

# Curs 6

## 4. Conectarea porturilor de intrare/ ieșire

- Prin port de intrare/ ieșire sau, mai scurt, port, se înțelege un circuit, sau grup de circuite, care fac legătura între unitatea centrală a unui microsistem și dispozitivele și echipamentele periferice.
- Porturi + circuite de legătură între ele = interfață.
- Există o gamă diversă de dispozitive și echipamente periferice: led – uri, comutatoare, module de afișare cu segmente, module de afișare cu LCD, tastaturi, imprimante etc.
- Implementarea porturilor:
  - Cu 1 sau puține circuite: dacă UC se leagă la led – uri, comutatoare, module de afișare cu segmente, module de afișare cu LCD, tastaturi, bistabile cu funcțiuni speciale etc.
  - Cu mai multe circuite, dacă UC se leagă la un echipament periferic, uneori nefăcându-se distincția între port și interfață; ex.: microcalculatorul PC: porturile serial, paralel, USB; noțiunea de port este mai largă cuprinzând mai multe circuite, inclusiv un circuit specializat programabil sau o parte a unui circuit specializat programabil mai complex, precum și elemente mecanice de conectare.

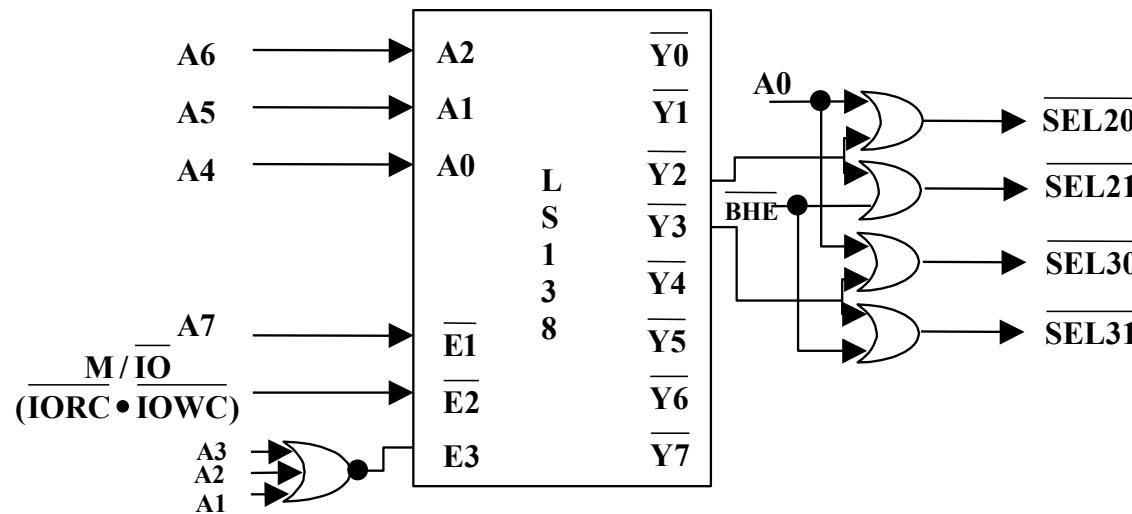
# Proiectarea cu Microprocesoare

## 4.1. Proiectarea unui decodificator de porturi

- Similară proiectării unui decodificator de memorie; diferențele care apar constau în:
  - Numărul de linii pe care apare adresa portului: la memorii adresele apar pe toată magistrala pe când la porturi adresele apar fie pe A7 – 0 fie pe A15 – 0;
  - Semnalul de comandă care indică transferul cu porturile: fie este altul decât cel pentru transferurile cu memoria fie este același dar cealaltă valoare logică este cea activă.
- Exemplu: să se proiecteze decodificatorul corespunzător următoarei hărți a porturilor:
  - P1 cu adresa 20H;
  - P2 cu adresa 21H;
  - P3 cu adresa 30H și
  - P4 cu adresa 31H;
  - Porturile sunt pe 8 biți, cu adrese pe 8 biți.

# Proiectarea cu Microprocesoare

- Schema decodificatorului este:



- Decodificarea este completă, poate fi și incompletă.

# Proiectarea cu Microprocesoare

- Există 2 posibilități de conectare a porturilor la unitatea centrală a unui microsistem:
  - În spațiul de intrare/ ieșire al microprocesorului: comunicarea microprocesorului cu ele se va face prin instrucțiuni dedicate de intrare/ ieșire, de tip IN/ OUT, iar în sistem va exista un decodificator de porturi, distinct de cel de memorie;
  - În spațiul de memorie al microprocesorului: comunicarea microprocesorului cu ele se va face cu instrucțiuni de lucru cu memoria Întrucât microprocesorul nu va face distincție între porturi și memorie, în sistem va exista doar decodificatorul de memorie iar porturilor le sunt alocate adrese sau zone din spațiul de memorie.
- Avantajele plasării porturilor în spațiul de memorie:
  - posibilitatea utilizării subsetului de instrucțiuni de lucru cu memoria mai bogat decât subsetul de instrucțiuni de lucru cu porturile;
  - reducerea resurselor microprocesorului, inclusiv al numărului de terminale.
- Dezavantaj:
  - Porturile vor ocupa zone din spațiul de adresare alocat memoriilor.

# Proiectarea cu Microprocesoare

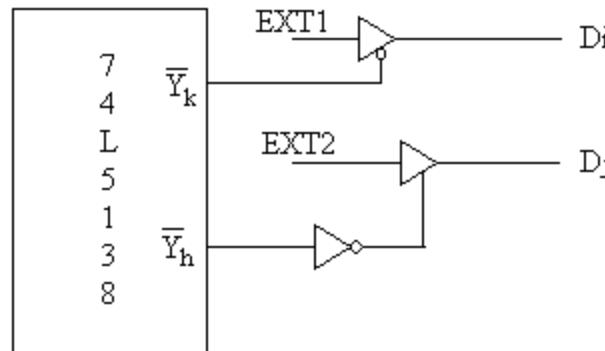
## 4.2. Tipuri de porturi

### ■ Portul implementat cu porți

- Este folosită pentru porturile de intrare;
- Ieșirile porților se conectează la linii de date ale microprocesorului, ca urmare este necesar să aibă facilitatea de a 3 – a stare;
- Semnalele care se citesc se conectează la intrările de date ale porților;
- Intrările de validare ale porților sunt comandate de una sau mai multe ieșiri dintr - un decodificator de porturi;
- Semnalele care se citesc sunt generate din exterior și se presupune că circuitul care le generează le și memorează până când este necesar; în caz contrar sunt prevăzute bistabile;
- Citirea se face în timpul ciclului de intrare.

# Proiectarea cu Microprocesoare

## ■ Schema de principiu:



- Schema trebuie să fie însoțită de o secvență de instrucțiuni; fie  $i = k = 0$  și  $j = h = 1$ , atunci secvența pentru citirea semnalului EXT1 este:
  - IN AL,00H
  - RCR AL,1
  - JNC ZERO; s-a citit 0 logic
    - ; s-a citit 1 logic
- S-au considerat conexiunile: A7 la A2, A6 la A1, A5 la A0, M//IO la /E1, GND la /E2 și 1 la E3

# Proiectarea cu Microprocesoare

- ❑ Explicarea secvenței:

**IN AL,00H**

- instrucțiunea provoacă citirea unui octet de la portul cu adresa 00H și încărcarea lui în jumătatea mai puțin semnificativă a registrului acumulator, AL; în timpul execuției acestei instrucțiuni se activează ieșirea  $Y_k$ , unde  $k = 0$ , ținând seama și de restul conexiunilor (A7 la A2, A6 la A1, A5 la A0, M//IO la /E1, GND la /E2 și 1 la E3); dacă s-ar fi folosit  $Y_1$  atunci adresa de port ar fi trebuit să fie 20H; semnalul EXT1 intră pe linia  $D_0$  a mag. de date și apoi în rangul  $A_0$  al registrului AL;

**RCR AL,1**

- RCR (Rotate Right through Carry) rotește conținutul registrului AL la dreapta prin indicatorul Carry (C); astfel rangul  $A_0$  ajunge în C;

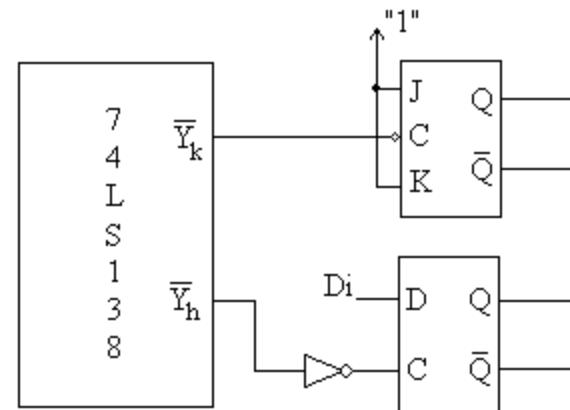
**JNC ZERO; s-a citit 0 logic**

**; s-a citit 1 logic**

- JNC (Jump if Not Carry) ramifică programul în funcție de ce s-a citit pe EXT1; dacă s-a citit 0, programul continuă la instrucțiunea cu eticheta ZERO; dacă s-a citit 1, programul continuă cu instrucțiunea de după JNC;

# Proiectarea cu Microprocesoare

- Portul implementat cu bistabile
  - Este folosită pentru porturile de ieșire;
  - Există mai multe variante:
    - cu bistabile:
      - de tip JK;
      - de tip D;
    - cu bistabile care comută:
      - pe nivelul impulsului de tact;
      - pe frontul impulsului de tact:
        - anterior;
        - posterior.
  - Schema de principiu:

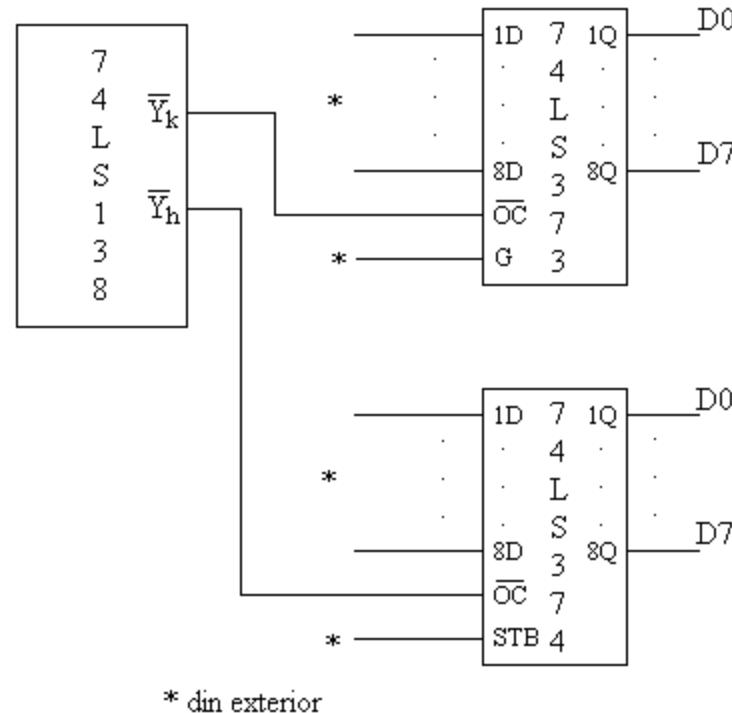


# Proiectarea cu Microprocesoare

- Fie  $k = 0$ ,  $h = 1$  și  $i = 7$ :
  - pentru modificarea ieșirii bistabilului de tip JK este suficientă execuția instrucțiunii OUT 00H,AL și bistabilul va trece în starea complementară celeia în care a fost;
  - utilizarea acestei soluții are dezavantajul că trebuie cunoscută starea prezentă a bistabilului înainte de a lansa în execuție instrucțiunea care îl modifică starea;
  - Ieșirea bistabilului de tip D poate fi comandată fără a fi necesară cunoașterea stării prezente întrucât are intrare de date;
  - Următoarea secvență provoacă trecerea bistabilului în starea 1:  
MOV AL,80H; D<sub>7</sub> = 1  
OUT 20H,AL
  - Următoarea secvență provoacă trecerea bistabilului în starea 0:  
MOV AL,00H; D<sub>7</sub> = 0  
OUT 20H,AL
  - Modificarea stării bistabilelor se face în timpul ciclului de ieșire.

# Proiectarea cu Microprocesoare

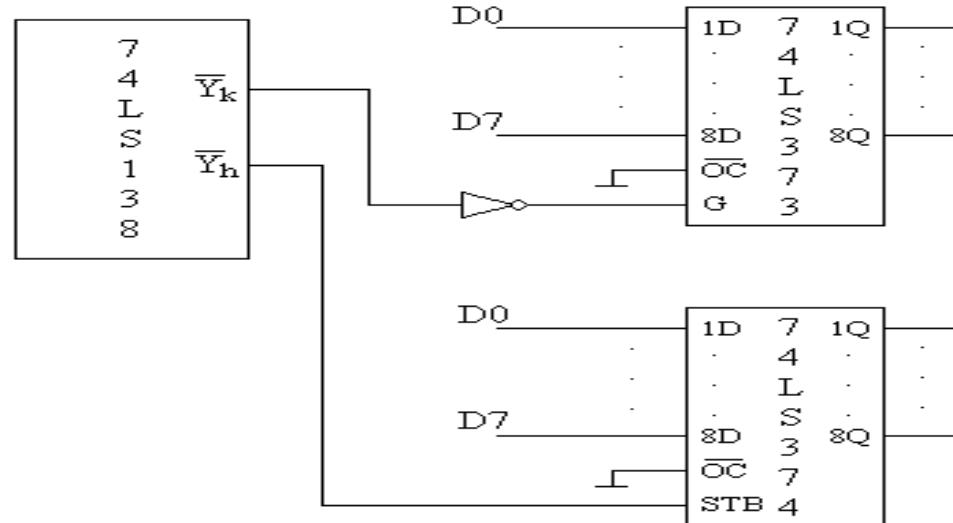
- Portul implementat cu registre
  - Porturi de intrare: soluția principală:



- Preluarea informației de la cele 2 registre se face prin instrucțiuni IN de la adresele care le corespund, mai exact în timpul ciclului de intrare;
- Înscrierea informației în registre se face din exterior.

# Proiectarea cu Microprocesoare

- ❑ Porturi de ieșire: extensie a implementării cu bistabile;
- ❑ Soluția principală pentru comanda registrelor 74x373 și 74x374:



- ❑ Fie  $k = 0$  și  $h = 1$ ; încărcarea unei configurații în registrele 74x373 și 74x374:

MOV AL, \_\_\_\_ H

OUT 00H,AL; pentru 373

OUT 20H,AL; pentru 374

- ❑ Încărcarea se va face în timpul ciclului de ieșire.

# Proiectarea cu Microprocesoare

- Portul implementat cu circuite specializate
  - Există circuite programabile, specializate pe diferite tipuri de aplicații, de exemplu interfață serială, interfață paralelă etc.
  - Datorită complexității sale, un asemenea circuit este comandat de microprocesor ca un grup de porturi ale căror adrese au un câmp comun și un câmp variabil;
  - Câmpul comun corespunde unei conexiuni făcute între o ieșire din decodificatorul de porturi și intrarea de selecție a circuitului specializat iar câmpul variabil corespunde legăturilor făcute între linii mai puțin semnificative din magistrala de adrese și intrări specifice ale circuitului specializat.

# Proiectarea cu Microprocesoare

## 4.3. Comanda unui semnal prin program

### ■ Semnificații:

- Citirea unui semnal din exteriorul microsistemu: soluția este cea prezentată la citirea unui semnal prin intermediul unui port implementat cu o poartă:
  - Semnalul este legat la intrarea de date a unei porți cu 3 stări;
  - Intrarea de validare a porții este comandată de o ieșire a unui decodificator de porturi;
  - Ieșirea porții este legată la o linie a magistralei de date a unității centrale;
  - Citirea se face cu o instrucțiune IN și apoi se face ramificarea.
- Poziționarea unui semnal la 1 sau 0 logic: soluția este cea prezentată la implementarea unui port cu un bistabil:
  - Valoarea semnalului este transmisă la intrarea de date a unui bistabil prin intermediul unei linii a magistralei de date;
  - Tactul bistabilului este comandat de o ieșire a unui decodificator de porturi;
  - Iesirea bistabilului este semnalul care trebuie comandat;
  - Poziționarea semnalului se face cu o instrucțiune OUT.

# Proiectarea cu Microprocesoare

## 5. Circuite specializate programabile (CSP)

### 5.1. Definiție și caracteristici

- Standardizarea soluțiilor pentru diversele tipuri de aplicații rezolvabile cu sisteme cu microprocesoare → circuite specializate pe o problemă.
- Cerințe:
  - Flexibilitate: pentru a acoperi toate modurile de lucru;
  - Programabilitate: pentru comunicare cu programatorul.
- Un circuit specializat este programabil în sensul că i se poate cere, prin intermediul unuia sau mai multor cuvinte de comandă, alegerea unor particularități ale tipului de aplicație pentru care circuitul a fost conceput.
- Cuvintele de comandă sunt transmise circuitului de unitatea centrală.
- Oferă informație de stare legată de ultimul transfer.

# Proiectarea cu Microprocesoare

## ■ Trăsături generale ale CSP:

- referitor la structura lor internă, circuitele specializate sunt văzute de unitatea centrală ca un grup de porturi de intrare/ ieșire ale căror adrese au un câmp comun, diferind, în general, prin valorile pe care le iau ultimele 1 – 2 ranguri;
- tot referitor la structura lor internă, se pot distinge 3 tipuri de porturi în cadrul circuitelor specializate: de comandă, de date și de stare;
- referitor la terminalele acestor circuite, ele se pot grupa în 2 mari categorii: pentru dialogul unitate centrală – circuit și circuit – periferic;
- referitor la tehnologie, sunt circuite realizate în tehnologia MOS, introducând încărcări mici, de ordinul  $\mu\text{A}$ ;
- referitor la viteza de lucru a acestor circuite, ea este adaptată la viteza de lucru a microprocesorului din a cărui familie fac parte.

## ■ CSP fac parte din familia unui microprocesor:

- pot fi conectate direct la microprocesorul respectiv și ușor, cu minim de circuite, la un microprocesor obținut din primul prin evoluție,
- se conectează dificil, cu multe circuite, la alte microprocesoare;

# Proiectarea cu Microprocesoare

## 5.2. Interfața serială

- Interfața serială constă în totalitatea circuitelor și programelor de bază care asigură comunicarea între unitatea centrală și un echipament periferic, aceasta fiind de tip bit după bit; comunicarea poate fi cu sau fără fir.
- Există multe tipuri de interfețe seriale:
  - Fără fir: Bluetooth, WiFi, ZigBee etc.
  - Cu fir: RS232, RS485, I<sup>2</sup>C, CAN, USB, Flexray etc.
  - Diferă prin:
    - Mediul de transmitere a informației;
    - Modalitatea de alocare a tensiunilor nivelor logice;
    - Parametri: rată de transfer, rezistență la perturbații, corecție a erorilor, metode de codificare, utilizarea sau nu a tactului, numărul de module care se pot conecta, ierarhizarea modulelor etc.
  - La PC: RS232 și USB.

# Proiectarea cu Microprocesoare

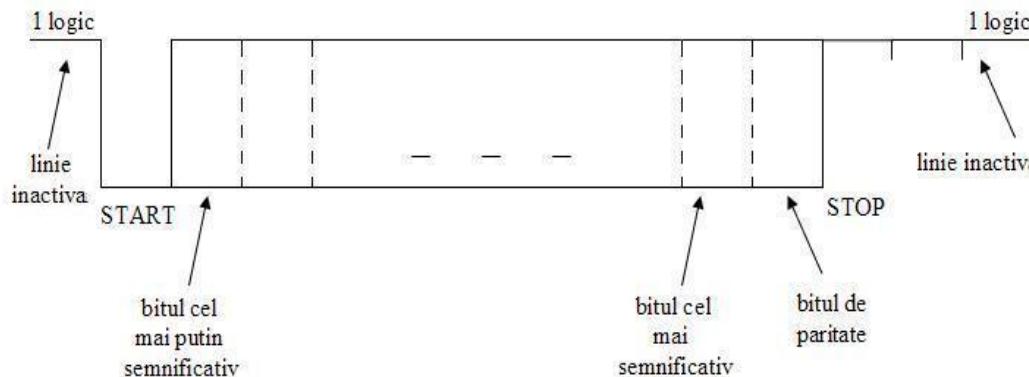
- Cea mai veche dar și cea mai răspândită: RS232; se folosește la transmiterea la distanță a datelor;
  - În mediu industrial: RS485;
  - În domeniile audio, video: I<sup>2</sup>C;
  - În industria automobilelor: CAN, Flexray etc.
- 
- Caracteristici de bază ale interfeței RS232:
  - Nivelele de tensiune:
    - Nu sunt TTL ci EIA și au următoarele valori:
      - - 25V ÷ - 3V pentru "1" logic și
      - + 3V ÷ + 25V pentru "0" logic;
    - Sunt necesare circuite de conversie TTL → EIA și EIA → TTL: 232;
  - Direcția de deplasare a datelor:
    - Simplex (unidirectionale), semi-duplex sau half duplex (bidirectionale dar la momente de timp diferite) și duplex sau full duplex (bidirectionale ce pot avea loc simultan);

# Proiectarea cu Microprocesoare

- Viteza:
  - Mică;
  - Există două unități de măsură a vitezei transferului: bit pe secundă (bps) și baud; dacă transferul este digital atunci  $1 \text{ bps} = 1 \text{ baud}$ ;
  - Treptele de viteză mai utilizate sunt: 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200 mai rar 230400, 460800 bps.
- Transferul serial de tip RS232 este util atunci când există distanțe mari (peste 3 m) între echipamentele care comunică.
- Există două motive care susțin această recomandare: costul și rezistența la perturbații:
  - Costul este determinat de numărul firelor din cablul care leagă cele 2 echipamente: dacă acest număr este mai mic, costul va fi mai redus;
  - Transferul serial de tip RS232 are o rezistență mai mare la perturbații decât cel paralel din două motive:
    - posibilitatea de perturbare a liniilor scade dacă numărul acestora este mai mic și
    - distanța dintre nivelele de tensiune corespunzătoare celor 2 nivele logice este mai mare decât la transferul paralel.

# Proiectarea cu Microprocesoare

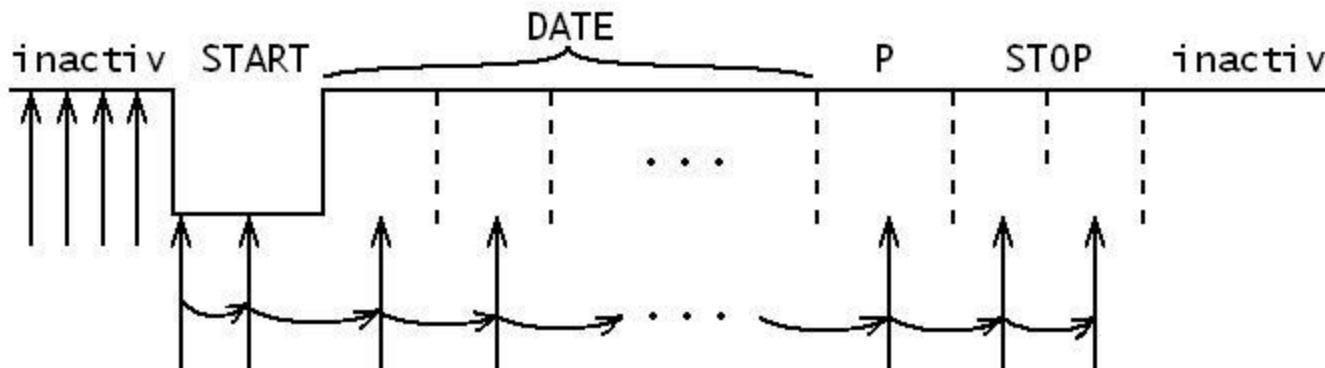
- Sincronizarea între emițător și receptor:
  - Transfer sincron;
  - Transfer asincron: transfer pe caracter, un caracter poate fi transmis în orice moment;
  - Forma de undă a unui caracter:



- Caracteristici:
  - Sincronizare la nivel de caracter prin bitul START;
  - 5 – 8 biți de date;
  - Opțional bit de paritate;
  - 1, 1.5, 2 biți de STOP.

# Proiectarea cu Microprocesoare

- ❑ Rejecția impulsurilor de START false și citirea informației la receptor:



- ❑ Avantaje:
  - Transmițătorul poate transmite un caracter oricând dorește;
  - Simplu de implementat.
- ❑ Dezavantaje:
  - Se transferă și informație inutilă (de cadrare): minim 20%;
  - Rată de transfer mică.

# Proiectarea cu Microprocesoare

- ❑ Transfer sincron: transfer pe bloc, transferul se face la nivel de bloc sau de mesaj.
- ❑ Avantaj:
  - Asigură viteză mai mare datorită:
    - ❑ Existenței tactului;
    - ❑ Elimină informația de cadrare la nivel de caracter.
- ❑ Dezavantaje:
  - Sunt necesare caractere de sincronizare deci dacă blocul este scurt, sub 8 caractere, modul sincron nu mai este eficient;
  - Linia suplimentară de tact.
- Codificarea informației alfanumerice: prin coduri:
  - ❑ BAUDOT: folosea doar 5 biți pentru codificarea unui caracter, ceea ce a limitat la 32 numărul de caractere ce puteau fi transferate;
  - ❑ EBCDIC;
  - ❑ ASCII (“American Standard Code for Information Interchange”): 96 de octeți corespund la litere, numere sau caractere speciale, iar 32 de octeți corespund la caractere de comandă.

# Proiectarea cu Microprocesoare

- Controlul fluxului: cum află transmițătorul că receptorul nu mai poate prelua caractere? - din diferite motive, de exemplu: tamponul său este plin, este deconectat de la linie etc., ca urmare nu mai are rost să le trimită fiindcă, în caz contrar, acestea se pierd.
  - Există trei soluții:
    - fără control;
    - cu control software ( “software handshaking” ) și
    - cu control hardware ( “ hardware handshaking” ).
  - Prima soluție: emițătorul și receptorul lucrează la aceeași viteză, mică;
  - A doua soluție: utilizarea a două caractere de control: XON având codul DC1 (11h) și XOFF având codul DC3 (13h); receptorul trimite XON dacă dorește transfer și XOFF dacă nu dorește transfer.
  - A treia soluție: realizează un dialog între transmițător și receptor prin intermediul a două semnale. Atunci când transmițătorul dorește să transfere ceva, activează linia RTS. Dacă receptorul poate primi informația, va răspunde prin activarea liniei CTS iar când nu mai poate primi informație va dezactiva linia CTS. Această soluție este mai rapidă decât cea dinainte dar cere un cablu cu 2 fire în plus.

# Proiectarea cu Microprocesoare

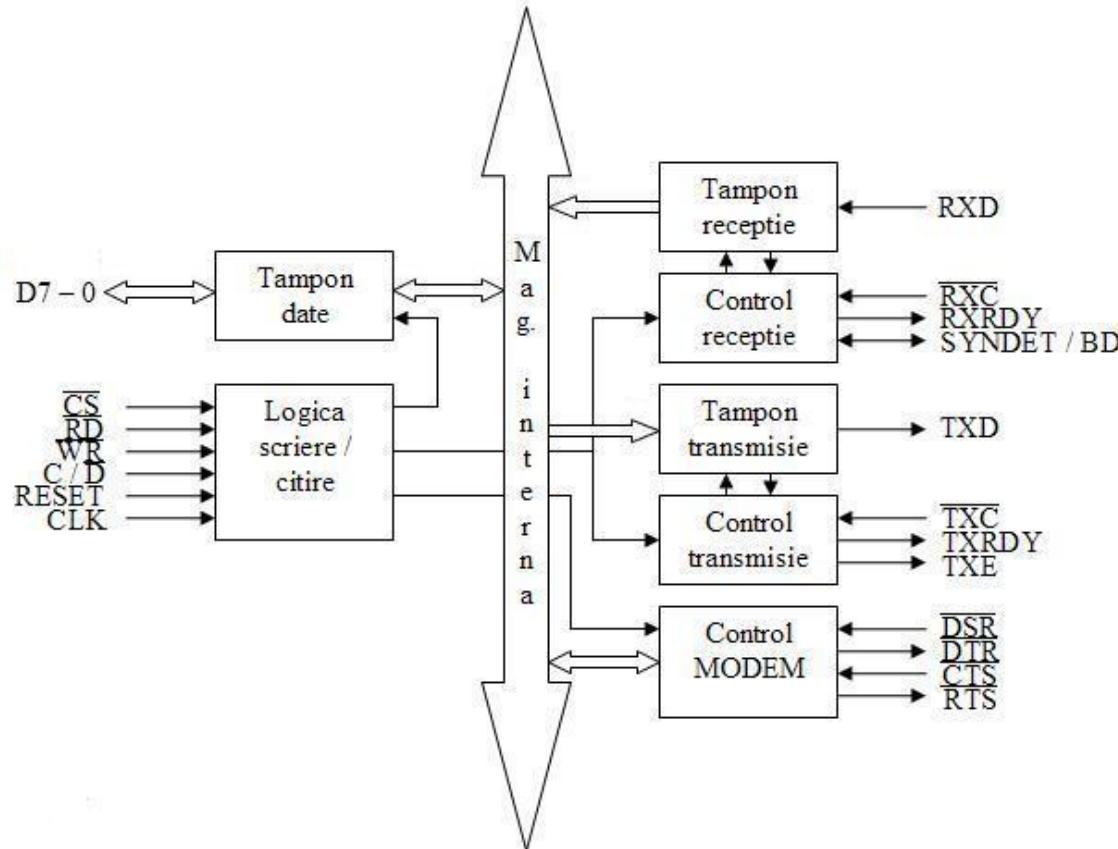
- Erori:
  - de cadrare (“Frame Error”);
  - de paritate (“Parity Error”);
  - de ritm (“Overrun Error”) și
  - de oprire (“Break”).
  - “Break”: o situație deosebită creată în mod voit de transmițător; în mod normal, dacă linia este inactivă, ea este la 1 logic; condiția de Break este forțată de transmițător care trece linia la 0 logic pentru o perioadă relativ lungă, 0,25 – 0,5 sec. și este sesizată de receptor;
  - În cazul tuturor erorilor, circuitul specializat nu oprește transferul ci le anunță doar prin activarea unor ranguri din octetul sau octeții săi de stare; este sarcina software-ului să trateze aceste situații, de exemplu cerând retransmisia octețiilor.
- Standardizare:
  - Electronic Industries Association (EIA): standardul RS232, ...
  - International Telegraph and Telephone Consultative Communitree (CCIT) care și-a schimbat denumirea în International Telecommunications Union (ITU): standardul V24, ...

# Proiectarea cu Microprocesoare

- Circuitul specializat programabil 8251
  - Specializat pentru transferurile seriale;
  - Face parte din categoria circuitelor de tip USART (“Universal Synchronous Asynchronous Receiver Transmitter”); moduri de lucru:
    - Sincron;
    - Asincron;
  - Poate să primească un octet în paralel de la unitatea centrală, să – l serializeze și să – l transmită la un echipament serial;
  - Poate să preia de pe linie, de la un echipament periferic serial, un octet, să – l asambleze și să – l predea, în paralel, unității centrale;
  - Circuitul comunică unității centrale când are un caracter gata pentru ea sau când a terminat de transmis un octet și poate prelua altul; poate comunica:
    - Prin program;
    - Prin Întreruperi;
  - Transmisie și receptie cu dublu tampon.

# Proiectarea cu Microprocesoare

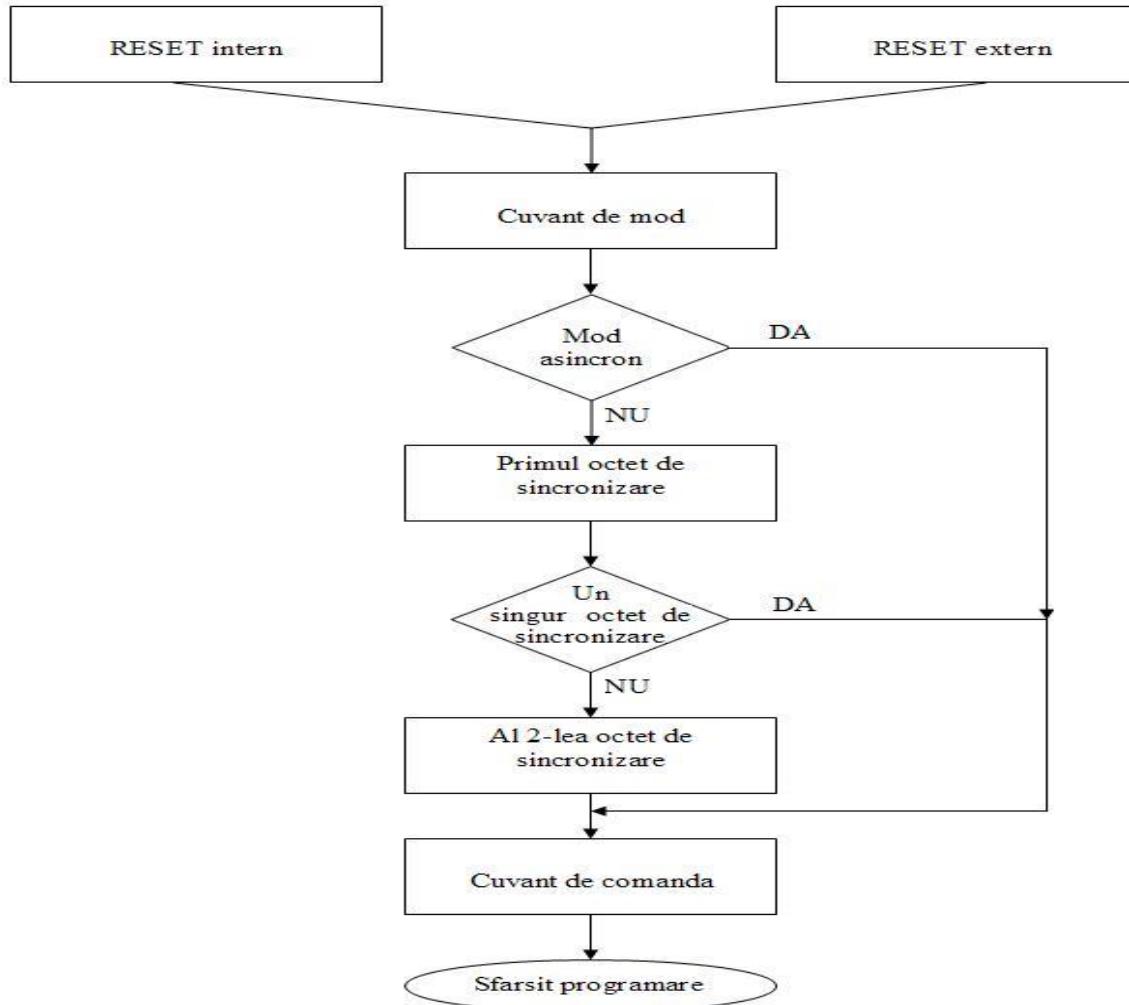
## ■ Structura internă a circuitului:



- C//D este intrarea prin care se alege unul din cele 2 porturi interne ale circuitului: portul de comandă/stări, ptr. C = 1 și portul de date ptr. C = 0; aici se leagă o linie din magistrala de adrese (de obicei A1 sau A2);

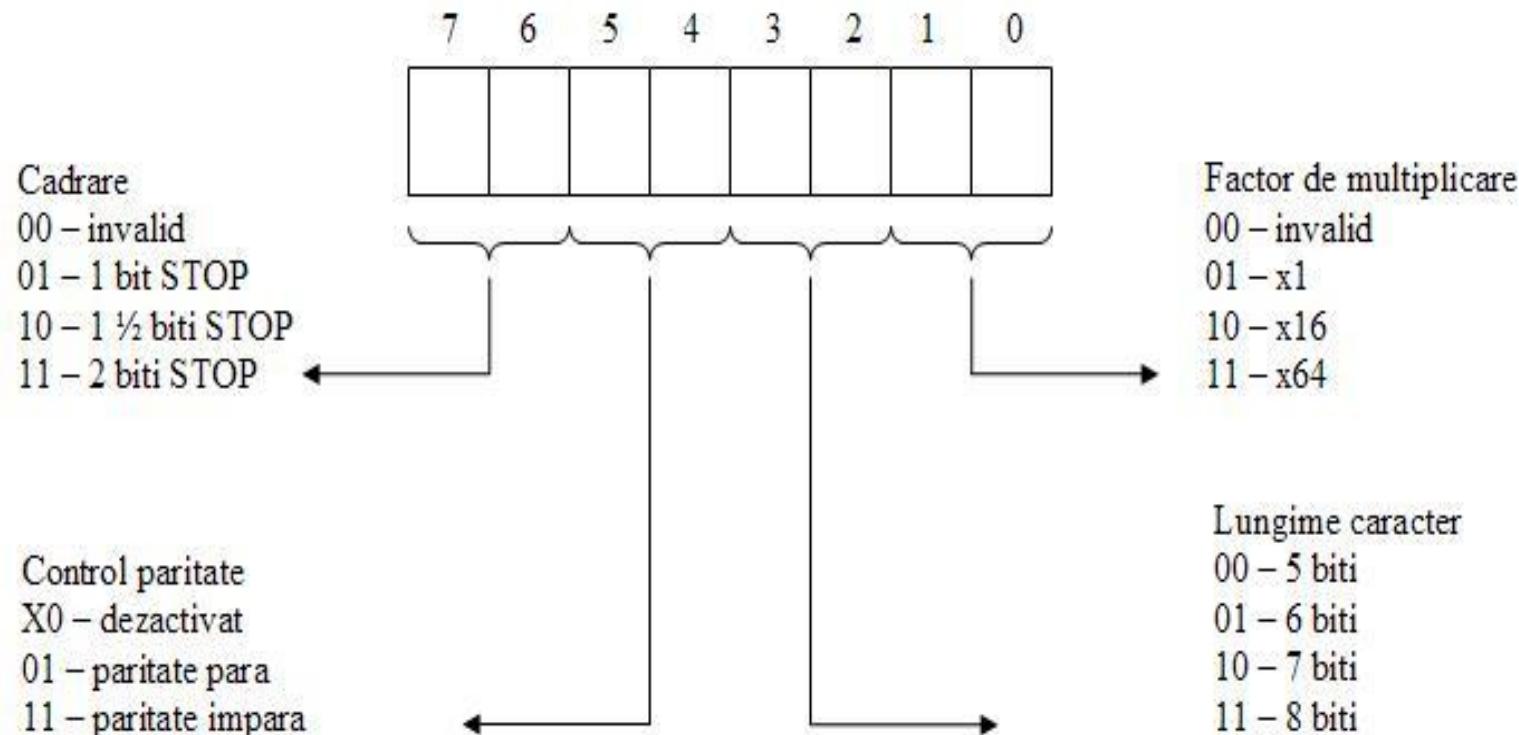
# Proiectarea cu Microprocesoare

## ■ Programarea circuitului:



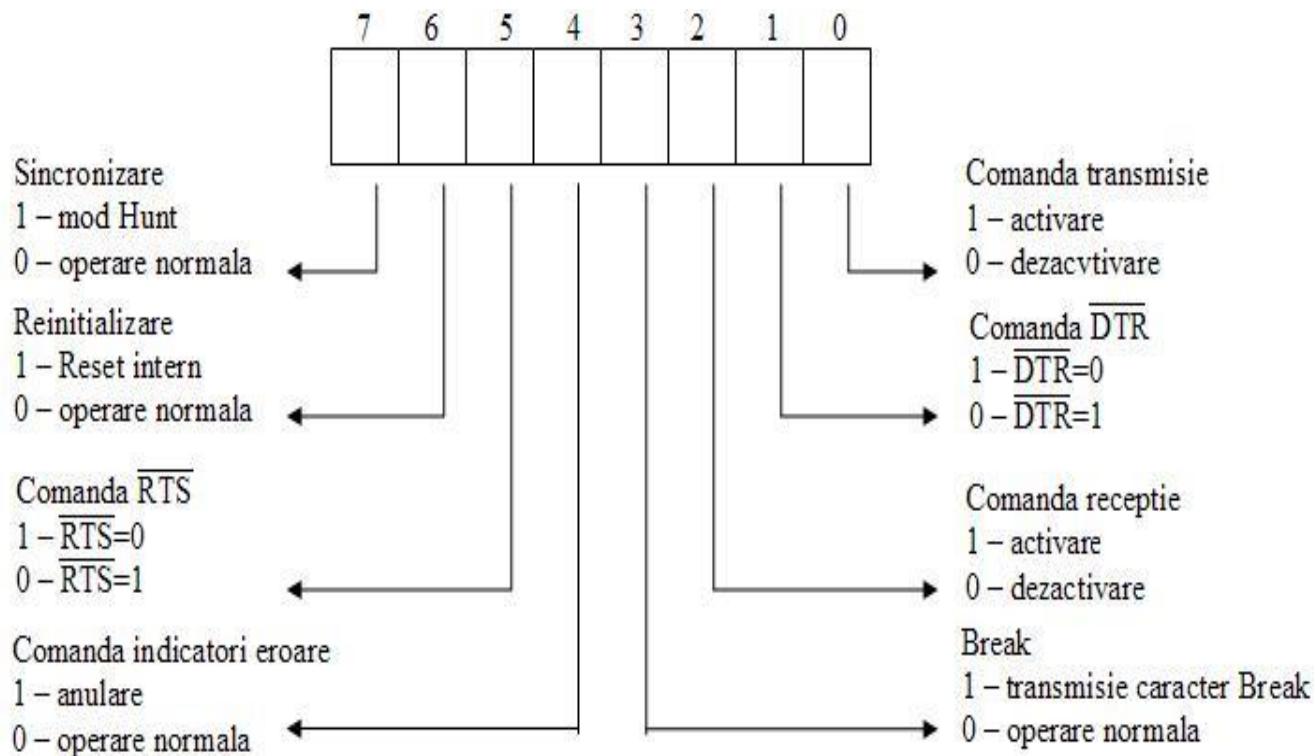
# Proiectarea cu Microprocesoare

- Structura cuvântului de mod pentru mod asincron:



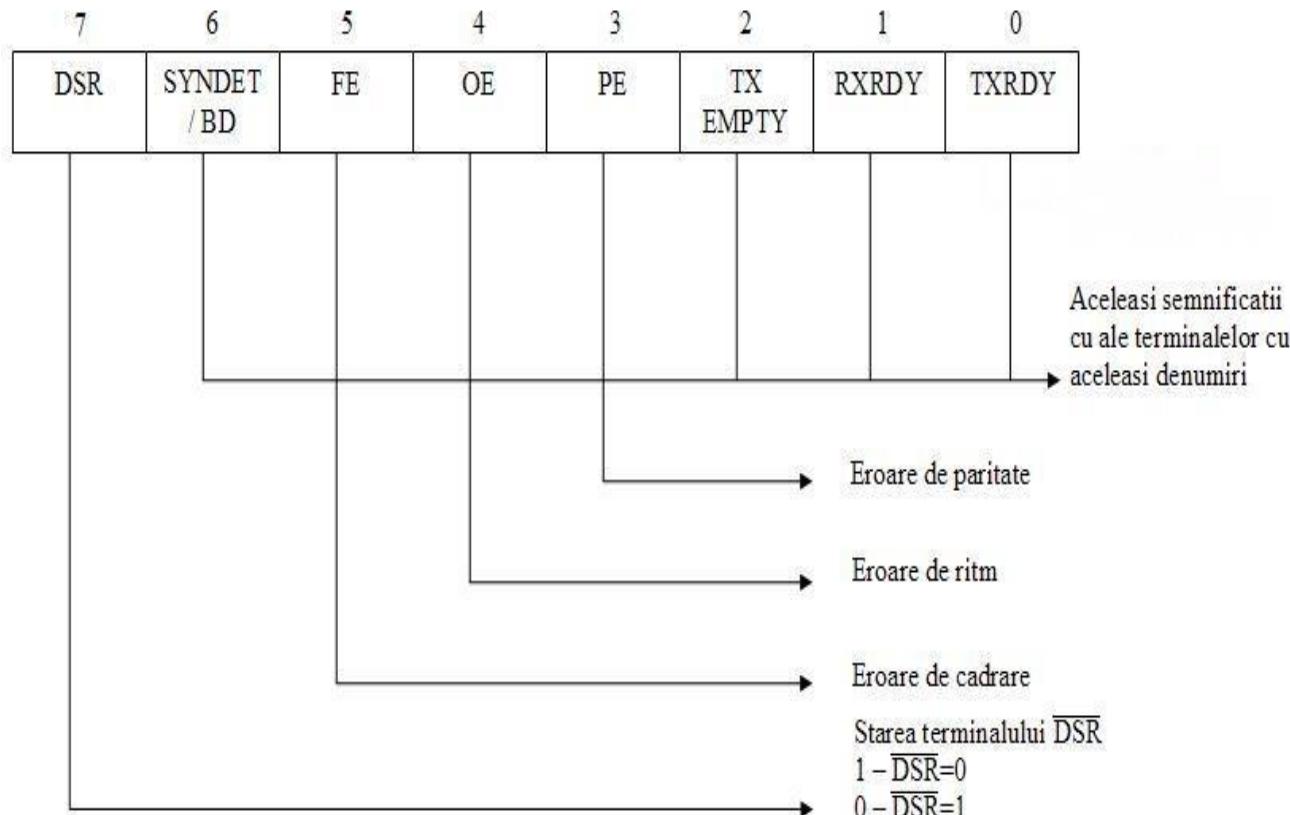
# Proiectarea cu Microprocesoare

## ■ Structura cuvântului de comandă:



# Proiectarea cu Microprocesoare

## □ Structura cuvântului de stare:



- Prin RxRDY activ, 8251 anunță procesorul că a terminat de preluat un caracter de pe linia serială, l-a asamblat și poate să i-l predea;
- Prin TxRDY activ, 8251 anunță procesorul că a terminat de transmis pe linie un caracter, l-a serializat și îi cere altul;

# Proiectarea cu Microprocesoare

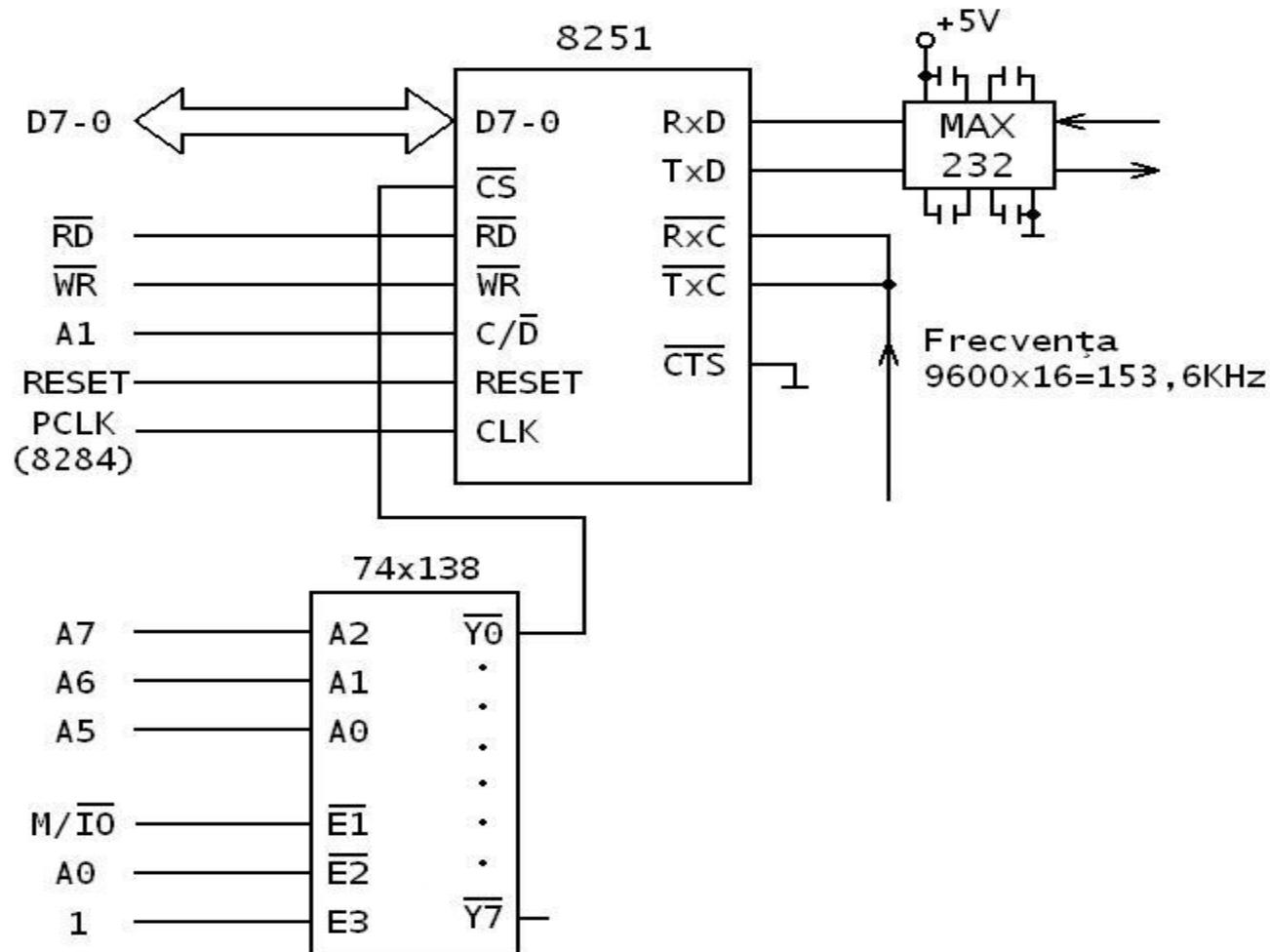
## ■ Aplicații:

Să se proiecteze o interfață serială cu circuitul 8251, conectată la o UC cu 8086, transferul făcându – se prin program. Datele inițiale ale transferului sunt:

- 8 biți de date,
- Fără paritate,
- 2 biți de STOP,
- Factor de multiplicare 16,
- Rata de transfer: 9600 bps.
- Se vor prezenta:
  - Schema interfeței,
  - Rutina de programare a circuitului specializat,
  - Rutinele de emisie și receptie.

# Proiectarea cu Microprocesoare

- Schema interfeței:



# Proiectarea cu Microprocesoare

## □ Adresele de port:

- Adresa de port este plasată de procesor pe liniile A7-A0 (dacă adresa este pe 8 biți);
- În exemplul considerat:

A7	A6	A5	A4	A3	A2	A1	A0
0	0	0	x	x	x	0	0

pentru date și

A7	A6	A5	A4	A3	A2	A1	A0
0	0	0	x	x	x	1	0

pentru comenzi/stări;

- 00H pentru date și 02H pentru comenzi/ stări; se observă că adresele nu sunt unice, mai există 7 variante pentru fiecare;

# Proiectarea cu Microprocesoare

- ❑ Rutina de programare:

MOV AL,0CEH ; cuvânt de mod

OUT 02H,AL

MOV AL,15H ; cuvânt de comandă

OUT 02H,AL

- ❑ Rutina de transmisie caracter:

TR: IN AL,02H ; citire și testare rang TxRDY din cuvântul de stare

RCR AL,1

JNC TR

MOV AL,CL ; se preia data din registrul CL

OUT 00H,AL

RET

# Proiectarea cu Microprocesoare

- ❑ Rutina de recepție caracter:

```
REC:IN      AL,02H ; citire și testare rang RxRDY din câvântul de stare  
            RCR     AL,2  
            JNC     REC  
            IN      AL,00H ; se preia data de la 8251  
            MOV     CL,AL ; se depune data în registrul CL  
            RET
```

- ❑ Soluția prezentată realizează transferul serial prin program.
- ❑ Dezavantaj: utilizarea ineficientă a timpului microprocesorului, întrucât acesta va trebui să aștepte mult timp până când circuitul 8251 poate realiza un transfer.
- ❑ Alternativa o constituie transferul serial prin intreruperi, în care microprocesorul execută un program și va fi întrerupt de circuitul 8251 doar când acesta poate realiza un transfer.
- ❑ Se vor utiliza terminalele RxRDY și TxRDY.