

Proiectare cu Microprocesoare

Curs 12 – DRAM, DMA

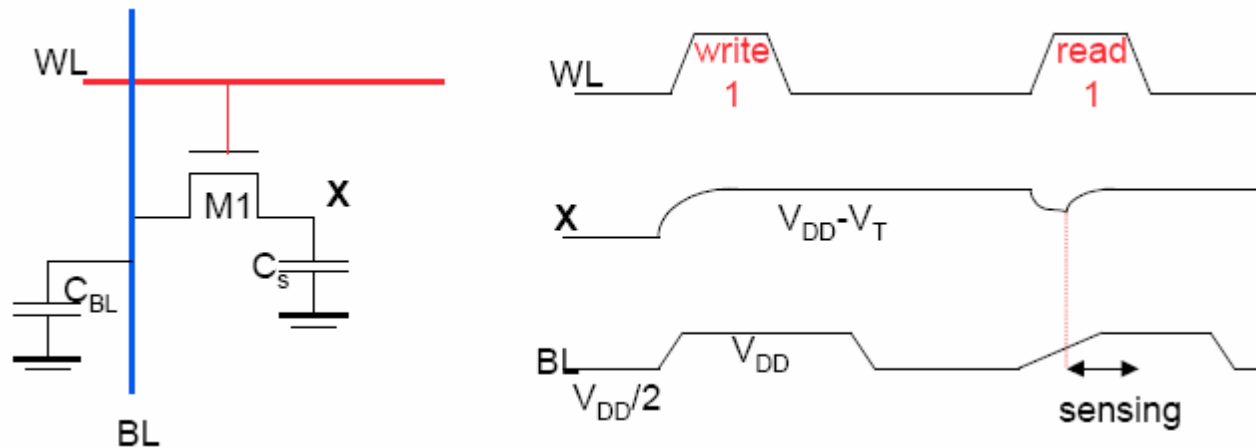
An 3 CTI

An universitar 2018/2019

Semestrul 1

Lector: Radu Danescu

Celula DRAM



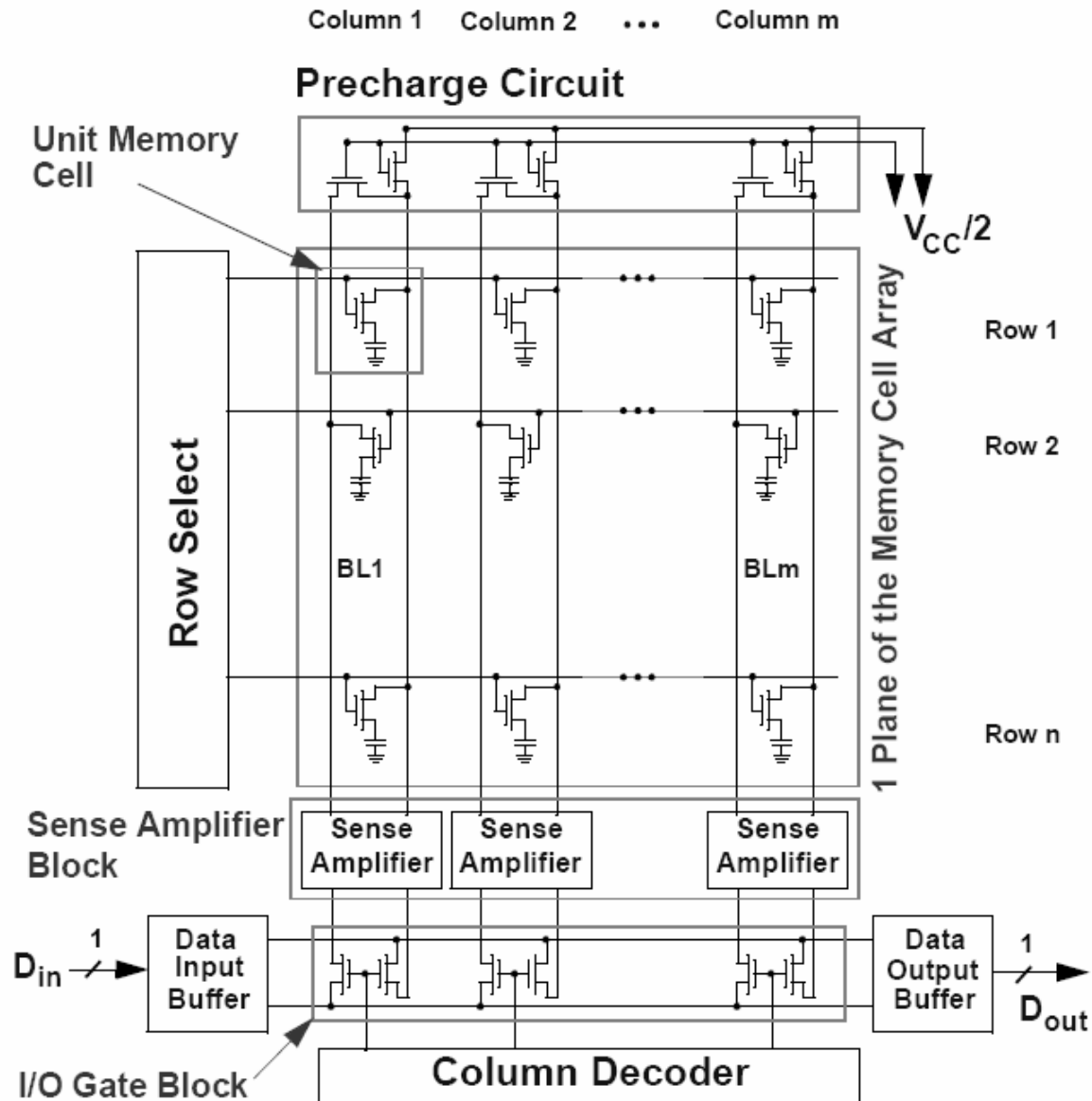
X- tensiunea pe C_s ; $C_s \sim 25\text{fF}$

Scriere: C_s se încarcă sau se descarcă (WordLine, BitLine activ)

Citire: Redistribuirea sarcinii între BL și C_s

Înainte de citire - precharge: $V_{\text{bit line}} = V_{DD}/2$:

Matricea de memorie DRAM



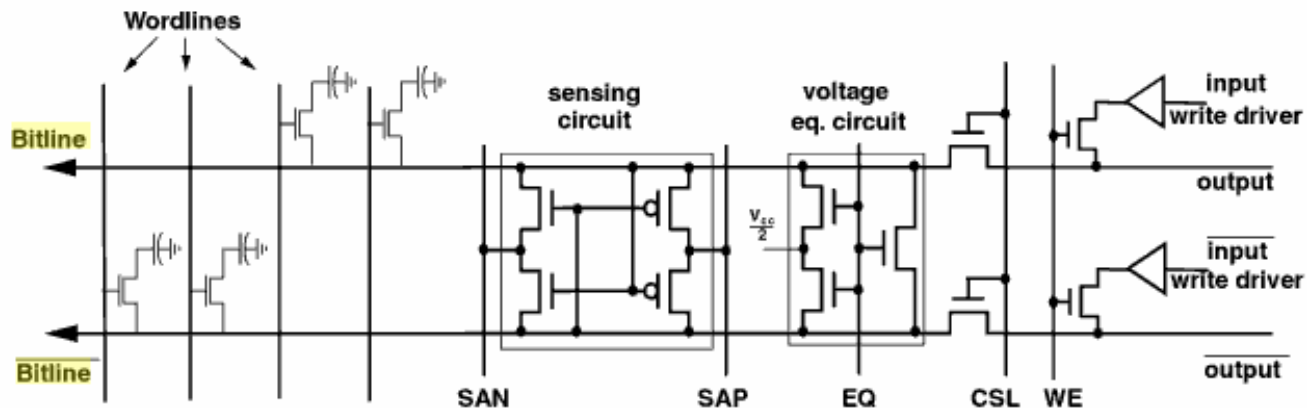
DRAM – Principiul de functionare - CITIRE

1. Se dezactivează amplificatorul de citire, și se pre-încarcă liniile de bit la nivelul de tensiune intermediar între logic '0' și logic '1'.
2. Se dezactivează circuitul de pre-încărcare. Din cauza lungimii liniilor, se va păstra tensiunea de pre-încărcare pentru o scurtă perioadă de timp.
3. Se activează linia de cuvânt pentru rândul selectat. Această acțiune conectează condensatorul de memorare la linia de bit, și sarcina electrică se împarte între linie și condensator, modificând tensiunea de pe linie.
4. Se activează amplificatorul de citire, care funcționează pe principiul reacției pozitive, amplificând orice diferență de tensiune dintre cele doua linii de bit, până când una este saturată la nivelul '1' și cealaltă la nivelul '0'. În acest moment rândul este 'deschis' și se poate selecta coloana.
5. Datele citite din DRAM se preiau din amplificatoarele de citire, selectate de adresa coloanei. Se pot efectua mai multe citiri pentru același rând.
6. În timp ce se fac citirile, curentul circulă înapoi în liniile de bit, de la amplificatorul de citire. Acest curent reface sarcina condensatorului.
7. Când citirea rândului curent este finalizată, linia cuvântului este dezactivată, condensatorul este deconectat, amplificatorul de citire este deconectat, iar liniile de bit sunt pre-încărcate din nou.

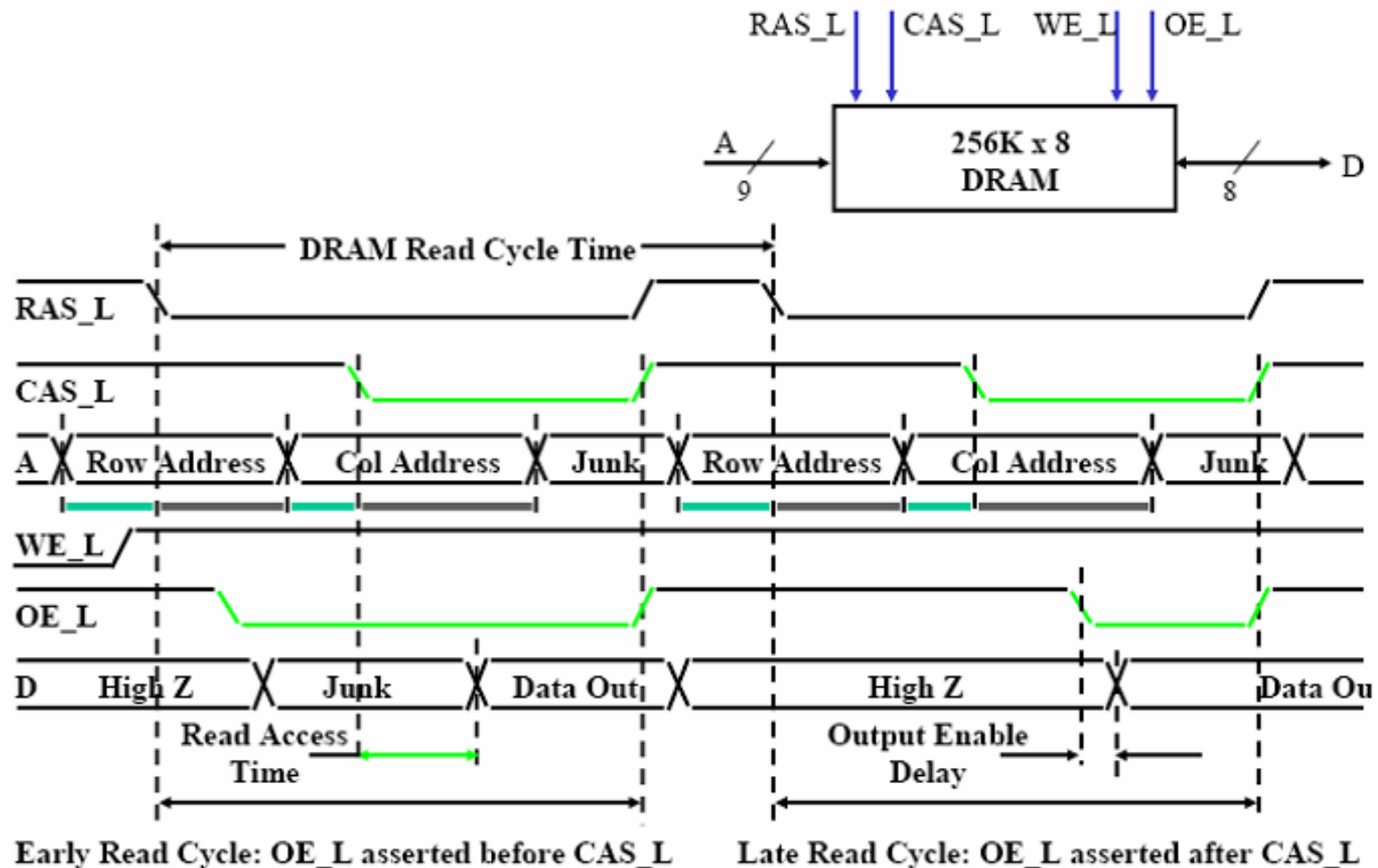


DRAM – Principiul de functionare - Scriere

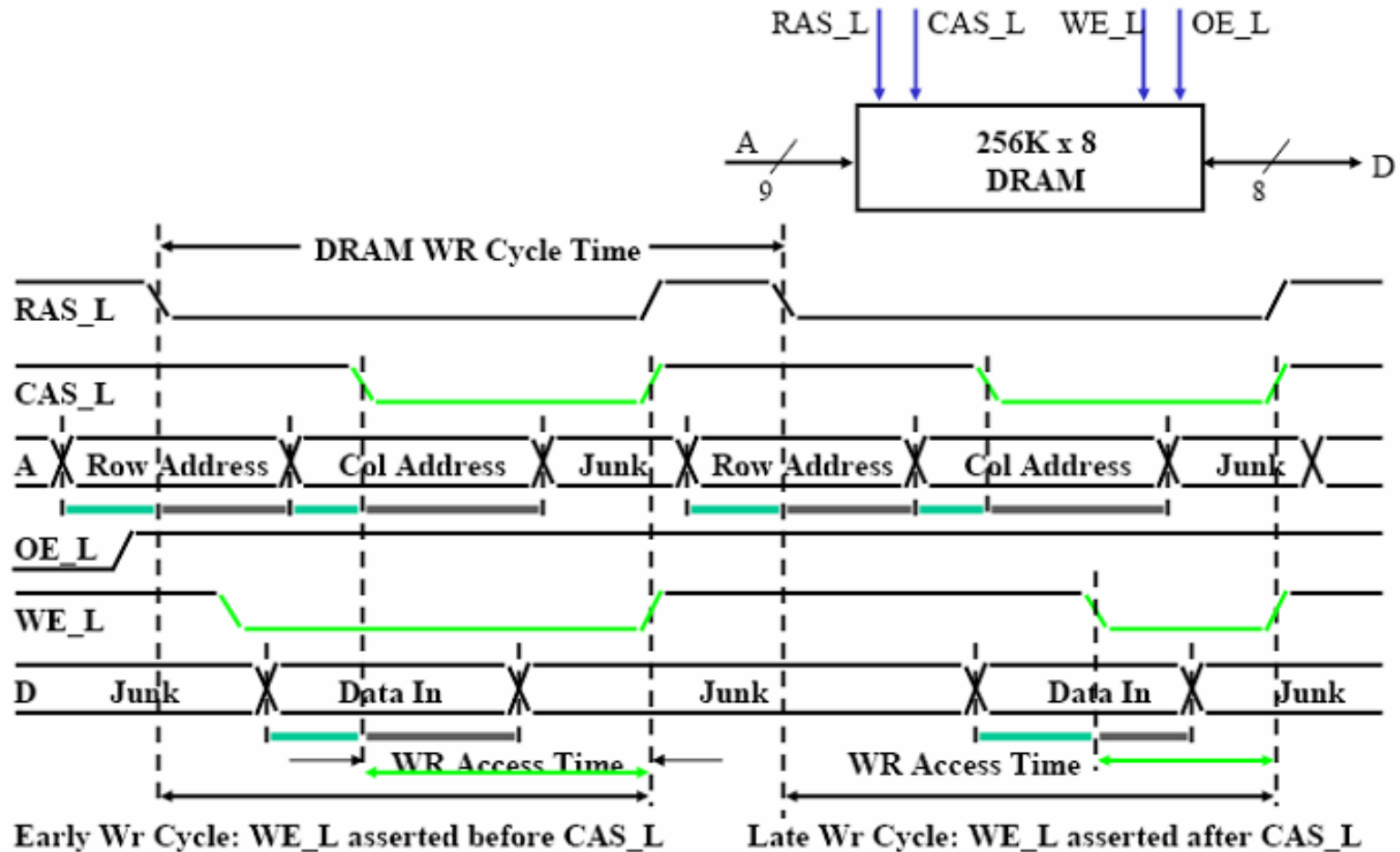
1. Se forțează starea dorită în amplificatorul de citire/scriere.
2. Se deschide linia de cuvânt, și sarcina condensatorului se va încărca sau descărca în funcție de tensiunile de pe liniile de bit.
3. Pentru scrierea unei celule, se citește tot rândul, se modifică locațiile dorite, și întreg rândul se scrie înapoi.



DRAM – Diagrama de timp - Citire

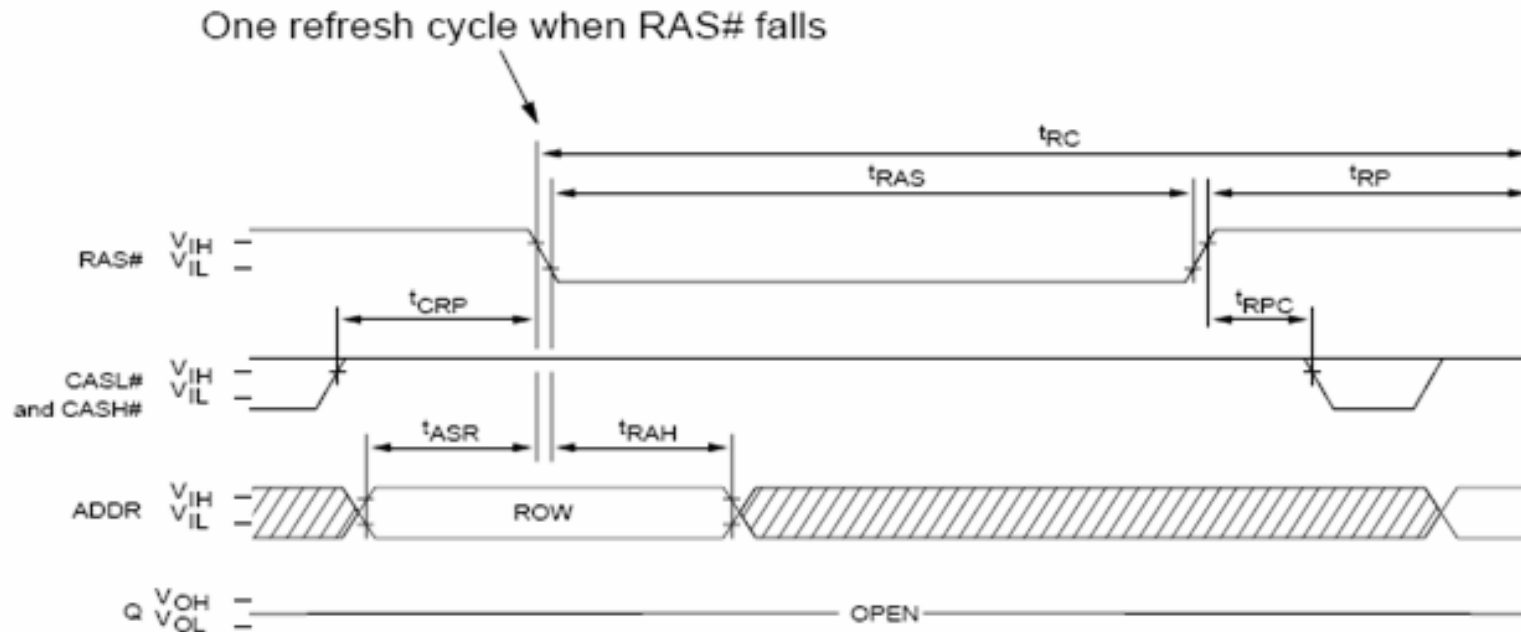


DRAM – Diagrama de timp - Scriere



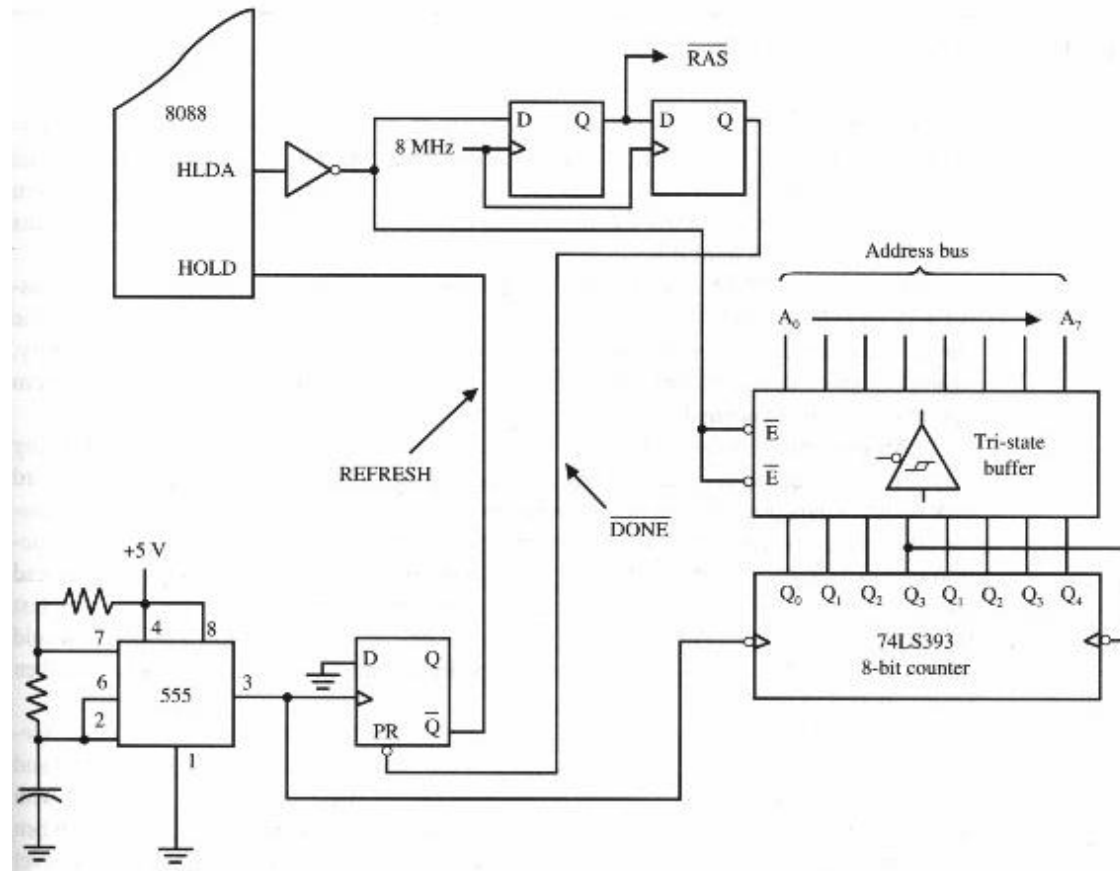
Reîmprospătare (refresh)

- Reîmprospătarea se face la accesarea unui rând
- Nu este necesară selecția coloanei
- Se generează adrese de rând succesive, și se activează RAS



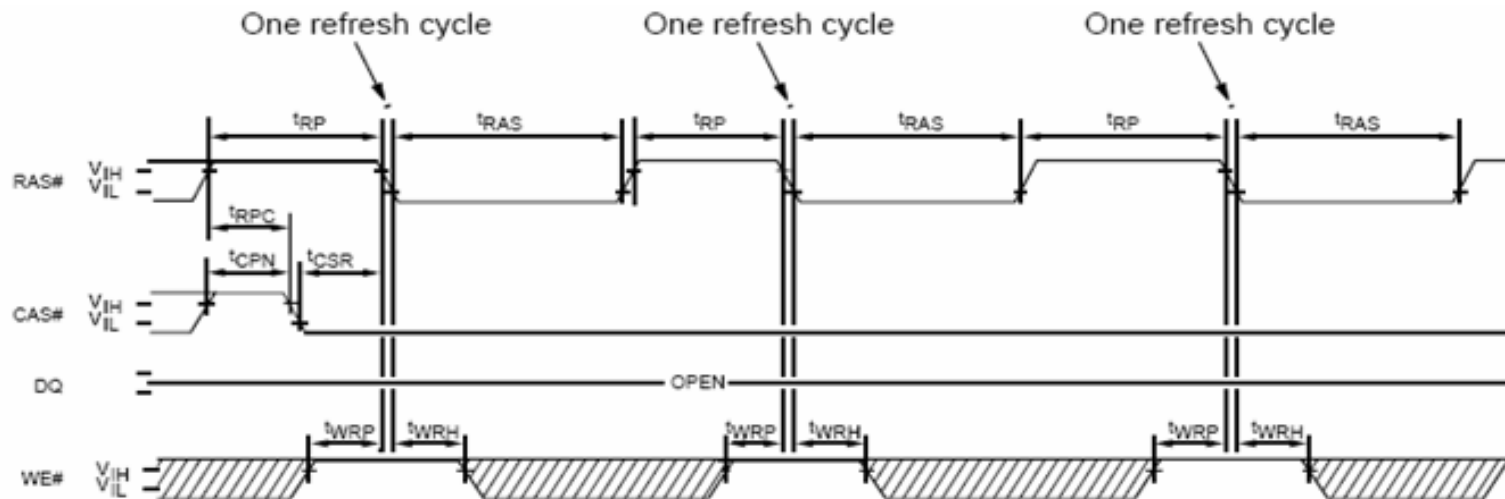
Reîmprospătare (refresh)

- Circuit de refresh extern:
- Se folosește HOLD-HLDA pentru controlul accesului concurent la magistrală



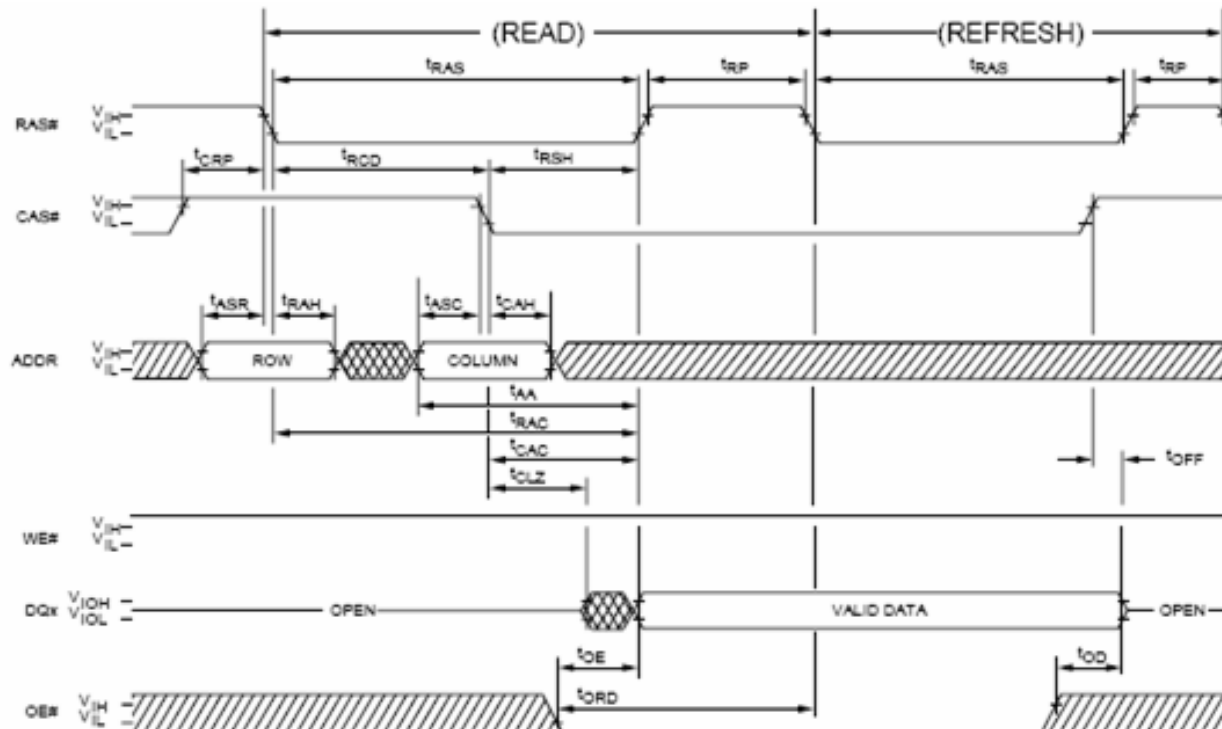
Reîmprospătare (refresh)

- Circuit de refresh intern:
- Circuitul de memorie încorporează un numărător de rânduri
- CAS before RAS – combinație în mod normal nepermisă, indică memoriei să facă o reîmprospătare folosind contorul intern



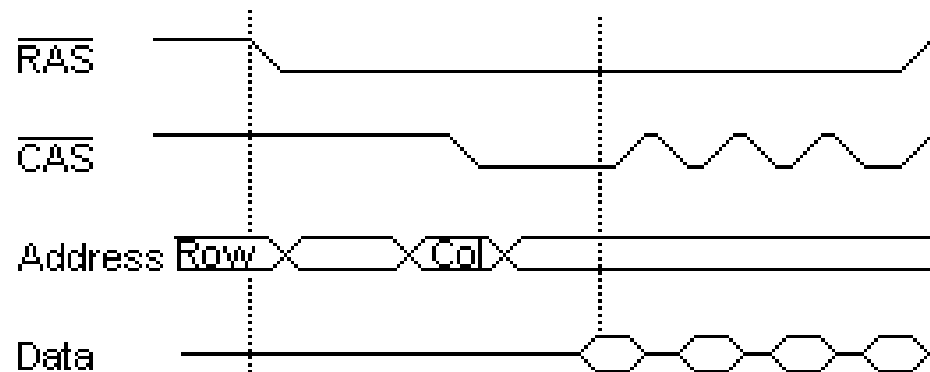
Reîmprospătare (refresh)

- Refresh ascuns ('Hidden refresh'):
- Folosește combinația CAS before RAS
- Urmează unei citiri
- Se dezactivează RAS menținând CAS, datele sunt valide la ieșire
- Se activează RAS, și se produce refresh fără afectarea ieșirilor



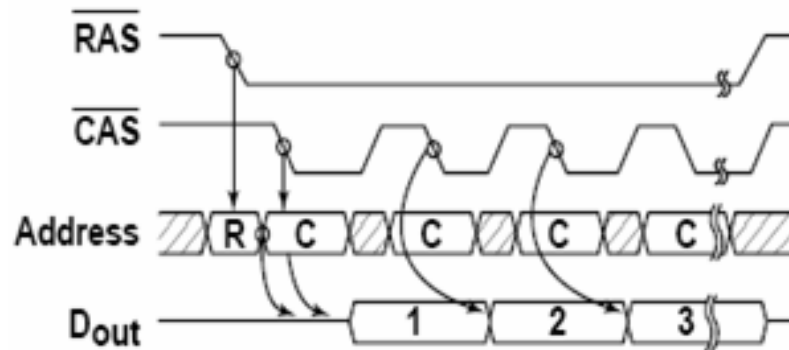
DRAM – Fast Page Mode (FPM)

- Permite citirea mai multor coloane pentru același rând
- 1 RAS, multiple CAS
- Variații:
 - **Static column** – nu necesită pulsarea CAS pentru schimbarea coloanei, ci CAS este menținut activ, adresa coloanei se schimbă și ieșirea o urmează cu o anumită întârziere
 - **Nibble mode** – adresele consecutive primei coloane sunt generate de un contor intern, la fiecare puls CAS, nemaifiind preluate de pe liniile de adresă.

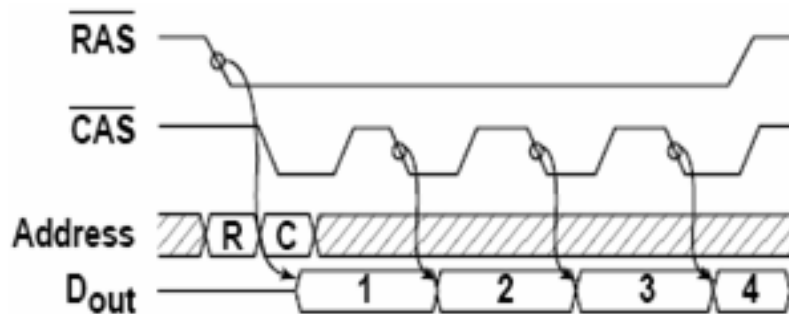


DRAM – Extended Data Out (EDO)

- Un ciclu de acces poate menține datele din ciclul anterior active

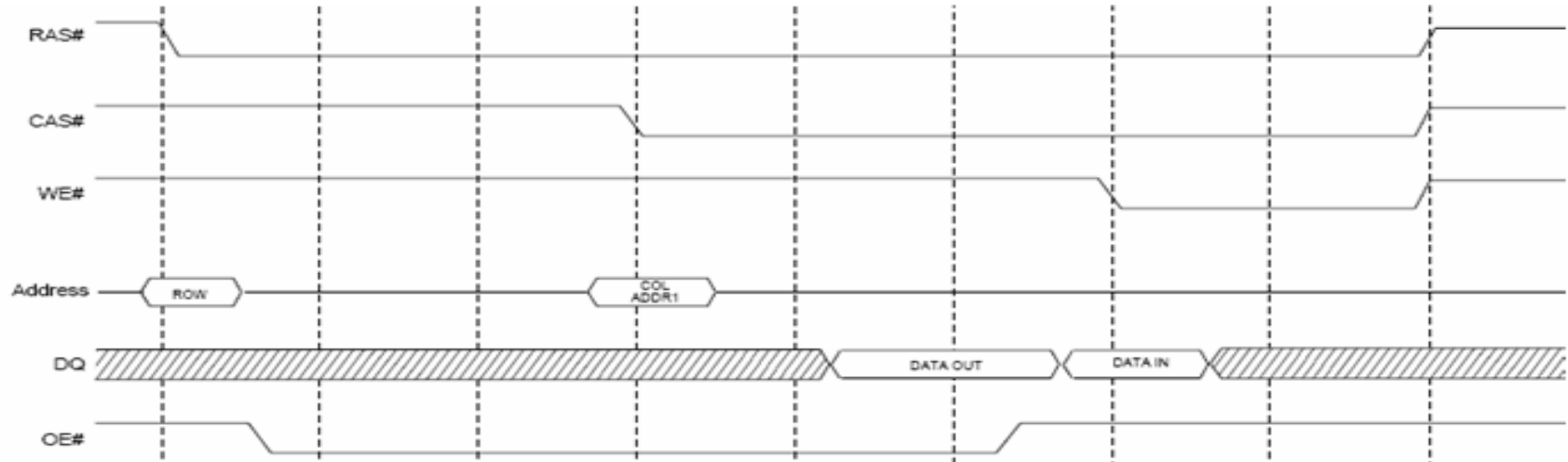


- Burst EDO (BEDO) – adresele coloanei sunt generate intern după primul CAS

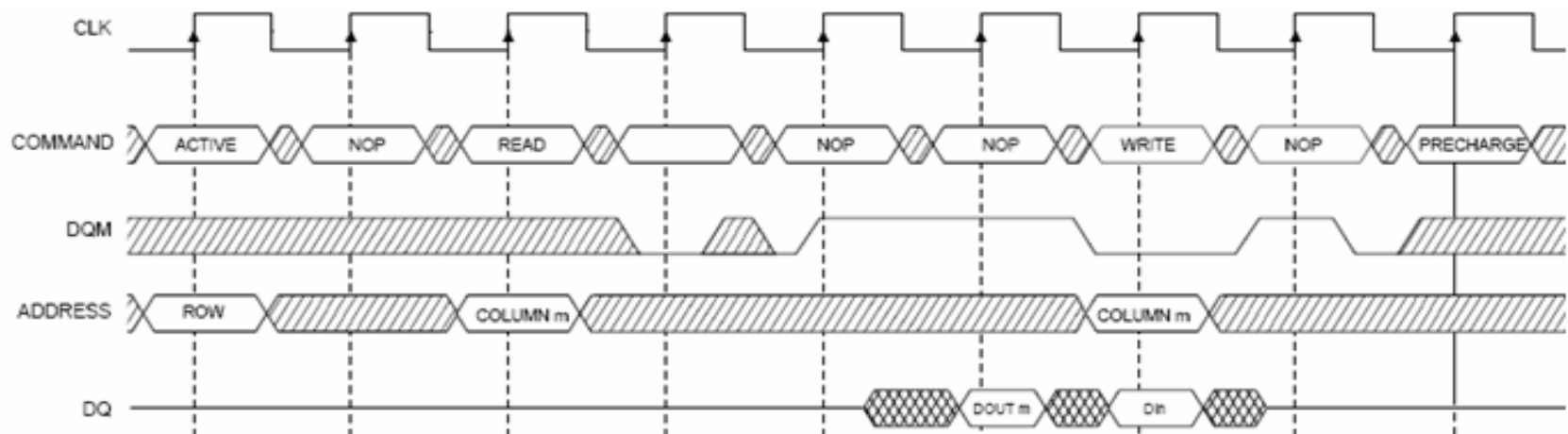


DRAM asincron / sincron

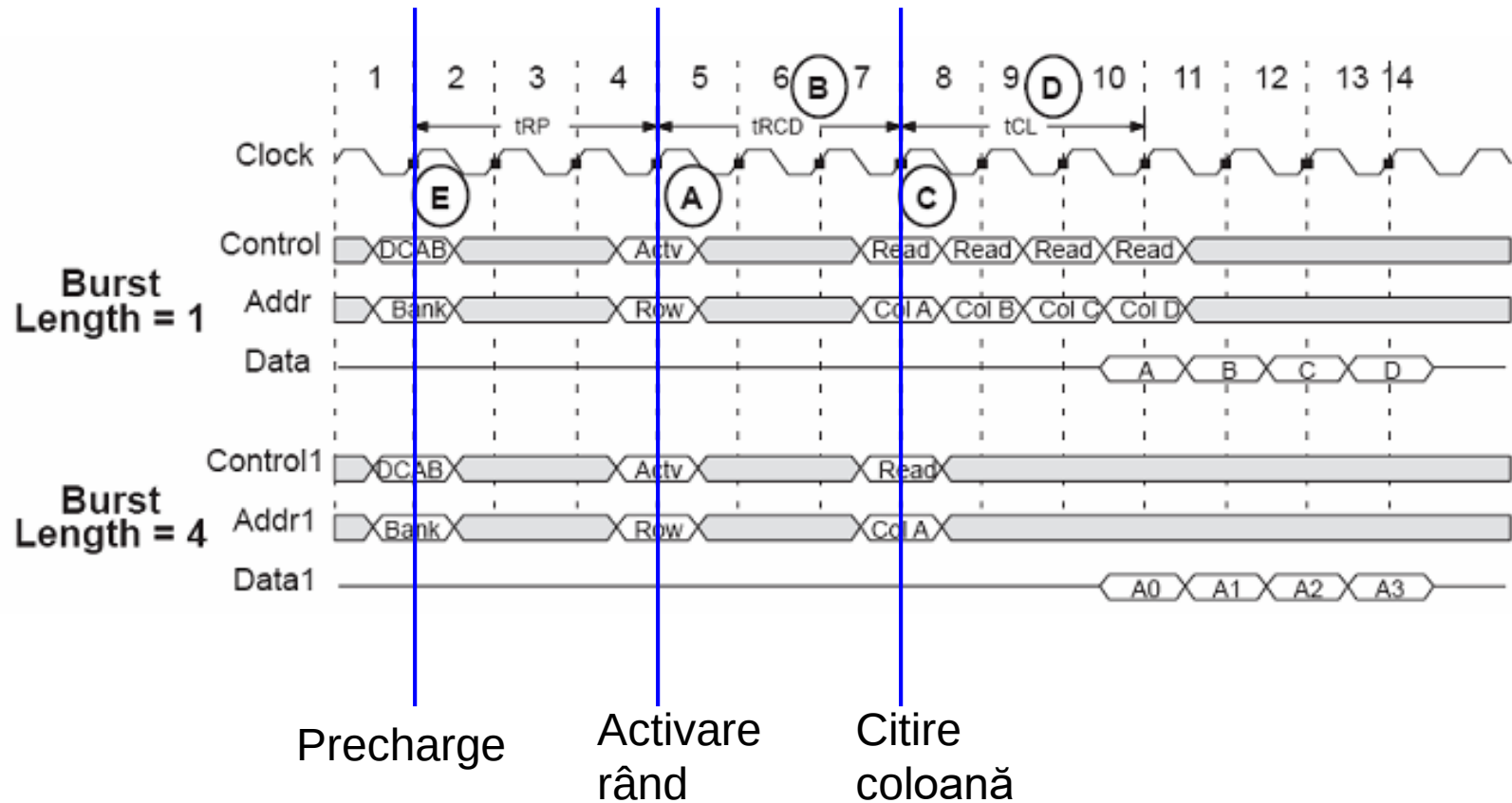
Asincron



Sincron



SDRAM – Synchronous DRAM



SDRAM

Semnale (selecție)

- * **CKE Clock Enable.** Dacă acest semnal este zero, memoria nu execută nici o operație.
- * **/CS Chip Select.** Când acest semnal este 1, memoria ignoră toate intrările.
- * **DQM Data Mask.** Maska pentru octeții din cuvântul de date (1 linie de mască pentru fiecare octet).
- * **/RAS Row Address Strobe.** Formează, împreună cu /CAS și /WE, comenzi pentru memorie.
- * **/CAS Column Address Strobe.**
- * **/WE Write enable.**

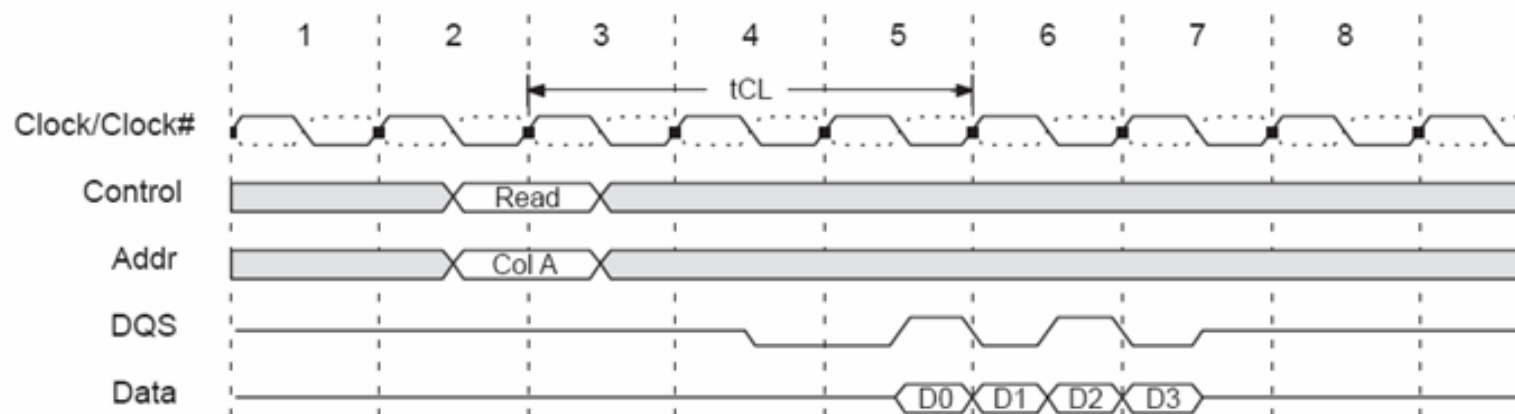
SDRAM

Comenzi

/CS	/RAS	/CAS	/WE	BAn	A10	An	Command
H	x	x	x	x	x	x	Command inhibit (No operation)
L	H	H	H	x	x	x	No operation
L	H	H	L	x	x	x	Burst Terminate: stop a burst read or burst write in progress.
L	H	L	H	bank	L	column	Read: Read a burst of data from the currently active row.
L	H	L	H	bank	H	column	Read with auto precharge: As above, and precharge (close row) when done.
L	H	L	L	bank	L	column	Write: Write a burst of data to the currently active row.
L	H	L	L	bank	H	column	Write with auto precharge: As above, and precharge (close row) when done.
L	L	H	H	bank	row		Active (activate): open a row for Read and Write commands.
L	L	H	L	bank	L	x	Precharge: Deactivate current row of selected bank.
L	L	H	L	x	H	x	Precharge all: Deactivate current row of all banks.
L	L	L	H	x	x	x	Auto refresh: Refresh one row of each bank, using an internal counter. All banks must be precharged.
L	L	L	L	0 0	mode		Load mode register: A0 through A9 are loaded to configure the DRAM chip. The most significant settings are CAS latency (2 or 3 cycles) and burst length (1, 2, 4 or 8 cycles)

DDRAM – Double Data Rate SDRAM

- Transferul de date se efectuează pe ambele fronturi ale semnalului de ceas
- Modulul are componente pe ambele fețe ale plăcuței



- Variații: DDR1 ... DDR4 (până la 64 GB capacitate, până la 4 GTransferuri/ secundă)

Direct memory access (Accesul Direct la Memorie / DMA)

Direct memory access (DMA) este un proces prin care un dispozitiv extern preia controlul magistralei, în locul procesorului.

DMA se folosește la **transferul de mare viteză a datelor** la/de la dispozitive de stocare precum harddisk-uri, CD-ROM, uneori și controllere video.

Ideea fundamentală a DMA este de a transfera blocuri de date **în mod direct între memorie și periferice**. Datele nu mai trec prin procesor, dar magistrala este ocupată.

Un transfer “normal” al unui byte de date poate lua până la 29 de perioade de ceas. Un transfer prin DMA necesită doar 5 perioade.

În calculatoarele moderne, DMA poate transfera sute de Mbytes pe secundă. Rata de transfer este limitată de viteza memoriei și de viteza dispozitivelor periferice.

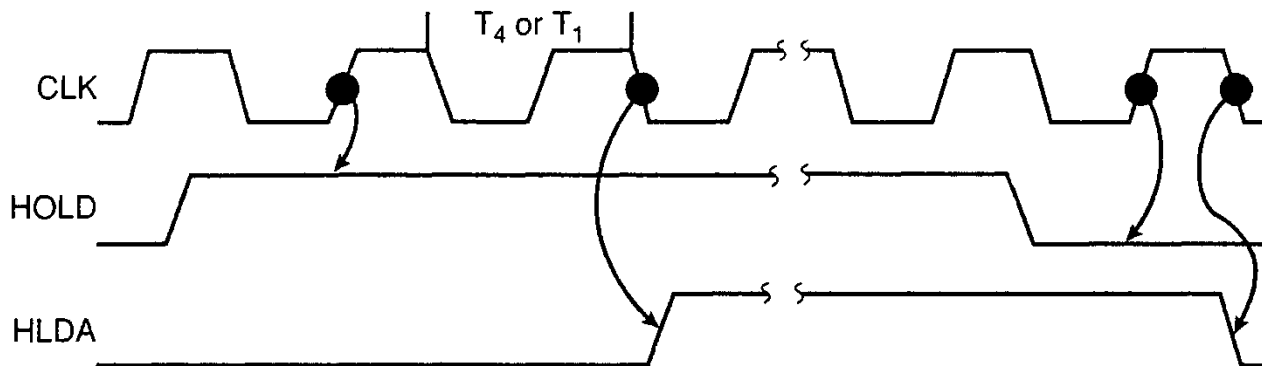
Funcționarea DMA la 8088/8086 în mod minimal:

Pinii **HOLD** și **HLDA** sunt folosiți pentru a primi și a confirma cererea pentru eliberarea magistralei.

În mod normal, CPU are controlul absolut al magistralei. În modul de operare DMA, componentul periferic preia controlul magistralei, temporar.

Pașii unui proces DMA tipic:

- 1) Controllerul DMA anunță cererea, pe pinul HOLD.
- 2) 8086 termină ciclul de magistrala curent, și intră în starea HOLD.
- 3) 8086 cedează controlul magistralei prin activarea pinului HOLDA. Pinii de lucru cu magistrala de la 8086 (Adrese, Date, Control) se pun în înaltă impedanță.
- 4) Începe operația DMA.
- 5) La terminarea operației DMA, controllerul DMA dezactivează HOLD, pentru a elibera controlul magistralei.

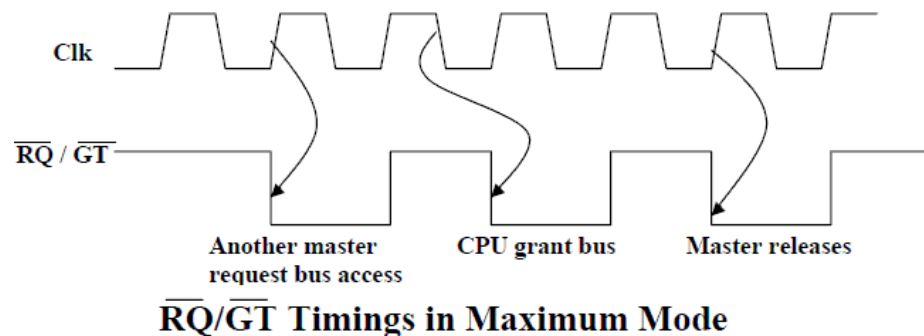


Funcționarea DMA la 8088/8086 în modul maximal:

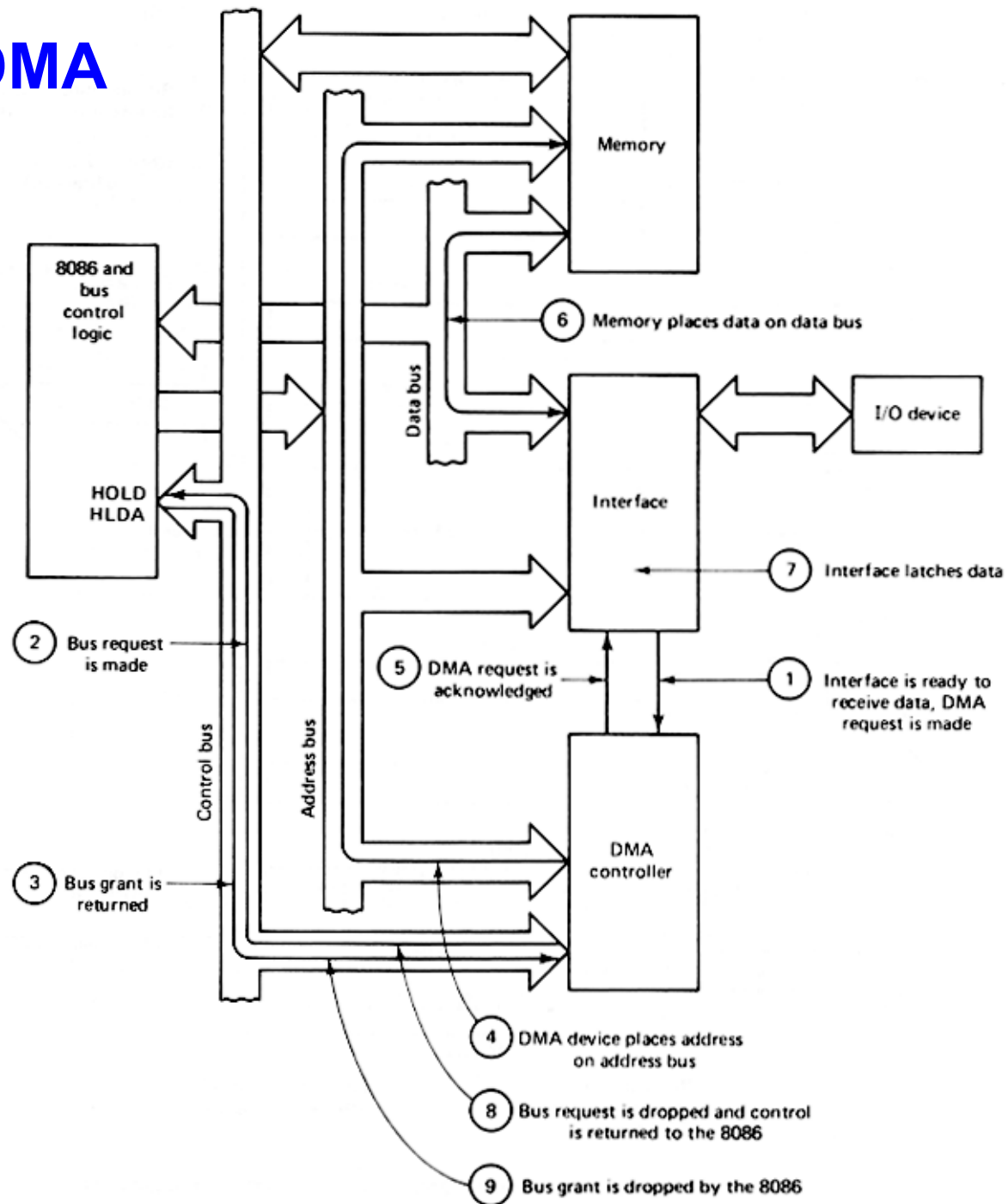
Pinii RQ/GT1 și RQ/GT0 sunt folosiți pentru a emite o cerere DMA și pentru a primi semnalele de confirmare.

Secvența de evenimente pentru un proces DMA tipic:

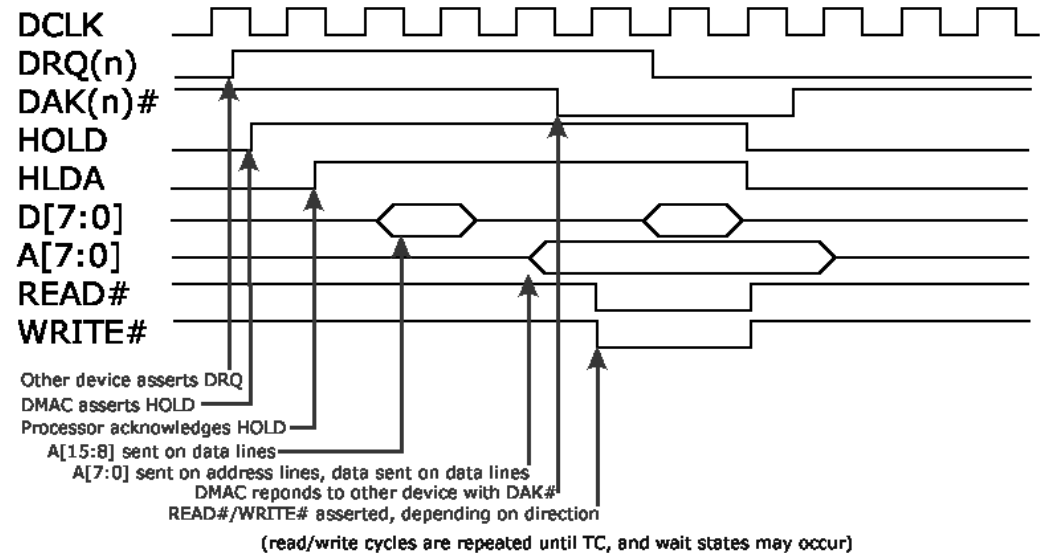
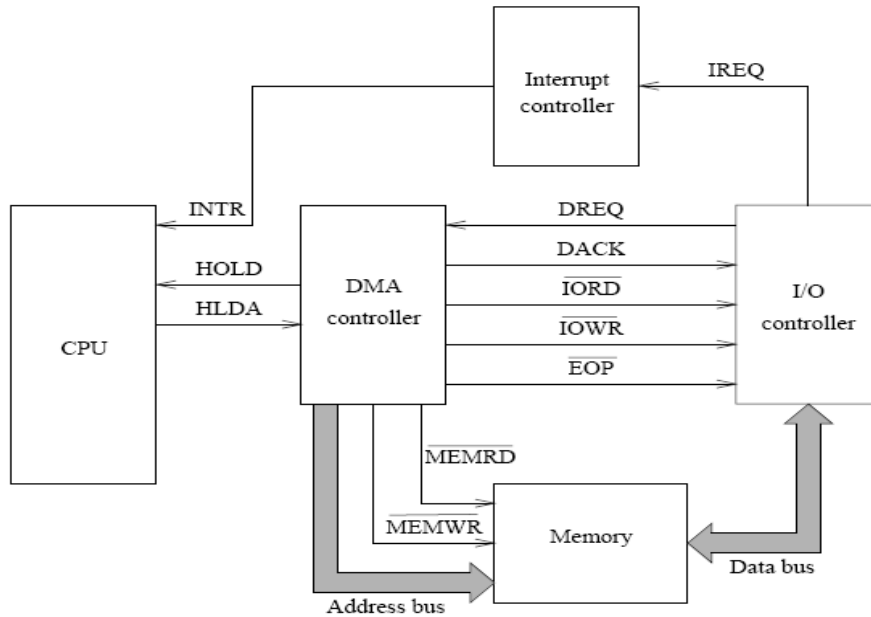
- 1) Controllerul DMA activează unul din pinii de cerere, ex. RQ/GT1 sau RQ/GT0 (RQ/GT0 este prioritar)
- 2) 8086 termină ciclul de magistrală curent, și intră în starea de eliberare a magistralei (stare HOLD)
- 3) 8086 cedează controlul magistralei prin activarea unui semnal pe același pin pe care a fost primit semnalul de cerere
- 4) Începe operația DMA
- 5) La terminarea operației, controllerul DMA activează pinul de cerere pentru a semnala eliberarea magistralei.



Secvența DMA



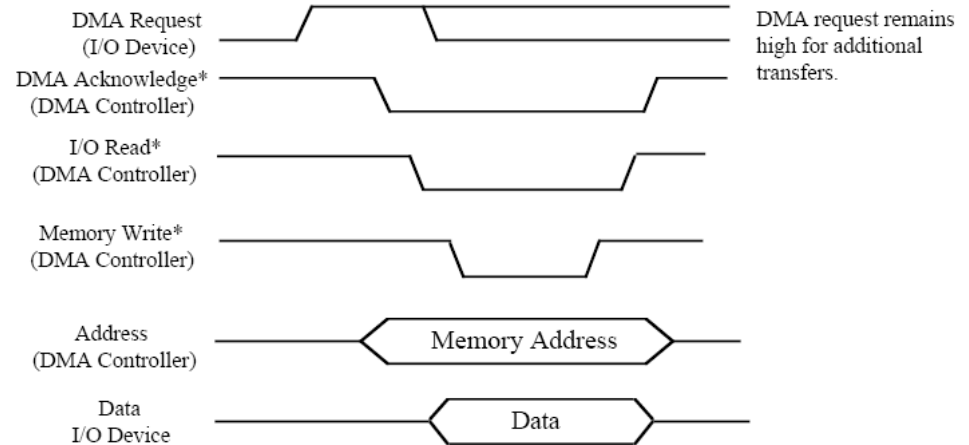
Semnalele pentru transferul DMA



Tipuri de transfer DMA

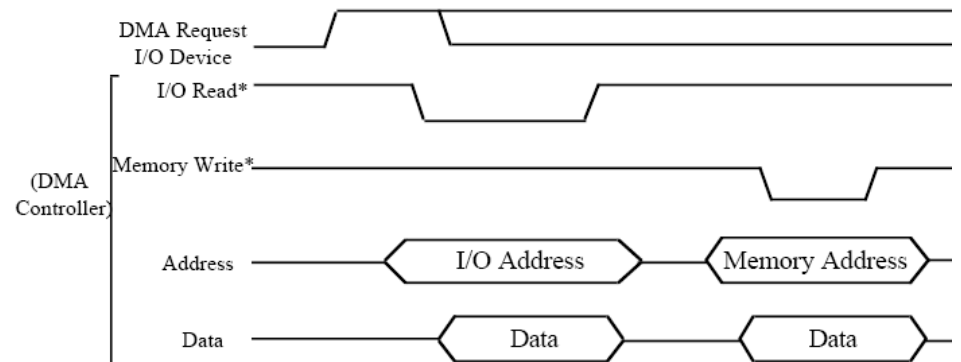
Transfer “Fly-by”

- Datele nu trec prin controllerul DMA
- 1 ciclu de magistrală per transfer
- Mem ↔ I/O
- Semnale de control simultane



Transfer “Flow-through”

- Datele trec prin controller
- Transfer cu preluare și stocare: 2 cicluri/transfer
- Mem – Mem, I/O – I/O



Controller DMA

Un controller DMA se interfațează cu mai multe periferice care pot cere transferuri DMA.

Controllerul decide prioritatea în cazul cererilor simultane, și ofera adresele de memorie pentru transferul datelor.

Controllerul DMA folosit cu 8088/8086 este **Controllerul DMA programabil 8237.**

8237 este un dispozitiv cu 4 canale. Fiecare canal este dedicat unui dispozitiv periferic, și este capabil de a adresa o secțiune de memorie de 64 K Bytes.

Organizarea generală a unui controller DMA

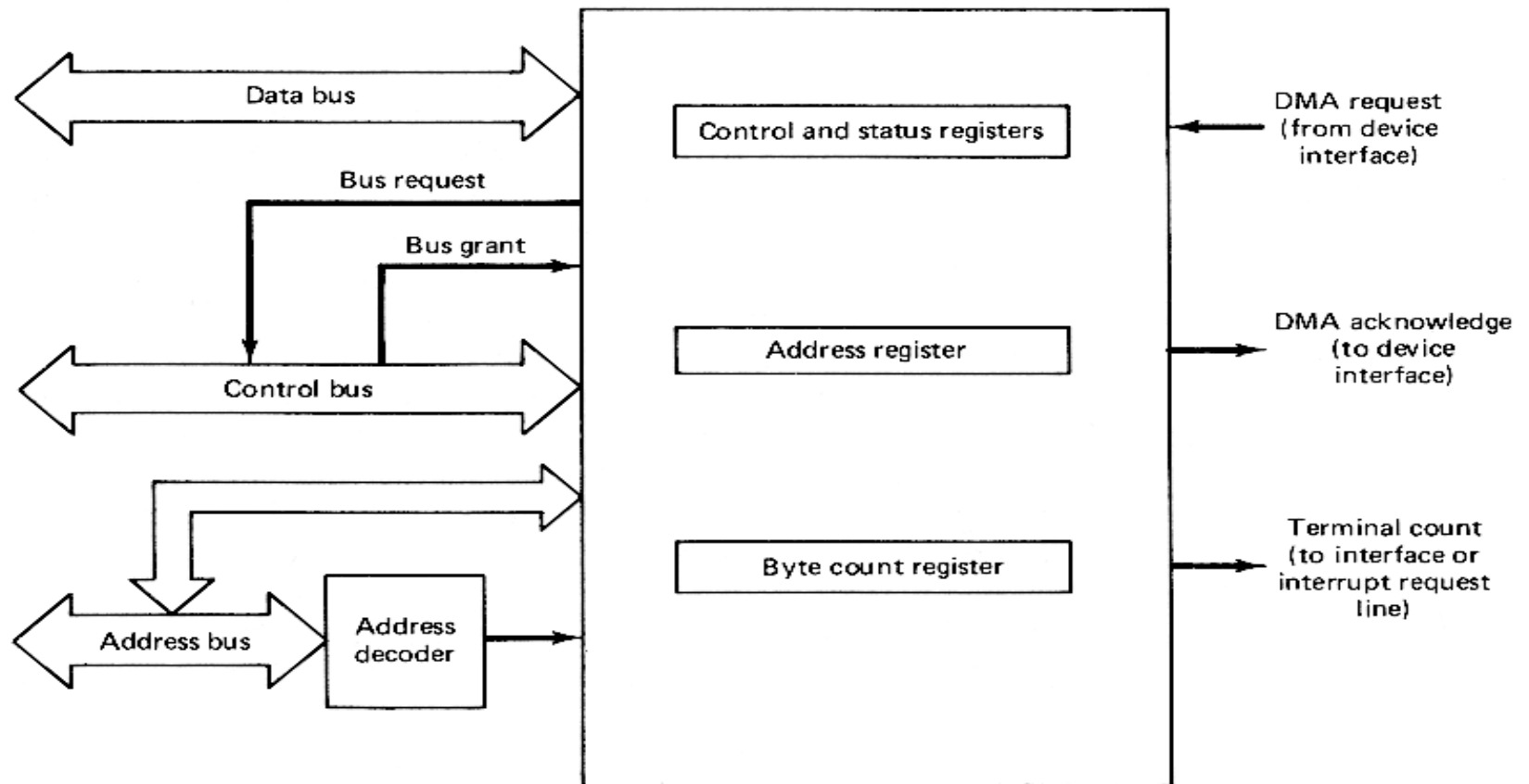
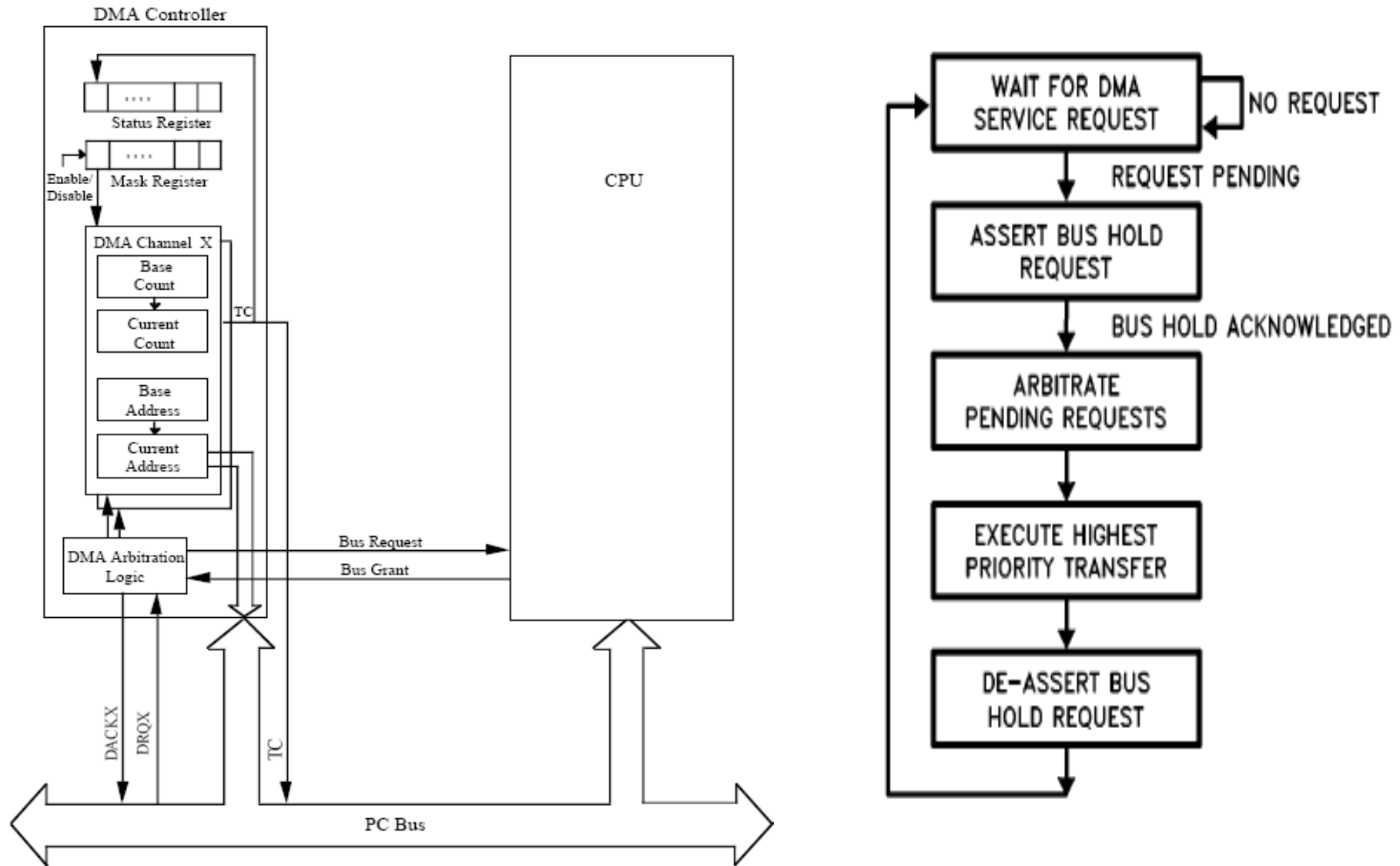
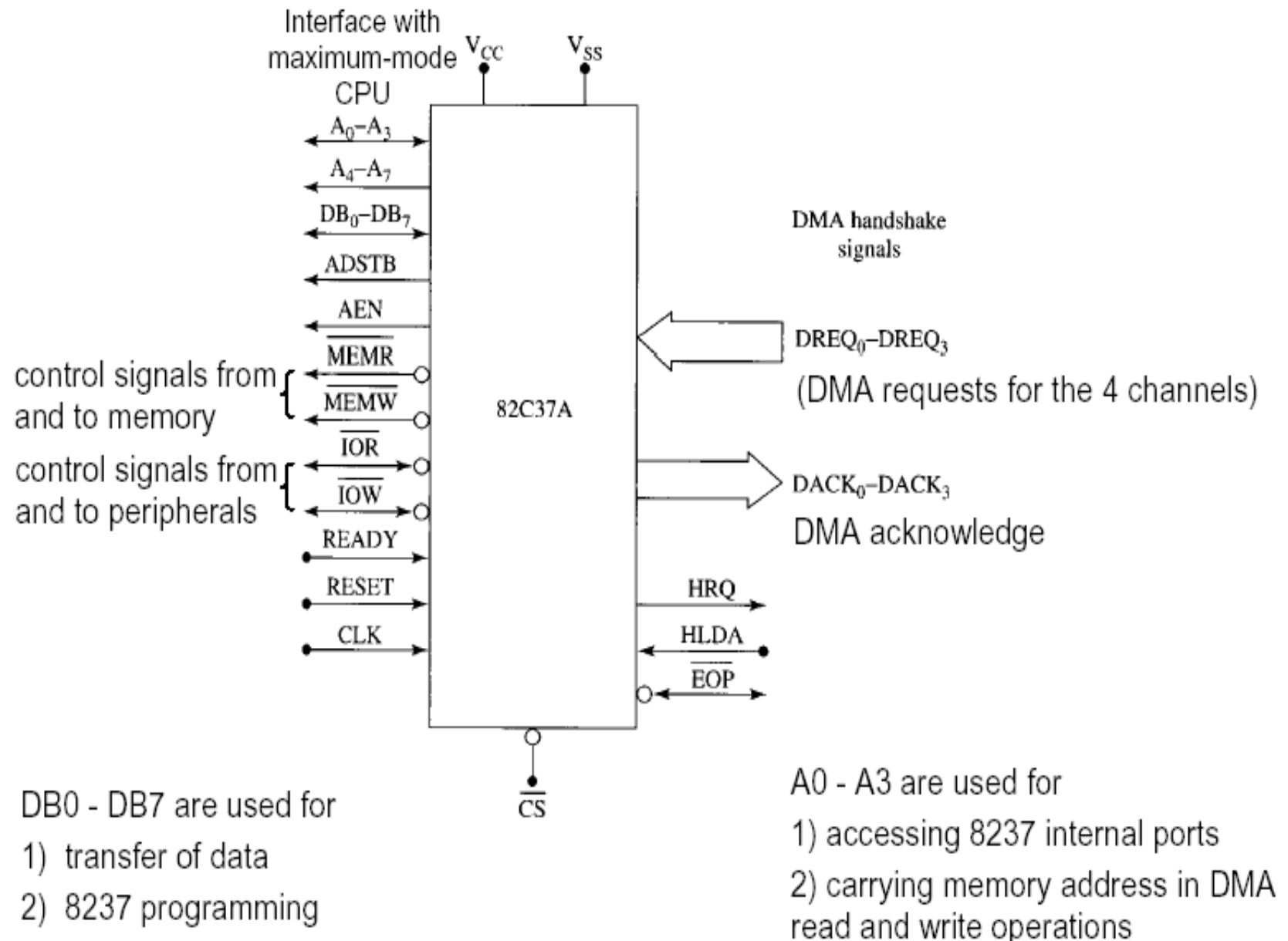


Figure 9-36 General organization of a DMA controller.

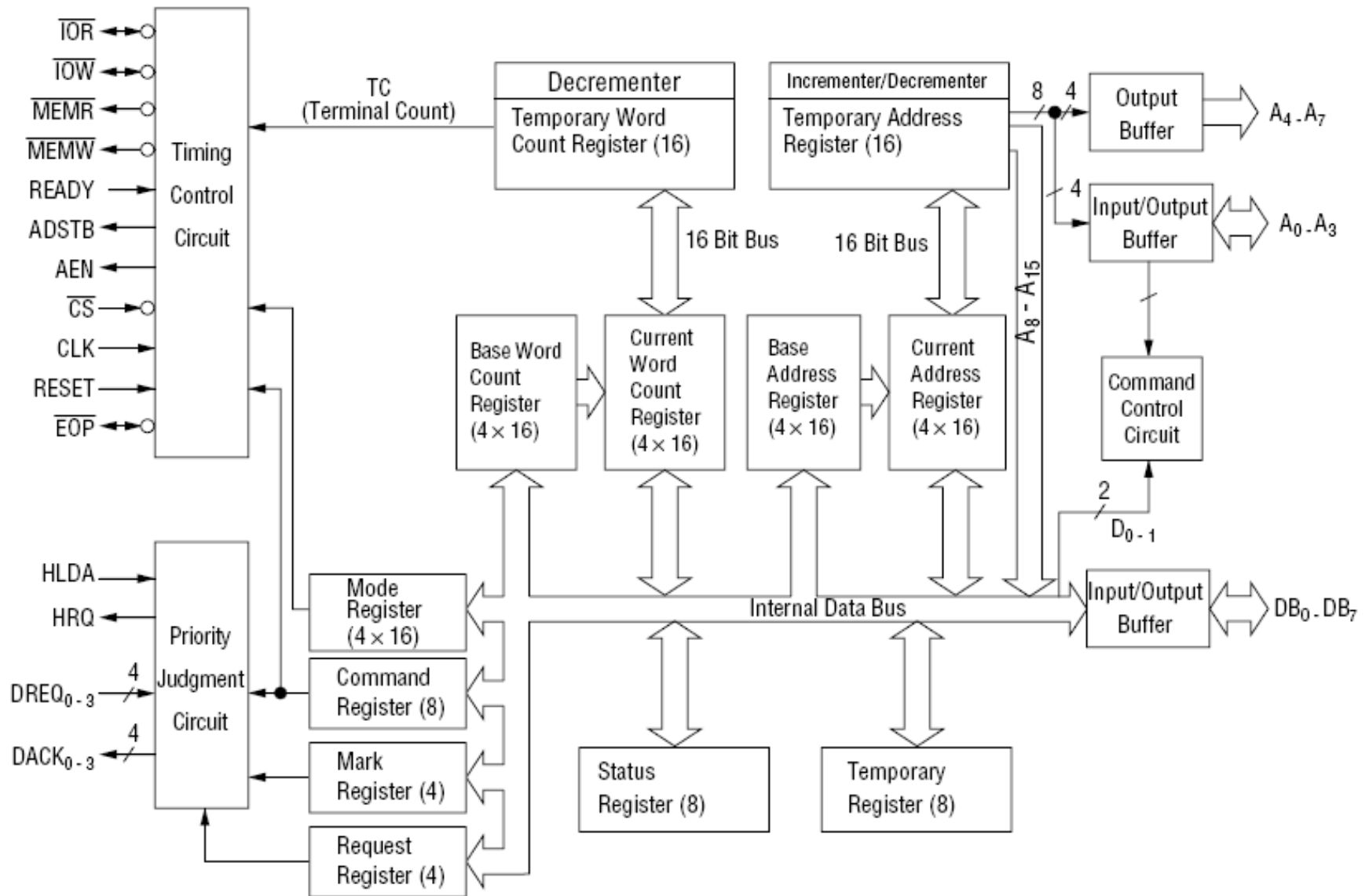
Organizarea generală a unui controller DMA



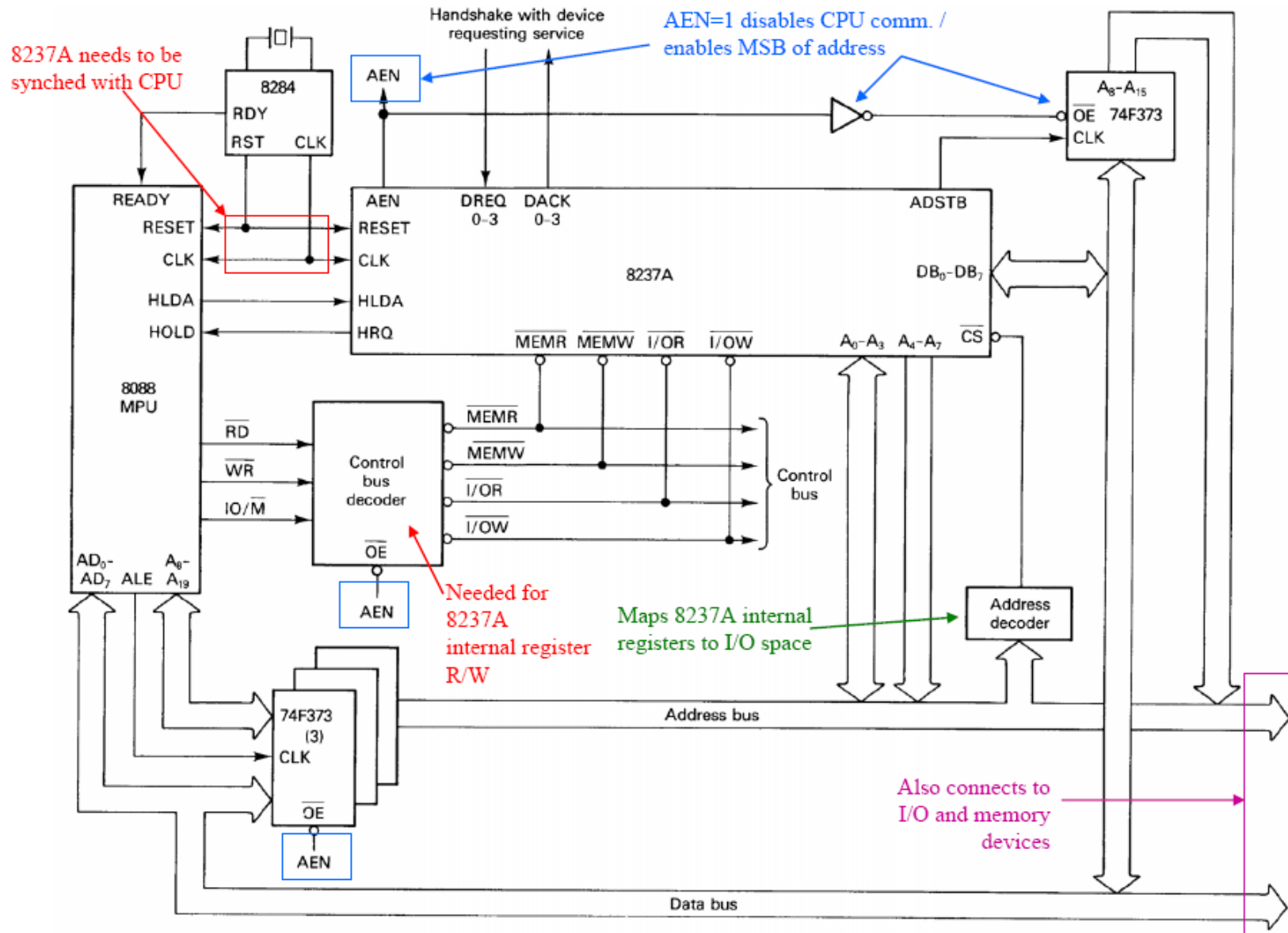
8237 DMA controller



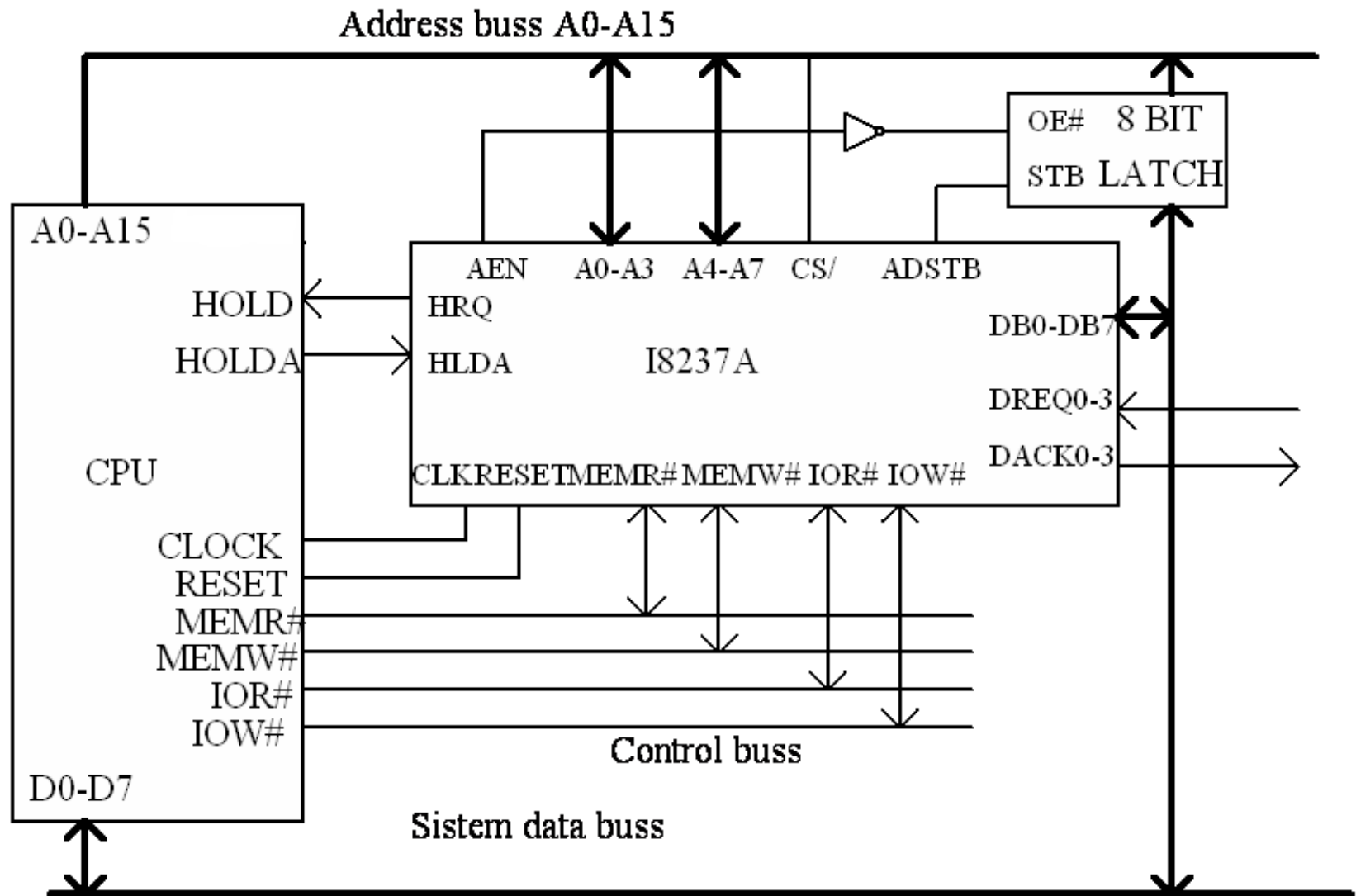
i8237A – Schema bloc



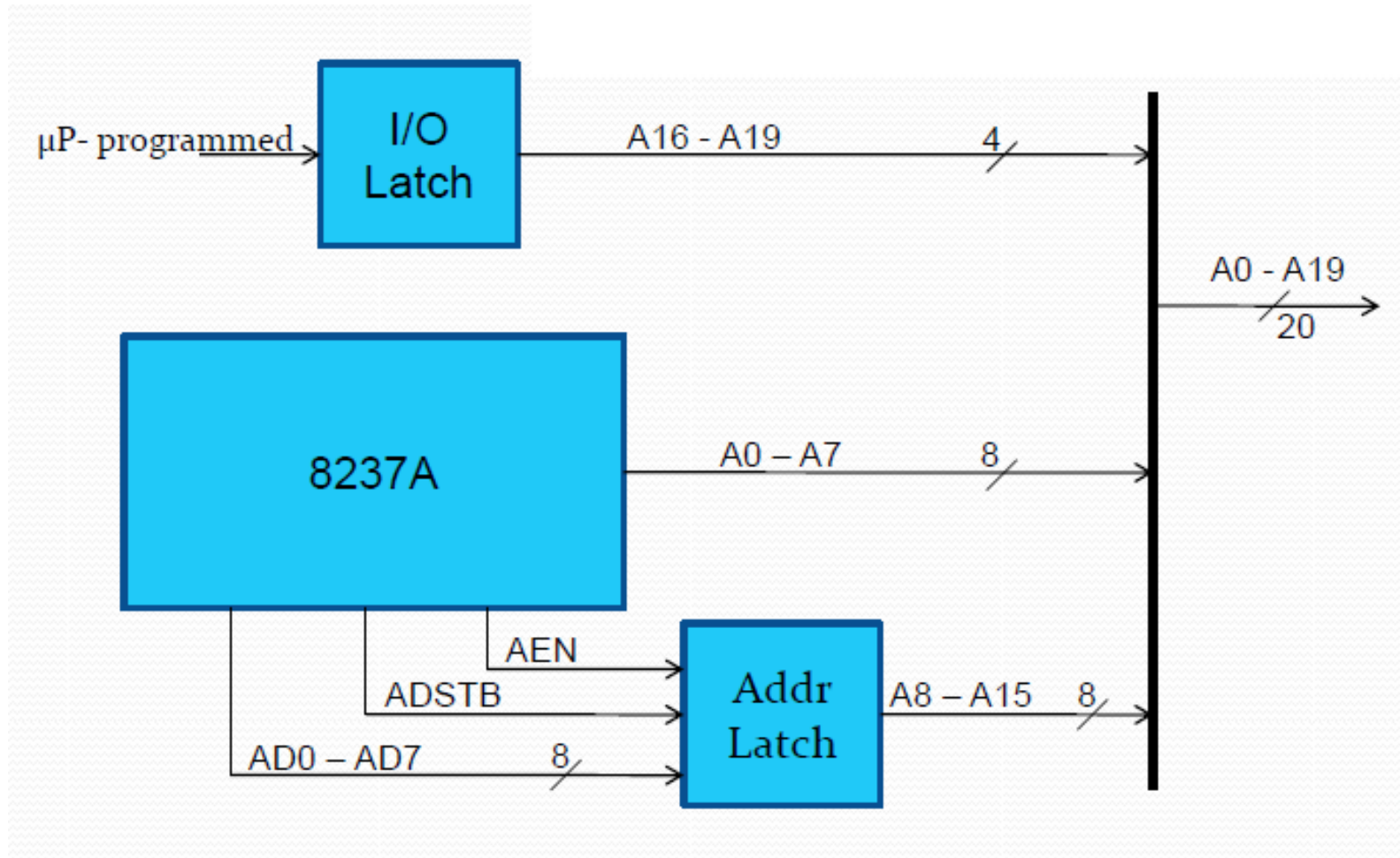
8088 + i8237A



Generarea adreselor de 16 biți

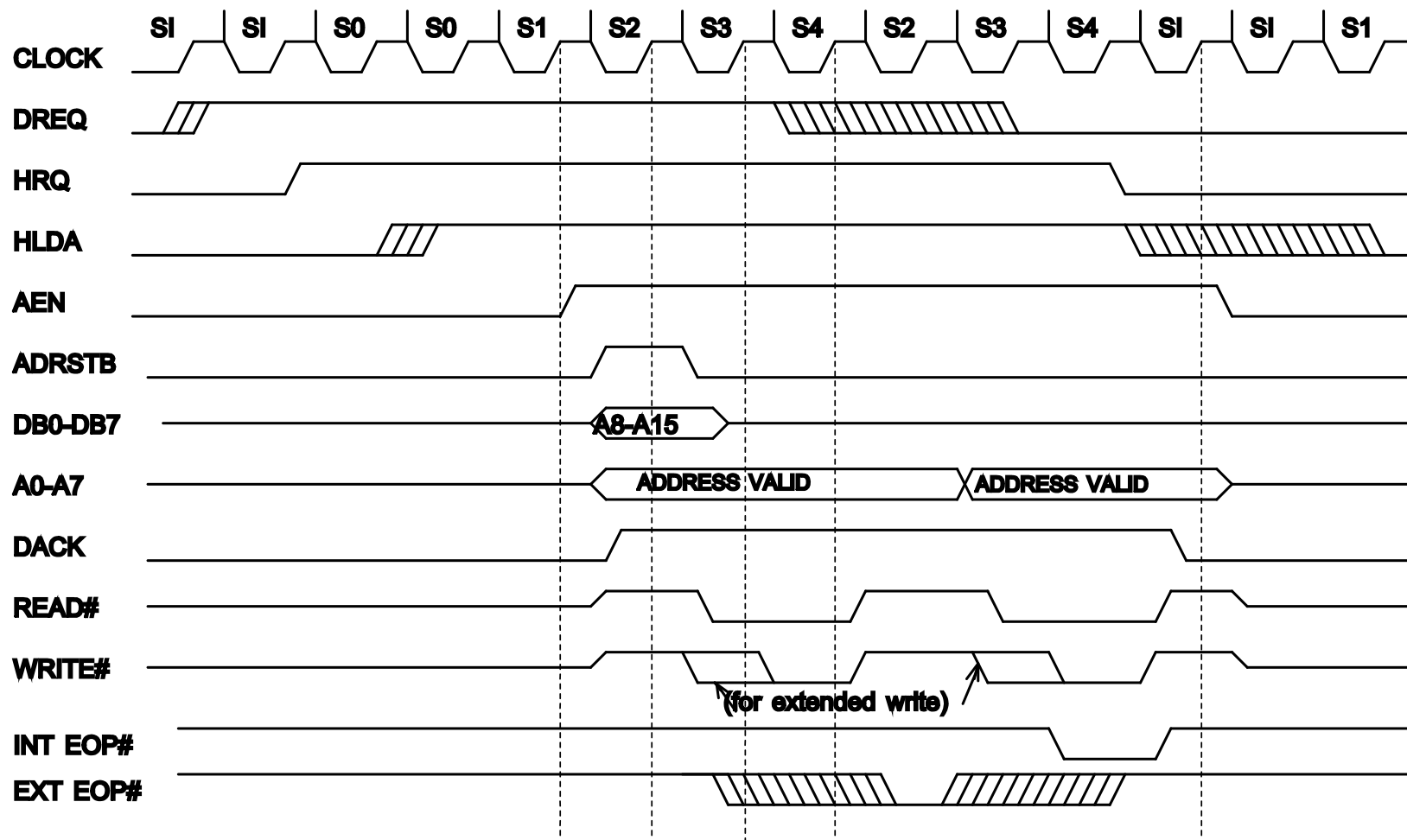


Generarea adreselor de 20 de biți



Ciclu de transfer DMA

(memorie la I/O sau I/O la memorie)



Ciclu comprimat : S2 (schimbă adresa) + S3,S4 (citește/scrie)

Moduri de lucru

Inactiv (slave)

- programare (#CS=0, HOLDA=0)
- DREQ se verifica pe frontul descrescător al Clk
- Daca DREQ este activ pe un canal ne-mascat, sau se face o solicitare software (transfer mem-to-mem) ⇒ Dispozitivul devine Activ

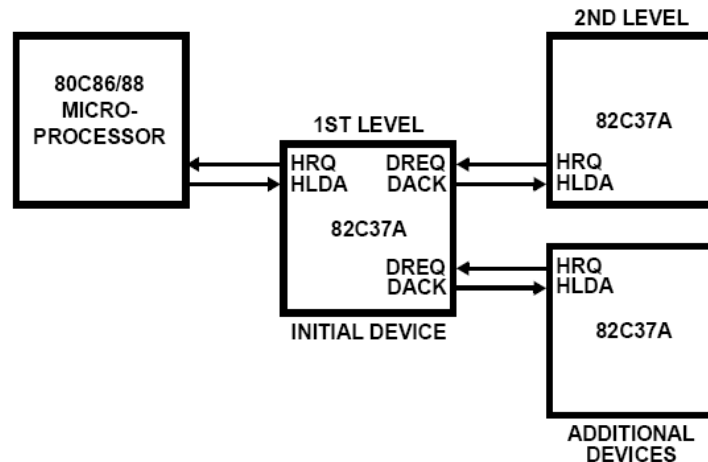
Activ (master) – Transfer DMA:

Modul transfer individual – se eliberează HOLD dupa fiecare octet transferat. Daca DREQ este menținut activ, se cere HOLD din nou.

Modul transfer bloc – se transferă un bloc de date, de dimensiune specificată într-un registru de numărare (DREQ nu trebuie menținut activ).

Transfer la cerere – se transferă date în mod continuu pâna când se primește un #EOP extern, sau până când DREQ devine inactiv.

Mod cascadat -



Tipuri de transfer

Scriere: transfer de date de la dispozitiv I/O la memorie prin activarea **MEMW** și **IOR**.

Citire: transfer de date de la memorie la dispozitiv I/O prin activarea **MEMR** și **IOW**.

Transfer de verificare: sunt pseudo-transferuri. Dispozitivul 82C37A funcționează ca în modurile Citire sau Scriere, generând adrese și răspunzând la EOP, etc, dar semnalele pentru controlul memoriei și al I/O rămân inactive. Modul de verificare nu este permis pentru operații memorie-memorie.

Memorie la memorie

- Canal 0 – adresa și numărătorul pentru sursă.
- Canal 1 – adresa și numărătorul pentru destinație.
- Octetul de date citit din memorie este stocat în registrul intern temporar din 82C37A .
- Transferul este inițiat prin software, sau prin setarea hardware a lui DREQ pentru canalul 0. 82C37A cere un transfer DMA în maniera obișnuită.

Autoinițializare – un canal poate fi configurat ca un canal de autoinițializare. În timpul autoinițializării, valorile originale ale adresei curente și ale numărătorului curent de cuvinte sunt automat restaurate din valoarea adresei de bază și a numărătorului de bază, în urma EOP. După autoinițializare, canalul este gata să efectueze o nouă operație DMA, fără intervenția CPU, dacă se detectează un nou DREQ, sau se face o cerere software.