UNIVERSITATEA POLITEHNICA TIMIȘOARA

Facultatea de Automatică și Calculatoare

Secția

Calculatoare și Tehnologia Informației

**PROIECTAREA MICROSISTEMELOR DIGITALE**

- MICROSISTEM CU MICROPROCESORUL 8086 –

Student: **Galuska Vlad-Cristian**

Anul: **III**

Grupa: **3.1**

An universitar: **2019-2020**

**Tema proiectului:**

Să se proiecteze un microsistem cu următoarea structură:

* Unitate centrală cu microprocesorul 8086
* 256 KB memorie EPROM, utilizând circuite 27C1024
* 64 KB memorie SRAM, utilizând circuite 62256
* Interfață serială, cu circuitul 8251, plasată în zona 03A0H – 03A2H sau 0BA0H – 0BA2H, în funcție de poziția microcomutatorului S1
* Interfață paralelă, cu circuitul 8255, plasată în zona 04D0H – 04D6H sau 0CD0H – 0CD6H, în funcție de poziția microcomutatorului S2
* O minitastatură cu 16 contacte;
* 15 led-uri
* Două module de afișare cu segmente, cu 8 ranguri

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.

**Descrierea Hardware**

**Microprocesorul 8086**

Primul microprocesor pe 16 biţi care a cunoscut o largă utilizare.

Apariţia lui a fost urmată la scurt timp de o familie de componente: generatorul de tact 8284, controlerul de magistrală 8288, coprocesorul matematic 8087 şi coprocesorul de intrare / ieşire 8089.

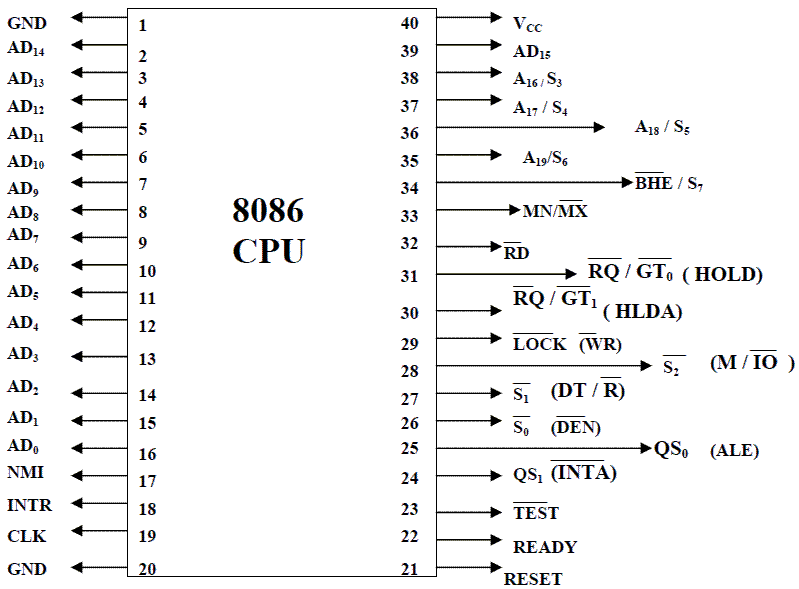
**Moduri de lucru:**

* minim: pentru aplicaţii relativ simple, în care microprocesorul generează el însuşi semnalele necesare transferurilor cu memoria şi cu porturile de intrare/ ieşire,
* maxim: pentru aplicaţii complexe, inclusiv sisteme multiprocesor, în care semnalele de comandă pentru memorii şi porturi sunt generate de un controler de magistrală, 8288,

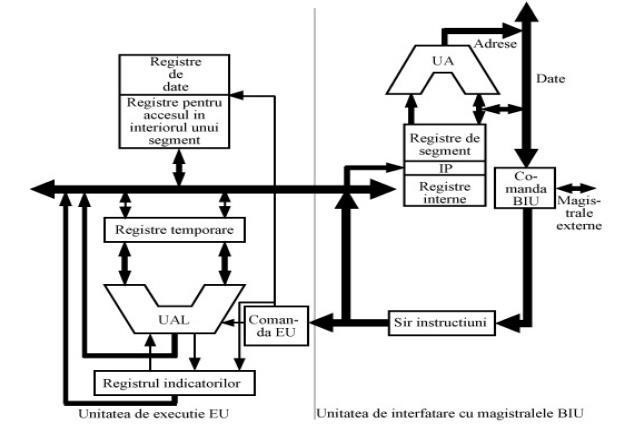
Trecerea dintr-un mod în altul se face prin hardware: există terminalul MN/ /MX la care, prin 1 logic se cere modul minim iar prin 0 logic se cere modul maxim.

**Caracteristici:**

* registrele interne şi magistrala de date externă sunt pe 16 biţi;
* posibilitatea de a adresa direct 1 Mo de memorie;
* viteză mărită de lucru datorită atât frecvenţei tactului cât şi unei structuri interne bazată pe conceptul de suprapunere care permite aducerea din memorie, în avans, a instrucţiunilor în timpul unor cicluri fără acces la magistrale;
* poate acoperi o gamă largă de aplicaţii datorită celor două moduri de lucru ale sale: minim şi maxim,
* magistralele de date şi adrese sunt multiplexate iar o parte dintre terminalele de comandă au rol dublu; aceasta a permis încapsularea circuitului într-o capsulă cu doar 40 terminale.



Structura interna:



|  |  |  |
| --- | --- | --- |
| **Pini** | **Simbol** | **Descriere** |
| 1 | GND | Masa |
| 2-16 | AD14-AD0 | Magistrale multiplexate de adrese/date pe 16 biți |
| 17 | NMI | Întrerupere nemascabilă |
| 18 | INTR | Întrerupere mascabilă |
| 19 | CLK | Generează semnale clock care sincronizează operația procesorului |
| 20 | GND | Masa |
| 21 | RESET | Încheie activitatea și setează următoarele PSW – clear; IP – 0000H; CS – FFFFH; DS – 0000H; SS – 0000H; ES – 0000H; Queue – empty. |
| 22 | READY | Confirmarea de la memorie sau interfața I/O că procesorul poate completa actualul ciclu de magistrale. |
| 23 | TEST | Este folosit doar de instrucțiunea WAIT. Microprocesorul intră într-o stare de așteptare până când este aplicat un 0 pe acest pin. |
| 24-31 |  | Definiția depinde de mod. |
| 32 | RD | Indică faptul că urmează să se facă o citire de memorie sau o citire I/O |
| 33 | MN/MX | Procesorul se află în modul minim când se aplică o tensiune de +5V și în modul maxim când este legat la masă. |
| 34 | BHE/s7 | Este ieşire cu trei stări, în timpul stării T1 se activează când are loc un transfer pe octetul superior al magistralei de date, validând acest transfer, iar în stările T2 - T4 este bit de stare |
| 35-38 | A19/s6 – A16/s3 | în prima perioada clock adresele A19-A16 sunt folosite ca output și în rest sunt folosite semnalele de status S6-S3. S6 e mereu 0, S5 da setarea lui IF, iar în functie de S4 și S3, registrul folosit poate fi ES, SS, CS sau DS |
| 39 | AD15 | La fel ca AD14-AD0 |
| 40 | Vcc | Primește tensiunea de +5V |

**Generatorul de tact 8284A**- generează tactul către microprocesor(CLK) și pentru circuitele specializate pentru interfețe(PCLK);generează semnalul READY către microprocesor,sincronizându-l cu tactul și generează semnalul de inițializare, RESET, către microprocesor, sincronizându-l cu tactul.

|  |
| --- |
|  |
| Schema circuitului 8284A |

**Circuitul amplificator/separator 74LS245**

Circuitul 74LS245 este folosit pentru amplificarea/separarea magistralelor bidirecționale. Astfel, el va avea drept intrări rangurile 15–0 ale magistralei multiplexate de adrese/date ale microprocesorului 8086, ieșirea DT//R a acestuia, care indică sensul de circulare a datelor, precum și iesirea /DEN, care validează transferul de date pe magistrală.

|  |
| --- |
|  |
| Configurația terminalelor |

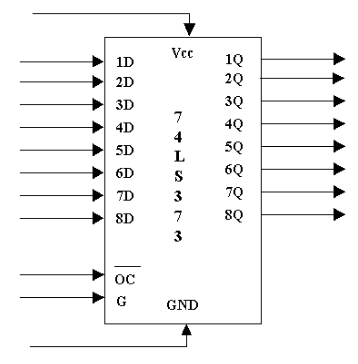
|  |
| --- |
|  |
| Schemă internă |

Funcționare:

|  |  |  |  |
| --- | --- | --- | --- |
| **/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 |

**Circuit registru 73LS373**

Circuitul 74LS373 este un registru cu 8 ranguri, cu 3 stări. Circuitul este folosit pentru demultiplexarea și amplificarea liniilor de adrese ale microprocesorului 8086, întrucât pe durata unui acces la memorie este necesar ca informația de pe liniile de adrese/date să rămână stabilă.



Funcționarea circuitului este următoarea:

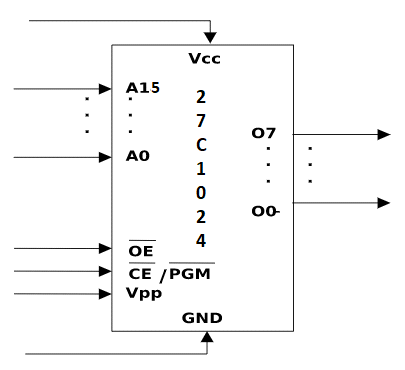
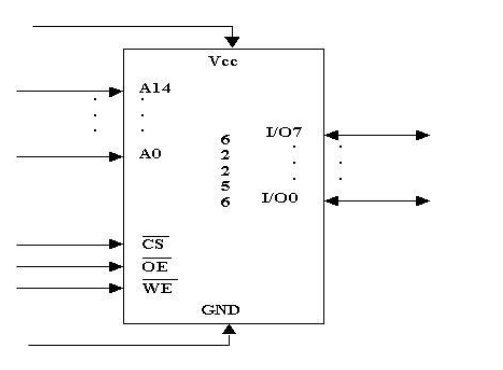
|  |  |  |
| --- | --- | --- |
| **/OC** | **G** | **8Q – 1Q** |
| 0 | 0 | Vechiul conţinut |
| 0 | 1 | 8D – 1D |
| 1 | X | A 3-a stare |

**Conectarea memoriilor**

Vom folosi memoriile:

EPROM de 256KB utilizând 2 circuite 27C1024 cu capacitatea=128KB

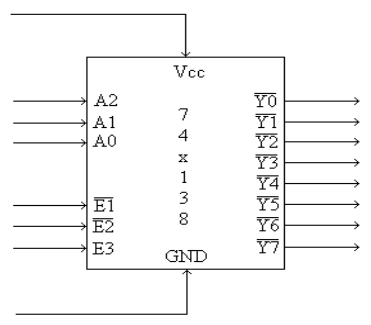
SRAM de 64KB utilizând 2 circuite 62256: cu capacitatea=32KB



Semnalele sunt:

* **A0-A14** - intrări de adrese
* **I/O0-I/O7** - ieșiri de date
* **/CS**=Chip Select - selectarea circuitului
* **/OE**=Output Enable – specifică că se citește
* **/WE**=Write Enable –spune că se citește
* **/CE**=Chip Enable –selectarea circuitului
* **/PGM**=Program Enable
* **Vpp**=Program Supply Voltage

Decodificatorul 74x138:



256KB=28\*210B=218=>A18-1-A0

64KB=26\*210B=216=>A16-1-A0

B1: 00000H - 07FFFH (256KB –EPROM)

B2: 80000H - 9FFFFH (64KB – SRAM)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | A19 | A18 | A17 | A16 | A15 | A14 | A13 | A12 | A11 | A10 | A9 | A8 | A7 | A6 | A5 | A4 | A3 | A2 | A1 | A0 |
| B0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| B1 | 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 | 1 | 1 |

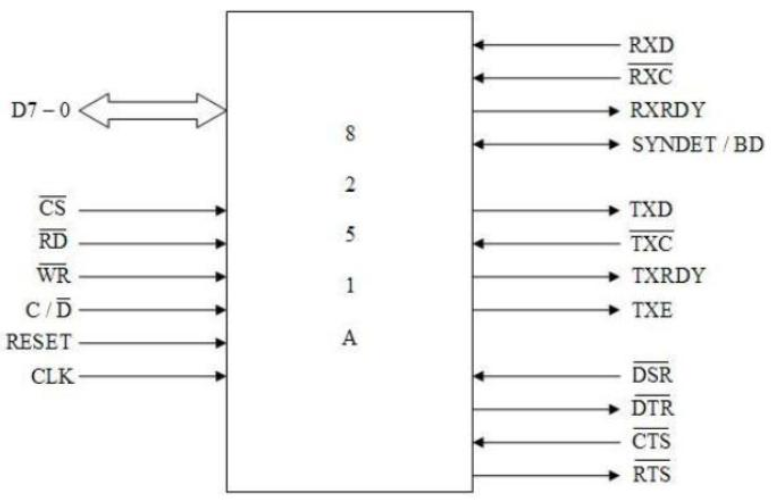
/SelSRAM=A19\*/A18\*/A17 (/Y4)

/SelEPROM=/A19

**Interfața serială**

**Circuitul 8251** este produs de Intel, standard numit și USART (Universal Synchronous/Asynchronous Receiver/Transmitter). A fost creat pentru comunicarea datelor cu microprocesoarele Intel și în mod specific microprocesorul Intel 8086.

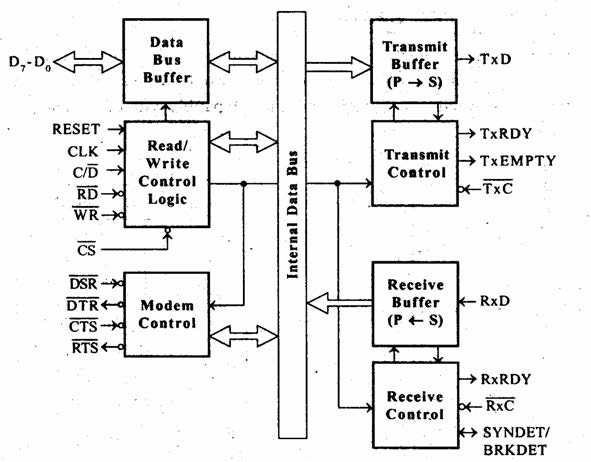
Schemă:



Modul de funcționare al **circuitului 8251**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **/CS** | **/RD** | **/WR** | **C//D** | **Operație** |
| 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 |

Structura interna a circuitului **8251**



Transmisia serială permite transferul de date între sisteme care se află la distanțe mari unul de altul folosind un număr redus de fire.Transferul datelor se face bit după bit.

Semnificație semnale:

D0-D7=magistrala de date a unității centrale.

RESET=intrare de inițializare.

CLK=intrare de tact.

C//D(Control/Data)=selecție comandă/date(1=comandă,0-date).

/CS(Chip Select)=specifică dacă comunicarea dintre 8251A și unitatea centrală este activă.

/RD(Read)=citire date sau stare de către unitatea centrală.

/WR(Write)=scriere date sau cuvânt de control de către unitatea centrală.

RXD(Reception Data)=intrare serială de date.

TXD(Transmission Data)=ieșire serială de date.

RXRDY(Receptor Ready)=caracter recepționat,ce poate fi citit de procesor.

TXRDY(Transmitter Ready)=transmițătorul e disponibil pentru un nou caracter.

TXE(Transmitter Empty)=transmițătorul este vid(nu are date de transmis).

/RXC(Reception Clock)=intrare de tact pentru recepție.

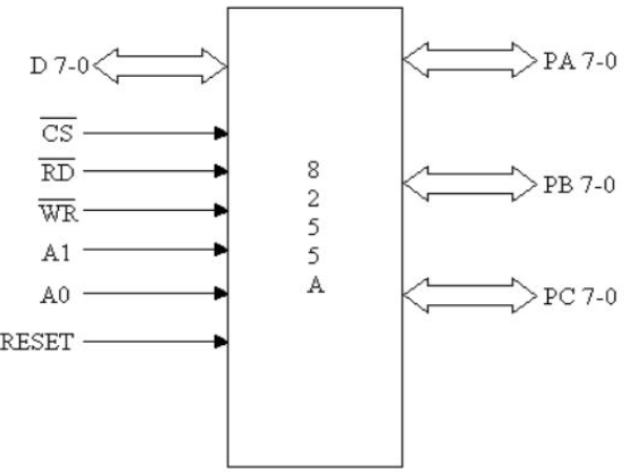
/TXC(Transmission Clock)=intrare de tact pentru transmisie.

**Interfața paralelă**

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.

**Circuitul 8255** a fost dezvoltat de Intel în prima parte a anilor 1970 pentru microprocesorul Intel 8085. Mai târziu a început să fie utilizat cu microprocesorul Intel 8086.

Schemă:



Modul de funcționare al **circuitului 8255**



Semnificație semnale:

D0-D7=magistrala de date a unității centrale.

/RD(Read)=citire date sau stare de către unitatea centrală.

/WR(Write)=scriere date sau cuvânt de control de către unitatea centrală.

/CS(Chip Select)=specifică dacă comunicarea dintre 8255A și unitatea centrală este activă.

A0 și A1=selecție port 0 și selecție port 1.

RESET=intrare de inițializare.

Porturile A,B și C=porturi de citire și scriere

**Decodificarea porturilor interfeței seriale și interfeței paralele**

Interfața serială este plasată în zona 03A0H-03A2H sau 0BA0H-0BA2H, în funție de poziția microcomutatorului S1,iar interfața paralelă este plasată în zona 04D0H-04D6H sau 0CD0H-0CD6H, în funcție de poziția microcomutatorului S2.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| A15 | A14 | A13 | A12 | A11 | A10 | A9 | A8 | A7 | A6 | A5 | A4 | A3 | A2 | A1 | A0 | Nume port |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 03A0H-ISA |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 03A2H-ISA |
| 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0BA0H-ISB |
| 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0BA2H-ISB |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 04D0H-IFA |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 04D2H-IFB |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 04D4H-IFC |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 04D6H-IFR |
| 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0CD0H-IFA |
| 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0CD2H-IFB |
| 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0CD4H-IFC |
| 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0CD6H-IFR |

/SELISA=/A11 \* A5 \* /A4 => /Y2

/SELISB=A11 \*A5 \*/A4 => /Y6

/SELIPA=/A11 \*/A5 \*A4 => /Y1

/SELIPB=A11 \*/A5 \*A4 => /Y5

**Conectarea LED-urilor,afișajelor și a minitastaturii:**

Pentru conectarea LED-urilor la unitatea centrală cu microprocesor 8086, sunt necesare porturi de ieșire cu posibilitatea de memorare.Ca port de ieșire am folosit circuitul registru 74LS373.Se folosesc 15 LED-uri și vom avea nevoie de 2 circuite 74LS373 conectate la decodificatorul adreselor de port.

Comandarea unui modul de afișaj cu segmenete cu mai multe ranguri se face prin soluția nemultiplexată,folosindu-se un registru pentru fiecare rang, registrele fiind comandate ca 2 porturi de ieșire ce memorează informațiile ce se vor afișa. Se foloșește soluția cu anod comun.Se folosesc module de afișare cu segmente și 8 cicuite 74LS373,circuitele 74LS373 fiind conectate la decodificatorul adreselor de port.

Pentru conectarea minitastaturii este necesar un port de ieșire cu posibilitatea de memorare(registru) și un port de intrare(poartă cu trei stări).Se baleiază coloanele cu un singur 0 și se citesc liniile pentru citirea tastaturii astfel,în portul de ieșire se va scrie succesiv căte o combinație cu un singur 0 logic în rangul 0,apoi în rangul 1, apoi în rangul 2,și apoi în rangul 3, după care ciclul se reia.În acest caz se va scrie un 0 pe coloanele minitastaturii.Fiecare scriere e urmată de citirea liniilor minitastaturii. Dacă în combinația citită exista un 0 logic înseamnă că a fost acționată o tastă, altfel nu.Dacă a fost acționată o tastă,ea va fi identificată din poziția pe care o ocupă 0 logic generat pe coloane și 0 logic citit pe linii.Vom avea nevoie de un circuit 74LS373 conectat la decodificatorul adreselor de port și un circuit 74LS244.



**Decodificarea porturilor pentru LED-uri,afișaje și minitastatură:**

Pentru decodificare adreselor de port se folosesc 2 circuite 74LS138.

Adrese de port pentru LED-uri:1000H și 1002H.

Adrese de port pentru tastatură:1400H(port de intrare) și 1402H(port de ieșire).

Adrese de port pentru afișaje: 0040H, 00C0H, 0440H,04C0H,0840H,08C0H,0C40H, 0CC0H.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| A15 | A14 | A13 | A12 | **A11** | **A10** | A9 | A8 | A7 | A6 | A5 | A4 | A3 | A2 | **A1** | A0 | Nume port |
| 0 | 0 | 0 | 1 | **0** | **0** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | **0** | 0 | 1000H-LD1 |
| 0 | 0 | 0 | 1 | **0** | **0** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | **1** | 0 | 1002H-LD2 |
| 0 | 0 | 0 | 1 | **0** | **1** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | **0** | 0 | 1400H-TIN |
| 0 | 0 | 0 | 1 | **0** | **1** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | **1** | 0 | 1402H-TIE |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0040H-AF1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 00C0H-AF2 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0440H-AF3 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 04C0H-AF4 |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0840H-AF5 |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 08C0H-AF6 |
| 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0C40H-AF7 |
| 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0CC0H-AF8 |

/SELLD1=/A11 \*/A10 \*/A1 => /Y0

/SELLD2=/A11 \*/A10 \*A1 => /Y1

/SELTIN=/A11 \*A10 \*/A1 => /Y2

/SELTIE=/A11 \*A10 \*A1 => /Y3

/SELAF1=/A11 \*/A10 \*/A7 => /Y0

/SELAF2=/A11 \*/A10 \*A7 => /Y1

/SELAF3=/A11 \*A10 \*/A7 => /Y2

/SELAF4=/A11 \*A10 \*A7 => /Y3

/SELAF5=A11 \*/A10 \*/A7 => /Y4

/SELAF6=A11 \*/A10 \*A7 => /Y5

/SELAF7=A11 \*A10 \*/A7 => /Y6

/SELAF8=A11 \*A10 \*A7 => /Y7

**Implementare subrutine:**

**Rutina de programare a circuitului 8251**

Cuvântul de mod:2 biți de stop:11,control paritate dezactivat:00,lungime caracter 8 biți:11 ,factor de multiplicare 16 :10 (1100 1110B=0CEH)

Cuvântul de comandă:operare normală pentru sincronizare:0; și reinițializare:0;comanda /RTS: 1-/RTS=0; comanda indicatori eroare: 1-anulare; break:0-operare normală;comanda recepție: 1-activare;comanda /DTR: 1-/DTR=0;comanda transmisie: 1-activare (0001 0101B=15H).

Adresele de port:03A0H(0BA0H)-comenzi/stări.

MOV DX,03A0H ;sau 0BA0H

MOV AL,0CEH ; încarcă 0CEH-cuvânt de mod pentru modul sincron

OUT DX,AL; octet de ieșire in AL la adresa de port in DX

MOV AL,15H ;încarcă 15H-cuvânt de comandă pentru transmisie sau recepție

OUT DX,AL

RET; reintoarcere din procedura

**Rutina de programare a circuitului 8255**

Cuvântul de mod:modul de intrare/ieșire cu dialog:01;port A:0-ieșire;port C superior:0-ieșire;Grup B:selecție mod:0-mod 0,port B:0-ieșire,port C inferior:1- intrare(0100 0001B=81H).

MOV DX,04D6H;sau 0CD6H,adică adresele de port pentru RCC

MOV AL,81H ;încarcă 81H-cuvânt de comandă

OUT DX,AL; octet de ieșire in AL la adresa lui DX

RET

**Rutine de emisie/recepție caracter pe interfața serială**

Emisie:verificăm bitul corespunzător lui TxRDY,prin care 8251 anunță procesorul că a terminat de transmis pe linie un caracter,l-a serializat și îi cere altul.

TR: MOV DX,03A0H;sau 0BA0H,adică adresele de port pentru comenzi/stări

IN AL,DX;citire și testare BUSY

RCR AL,1;verificăm bit-ul 0 corespunzător lui TxRDY

JNC TR;dacă bitul corespunzător e 0,reluăm operația

MOV AL,CL;se preia caracterul din registrul CL

MOV DX,03A2H;sau 0BA2H,adică adresele de port pentru date

OUT DX,AL;transmitem caracterul dorit

RET

Recepție:verificăm bitul corespunzător lui RxRDY,prin care 8251 anunță procesorul că a terminat de preluat un caracter de pe linia serială,l-a asamblat și poate să i-l predea.

REC:MOV DX,03A0H

IN AL,DX

RCR AL,2:verificăm bit-ul 1 corespunzător lui RxRDY

JNC REC;dacă bitul corespunzător e 0,reluăm operația

MOV DX,03A2H

IN AL,DX;se preia data de la 8251

MOV CL,AL;se depune data în registrul CL

RET

**Rutina de emisie caracter pe interfața paralelă**

Emisie:verificăm bitul corespunzător lui BUSY,pentru a vedea dacă receptorul este liber,microprocesorul așteaptă până cănd receptorul este liber și apoi trimite data.

PAR:MOV DX,04D4H;sau 0CD4H,adică adresele de port pentru portul C

IN AL,DX;

RCR AL,1;verificăm bit-ul 0 corespunzător lui BUSY

JC PAR;daca bitul respectiv e 1,reluăm operația

MOV AL,CL;se preia caracterul din registrul CL

MOV DX,04D0H;sau 0CD0H,adică adresele de port pentru portul A

OUT DX,AL;

OR AL,01H;modificăm bitul 0 pe 1

MOV DX,04D2H;sau 0CD2H,adică adresele de port pentru portul B

OUT DX,AL;/STB=1 ,adică nu sunt date de citit

AND AL,00H

MOV DX,04D2H

OUT DX,AL;/STB=0,adică sunt date de citit

OR AL,01H

MOV DX,04D2H;/STB=1,adică nu mai sunt date de citit

OUT DX,AL

RET

**Rutina de scanare a minitastaturii**

0 1 2 3

4 5 6 7

8 9 A B

C D E F

TASTATURA:MOV AL,0FEH;0FEH->11111110 ,generăm 0 logic pe prima coloană pentru a scana tastele 0,4,8,C

MOV DX,1400H

OUT DX,AL

MOV DX,1402H

IN AL,DX;se citește linia

AND AL,01H;se verifică dacă tasta apăsată este tasta 0

JZ TASTA0

IN AL,DX

AND AL,02H;se verifică dacă tasta apăsată este tasta 4

JZ TASTA4

IN AL,DX

AND AL,04H;se verifică dacă tasta apăsată este tasta 8

JZ TASTA8

IN AL,DX

AND AL,08H;se verifică dacă tasta apăsată este tasta C

JZ TASTAC

MOV AL,0FDH;generăm 0 logic pe a doua coloană pentru a scana tastele 1,5,9,D

MOV DX,1400H

OUT DX,AL

MOV DX,1402H

IN AL,DX;se citește linia

AND AL,01H;se verifică dacă tasta apăsată este tasta 1

JZ TASTA1

IN AL,DX

AND AL,02H;se verifică dacă tasta apăsată este tasta 5

JZ TASTA5

IN AL,DX

AND AL,4H;se verifică dacă tasta apăsată este tasta 9

JZ TASTA9

IN AL,DX

AND AL,08H;se verifică dacă tasta apăsată este tasta D

JZ TASTAD

MOV AL,0FBH;generăm 0 logic pe a doua coloană pentru a scana tastele 2,6,A,E

MOV DX,1400H

OUT DX,AL

MOV DX,1402H

IN AL,DX;se citește linia

AND AL,01H;se verifică dacă tasta apăsată este tasta 2

JZ TASTA2

IN AL,DX

AND AL,02H;se verifică dacă tasta apăsată este tasta 6

JZ TASTA6

IN AL,DX

AND AL,04H;se verifică dacă tasta apăsată este tasta A

JZ TASTAA

IN AL,DX

AND AL,08H;se verifică dacă tasta apăsată este tasta E

JZ TASTAE

MOV AL,0F7H;generăm 0 logic pe a doua coloană pentru a scana tastele 3,7,B,F

MOV DX,1400H

IN AL,DX;se citește linia

AND AL,01H;se verifică dacă tasta apăsată este tasta 3

JZ TASTA3

IN AL,DX

AND AL,02H;se verifică dacă tasta apăsată este tasta 7

JZ TASTA7

IN AL,DX

AND AL,04H;se verifică dacă tasta apăsată este tasta B

JZ TASTAB

IN AL,DX

AND AL,08H;se verifică dacă tasta apăsată este tasta F

JZ TASTAF

JMP TASTATURA;se reia scanarea tastaturii

TASTA0:CALL TIM;se așteaptă oprirea vibrațiilor

TAS0:MOV DX,1400H

IN AL,DX

AND AL,01H

JZ TAS0;se așteaptă dezactivarea tastei

CALL TIM;se așteaptă oprirea vibrațiilor

RET

TASTA1:CALL TIM;se așteaptă oprirea vibrațiilor

TAS1:MOV DX,1400H

IN AL,DX

AND AL,01H

JZ TAS1;se așteaptă dezactivarea tastei

CALL TIM;se așteaptă oprirea vibrațiilor

RET

TASTA2:CALL TIM;se așteaptă oprirea vibrațiilor

TAS2:MOV DX,1400H

IN AL,DX

AND AL,01H

JZ TAS2;se așteaptă dezactivarea tastei

CALL TIM;se așteaptă oprirea vibrațiilor

RET

TASTA3:CALL TIM;se așteaptă oprirea vibrațiilor

TAS3:MOV DX,1400H

IN AL,DX

AND AL,01H

JZ TAS3;se așteaptă dezactivarea tastei

CALL TIM;se așteaptă oprirea vibrațiilor

RET

TASTA4:CALL TIM;se așteaptă oprirea vibrațiilor

TAS4:MOV DX,1400H

IN AL,DX

AND AL,02H

JZ TAS4;se așteaptă dezactivarea tastei

CALL TIM;se așteaptă oprirea vibrațiilor

RET

TASTA5:CALL TIM;se așteaptă oprirea vibrațiilor

TAS5:MOV DX,1400H

IN AL,DX

AND AL,02H

JZ TAS5;se așteaptă dezactivarea tastei

CALL TIM;se așteaptă oprirea vibrațiilor

RET

TASTA6:CALL TIM;se așteaptă oprirea vibrațiilor

TAS6:MOV DX,1400H

IN AL,DX

AND AL,02H

JZ TAS6;se așteaptă dezactivarea tastei

CALL TIM;se așteaptă oprirea vibrațiilor

RET

TASTA7:CALL TIM;se așteaptă oprirea vibrațiilor

TAS7:MOV DX,1400H

IN AL,DX

AND AL,02H

JZ TAS7;se așteaptă dezactivarea tastei

CALL TIM;se așteaptă oprirea vibrațiilor

RET

TASTA8:CALL TIM;se așteaptă oprirea vibrațiilor

TAS8:MOV DX,1400H

IN AL,DX

AND AL,04H

JZ TAS8;se așteaptă dezactivarea tastei

CALL TIM;se așteaptă oprirea vibrațiilor

RET

TASTA9:CALL TIM;se așteaptă oprirea vibrațiilor

TAS9:MOV DX,1400H

IN AL,DX

AND AL,04H

JZ TAS9;se așteaptă dezactivarea tastei

CALL TIM;se așteaptă oprirea vibrațiilor

RET

TASTAA:CALL TIM;se așteaptă oprirea vibrațiilor

TASA:MOV DX,1400H

IN AL,DX

AND AL,04H

JZ TASA;se așteaptă dezactivarea tastei

CALL TIM;se așteaptă oprirea vibrațiilor

RET

TASTAB:CALL TIM;se așteaptă oprirea vibrațiilor

TASB:MOV DX,1400H

IN AL,DX

AND AL,04H

JZ TASB;se așteaptă dezactivarea tastei

CALL TIM;se așteaptă oprirea vibrațiilor

RET

TASTAC:CALL TIM;se așteaptă oprirea vibrațiilor

TASC:MOV DX,1400H

IN AL,DX

AND AL,08H

JZ TASC;se așteaptă dezactivarea tastei

CALL TIM;se așteaptă oprirea vibrațiilor

RET

TASTAD:CALL TIM;se așteaptă oprirea vibrațiilor

TASD:MOV DX,1400H

IN AL,DX

AND AL,08H

JZ TASD;se așteaptă dezactivarea tastei

CALL TIM;se așteaptă oprirea vibrațiilor

RET

TASTAE:CALL TIM;se așteaptă oprirea vibrațiilor

TASE:MOV DX,1400H

IN AL,DX

AND AL,08H

JZ TASE;se așteaptă dezactivarea tastei

CALL TIM;se așteaptă oprirea vibrațiilor

RET

TASTAF:CALL TIM;se așteaptă oprirea vibrațiilor

TASF:MOV DX,1400H

IN AL,DX

AND AL,08H

JZ TASF;se așteaptă dezactivarea tastei

CALL TIM;se așteaptă oprirea vibrațiilor

RET

**Rutina de aprindere stingere a unui led**

0-aprins

1-stins

Secvența ca LED-urile să lumineze:

MOV DX,1000H;primul grup de LED-uri

MOV AL,11111110;punem 0 pe pozitia 0 deci LED-ul 0

OUT DX,AL

MOV DX,1000H;primul grup de LED-uri

MOV AL,11111101;punem 0 pe pozitia 1 deci LED-ul 1

OUT DX,AL

MOV DX,1000H;primul grup de LED-uri

MOV AL,11111011;punem 0 pe pozitia 2 deci LED-ul 2

OUT DX,AL

MOV DX,1000H;primul grup de LED-uri

MOV AL,01111111;punem 0 pe pozitia 7 deci LED-ul 7

OUT DX,AL

MOV DX,1002H;al doilea grup de LED-uri

MOV AL,11111110;punem 0 pe pozitia 0 deci LED-ul 8

OUT DX,AL

MOV DX,1002H;al doilea grup de LED-uri

MOV AL,10111111;punem 0 pe pozitia 6 deci LED-ul 14

OUT DX,AL

RET

MOV DX,1000H

MOV AL,00H;aprind toate primele 8 LED-uri

OUT DX,AL

MOV DX,1002H

MOV AL,00H;aprind restul de 7 LED-uri

OUT DX,AL

RET

MOV DX,1000H

MOV AL,FFH;sting primele 8 LED-uri

OUT DX,AL

MOV DX,1002H

MOV AL,FFH;sting restul de 7 LED-uri

OUT DX,AL

RET

**Rutina de afișare a unui caracter hexa pe un rang cu segmente**

SEG:

MOV DX,0040H;afișăm caractere pe primul rang

CAR0:MOV AL,0C0H ;p=1,g=1,f=0,e=0,d=0,c=0,b=0,a=0

OUT DX,AL

RET

CAR1:MOV AL,0F9H ;p=1,g=1,f=1,e=1,d=1,c=0,b=0,a=1

OUT DX,AL

RET

CAR2:MOV AL,0A4H ;p=1,g=0,f=1,e=0,d=0,c=1,b=0,a=0

OUT DX,AL

RET

CAR3:MOV AL,0B0H ;p=1,g=0,f=1,e=1,d=0,c=0,b=0,a=0

OUT DX,AL

RET

CAR4:MOV AL,099H ;p=1,g=0,f=0,e=1,d=1,c=0,b=0,a=1

OUT DX,AL

RET

CAR5:MOV AL,092H ;p=1,g=0,f=0,e=1,d=0,c=0,b=1,a=0

OUT DX,AL

RET

CAR6:MOV AL,082H ;p=1,g=0,f=0,e=0,d=0,c=0,b=1,a=0

OUT DX,AL

RET

CAR7:MOV AL,0F8H ;p=1,g=1,f=1,e=1,d=1,c=0,b=0,a=0

OUT DX,AL

RET

CAR8:MOV AL,080H ;p=1,g=0,f=0,e=0,d=0,c=0,b=0,a=0

OUT DX,AL

RET

CAR9:MOV AL,090H ;p=1,g=0,f=0,e=1,d=0,c=0,b=0,a=0

OUT DX,AL

RET

CARA:MOV AL,088H ;p=1,g=0,f=0,e=0,d=1,c=0,b=0,a=0

OUT DX,AL

RET

CARB:MOV AL, 080H ;p=1,g=0,f=0,e=0,d=0,c=0,b=0,a=0

OUT DX,AL

RET

CARC:MOV AL, 0C6H ;p=1,g=1,f=0,e=0,d=0,c=1,b=1,a=0

OUT DX,AL

RET

CARD:MOV AL, 0C0H ;p=1,g=1,f=0,e=0,d=0,c=0,b=0,a=0

OUT DX,AL

RET

CARE:MOV AL, 086H ;p=1,g=0,f=0,e=0,d=0,c=1,b=1,a=0

OUT DX,AL

RET

CARF:MOV AL, 08EH ;p=1,g=0,f=0,e=0,d=1,c=1,b=1,a=0

OUT DX,AL

RET

**Bibliografie**

* <https://sites.google.com/site/uptacpmd/proiect-pmd>

* <https://sites.google.com/site/uptacpmd/Laborator>

* <https://datasheet4u.com/datasheet-pdf/Intel/8086/pdf.php?id=544568>

* <https://pdf1.alldatasheet.com/datasheet-pdf/view/124124/INTEL/8284A.html>
* “Sisteme cu microprocesoare”-MIRCEA POPA

* <https://pdf1.alldatasheet.com/datasheet-pdf/view/37318/SAMSUNG/62256.html>

* <https://pdf1.alldatasheet.com/datasheet-pdf/view/66100/INTEL/8255A.html>

* <https://pdf1.alldatasheet.com/datasheet-pdf/view/74454/MCNIX/27C1024.html>

* <http://map.grauw.nl/resources/midi/intel_8251.pdf>