

Universitatea Politehnica Timișoara  
Facultatea de Automatică și Calculatoare  
Proiectarea Microsistemelor Digitale

## Microsistem cu microprocesorul 8086

Ștefănică Nicoleta Rodica

2023-2024

## **Tema proiectului**

Proiectarea unui microsistem cu următoarea structură:

- unitate centrală cu microprocesorul 8086;
- 128 Ko memorie EPROM, utilizând circuite 27C512;
- 64 Ko memorie SRAM, utilizând circuite 62256;
- interfață serială, cu circuitul 8251, plasată în zona 0650H – 0652H sau 0E50H – 0E52H, în funcție de poziția microcomutatorului S1;
- interfață paralelă, cu circuitul 8255, plasată în zona 0260H– 0266H sau 0360H – 0366H, în funcție de poziția microcomutatorului S2;
- o minitastatură cu 9 contacte; - 24 LED-uri;
- un modul de afișare cu 7 segmente, cu 6 ranguri (se pot afișa maxim 6 caractere hexa simultan);
- un modul LCD, cu 2 linii a câte 16 caractere fiecare, cu o interfață la alegerea studentului.

Toate programele în limbaj de asamblare vor fi concepute sub formă de subrutine. Programele necesare sunt:

- rutinele de programare ale circuitelor 8251 și 8255;
- rutinele de emisie/ recepție caracter pe interfața serială; - rutina de emisie caracter pe interfață paralelă;
- rutina de scanare a minitastaturii;
- rutina de aprindere/ stingere a unui led;
- rutina de afișare a unui caracter hexa pe un rang cu segmente.

# DESCRIERE HARDWARE

## 1. Unitatea centrală

Unitatea centrală este formată din microprocesorul 8086, 3 circuite registru 74x373, 2 circuite 74x245 și un circuit 8284A.

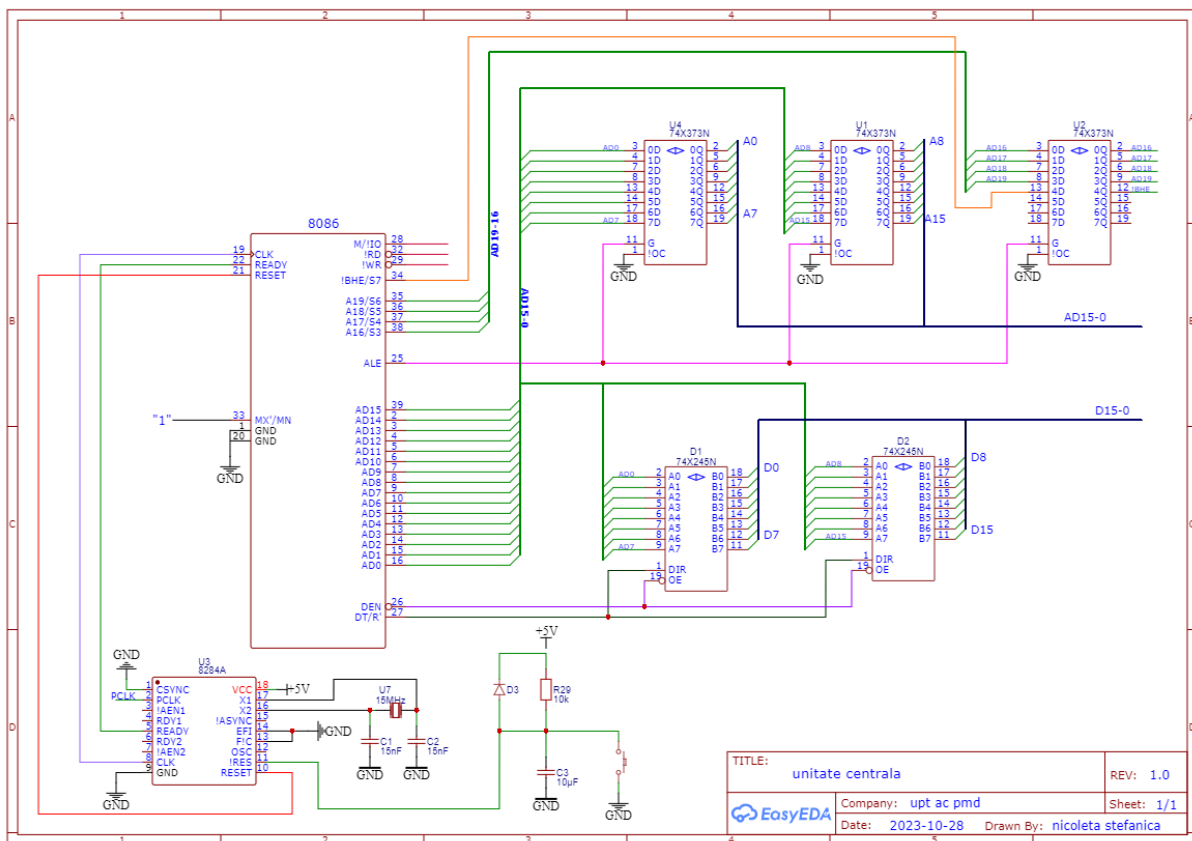
Microprocesorul Intel 8086 este versiunea îmbunătățită a microprocesorului 8085, fiind proiectat în 1976. Acesta utilizează o arhitectură pe 16 biți, putând manipula date pe 16 biți la un moment dat. 8086 folosește 20 de linii de adresă și 16 de linii de date, deci se poate adresa direct până la 1 MB de memorie. Folosește o frecvență de 5MHz, asigurând viteză de lucru eficientă.

Având în vedere că magistrala de date și cea de adrese a microprocesorului folosesc aceleași terminale, este necesară demultiplexarea magistralei cu ajutorul a 3 circuite 74x373, care sunt activate de semnalul ALE, atunci când pe magistrală sunt active adresele.

Cele 2 circuite amplificatoare 74x245 facilitează transferul de date bidirecțional, fiind activate de semnalul !DEN, care validează transferul de date pe magistrală.

Microprocesorul beneficiază de 2 moduri posibile de lucru, iar în cazul de față este folosit modul minim. Acesta este indicat de 1 logic, mod în care procesorul generează el însuși semnalele pentru transferul cu memoria și cu porturile de intrare/ieșire.

Circuitul 8284A este un generator de semnal pentru microprocesorul 8086. 8284A produce semnalul de tact corect, stabilește condițiile de sincronizare și de asemenea furnizează semnal de reset care este necesar pentru a inițializa sau reseta microprocesorul conectat, dar furnizează și semnalul PCLK pentru interfața serială, având o valoare mai mică decât CLK. Semnalul ~RST este activ pe 0 la momentul conectării la sursa de alimentare, sau în urma apăsării butonului, iar semnalul de reset se obține prin sincronizarea lui ~RST cu semnalul de tact.



## 2. Conectarea memoriilor

Pentru memoria EPROM de 128 KB vom utiliza 2 circuite 27C512, fiecare având o capacitate de 64 KB, iar pentru memoria SRAM de 64 KB vom utiliza 2 circuite 62256, fiecare cu o capacitate de 32 KB.

Pentru EPROM am ales adresa de început 00000H și cea de sfârșit 1FFFFH, acoperind un număr de 20000H locații, iar pentru SRAM adresa de început 20000H și cea de final 2FFFFH, acoperind 10000H locații.

EPROM:

$$128\text{Ko} = 2^{17} \text{ B} = 0010\ 0000\ 0000\ 0000\ 0000 \text{ de locații}$$

00000H -> 1FFFFH adică 20000H locații

SRAM:

$$64\text{Ko} = 2^{16} \text{ B} = 0001\ 0000\ 0000\ 0000\ 0000 \text{ de locații}$$

20000H -> 2FFFFH adică 10000H locații

	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
EPROM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
SRAM	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Ecuatiile semnalelor de selecție:

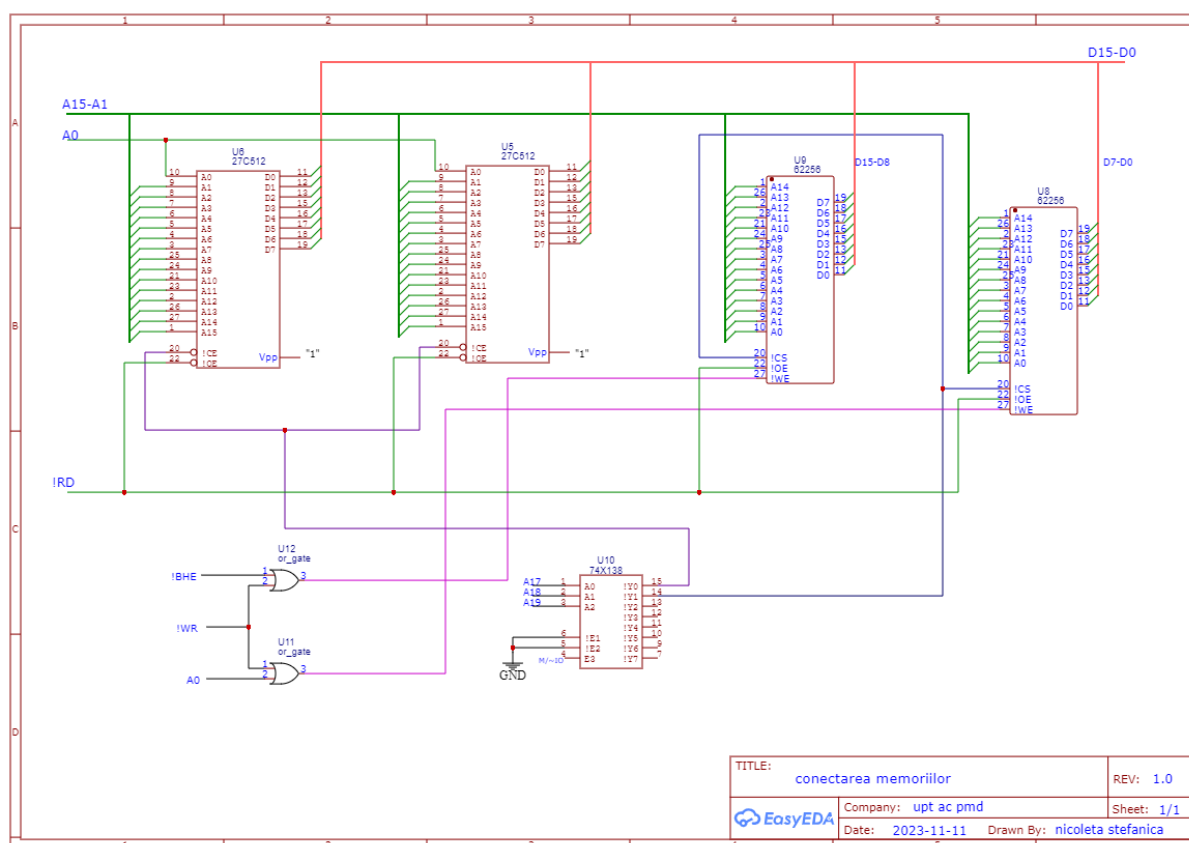
EPROM:

$$\text{SEL1} = \sim A19 * \sim A18 * \sim A17$$

SRAM:

$$\text{SEL2} = \sim A19 * \sim A18 * A17$$

Se realizează o decodificare incompletă, având în vedere că și bitul 16 rămâne neschimbat în cazul SRAM.



### 3. Interfețele serială și paralelă. Decodificatorul de porturi

Folosim un decodificator 4x16 atât pentru interfețe cât și pentru periferice, în funcție de nibble-ul format din biții A11, A10, A9, A8.

Pentru interfața serială, va fi aleasă una din zonele de memorie 0650H – 0652H sau 0E50H – 0E52H, în funcție de poziția microcomutatorului S1. Pentru cea paralelă, se va alege dintre zonele 0260H– 0266H sau 0360H – 0366H, în funcție de poziția microcomutatorului S2.

		A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
Seriala s1	0650H	0	0	0	0	0	1	1	0	0	1	0	1	0	0	0	0
	0652H	0	0	0	0	0	1	1	0	0	1	0	1	0	0	1	0
Seriala s2	0E50H	0	0	0	0	1	1	1	0	0	1	0	1	0	0	0	0
	0E52H	0	0	0	0	1	1	1	0	0	1	0	1	0	0	1	0
Paralela s1	0260H	0	0	0	0	0	0	1	0	0	1	1	0	0	0	0	0
	0262H	0	0	0	0	0	0	1	0	0	1	1	0	0	0	1	0
	0264H	0	0	0	0	0	0	1	0	0	1	1	0	0	1	0	0
	0266H	0	0	0	0	0	0	1	0	0	1	1	0	0	1	1	0
Paralela s2	0360H	0	0	0	0	0	0	1	1	0	1	1	0	0	0	0	0
	0362H	0	0	0	0	0	0	1	1	0	1	1	0	0	0	1	0
	0364H	0	0	0	0	0	0	1	1	0	1	1	0	0	1	0	0
	0366H	0	0	0	0	0	0	1	1	0	1	1	0	0	1	1	0

Ecuatiile de selecție pentru interfața serială

$$S51\_1 = \sim A11 * A10 * A9 * \sim A8 \Rightarrow Y6$$

$$S51\_2 = A11 * A10 * A9 * \sim A8 \Rightarrow Y14$$

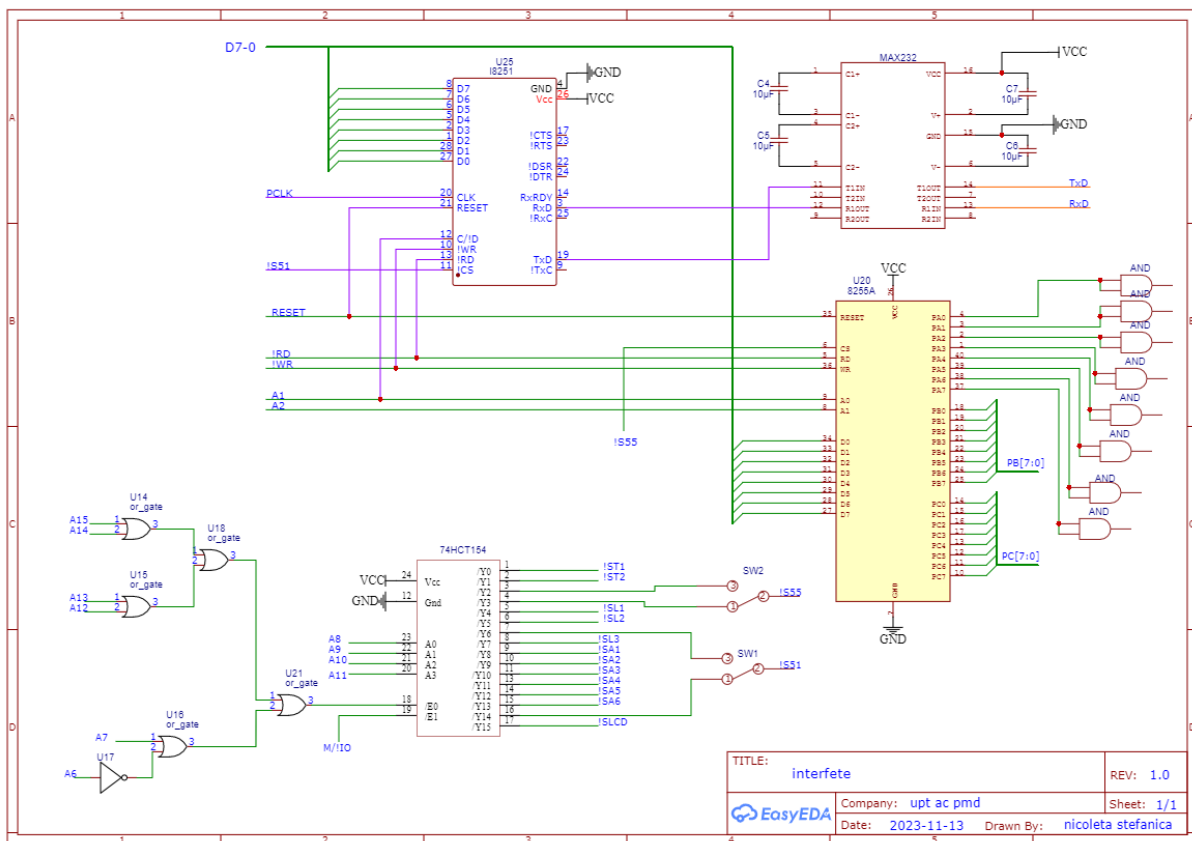
Ecuatiile de selecție pentru interfața paralelă

$$S55\_1 = \sim A11 * \sim A10 * A9 * \sim A8 = Y2$$

$$S55\_2 = \sim A11 * \sim A10 * A9 * A8 = Y3$$

Condiție de activare a ieșirilor:

$$E0 = (A15 + A14) + (A13 + A12) + (A7 + \sim A6)$$



#### 4. Conectarea afişajelor şi a minitastaturii

Având în vedere că în urma decodificării necesare interfeţelor cu un decodificator 4 la 16, am rămas cu 12 ieşiri disponibile, le-am folosit şi pentru decodificarea perifericelor. Am ales adresele arbitrar, dar ținând cont de biții A11, A10, A9 şi A8, lucru vizibil în tabelul de mai jos.

	adresele	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
ST1	0040H	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
ST2	0140H	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0
SL1	0440H	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0
SL2	0540H	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0	0
SL3	0740H	0	0	0	0	0	1	1	1	0	1	0	0	0	0	0	0
SA1	0840H	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0
SA2	0940H	0	0	0	0	1	0	0	1	0	1	0	0	0	0	0	0
SA3	0A40H	0	0	0	0	1	0	1	0	0	1	0	0	0	0	0	0
SA4	0B40H	0	0	0	0	1	0	1	1	0	1	0	0	0	0	0	0
SA5	0C40H	0	0	0	0	1	1	0	0	0	1	0	0	0	0	0	0
SA6	0D40H	0	0	0	0	1	1	0	1	0	1	0	0	0	0	0	0
SLCD	0F40H	0	0	0	0	1	1	1	1	0	1	0	0	0	0	0	0

## Ecuatiile de selecție

- tastatura

$$ST1 = \sim A15 * \sim A14 * \sim A13 * \sim A12$$

$$ST2 = \sim A15 * \sim A14 * \sim A13 * A12$$

- leduri

$$SL1 = \sim A15 * A14 * \sim A13 * \sim A12$$

$$SL2 = \sim A15 * A14 * \sim A13 * A12$$

$$SL3 = \sim A15 * A14 * A13 * A12$$

- afisaje

$$SA1 = A15 * \sim A14 * \sim A13 * \sim A12$$

$$SA2 = A15 * \sim A14 * \sim A13 * A12$$

$$SA3 = A15 * \sim A14 * A13 * \sim A12$$

$$SA4 = A15 * \sim A14 * A13 * A12$$

$$SA5 = A15 * A14 * \sim A13 * \sim A12$$

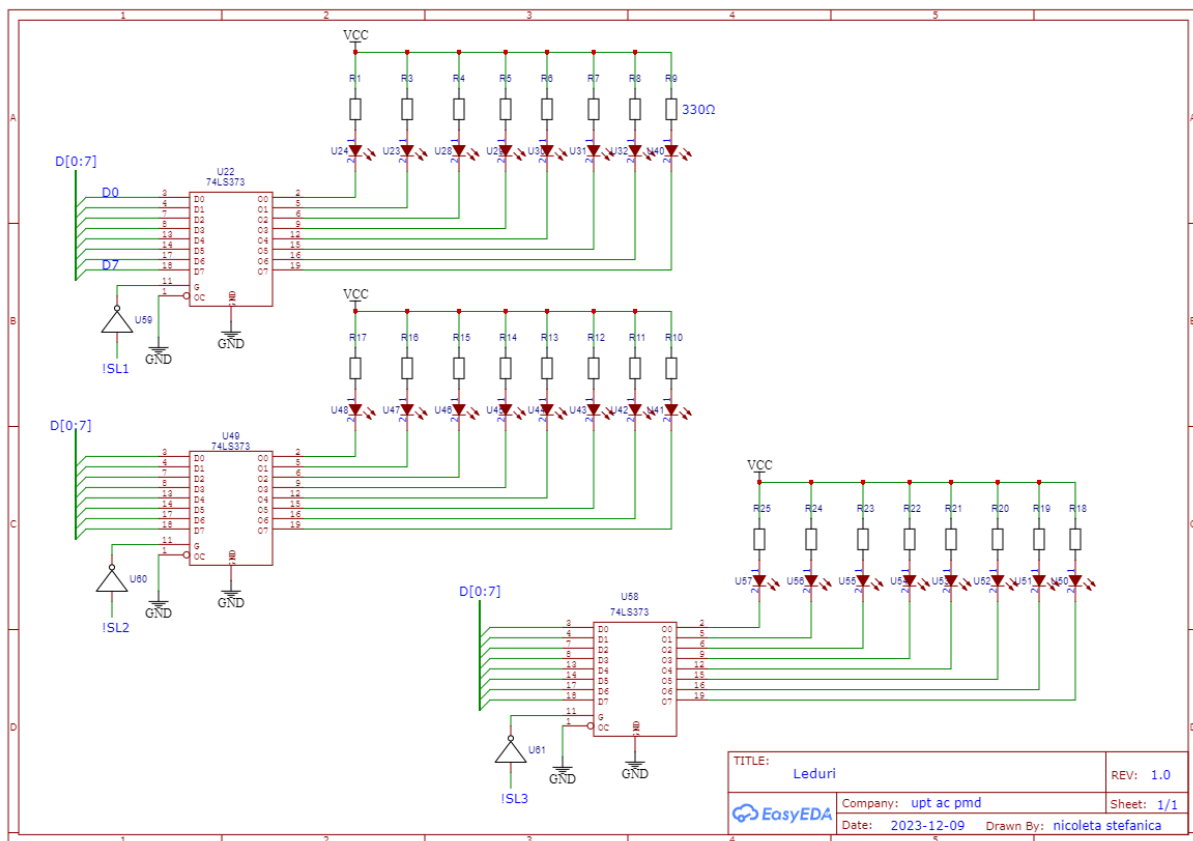
$$SA6 = A15 * A14 * \sim A13 * A12$$

- LCD

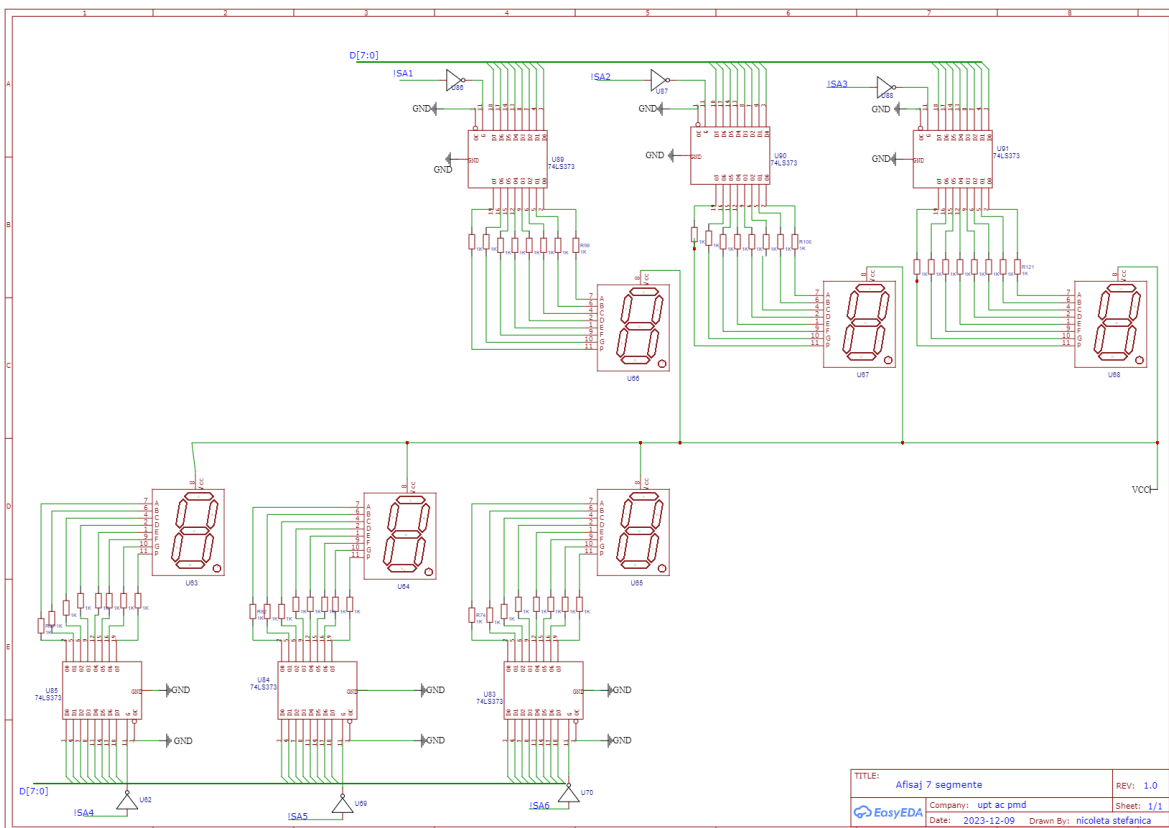
$$SLCD = A15 * A14 * A13 * A12$$



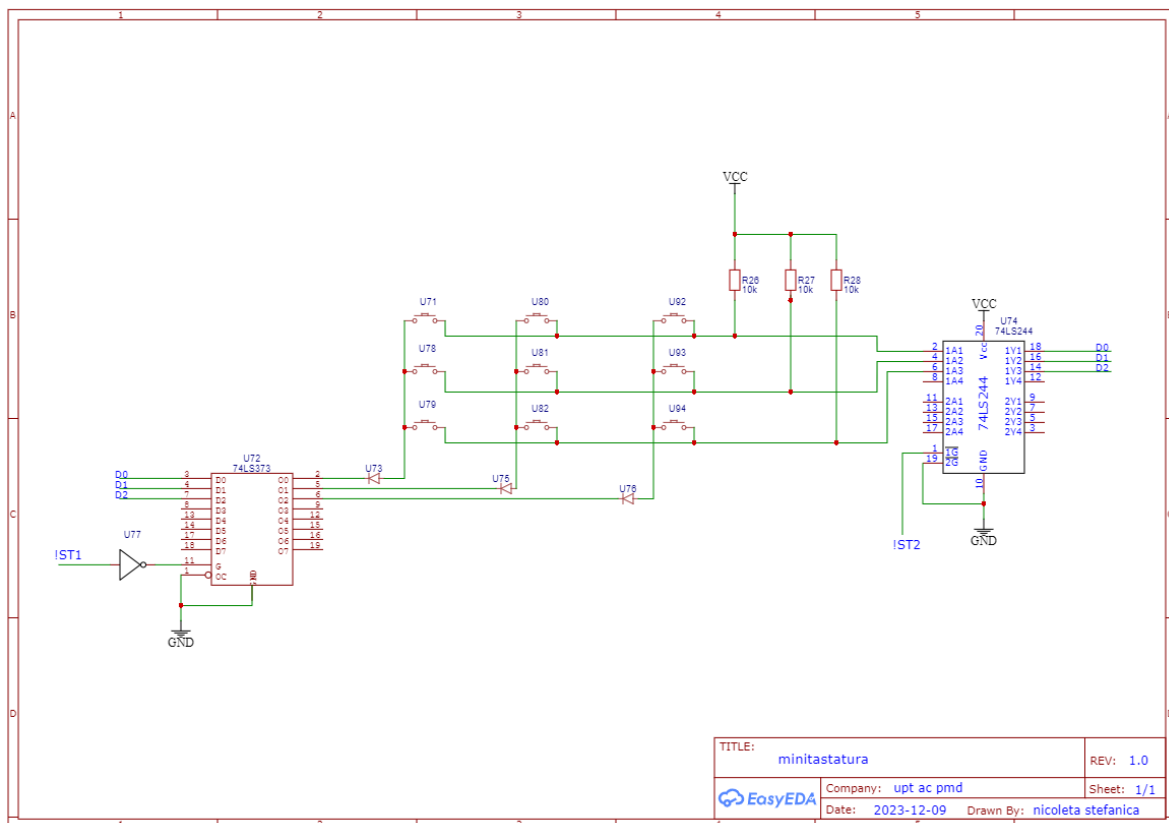
## Conectarea celor 24 de leduri:



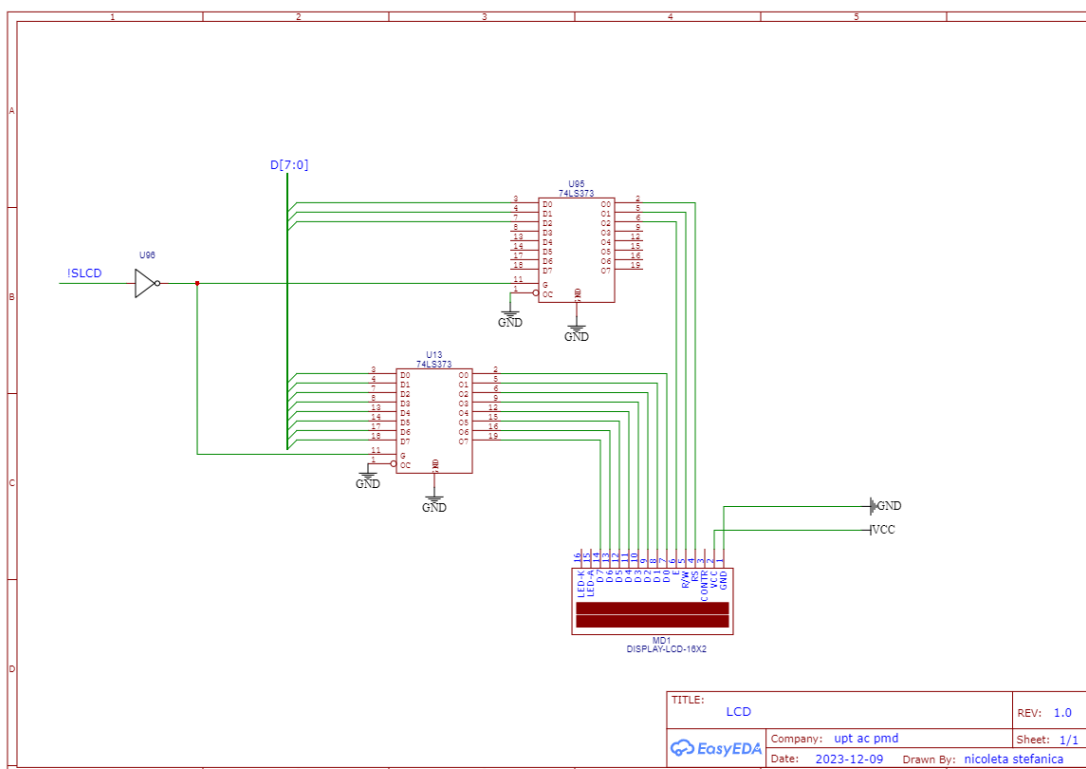
## Conectarea afişajelor cu 7 segmente, cu 6 ranguri:



### Conectarea minitastaturii:



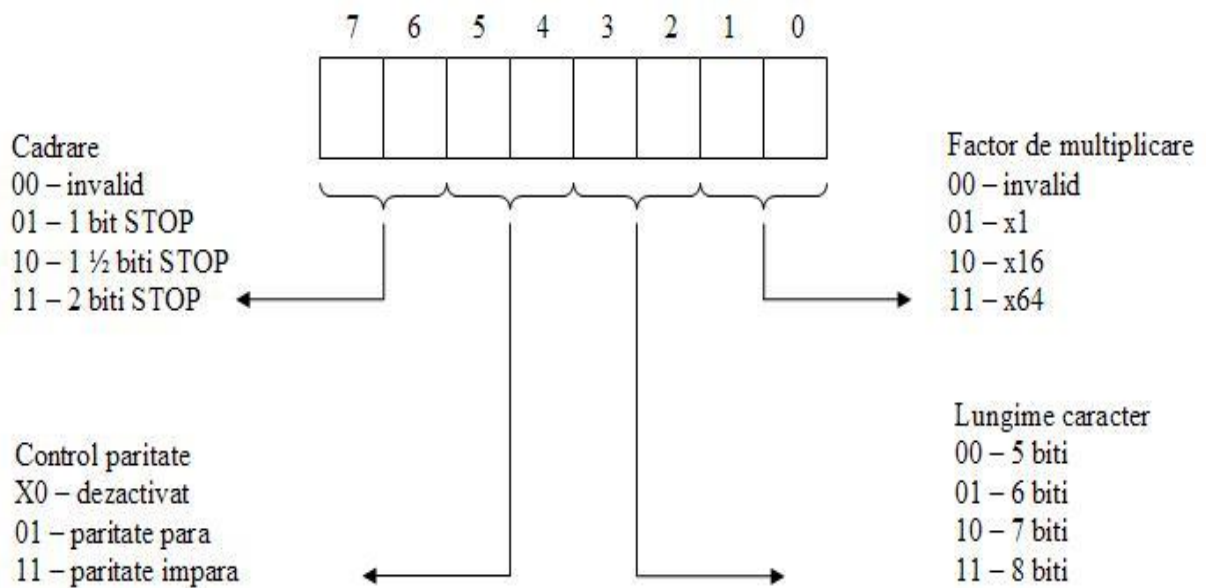
### Conectarea modulului LCD:



# RUTINE DE PROGRAMARE

## 1. Programare 8251 (interfața serială)

Structura cuvântului de mod:



11 – biți de stop

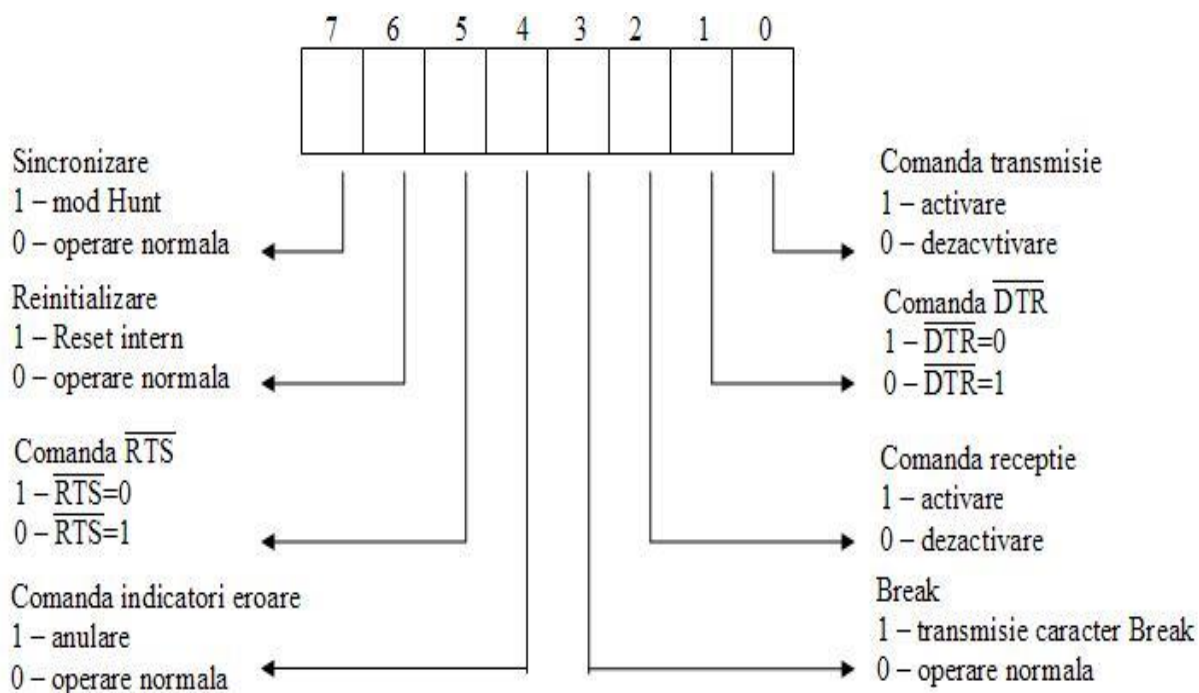
00 – fără paritate

11 – 8 biți de date

10 – factor de multiplicare 16

- 1100 1110 => 0CEH cuvânt de mod

Structura cuvântului de comandă:



Bitul de pe poziția 0: comanda transmisie activa

Bitul de pe poziția 1: comanda !DTR = 1

Bitul de pe poziția 2: comanda receptie activa

Bitul de pe poziția 3: break – operare normală

Bitul de pe poziția 4: anulare indicatori eroare

Bitul de pe poziția 5: comanda !RTS = 1

⇒ 0001 0101 ⇒ 015H cuvânt de comandă

În funcție de poziția comutatorului, se schimbă adresa de la anumite instrucțiuni, alternativa am trecut-o în paranteză.

Rutina de programare:

MOV AL, 0CEH ; cuvânt de mod

OUT 0652H, AL ( OUT 0E52H, AL )

MOV AL, 15H ; cuvânt de comandă

OUT 0652H, AL ( AUT 0E52H, AL )

### Rutina de transmisie caracter:

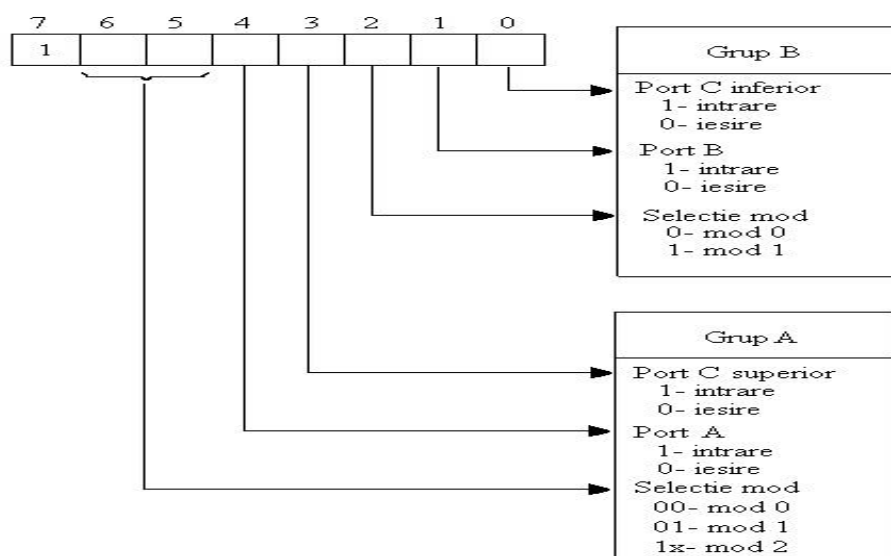
```
TR:  IN AL, 0652H (IN AL, 0E52H) ; citire și testare rang TxRDY din cuvântul de stare  
      RCR AL, 1  
      JNC TR  
      MOV AL, CL ; se preia data din registrul CL  
      OUT 0650H, AL ( OUT 0E50H, AL )  
      RET
```

### Rutina de recepție caracter:

```
REC: IN AL, 0652H (IN AL, 0E52H) ;citire și testare rang RxRDY din cuvântul de stare  
      RCR AL, 2  
      JNC REC  
      IN AL, 0650H ( IN AL, 0E50H ) ; se preia data de la 8251  
      MOV CL, AL ; se depune data în registrul CL  
      RET
```

## 2. Programare 8255 (interfața paralelă)

Structura cuvântului de comandă:



Biții de pe pozițiile 6-5: modul 0, mod intrare/ieșire

Bitul de pe poziția 4: 0, ieșire pentru portul A

Bitul de pe poziția 3: 0, ieșire pentru portul C superior

Bitul de pe poziția 2: 0, mod intrare/ieșire

Bitul de pe poziția 1: 0, ieșire pentru portul B

Bitul de pe poziția 0: 1, intrare pentru portul C inferior

⇒ 1000 0001 ⇒ 81H

Adresele:

0260H (0360H) – portul A

0262H (0362H) – portul B

0264H (0364H) – portul C

0266H (0366H) - RCC

Rutina de programare:

MOV AL, 81H

OUT 0266H, AL ( OUT 0366H, AL )

Rutina de emisie caracter pe interfață paralelă:

PAR: IN AL, 0264H ( IN AL, 0364H ) ; citire și testare BUSY

RCR AL, 1

JC PAR

MOV AL, CL ; se preia caracterul din registrul CL

MOV DX, 0260H ( MOV DX, 0360H )

OUT DX, AL

OR AL, 01H

MOV DX, 0262H ( MOV DX, 0362H )

OUT DX, AL ; /STB = 1

```

AND AL, 00H

OUT DX, AL      ; /STB = 0

OR AL, 01H

OUT DX,AL       ; /STB = 1

RET

```

### 3. Rutina de scanare a minitastaturii

; se pune 0 pe prima coloană și se verifică dacă s-au acționat tastele 1, 4, 7

```

REIA: MOV AL, 0FEH

      OUT 0040H, AL

      IN AL, 0140H

      AND AL, 01H

      JZ TASTA1

      IN AL, 0140H

      AND AL, 02H

      JZ TASTA4

      IN AL, 0140H

      AND AL, 04H

      JZ TASTA7

```

; se pune 0 pe a 2-a coloană și se verifică dacă s-au acționat tastele 2, 5, 8

```

MOV AL, 0FDH

OUT 0040H, AL

IN AL, 0140H

AND AL, 01H

JZ TASTA2

IN AL, 0140H

```

AND AL, 02H

JZ TASTA5

IN AL, 0140H

AND AL, 04H

JZ TASTA8

; se pune 0 pe a 3-a coloană și se verifică dacă s-au acționat tastele 3, 6, 9

MOV AL, 0FBH

OUT 0040H, AL

IN AL, 0140H

AND AL, 01H

JZ TASTA3

IN AL, 0140H

AND AL, 02H

JZ TASTA6

IN AL, 0140H

AND AL, 04H

JZ TASTA9

; se reia baleierea

JMP REIA

.....

; tratarea acționării tastei 1

TASTA1: CALL DELAY ; se așteaptă stabilizarea contactelor

AST1: IN AL, 0140H ; se citește din nou linia și se așteaptă dezactivarea tastei

AND AL, 01H

JZ AST1

CALL DELAY

; operația corespunzătoare acționării tastei 1



#### 4. Rutina de afișare a unui caracter hexa pe un rang cu segmente

Afișaj cu 7 segmente cu anod comun => activare pe 0. La aceste circuite, toți anozii segmentelor sunt legați împreună la un terminal la care se conectează alimentarea circuitului, iar catodii sunt accesibili la terminale ale circuitului.

Număr de afișat	DP	g	f	e	d	c	b	a	Format hexa
0	1	1	0	0	0	0	0	0	0C0H
1	1	1	1	1	1	0	0	1	0F9H
2	1	0	1	0	0	1	0	0	0A4H
3	1	0	1	1	0	0	0	0	0B0H
4	1	0	0	1	1	0	0	1	099H
5	1	0	0	1	0	0	1	0	092H
6	1	0	0	0	0	0	1	0	082H
7	1	1	1	1	1	0	0	0	0F8H
8	1	0	0	0	0	0	0	0	080H
9	1	0	0	1	0	0	0	0	090H
A	1	0	0	0	1	0	0	0	088H
b	1	0	0	0	0	0	1	1	083H
C	1	1	0	0	0	1	1	0	0C6H
d	1	0	0	1	0	0	0	1	091H
E	1	0	0	0	0	1	1	0	086H
F	1	0	0	0	1	1	1	0	08EH

Afișarea cifrei 0 pe primul rang:

```
MOV AL, 0C0H
```

```
OUT 0840H, AL
```

Afișarea cifrei 1 pe primul rang:

```
MOV AL, 0F9H
```

```
OUT 0840H, AL
```

Pentru afișarea pe celelalte ranguri, ceea ce se schimbă este adresa de la instrucțiunea OUT, astfel:

- Rang 2: OUT 0940H, AL
- Rang 3: OUT 0A40H, AL
- Rang 4: OUT 0B40H, AL
- Rang 5: OUT 0C40H, AL
- Rang 6: OUT 0D40H, AL

## 5. Rutina de aprindere/ stingere a unui led

Ledurile sunt active pe 0 și astfel putem să aprindem câte un led cu ajutorul următoarelor combinații:

LED1 1111 1110

LED2 1111 1101

LED3 1111 1011

LED4 1111 0111

LED5 1110 1111

LED6 1101 1111

LED7 1011 1111

LED8 0111 1111

Exemplu de aprindere al ledului 4

```
MOV AL, F7H
```

```
MOV DX, 0440H
```

```
OUT DX, AL
```

Stingerea ledului

```
MOV AL, 0FFH
```

```
MOV DX, 0440H
```

```
OUT DX, AL
```

Exemplu de aprindere al ledului 10

```
MOV AL, FDH
```

```
MOV DX, 0540H
```

```
OUT DX, AL
```

Se pot aprinde mai multe leduri în același timp dacă se scriu combinații cu mai multe leduri active, adică setând 0 pe pozițiile corespunzătoare lor.

## **BIBLIOGRAFIE**

1. Proiectarea Microsistemelor Digitale - curs
2. <https://www.electronicwings.com/sensors-modules/lcd-16x2-display-module>
3. <https://electronicsdesk.com/8284-clock-generator.html>
4. <https://www.geeksforgeeks.org/pin-diagram-8086-microprocessor/>
5. <https://www.electronicshub.org/7-segment-display-pinout/>
6. <https://www.engineersgarage.com/max232-pin-out-and-description/>