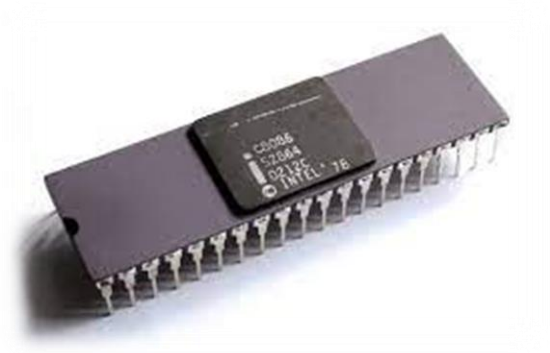
Universitatea Politehnica din Timișoara
Facultatea de Automatică și Calculatoare

Proiectarea Microsistemelor Digitale

Microsistem cu microprocesorul 8086



TURCIN Cristian

An universitar 2023-2024

Tema proiectului:

Să se proiecteze un microsistem cu următoarea structură:

- unitate centrală cu microprocesorul 8086;

- 128 KB memorie EPROM, utilizând circuite 27C1024;

- 256 KB memorie SRAM, utilizând circuite 62512;

- interfaţă serială, cu circuitul 8251, plasată în zona 0DD0H – 0DD2H sau 0F50H – 0F52H, în funcţie de poziţia microcomutatorului S1;

- interfaţă paralelă, cu circuitul 8255, plasată în zona 0150H – 0156H sau 0A50H – 0A56H, în funcţie de poziţia microcomutatorului S2;

- o minitastatură cu 16 contacte;

- 16 led-uri;

- doua module de afişare cu segmente, cu 4 ranguri fiecare.

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

- rutinele de programare ale circuitelor 8251 şi 8255;

- rutinele de emisie/ recepţie caracter pe interfaţa serială;

- rutina de emisie caracter pe interfaţă paralelă;

- rutina de scanare a minitastaturii;

- rutina de aprindere/ stingere a unui led;

- rutina de afişare a unui caracter hexa pe un rang cu segmente.

Structura rutinelor (intrări, secvenţe, ieşiri) va fi stabilită de fiecare student.

Descrierea hardware

1. Unitatea centrală cu procesor 8086

Unitatea centrală cuprinde:

- microprocesorul 8086

- generator de tact 8284A

- circuite pentru amplificarea și demultiplexarea magistralelor de adrese şi date

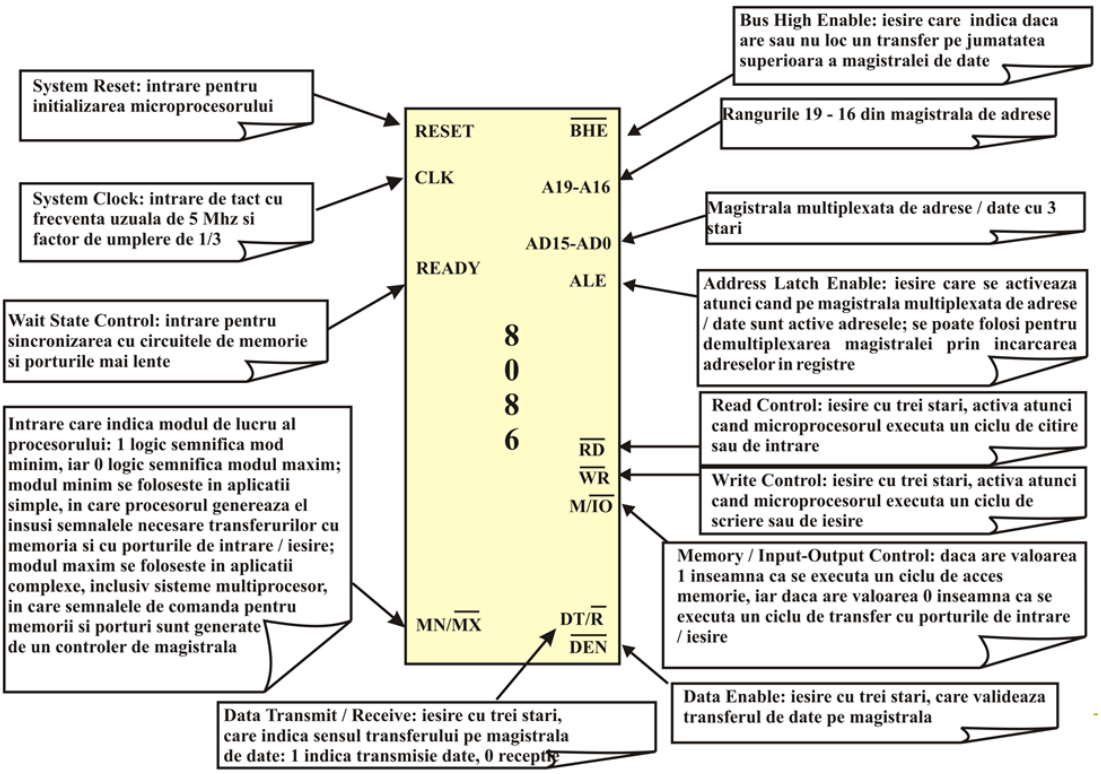
Microprocesorul Intel 8086

Este primul microprocesor pe 16 biţi care a cunoscut o largă utilizare. Microprocesorul Intel 8086 lucrează cu date de 16 biţi, numite cuvinte de date. Transferurile între UE și UI sau între microprocesor și MP sau IO se fac, în general, sub formă de cuvinte de date de 16 biți. De aceea bus-ul intern prin care comunică UE și UI al microprocesorului Intel 8086 este de 16 biți.

Apariția lui a fost urmată la scurt timp de o familie de componente: generatorul de tact 8248, controlerul de magistrală 8288, coprocesorul matematic 8087 si coprocesorul de intrare/ ieșire 8089.

Generatorul de tact 8284A

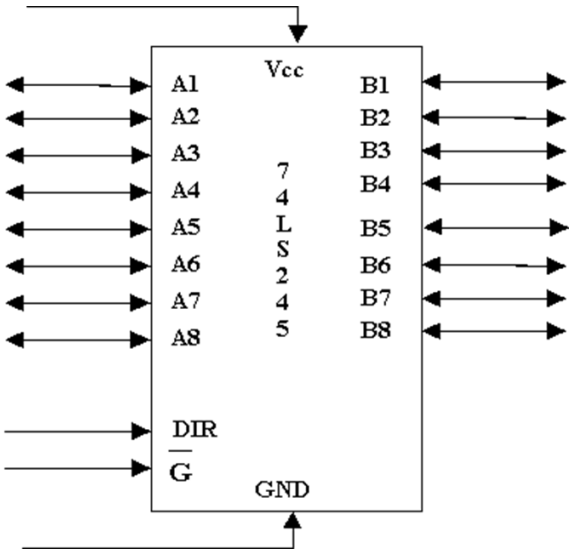
Circuitul 8284A este folosit pentru a genera tactul CLK către microprocesor și către circuitele specializate ale interfeței seriale. Acesta mai are rolul de a genera READY și RESET către microprocesor, sincronizându-le cu tactul.



Circuitul amplificator/ separator bidirecţional 74x245:

-Este un circuit folosit pentru amplificarea/ separarea magistralelor bidirecţionale ale microprocesoarelor;

-Configuraţia terminalelor:



-Funcţionarea:

/G	DIR	A8 – A1	B8 – B1
0	0	B8 – B1	Intrări
0	1	Intrări	A8 – A1
1	X	A 3 – a stare	A 3 – a stare

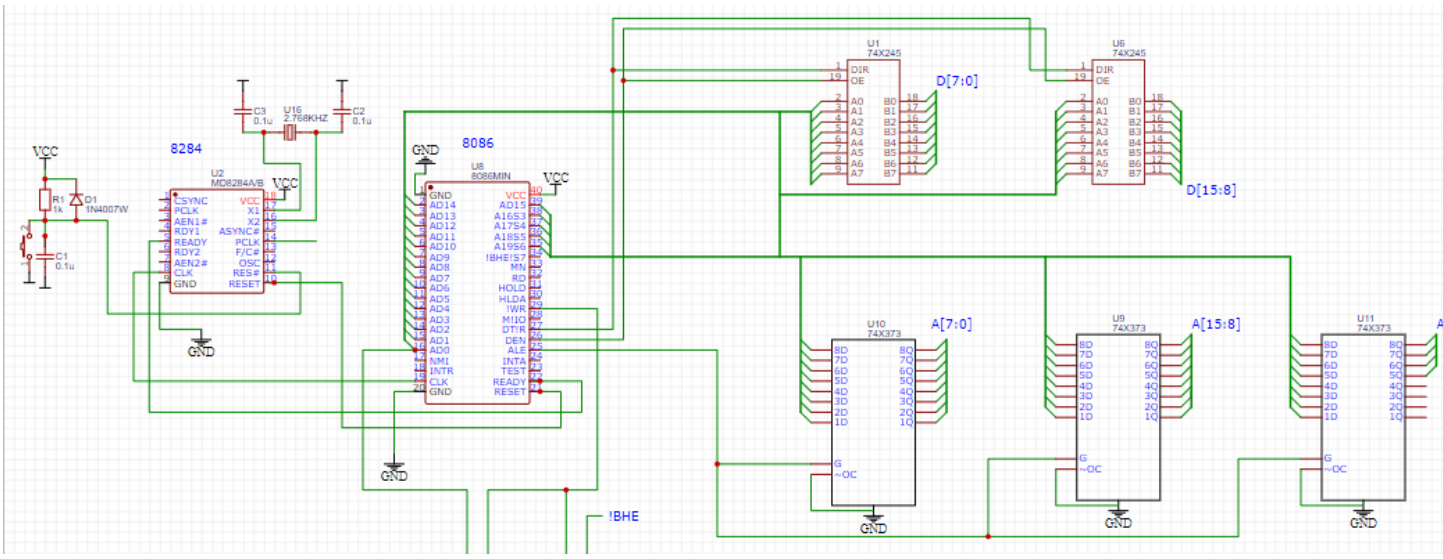
Circuitul registru 74x373

-este folosit pentru demultiplexarea liniilor de adresă.

-este un registru cu ieșiri cu 3 stări, alcătuit din 8 bistabile. Are o intrare de validare pentru toate ieșirile /OC (dacă este pe 1 logic bistabilile trec în a 3-a stare) și o intrare pentru încărcarea bistabilelor G activă la 1 logic.

/OC	G	8Q – 1Q
0	0	Vechiul conţinut
0	1	8D – 1D
1	X	A 3 – a stare

UNITATEA CENTRALA

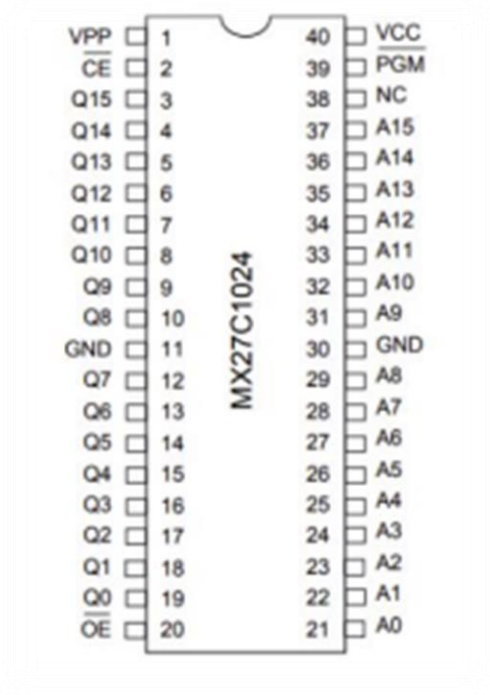


2. Circuitele de memorie

Memoria EPROM (traducere liberă: memorie read only programabilă sau anulabilă) este un tip de memorie de calculator nevolatilă, adică o

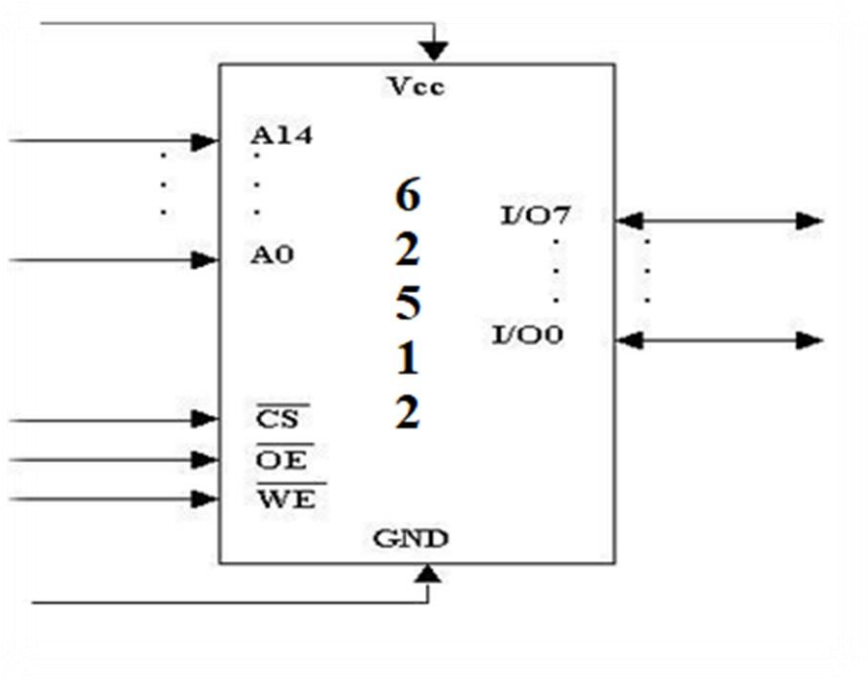
memorie care își păstrează datele chiaă și când i se întrerupe alimentarea cu curent electric.

Circuitul de memorie EPROM 27C1024 este un circuit EPROM avand capacitatea de 128Ko. Se va folosi un singur circuit.



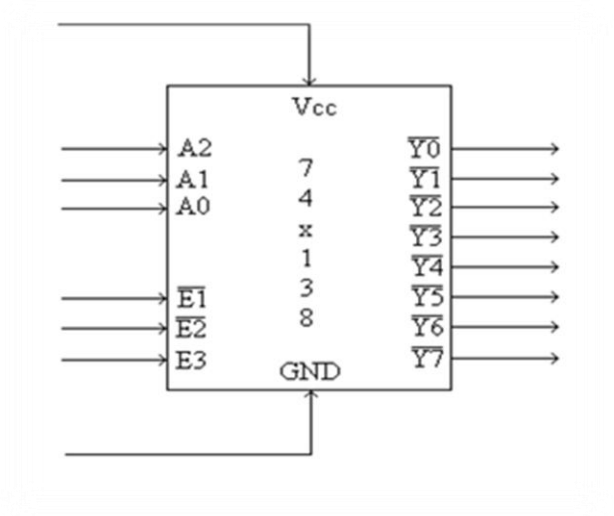
Memoria SRAM (Static Random Acces Memory) este un tip de memorie semiconductoare; static sublinează faptul că, spre deosebire de memoriile DRAM (Dynamic Random Acces Memory), nu mai este necesar un ciclu periodic de reîmprospătare (refresh). Memoriile SRAM folosesc circuite logice combinaționale pentru a memora fiecare bit. Datele depuse în memorie sunt stabile.

Circuitul de memorie SRAM 62512 este un circuit SRAM avand capacitatea de 64KB. Se vor folosi 4 astfeld de circuite.



Circuitul decodificator 74x138:

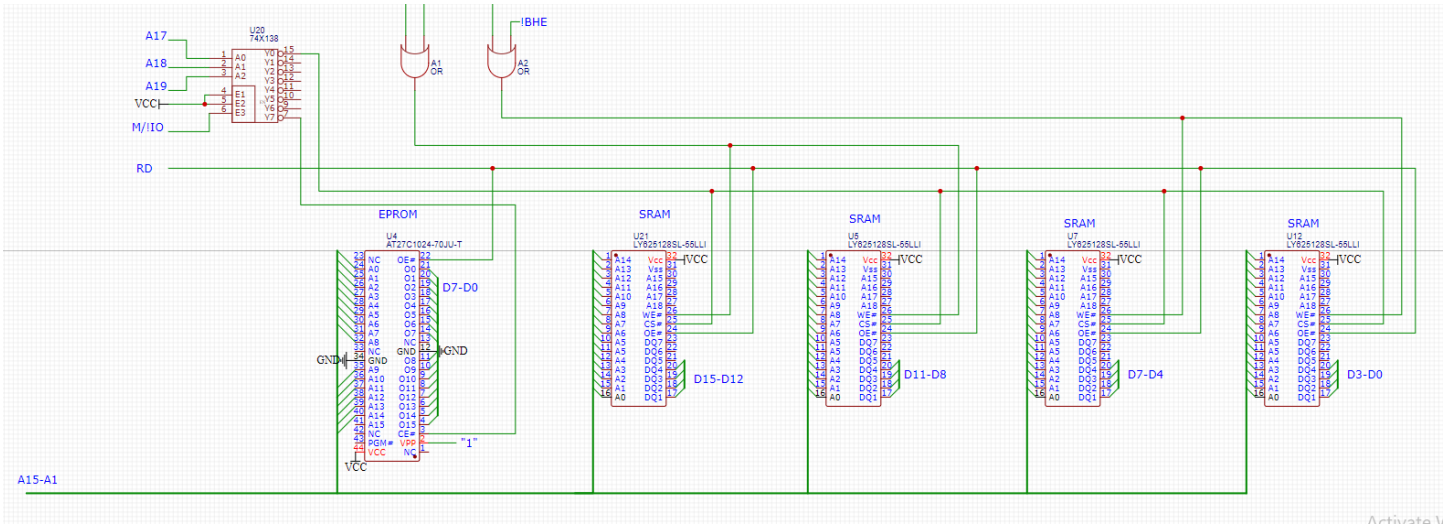
-Configuraţia terminalelor:



-Funcţionarea:

E3	/E2	/E1	A2	A1	A0	/Y7	/Y6	/Y5	/Y4	/Y3	/Y2	/Y1	/Y0
1	0	0	0	0	0	1	1	1	1	1	1	1	0
1	0	0	0	0	1	1	1	1	1	1	1	0	1
1	0	0	0	1	0	1	1	1	1	1	0	1	1
1	0	0	0	1	1	1	1	1	1	0	1	1	1
1	0	0	1	0	0	1	1	1	0	1	1	1	1
1	0	0	1	0	1	1	1	0	1	1	1	1	1
1	0	0	1	1	0	1	0	1	1	1	1	1	1
1	0	0	1	1	1	0	1	1	1	1	1	1	1
0	X	X	X	X	X	1	1	1	1	1	1	1	1
X	1	X	X	X	X	1	1	1	1	1	1	1	1
X	X	1	X	X	X	1	1	1	1	1	1	1	1

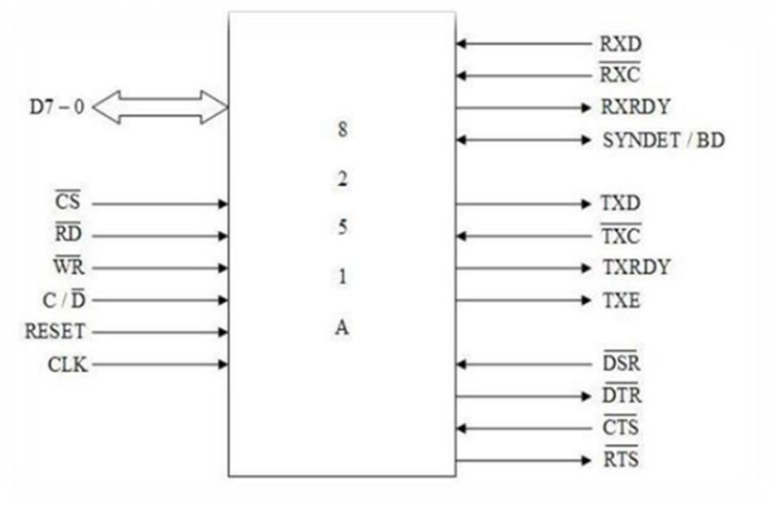
CONECTAREA MEMORIILOR



3.Interfaţa serială şi interfaţa paralelă

Interfaţa serială este reprezentată de circuitele și programele de bază care asigură comunicare între unitatea centrală și echipamentele periferice, comunicare bit după bit.

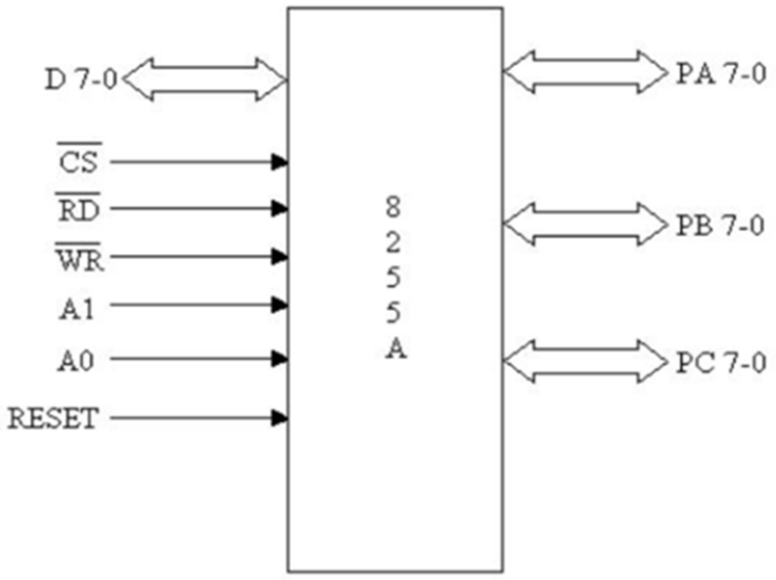
Circuitul 8251 (specializat pentru transferurile seriale) este un circuit USART (Universal Synchronous Asynchronous Receiver Transmitter) pentru comunicarea de tip serial.



Legătura între operaţiile realizate de circuit şi starea terminalelor de comandă

/CS	/RD	/WR	C//D	Operaţie
1	X	X	X	Magistrala de date în a 3-a stare
0	1	1	X	Magistrala de date în a 3-a stare
0	0	1	1	Citire a octetului de stare
0	0	1	0	Citire a datei
0	1	0	1	Scriere a cuvintelor de comandă
0	1	0	0	Scriere a datei

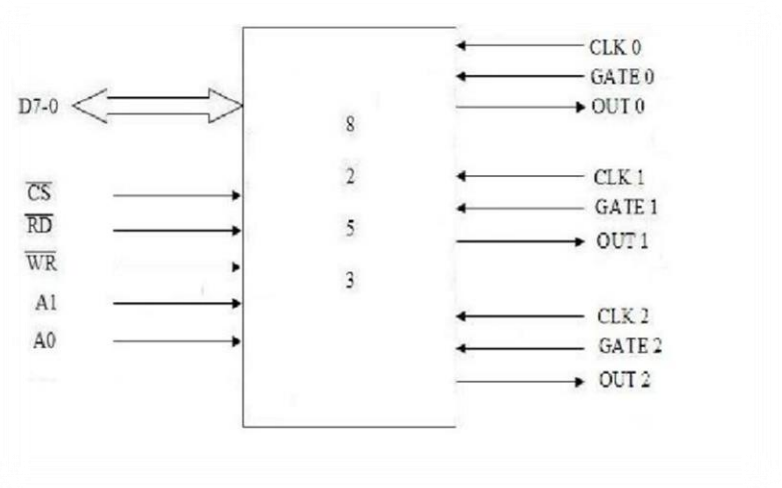
Interfaţa paralelă – circuitul 8255A Spre deosebire de transferul serial, la care transferul datelor se face bit după bit, la transferul paralel se transferă 8 biți simultan iar transferul este însoțit şi de semnale de dialog. Interfața paralelă se realizează cu ajutorul circuitului 8255



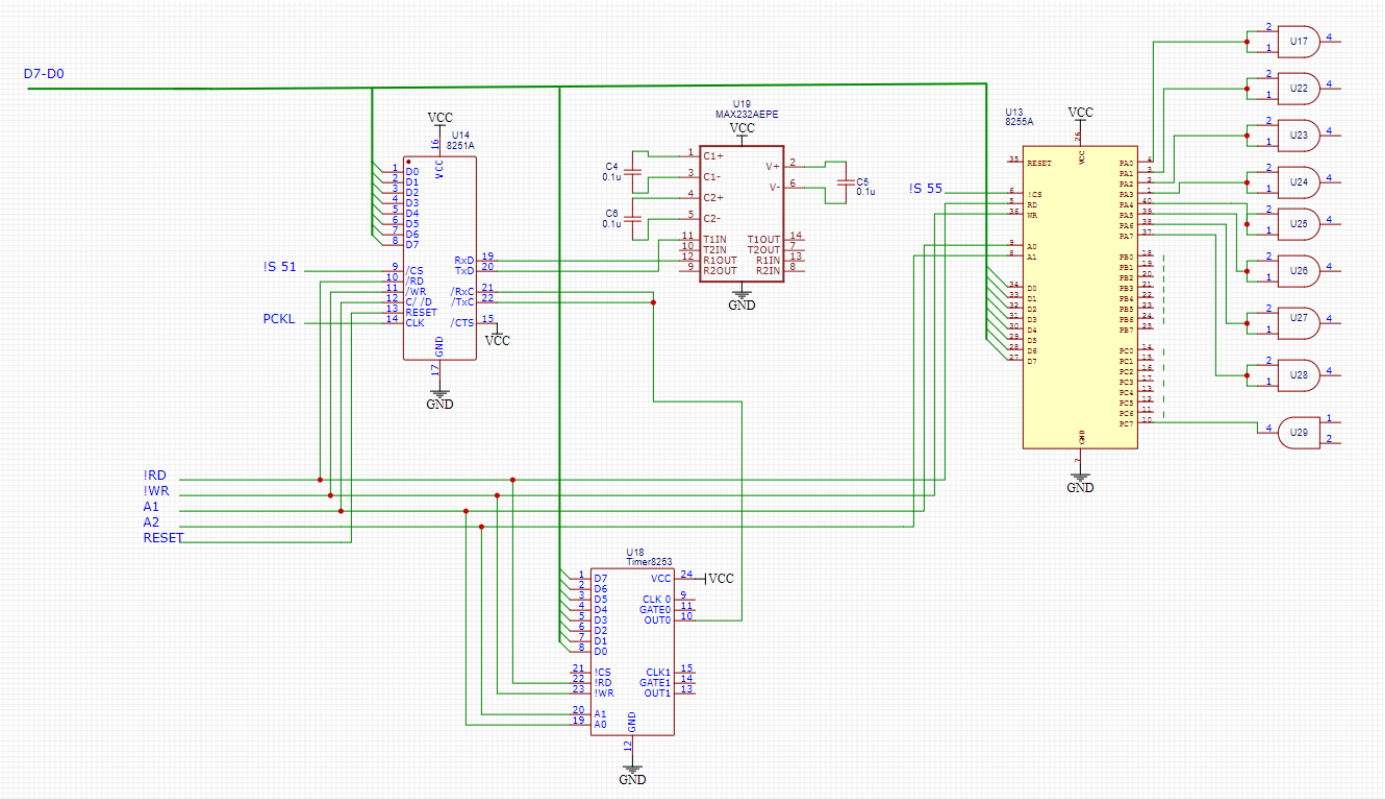
Semnificaţiile terminalelor

/CS	/RD	/WR	A1	A0	Operaţia
0	1	0	0	0	Scriere în portul A
0	1	0	0	1	Scriere în portul B
0	1	0	1	0	Scriere în portul C
0	1	0	1	1	Scriere în portul cuvântului de comandă
0	0	1	0	0	Citire din portul A
0	0	1	0	1	Citire din portul B
0	0	1	1	0	Citire din portul C
0	0	1	1	1	Fără operaţie – magistrala de date este în a 3-a stare
0	1	1	x	x	Fără operaţie – magistrala de date este în a 3-a stare
1	x	x	x	x	Magistrala de date este în a 3-a stare

Circuitul 8253 (contor temporizator) generează rata de transfer pentru interfaţa serială, fiind conectat la terminalele TxC şi Rxc ale 8251



CONECTAREA INTERFETELOR



4. Interfaţa cu utilizatorul

Conectarea LED-urilor

-LEDul (light-emittingg diode) este o diodă semiconductoare, care emite lumină la polarizarea directă joncțiunii p-n.

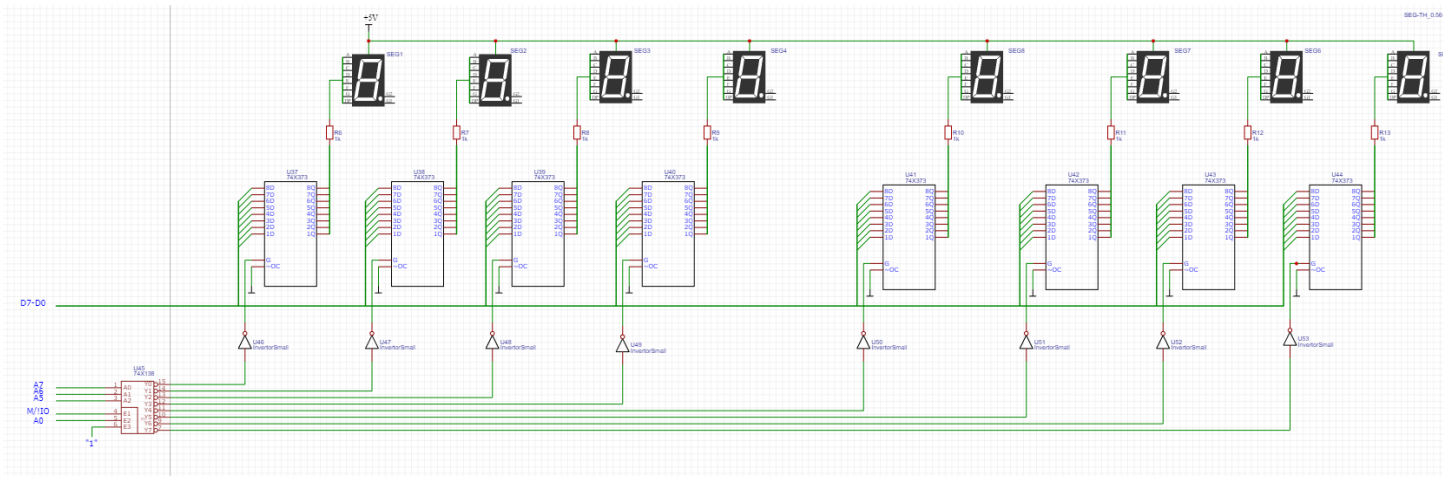
Conectarea minitastaturii cu 16 contacte

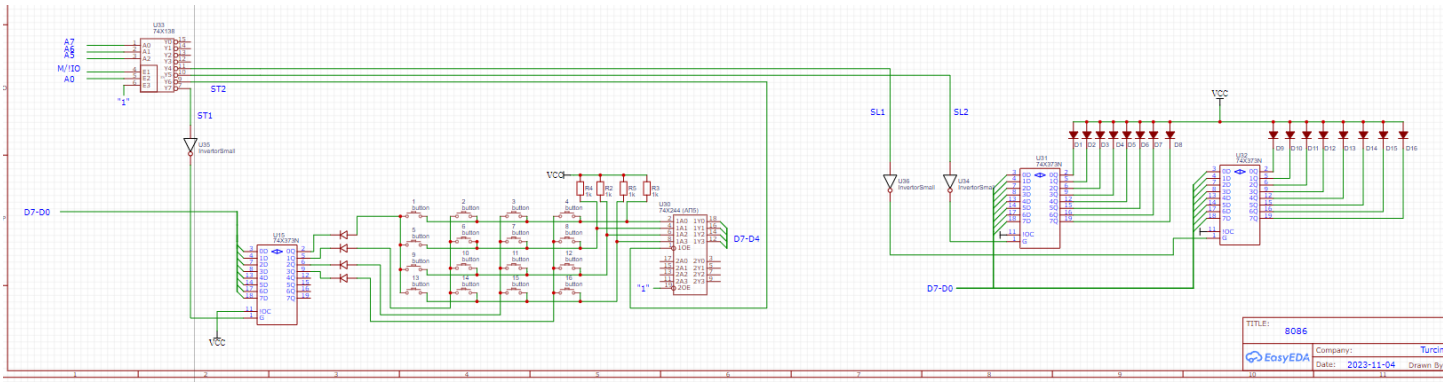
-Minitastatura este formată din 16 contacte. Pentru a identifica tasta acţionată trebuie citită tastatura. Tastatura necesită 2 porturi pentru funcţionare: unul de ieşire, prin care se activează coloana şi unul de intrare, prin care se citeşte linia activă

Conectarea modulului de afişare cu segmente

-Circuitele de afişare , pot afișa valori hexazecimale, în funcţie de setarea intrărilor A-G

CONECTAREA AFISAJELOR SI A TASTATURII





Descrierea Software

Rutinele de programare

Interfata seriala

8 biti de data, fara paritate, factor de multiplicare 16, rata de transfer 9600 bps;

```
MOV DX, 0DD0H ; adresa circuitului 8251
MOV AL, 16H ; cuvântul de comanda pentru contorul 0
OUT DX, AL
MOV AL ,10H ; constanta pentru contorul 0
MOV DX, 0DE0H
OUT DX, AL
RET
```

Rutina de transmisie caracter serial

```
TR:   IN AL,DX ; citire si testare rang TxRDY din ;cuvântul de stare
RCR AL,1
JNC TR
MOV AL,CL ; se preia data din registrul CL
MOV DX,0DD0H ;se pune în DX în functie de comutator
OUT DX,AL
RET
```

Rutina de receptie caracter serial

```
REC: IN AL,DX ; citire si testare rang RxRDY din ;cuvântul de stare
RCR AL,2
JNC REC
```

```
MOV DX, 0DD0H ;se pune in DX 0330H sau 0730H, în functie de comutator
```



```
IN AL,DX ;se preia data de la 8251
MOV CL,AL ; se depune data în registrul CL
RET
```

Rutina de programare a circuitului

```
PROG:      MOV DX, 0D56H ;sau 0B56H
MOV AL, 81H
OUT DX,AL ;trimitere cuvant de comanda
RET
```

Rutina de emisie caracter paralel

```
PAR:       IN AL,DX ; citire și testare BUSY
RCR AL,1
JNC PAR ;așteptam ca procesorul sa fie liber
MOV AL,CL ; se preia caracterul din registrul CL
MOV DX,0D50H ;se pune în DX în funcție de comutator
OUT DX,AL ;se trimite caracterul
OR AL, 01H
MOV DX,0D52H
OUT DX,AL ; /STB = 1
AND AL,00H
```

```
OUT DX,AL ; /STB = 0
OR AL,01H
OUT DX,AL ; /STB = 1
RET
```

Rutina pentru minitastatura

```
; se pune 0 pe prima coloana si se verifica daca s-au actionat tastele 0, 4, 8, C
REIA:       MOV AL,0FEH
MOV DX, 1040H ;Locația de memorie TA1
OUT DX,AL
IN AL, 08C0H;citim linia
AND AL,01 ;se verifica daca tasta apasata este TASTA 0
JZ TASTA_0
IN AL, 08C0H
AND AL,02 ;se verifica daca tasta apasata este TASTA 4
JZ TASTA_4
IN AL, 08C0H
AND AL,04 ; se verifica daca tasta apasata este TASTA 8
JZ TASTA_8
IN AL,08C0H
AND AL,08 ; se verifica daca tasta apasata este TASTA C
JZ TASTA_C
```

MOV AL, 0FDh ; Se generează 0 logic pe a doua coloană, pentru a scana în

; acelaşi mod tastele 1, 5, 9, D

OUT 1040H, AL

IN AL, 08C0H ; Citim linia

AND AL,01 ; Se verifica dacaTasta apasata este TASTA 1

JZ TASTA_1 ; Dacă da, apelăm rutina TASTA1

IN AL, 08C0H ; Citim linia

AND AL,02 ; Se verifica dacaTasta apasata este TASTA 5

JZ TASTA_5 ; Dacă da, apelăm rutina TASTA5

IN AL, 08C0H ; Citim linia

AND AL,04 ; Se verifica dacaTasta apasata este TASTA 9

JZ TASTA_9 ; Dacă da, apelăm rutina TASTA 9

IN AL, 08C0H ; Citim linia

AND AL,08 ; Se verifica dacaTasta apasata este TASTA D

JZ TASTA_D ; Dacă da, apelăm rutina TASTA D

MOV AL, 0FBH ; Se generează 0 logic pe a patra coloană, pentru a scana în acelaşi mod tastele 2,6,A,E

OUT 1040H, AL

IN AL, 08C0H ; Citim linia

AND AL,01 ; Se verifica dacaTasta apasata este TASTA 2

JZ TASTA_2 ; Dacă da, apelăm rutina TASTA2

IN AL, 08C0H ; Citim linia

AND AL,02 ; Se verifica dacaTasta apasata este TASTA 6

JZ TASTA_6 ; Dacă da, apelăm rutina TASTA6

IN AL, 08C0H ; Citim linia

AND AL,04 ; Se verifica dacaTasta apasata este TASTA A

JZ TASTA_A ; Dacă da, apelăm rutina TASTA A

IN AL, 08C0H ; Citim linia

AND AL,08 ; Se verifica dacaTasta apasata este TASTA E

JZ TASTA_E ; Dacă da, apelăm rutina TASTA E

MOV AL, 0F7H ; Se generează 0 logic pe a patra coloană, pentru a scana în acelaşi mod tastele 3, 7, B, F

OUT 1040H, AL

IN AL, 08C0Hh ; Citim linia

AND AL,01 ; Se verifica dacaTasta apasata este TASTA 3

JZ TASTA_3 ; Dacă da, apelăm rutina TASTA3

IN AL, 08C0H ; Citim linia

AND AL,02 ; Se verifica dacaTasta apasata este TASTA 7

JZ TASTA_7 ; Dacă da, apelăm rutina TASTA7

IN AL, 08C0H ; Citim linia

AND AL,04 ; Se verifica dacaTasta apasata este TASTA B

JZ TASTA_B ; Dacă da, apelăm rutina TASTA B

IN AL, 08C0H ; Citim linia

AND AL,08 ; Se verifica dacaTasta apasata este TASTA F

JZ TASTA_F ; Dacă da, apelăm rutina TASTA F

JMP REIA ; Se reia scanarea tastaturii

```
TASTA_0:        CALL DELAY ; se asteapta stabilizarea contactelor
AST_0: IN AL, 08C0H ; se citeste din nou linia si se asteapta dezactivarea
;tastei
AND AL,01
JZ AST_0
CALL DELAY
MOV CL,00H
RET

TASTA_1:    CALL DELAY ; se asteapta stabilizarea contactelor
AST_1: IN AL,08C0H ; se citeste din nou linia si se asteapta dezactivarea
;tastei
AND AL,01
JZ AST_1
CALL DELAY
MOV CL,01H
RET
…..
TASTA_E:        CALL DELAY ; se asteapta stabilizarea contactelor
AST_E: IN AL,08C0H ; se citeste din nou linia si se asteapta dezactivarea
;tastei
AND AL,08
JZ AST_E
CALL DELAY
MOV CL,0EH
```

```
RET

TASTA_F:        CALL DELAY ; se asteapta stabilizarea contactelor
ASTF: IN AL,08C0H ; se citeste din nou linia si se asteapta dezactivarea
;tastei
AND AL,08
JZ AST_F
CALL DELAY
MOV CL,0FH
RET
```

Rutina de aprindere/stingere a unui led

Starea ledurilor: 0 - aprins

1 - stins

```
MOV DX, 1440H
CMP AH,1
JE STINGE
MOV AL,0FEH
OUT DX,AL
RET
STINGE:MOV AL,0FFH
OUT DX,AL
RET
```

Rutina de afisare a unui caracter hexa pe un rang cu segmente

Intrare: AH – rangul pe care se va afisa carcterul

AL – caracterul ce va fi afisat ;vedem pe care rang se face afisarea

```
CMP AH,0
JE RANG_0
CMP AH,1
JE RANG_1
CMP AH,2
JE RANG_2
CMP AH,3
JE RANG_3
CMP AH,4
JE RANG_4
CMP AH,5
JE RANG_5
CMP AH,6
JE RANG_6
CMP AH,7
JE RANG_7

RANG_0: MOV DX, 0040H
JMP AFIS
RANG_1: MOV DX, 00C0H
JMP AFIS
```

```
RANG_2:  MOV DX, 0440H
JMP AFIS
RANG_3:  MOV DX, 04C0H
JMP AFIS
RANG_4:  MOV DX, 0040H
JMP AFIS
RANG_5:  MOV DX, 00C0H
JMP AFIS
RANG_6:  MOV DX, 0440H
JMP AFIS
RANG_7:  MOV DX, 04C0H
JMP AFIS


AFIS:;testam caracterul din AL
CMP AL,00H
JE CAR_0
CMP AL,01H
JE CAR_1
CMP AL,02H
JE CAR_2
…
CMP AL,0EH
JE CAR_E
```

```
CMP AL,0FH
JE CAR_F
CAR_0: MOV AL,C0H ;combinatia de segmente
OUT AL,DX
RET
CAR_1: MOV AL,F9H
OUT AL,DX
RET
CAR_2: MOV AL,A4H
OUT AL,DX
RET
….
CAR_E: MOV AL,86H
OUT AL,DX
RET
CAR_F: MOV AL,8EH
OUT AL,DX
RET
```

BIBLIOGRAFIE

-M. Popa, Proiectarea microsistemelor digitale; Orizonturi Universitare, Timişoara, 2003

-M. Popa, Sisteme cu microprocesoare; Orizonturi Universitare, Timişoara, 2003

-https://sites.google.com/site/labpmd/proiect-pmd

-https://en.wikipedia.org/wiki/MAX232

https://www.uotechnology.edu.iq/depeee/lectures/3rd/Shared%20all/Microprocessor%20engineering%201/part2_P.pdf