UNIVERSITATEA POLITEHNICA TIMIȘOARA

FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE

Proiectarea microsistemelor digitale

**Microsistem cu microprocesorul 8086**

Arșeanu Andreea

An universitar:2024-2025

**Tema Proiectului**

Să se proiecteze un microsistem cu următoarea structură:

* unitate centrală cu microprocesorul 8086;
* 256 KB memorie EPROM, utilizând circuite 27C2048;
* 64 KB memorie SRAM, utilizând circuite 62512;
* interfață serială, cu circuitul 8251, plasată în zona 0AF0H – 0AF2H sau 0BF0H – 0BF2H, în funcție de poziția microcomutatorului S1;
* interfață paralelă, cu circuitul 8255, plasată în zona 0D70H – 0D76H sau 0C70H – 0C76H, în funcție de poziția microcomutatorului S2;
* minitastatură cu 12 contacte;
* 10 led-uri;
* un modul de afișare cu 7 segmente, cu 6 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.

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

**DESCRIERE HARDWARE**

**1.Unitatea centrală a microprocesorului 8086**

Unitatea centrală a microsistemului este alcătuită din:

* Microprocesorul Intel 8086
* Generatorul de tact 8284A
* Circuite pentru amplificarea și separarea magistralelor de adrese și date

Microprocesorul Intel 8086 este unul dintre cele mai răspândite microprocesoare pe 16 biți. Atât registrele interne, cât și magistralele de date interne și externe au o lățime de 16 biți. Lansarea acestuia a fost urmată rapid de dezvoltarea unei familii de componente asociate, inclusiv:

* 8284 – generatorul de tact
* 8288 – controlerul de magistrală
* 8087 – coprocesorul matematic
* 8089 – coprocesorul de intrare/ieșire

Structura internă

Microprocesorul 8086 are două unități funcționale principale care lucrează asincron și independent:

* EU (Execution Unit – Unitatea de Execuție): Execută instrucțiunile.
* BIU (Bus Interface Unit – Unitatea de Interfațare cu Magistralele): Se ocupă cu aducerea instrucțiunilor din memorie și transferul operandurilor între EU și exteriorul microprocesorului.

Moduri de operare

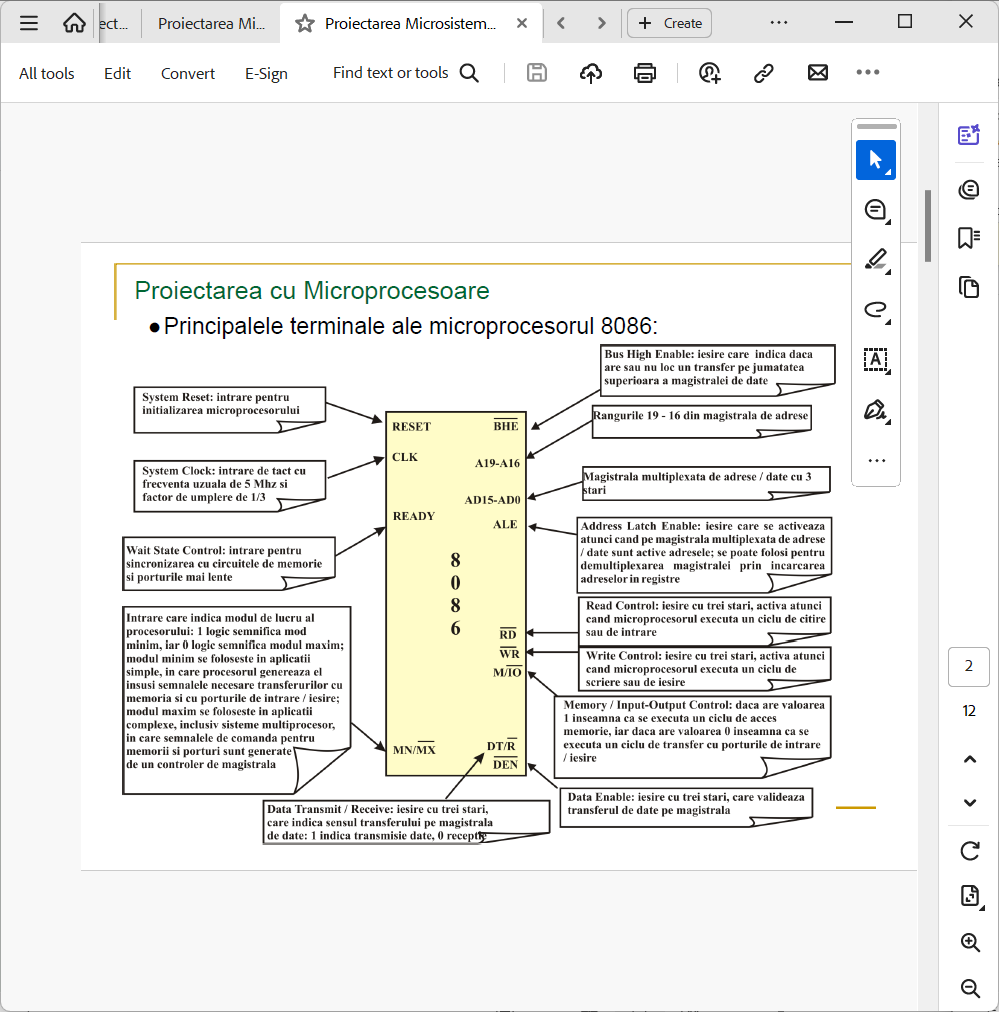
Microprocesorul poate funcționa în două moduri distincte:

1. Modul minim:
   * Utilizat pentru aplicații relativ simple.
   * Microprocesorul generează singur semnalele necesare transferurilor cu memoria și porturile de intrare/ieșire.
2. Modul maxim:
   * Utilizat pentru aplicații complexe, inclusiv în sisteme multiprocesor.
   * Semnalele de comandă pentru memorii și porturi sunt generate de un controler de magistrală (8288).
   * Trecerea între moduri se realizează hardware prin terminalul MN/MX:
     + 1 logic: Modul minim.
     + 0 logic: Modul maxim.

Capacitatea de adresare

Microprocesorul Intel 8086 poate lucra cu un spațiu maxim de adresare de 1M octeți, ceea ce corespunde unui număr de 1M adrese distincte. Acest lucru este posibil deoarece adresele sunt reprezentabile pe 20 de biți.

Principalele terminale ale microprocesorului 8086:

****

**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.

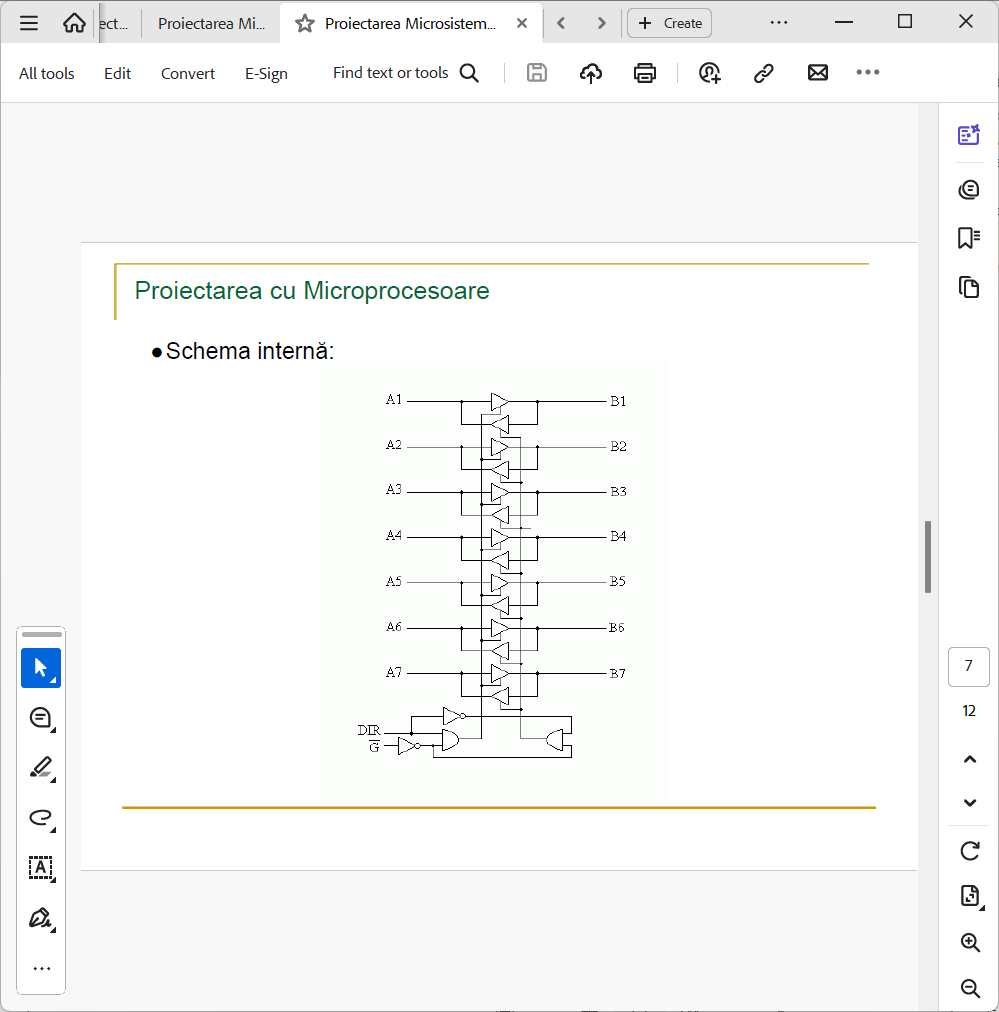
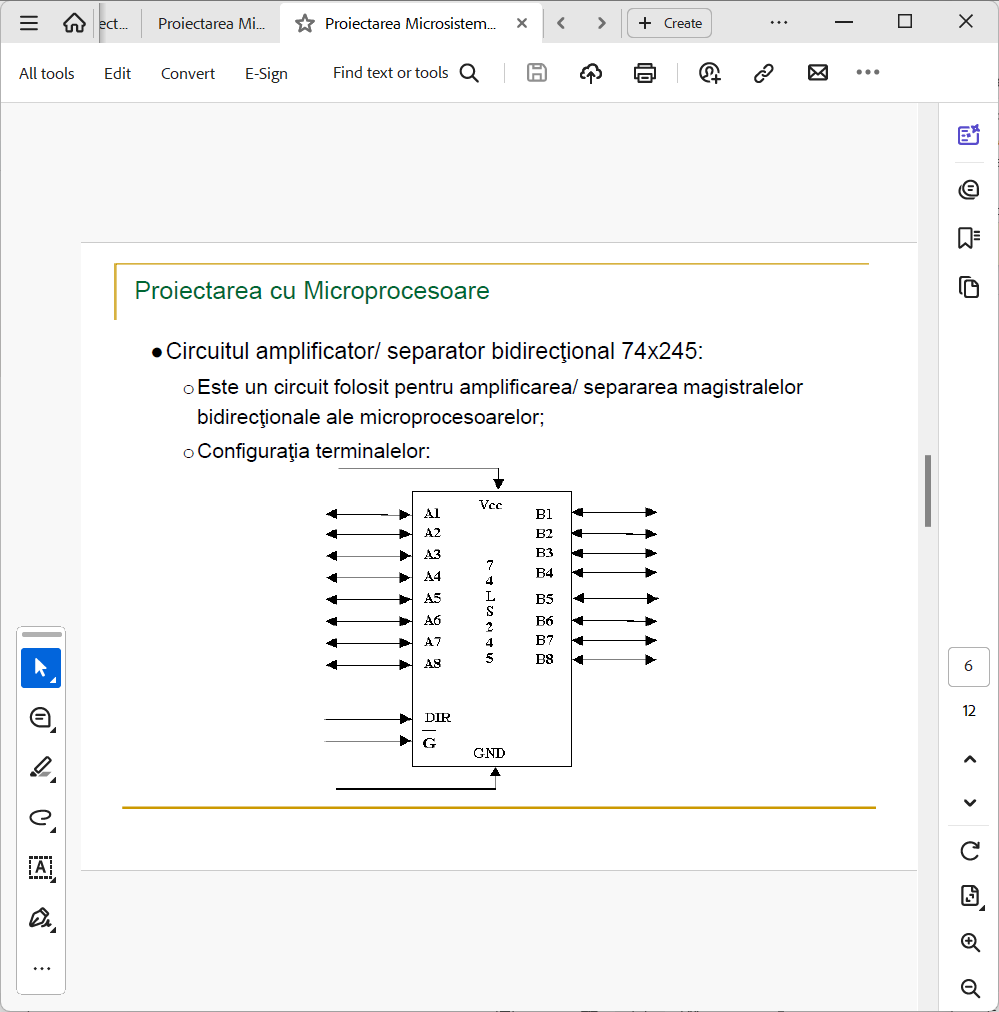
Folosește ca intrare o sursă de semnal, un cristal de cuarț pentru a genera semnalul de clock necesar procesorului. Acesta va fi 1/3 din valoarea frecvenței cristalului. Ieșirea PCLK pentru periferice funcționează ½ din frecvența CLK.

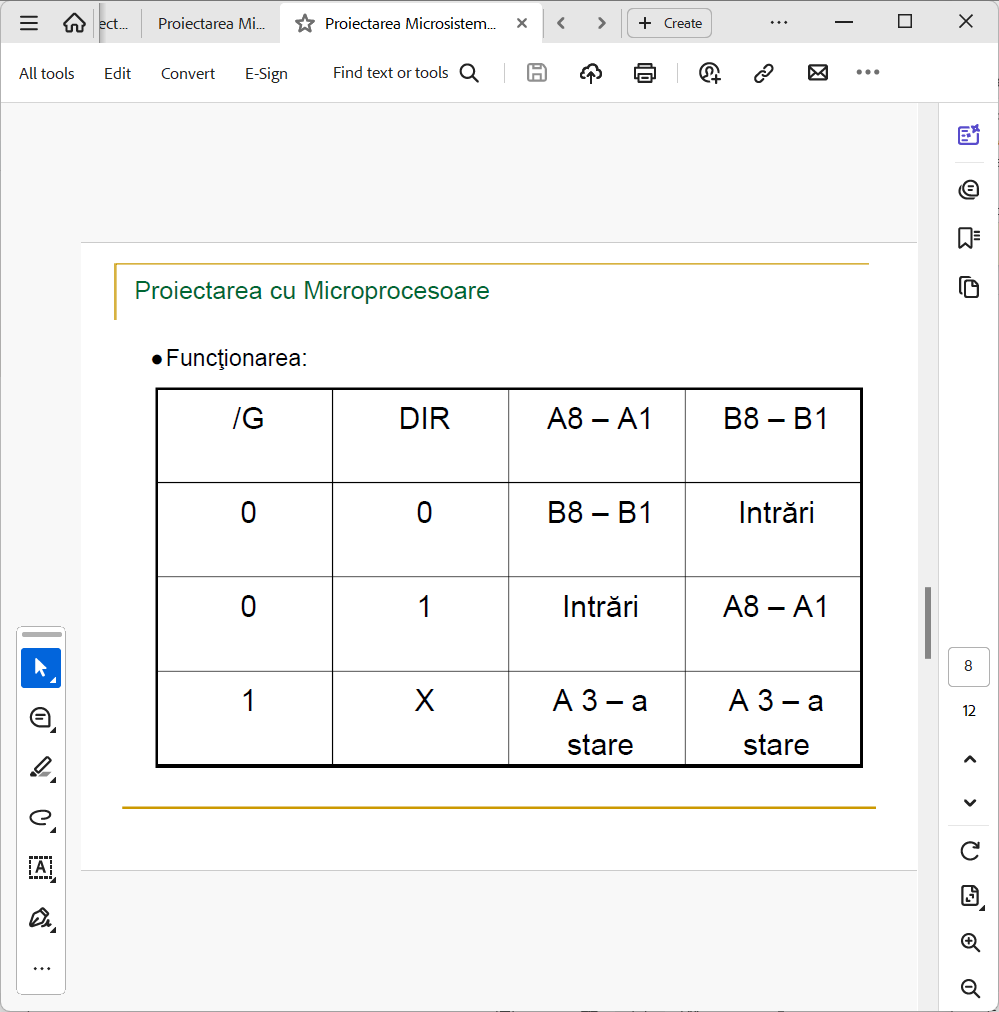
**Circuitul amplificator/ separator bidirecţional 74x245:**

Circuitul **74x245** este un **amplificator/separator bidirecțional cu trei stări (tri-state)**, utilizat frecvent pentru interconectarea magistralelor de date și gestionarea comunicării între dispozitive. Este parte a familiilor TTL și CMOS și este disponibil în diferite variante, cum ar fi 74LS245, 74HC245, 74HCT245, care diferă prin specificații electrice și tehnologii.

**Caracteristici principale**

1. **Funcție principală**:
   * Permite transferul de date bidirecțional între două magistrale de 8 biți.
   * Funcționează ca un buffer (amplificator) pentru a izola sau întări semnalele dintre cele două magistrale.
2. **Structura internă**:
   * Dispune de două grupuri de 8 pini (A0–A7 și B0–B7) care reprezintă cele două magistrale de date.
   * Controlează direcția transferului de date și starea ieșirilor prin pini de control.
3. **Pini de control**:
   * **!G** : Activează ieșirile. Este activ pe nivel logic jos (0).
   * **DIR** (Direction): Controlează direcția transferului de date.
     + DIR = 0: Transferul datelor se face de la B către A.
     + DIR = 1: Transferul datelor se face de la A către B.
4. **Ieșiri tri-state**:
   * Ieșirile pot fi în una dintre cele trei stări:
     + **0**: Nivel logic jos.
     + **1**: Nivel logic înalt.
     + **High-impedance (Z)**: Stare de înaltă impedanță (deconectat).

****

****

**Circuitul 74x373:**

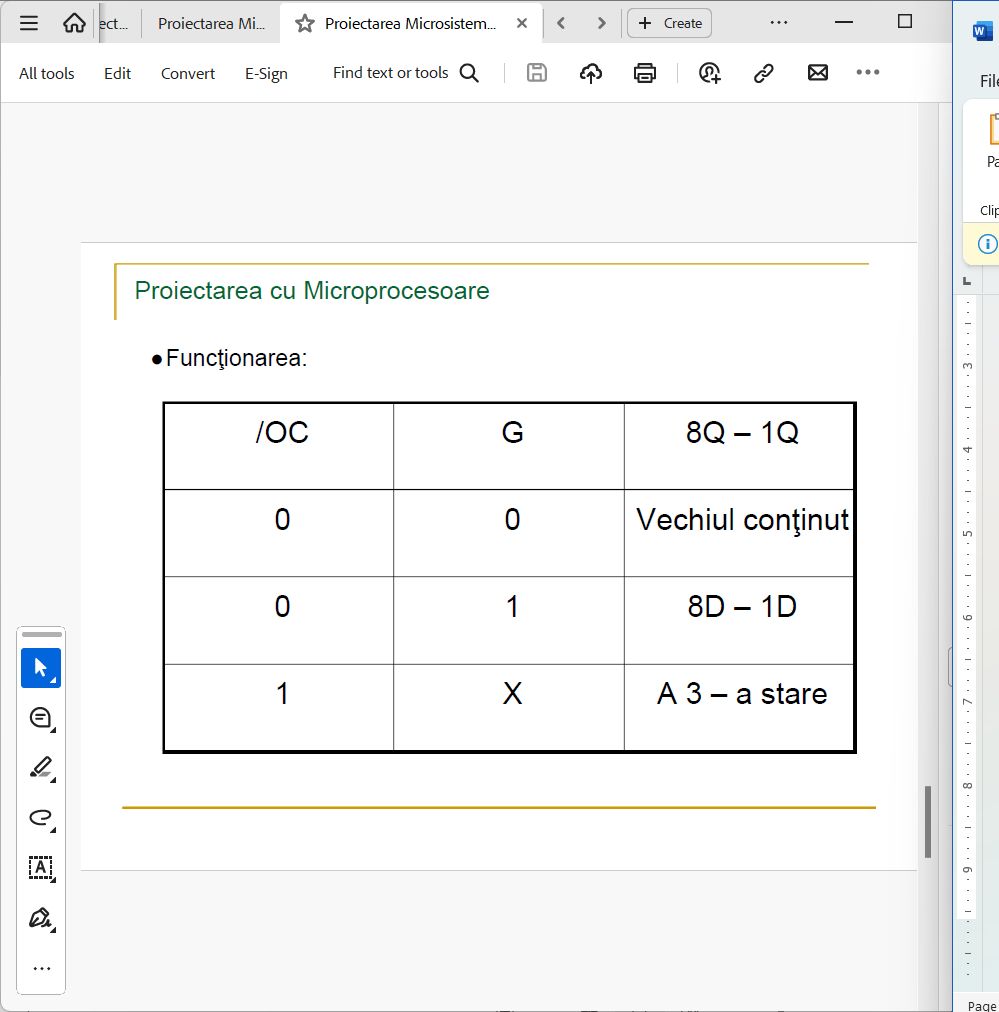
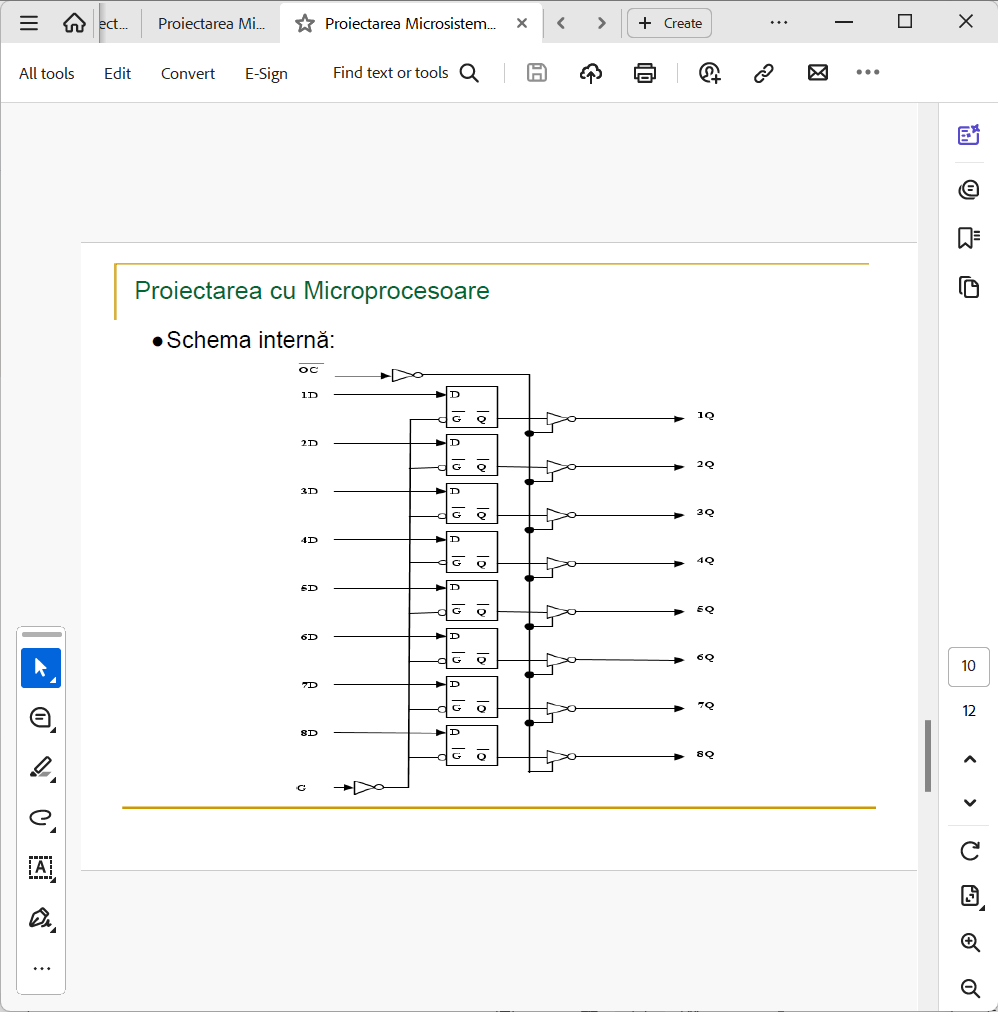
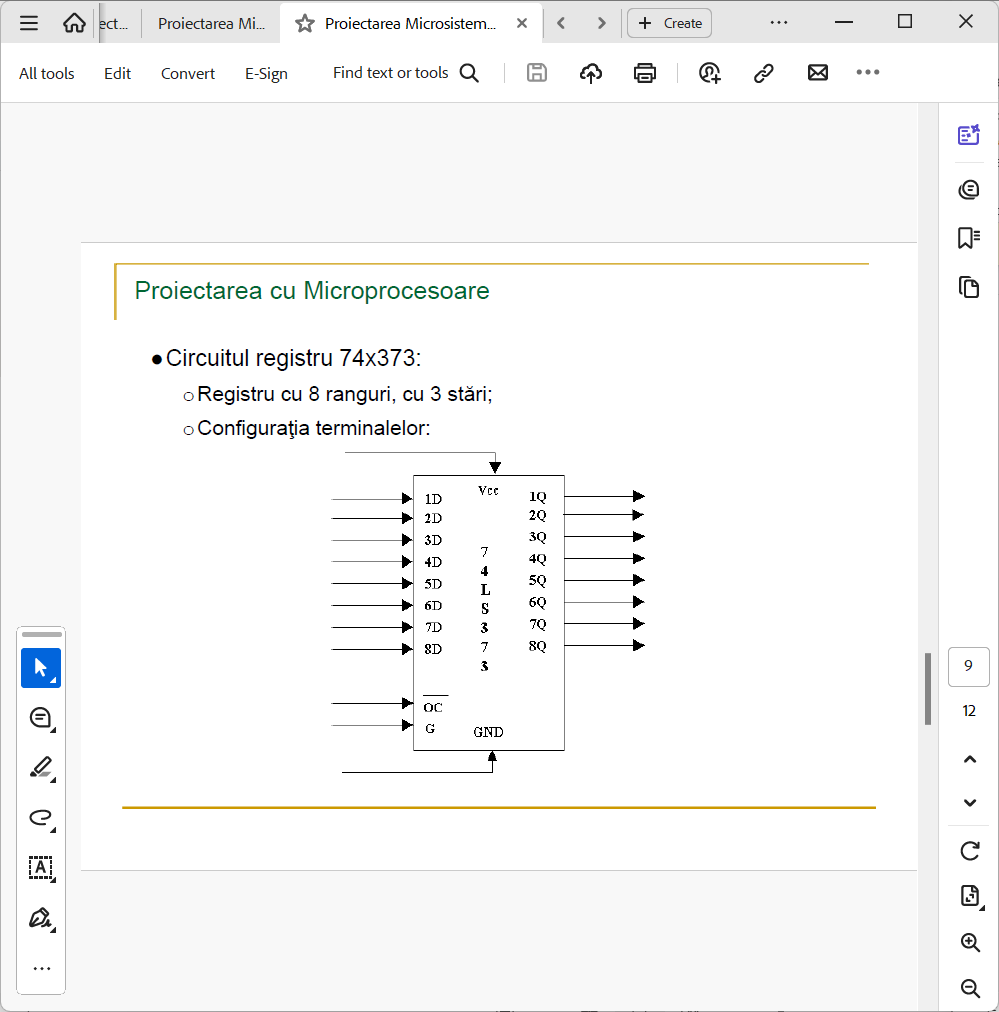
Registru cu 8 ranguri, cu 3 stări;

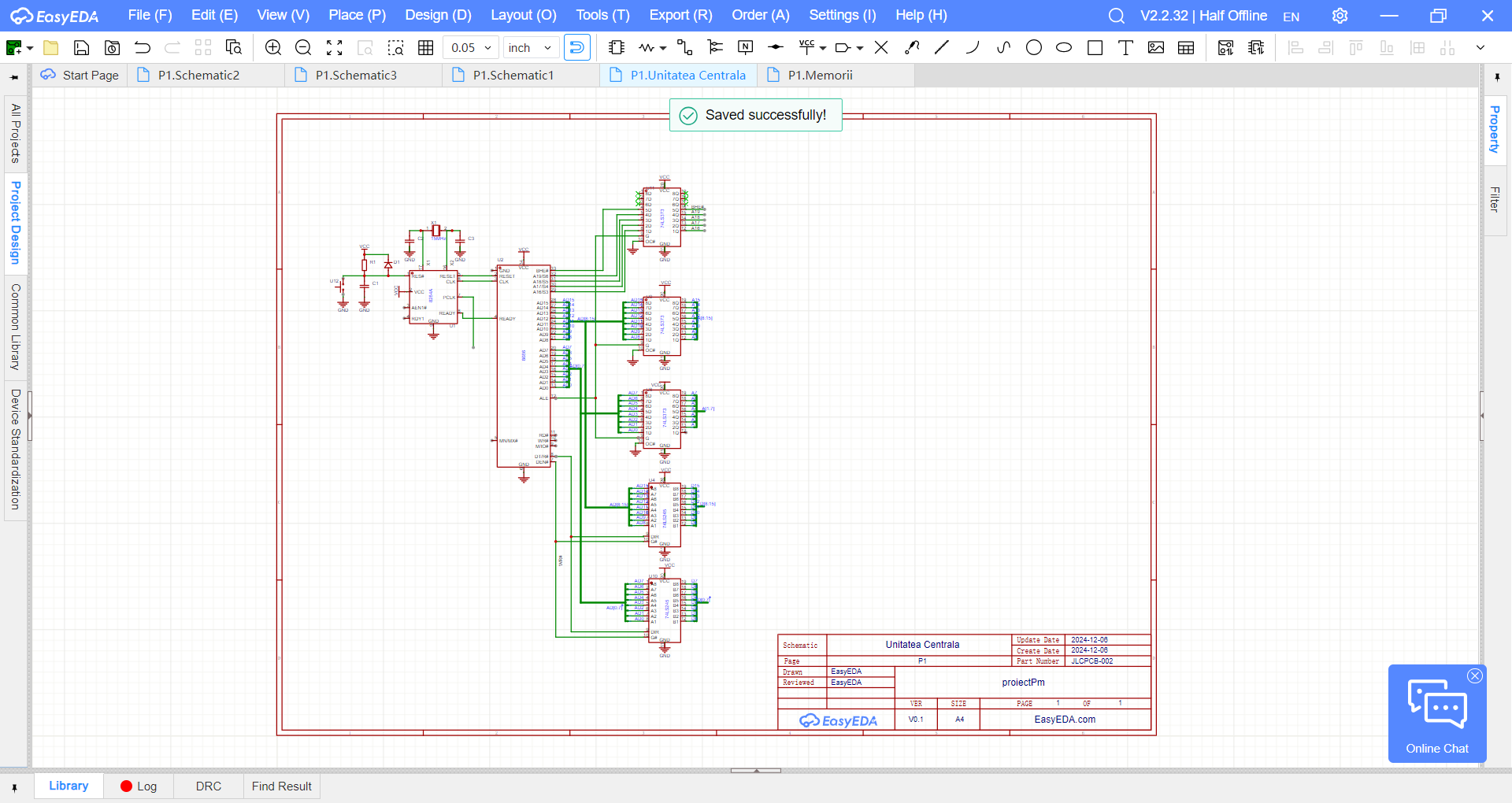
Circuitul **74x373** este un registru octet (8 biți) de tip latch, cu ieșiri **tri-state**, utilizat frecvent în aplicații de memorare temporară a datelor și gestionarea magistralelor de date. Acesta permite stocarea datelor și izolarea lor pe o magistrală comună.

**Caracteristici principale**

1. **Funcție principală**:
   * Funcționează ca un registru **latch transparent D**, ceea ce înseamnă că datele de pe intrări sunt transferate la ieșiri în funcție de semnalul de control.
   * Dispune de ieșiri tri-state, permițând conectarea mai multor dispozitive la aceeași magistrală.
2. **Structură internă**:
   * Are 8 etaje (biți) de stocare.
   * Pinii principali includ:
     + **D0 – D7**: Intrări de date.
     + **Q0 – Q7**: Ieșiri de date (active pe nivel logic jos sau ridicat).
     + **!OC** (Output Control): Controlează starea ieșirilor.
     + **G** (Latch Enable): Controlează stocarea datelor.
3. **Controlul funcționalității**:
   * **Latch Enable (G)**:
     + G = 1: Latch-ul este transparent, iar datele de la intrări sunt transferate direct la ieșiri.
     + G = 0: Latch-ul stochează starea curentă a intrărilor, blocând actualizarea ieșirilor.
   * **Output Control (!OC)**:
     + !OC = 0: Ieșirile sunt active.
     + !OC = 1: Ieșirile sunt în stare de impedanță ridicată (tri-state).
4. **Ieșiri tri-state**:
   * Permite conectarea mai multor dispozitive pe aceeași magistrală, evitând conflictele de date.

Configuraţia terminalelor: Schema internă:

****

****

**2. Conectarea la memorie**

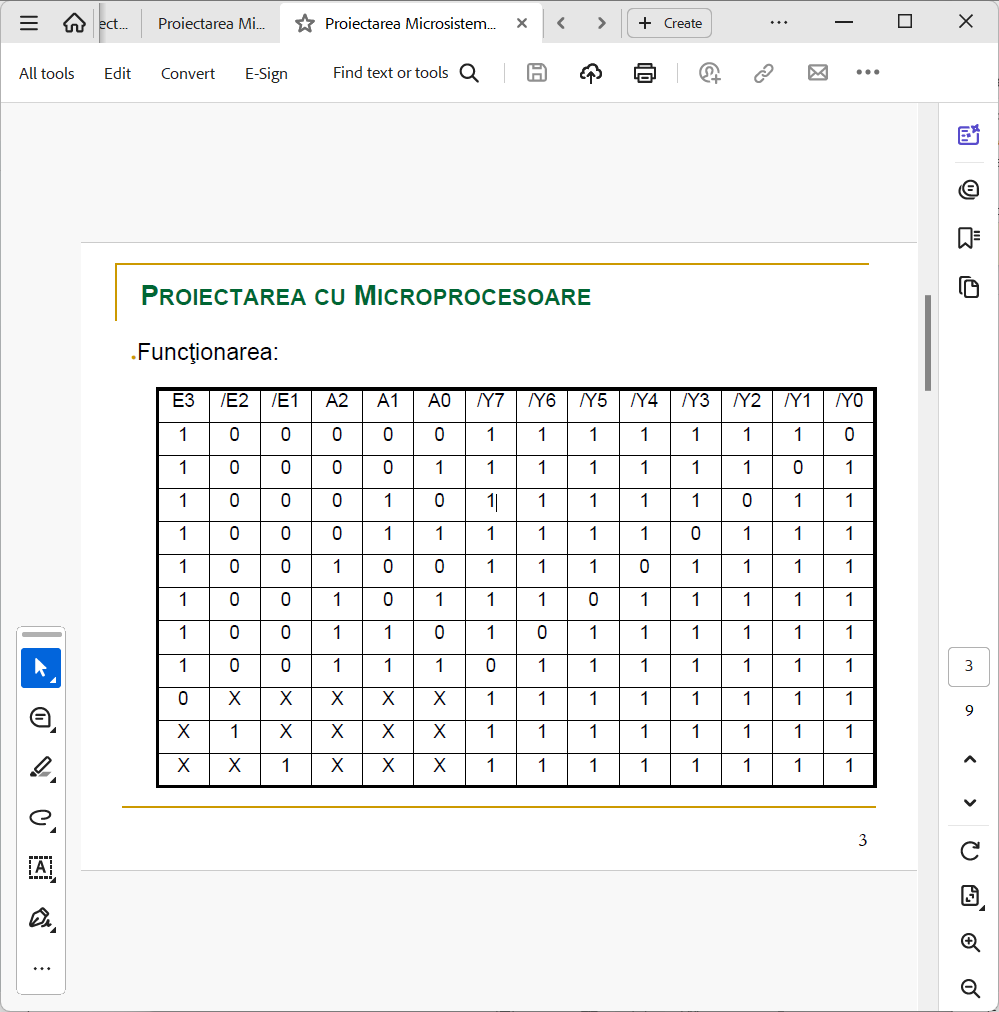
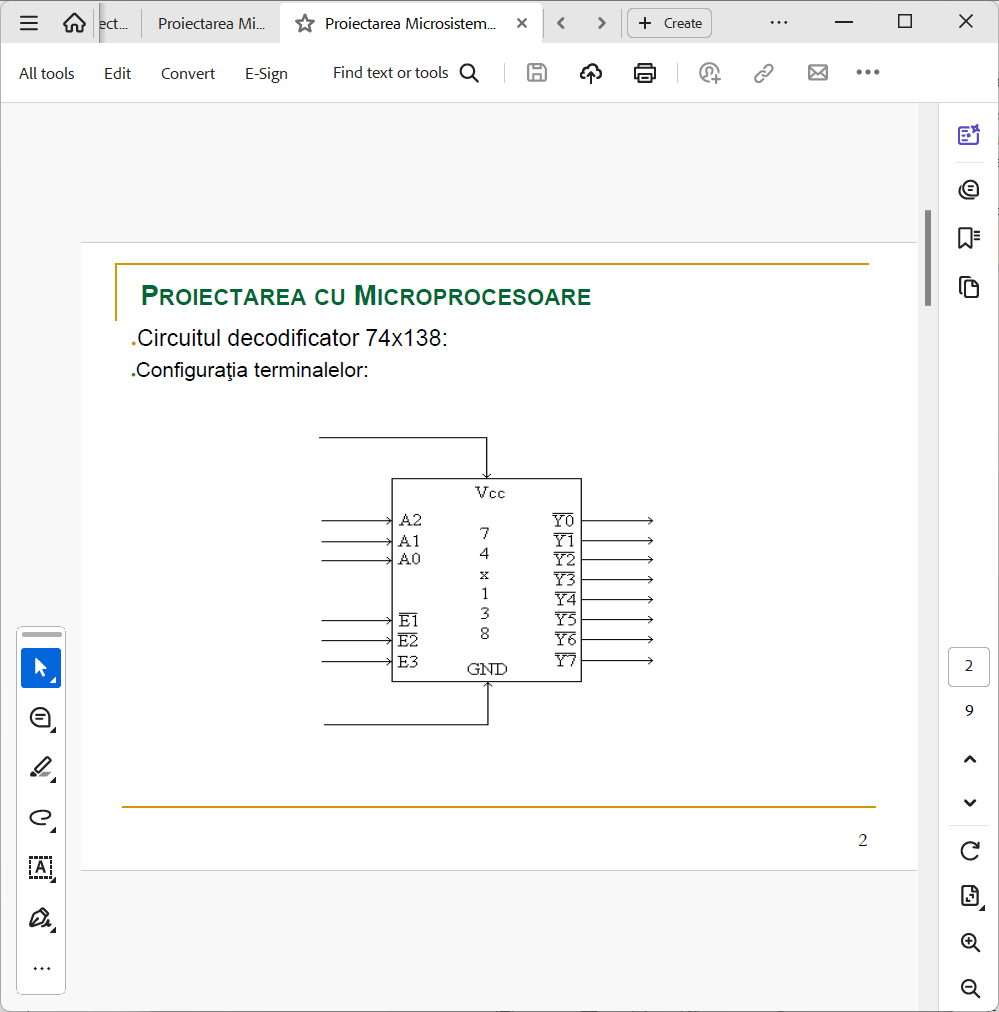
**Circuitul decodificator 74x138:**

Circuitul integrat 74x138 este un decodificator/demultiplexor foarte utilizat în aplicații logice digitale. Este parte din familia de circuite logice TTL (Transistor-Transistor Logic) și poate fi găsit sub diferite prefixe (cum ar fi 74LS138, 74HC138), care indică specificațiile detaliate și tehnologia utilizată.

**Caracteristici principale**

1. **Funcție principală:**
   * Decodificator de 3-to-8 linii: are 3 intrări de selecție și 8 ieșiri.
   * Selectează una dintre cele 8 ieșiri în funcție de combinația binară a intrărilor.
2. **Intrări**:
   * A, B, C: Intrări de selecție (3 biți).
   * E1,E2,E3: Linii de activare (enable lines).
     + E3 este activă pe nivel logic înalt (1).
     + E1 și E2 sunt active pe nivel logic jos (0).
3. **Ieșiri**:
   * Y0 până la Y7: Ieșiri active pe nivel logic jos (0).
     + Ieșirea selectată este 0, iar celelalte sunt 1.
4. **Funcționalitate de activare:**

* Ieșirile sunt active numai dacă liniile de activare sunt configurate corect (E3 = 1, E2 = 0, E1 = 0).

●Configuraţia terminalelor: Funcţionarea:

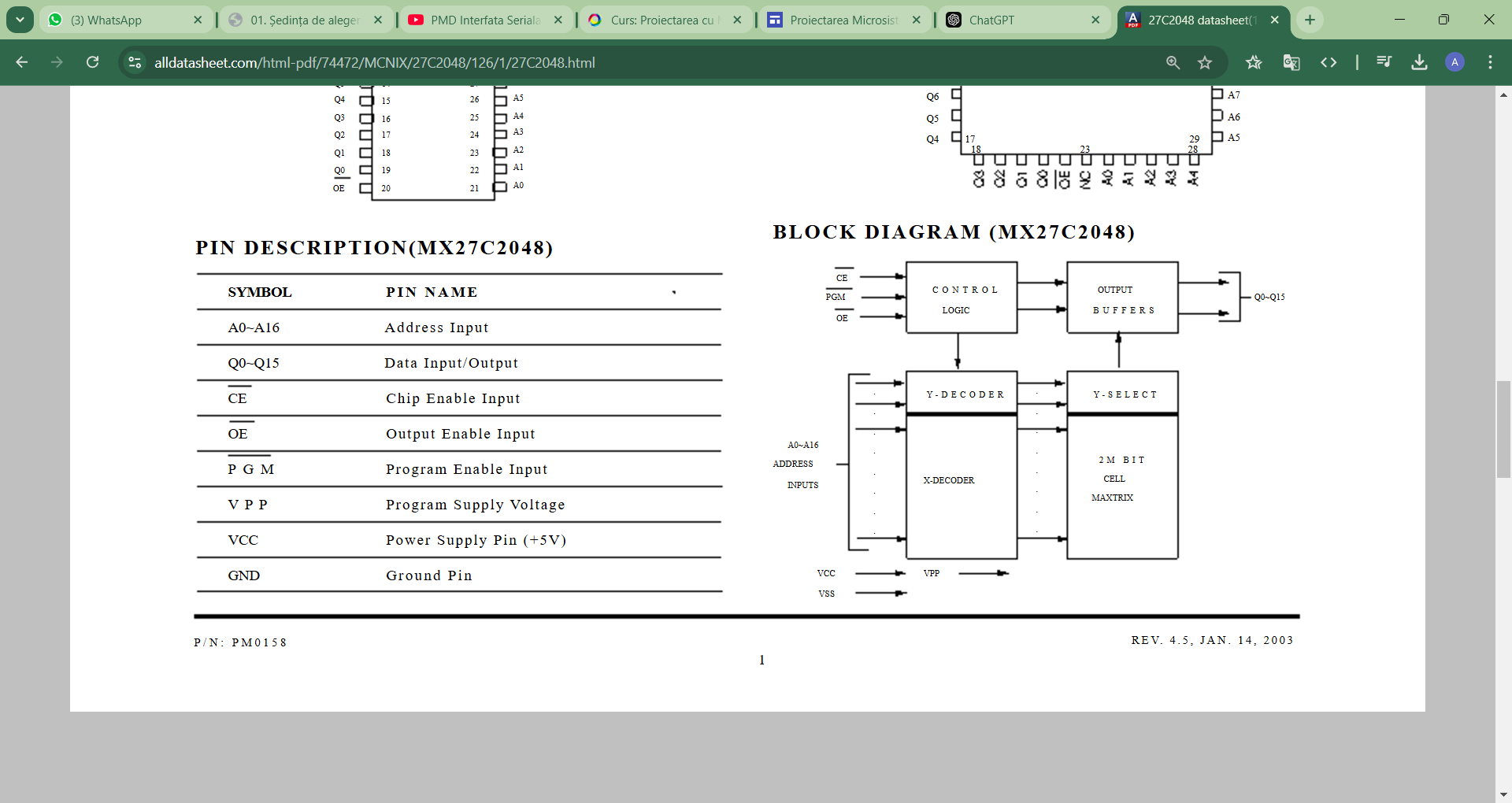
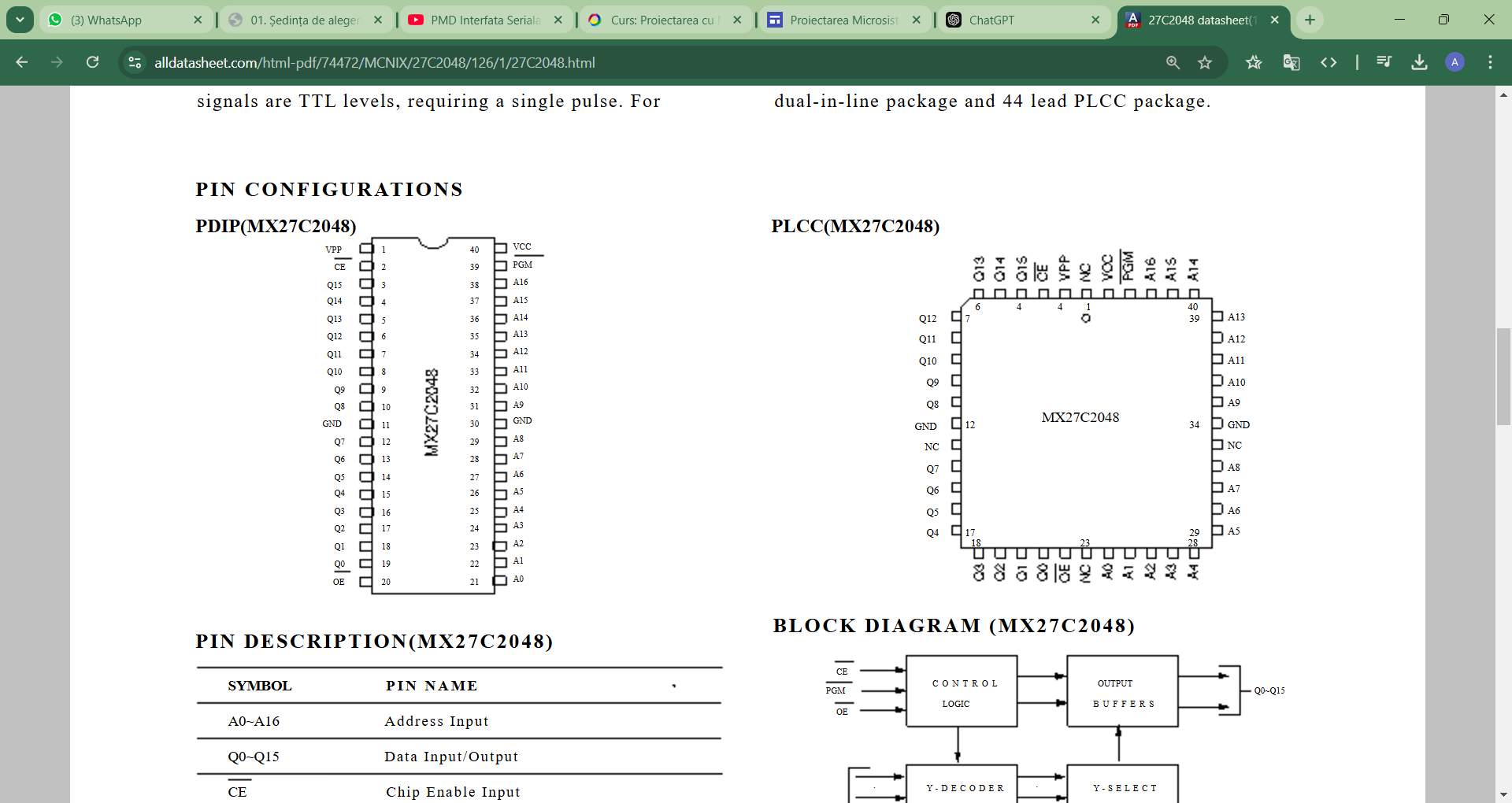
**Memoria EPROM** (traducere liberă: memorie *read only* programabilă sau anulabilă) este un tip de memorie de calculator nevolatilă, adică o memorie care își păstrează datele chiar și când i se întrerupe alimentarea cu curent electric. EPROM-urile sunt prin construcție o serie de tablouri de tranzistoare (porți logice) programate individual cu ajutorul unui dispozitiv electronic care furnizează tensiuni mai înalte decât cele pentru circuite digitale normale. Odată programate, un EPROM poate fi șters doar prin expunerea lui la lumină ultravioletă puternică. EPROM-urile au în partea superioară o fereastră de cuarț topit și resolidificat, prin care se vede cipul de siliciu propriu-zis. Acesta permite expunerea la lumină ultravioletă pentru ștergere (aducere la zero) a memoriei.

**Circuitul 27C2048**

Este un circuit EPROM avand capacitatea de 256KB și un timp de acces la memorie de obicei, între **100 ns** și **150 ns**, în funcție de model.

Circuitul are:

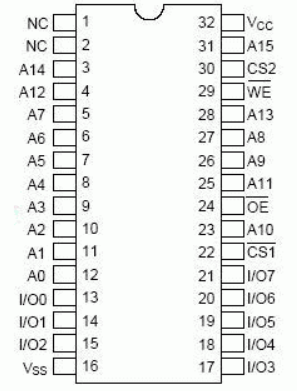
* 18 intrări de adrese, A17-A0;
* Linii de date, Q15-Q0;
* linie de selectie si programare, !CE(activează accesul la memorie) / !PGM
* Linie de validare a iesirilor !OE;
* intrare pentru tensiunea de programare, Vpp;
* Alimentarile Vcc si GND.



**Memoria SRAM** (Static Random Access Memory) este un tip de memorie semiconductoare; *static* subliniază 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 62512**

Este un circuit SRAM având capacitatea de 64KB și un timp de acces la memorie de 70 ns



Circuitul prezintă:

* 16 intrări de adrese, A15-A0;
* input/output, I/O7-I/O0;
* linie de selecție - !CS
* o linie de validare a ieșirilor !OE;
* o linie de validare a screrii !WE;
* alimentările VCC și GND.

**Decodificarea memoriilor**

**256KB memorie EPROM:**

256kb=28 \* 210 = 218 biţi

Adresa de început este: 00000H

Adresa finală este: 3FFFFH

Vom folosi un circuit de tipul 27C2048 care are capacitatea de 256KB.

**64KB memorie SRAM:**

64kb=26  \* 210= 216 biţi

Adresa de început este: 40000H

Adresa finală este: 4FFFFH

Vom folosi doua circuite de tipul 62512 care au capacitatea de 64KB(ne sunt necesare 16 iesiri, iar un singur circuit are doar 8, asadar vom imparti memoria sram in zone pare si impare).

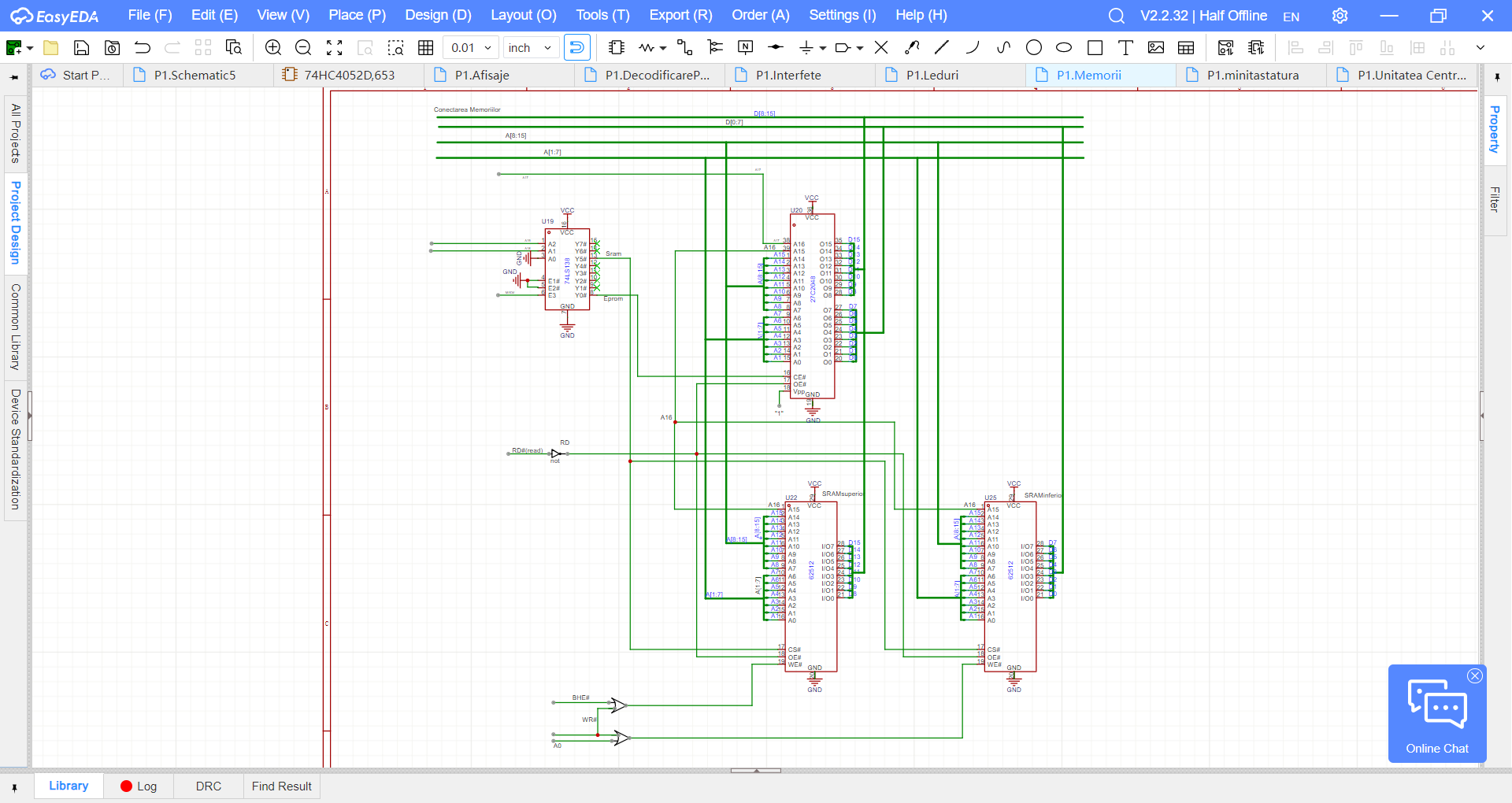
**Decodificarea completă a memoriei**

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

Memoria EPROM a fost plasată în zona 00000H-3FFFFH şi SRAM în zona 40000H-4FFFFH. În urma decodificării liniilor de adresă a celor două zone de memorie, liniile de adresă A19, A18 generează semnale de selecţie.Pentru semnalul de selecție rămas voi pune 0(adică îl voi lega la GND).

EPROM=(~A19)\*(~A18)\*(0)=(~Y0)

SRAM=(~A19)\*A18\*(0)=(~Y5)

****

**3. Interfața serială și paralelă**

**3.1 Conectarea porturilor**

Interfață serială zona 1: -port date: 0AF0H

-port comenzi: 0AF2H

Interfață serială zona 2: - port date: 0BF0H

- port comenzi: 0BF2H

Interfață paralelă zona 1: - port A: 0D70H

- port B: 0D72H

- port C: 0D74H

- port COMANDA: 0D76H

Interfață paralelă zona 2: - port A: 0C70H

- port B: 0C72H

- port C: 0C74H

- port COMANDA: 0C76H

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| mem | A15 | A14 | A13 | A12 | A11 | A10 | A9 | A8 | A7 | A6 | A5 | A4 | A3 | A2 | A1 | A0 |  |
| 0AF0H | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | Interfata Seriala |
| 0AF2H | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | Zona1 |
| 0BF0H | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | Interfata Seriala |
| 0BF2H | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | Zona2 |
| 0D70H | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | Interfata paralela |
| 0D72H | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 |  |
| 0D74H | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | Zona1 |
| 0D76H | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 |  |
| 0C70H | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | Interfata paralela |
| 0C72H | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 |  |
| 0C74H | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | Zona2 |
| 0C76H | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 |  |
| 0E10H | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | timer |
| 0E16H | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 |  |

Folosind **Decodificatorul 74x138** , se vor activa ieșirile dacă:

(~E1) = M/~IO

(~E2) = ((A15+A14)+(A12+A11))+A3

E3 = A11\*A4

Selecția pentru circuitul 8251 în funcție de comutatorul S1:

Pentru zona 1: Is51(1)=A9\*(~A8)\*A7=(~Y5)

Pentru zona 2: Is51(2)=A9\*A8\*A7=(~Y7)

Selecția pt circuitul 8255 în funcție de comutatorul S2:

Pentru zona 1: Is55(1)=(~A9)\*A8\*(~A7)=(~Y2)

Pentru zona 2: Is55(2)=(~A9)\*(~A8)\*(~A7)=(~Y0)

Pentru Timer:

Is53=A9\*(~A8)\*(~A7)

**3.2 Interfața serială**

**Circuitul 8251 USART (Universal Synchronous Asynchronous Receiver Transmitter)**

Funcție: Permite comunicarea serială sincronă sau asincronă între microprocesor și alte dispozitive.

Componente cheie:

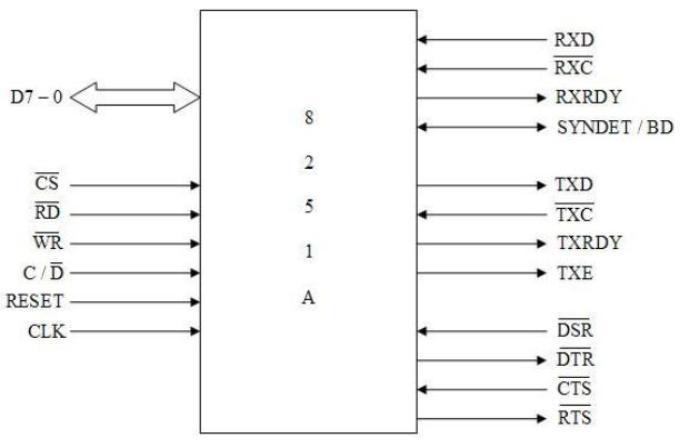
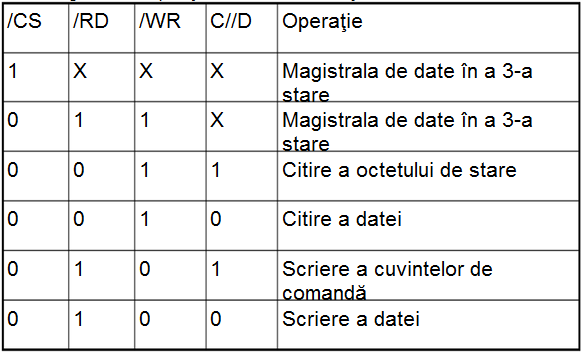
* Registru de comandă: Configurează modul (sincron/asincron), paritatea, lungimea cuvântului, etc.
* Registru de stare: Monitorizează starea transmisiei/recepției.
* Registru de date: Utilizat pentru trimiterea și primirea caracterelor.

Semnale principale:

* TxD/RxD: Transmitere și recepție date.
* CLK: Sincronizează comunicarea.

Aplicații: Interfațare cu modemuri, terminale, alte microcontrolere.

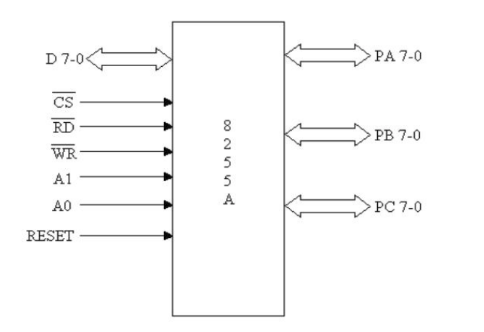
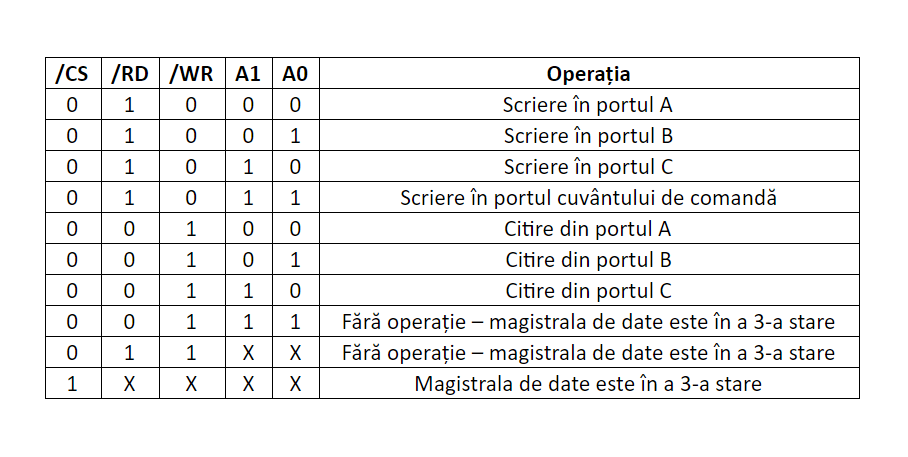
**Funcţionarea circuitului:**

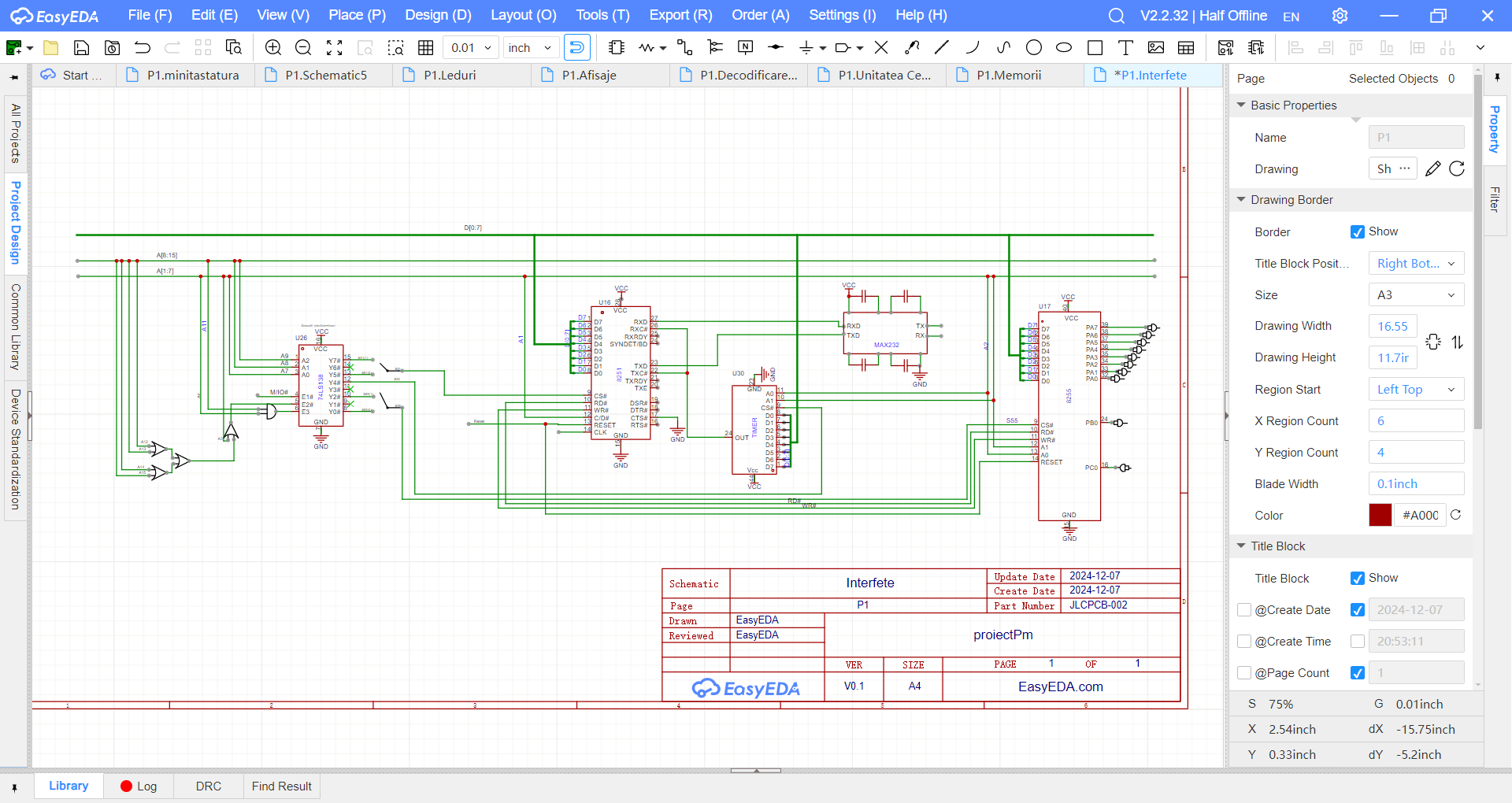
 

**3.3 Interfaţa paralelă**

* Funcție: Comunicare paralelă cu periferice prin porturi configurabile.
* Porturi:
  + Port A/B: 8 biți (intrare sau ieșire).
  + Port C: 8 biți (divizibil în două subporturi de 4 biți).
* Moduri de operare:
  + Mod 0: Intrare/ieșire simplă.
  + Mod 1: Intrare/ieșire sincronizată cu semnale de control.
  + Mod 2: Comunicare bidirecțională.
* Aplicații: Controlul tastaturilor, LED-urilor, afișajelor sau senzorilor.

**Funcţionarea circuitului:**



**4.Minitastatura cu 12 contacte**

Minitastatura cu 12 contacte este un dispozitiv de intrare organizat într-o matrice 4x3 (4 rânduri și 3 coloane), utilizată în sisteme embedded pentru introducerea de date numerice. Fiecare tastă închide un circuit între un rând și o coloană. Funcționează prin scanare: se activează succesiv rândurile și se verifică semnalele pe coloane pentru a identifica tasta apăsată. Este folosită frecvent în sisteme de securitate, controlere sau telefoane.

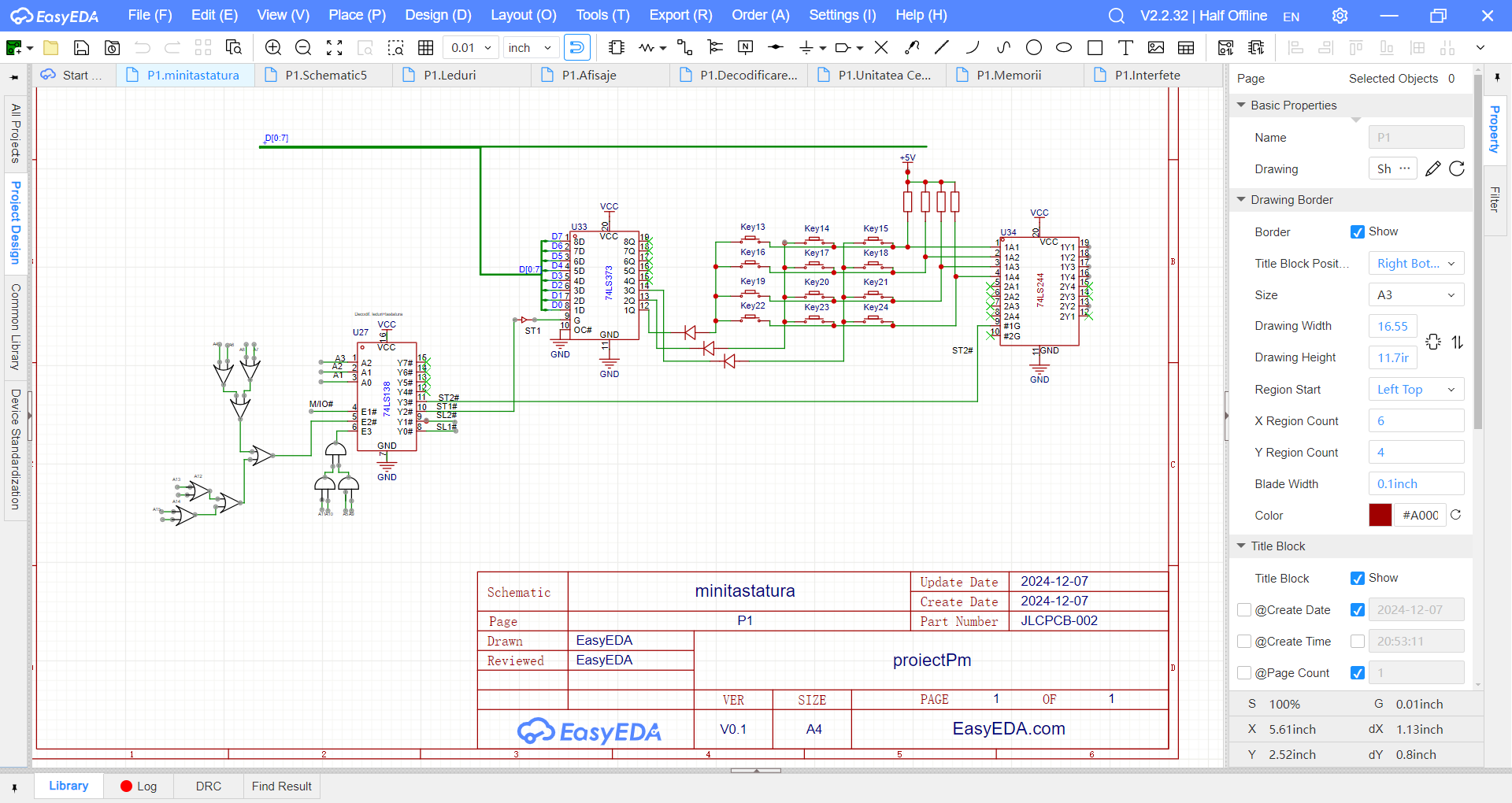
**Porturi de intrare -** 0E24H

**Porturi de iesire -** 0E26H

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| mem | A15 | A14 | A13 | A12 | A11 | A10 | A9 | A8 | A7 | A6 | A5 | A4 | A3 | A2 | A1 | A0 |  |
| 0AF0H | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | Interfata Seriala |
| 0AF2H | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | Zona1 |
| 0BF0H | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | Interfata Seriala |
| 0BF2H | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | Zona2 |
| 0D70H | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | Interfata paralela |
| 0D72H | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 |  |
| 0D74H | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | Zona1 |
| 0D76H | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 |  |
| 0C70H | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | Interfata paralela |
| 0C72H | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 |  |
| 0C74H | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | Zona2 |
| 0C76H | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 |  |
| 0E10H | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | timer |
| 0E16H | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 |  |
| 0E24H | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | TATST1 |
| 0E26H | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | TAST2 |

ST1= (~A3)\*A2\*(~A1)= (~Y2)

ST2=(~A3)\*A2\*A1 =(~Y3)

****

**5.Led-uri (10)**

LED (light emitting diode) este o diodă semiconductoare, care emite lumină la polarizarea directă joncțiunii p-n. Pentru ca un led sa fie aprins la ieșirea portului trebuie sa fie 0 logic. Se folosește un registru 74LS373 conectate la magistrală pentru a le menține aprinse sau stinse.

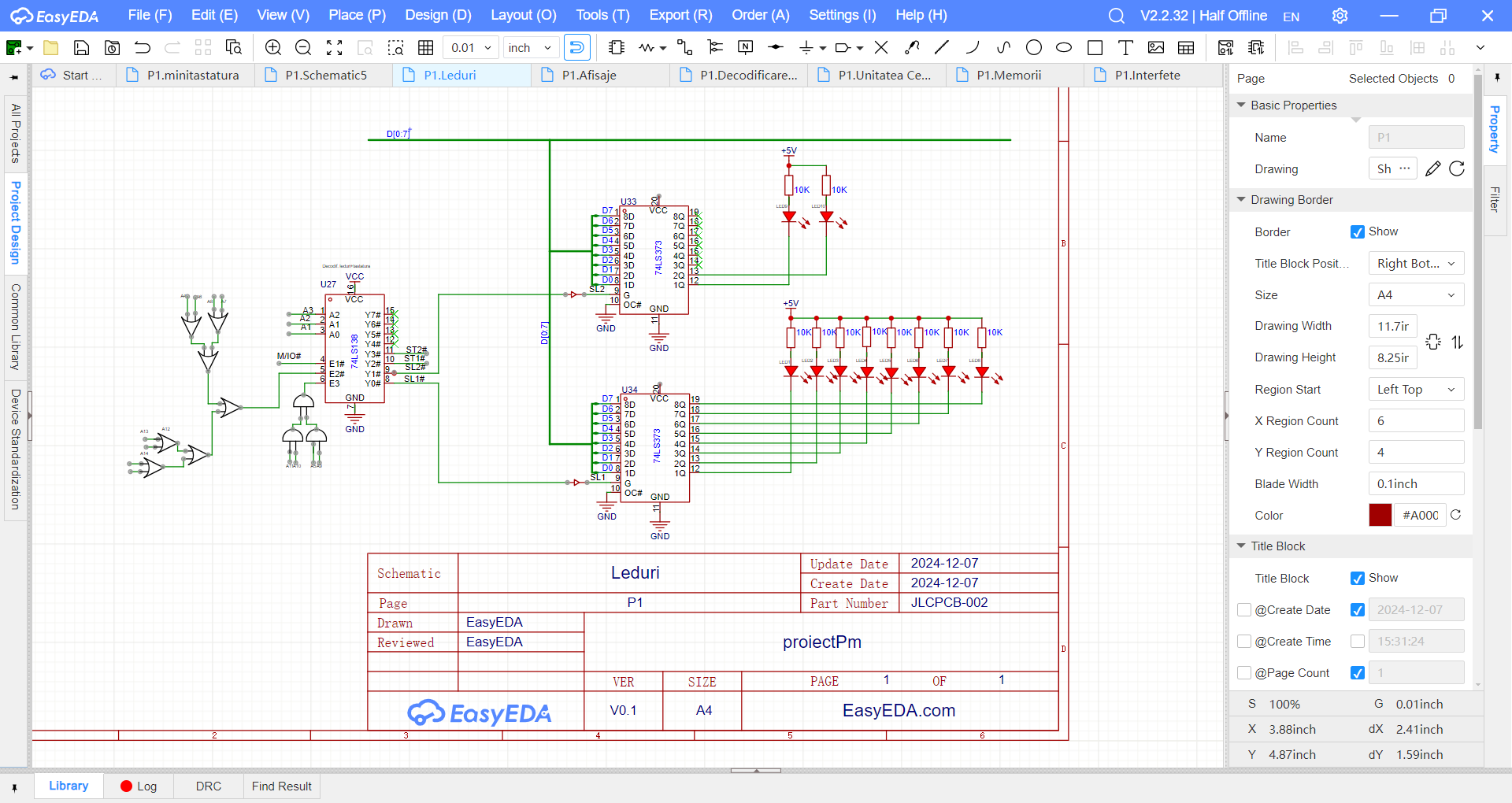
**Portul L1=** 0E20H

**Portul L2=**0E22H

SL1= (~A3)\*(~A2)\*(~A1) =(~Y0)

SL2= (~A3)\*A2\*(~A1)= (~Y1)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| mem | A15 | A14 | A13 | A12 | A11 | A10 | A9 | A8 | A7 | A6 | A5 | A4 | A3 | A2 | A1 | A0 |  |
| 0AF0H | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | Interfata Seriala |
| 0AF2H | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | Zona1 |
| 0BF0H | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | Interfata Seriala |
| 0BF2H | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | Zona2 |
| 0D70H | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | Interfata paralela |
| 0D72H | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 |  |
| 0D74H | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | Zona1 |
| 0D76H | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 |  |
| 0C70H | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | Interfata paralela |
| 0C72H | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 |  |
| 0C74H | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | Zona2 |
| 0C76H | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 |  |
| 0E10H | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | timer |
| 0E16H | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 |  |
| 0E20H | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | LED1 |
| 0E22H | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | LED2 |
| 0E24H | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | TATST1 |
| 0E26H | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | TAST2 |



**6.Modul de afişare cu 7 segmente, cu 6 ranguri**

Circuitele de afișare cu 7 segmente pot afișa valori hexazecimale, prin activarea corespunzătoare a intrărilor A-G. Cele 6 ranguri ale afișajului au fiecare câte un port de ieșire. Porturile sunt conectate la registre 74LS373, fiecare având un semnal de selecție dedicat: SA1, SA2, SA3, SA4, SA5, SA6.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| mem | A15 | A14 | A13 | A12 | A11 | A10 | A9 | A8 | A7 | A6 | A5 | A4 | A3 | A2 | A1 | A0 |  |
| 0AF0H | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | Interfata Seriala |
| 0AF2H | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | Zona1 |
| 0BF0H | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | Interfata Seriala |
| 0BF2H | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | Zona2 |
| 0D70H | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | Interfata paralela |
| 0D72H | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 |  |
| 0D74H | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | Zona1 |
| 0D76H | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 |  |
| 0C70H | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | Interfata paralela |
| 0C72H | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 |  |
| 0C74H | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | Zona2 |
| 0C76H | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 |  |
| 0E10H | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | timer |
| 0E16H | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 |  |
| 0E00H | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | AFISAJ1 |
| 0E02H | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | AFISAJ2 |
| 0E04H | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | AFISAJ3 |
| 0E06H | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | AFISAJ4 |
| 0E08H | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | AFISAJ5 |
| 0E0AH | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | AFISAJ6 |
| 0E20H | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | LED1 |
| 0E22H | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | LED2 |
| 0E24H | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | TATST1 |
| 0E26H | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | TAST2 |

SA1: (~A3)\*(~A2)\*(~A1)= (~Y0)

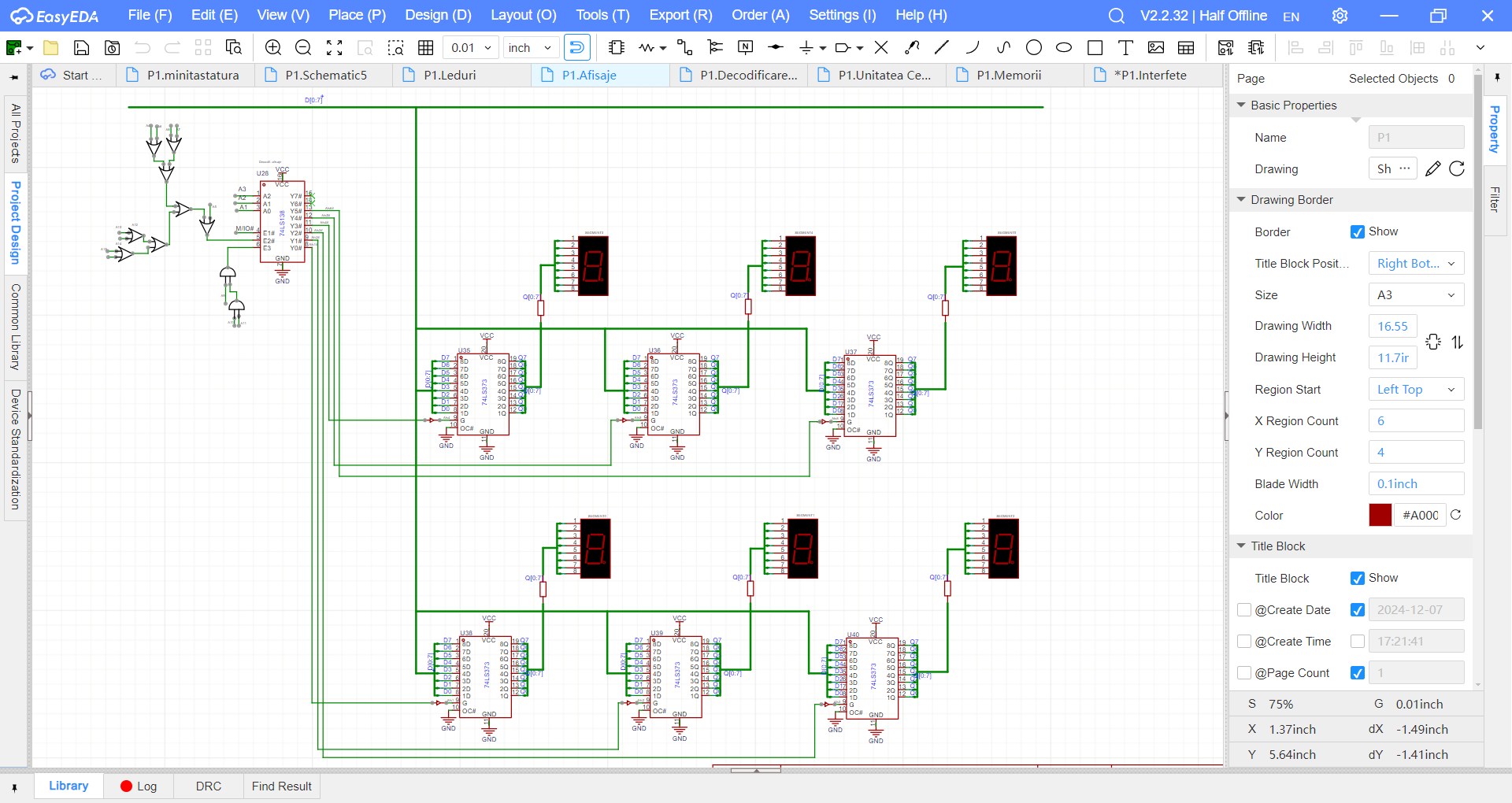
SA2: (~A3)\*(~A2)\*(A1) =(~Y1)

SA3: (~A3)\*(A2)\*(~A1) =(~Y2)

SA4: (~A3)\*(A2)\*(A1) =(~Y3)

SA5: (A3)\*(~A2)\*(~A1)= (~Y4)

SA6: (A3)\*(~A2)\*(A1) =(~Y5)



**Decodificare Porturi :**

Am folosit 3 decodificatoare 74x138, unul pentru interfețe și timer, unul pentru LED-uri și tastatură și altul pentru afișaje.

Decodificatorul pentru interfețe și timer se activează dacă:

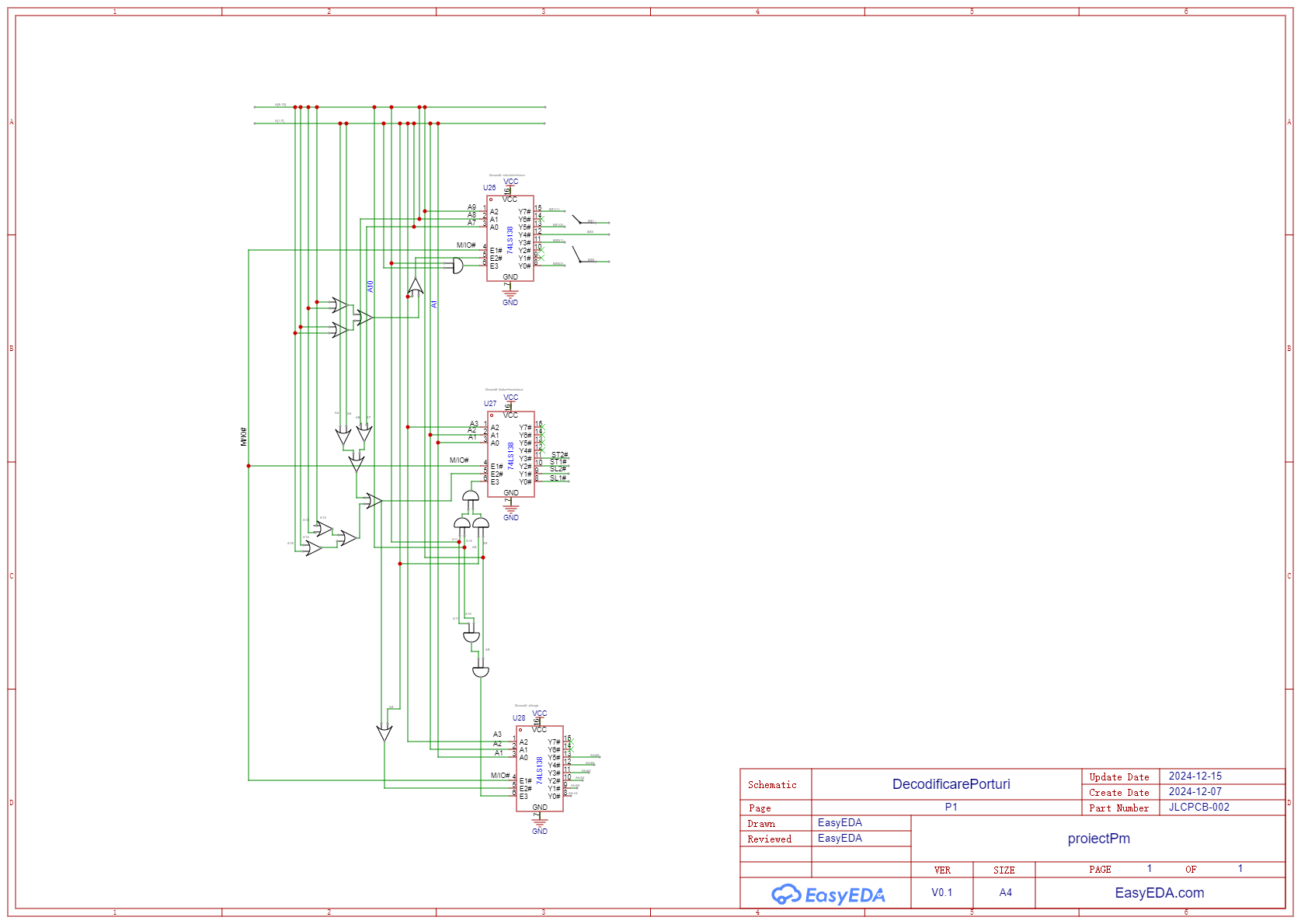
* (~E1) = M/~IO
* (~E2) = ((A15 + A14) + (A12 + A11)) + A3
* E3 = A11 \* A4

Decodificatorul pentru LED-uri și tastatură se activează dacă:

* (~E1) = M/~IO
* (~E2) = (((A15 + A14) + (A12 + A13)) + ((A8 + A7) + (A6 + A4)))
* E3 = A11 \* A10 \* A9 \* A5

Decodificatorul pentru afișaje se activează dacă:

* (~E1) = M/~IO
* (~E2) = (((A15 + A14) + (A12 + A13)) + ((A8 + A7) + (A6 + A5))) + A4
* E3 = A11 \* A10 \* A9



**7.Subrutine (programe necesare)**

**7.1** **Rutina de programare a circuitului 8251**

**Configurarea circuitului 8251**

8251 este configurat folosind cuvinte de mod și de comandă:

Cuvânt de mod: Definește modul de funcționare (sincron/asynchron), lungimea cuvântului de date, paritatea etc.

Cuvânt de comandă: Activează funcționalități precum transmisia, recepția, resetarea etc.

Datele iniţiale ale transferului : 8 biţi de date, fără paritate , factor de multiplicare 16, rata de transfer 9600 bps;

Adresele de port:

0AF0H (0BF0H )date

0AF2H (0BF2H) comenzi / stări;

**Rutina de programare**

CONFIG\_8251:

MOV DX, 0AF2H ; (sau 0BF2H)Adresa portului de stare/comandă

MOV AL, 0CEH ; Cuvânt de mod: 8 biți, fără paritate, factor multiplicare 16(11001110B)

OUT DX, AL ; Scriere în portul de comandă

MOV AL, 15H ; Cuvânt de comandă: activare transmisie și recepție(00010101B)

OUT DX, AL ; Scriere în portul de comandă

RET

**7.2** **Rutina de programare a circuitului 8255**

Modul de lucru 0 se mai numește și mod intrare/ ieșire și se folosește pentru operații simple de intrare/ ieșire fără dialog.

Adresele de port:0D70H (0C70H) portul A,

0D72H (0C72H) portul B,

0D74H (0C74H) portul C

0D76H (0C76H) RCC;(portul de comanda)

**Rutina de programare**

CONFIG\_8255:

MOV DX, 0D76H ;(sau 0C76H) Adresa registrului de control

MOV AL, 81H ; Cuvânt de comandă: 1000 0001B

; - Port A: ieșire

; - Port B: ieșire

; - Port C inferior: intrare

OUT DX, AL ; Scriere în registrul de control

RET

**7.3 Rutina de emisie caracter pe interfaţa serială**

TR: MOV DX, 0AF2H ;(sau 0BF2H)adresa portului de stare/comanda

IN AL, DX ; citire stare 8251

RCR AL, 1 ; testare bit TxRDY (daca e egal cu 0)

JNC TR ; buclă până când TxRDY este active(dacă bitul corespunzător e 0, reluăm operația)

MOV AL, CL ; caracterul de transmisie este înregistrat în CL il copiaza in AL

MOV DX, 0AF0H ; (sau 0BF0H)adresa portului de date

OUT DX, AL ; transmisie caracter

RET

**7.4 Rutina de recepţie caracter pe interfaţa serială**

REC: MOV DX, 0AF2H ;(sau 0BF2H)adresa portului de stare/comanda

IN AL, DX ; citire stare 8251

RCR AL, 2 ; testare bit RxRDY(verificam bitul 1)

JNC REC ; buclă până când RxRDY este active (dacă bitul corespunzător e 0, reluăm operația)

MOV DX, 0AF0H ; (sau 0BF0H)adresa portului de date

IN AL, DX ; citire caracter din 8251

MOV CL, AL ; stocare caracter în registrul CL

RET

**7.5 Rutina de emisie caracter pe interfaţă paralelă**

PAR: MOV DX, 0D74H ; (sau 0C74H)Adresa portului C (stare BUSY)

IN AL, DX ; Citire stare BUSY (port C)

RCR AL, 1 ; Testare bit BUSY (Bit 0) pentru a verifica dacă receptorul este liber

JNC PAR ; Buclă până când perifericul nu mai este ocupat(daca bitul respectiv e 0, reluăm operația)

MOV AL, CL ; Caracterul de transmisie este înregistrat în CL

MOV DX, 0D70H ; (sau 0C70H) Adresa portului A

OUT DX, AL ; Scriere caracter pe portul A (port de date)

OR AL, 01H ; Activare /STB = 1(modificăm bitul 0 pe 1)

MOV DX, 0D72H ;(sau 0C72H) Adresa portului B

OUT DX, AL ; Semnalizare către receptor că datele sunt disponibile (/STB = 1)

AND AL, 0FEH ; Dezactivare /STB = 0

OUT DX, AL ; Semnalizez receptorului că sunt date de citit

; (Opțional) Activare din nou /STB pentru o a doua notificare

OR AL, 01H ; Activare din nou /STB = 1

OUT DX, AL

RET

**7.6 Rutina de scanare a minitastaturii**

Minitastatura are structura:

1 2 3

4 5 6

7 8 9

\* 0 #

; se pune 0 pe prima coloană şi se verifică dacă s-au acţionat tastele 1, 4, 7, \*

TASTATURA:

MOV AL, 0FEH ; 0FEH -> 11111110, generăm 0 logic pe prima coloană pentru a scana tastele 1, 4, 7, \*

MOV DX, 0E24H ; Adresa pentru scrierea pe coloane

OUT DX, AL

MOV DX, 0E26H ; Adresa pentru citirea liniilor

IN AL, DX ; se citește linia

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

JZ TASTA1 ; dacă da, apelăm rutina TASTA1

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 7

JZ TASTA7

IN AL, DX

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

JZ TASTA\_STAR

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

MOV AL, 0FDH ; generăm 0 logic pe a doua coloană pentru a scana tastele 2, 5, 8, 0

MOV DX, 0E24H

OUT DX, AL

MOV DX, 0E26H

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 5

JZ TASTA5

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 0

JZ TASTA0

; se pune 0 pe a 3-a coloană şi se verifică dacă s-au acţionat tastele 3, 6, 9, #

MOV AL, 0FBH ; generăm 0 logic pe a treia coloană pentru a scana tastele 3, 6, 9, #

MOV DX, 0E24H

OUT DX, AL

MOV DX, 0E26H

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 6

JZ TASTA6

IN AL, DX

AND AL, 04H ; se verifică dacă tasta apăsată este tasta 9

JZ TASTA9

IN AL, DX

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

JZ TASTA\_HASH

JMP TASTATURA ; se reia scanarea tastaturii

; Rutine pentru fiecare tastă

TASTA1: CALL DELAY ; se așteaptă oprirea vibrațiilor

TAS1: MOV DX, 0E26H

IN AL, DX

AND AL, 01H

JZ TAS1 ; se așteaptă dezactivarea tastei

CALL DELAY ; se așteaptă oprirea vibrațiilor

MOV CL, 01H ; CL – număr corespunzător tastei

RET

TASTA2: CALL DELAY

TAS2: MOV DX, 0E26H

IN AL, DX

AND AL, 01H

JZ TAS2

CALL DELAY

MOV CL, 02H

RET

TAS4: MOV DX, 0E26H

IN AL, DX

AND AL, 02H

JZ TAS2

CALL DELAY

MOV CL, 04H

RET

…

TAS7: MOV DX, 0E26H

IN AL, DX

AND AL, 04H

JZ TAS2

CALL DELAY

MOV CL, 07H

RET

…

; Continuă similar pentru celelalte taste...

TASTA\_STAR: CALL DELAY

TAS\_STAR: MOV DX, 0E26H

IN AL, DX

AND AL, 08H

JZ TAS\_STAR

CALL DELAY

MOV CL, 0AH ; Exemplu: atribuire simbol pentru tasta '\*'

RET

TASTA\_HASH: CALL DELAY

TAS\_HASH: MOV DX, 0E26H

IN AL, DX

AND AL, 08H

JZ TAS\_HASH

CALL DELAY

MOV CL, 0BH ; Exemplu: atribuire simbol pentru tasta '#'

RET

**7.7 Rutinele de aprindere/ stingere a unui led**

;anod comun : ledul va fi aprins când „0” este activ

0 - aprins

1 – stins

Secvența ca ledurile să lumineze(cate unul pe rand):

MOV DX, 0E20H ; Adresa primelor 8 LED-uri

MOV AL, 11111110B ; punem 0 pe poziția 0, deci ledul 1

OUT DX, AL

MOV DX, 0E20H ; Adresa primelor 8 LED-uri

MOV AL, 11111101B ; punem 0 pe poziția 1, deci ledul 2

OUT DX, AL

MOV DX, 0E20H ; Adresa primelor 8 LED-uri

MOV AL, 11111011B ; punem 0 pe poziția 2, deci ledul 3

OUT DX, AL

MOV DX, 0E20H ; Adresa primelor 8 LED-uri

MOV AL, 11110111B ; punem 0 pe poziția 3, deci ledul 4

OUT DX, AL

MOV DX, 0E20H ; Adresa primelor 8 LED-uri

MOV AL, 11101111B ; punem 0 pe poziția 4, deci ledul 5

OUT DX, AL

MOV DX, 0E20H ; Adresa primelor 8 LED-uri

MOV AL, 11011111B ; punem 0 pe poziția 5, deci ledul 6

OUT DX, AL

MOV DX, 0E20H ; Adresa primelor 8 LED-uri

MOV AL, 10111111B ; punem 0 pe poziția 6, deci ledul 7

OUT DX, AL

MOV DX, 0E20H ; Adresa primelor 8 LED-uri

MOV AL, 01111111B ; punem 0 pe poziția 7, deci ledul 8

OUT DX, AL

MOV DX, 0E22H ; Adresa următorilor 2 LED-uri

MOV AL, 11111110 ; punem 0 pe poziția 0, deci ledul 9

OUT DX, AL

MOV DX, 0E22H ; Adresa următorilor 2 LED-uri

MOV AL, 11111101B ; punem 0 pe poziția 1, deci ledul 10

OUT DX, AL

Secvența ca ledurile să lumineze(toate):

MOV DX, 0E20H

MOV AL, 00H ; aprind ledurile 1-8

OUT DX, AL

MOV DX, 0E22H

MOV AL, 00H ; aprind ledurile 9-10

OUT DX, AL

RET

Secvența ca ledurile să se stinga(toate):

MOV DX, 0E20H

MOV AL, FFH ; sting ledurile 1-8

OUT DX, AL

MOV DX, 0E22H

MOV AL, FFH ; sting ledurile 9-10

RET

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

SEG: MOV DX, 0E00H ; Adresa portului pentru primul afișaj

CAR0:

MOV AL, 0C0H ; Cod pentru afișarea '0'

OUT DX, AL

RET

CAR1:

MOV AL, 0F9H ; Cod pentru afișarea '1'

OUT DX, AL

RET

CAR2:

MOV AL, 0A4H ; Cod pentru afișarea '2'

OUT DX, AL

RET

CAR3:

MOV AL, 0B0H ; Cod pentru afișarea '3'

OUT DX, AL

RET

CAR4:

MOV AL, 099H ; Cod pentru afișarea '4'

OUT DX, AL

RET

CAR5:

MOV AL, 092H ; Cod pentru afișarea '5'

OUT DX, AL

RET

CAR6:

MOV AL, 082H ; Cod pentru afișarea '6'

OUT DX, AL

RET

CAR7:

MOV AL, 0F8H ; Cod pentru afișarea '7'

OUT DX, AL

RET

CAR8:

MOV AL, 080H ; Cod pentru afișarea '8'

OUT DX, AL

RET

CAR9:

MOV AL, 090H ; Cod pentru afișarea '9'

OUT DX, AL

RET

CARA:

MOV AL, 088H ; Cod pentru afișarea 'A'

OUT DX, AL

RET

CARB:

MOV AL, 083H ; Cod pentru afișarea 'B'

OUT DX, AL

RET

CARC:

MOV AL, 0C6H ; Cod pentru afișarea 'C'

OUT DX, AL

RET

CARD:

MOV AL, 0A1H ; Cod pentru afișarea 'D'

OUT DX, AL

RET

CARE:

MOV AL, 086H ; Cod pentru afișarea 'E'

OUT DX, AL

RET

CARF:

MOV AL, 0A6H ; Cod pentru afișarea 'F'

OUT DX, AL

RET

|  |  |
| --- | --- |
| **Caracter** | **Valoare in hexa** |
| **0** | **C0H** |
| **1** | **F9H** |
| **2** | **A4H** |
| **3** | **B0H** |
| **4** | **99H** |
| **5** | **92H** |
| **6** | **82H** |
| **7** | **F8H** |
| **8** | **80H** |
| **9** | **90H** |
| **A** | **88H** |
| **B** | **83H** |
| **C** | **C6H** |
| **D** | **A1H** |
| **E** | **86H** |
| **F** | **A6H** |

Bibliografie

* Curs Proiectarea cu Microprocesoare
* <https://sites.google.com/site/labpmd/Laborator/conectarea-memoriilor>
* <https://sites.google.com/site/labpmd/Laborator/interfata-paralela?authuser=0>
* <https://sites.google.com/site/labpmd/Laborator/serial?authuser=0>
* <https://sites.google.com/site/labpmd/Laborator/interfata-seriala---receptia?authuser=0>
* <https://www.youtube.com/watch?v=_EuN5_T2Ga8&list=PL9CHjpufJT0uJkAzUIvX1yY7kZzvUAyLG&index=2> (PMD - Conectarea memoriilor)
* <https://www.youtube.com/watch?v=Sq-Ptc5kpbU&list=PL9CHjpufJT0uJkAzUIvX1yY7kZzvUAyLG&index=3> (PMD Interfata Seriala si Paralela)
* <https://www.youtube.com/watch?v=BL-0bs9kDjk&list=PL9CHjpufJT0uJkAzUIvX1yY7kZzvUAyLG&index=4> (PMD Conectarea Perifericelor)