

15 Familia Intel

15.1 Interfața paralelă programabilă Intel 8255

Circuit integrat realizat în tehnologie NMOS, 40 de terminale, cu o singură tensiune de alimentare, +5 V. Toate intrările și ieșirile sunt compatibile TTL.

Conține 3 porturi de intrare / ieșire programabile, în total 24 linii I/O. Fiecare port poate fi programat ca port de intrare sau ca port de ieșire, în mod independent. Deși face parte din familia Intel, poate fi utilizat în orice sistem cu microprocesor de cel puțin 8 biți. Se utilizează ca interfață paralelă între un sistem cu microprocesor și diverse periferice inteligente sau echipamente de execuție.

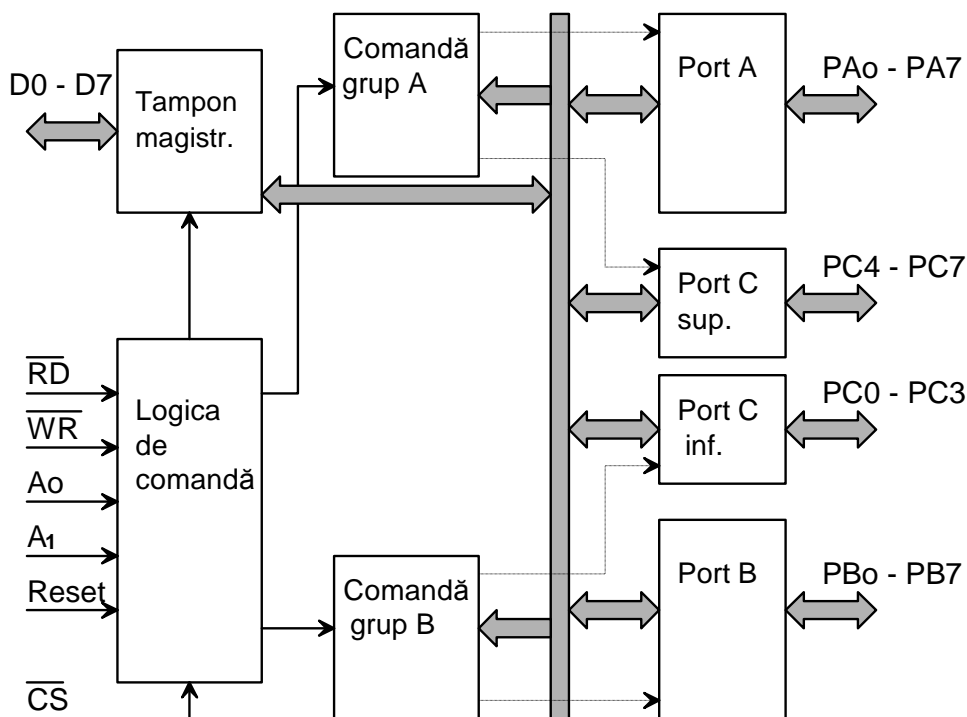


Fig.1 Intel 8255 - structura internă

Structura internă conține trei blocuri funcționale:

- ◆ Interfața cu magistralele sistemului de calcul.
- ◆ Magistrala internă de date și două blocuri de comandă grup.
- ◆ Interfața cu perifericele;

15.1.1. Interfața cu magistralele

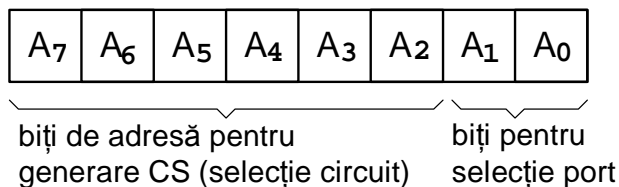
Conține un registru tampon de date și un bloc logic pentru selecție, adresare și operații de citire/scriere.

Selecția porturilor se realizează cu liniile A_0 , A_1 ale magistralei de adrese și cu semnalul CS (*Chip Select*) generat de sistemul de calcul pe baza liniilor de adresă, altele decât A_0 , A_1 . Semnalul RESET, activ în "1", produce inițializarea circuitului, adică anularea informațiilor de programare anterioare. Semnalele RD și WR sunt generate de microprocesor, fiind comanda de citire, respectiv scriere port.

Tabelul 1. Selecția porturilor și a registrului de comandă

CS	A_1	A_0	Obiectul selectat
0	0	0	Port A (I/O)
0	0	1	Port B (I/O)
0	1	0	Port C (I/O)
0	1	1	Registrul de comandă
1	X	X	Circuit 8255 neselectat

Formarea adresei unui port :
(de 8 biți)



2. Interfața cu perifericele. Conține 24 linii de intrare / ieșire, logica internă de comandă și registrele temporare de date pentru ieșiri.

Caracteristicile și funcțiile liniilor de interfață sunt determinate de modul de operare stabilit prin programare.

Sunt puse la dispoziția utilizatorului 3 moduri de lucru, care acoperă o mare varietate de aplicații.

Modul 0. Este modul de bază de intrare / ieșire, care oferă mai multe variante de utilizare:

- ♦ două porturi de 8 biți (PA și PB) care se programează independent ca porturi de intrare sau de ieșire;
- ♦ două porturi de 4 biți (PC4-7, PC0-3) care se programează independent, cu posibilitatea poziționării individuale în "0" sau "1" a fiecărei linii de ieșire;

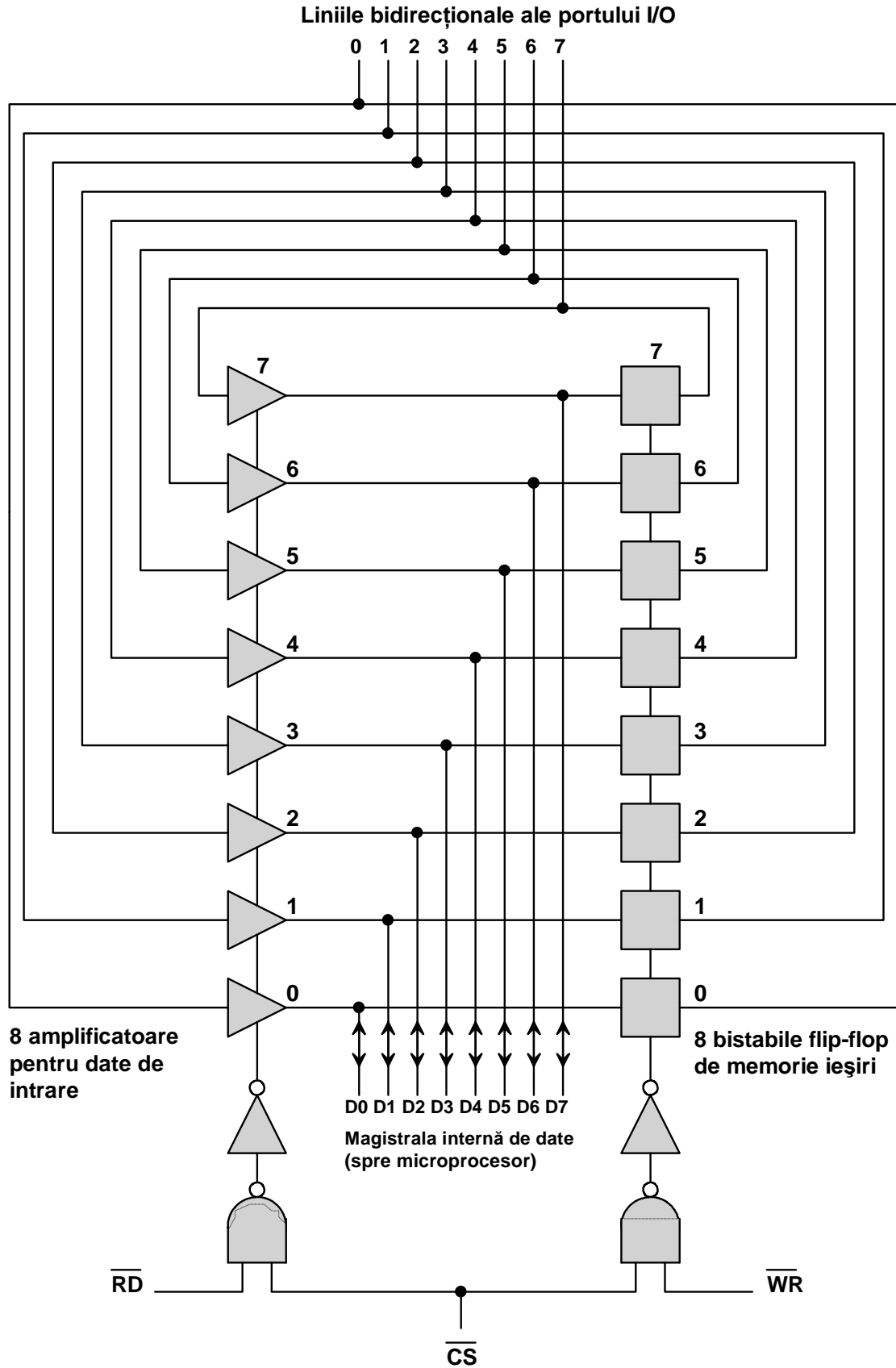


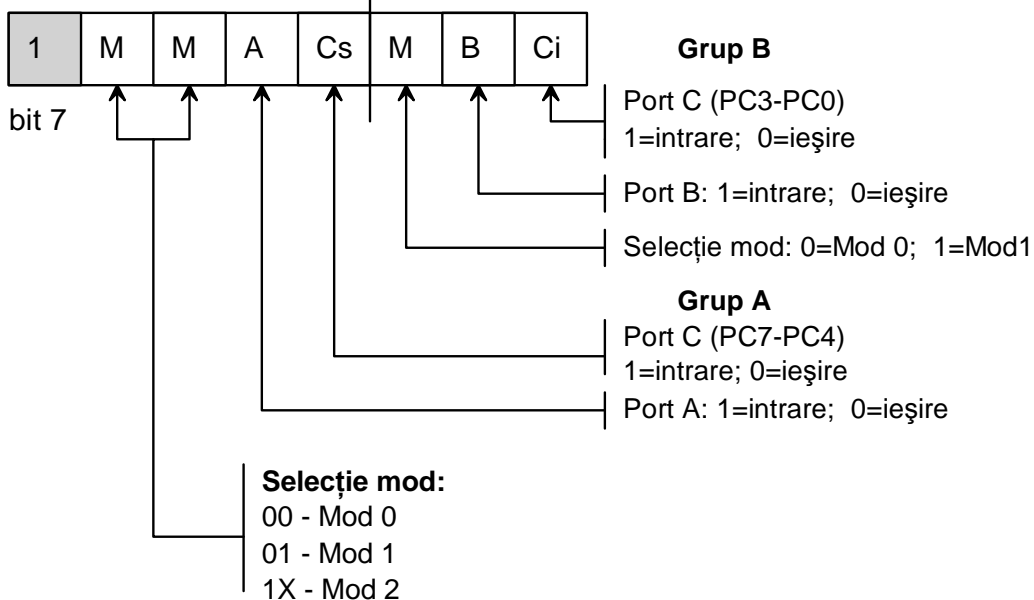
Fig. 2 Structura internă a unui port de intrare/ ieșire de 8 biți (PA, PB, PC)

-
- ♦ porturile de ieșire dispun de registre de memorare temporară a datelor;
 - ♦ porturile de intrare sunt fără memorie; transferă datele existente în momentul comenzii de citire.

Modul 1 și 2. PA și PB sunt utilizate ca porturi de intrare / ieșire iar liniile portului PC au diferite funcții ce deservește liniile de date ale porturilor PA, PB (validare intrare sau ieșire, cereri de întrerupere).

Stabilirea modului de lucru, funcția fiecărui port și poziționarea liniilor în "0" sau "1" se realizează prin încărcarea unor cuvinte de comandă sub controlul programului. Cuvintele de comandă sunt încărcate și memorate într-un registru special, registrul de comandă, care are adresă proprie (tabelul 1).

Structura cuvântului de comandă pentru definirea modului de operare



Structura cuvântului de comandă pentru poziționarea în 1 / 0 a biților din PC

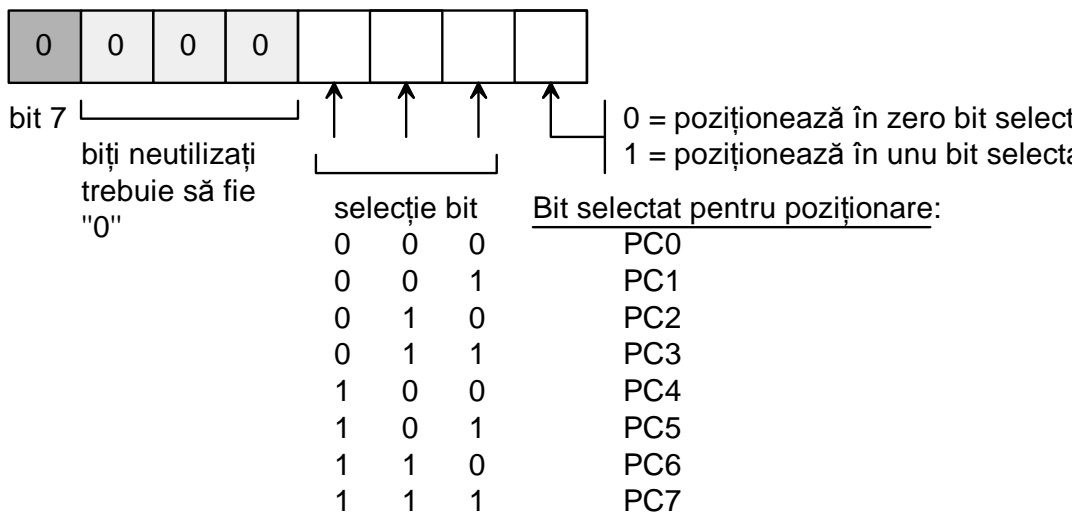


Fig.3 Structura cuvintelor de comandă (programare)

Cuvintele de comandă (fig.3) sunt de două categorii:

- pentru definirea modului de operare (bit 7 = 1); un singur cuvânt de comandă de 8 biți programează modul de lucru și tipul porturilor PA, PB, PC.

- pentru poziționarea individuală a liniilor de ieșire ale portului PC (bit 7 = 0).

Adresele porturilor se stabilesc la proiectarea sistemului de calcul; acestea pot fi de 8 sau 16 biți, în funcție de tipul microprocesorului.

Exemplu: Dacă PA are adresa: PA: 1 1 1 1 1 0 0 0 = F8 H,
atunci celelalte adrese trebuie să fie: PB: 1 1 1 1 1 0 0 1 = F9 H

PC: 1 1 1 1 1 0 1 0 = FA H

adresa registrului de comandă: 1 1 1 1 1 0 1 1 = FB H.

La programare, cuvântul de comandă se transmite la adresa FB iar octeții de date se scriu sau se citesc de la adresele corespunzătoare fiecărui port. Deși se utilizează aceeași adresă atât pentru programarea modului de operare cât și pentru poziționarea individuală a biților portului PC, cuvintele de comandă se memorează în registre diferite, în funcție de valoarea bitului b7.

Pentru programarea porturilor PA, PB ca porturi de intrare și PC ca port de ieșire, toate în modul M0, rezultă cuvântul de programare:

1 0 0 1 0 0 1 0 = 92 H.

În limbaj de asamblare Intel 8086, secvența de program este:

```
MOV AL, 92H ; se încarcă octetul de programare în acumulator
OUT 0FBH, AL ; se transferă octetul la 8255 în registrul de
; comandă.
```

Următoarea secvență de program (Intel 8086) citește porturile PA și PB, pune datele în registrele DL respectiv EI și transferă octetul FF la portul PC:

```
IN AL, 0F8H ; datele de la PA se transferă în acc. (citire PA)
MOV DL, AL ; se salvează datele în DL
IN AL, 0F9H ; datele de la PB se transferă în acc. (citire PB)
MOV EI, AL ; se salvează datele în EI
MOV AL, 0FFH ; se încarcă octetul FF în acumulator
OUT FAH, AL ; se transmite FF la portul PC (scriere în port PC)
```

Secvența următoare poziționează biții b5=1 și b6=0 ai portului PC:

```
MVI AL, 0BH ; cuvântul de poziționare pentru b5 = 1 în acc.
OUT FBH, AL ; se transferă cuvântul de poziționare în registrul de
; comandă, care are adresa FB
MVI AL, 0CH ; cuvântul de poziționare pentru b6 = 0 în acc.
OUT FBH, AL ; se transferă cuvântul de poziționare în
; registrul de comandă, care are adresa FB
```

După această secvență, ieșirea PC5 trece în "1" și ieșirea PC6 trece în "0"; valorile logice se mențin până la re-poziționare sau schimbarea modului de operare.

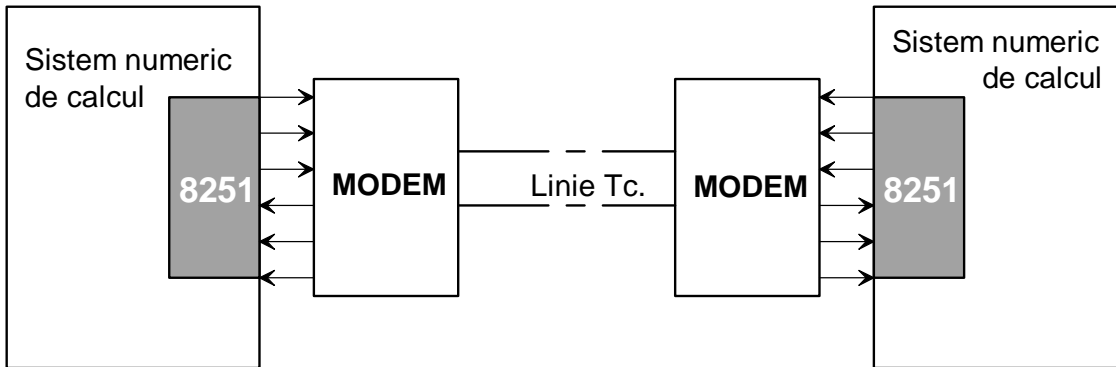


Fig.5 Schema unui sistem de comunicații de date

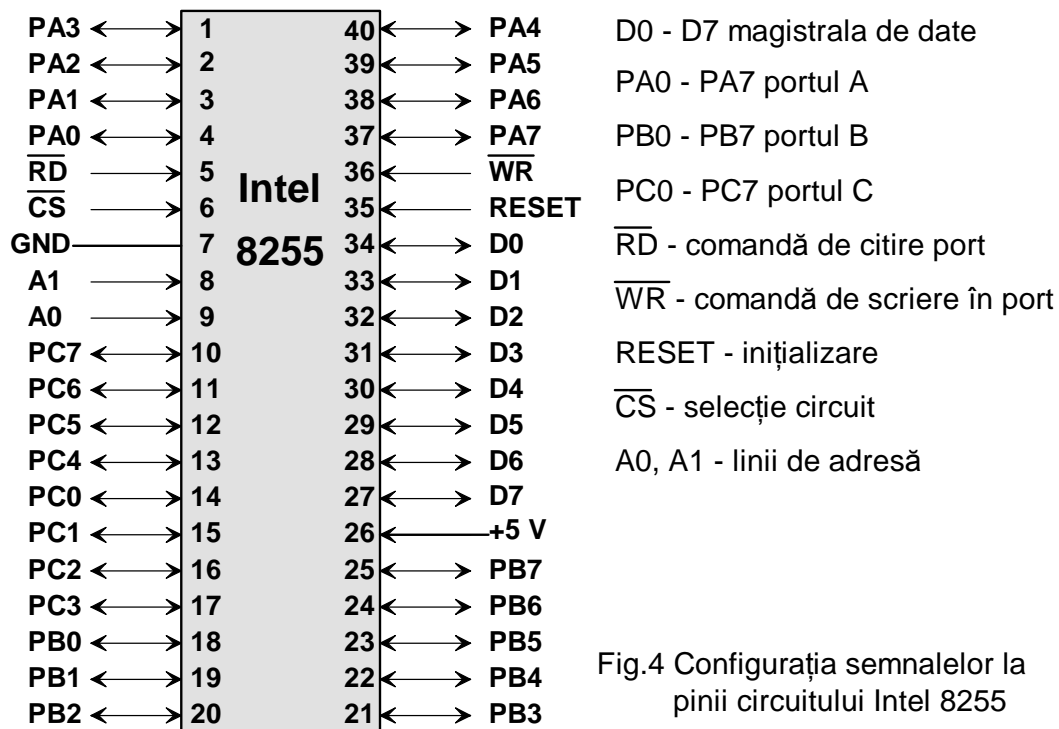


Fig.4 Configurația semnalelor la pini circuitului Intel 8255

15.2 Interfața serială programabilă Intel 8251 (USART)

Cunoscut sub numele de USART (*Universal Synchronous - Asynchronous Receiver Transmitter*), Intel 8251 este un circuit realizat în tehnologie NMOS cu 28 de terminale, destinat comunicației seriale de date între două sisteme numerice (port serial).

Pentru comunicații seriale la distanțe mari (zeci, sute de km), circuitul 8251 dispune de intrări /ieșiri standard pentru conectare cu echipamente MODEM specifice liniilor de telecomunicații (în cablu, pe fibre optice, radio).

Intel 8251 poate funcționa atât în modul sincron cât și în modul asincron; modul de operare, viteza de transmisie/recepție și formatul datelor se stabilesc prin cuvinte de comandă (programare). Se poate conecta la un sistem numeric cu microprocesor (Intel 8080 sau orice microprocesor de cel puțin 8 biți), prin magistrala de date de 8 biți și 6 semnale de intrare prin care sistemul numeric controlează toate funcțiile sale.

Având în structură un transmițător și un receptor cu funcționare independentă, circuitul 8251 poate realiza simultan funcțiile de recepție și transmisie.

Dispune de semnale de ieșire ce pot fi utilizate drept cereri de întrerupere către unitatea centrală, activate la recepția completă a unui caracter (pentru funcția de receptor) sau la terminarea transmisiei unui caracter (pentru funcția de transmițător).

Principalele funcții ale circuitului 8251 sunt:

- ♦ Acceptă și transmite date, în paralel, sub formă de octeți, prin magistrala de date conectată la magistrala sistemului cu microprocesor;
- ♦ Realizează conversia datelor în format serial de 5, 6, 7 sau 8 biți pe caracter și le transmite la ieșirea serială de date, cu viteza programată (în biți / secundă);
- ♦ Recepționează date în formatul serial prestabilit, la intrarea de date, realizează asamblarea lor în octeți pe care îi transmite paralel către microprocesor;
- ♦ Generează semnale pentru microprocesor cu semnificația "recepție gata" sau "transmisie gata" care pot fi utilizate ca întreruperi sau pot fi testate periodic de către microprocesor pentru stabilirea momentelor în care acesta trebuie să preia datele recepționate sau să încarce date pentru transmisie;
- ♦ Interfața serială dispune de un registru de stare care poate fi citit în orice moment de către microprocesor;
- ♦ Se poate conecta cu un echipament MODEM prin patru semnale standard de protocol și două semnale de date.

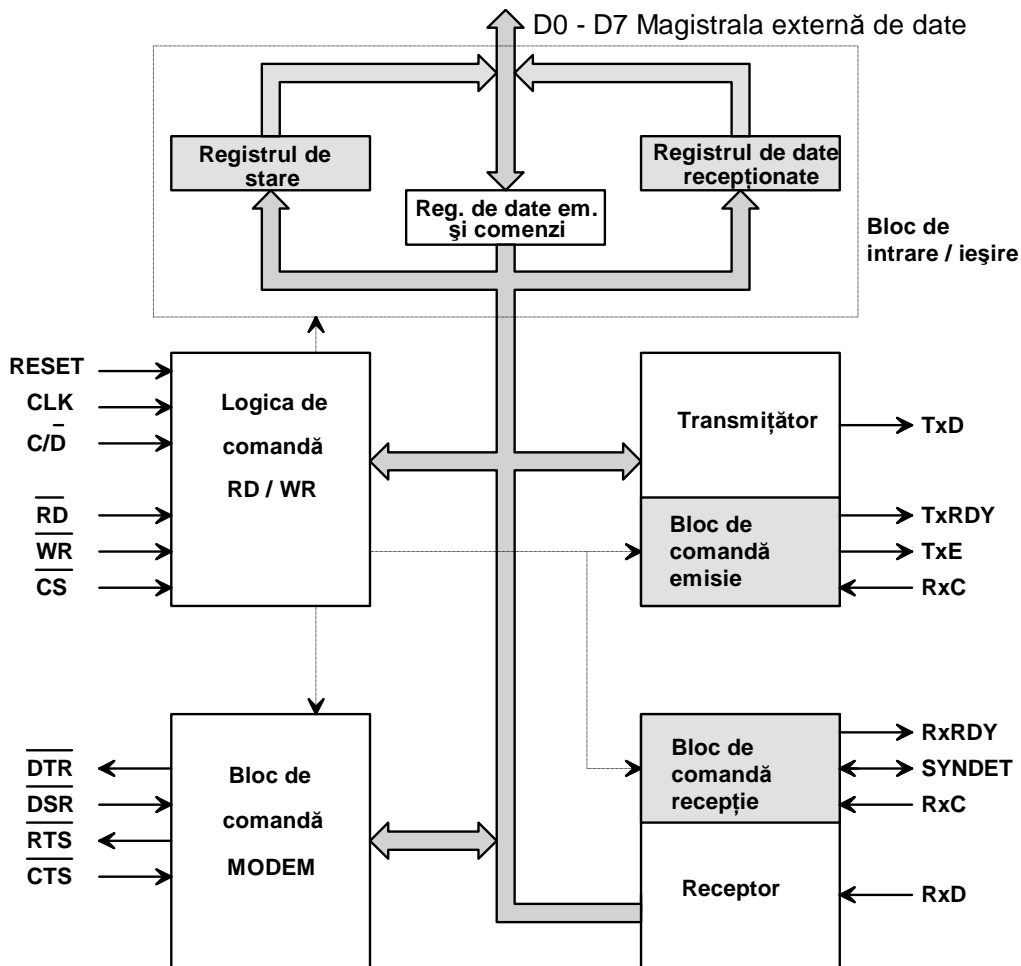


Fig. 5 Structura internă a circuitului Intel 8251

Structura internă conține 5 blocuri funcționale: blocul de intrare/ieșire, blocul de comandă, transmițătorul, receptorul și blocul de interfață cu echipamentul MODEM.

Toate blocurile funcționale sunt conectate la magistrala internă de date de 8 biți.

Blocul de intrare/ieșire. Conectează circuitul la magistrala de date a sistemului de calcul. Prin această magistrală se transmit cuvintele de comandă (programare), datele de emisie, datele recepționate și cuvântul de stare, toate pe 8 biți.

Registru de stare este permanent la dispoziția microprocesorului și conține toate informațiile cu privire la funcționarea interfeței seriale.

Registru datelor de emisie și de comenzi, memorează temporar octetul de date încărcat de pe magistrala externă și cuvântul de comandă în etapa de programare. Octetul de date urmează a fi transferat prin magistrala internă la emițător, de îndată ce acesta eliberează registrul propriu de emisie.

Registrul datelor recepționate memorează temporar octetul de date furnizat de receptor până când microprocesorul îl transferă printr-o comandă de citire date, într-un registru intern. În acest interval, receptorul stochează datele ce sosesc în flux continuu într-un registru propriu.

Transmițătorul face conversia datelor din format paralel în format serial, adaugă biții de start, stop și eventual de paritate (conform programării) și transmite caracterul astfel format, bit cu bit, la ieșirea TxD (*Transmission Data*), cu viteza (biți / secundă) programată.

Viteza de transmisie este determinată de frecvența semnalului TxC (*Transmission Clock*) dar în modul asincron pot fi selectate și opțiunile 1:16 și 1:64 din frecvența de tact, când linia de comunicație nu permite o frecvență egală cu TxC. Transmițătorul conține un bloc de comandă a emisiei, care coordonează funcționarea conform programării și generează semnalele TxRDY (*Transmitter Ready*) și TxE (*Transmitter Empty*).

Receptorul primește date bit cu bit la intrarea RxD (*Reception Data*) în formatul serial, face conversia lor în format paralel pe 8 biți după eliminarea biților de start, stop, paritate și le transferă în registrul de date recepționate din blocul de intrare/ieșire.

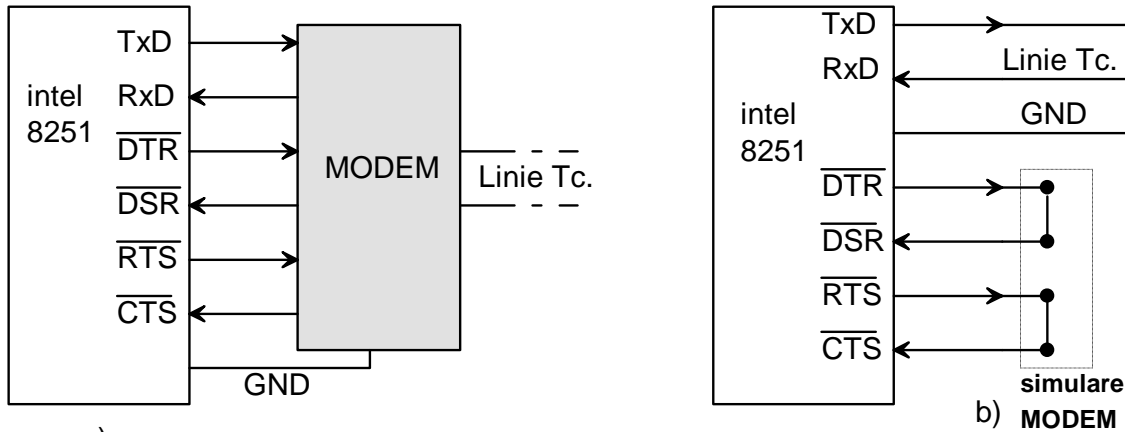
După acest transfer, activează semnalul RxRDY (*Reception Ready*), pentru a semnaliza că un caracter este pregătit pentru a fi citit de microprocesor. Semnalul de intrare RxC este semnalul de tact pentru recepție; el poate avea frecvență diferită față de TxC.

Semnalul SYNDET (*Synchrony Detection*) este utilizat în modul de lucru sincron. El este activat la detecția caracterelor de sincronizare sau în cazul în care se utilizează sincronizare externă.

Blocul de comandă MODEM lucrează pe baza a două perechi de semnale de protocol de tip dialog. Generează semnalul DTR (*Data Terminal Ready*), activ în zero, indicând că interfața 8251 este pregătită, la care modemul răspunde cu DSR (*Data Set Ready*), activ în zero, indicând că modemul este de asemenea pregătit.

Interfața 8251 generează semnalul RTS (*Request to Send*), activ în zero, cu semnificația "cerere în vederea transmisiei", la care modemul răspunde cu CTS (Clear to Send), activ în zero, anulare în vederea transmisiei. Ieșirea de date TxD se conectează ca intrare la modem iar ieșirea de date de la MODEM, se conectează la intrarea de date recepționate, RxD.

În cazul funcționării fără MODEM, acesta trebuie simulat prin conectarea directă a semnalelor de comandă pereche (fig. 6).



a) Fig. 6 Funcționarea cu MODEM (a) și fără MODEM (b).

Logica de comandă decodifică semnalele de comandă (de la procesor) și pe cele de selecție și realizează funcțiile corespunzătoare pe baza controlului pe care îl exercită asupra tuturor unităților interne.

Tabel 2. Semnalele de comandă și funcțiile lor.

CS	C/D	RD	WR	Funcția
0	0	0	1	Unitatea centrală citește data de la 8251
0	1	0	1	Unitatea centrală citește starea lui 8251
0	0	1	0	Unitatea centrală scrie data în 8251
0	1	1	0	Unitatea centrală scrie o comandă în 8251
1	x	x	x	Interfață 8251 neoperațională (neselectată)

Interfața 8251 necesită două adrese de port I/O: o adresă pentru portul de date ($C/\overline{D} = 0$) și o adresă pentru portul de comenzi și stări ($C/\overline{D} = 1$).

Cuvântul de comandă și cuvântul de stare sunt memorate într-un registru de 16 biți și transferate pe rând, ca octeți. Cele două cuvinte au aceeași adresă dar cuvântul de comandă se scrie (WR activ) iar cuvântul de stare se citește (RD activ), așadar nu se face confuzie între ele deși au aceeași adresă de selecție.

15.2.1. Semnificația semnalelor.

D₀ - D₇ : magistrala de date a sistemului numeric;

RESET : intrare de inițializare;

CLK : intrare de tact (comun cu tact sistem);

C/D: selecție Comandă/Date (de regulă este linia A₀ a magistralei de adrese, "1" - comandă, "0"-date);

\overline{RD} : (*Read*) citire date sau stare de către unitatea centrală;

WR: (*Write*) scriere date sau cuvânt de programare de către unitatea centrală;

RxD : (*Reception Data*) intrare serială de date;

TxD : (*Transmission Data*) ieșire serială de date;

RxRDY : (*Receptor Ready*) caracter recepționat, ce poate fi citit de procesor;

TxRDY : (*Transmitter Ready*) transmițătorul disponibil pentru un nou caracter;

TxE : (*Transmitter Empty*) transmițătorul este vid (nu are date de transmis);

RxC : (*Reception Clock*) intrare de tact pentru recepție;

TxC : (*Transmission Clock*) intrare de tact pentru transmisie;

SYNDET : (*Synchrony Detection*) forțare sau detecție sincronă de date;

DTR : (*Data Terminal Ready*) terminal de date pregătit;

DSR : (*Data Set Ready*) echipament de date pregătit;

RTS : (*Request To Send*) cerere în vederea transmisiei;

CTS : (*Clear To Send*) anulare în vederea transmisiei.

Comunicația serială, cu echipamente locale sau plasate la distanță, legate prin modemi, poate fi realizată de către interfața 8251, fie sub formă *asincronă*, fie sub formă *sincronă*.

15.2.2. Transmisia asincronă

Sincronizarea între emițător și receptor se face la începutul fiecărui caracter, fără a se utiliza semnal special de sincronizare. Frecvența semnalului de tact la emițător poate să difere de cea de la receptor cu max. 5%, de unde și denumirea.

Fiecărui caracter (5 - 8 biți de date) i se adaugă informația de cadru, constând dintr-un bit de START, definit ca zero logic și unul sau doi biți de STOP (în funcție de opțiunea programatorului), definiți ca unu logic. Tranziția 1 → 0, de la bitul de STOP al unui caracter la bitul de START al caracterului următor, asigură sincronizarea între emițător și receptor, deoarece indică receptorului începutul unui nou caracter în fluxul serial de biți.

Dacă diferența frecvențelor de tact la emițător și receptor este în limite normale, sincronizarea se menține pe toată durata recepției unui caracter și datele recepționate vor fi corecte.

Bitul sau biții de STOP adăugați la sfârșitul fiecărui caracter asigură tranziția necesară sincronizării.

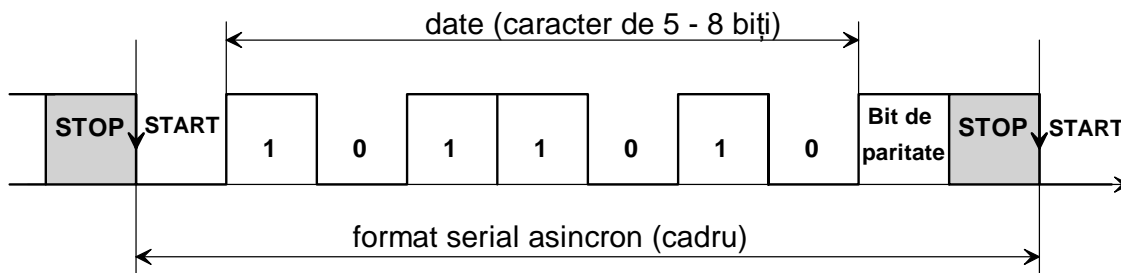


Fig.7 Formatul asincron pentru transmisia serială

Prin cuvântul de programare, se poate selecta opțiunea privind inserarea în fiecare cadru serial a unui bit de paritate (de control). Dacă pe durata transmisiei, numărul erorilor de bit este impar, bitul de paritate indică existența erorilor; dacă este afectat chiar bitul de paritate, efectul este același: interfața serială semnalizează către unitatea centrală recepționarea unui caracter eronat, prin bitul PE (*Parity Error*) din cuvântul de stare.

15.2.3. Transmisia sincronă

Sincronizarea emițătorului cu receptorul se face permanent, la fiecare bit, pe durata comunicației, fie utilizând informația de sincronizare furnizată de semnalul modulat, fie pe baza unei surse externe de semnal de tact, comun pentru emițător și receptor. Caracterele de 5 - 8 biți se transmit în șir continuu, de orice lungime, după ce se adaugă la începutul șirului unu sau două caractere speciale SYN care servesc receptorului pentru identificarea începerii transmisiei sincrone.

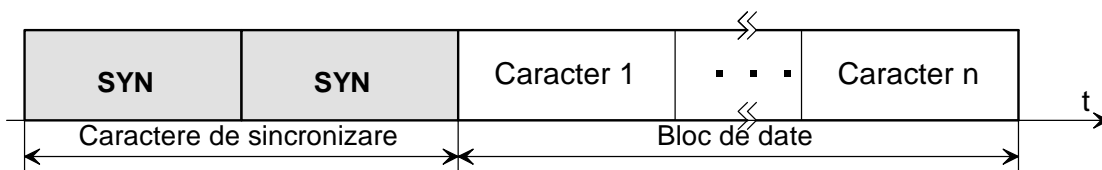


Fig. 8 Formatul sincron pentru transmisie serială

Un caracter poate fi de 5, 6, 7 sau 8 biți, la care se adaugă eventual un bit de paritate. Sincronizarea se realizează extern, prin echipamente hardware adecvate, sau intern prin detectarea caracterelor speciale SYN. Acest caracter este specificat prin software în cadrul procedurii de inițializare și programare. El este memorat atât la emițător cât și la receptor, care verifică bit cu bit caracterul de sincronizare recepționat și la recunoașterea sa validează începerea recepției datelor. Dacă unitatea

centrală nu furnizează la timp datele de emisie, interfața inserează automat caractere SYN pentru a nu pierde sincronizarea.

Comparând cele două modalități de transmisie, se constată că pentru mesaje ce dăpășesc 8 caractere, devine mai eficientă transmisia sincronă, ținând seama de numărul de biți suplimentari ce se adaugă în cele două cazuri. Transmisia sincronă însă, necesită semnal de sincronizare la fiecare bit, deci echipament suplimentar și ca urmare, este mai scumpă.

La transmisia sincronă, echipamentul MODEM furnizează semnalul de sincronizare către terminal și impune ca datele să-i fie livrate sincron cu acest semnal. MODEM - ul receptor dispune de un generator de tact cu aceeași frecvență cu cel de la emisie, a cărui fază este ajustată cu faza emițătorului.

În unele cazuri, când se urmărește creșterea vitezei de operare fără a schimba protocolul, se operează într-un mod hibrid cu format asincron și transmisie sincronă; această transmisie se numește *isosincronă*.

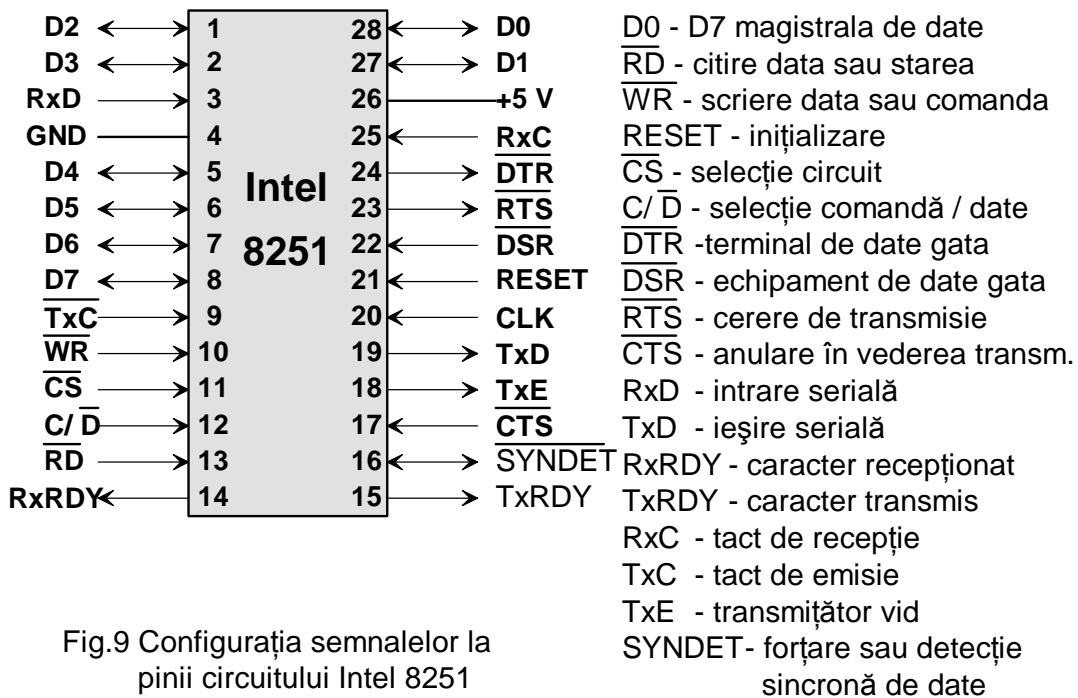


Fig.9 Configurația semnalelor la pini circuitului Intel 8251

15.2.4. Funcționare

La aplicarea tensiunii de alimentare interfața trece în starea "selecție mod" în care așteaptă cuvintele de comandă pentru stabilirea modului de operare.

Dacă este programat **modul asincron**, transmițătorul așteaptă date de transmis iar receptorul așteaptă date seriale pe intrarea RxD.

Receptorul, pregătit pentru recepție asincronă, urmărește tranziția $1 \rightarrow 0$ la intrarea RxD, la care declanșează un generator intern cu

perioada $T/2$ față de durata T a unui bit; bitul recepționat este testat la mijlocul duratei sale, pe tranziția $1 \rightarrow 0$ a tactului intern (fig.10).

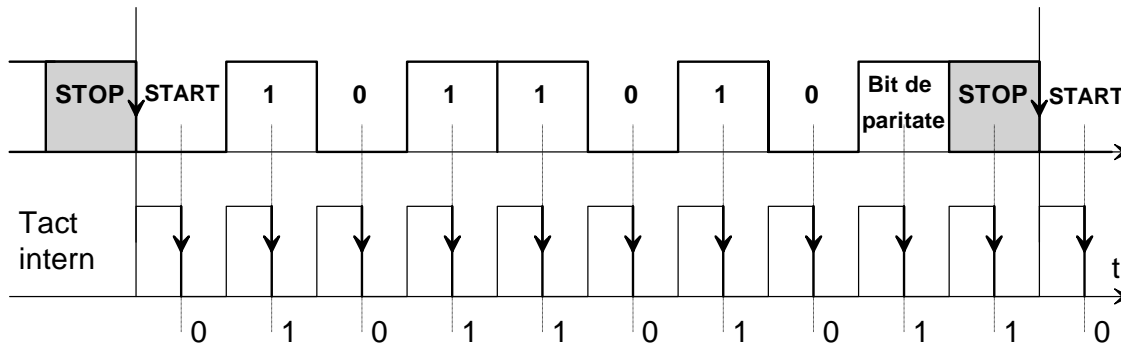


Fig.10 Recepționarea valorilor logice prin sondare la mijloc

Cuvântul recepționat este memorat temporar în registrul tampon de recepție și se activează semnalul RxRDY, pentru a indica unității centrale că un caracter este disponibil pentru transfer.

Transmițătorul, în modul asincron, pe durata START activează semnalul TxRDY, pentru a indica unității centrale că poate transfera un nou caracter în registrul tampon de emisie, deoarece caracterul curent este în curs de transmisie. Unitatea centrală răspunde cu WR activ și caracter pe magistrala de date, ceea ce anulează semnalul TxRDY. Dacă unitatea centrală nu încarcă un nou caracter până la terminarea transmisiei celui curent, se activează semnalul TxE (transmițător vid).

Dacă este programat **modul sincron**, receptorul înregistrează datele bit cu bit, și le grupează în caractere (conform modului prestabilit prin programare).

Pentru a grupa biții corect, receptorul trebuie sincronizat cu emițătorul, ceea ce se realizează în modul HUNT (vânătoare).

În modul HUNT, interfața încarcă fiecare bit recepționat la intrarea RxD, în registrul de recepție și compară conținutul acestuia cu cel care păstrează caracterul SYN, încărcat prin program.

Modul HUNT ia sfârșit când cele două caractere sunt identice; realizarea sincronizării determină activarea semnalului SYNDET (când sunt două caractere SYN, modul HUNT se termină după coincidența celor două).

Transmițătorul sincron transmite unul sau două caractere SYN la ieșirea TxD, după care urmează, fără pauză, caracterele blocului de date, în număr nelimitat.

Dacă nu are caracter de transmis, va transmite SYN, pentru a nu pierde sincronizarea cu receptorul.

15.2.5. Programarea

Constă în două cuvinte de comandă încărcate succesiv de unitatea centrală prin instrucțiunea "OUT".

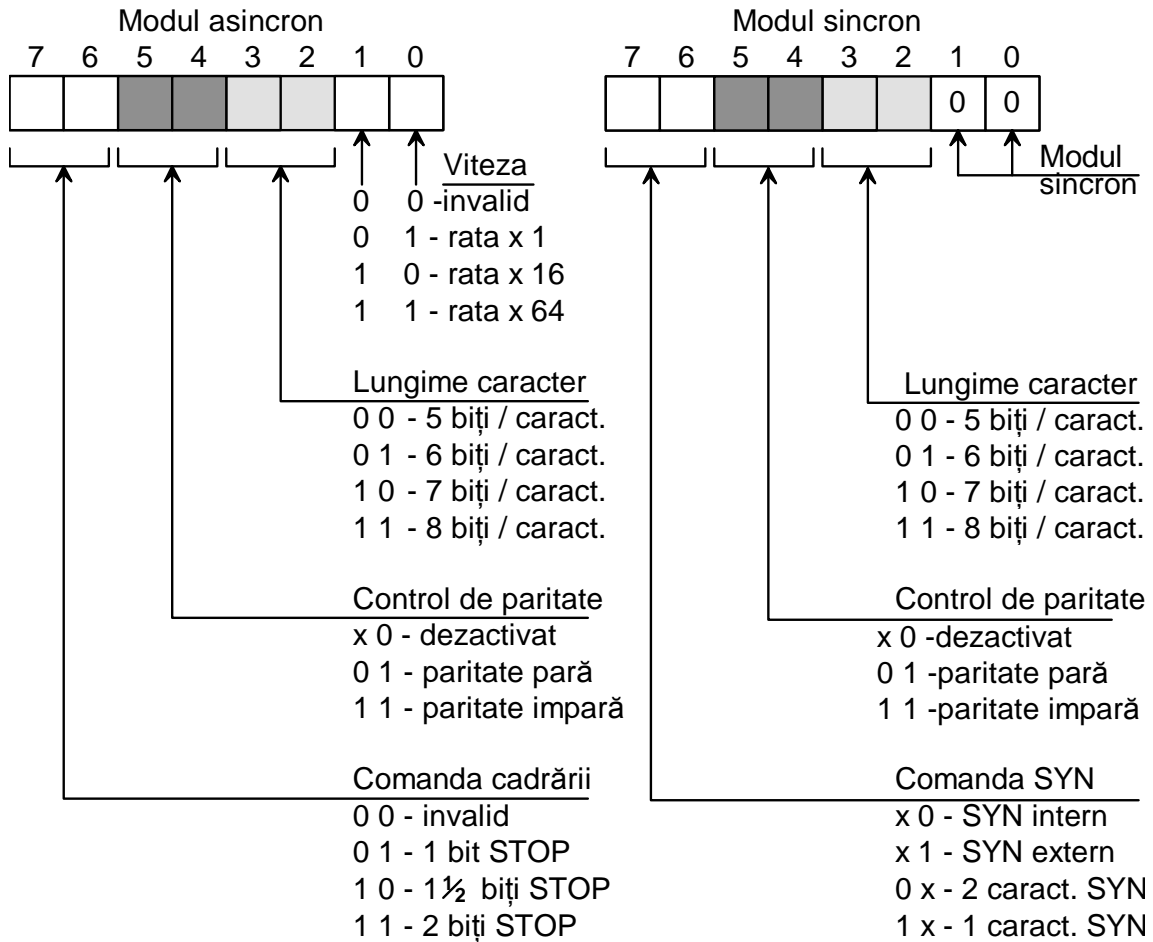


Fig.11 Structura primului cuvânt de comandă pentru cele două moduri

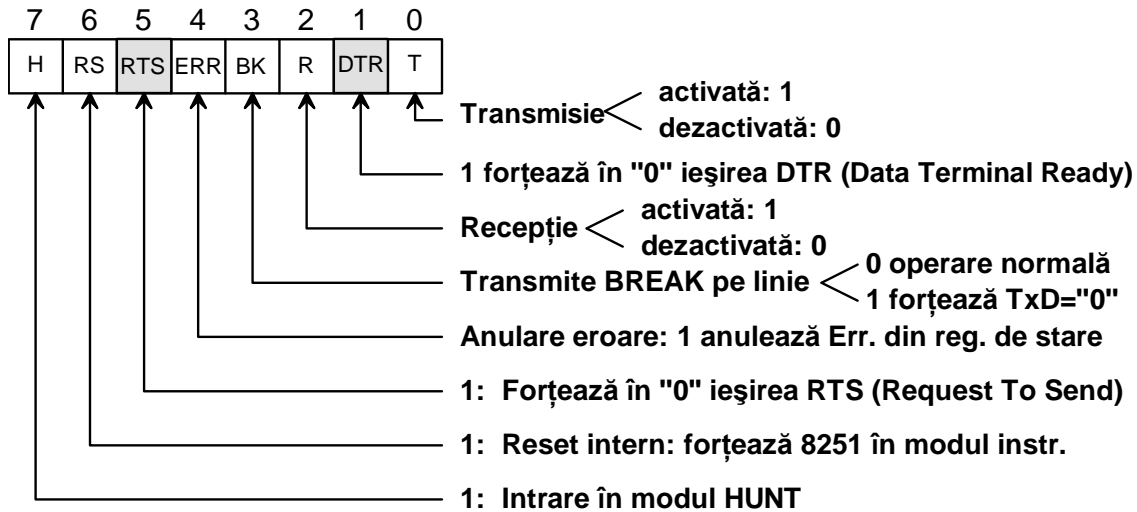


Fig. 12 Structura celui de-al doilea cuvânt de comandă

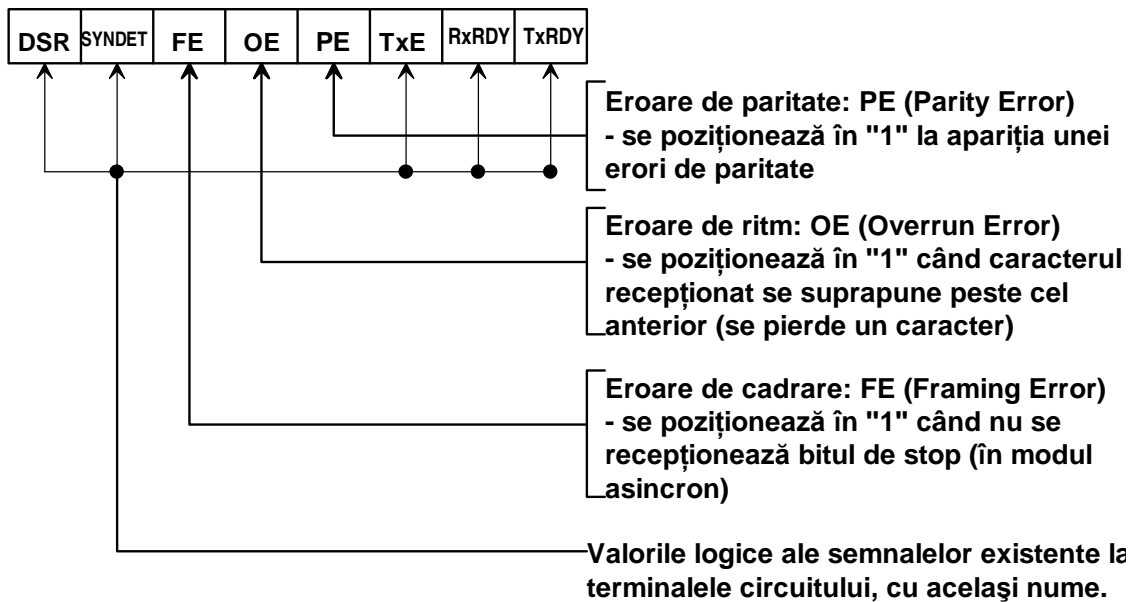


Fig. 13 Structura cuvântului de stare.

Structura primului cuvânt de comandă diferă după modul de lucru sincron sau asincron.

Al doilea cuvânt de comandă are aceeași structură pentru ambele moduri de lucru. În orice moment, unitatea centrală poate citi starea interfeței, cu instrucțiunea "IN <adresă port>", unde *adresă port* este adresa cuvântului de stare.

15.2.6. Adresarea circuitului 8251

Interfața serială 8251 conține din punct de vedere logic două porturi de 8 biți, care pot fi adresate pe minim 8 biți.

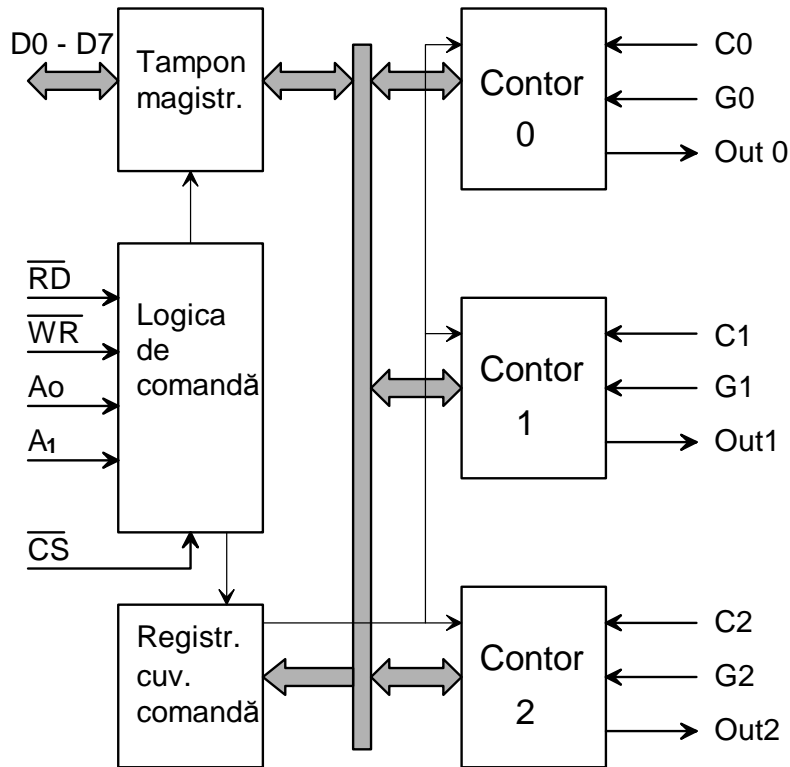


Fig.15 Intel 8253 - structura internă

Bitul A_0 , se utilizează de regulă ca separator între cele două adrese, fiind conectat la intrarea C/D.

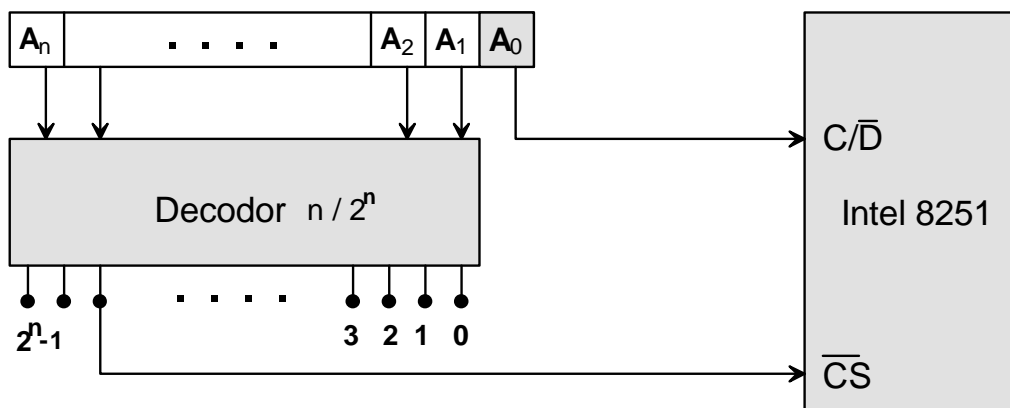


Fig. 14 Formarea adreselor de port.

Dacă $C/D = 0$, se adresează portul de date (adresă pară) iar când $C/D = 1$, se adresează portul de comandă sau stare (adresă impară). În ambele cazuri, valorile biților de adresă pentru generarea lui CS, sunt aceleași.

Exemplu:

Să considerăm adresa pe 8 biți:

1 1 1 0 1 1 1 0 = EE H; ($A_0=0$, deci este adresă de port de date)

Adresa următoare, va fi:

1 1 1 0 1 1 1 1 = EF H (adresa portului de comandă sau stare).

Utilizând aceste adrese, cele 4 tipuri de comenzi sunt:

OUT 0EEH, AL ; Acc. se transferă la portul de date (scriere caracter)

IN AL, 0EEH ; se transferă un octet în Acc. (citire caracter)

OUT 0EFH, AL ; cuvânt de programare la portul de comandă

IN AL, 0EFH ; citire cuvânt de stare de la portul de comandă/stare

15.3 Circuitul contor / periodizator Intel 8253 (Timer)

Circuit integrat LSI, realizat în tehnologie NMOS, cu 24 de pini, conține trei numărătoare independente de 16 biți. Fiecare numărător poate fi programat în 6 moduri de lucru, acoperind practic toate variantele curente de utilizare.

Structura internă a circuitului este prezentată în fig.15.

Blocurile funcționale principale sunt: registrul tampon de date, blocul de comandă, registrul cuvântului de comandă și cei trei contori, fiecare cu logica proprie de numărare și de conectare la magistrala internă de date.

Registrul tampon de date stochează temporar datele de intrare, la comanda de "scriere" (WR), pe care le transferă pe magistrala internă sau datele de ieșire pe care le transferă pe magistrala externă la comanda de "citire date" (RD), inițiată de unitatea centrală. Ieșirile sunt cu trei stări ("0", "1" și starea SIR, de înaltă impedanță).

Logica de comandă, pe baza celor 5 semnale de intrare, generează comenzi interne către toate blocurile circuitului, determinând realizarea funcțiilor acestora.

\overline{RD} (Read) determină citirea conținutului unui contor (octet inferior, octet superior), prin transferul octetului în registrul tampon, orientat spre magistrala de date externă.

\overline{WR} (Write) determină scrierea datelor în contori (inițializare contori) sau a cuvântului de comandă, traseul fiind: magistrală externă, registru tampon, magistrală internă, destinație (contor sau registru de comandă).

A_0, A_1 linii ale magistralei de adrese; sevesc selectării contorilor și registrului de comandă, conform tabelului de mai jos.

\overline{CS} (Chip Select) semnal de intrare pentru selecție circuit, care devine activ când pe magistrala de adrese se încarcă adresa circuitului, stabilită prin hardware.

CS	RD	WR	A1	A0	Funcția
0	1	0	0	0	scrie în contor 0
0	1	0	0	1	scrie în contor 1
0	1	0	1	0	scrie în contor 2
0	1	0	1	1	scrie în registrul de comandă
0	0	1	0	0	citește contor 0
0	0	1	0	1	citește contor 1
0	0	1	1	0	citește contor 2
0	0	1	1	1	combinație inoperantă
1	X	X	X	X	circuit neselectat
0	1	1	X	X	nici o operație

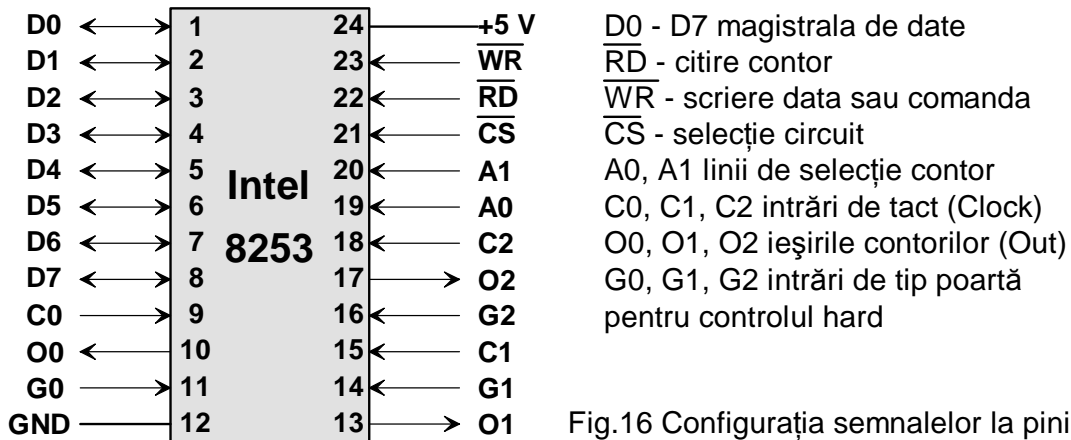


Fig.16 Configurația semnalelor la pini

Registrul cuvântului de comandă, de 24 de biți, memorează informația privind modul de lucru al fiecărui contor (trei cuvinte de comandă de 8 biți); se selectează conform tabelului de mai sus și nu poate fi citit.

Contorii 0, 1, 2, sunt numărătoare de 16 biți identice. Conținutul fiecărui contor este stabilit prin program, numărarea efectuându-se prin decrementare.

Contorii operează în cod binar sau în cod BCD, în diverse configurații privind intrarea de tact, poarta de comandă și ieșirea.

15.3.1. Modul de funcționare

Se stabilește prin cuvinte de comandă încărcate în registrul corespunzător. Valoarea inițială din fiecare contor se transferă direct, utilizând adresa contorului. După programare, fiecare contor este decrementat la fiecare front descrescător al semnalului său de tact.

Semnalul de tact (0 - 3 MHz) poate fi asincron cu semnalul de tact al unității centrale sau sincron cu acesta. În primul caz, contorul va fi utilizat la numărarea unor evenimente aleatoare, externe; în cel de-al doilea caz, va fi utilizat pentru generarea unor intervale de timp sincrone cu semnalul de tact al sistemului numeric.

Semnalul de intrare G (*Gate*) poate activa (cu "1" logic) contorul corespunzător sau poate bloca funcționarea contorului (cu "0" logic).

Ieșirile contorilor pot fi utilizate, în funcție de programare, ca semnale de tact, impulsuri singulare sau cereri de întrerupere.

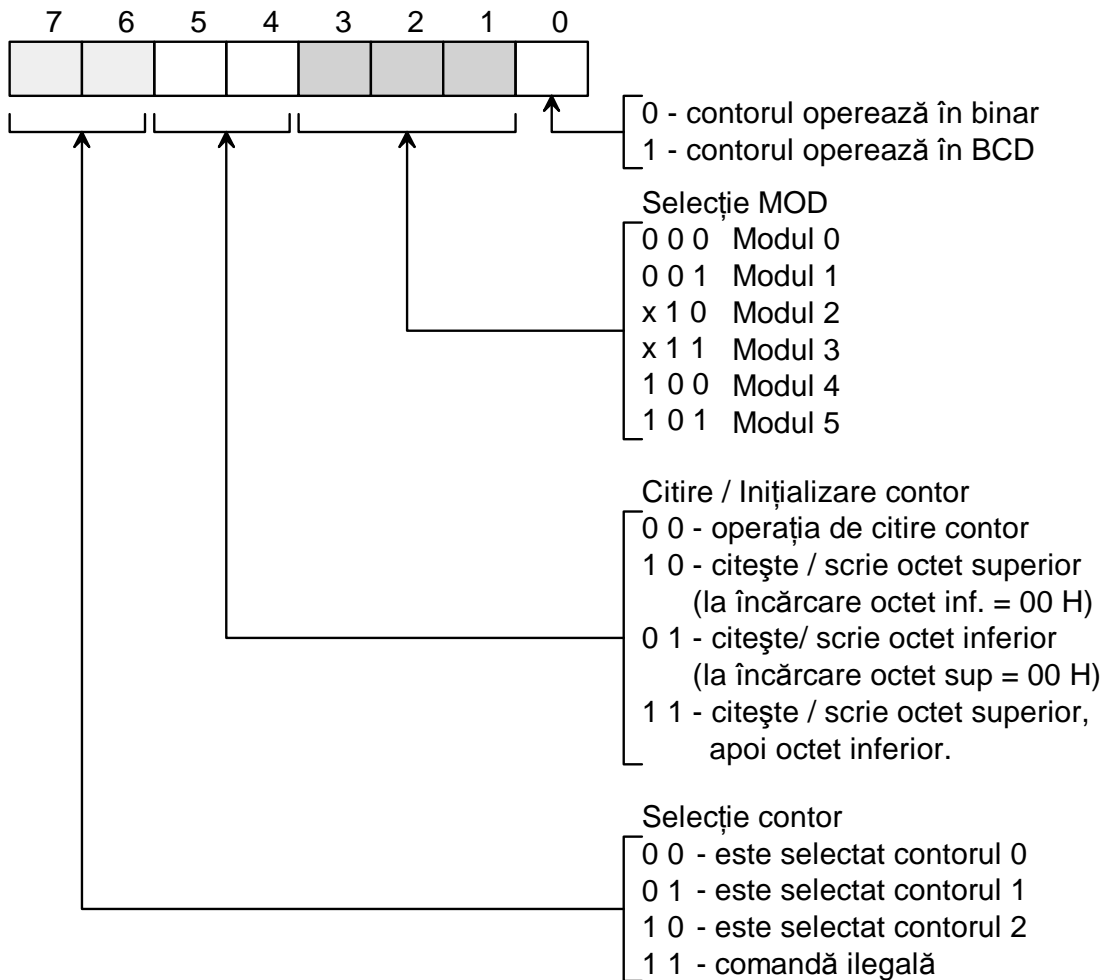


Fig. 17 Structura cuvântului de comandă

Un contor se consideră încărcat (inițializat) când s-au înscris în el 16 biți; în funcție de programarea biților b5, b4 din cuvântul de comandă,

dacă se înscrie numai octetul superior (D8 - D15) octetul inferior (D0 - D7) se inițializează automat cu 00 H iar dacă se înscrie numai octetul inferior, atunci octetul superior se inițializează automat cu 00 H. Dacă se înscriu ambii octeți, primul octet încărcat se consideră a fi cel superior.

Biții b3, b2, b1 programează modul de operare:

Modul 0: este definit ca întrerupere la terminarea numărării; ieșirea trece din "0" în "1" când numărătorul trece în starea 0000 H.

Modul 1: este definit ca monostabil programabil;

Modul 2: este definit ca generator de impulsuri divizate cu N;

Modul 3: este definit ca generator de impulsuri cu factor de umplere 50%;

Modul 4: este definit ca *strobe* comandat prin software; ieșirea este "0" o perioadă de tact, când starea numărătorului este 0000 H.

Modul 5: este definit ca *strobe* comandat prin hardware.

Decrementarea începe pe frontul crescător al intrării G; ieșirea este "0" o perioadă de tact, când starea numărătorului este 0000 H.

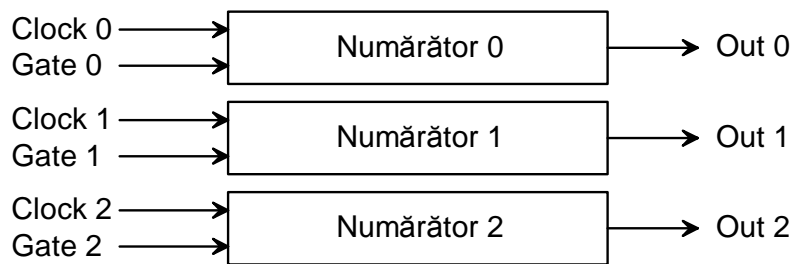


Fig. 18

Modul 0

Este definit ca întrerupere la terminarea numărării a N impulsuri de tact, unde N este numărul încărcat inițial; ieșirea trece din "0" în "1" când numărătorul trece în starea 0000 H.

După încărcarea cuvântului de comandă și a contorului cu valoarea inițială, ieșirea contorului trece în "0" și prin decrementare la fiecare impuls de tact, când valoarea contorului devine 0000 H, ieșirea trece în "1" logic. Decrementarea continuă și după atingerea valorii finale. În figura 19 sunt prezentate diagramele pentru N=5.

Decrementarea este blocată pe durata aplicării valorii "0" logic la intrarea G.

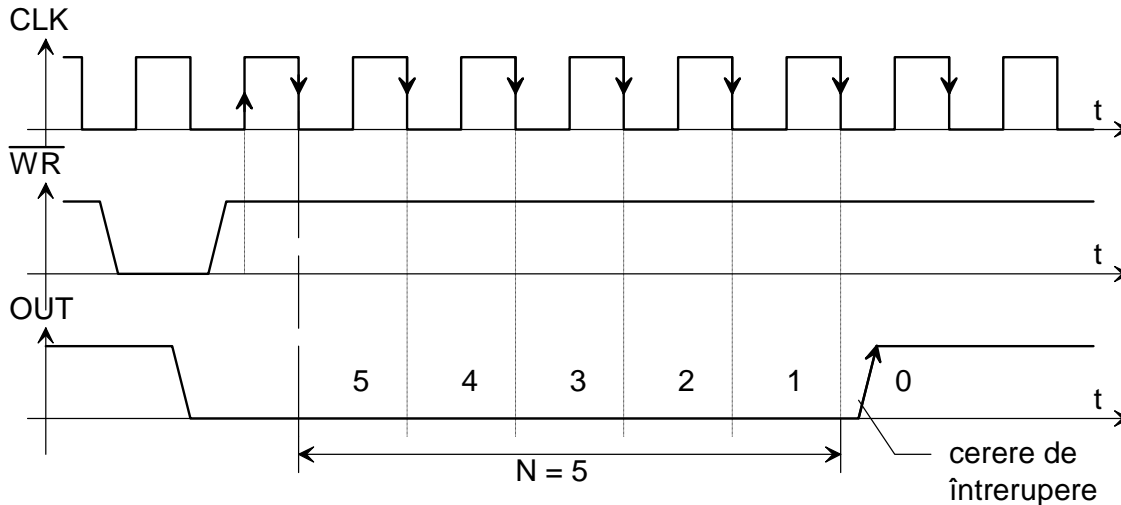


Fig. 19 Funcționarea unui numărător în modul 0.

Exemplu:

Considerăm adresele de selecție:

F4 H pentru N0,

F5 H pentru N1

F6 H pentru N2,

F7 H pentru registrul de comandă.

Secvența de program (8086) următoare, stabilește Mod 0 pentru N2, numărare binară și $N=128$, deci după 128 perioade de tact ieșirea trece în "1" logic.

Cuvântul de comandă este $1\ 0\ 0\ 1\ 0\ 0\ 0\ 0 = 90\ H$.

```

-----
MOV  AL, 90H          ; se încarcă în AL cuvântul de comandă
OUT  0F7H, AL        ; transferă cuv. de comandă la 8253
MOV  AL, 80H         ; se încarcă  $N = 128Z = 80H$  în registrul A
OUT  0F6H, AL        ; transferă N la contor 2, ca octet inferior
                          ; octetul superior = 00H (automat)
-----

```

Modul 1

Este definit ca monostabil programabil declanșat de o tranziție $0 \rightarrow 1$ la intrarea G (*Gate*), când ieșirea OUT trece în "0". Durata menținerii în "0" a ieșirii OUT este dată de valoarea N încărcată inițial în contor ($N \times T$).

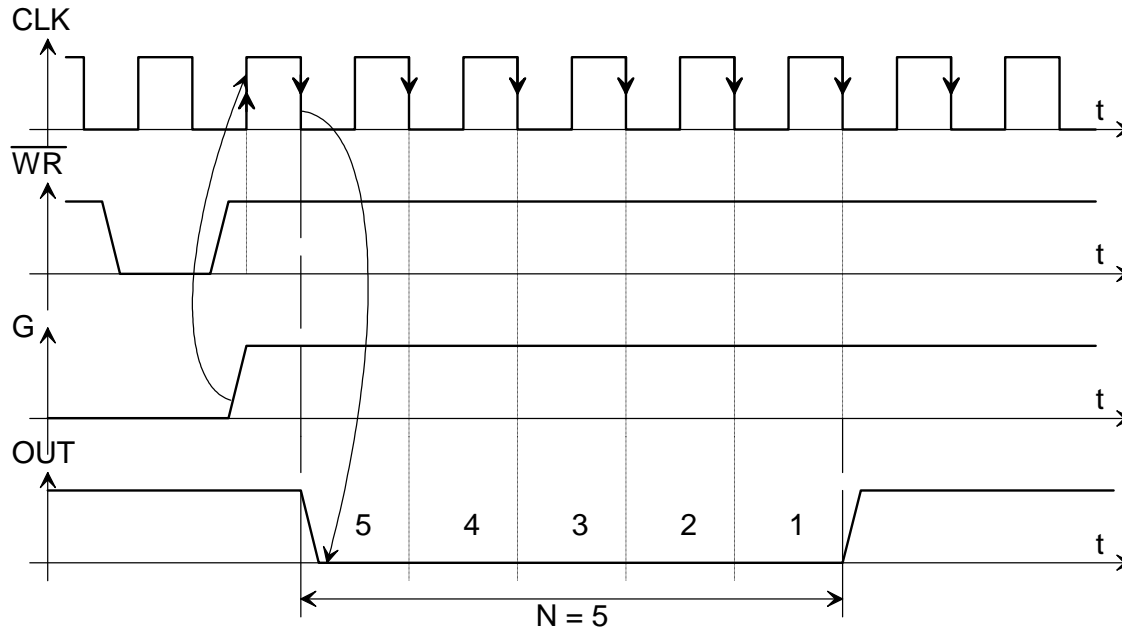


Fig. 20 Funcționarea unui numărător în modul 1.

Dacă la intrarea G apare o tranziție pozitivă, indiferent dacă decrementarea s-a terminat sau nu, contorul este relansat cu valoarea încărcată inițial (monostabil retrigerabil).

Exemplu:

Se programează contorul 1 în Mod 1, numărare în BCD, $N=3000$.

Cuvântul de comandă este: $01100011 = 63H$.

Secvența de program (8086) este următoarea:

```

-----
MVI AL, 63H           ; se încarcă în A cuvântul de comandă
OUT 0F7H, AL         ; transferă cuv. de comandă la 8253
MVI AL, 30H          ; se încarcă N =30 Z (30 BCD) în registrul A
OUT 0F5H, AL         ; transferă N la contor 1, ca octet superior
                       ; octetul inferior = 00H (automat)
-----

```

Modul 2

Este definit ca generator de impulsuri cu frecvența de N ori mai mică decât frecvența impulsurilor de tact (divizor cu N).

Ieșirea trece la nivel coborât pe durata unei perioade de tact, după numărarea celor N impulsuri (N este numărul încărcat inițial în contor).

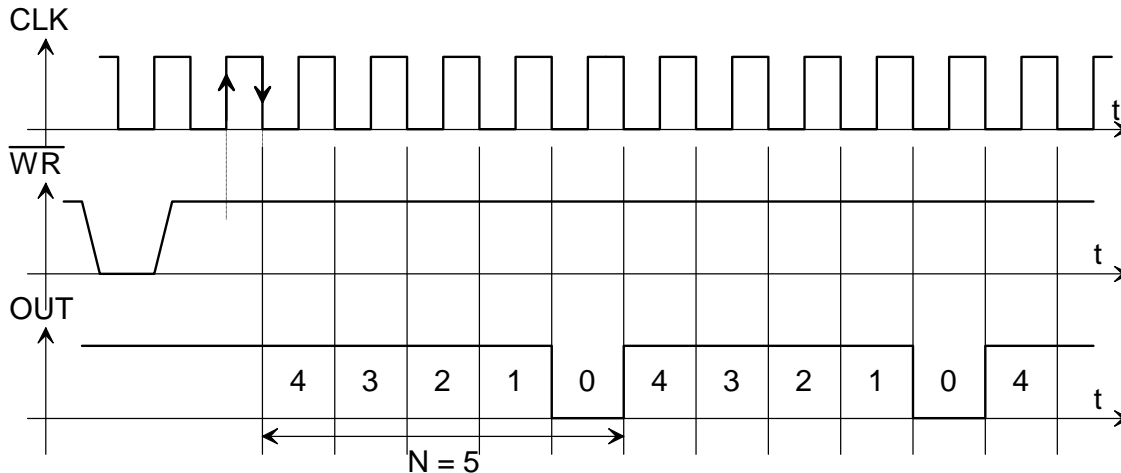


Fig. 21 Funcționarea unui numărător în modul 2 ($G=1$).

Contorul se blochează (ieșirea se menține în "1" logic) dacă intrarea G (*Gate*) trece în "0" logic pe durata decrementării. Când intrarea G revine în "1" logic, contorul pornește de la valoarea inițială (N); astfel, intrarea G poate fi utilizată pentru sincronizarea contorului cu evenimente externe.

Exemplu:

Se programează contorul 0 în Mod 2, fiind încărcat inițial cu valoarea 02F0 H.

Încărcarea se face începând cu octetul mai puțin semnificativ iar datele vor fi tratate în binar.

Cuvântul de comandă este: 0 0 1 1 0 1 0 0 = 34 H

```

-----
MOV AL, 34H      ; se încarcă în AL cuvântul de comandă
OUT 0F7H, AL     ; transferă cuv. de comandă la 8253
MOV AL, 02H     ; se încarcă în AL octetul superior
OUT 0F4H, AL     ; transferă 02 la contor 0, ca octet superior
MOV AL, 0F0H    ; se încarcă în AL octetul inferior
OUT 0F4H, AL     ; transferă F0 la contor 0, ca octet inferior

```

Modul 3

Este definit ca generator de impulsuri dreptunghiulare, similar cu cel din mod 2 dar cu factor de umplere 50%, mai precis, $f = \frac{N/2}{N} = \frac{1}{2}$ pentru N par și $f = \frac{(N+1)/2}{N} = \left(1 + \frac{1}{N}\right) \cdot \frac{1}{2}$ pentru N impar. Funcționarea este continuă, prin reluarea decrementării de la valoarea inițială, N .

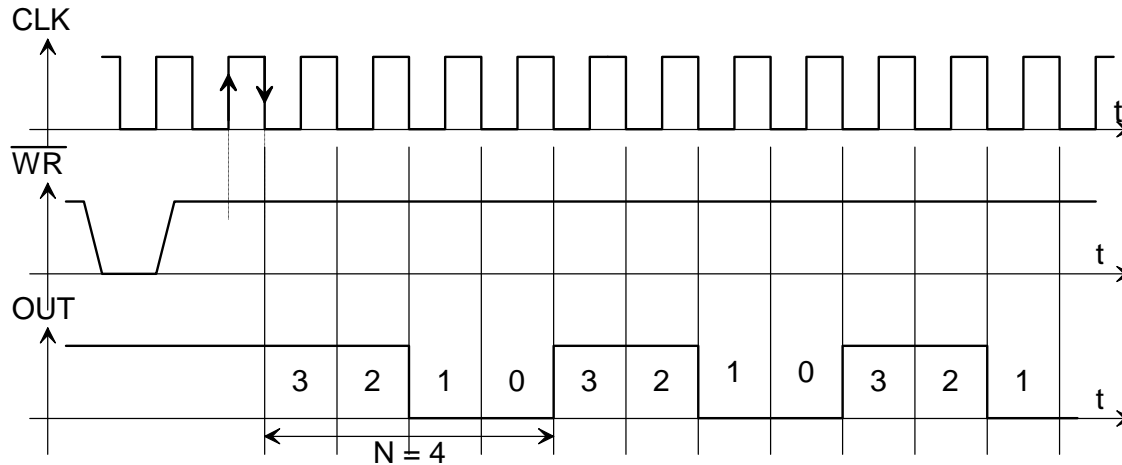


Fig. 22 Funcționarea unui numărător în modul 3 ($G=1$).

Modul 4

Definit ca *strobe* comandat prin software, ieșirea trece pe nivel coborât pentru o perioadă de tact, la terminarea decrementării.

După stabilirea modului de operare și după încărcarea valorii inițiale, N , ieșirea se menține la nivel ridicat cât timp starea contorului este diferită de 0000 H.

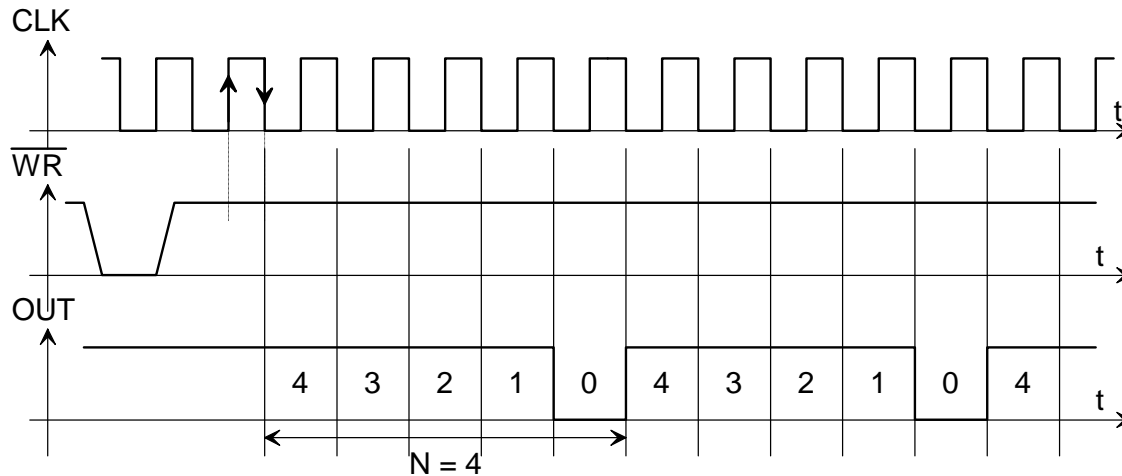


Fig. 23 Funcționarea unui numărător în modul 4.

Modul 5

Este definit ca *strobe* comandat prin hardware. Decrementarea contorului începe la primul impuls de tact după frontul crescător al intrării G . Ieșirea va fi în starea "0" pe o durată egală cu perioada impulsurilor de tact.

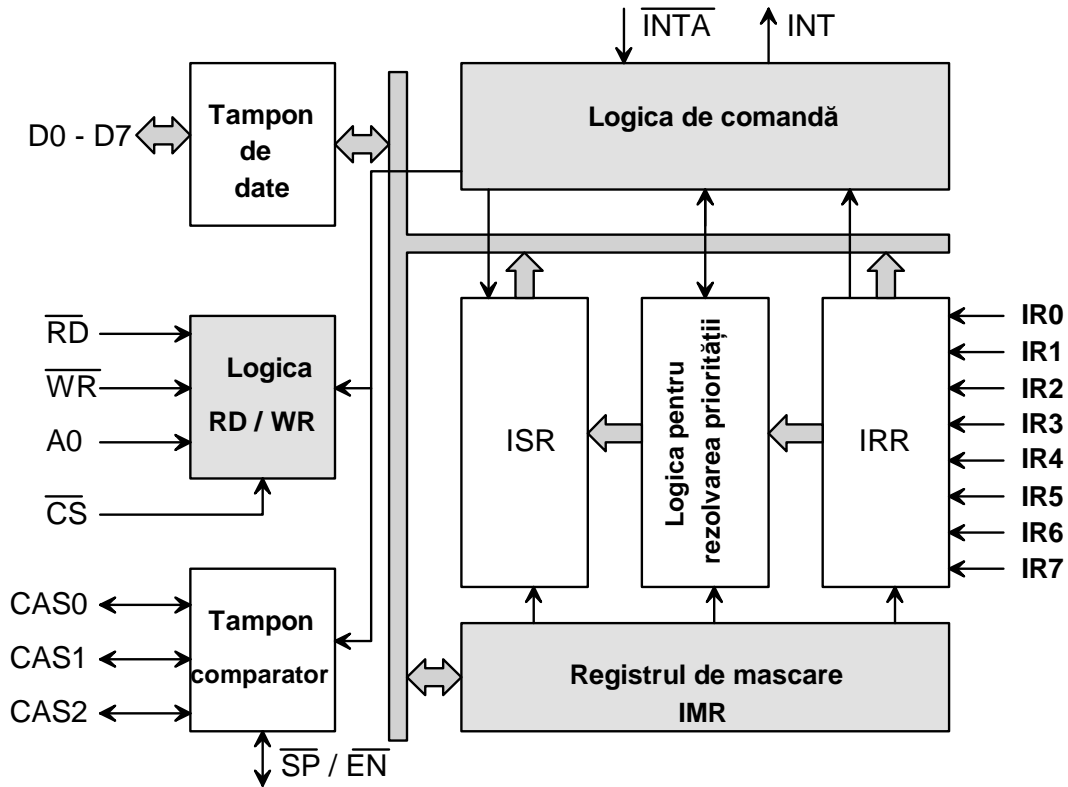


Fig. 26 Structura internă a circuitului Intel 8259

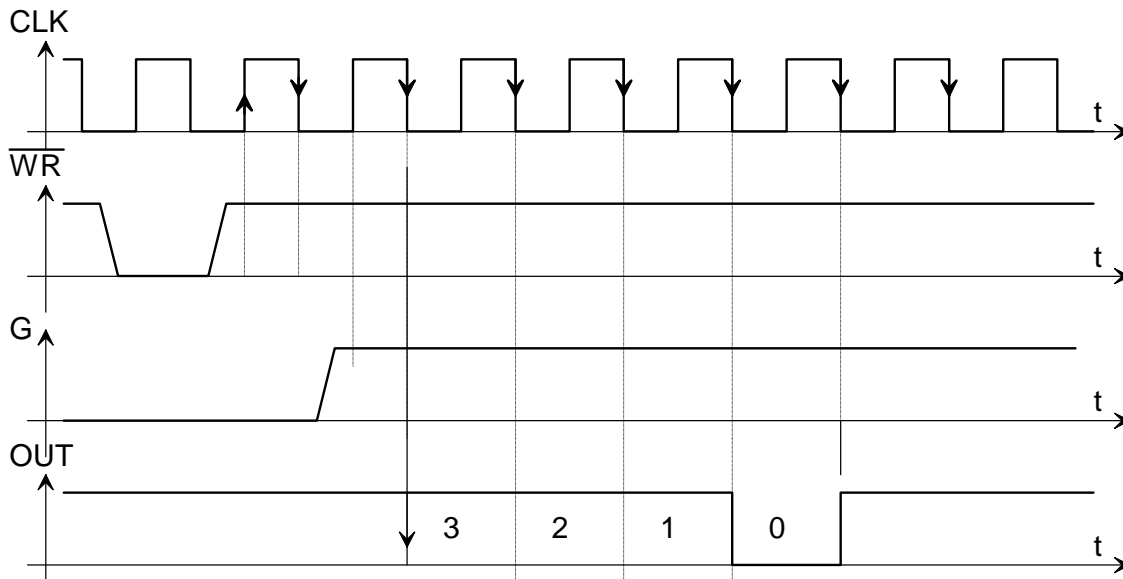


Fig. 24 Funcționarea unui numărător în modul 5.

Citirea conținutului unui contor

Se poate face în două moduri: în timpul funcționării sau prin blocarea tactului.

I. Conținutul contorului se transferă într-un registru intern al procesorului; decrementarea contorului continuă. Pentru citire este necesar un cuvânt special de comandă transmis la adresa portului de comandă.

7	6	5	4	3	2	1	0
		0	0	x	x	x	x
0	0	citire contor 0					
0	1	citire contor 1					
1	0	citire contor 2					

Fig. 25 Cuvântul de comandă pentru citirea unui contor

Biții b5=0, b4=0 specifică operația de memorare a conținutului unui contor.

II. Citirea contorului se face după blocarea semnalului de tact printr-o logică externă adecvată sau prin controlul unității centrale asupra intrării G, care poate bloca în anumite moduri de lucru decrementarea, chiar dacă semnalul de tact este prezent.

Exemplu:

I.

```

MOV AL, 40 H           ; cuvântul de comandă în AL (citire contor 1)
OUT 0F7H, AL          ; se transferă cuv. com. la 8253
IN AL, 0F5H           ; se citește octet inferior în AL.
MOV CL, AL            ; se salvează octet în registrul CL
IN AL, 0F5H           ; se citește octet inferior în AL.
MOV BL, AL            ; se salvează octet în registrul BL

```

II. Pentru citirea contorului cu blocarea decrementării, nu mai este necesar cuvântul de comandă (linia 1,2) dar trebuie oprită decrementarea.

15.4 Controlerul de întreruperi Intel 8259, 8259A, 82C59A (Programmable Interrupt Controller)

Circuit integrat LSI, realizat în tehnologie NMOS, 28 de pini, deservește 8 cereri de întrerupere de tip INT (mascabile). În cazul conectării în cascadă a 8 circuite 8259, considerate "*slave*" și unul "*master*", numărul cererilor de întrerupere ce pot fi deservite crește la 64.

Controlerul 8259 memorează cererile de întrerupere, stabilește ordinea de tratare, lansează cererea INT către unitatea centrală și susține dialogul necesar cu aceasta până la servirea completă a fiecărei cereri.

Blocurile componente principale (fig. 26) sunt: registrul tampon de date, blocul de control pentru citire / scriere, blocul pentru conectare în cascadă, blocul de comandă (logica de comandă), logica de tratare a priorităților și trei registre de 8 biți pentru evidența cererilor de întrerupere.

15.4.1. Registrele

1. Registrul cererilor de întrerupere (*Interrupt Request Register - IRR*), este conectat cu cele 8 intrări IR0 - IR7 (*Interrupt Request*) - cereri de întrerupere, pe care le memorează în momentul apariției unui front pozitiv. Înregistrarea unei cereri este anulată în momentul în care unitatea centrală acceptă tratarea acesteia prin semnalul INTA activ.

2. Registrul de servire a întreruperilor (*In Service Register - ISR*) specifică întreruperea activă, atunci când sunt înregistrate mai multe cereri; întreruperea activă este selectată pe baza logicii de tratare a prioritare, care funcționează conform priorităților stabilite prin programare.

3. Registrul de mascare (*Interrupt Mask Register - IMR*) înregistrează acele întreruperi care nu trebuie luate în considerație, acestea fiind stabilite de program și transmise controlerului 8259, prin cuvinte de comandă.

Registrul tampon de date conectează unitatea 8259 la magistrala de date a sistemului de calcul. Stochează temporar datele de intrare (D0-D7), la comanda de scriere (WR), pe care le transferă pe magistrala internă sau datele de ieșire pe care le transferă pe magistrala externă la comanda de "citire date" (RD), inițiată de unitatea centrală. Ieșirile sunt cu trei stări ("0", "1" și starea SIR, de înaltă impedanță).

15.4.2. Logica de comandă citește/scrie

Pe baza celor 4 semnale de intrare, generează comenzi interne către toate blocurile circuitului, determinând realizarea funcțiilor acestora.

$\overline{\text{RD}}$ (*Read*) determină citirea stării circuitului.

$\overline{\text{WR}}$ (*Write*) determină scrierea cuvintelor de comandă în 8259.

A_0 linie de adrese; sevește selectării celor două porturi I/O.

$\overline{\text{CS}}$ (*Chip Select*) semnal de intrare pentru selecție circuit, care devine activ când pe magistrala de adrese se încarcă adresa circuitului, stabilită prin hardware.

15.4.3. Blocul de cascading

Permite conectarea în paralel a 2, 3 . . . ,8 circuite 8259 de tip *slave* ($\overline{SP/EN}= 0$) cu un circuit *master* ($\overline{SP/EN}= 1$), pentru creșterea numărului de cereri de întrerupere deservite de la 8 la 64.

Dacă se utilizează un singur controler 8259, pinii de cascadare CAS0 - CAS2 rămân neconectați, iar unitatea se consideră *master*.

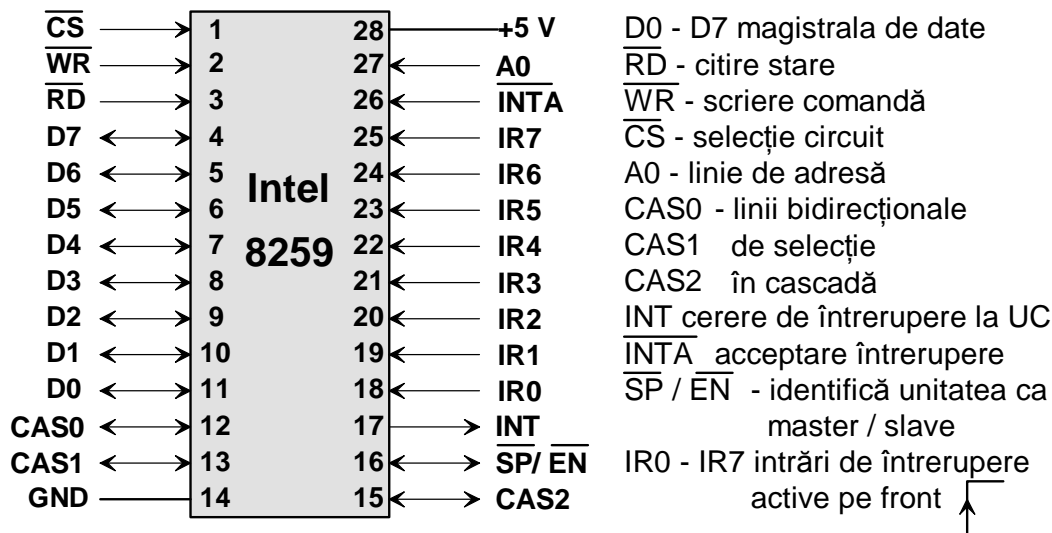


Fig.26 Configurația semnalelor la pinii circuitului 8259A

15.4.4. Secvențele de lucru în servirea cererilor de întrerupere

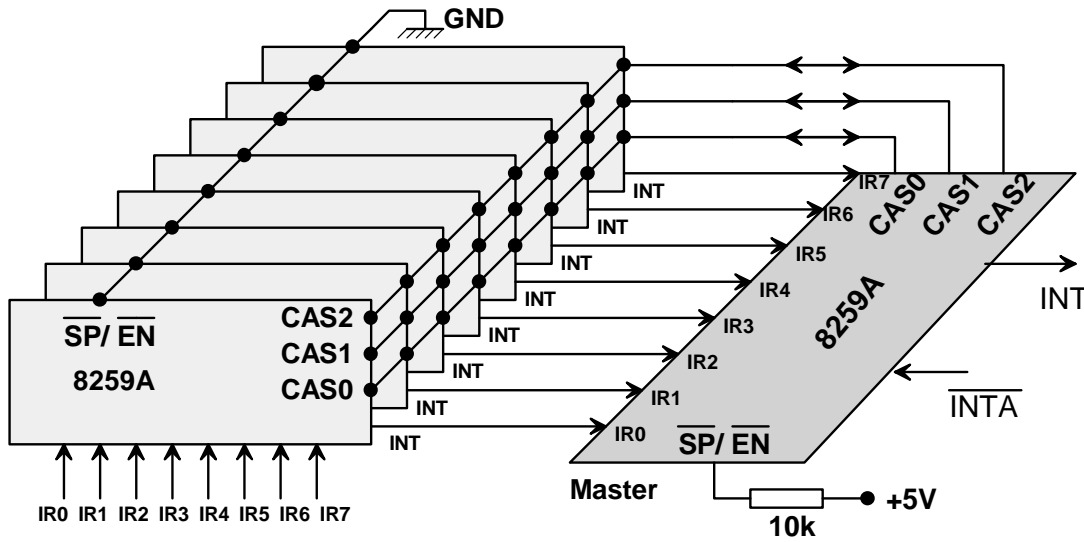
1. Una sau mai multe intrări IR sunt activate de fronturile crescătoare ale cererilor de întrerupere;

2. Unitatea 8259 transmite $INT = 1$ către unitatea centrală, ca urmare a acceptării unei cereri de întrerupere, după rezolvarea priorității.

3. Unitatea centrală acceptă tratarea și generează INTA (*Interrupt Acknowledge*);

4. Unitatea 8259 recunoaște INTA și răspunde cu codul instrucțiunii CALL, pe magistrala de date.

5. Unitatea centrală generează succesiv încă două semnale INTA, la care 8259 răspunde de fiecare dată cu octet pe magistrala de date, în total 16 biți, care reprezintă adresa subrutinei de tratare a cererii de întrerupere. Se formează astfel cei 3 octeți ai instrucțiunii CALL <Adresa>, de salt necondiționat la subrutina de tratare.



8 x 8259A Slave

Fig. 27 Conectarea unităților 8259A în cascadă

15.4.5. Programarea unității PIC 8259A

Pentru programare se transmite circuitului două tipuri de cuvinte de comandă:

- ◆ *Primul tip* este de inițializare (*Initialization Command Word-ICW*), de 2 sau 4 octeți, notați ICW1, ICW2, ICW3, ICW4.

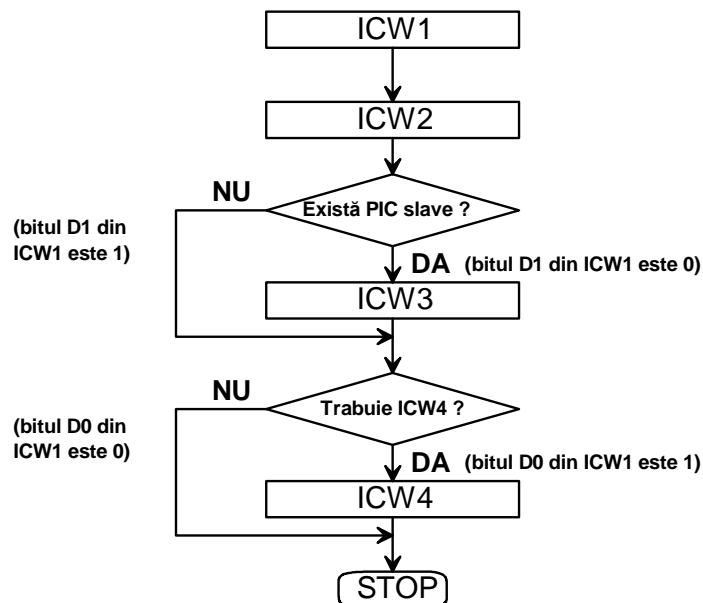


Fig. 25 Schema logică a cuvintelor de comandă pentru inițializare

- Primul cuvânt de comandă, ICW1, transmis cu $A0 = 0$, are ca efect:
- aducerea la zero a registrului IRR (anularea unor cereri înregistrate);
 - aducerea la zero a registrului IMR (de mascare cereri de întrerupere);

- atribuirea priorităților standard (prioritate maximă IR0, minimă IR7);
- stabilirea existenței unor unități slave.

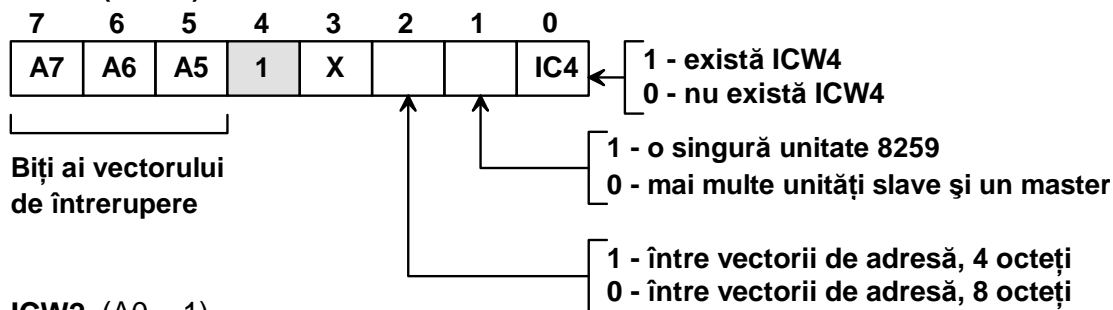
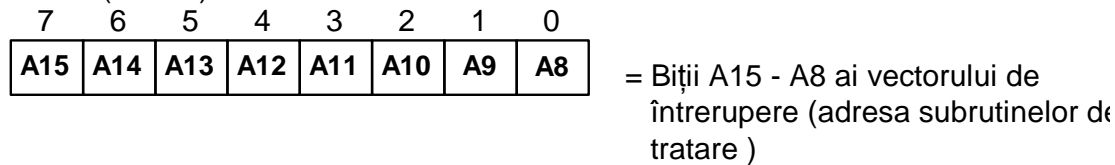
ICW1 (A0 = 0)**ICW2 (A0 = 1)**

Fig. 28 Structura cuvintelor de comandă ICW1 și ICW2.

Al doilea cuvânt de comandă, ICW2, specifică o parte din adresa subrutinelor de tratare a întreruperilor și anume biții A15 - A8 din cei 16 biți de adresă.

Biții A0, A1 sunt "0" logic pentru toate adresele, biții A2, A3, A4 sunt poziționați conform nivelului întreruperii (Tab. 2) iar A5, A6, A7 sunt specificați de ICW1.

Tab.2a

A7	A6	A5	A4	A3	A2	A1	A0	Linia
0	0	0	0	0	0	0	0	IR0
0	0	0	0	0	1	0	0	IR1
0	0	0	0	1	0	0	0	IR2
0	0	0	0	1	1	0	0	IR3
0	0	0	1	0	0	0	0	IR4
0	0	0	1	0	1	0	0	IR5
0	0	0	1	1	0	0	0	IR6
0	0	0	1	1	1	0	0	IR7

Tab.2b

A7	A6	A5	A4	A3	A2	A1	A0	Linia
0	0	0	0	0	0	0	0	IR0
0	0	0	0	1	0	0	0	IR1
0	0	0	1	0	0	0	0	IR2
0	0	0	1	1	0	0	0	IR3
0	0	1	0	0	0	0	0	IR4
0	0	1	0	1	0	0	0	IR5
0	0	1	1	0	0	0	0	IR6
0	0	1	1	1	0	0	0	IR7

Tabelul 2 prezintă formarea octetului inferior de adresă în cele două cazuri ce rezultă din structura lui ICW1: între vectorii de adresă 4 locații (2a) și între vectorii de adresă 8 locații (2b).

Astfel vectorii de adresă sunt determinați de cele două cuvinte de comandă.

Dacă, de exemplu, ICW2 = 86 H și bitul D2 = 1 în ICW1 (4 locații între adrese), rezultă următoarele 8 adrese ale subrutinelor de tratare întrerupere:

- ♦ IR0 determină CALL 8600 H ; subrutina la 8600H
- ♦ IR1 determină CALL 8604 H ; subrutina la 8604H
- ♦ IR2 determină CALL 8608 H ; subrutina la 8608H
- ♦ IR3 determină CALL 860C H ; subrutina la 860CH
- ♦ IR4 determină CALL 8610 H ; subrutina la 8610H
- ♦ IR5 determină CALL 8614 H ; subrutina la 8614H
- ♦ IR6 determină CALL 8618 H ; subrutina la 8618H
- ♦ IR7 determină CALL 861C H ; subrutina la 861CH

Dacă ICW2 = 86 H și bitul D2 = 0 în ICW1 (8 locații între adrese), rezultă adresele subrutinelor de tratare întrerupere sunt:

- ♦ IR0 determină CALL 8600 H ; subrutina la 8600H
- ♦ IR1 determină CALL 8608 H ; subrutina la 8608H
- ♦ IR2 determină CALL 8610 H ; subrutina la 8610H
- ♦ IR3 determină CALL 8618 H ; subrutina la 8618H
- ♦ IR4 determină CALL 8620 H ; subrutina la 8620H
- ♦ IR5 determină CALL 8628 H ; subrutina la 8628H
- ♦ IR6 determină CALL 8630 H ; subrutina la 8630H
- ♦ IR7 determină CALL 8638 H ; subrutina la 8638H

- ♦ *Al doilea tip de cuvânt de comandă* este de operare (*Operation Command Word - OCW*) și permite mascarea unor întreruperi, stabilirea "sfârșit tratare", modificarea nivelurilor de prioritate și pregătirea pentru citirea stării.

Următoarea secvență în limbaj de asamblare (80286) realizează inițializarea unei unități 8259A master și a unei unități slave.

```

; Initializarea lui 8259 master
MOV AL, 11H ; ICW1
OUT 20H, AL
JMP $+2 ; Timp de asteptare
MOV AL, 08H ; ICW2, octetul superior de adresa
OUT 21H, AL
JMP $+2 ; Timp de asteptare
MOV AL, 04H ; ICW3
OUT 21H, AL
JMP $+2 ; Timp de asteptare
MOV AL, 01H ; ICW4
OUT 21H, AL
JMP $+2 ; Timp de asteptare
MOV AL, 0FFH ; Validarea intreruperilor
OUT 21H, AL

```

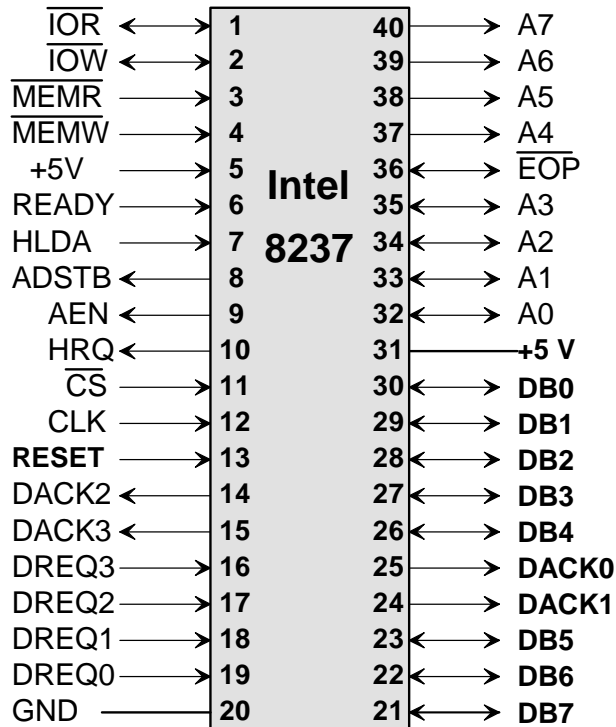


Fig. 30 DMA - 8237 Configurația semnalelor la pini

```

; Initializarea lui 8259 slave
MOV AL, 11H      ; ICW1
OUT A0H, AL
JMP $+2         ; Timp de asteptare
MOV AL, 70H     ; ICW2, octetul superior de adresa
OUT A1H, AL
JMP $+2         ; Timp de asteptare
MOV AL, 02H     ; ICW3
OUT A1H, AL
JMP $+2         ; Timp de asteptare
MOV AL, 01H     ; ICW4
OUT A1H, AL
JMP $+2         ; Timp de asteptare
MOV AL, 0FFH   ; Validarea intreruperilor
OUT A1H, AL

```

La terminarea tratării unei întreruperi, unitatea centrală trebuie să trimită la 8259, cu A0 = 1, un cuvânt de "terminare normală" (20H); există și variante de "terminare specială".

Citirea stării se face cu instrucțiunea IN, după ce în prealabil s-a transmis la unitatea 8259 un cuvânt special pentru citire.

15.5 Controlerul programabil Intel 8237

(Direct Memory Access - DMA)

Circuit LSI realizat în tehnologie NMOS, 40 de pini, cu o singură tensiune de alimentare (+5V), servește pentru transferul rapid al blocurilor de date, fără implicarea procesorului, care poate efectua, pe durata transferului, operații interne din programul propriu, fără utilizarea magistralelor. Principalele caracteristici sunt:

- ♦ poate deservi 4 periferice independente (are 4 canale DMA);
- ♦ viteza maximă de transfer este 1,6 MB/sec.;
- ♦ dimensiunea maximă a blocului de date transferat este de 64 kB;

Transferul DMA se poate efectua între:

- ♦ zonă de memorie - zonă de memorie;
- ♦ memorie - disc magnetic;
- ♦ memorie - periferic rapid (disc optic, bandă, etc.);

Obiectivul tehnicii DMA este de a câștiga timp; microprocesorul este degrevat de operații repetitive, de rutină, care sunt efectuate de un circuit specializat, sub controlul microprocesorului.

Pentru organizarea unui transfer DMA, trebuie să se specifice:

- ♦ Adresa de început a zonei de memorie care se transferă (sursă);
- ♦ Adresa de început a zonei de memorie în care se transferă (destinație);
- ♦ Numărul de octeți ce se transferă.

Circuitul DMA este încărcat de procesor cu aceste informații și apoi girează transferul, cedând controlul asupra magistralelor sistemului pe durata transferului.

Pentru cedarea magistralelor, procesorul se autodeconectează prin trecerea ieșirilor sale în starea de înaltă impedanță; controlerul DMA raportează terminarea transferului prin dezactivarea semnalului de solicitare, ceea ce determină procesorul să preia controlul asupra magistralelor.

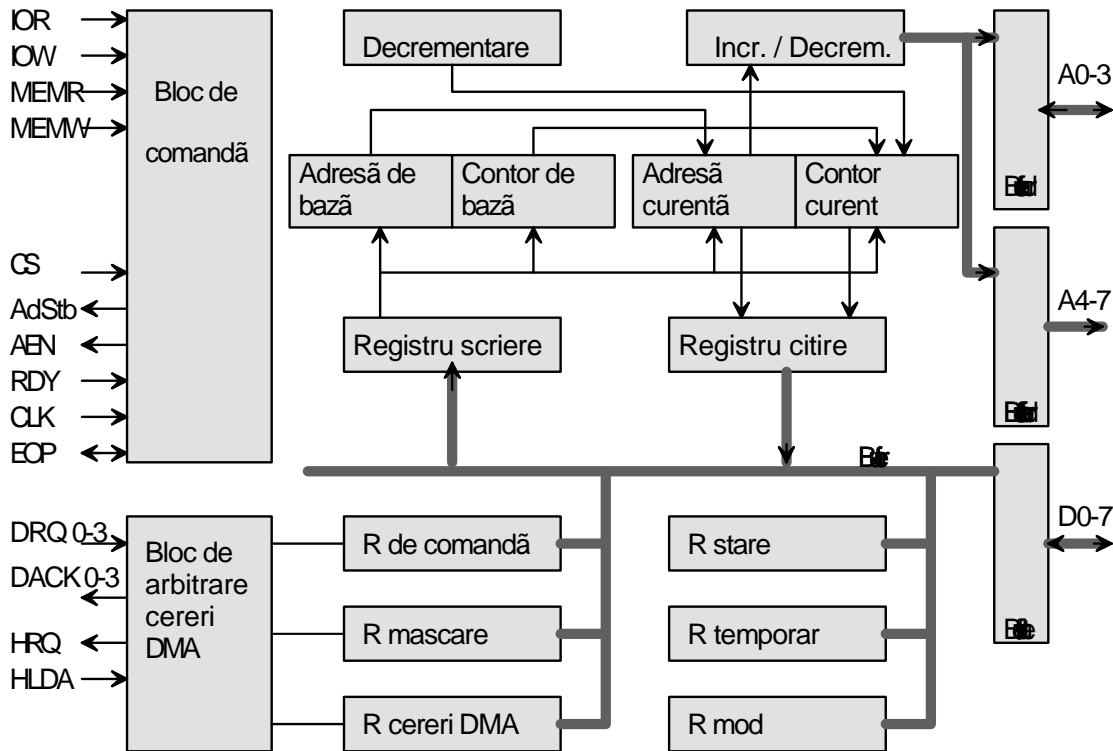


Fig. 29 Structura internă a controlerului DMA - Intel 8237

Controlerul DMA 8237 conține în structură un număr mare de registre (27), din care cele mai importante sunt:

- ♦ Registrul pentru adresa de bază: conține adresa de început a zonei destinație care nu se modifică pe durata transferului;
- ♦ Registrul numărului de octeți: conține dimensiunea, în octeți a blocului de date care se transferă, nu se modifică pe durata transferului;
- ♦ Registrul adresei curente: conține adresa octetului ce se transferă și este incrementat după fiecare transfer de octet;
- ♦ Registrul numărător de octeți: conține numărul de octeți de transferat și este decrementat după fiecare transfer; când ajunge la zero, transferul este încheiat;
- ♦ Registrul de comandă: este destinat memorării cuvântului de programare;
- ♦ Registrul de mod: determină modul de funcționare (unul din trei posibile);
- ♦ Registrul cererilor DMA: răspunde cererilor de transfer și le înregistrează;
- ♦ Registrul de mascare: servește mascării unor linii de solicitare DMA;

- ♦ Registrul de stări: memorează starea circuitului și poate fi citit de procesor;
- ♦ Registrul de adresă temporară;
- ♦ Registrul de numărare temporară a cuvintelor transferate.

15.5.1. Semnalele și modurile de funcționare

Controlerul programabil pentru DMA, 8237 dispune de 4 intrări pentru cereri de tip DMA, notate DREQ0 - DREQ3 (*DMA Request*), ceea ce era suficient pentru primele PC-uri. Prin conectarea în cascadă a două circuite 8237, rezultă 7 linii de intrare DMA, necesare la PC - AT.

- ♦ Când un periferic necesită un transfer masiv de date, emite o cerere către 8237, la una din intrările DREQ; aceste intrări sunt independente și au fiecare un nivel de prioritate, ca și cererile de întrerupere. Cererea DMA cea mai prioritară este prima luată în considerație.

De regulă prioritatea cea mai înaltă este atribuită liniei 0.

- ♦ La apariția unei cereri DMA, 8237 emite un semnal HRQ (*Hold Request*) la microprocesor, prin care se cere acestuia cedarea magistralelor.
- ♦ Microprocesorul răspunde prin HLDA (*Hold Acknowledge*), după trecerea magistralelor în starea de înaltă impedanță.
- ♦ Controlerul 8237 emite acordul DACK către solicitatorul de DMA.
- ♦ Se execută transferul de date prin acces direct la memorie.

Unitatea 8237 trebuie să contabilizeze două adrese, pe care să le țină "la zi": adresa sursă și adresa destinație. Octetul citit este stocat temporar într-un registru intern; etapele transferului sunt:

- ♦ 8237 citește un octet de la adresa sursă;
- ♦ memorează octetul citit în registrul temporar;
- ♦ comută pe adresa destinație și transferă octetul;

Tabel 2. Semnalele controlerului DMA 8237

Semnal	I - intrare E - ieșire	Funcție
CLK	I	Semnalul de tact al sistemului
CS	I	Selecție circuit
RESET	I	Inițializare
READY	I	Gata - acordare de timp de așteptare memoriilor lente
HLDA	I	Acceptare de către procesor a cererii DMA
DREQ0 - DREQ3	I	DMA Request - 4 intrări, cereri DMA
DB0 - DB7	I / E	Magistrală de date de 8 biți
IOR	I / E	Input/Output Read - Citire port

IOW	I / E	Input/Output Write - Scriere port
EOP	I / E	End of Process - Terminare transfer
A0, A1, A2, A3	I / E	Adresă pentru registre (I) / Adresă DMA (E)
A4, A5, A6, A7	E	Adresă pentru DMA
HRQ	E	Hold Request - Cerere de DMA la procesor
DACK0 - DACK3	E	DMA Acknowledge - Acceptare DMA
AEN	E	Address Enable - Autorizare adresă
ADSTB	E	Address Strobe - Încărcare octet sup. de adresă
MEMR	I	Memory Read - Citire pentru transfer
MEMW	I	Memory Write - Scriere pentru transfer

Adresa sursă poate fi menținută constantă, caz în care se transferă același octet în toată zona destinație.

Prioritățile programate pot fi:

1. Fixe: nivelul de prioritate corespunde numărului cererii, "0" fiind cea de prioritate maximă.

2. Variabilă: ultimul nivel care a fost servit pentru transfer devine cel mai puțin prioritar, din cele rămase se elimină ultimul servit și așa se determină nivelul ce va fi servit (regula se aplică dacă apar cereri simultane).

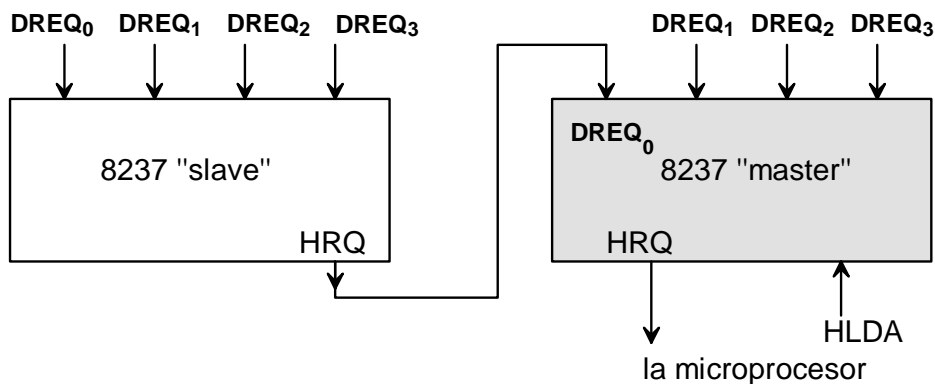


Fig. 31 Montarea în cascadă a două circuite 8237. Numărul de intrări devine 7

Într-un PC - AT, unitățile DMA 8237 sunt cascadeate (fig. 31) și furnizează astfel 7 intrări pentru cereri DMA. Ceea ce complică schema, este adăugarea unor circuite externe pentru stocarea octeților suplimentari de adresă, dat fiind că un 8237 are magistrala de adrese de 8 biți (A0 - A7) iar un PC - AT lucrează cu adrese de 20 sau 32 de biți.

Pentru evitarea conflictelor logice în alocarea intrărilor DMA, se alocă, fiecărui periferic nou introdus, o linie pentru cereri DMA.

