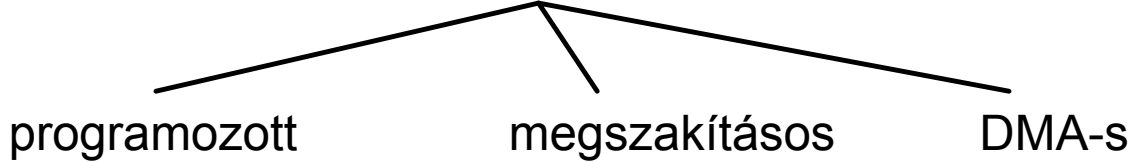


## Periféria kezelési módszerek



- közvetlen szoftver  
ütemezés

- lekérdezéses  
ütemezés

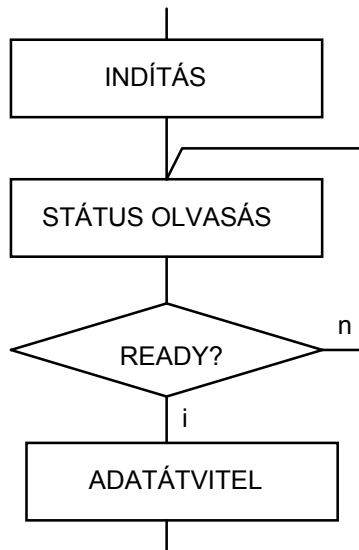
### **Programozott periféria kezelés**

#### **Közvetlen szoftver ütemezés**

Egyes perifériáknál *nincs szükség szinkronizálásra* mert vagy nincs mihez szinkronizálódni (pl: egy kijelzőt meghajtó regiszter, egyszerű bementi port), vagy a periféria egy művelet elindítása után a következő utasítás végrehajtására már biztosan elkészül (gyors A/D konverter). Az ilyen *perifériákhoz a processzor tetszőleges időpontban fordulhat.*

## Lekérdezéses ütemezés

Ha a periféria sebessége olyan, hogy *egy perifériás művelet néhány utasítás végrehajtási időn belül fejeződik be*, akkor a szinkronizálást státus figyeléssel célszerű megoldani.



*Lassú periféria esetében sok várakozással jár, ami alatt a CPU nem végez érdemleges működést. Ezért csak akkor engedhető meg, ha CPU egyéb feladatait így is képes elvégezni.*

Az igényelt hardver tekintetében általában a *lekérdezéses ütemezés a legolcsóbb megoldás.*

## Program megszakításos periféria kezelés

*Ha a periféria csak viszonylag hosszú idő vagy előre kiszámíthatatlan idő múlva lesz kész az újabb adatátvitelre, a lekérdezéses ütemezés feleslegesen foglalja a processzor idejét. Ezért ilyenkor **interruptos periféria kezelést célszerű alkalmazni.***

*A perifériához fordulást maga a periféria **kezdeményezi** egy logikai jellel a CPU vagy IT vezérlő interrupt kérő bemenetén. Az interruptot hardvertől függően az IT vonal szintje (**szintérzékeny IT**), vagy a jel valamely éle (**élérzékeny IT**) váltja ki.*

Az IT kezdeményezés hatására valamilyen mechanizmussal előbb-utóbb egy **speciális szubrutinra (interrupt rutin) adódik a vezérlés, ami elvégzi a periféria kezelését.** Mivel a CPU és a periféria szinkronizálása automatikus, **nem kell feleslegesen státusz figyeléssel tölni az időt.**

Az interruptos kezelés **a program áttekinthetőségét is növeli.**

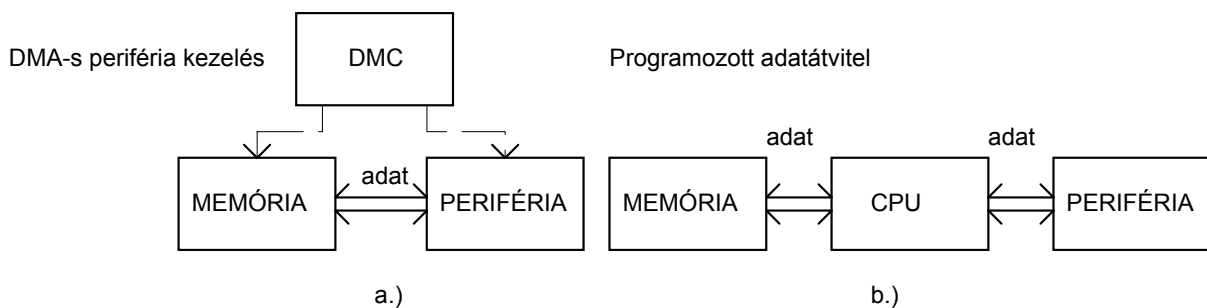
Mivel az IT rutinban nem a periféria kezelését szolgáló, ugyanakkor mindig végrehajtandó feladatok is vannak (regiszter mentés, regiszter visszaállítás, stack kezelés), **túl sűrűn érkező IT-k szintén leronthatják a CPU kihasználtságát.**

IT-s periféria kezelést akkor célszerű alkalmazni, ha:

- *a CPU sebessége elegendő a periféria kezelésére és*
- *az interruptok várhatóan nem túl sűrűn jönnek,* közöttük a CPU elég sok utasítást képes végrehajtani, vagy
- ha *az interrupt bekövetkezésének időpontja véletlenszerű,* de ha bekövetkezik, akkor gyorsan ki kell szolgálni,
- *az interrupt alatt elvégzendő feladatok nem vesznek túl sok időt igénybe,* vagy ha igen, akkor azt egyéb IT is megszakíthatja (többszintű IT rendszer) .
- *az interruptos szervezés a program strukturáltságát, áttekinthetőségét szolgálja.*

## *A közvetlen memória hozzáférés (DMA)*

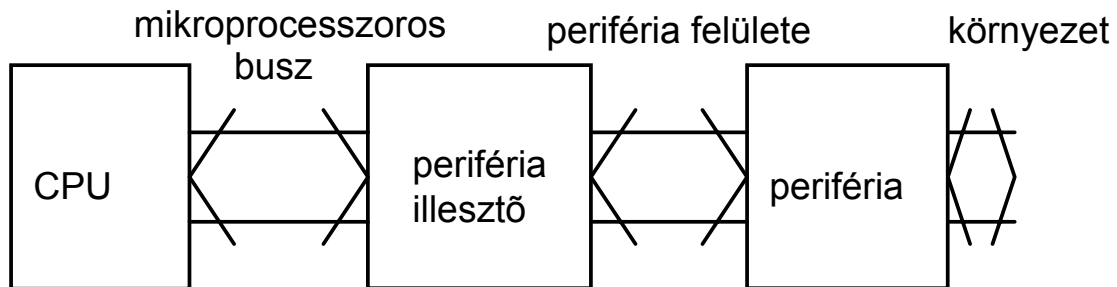
- Ha *a processzor sebessége nem elegendő az adatátvitel lebonyolítására* a periféria és memória között, DMA-s kezelést kell alkalmazni.
- Az ún. DMA vezérlő segítségével *a processzort kikerülve, közvetlen adatátvitel lehetséges a memória és a periféria között*. Ezt nevezik közvetlen memória hozzáférésnek (Direct Memory Access).
- *A DMA átvitel általában gyorsabb, mintha a CPU végezné* (nem kell közben a memóriából utasításokat olvasni és nem kell az adatot a CPU-n keresztül áramoltatni).
- A DMA-s kezelést *tipikusan nagy sebességű, blokkos adatátvitelt igénylő perifériáknál alkalmazzák* (winchester, nagyssebességű soros adatátvitel stb).



## Periféria illesztés

A *perifériák* biztosítják a processzor és a környezet közötti kapcsolatot.

A *periféria illesztő* áramkör a periféria kommunikációs felületét illeszti a mikroprocesszor buszára

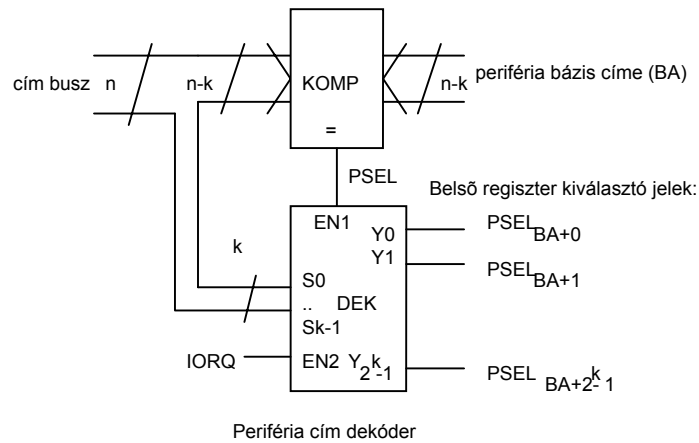


### Periféria regiszterek típusai:

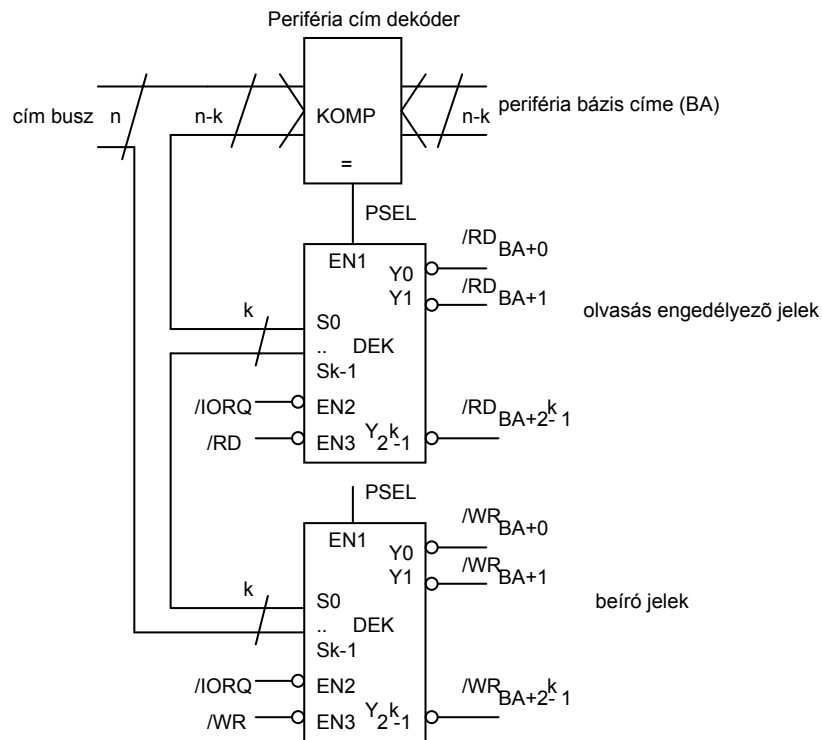
- *üzemmód regiszter* (a periféria használati jellemzőinek (pl. soros vonal sebessége) beállítása)
- *parancs regiszter* (utasítás a perifériának, pl. A/D konverter indítása)
- *státus regiszter* (a periféria állapota pl. A/D konverter kész)
- *adat regiszter* (adat a perifériának vagy a perifériától pl. A/D konvertertől jövő adat, a bementi feszültséggel arányos bináris szám)

# A periféria címdekóder felépítése

A periféria kijelölése (PSEL) és a periférián belüli címek ( $PSEL_{BA+i}$ ) kijelölésének elve:

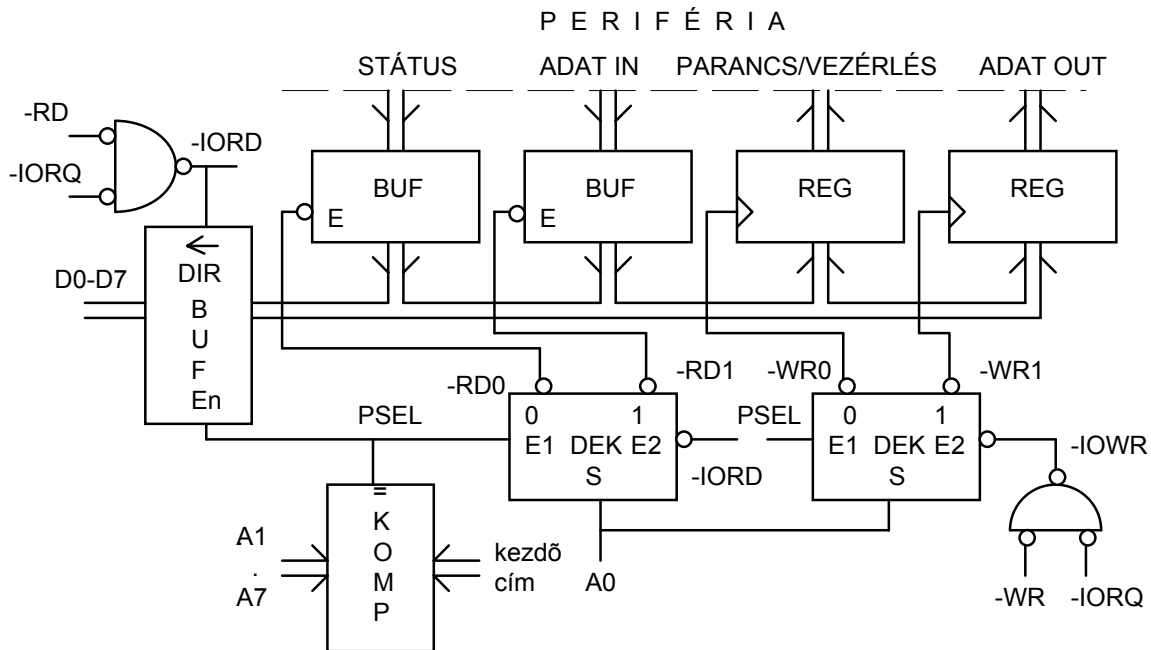


Olvasás engedélyező és beíró impulzusok előállításása:



## Tipikus periféria illesztő felépítése

2 olvasható és 2 írható regiszter, adat busz leválasztással (nem IT-os)

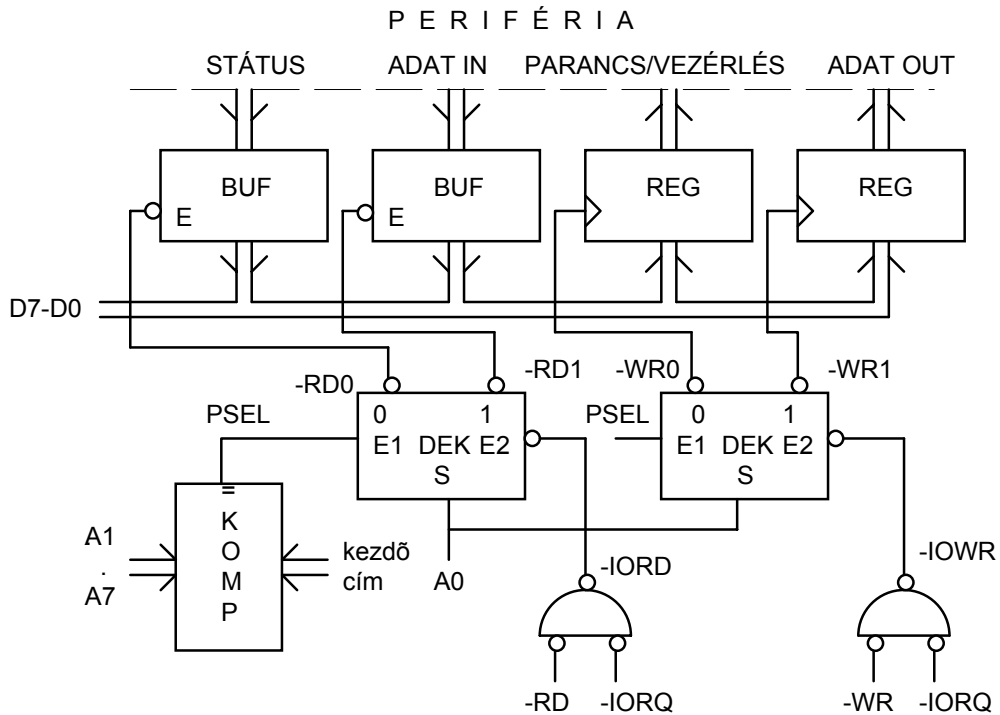


## Programozási felület

regiszter címe	funkciója	az egyes bitek szerepe	írható (W), olvasható (R), vagy mindkettő (R/W)
kezdő cím	parancs regiszter	x	W
kezdő cím	státus regiszter	x	R
kezdő cím+1	adat regiszter	x	R
kezdő cím+1	adat regiszter	x	W



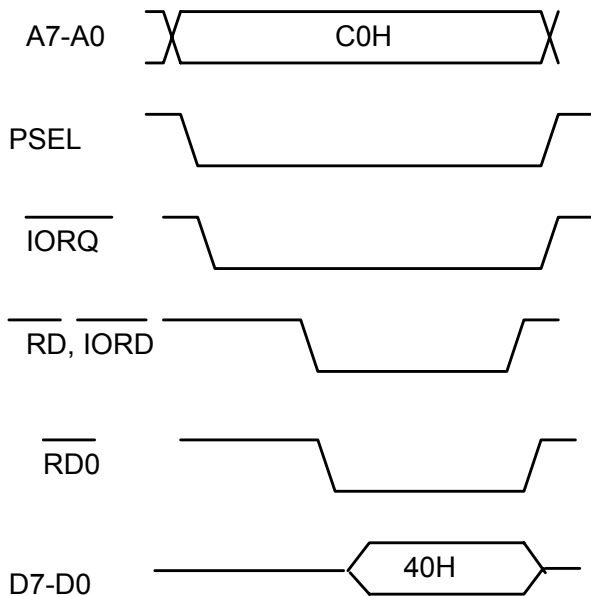
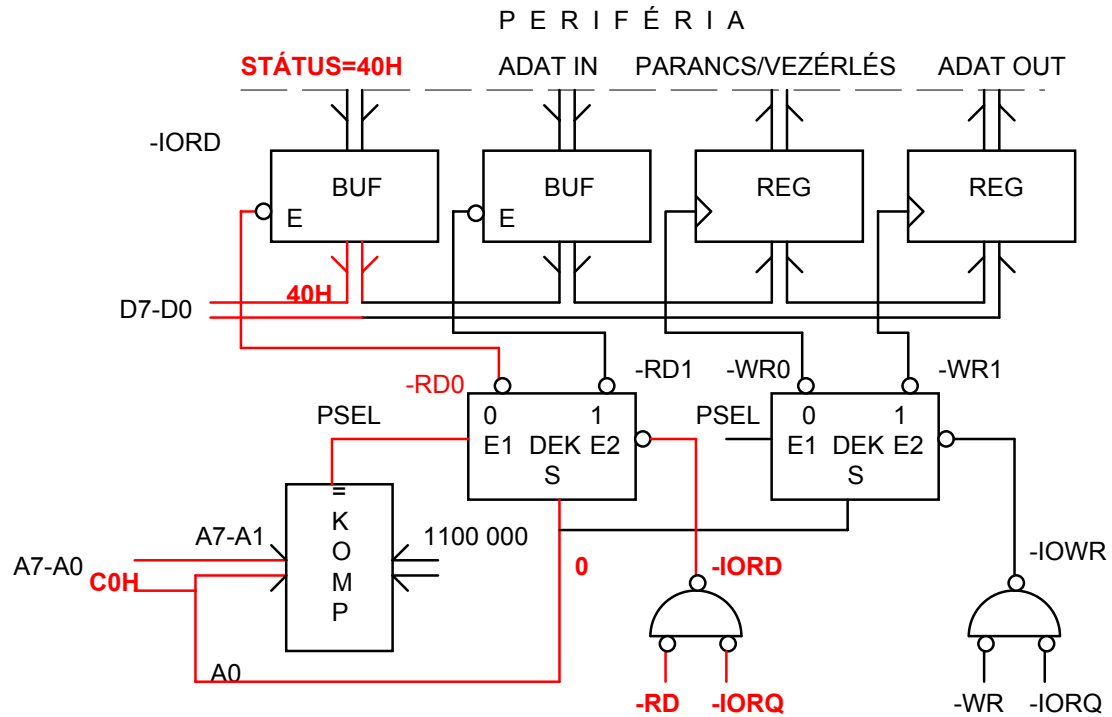
## 2 olvasható és 2 írható regiszter, adat busz leválasztás nélkül



# Olvasás a státus regiszterből

## A periféria bázis címe C0H, státus reg:C0H

### IN A,(C0H) utasítás hatása

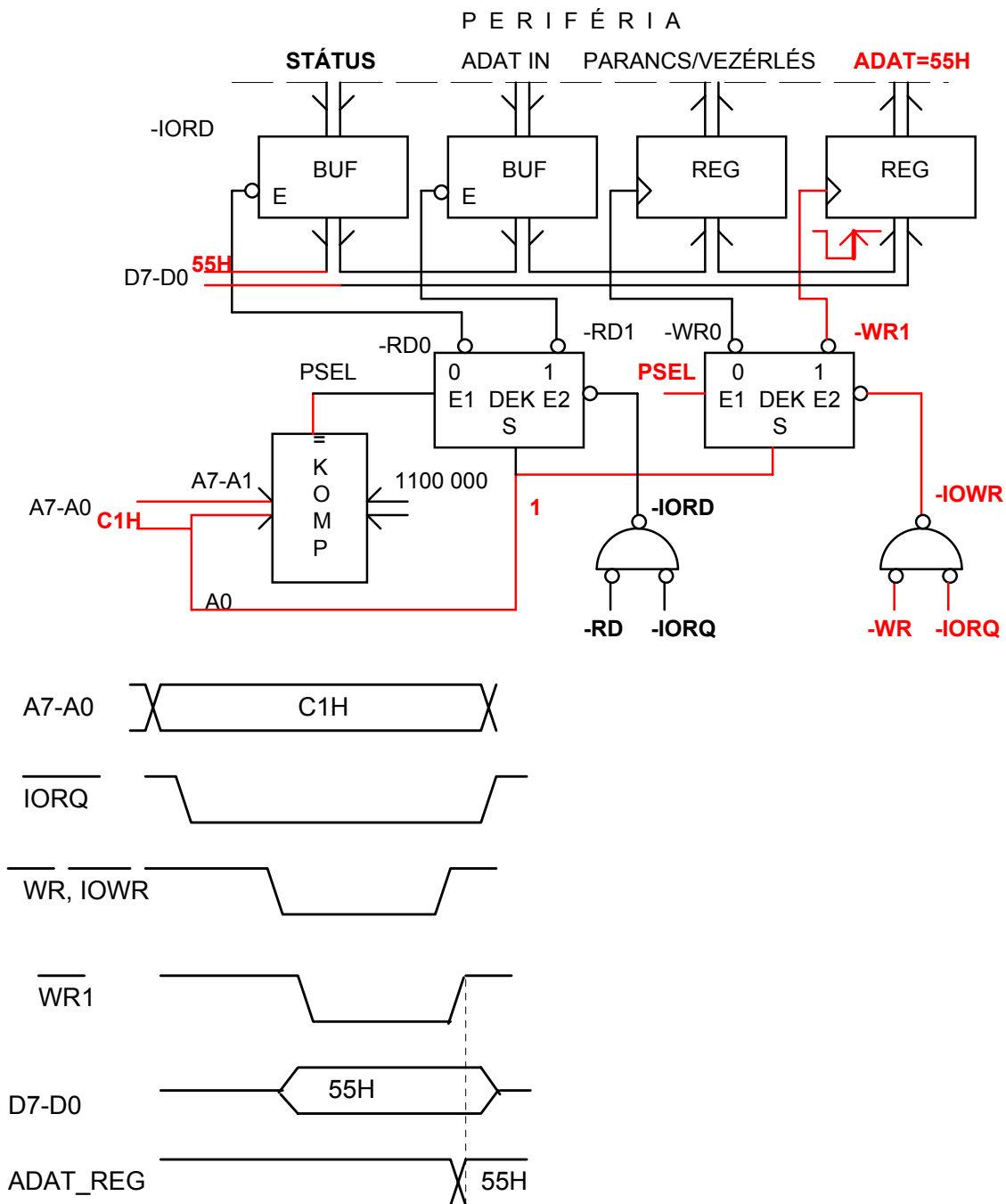


# Írás az adat regiszterbe

A periféria bázis címe C0H, adat reg: C1H

LD A,#55H

OUT (C1H,A)



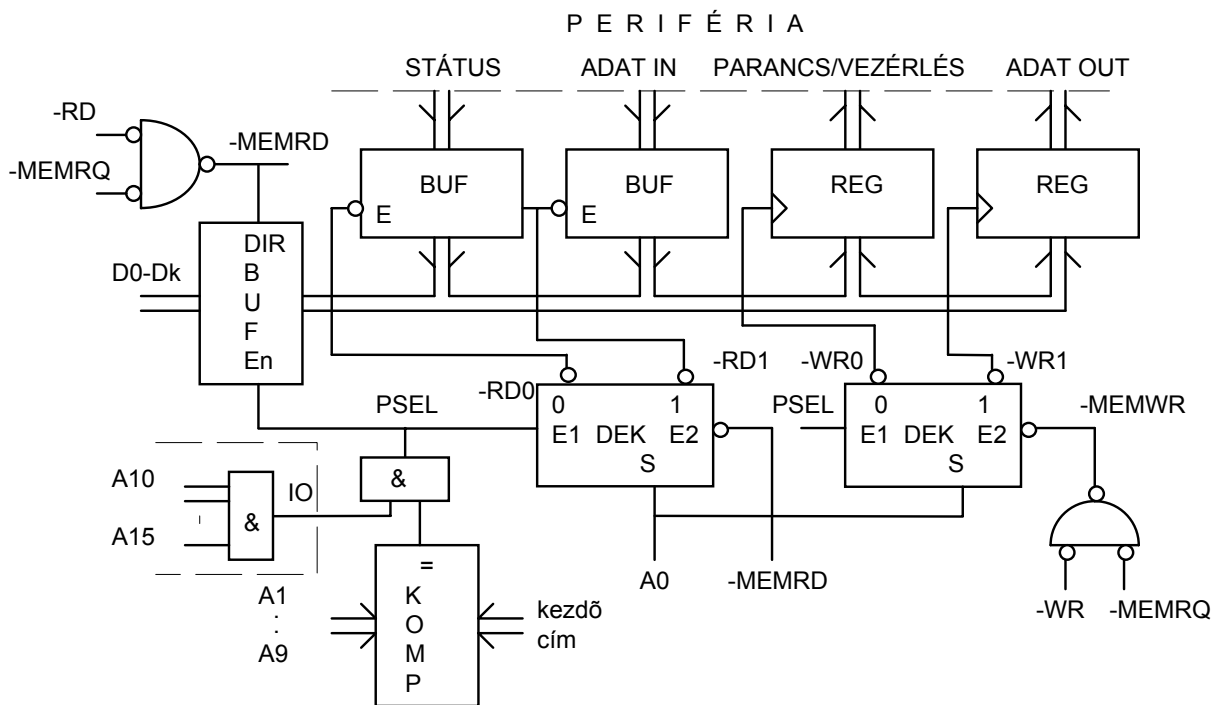
Programozott kezelés (program részlet Z80 utasításokkal):  
A periféria parancs regiszterébe 1-et kell írni az indításhoz.  
A státus regiszter 7. bitjén jelzi, hogy kész van.

```
BASE      equ C0H
PAR_REG   equ BASE
STAT_REG  equ BASE
DATA_IN   equ BASE+1
START     equ 01H
READY     equ 80H
```

```
...
LD        A,#START
OUT       (PAR_REG),A    ;indítás
loop:    IN        A,(STAT_REG)
AND       A,#READY     ;periféria kész figyelés
JZ        loop
IN        A,(DATA_IN)   ;adat beolvasás
...
```

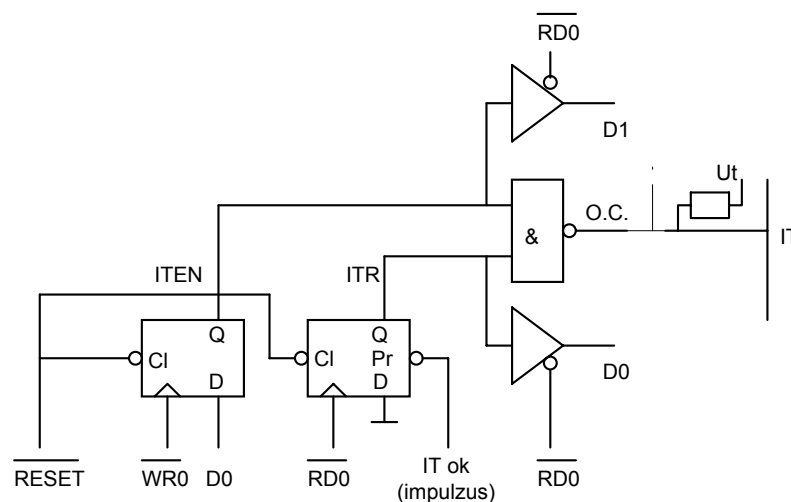
# MEMÓRIÁBA ÁGYAZOTT PERIFÉRIA ILLESZTÉS

- *A memória egy területét IO területnek tartjuk fenn*, erre a címtartományra nem szabad memóriát illeszteni.
- *A periféria memória referenciás utasításokkal érhető el.*



## Interruptos periféria IT kérő logikája

- Ebben a megvalósításban perifériában az eseményre egy impulzust kell generálni.
- Az impulzus billenti be az IT kérő flip-floppot és okoz IT-ot, ha a parancs regiszter ITEN bitjével engedélyezve van.
- E megvalósításban az IT rutinban kiadott státus regiszter olvasás törli az IT kérést.
- A státus regiszterből kiolvasható az IT kérő flip-flop állapota és az ITEN bit.
- A uP IT vonalára több periféria csatlakozhat.
- A perifériáknak open collectorosan kell meghajtani az IT vonalat.
- Az IT kérést kiszolgálásig fenn kell tartatni.



Adott egy mikroprocesszoros busz, melynek jelei:

D7-D0 (adat busz),

A15-A0 (címbusz),

-RD (alacsony aktív olvasás engedélyezés),

-WR (alacsony aktív beíró impulzus, az adat a jel hátsó élénél stabil),

-MREQ (memóriához fordulás jelzése, alacsony aktív, amíg a jel aktív, a memória cím stabil),

-IORQ (perifériához fordulás jelzése, alacsony aktív, amíg a jel aktív, a periféria cím stabil),

-IRQ, alacsony aktív interrupt kérő jel,

-RESET, alacsony aktív alaphelyzetbe hozó jel,

SYSCLK (rendszer órajel).

A processzor 256 IO címmel rendelkezik (A7-A0).

Illesszen a fenti buszra egy **szinkron soros adó egységet**. Az egység az ADAT regiszterébe írt adatot **az írást követően** (annak hatására) sorosan kiadja az SOUT kimenetén egy SCLK órajellel együtt, amelynek felfutó élénél változik az adat. Az adás előtt és után az SCLK órajel 0 értékű, az SOUT kimenet pedig tetszőleges. A periféria a teljes 8 bit kiküldése után jelez a státuszregiszter RDY bitjén és IT-ot kér, ha engedélyezett (parancsregiszter ITEN bitje). (15p)

Az alábbiakban részekre bontottuk a feladatot. Az összes részfeladat megoldása után áll elő a teljes feladatot megvalósító interfész. *Az egyes részekenél azok részletes funkcionális blokkvázlatát kérjük, egymástól jól elhatároltan lerajzolni!*

**a.** Rajzolja le a periféria cím és utasítás dekóderét, továbbá a parancsregisztert (ITEN), az adatregisztert, a státuszregisztert és a beírásukat/kiolvasásukat lehetővé tevő logikát! Az adatregiszter az A0H címen, a parancsregiszter az A1H, a státuszregiszter az A2H címen legyen. A parancsregiszter bitkiosztása: ITEN: D0. Az állapot regiszterből legyen kiolvasható az RDY bit (D1 bit) és az ITRQ = (ITEN & RDY) bit (D0 bit) értéke! (6p)

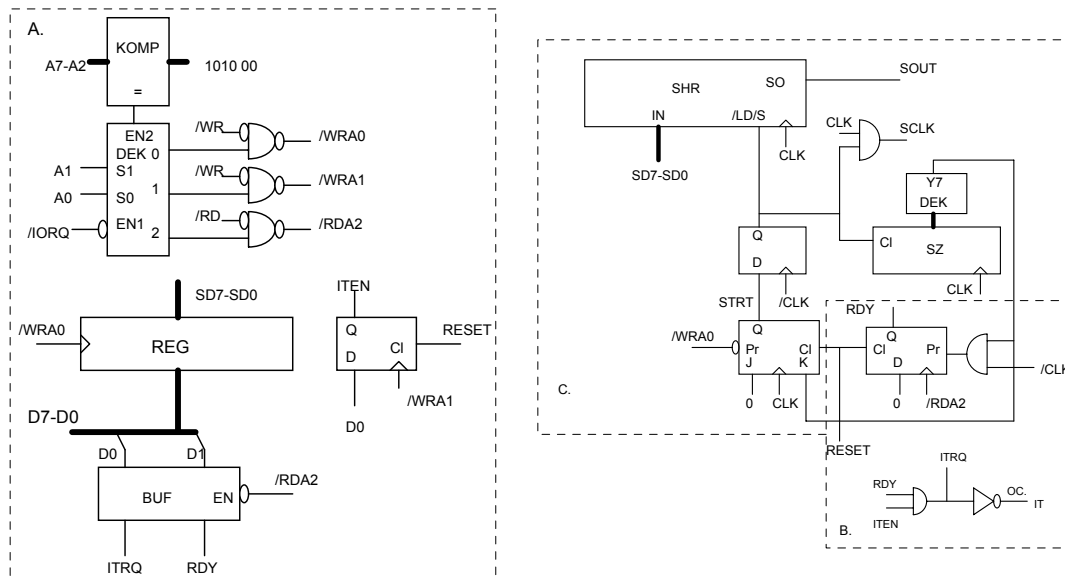
**b.** Rajzolja le a teljes interrupt kérő logikát, mely az RDY bebillenésekor interruptot kér! Az interrupt kérés programból legyen tiltható (ITEN bit) és RESET hatására is tiltódjon le. Az interrupt kérő flip-flop (RDY bit) törlődjön a státuszregiszter olvasása *után* és RESET hatására. (4p)

**c.** Rajzolja le a fent specifikált működést megvalósító soros adó egységet! (4p)

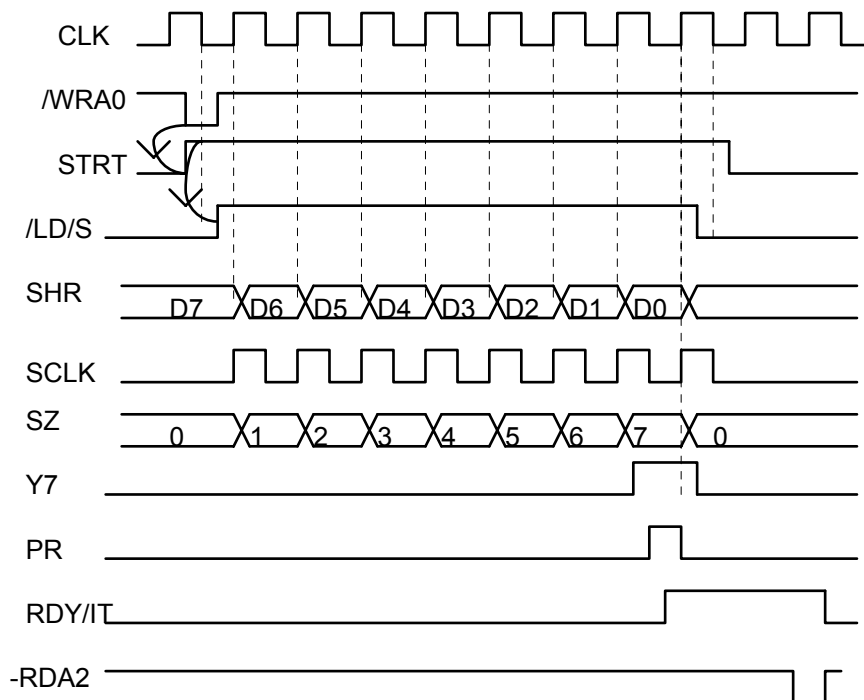
**d.** Adja meg az egység programozási felületét, az alább megadott táblázatnak megfelelő formában! (1p)

regiszter címe (hexa)	funkciója	bitek szerepe D7 D6 D5 D4 D3 D2 D1 D0	típusa R (olvasható), W (írh.), R/W (mindkettő)
--------------------------	-----------	--	--

# Részletes funkcionális blokkvázlat



## Működést bemutató idődiagram:



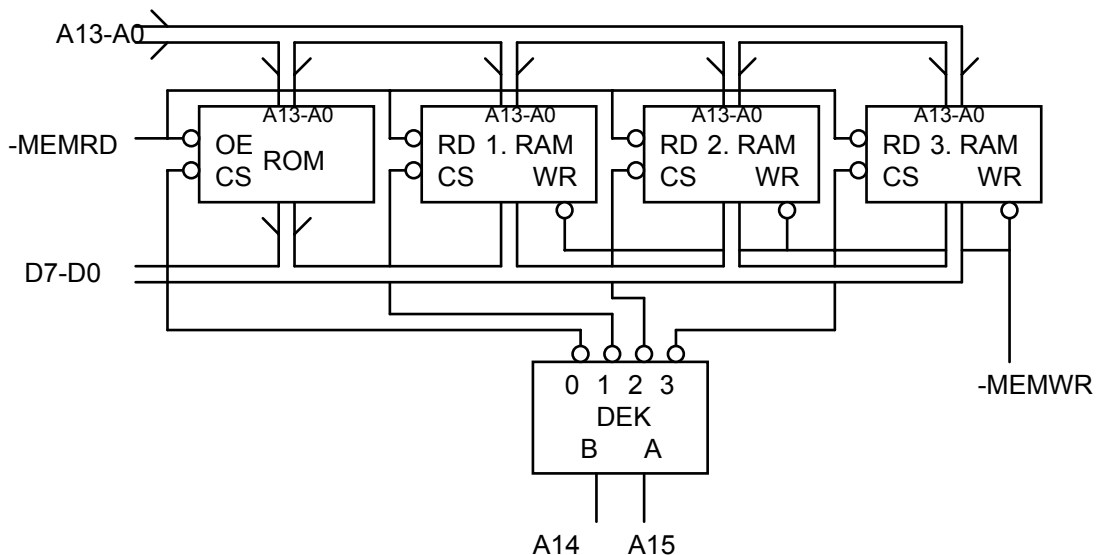


## Memória illesztés

- *Memória térkép*en reprezentáljuk a memóriák elhelyezkedését a címtartományban:

	16kbyte ROM	16kbyte RAM	16kbyte RAM	16kbyte RAM
A15	0	0	1	1
A14	0	1	0	1

- A *legfelső címbitek* alapján állítjuk elő az egyes memória chipek engedélyező jelét (*-CS*).
- *Egyszerre csak egy memória kaphat engedélyezést.*
- Az *alsó címbitek jelölik ki* a kiválasztott memória egy *rekeszét*.

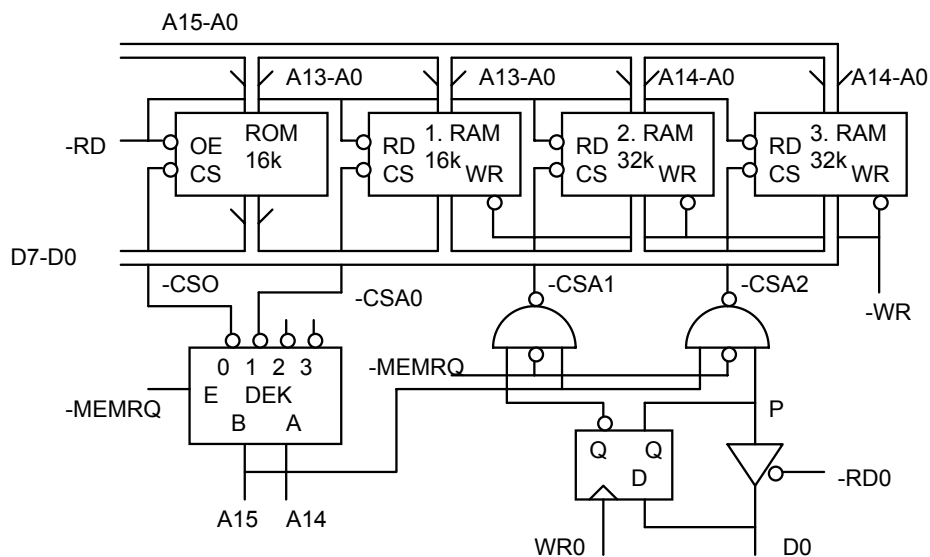


## Egyszerű lapozásos memória illesztése (tömbkapcsolás)

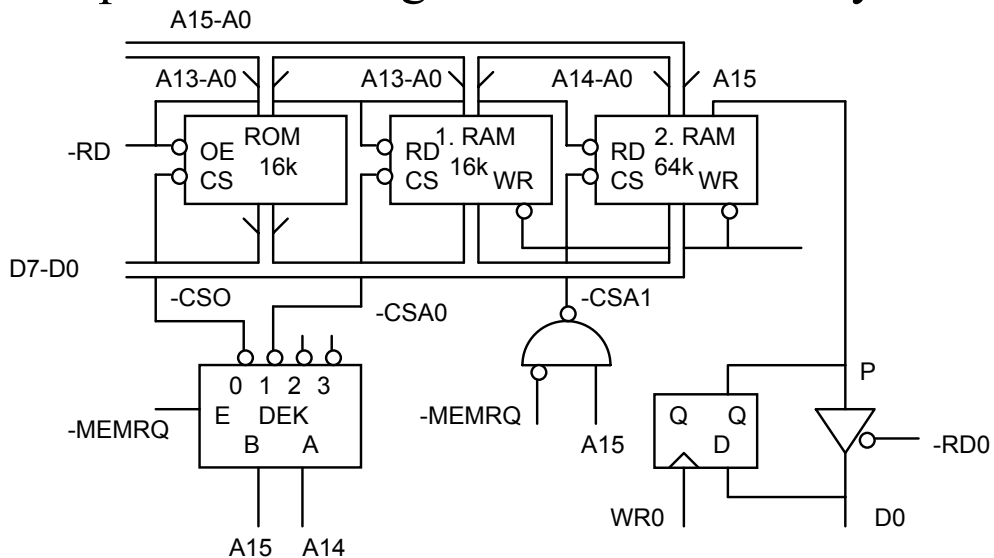
16kbyte ROM	16kbyte RAM	32 kbyte RAM	0. lap	P=0
		32 kbyte RAM	1. lap	P=1

A15	0	0	1
A14	0	1	x

### A lapozós rész megvalósítása 2db 32kbyte-os chippel

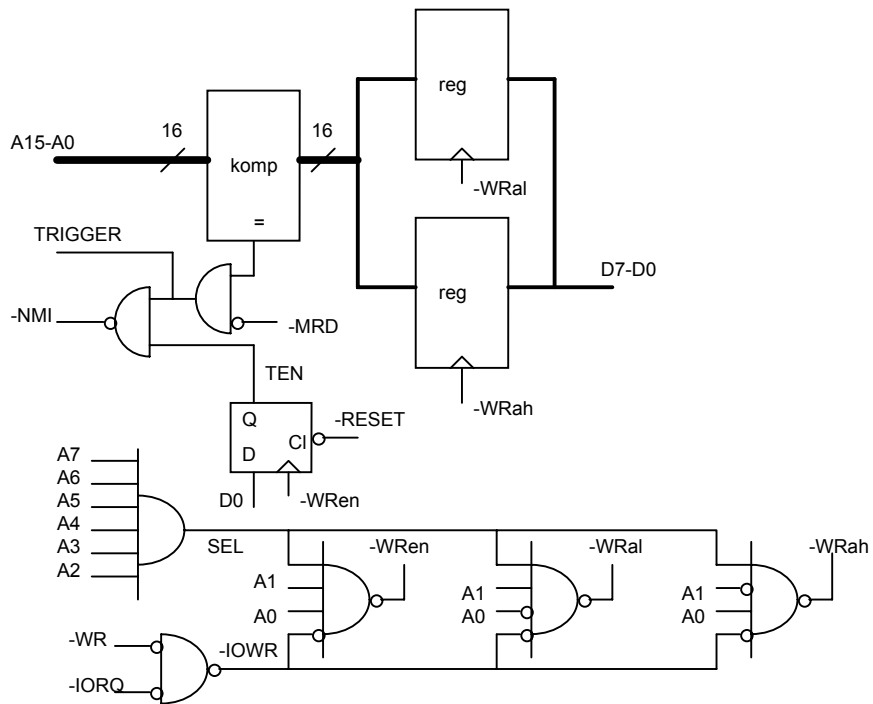


### A lapozós rész megvalósítása 1db 64kbyte-os chippel



# TÖRÉSPONTI EGYSÉG

Megadható címről olvasáskor NMI-t kér és trigger jelet generál



## SOROS A/D + FESZÜLTSG HATÁR FIGYELŐ

Az A/D STRT bemenetén levő jel felfutó éle indítja az átalakítást, majd az ezt követő 8 órajel alatt ( $f=10\text{kHz}$ ) az SD kimenetén jelenik meg az adat bitsorosan, az órajel *lefutó* élével szinkronban, 2-es komplementes kódban. Az átalakítás végét a RDY vonal jelzi.

A feladat, az A/D felhasználásával egy programozható feszültség határ figyelő megtervezése. Az áramkör-programból való engedélyezés után (EN) - 1 msec-onként ciklikusan elindítja a konverziót. Az órajel *lefutó* élével szinkronban, 2-es komplementes kódban. Az átalakítás végét a RDY vonal jelzi. IT-t kér, ha az átalakított adat egy 8 biten programozható VL értéket meghalad. Egyszerű IT rendszert tételünk fel. Rendelkezésre áll egy 10 kHz-es órajel. A periféria kezdőcíme A0H.

