



Proiectarea cu Micro-Procesoare

Lector: Mihai Negru

An 3 – Calculatoare și Tehnologia Informației

Seria B

Curs 12: Direct Memory Access – DMA

<http://users.utcluj.ro/~negrum/>



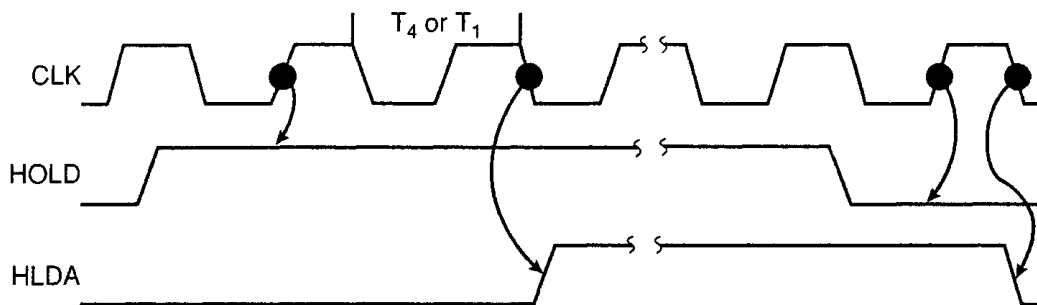
Accesul direct la memorie– DMA



- **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 periferice.
- 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 date la viteză foarte mare. Rata de transfer este limitată de viteza memoriei, de viteza dispozitivelor periferice și de lățimea magistralei.

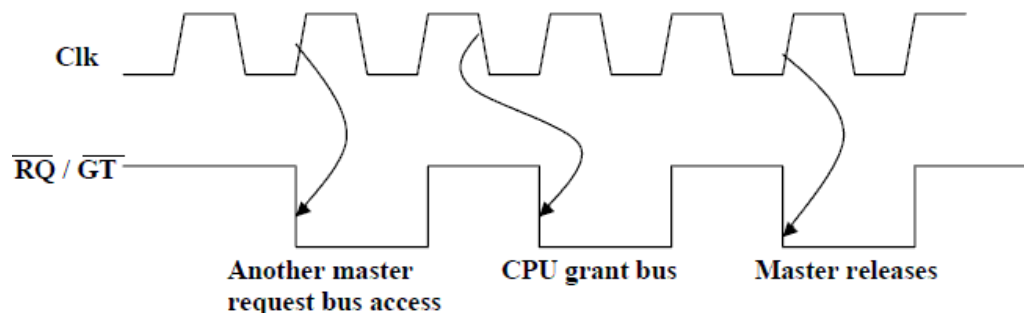


- Pinii **HOLD** și **HLDA** sunt folosiți pentru a primi și a confirma cererea pentru eliberarea magistralei.
- În mod normal, CPU are controlul absolut al magistralei. **În modul de operare DMA, componenta periferică preia controlul magistralei, temporar.**
- Pașii unui proces DMA tipic:
 - 1) Controllerul DMA anunță cererea, pe pinul HOLD.
 - 2) 8086 termină ciclul curent de magistrală, și intră în starea HOLD.
 - 3) 8086 cedează controlul magistralei prin activarea pinului HOLDA. Pinii de lucru cu magistrala de la 8086 (Addressa, Data, Control) se pun pe înaltă impedanță.
 - 4) Incepe transferul DMA.
 - 5) La terminarea transferului DMA, controllerul DMA dezactivează HOLD, pentru a elibera controlul magistralei.





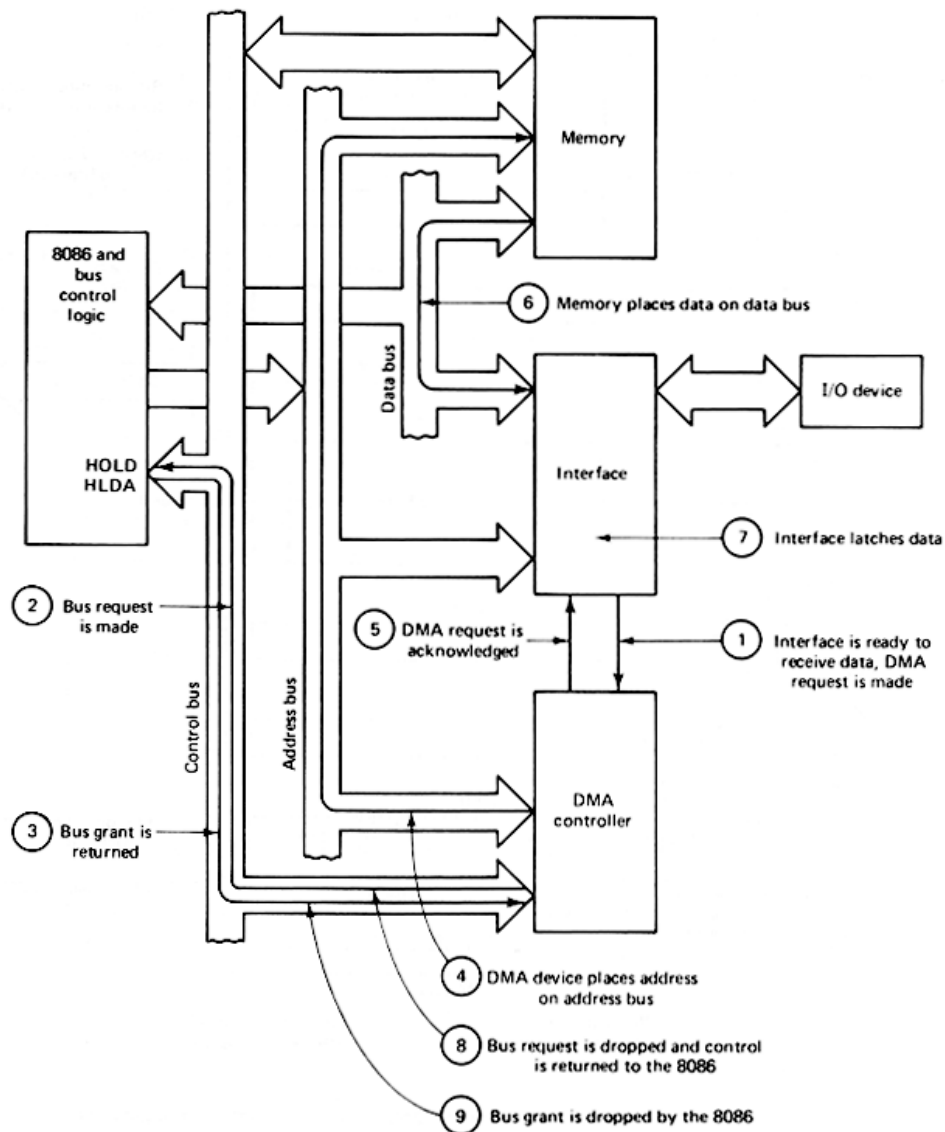
- Pinii RQ/GT1 și RQ/GT0 sunt folositi pentru a emite o cerere DMA și pentru a primi semnalele de confirmare.
- Secvența de evenimente pentru un proces DMA tipic
 - 1) Controllerul DMA activează unul din pinii de cerere, ex. RQ/GT1 sau RQ/GT0 (RQ/GT0 este prioritar)
 - 2) 8086 termina ciclul de magistrală 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) Incepe transferul DMA
 - 5) La terminarea operatiei, controllerul DMA activează pinul de cerere pentru a semnala eliberarea magistralei.



$\overline{RQ} / \overline{GT}$ Timings in Maximum Mode

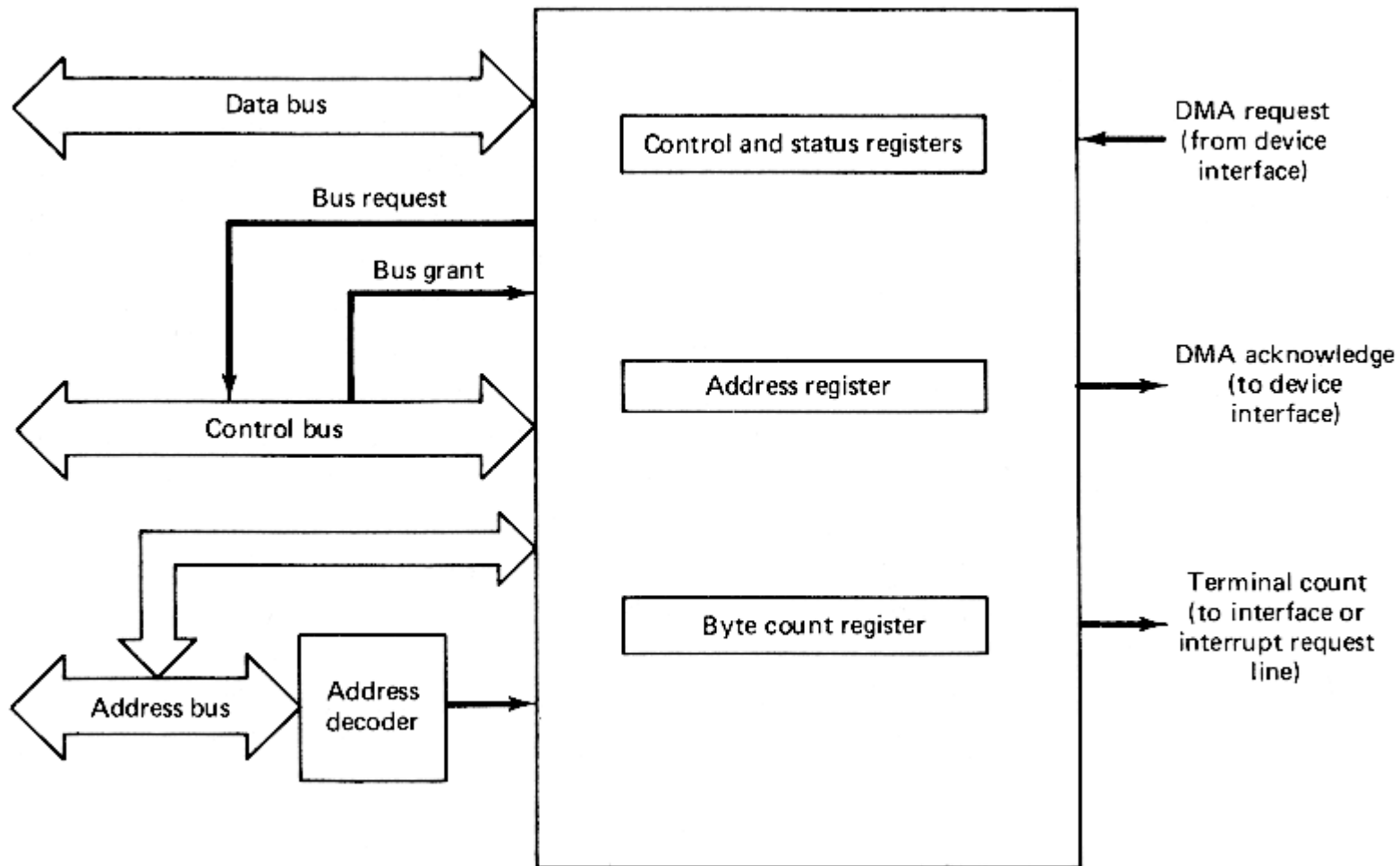


Accesul direct la memorie– DMA



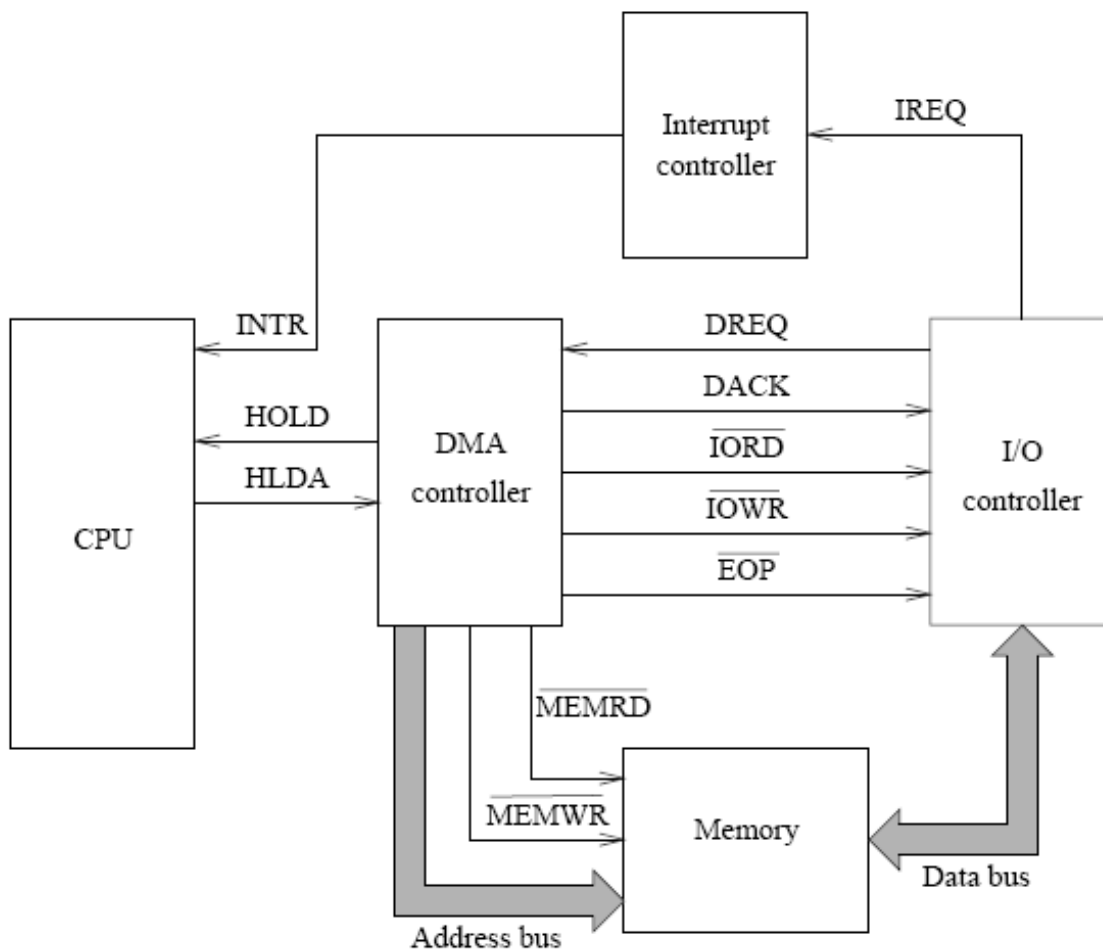


Organizarea generală a unui controller DMA





Semnalele pentru transferul DMA



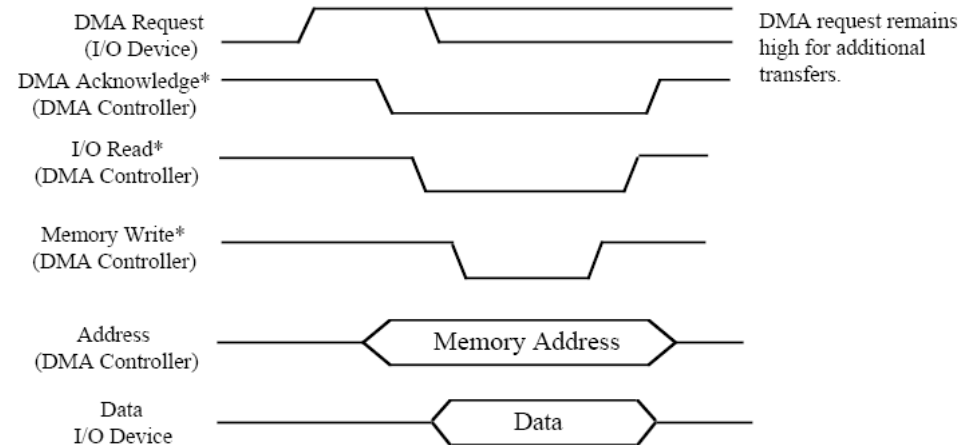


Tipuri de transfer DMA



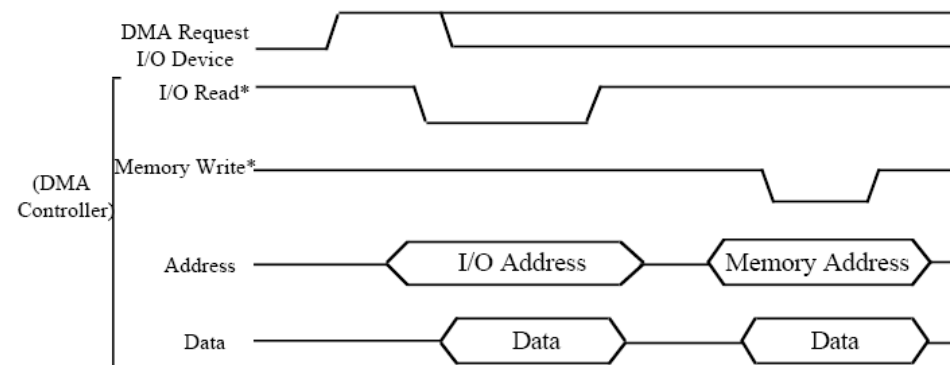
• Transfer “Fly-by”

- 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





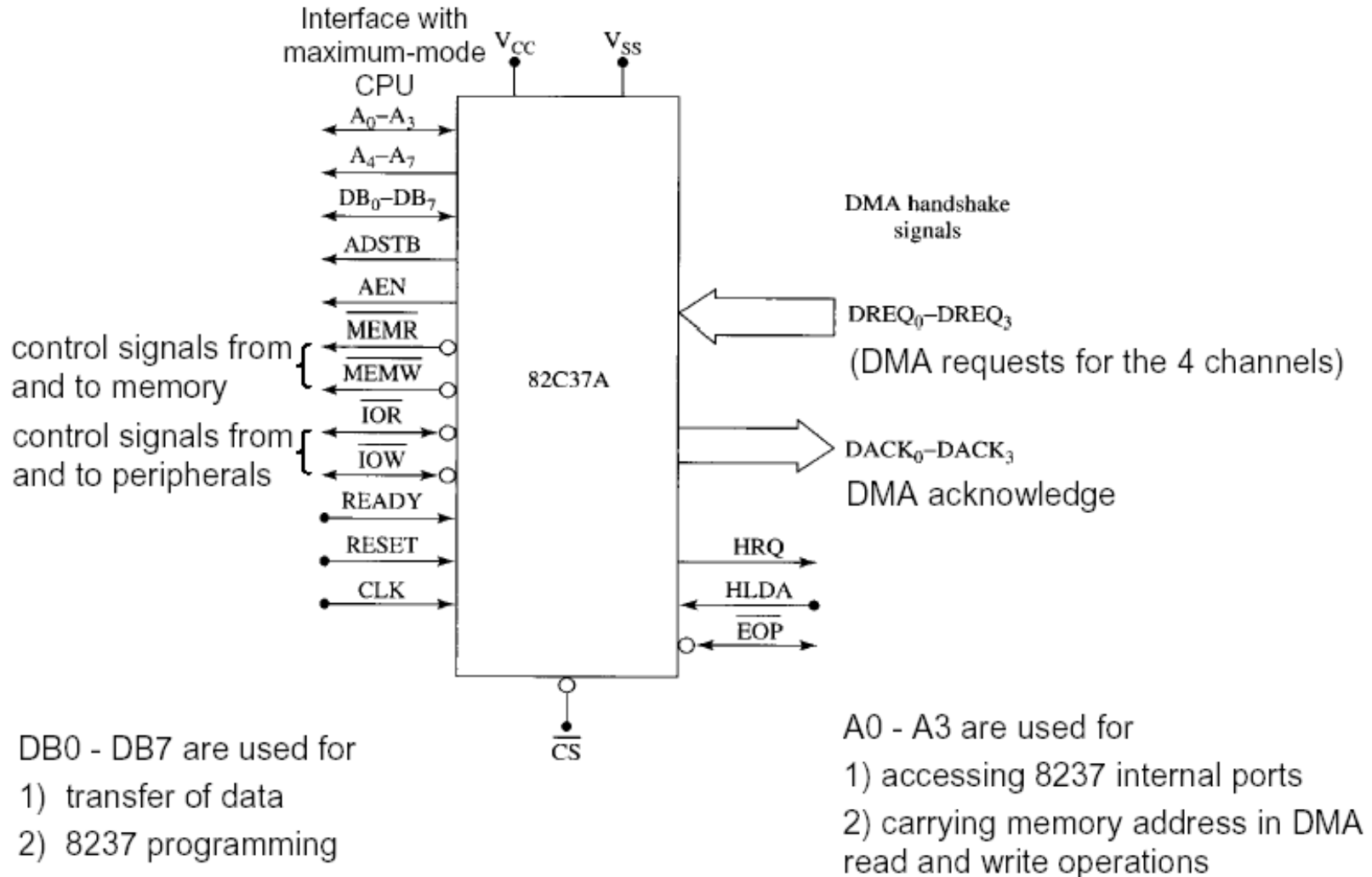
Controller DMA



- Un controller DMA se interfațează cu mai multe periferice care pot cere transfer DMA.
- Controllerul decide prioritatea in cazul cererilor simultane, și oferă adresele de memorie pentru transferul datelor.
- Controllerul DMA folosit cu 8086/8088 este **dispozitivul 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.

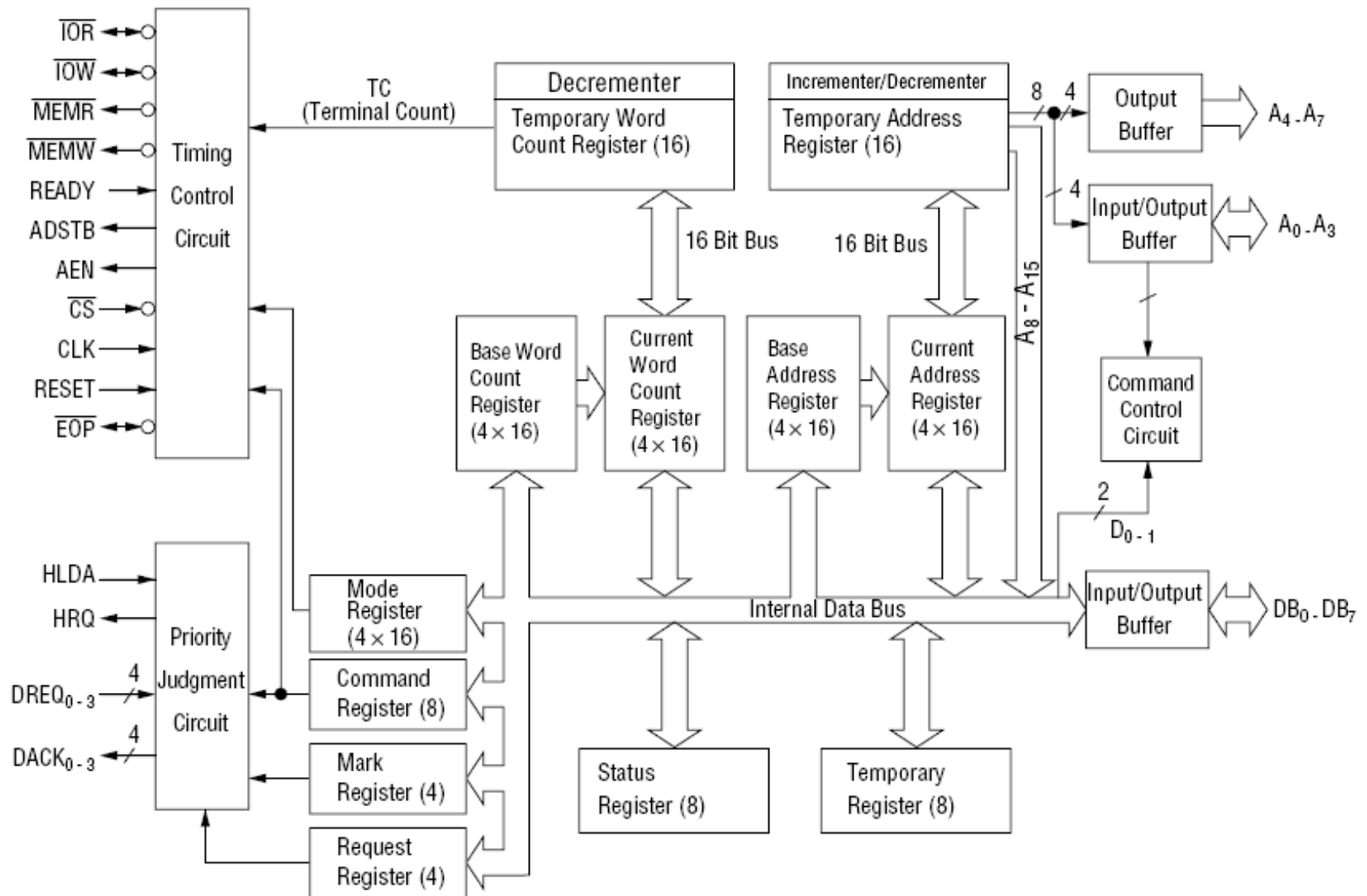


Controller DMA – 8237



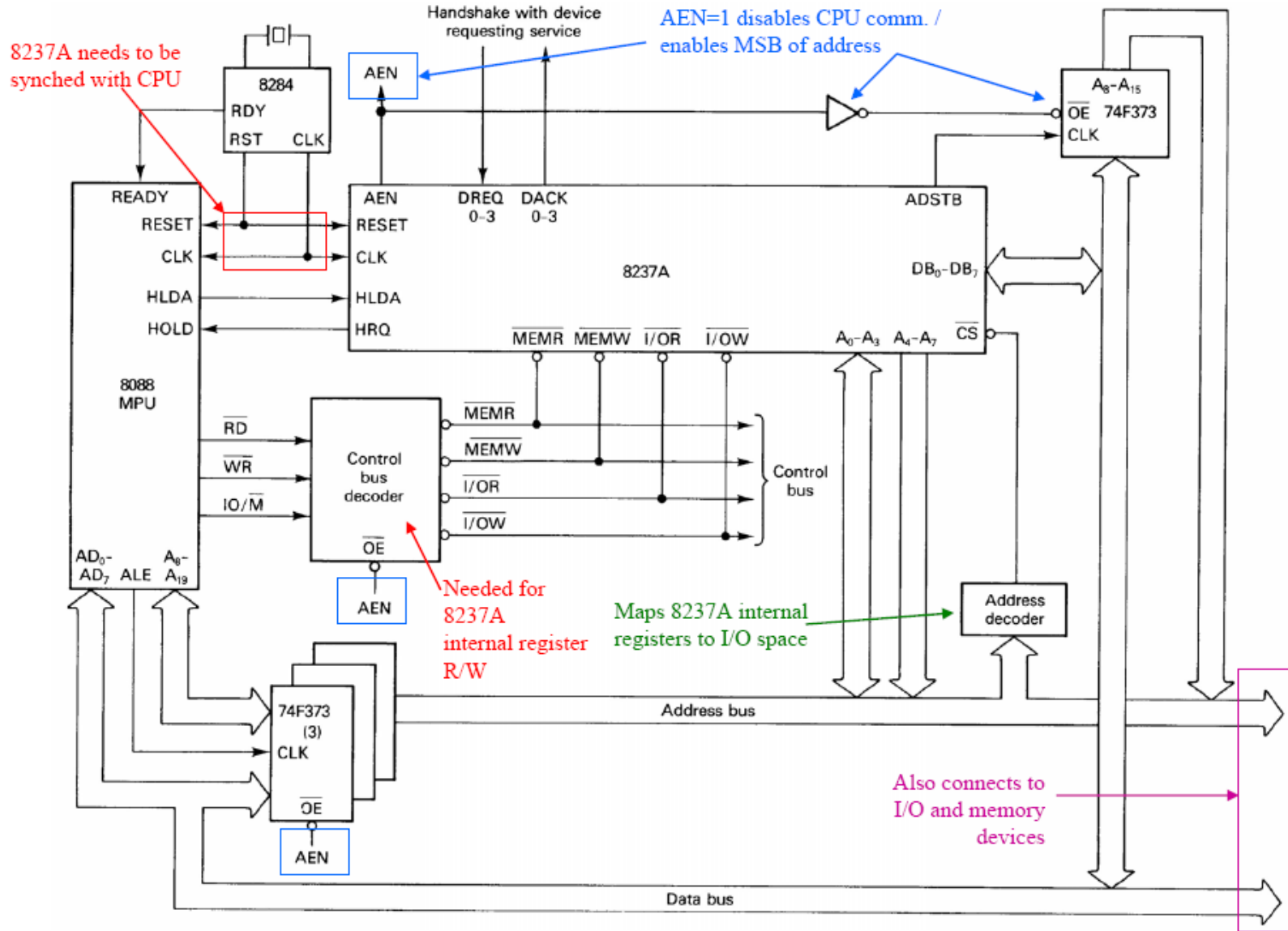


Controller DMA – 8237 (schema bloc)





8088 + i8237A

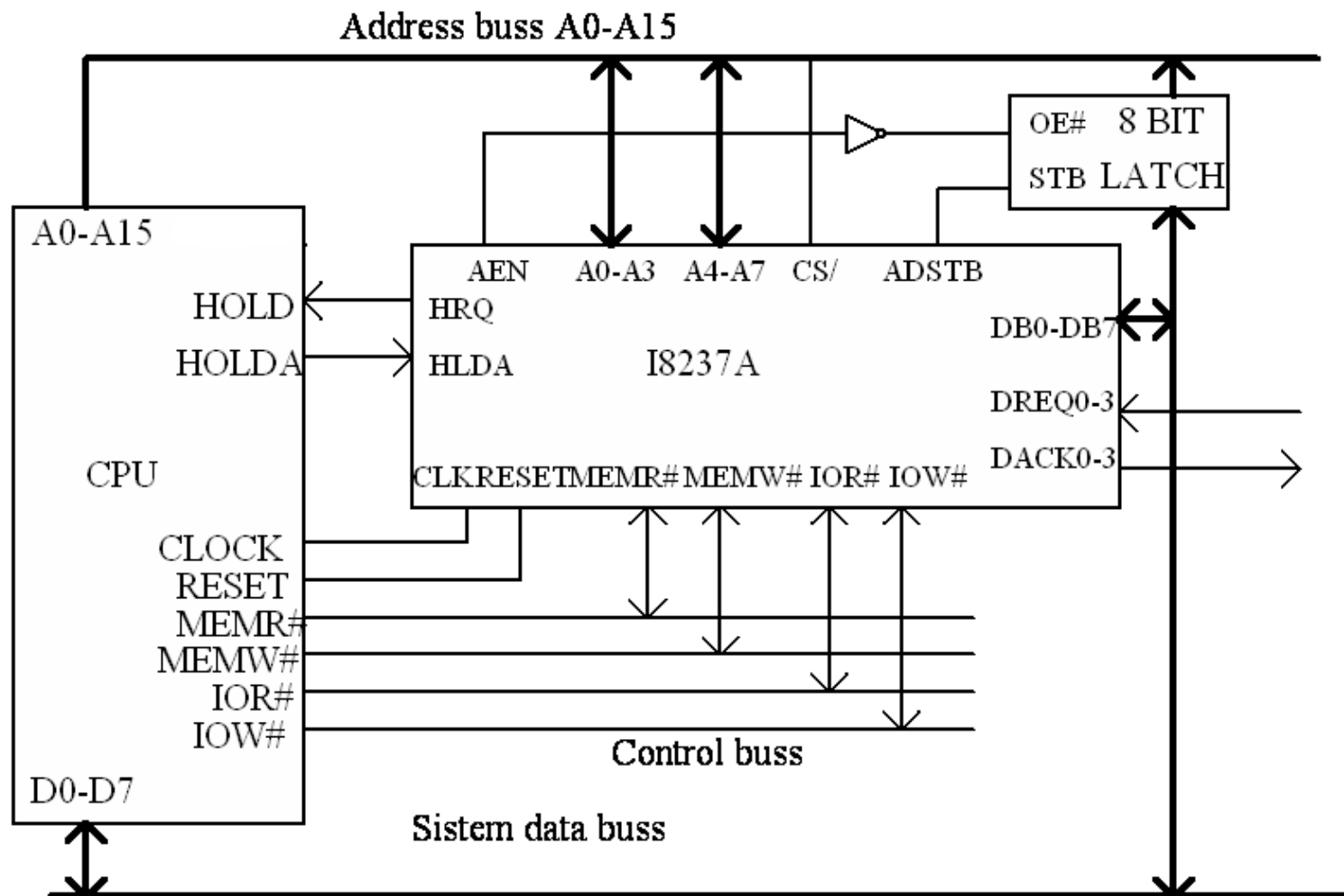




- **Sciere:** transfer de la dispozitiv I/O la memorie prin activarea MEMW și IOR.
- **Citare:** transfer de la memorie la dispozitiv I/O prin activarea MEMR și IOW.
- **Transfer de verificare:** pseudo-transferuri.
 - Dispozitivul 82C37A funcționează ca în modurile Citire sau Sciere, generând adrese 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 manieră obisnuită.
- **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 baza 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.

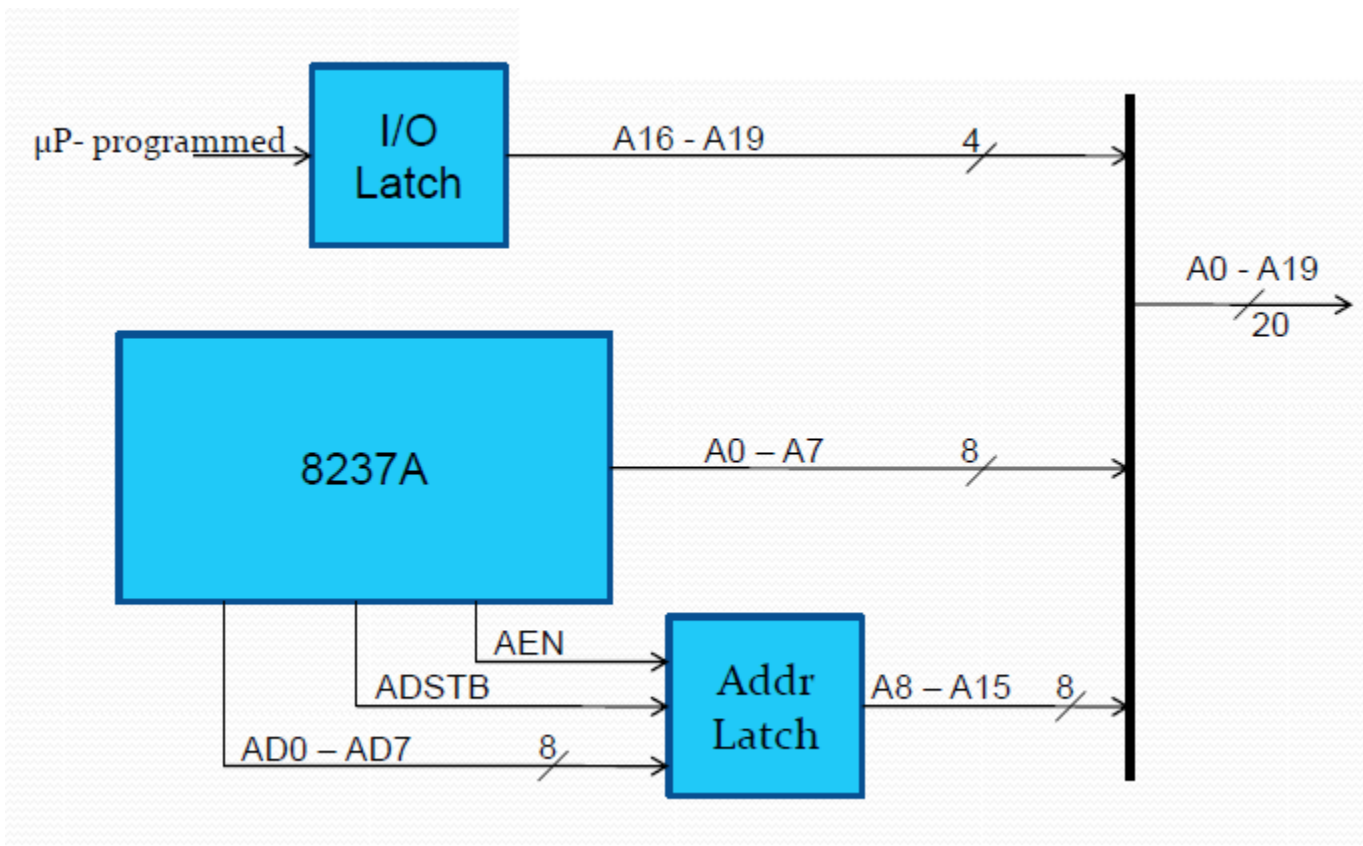


Generarea adreselor de 16 biți





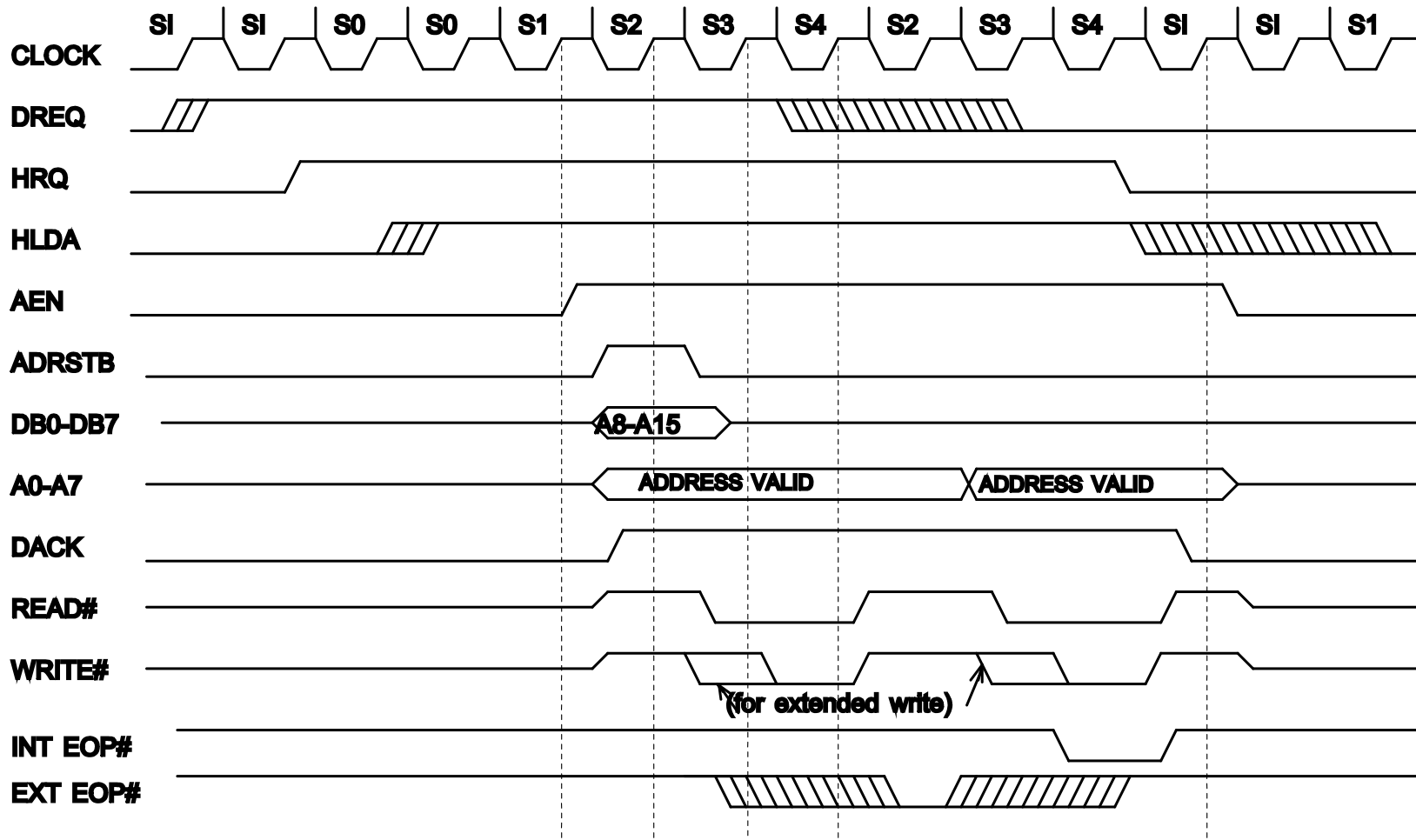
Generarea adreselor de 20 biți





Ciclu de transfer DMA

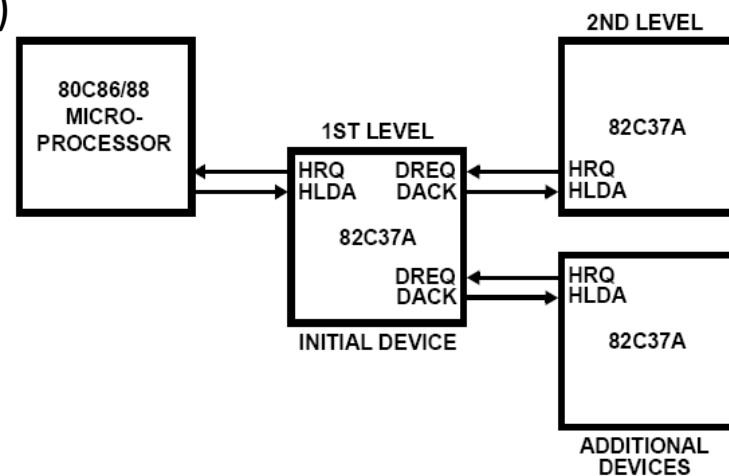
memorie la I/O sau I/O la memorie



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



- Inactiv (slave)
 - programare (#CS=0, HOLDA=0)
 - DREQ se verifică pe frontul descrescător al clk
 - Dacă DREQ este activ pe un canal ne-mască, sau se face o solicitare software (transfer mem-to-mem) → Dispozitivul devine Activ
- Activ (master) – Transfer DMA
 - **Transfer individual** – se eliberează HOLD după fiecare byte transferat. Dacă DREQ este menținut activ, se cere HOLD din nou.
 - **Transfer bloc** – se transferă un bloc de date, de dimensiune specificată într-un registru de numărare (DREQ nu trebuie menținut)
 - **Transfer la cerere** – se transferă date în mod continuu până când se primește un #EOP extern, sau până când DREQ devine inactiv.
 - **Mod cascadat**





1. 82C37A datasheet