**UNIVERSITATEA POLITEHNICA DIN TIMIŞOARA**

**FACULTATEA DE AUTOMATICĂ ŞI CALCULATOARE**

Proiectarea cu Microprocesoare

Microsistem cu microprocesorul 8086

Coordonator: Ș. l. dr. ing. Sergiu NIMARĂ

nume: Solomon Alexandru-Paul

anul: 3

Tema proiectului

Să se proiecteze un microsistem cu următoarea structură:

- unitate centrală cu microprocesorul 8086;

- 128 KB memorie EPROM, utilizând circuite 27C512;

- 64 KB memorie SRAM, utilizând circuite 62256;

- interfaţă serială, cu circuitul 8251, plasată în zona 04D0H – 04D2H sau 05D0H – 05D2H, în funcţie de poziţia microcomutatorului S1;

- interfaţă paralelă, cu circuitul 8255, plasată în zona 0250H – 0256H sau 0A50H – 0A56H, în funcţie de poziţia microcomutatorului S2;

- o minitastatură cu 9 contacte;

- 10 led-uri;

- un modul de afişare cu 7 segmente, cu 8 ranguri (se pot afişa maxim 8 caractere hexa simultan);

- un modul LCD, cu 2 linii a câte 16 caractere fiecare, cu o interfaţă 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 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**

Microsistemul conţine microprocesorul 8086 împreună cu 128KB de memorie EPROM şi 64KB de memorie SRAM. Pe partea de periferice, sistemul asigură comunicarea cu exteriorul prin porturi seriale și paralele. De asemenea, microsistemul este echipat cu un afișaj cu 8 ranguri de segmente, 10 LED-uri, o minitastatură cu 9 contacte, precum și un modul LCD format din două linii a câte 16 caractere fiecare.

**1.Unitatea centrala cu procesor 8086**

Unitatea centrală a unui microsistem bazat pe 8086 este alcătuită din următoarele componente principale:

* **Microprocesorul Intel 8086**
* **Generatorul de tact 8284A**
* **Circuite pentru separarea și amplificarea magistralelor de adrese și date**, precum:
  + **74LS245**
  + **74LS373**

Microprocesorul **Intel 8086**, produs de firma Intel, este unul dintre cele mai răspândite procesoare pe **16 biți**, fiind baza întregii arhitecturi x86. Acesta poate adresa direct până la **1 MB de memorie**, folosind o magistrală multiplexată de adrese și date.

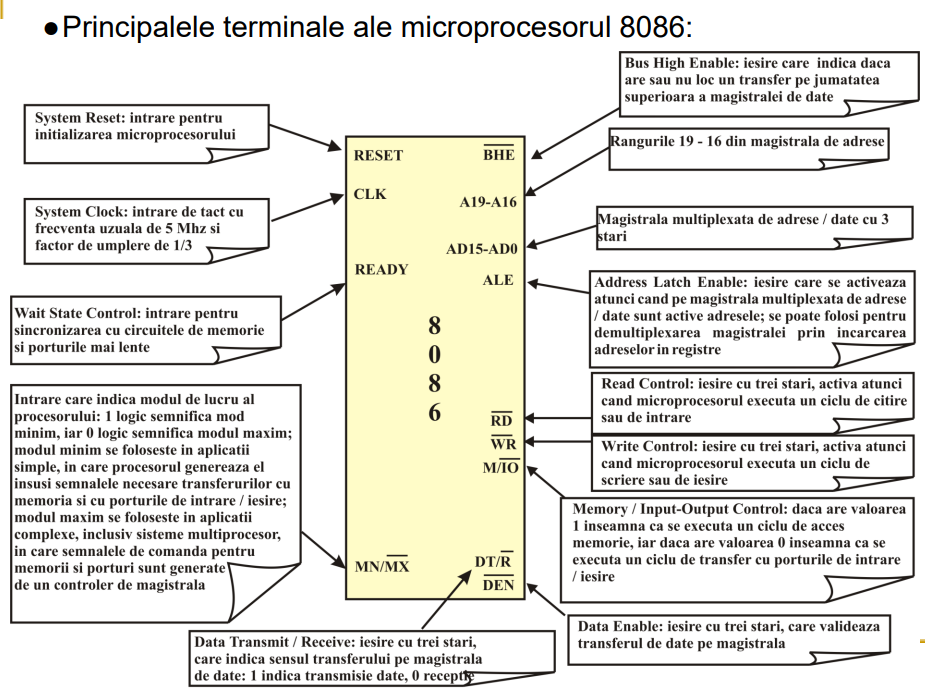
Structura internă a procesorului este împărțită în două unități funcționale care lucrează asincron și independent:

* **EU – Unitatea de Execuție**, care realizează operațiile corespunzătoare instrucțiunilor (aritmetice, logice, operații pe biți etc.);
* **BIU – Unitatea de Interfață cu Magistrale**, care se ocupă cu preluarea instrucțiunilor din memorie și transferul datelor între EU și memorie sau porturile de I/O.

Microprocesorul poate funcționa în două moduri hardware:

* **Mod minim (MN)** – utilizat în aplicații simple, unde 8086 generează singur semnalele pentru memorie și porturi.
* **Mod maxim (MX)** – folosit în sisteme complexe, unde semnalele sunt generate de controlerul extern 8288.

8086 poate efectua operații cu numere **cu sau fără semn**, pe **8 sau 16 biți**, și permite lucru la nivel de bit, octet sau cuvânt, făcându-l versatil și puternic pentru arhitecturile de la acea vreme.



**Generatorul de tact**

**Generatorul de tact 8284A** este circuitul responsabil de furnizarea semnalului de ceas necesar funcționării microprocesorului 8086. El asigură un semnal stabil și precis, absolut necesar sincronizării tuturor operațiilor din sistem. Pe lângă generarea frecvenței de tact, 8284A mai include circuite pentru resetarea procesorului și pentru sincronizarea semnalelor externe, contribuind astfel la funcționarea corectă și fiabilă a microsistemului.

**Registru 74LS373**

* Registru cu **8 ranguri**, folosit la **demultiplexarea** magistralei de adrese a procesorului 8086.
* Conține **8 bistabile tri-state** care mențin stabilă adresa când magistrala este eliberată.

Când G este activ (1 logic), circuitul este în mod transparent, iar datele aplicate pe intrările D0-D7 sunt transmise direct la ieșirile Q0-Q7. Când G devine inactiv (0 logic), latch-urile rețin ultima valoare a datelor, iar ieșirile rămân constante indiferent de modificările intrărilor D0-D7.

|  |  |  |
| --- | --- | --- |
| A diagram of a circuit board  Description automatically generated | A diagram of a circuit  Description automatically generated | A white grid with black text  Description automatically generated |
| Configurația terminalelor | Schema internă | Funcționarea |

**Amplificator/separator 74LS245**

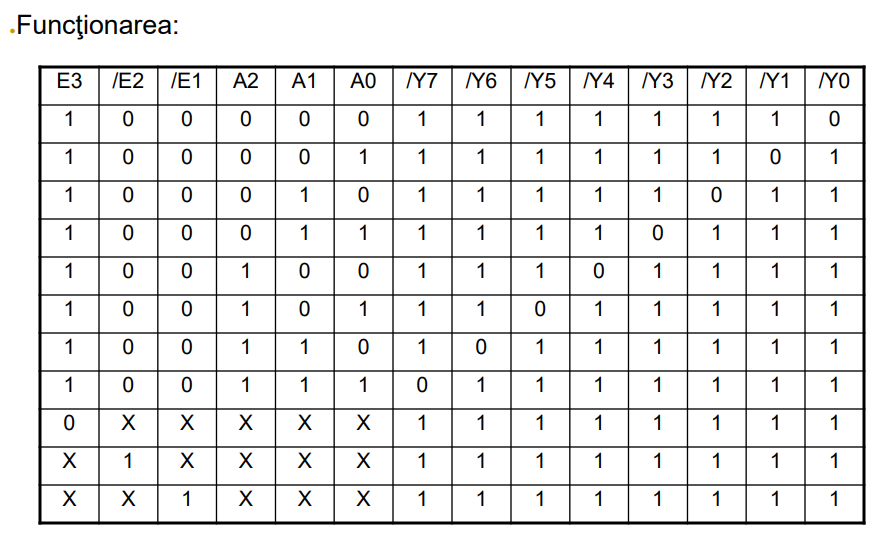
* Asigură **amplificarea și separarea** magistralei de date bidirecționale.
* Permite transferul datelor în **ambele sensuri** și previne conflictele pe magistrală.
* Permite schimbul de date între două magistrale, controlat de semnalul DIR.

|  |  |  |
| --- | --- | --- |
| A diagram of a circuit board  Description automatically generated | A diagram of a line of arrows  Description automatically generated | A white grid with black text  Description automatically generated |
| Configurația terminalelor | Schema internă | Funcționarea |

A diagram of a computer

AI-generated content may be incorrect.

**Funcționarea circuitul decodificator 74LS138**

****

A diagram of a circuit

AI-generated content may be incorrect.**Decodificarea porturilor**

Din decodificatorul de porturi (74LS138) ies semnale de selecție care activează un anumit circuit periferic.

**2.Conectarea memoriilor**

Pentru implementarea memoriei nevolatile și volatile, microsistemul cu microprocesorul 8086 utilizează circuite **EPROM** și **SRAM**, conectate la magistrala procesorului prin intermediul latch-ului **74LS373** (pentru demultiplexarea adreselor) și al circuitelor de decodare a adreselor

**2.1. Memoria EPROM – 27C256**

Memoria EPROM (Erasable Programmable Read Only Memory) reprezintă o memorie nevolatilă, capabilă să păstreze datele chiar și atunci când alimentarea este întreruptă. Programarea acesteia se face prin aplicarea unor tensiuni ridicate, iar ștergerea se realizează prin expunerea la lumină ultravioletă puternică, datorită ferestrei de cuarț aflate în capsula circuitului.

În cadrul proiectului sunt utilizate două circuite **27C256**, fiecare având o capacitate de 64 KB, realizând împreună un spațiu total de **128 KB EPROM**.

Fiecare EPROM dispune de:

* 16 linii de adresă **A0–A15**, conectate la ieșirile latch-ului 74LS373;
* 8 linii bidirecționale de date **D0–D7**, comune magistralei procesorului;
* semnalul de selectare **!CE**, generat prin decodarea adreselor;
* semnalul de validare a ieșirii **!OE**, conectat la linia **RD** a microprocesorului;
* alimentările standard **VCC** și **GND**.

Selecția fiecărui EPROM este realizată prin circuite logice (74LS138), astfel încât fiecare ocupă un bloc de memorie bine definit.

**2.2 Memoria SRAM – 62256**

Memoria SRAM (Static Random Access Memory) este o memorie volatilă, dar stabilă, care nu necesită cicluri de reîmprospătare, spre deosebire de DRAM. Circuitele SRAM păstrează datele atâta timp cât sunt alimentate, utilizând celule logice statice pentru fiecare bit.

În proiect sunt folosite două circuite **62256**, fiecare având o capacitate de 32 KB, obținându-se un total de **64 KB SRAM**.

Fiecare SRAM dispune de:

* 15 linii de adresă **A0–A14**, conectate la aceleași ieșiri demultiplexate ale 74LS373;
* 8 linii bidirecționale de date **I/O0–I/O7**, comune magistralei sistemului;
* semnalul de selectare **!CS**, generat prin decodarea adreselor;
* semnalul de citire **!OE** (legat la RD);
* semnalul de scriere **!WE** (legat la !WR);
* alimentările **VCC** și **GND**.

Fiecare circuit este activat doar când zona de adresă aferentă este accesată de procesor, restul timpului rămânând în stare inactive pentru a evita conflictele de magistrală.

**Decodificarea memoriilor**

128KB memorie EPROM:

2^7\*2^10=2^17B => Numarul de locatii in hexa 20000H

Adresa început este: **00000H**

Adresa finală este: **1FFFFH**

64KB memorie SRAM :

2^6\*2^10=2^16B=> Numarul de locatii in hexa 10000H

Adresa început este: **20000H**

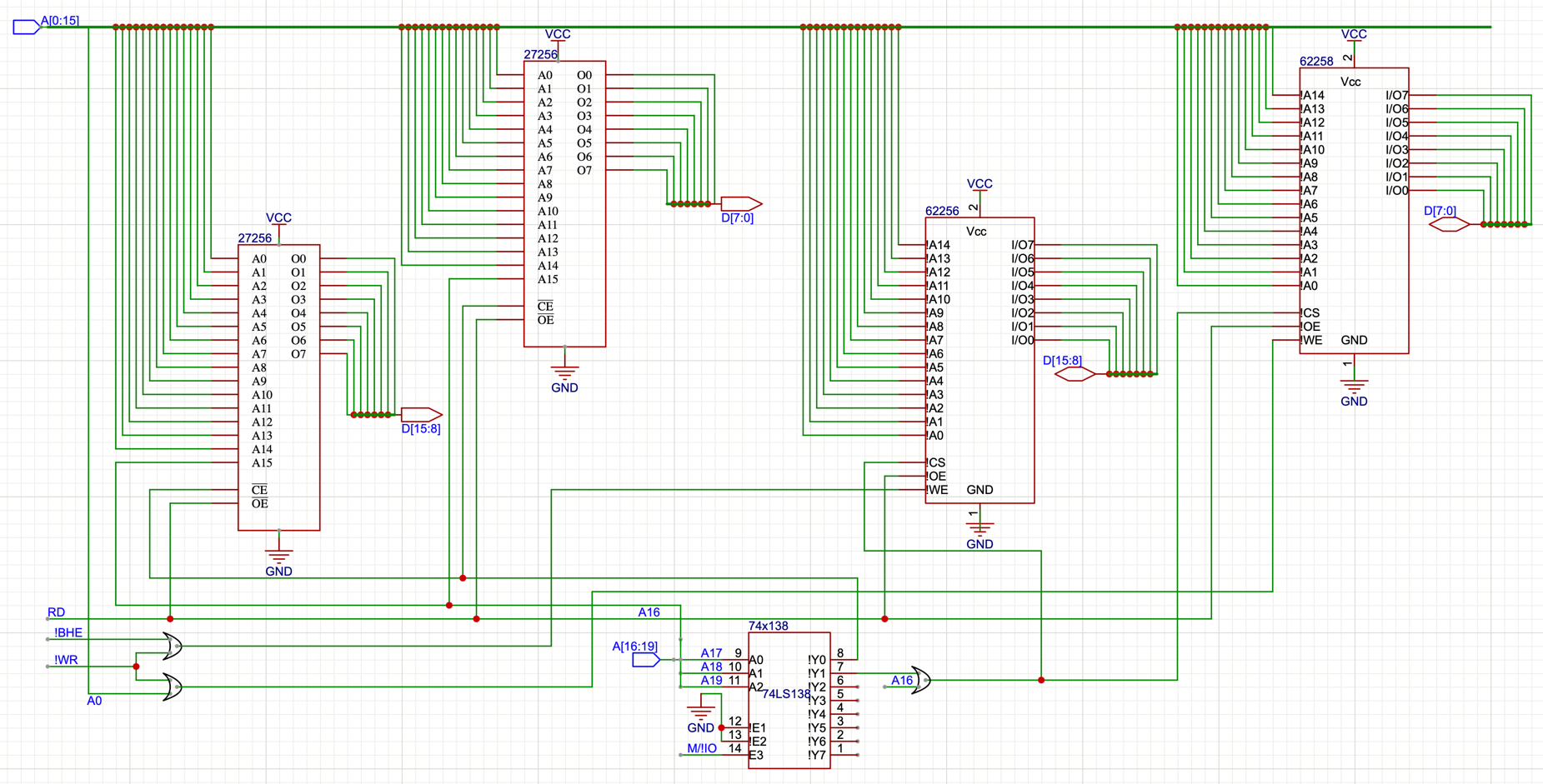
Adresa finală este: 2FFFF**H**

**Adrese porturi memorii**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | A19 | A18 | A17 | A16 | A15 | A14 | A13 | A12 | A11 | A10 | A9 | A8 | A7 | A6 | A5 | A4 | A3 | A2 | A1 | A0 |  |
| 00000H | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | EPROM1 |
| 0FFFFH | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 10000H | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | EPROM2 |
| 1FFFFH | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 20000H | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | SRAM1 |
| 27FFFH | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 28000H | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | SRAM2 |
| 2FFFFH | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

**SELEPROM =( !A19) && (!A18) &&( !A17) = (!Y0)**

**SELSRAM=( (!A19) && (!A18) && (A17) ) ||( A16) = (!Y1) || (A16)**



**3.Interfata seriala si paralela**

3.1 Interfața serială 8251

Circuitul **Intel 8251 USART** (Universal Synchronous/Asynchronous Receiver/Transmitter) este folosit pentru **realizarea** comunicației seriale în sistem. Acesta permite transmisia și recepția de date în mod sincron sau asincron, fiind configurat prin registre interne de comandă.

În proiect, 8251 este conectat la magistrala de date și control a procesorului 8086 și este mapat în spațiul de I/O în două zone posibile:

* **04D0H** (port de date) **– 04D2H** (port de comenzi)
* sau **05D0H (**port de date) **– 05D2H** (port de comenzi),

în funcție de poziția microcomutatorului **S1**. Astfel, selecția circuitului se realizează prin logica de decodare a adreselor, activată doar în intervalul corespunzător, iar accesul la registrele 8251 se face prin semnalele RD, WR ,!CS și C/!D.

A white rectangular box with black text

AI-generated content may be incorrect.A diagram of a rectangular object with arrows

AI-generated content may be incorrect.

**Pinii de Interfață cu Microprocesorul (Control și Date)**

|  |  |  |
| --- | --- | --- |
| Pin | Tip | Descriere |
| D7-D0 | Bidirecțional | Magistrala de date pentru transferul octeților și a comenzilor/stării. |
| !CS | Intrare | Activează cipul pentru comunicarea cu procesorul. |
| !RD | Intrare | Semnal de citire a datelor sau stării. |
| C/!D | Intrare | Selectează registrul accesat: 0 pentru **Date**; 1 pentru **Comandă/Stare.** |
| !WR | Intrare | Semnal de scriere a datelor sau comenzilor. |
| RESET | Intrare | Inițializează și resetează circuitele interne. |
| CLK | Intrare | Sincronizează funcțiile interne ale cipului. |

**Pinii de Transmisie - TX**

|  |  |  |
| --- | --- | --- |
| Pin | Tip | Descriere |
| TxD | Ieșire | Ieșirea datelor seriale transmise (bit cu bit). |
| !TxC | Intrare | Semnal de ceas care sincronizează transmisia datelor pe TxD. |

**Pinii de Recepție – RX**

|  |  |  |
| --- | --- | --- |
| Pin | Tip | Descriere |
| RxD | Intrare | Intrarea datelor seriale recepționate. |
| !RxC | Intrare | Semnal de ceas care sincronizează recepția datelor pe RxD. |

**Pinii Multifuncționali**

|  |  |  |
| --- | --- | --- |
| Pin | Tip | Descriere |
| !CTS | Intrare | Semnal de control primit care oferaă permisiunea de a transmite. |

3.2 Interfața paralelă 8255

Circuitul **Intel 8255 PPI** (Programmable Peripheral Interface) oferă trei porturi paralele programabile (Port A, Port B și Port C), utilizate în proiect pentru conectarea elementelor de I/O.

Acesta este mapat în spațiul de I/O al sistemului în una dintre următoarele două zone:

* **0250H – 0256H**
* sau **0A50H – 0A56H**

A diagram of a rectangular object with arrows

AI-generated content may be incorrect.A table with numbers and letters

AI-generated content may be incorrect.în funcție de starea microcomutatorului **S2**. Fiecare adresă corespunde unui registru intern (Port A, Port B, Port C și registrul de control-RCC), iar selectarea circuitului se face utilizând un semnal /CS generat prin decodarea adreselor.

**Pinii de Interfață cu Microprocesorul**

|  |  |  |
| --- | --- | --- |
| Pin | Tip | Descriere |
| D7-D0 | Bidirecțional | Magistrala de date pentru transferul octeților între microP și porturi. |
| !CS | Intrare | Activează cipul pentru comunicare. |
| !RD | Intrare | Semnal de citire a datelor de la porturi. |
| !WR | Intrare | Semnal de scriere a datelor sau a cuvântului de control. |
| A1,A0 | Intrare | Selectează unul din cele patru registre interne: Port A, Port B, Port C sau Registrul de Comandă/Control (RCC). |
| RESET | Intrare | Resetează circuitele interne și inițializează 8255. |

**Pinii Porturilor de I/O**

|  |  |  |
| --- | --- | --- |
| Pin | Tip | Descriere |
| PA7-PA0 | Bidirecțional | Pinii pentru transferul de date pe **Portul A**. Poate fi configurat ca intrare sau ieșire (8 biți). |
| PB7-PB0 | Bidirecțional | Pinii pentru transferul de date pe **Portul B**. Poate fi configurat ca intrare sau ieșire (8 biți). |
| PC7-PC0 | Bidirecțional | Pinii pentru transferul de date pe **Portul C**. Poate fi utilizat fie pentru I/O de date, fie pentru semnale de control (Handshaking). |

**Adrese porturi pentru interfața serială și paralelă**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | A15 | A14 | A13 | A12 | A11 | A10 | A9 | A8 | A7 | A6 | A5 | A4 | A3 | A2 | A1 | A0 |
| 04D0H | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | S1 |
| 04D2H | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| 05D0H | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | S2 |
| 05D2H | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| 0250H | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | P1 |
| 0252H | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| 0254H | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| 0256H | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 |
| 0A50H | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | P2 |
| 0A52H | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| 0A54H | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| 0A56H | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 |

!E1=!( ( !(A15 || A14 || A13 || A12 ) ) && ( !(A3 || A5) ) )

!E2 = M/~IO

E3= (!A0 && A4)

!IS1 = (!A11) && (!A9) && (!A8) = (!Y0)

!IS2 = (!A11) && (!A9) && (A8) = (!Y1)

!IP1 = (!A11) && (A9) && (!A8) = (!Y2)

!IP2 = (A11) && (A9) && (!A8) = (!Y6)

Circuitul MAX232

Circuitul MAX232 este un dispozitiv integrat utilizat pentru a realiza conversia nivelurilor de tensiune între standardele RS-232 și TTL (Transistor-Transistor Logic), fiind esențial pentru comunicarea între dispozitive care operează la tensiuni diferite, cum ar fi computere, microcontrolere și periferice seriale.

Standardul RS-232 folosește niveluri de tensiune mai mari, de obicei între +3V și +15V pentru un „1 logic” și între -3V și -15V pentru un „0 logic”, în timp ce logica TTL operează la niveluri mai mici, de obicei între 0V și 5V. MAX232 convertește bidirecțional aceste semnale, facilitând comunicarea între dispozitive.

A diagram of a circuit

AI-generated content may be incorrect.

Circuitul Intel 8253 (Programmable Interval Timer)

8253 este un circuit de interfață I/O programabil care conține **trei numărătoare/temporizatoare independente pe biți**. Rolul său principal este de a furniza funcții de sincronizare, numărare de evenimente și generare de frecvențe.

In acest proiect este utilizat exclusiv pentru a gestiona **timpii de comunicare serială**.

1. Ceasul de Intrare (CLK): Numărătorul selectat (de obicei Counter 1) primește semnalul  (Peripherial Clock) generat de 8285A.
2. Programarea: Microprocesorul scrie un factor de divizare în numărător.
3. Semnalul de Ieșire (OUT): Pinul  furnizează un semnal pătrat la o frecvență mult mai mica.
4. **Conectarea la 8251:** Acest semnal de ieșire de la  8253 este conectat direct la pinii !RxC si !TxC al circuitului 8251(Interfața Serială), stabilind astfel rata de transmisie (Baud Rate).

A diagram of a computer

AI-generated content may be incorrect.

4. Conectarea afişajelor şi a minitastaturii

4.1 Minitastura cu 9 contacte

Minitastatura are o structură matriceală ,iar la intersecţia liniilor şi coloanelor se găsesc tastele. Se folosește un port de ieşire cu posibilitate de memorare (registru) şi un port de intrare (porţi cu 3 stări). În portul de ieșire se va scrie 0 logic numai pe o coloană, iar pe restul 1 și se citesc linii. Dacă pe o linie se detectează 0 logic, atunci tasta a fost acționată. Tastatura are o structură matricială 3x3, la intersecţia liniilor şi coloanelor găsindu–se tastele.

A diagram of a circuit

AI-generated content may be incorrect.

Pentru conectarea unei minitastaturi mecanice la o unitate centrala cu microprocesorul 8086, este necesar un port de iesire cu posibilitate de memorare (registru) si un port de intrare (porti cu 3 stari).

Buffer-ul de Intrare (74LS244)

Circuitul 74LS244 este un buffer / driver cu 8 canale, folosit pentru amplificarea și izolarea semnalelor digitale pe magistralele de date și adrese.

Este împărțit în două grupuri de câte 4 canale, fiecare având propria linie Enable, ceea ce permite activarea selectivă a grupei de canale. Rolul lui principal este să ofere curent mare, să reducă încărcarea magistralei și să permită transfer rapid de date între circuite, fiind util în aplicații cu magistrale partajate sau multiplexate.

Latch-ul de Ieșire (74LS373)

Circuitul 74LS373 (latch transparent) are rolul de a memora starea electronică a coloanelor tastaturii sau de a reține datele scrise de microprocesor, pentru a le menține stabile la ieșire. Acesta acționează ca un registru de ieșire (driver) care transmite semnalele către structura matricială a tastaturii.

**Adrese porturi pentru tastatură**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | A15 | A14 | A13 | A12 | A11 | A10 | A9 | A8 | A7 | A6 | A5 | A4 | A3 | A2 | A1 | A0 |
| 0840H | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 08C0H | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |

Port de intrare : 08C0H (!ST2)

Port de iesire : 0840H (ST1)

ST1 = !( (A11) && (!A10) && (!A7) ) = !( !Y4 )

!ST2 = (A11) && (!A10) && (A7) = (!Y5)

A diagram of a circuit

AI-generated content may be incorrect.

4.2 Conectare 10 LED-uri

LED-ul (Light Emitting Diode) este un dispozitiv semiconductor care emite lumină atunci când joncțiunea sa p-n este polarizată în sens direct, permițând trecerea curentului electric.

Se folosește un registru 74LS373 conectat la magistrală pentru a le menține aprinse sau stinse.

Rezistorul conectat la VCC are rolul de a limita curentul care trece prin LED-uri pentru a le proteja împotriva arderii.

LED-urile sunt conectate în configurație Anod Comun, ceea ce înseamnă că anodul este legat la VCC prin resistor(330 Ohm), iar pentru a le aprinde (ON) este necesară o comandă de 0 logic (LOW) pe catod, de la portul de ieșire.

**Adrese porturi pentru LED-uri**

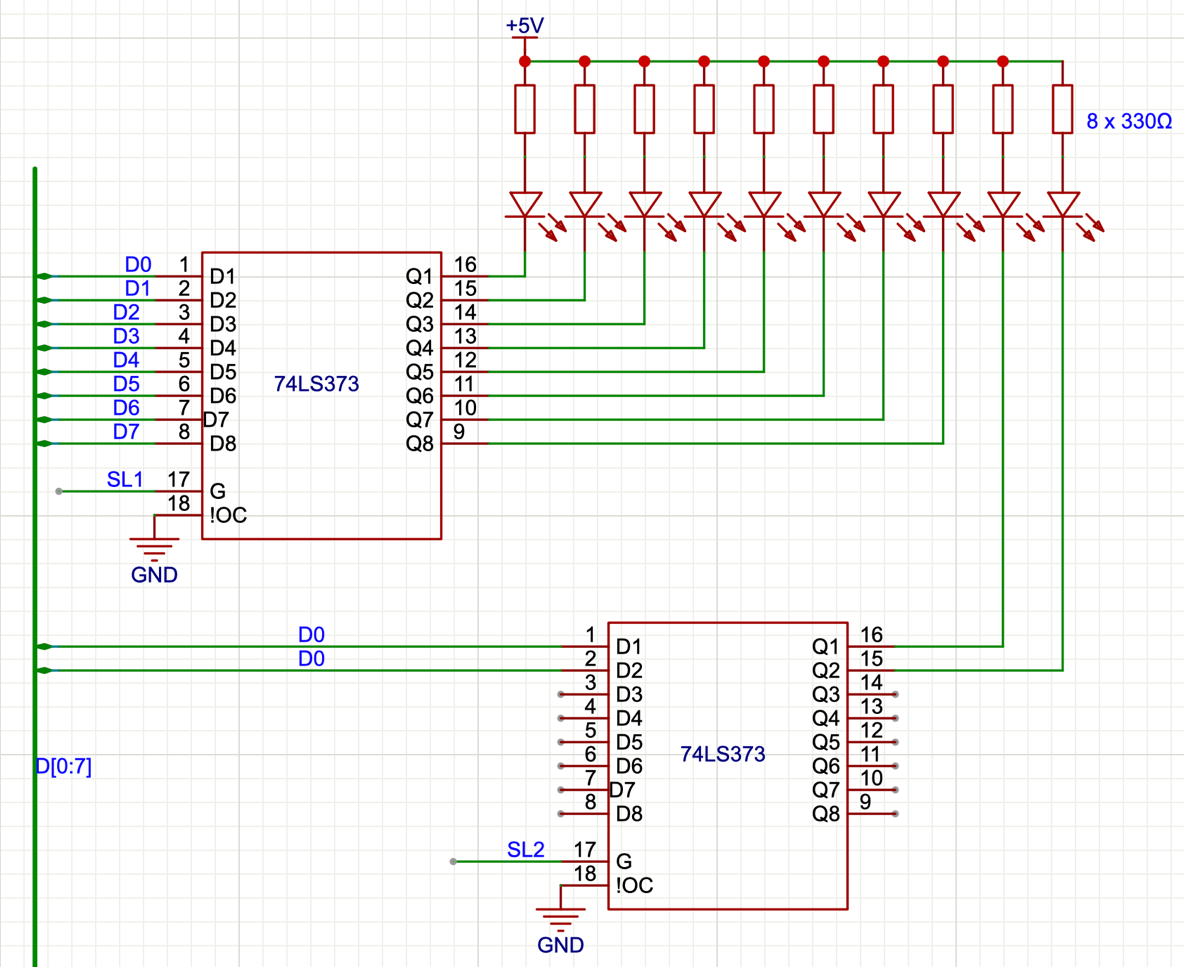
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | A15 | A14 | A13 | A12 | A11 | A10 | A9 | A8 | A7 | A6 | A5 | A4 | A3 | A2 | A1 | A0 |
| SL1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| SL2 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |

Port de intrare – 0C40h (primele 8 LED-uri)

Port de intrare – 0CC0h (ultimele 2 LED-uri)

SL1 = !( (A11) && (A10) && (!A7) ) = !( !Y6 )

SL2 = !( (A11) && (A10) && (A7) ) = !( !Y7 )



4.3 Modulul de afișaj cu 7 segmente

Acest modul este responsabil de conversia codurilor Hexa/Decimale trimise de microprocesor în caractere vizibile (cifre și litere A-F).

Afișajul utilizează 7 LED-uri dispuse în formă de opt (segmentele a până la g), plus un punct zecimal (dp).

Afișajul nu este conectat direct la magistrala de date, ci printr-un latch transparent 74LS373, care joacă rolul esențial de registru de ieșire:

Menținerea Stării: Deoarece afișajul este un dispozitiv lent, 74LS373 memorează octetul de date (codul segmentului) trimis de microprocesor, menținând starea afișată stabilă pe ecran.

Decodificarea Adresei: Latch-ul este activat (pinul G) de semnalul SAx (de exemplu, SA1) generat de decodificatorul de adrese. Când microprocesorul scrie la adresa alocată (de exemplu, 0040H), semnalul SA1 este activat, iar latch-ul captează datele de pe magistrala D[0..7] și le plasează la ieșirile sale (Q1 la Q8).

**Adrese porturi pentru 8 ranguri 7 Segment Display**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | A15 | A14 | A13 | A12 | A11 | A10 | A9 | A8 | A7 | A6 | A5 | A4 | A3 | A2 | A1 | A0 |
| 0040H | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 00C0H | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0440H | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 04C0H | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0000H | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0080H | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0400H | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0480H | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

SA1 = !( (!A11) && (!A10) && (!A7) )= !(!Y0)

SA2 = !( (!A11) && (!A10) && (A7) ) = !(!Y1)

SA3 = !( (!A11) && (A10) && (!A7) ) = !(!Y2)

SA4 = !( (!A11) && (A10) && (A7) ) = !(!Y3)

SA5 = !( ( (!A11) && (!A10) && (!A7) ) || (!A6) ) = !( (!Y0) || (!A6) )

SA6 = !( ( (!A11) && (!A10) && (A7) ) || (!A6) ) = !( (!Y1) || (!A6) )

SA7 = !( ( (!A11) && (A10) && (!A7) ) || (!A6) ) = !( (!Y2) || (!A6) )

SA8 = !( ( (!A11) && (A10) && (A7) ) || (!A6) ) = !( (!Y3) || (!A6) )

**A diagram of a circuit board

AI-generated content may be incorrect.**

| **Valoare Hex** | **Segmente Aprinse (0 logic)** | **Cod (Hex)** |
| --- | --- | --- |
| 0 | a,b,c,d,e,f | 03H |
| 1 | b,c | 9FH |
| 2 | a,b,d,e,g | 25H |
| 3 | a,b,c,d,g | 0DH |
| 4 | b,c,f,g | 99H |
| 5 | a,c,d,f,g | 49H |
| 6 | a,c,d,e,f,g | 41H | |
| 7 | a,b,c | 4FH | |
| 8 | a,b,c,d,e,f,g | 01H | |
| 9 | a,b,c,d,f,g | 09H | |
| A | a,b,c,e,f,g | 11H | |
| B | c,d,e,f,g | C1H | |
| C | a,d,e,f | 63H | |
| D | b,c,d,e,g | 85H | |
| E | a,d,e,f,g | 61H | |
| F | a,e,f,g | 71H | |

4.4 Modulul LCD 16 X 2

Acesta funcționează ca un ecran compact, capabil să afișeze două linii de text, fiecare cu o capacitate de 16 caractere.

Tipul de Afișaj: Este un modul alfanumeric, capabil să afișeze caractere din setul ASCII standard (litere, cifre, simboluri) și caractere personalizate programabile de utilizator.

Structură: Modulul este organizat pe două linii vizibile, cu 16 poziții de caractere pe fiecare linie.

Controler Intern: Modulul include un controler intern (cel mai adesea compatibil cu Hitachi HD44780) care gestionează memoria de afișare (DDRAM) și conversia codurilor ASCII în pixeli.

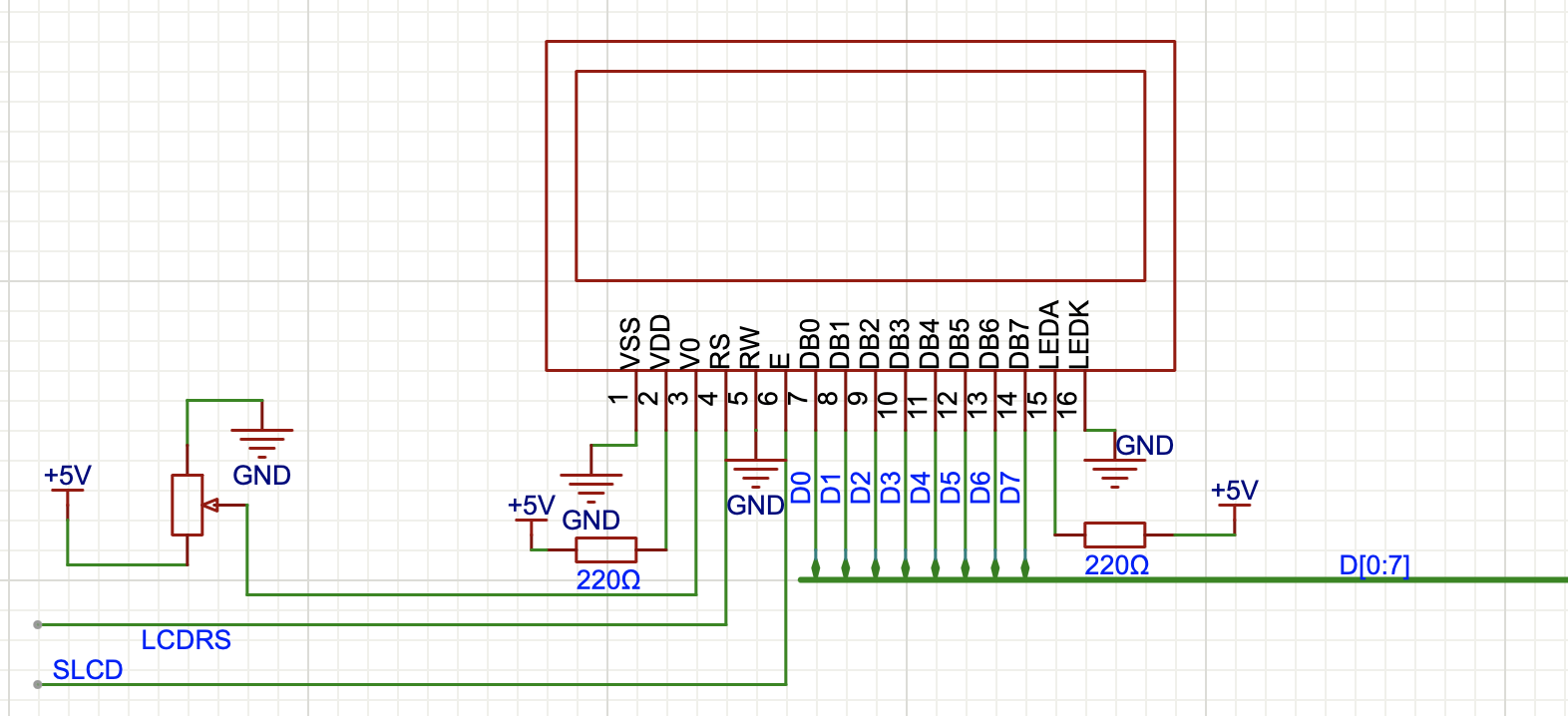
A close-up of a pin description

AI-generated content may be incorrect.

Adresa portului pentru modulul LCD

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | A15 | A14 | A13 | A12 | A11 | A10 | A9 | A8 | A7 | A6 | A5 | A4 | A3 | A2 | A1 | A0 |
| 08D0H | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |

SLCD = (A11) && (!A9) && (!A8) = (!Y4)



**Subrutina de Programare a Circuitelor 8251 si 8255**

Subrutina de programarea a interfeței seriale

|  |  |
| --- | --- |
| SERIAL\_PROG:  MOV DX, 04D2H  MOV AL, 0CEH  OUT DX, AL  MOV AL, 015H  OUT DX, AL    RET | ;Adresa portului de comenzi/stare  ;AL<-cuvant de mod 8 biți date, fără paritate,factor x16  ;Scrie cuvântul de mod în 8251  ;AL<-Cuvântul de Comandă (15H)  ;Scrie cuvântul de comandă în 8251 |

Subrutina de programarea a interfeței paralele

|  |  |
| --- | --- |
| PARALEL\_PROG:  MOV DX, 0256H  MOV AL, 81H  OUT DX, AL  RET | ;Adresa portului de comandă/control RCC  ;AL<-Cuvânt de Comandă (81H): Port A (Ieșire), Port B (Ieșire), Port C Inferior (Intrare)  ;Scrie Cuvântul de Comandă în 8255 |

Subrutina de transmisie a unui caracter pe interfața serială

|  |  |
| --- | --- |
| SERIAL\_TRANSM:  MOV DX, 04D2H  TX\_TEST:  IN AL, DX  RCR AL, 1  JNC TX\_TEST  MOV DX, 04D0H  MOV AL, CL  OUT DX, AL  RET | ;DX<-Adresa portului de stare (pentru a citi TxRDY)  ;Etichetă pentru bucla de așteptare (Polling)  ;Citește Cuvântul de Stare de la 8251  ;Rotește la dreapta. Bitul 0 (TxRDY) ajunge în flagul Carry.  ;Salt înapoi dacă TxRDY = 0 (Nu este gata de transmisie)  ;DX <- Adresa portului de date pentru scriere  ;AL <- Preia caracterul din CL  ;Transmite caracterul prin 8251 |

Subrutina de recepție a unui caracter pe interfața serială

|  |  |
| --- | --- |
| SERIAL\_REC:  MOV DX, 04D2H  RX\_TEST:  IN AL, DX  RCR AL, 2  JNC RX\_TEST  MOV DX, 04D0H  IN AL, DX  MOV CL, AL  RET | ;DX <- Adresa portului de stare (pentru a citi RxRDY)  ; Citește Cuvântul de Stare de la 8251  ; Rotește la dreapta cu 2 poziții. Bitul 1 (RxRDY) ajunge în flagul Carry  ; Salt înapoi dacă RxRDY = 0 (Data nu a sosit încă)  ; DX<-Adresa portului de date pentru citire  ; Citește data de la 8251  ; Depune caracterul recepționat în registrul CL |

Subrutina de transmisie a unui caracter pe interfața paralelă

|  |  |
| --- | --- |
| PARALEL\_TRANSM:  MOV DX, 0250h  MOV AL, CL  OUT DX, AL  RET | ;DX <- Adresa Portului A (Portul de Ieșire pentru Date  ; AL <- Preia caracterul din CL  ; Trimite caracterul pe Port A. Datele sunt acum disponibile la ieșire |

Subrutina de scanare a minitastaturii

|  |  |
| --- | --- |
| TASTATURA\_SCAN:  MOV DX, 0840H  MOV AL, 0FEH  OUT DX, AL  IN AL, 08C0H  AND AL, 80H  JZ TASTA1  IN AL, 08C0H  AND AL, 40H  JZ TASTA4  IN AL, 08C0H  AND AL, 20H  JZ TASTA7  MOV DX, 0840H  MOV AL, 0FDH  OUT DX, AL  IN AL, 08C0H  AND AL, 80H  JZ TASTA2  IN AL, 08C0H  AND AL, 40H  JZ TASTA5  IN AL, 08C0H  AND AL, 20H  JZ TASTA8  MOV DX, 0840H  MOV AL, 0FBH  OUT DX, AL  IN AL, 08C0H  AND AL, 80H  JZ TASTA3  IN AL, 08C0H  AND AL, 40H  JZ TASTA6  IN AL, 08C0H  AND AL, 20H  JZ TASTA9  JMP TASTATURA\_SCAN | ;DX <- Adresa portului de Ieșire/Scanare  ;COLOANA 1  ;AL <- Cod scanare C1 (1111 1110B)  ;Activează Coloana 1 (Ieșire ST2)  ;Citește rândurile (R1, R2, R3) de la ST1  ;Verifică Rândul 1  ;Saltă dacă Rândul 1 este LOW (Tasta 1)  ;Citește rândurile din nou  ;Verifică Rândul 2  ;Saltă dacă Rândul 2 este LOW (Tasta 4)  ;Citește rândurile din nou  ;Verifică Rândul 3  ;Saltă dacă Rândul 3 este LOW (Tasta 7)  ;COLOANA 2  ;DX <- Adresa portului de Ieșire/Scanare  ;AL <- Cod scanare C2 (1111 1101B)  ;Activează Coloana 2 (Ieșire ST2)  ;Citește rândurile (R1, R2, R3) de la ST1;  ;Verifică Rândul 1  ;Saltă dacă Rândul 1 este LOW (Tasta 2)  ;Citește rândurile din nou  ;Verifică Rândul 2  ;Saltă dacă Rândul 2 este LOW (Tasta 5)  ; Citește rândurile din nou  ; Verifică Rândul 3  ;Saltă dacă Rândul 3 este LOW (Tasta 8)  ;COLOANA 3  ;DX <- Adresa portului de Ieșire/Scanare  ;AL <- Cod scanare C3 (1111 1011B)  ;Activează Coloana 3 (Ieșire ST2)  ;Citește rândurile (R1, R2, R3) de la ST1  ;Verifică Rândul 1  ;Saltă dacă Rândul 1 este LOW (Tasta 3)  ;Citește rândurile din nou  ;Verifică Rândul 2  ;Saltă dacă Rândul 2 este LOW (Tasta 6)  ;Citește rândurile din nou  ;Verifică Rândul 3  ;Saltă dacă Rândul 3 este LOW (Tasta 9)  ;Reia ciclul de scanare dacă nu s-a găsit nicio tastă |

Subrutina de aprindere a unui LED

|  |  |
| --- | --- |
| LED\_ON:  MOV DX, 0C40H  IN AL, DX  AND AL, 0FEH  OUT DX, AL  RET | ;DX <- Adresa portului SL1  ;Citește starea curentă a portului  ;Masca 1111 1110B. Setează bitul D0 la 0 (ON)  ;Scrie noua stare înapoi pe SL1 |

Subrutina de stingere a unui LED

|  |  |
| --- | --- |
| LED\_OFF:  MOV DX, 0C40H  IN AL, DX  OR AL, 01H  OUT DX, AL  RET | ;DX <- Adresa portului SL1  ;Citește starea curentă a portului  ;Masca 0000 0001B. Setează bitul D0 la 1 (OFF)  ;Scrie noua stare înapoi pe SL1 |

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

|  |  |
| --- | --- |
| SEGMEN\_CODES: DB 0COH, OF9H,  08FH, 0DH, 99H, 49H, 41H, 4FH, 01H, 09H, 11H, C1H, 63H, 85H, 61H, 71H  AFISEAZA HEXA:  MOV BL, CL  MOV BH, 00H  MOV SI, OFFSET SEGMENT\_CODES  ADD SI, BX  MOV AL, [SI]  MOV DX, 0040H  OUT DX, AL  RET | ; Definiția tabelului de 16 octeți pentru 0-F  ;BL <- Valoarea Hex(in funcţie de valoarea din CL se va afişa valoara hex corespunzataore)  ;Curăță BH (pregătire pentru BX)  ;SI <- Adresa de început a tabelului de coduri  ;Calculează adresa efectivă a codului 7-segmente  ;AL <- Citește codul 7-segmente din table (valoarea)  ;DX <- Adresa de bază a primului rang (SA1)  ;Trimite codul segmentelor la rangul selectat |

**Bibliografie**

* [1] Mircea Popa, Sisteme cu microprocesoare, Editura Orizonturi Universitare, Timișoara, 2003
* [2] Materialele de curs, laborator și proiect ale disciplinei Proiectarea cu Micrprocesoare
* [3]Microprocesorul 8086, lucrare de laborator UTCN <https://elupu.utcluj.ro/wp-content/uploads/2018/10/L2.pdf>

ultima accesare : 5 Noiembrie

* [4]Interfaţa paralelă de intrări/ ieşiri PIO 8255A, lucrare de laborator UTCN

<https://users.utcluj.ro/~apateana/PIO8255.pdf>

ultima accesare : 10 Noiembrie

* [5] Utilizare potentiometru <https://sites.google.com/site/bazeleelectronicii/home/rezistoare/12-potentiometers>

ultima accesare : 12 Decembrie

* [6] Foile de catalog accesate pentru urmatoarele componente:

|  |  |  |
| --- | --- | --- |
| Componenta | Sursa | Utlima accesare |
| Intel 8086 | <https://www.inf.pucrs.br/calazans/undergrad/orgcomp_EC/mat_microproc/intel-8086_datasheet.pdf> | 5 Noiembrie |
| 74LS373 | <https://www.alldatasheet.com/datasheet-pdf/pdf/28029/TI/74LS373.html> | 10 Noiembrie |
| 74LS245 | <https://www.alldatasheet.com/datasheet-pdf/pdf/177341/TI/74LS245.html> | 10 Noiembrie |
| 27256 | <https://www.alldatasheet.com/datasheet-pdf/pdf/157883/INTEL/27256.html> | 10 Noiembrie |
| 62256 | <https://www.alldatasheet.com/html-pdf/37318/SAMSUNG/62256/493/2/62256.html> | 12 Noiembrie |
| 8251 | <https://www.alldatasheet.com/datasheet-pdf/pdf/66096/INTEL/8251A.html> | 15 Noiembrie |
| 8255 | <https://www.alldatasheet.com/html-pdf/66100/INTEL/8255A/511/4/8255A.html> | 16 Noiembrie |
| 74LS244 | <https://www.ti.com/lit/ds/symlink/sn74s244.pdf> | 1 Decembrie |
| 16 x 2 LCD | <https://www.electronicwings.com/sensors-modules/lcd-16x2-display-module> | 14 Decembrie |