

Problema 7.

Să se proiecteze un microsistem bazat pe microcontrolerul I8051 care controlează un sistem de afișare alcătuit din afișoare „matrice de puncte” cu LEDuri, de tip LTP-1057AE, cu următoarele specificații:

- Afișorul va afișa doar caracterele ASCII cu codurile cuprinse între 20h (Space) -7fh. Configurația LED-urilor pentru fiecare caracter este prezentată în anexă. Spațiu înseamnă toate segmentele stinse.
- Pentru cod și date se folosește numai memoria internă.

I

Să se cupleze la microcontrolerul I8051 UN SINGUR afișor LTP-1057AE. Afișorul va fi controlat folosind tehnica multiplexării, prezentată în prelegerea 4, paragraful „Sistemul de afișare multiplexat”.

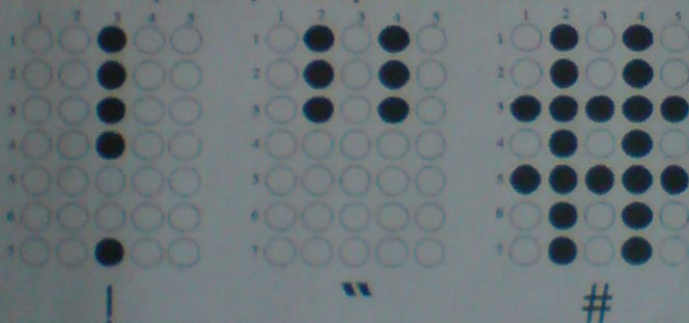
Se cere schema simplificată a microsistemului format din I8051, afișor, porturile necesare pentru conectarea acestuia și logica de decodificare a porturilor. Porturile necesare se vor decodifica incomplet în zona de adrese x:4000h-x:7FFFh. Logica de decodificare se va figura ca un bloc. Nu este necesară implementarea acesteia, fiind suficientă specificarea forme booleene minime pentru fiecare selecție de port.

Pentru orice port de ieșire se va folosi un registru paralel-paralel pe 8 biți de tip 74HCT374. Se preferă reprezentarea generică, conform anexei.

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. Schema – 1 punct, logica de decodificare – 1 punct.

II- 3 puncte

Funcția care face afișarea se va numi *disp_Mux1x5x7()*. Afișarea trebuie să se facă de 60 de ori pe secundă cu precizie de cel puțin 1%. Frecvența ceasului microcontrolerului I8051 este 12 MHz. Se vor respecta și celelalte două reguli enunțate în curs. Se va defini configurația punctelor doar pentru caracterele *space*, *!* și *#* iar apoi se va continua cu, conform figurii următoare.



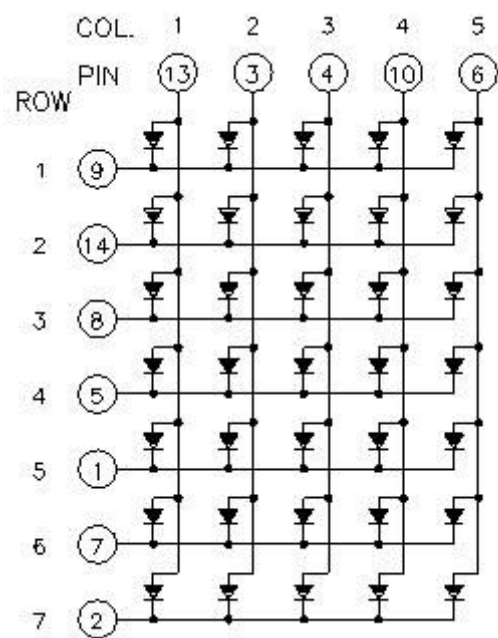
Să se scrie programul principal care face toate configurările necesare. De asemenea, programul principal (sau funcții apelate din programul principal) scrie codul caracterului ce trebuie afișat într-o variabilă globală iar *disp_Mux1x5x7()* o afișează.

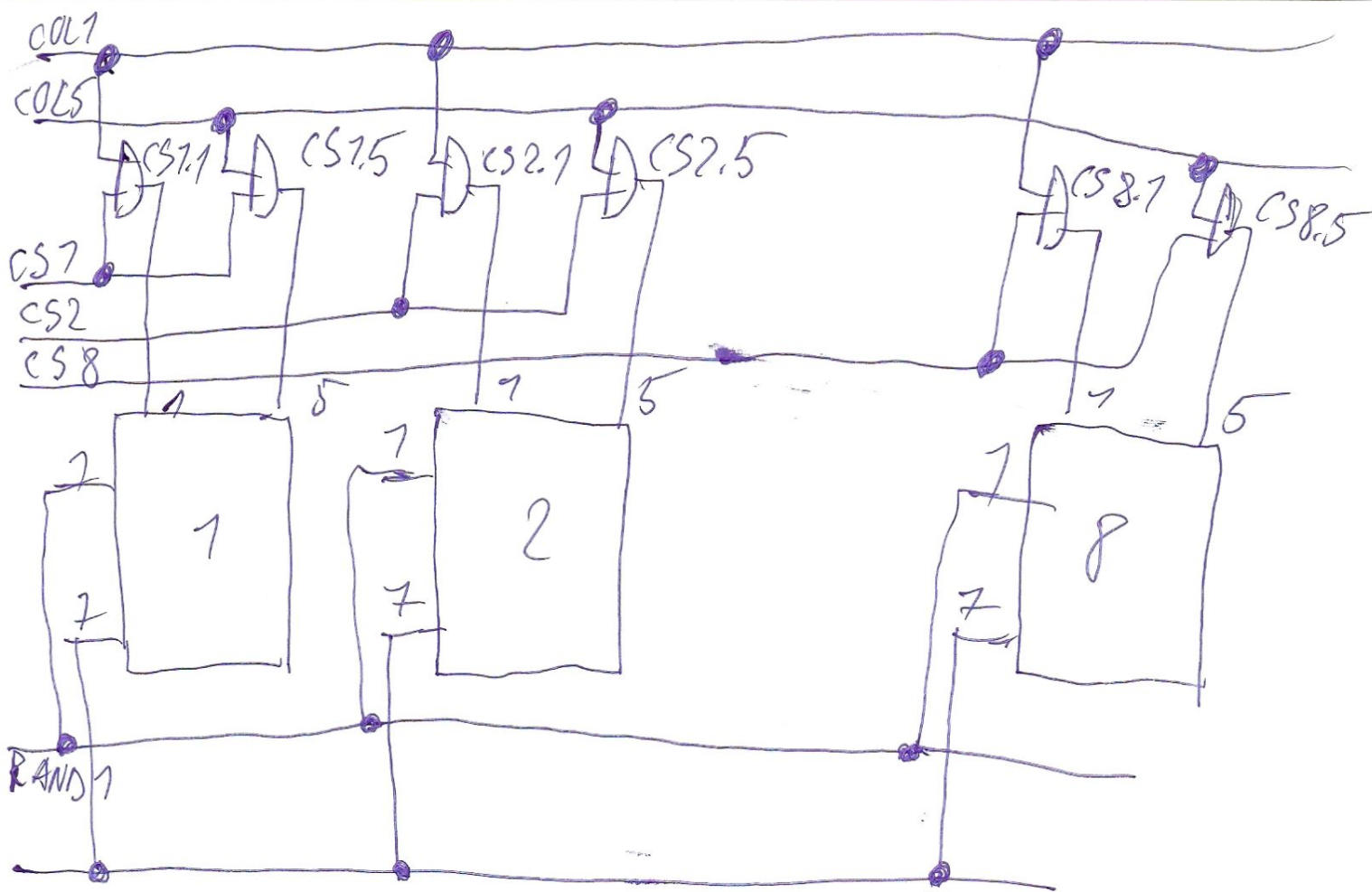
III-1 punct

Să se adauge porturile necesare pentru cuplarea a încă 7 afișoare LTP-1057AE. Se va preciza tipul porturilor și se va explica modul în care sunt folosite aceste porturi. Nu se cere desenarea schemei. De asemenea se va preciza ce adrese se alocă acestor porturi, fără a mai fi necesară specificarea funcțiilor de selecție.

IV-3 puncte

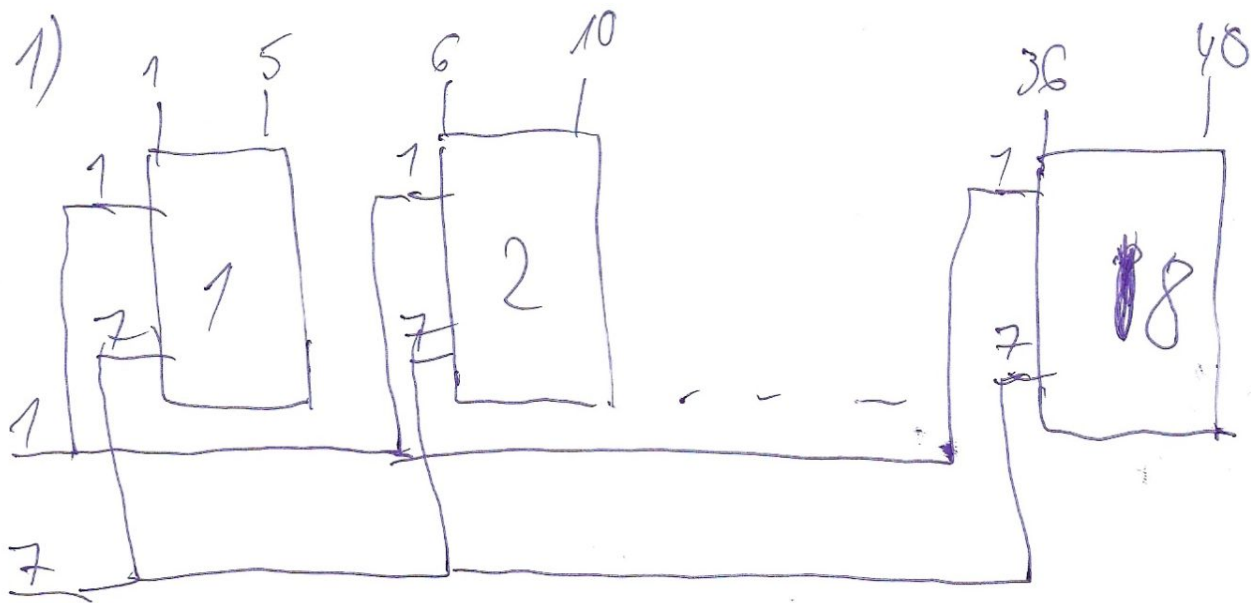
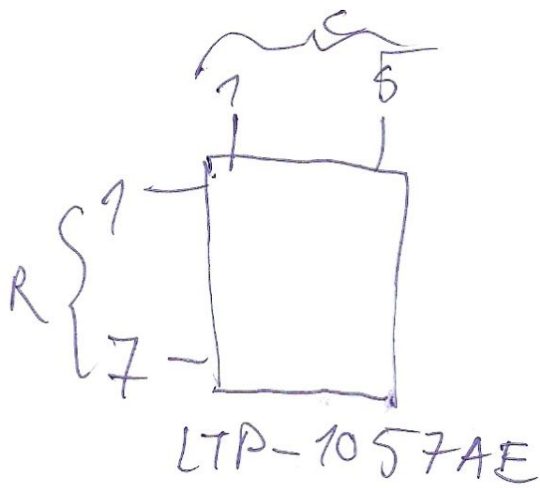
Să se scrie funcția *disp_Mux8x5x7()* care afișează 8 caractere memorate într-un buffer alocat global. Acest buffer servește ca mijloc de comunicare între *disp_Mux8x5x7()* și restul funcțiilor.





RAND 7

RAND 1..7 \Rightarrow PORT 1
 COL 1..5 \Rightarrow PORT 2
 CS 1..8 \Rightarrow PORT 3

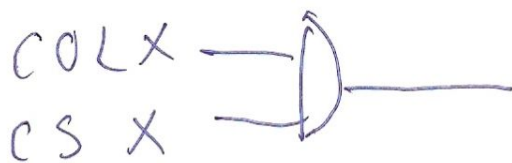


1 PORT pt rânduri; 7 rânduri

5 x 8 bits \Rightarrow 5 PORT-uri pt coloane, 40 de coloane

\Rightarrow 6 PORT-uri

2) „chip enable”



\Rightarrow 8 CS

$x:4000h - x:7FFFh$

0 1 0 0 0 0 0 0 . . .
0 1 1 1 1 1 1 1 . . .

$$A_{15} + A_{14} -$$

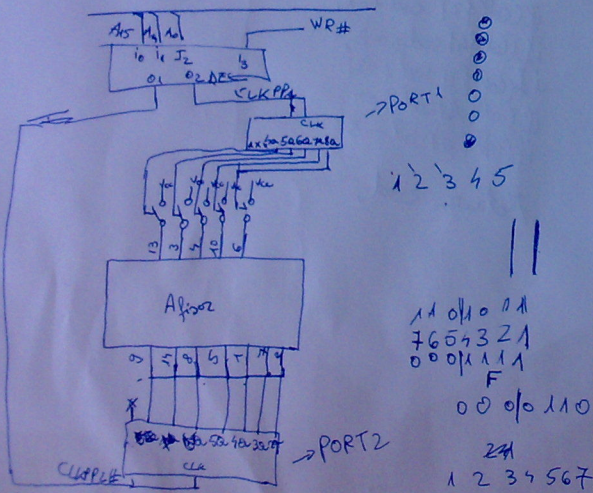
Vom avea 2 porturi de intrare: un port de selectie
si un port segmente

la postul de selectie digit vai lega colosanele iar
la cel selectie segun linie.

part 1: 4000h

$$CLRPP_1 = A_{15} + \overline{A}_{14} + A_0 + WR\#$$

part₂: 400ah

$$CLKPP_2 = A_{15} + \overline{A_{14}} + \overline{A_6} + WR \neq$$


```
#define P_Dig 0x4000
```

```
#define LSEG 0x4001
```

~~code char~~ ~~seg 7[] = {0x06}~~

pentru 1 avem in

PORT 1 0110 000 1B

Part, 1/0 10 1 ¹⁵

code char reg mux[1][2] = { {0x1B, 0x06}, {0x15, 0x08} }

$$\{0115, 0\} \cup \{0, 30\}$$

```
void disp_Mux < 5 x 7 x (char ch) {
```

if (ch == "!") {

```
xBYTE[P_DIG] = Segmax[0][0];
```

$$\text{X BYTE [P_SEG]} = \text{segment} [1] [1];$$
$$\{ \text{if } cch == " " \} \{$$
$$x_{BYTE}[P_DIG] = \text{segment} \times [1][0];$$
$$XBYTE[P-SEF] = \arg \max [1] [1]$$

if $cc_h = \text{"\#"} \{$

$$\text{X BYTE [P-DIG]} = \text{segment} \times 256[0];$$
$$X_{BYTE} [P-SLE \&] = \text{segment} [2] [1];$$
$$\text{BYTE}[P_DI_E] = \text{segmax}[3][0]$$
$$\text{BYTE}[P_SEG] = \text{segment} [3] [0]$$