

## 1.1. Descrierea generala a procesoarelor x86 pe 16 biti

În figurile 1.2 și 1.3 sunt prezentate schemele bloc ale celor două unități centrale. Așa cum am spus ambele încorporează două blocuri de prelucrare separate, UE și UIM. UE este identică la ambele procesoare, UIM diferind prin aceea că la 8086 ea lucrează pe o magistrală de date de 16 biți cu coadă de instrucțiuni de 6 octeți; la 8088 magistrala este de 8 biți iar coada de 4 octeți.

UE are funcția de a executa toate instrucțiunile, comunicând prin date și adrese cu UIM și manipulând registrele generale, indicatorii de condiții și unitatea aritmetică/logică. Cu excepția câtorva conexiuni externe de comandă, UE este complet izolată de exteriorul procesorului. UIM are funcția de a executa toate ciclurile de acces la magistrala externă, fiind alcătuită din registrele de segment, registrele de comunicație internă, *pointer*-ul de instrucțiuni, stiva pentru codul obiect al instrucțiunii și un sumator specializat. Operațiile pe care le face UIM sunt: obținerea adresei prin adunarea valorii segmentului cu valoarea deplasamentului, transferarea datelor spre UE pe magistrala de date internă, de 16 biți, a unității aritmetice/logice, și încărcarea în avans, preextragerea, codurilor obiect ale instrucțiunilor din memoria sistemului în coada de așteptare, de unde vor fi preluate de UE.

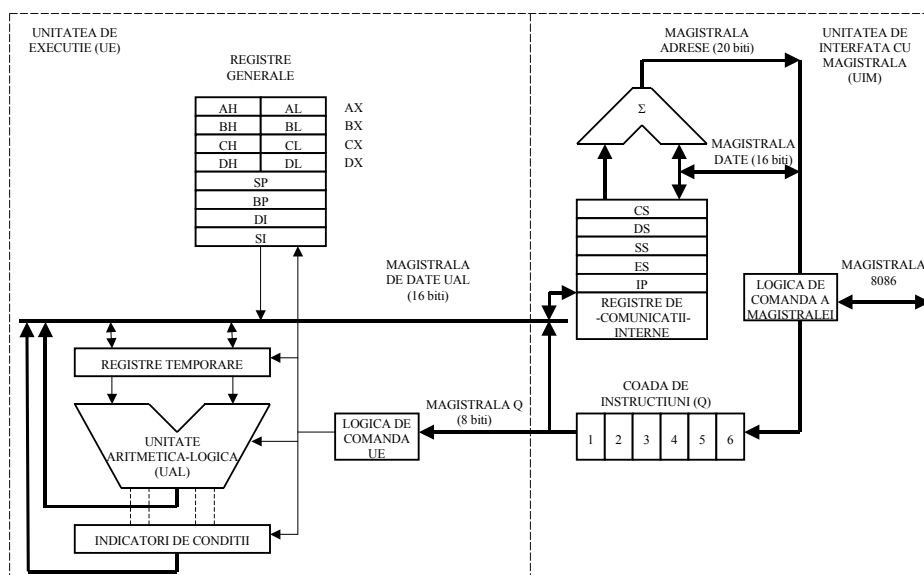


Figura 1.2. Schema bloc a microprocesorului 8086

UE extrage, atunci când este gata să execute o instrucțiune, codul obiect al acesteia din coada de așteptare, trecând apoi la execuția ei. În cazul în care coada de așteptare nu are nici-un octet, UE va aștepta până ce UIM va extrage codul din memoria sistemului. Dacă în cursul execuției unei instrucțiuni este necesar accesul la o locație de memorie sau la un *port* de I/O, UE va face, de asemenea, o cerere către UIM pentru ca aceasta să execute ciclul de magistrală dorit.

Cele două blocuri, UE și UIM, funcționează independent. La 8086, dacă cel puțin doi octeți din coadă sunt liberi și UE nu are nici-o cerere către

UIM, aceasta din urmă va executa ciclul de extragere în scopul ocupării întregii cozi. La 8088, UIM extrage cod obiect atunci când coada are cel puțin un octet liber. Dacă UE are nevoie de un acces la magistrală și UIM se află în timpul unui ciclu de extragere, UE va aștepta terminarea acestui ciclu înainte de a fi luată în considerare cererea.

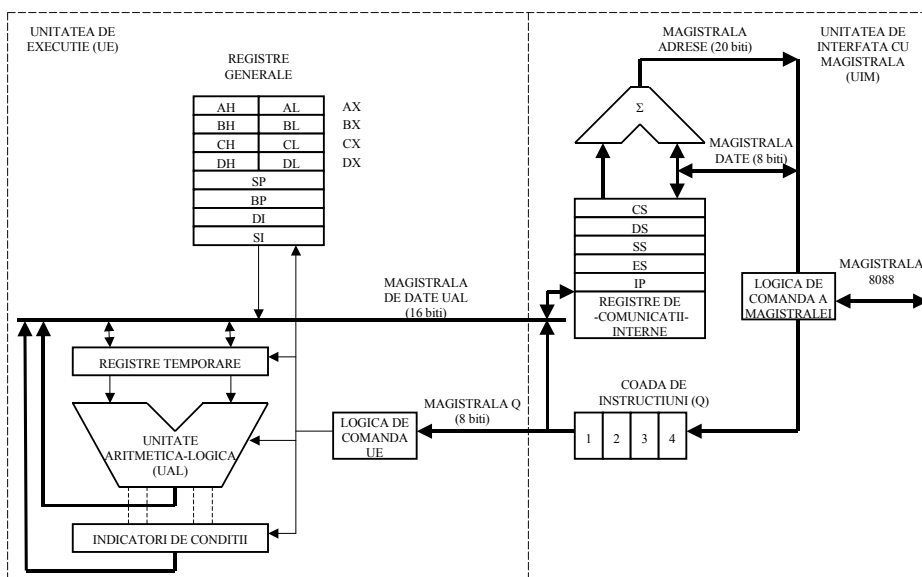


Figura 1.3. Schema bloc a microprocesorului 8088

Registrele programabile ale microprocesorului 8086 sunt înfățișate în figura 1.4. Prezentăm în continuare registrele în legătură cu modurile de adresare ale lui 8086 deoarece multe din acestea sunt utilizate de logica de adresare a memoriei.

AX este acumulatorul principal utilizat, de exemplu, de instrucțiunile de I/O care transferă datele de obicei prin intermediul acestui registru. BX poate fi folosit atât ca acumulator cât și ca registru bază pentru calcularea adreselor de memorie. CX poate fi acumulator sau se utilizează ca numărător în execuția instrucțiunilor iterative. DX poate fi folosit ca acumulator sau ca registru numărător pentru adresarea zonelor de memorie în transferuri cu *port*-urile de intrare/ieșire.

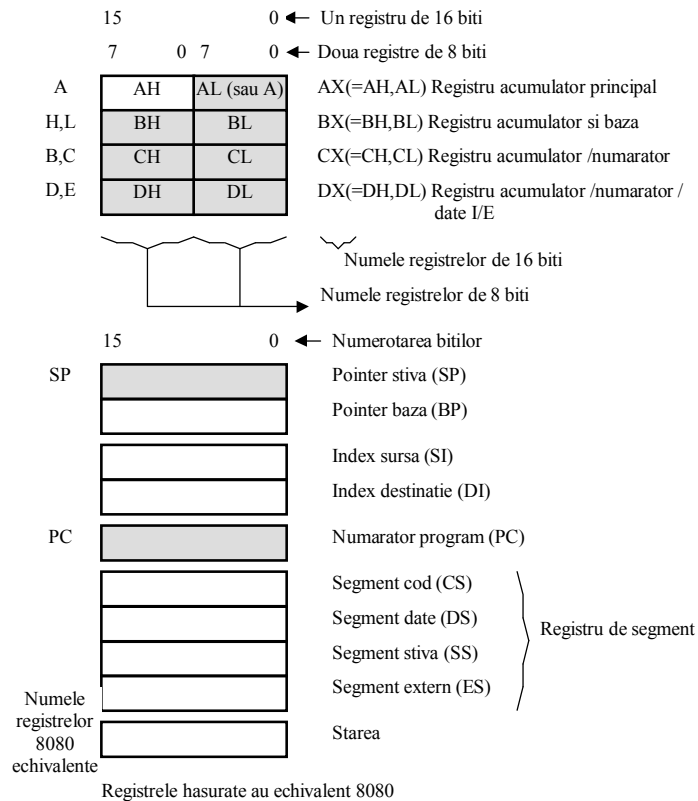


Figura 1.4. Registrele programabile ale microprocesorului 8086

Adresele de memorie pe 20 de biți, generate de 8086, se calculează prin însumarea conținutului unui *registru de segment* cu o *adresă efectivă*. Adresa efectivă se calculează printr-o varietate de moduri de adresare ca și la alte microprocesoare. Registrul de segment selectat este întâi deplasat la stânga cu patru poziții și apoi adunat cu adresa efectivă, generându-se o adresă de 20 de biți:

Conținutul registrului de segment:            SSSS SSSS SSSS SSSS 0000 +

Adresa efectivă de memorie:                    0000 EEEE EEEE EEEE EEEE

Adresa de memorie pe 20 de biți:                SSSM MMMM MMMM MMMM MMMM

Registrele de segment sunt folosite deci ca registre de bază care pot puncta orice locație de memorie aflată la o adresă multiplu de 16. Fiecare registru de segment identifică începutul unei zone de memorie, al unui *segment*, de 64

kocteți. Deoarece 8086 are patru registre de segment în orice moment, din întreaga memorie vor fi selectate numai patru segmente de câte 64 kocteți. Nu se impune nici-o restricție asupra conținutului registrelor de segment, ceea înseamnă că aceste registre identifică numai *originea* segmentelor care pot fi plasate oriunde în spațiul de adresare de 1 Moctet al microprocesorului. Memoria de 1 Moctet nu este deci împărțită în pagini de 64 kocteți, ci adresată pe segmente care pot fi *separate* sau *suprapuse*.

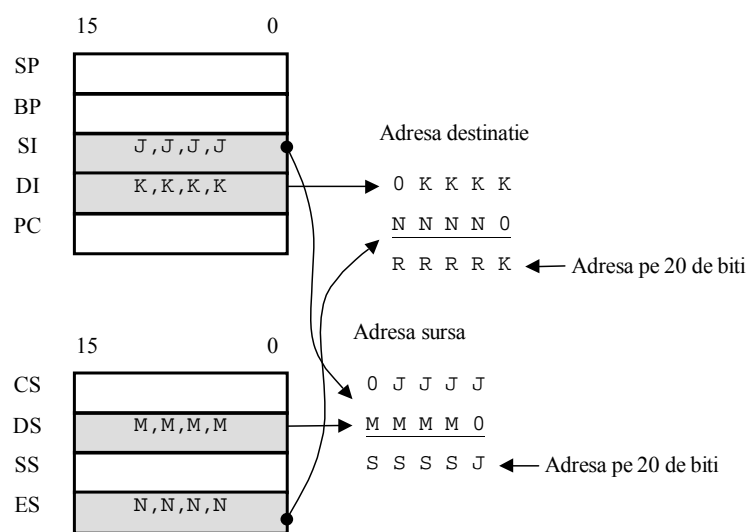


Figura 1.5. Calculul adresei de memorie pentru instrucțiuni pe șiruri de date

Cele patru registre de segment îndeplinesc funcții de adresare diferite: în funcție de tipul accesului la memorie, – cod, date, stivă și extern –, calculul adresei folosește registrul de segment asociat. Astfel la o extragere de instrucțiune, numărătorul de program, PC, este adunat cu conținutul registrului de segment de cod, CS. La orice instrucțiune de lucru cu stiva, de tip Push, Pop, Call sau Return, adresa de memorie se calculează adunând conținutul *pointer*-ului de stivă, SP, cu conținutul registrului de segment de stivă, SS. La instrucțiunile pe șiruri de date pentru calculul adreselor sursă și destinație sunt utilizate registrele index SI și DI împreună cu registrele de segment DS și ES, figura 1.5. Observăm că pentru aceste instrucțiuni este necesar ca șirurile sursă și destinație să fie plasate fiecare în cadrul unui singur spațiu de adresare de 64 kocteți, nu neapărat același. Instrucțiunile care accesează date calculează adresa prin adunarea adresei efective cu conținutul registrelor de segment DS sau SS.

Dăm în tabelul 1.1 o prezentare sintetică a modurilor de calculare a adresei de memorie.

Tabelul 1.1. Calculul adreselor de memorie

Referința de memorie	Registrul de segment	Registrul bază	Registrul index	Deplasament pe 16 biți, fără semn	Deplasament pe 8 biți, semn extern	Fără deplasament
Date	DS (CS, SS, ES) <sup>1</sup>	–	SI	* <sup>2</sup>	*	*
			DI	*	*	*
		BX	SI	*	*	*
			DI	*	*	*
			–	*	*	*
	DS	–	–	*		
	SS (CS, DS, ES) <sup>1</sup>	BP	SI	*	*	*
			DI	*	*	*
–			*	*		
Stivă	SS	SP	–			
Șiruri	DS	–	SI			
	ES	–	DI			
Extragere instrucțiune	CS	PC	–			
Salt	CS	PC	–		*	
Date I/O	DS	DX	–			

Note: 1. Depășirea segmentului permite înlocuirea registrelor de segment DS sau SS cu unul din celelalte registre.

2. \* semnifică deplasamentele care pot fi utilizate la calculul adresei

Vom prezenta întâi opțiunile de adresare a datelor. Cea mai simplă adresare este cea *directă*. În acest caz registrul DS este adunat cu un *offset* sau deplasament pe 16 biți dat de doi octeți din codul obiect al instrucțiunii. Dacă se folosesc registrele index SI sau DI, prin adunarea conținutului unuia din acestea cu conținutul registrului DS se obține o adresare *implicită*. Prin adăugarea deplasamentului, de 16 sau 8 biți, obținem o adresare *directă indexată*. Dacă deplasamentul e pe 8 biți atunci cel mai semnificativ bit, semnul, se extinde pe încă 8 biți, mai semnificativi, pentru a forma un deplasament pe 16 biți, de exemplu:

```

Deplasament:      ...      1010 1101
Semn extins:    1111 1111 1010 1101  sau
Deplasament:      ...      0101 1010
Semn extins:    0000 0000 0101 1010
    
```

Adresa mai poate fi calculată și prin utilizarea unui registru-bază obținându-se o adresare *relativă*. Pentru adresarea relativă a datelor se pot utiliza două registre de bază, BX și BP, adresele găsindu-se în segmentul DS, pentru date, și, respectiv, în segmentul SS, pentru stivă. În primul fel de adresare relativă cu bază a datelor, conținutul registrului BX este adunat la adresa efectivă calculată în unul din modurile exemplificate mai sus. Putem avea deci trei feluri de adresări relative: directe, implicite sau directe indexate. În al doilea fel de adresare, calculul adresei se face adunând la adresa efectivă conținutul registrului de segment SS.

În legătură cu utilizarea registrelor generale ca registre de bază atragem aici atenția asupra unui mod de adresare a dispozitivelor de I/O prin intermediul registrului DX. În acest mod de adresare conținutul registrului DX este plasat pe magistrala de adrese fără a fi adunat cu vreun registru de segment. Acesta este singurul mod de adresare în care conținutul unui registru este scos direct pe magistrală ca adresă fără a fi în prealabil prelucrat în logica de segmentare.

Toate instrucțiunile de salt condiționat utilizează adresarea relativă la programe, cu registru de bază PC, permițându-se astfel generarea unui cod relocabil. Deplasamentul pe 8 biți reprezintă un număr binar cu semn cuprins în gama (-128, +127).

Pentru instrucțiunile de salt necondiționat și de apel sunt folosite trei moduri de adresare: adresare relativă la program, registru de bază PC, cu deplasament pe 8 sau 16 biți; adresare directă; adresare *indirectă*. În adresarea indirectă se utilizează una din formele de adresare descrise mai sus pentru a citi din memorie o nouă adresă de memorie. Există două opțiuni de adresare indirectă: una în care cuvântul de 16 biți citit din memorie e încărcat în PC și instrucțiunea de salt sau apel adresează o locație de memorie din cadrul segmentului CS curent; cealaltă în care se citesc două cuvinte de 16 biți pentru a fi încărcate în PC, respectiv CS și care permite adresarea oricărei locații din spațiul de memorie al microprocesorului.

În figura 1.6 dăm structura registrului de stare al microprocesorului 8086 conținând indicatorii de condiții.

Indicatorul *Transport*, C sau CF, este inversat indicând de fapt un împrumut: după o scădere, realizată de procesor în complement față de 2, acest bit va fi pus pe "1" dacă nu a existat transport la bitul cel mai semnificativ și pus pe "0" dacă a existat un transport.

Indicatorul *Paritate*, P sau PF, este pus pe "1" dacă rezultatul conține un număr par de "1" și pe "0" dacă rezultatul operației are un număr impar de "1"-uri.

Indicatorul *Zero*, Z sau ZF, este standard fiind pus pe "1" dacă rezultatul operației e zero și pe "0" dacă rezultatul nu este zero.

Indicatorul *Semn*, S sau SF, este pus pe "1" dacă rezultatul unei operații este negativ având bitul cel mai semnificativ "1" și pus pe "0" dacă rezultatul este pozitiv.

Indicatorul *Direcție*, D sau DF, determină dacă operațiile pe șiruri vor incrementa sau decremента registrele index. Dacă *Direcție* este "1" atunci conținuturile registrelor SI și DI vor fi decrementate, șirurile fiind accesate începând cu adresele cele mai mari. Dacă indicatorul este "0" conținutul registrelor SI și DI va fi incrementat, șirurile fiind accesate începând cu adresele cele mai mici.

Indicatorul *Validare/Invalidare Întrerupere*, I sau IF, trebuie să fie "1" pentru a valida întreruperile și "0" pentru a le invalida (cu excepția NMI).

Indicatorul *Derută*, T sau TF, face ca microprocesorul 8086 să intre în modul de lucru "pas-cu-pas" și este strâns legat de sistemul de întreruperi (vezi §1.4.5.1 și §2.3.6).

Indicatorul *Depășire*, O sau OF este, de asemenea, legat de sistemul de întreruperi și de execuția instrucțiunii INTO.

Indicatorul *Transport Auxiliar*, A sau AF indică, după o operație, un transport din bitul 3 în 4.

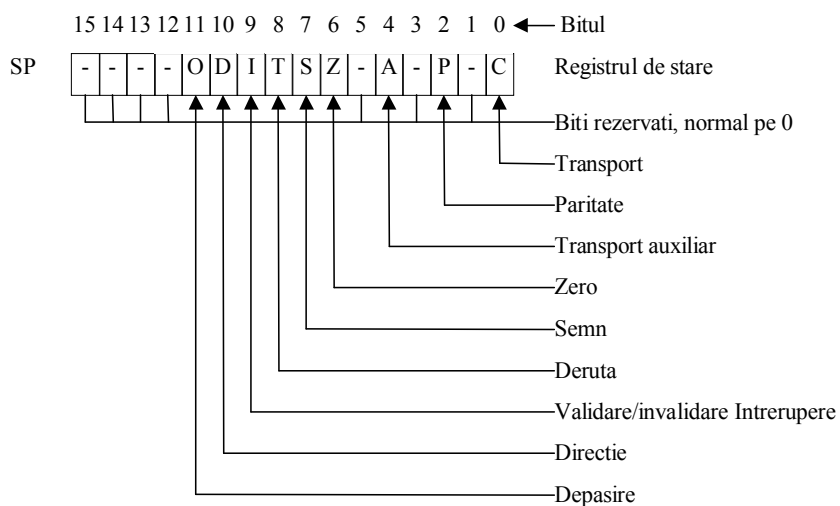


Figura 1.6. Registrul de stare



## 1.2. CONEXIUNILE EXTERNE

Microprocesoarele 8086 și 8088 sunt fabricate în capsule DIL cu 40 de conexiuni externe. Aceste conexiuni sunt indicate în figurile 1.7 și 1.8. Vom descrie în continuare semnificațiile lor.

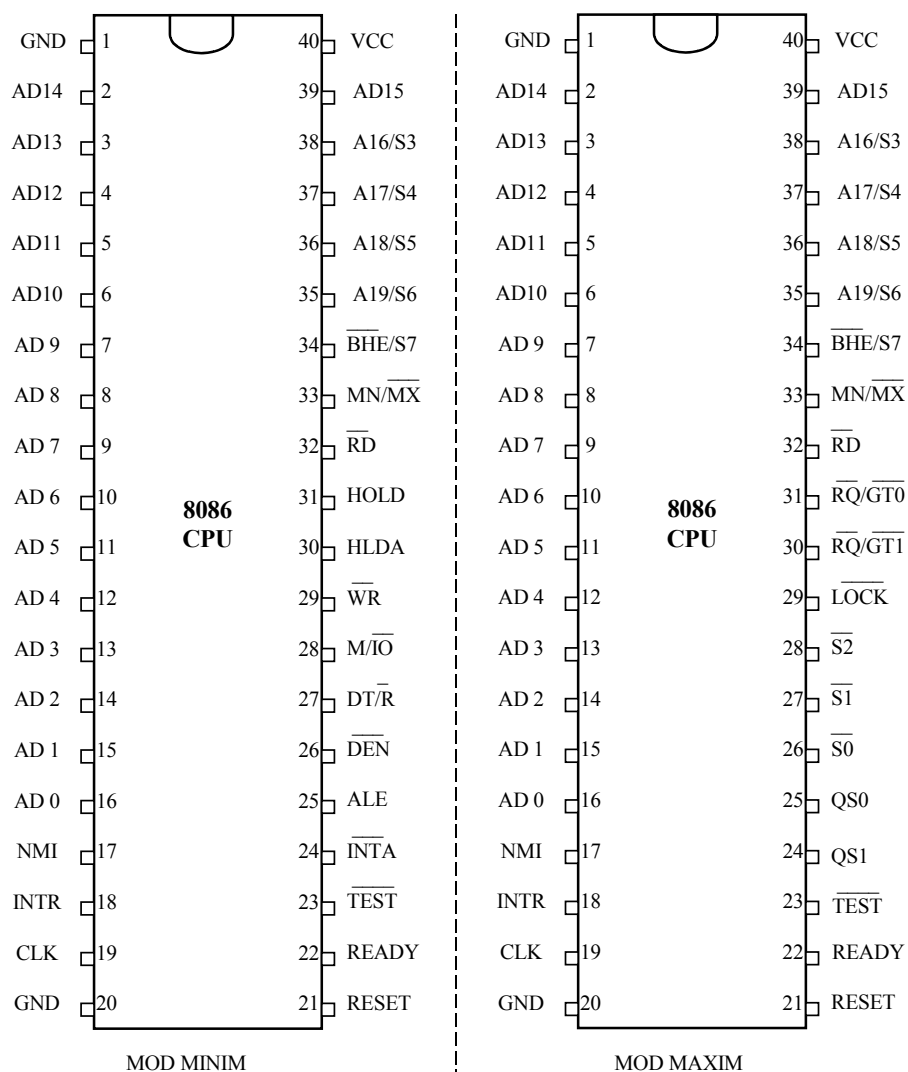


Figura 1.7. Conexiunile externe ale microprocesorului 8086

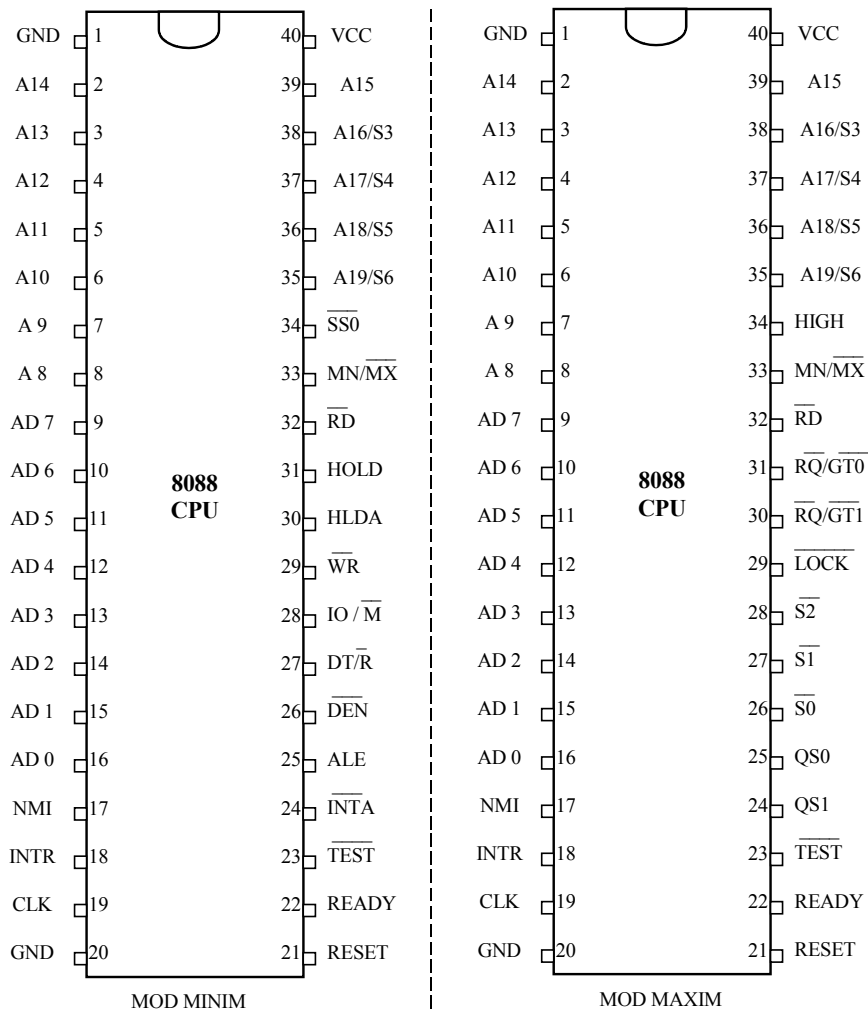


Figura 1.8. Conexiunile externe ale microprocesorului 8088

## 1.2.1. CONEXIUNILE EXTERNE ALE MICROPROCESORULUI 8086

### 1.2.1.1. Semnalele comune modurilor de lucru minim și maxim

AD15÷AD0, *Address/Data Bus*, magistrala de adrese/date, intrări/ieșiri 3-stări active pe "1". Pe aceste linii sunt multiplexate în timp adresele de memorie (cei mai puțin semnificativi 16 biți) sau de I/O, în starea T1, și

magistrala de date, în stările T2, T3, TW, T4<sup>1</sup>. Multiplexarea magistralelor a fost impusă de necesitatea de a împacheta microprocesorul într-o capsulă de 40 de conexiuni. AD0, ca adresă A0, are pentru selecția octetului de date mai puțin semnificativ, D7÷D0, aceeași funcție ca și semnalul  $\overline{\text{BHE}}$ , descris mai jos; pentru octetul mai semnificativ, D15÷D8, A0 este "0" în timpul stării T1, la transferarea unui octet de date pe porțiunea mai puțin semnificativă a magistralei, în operații cu memoria sau cu dispozitivele de I/O. AD15÷AD0 sunt trecute în starea a treia, de impedanță înaltă, în timpul operațiilor de achitare a întreruperii și a cererii de preluare a magistralei.

A19/S6÷A16/S3, *Address/Status*, adrese/stări, ieșiri 3-stări. În timpul stării T1 reprezintă cei mai semnificativi biți de adresă în operațiile cu memoria; în operațiile de I/O aceste semnale sunt "0". Pentru ambele tipuri de operații, pe timpul stărilor T2, T3, TW și T4, la aceste ieșiri se găsește o parte din starea microprocesorului: S6=0, indicând că 8086 accesează magistrala, S5=I, *Validare/Invalidare Întrerupere*, actualizat la începutul fiecărei perioade a ceasului CLK, iar S4 și S3 precizând ce registru de relocare, DS, CS, SS sau ES, va fi utilizat în acel moment pentru accesarea datelor. Semnificația biților de stare S4 și S3 este următoarea:

A17/S4	A16/S3	Semnificație
0	0	Date externe (relativ la segmentul din ES)
0	1	Stivă (relativ la segmentul din SS)
1	0	Cod sau nimic (relativ la segmentul din CS sau o valoare în lipsă "0")
1	1	Date (relativ la segmentul din DS)

Informația de stare este necesară în acțiunile de diagnosticare a structurilor realizate în jurul procesorului 8086. S4 și S3 mai pot fi utilizați pentru selecția bancurilor de memorie asociate fiecare câte unui registru de segment. Această tehnică permite partajarea memoriei în funcție de segment în vederea expandării ei peste spațiul de adresare directă de 1 Moctet. De asemenea, în acest fel, se asigură o posibilitate de protecție în cazul operațiilor eronate de scriere prin suprapunere în două segmente și distrugerea informației în unul dintre segmente. Conexiunile A19/S6÷A16/S3 sunt trecute în starea a treia în timpul operației de achitare a unei cereri de preluare a magistralei microprocesorului.

$\overline{\text{BHE}}/S7$ , *Bus High Enable/Status*, validare octet mai semnificativ/stare. Ieșire 3-stări. În timpul stării T1 ieșirea  $\overline{\text{BHE}}$  are funcția de a selecta transferul datelor pe porțiunea mai semnificativă a magistralei de date, biții D15÷D8, fiind "0" în T1 timpul ciclilor de citire, scriere și întrerupere.  $\overline{\text{BHE}}$  poate fi utilizat la selecția dispozitivelor de I/O pe 8 biți conectate pe porțiunea mai semnificativă

<sup>1</sup> Descrierea stărilor T este dată în §1.4.1

a magistralei de date. Bitul de stare S7 este validat în timpul stărilor T2, T3 și T4. Conexiunea este trecută în starea a treia în timpul unei operații de preluare a magistralei.

$\overline{MN} / \overline{MX}$ , *Minimum/Maximum*, comanda modului de lucru minim/maxim. Intrare. Dacă  $\overline{MN} / \overline{MX}$  este conectat la "0" microprocesorul tratează conexiunile 24÷31 în modul maxim, pentru care un circuit specializat pentru comanda magistralei, 8288, interpretează biții de stare  $\overline{S2}$ ,  $\overline{S1}$  și  $\overline{S0}$  pentru a genera semnale de comandă compatibile cu standardul de interfață Multibus. Pentru  $\overline{MN} / \overline{MX}$  conectat la +5V, modul minim, 8086 generează singur, la conexiunile 24÷31, semnalele de comandă. În §1.4.2 și §1.4.3 se dau două exemple de utilizare a procesorului 8086 în modurile minim și maxim.

$\overline{RD}$ , *Read*, citire. Ieșire 3-stări activă pe "0". Comandă de citire indicând că procesorul efectuează un ciclu de citire memorie sau I/O, funcție de starea conexiunii  $\overline{S2}$  ( $M / \overline{IO}$ ).  $\overline{RD}$  este activat pe "0" în timpul stărilor T2, T3 și TW ale oricărui ciclu de citire, după trecerea în starea a treia a magistralei locale a microprocesorului. Conexiunea este trecută în starea de impedanță înaltă în timpul unei operații de preluare a magistralei.

READY, gata. Intrare activă pe "1". Reprezintă, în timpul unei operații de scriere sau citire, un semnal de achitare din partea memoriei sau dispozitivului de I/O adresat, certificând validitatea transferului de date cu microprocesorul. Semnalul de achitare emis de memorie sau de I/O trebuie să fie sincronizat pentru a putea fi utilizat în mod corect de procesor prin satisfacerea timpilor de *set-up* și *hold*. Această sincronizare se face cu ajutorul unui alt circuit specializat necesar în sistemele cu 8086: generatorul de ceas 8284.

INTR, *Interrupt Request*, cerere întrerupere. Intrare activă pe nivel "1". Procesorul eșantionează această intrare sincronizată intern în ultimul ciclu de ceas al fiecărei instrucțiuni, pentru a intra, dacă INTR=1, într-o operație de achitare a întreruperii. La achitarea întreruperii se va apela o subrutină pe baza unei tabele de vectori localizate în memoria sistemului. Întreruperea poate fi mascată intern prin program punând pe "0" bitul de validare a întreruperii.

$\overline{TEST}$ , intrare activă pe "0". Această intrare, sincronizată intern cu frontul pozitiv al fiecărui impuls de ceas CLK, este examinată de instrucțiunea WAIT: dacă intrarea este "0", microprocesorul își continuă execuția, dacă este "1", va aștepta intrând în așa-numita stare inactivă, TI, executată în general de UIM atunci când nu poate să execute un ciclu de magistrală (vezi §1.4.1).

NMI, *Non-Maskable Interrupt*, întrerupere nemascabilă. Intrare activă pe front pozitiv producând la sfârșitul instrucțiunii în curs o întrerupere de tipul 2 (vezi §1.4.5.1). Achitarea acestui tip de întrerupere conduce la apelarea unei subrutine pornind de la o tabelă de vectori aflată în memorie. Intrarea este sincronizată intern și nu poate fi mascată prin program.

RESET, inițializare. Intrare activă pe "1". Semnalul aplicat la această conexiune este sincronizat intern și, dacă este activ cel puțin patru perioade de ceas, conduce la terminarea activității curente a microprocesorului și inițializarea execuției după revenirea lui pe "0".

CLK, *Clock*, ceas. Intrare ce asigură funcționarea sincronă a microprocesorului și a controlorului de magistrală. Factorul de umplere al ceasului este 33%.

### 1.2.1.2. Semnalele specifice modului de lucru minim

Așa cum am menționat în §1.1, una dintre caracteristicile cele mai interesante ale microprocesoarelor 8086/8088 este și posibilitatea de a selecta configurația de bază a mașinii, modul de lucru cel mai potrivit aplicației, prin conectarea la VCC sau GND a intrării MN /  $\overline{MX}$ .

În modul minim, 8086 permite realizarea de unități centrale mai restrânse ca volum, satisfăcând zona aplicațiilor pe 16 biți mici și medii. În acest mod de lucru procesorul își menține capacitatea de adresare a memoriei de 1M, a I/O de 64k, precum și magistrala de 16 biți, generând conexiunile 24÷31 direct, fără ajutorul unui circuit de comandă specializat. Sunt toate semnalele necesare manipulării magistralelor, DT /  $\overline{R}$ ,  $\overline{DEN}$ , ALE, M /  $\overline{IO}$ , semnalele de comandă pentru operațiile de citire, scriere, achitarea întreruperii,  $\overline{RD}$ ,  $\overline{WR}$ ,  $\overline{INTA}$ , precum și semnalele HOLD, HLDA necesare operațiilor simple de transfer cu acces direct implementate cu ajutorul controloarelor DMA obișnuite, de exemplu 8257 (vezi și [9]).

În continuare, vom descrie, pe scurt, semnificația conexiunilor 24÷31 ale circuitului 8086 în modul de lucru minim.

M /  $\overline{IO}$ , *Memory/Input-Output*, memorie/intrare-ieșire. Ieșire 3-stări utilizată pentru a distinge accesesele la memorie, M /  $\overline{IO}$ =1, de cele la dispozitivele de I/O, M /  $\overline{IO}$ =0. Ieșirea este validată în starea T4 precedentă unui ciclu de magistrală rămânând activă până la sfârșitul stării T4 a ciclului curent. Ieșirea, echivalentă cu  $\overline{S2}$  în modul maxim, este trecută în starea a treia în ciclul de achitare a cererii de preluare a magistralei microprocesorului.

$\overline{WR}$ , *Write*, scriere. Ieșire 3-stări activă pe "0" ce indică faptul că procesorul execută un ciclu de scriere a memoriei sau a I/O. Ieșirea este activă în stările T2, T3 și TW, fiind trecută în starea de impedanță înaltă în ciclul de achitare a cererii de preluare a magistralei microprocesorului.

$\overline{INTA}$ , *Interrupt Acknowledge*, achitare întrerupere. Ieșire activă pe "0" utilizată pentru generarea unui semnal de eșantionare a citirii în ciclul de achitare a întreruperii. Ieșirea  $\overline{INTA}$  este activă în timpul stărilor T2, T3 și TW ale fiecărui ciclu de întrerupere.

ALE, *Address Latch Enable*, validarea *latch*-urilor de adrese. Ieșire activă pe "1" utilizată pentru memorarea adreselor generate de microprocesor pe magistrala multiplexată locală în *latch*-urile de adrese ale sistemului.

DT /  $\overline{R}$ , *Data Transmit/Receive*, transmisie/recepție date. Ieșire utilizată pentru comanda direcției *transceiver*-elor de date: transmisie pentru DT /  $\overline{R}$  =1, recepție pentru DT /  $\overline{R}$  =0, sensul fiind raportat la microprocesor. Ieșirea este trecută în starea a treia în timpul ciclilor de achitare a cererii de preluare a magistralei.

$\overline{DEN}$ , *Data Enable*, validare date. Ieșire 3-stări activă pe "0" folosită pentru validarea *transceiver*-elor de date. Este "0" în timpul operațiilor cu memoria și cu dispozitivele de I/O precum și în timpul ciclilor de achitare întrerupere ( $\overline{INTA}$ ), pentru citiri (și  $\overline{INTA}$ ) de la mijlocul lui T2 la mijlocul lui T4, iar pentru scrieri de la începutul lui T2 la mijlocul lui T4. Ieșirea este trecută în starea de impedanță mare în timpul ciclilor de achitare a unei cereri de preluare a magistralei.

HOLD, cerere de preluare a magistralei microprocesorului. Intrare activă pe "1".

HLDA, achitare a cererii de preluare a magistralei. Ieșire activă pe "1".

După recepționarea unei cereri HOLD microprocesorul va activa, la mijlocul stării T1, semnalul de achitare HLDA, trecând în același timp în starea a treia magistrala locală (AD15÷AD0, A19/S6÷A16/S3) și ieșirile de comandă ( $\overline{RD}$ ,  $\overline{BHE}/S7$ , M /  $\overline{IO}$ , DT /  $\overline{R}$ ,  $\overline{WR}$ ,  $\overline{DEN}$ ). La terminarea operației de transfer direct dispozitivul care a activat semnalul HOLD îl va dezactiva trecându-l pe "0", după care procesorul va dezactiva la rândul lui semnalul de achitare HLDA. Magistrala locală și ieșirile de comandă vor fi activate când microprocesorul va avea nevoie să execute un ciclu de magistrală. Menționăm că intrarea HOLD nu este asincronă și că, pentru asigurarea timpului de *set-up*, este necesară sincronizarea externă.

### 1.2.1.3. Semnalele specifice modului de lucru maxim

În acest mod de lucru, obținut prin legarea conexiunii MN /  $\overline{MX}$  la GND, 8086 permite implementarea configurațiilor multiprocesor și/sau cuplarea procesoarelor pentru extinderea setului de instrucțiuni, a *coprocesoarelor*. În modul maxim este necesară folosirea controlorului de magistrală 8288 pentru ca, prin redefinirea conexiunilor 24÷31, să se obțină această creștere a performanțelor.

$\overline{RQ} / \overline{GT1}, \overline{RQ} / \overline{GT0}$ , *Request/Grant Bus Access Control*, comenzi de cerere/achitare a accesului la magistrala microprocesorului. Intrări/ieșiri active pe "0" folosite de alte dispozitive de tip *master*, aflate pe magistrala locală, pentru a forța microprocesorul să elibereze această magistrală la sfârșitul ciclului în curs. Aceste conexiuni sunt prevăzute cu rezistențe interne legate la VCC, ceea ce permite să fie lăsate în gol.  $\overline{RQ} / \overline{GT0}$  este prioritară față de  $\overline{RQ} / \overline{GT1}$ .

$\overline{LOCK}$ , blocare. Ieșire 3-stări activă pe "0". Indică celorlalte dispozitive de tip *master* aflate pe magistrala locală a microprocesorului, că nu pot prelua această magistrală atâta timp cât  $\overline{LOCK}=0$ . Ieșirea este activată de o instrucțiune cu prefix  $\overline{LOCK}$  rămânând activă încă o perioadă de ceas după execuția acestei instrucțiuni cu prefix. Ieșirea este trecută în starea a treia pe timpul ciclurilor de achitare a unei cereri de preluare a magistralei.

$\overline{S2} \div \overline{S0}$ , *Status*, stare. Ieșiri 3-stări active în timpul stărilor T4, T1 și T2 și inactive, în starea pasivă (1,1,1) în timpul stărilor T3 și TW dacă semnalul de la conexiunea  $\overline{READY}$  este "1". Starea  $\overline{S2} \div \overline{S0}$  este folosită de controlorul 8288 pentru a genera semnalele de comandă necesare în operațiile de acces la memorie sau la I/O. Orice schimbare a acestor semnale în timpul stării T4 indică începutul unui ciclu de magistrală, revenirea în starea pasivă în T3 sau TW indicând sfârșitul ciclului de magistrală. Semnificația ieșirilor  $\overline{S2} \div \overline{S0}$  este următoarea:

$\overline{S2}$	$\overline{S1}$	$\overline{S0}$	Semnificație
0	0	0	Achitare întrerupere
0	0	1	Citire <i>port</i> I/O
0	1	0	Scriere <i>port</i> I/O
0	1	1	Oprire ( <i>Halt</i> )
1	0	0	Citire instrucțiune
1	0	1	Citire memorie
1	1	0	Scriere memorie
1	1	1	Stare pasivă (nici-un ciclu de magistrală)

Conexiunile  $\overline{S2} \div \overline{S0}$  sunt trecute în starea a treia în timpul ciclurilor de achitare a unei cereri de preluare a magistralei.

QS1, QS0, *Queue Status*, starea cozii de instrucțiuni. Ieșiri validate în ciclul de ceas următor unei operații cu stiva. Aceste ieșiri permit urmărirea din exterior a execuției instrucțiunilor având, atunci când sunt validate, următoarea semnificație:

QS1	QS0	Semnificație
0	0	Nici o operație
0	1	Extragerea primului octet al codului-operație
1	0	Coadă goală (nici un octet în coadă)
1	1	Extragerea octetului următor al codului-operație

### 1.2.2. CONEXIUNILE EXTERNE ALE MICROPROCESORULUI 8088

Precizăm în acest subcapitol numai conexiunile a căror semnificație diferă de cea a microprocesorului 8086, prezentată în §1.3.1.

#### 1.2.2.1. Semnalele comune modurilor de lucru minim și maxim

$AD7\div AD0$ , *Address/Data Bus*, magistrală de adrese/date. Diferă de magistrala multiplexată a procesorului 8086 numai prin mărimea de 8 biți față de 16.

$A15\div A8$ , *Address Bus*, magistrală de adrese. Ieșiri 3-stări reprezentând biții 8÷15 ai adresei. Spre deosebire de  $AD15\div AD8$ , acești biți nu mai sunt multiplexați nefiind necesară memorarea lor în *latch*-uri cu ajutorul semnalului ALE.  $A15\div A8$  rămân valizi pe durata întregului ciclu de magistrală, stările  $T1\div T4$ .

Conexiunile  $MN / \overline{MX}$ ,  $\overline{RD}$ ,  $\overline{READY}$ ,  $\overline{INTR}$ ,  $\overline{TEST}$ ,  $\overline{NMI}$ ,  $\overline{RESET}$ ,  $CLK$  au aceleași semnificații cu cele ale microprocesorului 8086.

#### 1.2.2.2. Semnalele specifice modului de lucru minim

$IO / \overline{M}$ , *Input-Output/Memory*, intrare-ieșire/memorie. Ieșire 3-stări semnificând pentru  $IO / \overline{M}=1$  un ciclu de I/O iar pentru  $IO / \overline{M}=0$  un ciclu de memorie. Activarea acestei ieșiri respectă aceleași relații de timp ca și ieșirea  $M / \overline{IO}$  a lui 8086.

$\overline{SS0}$ , *S0 Status*, starea  $S0$ . Ieșire 3-stări echivalentă cu bitul de stare  $S0$  din modul maxim. Împreună cu  $IO / \overline{M}$  și  $DT / \overline{R}$ , ultima conexiune având aceeași semnificație ca și la 8086, codifică starea ciclului de magistrală în curs:



IO / M	DT / R	SS0	Semnificație
1	0	0	Achitare întrerupere
1	0	1	Citire <i>port</i> I/O
1	1	0	Scriere <i>port</i> I/O
1	1	1	Opre (Halt)
0	0	0	Citire instrucțiune
0	0	1	Citire memorie
0	1	0	Scriere memorie
0	1	1	Stare pasivă

Celelalte conexiuni specifice modului minim de lucru al microprocesorului 8088,  $\overline{WR}$ ,  $\overline{INTA}$ , ALE,  $\overline{DEN}$ , HOLD și HLDA, rămân cu aceleași funcții ca la 8086.

### 1.2.2.3. Semnalele specifice modului de lucru maxim

Semnalele specifice modului de lucru maxim al microprocesorului 8088,  $\overline{S2}$ ,  $\overline{S1}$ ,  $\overline{S0}$ ,  $\overline{RQ}/\overline{GT1}$ ,  $\overline{RQ}/\overline{GT0}$ ,  $\overline{LOCK}$ , QS1 și QS0 sunt aceleași cu ale microprocesorului 8086 în modul maxim. Conexiunea  $\overline{SS0}$  este "1" în acest mod de lucru.

## 1.3. FUNCȚIONAREA MICROPROCESORULUI 8086

Vom descrie în cele ce urmează funcționarea microprocesorului 8086 din punct de vedere al utilizării lui hardware.

### 1.3.1. CICLII DE MAGISTRALĂ AI MICROPROCESORULUI 8086. PREZENTARE GENERALĂ

8086 comunică cu exteriorul prin intermediul unei magistrale de comenzi și a unei magistrale multiplexate în timp de adrese, stări și date. Așa cum am menționat, tehnica multiplexării în timp a permis, atunci când a fost dezvoltat circuitul, utilizarea cea mai eficientă a celor 40 de conexiuni exterioare ale capsulei în care s-a împachetat microprocesorul. Această magistrală multiplexată, denumită și magistrală locală, poate fi amplificată direct și condusă în sistem, *latch*-area, memorarea adresei făcându-se distribuit în modulele de memorie sau de I/O. O altă abordare este cea a demultiplexării unice a adreselor și datelor lângă procesor, cu ajutorul unui singur grup de

*latch*-uri pentru adrese și a *transceiver*<sup>2</sup>-elor pentru date. Microprocesorul extrage cod sau transferă date executând așa numiții *cicli de magistrală*. Un ciclu de magistrală poate fi definit ca un eveniment asincron în care întâi se validează o adresă, a unei locații de memorie sau a unui dispozitiv periferic, apoi se generează fie un semnal de comandă a citirii, pentru capturarea datelor de la dispozitivul adresat, fie un semnal de comandă a scrierii asociat cu emiterea datelor ce trebuie transmise dispozitivului adresat. Observăm întrebuințarea termenului ciclu de magistrală în locul mai vechiului termen *ciclu mașină* folosit în descrierea funcționării unor procesoare pe 8 biți ca 8080 sau Z80. Ciclii de magistrală trebuie înțeleși ca grupuri de perioade de ceas reprezentând activitatea externă a microprocesorului fără a avea o legătură explicită și ordonată cu execuția curentă a unei instrucțiuni așa cum aveau ciclii mașinii (vezi de exemplu ciclii  $M_1, M_2, M_3$  ai unui ciclu-instrucțiune Z80 [9]). În figura 1.9 este înfățișată diagrama de bază reprezentând ciclii de citire și scriere pe care îi execută microprocesorul pe magistrala sa de comunicație cu exteriorul. Această diagramă va fi detaliată în capitolele următoare pentru modurile de lucru minim și maxim.

Orice ciclu de magistrală, TCY, are cel puțin patru perioade de ceas, numite *stări T*. În timpul primei stări T, T1, procesorul validează adresa A19÷A0 pe magistrala multiplexată. Tot acum se generează semnalul ALE, de către 8086 sau de controlorul de magistrală 8288, în funcție de modul de lucru. Acest semnal servește memorării adresei în circuite de tip *latch*, de exemplu 74LS373 sau 8282, pe frontul negativ al acestui impuls garantându-se validitatea adresei. A doua stare, T2, este destinată schimbării direcției magistralei, microprocesorul invalidând adresa și, pentru un ciclu de citire, trecând magistrala, cei mai puțin semnificativi 16 biți, în starea a treia, iar pentru un ciclu de scriere validând datele. *Transceiver*-ele de date sunt validate, cu ajutorul semnalului DEN, în T1 sau T2, în funcție de configurația sistemului și direcția transferului. Tot în T2 sunt generate și comenzile de citire,  $\overline{RD}$ , scriere,  $\overline{WR}$  sau achitare întrerupere,  $\overline{INTA}$ , precum și starea S7÷S3 în locul biților de adresă A19÷A16 și a semnalului  $\overline{BHE}$ .

În T3, pentru un ciclu de scriere, microprocesorul menține biții de stare și datele ce trebuie transmise, iar în cazul unui ciclu de citire 8086 eșantionează datele păstrând  $\overline{RD}=0$ . Dacă memoria sau dispozitivul de I/O selectat nu sunt capabile să asigure transferul datelor până în T4 ele trebuie să avertizeze de acest lucru procesorul poziționând, înainte de T3, semnalul READY pe "0". În acest fel ciclul de magistrală se prelungește prin introducerea de către microprocesor, între T3 și T4, a unor stări de așteptare TW al căror număr depinde de timpul necesar dispozitivului adresat pentru a asigura transferul. Activitatea procesorului pe magistrală în timpul unei stări TW este aceeași ca în

<sup>2</sup> Circuite amplificatoare bidirecționale cum sunt, de exemplu, 8286 sau 74LS245

T3. Ieșirea din stările TW se face la inițiativa dispozitivului, după trecerea timpului necesar asigurării corectitudinii transferului de date, prin trecerea semnalului READY pe "1".

Un ciclu de magistrală se termină cu starea T4 prin dezactivarea comenzilor și eliberarea magistralei locale.

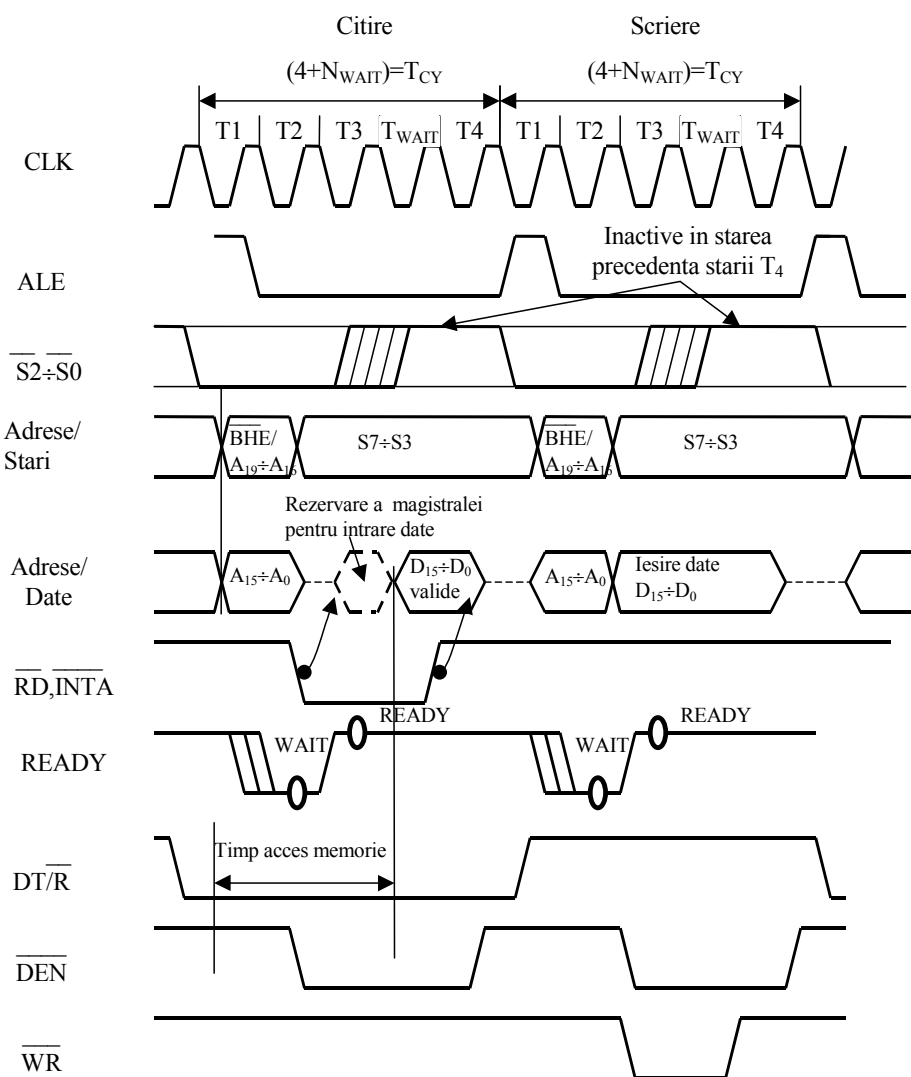


Figura 1.9. Ciclii de magistrală ai microprocesorului 8086. Diagrama de bază

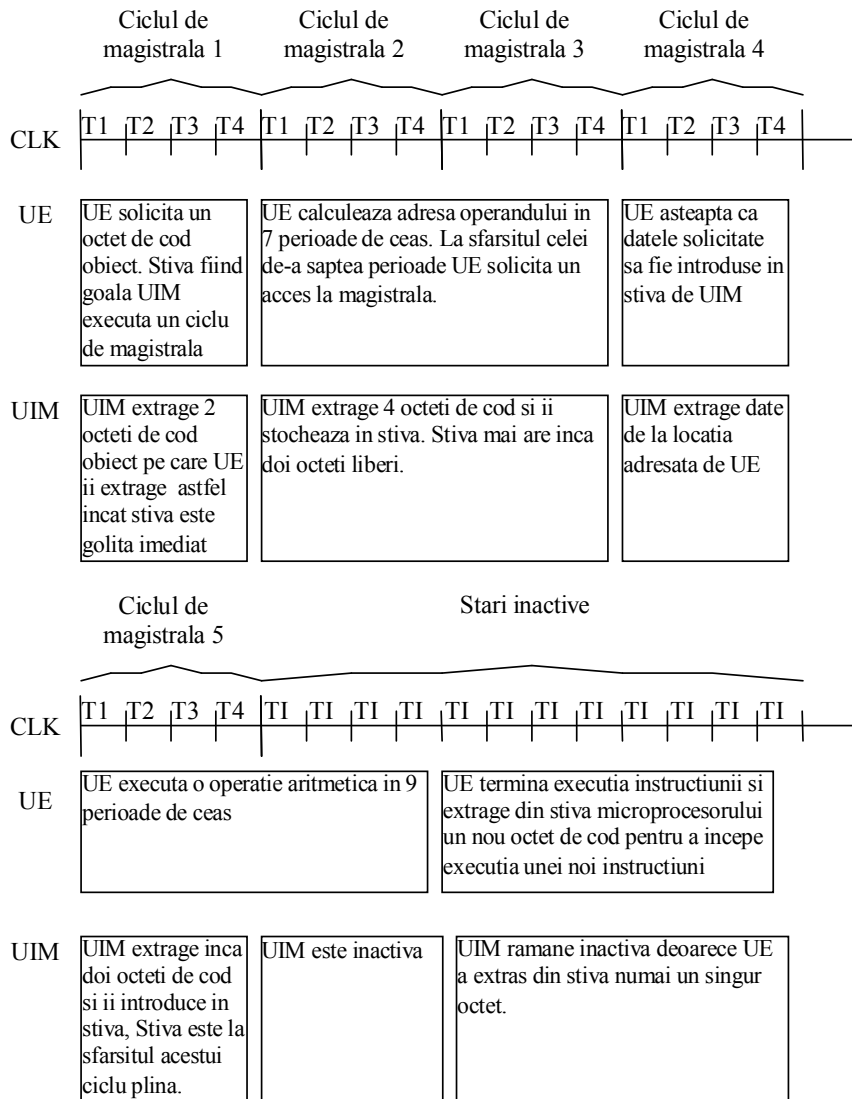


Figura 1.10. Un exemplu de funcționare suprapusă a celor două blocuri funcționale ale microprocesorului 8086

Între ciclii de magistrală, execuții de unitatea centrală numai atunci când trebuie transferate instrucțiuni sau operanzi la memorie sau dispozitivele de I/O, pot apărea așa numitele stări inactive, TI, în timpul cărora procesorul

execută operații interne. Dacă ciclul precedent a fost o scriere, 8086 va menține în timpul stării TI informația de stare  $S7 \div S3$  din ciclul de magistrală precedent și datele anterior emise pe magistrala multiplexată. Atunci când operația precedentă a fost o citire, în afara menținerii stării  $S7 \div S3$ , microprocesorul va lăsa în TI magistrala de date în starea de impedanță înaltă.

Datorită modului de lucru suprapus al celor două blocuri funcționale, UE și UIM, activitatea exterioară a microprocesorului nu va apărea, ca în cazul mașinilor din generațiile anterioare, de exemplu Z80, ca o succesiune de operații de extragere a codului, de *fetch*-uri, și de transferuri corespunzătoare de date cu memoria sau dispozitivele de I/O. La 8086 extragerea codului și transferurile de operanzi asociate unei instrucțiuni pot fi separate prin ciclul de magistrală executat de UIM în scopul umplerii stivei interne de 6 octeți. De asemenea, pot apărea desincronizări, întârzieri, între citirea și începutul execuției unei instrucțiuni. În figura 1.10, [4], se exemplifică modul de lucru suprapus al celor două blocuri funcționale ale microprocesorului. Observăm că UE este activă în timpul execuției instrucțiunilor și inactivă atunci când așteaptă codul obiect sau date pe care nu le poate obține decât prin intermediul unor cicluri de magistrală executate de UIM. Activitatea UE se desfășoară în secvențe cu un număr variabil de perioade de ceas, negrupate, așa cum am spus mai sus, în ciclul mașină de lungime fixă. Pe de altă parte, UIM grupează perioadele de ceas în cicluri de magistrală, dar numai când microprocesorul execută un transfer de cod sau date cu exteriorul. În restul timpului UIM rămâne inactivă.

### 1.3.2. FUNCȚIONAREA MICROPROCESORULUI ÎN MODUL MINIM

Conectarea intrării  $MN / \overline{MX}$  a microprocesorului 8086 la +5V permite selectarea semnificației funcționale a grupului de conexiuni externe 24÷31 ale circuitului în vederea utilizării lui în modul minim, pentru realizarea de configurații simple sau medii, cu o singură unitate centrală. În figura 1.11 este prezentată o astfel de configurație organizată în jurul lui 8086 funcționând în modul minim. Cu linie continuă sunt desenate circuitele și semnalele necesare pentru alcătuirea unei structuri foarte simple. Cu linie punctată sunt indicate circuitele și semnalele suplimentare necesare pentru implementarea unor structuri de complexitate medie.

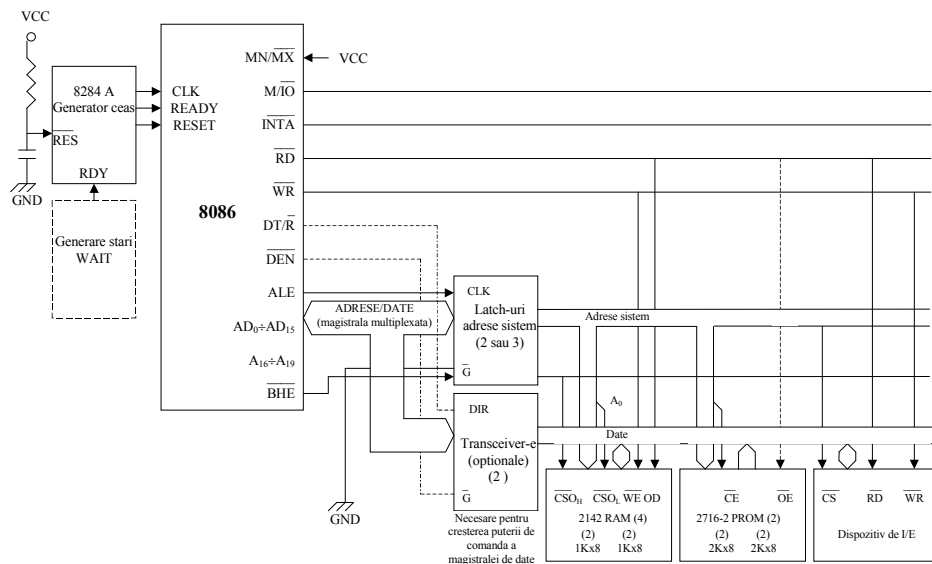


Figura 1.11. Microprocesorul 8086 în modul minim

În acest subcapitol vom interpreta diagramele de timp corespunzătoare operațiilor externe, de magistrală, ale microprocesorului 8086 lucrând în modul minim. În analizele de timp care urmează ne vom referi la figura 1.12 unde sunt date diagramele detaliate de catalog, pentru modul de lucru minim și la tabelele 1.2 și 1.3 cu valorile parametrilor asociați acestor diagrame [6]. În funcționarea pe magistrală a lui 8086 se pot pune în evidență, în modul minim ca și în modul maxim, următoarele secvențe de timp principale: adresarea, ciclul de citire, ciclul de scriere, achitarea întreruperii, introducerea stărilor de așteptare, preluarea magistralei.

### 1.3.2.1. Adresarea

Adresarea memoriei sau a dispozitivelor de I/O se face prin demultiplexarea în timp a informațiilor de pe magistrala locală a microprocesorului, cu ajutorul semnalului ALE destinat memorării adreselor în *latch*-urile de adrese ale sistemului. În operația de adresare vom analiza întâi raportul între adrese și ALE apoi întârzierea adreselor până la apariția lor pe magistrala demultiplexată a sistemului.

Tabelul 1.2. Parametrii 8086 pentru modul minim. Cerințe de timp

Parametru	Semnificație	Valoare minimă	Valoare maximă	Condiții de test și observații
TCLCL	Perioada ceasului CLK	200ns	500ns	
TCLCH	Timpul cât CLK=0	118ns		
TCHCL	Timpul cât CLK=1	69ns		
TCH1CH2	Frontul crescător al ceasului CLK		10ns	Măsurat între valorile 1,0V și 3,5V ale semnalului CLK
TCL2CL1	Frontul descrescător al ceasului CLK		10ns	Măsurat între valorile 3,5V și 1,0V ale semnalului CLK
TDVCL	Timpul de stabilizare, de <i>set-up</i> , al datelor la citire	30ns		
TCLDX	Timpul de menținere, de <i>hold</i> , al datelor la citire	10ns		
TR1VCL	Timpul de <i>set-up</i> al semnalului RDY la intrarea în circuitul 8284A măsurat față de frontul descrescător al lui CLK	35ns		Acest timp este necesar pentru semnalul asincron RDY pentru a garanta recunoașterea lui în următoarea perioadă a lui CLK
TCLR1X	Timpul de <i>hold</i> al semnalului RDY la intrarea în 8284A	0ns		
TRYHCH	Timpul de <i>set-up</i> al semnalului READY la intrarea în 8086	118ns		
TCHRYH	Timpul de <i>hold</i> al semnalului READY la intrarea în 8086	30ns		
TRYLCL	Timpul necesar ca intrarea READY să fie considerată inactivă față de începutul stării T3	-8ns		READY mai poate deveni inactiv maximum 8ns în T3 transformând această stare într-o stare de așteptare TW
THVCH	Timpul de <i>set-up</i> al semnalului HOLD	35ns		
TINVCH	Timpul de <i>set-up</i> al semnalelor INTR, NMI, $\overline{\text{TEST}}$	30ns		Timp necesar pentru ca aceste semnale asincrone să fie recunoscute în următoarea perioadă a ceasului CLK
TILIH	Frontul de creștere, pozitiv, al semnalelor de intrare (cu excepția lui CLK)		20ns	De la 0,8V la 2,0V
TIHIL	Frontul descrescător, negativ, al semnalelor de intrare (exceptând CLK)		12ns	De la 2,0V la 0,8V

Tabelul 1.3. Parametrii 8086 pentru modul minim. Răspunsuri în timp

Parametru	Semnificație	Valoare minimă	Valoare maximă	Condiții de test și observații
TCLAV	Întârzierea validării adresei față de CLK	10ns	110ns	Măsurătorile au fost făcute adăugându-se la ieșirile corespunzătoare ale microprocesorului $C_L=20\dots100\text{pF}$
TCLAX	Timpul de menținere al adresei	10ns		
TCLAZ	Întârzierea până la trecerea adreselor în starea a treia	TCLAX	80ns	
TLHLL	Durata impulsului ALE	TCLCH-20ns		
TCLLH	Întârzierea activării semnalului ALE		80ns	
TCHLL	Întârzierea dezactiv. semnalului ALE		85ns	
TLLAX	Timpul de <i>hold</i> al adresei vs. ALE	TCHCL-10ns		
TCLDV	Întârzierea validării datelor vs. CLK	10ns	110ns	
TCHDX	Timpul de <i>hold</i> al datelor vs. CLK	10ns		
TWHDX	Timpul de <i>hold</i> al datelor după $\overline{\text{WR}}$	TCLCH-30ns		
TCVCTV	Întârzierea de activare a comenzilor $\overline{\text{DEN}}, \overline{\text{WR}}, \overline{\text{INTA}}$	10ns	110ns	
TCHCTV	Întârzierea de activare/dezactivare a comenzii $\overline{\text{DT/R}}$	10ns	110ns	
TCVCTX	Întârzierea la inactivarea comenzilor $\overline{\text{DEN}}, \overline{\text{WR}}, \overline{\text{INTA}}$	10ns	110ns	
TAZRL	Întârzierea între momentul trecerii adreselor în starea a treia și activarea comenzii de citire $\overline{\text{RD}}$	0ns		
TCLRL	Întârzierea activării comenzii $\overline{\text{RD}}$	10ns	165ns	
TCLRH	Întârzierea la inactivarea comenzii $\overline{\text{RD}}$	10ns	150ns	
TRHAV	Întârzierea între inactivarea $\overline{\text{RD}}$ și activarea următoarei adrese	TCLCL-45ns		
TCLHAV	Întârzierea validării semnalului HLDA	10ns	160ns	
TRLRH	Durata comenzii $\overline{\text{RD}}$	2·TCLCL-75ns		
TWLWH	Durata comenzii $\overline{\text{WR}}$	2·TCLCL-60ns		
TAVAL	Întârzierea între validarea adresei și terminarea impulsului ALE	TCLCH-60ns		
TOLOH	Frontul crescător al semnalelor de ieșire		20ns	De la 0,8V la 2,0V
TOHOL	Frontul descrescător al semnalelor de ieșire		12ns	De la 2,0V la 0,8V



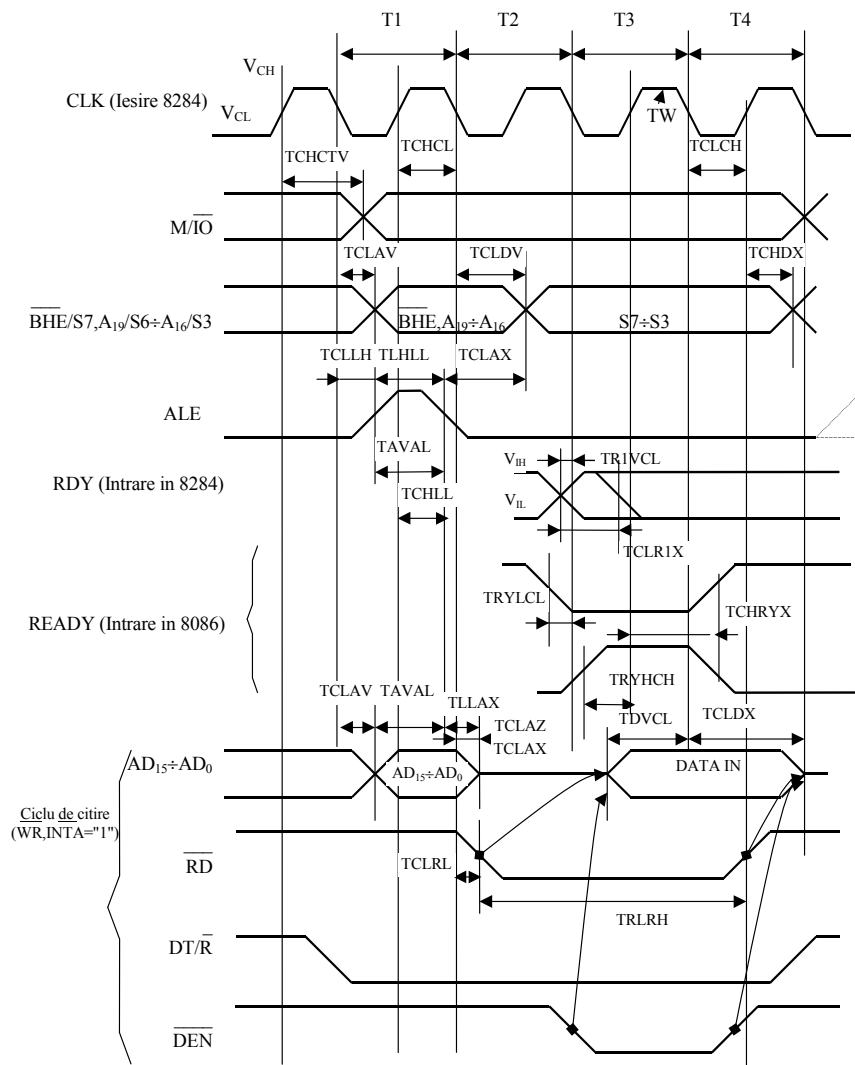


Figura 1.12. Diagrame de timp pentru modul minim



detaliat relația de timp între adresă și ALE ea fiind importantă în aprecierea corectitudinii memorării unei adrese valide de pe magistrala multiplexată a lui 8086. Se cunoaște că *latch*-urile folosite în mod obișnuit, ca 8282 sau 74LS373, sunt transparente atunci când ALE=1 ele memorând, capturând, datele de la intrare, în cazul nostru o adresă, pe frontul negativ al semnalului ALE. În acest caz ne interesează starea liniilor de adresă față de momentul terminării impulsului ALE. O situație defavorabilă va surveni atunci când întârzierea de apariție a adreselor, TCLAV, este maximă și ALE se termină cel mai repede, TCHLL fiind minim, zero. Rezultă de aici că adresa ar fi garantată validă, înainte de terminarea lui ALE, cu  $TCLCH_{\min} - TCLAV_{\max} = 118\text{ns} - 110\text{ns} = 8\text{ns}$  ceea ce, deși ar asigura un timp de stabilizare, de *set-up*, suficient pentru *latch*-urile menționate mai sus (0ns pentru 8282 și 5ns pentru 74LS373), nu constituie o garanție de proiectare sigură. Într-adevăr, analiza de mai sus ignoră un alt parametru, cu adevărat important în relația de timp adresă/ALE și anume TAVAL, timpul de garantare al unei adrese valide înaintea terminării semnalului ALE. Acest timp este dat, conform tabelului 1.2, de ecuația  $TAVAL = TCLCH - 60\text{ns}$  și este prioritar față de rezultatul obținut mai sus prin luarea în considerare numai a relațiilor de timp independente ale semnalelor față de ceasul microprocesorului. Înlocuind, obținem în situația cea mai defavorabilă, pentru 8086 lucrând la 5MHz,  $TAVAL = 118\text{ns} - 60\text{ns} = 58\text{ns}$ .

Timpul de menținere, de *hold*, al adresei după frontul negativ al semnalului ALE este dat de  $TLLAX = TCHCL - 10\text{ns}$  prioritar, de asemenea, față de valoarea obținută prin luarea în considerare a întârzierilor TCHLL și TCLAX față de CLK. În situația cea mai defavorabilă fabricantul garantează un timp de menținere minim  $TLLAX_{\min} = TCHCL_{\min} - 10\text{ns} = 69\text{ns} - 10\text{ns} = 59\text{ns}$  față de o valoare minimă negativă obținută artificial prin  $TCHCL_{\min} - TCHLL_{\max} + TCLAX_{\min} = 69\text{ns} - 85\text{ns} + 10\text{ns} = -6\text{ns}$ . Semnificația lui  $TCLAX_{\min}$  se referă la performanța microprocesorului de a-și inactiva magistrala în cazul în care nu este condiționată de un semnal ALE prea lent datorat, de exemplu, încărcării capacitive mai mari a ieșirii. Timpii TLLAX și TCLAX definesc momentul comutării, după terminarea impulsului ALE, a întregii magistrale multiplexate, atât pentru ciclul de citire cât și pentru cei de scriere. În timpul operațiilor de citire, după ALE, conexiunile AD15÷0 vor fi trecute în starea de impedanță înaltă iar, pentru operațiile de scriere, comutate pe datele ce urmează a fi scrise. AD19÷16 comută din adrese în stări pentru ambele tipuri de ciclul. Durata impulsului ALE este garantată de relația  $TLHLL = TCLCH - 20\text{ns}$ , minimum 98ns, și nu de  $TCLCH - TCLKH + TCHLL$  care, în cazul cel mai defavorabil, ar fi condus la o valoare minimă pentru ALE de  $118\text{ns} - 80\text{ns} + 0\text{ns} = 38\text{ns}$ .

O altă relație de timp care ne interesează în faza de adresare este întârzierea adreselor demultiplexate, după *latch*-uri, față de adresele emise de

microprocesor pe magistrala locală multiplexată. Utilizând *latch*-uri ca 8282 sau 74LS373 adresa nu se va transmite decât după ce  $ALE=1$ . Comparând întârzierea maximă a emiterii adreselor,  $TCLAV_{max}=110ns$ , cu întârzierea maximă a apariției impulsului ALE,  $TCLLH_{max}=80ns$ , deducem că întârzierea maximă a apariției adreselor pe magistrala demultiplexată de adrese a sistemului este dată de  $TCLAV_{max}+t_{latch}$  unde  $t_{latch}$  este întârzierea maximă de propagare prin *latch*-uri (30ns pentru 8282, 18ns pentru 74LS373)

### 1.3.2.2. Ciclul de citire

Operația de citire propriu-zisă constă din: (1) trecerea magistralei locale AD15÷AD0 în starea a treia pentru a permite preluarea de către microprocesor a datelor, (2) activarea comenzii de citire  $\overline{RD}$ , (3) validarea *transceiver*-elor de date, dacă sunt prevăzute, cu ajutorul comenzii  $\overline{DEN}$  și (4) stabilirea direcției acestora prin intermediul semnalului  $DT/\overline{R}$ . Precizarea direcției de circulație a datelor în *transceiver* se face prin poziționarea  $DT/\overline{R}=0$ , recepție-date, la începutul ciclului de magistrală astfel încât acest semnal nu ridică probleme de timp. Problema cea mai importantă, ca relație de timp, la citirea datelor de către microprocesor este ca acestea să fie validate, plasate, pe magistrala locală respectându-se timpii de stabilizare, *set-up*, și menținere, *hold*, impuși de 8086, TDVCL respectiv TCLDX. Corectitudinea datelor depinde, pe de-o parte, de selecția și întârzierea în răspuns ale dispozitivului de memorie sau de I/O adresat, iar pe de altă parte de validarea și întârzierea *transceiver*-elor folosite în sistem.

Activarea comenzii  $\overline{DEN}$  asigură un timp suficient pentru ca *transceiver*-ele să fie deschise circulației datelor spre magistrala locală a microprocesorului asigurându-se timpul minim de stabilizare a datelor. Valoarea minimă a acestui timp pentru frecvența de 5MHz a ceasului e dată de relația  $TCLCL+TCHCL_{min}-TCVCTV_{max}-TDVCL_{min}=200ns+69ns-110ns-30ns=129ns$ . Invalidarea datelor prin invalidarea *transceiver*-ului asigură timpul minim de menținere al datelor,  $TCLDX_{min}=10ns$ , deoarece întârzierea minimă la dezactivarea comenzii  $\overline{DEN}$ , față de aceeași referință ca și TCLDX, este  $TCVCTX_{min}=10ns$  și întârzierea trecerii în starea a treia a *transceiver*-elor este cel puțin 0ns. De asemenea, importantă aici este și durata între momentul invalidării *transceiver*-elor, implicând eliberarea magistralei locale a microprocesorului, și momentul emiterii de către 8086 pe această magistrală a unei noi adrese. Această durată este pentru frecvența de 5MHz a ceasului de minimum  $TCLCL-TCVCTX_{max}+TCLAV_{min}=200ns-110ns+10ns=100ns$ .

Întârzierea plasării datelor pe magistrală de către dispozitivul citit se datorează întâi selecției circuitului, apoi comenzii efective de citire. De obicei,

selecția se face pe baza adreselor, ceea ce nu pune probleme de timp, adresarea făcându-se, așa cum am văzut, la începutul ciclului de magistrală. Rămâne întârzierea datorată comenzii efective de citire  $\overline{RD}$ . Timpul minim "oferit" de microprocesor dispozitivului citit pentru a plasa datele pe magistrala locală, cu ajutorul comenzii  $\overline{RD}$ , fără introducerea unor stări de așteptare TW, la frecvența ceasului de 5MHz și asigurându-se timpul minim de stabilizare este dat de formula  $2 \cdot TCLCL - TCLRL_{\max} - TDVCL_{\min} = 2 \cdot 200ns - 165ns - 30ns = 205ns$ . Acesta este timpul pe care îl au la dispoziție datele pentru a fi "extrase", "accesate", cu ajutorul comenzii  $\overline{RD}$  din dispozitivul adresat și a parcurge magistralele și circuitele interpușe până la microprocesor. Introducerea stărilor de așteptare adaugă la acest timp câte o perioadă TCLCL pentru fiecare stare TW introdusă. Durata minimă a impulsului de citire, parametru important pentru dispozitivele de memorie sau I/O ce urmează a fi utilizate în sistem, este garantată de relația  $TRLRH = 2 \cdot TCLCL - 75ns = 325ns$  prioritară față de formula  $2 \cdot TCLCL - TCLRL + TCLRH$  care ar conduce în cazul cel mai defavorabil la  $2 \cdot 200ns - 165ns + 10ns = 245ns$ . Timpul de menținere minim al datelor,  $TCLDX_{\min} = 10ns$ , este asigurat și pe această cale datorită faptului că întârzierea minimă a dezactivării comenzii  $\overline{RD}$ , față de aceeași referință ca și TCLDX, este  $TCLR_{\min} = 10ns$ .

În sisteme configurate minimal, fără *transceiver*-e, dispozitivele de memorie și/sau I/O sunt plasate direct pe magistrala locală multiplexată a microprocesorului. În această situație nu mai este necesară folosirea semnalelor  $\overline{DEN}$  și  $DT / \overline{R}$ , adresele și comanda efectivă de citire,  $\overline{RD}$ , fiind suficiente. Utilizarea corectă în timp a magistralei locale este asigurată de garantarea activării semnalului  $\overline{RD}$  cel puțin odată cu trecerea magistralei locale în starea a treia,  $TAZRL_{\min} = 0ns$ , și de asigurarea unui timp minim între dezactivarea lui  $\overline{RD}$ , și validarea unei adrese noi de către 8086,  $TRHAV_{\min} = TCLCL - 45ns = 155ns$ .

### 1.3.2.3. Ciclul de scriere

Scrierea efectivă presupune generarea datelor, activarea comenzii de scriere și comanda *transceiver*-elor. După cum se vede în figura 1.12, pe timpul operației de scriere  $DT / \overline{R}$  este menținut pe "1" ceea ce se asigură prin trecerea pe "1" a acestui semnal la sfârșitul oricărui ciclu de magistrală care implică citire de date, așa cum am arătat deja în §1.4.2.2. Deci comanda direcției *transceiver*-elor,  $DT / \overline{R}$ , vine în ciclul de scriere poziționată pe "1" din ciclul sau ciclul precedent, ea rămânând nemodificată, pe transmisie, în timpul operației de scriere. Astfel microprocesorul poate să activeze comanda de

validare a *transceiver*-elor,  $\overline{DEN}$ , încă din starea T1, pe timpul operației de adresare, fără să perturbe adresa emisă pe magistrala locală. Această validare avansată este necesară pentru minimizarea întârzierii datelor, momentul emiterii efective a lor găsind *transceiver*-ele deschise. Comanda de scriere,  $\overline{WR}$ , și datele sunt activate cu același front negativ al ceasului CLK de la începutul stării T2. La începutul scrierii apare o zonă de incertitudine datorată întârzierilor diferite  $TCVCTV=10\div 110ns$  pentru  $\overline{WR}$  și  $TCLDV=10\div 110ns$  pentru date, ea fiind de maximum 100ns între cele două evenimente, de exemplu dacă  $TCVCTV=10ns$  și  $TCLDV=110ns$ . Această incertitudine a relației între date și comanda  $\overline{WR}$  la începutul comenzii de scriere impune proiectantului să utilizeze în sistem dispozitive de memorie sau I/O care să captureze datele pe frontul pozitiv al semnalului  $\overline{WR}$  sau să imagineze diverse adaptări, circuite, care să asigure preluarea de către dispozitive a datelor de scriere după trecerea perioadei de incertitudine. Microprocesorul 8086 garantează validitatea datelor față de frontul pozitiv al comenzii de scriere  $\overline{WR}$ , fără introducerea unor stări de așteptare, o durată dată de formula  $2 \cdot TCLCL - TCLDV + TCVCTX$ . Introducerea stărilor TW adaugă un timp TCLCL pentru fiecare stare de așteptare. În situația cea mai defavorabilă, pentru frecvența de 5MHz a ceasului, obținem un timp garantat de minimum  $2 \cdot 200ns - 110ns + 10ns = 300ns$ . Timpul de menținere al datelor după dezactivarea comenzii de scriere este garantat de  $TWHDX = TCLCH - 30ns$  fiind de minimum  $118ns - 30ns = 80ns$ . TWHDX este prioritar față de timpul obținut prin referiri la ceasul microprocesorului cu formula  $TCLCH + TCHDX - TCVCTX$  care ar conduce la o valoare minimă de  $118ns + 10ns - 110ns = 18ns$ .

La sfârșitul scrierii microprocesorul comută datele în adrese, dacă urmează imediat un nou ciclu de magistrală, T1 după T4, sau se trece în starea a treia, în cazul în care magistrala va fi cedată datorită achitării unei cereri HOLD sau  $\overline{RQ}$ . Așa cum am spus și în §1.4.1, dacă nu urmează imediat un nou ciclu de magistrală procesorul va menține datele emise anterior în timpul operației de scriere. Asigurarea timpului de menținere a datelor la sfârșitul scrierii, după *transceiver*-e, la dispozitivul receptor, se face și prin invalidarea comenzii  $\overline{DEN}$  cu o întârziere de minimum  $TCLCH_{min} + TCVCTX_{min} - TCVCTX_{max} = 118ns + 10ns - 110ns = 18ns$ . Menționăm acum că acest rezultat este abstract el presupunând în formula de mai sus că o întârziere de același tip, aici TCVCTX, poate fi simultan minimă și maximă. În realitate acest lucru nu este posibil, o componentă electronică nefiind capabilă să demonstreze în același timp, pentru același parametru, întârziere maximă și minimă. Argumentul conduce la concluzia că rezultatele obținute în analize de timp cum este și cea de mai sus reprezintă într-adevăr cazurile cele mai defavorabile, nemaî fiind necesare amendări suplimentare pentru a da garanții proiectantului. În situația de față

întârzierea reală între momentul inactivării comenzilor  $\overline{WR}$  și  $\overline{DEN}$  este de aproximativ 60ns [10].

#### 1.3.2.4. Ciclul de achitare a întreruperii

Vom descrie aici numai ciclul de magistrală specific achitării întreruperii mascabile, urmând ca într-un alt capitol să prezentăm pe larg structura mecanismului de întreruperi al microprocesorului 8086 și întreaga lui activitate de pe magistrală la achitarea unei întreruperi.

Întreruperile mascabile generate de sistem sunt sesizate de procesor la intrarea INTR și sunt mascate de bitul I, – indicatorul de condiții pentru Validare/Invalidare Întreruperi –, din registrul de stare. În timpul ultimului ciclu de ceas al fiecărei instrucțiuni, cu unele excepții ce vor fi precizate ulterior, microprocesorul eșantionează linia INTR. Dacă INTR este găsit pe "1" și I poziționat pe Validare-întreruperi 8086 va executa o secvență de achitare a întreruperii. Pentru a garanta achitarea de către microprocesor a întreruperii INTR va trebui menținut pe "1" până când procesorul intră în ciclul de achitare și activează semnalul de răspuns  $\overline{INTA}$ . Semnalul de cerere de întrerupere este de tip activ pe nivel el fiind sincronizat intern de către procesor cu frontul pozitiv al ceasului CLK și testat apoi, cum am mai spus, în ultimul ciclu de ceas al instrucțiunii în curs. Dacă detectarea întreruperii, la sfârșitul execuției instrucțiunii în curs, se produce în timp ce UIM execută un ciclu de magistrală, extrăgând o nouă instrucțiune în vederea umplerii cozii de așteptare, INTR trebuie să satisfacă și un timp de stabilizare de cel puțin două perioade de ceas înainte de ultima stare, T4, a ciclului executat de UIM, pentru ca întreruperea să fie achitată imediat după terminarea acestuia. În cazul în care timpul de *set-up* față de sfârșitul acestui eventual ciclu de magistrală executat de UIM nu este asigurat, achitarea întreruperii se mai poate amâna până după derularea a încă unui ciclu de magistrală, dacă există vreunul în așteptare.

Secvența hardware, specifică, de achitare a întreruperii mascabile, INTR, de către microprocesorul 8086 este compusă din cicli  $\overline{INTA}$  separați prin două stări inactive TI, figura 1.13. Cei doi cicli de magistrală  $\overline{INTA}$  sunt, așa cum se poate vedea și în figura 1.12, asemănători logic cu ciclul de citire, comanda  $\overline{RD}$  înlocuindu-se cu semnalul de achitare  $\overline{INTA}$ . Diferă doar relațiile de timp referitoare la  $\overline{INTA}$  și magistrala AD15÷AD0.





3-a până în ciclul de ceas următor stării T4 a fiecăruia din cei doi cicli de magistrală  $\overline{INTA}$ . În cazul în care, pe timpul achitării întreruperii, coada de așteptare a microprocesorului nu este plină, pe magistrala AD pot fi plasate, în timpul stărilor inactive TI, date în condițiile discutate anterior în §1.4.1. Datele reprezentând tipul întreruperii trebuie să satisfacă aceleași cerințe pentru timpii de stabilizare și menținere, TCLDV și TCLDX, față de frontul negativ al ceasului de la începutul stării T4 din al doilea ciclu  $\overline{INTA}$ , ca și datele dintr-un ciclu de citire.

Ieșirea  $M/\overline{IO}$  va fi poziționată pe "0" indicând sistemului, în timpul ciclilor  $\overline{INTA}$ , o operație de intrare/ieșire. Semnalul  $\overline{LOCK}$ , intern în cazul funcționării microprocesorului 8086 în modul minim, va fi și el activat între stările T2 ale celor doi cicli de achitare pentru a preveni achitarea de către UIM a unei cereri de magistrală între cei doi cicli  $\overline{INTA}$ . Asupra priorității între cererea de întrerupere și cererea de magistrală vom reveni în §1.4.5.4. Comenzile de validare și sens pentru *transceiver-e*,  $\overline{DEN}$  și  $DT/\overline{R}$ , sunt activate în fiecare ciclu  $\overline{INTA}$  având aceleași relații de timp ca și în cazul ciclilor de citire. Între cei doi cicli  $\overline{INTA}$ ,  $\overline{DEN}$  și  $DT/\overline{R}$  sunt dezactivate.

Relațiile de timp pentru semnalul de achitare  $\overline{INTA}$  sunt identice cu cele ale comenzii de scriere  $\overline{WR}$ . Astfel, pe baza acestor relații obținem un timp de acces, de la validarea comenzii  $\overline{INTA}$  până la stabilizarea pe magistrala microprocesorului a datelor reprezentând timpul întreruperii, dat de formula  $2 \cdot TCLCL - TCVCTV - TDVCL$  care, pentru cazul cel mai defavorabil conduce la valoarea  $2 \cdot TCLCL - TCVCTV_{\max} - TDVCL_{\min} = 2 \cdot 200ns - 110ns - 30ns = 260ns$ . Mărirea timpului de acces se poate face prin introducerea de stări TW care, pentru fiecare stare introdusă, adaugă la acest timp o perioadă TCLCL. Garantarea timpului de menținere TCLDX necesar procesorului pentru capturarea datelor pe frontul negativ de la începutul stării T4 se asigură prin condiționarea menținerii datelor pe magistrală cu  $\overline{INTA} = 0$  și  $\overline{DEN} = 0$  și invalidarea acestor comenzi cu cel puțin  $TCVCTX_{\min} = 10ns$  după frontul CLK menționat. Durata minimă a impulsului  $\overline{INTA}$  rezultă din relația  $2 \cdot TCLCL - TCVCTV + TCVCTX$ , conducând, pentru un ceas de 5MHz, la o valoare teoretică minimă de  $2 \cdot 200ns - 110ns + 10ns = 300ns$ . Având în vedere că în realitate microprocesorul nu poate să manifeste simultan, pentru același semnal, atât întâzieri maxime cât și minime, rezultă o valoare practică minimă de 340ns [10]. Pentru eliberarea magistralei la sfârșitul celor doi cicli  $\overline{INTA}$  trebuie ținut cont, pe de-o parte, de întâziera maximă a comenzii  $\overline{INTA}$ ,  $TCVCTX_{\max} = 110ns$ , iar pe de altă parte de momentul când microprocesorul activează magistrala în starea T1 imediat următoare stării T4 din al doilea ciclu  $\overline{INTA}$ . Rezultă un timp minim de  $TCLCL - TCVCTX_{\max} + TCLAV_{\min} = 200ns - 110ns + 10ns = 100ns$  în care dispozitivul care a plasat pe magistrală tipul întreruperii trebuie să-și invalideze ieșirile. Același timp rezultă și luând în

considerare invalidarea datelor datorită comenzii  $\overline{DEN}$ , pentru sisteme în care se folosesc *transceiver-e*.

### 1.3.2.5. Introducerea stărilor de așteptare

Stările de așteptare, TW, pot fi inserate, în timpul operațiilor de citire, scriere sau achitare a întreruperii pentru a se facilita transferul de date cu dispozitive de memorie sau de I/O mai lente. Stările TW se inserează între stările T3 și T4. Pentru a introduce o stare de așteptare într-un ciclu de magistrală semnalul READY trebuie inactivat, pus pe "0", la sfârșitul stării T2 iar pentru a nu intra într-o stare de așteptare sau pentru a ieși din ea, intrarea READY a microprocesorului trebuie activată, trecută pe "1", înainte de frontul pozitiv al ceasului din T3, respectiv din TW.

Introducerea stărilor de așteptare se poate face în două moduri dictate, în general, de complexitatea configurațiilor: primul este acela al sistemelor considerate în mod normal *not ready* – nepregătite, al doilea al sistemelor aflate în mod normal în starea *ready* – pregătite pentru transfer.

În cazul primei abordări, considerate clasică, dacă dispozitivul selectat care primește o comandă,  $\overline{RD}$ ,  $\overline{WR}$  sau  $\overline{INTA}$ , are timp să o execute în ritmul impus de microprocesor el va trebui să activeze în timp util intrarea READY pentru a permite procesorului să-și încheie ciclul de magistrală prevenindu-se astfel introducerea stărilor TW, ca în figura 1.14. Dacă dispozitivul selectat nu poate executa comanda în ritmul microprocesorului el va introduce una sau mai multe stări de așteptare menținând pe "0" un timp corespunzător intrarea READY a lui 8086. Această soluție se pretează, de obicei, sistemelor complexe cu mai multe procesoare sau celor realizate pe magistrale standard cum ar fi Multibus, unde întârzierile de propagare și de acces la magistrală sunt relativ mari și multe din operațiile externe microprocesorului nu se pot executa la viteza lui maximă. Dispozitivele cele mai rapide, puține în cazul adaptării prezentei soluții, vor trebui să activeze semnalul READY ca în figura 1.14 pentru a permite funcționarea sistemului la viteza maximă. În cazul în care nu se respectă relațiile de timp necesare poate fi introdusă o stare TW în plus.

A doua abordare presupune că majoritatea operațiilor externe microprocesorului se pot face în ritmul impus de el și deci că semnalul READY este în mod normal activ, pe "1". Dispozitivele mai lente, mai puține sau mai puțin solicitate în configurațiile pentru care alegem această implementare, vor trebui să dezactiveze semnalul READY pentru a introduce stări TW ca, de exemplu, în figura 1.15. Metoda sistemelor în stare normală *ready* conduce în general la scheme mai simple și este potrivită cu configurațiile mici, monoprosesor, unde nu se pun probleme de viteză legate, cum am menționat, de accesul la o magistrală complexă sau la resurse localizate pe module diferite

cum e cazul configurațiilor complexe, multiprocesor. Dezavantajul acestei metode apare în posibilitatea terminării anormale a unui ciclu de magistrală datorită dezactivării incorecte, fără respectarea relațiilor de timp impuse, a semnalului READY. De aceea, pentru a obține siguranță în funcționarea sistemului, metoda impune analizarea atentă a relațiilor de timp legate de accesul microprocesorului la toate dispozitivele de memorie și/sau I/O.

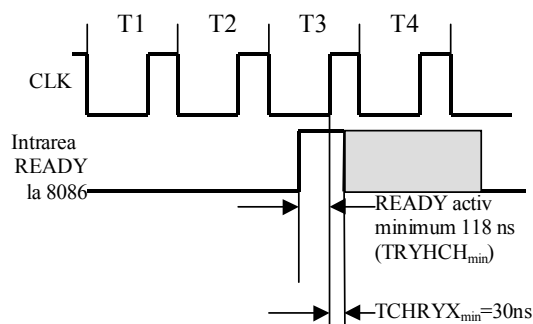


Figura 1.14. Prevenirea introducerii unor stări de așteptare în sisteme normale “not-ready”

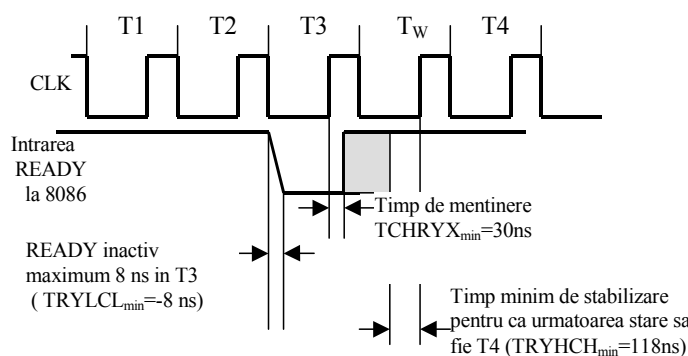


Figura 1.15. Introducerea unei stări de așteptare în sisteme normal “ready”

După cum vedem în figurile detaliate 1.14 și 1.15 dar și în diagramele de catalog din figura 1.12 utilizarea intrării READY ascultă de cerințe diferite în funcție de metoda folosită. În cazul sistemelor în starea normală *ready*, pentru a introduce o stare *tw*, intrarea READY a microprocesorului trebuie pusă pe 0 pe timpul cât CLK mai este "0" în starea T3. Menționăm că aceste restricții sunt rezolvate în cazul utilizării circuitului de ceas 8284 care asigură și în cel mult 8ns ( $TRYLCL_{min} = -8ns$ ) după sfârșitul stării T2, începutul stării T3. După depășirea acestei limite permise, pentru garantarea unei funcționări

în starea normală *not ready* pentru a preveni introducerea stărilor de așteptare intrarea READY trebuie trecută pe "1" cu cel puțin 118ns,  $TRYHCH_{min}$ , înainte de frontul pozitiv al ceasului din T3. În cele două situații prezentate mai sus, ilustrând introducerea respectiv prevenirea introducerii unei stări TW, pentru ca procesorul să ia o decizie corectă este necesar ca semnalul READY, "0" sau "1", să fie menținut un timp minim dat de parametrul  $TCHRYX_{min} = 30ns$  față de frontul pozitiv din T3.

Este necesar să precizăm că în sistemele în stare normală *not ready* programatorii pot să ia o măsură de precauție specifică nescriind cod executabil în ultimii șase octeți ai memoriei existente. Acest lucru elimină situația în care, datorită preextragerilor de instrucțiuni și existenței codului executabil în ultimele locații fizice, 8086 ar putea, depășind spațiul de memorie fizică, să se blocheze în TW deoarece, în acest caz, fără măsuri speciale de proiectare, memoria nu mai răspunde prin activarea semnalului READY. Neasignând cod executabil ultimilor șase octeți rezervăm această zonă de memorie extragerilor în avans ale UIM, executate, așa cum știm, de microprocesor în scopul umplerii cozii de așteptare, și pentru care memoria poate răspunde corect, fără complicații hardware, prin trecerea pe "1" a semnalului READY.

Semnalul READY este generat de obicei fie numai pe baza decodificării de adresă fie pe baza decodificării de adresă și condiționării cu ajutorul comenzii,  $\overline{RD}$ ,  $\overline{WR}$  sau  $\overline{INTA}$ . Pentru sisteme normale *not ready* timpul în care trebuie activat semnalul READY pornind de la adresă pentru a preîntâmpina introducerea unei stări TW, cazul dispozitivelor rapide ce pot funcționa în ritmul impus de microprocesor, este dat de  $2 \cdot TCLCL + TCLCH - TCLAV - TRYHCH$  care, în situația cea mai defavorabilă, conduce la  $2 \cdot 200ns + 118ns - 110ns - 118ns = 290ns$ . Acesta este timpul pus la dispoziție de microprocesorul 8086 unui sistem normal *not ready* pentru ca el să decodifice adresa și să activeze semnalul READY luând astfel decizia preîntâmpinării introducerii unei stări TW. În cazul în care este necesară introducerea unor stări de așteptare sistemul nu are altceva de făcut decât să întârzie activarea liniei READY cu numărul de perioade de ceas corespunzător.

Pentru sistemele normale *ready* timpul în care sistemul poate de data aceasta inactiva semnalul READY, pentru a introduce stări TW pe baza decodificării de adresă, este dat de  $2 \cdot TCLCL - TCLAV - TRYLCL$ . În situația cea defavorabilă obținem  $2 \cdot 200ns - 110ns - (-8ns) = 298ns$ . Pentru dispozitivele rapide sistemul nu va acționa asupra liniei READY lăsând-o în "1".

Revenirea în starea normală a semnalului READY, în cazul modificării lui, se face la începutul ciclului de magistrală următor imediat după schimbarea adresei prin deschiderea *latch*-urilor datorită semnalului ALE.

În cazul condiționării semnalului READY de comenzile  $\overline{RD}$ ,  $\overline{WR}$  sau  $\overline{INTA}$  timpii puși la dispoziție de microprocesor logicii externe sunt mult mai mici. Astfel pentru citiri și sisteme normale *not ready* acest timp, dat de relația  $TCLCL+TCLCH-TCLRL-TRYHCH$ , este în situația cea mai defavorabilă de  $200ns+118ns-165ns-118ns=35ns$ . Pentru sisteme normale *ready* rezultă din relația  $TCLCL-TCLRL-TRYLCL$  un timp de  $200ns-165ns-(-8ns)=43ns$ . Pentru scrieri și cicli  $\overline{INTA}$  formulele se modifică prin înlocuirea parametrului  $TCLRL$  cu  $TCVCTV$  rezultând timpii de  $90ns$  și respectiv  $98ns$ . Atragem atenția că timpii obținuți mai sus se referă la semnalul READY de la intrarea microprocesorului 8086. În §1.4.4.4, referitor la generarea semnalului READY cu ajutorul circuitului 8284, vom reveni precizând și restricțiile impuse semnalului RDY de la intrarea acestui circuit. Analiza relațiilor de timp ale semnalului READY de la intrarea microprocesorului 8086 este utilă mai ales la realizarea unor structuri care nu folosesc circuitul specializat 8284.

Din compararea rezultatelor obținute mai sus concluzionăm că restricțiile de timp impuse sistemelor la generarea semnalului READY sunt mai largi pentru situațiile în care semnalul se formează prin decodificarea adresei față de situațiile în care acesta este condiționat de comenzi. De aici recomandările ca pentru sistemele în stare normală *not ready* formarea semnalului READY să se facă, pentru dispozitivele rapide care nu impun introducerea de stări TW, numai prin decodificarea de adresă, iar pentru dispozitivele lente și prin folosirea, dacă este cazul, a comenzilor. Pentru sistemele în stare normală *ready* în cazul dispozitivelor rapide READY nu trebuie acționat iar în situația dispozitivelor lente semnalul READY se recomandă a fi dezactivat din timp, numai pe baza decodificării de adresă, și reactivat ținând cont eventual și de comandă. Există, de asemenea, posibilitatea utilizării semnalului  $M/\overline{IO}$  la formarea lui READY în sistemele pentru care memoria este rapidă și nu are nevoie de prelungirea ciclilor de magistrală, fiind în schimb necesară introducerea unui număr fix de stări TW pentru toate dispozitivele de I/O.  $M/\overline{IO}$  permite declanșarea unei scheme care să introducă un astfel de număr prefixat de stări TW la orice ciclu  $\overline{RD}$ ,  $\overline{WR}$  sau  $\overline{INTA}$  indiferent de adresa circuitului accesat.

### 1.3.2.6. Preluarea controlului magistralei locale (accesul direct)

Preluarea controlului magistralei locale a microprocesorului 8086 funcționând în modul minim se face printr-un procedeu de cerere-achitare

analog cu cel folosit de microprocesoarele 8080 și Z80 [5, 9, 27, 28]. Semnalele utilizate sunt HOLD pentru cererea de magistrală, analog cu semnalele HOLD și  $\overline{\text{BUSRQ}}$  de la 8080 și respectiv Z80, și HLDA pentru achitarea cererii, echivalent cu HLDA și  $\overline{\text{BUSAK}}$ . Pentru a prelua controlul magistralei locale un dispozitiv, de exemplu un coprocesor sau un circuit pentru comanda accesului direct la memorie, trebuie să activeze intrarea HOLD a procesorului și să aștepte activarea semnalului de achitare pentru a putea efectua transferuri de date pe magistrala locală. Intrarea HOLD este eșantionată de 8086 cu frontul negativ al ceasului, figura 1.16. Decizia de a răspunde la cererea HOLD este luată în unitatea de interfață cu magistrala fiind influențată de următorii factori: activitatea curentă a magistralei, starea semnalului intern  $\overline{\text{LOCK}}$ , activat de prefixul software LOCK, și starea întreruperilor. După eșantionare microprocesorul poate achita cererea HOLD numai dacă următoarea stare T este T4 sau TI. Dacă nu, el așteaptă sfârșitul ciclului de magistrală în curs pentru a trece pe "1" ieșirea HLDA. Dispozitivul care a emis HOLD trebuie să mențină acest semnal activ pe toată durata cât are nevoie de magistrală. Așa cum am spus mai sus, dialogul pentru preluarea magistralei are loc între dispozitivul solicitant și UIM. UE își poate continua activitatea până la golirea stivei sau până la apariția necesității transferării unui operand. La terminarea operației, HOLD va fi trecut pe "0" microprocesorul răspunzând la rândul lui prin HLDA=0 și, dacă este cazul, prin reactivarea magistralei locale. Precizăm, din nou, că datorită funcționării de tip *pipeline* a celor două blocuri constructive, UE și UIM, operația HOLD/HLDA se poate face în paralel cu execuția unor operații interne de către UE. În această situație este posibil ca la încheierea ciclului de cerere/achitare de magistrală microprocesorul să nu aibă nevoie de un acces în exterior și deci, nefiind cazul, să nu activeze imediat magistrala locală. Va apărea atunci o perioadă de timp în care magistrala locală nu va fi controlată de nici-un dispozitiv rămânând în starea a treia. De aceea se recomandă, pentru sistemele care folosesc facilitatea HOLD/HLDA, să se conecteze ieșirile de comandă la +5V prin rezistențe de 22k $\Omega$ , asigurându-se astfel menținerea, în perioada de incertitudine, a nivelului inactiv "1".

Cererea de preluare a controlului, HOLD, nefiind sincronizată de procesor, trebuie să respecte un timp minim de stabilizare față de frontul pozitiv al ceasului. Acest timp,  $\text{THVCH}_{\min} = 35\text{ns}$ , poate fi asigurat dacă se face o sincronizare externă folosindu-se ceasul microprocesorului, oricare din cele două fronturi fiind utilizabil. Trecerea magistralei locale în starea a treia se poate face teoretic după validarea semnalului de achitare HLDA. Întârzierea între activarea semnalului HLDA și eliberarea magistralei este dată de diferența  $\text{TCLAZ} - \text{TCLHAV}$  fiind de maximum 70ns. Această situație poate să deranjeze în sistemele în care activarea lui HLDA conduce imediat la activarea de către dispozitivul solicitant a liniilor de comandă. De aceea pentru aceste cazuri se

recomandă întârzierea lui HLDA astfel încât să se anuleze zona de suprapunere de maximum 70ns. Întârzierile minime cu care microprocesorul conduce din nou magistrala locală, după inactivarea cererii HOLD, sunt date, pentru comenzi, de  $THVCH_{\min}+3\cdot TCLCL= 635\text{ns}$  și pentru adrese/date/stări de  $THVCH_{\min}+3\cdot TCLCL+ TCHCL_{\min}= 704\text{ns}$ , figura 1.17 (s-a considerat că între fronturile ceasului ultimei perioade și activarea semnalelor nu există întârziere). Dacă aceste întârzieri nu sunt suficiente pentru ca dispozitivul solicitant să elibereze magistrala, el trebuie să întârzie la microprocesor momentul trecerii lui HOLD pe "0". În sfârșit, întârzierea între momentul trecerii lui HLDA pe "0" și preluarea comenzilor este de minimum  $TCLCL+TCLCH_{\min}-TCLHAV_{\max}= 200\text{ns}+118\text{ns}-160\text{ns}= 158\text{ns}$  iar până la preluarea adreselor/datelor de  $2\cdot TCLCL-TCLHAV_{\max}= 240\text{ns}$ . Întârzierile legate de semnalele HOLD/HLDA și magistrala locală sunt importante pentru proiectant care, ținând cont de ele, va trebui să asigure controlul disjunct, fără suprapuneri, al magistralei locale din cele două direcții: dinspre microprocesor și dinspre dispozitivele solicitante.

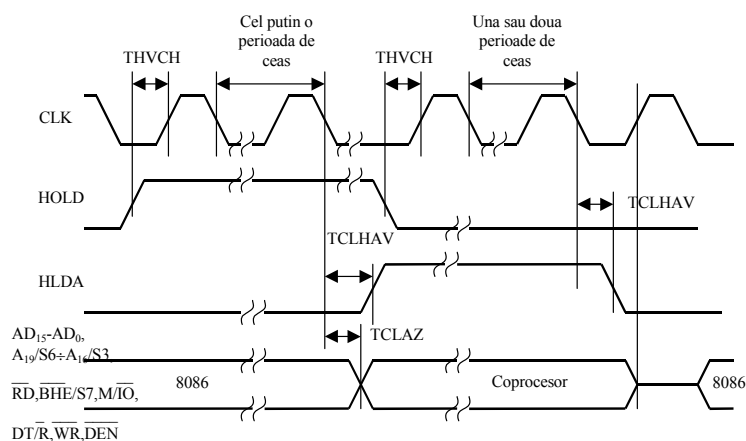


Figura 1.16. Secvența de preluare a controlului magistralei locale în modul minim

Răspunsul microprocesorului la cererea de preluare a magistralei, HOLD, depinde așa cum am spus mai sus de activitatea curentă de pe magistrală, starea semnalului  $\overline{LOCK}$  și a întreruperilor. Dacă cererea HOLD apare când  $\overline{LOCK}$  nu este activ, microprocesorul nu execută un ciclu de achitare a întreruperii și UIM este la sfârșitul unui ciclu de magistrală sau într-o stare TI achitarea cu HLDA va avea întârzierea minimă dată de  $THVCH_{\min}+ TCHCL_{\min}+TCLCL+TCLHAV_{\min}= 35\text{ns}+65\text{ns}+200\text{ns}+10\text{ns}= 310\text{ns}$ . În aceleași condiții întârzierea maximă este dată de  $(THVCH_{\min}-1\text{ns})+ TCLCL+ TCHCL_{\max}+TCLCL+ TCLHAV_{\max}= 34\text{ns}+200\text{ns}+82\text{ns}+200\text{ns}+160\text{ns}= 676\text{ns}$  și apare atunci când se pierde o perioadă de ceas la eșantionarea lui HOLD

datorită nerespectării cu puțin, în relația de mai sus cu  $1ns$ , a timpului de stabilizare minim. Dacă cererea HOLD apare când UIM trebuie să înceapă un ciclu de magistrală întârzierea maximă a lui HLDA, incluzând pierderea unei perioade la eșantionare, ca mai înainte, și o perioadă de stabilizare internă pentru HLDA, este dată de  $(THVCH_{min}-1ns) + TCLCL + TCHCL_{max} + TCLCL + 4 \cdot TCLCL + TCLCL + TCLHAV_{max} = 34ns + 7 \cdot 200ns + 82ns + 160ns = 1,676\mu s$ . La această întârziere se mai pot adăuga stări TW. De asemenea, dacă transferul pe care îl are de executat UIM este pe cuvânt la o adresă impară, vor trebui efectuați doi cicli consecutivi de magistrală înainte ca procesorul să achite cererea HOLD ceea ce conduce la o întârziere suplimentară de încă patru stări T. Se obține în această situație, neținând cont de eventuale stări de așteptare o întârziere de  $2,476\mu s$ .

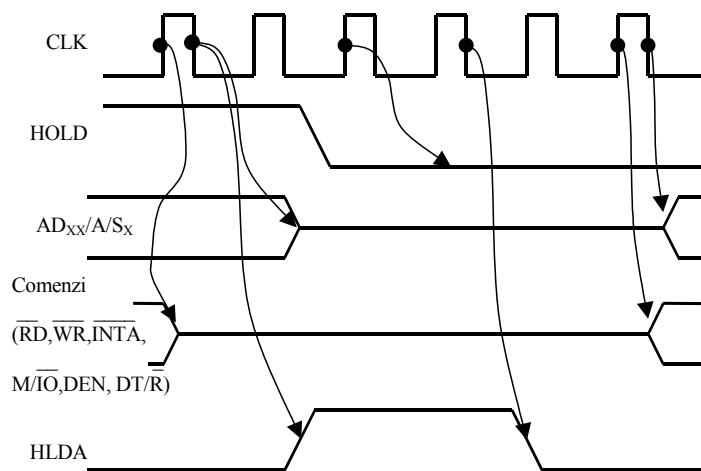


Figura 1.17. Eliberarea și preluarea magistralei locale în modul minim

Cu toate că în modul minim 8086 nu utilizează ieșirea  $\overline{LOCK}$ , prefixul LOCK întrebuințat la programare înaintea unei instrucțiuni nu va permite achitarea unei eventuale cereri HOLD decât după terminarea instrucțiunii respective. În acest mod se poate asigura execuția fără întrerupere a oricăror instrucțiuni care necesită mai mulți cicli de magistrală. Ținând cont de faptul că semnalul intern LOCK este activ încă o stare T după terminarea instrucțiunii "protejate" întârzierea maximă a activării lui HLDA va fi dată de  $(THVCH_{min}-1ns) + TCLCL + TCHCL_{max} + (M+1) \cdot TCLCL + TCLCL + TCLHAV_{max}$ , unde M este numărul de stări T necesar execuției instrucțiunii care urmează



prefixului LOCK iar adăugarea ultimei perioade TCLCL este impusă de necesitatea respectării unui timp de stabilizare intern pentru HLDA. Rezultă, la o frecvență a ceasului de 5MHz, o întârziere de  $M \cdot 200\text{ns} + 876\text{ns}$ .

Pentru situațiile în care cererea de preluare a magistralei se face la începutul unei secvențe de achitare a întreruperii întârzierea maximă până la activarea semnalului de achitare HLDA va fi dată de  $(\text{THVCH}_{\min} - 1\text{ns}) + \text{TCLCL} + \text{TCHCL}_{\max} + \text{TCLCL} + 10 \cdot \text{TCLCL} + \text{TCLCL} + \text{TCLHAV}_{\max} = 2,876\mu\text{s}$ . Ultima perioadă T reprezintă, ca și în cazul prefixului LOCK, un timp de stabilizare intern.

### 1.3.3. FUNCȚIONAREA MICROPROCESORULUI ÎN MODUL MAXIM

Prin legarea intrării  $\text{MN} / \overline{\text{MX}}$  la GND se obține funcționarea microprocesorului în modul maxim. În acest mod de lucru, pentru controlul sistemului, se utilizează circuitul specializat de comandă a magistralei 8288, obținându-se structuri tipice cum este și cea din figura 1.18.

Modul de lucru maxim a fost conceput pentru a asigura microprocesorului posibilități de funcționare în sisteme multi-microprocesor sau cu procesoare de extensie a setului de instrucțiuni, așa numitele *coprocesoare*. Prin adăugarea controlorului 8288 conexiunile externe, destinate în modul minim comenzilor și controlului magistralei, sunt redefinite, §1.3.2.3, oferindu-se următoarele facilități: două canale,  $\overline{\text{RQ}} / \overline{\text{GT0}}$  și  $\overline{\text{RQ}} / \overline{\text{GT1}}$ , pentru preluarea magistralei locale de către procesoare suplimentare aflate pe această magistrală și care pot utiliza circuitele de interfață ale lui 8086 pentru a avea acces la magistrala sistemului; semnalele QS0 și QS1 indicând starea cozii de așteptare utilizabile de către module specializate sau de coprocesoare pentru a "urmări" execuția instrucțiunilor; mecanism de blocare hardware a magistralei cu ajutorul semnalului  $\overline{\text{LOCK}}$  permițând controlul accesului la resursele divizate în sisteme multiprocesor prin prevenirea pierderii voluntare sau forțate a magistralei de către unitatea centrală, activă la un moment dat, și garantarea execuției mai multor cicli de magistrală fără intervenția unei alte unități centrale; alte posibilități de extindere a comenzilor și configurațiilor de sistem cu ajutorul unor circuite suplimentare, cum este circuitul 8289 pentru arbitrarea magistralei de sistem. Analizele de timp descrise în continuare vor ține cont atât de semnalele generate de 8086 cât și de cele generate de controlorul de magistrală. 8288 generează și semnalele suplimentare de control și comandă care permit alcătuirea unor sisteme complexe.



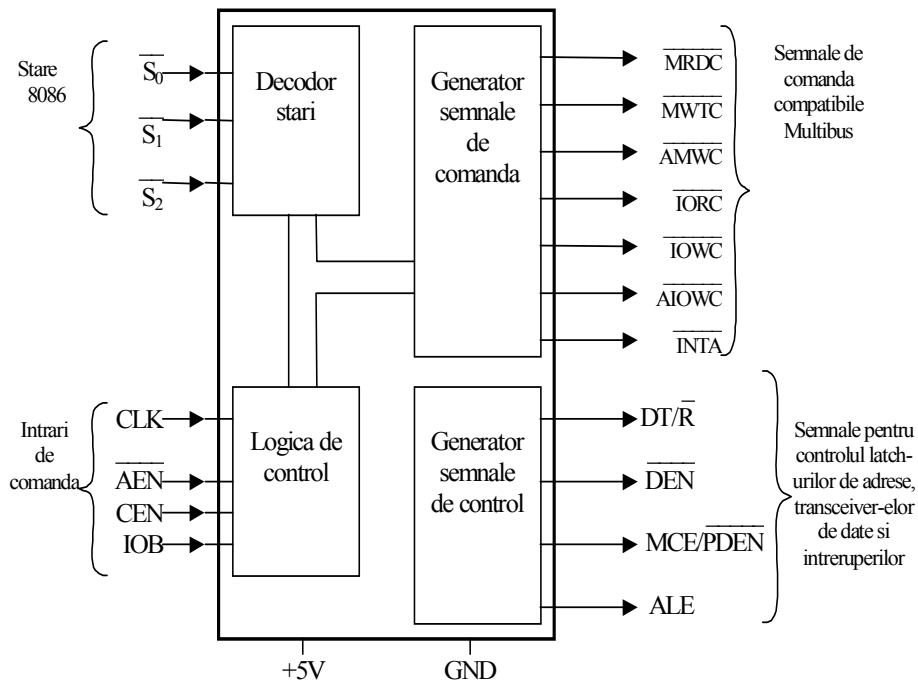


Figura 1.19. Schema bloc a controlului de magistrală 8288

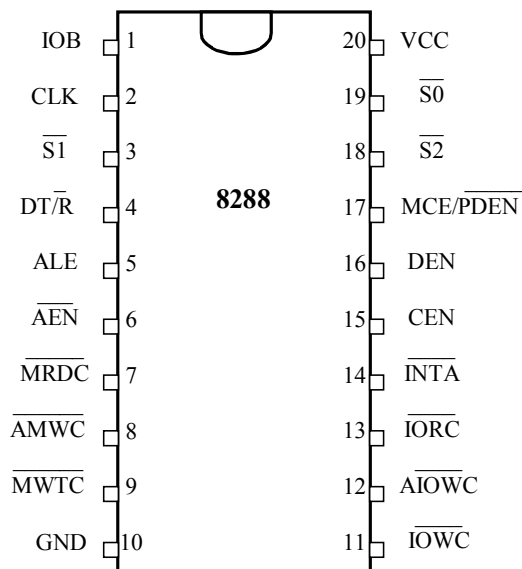


Figura 1.20. Conexiunile externe ale controlului de magistrală 8288

DEN, *Data Enable*, validare date. Ieșire activă pe "1" utilizată pentru validarea *transceiver*-elor de date pe magistrala locală sau pe magistrala sistemului.

DT/ $\bar{R}$ , *Data Transmit/Receive*, transmisie/recepție date. Ieșire folosită pentru precizarea direcției *transceiver*-elor de date, "1" pentru transmisie, scriere în dispozitivele de I/O sau memorie, "0" pentru recepție, citire din dispozitivele de I/O sau memorie.

$\overline{AEN}$ , *Address Enable*, validare adrese. Intrare activă pe "0" utilizată pentru validarea semnalelor de comandă ale controlorului de magistrală.  $\overline{AEN}$  nu va afecta liniile de comandă ale circuitului dacă intrarea IOB a lui este conectată la "1", 8288 lucrând în această situație în modul *magistrală de I/O*.

CEN, *Command Enable*, validare comenzi. Intrare activă pe "1" utilizată pentru validarea semnalelor de comandă ale controlorului precum și a ieșirilor DEN și  $\overline{PDEN}$ . Când CEN=0 semnalele de comandă și DEN și  $\overline{PDEN}$  sunt forțate în starea inactivă. Intrarea poate fi folosită pentru a implementa partiționări de memorie și a elimina conflicte de adresare între dispozitivele plasate pe magistrala de sistem și cele de pe magistralele locale.

IOB, *Input/Output Bus Mode*, modul *magistrală de I/O*. Intrare cu ajutorul căreia este selectat modul de lucru al controlorului: "0" pentru modul *magistrală sistem*, "1" pentru modul *magistrală de I/O*.

$\overline{AIOWC}$ , *Advanced Input/Output Write Command*, comandă de scriere I/O avansată. Ieșire activă pe "0" reprezentând o comandă de scriere avansată, anticipată, pentru dispozitivele de I/O care au nevoie mai devreme de un astfel de semnal pentru decodificări de adrese sau validări. Utilizarea acestui semnal avansat elimină necesitatea introducerii unei stări de așteptare pentru unele dispozitive de I/O mai lente. Semnalul are aceeași cronogramă cu a semnalelor de citire.

$\overline{IOWC}$ , *Input/Output Write Command*, comandă de scriere I/O. Ieșire activă pe "0" reprezentând comanda normală de scriere pentru dispozitivele de I/O.

$\overline{IORC}$ , *Input/Output Read Command*, comandă de citire I/O. Ieșire activă pe "0" reprezentând comanda de citire pentru dispozitivele de I/O.

$\overline{AMWC}$ , *Advanced Memory Write Command*, comandă de scriere memorie avansată. Ieșire activă pe "0" utilizată pentru generarea unei comenzi de scriere anticipate solicitate de unele memorii mai lente pentru decodificări sau validări. Ca și  $\overline{AIOWC}$  are aceeași cronogramă cu cea a semnalelor de citire.

$\overline{MWTc}$ , *Memory Write Command*, comandă de scriere memorie. Ieșire activă pe "0" reprezentând comanda normală de scriere la memorie.

$\overline{MRDC}$ , *Memory Read Command*, comandă de citire memorie. Ieșire activă pe "0" reprezentând comanda de citire din memorie.

$\overline{INTA}$ , *Interrupt Acknowledge*, achitare întreruperi. Ieșire activă pe "0" utilizată pentru a comunica dispozitivului care întrerupe achitarea întreruperii și pentru a comanda plasarea de către acesta din urmă, pe magistrala de date, a vectorului de întrerupere.

$MCE / \overline{PDEN}$ , *Master Cascade Enable/Peripheral Data Enable*. Este o ieșire având două funcții, MCE și  $\overline{PDEN}$ , în funcție de tensiunea aplicată intrării de comandă IOB, "0" respectiv "1". MCE este o ieșire activă pe "1" care se utilizează în timpul unei secvențe de achitare a întreruperii pentru citirea dintr-un controlor de întreruperi *master* a adresei de cascadă corespunzătoare controlorului de întreruperi *slave*, conectat în cascadă, care a solicitat efectiv întreruperea.  $\overline{PDEN}$  este o ieșire activă pe "0" realizând, analog cu ieșirea DEN, funcția de validare a *transceiver*-elor de date pentru dispozitivele de I/O.

În figurile 1.21, 1.22 și 1.23 se prezintă diagramele care ilustrează relațiile de timp între semnalele utilizate și generate de circuitul 8288 iar în tabelele 1.4 și 1.5 se dau valorile minime și maxime ale parametrilor ce caracterizează aceste relații [11].

Așa după cum se poate înțelege din figura 1.19 liniile de stare activate de microprocesor sunt decodificate de controlorul de magistrală 8288 pentru a emite comanda corespunzătoare conform tabelului 1.6. Comenzile se emit în două feluri, în funcție de modul de lucru al circuitului selectat cu ajutorul conexiunii IOB.

Modul *magistrală de I/O*. În acest mod de lucru, IOB=1, toate comenzile de I/O,  $\overline{IORC}$ ,  $\overline{IOWC}$ ,  $\overline{AIOWC}$ ,  $\overline{INTA}$ , sunt permanent validate, indiferent de starea semnalului  $\overline{AEN}$ . Atunci când microprocesorul inițiază un ciclu de magistrală pentru o operație de I/O, 8288 va activa imediat liniile de comandă corespunzătoare împreună cu semnalele  $\overline{PDEN}$  și DT/R pentru controlul *transceiver*-elor de magistrală. În acest mod de lucru comenzile de I/O nu pot fi utilizate pe o magistrală unică de sistem, aceeași pentru memorie și dispozitivele de I/O, datorită lipsei unui arbitraj. Acest mod de lucru se folosește atunci când în cadrul sistemului există două magistrale separate pentru memorie și I/O. În aceste situații microprocesorul nu va aștepta nimic când accesează magistrala de I/O. Comenzile pentru memorie,  $\overline{MRDC}$ ,  $\overline{MWTC}$ ,  $\overline{AMWC}$ , rămân dependente de intrarea  $\overline{AEN}$  fiind validate numai după ce această intrare devine "0" datorită unui semnal de tip *magistrală gata*. Modul de lucru *magistrală de I/O* este avantajos în sistemele multi-microprocesor pentru microprocesoarele care au toate dispozitivele de I/O dedicate.

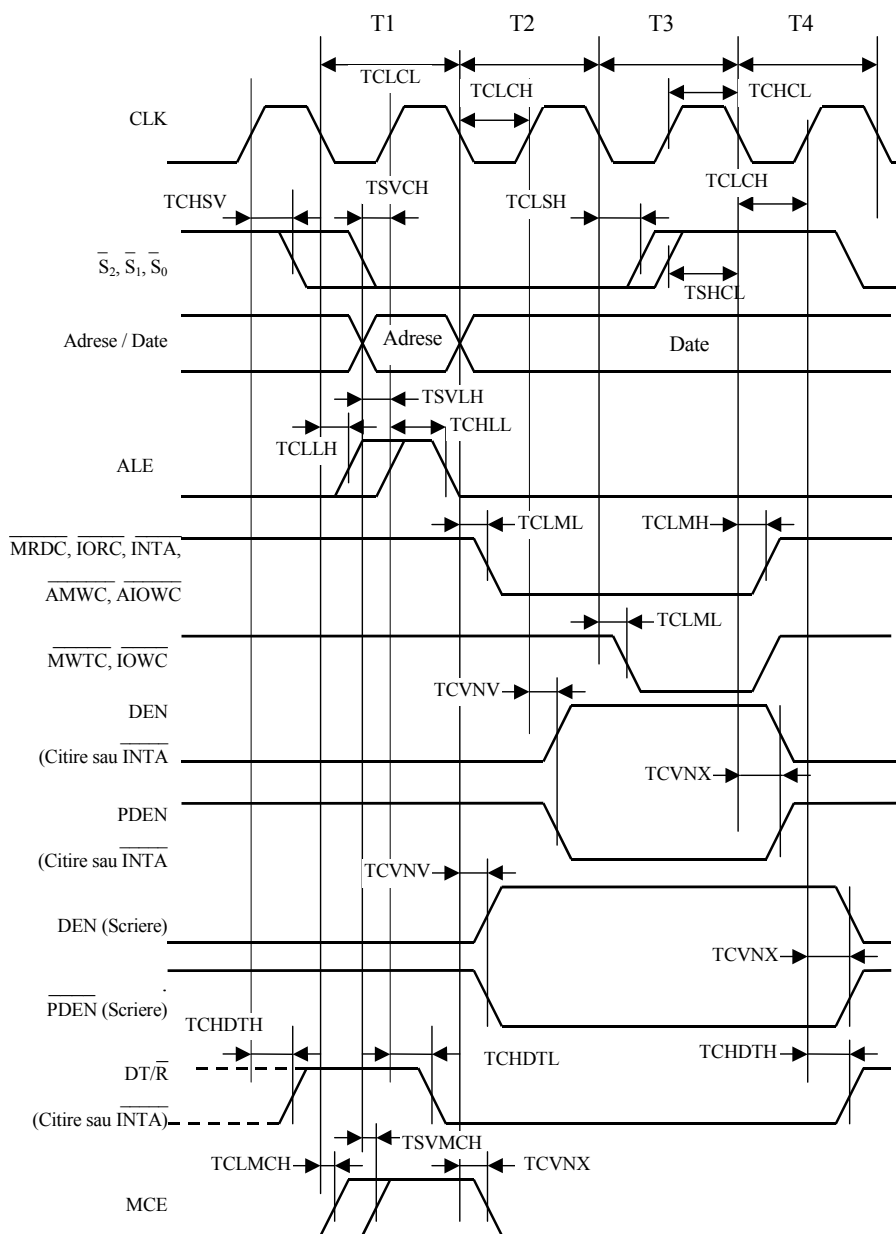


Figura 1.21. Diagramele de timp pentru circuitul 8288

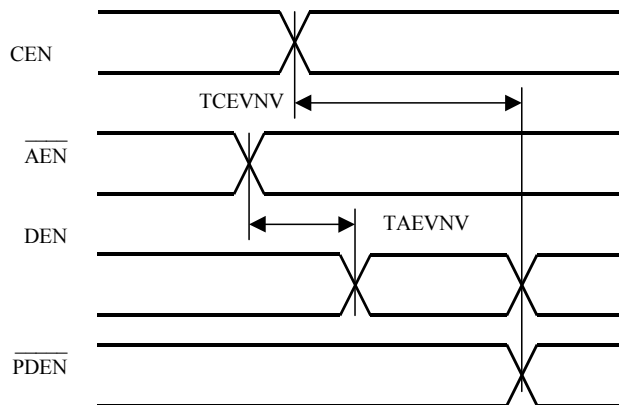


Figura 1.22. Condiționarea semnalelor DEN și  $\overline{\text{PDEN}}$

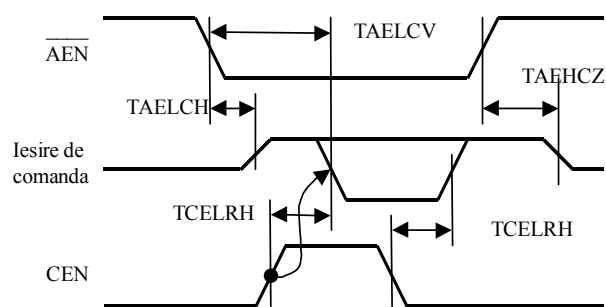


Figura 1.23. Întârzierile la validarea/invalidarea comenzilor

Tabelul 1.4. Parametrii 8288. Cerințe de timp

Parametru	Semnificație	Valoare minimă
TCLCL	Perioada ceasului CLK	125ns
TCLCH	Timpul cât CLK=0	66ns
TCHCL	Timpul cât CLK=1	40ns
TSVCH	Timpul de stabilizare, de <i>set-up</i> , al semnalelor de stare active	65ns
TCHSV	Timpul de menținere, de <i>hold</i> , al semnalelor de stare inactive	10ns
TSHCL	Timpul de stabilizare al semnalelor de stare inactive	55ns
TCLSH	Timpul de menținere al semnalelor de stare active	10ns

Tabelul 1.5. Parametrii 8288. Răspunsuri în timp

Parametru	Semnificație	Valoare minimă	Valoare maximă
TCVNV	Întârzierea la activarea semnalelor de control	5ns	45ns
TCVNX	Întârzierea la inactivarea semnalelor de control	10ns	45ns
TCLLH	Întârzierea la activarea semnalului ALE în raport cu CLK		15ns
TCLMCH	Întârzierea la activarea semnalului MCE în raport cu CLK		15ns
TMHNL	Întârzierea între activarea semnalului DEN și activarea semnalelor de comandă	TCLCH-5ns	
TSVLH	Întârzierea între activarea stărilor și activarea semnalului ALE		15ns
TSVMCH	Întârzierea între activarea stărilor și activarea semnalului MCE		15ns
TCHLL	Întârzierea la inactivarea semnalului ALE		15ns
TCLML	Întârzierea la activarea comenzilor	10ns	35ns
TCLMH	Întârzierea la inactivarea comenzilor	10ns	35ns
TCHDTL	Întârzierea la activarea semnalului DT / $\bar{R}$		50ns
TCHDTH	Întârzierea la inactivarea semnalului DT / $\bar{R}$		30ns
TAE LCH	Întârzierea la validarea comenzilor		40ns
TAEHCZ	Întârzierea la invalidarea comenzilor		40ns
TAE LCV	Întârzierea la activarea comenzilor	115ns	200ns
TAEVNV	Întârzierea între $\bar{AEN}$ și DEN		20ns
TCEVNV	Întârzierea între CEN și DEN sau $\bar{PDEN}$		20ns
TCELRH	Întârzierea între CEN și comenzi		TCLML

Tabelul 1.6. Decodificarea stărilor și comenzile emise de 8288

$\bar{S2}$	$\bar{S1}$	$\bar{S0}$	Starea microprocesorului	Comanda emisă de 8288
0	0	0	Achitare întrerupere	$\bar{INTA}$
0	0	1	Citire <i>port</i> de I/O	$\bar{IORC}$
0	1	0	Sciere <i>port</i> de I/O	$\bar{IOWC}, \bar{AIOWC}$
0	1	1	Oprire ( <i>Halt</i> )	-
1	0	0	Acces cod	$\bar{MRDC}$
1	0	1	Citire memorie	$\bar{MRDC}$
1	1	0	Sciere memorie	$\bar{MWTC}, \bar{AMWC}$
1	1	1	Stare pasivă	-

Modul *magistrală de sistem*. În acest mod de lucru, IOB=0, toate comenzile sunt validate cu  $\bar{AEN}=0$ . Modul de funcționare se utilizează în sistemele realizate pe o singură magistrală, memoria cât și dispozitivele de I/O fiind accesibile numai prin intermediul acestei magistrale. Folosirea circuitului 8288 în acest mod de lucru impune realizarea în cadrul sistemului a unei logici



de arbitraj care va informa controlorul, prin intermediul liniei  $\overline{AEN}$ , dacă magistrala este liberă pentru a fi accesată de microprocesor.

O facilitate suplimentară pentru utilizarea controlorului în sisteme multi-microprocesor este și ieșirea MCE activă numai în modul *magistrală de sistem*. Așa cum s-a arătat și în §1.4.2.4 pe timpul unei secvențe de achitare a întreruperii apar doi cicli de întrerupere succesivi. Pe timpul primului ciclu, necesar pentru efectuarea unor operații interne în microprocesor, nu are loc nici-un transfer de adrese sau date. De aceea, logica asociată controloarelor de întreruperi va trebui să mascheze semnalul MCE pe durata acestui prim ciclu  $\overline{INTA}$ . Imediat însă înaintea celui de-al doilea ciclu, semnalul MCE activat va putea, pentru situații în care se folosesc mai multe controloare de întreruperi legate în cascadă, să valideze o adresă de cascadă emisă de controlorul *master* pe magistrala locală a procesorului pentru a fi memorată cu ajutorul semnalului ALE în *latch*-uri de adrese. În acest fel, la sfârșitul celui de-al doilea ciclu  $\overline{INTA}$ , controlorul *slave* adresat cu adresa de cascadă va plasa pe magistrala de date a sistemului vectorul propriu de întrerupere. Vom reveni cu detalii referitoare la organizarea sistemului de întreruperi în §1.4.4.

### 1.3.3.2. Adresarea

În cele ce urmează ne vom referi atât la diagramele și relațiile de timp pentru microprocesorul 8086, figura 1.24 și tabelele 1.7 și 1.8, cât și la diagramele și relațiile de timp ale controlorului de magistrală 8288, date în figurile 1.21, 1.22, 1.23 și în tabelele 1.4 și 1.5.

În modul de lucru maxim adresele sunt generate de microprocesor, ca și în modul minim, în timp ce semnalul de strobare ALE este emis de 8288. Pentru aprecierea situației celei mai defavorabile observăm că activarea semnalului ALE are două căi de întârziere: față de stări și față de ceas. În raport cu stările  $\overline{S2}$ ,  $\overline{S1}$ ,  $\overline{S0}$  apare o întârziere, măsurată de la frontul pozitiv anterior stării T1, dată de suma  $TCHSV+TSVLH$ . A doua cale de întârziere TCLKH măsurată de la frontul negativ al ceasului CLK care marchează începutul stării T1. De aici rezultă că, în situația când microprocesorul este mai lent pe calea de activare a stărilor și alternanța pozitivă a ceasului este minimă, ALE va fi generat cu o întârziere TSVLH după schimbarea stărilor. Pe de altă parte, dacă stările  $\overline{S2}$ ,  $\overline{S1}$ ,  $\overline{S0}$  se schimbă înainte de începerea stării T1 8288 garantează activarea semnalului ALE cel mult cu o întârziere TCLKH față de începutul stării T1. Pentru prima situație, cea mai defavorabilă comparativ cu  $TCLKH_{max}=15ns$ , se obține, față de începutul stării T1, o întârziere maximă de  $TCHSV_{max} + TSVLH_{max}-TCHCL_{min}=110ns+15ns-69ns=56ns$ .

Tabelul 1.7. Parametrii 8086 pentru modul maxim. Cerințe de timp

Parametru	Semnificație	Valoare minimă	Valoare maximă	Observații
TCLCL	Perioada ceasului CLK	200ns	500ns	
TCLCH	Timpul cât CLK=0	118ns		
TCHCL	Timpul cât CLK=1	69ns		
TCH1CH2	Frontul crescător al ceasului CLK		10ns	Măsurat între valorile 1,0V și 3,5V
TCL2CL1	Frontul descrescător al ceasului		10ns	Măsurat între valorile 3,5V și 1,0V
TDVCL	Timpul de stabilizare, de <i>set-up</i> , al datelor la citire	30ns		
TCLDX	Timpul de menținere, de <i>hold</i> , al datelor la citire	10ns		
TR1VCL	Timpul de stabilizare al semnalului RDY la intrarea în 8284A măsurat față de frontul descrescător al ceasului CLK	35ns		Acest timp este necesar semnalului asincron RDY pentru a garanta recunoașterea lui în următoarea perioadă a ceasului CLK
TCLR1X	Timpul de menținere al semnalului RDY la intrarea în 8284A	0ns		
TRYHCH	Timpul de stabilizare al semnalului READY la intrarea în 8086	118ns		
TCHRYX	Timpul de menținere al semnalului READY la intrarea în 8086	30ns		
TRYLCL	Timpul necesar ca intrarea READY să fie considerată inactivă față de începutul stării T3	-8ns		READY mai poate deveni inactiv maximum 8ns în T3 transformând această stare într-o stare de așteptare TW
TINVCH	Timpul de stabilizare pentru recunoașterea semnalelor INTR, NMI, TEST	30ns		
TGVCH	Timpul de stabilizare pentru semnalele RQ / GT	30ns		
TCHGX	Timpul de menținere al semnalului RQ la intrarea în 8086	40ns		
TILIH	Frontul crescător al semnalelor la intrările în 8086		20ns	Măsurat între 0,8V și 2V
TIHIL	Frontul descrescător al semnalelor la intrările în 8086		12ns	Măsurat între 2V și 0,8V

Tabelul 1.8. Parametrii 8086 pentru modul maxim. Răspunsuri în timp

Parametru	Semnificație	Valoare minimă	Valoare maximă	Observații
TCLML	Întârzierea la activarea comenzilor	10ns	35ns	Comenzile generate de 8288
TCLMH	Întârzierea la inactivarea comenzilor	10ns	35ns	
TRYHSH	Întârzierea între activarea lui READY și trecerea stărilor $\overline{S2}$ , $\overline{S1}$ , $\overline{S0}$ în starea pasivă		110ns	
TCHSV	Întârzierea la activarea stărilor $\overline{S2}$ , $\overline{S1}$ , $\overline{S0}$ față de CLK	10ns	110ns	
TCLSH	Întârzierea la inactivarea stărilor $\overline{S2}$ , $\overline{S1}$ , $\overline{S0}$ față de CLK	10ns	130ns	
TCLAV	Întârzierea la validarea adresei vs. CLK	10ns	110ns	
TCLAX	Timpul de menținere al adresei	10ns		
TCLAZ	Întârzierea până la trecerea adreselor în starea a 3-a	TCLA X	80ns	
TSVLH	Întârzierea între validarea stării și punerea pe "1" a lui ALE		15ns	ALE și MCE generate de 8288
TSVMCH	Întârzierea între validarea stării și punerea pe "1" a lui MCE		15ns	
TCLLH	Întârzierea între frontul descrescător al ceasului CLK și validarea lui ALE		15ns	
TCLMCH	Întârzierea între frontul descrescător al ceasului CLK și validarea lui MCE		15ns	
TCHLL	Întârzierea la inactivarea lui ALE		15ns	
TCLMCL	Întârzierea la inactivarea lui MCE		15ns	
TCLDV	Întârzierea validării datelor față de CLK	10ns	110ns	
TCHDX	Timpul de menținere al datelor vs. CLK	10ns		
TCVNV	Întârzierea față de CLK la activarea semnalului de control DEN	5ns	45ns	DEN generat de 8288
TCVNX	Întârzierea la inactivarea semnalelor de control	10ns	45ns	
TAZRL	Întârzierea între trecerea în starea a 3-a a magistralei și activarea lui $\overline{RD}$	0ns		
TCLRL	Întârzierea față de CLK a activării lui $\overline{RD}$	10ns	165ns	
TCLRHL	Întârzierea față de CLK a inactivării lui $\overline{RD}$	10ns	150ns	
TRHAV	Întârzierea între inactivarea lui $\overline{RD}$ și activarea adresei următoare	TCLCL -45ns		
TCHDTL	Întârzierea față de CLK la activarea semnalului $\overline{DT}/\overline{R}$		50ns	
TCHDTH	Întârzierea la inactivarea semnalului $\overline{DT}/\overline{R}$		30ns	
TCLGL	Întârzierea la activarea lui GT	0ns	85ns	
TCLGH	Întârzierea la inactivarea lui GT	0ns	85ns	
TRLRH	Lățimea semnalului $\overline{RD}$	2TCLC L-75ns		
TOL0H	Frontul crescător al semnalelor generate		20ns	
TOH0L	Frontul descrescător al semnalelor generate		12ns	0,8V $\uparrow$ 2V 2V $\downarrow$ 0,8V



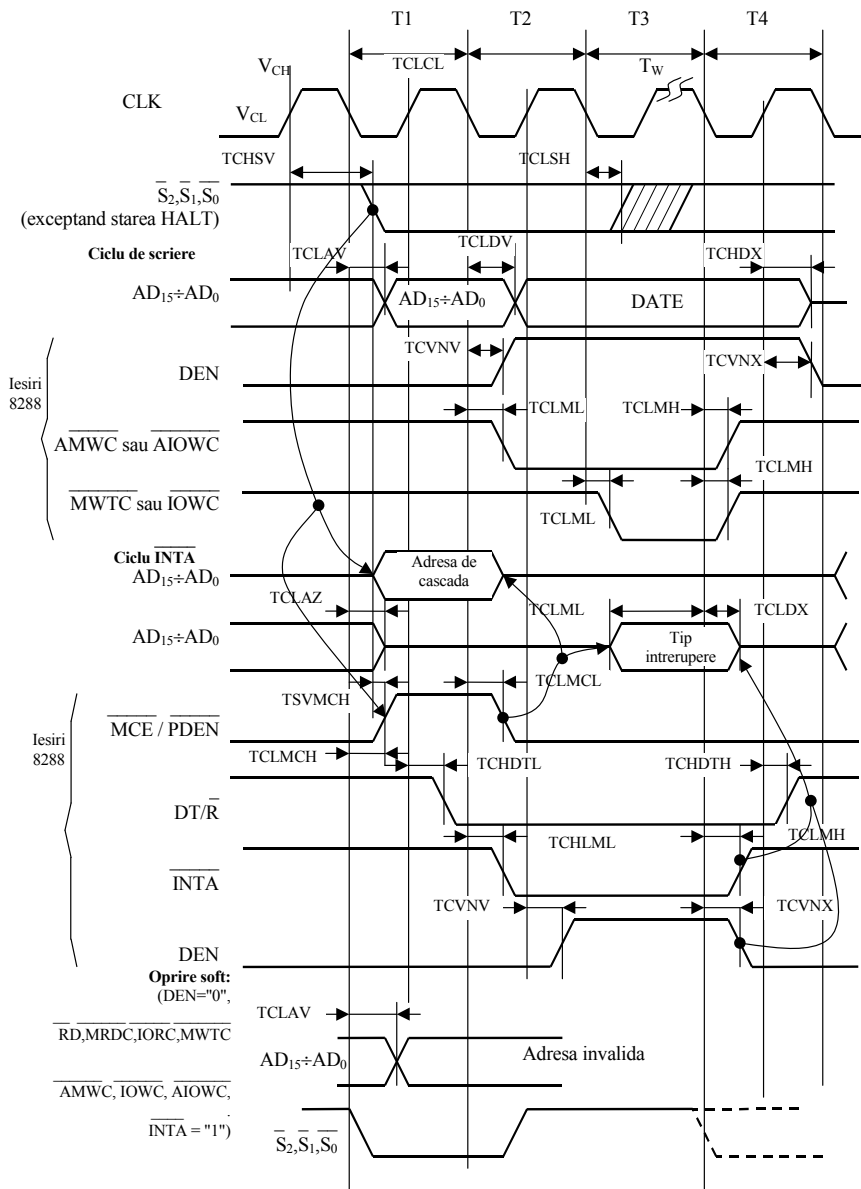


Figura 1.24. Continuare

Dezactivarea semnalului ALE se face cu frontul pozitiv din T1 indiferent de întârzierea la activare. De aici rezultă, la o frecvență a ceasului de 5MHz, o durată minimă pentru ALE de  $T_{CLCH}_{max} + T_{CHLL}_{min} - 56ns = 131ns + 0ns - 56ns = 75ns$ .

Timpul de stabilizare minim al adreselor față de frontul negativ al lui ALE, necesar pentru asigurarea capturării adreselor în *latch*-uri, este dat de  $TCLCH_{\min}-TCLAV_{\max}+TCHLL_{\min}$ . Se obține la 5MHz, presupunând din nou aproape teoretic  $TCHLL_{\min}=0\text{ns}$ , un timp de *set-up* garantat de minimum  $118\text{ns}-110\text{ns}+0\text{ns}=8\text{ns}$  ceea ce permite utilizarea unor *latch*-uri ca 8282, 8283 sau 74LS373. Timpul de menținere al adreselor față de frontul căzător al semnalului ALE se poate obține din relația  $TCHCL-TCHLL$  care, în cazul cel mai defavorabil și pentru o frecvență a ceasului de 5MHz, conduce la valoarea minimă de  $TCHCL_{\min}-TCHLL_{\max}=69\text{ns}-15\text{ns}=54\text{ns}$ .

Timpii de comutare ai magistralei multiplexate din adrese în stări și în starea a treia sau date funcție de tipul ciclului, citire respectiv scriere, rămân aceleași ca în modul minim.

### 1.3.3.3. Ciclu de citire

În modul maxim se pot folosi comenzile de citire generate de 8288, dar și comanda  $\overline{RD}$  activată direct de microprocesor, ale cărei relații de timp sunt aceleași ca în modul minim. Deoarece parametrii dinamici ai semnalelor emise de controlorul de magistrală sunt mai buni decât cei ai semnalelor emise de microprocesor, cum se poate vedea din comparația tabelor 1.3 și 1.5, se recomandă folosirea acestor semnale pentru accesul la dispozitivele de memorie sau I/O plasate pe magistrala demultiplexată, de sistem, magistrală care datorită unor *buffer*-e suplimentare și a unei încărcări capacitive mai mari introduce, de obicei, întârzieri suplimentare. Semnalul  $\overline{RD}$  poate fi utilizat pentru citirea unor dispozitive apropiate, conectate direct pe magistrala locală multiplexată.

Comenzile de citire a dispozitivelor de memorie sau de I/O emise de 8288,  $\overline{MRDC}$ ,  $\overline{IORC}$  și  $\overline{INTA}$ , au aceleași caracteristici dinamice: sunt activate cu o întârziere  $TCLML$  față de începutul stării T2 și sunt dezactivate cu o întârziere  $TCLMH$  față de începutul stării T4. Durata minimă a unei astfel de comenzi va deci dată de  $2 \cdot TCLCL - TCLML_{\max} + TCLMH_{\min}$  și va fi la 5MHz de  $400\text{ns} - 35\text{ns} + 10\text{ns} = 375\text{ns}$ . Timpul minim oferit pentru accesarea datelor de către microprocesor va fi  $2 \cdot TCLCL - TCLML_{\max} - TDVCL_{\max} = 400\text{ns} - 35\text{ns} - 30\text{ns} = 335\text{ns}$  simțitor mai mare decât cel de 205ns pe care îl avem la dispoziție în modul minim.

După cum se vede în figurile 1.21 și 1.24 direcția *transceiver*-elor se stabilește în T1 iar validarea lor pe magistrala locală a microprocesorului se face în T2 asigurându-se o întârziere minimă de  $TCLCH_{\min} + TCVNV_{\min} = 118\text{ns} + 5\text{ns} = 123\text{ns}$  față de momentul comandării trecerii în starea a treia a magistralei locale. Timpul minim rămas la dispoziție pentru propagarea

datelor spre procesor cu respectarea intervalului de stabilizare, de *set-up*, este dat de relația  $TCHCL_{\min} + TCLCL - TCVNV_{\max} - TDVCL_{\min} = 69\text{ns} + 200\text{ns} - 45\text{ns} - 30\text{ns} = 194\text{ns}$ , mai mare decât timpul de  $129\text{ns}$  asigurat în modul minim. Timpul de menținere al datelor,  $TCLDX_{\min} = 10\text{ns}$ , este garantat din punctul de vedere al controlului *transceiver*-elor deoarece atât DEN cât și  $DT/\bar{R}$  sunt menținute în T4 minimum  $TCVNX_{\min} = 10\text{ns}$ , respectiv  $TCLCH_{\min} + TCHDTH_{\min} = 118\text{ns} + 0\text{ns} = 118\text{ns}$ . Pe de altă parte, *transceiver*-ele vor fi invalidate înainte de următorul ciclu al microprocesorului cu minimum  $TCLCL - TCVNX_{\max} = 200\text{ns} - 45\text{ns} = 155\text{ns}$ .

#### 1.3.3.4. Ciclul de scriere

Spre deosebire de modul de lucru minim, în modul maxim 8288 generează două tipuri de comenzi de scriere pentru memorie și I/O: avansate,  $\overline{AMWC}$  și  $\overline{AIOWC}$ , și normale,  $\overline{MWTC}$ ,  $\overline{IOWC}$ . Comenzile avansate de scriere, generate cu o perioadă de ceas înaintea celor normale, au aceeași desfășurare în timp ca și comenzile de citire, durata lor fiind de  $2 \cdot TCLCL - TCLML + TCLMH$ . La 5MHz această durată este de minimum  $2 \cdot 200\text{ns} - 35\text{ns} + 10\text{ns} = 375\text{ns}$ . Pentru comenzile normale lățimea impulsului de scriere, dată de  $TCLCL - TCLML + TCLMH$ , va fi, la 5MHz, de minimum  $175\text{ns}$ . Întârzierea datelor, calculată față de începutul stării T2, depinde atât de microprocesor, parametrul  $TCLDV$ , cât și de comanda DEN pentru validarea *transceiver*-elor,  $TCVNV$ . Se observă din tabelele 1.3 și 1.5 că întârzierea cea mai mare se datorează microprocesorului:  $TCLDV_{\max} = 110\text{ns}$  față de  $TCVNV_{\max} = 45\text{ns}$ . Pentru comenzile avansate va fi deci posibil ca datele să fie valide la ieșirile microprocesorului după activarea comenzii de scriere de către 8288 cu o întârziere dată de diferența  $TCLDV - TCLML$ , maximum  $100\text{ns}$ . Pentru comenzile normale datele vor fi întotdeauna valide înaintea activării acestor comenzi cu cel puțin  $TCLCL - TCLDV_{\max} + TCLML_{\min} = 200\text{ns} - 110\text{ns} + 10\text{ns} = 100\text{ns}$ . Față de frontul pozitiv al ambelor tipuri de comenzi de scriere datele sunt stabile, la o frecvență a ceasului de 5MHz, cu minimum  $2 \cdot TCLCL - TCLDV_{\max} + TCLMH_{\min} = 2 \cdot 200\text{ns} - 110\text{ns} + 10\text{ns} = 300\text{ns}$ . Timpul de menținere, de *hold*, al datelor după dezactivarea comenzilor de scriere este, ca și timpul de *set-up*, dependent atât de microprocesor cât și de comanda de invalidare a *transceiver*-elor emisă de 8288. Procesorul garantează stabilitatea datelor după frontul crescător al comenzilor de scriere minimum  $TCLCH_{\min} - TCLMH_{\max} + TCHDX_{\min} = 118\text{ns} - 35\text{ns} + 10\text{ns} = 93\text{ns}$ . Invalidarea *transceiver*-elor cu DEN se face cu cel puțin  $TCLCH_{\min} - TCLMH_{\max} + TCVNX_{\min} = 118\text{ns} - 35\text{ns} + 10\text{ns} = 93\text{ns}$ . Astfel, din punct de vedere al

procesorului, 8086+8288, datele vor fi garantat stabile cel puțin 93ns după comanda de scriere.

### 1.3.3.5. Achitarea întreruperii

Secvența de achitare a întreruperii, generarea celor doi cicli  $\overline{\text{INTA}}$ , este logic identică cu cea din modul minim, diferențele funcționale fiind legate de posibilitățile oferite de 8288, prin generarea semnalului MCE, de a realiza sisteme cu mai multe controloare de întreruperi. Relațiile de timp ale semnalelor DEN, DT/ $\overline{\text{R}}$  și  $\overline{\text{INTA}}$  sunt aceleași ca în ciclul de citire,  $\overline{\text{INTA}}$  fiind echivalent cu o comandă de citire,  $\overline{\text{MRDC}}$  sau  $\overline{\text{IORC}}$ . Ca și în modul minim, pe timpul celor doi cicli  $\overline{\text{INTA}}$  magistrala locală multiplexată a microprocesorului va fi trecută în starea a treia la începutul stării T1 a fiecărui ciclu. Timpii de stabilizare și menținere pentru datele ce reprezintă tipul de întrerupere, plasate pe magistrala locală a procesorului, sunt aceiași ca pentru orice citire: TDVCL și TCLDX.

Diferențele de funcționare în modul maxim sunt datorate posibilităților oferite de controlorul de magistrală în vederea implementării de sisteme cu mai multe circuite de control al întreruperilor folosind semnalele MCE și ALE. După cum s-a menționat în §1.4.3.1., în modul de lucru *magistrală de sistem* 8288 generează, pe timpul stării T1, în același timp cu semnalul de strobare ALE, semnalul MCE. MCE poate fi utilizat în vederea validării unei adrese de cascadă, de exemplu CAS2÷0 pentru 8259A, generate de controlorul de întreruperi principal, *master*, pe liniile cele mai semnificative, AD15÷AD13, ale magistralei locale. Această adresă poate fi capturată cu ALE în *latch*-urile de adresă ale sistemului pentru a selecta controlorul de întreruperi secundar, *slave*, care a generat efectiv întreruperea. Un exemplu de schemă care utilizează această facilitate este dat în figura 1.24A. MCE este generat în fiecare ciclu  $\overline{\text{INTA}}$  dar el trebuie folosit numai în al doilea ciclu  $\overline{\text{INTA}}$  deoarece la începutul primului ciclu microprocesorul nu garantează starea de impedanță înaltă a magistralei locale:  $\text{TCLAZ}_{\text{max}}=80\text{ns}$ , în timp ce MCE poate fi activat cu o întârziere maximă de  $\text{TSVMCH}_{\text{max}}=15\text{ns}$ . De asemenea, 8259A, controlorul de întreruperi folosit în mod uzual în sistemele cu 8086 sau 8088, generează adresa de cascadă în al doilea ciclu  $\overline{\text{INTA}}$ . Selecția celui de-al doilea semnal MCE se face cu ajutorul semnalului  $\overline{\text{LOCK}}$ . Reamintim că în modul minim acest semnal neutilizabil este generat intern, la ieșirea corespunzătoare a procesorului generându-se comanda de scriere  $\overline{\text{WR}}$ . În modul maxim  $\overline{\text{LOCK}}$  va fi activat în T2 din primul ciclu  $\overline{\text{INTA}}$  și dezactivat în starea T2 a celui de-al doilea ciclu, figura 1.25. Întârzierea la schimbarea stării semnalului  $\overline{\text{LOCK}}$ , TCLAV, figura 1.26, este aceeași cu întârzierea de validare a adreselor,



cuprinsă între 10ns și 110ns. Timpul de stabilizare minim al adresei de cascadă, față de frontul negativ al semnalului de *latch*-are ALE, este condiționat atât de întârzierea apariției adresei la ieșirile controlorului de întreruperi cât și de raportul între semnalele MCE și ALE. Suprapunerea minimă între MCE și ALE garantează, din al doilea punct de vedere, un timp minim de stabilizare de  $TCLCH_{\min} + TCHLL_{\min} - TCLMCH_{\max} = 118ns + 0ns - 15ns = 103ns$ . Timpul de menținere al adresei de cascadă este asigurat, de asemenea, din punct de vedere al semnalului MCE, mai durând după ALE cel puțin  $TCHCL_{\min} + TCVNX_{\min} - TCHLL_{\max} = 69ns + 10ns - 15ns = 64ns$ .

În modul de lucru *magistrală de I/O*, 8288 nu mai generează semnalul MCE și toate operațiile de intrare/ieșire, inclusiv ciclul  $\overline{INTA}$ , se presupune că se referă la dispozitive aflate pe magistrala locală și nu pe magistrala de sistem. În acest caz nu mai este necesară validarea unei adrese de cascadă pe magistrala de sistem, vezi și figura 1.26A, prin capturare în *latch*-urile de adresă. În această situație toate întreruperile mascabile vin din zona locală a sistemului. În acest mod de lucru, în timpul ciclilor  $\overline{INTA}$  se va genera semnalul de validare  $\overline{PDEN}$  în locul semnalului DEN utilizabil în cazul când și pe magistrala locală se folosesc *transceiver*-e.

### 1.3.3.6. Introducerea stărilor de așteptare

Introducerea stărilor de așteptare, TW, se face la fel ca în modul minim între stările T3 și T4, prin implementarea uneia din cele două metode: sisteme normale *ready* sau sisteme normale *not ready*. Timpul minim oferit dispozitivelor de memorie sau de I/O pentru a dezactiva, trece pe "0", intrarea READY a microprocesorului, timp măsurat între momentul activării unei comenzi și sfârșitul stării T2 (este vorba despre comenzile de citire inclusiv  $\overline{INTA}$  și cele avansate), este dat de  $TCLCL - TCLML_{\max} - TRYLCL_{\min} = 200ns - 35ns - (-8ns) = 173ns$ . După cum se vede în figura 1.24 comenzile normale sunt validate în T3 ele neasigurând generarea corectă a semnalului READY. Având în vedere însă că 8288 generează în același timp comenzi normale și comenzi avansate, la generarea semnalului READY se pot utiliza întotdeauna numai comenzile avansate beneficiind de timpul calculat mai sus. În acest timp minim trebuie inclusă și întârzierea datorată circuitului 8284 de sincronizare a semnalului READY, aspect asupra căruia vom reveni în §1.4.4.4. Timpul minim oferit dispozitivelor periferice pentru activarea, trecerea pe "1", a intrării READY a procesorului, măsurat între momentul activării comenzii și frontul pozitiv al ceasului din T3, este  $TCLCL - TCLML_{\max} + TCLCH_{\min} - TRYHCH_{\min} = 200ns - 35ns + 118ns - 118ns = 165ns$ .

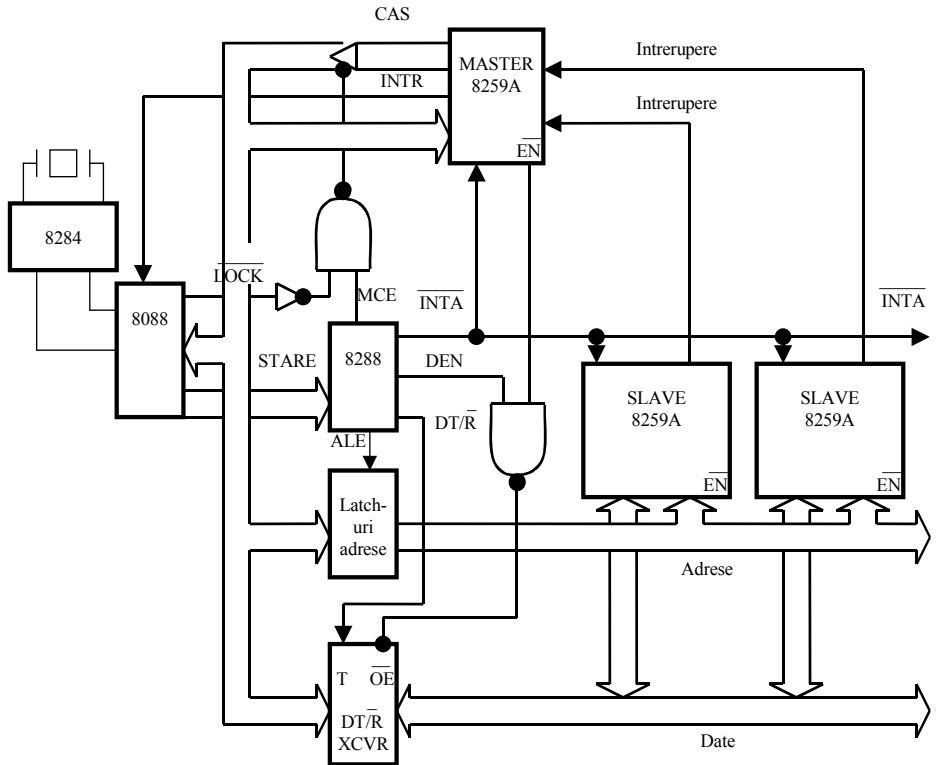


Figura 1.24A. 8086 în modul maxim cu un controlor de întreruperi *master* pe magistrala locală și mai multe controloare de întreruperi *slave* pe magistrala de sistem

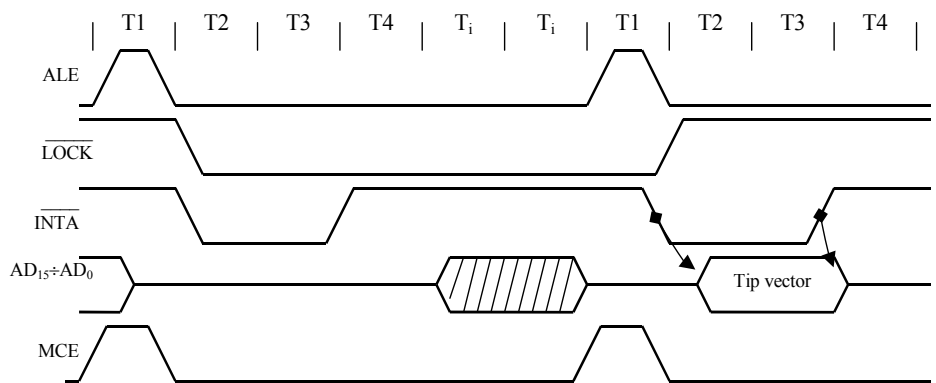


Figura 1.25. Secvența de achitare a întreruperii mascabile în modul maxim

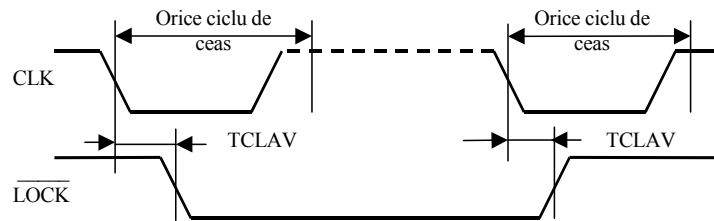


Figura 1.26. Întârzierile semnalului  $\overline{\text{LOCK}}$

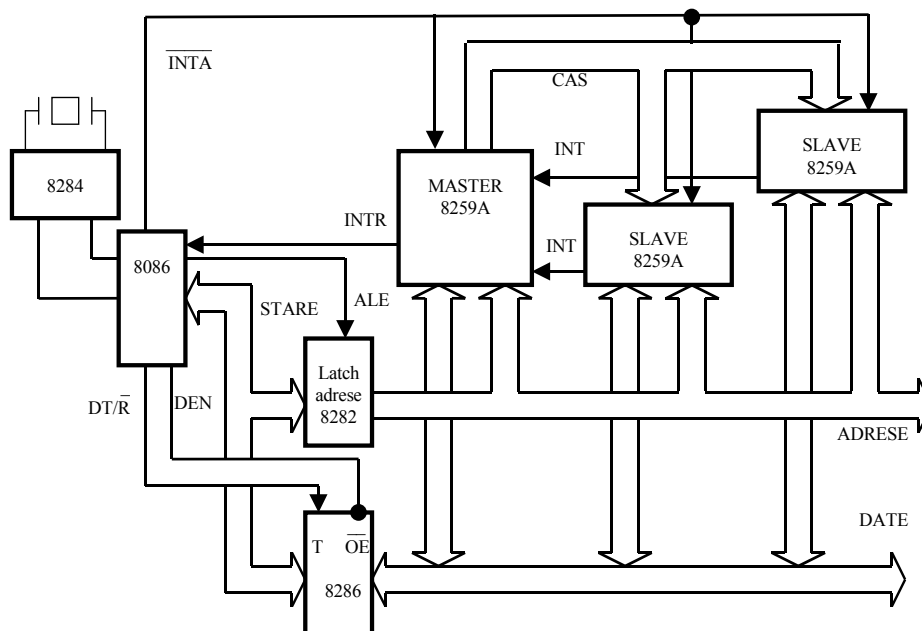


Figura 1.26A. 8086 în modul maxim cu un controlor de întreruperi *master* pe magistrala locală și mai multe controloare de întreruperi *slave* pe magistrala de sistem

### 1.3.3.7. Preluarea controlului magistralei locale (accesul direct)

În modul de lucru maxim al microprocesorului preluarea controlului magistralei locale, spre deosebire de modul minim unde aceasta se realiza printr-un procedeu de cerere/achitare cu ajutorul semnalelor HOLD și HLDA, se face prin intermediul unei secvențe mai sofisticate implementate pe două niveluri de prioritate cu ajutorul unor conexiuni bidirecționale  $\overline{\text{RQ}}/\overline{\text{GT0}}$  și  $\overline{\text{RQ}}/\overline{\text{GT1}}$ , figura 1.27.



Așa cum se poate vedea în figura 1.27 după primirea și achitarea unei cereri, 8086 va trece în starea a treia magistrala multiplexată,  $\overline{AD15}/\overline{AD0}$ ,  $\overline{BHE}/S7$ ,  $A19/S6/A16/S3$ , liniile de stare  $\overline{S2}/\overline{S0}$ , ieșirile  $\overline{LOCK}$  și  $\overline{RD}$ . Această acțiune nu va invalida ieșirile controlorului 8288 ci, datorită existenței în 8288 a unor rezistențe interne pentru semnalele  $\overline{S2}/\overline{S0}$ , se va confunda cu apariția unor stări pasive pe timpul cărora, așa cum s-a arătat, 8288 nu va emite nici-o comandă sau semnal de validare a *transceiver*-elor. În această situație, 8288 poate fi controlat de coprocesorul căruia i s-a acordat magistrala. Dacă coprocesorul nu folosește controlorul 8288 el trebuie să-i invalideze ieșirile de comandă, trecându-le în starea a treia prin poziționarea pe "1" a intrării  $\overline{AEN}$  a lui 8288.

O cerere de preluare a magistralei locale pe una din liniile  $\overline{RQ}/\overline{GT}$  se materializează printr-un impuls pe zero, figura 1.27, a cărui durată nu va fi mai mare decât perioada ceasului CLK dar care trebuie sincronizat cu ceasul microprocesorului pentru a fi asigurați timpii de stabilizare și menținere față de frontul pozitiv al ceasului cu care se eșantionează liniile  $\overline{RQ}/\overline{GT}$ . După generarea impulsului de cerere solicitantul, coprocesor sau procesor specializat, va trebui să urmărească starea liniei  $\overline{RQ}/\overline{GT}$  pe care a emis cererea. Achitarea de către microprocesor a unei cereri se face de asemenea printr-un impuls pe zero emis de microprocesor pe linia bidirecțională pe care s-a făcut cererea. Precizăm că, deoarece microprocesorul poate emite achitarea cu frontul negativ al ceasului CLK imediat următor frontului pozitiv cu care a strobato cererea, linia  $\overline{RQ}/\overline{GT}$  pe care se desfășoară dialogul poate să rămână pe "0" datorită unei scurte suprapuneri a începutului achitării cu cererea care nu s-a terminat. Această menținere pe "0" a liniei, semnificând totuși un dialog, impune, în vederea capturării unei achitări, folosirea unei logici sincrone acționate pe ceasul CLK. O logică bazată pe acționări pe front nu va fi potrivită datorită tocmai posibilității apariției unor eventuale suprapuneri, așa cum s-a explicat mai sus.

După recepționarea achitării, dispozitivul solicitant poate să preia magistrala locală. Datorită faptului că microprocesorul va trece magistrala în starea a treia după o perioadă de ceas de la emiterea achitării se impune dispozitivului solicitant să respecte la activarea magistralei întârzierea  $TCLAZ$  față de frontul negativ al ceasului CLK, figura 1.27. Această precauție este recomandată deoarece detectarea unei achitări de către dispozitivul solicitant se poate face chiar cu frontul pozitiv al ceasului cuprins în perioada de ceas în care microprocesorul emite achitarea, deci înainte de dezactivarea magistralei de către 8086. Aceasta ar putea conduce la apariția prin suprapunerea a doi emițători a unei zone de incertitudine a magistralei la preluarea ei de către solicitant.

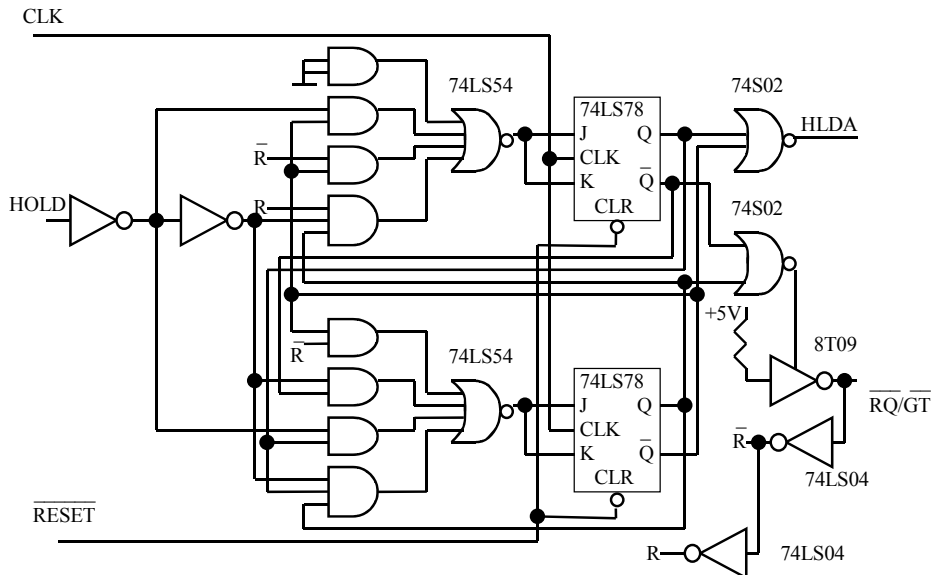


Figura 1.28. O schemă de conversie HOLD/HLDA în  $\overline{RQ}/\overline{GT}$

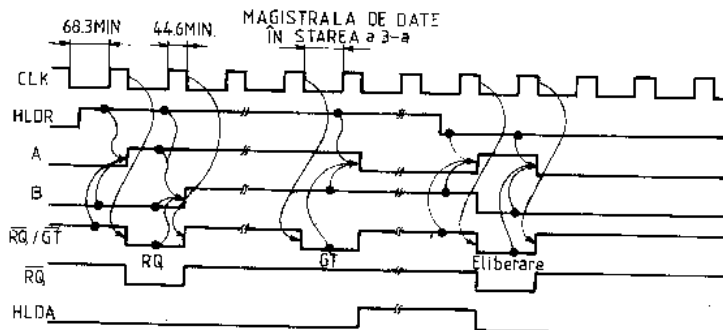


Figura 1.29. Diagramele de timp pentru schema din figura 1.28

Pentru a reda controlul magistralei procesorului dispozitivul solicitant va trebui să elibereze magistrala și să emită pe linia  $\overline{RQ}/\overline{GT}$  un impuls de eliberare a magistralei. 8086 va activa liniile de stare  $\overline{S2} \div \overline{S0}$  și  $\overline{LOCK}$  trei perioade de ceas după detectarea impulsului de eliberare. Liniile de adrese/date vor fi activate după cel puțin un interval  $TCHCL_{min}$  față de liniile de stare. Între timp *latch*-urile de adresă și 8288-ul asociate lui 8086 vor trebui și ele activate pentru ca microprocesorul să preia efectiv controlul magistralei locale.

Pentru ilustrarea mai bună a protocolului de preluare a magistralei locale în modul maxim dăm în figurile 1.28 și 1.29 o schemă de conversie a semnalelor HOLD/HLDA în  $\overline{RQ}/\overline{GT}$  și diagrama de timp corespunzătoare [10]. Această schemă poate fi utilizată pentru adaptarea de exemplu a unui circuit de control al accesului direct în sisteme cu 8086 lucrând în modul maxim.

Răspunsul microprocesorului la o cerere pe una din liniile  $\overline{RQ}/\overline{GT}$  depinde, ca și în modul minim, de activitatea curentă pe magistrală, de starea semnalului  $\overline{LOCK}$  și de întreruperi. În [10] se dă pentru calculul întârzierii între detecția unei cereri de către microprocesor și detecția unei achitări de către solicitant următoarea formulă: (Întârzierea de la HOLD la HLDA)–(THVCH+TCHCL+TCLHAV). Aplicarea acestei formule conduce pentru situațiile în care unitatea de interfață cu magistrala UIM se află într-o stare T4 sau TI la o întârziere de o perioadă de ceas. Spre deosebire de modul minim unde dialogul cu procesorul pentru preluarea magistralei se făcea numai pe un canal HOLD/HLDA, în modul maxim pot apărea întârzieri datorită dialogului pe cele două canale  $\overline{RQ}/\overline{GT}$ . Astfel, dacă 8086 a generat o achitare pe una din liniile  $\overline{RQ}/\overline{GT}$ , o posibilă cerere apărută pe cealaltă linie nu va primi achitarea decât după ce primul solicitant va elibera magistrala. Întârzierea între eliberarea unei linii  $\overline{RQ}/\overline{GT}$  și achitarea unei cereri în așteptare emise pe cealaltă linie este în general de o perioadă de ceas, figura 1.30. Uneori, situațiile în care pe timpul preluării magistralei pe  $\overline{RQ}/\overline{GT1}$  apare o cerere pe canalul  $\overline{RQ}/\overline{GT0}$  întârzierea între eliberarea canalului 1 și achitarea canalului 0 poate fi de două perioade de ceas, în funcție de existența unei cereri de transfer în așteptare emise de unitatea de execuție. Această cerere de transfer emisă de UE va fi amânată în favoarea rezolvării cererii de pe  $\overline{RQ}/\overline{GT0}$ , dar cu "prețul" decalării achitării pe canalul 0 cu încă o perioadă de ceas. Observăm deci că microprocesorul, fiind cel mai puțin prioritar ca solicitant de magistrală, sistemul poate fi blocat, *agățat*, de unul dintre solicitanții exteriori. Ieșirea dintr-o astfel de blocare se poate face cu ajutorul unei tehnici de tip *ceas de gardă*, *watchdog*.

#### 1.3.3.8. Alte particularități ale funcționării în modul maxim

Starea cozii de așteptare dată de ieșirile QS1, QS0, indică ce tip de informații este extras din coada de așteptare internă și când aceasta din urmă este inițializată datorită unui transfer al controlului. Monitorizând liniile de stare  $\overline{S2}$ ,  $\overline{S1}$  și  $\overline{S0}$  pentru instrucțiunile extrase de microprocesor,  $-\overline{S2}, \overline{S1}, \overline{S0}=1,0,0$  indică acces-cod  $-\text{A0}$  și  $\overline{\text{BHE}}$  care precizează dacă accesul se face pe octet sau pe cuvânt, și liniile QS1 și QS0 pentru instrucțiunile care

părăsesc coada de așteptare, se poate "urmări" din exterior execuția instrucțiunilor. Această tehnică este utilizată atât de coprocesoare pentru detectarea execuției unei instrucțiuni *ESCAPE* prin care li se comunică sarcini specifice cât și, de exemplu, pentru depanare și punere la punct, în cadrul unor module de logică specializate, de tip *analizor logic*, care pot să "prindă" o execuție la o anumită adresă de memorie. Starea cozii de așteptare este validă în ciclul de ceas următor efectuării operației.

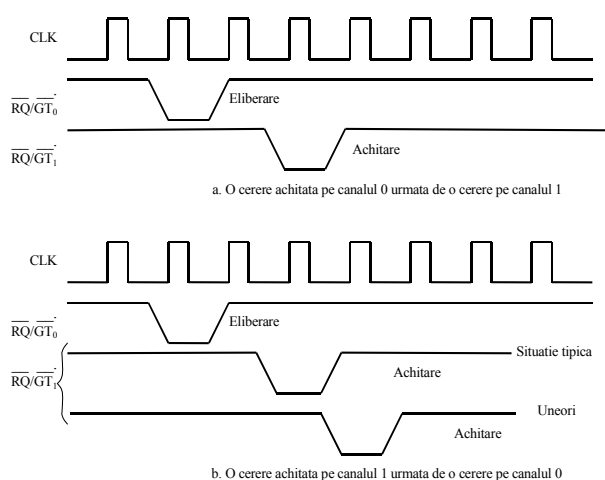


Figura 1.30. Întârzieri la achitarea cererii de preluare a magistralei pe liniile  $\overline{RQ}/\overline{GT}$

Controlul accesului la resursele divizate în sisteme complexe, multi-microprocesor, se poate face în modul de lucru maxim, așa cum am mai spus, cu ajutorul ieșirii  $\overline{LOCK}$ . Această ieșire este activată la execuția unei instrucțiuni precedate de prefixul *LOCK*, mai exact, în primul ciclu de ceas următor "execuției" prefixului. Ieșirea va rămâne activă încă un ciclu de ceas după executarea instrucțiunii precedate de prefixul *LOCK*, figura 1.31. În această figură se poate vedea, de exemplu, comportarea ieșirii  $\overline{LOCK}$  la execuția instrucțiunii *XCHG* cu prefix *LOCK* utilizate pentru implementarea unui mecanism uzual de tip *testare și poziționare semafor*:

```
lock xchg reg, memorie      ; reg este orice registru
                             ; MEMORIE este adresa
                             ; semaforului
```



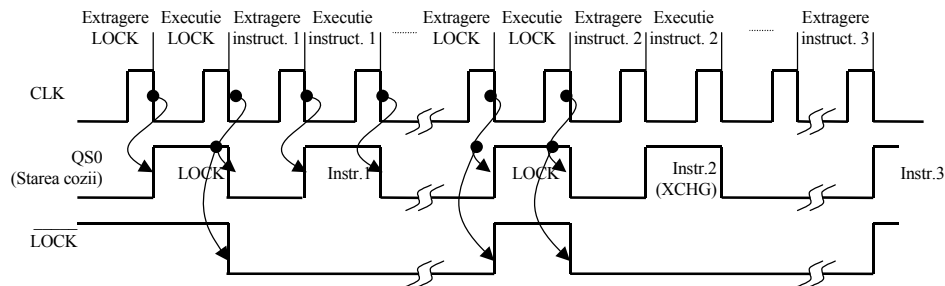


Figura 1.31. Activarea semnalului  $\overline{LOCK}$

Aici prefixul  $\overline{LOCK}$  previne accesul unei alte unități centrale la semafor, locația MEMORIE, pe timpul operației de testare și poziționare a semaforului. O altă întrebuințare eficientă a prefixului  $\overline{LOCK}$  este în cazul transferurilor pe blocuri folosite pentru schimbarea rapidă de mesaje între unitățile centrale ale sistemelor multiprocesor. În figura 1.31 starea cozii, validată în timpul ciclului de ceas următor operației efectuate asupra cozii, este reprezentată de QS0, poziționat pe "1" la extragerea din coadă a primului octet al codului-operație. În figură este prezentată execuția succesivă a trei instrucțiuni. Observăm întâi că sunt necesare două perioade de ceas pentru extragere și decodare – execuția prefixului  $\overline{LOCK}$  –, echivalentă cu întârzierea până la activarea ieșirii  $\overline{LOCK}$ . Ieșirea  $\overline{LOCK}$  va deveni așadar activă odată cu începutul instrucțiunii pe care o precede prefixul. Ea va rămâne activă încă o perioadă după terminarea instrucțiunii, trecând pe "1" între execuțiile a două instrucțiuni succesive, chiar dacă ambele sunt precedate de prefix, cazul din figură. Dacă în coada de așteptare se află numai prefixul, instrucțiunea pe care o precede nefiind încă extrasă din memorie, ieșirea  $\overline{LOCK}$  va fi activată imediat după execuția prefixului și va rămâne așa până după aducerea instrucțiunii în coadă și executarea ei. Mai trebuie făcută și observația că pe timpul execuției unei instrucțiuni "blocate", precedate de prefixul  $\overline{LOCK}$ , UIM mai poate să încarce coada de așteptare a microprocesorului, magistrala fiind de fapt numai a lui pe durata execuției instrucțiunii datorită prefixului  $\overline{LOCK}$ .

Pe timpul execuției unei instrucțiuni "blocate" orice cerere  $\overline{RQ}/\overline{GT}$  va fi înregistrată urmând a fi achitată după terminarea execuției instrucțiunii. De asemenea, întreruperile care apar în timpul execuției unui prefix nu vor fi achitate, dacă sunt evident validate, decât după execuția instrucțiunii precedate de prefix. Excepție fac instrucțiunile care permit servirea întreruperilor pe timpul execuției lor: HALT, WAIT sau instrucțiunile pe șiruri.

### 1.3.4. GENERAREA CEASULUI ȘI A SEMNALELOR DE INIȚIALIZARE ȘI READY CU AJUTORUL CIRCUITULUI 8284

#### 1.3.4.1. Circuitul 8284

Circuitul 8284, realizat în tehnologie bipolară, asigură generarea semnalelor de ceas în sistemele realizate în jurul microprocesoarelor 8086 sau 8088, sincronizarea semnalului READY și generarea semnalului de inițializare RESET. Schema-bloc a circuitului este dată în figura 1.32 iar conexiunile externe în figura 1.33. Prezentăm în continuare succint semnificația conexiunile externe.

$\overline{AEN1}, \overline{AEN2}$ , *Address Enable*, validare adresă. Intrări active pe "0" servind pentru validarea semnalelor READY corespunzătoare, RDY1 respectiv RDY2. Cele două perechi de semnale  $\overline{AEN1,2}/RDY1,2$  sunt utilizabile în configurații de sisteme în care procesorul deservit de circuitul 8284, poate avea acces la două magistrale, pe fiecare putându-se găsi mai multe unități centrale, de exemplu două magistrale de tip MULTIBUS. În configurațiile simple, cu o singură unitate centrală intrările  $\overline{AEN}$  se pot conecta la "0" validând tot timpul sincronizarea semnalelor READY.

RDY1, RDY2, *Bus Ready (Transfer Complete)*, magistrală liberă (transfer terminat). Intrare activă pe "1" indicând procesorului că transferul cu unul din dispozitivele aflate pe magistrala de date a sistemului s-a încheiat: octetul a fost recepționat sau emis.

READY, Gata. Ieșire activă pe "1" reprezentând semnalul cu care se poate comanda intrarea READY a microprocesoarelor 8086 sau 8088 pentru a le trece în starea de așteptare. Semnalul se obține prin sincronizarea cu ceasul CLK a semnalelor asincrone apărute la intrările RDY1 sau RDY2. READY va fi șters după un timp ce satisface timpul de menținere impus de microprocesor.

X1, X2. Intrări la care se conectează cuarțul a cărui frecvență trebuie să fie de trei mai mare decât frecvența dorită a ceasului microprocesorului.

$F/\overline{C}$ , *Frequency/Cristal Select*, selecție frecvență/cuarț. Intrare prin care se poate selecta sursa de obținere a semnalelor de ceas pentru microprocesor: frecvență externă la intrarea EFI sau oscilator cu cuarț.  $F/\overline{C}=1$  selectează generarea ceasului cu ajutorul intrării EFI.

EFI, *External Frequency Input*, intrare de frecvență externă. Semnalul prezentat la această intrare este de tip TTL cu factor de umplere 1/2 și frecvență de trei ori mai mare decât frecvența dorită a ceasului microprocesorului.

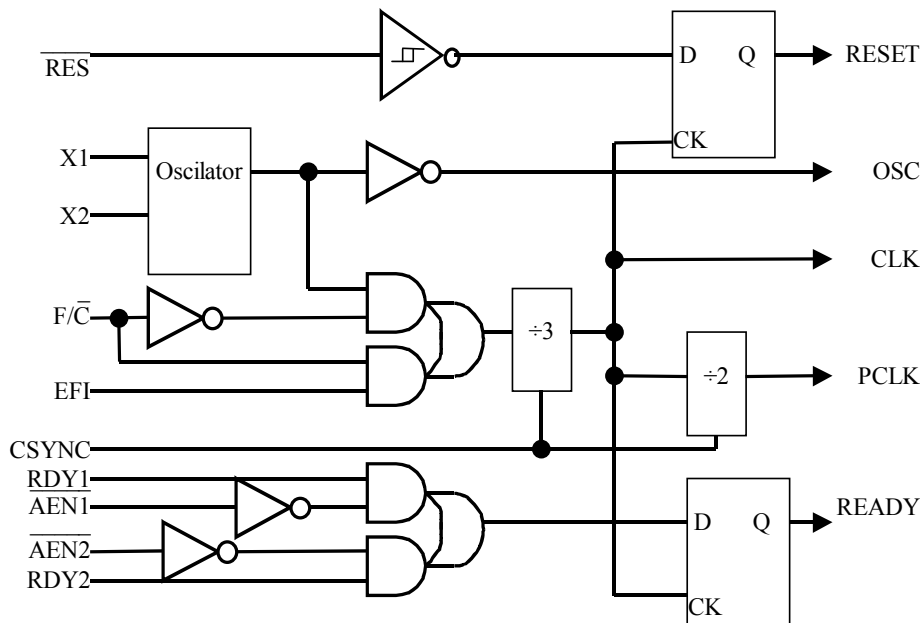


Figura 1.32. Schema bloc a circuitului 8284

CLK, *Processor Clock*, ceas pentru procesor. Ieșire de ceas utilizată de microprocesor și dispozitivele conectate pe magistrala lui locală. Frecvența ceasului CLK este de trei mai mică decât frecvența cuarțului sau a semnalului de la intrarea EFI și are un factor de umplere 1/3. Nivelul "1" al acestei ieșiri, garantat de cel puțin 4V la o tensiune de alimentare  $VCC=5V\pm 10\%$  și un consum de 1mA, permite utilizarea ceasului CLK atât pentru dispozitive bipolare cât și pentru cele MOS.

PCLK, *Peripheral Clock*, ceas pentru periferice. Ieșire de ceas utilizabilă pentru dispozitivele periferice mai lente. Frecvența este de două ori mai mică decât a lui CLK și factorul de umplere 1/2. PCLK este de tip TTL.

OSC, *Oscillator Output*, ieșirea oscilatorului. Ieșirea TTL, negată, a oscilatorului având frecvența cuarțului. Ieșirea nu este afectată de opțiunea  $F/\bar{C}$  fiind comandată direct de oscilatorului cu cuarț. Dacă nu se folosește cuarțul ci intrarea externă EFI ieșirea OSC este nedeterminată. Raporturile de frecvență între OSC, CLK și PCLK se pot vedea în figura 1.34.

$\overline{RES}$ , *Reset In*, intrare de inițializare. Intrare de tip *trigger Schmitt* activă pe "0" la care se poate conecta o rețea RC pentru generarea semnalului de inițializare la punerea sub tensiune sau cu ajutorul unui comutator.

RESET, inițializare. Ieșire activă pe "1" destinată inițializării microprocesorului și circuitelor aferente.

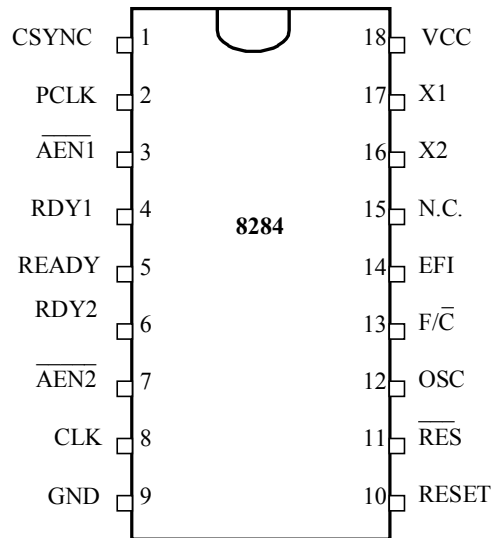


Figura 1.33. Conexiunile externe ale circuitului 8284

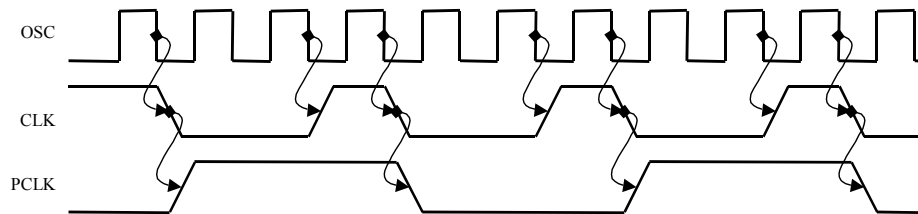


Figura 1.34. Relațiile între OSC, CLK și PCLK

CSYNC, *Clock Synchronization*, sincronizare ceas. Intrare de sincronizare activă pe "1" cu ajutorul căreia se pot controla mai multe circuite 8284 pentru a genera ceas în fază. Numărătoarele interne ale circuitului sunt inițializate cu CSYNC=1 și validate cu CSYNC=0. Generarea semnalului de sincronizare aplicat la intrarea CSYNC trebuie făcută sincron cu EFI. CSYNC este utilizat deci împreună cu EFI, în cazul folosirii ca sursă a ceasului a oscilatorului intern intrarea CSYNC legându-se la masă.

Diagramele de timp care ilustrează funcționarea circuitului se dau în figura 1.35 iar parametrii de timp în tabelele 1.9 și 1.10.

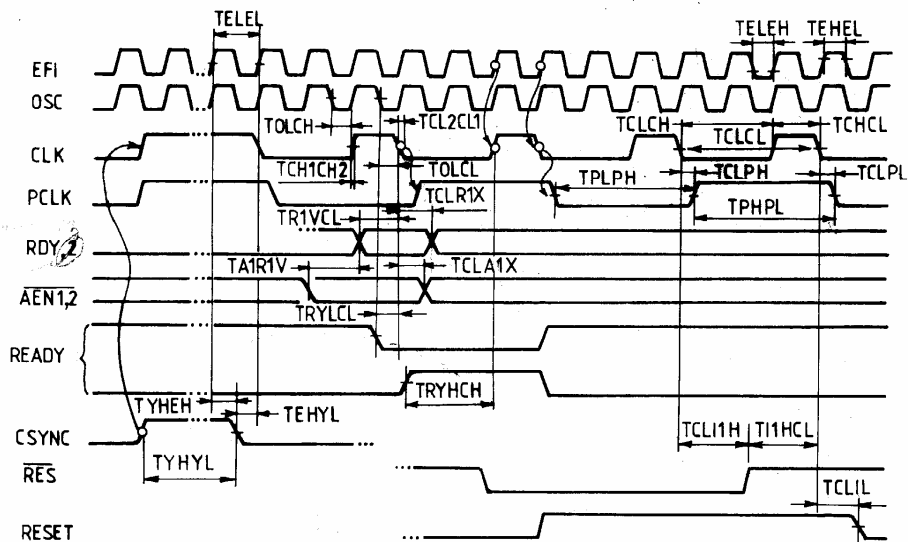


Figura 1.35. Diagramele de timp pentru circuitul 8284

Tabelul 1.9. Parametrii 8284. Cerințe de timp

Parametru	Semnificație	Valoare minimă	Valoare maximă
TEHEL	Timpul cât EFI este "1"	13ns	
TELEH	Timpul cât EFI este "0"	13ns	
TELEL	Perioada EFI	$TEHEL+TELEH+\delta^1$	
	Frecvența cuarțului	12MHz	25MHz
TR1VCL	Timpul de stabilizare al semnalelor RDY1,2 față de CLK	35ns	
TCLR1X	Timpul de menținere al semnalelor RDY1,2 față de CLK	0ns	
TA1VR1V	Timpul de stabilizare al semnalelor AEN1,2 față de RDY1,2	15ns	
TCLA1X	Timpul de menținere al semnalelor AEN1,2 față de CLK	0ns	
TYHEH	Timpul de stabilizare al semnalului CSYNC față de EFI	20ns	
TEHYL	Timpul de menținere al semnalului CSYNC față de EFI	20ns	
TYHYL	Durata lui CSYNC	$2 \cdot TELEL$	
TI1HCL	Timpul de stabilizare al semnalului RES față de CLK	65ns	
TCLI1H	Timpul de menținere al semnalului RES față de CLK	20ns	

Nota 1:  $\delta$ =durata frontului crescător + durata frontului descrescător al semnalului EFI(maximum 5ns+5ns=10ns)

Tabelul 1.10. Parametrii 8284. Răspunsuri în timp

Parametru	Semnificație	Valoare minimă	Valoare maximă
TCLCL	Perioada ceasului CLK	125ns	
TCHCL	Timpul cât CLK este "1"	$(1/3) \cdot TCLCL + 2ns$	
TCLCH	Timpul cât CLK este "0"	$(2/3) \cdot TCLCL - 15ns$	
TCH1CH2 TCL2CL1	Durata fronturilor crescător și descrescător ale ceasului CLK		10ns
TPHPL	Timpul cât PCLK este "1"	$TCLCL - 20ns$	
TPLPH	Timpul cât PCLK este "0"	$TCLCL - 20ns$	
TRYLCL	Timpul între dezactivarea semnalului READY și CLK	-8ns	
TRYHCH	Timpul între activarea semnalului READY și CLK	$(2/3) \cdot TCLCL - 15ns$	
TCLIL	Întârzierea între CLK și RESET	40ns	
TCLPH	Întârzierea între CLK și frontul crescător al PCLK		22ns
TCLPL	Întârzierea între CLK și frontul descrescător al PCLK		22ns
TOLCH	Întârzierea între OSC și frontul crescător al CLK	-5ns	12ns
TOLCL	Întârzierea între OSC și frontul descrescător al CLK	2ns	20ns

#### 1.3.4.2. Generarea ceasului

În figura 1.36 se dau caracteristicile pe care trebuie să le îndeplinească semnalul de ceas CLK al microprocesorului 8086. După cum se vede, 8086 are nevoie de un ceas cu fronturi rapide, maximum 10ns, și niveluri de tensiune "0" între -0,5V și +0,6V și "1" între 3,9V și VCC+1V. Frecvența maximă a ceasului pentru 8086, varianta standard, este de 5MHz. Frecvența minimă a ceasului, având în vedere existența în interiorul microprocesorului a unor celule de memorie dinamică, este de 2MHz. Din cauza acestei ultime restricții funcționarea pas-cu-pas a lui 8086, la nivel de instrucțiune sau de ciclu-mașină, nu poate fi implementată hardware prin blocarea ceasului. Ea este posibilă numai prin poziționarea software a indicatorului de condiții T, așa cum vom vedea în §1.4.5.1 și în §2.3.6. Pentru a satisface cerințele minime de timp care să asigure o funcționare internă optimă, ceasul microprocesorului se recomandă, după cum am menționat, să aibă un factor de umplere 1/3. Această recomandare apare mai stringentă cu cât ne apropiem de frecvența maximă de lucru. Circuitul 8284, prezentat în capitolul anterior, asigură generarea unui ceas cu factor de umplere 1/3, nivelurile de tensiune și fronturile lui respectând cerințele impuse de microprocesor.

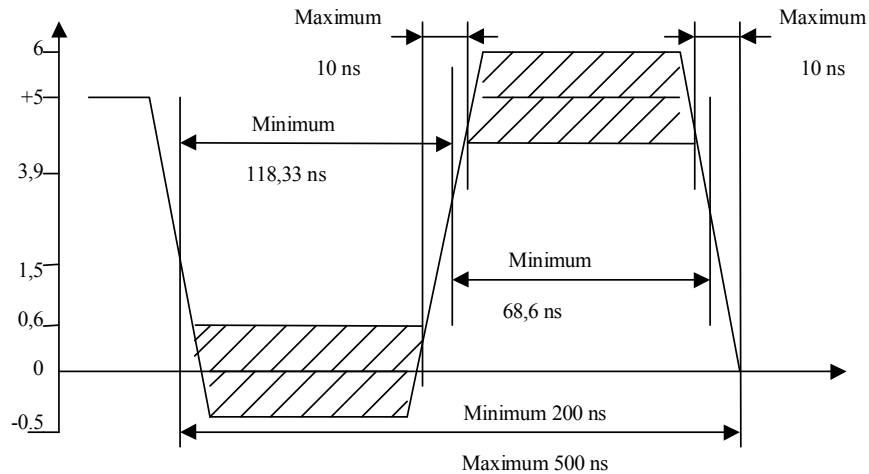


Figura 1.36. Caracteristicile ceasului pentru microprocesorul 8086

După cum se poate deduce din prezentarea conexiunilor externe și a schemei bloc ale circuitului 8284, ca sursă pentru generarea ceasului se poate folosi fie un semnal extern la intrarea EFI, fie oscilatorul din interior proiectat să oscileze cu un cuarț extern conectat între X1 și X2 pe frecvența de rezonanță serie a acestuia. Sursa trebuie să oscileze pe o frecvență de trei ori mai mare decât frecvența dorită a ceasului CLK. Pentru o generare cât mai stabilă și precisă a ceasului fabricanții circuitului recomandă utilizarea cuarțurilor ce oscilează pe frecvența fundamentală și au o rezistență serie cât mai mică. De asemenea, deoarece oscilatorul nu apare pentru cuarț ca o sursă ideală de semnal având o componentă inductivă, pentru a anula efectul acestei componente – oscilația pe o frecvență mai mică decât frecvența de rezonanță serie pură –, trebuie adăugat în serie cu cristalul, la intrarea X2, un condensator, CL în figura 1.37. Acest condensator servește, de asemenea, și la separarea în curent continuu a cuarțului ceea ce asigură o protecție a acestuia din urmă împotriva unor polarizări în tensiune continuă care ar putea să deranjeze și chiar să distrugă structura cristalină a cuarțului.

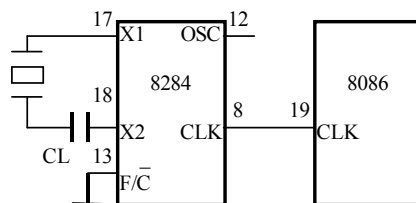


Figura 1.37. Conectarea unui cuarț la 8284

Impedanța condensatorului,  $XCL$ , depinde de frecvența de oscilație, fiind data de:

$$XCL = \frac{1}{2 \cdot \pi \cdot F \cdot CL}$$

Se recomandă ca rezistența-serie a cristalului plus  $XCL$  să fie mai mică decât  $1k\Omega$  pentru a nu opri funcționarea oscilatorului prin reducerea amplificării în buclă sub 1.

Valoarea condensatorului descrește deci cu frecvența cristalului fiind, de exemplu, aproximativ 24pF la 12MHz și 8pF la 22MHz. Dacă sistemul nu necesită alegerea precisă a unei frecvențe, ceea ce ar fi impus corelarea exactă a valorii condensatorului cu frecvența de rezonanță serie a cuarțului, ci doar stabilitatea ei, fabricanții recomandă utilizarea unui condensator de 12÷15pF pentru un cuarț de 15MHz în vederea generării ceasului CLK la 5MHz. Circuitul imprimat, în general elementele tehnologice folosite, fire, lipituri, socluri, pot modifica valoarea componentei inductive și deci conduce la schimbarea valorii condensatorului CL.

Intrarea EFI se utilizează atunci când este necesară o sursă de semnal a cărei frecvență să fie foarte precisă sau să fie variabilă și pentru situațiile când mai multe circuite 8284 comandând unități centrale care trebuie sincronizate, au nevoie de o sursă comună, unică, de oscilație, figura 1.38. Semnalul extern aplicat la intrarea EFI a circuitului trebuie să fie compatibil TTL, să aibă un factor de umplere 1/2 și o frecvență de trei ori mai mare decât frecvența dorită a ceasului CLK. Frecvența maximă a acestui semnal este puțin peste 24MHz, timpul minim pe "0" sau pe "1" fiind de 13ns. Frecvența minimă este impusă de microprocesor. La folosirea unei surse comune pentru generarea ceasului de către mai multe circuite 8284 distribuite în cadrul sistemului fiecare 8284 va trebui comandat direct de la sursă printr-o legătură proprie. Se recomandă, pentru minimizarea zgomotului, ca aceste legături să fie realizate cu fir bifilar, torsadat, semnalul fiind emis și recepționat cu porți ca 74LS04, având fronturi mai lente, masa firului torsadat conectând mesele sursei și receptorului, figura 1.37. De asemenea, pentru micșorarea alunecării ceasului aceste legături trebuie să fie de aceeași lungime. Cu toate acestea variația întârzierii între EFI și ceasurile CLK generate de mai multe circuite 8284, ca în figura 1.38, poate ajunge la 35÷40ns. Această variație se reduce la 15÷25ns dacă circuitele sunt împachetate în același tip de capsulă, au același tensiune de alimentare și lucrează la aceeași temperatură.

Pentru sincronizarea ceasurilor cu evenimente externe proiectanții au prevăzut intrarea CSYNC: CSYNC=1 forțează ieșirile de ceas CLK și PCLK pe "1" iar CSYNC=0 validează generarea ceasurilor, prin pornirea numărătoarelor,



cu primul front pozitiv al semnalului emis de sursa de frecvență, oscilatorul cu cuarț sau EFI. CSYNC trebuie să fie activ pe "1" cel puțin două perioade ale semnalului emis de sursa de frecvență. În figurile 1.39a și b se prezintă două moduri de obținere a unui semnal de comandă CSYNC sincronizat, pornind de la o condiție de sincronizare externă și funcție de sursa de generare a ceasurilor, EFI respectiv OSC. Cele două bistabile din figura 1.39a sunt acționate cu  $\overline{EFI}$  pentru a se asigura timpii de stabilizare și menținere impuși comenzii CSYNC, figura 1.39c. Această negare a semnalului sursă nu mai este necesară la utilizarea oscilatorului local deoarece OSC reprezintă ieșirea negată a semnalului emis de acesta. Încă o observație: deoarece activarea lui CSYNC poate conduce, prin trecerea imediată pe "1" a ceasului CLK, la violarea timpului TCLCH de minimum 118ns, se recomandă ca această activare să se facă fie pe timpul inițializării, fie pe timpul cât CLK=1. Dacă se face pe timpul inițializării, invalidarea lui CSYNC trebuie realizată cu minimum patru perioade ale ceasului CLK înainte de terminarea RESET-ului pentru a se garanta inițializarea microprocesorului.

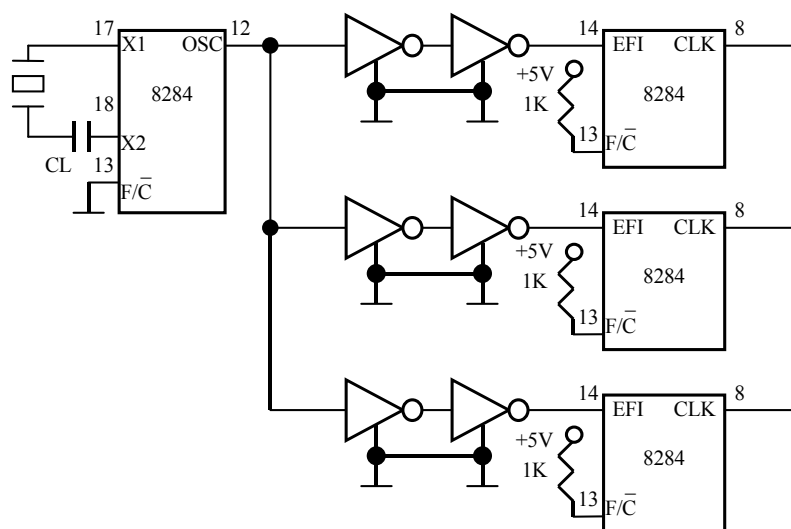


Figura 1.38. Mai multe circuite 8284 având o sursă comună de oscilație



### 1.3.4.3. Inițializarea

Inițializarea microprocesorului 8086 se realizează prin activarea, trecerea pe "1", a conexiunii RESET. Durata minimă de activare a intrării RESET este de patru perioade ale ceasului CLK, cu excepția inițializării la punerea sub tensiune când este necesar un impuls de 50μs. La apariția semnalului RESET 8086 va termina operația în curs și va rămâne inactiv până la terminarea impulsului de inițializare. Semnalul RESET, sincronizat intern, va declanșa după revenirea sa pe "0" o secvență internă de inițializare de aproximativ zece cicluri CLK. După acest interval, microprocesorul își va relua funcționarea normală începând cu instrucțiunea de la adresa FFFF0H.

La inițializare, microprocesorul 8086 va elibera magistrala de date, adrese și comenzi conform tabelului 1.11. Așa cum se vede magistrala multiplexată va fi trecută odată cu activarea intrării RESET în starea a treia. O altă parte a semnalelor va fi întâi poziționată pe "1", pe durata unei semiperioade a ceasului, când CLK este "0" – figura 1.40, după care va fi trecută în starea a treia. În modul minim ALE și HLDA vor fi inactivate, trecute pe "0". În modul maxim liniile  $\overline{RQ}/\overline{GT}$  sunt, de asemenea, inactivate, trecute pe "1", iar liniile de stare a cozii QS1,0 devin 0,0 – *nici o operație*. Deoarece în această din urmă situație la inițializare indicatorii stivei, după cum se vede, nu sunt reșetați, QS1,0=1,0, se impune, în sistemele unde se folosește o logică externă de urmărire a stării cozii, inițializarea ei cu semnalul general RESET.

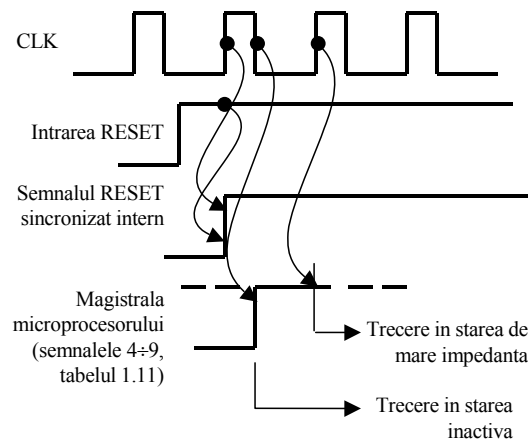


Figura 1.40. Inactivarea magistralei microprocesorului 8086 la inițializare

Tabelul 1.11. Starea magistralei microprocesorului 8086 la inițializare

Denumirea semnalului	Starea semnalului la inițializare
AD15÷AD0	Starea a 3-a
A19/S6÷A16/S3	Starea a 3-a
$\overline{\text{BHE}} / \text{S7}$	Starea a 3-a
$\overline{\text{S2}}(\text{M} / \overline{\text{IO}})$	Trecut pe "1", apoi în starea a 3-a
$\overline{\text{S1}}(\text{DT} / \overline{\text{R}})$	Trecut pe "1", apoi în starea a 3-a
$\overline{\text{S0}}(\overline{\text{DEN}})$	Trecut pe "1", apoi în starea a 3-a
LOCK / $\overline{\text{WR}}$	Trecut pe "1", apoi în starea a 3-a
$\overline{\text{RD}}$	Trecut pe "1", apoi în starea a 3-a
INTA	Trecut pe "1", apoi în starea a 3-a
ALE	"0"
HLDA	"0"
$\overline{\text{RQ}} / \overline{\text{GT0}}$	"1"
$\overline{\text{RQ}} / \overline{\text{GT1}}$	"1"
QS0	"0"
QS1	"0"

Pentru a se asigura starea inactivă a liniilor de comandă și control ale microprocesorului în special în sistemele unde curenții de scurgere sau capacitatea parazită pot conduce la niveluri de tensiune "1" sub nivelul admis de dispozitivele utilizate, se recomandă legarea la VCC a acestor linii cu rezistențe de 22k $\Omega$ . În sistemele care lucrează în modul maxim această cerință este asigurată de 8288 care conține rezistențe interne legate la VCC. Datorită acestor rezistențe 8288 interpretează la inițializare liniile de stare S2÷S0, pe "1", ca reprezentând starea pasivă a microprocesorului și își poziționează ieșirile conform tabelului 1.12. Dacă inițializarea apare în timpul unui ciclu de magistrală trecerea liniilor S2÷S0 în starea pasivă va conduce la terminarea ciclului și revenirea liniilor de comandă în starea inactivă. Reținem deci că 8288 nu își trece ieșirile de comandă în starea a treia atunci când S2÷S0 semnifică starea pasivă a microprocesorului. Dacă în sistem, la inițializare, este necesară această trecere a comenzilor în starea de impedanță înaltă, semnalul de inițializare va trebui conectat la intrarea  $\overline{\text{AEN}}$  a lui 8288 precum și, pentru invalidarea întregii unități centrale, la intrările de validare-ieșire,  $\overline{\text{OE}}$ , ale *latch*-urilor de adrese. *Transceiver*-ele pentru date sunt dezactivate cu ajutorul ieșirii DEN, figura 1.41. Un nivel "1" corect al ieșirilor de comandă ale circuitului 8288 la trecerea lor în starea a treia se asigură prin legarea lor la VCC cu ajutorul unor rezistențe de 2,2k $\Omega$ .

Tabelul 1.12. Ieșirile circuitului 8288 la inițializare

Denumirea semnalului	Starea semnalului la inițializare
ALE	0
DEN	0
DT/ $\bar{R}$	1
MCE/ $\bar{PDEN}$	0/1
Comenzi	1

Semnalul de inițializare a microprocesorului, RESET, poate fi generat, așa cum am spus în §1.4.4.1, cu ajutorul circuitului 8284. Comanda acestui semnal se face la intrarea de tip *trigger Schmitt*,  $\bar{RES}$ , figura 1.42a. După cum se știe, astfel de intrări asigură, prin utilizarea unei reacții interne pozitive care accelerează tranzițiile lente și fixează praguri diferite pentru fronturile pozitive respectiv negative, transformarea semnalelor de intrare cu variație lentă în semnale de ieșire clar definite, fără oscilații, stabile. Diferența între cele două praguri, *hysteresis*-ul la intrarea  $\bar{RES}$ , este de minimum 0,25V. Aceasta înseamnă că un semnal "0" la intrarea  $\bar{RES}$ , a cărui valoare maximă este  $V_{ILmax}=0,8V$ , va rămâne activ în tranziția pozitivă spre valoarea "1" până la 1,05V. În sensul invers un semnal "1" la intrarea  $\bar{RES}$ , a cărui valoare minimă este  $V_{IHmin}=2,6V$ , va fi considerat în continuare "1", în tranziția spre "0", până va scădea sub 2,35V.

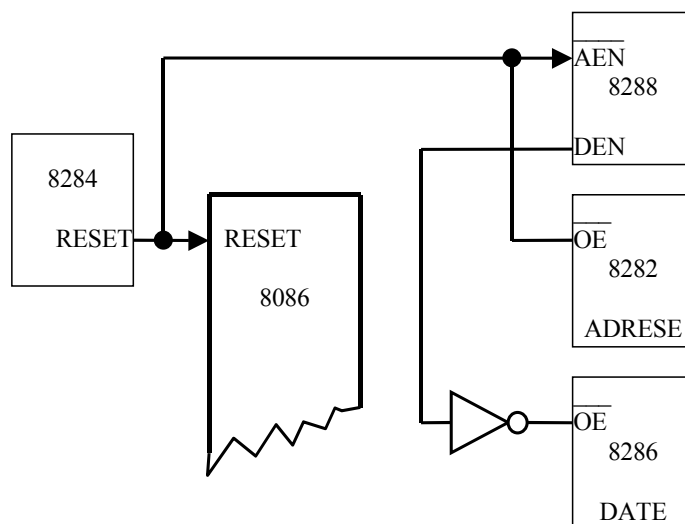


Figura 1.41. Dezactivarea la inițializare a controlorului 8288, a *latch*-urilor de adrese și a *transceiver*-elor de date

La punerea sub tensiune, intrarea  $\overline{\text{RES}}$  trebuie să rămână sub 1,05V cel puțin 50 $\mu\text{s}$  după ce tensiunea de alimentare VCC a atins 4,5V, figura 1.42b. *Hysteresis*-ul de la intrarea  $\overline{\text{RES}}$  permite asigurarea acestei cerințe prin conectarea unui circuit RC simplu, figura 1.42a. Constanta RC va rezulta din formula obișnuită:

$$V(t) = V \left( 1 - e^{-\frac{t}{RC}} \right),$$

unde  $V=4,5\text{V}$ . Valoarea constantei RC rezultată din această formulă nu ține cont de timpul în care tensiunea de alimentare atinge 4,5V și de sarcina acumulată în condensator în acest timp. Pentru  $t=50\mu\text{s}$  și  $V(t)=1,05\text{V}$ , în formula de mai sus, se obține întâi constanta de timp  $RC=188 \cdot 10^{-6}$  și apoi durata inițializării de 162 $\mu\text{s}$  după ce tensiunea de alimentare a atins 4,5V. Această durată reprezintă timpul de creștere a tensiunii la intrarea  $\overline{\text{RES}}$  până la valoarea de comutare  $V(t)=2,6\text{V}$ . Dacă se dorește o precizie mai bună pentru durata inițializării încărcarea condensatorului de la intrarea  $\overline{\text{RES}}$  poate fi făcută cu un generator de curent constant care va asigura o creștere liniară și nu invers exponențială a tensiunii la intrarea  $\overline{\text{RES}}$ .

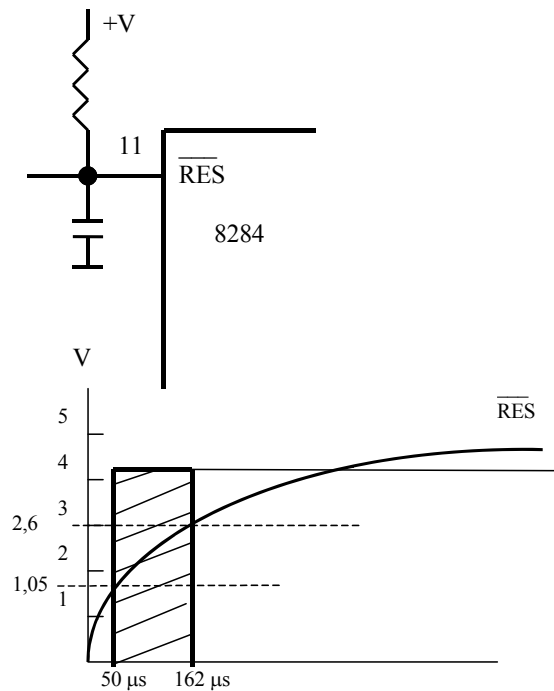


Figura 1.42. Comanda inițializării cu ajutorul circuitului 8284 : circuitul RC de la intrarea  $\overline{\text{RES}}$  (a); semnalul aplicat la intrarea  $\overline{\text{RES}}$  (b)

### 1.3.4.4. Generarea semnalului READY

Pentru formarea către microprocesor a semnalului READY, necesar pentru introducerea stărilor de așteptare, care să satisfacă cerințele de timp prezentate în §1.4.2.5 și §1.4.3.6, se poate utiliza, așa cum am mai spus, circuitul 8284 care are prevăzute în acest scop intrările separate RDY1 și RDY2 validate de  $\overline{AEN1}$ , respectiv  $\overline{AEN2}$ . La aceste intrări se va conecta logica propriu-zisă de generare a stărilor de așteptare realizată în funcție de configurația sistemului și dispozitivele de memorie și/sau I/O folosite. Cele două intrări RDY1 și RDY2, validate, sunt trecute printr-o poartă SAU LOGIC formând un semnal intern eșantionat la începutul fiecărui ciclu de ceas, figura 1.43. Semnalul eșantionat este generat la ieșirea READY cu o întârziere de maximum 8ns, ceea ce satisface timpii de stabilizare la intrarea READY a microprocesorului, figura 1.15. Semnalul eșantionat, fiind memorat, nu se poate modifica decât cel mai devreme cu următorul front negativ al ceasului CLK, ceea ce asigură și timpii de menținere impuși de 8086. Semnalele generate la intrările RDY1 și RDY2 ale circuitului 8284 trebuie să satisfacă timpul de stabilizare minim  $TR1VCL_{min}=35ns$  față de frontul negativ al ceasului iar cele generate la intrările AEN, minimum  $TR1VCL_{min} + TA1VR1V_{min} = 35ns+15ns = 50ns$ .

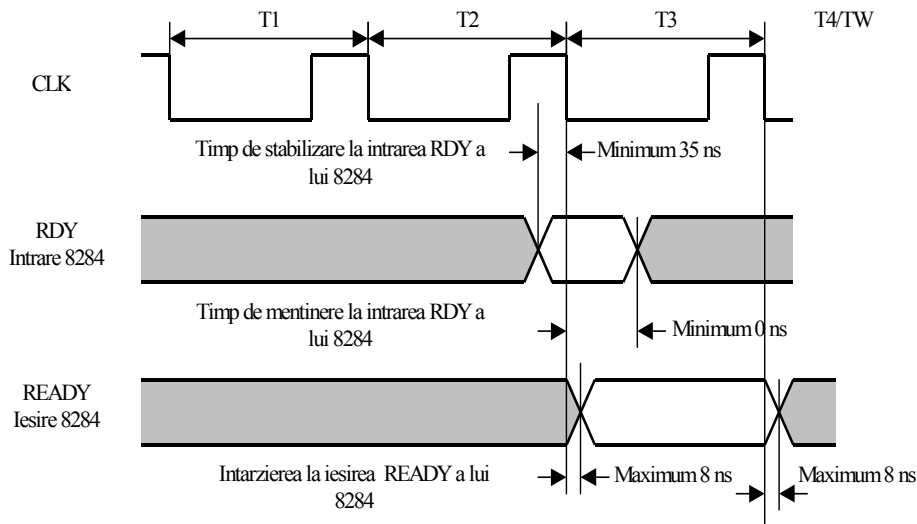


Figura 1.43. Relații de timp la generarea semnalului READY cu ajutorul circuitului 8284

Pentru sistemele care au nevoie de o singură intrare RDY intrarea  $\overline{AEN}$  corespunzătoare se poate lega la GND, cealaltă intrare de validare conectându-se la VCC printr-o rezistență de aproximativ  $1k\Omega$ , figura 1.44a. Dacă semnalul generat de logica de introducere a stărilor de așteptare este activ pe "0" se pot utiliza intrările  $\overline{AEN}$ , figura 1.44b. Un exemplu de logică pentru inserarea unei singure stări TW, suficientă pentru marea majoritate a dispozitivelor de memorie sau I/O care nu pot să lucreze la viteza maximă a microprocesorului, este dat figura 1.45. Selecția unuia dintre dispozitivele lente din sistem, la noi  $\overline{CS1}$  sau  $\overline{CS2}$ , conduce la bascularea bistabilului și trecerea semnalului RDY pe "0", figura 1.46. Trecerea pe "0" a bistabilului se face cu frontul pozitiv din T2, ceea ce asigură formarea corectă a semnalului READY către microprocesor și introducerea unei stări TW după T3. Revenirea pe "1" a bistabilului cu frontul pozitiv din T3 indică procesorului ieșirea din TW și intrarea în T4 pentru a încheia ciclul. Următoarea trecere pe "0" a bistabilului în TW nu mai influențează desfășurarea operației. Schema se inițializează cu ALE la începutul fiecărui ciclu-mașină.

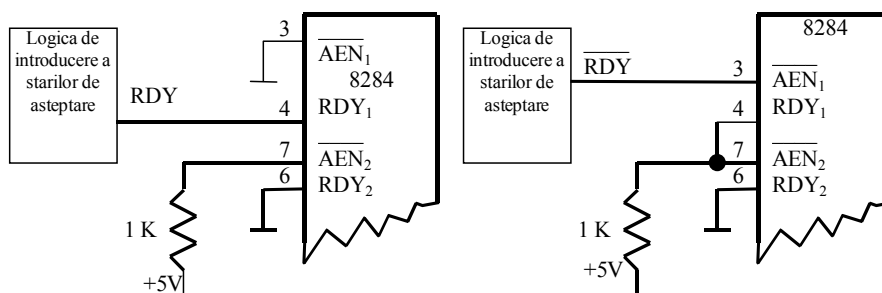


Figura 1.44. Generarea semnalului READY cu ajutorul circuitului 8284 : utilizarea intrărilor RDY1 sau RDY2 (a); utilizarea intrărilor  $\overline{AEN1}$ ,  $\overline{AEN2}$  (b)

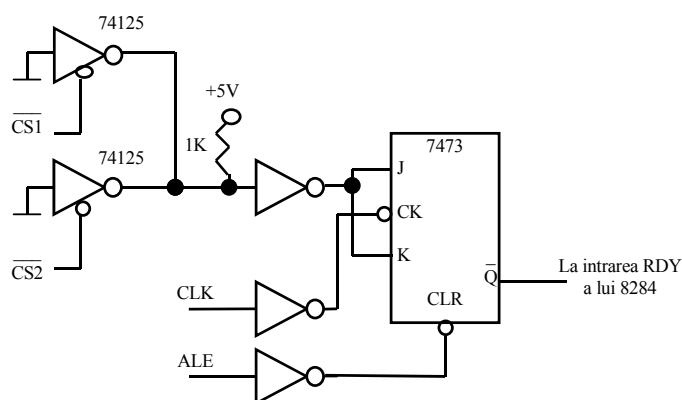


Figura 1.45. Un circuit pentru inserarea unei singure stări TW



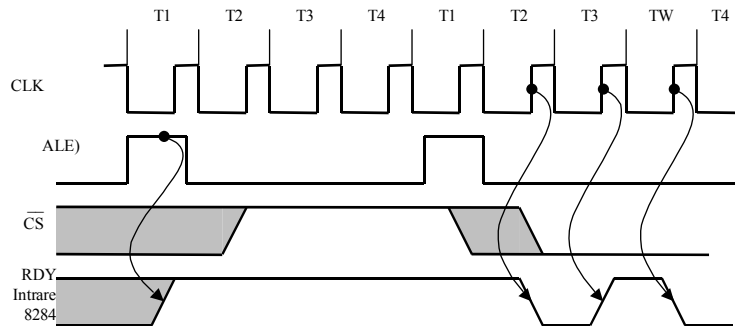


Figura 1.46. Diagrama de funcționare a circuitului din figura 1.45

### 1.3.5. SISTEMUL DE ÎNTRERUPERI

La 8086 întreruperile pot fi cauzate atât de evenimente hardware, cât și de evenimente software. Întreruperile hardware, generate de circuite exterioare microprocesorului, pot fi mascabile sau nemascabile. Întreruperile software, nemascabile, grupate mai demult în categoria *derute* [12], sunt generate de logica programelor executate pe 8086.

Sistemul de întreruperi al procesorului 8086 are la bază o tabelă a vectorilor de întrerupere plasată în memoria sistemului între adresele 00000H și 003FFH, figura 1.47. Fiecare vector constă din doi octeți reprezentând numărătorul de program, *pointer*-ul de instrucțiune, IP, și doi octeți ce specifică valoarea asociată a segmentului de cod, registrul CS. Cele două valori, IP și CS, formează adresa rutinei de serviciu (vezi capitolul 1.2). Tabela conține maximum 256 de vectori de întrerupere precizând deci adresele de început ale rutinelor de serviciu ale întreruperilor. Aceste rutine pot fi plasate oriunde în spațiul de adresare de 1Moctet al microprocesorului. Fiecărui vector din tabelă îi este asociat, așa cum se poate vedea în figura 1.47, un număr ce reprezintă tipul întreruperii. Numărul, preluat din logica externă sau din program, multiplicat cu 4, dă deplasamentul față de începutul tabelii la care se găsește vectorul de întrerupere asociat. Acest sistem de întreruperi "vectorizat" este deosebit de flexibil permițând utilizatorului să specifice cum crede de cuviință adresele de memorie ale tuturor rutinelor de serviciu întrebuintate. Dacă nu sunt definite toate cele 256 de întreruperi, utilizatorul va preciza numai adresele rutinelor folosite, recomandându-se, totuși, cel puțin, în perioada de punere la punct, asignarea tipurilor de întrerupere neutilizate cu adresa unei rutine separate pentru detectarea ușoară a întreruperilor parazite.

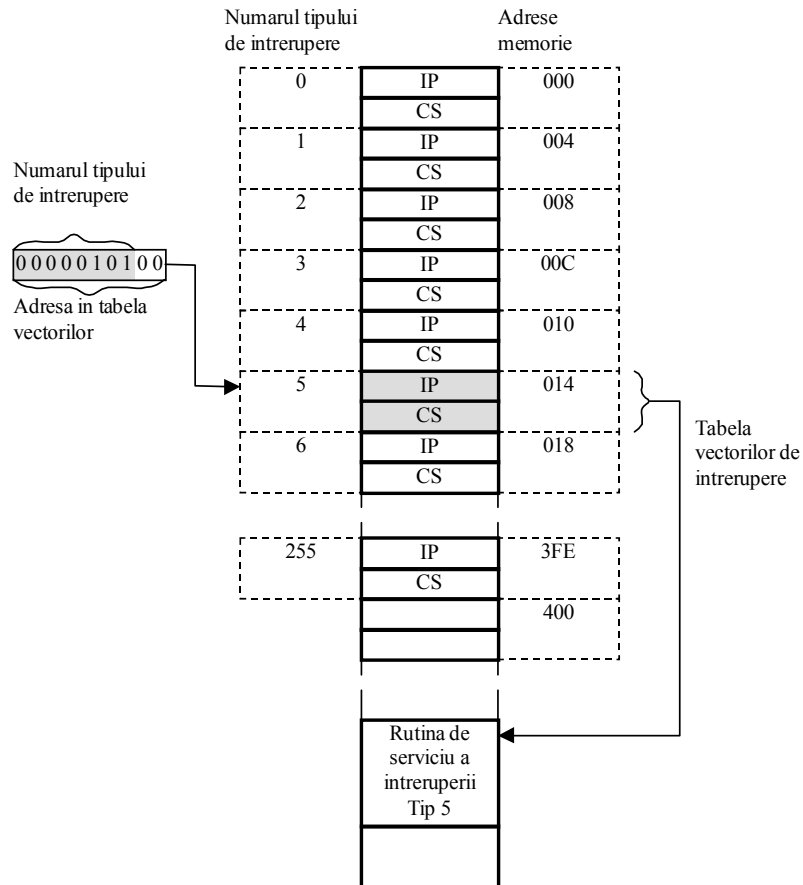


Figura 1.47. Tabela vectorilor de întrerupere la 8086

Întreruperile microprocesorului 8086 se mai pot clasifica în trei grupe: întreruperi predefinite generate de funcții speciale, întreruperi hardware definite de utilizator și întreruperi software definite de utilizator. Vom descrie pe scurt aceste tipuri de întreruperi, vezi și figura 1.47.

### 1.3.5.1. Întreruperi predefinite

La invocarea din hardware sau din software a unei întreruperi predefinite microprocesorul va transfera controlul rutinei a cărei adresă este specificată de vectorul asociat tipului de întrerupere. Utilizatorul are sarcina să scrie rutinele de serviciu și să inițializeze tabela vectorilor cu adresele corespunzătoare.

*Tipul 0* – împărțire la 0. Această întrerupere este invocată la orice încercare de împărțire pentru care câtul depășește valoarea maximă, cum este, de exemplu, cazul împărțirii la zero. Întreruperea nu este mascabilă și poate fi considerată ca o secvență aparținând operației de împărțire.

*Tipul 1* – pas-cu-pas. Întreruperea apare după o instrucțiune de la poziționarea indicatorului de condiție *Derută*, T, în registrul de stare al microprocesorului. Se permite astfel introducerea software a funcționării pas-cu-pas în cadrul unei secvențe de program. Secvența de întrerupere începe cu salvarea registrului de stare și a numărătorului de program și ștergerea indicatorului T permițându-se astfel execuția normală, nu pas-cu-pas, a rutinei de serviciu specifice. La revenire în programul principal, modul de execuție pas-cu-pas, este asigurat prin restaurarea IP, CS și a indicatorilor de condiții, deci și a lui T. Nici această întrerupere nu poate fi mascată.

*Tipul 2* – întrerupere nemascabilă, NMI. Întreruperea este de tip hardware și are prioritatea cea mai înaltă. Intrarea corespunzătoare a microprocesorului, conexiunea NMI, este comutată pe front și apoi sincronizată în interiorul procesorului cu ceasul CLK. Pentru ca o întrerupere NMI să fie recunoscută, semnalul intern sincronizat trebuie să fie activ cel puțin două perioade ale ceasului. De asemenea, dacă intrarea NMI rămâne pe "1" mai multă vreme, revenirea pe "0", timpul cât stă pe "0", înainte ca o nouă întrerupere să fie declanșată este de minimum două perioade CLK. Semnalul de la intrarea lui 8086 poate fi dezactivat înainte de intrarea în rutina de serviciu. O atenție deosebită trebuie acordată eliminării *spike*-urilor, impulsurilor parazite, care pot genera întreruperi. Întreruperea NMI este rezervată, de obicei, evenimentelor catastrofale cum sunt căderile de tensiune sau semnalizările de la un sistem de supraveghere de tip *ceas de gardă*, *watchdog*.

*Tipul 3* – întrerupere pe un octet. Este o întrerupere software nemascabilă invocată de o instrucțiune reprezentată pe un octet, INT 3, destinată în primul rând ca întrerupere de tip *breakpoint* pe parcursul punerii la punct a programelor.

*Tipul 4* – întrerupere la depășire. Este o întrerupere software nemascabilă care apare la execuția instrucțiunii INTO dacă indicatorul de condiție *Depășire*, O, este poziționat. Instrucțiunea permite derutarea programului la o rutină de serviciu în cazul apariției unei erori de depășire.

Întreruperile de tip 0 sau 2 pot apărea fără ca programatorul să acționeze în vreun fel specific, cu excepția unei împărțiri la 0 care ar putea fi o greșeală de programare. Întreruperile de tip 1, 3 și 4 necesită pentru a fi generate acte conștiente din partea programatorului. Toate tipurile de întreruperi prezentate mai sus, cu excepția NMI, sunt invocate software și sunt asociate direct cu câte o instrucțiune specifică.

### 1.3.5.2. Întreruperi software definite de utilizator

Întreruperile software sunt generate de instrucțiunea INT unde reprezintă numărul tipului de întrerupere definit de utilizator. Instrucțiunile INT, deci întreruperile software, nu sunt mascabile cu ajutorul indicatorului de condiții *Validare/Invalidare Întrerupere*, I. Revenirea din rutina de serviciu apelată printr-o întrerupere software se face cu instrucțiunea IRET.

Transferul controlului la o întrerupere software se face la sfârșitul instrucțiunii INT fără ca microprocesorul să inițieze pe magistrală un ciclu de achitare INTA. De asemenea, întreruperile software vor invalida întreruperile mascabile prin punerea pe "0" a indicatorilor I și T.

### 1.3.5.3. Întreruperi hardware definite de utilizator

Așa cum am mai spus, întreruperile hardware definite de utilizator sunt inițiate de circuite speciale prin activarea, trecerea pe "1", a semnalului de la intrarea INTR a procesorului. Aceste întreruperi sunt mascabile cu ajutorul bitului I din registrul de stare.

Starea intrării INTR este testată în timpul ultimului ciclu de ceas al fiecărei instrucțiuni. Excepție de la această regulă fac instrucțiunile MOV și POP cu un registru de segment, instrucțiunea WAIT, instrucțiunile pe șiruri precedate de prefixul de repetare REP.

În cazul instrucțiunilor MOV și POP cu un registru de segment microprocesorul va testa starea conexiunii INTR după executarea instrucțiunii următoare instrucțiunilor MOV și POP menționate. Astfel, se permite încărcarea unui *pointer* de stivă de 32 de biți în registrele SS și SP fără pericolul apariției unei întreruperi între cele două încărcări. O secvență cum este și cea de mai jos nu va fi deci interruptibilă:

```
mov ss, segment_stiva_nou
mov sp, pointer_stiva_nou
```

În timpul instrucțiunii WAIT care așteaptă trecerea pe "0" a intrării  $\overline{\text{TEST}}$  a microprocesorului se testează și starea semnalului la conexiunea INTR pentru a se permite executarea rutinelor de întrerupere în timpul așteptării. Particularitatea în această situație constă în faptul că atunci când se detectează o întrerupere, 8086 mai extrage încă o dată instrucțiunea WAIT înainte de a

transfera controlul rutinei de serviciu. Aceasta pentru a garanta revenirea din rutină tot la instrucțiunea de așteptare WAIT.

O altă situație specială apare și în cazul instrucțiunilor cu prefix. Deoarece prefixele sunt considerate ca parte a instrucțiunii pe care o preced, procesorul va eșantiona semnalul INTR la sfârșitul execuției instrucțiunii incluzând în aceasta și prefixele. Excepția survine la instrucțiunile pe șiruri precedate de prefixul REP. Pentru aceste instrucțiuni testarea întreruperii se face la sfârșitul fiecărei repetări. Atunci când instrucțiunile repetitive pe șiruri mai sunt precedate și de alte prefixe, de exemplu LOCK, și apare o întrerupere microprocesorul va restaura la revenirea din rutina de serviciu numai prefixul imediat precedent instrucțiunii. De aceea, pentru a nu perturba printr-o întrerupere execuția completă a instrucțiunii repetitive trebuie utilizată o secvență de program de tipul:

```
transfer_repetitiv: lock rep movs dest,cs:sursa
                   and cx,cx
                   jnz transfer_repetitiv
```

Codul obiect pe octeți generat de asamblorul 8086 pentru instrucțiunea MOVSB va fi în ordinea crescătoare a adreselor: prefixul LOCK, prefixul REP, prefixul *Depășire Segment* (:), codul propriu-zis al instrucțiunii MOVSB. La revenirea dintr-o rutină de serviciu a întreruperii va fi refăcut numai prefixul *Depășire Segment* garantându-se astfel execuția corectă a încă unui transfer. Următoarea instrucțiune verifică terminarea transferului șirului prin testarea valorii contorului asociat instrucțiunii MOVSB, registrul CX. Dacă această valoare e diferită de zero se sare la începutul secvenței pentru a se termina execuția instrucțiunii repetitive cu prefixe.

#### 1.3.5.4. Prioritatea întreruperilor și a cererii DMA

Ordinea de prioritate a întreruperilor respectă următoarele reguli:

– INTR, întreruperea hardware, este singura întrerupere mascabilă și dacă ea este detectată simultan cu alte întreruperi, ștergerea indicatorului I din registrul de stare o va masca. Aceasta face ca INTR să fie de fapt întreruperea cu prioritatea cea mai scăzută atâta timp cât rutinele de serviciu ale celorlalte întreruperi nu revalidează INTR prin poziționarea lui I;

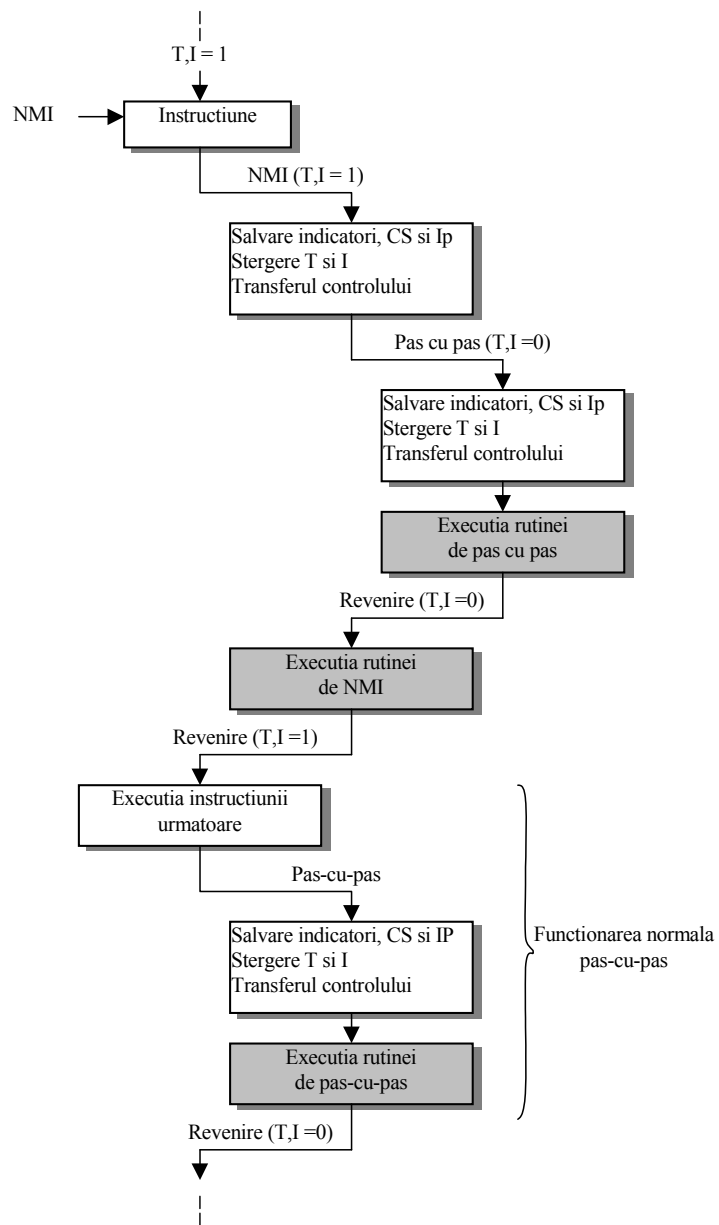


Figura 1.48. O întrerupere NMI pe timpul funcționării pas-cu-pas și funcționarea normală pas-cu-pas

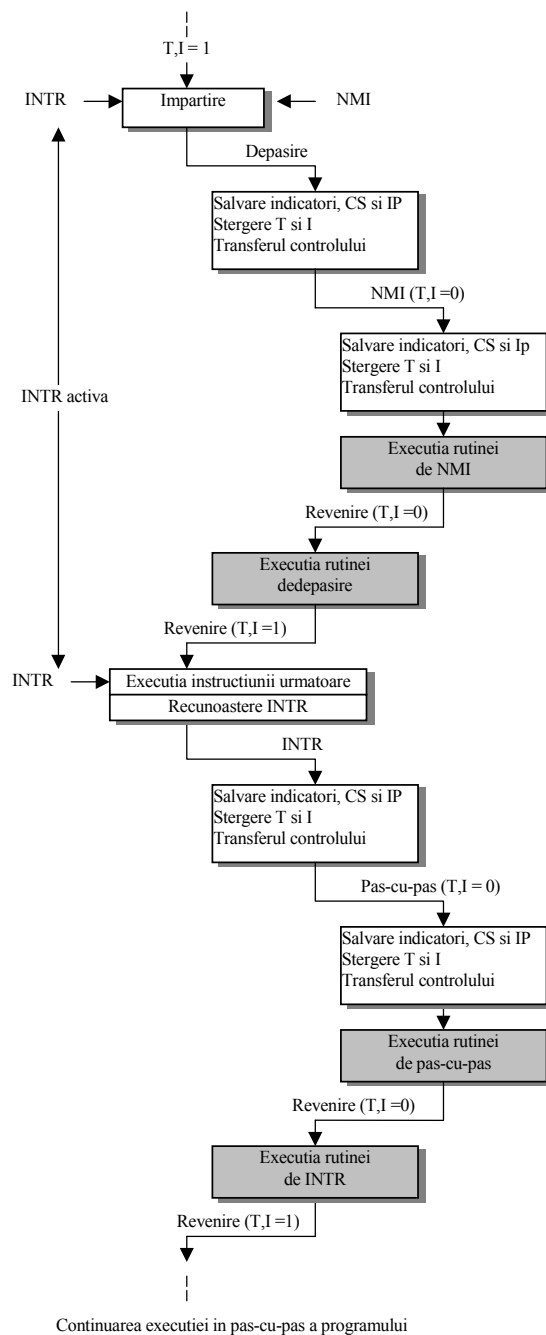


Figura 1.49. Apariția simultană a întreruperilor NMI, INTR, pas-cu-pas și întreruperii de depășire

– în cadrul întreruperilor nemascabile, NMI, pas-cu-pas și întreruperi software, întreruperea pas-cu-pas are prioritatea cea mai înaltă urmată de NMI și întreruperile software. Această ordine de prioritate este valabilă numai în relațiile dintre două tipuri de întreruperi nemascabile, ca în exemplul din figura 1.48. Excepția apare în cazul în care toate cele trei întreruperi nemascabile sunt în așteptare. În această situație ordinea de execuție a acestor rutine este următoarea: NMI, întreruperea software, întreruperea pas-cu-pas. Așadar funcționarea pas-cu-pas se reia la execuția instrucțiunii următoare instrucțiunii care a provocat întreruperea software, figura 1.49.

În legătură cu cererea DMA precizăm că apariția simultană a unei cereri DMA și a întreruperii INTR, de exemplu în timpul execuției unei instrucțiuni precedate de prefixul LOCK, se rezolvă prin servirea întâi a cererii DMA și apoi a întreruperii, HOLD fiind deci prioritar față de INTR.

### 1.3.6. STAREA HALT, OPERAȚII CU $\overline{\text{LOCK}}$ , SINCRONIZAREA EXTERNĂ CU AJUTORUL INTRĂRII $\overline{\text{TEST}}$

#### 1.3.6.1. Starea HALT a microprocesorului

La execuția unei instrucțiuni HALT, microprocesorul 8086 va indica intrarea în starea HALT, oprire, în funcție de modul de lucru. În modul de lucru minim, 8086 va genera semnalul de eșantionare ALE fără nici-o comandă ( $\overline{\text{RD}}$ ,  $\overline{\text{WR}}$  sau  $\overline{\text{INTA}}$ ), figura 1.12. În modul maxim, procesorul generează starea HALT la ieșirile  $\overline{\text{S2}}$ ,  $\overline{\text{S1}}$  și  $\overline{\text{S0}}$ , §1.3.1.3 și figura 1.24, ALE fiind emis de controlorul de magistrală 8288. Starea HALT nu este părăsită decât la apariția unei întreruperi sau la inițializare. Microprocesorul nu va ieși din HALT la apariția unei cereri HOLD de preluare a magistralei. În ultima situație, 8086 va regenera starea HALT.

#### 1.3.6.2. Operații de citire/modificare/scriere cu $\overline{\text{LOCK}}$

Activarea semnalului  $\overline{\text{LOCK}}$  se poate face cu ajutorul prefixului LOCK atunci când este necesară execuția consecutivă a două cicluri de magistrală. Această necesitate poate apărea la execuția unei instrucțiuni de interschimbare, XCHG, memorie/registru când procesorul trebuie să realizeze cu memoria o operație de citire/modificare/scriere, de exemplu a unui semafor, fără a fi perturbat. Semnalul  $\overline{\text{LOCK}}$  este pus pe "0" în ciclul de ceas următor celui în



care UE decodifică codul-obiect al prefixului LOCK și este dezactivat, pus pe "1", la sfârșitul ultimului ciclu de magistrală aparținând instrucțiunii precedate de prefixul LOCK. Pe timpul cât  $\overline{LOCK}$  este activ, în modul maxim, cererile de preluare a magistralei de la intrările RQ/GT sunt memorate pentru a fi achitate după dezactivarea lui  $\overline{LOCK}$ .

### 1.3.6.3. Sincronizarea externă cu ajutorul intrării $\overline{TEST}$

Pentru legătura cu exteriorul, microprocesorul 8086 mai poate testa prin program starea intrării  $\overline{TEST}$  cu ajutorul instrucțiunii WAIT. Dacă instrucțiunea WAIT găsește intrarea  $\overline{TEST}$  inactivă, pe "1", procesorul intră în așteptare executând în mod repetitiv instrucțiunea WAIT. Pe durata execuției acestei instrucțiuni, microprocesorul e în starea pasivă și nu execută cicluri de magistrală. Ieșirea din starea de așteptare se face prin activarea, cel puțin 5 perioade de ceas, a intrării  $\overline{TEST}$ . Dacă în timpul așteptării, execuției instrucțiunii WAIT, apare o cerere HOLD toate ieșirile microprocesorului trec în starea a treia. De asemenea, apariția unei cereri de întrerupere conduce la suspendarea așteptării, procesarea întreruperii și revenirea la execuția instrucțiunii de așteptare WAIT. Pentru a reveni la WAIT procesorul va mai extrage o dată după apariția întreruperii codul instrucțiunii WAIT pentru a putea memora în stivă adresa de revenire din rutina de serviciu a întreruperii.

## 1.4. FUNCȚIONAREA MICROPROCESORULUI 8088

Unitatea centrală 8088 are aceeași structură internă ca 8086 majoritatea funcțiilor fiind identice, figurile 1.2 și 1.3. 8088 manevrează magistralele de adrese, date și comenzi în același mod cu 8086 cu deosebirea că schimbul de date se face pe 8 biți. Operanzii de 16 biți vor fi deci accesați în două cicluri de magistrală consecutivi. Cele două unități centrale sunt identice din punct de vedere al programatorului, diferențele apărând numai la timpii de execuție. De asemenea, structura registrelor este aceeași.

Prezentăm sintetic diferențele dintre microprocesoarele 8088 și 8086:

– lungimea cozii la 8088 este de 4 octeți față de 6 cât este la 8086. Scurtarea s-a făcut cu scopul de a preveni ocuparea excesivă a magistralei de către UIM din 8088 datorită timpului suplimentar impus de accesul pe 8 biți;

– în același timp cu micșorarea cozii, pentru optimizare, proiectanții au modificat și algoritmul de pre-extragere. UIM din 8088 va extrage o nouă





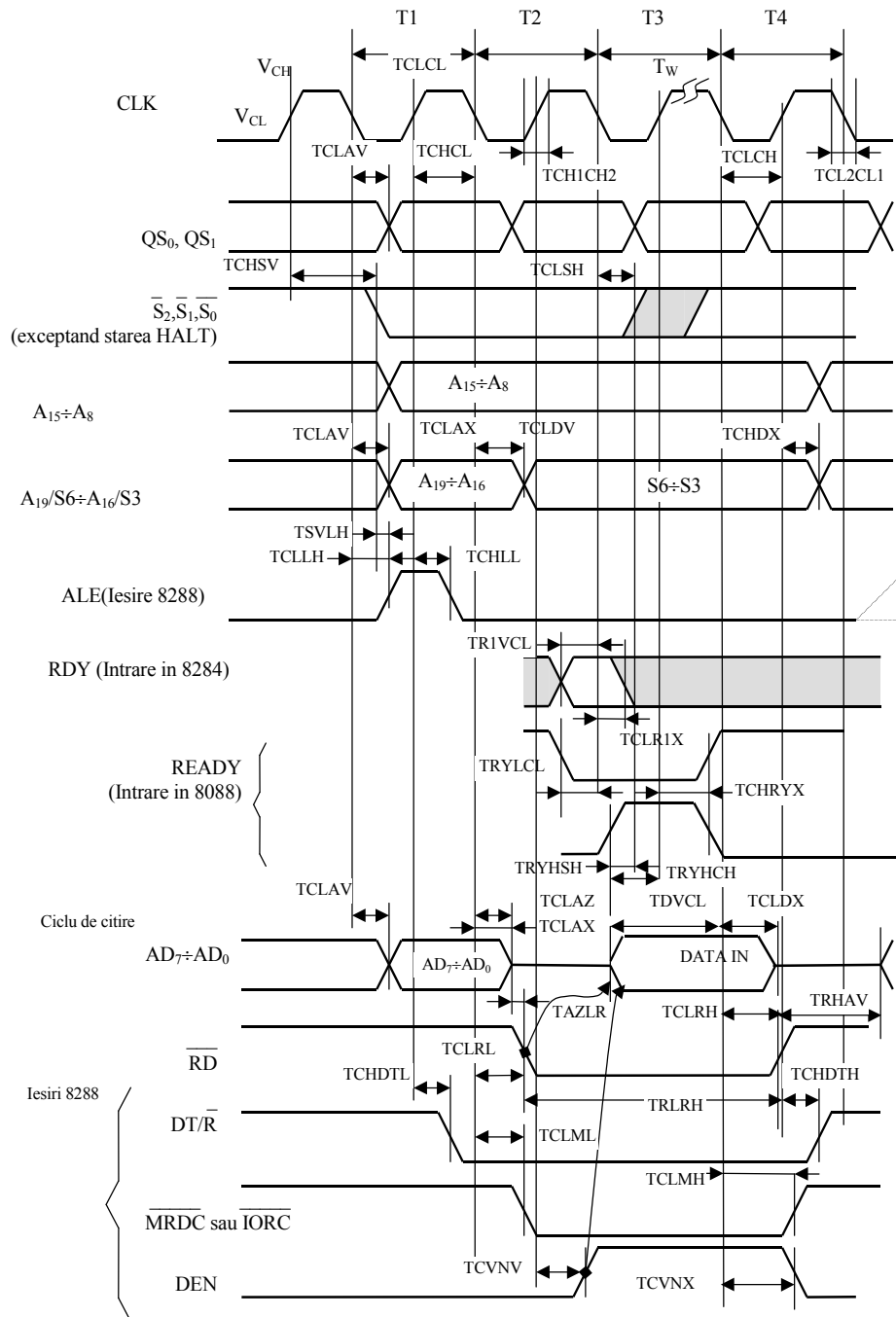


Figura 1.51. Diagrame de timp pentru 8088 în modul maxim

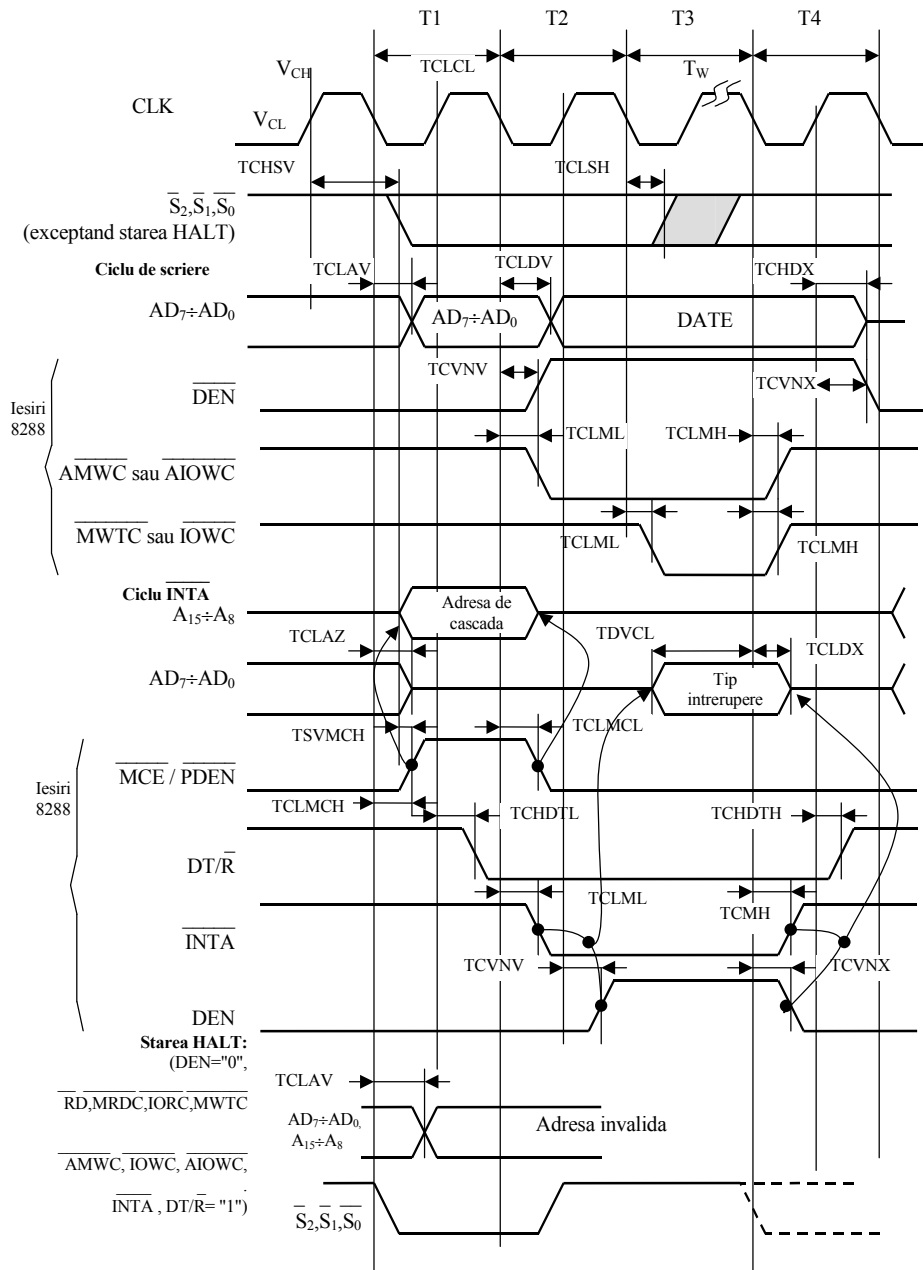


Figura 1.51. Continuare

– timpii de execuție ai instrucțiunilor sunt afectați de accesul pe 8 biți la resursele externe, memoria și dispozitivele de I/O. Toate scrierile și citirile operanzilor de 16 biți se prelungesc cu încă un ciclu de magistrală. Viteza de execuție este, de asemenea, micșorată și de întârzierile apărute la extragerea codului. Această a doua micșorare a vitezei de execuție apare numai în cazul succesiunilor de operații simple. În situațiile utilizării unor instrucțiuni mai sofisticate UIM din 8088 are timp să umple stiva și execuția nu va mai fi limitată decât de viteza UE din microprocesor.

Cum am mai spus 8088 și 8086 sunt complet compatibile software deoarece au unitățile de execuție identice. De aceea programele care nu sunt dependente de organizarea sistemului vor funcționa corect pe amândouă microprocesoarele. Pe de altă parte, programele care depind de organizarea sistemului vor trebui revăzute pentru a putea fi transferate între 8088 și 8086.

Conexiunile externe ale microprocesorului 8088 sunt aproape identice cu cele ale lui 8086, figura 1.7 și 1.8, apărând la 8088 următoarele modificări funcționale:

– biții de adresă A8÷A15 sunt la 8088 numai ieșiri fiind *latch*-ate intern și rămânând deci valizi pe durata întregului ciclu de magistrală, asemănător cu 8085;

– semnalul  $\overline{BHE}$  a fost eliminat, nemaifiind necesar la 8088;

– ieșirea  $\overline{SS0}$  asigură în modul minim informații de stare având semnificațiile bitului de stare  $\overline{S0}$ , vezi §1.3.2.2. Această ieșire este activă numai în modul minim fiind "1" în modul maxim. Împreună cu  $IO/\overline{M}$  și  $DT/\overline{R}$  codifică în modul de lucru minim starea ciclului de magistrală în curs;

– ieșirea  $IO/\overline{M}$  a fost inversată pentru a se obține o compatibilitate la nivel de magistrală cu microprocesorul 8085;

– la o instrucțiune HALT în modul minim, semnalul ALE este întârziat cu o perioadă de ceas pentru a se putea *latch*-a cu el starea microprocesorului.

Diagramele de timp pentru procesorul 8088 lucrând în modul minim sunt date în figura 1.50 iar pentru modul maxim în figura 1.51. Parametrii lui 8088 atât în modul minim cât și în cel maxim, cerințe și răspunsuri în timp, sunt aceși ca ai lui 8086, tabelele 1.2, 1.3 și 1.7, 1.8.

## SETUL DE INSTRUCȚIUNI AL MICROPROCESOARELOR 8086/8088

### 2.1. GENERALITĂȚI. ARHITECTURA UNUI SET DE INSTRUCȚIUNI

Arhitectura unui de instrucțiuni este, așa cum se cunoaște, *partea vizibilă* a mașinii, accesibilă programatorului. Seturile de instrucțiuni se pot clasifica pe baza a cinci direcții de structurare [15]:

- memorarea operanzilor în UC;
- numărul de operanzi explicit specificați într-o instrucțiune;
- localizarea operanzilor;
- tipurile de operații;
- tipurile și dimensiunile operanzilor.

Memorarea operanzilor în UC diferențiază arhitecturile în trei mari clase: arhitecturi de tip *stivă*, de tip *acumulator* și de tip *registre generale* (*GPR* – *General Purpose Registers*). Mașinile mai vechi reprezintă arhitecturi de tip *stivă* sau *acumulator* în timp ce mașinile actuale au la bază arhitecturi cu registre generale. Această evoluție are două motivații principale: întâi că registrele, ca și alte structuri de memorare din cadrul UC, sunt mai rapide decât memoria, și apoi, ele sunt mai ușor și mai eficient de utilizat la compilare în comparație cu celelalte forme de memorare internă.

Arhitecturile GPR se pot diferenția la rândul lor cu ajutorul a două caracteristici mari. Prima se referă la *numărul de operanzi ai unei instrucțiuni UAL*: doi sau trei. În formatul cu trei operanzi instrucțiunea conține doi operanzi sursă și un operand rezultat. În formatul cu doi operanzi unul dintre aceștia este atât sursă cât și destinație. A doua diferențiere se referă la *numărul de operanzi dintr-o instrucțiune UAL ce pot fi adrese de memorie*. Acest număr poate varia pentru o instrucțiune UAL tipică între 0 și 3. Combinațiile celor două caracteristici sunt exemplificate în tabelul 2.1.

Tabelul 2. 1. Arhitecturi de tip "registru general" – GPR

Număr de adrese de memorie în instrucțiuni UAL	Număr maxim de operanzi în instrucțiuni UAL	Exemple de mașini
0	2	IBM RT-PC
	3	SPARC, MIPS, HP Precision
1	2	PDP-10, 68000, IBM 360, 8086
	3	IBM 360 (instrucțiunile RS)
2	2	PDP-11, 32X32, IBM 360 (instr. SS)
	3	–
3	3	VAX (are și formate de doi operanzi)

Deși, așa cum se poate observa, sunt șapte combinații posibile, numai trei sunt considerate tipice și permit clasificarea majorității mașinilor de calcul actuale. Acestea sunt denumite *registru-registru* (sau *încărcare/memorare – load/store*), *registru-memorie* și *memorie-memorie*. Fiecare din cele trei tipuri de arhitecturi are avantajele și dezavantajele sale. Astfel, arhitecturile *registru-registru* sunt în general simple, au instrucțiuni de lungime fixă care se execută de obicei în același număr de ceasuri. Dezavantajul lor constă în aceea că programele vor avea un număr de instrucțiuni mai mare. Arhitecturile *registru-memorie* prezintă avantajul accesării directe a datelor din registre, fără încărcări prealabile din memorie. Dezavantajul lor principal apare datorită neechivalenței operanzilor, cel care este sursă și destinație fiind distrus în urma unei operații binare. Mașinile *memorie-memorie* sunt cele mai compacte și pot să nu mai folosească registre pentru memorări temporare. Pe de altă parte aceste arhitecturi au instrucțiuni de lungimi și complexități variabile, care se pot executa în perioade de timp foarte diferite. Eficiența acestor ultime tipuri de mașini poate fi sever limitată de accesul la memorie, *gâtuitura von Neumann*. Avantajele și dezavantajele arhitecturale de mai sus sunt calitative, nu au un caracter absolut, și, în ultimă instanță, impactul lor real depinde atât de calitatea compilatoarelor cât și de realizarea hardware a mașinii.

8086, este, așa cum am spus deja în §1.1, o arhitectură hibridă acumulator/registre generale, cu instrucțiuni UAL tipice cu doi operanzi dintre care unul poate fi adresă de memorie. Setul de instrucțiuni prevede operații atât pe 8 biți, octeți, cât și pe 16 biți, cuvinte. Această distincție referitoare la tipul operandului apare atât în operațiile cu registre cât și în accesele la memorie.

Așa cum am văzut în primul capitol, spațiul de adrese al microprocesorului 8086, deși este de 20 de biți, este "spart" în segmente de 64kB adresabile cu ajutorul unor deplasamente de 16 biți. Adresa de 20 de biți este astfel formată dintr-o adresă efectivă de 16 biți – *offset*-ul sau



deplasamentul în cadrul unui segment –, care se adaugă la o altă adresă pe 16 biți – baza segmentului. Adresa bazei segmentului se obține prin deplasarea stânga cu 4 poziții a conținutului unui registru de segment de 16 biți.

Cele 14 registre ale microprocesorului 8086, prezentate pe larg în prima parte a lucrării, figura 1.4, pot fi împărțite din punctul de vedere al arhitecturii setului de instrucțiuni în patru clase: registre de date (AX, BX, CX și DX), registre de adresare (SP, BP, SI, DI), registre de segment (CS, SS, DS, ES) și registre de control (IP, Starea). Registrele de date sunt utilizate pentru memorarea datelor și operații asupra lor, registrele de adresare sunt folosite pentru formarea adreselor de memorie efective de 16 biți (în cadrul segmentelor), registrele de segment pentru formarea unei adrese de memorie reale pe 20 de biți iar registrele de control pentru păstrarea stării mașinii și controlul programului.