

# Digitális rendszerek

Perifériák illesztése



# Perifériakezelési módszerek

## Feltétel nélküli

A perifériának mindig rendelkezésre kell állnia!

Az átvitel in/out utasításokkal történik

Pl.: kapcsolósor állapotának beolvasása, LED-ek kigyújtása latch-en keresztül

Előnye: egyszerű megvalósítás.

Hátránya: nem minden esetben alkalmazható

# Perifériakezelési módszerek

## Feltételes

Állapotolvasással

- A periféria jelzi az átviteli igényét.
- A processzornak folyamatosan olvasnia kell a periféria állapotát, egyébként az igény elveszhet!
- Igény esetén a processzor végrehajtja az átvitelt.

Előnye: egyszerű hardver.

Hátránya: Az állapotolvasás foglalja a processzort.

# Perifériakezelési módszerek

## Feltételes

### Megszakítással

- A periféria a fenti módon jelzi az átviteli igényét.
- A periféria jelzését egy megszakításvezérlővel vagy anélkül a processzorhoz illesztettük.
- A periféria igényének beérkezésekor:
  - a CPU által éppen végrehajtott program futása megszakad
  - az igény kiszolgálásra kerül
  - a megszakított program futása folytatódik.

Előnye: a CPU-nak nem kell a periféria állapotát figyelnie.

Hátránya: valamivel bonyolultabb hardver, sok adat átvitele esetén a regiszterek értékének mentése és visszatöltése jelentősen ronthatja a teljesítményt.

# Perifériakezelési módszerek

## Feltételes

DMA vezérlő segítségével

- A periféria jelzi az átviteli igényét.
- A periféria jelzését a DMA vezérlő dolgozza fel.
- A DMA vezérlő elveszi a rendszersínt a processzortól és az átvitelt maga hajtja végre.
- Az átvitel megtörténtét a DMA vezérlő jelzi a CPU felé pl.: státusz beállításával vagy megszakítással.
- A DMA vezérlő visszaadja a rendszersínt a processzornak

Előnye: hatékonyabb adatátvitel

Hátránya: bonyolultabb hardver, DMA vezérlő szükséges

# Példák perifériák illesztésére

## Feltétel nélküli periféria kezelés

Illesszük a 8086 processzorhoz a következő perifériákat:

8db kapcsolót (ellenállások felhasználásával) az 1Ch perifériacímre és

8 db LED-et (8 bites latch felhasználásával) az 1Dh perifériacímre

# Példák perifériák illesztésére

## Állapotolvasásos periféria kezelés

Illesszünk 8086 processzorhoz

8kB ROM-ot a 0000h címre,

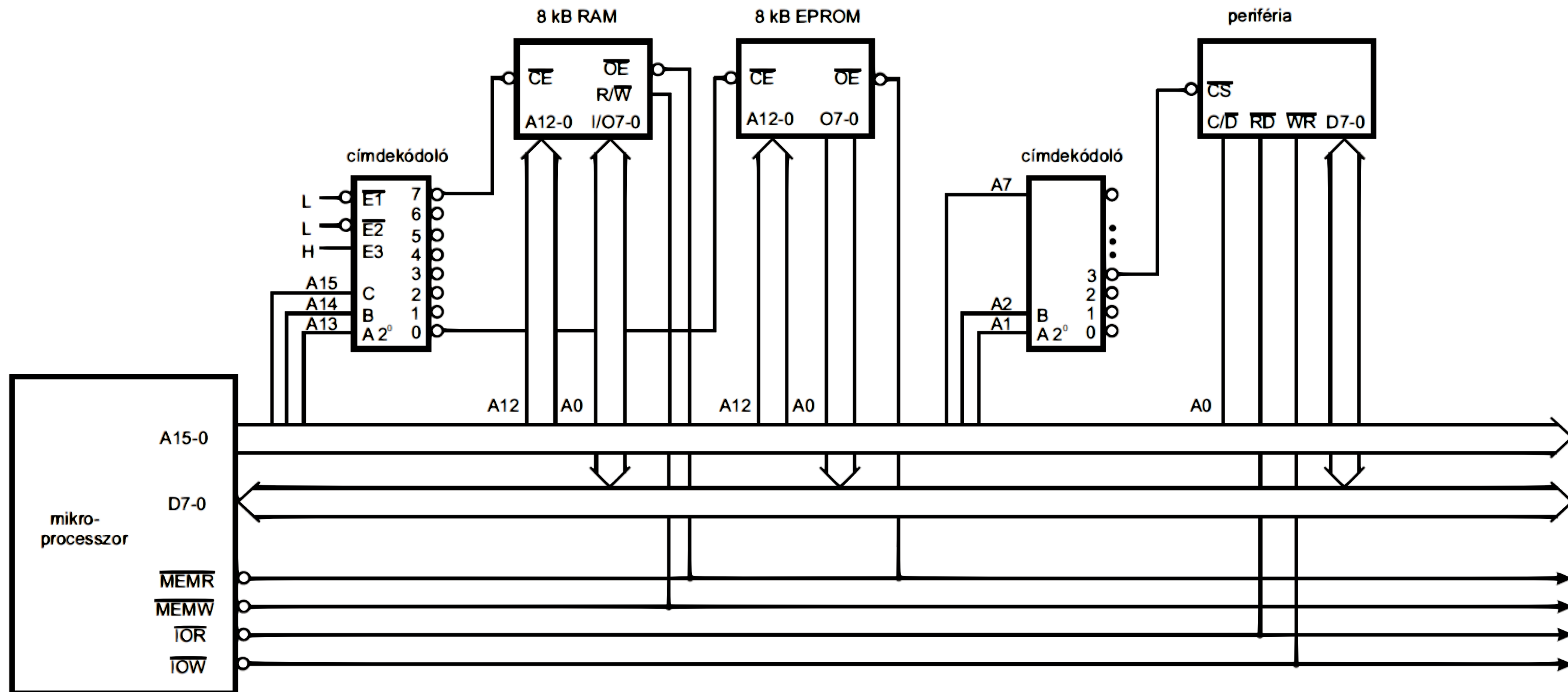
8kB RAM-ot az E000h címre,

Egy 2 I/O címet elfoglaló perifériát a 06h/07h perifériacímre:

a 07h címet íráskor vezérlésre, olvasáskor állapotolvasásra,

a 06h címet pedig adatátvitelre lehessen használni!

# Példák perifériák illesztésére





# Példák perifériák illesztésére

## Állapotolvasásos periféria kezelés

Illesszünk 8086 processzorhoz:

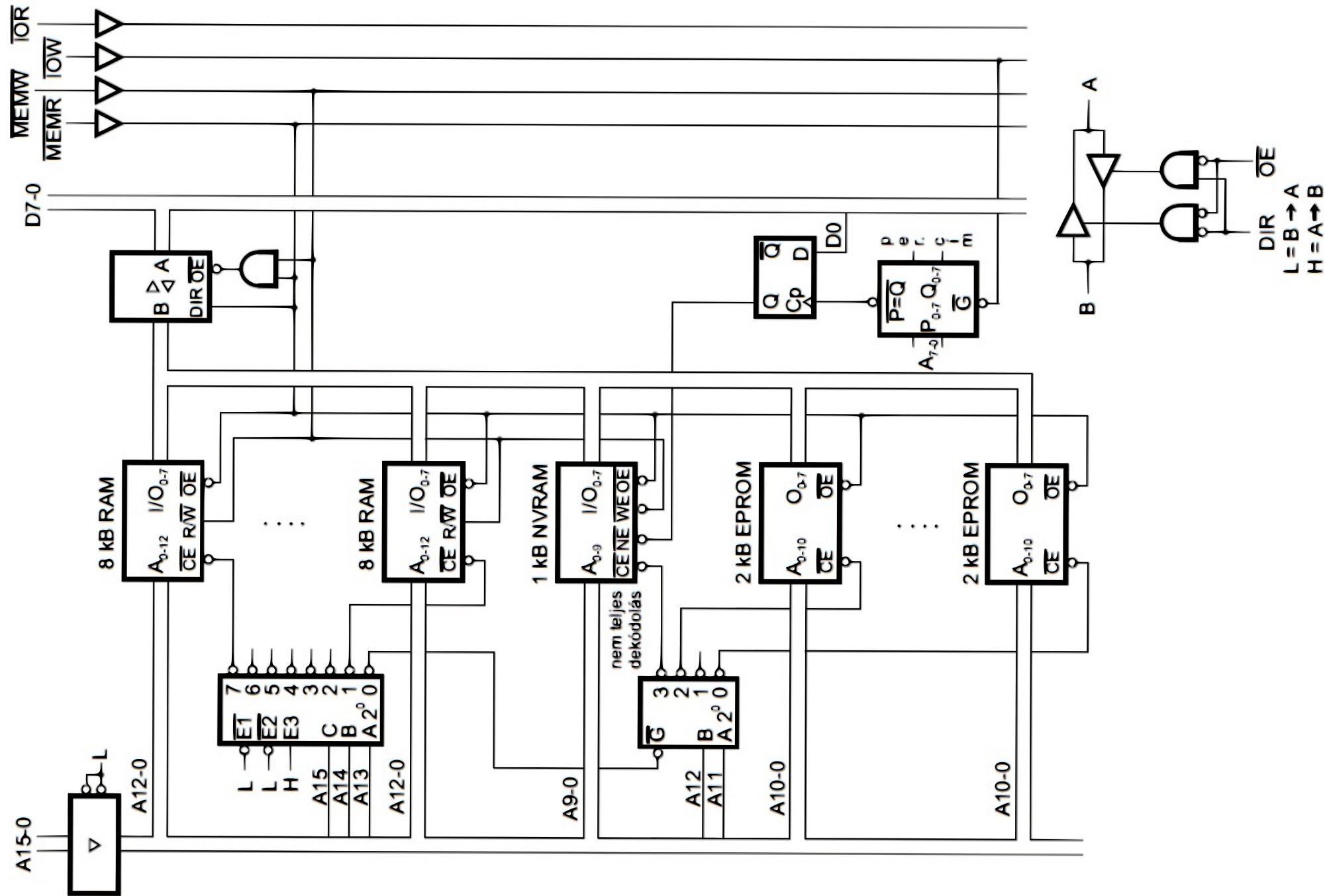
3 x 2kB ROM-ot a 0000h címtől,

1kB NVRAM-ot az 1800h címtől (nem teljes címdekódolás megengedett!) és

7 x 8kB RAM-ot a 2000h címtől.

Az NVRAM  $\overline{NE}$  bemenetét egy adott perifériacímre írt D0 adatbittel lehessen vezérelni!

# Példák perifériák illesztésére



# i8259A megszakításvezérlő

## Fontosabb jellemzők

8 megszakítási bemenet,  
kaskádosítással 64!

Programozható megszakítási  
üzemmódok

Perifériaként programozható, illeszthető

Két I/O címet használ

Prioritási üzemmódok

Megszakítási igények maszkolhatók

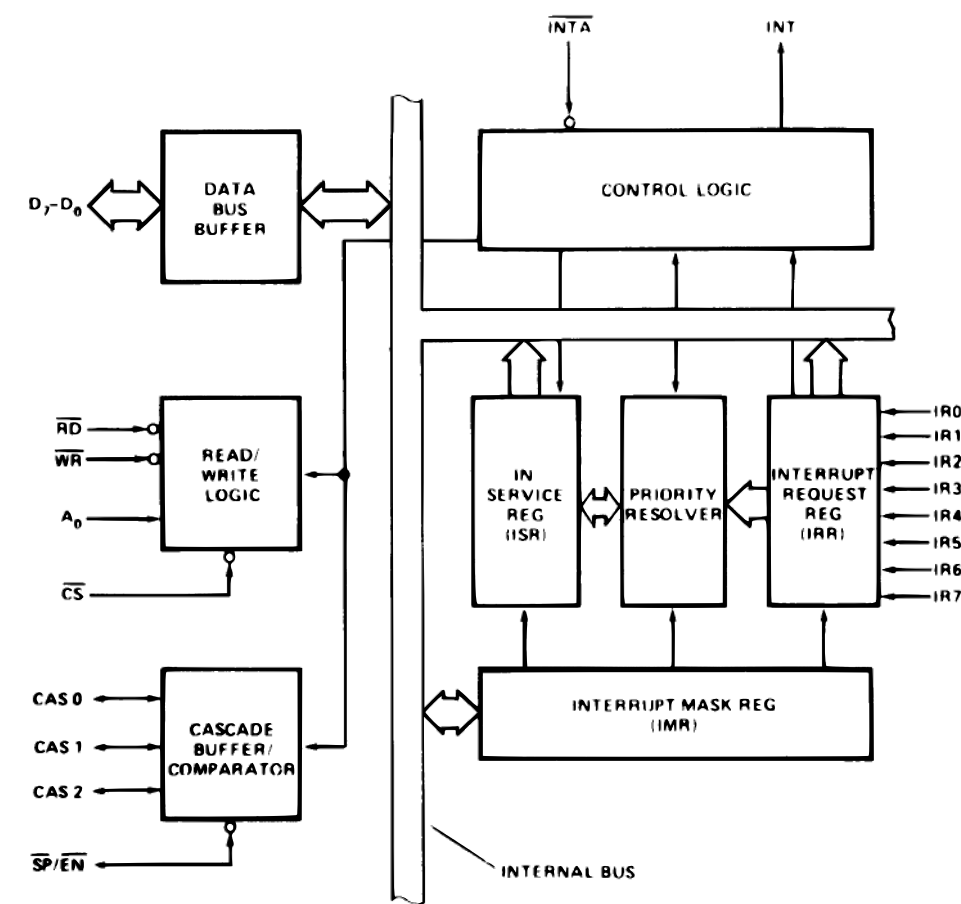


Figure 1. Block Diagram

231468-1

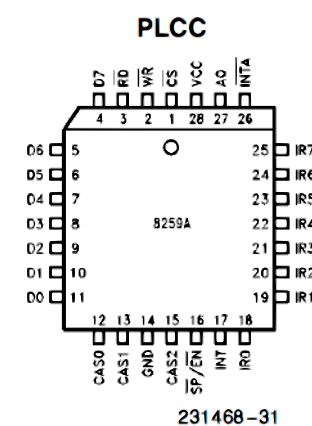
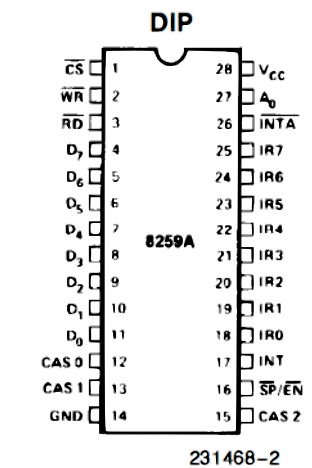


Figure 2. Pin Configurations

# i8259A megszakításvezérlő

## Illesztés CPU-hoz

INT (Interrupt)

magas aktív kimenet, megszakítás jelzése

$\overline{\text{INTA}}$  (Interrupt Acknowledge)

Megszakítás-vektor kérése

$\overline{\text{CS}}$  (Chip Select)

I8259A globális engedélyezése

$\overline{\text{WR}}$  (Write)

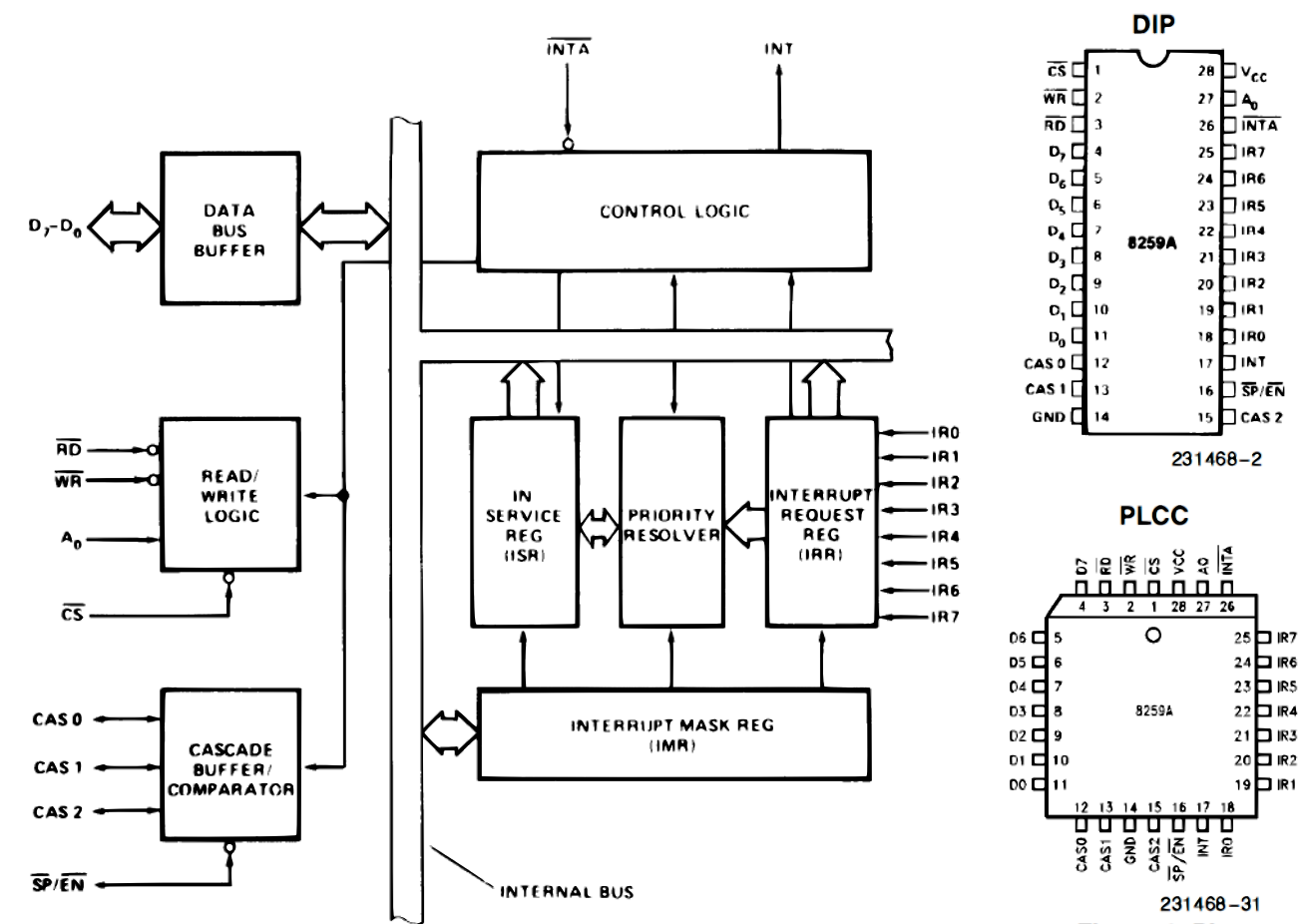
Regiszterek írásának engedélyezése

$\overline{\text{RD}}$  (Read)

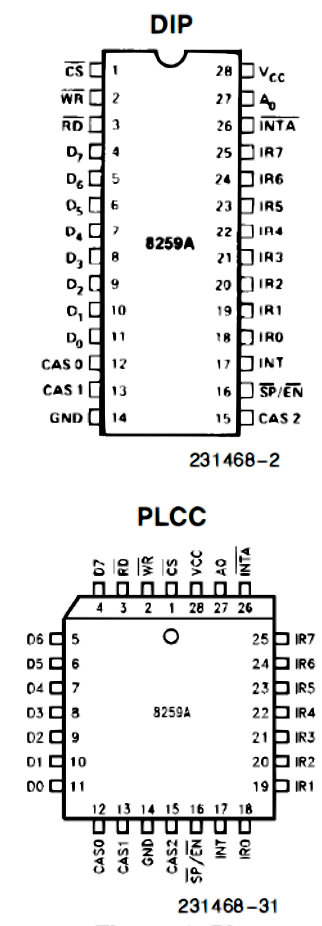
Regiszterek tartalmának kiolvasása

A0

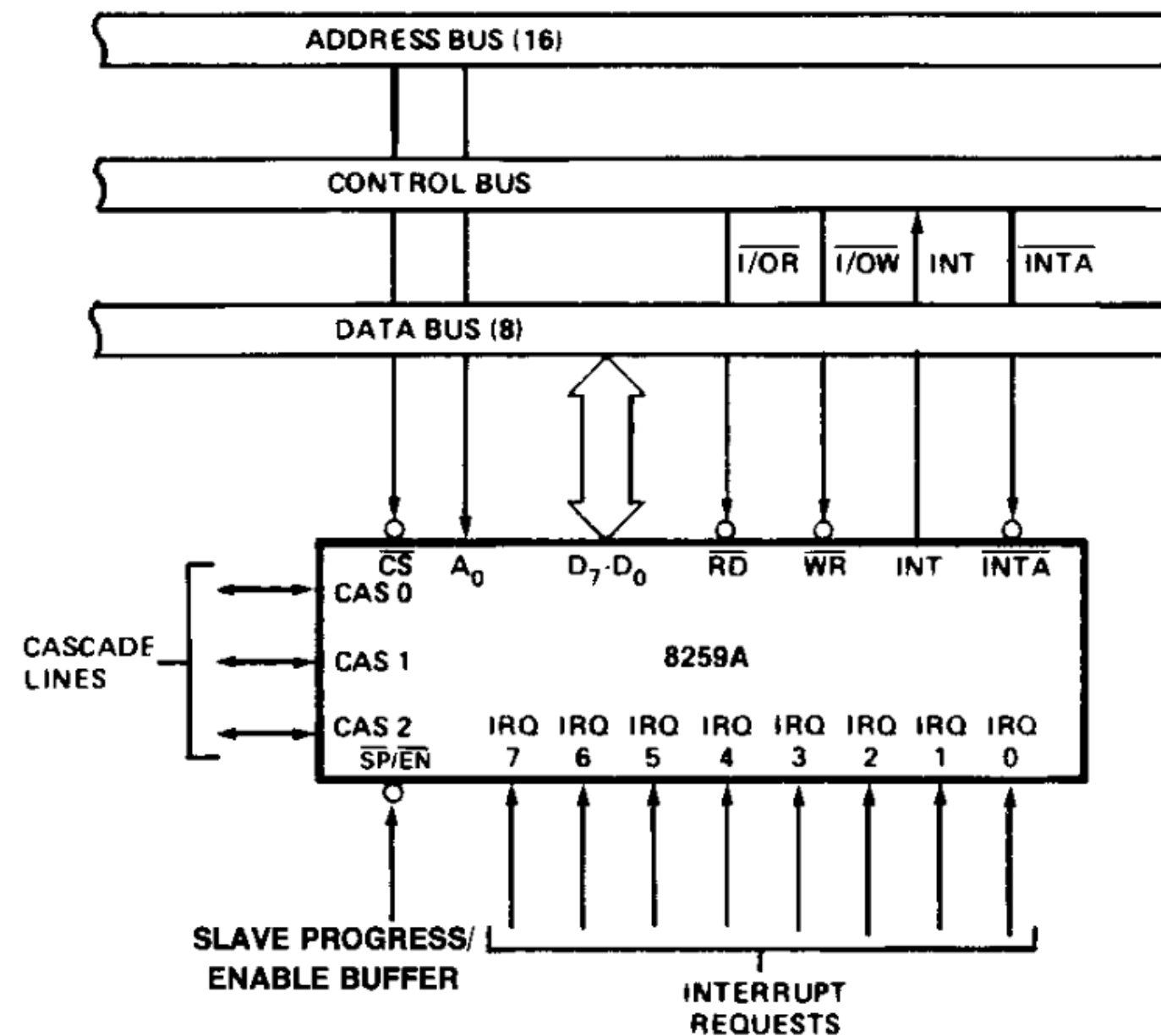
WR és RD jelekkel együttműködve engedélyezi a regiszterek írását és olvasását



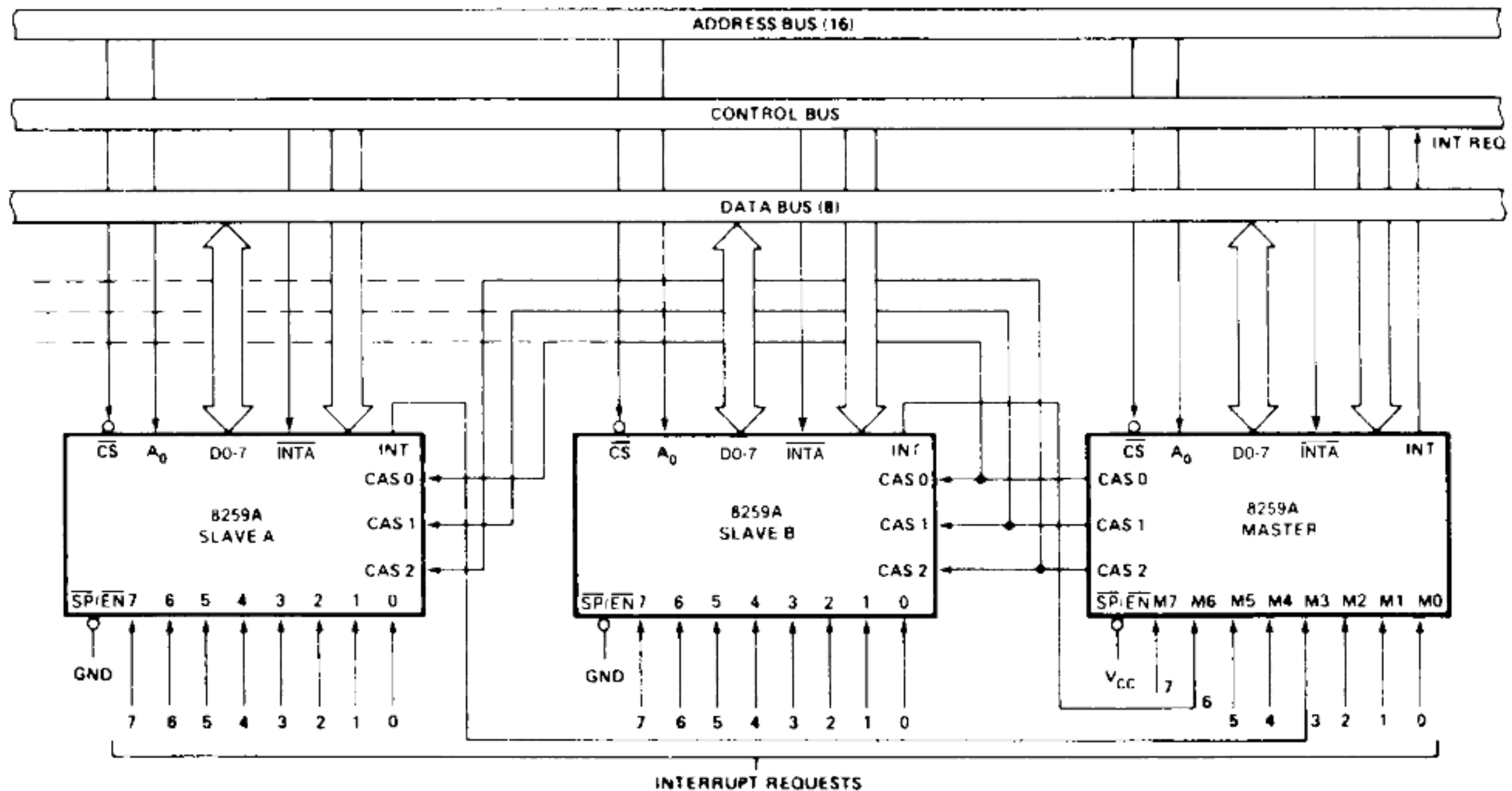
231468-1



# i8259A megszakításvezérlő Illesztés CPU-hoz



# i8259A megszakításvezérlő Illesztés CPU-hoz (kaszkádban)



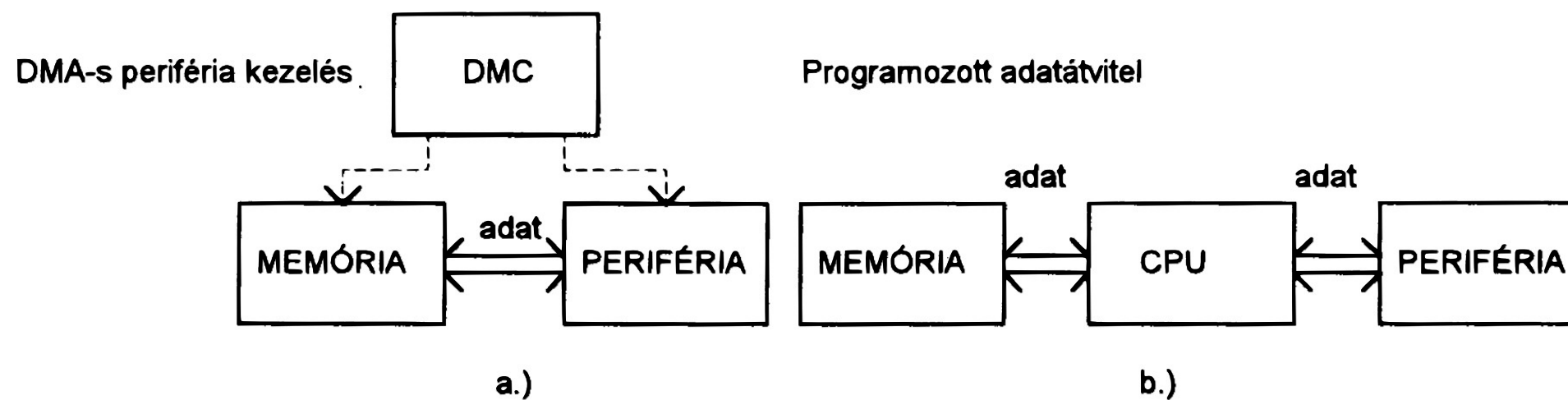
# Közvetlen memória hozzáférés (DMA)

## Jellemzők

Az adatátvitelt egy speciális célhardver (DMA vezérlő) végzi

DMA-s átvitel tehermentesíti a CPU-t

Általában nagy sebességű, blokkos átvitelt igénylő perifériáknál alkalmazzák



# Közvetlen memória hozzáférés (DMA)

## DMA átvitel

Inicializálás (felprogramozás)

Átvitel iránya,

Memóriablokk kezdőcíme

Blokkméret (szószám)

Átvitel módja

Átvitel kérése

Periféria valamelyik DRQi vonalon vagy a program parancsszóval kéri

Rendszersín átvétele a CPU-tól

DRQ-n keresztül elkéri buszt a CPU-tól, a CPU a DACK vonalon hagyja jóvá a kérést



# Közvetlen memória hozzáférés (DMA)

## DMA átvitel

### Adatátvitel

DMA vezérlő az inicializálásnak megfelelően, buszciklusok generálásával lebonyolítja az adatátvitelt.

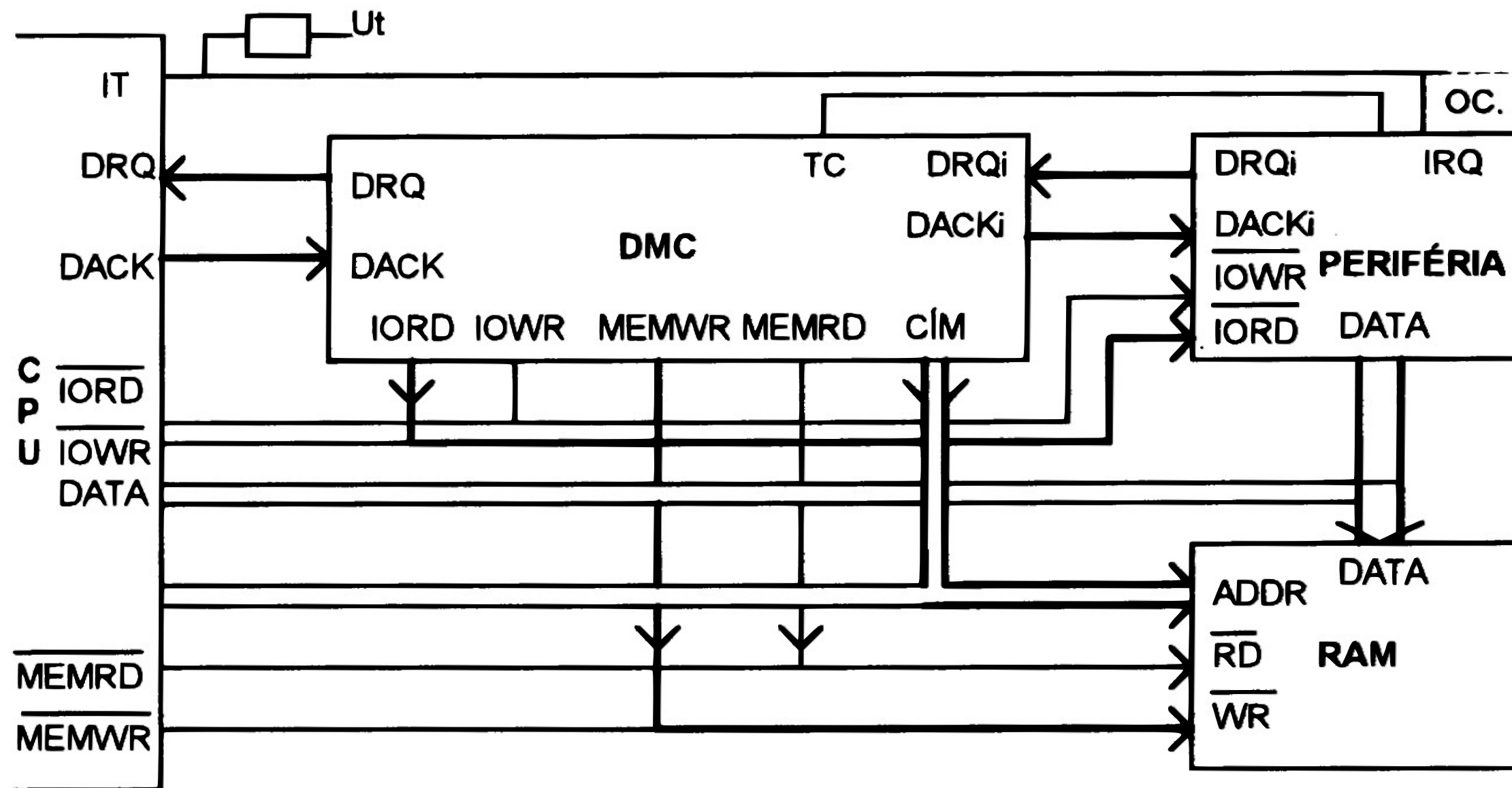
### Rendszersín visszaadása CPU-nak

Lehetséges a teljes adatblokkra „elkérni” a buszt

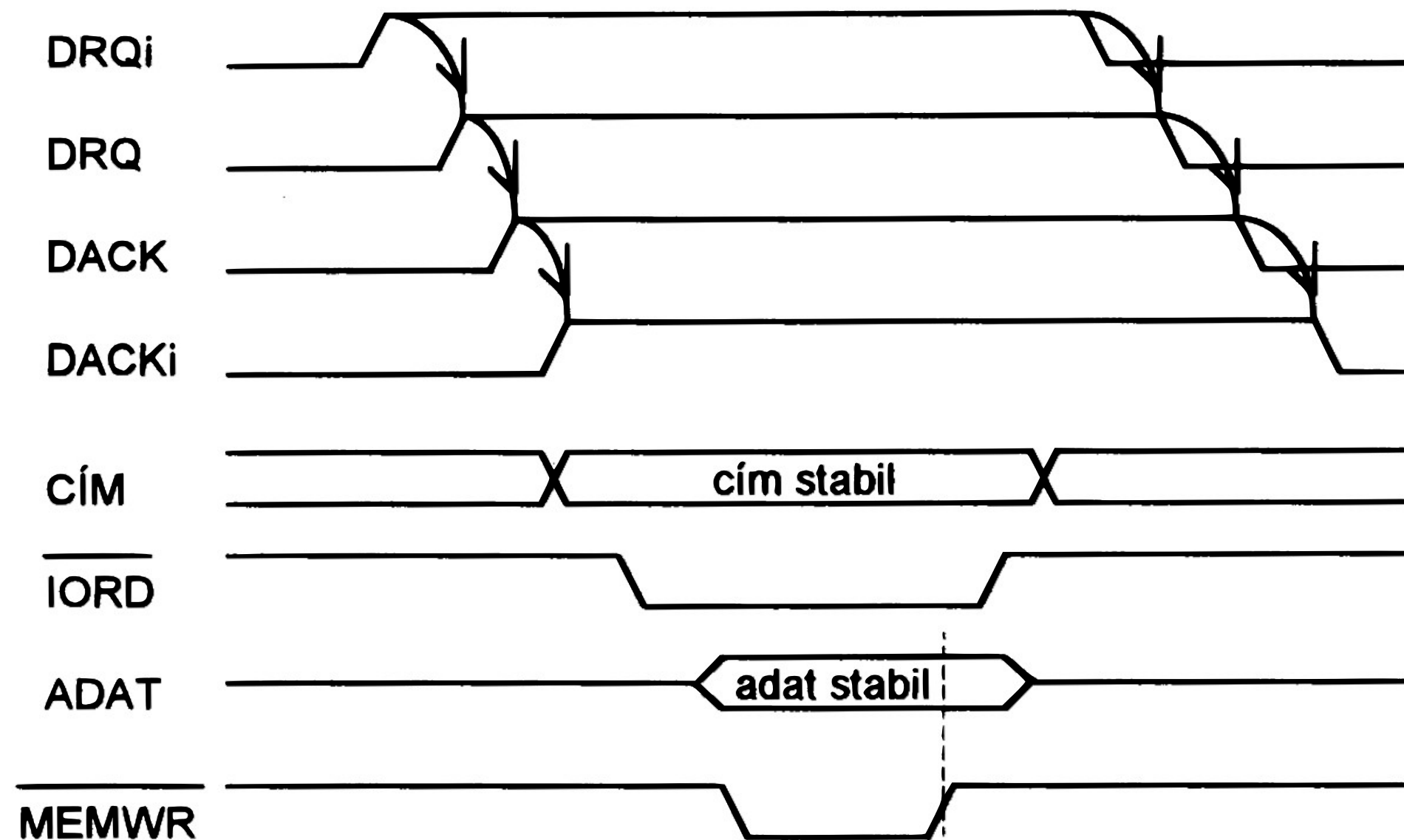
### Átvitel végének jelzése

DMA vezérlő vagy a periféria általában megszakítással jelzi a CPU-nak

# Közvetlen memória hozzáférés (DMA) Egyciklusú átvitel



# Közvetlen memória hozzáférés (DMA) Egyciklusú átvitel



# Közvetlen memória hozzáférés (DMA) Egyciklusú átvitel

## Üzem módok

Egyenkénti (single)

minden egyes szó átviteléhez külön DMA kérés

folyamatos DMA kérés esetén ciklusonként egy processzor ciklus beiktatása

Blokkos

Egykérés = egy adatblokk átvitele

Igény szerinti (demand)

A kérés végéig fennáll a DMA átvitel

## Nem DMA-s perifériák tiltása

Hibás működés (olvasási vagy írási) elkerülése érdekében

# Közvetlen memória hozzáférés (DMA)

## Kétciklusú átvitel

Első ciklusban megcímzi és beolvassa a forrásról a pufferbe az adato(ka)t

Második ciklusban kiírja a céleszközbe a puffer tartalmát

Előnye:

Nincs különbség a DMA-s és nem DMA-s perifériák engedélyezésében

Hátránya:

Lassabb (kb. fele akkora) sebesség, mint az egyciklusú átvitel során

# Példa DMA vezérlő illesztésére

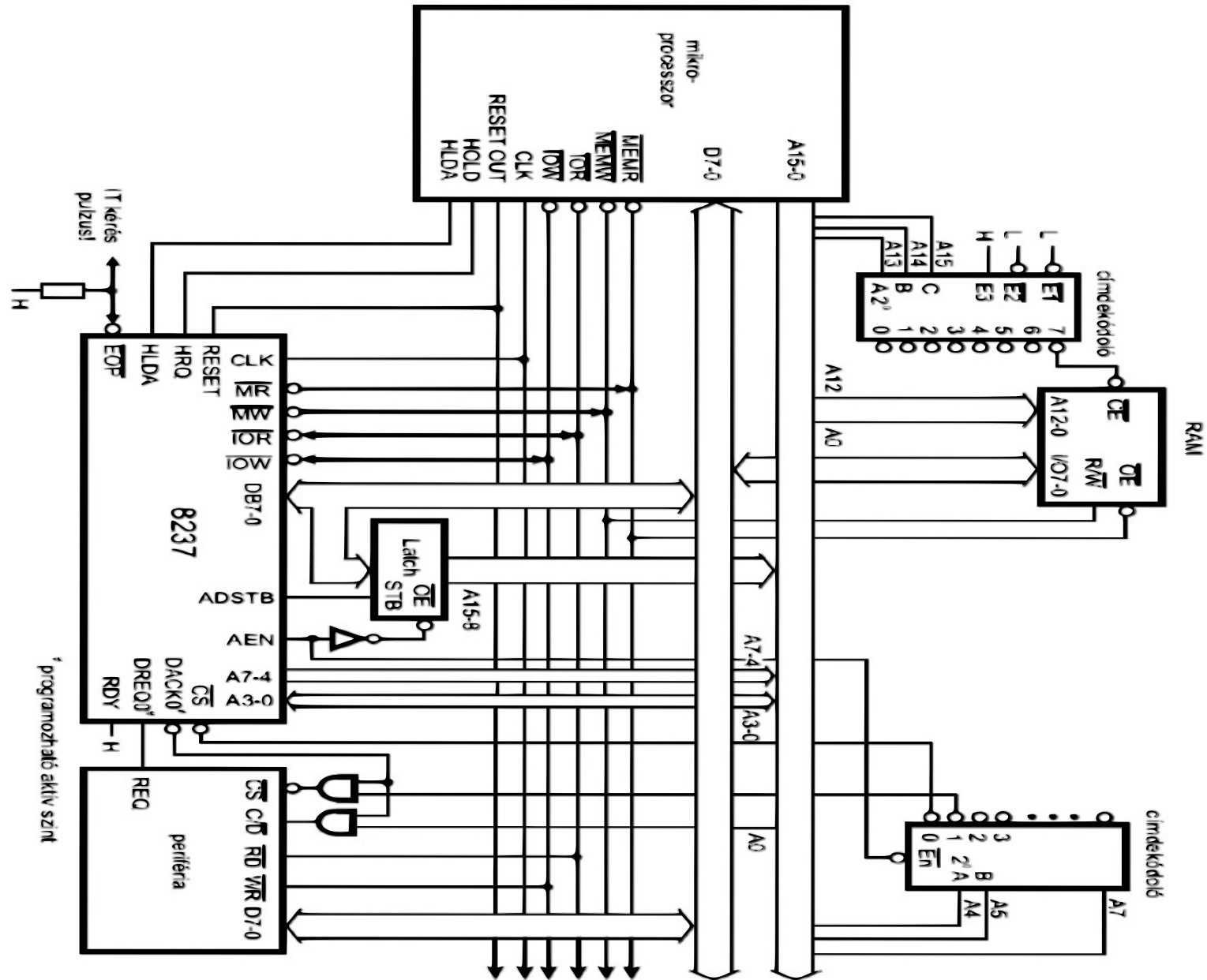
## Feladat

Adott egy olyan input periféria, amely két I/O portot foglal el (A0 címbitet kell belekötni). A0=1 esetén a parancs regiszterét érjük el, ide két egymást követő out utasítással (alsó bájt, felső bájt) kell beleírni, hogy mennyi adatot kérünk tőle. Ha az adatok készen állnak, ezt a REQ lábán 0→1 átmenettel jelzi. Ezután a megadott számú adat blokkosan (egymás után várakozás nélkül) kiolvasható az adat regiszteréből (A0=0).

A beolvasásra használjunk 8237 DMA vezérlőt!

Illesszünk még a rendszerbe 8kB EPROM-ot a 0000h címre és 8kB RAM-ot az E000h címre!

# Példa DMA vezérlő illesztésére



# További gyakorló feladatok

## 1. feladat:

i8086 processzorhoz illesszünk a 0000h címtől 8kB ROM-ot, a 2000h címtől pedig 2x8kB RAM-ot! Rajzoljunk memóriatérképet és készítsük el a blokkvázlatot!

## 2. feladat:

i8086 processzorhoz illesszünk a 0000h címtől 16kB ROM-ot, a 8000h címtől pedig 3x8kB RAM-ot. Rajzoljunk memóriatérképet és készítsük el a blokkvázlatot!

## 3. feladat:

Illesszünk az i8086 processzorhoz a 0000h címre 16kB EPROM-ot, a 4000h címre 16kB RAM-ot, a 8000h címre pedig 8x 32kB RAM-ot úgy, hogy a 00h címre beírt 0-7 közötti értékkel lehessen közülük a megfelelő sorszámút engedélyezni. (Az most nem követelmény, az a 00h címről az aktuális érték visszaolvasható is legyen.)