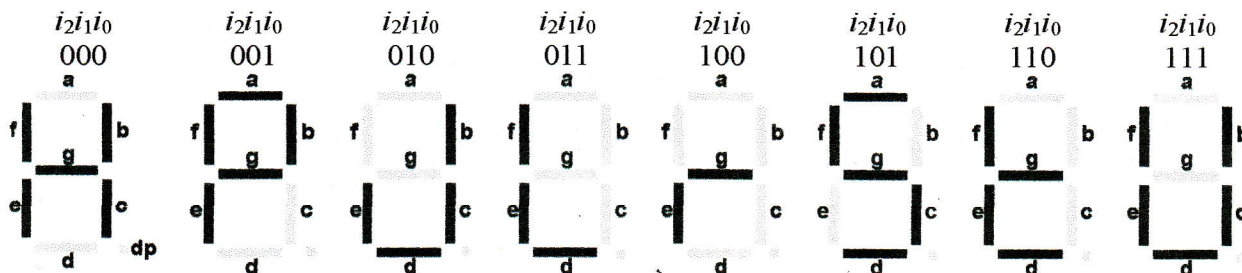


### Problema 13.

Să se proiecteze un microsistem bazat pe microcontrolerul I8051 care calculează funcții logice. Variabilele de intrare sunt  $i_0 - i_{10}$  și se citesc prin intermediul a două porturi de intrare. Funcțiile logice  $a, b, c, d, \dots, f, g$  și  $dp$  reprezintă segmentele unui afișor cu 7 segmente cu anod comun de tip LTS546AG. Funcțiile  $a, b, c, d, \dots, f$  și  $g$  (fără  $dp$ ) depind numai de variabilelor  $i_2 i_1 i_0$  și sunt definite conform tabelului următor:



Afișorul este controlat prin intermediul unui port de ieșire.

Punctul zecimal se folosește pentru afișarea funcției:

$$f(i_3, \dots, i_{10}) = i_3 \oplus i_4 \oplus i_5 \oplus i_6 \oplus i_7 \oplus i_8 \oplus i_9 \oplus i_{10}$$

Dacă valoarea lui  $f$  este ,1'  $dp$  se va aprinde iar dacă  $f$  este ,0'  $dp$  se va stinge.

### I- 3 puncte

Se cere să se prezinte schema simplificată a microsistemului (cu magistrale, cf. prelegere 1) care conține cele trei porturi, de 8 biți fiecare: două porturi de intrare pentru citirea variabilelor booleene  $i_0 - i_{10}$  și un port de ieșire pentru afișor. Schema va preciza ce bit și ce port de intrare se folosește pentru fiecare variabilă de intrare. Se va specifica complet modul de conectare al afișorului 7 segmente cu toate componentele suplimentare care se consideră necesare. 477f

Porturile se decodifică incomplet în zona de adrese  $x:4000h - x:43ffh$ . Logica de decodificare se va figura ca un bloc. Nu este necesară implementarea acesteia, fiind suficientă specificarea formelor booleene minime.

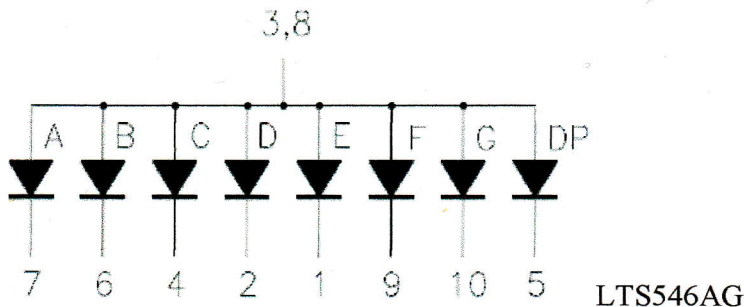
Pentru orice port de ieșire se va folosi un registru paralel-paralel pe 8 biți de tip 74HCT374. Pentru orice port de intrare se vor folosi 8 drivere 3-state încapsulate într-un singur IC de tip 74HCT541. Se preferă reprezentarea generică, conform anexei.

### II- 2 puncte

Să se scrie programul C care calculează valorile funcțiilor  $a, b, c, d, \dots, f$  și  $g$  (fără  $dp$ ) și le afișează conform tabelul de mai sus.

### III- 4 puncte

Să se adauge/modifice programul C de la punctul II pentru a calcula și afișa  $f(i_3, \dots, i_{10}) = i_3 \oplus i_4 \oplus i_5 \oplus i_6 \oplus i_7 \oplus i_8 \oplus i_9 \oplus i_{10}$  prin intermediul punctului zecimal. Folosiți `bdata` și `sbit`. Funcționalitatea implementată la punctul II nu trebuie alterată.



# Problema 13.

Variabile de intrare  $i_0-i_{10} \rightarrow 2$  porturi de intrare,

Fct. logice a b c d e f g  $\rightarrow$  seg unui afisor 7 seg,  
a b c d e f g (fără dp) depind numai de var  $i_2, i_1, i_0$

$$\begin{aligned} 4000h &= 0100 \ 0000 \ 0000 \ 0000 \\ 47F7h &= 0100 \ 0111 \ 1111 \ 1111 \end{aligned}$$

$$A_{15} + \bar{A}_{14} + A_{13} + A_{12} + A_{11} + \dots + A_0$$

PORT-OUT<sub>1</sub>

$$CLKPP_{PORT-OUT_1} = \bar{A}_{15} + A_{14} + A_{13} + A_{12} + A_{11} + A_0 + WR\#$$

$$CLKPP_{PORT-OUT_2} = A_{15} + \bar{A}_{14} + A_{13} + A_{12} + A_{11} + \bar{A}_0 + WR\#$$

$$OETS_{PORT_{IN_1}} = A_{15} + \bar{A}_{14} + A_{13} + A_{12} + A_{11} + A_0 + RD\#$$

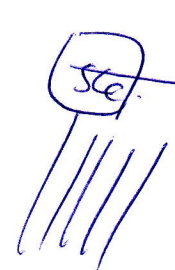
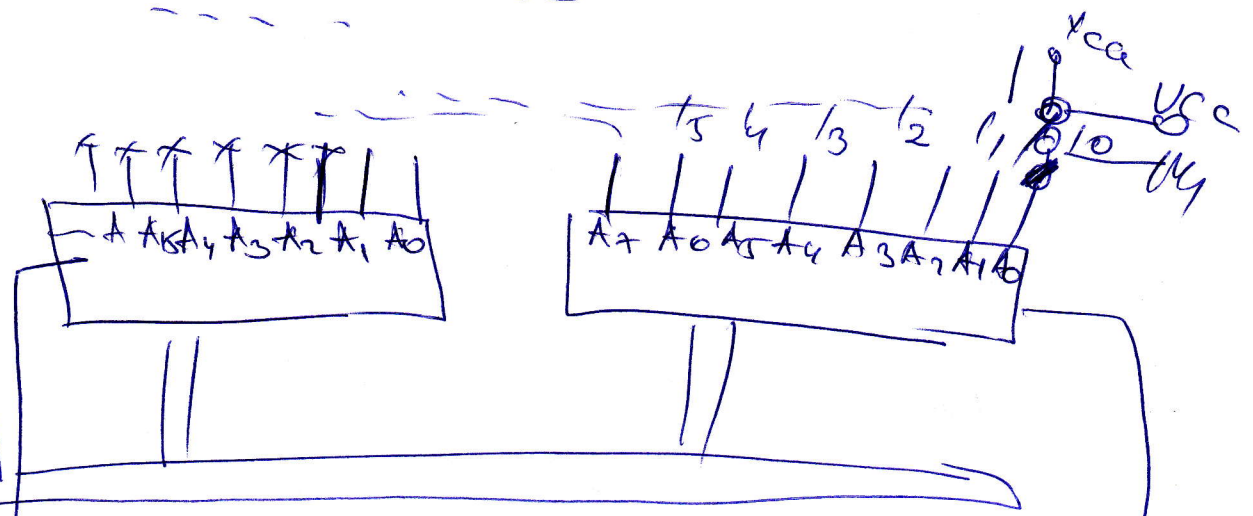
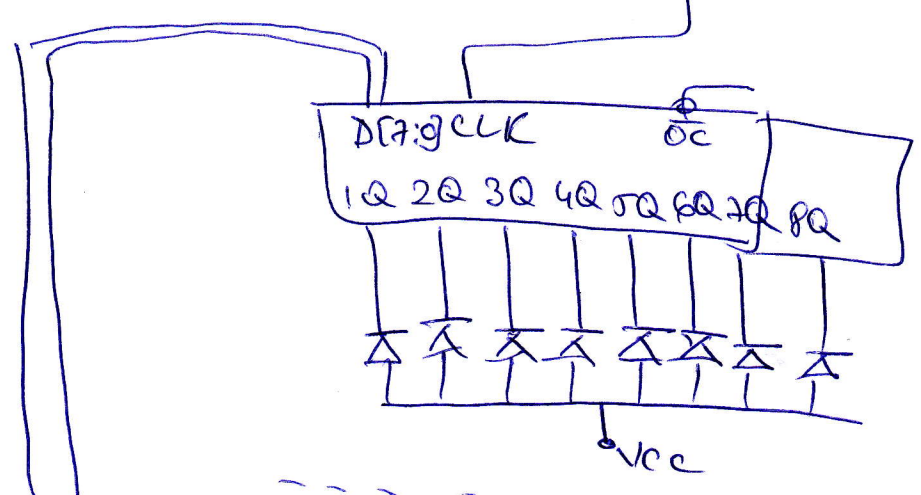
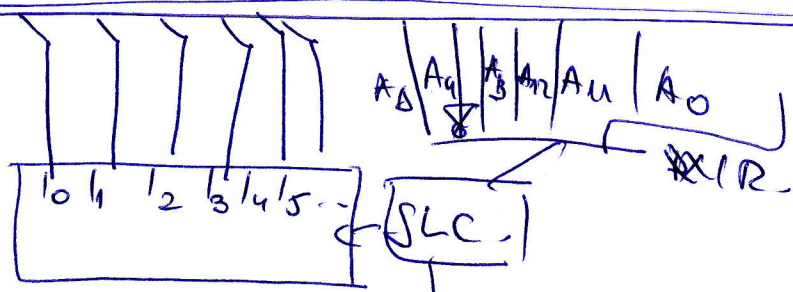
PORT-OUT<sub>1</sub> : 4000h.

PORT-OUT<sub>2</sub> : 4001h.

PORT-IN<sub>1</sub> : 4000h.

|   | $i_0$ | $i_1$ | $i_2$ | a | b | c | d | e | f | g | dp | HEXA           |
|---|-------|-------|-------|---|---|---|---|---|---|---|----|----------------|
| 0 | 0     | 0     | 0     | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1  | <del>91h</del> |
| 1 | 0     | 0     | 1     | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1  | 31h.           |
| 2 | 0     | 1     | 0     | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1  | 31h.           |
| 3 | 0     | 1     | 1     | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1  | 87h.           |
| 4 | 1     | 0     | 0     | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1  | E3h.           |
| 5 | 1     | 0     | 1     | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1  | 75h.           |
| 6 | 1     | 1     | 0     | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1  | 49h.           |
| 7 | 1     | 1     | 1     | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1  | E1h.           |
| 8 | 1     | 1     | 1     | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1  | 83h.           |





```

// include <absacc.h>
#define PORT_OUT1 0x4000
#define PORT_OUT2 0x4001
#define PORT_OUT 0x4000

```

```

main()
{
    char inputs;
    char i0;
    char i1;
    char i2;
    char temp = 0;
    while (1)
    {

```

```

        inputs = BYTE(PORT_IN);
        i0 = inputs & 1; // i0 = 00000000d0
        i1 = (inputs >> 1) & 1;
        i2 = (inputs >> 2) & 1;

```

```

        if (i0 == 0 && i1 == 0 && i2 == 0)

```

```

            outs = 0x91;

```

```

            BYTE(PORT_OUT1) = outs;

```

```

        if (i0 == 0 && i1 == 0 && i2 == 1)

```

```

            outs = 0x31;

```

```

            BYTE(PORT_OUT) = outs;

```

```

        if (i0 == 1 && i1 == 1 && i2 == 1)

```

```

            outs = 0x83;

```

```

            BYTE(PORT_OUT) = outs;
        }
    }

```

III

char bdata inputs[2]

short i0 = inputs[0] ^ 0,

short i1 = inputs[1] ^ 1,

short i2 = inputs[0] ^ 2,

short i3 = inputs[1] ^ 0,

short i4 = inputs[1] ^ 1,

short i10 = inputs[1] ^ 2,

int f(short i3, —, short i0)

{  
if (i3 ^ i4 — ^ i10 == 1) return 1;  
else return 0;  
}

main()

{

while

// — — —

//

xBYTE[PORT\_OUT] = outs + (f(i3, i4, — i10) << 2)