

Proiectarea cu Microprocesoare

# **Microsistem cu microprocesorul 8086**

Universitatea Politehnica Timișoara  
Facultatea de Automatică și Calculatoare

Perhaiță Daniel  
Anul universitar 2025-2026

# Tema proiectului

Lucrarea constă în proiectarea unui microsistem în software-ul de CAD EasyEDA după următoarea structură dată drept cerință:

- unitate centrală cu microprocesorul 8086;
- 128 KB memorie EPROM, utilizând circuite 27C512;
- 64 KB memorie SRAM, utilizând circuite 62256;
- interfață serială, cu circuitul 8251, plasată în zona 04D0H – 04D2H sau 05D0H – 05D2H, în funcție de poziția microcomutatorului S1;
- interfață paralelă, cu circuitul 8255, plasată în zona 0250H – 0256H sau 0A50H – 0A56H, în funcție de poziția microcomutatorului S2;
- o minitastatură cu 9 contacte;
- 10 led-uri;
- un modul de afișare cu 7 segmente, cu 8 ranguri (se pot afișa maxim 8 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.

# Unitatea Centrală

Unitatea centrală este formată din:

- Microprocesorul 8086
- Generator de clock 8284A
- Trei circuite registru 74x373
- Doua circuite amplificator/separator bidirectional 74x245

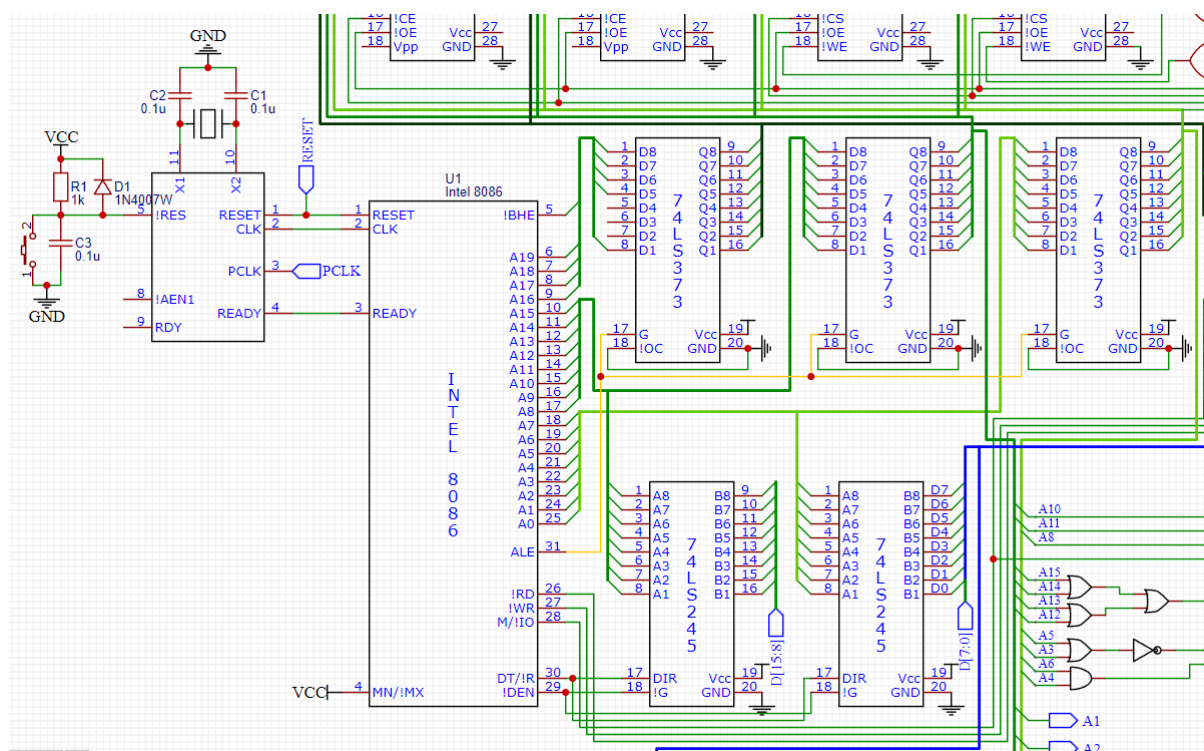


Fig. 1

# Microprocesorul 8086

Intel 8086 este cel mai răspândit microprocesor pe 16 biți.

Acesta acoperă o gamă largă de aplicații datorită celor două moduri de lucru: *modul minim*, folosit pentru aplicații simple și *modul maxim*, pentru aplicații complexe și chiar sisteme multiprocesor. Proiectarea este realizată cu 8086 în modul minim, pinul MN/!MX fiind conectat la VCC.

## **Generatorul de clock 8284A**

Circuitul 8284A este un generator de semnal de tact pentru procesoarele 8086 și 8088 care poate produce semnale la frecvențe de 5Mhz sau 8Mhz.

Frecvența dorită în cazul nostru este de 5Mhz. Ca să ajungem la acest rezultat trebuie să legăm la pinii X1 și X2 ai circuitului capetele unui cuarț cu o frecvență de trei ori mai mare, adică de 15Mhz. Acest lucru este necesar pentru că generatorul de clock al 8284A folosește un numărător intern numit “divide-by-three counter”, microprocesorul având nevoie de un factor de umplere de 33%.

Pentru resetarea semnalului la aprindere folosim un circuit RC legat la pinul !RES al 8284A, așa cum este specificat în documentație.

## **Circuitul amplificator/separator bidirectional 74x245**

Circuitul asigură comunicarea asincronă între două magistrale în ambele direcții. Direcția de date este specificată de pinul DIR, conectat la DT!/R (Data transmit/receive) care indică transmisie de date (pe “1” logic) și recepție de date (pe “0” logic).

Intrarea !G este de enable și când este pe 1 “izolează” cele două magistrale. Aceasta este conectată la semnalul !DEN (Data Enable, cu activare pe 0) al 8086.

Schema internă prezintă opt buffere cu trei stări orientate spre stânga și opt orientate spre dreapta activate în totalitate de semnalul de enable !G și poziționate pe o direcție de semnalul DIR.

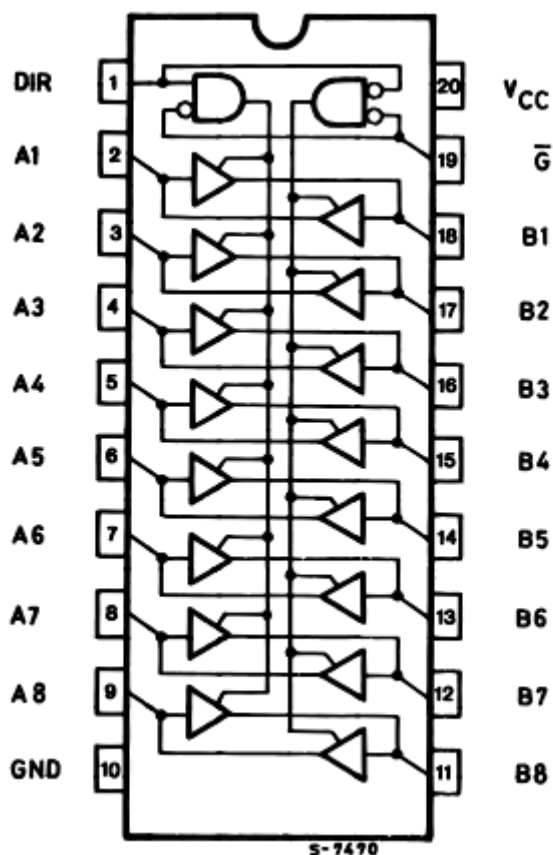
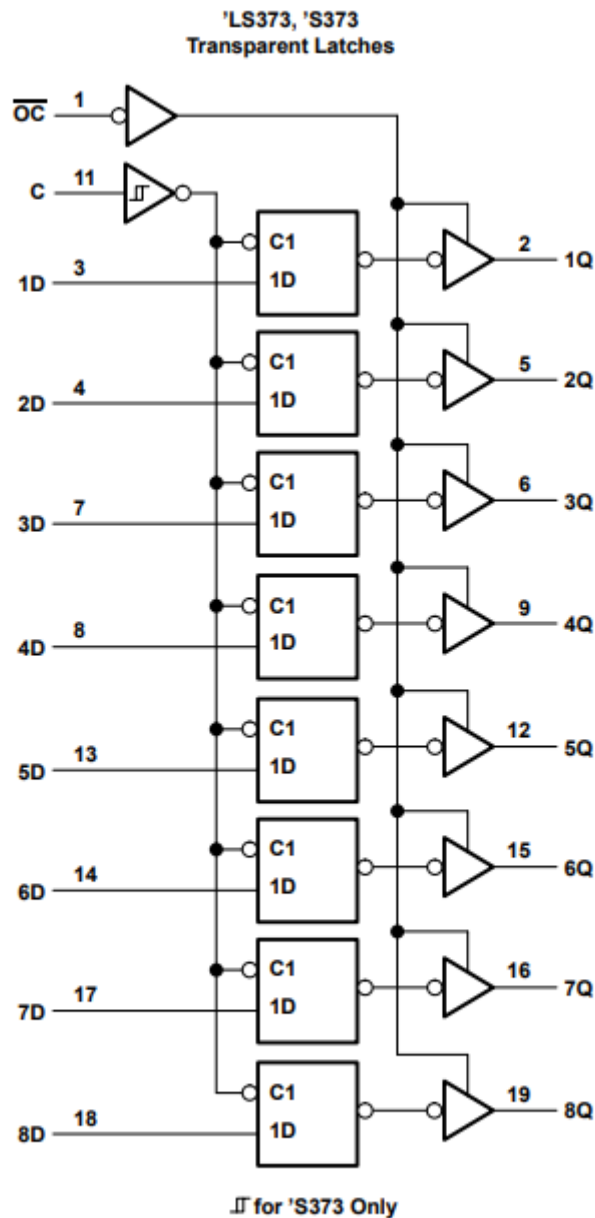


Fig. 2

## Circuit registru 74x373

Trei circuite 74LS373 sunt folosite in schema din Fig. 1 cu rol în demultiplexarea adreselor. Acestea își deservesc rolul atunci când semnalul ALE (conectat la intrarea G a circuitului) este activ.

În interior, circuitul este format din opt flip-flop-uri D a căror ieșire este conectată la un buffer cu trei stări. Acest lucru face posibilă reținerea biților de adresă chiar și atunci când pe pinii microprocesorului “se vor scrie” în continuare date.



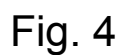
Pin numbers shown are for DB, DW, J, N, NS, and W packages.

Fig. 3

## Conectarea Memoriilor

Pentru conectarea memoriilor s-au folosit circuitele:

- Circuit decodificator 74x138
- Două circuite de memorie EPROM 27512
- Două circuite de memorie SRAM 62256

[illegible]

Se observă că biții A19, A18 și A17 au tot timpul valoarea 0 pentru memoria RAM și tot timpul valoarea 1 pentru memoria EPROM.

Astfel, aceștia pot fi folosiți ca biți de intrare în circuitul decodificator 74x138.

## **Explicația semnalelor pentru funcționarea circuitelor de memorie**

### **- Selecție circuit -**

Fiecare circuit de memorie are o intrare !CS sau !CE (Chip select / Chip enable) activată pe 0 și conectată la ieșirea specifică din decodificator (!Y0 pentru RAM și !Y7 pentru EPROM). Activarea blocului pentru RAM s-a validat și prin verificarea ca bitul A16 să aibă valoarea 0 (cu ajutorul unei porți SAU), astfel încât să evităm fenomenul de “mirroring” de adrese.

### **- Citire -**

Semnalul !OE al fiecărui circuit activează citirea din registru și este conectat direct la !RD al microprocesorului.

### **- Scriere -**

Semnalul !WE (Write enable) specific memoriei RAM este activat în funcție de circuit. Memoria este împărțită practic în două, fiecare circuit fiind adresabil individual. Setând amândouă semnalele !WR al 8086 și A0 pe 0 obținem acces de scriere pe adresele pare ale memoriei, iar dacă punem !WR și !BHE (Bus high enable) pe 0 putem scrie la adresele impare.

Ieșirile circuitelor duc, prin magistrale, la circuitele 74x245.



## Decodificarea porturilor

Memorie	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	Adresa	Semnal iesire
Serial1	0	0	0	0	0	1	0	0	1	1	0	1	0	0	0	0	04D0h	!Y2
	0	0	0	0	0	1	0	0	1	1	0	1	0	0	1	0	04D2h	
Serial2	0	0	0	0	0	1	0	1	1	1	0	1	0	0	0	0	05D0h	!Y3
	0	0	0	0	0	1	0	1	1	1	0	1	0	0	1	0	05D2h	
Paralel1	0	0	0	0	0	0	1	0	0	1	0	1	0	0	0	0	0250h	!Y0
	0	0	0	0	0	0	1	0	0	1	0	1	0	1	1	0	0256h	
Paralel2	0	0	0	0	1	0	1	0	0	1	0	1	0	0	0	0	0A50h	!Y4
	0	0	0	0	1	0	1	0	0	1	0	1	0	1	1	0	0A56h	
Tastatura (Linii)	0	0	0	0	1	1	0	0	0	1	0	1	0	0	0	0	0C50h	!Y6
Tastatura (Coloane)	0	0	0	0	1	1	0	1	0	1	0	1	0	0	0	0	0D50h	!Y7
	L	L	L	L	sel	sel	dc	sel	dc	H	L	H	L	dc	dc	dc		

$A2 = A10$

$A1 = A11$

$A0 = A8$

$!E1 = M/!IO$

$!E2 = (A15 | A14) | (A13 | A12)$

$E3 = !(A3 | A5) \& (A4 \& A6)$

Memorie	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	Adresa	Semnal iesire
Afisaj1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000h	!Y0
Afisaj2	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0010h	!Y1
Afisaj3	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0020h	!Y2
Afisaj4	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0030h	!Y3
Afisaj5	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0040h	!Y4
Afisaj6	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0050h	!Y5
Afisaj7	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0060h	!Y6
Afisaj8	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0070h	!Y7
LCD	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	00A0h	!Y10
LED 1-8	0	0	0	0	0	0	0	0	1	0	1	1	0	0	0	0	00B0h	!Y11
LED 9-10	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	00C0h	!Y12
	L	L	L	L	L	L	L	L	sel	sel	sel	sel	L	L	L	L		

$$A3 = A7$$

$$A2 = A6$$

$$A1 = A5$$

$$A0 = A4$$

$$!E1 = ((A15 | A14) | (A13 | A12)) | (M/!IO | !WR)$$

$$!E2 = ((A11 | A10) | (A9 | A8)) | ((A3 | A2) | (A1 | A0))$$

Există două tabele pentru adresele de porturi I/O, deci și două decodificatoare, unul 3x8 și unul 4x16.

## Interfețele serială și paralelă

Circuite folosite:

- Interfața programabilă de comunicare serială 8251A
- Interfața programabilă de comunicare paralelă 8255A

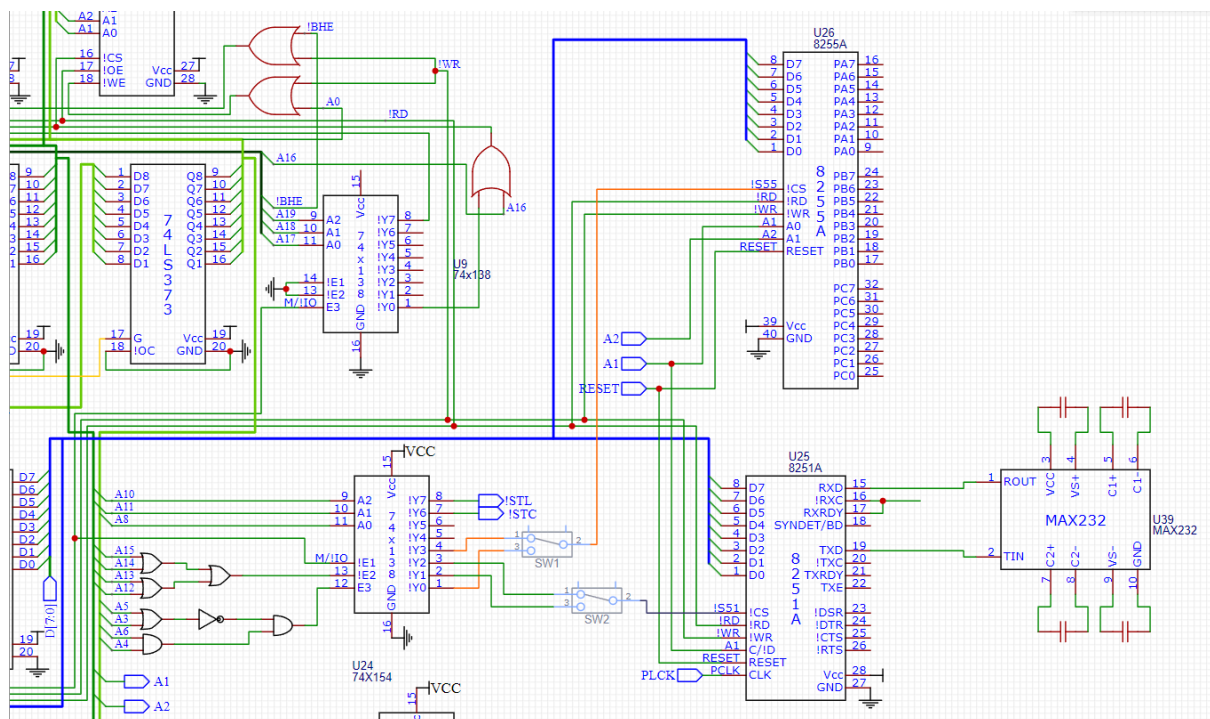


Fig. 5

## Interfața serială folosind 8251A

Circuitul “transformă” datele primite de la CPU în mod paralel pentru a fi transmise către periferice într-o modalitate serială.

C/D	RD	WR	CS	
0	0	1	0	8251A DATA → DATA BUS
0	1	0	0	DATA BUS → 8251A DATA
1	0	1	0	STATUS → DATA BUS
1	1	0	0	DATA BUS → CONTROL
X	1	1	0	DATA BUS → 3-STATE
X	X	X	1	DATA BUS → 3-STATE

Fig. 6

Conform documentației, în funcție de codificarea pinilor din figura de mai sus, se pot scrie și citi date de la circuit, se poate lua registrul de stare și se poate programa scriind în memoria de control.

Pentru a delimita între memoria de control și cea de date se leagă pinul A1 la C/!D.

Pentru transmitere, datele “ies” prin TXD, iar pentru primirea datelor folosim intrarea RXD. Amandoua sunt conectate la un circuit MAX232 care are rolul de a converti semnalele de la logică TTL la logică serială.

## Interfața serială folosind 8251A

Circuitul este un dispozitiv programabil de intrare-ieșire paralelă care dispune de 24 de linii de I/O grupate în trei porturi a câte 8 biți: A, B și C.

Selecția regiștrilor se face poziționând semnalele A1 și A0 într-o anumită manieră, după cum se observă în Fig. 7 de mai jos. A1 este conectat la A2 al 8086 și A0 la A1 al 8086 astfel

încât să se obțină patru adrese pare consecutive (primele trei pentru porturile A, B și C iar al patrulea pentru Control).

**8255A BASIC OPERATION**

A <sub>1</sub>	A <sub>0</sub>	$\overline{RD}$	$\overline{WR}$	$\overline{CS}$	Input Operation (READ)
0	0	0	1	0	Port A → Data Bus
0	1	0	1	0	Port B → Data Bus
1	0	0	1	0	Port C → Data Bus
					<b>Output Operation (WRITE)</b>
0	0	1	0	0	Data Bus → Port A
0	1	1	0	0	Data Bus → Port B
1	0	1	0	0	Data Bus → Port C
1	1	1	0	0	Data Bus → Control
					<b>Disable Function</b>
X	X	X	X	1	Data Bus → 3-State
1	1	0	1	0	Illegal Condition
X	X	1	1	0	Data Bus → 3-State

Fig. 7

## Microcomutatoare

Microcomutatoarele SW1 și SW2 au rolul de a alege între cele două adrese, prin închiderea circuitului doar la una din ieșirile decodicatorului. Acestea pun pe 0 semnalele !S51 și !S55 doar atunci când adresa la care sunt conectate este ieșire a decodicatorului, acționând !CS al circuitelor 8251A sau 8255A.

## Conectarea LED-urilor

Se folosesc:

- Două circuite registru 74LS373
- Zece LED-uri
- Zece rezistoare de 330 ohm

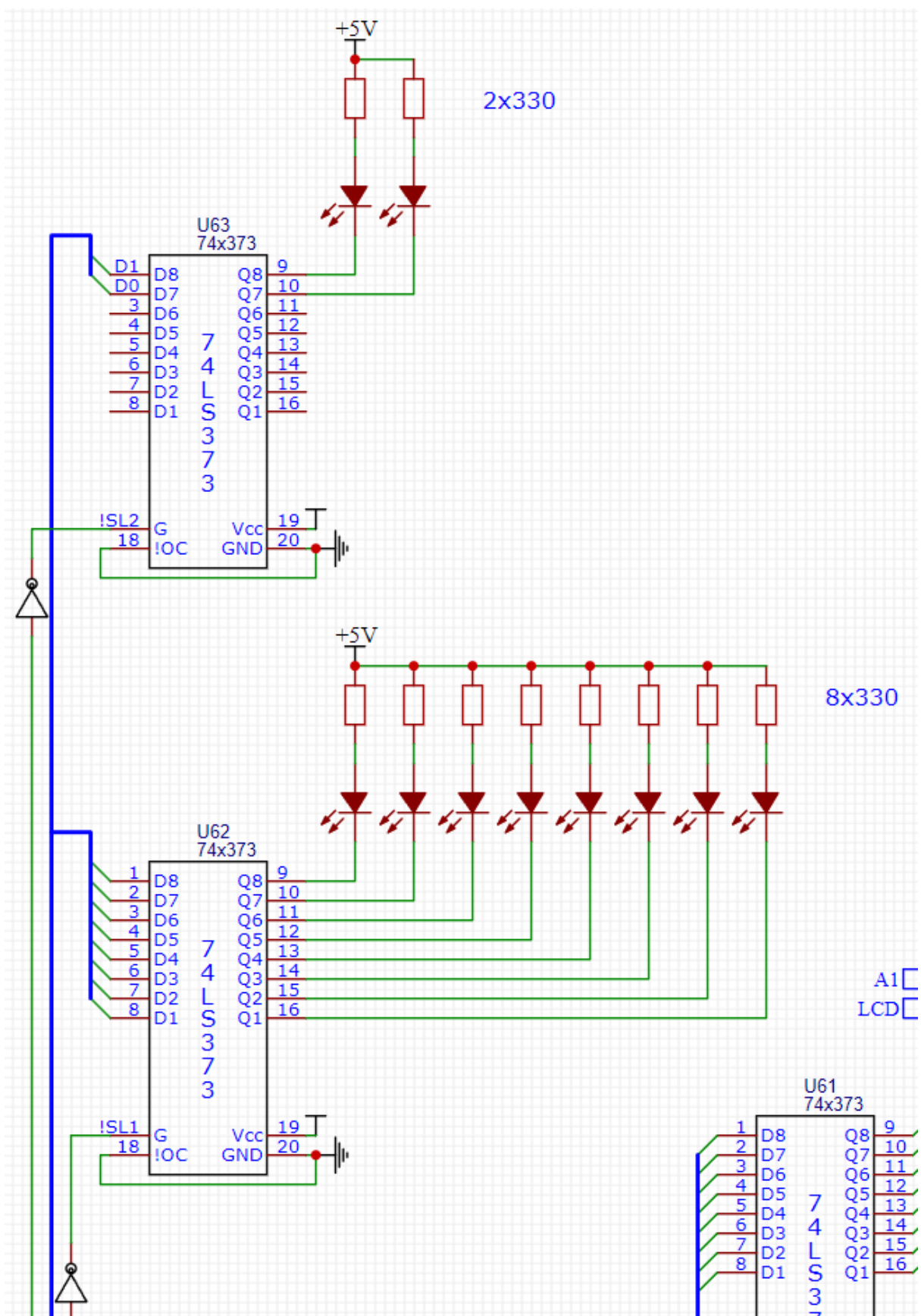


Fig. 8

Din cele 10 led-uri, 8 vor fi puse pe o adresă, iar restul de două pe o altă adresă.

Led-urile luminează la setarea ieșirii 373 pe 0.

## Conectarea afişajelor cu segmente

## Folosim:

- Opt circuite registru 74LS373
- Opt afişaje cu 7 segmente

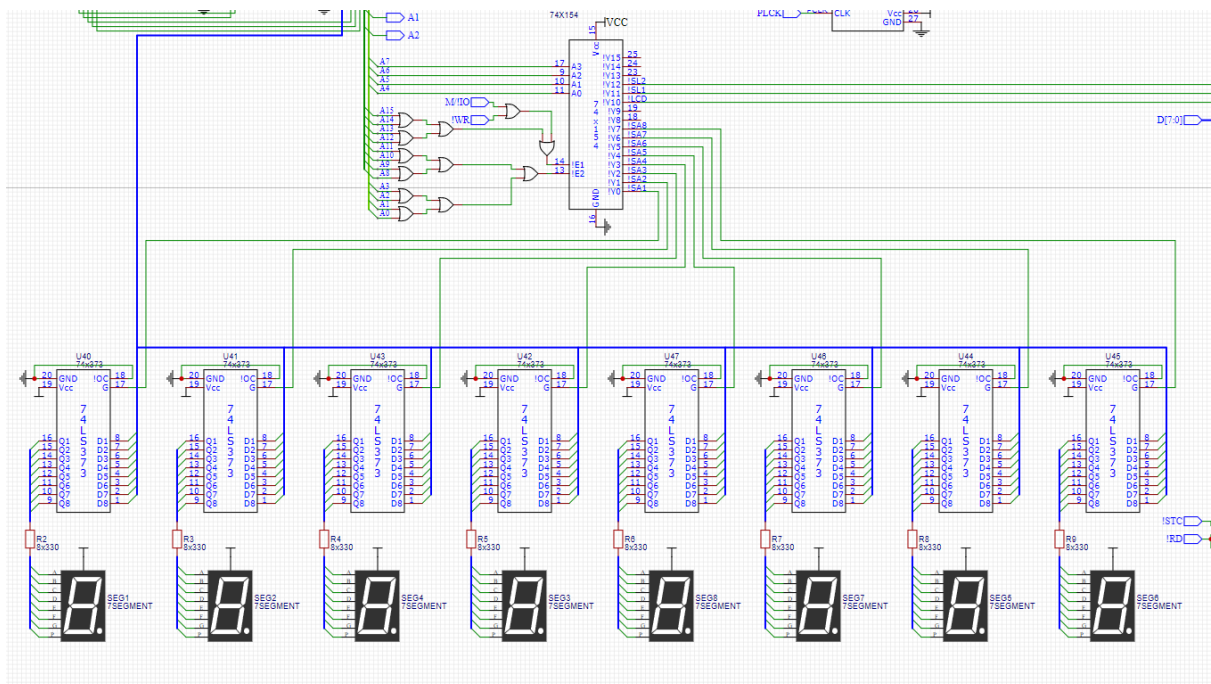


Fig. 9

Fiecare segment are propriul circuit registru activat de ieșirea decodificatorului. Led-urile segmentelor sunt luminate cand ieșirea circuitelor 373 este 0.

## Conectarea Minitastaturii

Folosim:

- Circuit registru 74LS373
- Circuit amplificator/separator bidirectional 74x245
- Nouă taste
- Trei rezistoare de 10K ohm

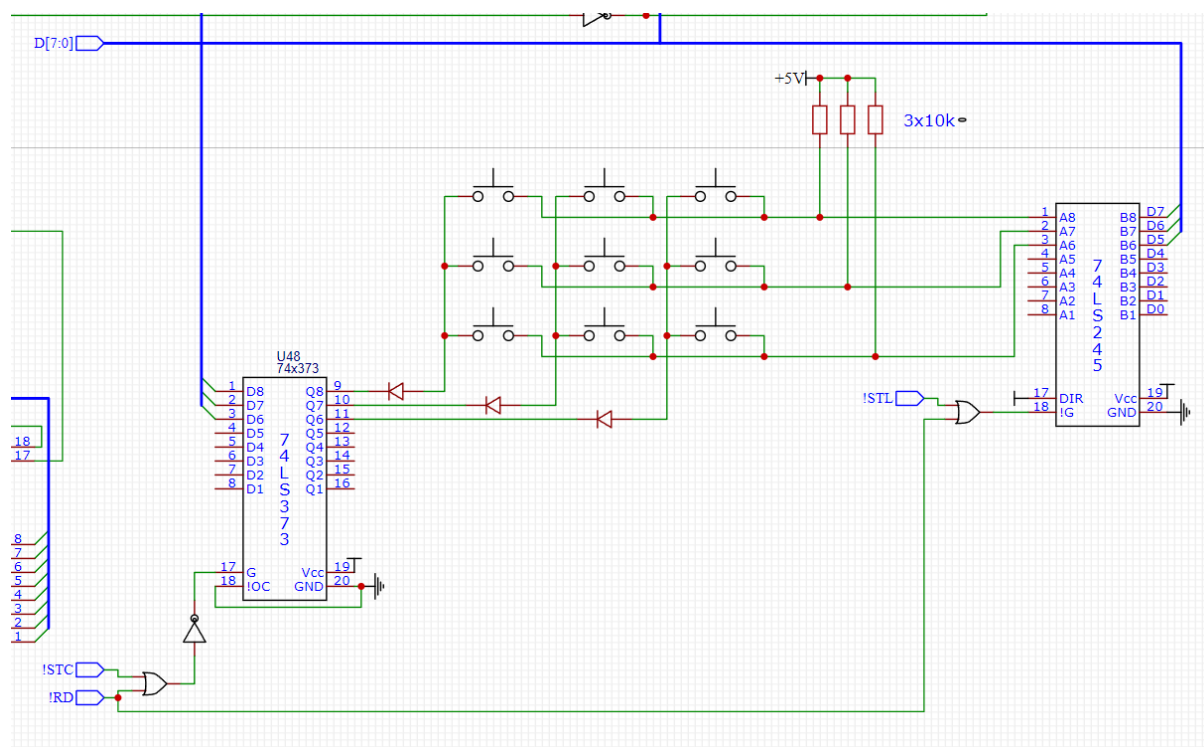


Fig. 10

Folosim, pentru selecția coloanelor, un circuit registru 74LS373 care activează pe rând câte o coloană. Blocul de citire al liniilor este reprezentat de un circuit 74LS245 cu intrarea DIR legată la VCC astfel încât datele se transmit “de la stânga la dreapta”, adică de la tastatură pe magistrala de date a microprocesorului.

## Conectarea LCD 16x2

Folosim:

- Circuit registru 74LS373

- Modul LCD 16x2 HS HS1602A

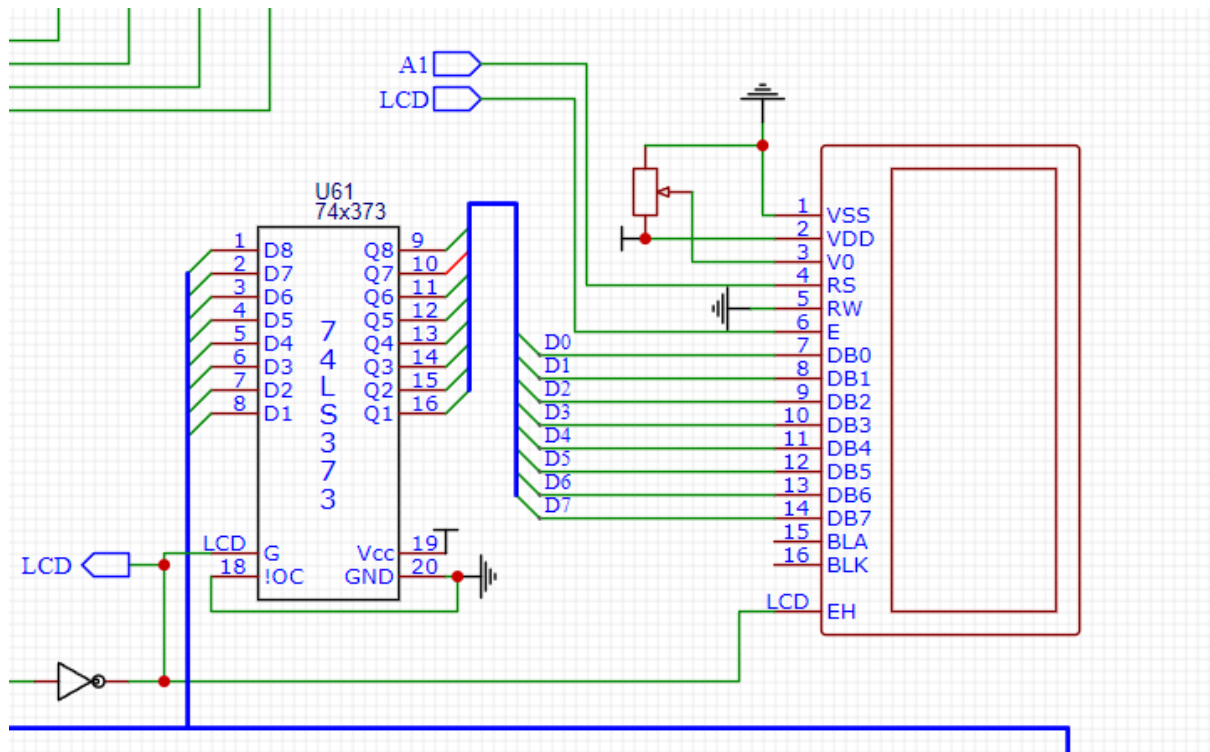


Fig. 11

### 三. interface definition

#### 1. 2.54\*16P

PIN	Name	Power	Functional Description
1	VSS	0V	Power ground
2	VCC	5.0V	Power supply
3	V0	0.2V	*
4	RS	H/L	RS=H, Display data RS=L, Instruction
5	R/W	H/L	R/W=H, Read data R/W=L, Write data
6	E	H,H→L	Enable signal
7	DB0	H/L	Data
8	DB1	H/L	Data
9	DB2	H/L	Data
10	DB3	H/L	Data
11	DB4	H/L	Data
12	DB5	H/L	Data
13	DB6	H/L	Data
14	DB7	H/L	Data
15	BLA	---	NC (LEDA)
16	BLK	---	NC (LEDK)

\*this pin need connect the VR(10K) to VSS.

Fig. 12



Am legat semnalele la pinii LCD conform documentației din Fig. 12.

R!/W este conectat la ground deoarece vom scrie mereu pe afișajul LCD.

RS este conectat la A1 deoarece este modul de adresare al adreselor pare ale microprocesorului. Astfel, vom avea două adrese pare consecutive: una pentru date și una pentru control.

V0 este conectat ca fiind “controllerul” potențiometrului care se ocupă de contrastul luminii de fundal.

## Subrutinele perifericelor

### Interfața serială

#### Adresele de port serial:

04D0h (05D0h) pentru date

04D2h (05D2h) pentru stări/comenzi

#### Programarea 8251

```
MOV DX, 04D2h ; adresa de control
MOV AL, 0CEh ; cuvant de mod, 2 biti de stop,
fara paritate, 8 biti transmisi, 16x baud rate
OUT DX, AL
MOV AL, 15h ; cuvant de comanda, error reset,
receive enable, transmit enable
OUT DX, AL
RET
```

#### Transmisie caracter

```
TR: MOV DX, 04D2h
IN AL, DX
```

```
RCR AL, 1 ; verificam TxRDY
JNC TR ; daca nu exista carry vom merge inapoi

MOV AL, CL
MOV DX, 04D0h
OUT DX, AL
RET
```

## Recepție caracter

```
REC: MOV DX, 04D2h
IN AL, DX
RCR AL, 2 ; verificam RxRDY
JNC REC

MOV DX, 04D0h ; adresa de date
IN AL, DX ; citim datele
MOV CL, AL
RET
```

## Interfața paralelă

Paralel modul 0 - basic input/output

### Adresele de port paralel:

0250h (0A50h) pentru portul A

0252h (0A52h) pentru portul B

0254h (0A54h) pentru portul C

0256h (0A56h) pentru RCC

### Rutina de programare:

```
MOV DX, 0256h
MOV AL, 81h ; A=out, B=out, lowC=in, highC=out
OUT DX, AL
RET
```

### Rutina de emisie caracter:

```
PAR: MOV DX, 0254h ; portul C de intrare, cu  
semnalul busy activ pe 0  
IN AL, DX ; citire si testare busy  
RCR AL, 1  
JNC PAR  
  
MOV AL, CL ; se preia caracterul din registrul  
CL  
MOV DX, 0250h  
OUT DX, AL  
  
; se genereaza manual semnalul strobe pentru  
emitere caracter  
OR AL, 01h  
MOV DX, 02562h  
OUT DX, AL ; !STB = 1  
  
AND AL, 00h  
OUT DX, AL ; !STB = 0  
  
OR AL, 01h  
OUT DX, AL ; !STB = 1  
RET
```

## Led-uri

Aprinderea este comandata pe 0 logic

### Rutina de aprindere a unui led:

```
MOV AL, FEh  
OUT 00B0h, AL ; led-ul 1  
OUT 00C0h, AL ; led-ul 1 din zona a doua de  
memorie = led-ul 9  
RET
```

## Rutina de stingere a unui led:

```
MOV AL, 01h
OUT 00B0h, AL ; led-ul 1
OUT 00C0h, AL ; led-ul 1 din zona a doua de
memorie = led-ul 9
RET
```

## Afişaj cu 7 segmente

Anod comun => activare pe 0

```
; afisarea cifrei 0 pe primul rang
MOV AL, 0C0h
OUT 00h, AL
```

## Tastatură

```
; se pune 0 pe prima coloana si se "asculta" daca
s-au actionat tastele 1, 4, 7
REIA:
MOV AL, 0FEh
OUT 0D50h, AL

IN AL, 0C50h
AND AL, 01h
JZ TASTA1

IN AL, 0C50h
AND AL, 02h
JZ TASTA4

IN AL, 0C50h
AND AL, 04h
JZ TASTA7

; se pune 0 pe a doua coloana si se "asculta" daca
s-au actionat tastele 2, 5, 8
MOV AL, 0FDh
OUT 0D50h, AL
```

```
IN AL, 0C50h
AND AL, 01h
JZ TASTA2
```

```
IN AL, 0C50h
AND AL, 02h
JZ TASTA5
```

```
IN AL, 0C50h
AND AL, 04h
JZ TASTA8
```

```
; se pune 0 pe a treia coloana si se "asculta" daca
s-au actionat tastele 3, 6, 9
```

```
MOV AL, 0FBh
OUT 0D50h, AL
```

```
IN AL, 0C50h
AND AL, 01h
JZ TASTA3
```

```
IN AL, 0C50h
AND AL, 02h
JZ TASTA6
```

```
IN AL, 0C50h
AND AL, 04h
JZ TASTA9
```

```
JMP REIA
```

```
; tratarea acționării tastei 1
```

```
TASTA1: CALL DELAY ; se așteaptă stabilizarea
contactelor
```

```
AST1: IN AL, 0C50h ; se citește din nou linia și
se așteaptă dezactivarea tastei
```

```
AND AL, 01H
```

```
JZ AST1
```

```
CALL DELAY
```

```
; operația corespunzătoare acționării tastei 1
```

# Bibliografie

Cursuri și materiale proiect conform [Campus Virtual](#)

<https://www.alldatasheet.com/datasheet-pdf/download/124124/Intel/8284A.html>

<https://www.alldatasheet.com/datasheet-pdf/download/83514/Siliconix/TMICROELECTRONICS/74LCX245.html>

[https://www.alldatasheet.com/datasheet-pdf/download/28029/Texas Instruments/74LS373.html](https://www.alldatasheet.com/datasheet-pdf/download/28029/Texas%20Instruments/74LS373.html)

<https://www.alldatasheet.com/datasheet-pdf/download/66096/Intel/8251A.html>

<https://www.alldatasheet.com/datasheet-pdf/download/66100/Intel/8255A.html>

<https://www.lcsc.com/datasheet/C5329588.pdf>