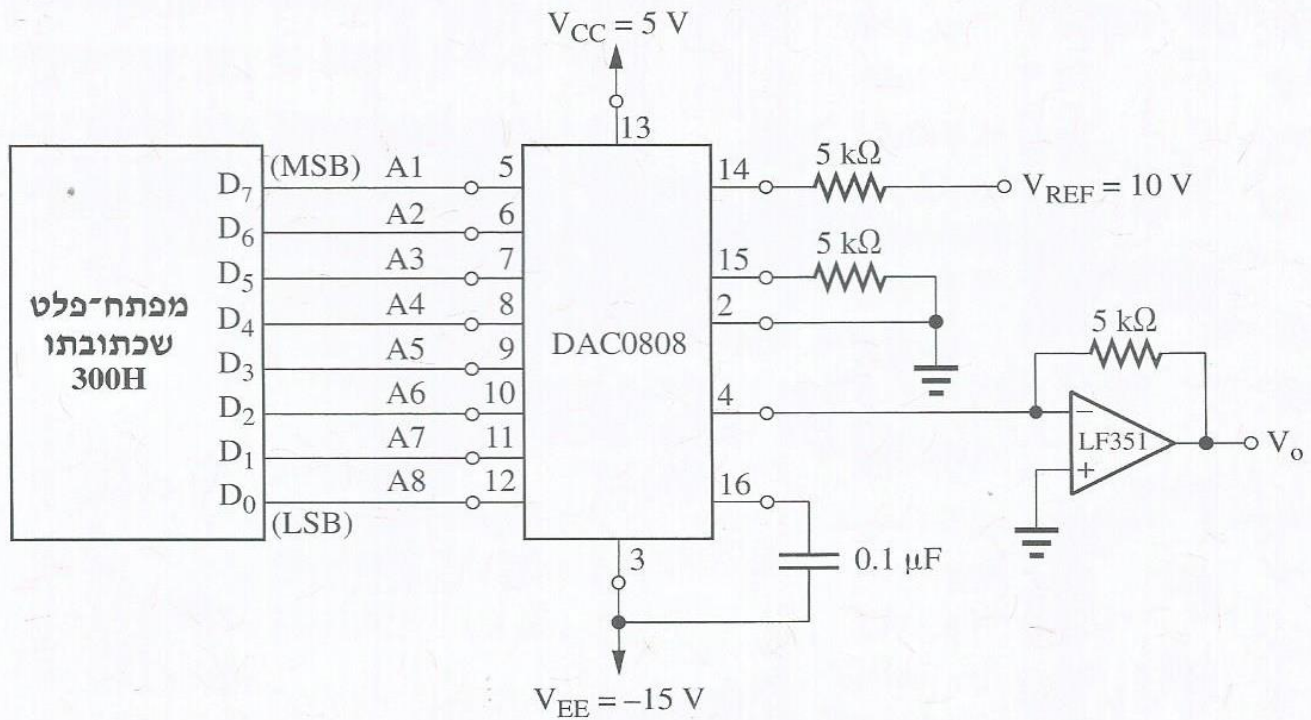
## פתרון שאלה 6

- א. שורה 7 `int *p_lsb, *p_msb;`  
הגדרה של 2 מצביעים מטיפוס שלם והשמות שלהם `p_lsb` ו `p_msb`.
- שורה 8 `p_lsb=&lsb;`  
העבר את הכתובת של המשתנה `lsb` אל המצביע `p_lsb`, כלומר הערך של המצביע `p_lsb` הוא הכתובת של המשתנה `lsb`.
- שורה 15 `*p_msb=*p_lsb+1;`  
העבר לכתובת עליה מצביע `p_msb` את הערך שנמצא בכתובת עליה מצביע `p_lsb` פלוס 1 או במילים אחרות הוסף 1 לערך הנמצא בכתובת עליה מצביע `p_lsb` והעבר את הסכום לכתובת עליה מצביע `p_msb`.
- שורה 22 : `out32(0x500, out);`  
קוראים (מזמנים) את הפונקציה `out32` ושולחים לה את הכתובת `0x500` ואת הערך הנמצא במשתנה `out`. הפונקציה `out32` יודעת להוציא את הנתון שקיבלה לפורט הפלט בכתובת `0x500` ( כתובת פורט הפלט היא 500 בהקסה – דצימלי).
- ב. בתצוגת 7 המקטעים נראה את ספירה כלפי מעלה מהמספר 01 ועד 99 בקצב של 100mSec (עשירית שנייה) בין מספר למספר. התצוגה תהיה ב BCD (לא יהיו ספרות כמו A עד F) כלומר 01 02 03 .... 09 10 11 ..... 19 20 ... עד 99. בסיום התכנית נשאר המספר 99 בתצוגת 7 המקטעים.
- ג. אין השפעה כי הכפלה של הערך הקיים במשתנה `msb` ב 16 או הזזה של המספר 4 פעמים ימינה נותן את אותה התוצאה.
- ד. נרשום פקודה : `out32(0x500,0x48);`

**שאלה 7**

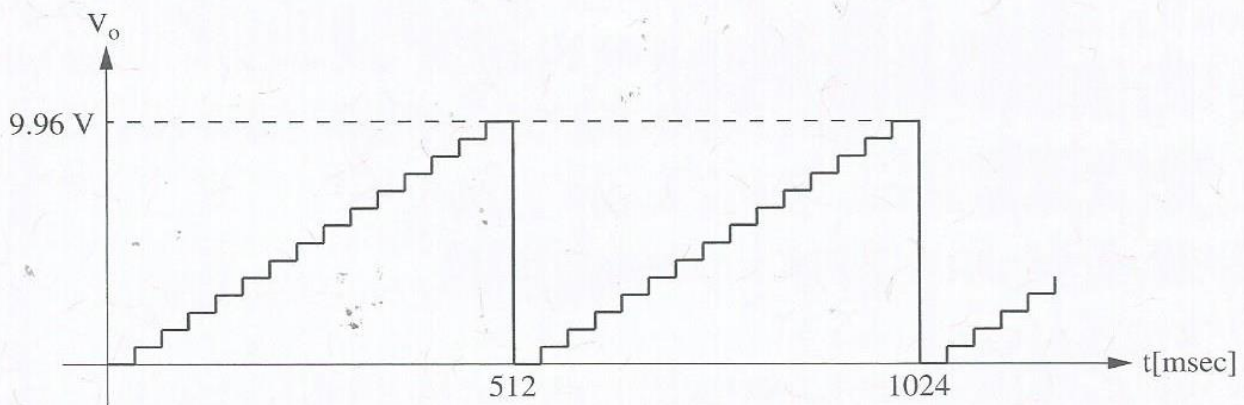
באיור א' לשאלה 7 מתואר חיבור עקרוני בין מפתח-פלט שכתובתו 300H ובין ממיר אות ספרתי לאות תקבילי DAC0808. דף המפרט של הממיר נתון בנספח לשאלה זו.

המתח  $V_o$ , המתקבל במוצא הממיר, נתון בנוסחה:  $V_o [V] = 10 \cdot \left( \frac{A1}{2} + \frac{A2}{4} + \dots + \frac{A8}{256} \right)$ .



**איור א' לשאלה 7**

- א. כתוב תוכנית בשפת C, שתפיק במוצא  $V_o$  של המעגל המתואר באיור א' מתח קבוע של 5 V.
- ב. כתוב תוכנית בשפת C, שתפיק במוצא  $V_o$  את גל שן המסור, המתואר באיור ב' לשאלה:



**איור ב' לשאלה 7**



May 1999

DAC0808 8-Bit D/A Converter

# DAC0808 8-Bit D/A Converter

## General Description

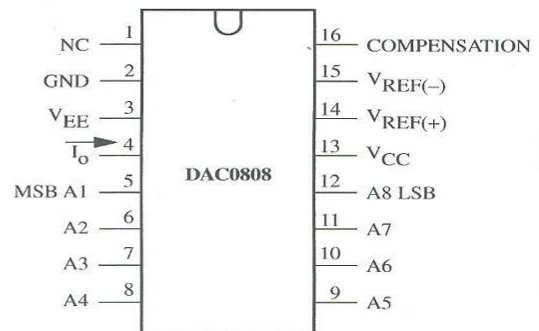
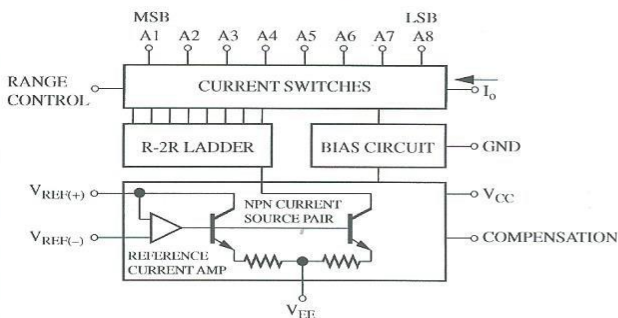
The DAC0808 is an 8-bit monolithic digital-to-analog converter (DAC) featuring a full scale output current settling time of 150 ns while dissipating only 33 mW with  $\pm 5V$  supplies. No reference current ( $I_{REF}$ ) trimming is required for most applications since the full scale output current is typically  $\pm 1$  LSB of  $255 I_{REF}/256$ . Relative accuracies of better than  $\pm 0.19\%$  assure 8-bit monotonicity and linearity while zero level output current of less than  $4 \mu A$  provides 8-bit zero accuracy for  $I_{REF} \geq 2$  mA. The power supply currents of the DAC0808 is independent of bit codes, and exhibits essentially constant device characteristics over the entire supply voltage range.

The DAC0808 will interface directly with popular TTL, DTL or CMOS logic levels, and is a direct replacement for the MC1508/MC1408. For higher speed applications, see DAC0800 data sheet.

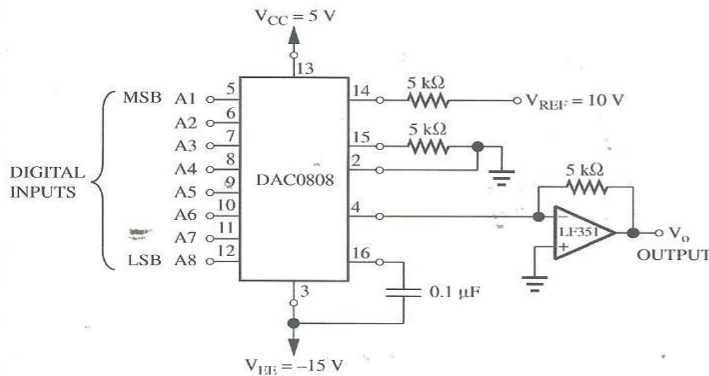
## Features

- Relative accuracy:  $\pm 0.19\%$  error maximum
- Full scale current match:  $\pm 1$  LSB typ
- Fast settling time: 150 ns typ
- Noninverting digital inputs are TTL and CMOS compatible
- High speed multiplying input slew rate: 8 mA/ $\mu s$
- Power supply voltage range:  $\pm 4.5V$  to  $\pm 18V$
- Low power consumption: 33 mW @  $\pm 5V$

## Block and Connection Diagrams



## Typical Application



$$V_o = 10V \cdot \left( \frac{A1}{2} + \frac{A2}{4} + \dots + \frac{A8}{256} \right)$$

FIGURE 1. +10V Output Digital to Analog Converter

## פתרון שאלה 7

א. כדי לקבל מתח קבוע של 5 וולט יש להוציא ל '1' A1 ולכל שאר הסיביות '0' לפי הנוסחה הרשומה ב Vo שבנספח :

$$V_o = 10 (1/2+0/4+0/8+.....+0/256) = 5V$$

בהנחה שהוגדרה השורה

```
void _stdcall Out32(short PortAddress, short data);
```

מכאן שהנתון שנוציא היא 80H = 10000000 והשורה שנכתוב היא:

```
Out32(0x300,0x80);
```

ב. מהציור רואים שיש לנו 15 רמות (מדרגות) מתח ( מ 0 וולט ועד 9.96 וולט ) והן נמשכות 512 שניות. מכאן שכל רמת מתח נמשכת זמן של :  $512/15 = 34.1333\text{mSec}$  . כדי להוציא 0 וולט יש לשלוח להדקים A1 עד A8 את הנתון 00000000 שהיא מדרגה 0. כדי לקבל 9.96 וולט יש להוציא את הנתון 11111111 וזוהי המדרגה האחרונה שהיא מספר 14 .

מכאן שההפרש בין מדרגה למדרגה הוא :  $255/14=18.2142$  .

מדרגת זמן נמשכת  $512/15 = 34.1333$  אלפיות שנייה. ניקח את המספר השלם 34 .

מכאן התכנית שנרשום היא :

```
float out;
while(1)
for(int i=0; i<15;i++)
{
out= 18.2142*i; // תהיה החלק השלם במכפלה
out32(0x300,out);
sleep(34);
}
```

## שאלה 8

בשירות המטאורולוגי מודדים כמה גשם יורד בכל יום. ביום ללא גשם – נרשם הערך 0 .  
ביום שבו יורד גשם – נרשם מספר שלם המייצג את כמות הגשם (במ"מ) שירדה באותו יום.  
כתוב תוכנית בשפת C , שתבצע את הפעולות שלהלן:

1. תקלוט מהמקלדת את כמות הגשם (במ"מ) שירדה מדי יום במשך תקופה מסוימת.  
קליטת הנתונים לתוכנית תיפסק כאשר נקלט הערך 1000 .
2. תספור ותציג כפלט כמה ימים גשומים וכמה ימים ללא גשם היו בתקופה הזו.
3. תספור כמה ימים עברו בין כל יום גשום ליום הגשום הבא אחריו (לא כולל את שני הימים שבהם ירד גשם) בתקופה הזו. התוכנית תציג כפלט את מספר הימים המרבי ללא גשם בין שני ימים גשומים סמוכים במהלך התקופה הזו.

לדוגמה:

עבור הקלט (הנקרא משמאל לימין): 5 22 44 0 0 0 45 0 0 0 0 43 23 1000  
הפלט בסעיף 2 יהיה 6 ימים גשומים ו-7 ימים ללא גשם, והפלט בסעיף 3 יהיה 4 ימים.

## פתרון שאלה 8

א נגדיר מערך a[] המציין תקופה בת 100 ימים

```
#include <stdio.h>
void howManyRainyDays(int b[]);
void maxDaysWithOutRain(int c[]);
void main()
{
int a[100],i=0,noRainDays,rainDays,flag=1;
while(flag) // קליטת נתונים למערך עד שהמשתמש יקיש 1000
{
printf("Please Enter amount of rain for day number %d : ",i+1);
scanf("%d",&a[i]);
if(a[i]==1000)
```

```
flag=0;
i++;
}
howManyRainyDays(a); // פונקציה שתמצא כמה ימי גשם היו וכמה לא
maxDaysWithOutRain(a); // פונקציה לכמה ימים רצופים ללא גשם
}
```

ב.

```
// פונקציה שתמצא כמה ימי גשם היו
void howManyRainyDays(int b[])
{
    int rainyDays=0, i=0;
    for ( ; b[i]!=1000;i++)
        if(b[i]!=0)
            rainyDays++;
    printf("\nIn this period of time there are %d rainy
days",rainyDays);
    printf("\nIn this period of time there are %d days with no
rain",i-rainyDays);
}
```

ג.

```
// פונקציה שתמצא מקסימום ימים רצופים ללא גשם
void maxDaysWithOutRain(int c[])
{
    int maxDays=0,temp=0,i;
    for(i=0;c[i]!=1000;i++)
    {
        if(c[i]==0)
            temp++;
        else if(temp>maxDays)
        {
            maxDays=temp;
            temp=0;
        }
    }
    printf("\nmaximum days without RAIN : %d days",maxDays);
}
```