

BELÉPTETŐ RENDSZER TERVEZÉSE - hardver

Számítógép-architektúrák

1. gyakorlat

Dr. Lencse Gábor

BME Hálózati Rendszerek és Szolgáltatások Tanszék

lencse@hit.bme.hu



Kiegészítések:

Dr. Koller István

BME Hálózati Rendszerek és Szolgáltatások Tanszék

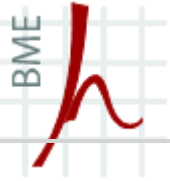
koller@hit.bme.hu

2014. szeptember 10.,
Budapest

- *A feladat megértése, rendszer specifikáció*
- A hardver specifikációja
- A tervezendő rendszer elemeinek megismerése (digitális technikából tanult potenciálisan alkalmazott elemek)
- A hardver rendszerterv lépésről lépésre
- Alternatív megoldások vizsgálata – a gyakorlati megvalósítás felé
- Összefoglalás

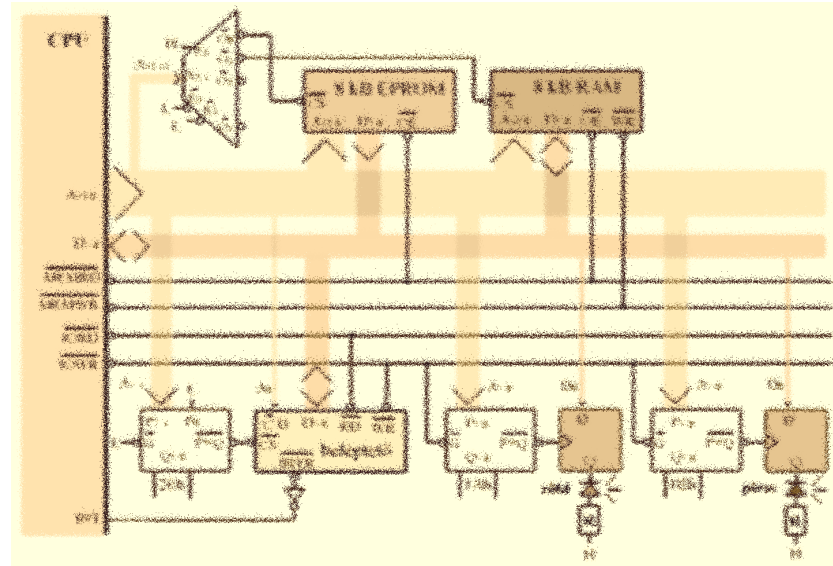


A FELADAT MEGÉRTÉSE



A feladat: elektronikus ajtózár

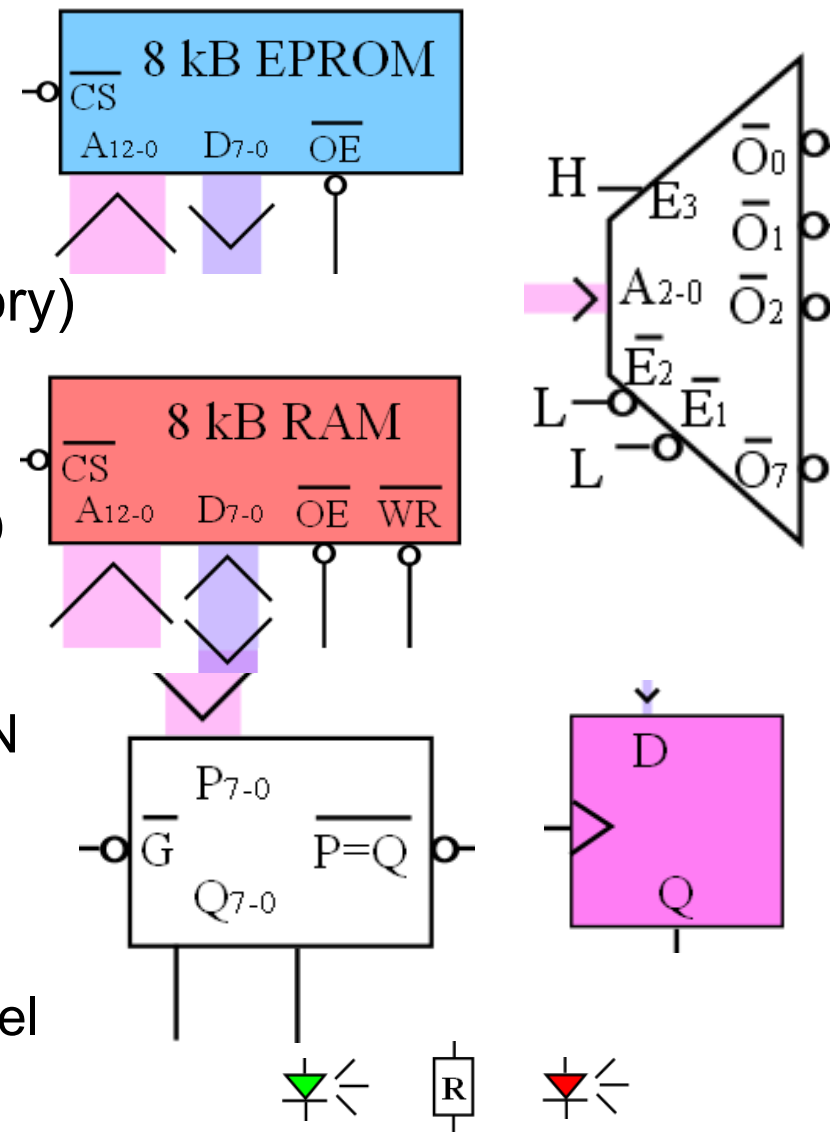
- Kulcs helyett elektronikusan olvasható kártya
- Alaphelyzetben az ajtó zárva van
- Elektronikusan vezérelhető zárszerkezet
- Nem elég a kártyát az olvasóval lehúzni, még egy kódot is ismerni kell: fokozott biztonság mechanikus kulcshoz képest
- Ha az elektronikusan olvasható kártya és a kód is megfelelő, zöld LED-et kigyújtunk, ajtózárát kinyitjuk, (majd egy idő múlva bezárjuk). Zárt állapotban piros LED világít
- Másodlagos kérdések:
 - Táplálás (elemes, hálózati)
 - Távoli felügyelet (naplózás, átkódolás)
 - Vezetékes, vagy vezeték nélküli távoli elérés



A RENDSZER ÉPÍTŐELEMEI

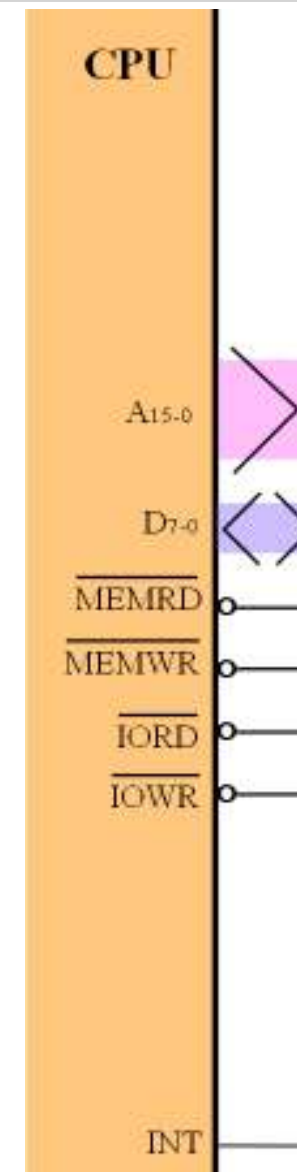
Adottak a következő építőelemek

- hipotetikus mikroprocesszor
- beléptető periféria
- 8KB ROM (Read Only Memory)
- 8KB RAM (Random Access Memory)
- 3/8-as dekóder
- 2 db D tároló
 - Az órajel felfutó élére tárolja el a D bemenet értékét.
- 3 db 8 bites címkomparátor
 - működése: IF $(/G=0) \& (P=Q)$ THEN „ $/(P=Q)$ ”=0 ELSE „ $/(P=Q)$ ”=1
- 2 db 250 Ohmos ellenállás
- egy zöld és egy piros LED
 - nyitó irányban 2V feszültségeséssel
 - 5-15 mA áramigénnyel



A mikroprocesszor illesztési felülete

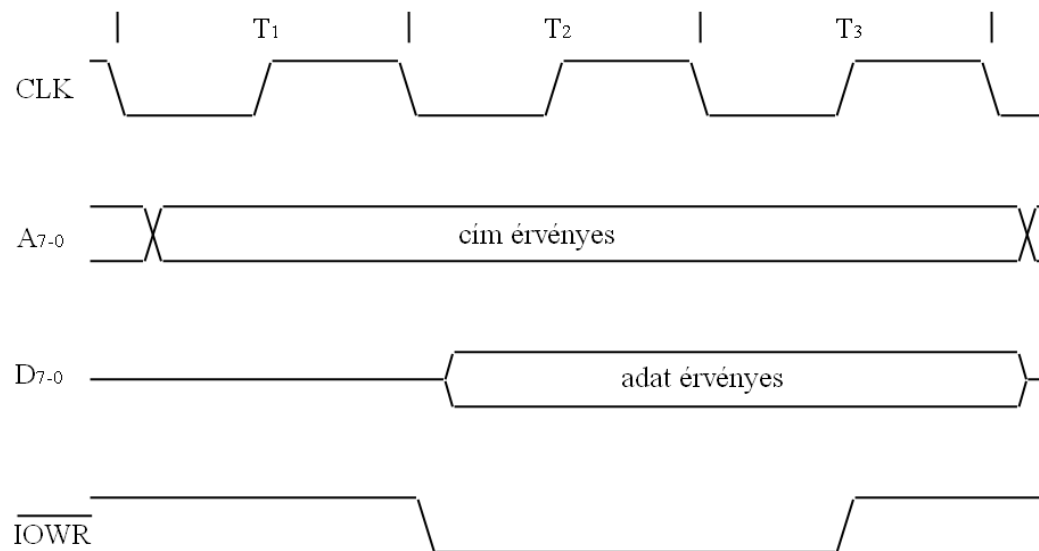
- A_{15-0} címbusz
 - megcímezhető 64 kB memória, de csak 256 I/O port (az A_{7-0} címbitekkel címezhető)
- D_{7-0} adatbusz
- $\overline{\text{MEMRD}}$, $\overline{\text{MEMWR}}$, $\overline{\text{IORD}}$, $\overline{\text{IOWR}}$
 - 0 aktív kimenő vezérlő jelek
- INT: felfutó élre érzékeny IT bemenet

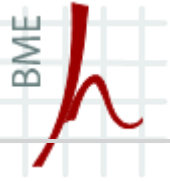


A mikroprocesszor működése

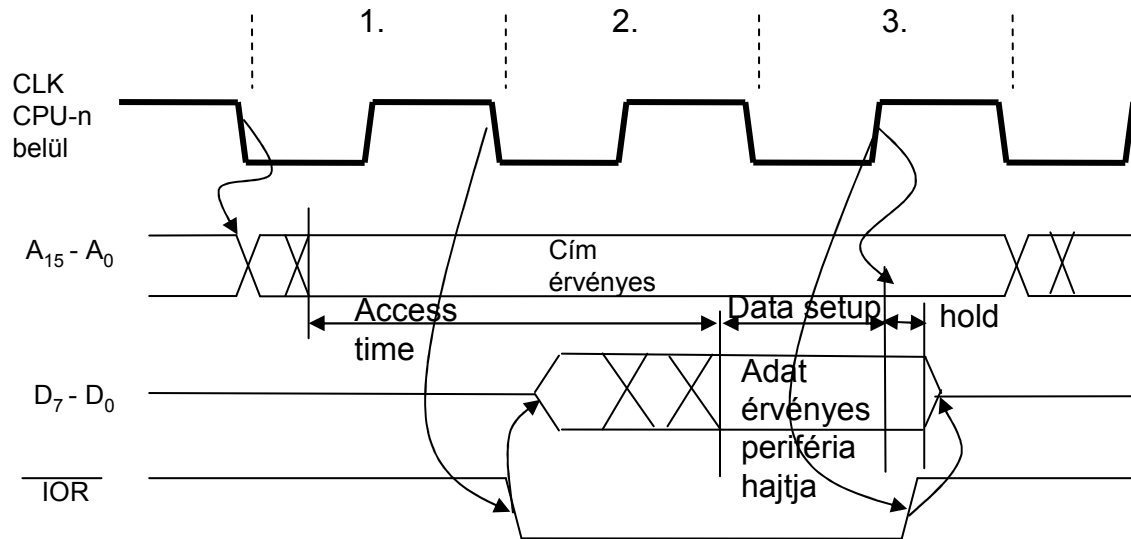
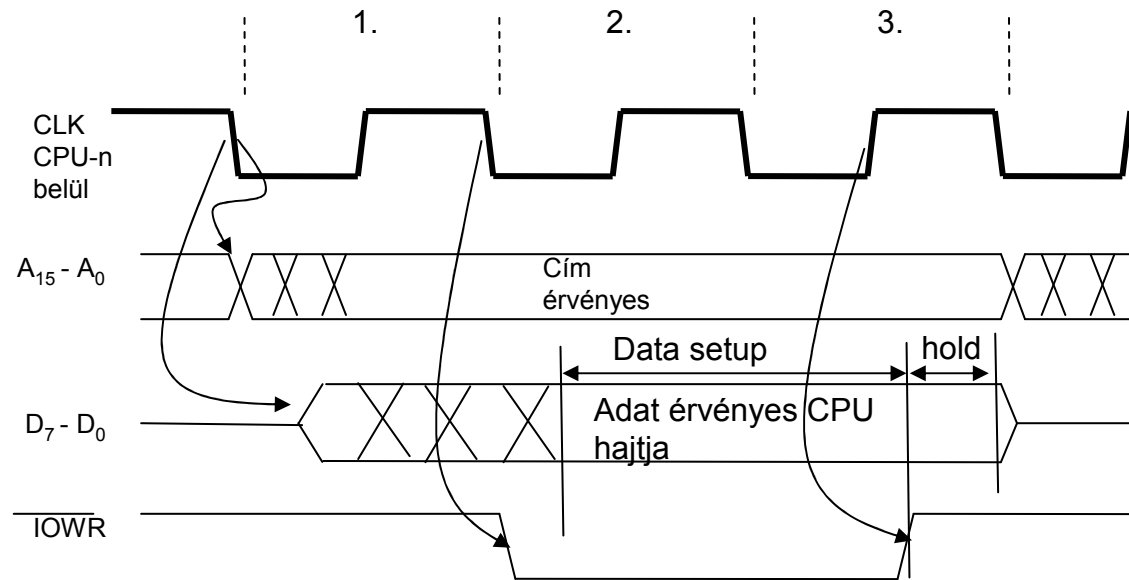
- Működés

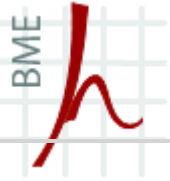
- Kezdetben a 0000h címről indul.
- Megszakításkor egy CALL 1000h utasítást hajt végre.
 - regisztereket nem ment, a további megszakításokat letiltja.
- Utasításkészletét később mutatjuk be...
- A perifériára való írás idődiagramja (az illesztéshez kell):





Write / Read



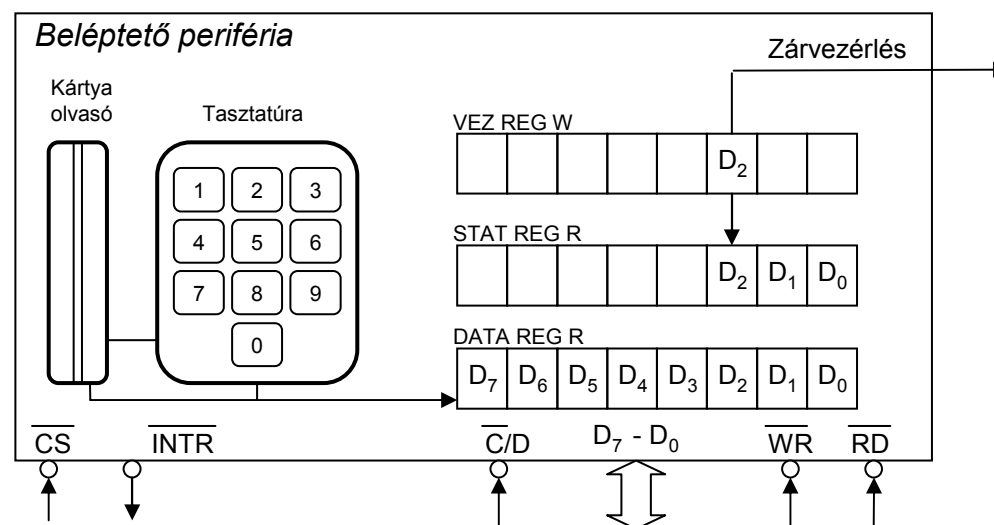
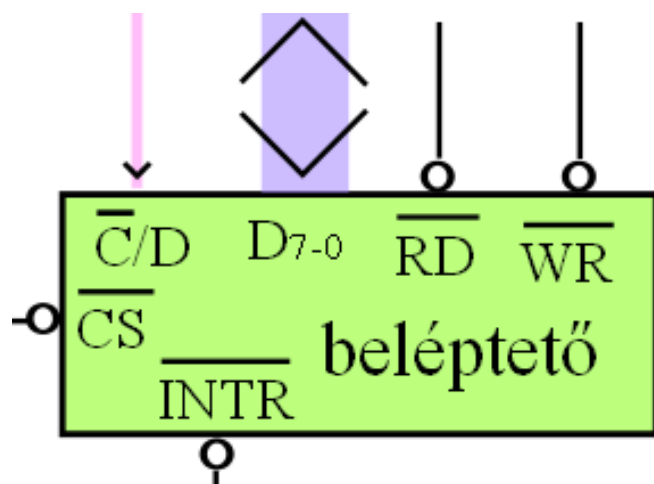


A beléptető periféria főbb részei

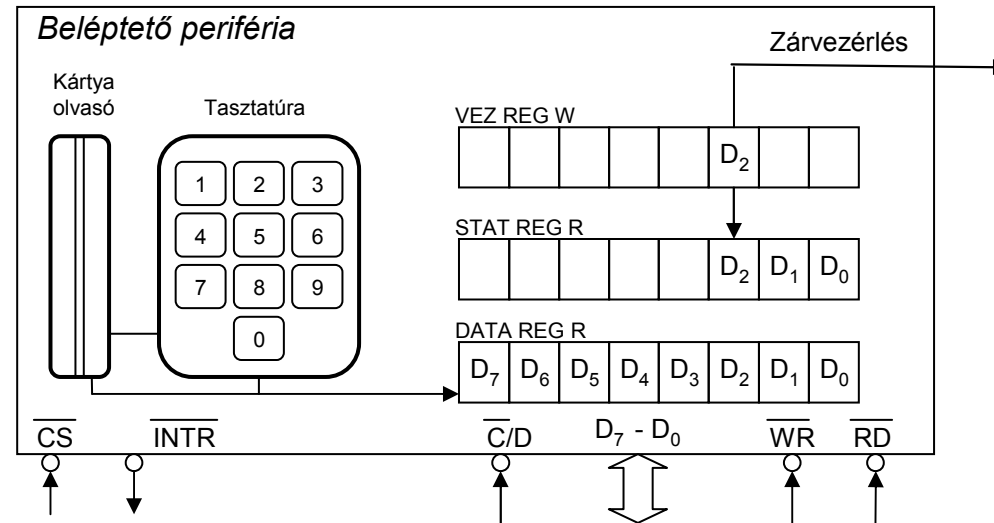
- egy kártyaolvasó
 - a belépőkártyák azonosítója 0-255 egész szám
- egy 0-9 számjegyeket tartalmazó billentyűzet
 - más gomb, pl. törlés nincs
- egy relé vezérlő egység
 - nyitott vagy zárt állásba vezérelhető a zár nyelve
 - az ajtó zárt állásban is becsukódik, de csak nyitott állásban nyitható ki
- a processzor felé való illesztési felület

A beléptető periféria illesztési felülete

- D_{7-0} : kétirányú adatbusz
- $/CS$, $/RD$ és $/WR$: 0 aktív bemenő vezérlő jelek,
- $(/C)/D$ (Command=0/Data=1): választó bemenet, hatása:
 - $(/C)/D=0$ érték esetén: D_{7-0} olvasáskor státusz, íráskor zárvezérlés
 - $(/C)/D=1$ érték esetén: az adatregiszterből az utolsó eseményhez tartozó számérték (utoljára lehúzott kártya azonosítója vagy utoljára megnyomott számjegy) olvasható ki, az adatregiszter írása hatástalan.
- $/INTR$: alacsony logikai szinttel jelző esemény kimenet

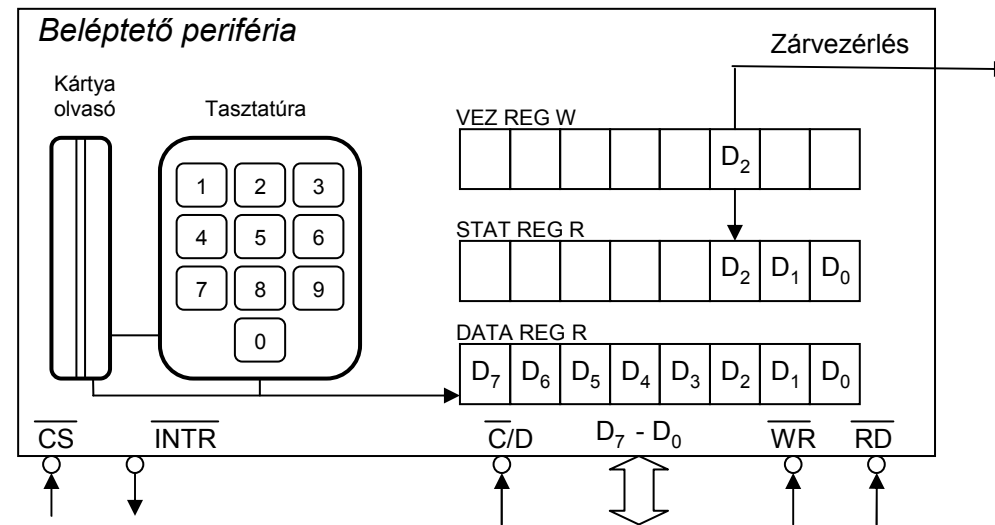


A beléptető periféria működése – 1



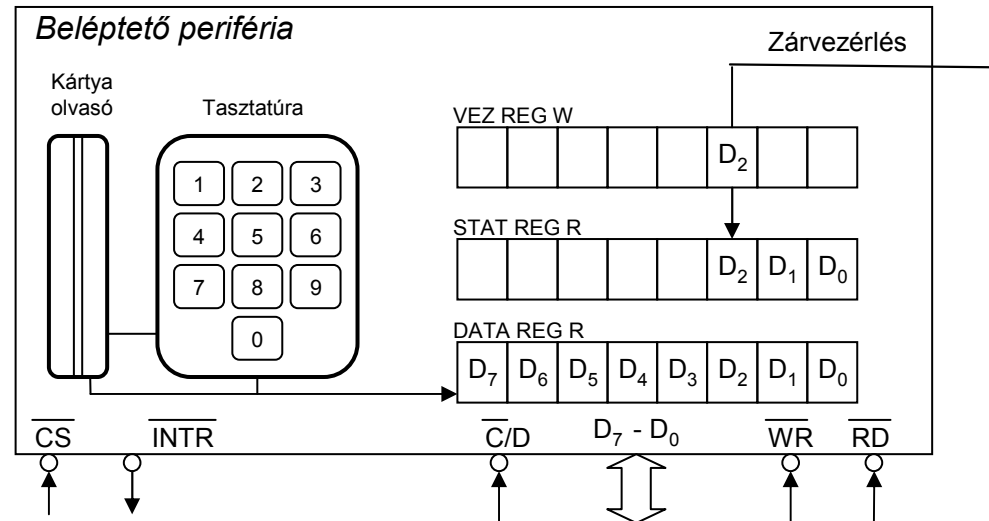
- Minden kártyához (kártyaazonosítóhoz) tartozik egy 4 decimális jegyből álló kód: a zárat akkor kell nyitni, ha a kártya lehúzása után megnyomott első 4 billentyű éppen a hozzá tartozó kódot adja. (Időzítéssel nem foglalkozunk!)

A beléptető periféria működése – 2



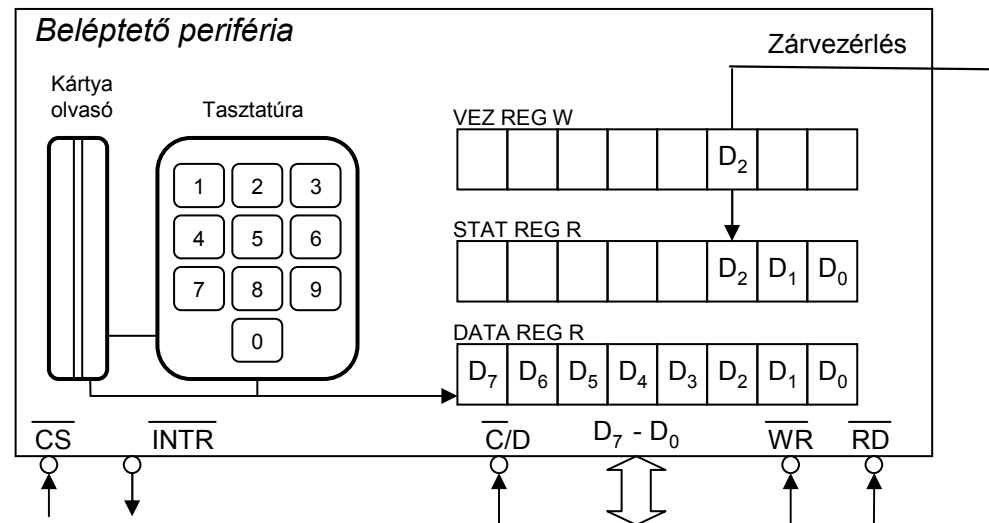
Amikor a kártyát lehúzzák, a beléptető /INTR lába alacsony szintűre vált, és a státuszregiszteréből olvasva a D0 bit 1-es értékű lesz (akárhányszor is kiolvasható): ezek mindaddig fennállnak, amíg az adatregiszterből ki nem olvasták a kártya azonosítóját. Ezután adatregiszterből kiolvasható (csak egyszer!) a lehúzott kártya azonosítója, utána rögtön /INTR magasra vált, és a státuszregiszteréből olvasva a D0 bit értéke 0 lesz.

A beléptető periféria működése – 3

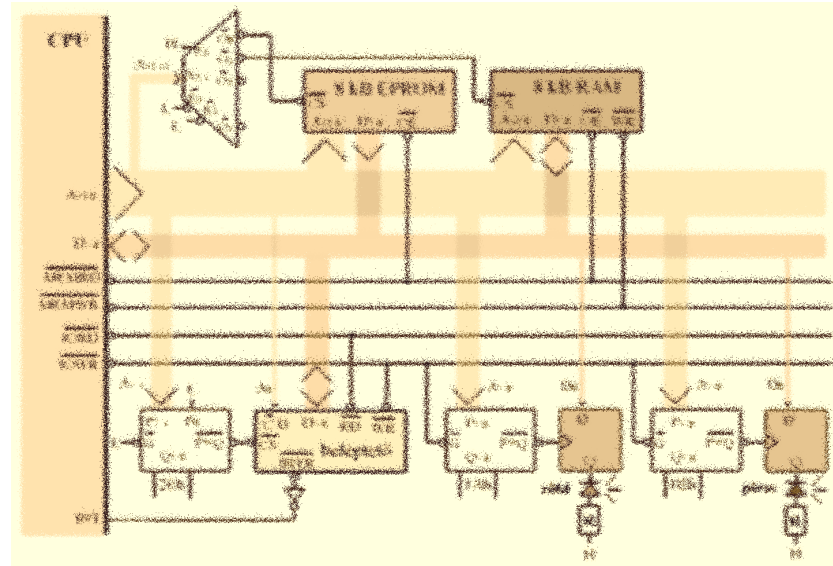


Amikor egy számjegyet beütöttek, a beléptető /INTR lába alacsony szintűre vált, és a státuszregiszteréből olvasva a D_1 bit 1-es értékű lesz (akárhányszor is kiolvasható): ezek mindaddig fennállnak, amíg az adatregiszterből ki nem olvasták a számjegy értékét. Ezután az adatregiszterből kiolvasható (csak egyszer!) a beütött számjegy értéke, utána rögtön /INTR magasra vált, és a státuszregiszteréből olvasva a D_1 bit értéke 0 lesz.

A beléptető periféria működése – 4

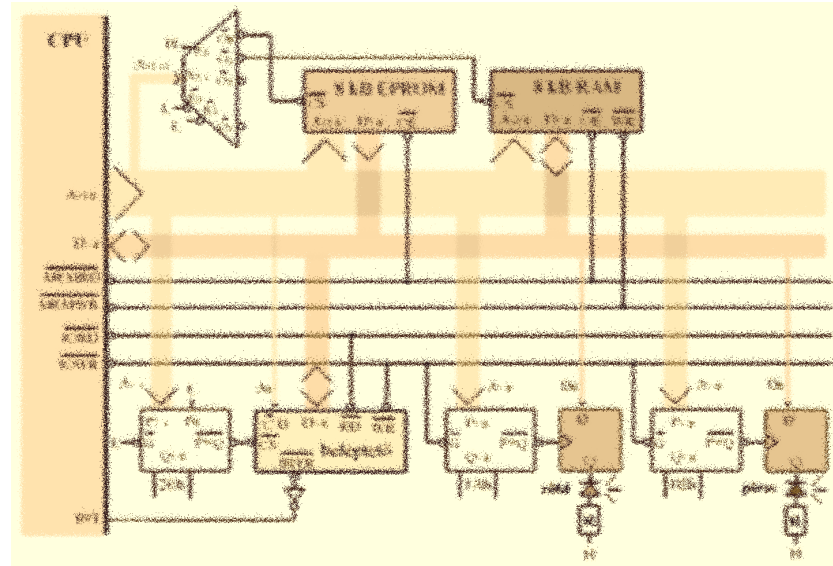


- A vezérlő regiszter D_2 bitjével állítható a zár állása: 0: zárás parancs, 1: nyitás parancs. A nyitás parancs után, amint az ajtót kinyitották rögtön, de legkésőbb 10s után (timeout: ha az ajtót addig nem nyitották ki) a beléptető zárnyelv vezérlője automatikusan átmegy zárt állapotba.
- A státuszregiszter D_2 bitjéből mindig kiolvasható a zár állása: 0: zárva, 1: nyitva

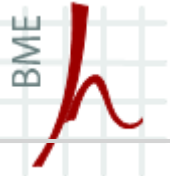


A HARDVER kapcsolási rajz vázlat

- Illesszünk a hipotetikus mikroprocesszorhoz
 - 8KB EPROM-ot a 0000h címre
 - 8KB SRAM-ot a 2000h címre
 - a beléptető perifériát a 2Eh báziscímre
 - így 2Eh/2Fh címeken érhető el
 - egy zöld és egy piros LED-et az 1Ah és az 1Bh portcímekre
 - D tárolók és ellenállások segítségével
 - az adatbusz D_0 bitjével lehessen az állapotukat vezérelni



A HARDVER TERVEZÉSÉNEK LÉPÉSEI



8K-s memória címtartománya

- K: * 1024 (k:*1000, pl. kg)
- $8K = 2^{13} = 8192$: 13 bites címbusz (0.. 12)
- 8K-s memória címtartománya:
 - 0.. 8191 decimálisan
 - 0000h.. 1FFFh hexadecimálisan

CPU
16 bites címbusz

8 K-s memória
címbusz

8 K-s memória
legkisebb cím bináris

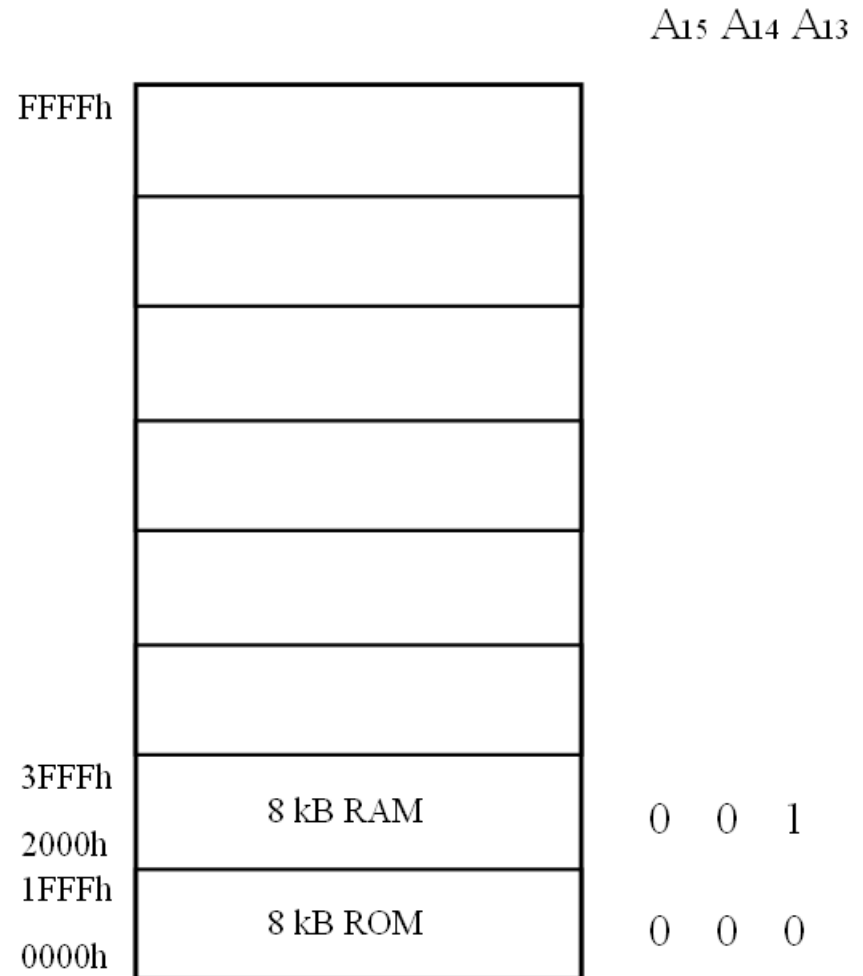
8 K-s memória
legkisebb cím hexa

8 K-s memória
legnagyobb cím bináris

8 K-s memória
legnagyobb cím hexa

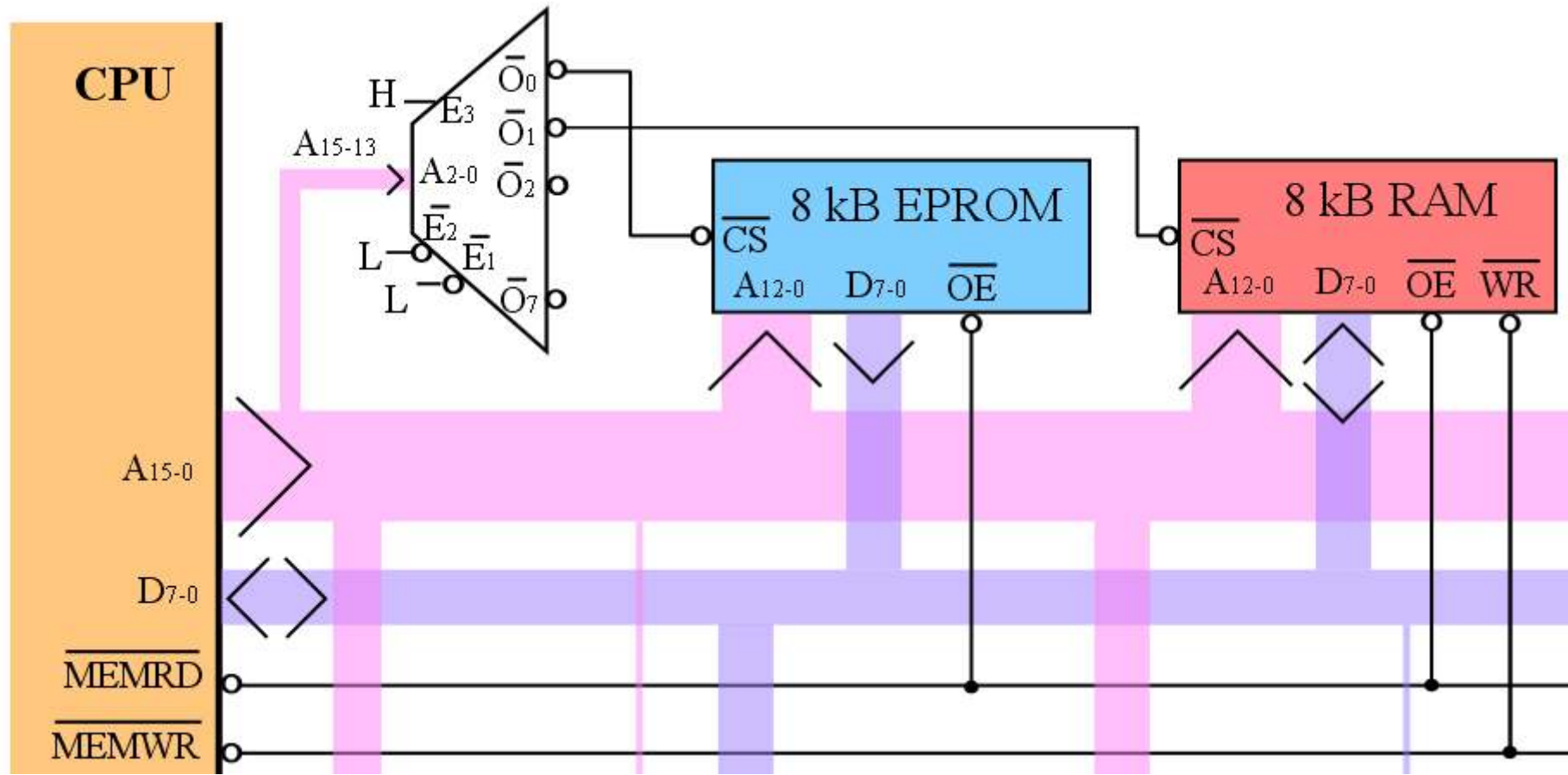
A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	
			A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	
			0	0	0	0	0	0	0	0	0	0	0	0	0	
			0					0				0				
			1	1	1	1	1	1	1	1	1	1	1	1	1	
			1					F				F				

- Készítsünk memóriatérképet!
 - Mellette: A_{15-13} címbitek
 - egy 3/8-as dekóderrel az EPROM és az SRAM illeszthető:
 - EPROM: $A_{15-13}=000$
 - SRAM: $A_{15-13}=001$
 - 3/8-as dekóder $/O_0$ és $/O_1$
 - Méretük 8kB
 - 13 címbit kell: A_{12-0}



Memóriák illesztése

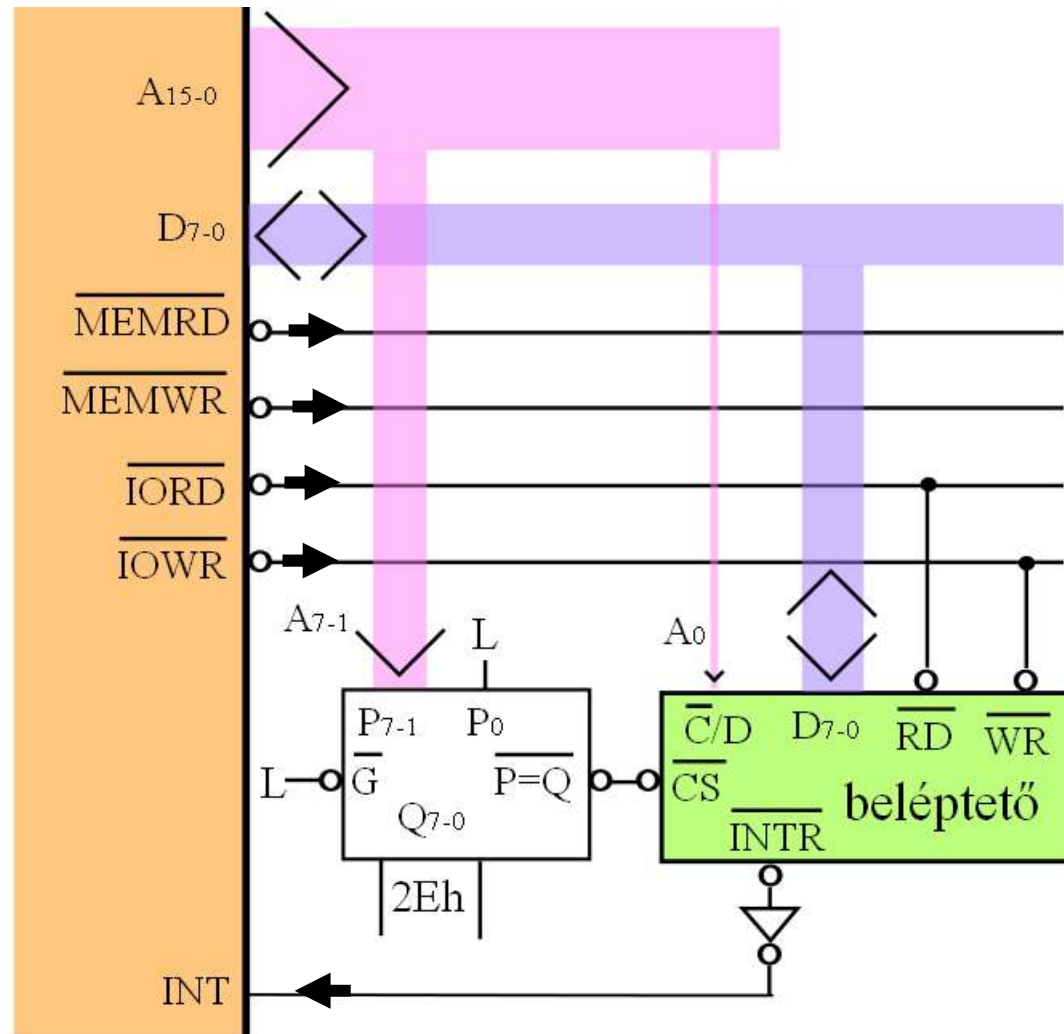
- A CPU és a hozzá illesztett memóriák



A beléptető periféria illesztése

- A beléptető perifériát címkomparátorral a 2Eh báziscímre illesztjük:
 - a címkomparátor P_{7-1} bemenetére a címbusz A_{7-1} bitjeit kötjük, de az A_0 címbit helyett P_0 -ra fixen logikai 0-t (L) kötünk, Q_{7-0} bemenetére pedig az 2Eh értéket kapuzzuk.
 - A címkomparátor /G engedélyező bemenetét fixen logikai 0-ra (L) kötjük.
 - A címkomparátor 0 aktív /(P=Q) kimenetével engedélyezzük a beléptető perifériát annak /CS bemenetén.
- Az A_0 címbitet rákötjük a beléptető periféria (/C)/D bemenetére;
- az /IORD, /IOWR, és D_{7-0} jeleket pedig értelemszerűen a beléptető periféria /RD, /WR, és D_{7-0} lábaira kötjük.
- A periféria /INTR kimenetét inverteren keresztül kötjük a CPU felfutó élre érzékeny INT lábára.

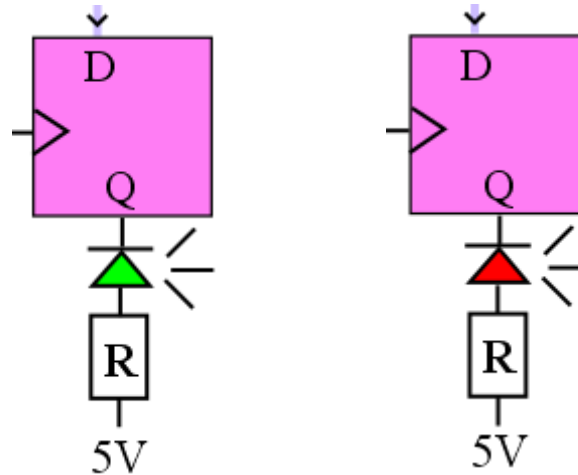
A beléptető periféria illesztése



LED-ek meghajtása

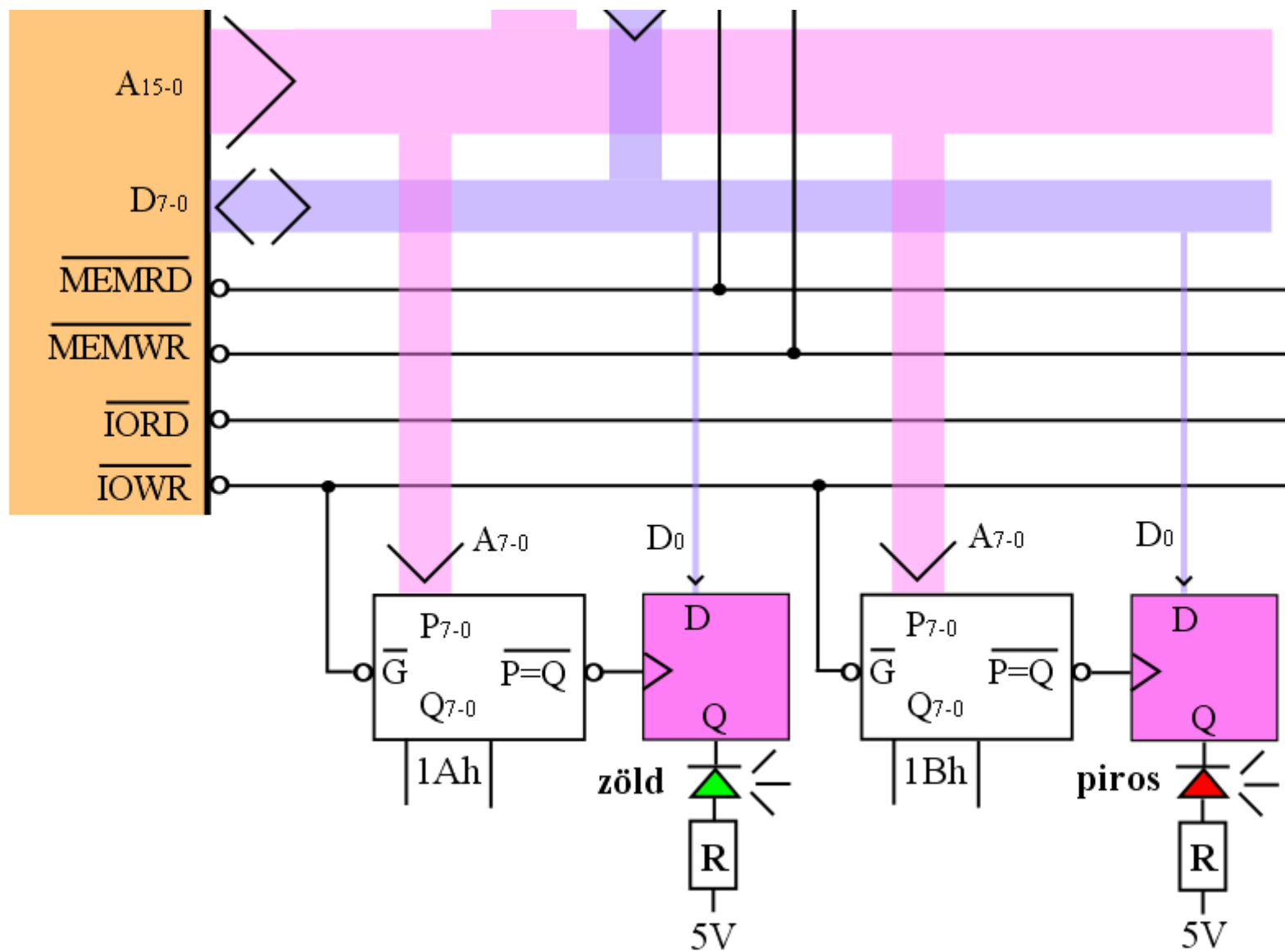
- A LED-ek meghajtásához fontoljuk meg:
 - A D tárolók TTL alkatrészek, kimeneti áram terhelhetőségük
 - logikai 1-nél: $< 1\text{mA}$
 - logikai 0-nál: 20mA környékén
 - csak logikai 0 esetén tudják a LED-eket meghajtani
 - Ha az egy LED-en nyitó irányban eső feszültség 2V , további veszteségekkel (kb. 1V) számolva a $250\ \Omega$ -os áramkorlátozó ellenálláson még kb. 2V feszültségesés 8mA áramot biztosít; ez a LED-ek számára megfelelő.
 - Tehát a LED-eket a D tárolók kimenetéről 250-os ellenálláson keresztül tápfeszültségre (5V) kötjük.
 - **FIGYELEM!** Így a LED-ek kigyújtásához 0-t kell a D tárolókba írunk, kioltásához pedig 1-et!

- Ügyeljünk, hogy a LED-ek rajzjele az elméleti áramirányt kövesse!

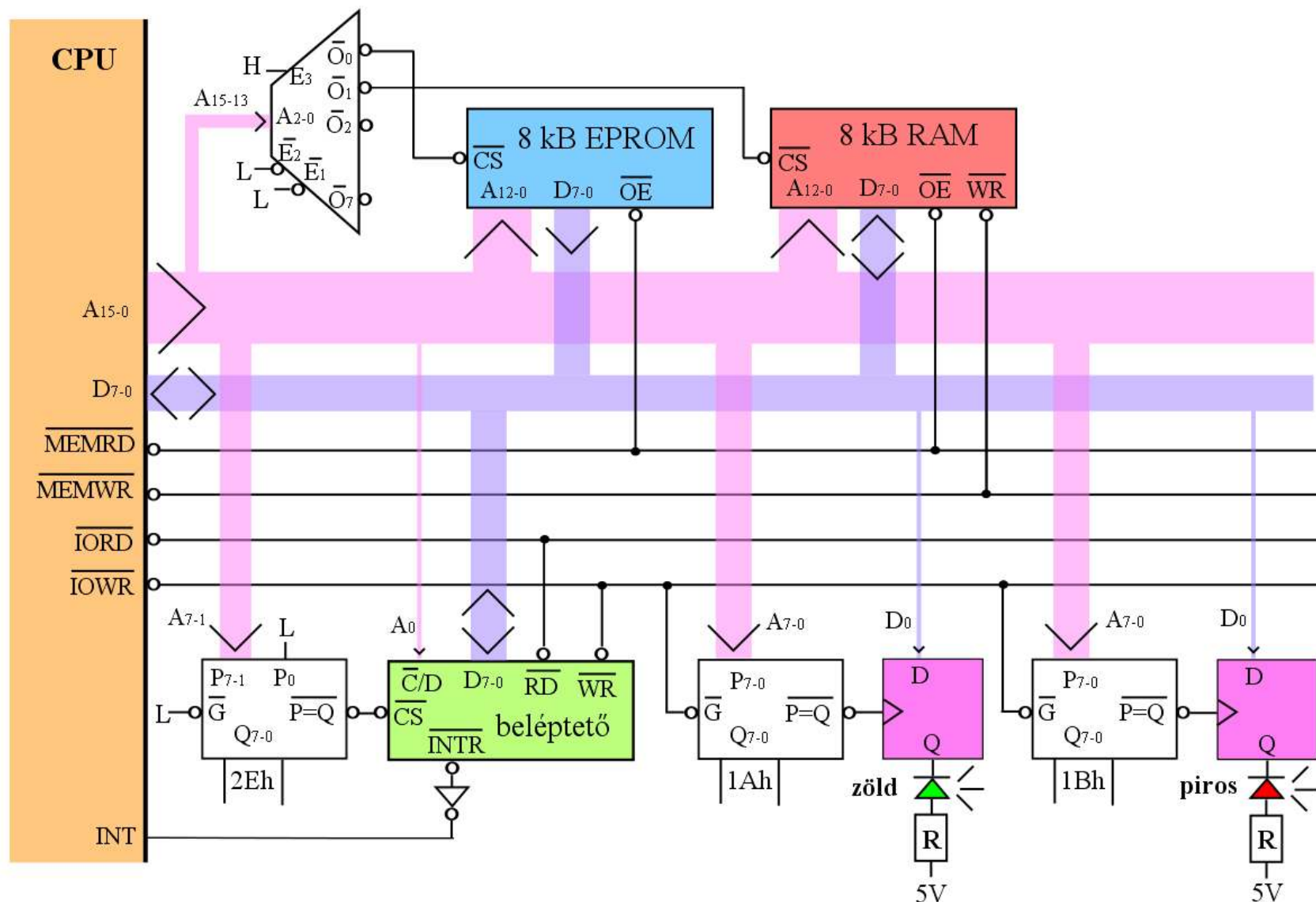


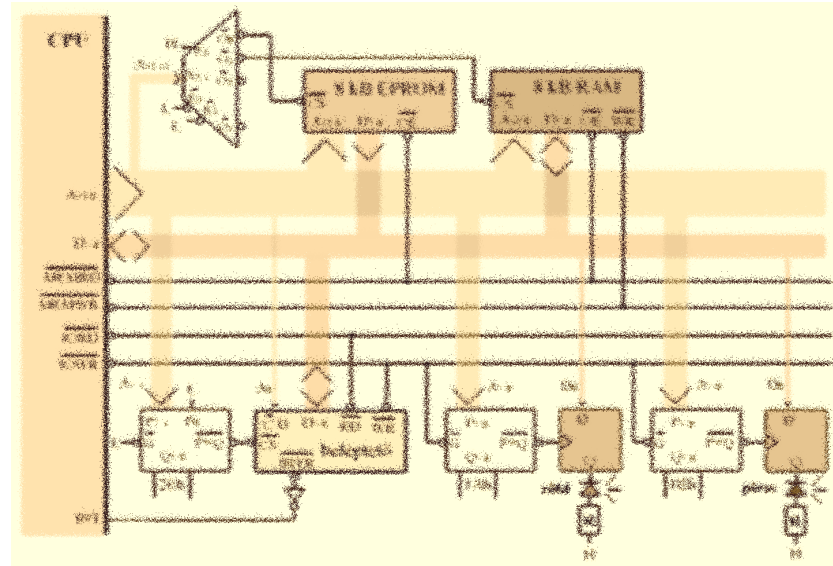
- A D tárolókat az 1Ah és az 1Bh portcímekre illesztjük címkomparátorokkal.
 - A címkomparátorok P_{7-0} bemenetére a címbusz A_{7-0} bitjeit kötjük, Q_{7-0} bemenetére pedig az 1Ah illetve 1Bh értékeket kapuzzuk.
 - A címkomparátorok engedélyezéséhez a /G engedélyező bemenetükre a CPU /IOWR kimenetét kötjük, mivel a tárolókat csak írunk kell.
 - A címkomparátorok 0 aktív /(P=Q) kimenetét a D tároló felfutó élre érzékeny órajel bemenetére kötjük; így az /IOWR megszűnésekor a felfutó él hatására a kiválasztott D tároló eltárolja a D bemenetére kötött adatbusz D_0 bitjének ekkor még stabilan tartott értékét.

D tárolók illesztése



A teljes logikai kapcsolási rajz

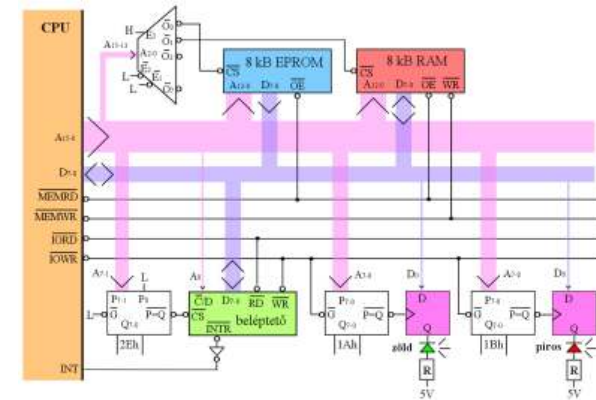


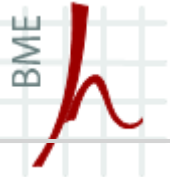


ALTERNATÍV MEGOLDÁSOK VIZSGÁLATA

- **Gondolkozzanak el önállóan az alábbi kérdéseken!**
 1. Hogyan módosítaná a hardvert, ha a 2000h címre nem 8 kB, hanem 16 kB RAM-ot kellene illeszteni?
 2. Milyen megoldást választana, ha a feladat úgy szólna, hogy a LED-eknek akkor kell világítani, ha vezérlésükhöz használt perifériacím 0. bitjére utoljára logikai 1 értéket írtunk és nem világítani, ha 0 értéket írtunk?
 3. Hogyan tudná egyszerűsíteni a rendszert, ha a specifikáció azt írja elő, hogy a piros és a zöld LED állapota mindig egymással ellentétes?

- Megismerkedtünk egy beleptető rendszer építéséhez használható hipotetikus és valós építőelemekkel
- Megadtunk egy konkrét specifikációt
- Megterveztük a kívánt hardvert
 - Memóriát illesztettünk
 - Perifériát illesztettünk
 - LED-ek meghajtásáról gondoskodtunk
 - D tárolókat illesztettünk (csak írásra)
- Alternatív megoldásokat is megvizsgáltunk





Memória technológiák

Általános kategória:

- **RAM** – Random Access Memory –
 - bármikor írható, olvasható
 - viszont a tartalom elveszik kikapcsoláskor – ***Volatile (felejtő)*** (de hogy legyen kivétel: NVRAM = SRAM+elem egy csipben)

- **ROM** – Read Only memory –
 - CSAK olvasható
 - viszont a tartalom örök – ***Non Volatile (nem felejtő)***



Kitérő: memória technológiák RAM

- SRAM (Static Random Access Memory)
 - olvasható, írható
 - Tápfeszültség kikapcsolásakor tartalom törlődik – *Volatile*
 - Gyors elérés, viszonylag kicsi kapacitás (2kbit.. 144Mbit - egy memória csipben)
 - Sok tranzisztor / bit – sokat fogyaszt, drága (Ft/bit)
- DRAM (Dynamic Random Access Memory)
 - olvasható, írható
 - Tápfeszültség kikapcsolásakor tartalom törlődik – *Volatile*
 - Nehézkes, lassú elérés (memória kontroller), viszonylag nagy kapacitás (4Mbit.. 4Gbit egy memória csipben)
 - Egy tranzisztor / bit – keveset fogyaszt, olcsó (Ft/bit)



Kitérő: memória technológiák ROM

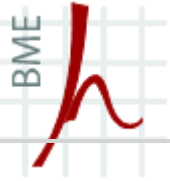
ROM technológiák

- **ROM** - Gyártáskor maszkprogramozott eszköz
 - *nem használatos ma már*
- Klasszikus **EPROM** – Erasable Programmable Read Only Memory
 - Erase – törlés UV fénnel – üveglap
 - Programozás (írás)
 - Hosszadalmas, bonyolult, lassú
 - ***Ma már nem használatos***
- Mai EPROM – (**OTP EPROM** – One Time Programmable ROM)
 - *Egyszer írható, viszonylag lassan (100us/bájt)*
 - Gyorsan olvasható, mint egy SRAM
 - Viszonylag kis tárolási kapacitás: 768bit.. 1Mbit



- **EEPROM** (Electrically Erasable and Programmable Read Only Memory)
 - Viszonylag kis tárhkapacitás (128bit.. 2Mbit)
 - Viszonylag gyors olvasás
 - Úgy írható, mint egy RAM (átmeneti tárbn tartja az írandókat, majd onnan programozza a memóriát)
 - Lehet párhuzamos adatbuszú, pl. 8 bit széles, de igen gyakran soros interfészű (SPI, I2C stb.)
 - Tipikusan viszonylag kevés setup, konfigurációs adat tárolására

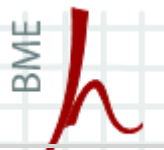




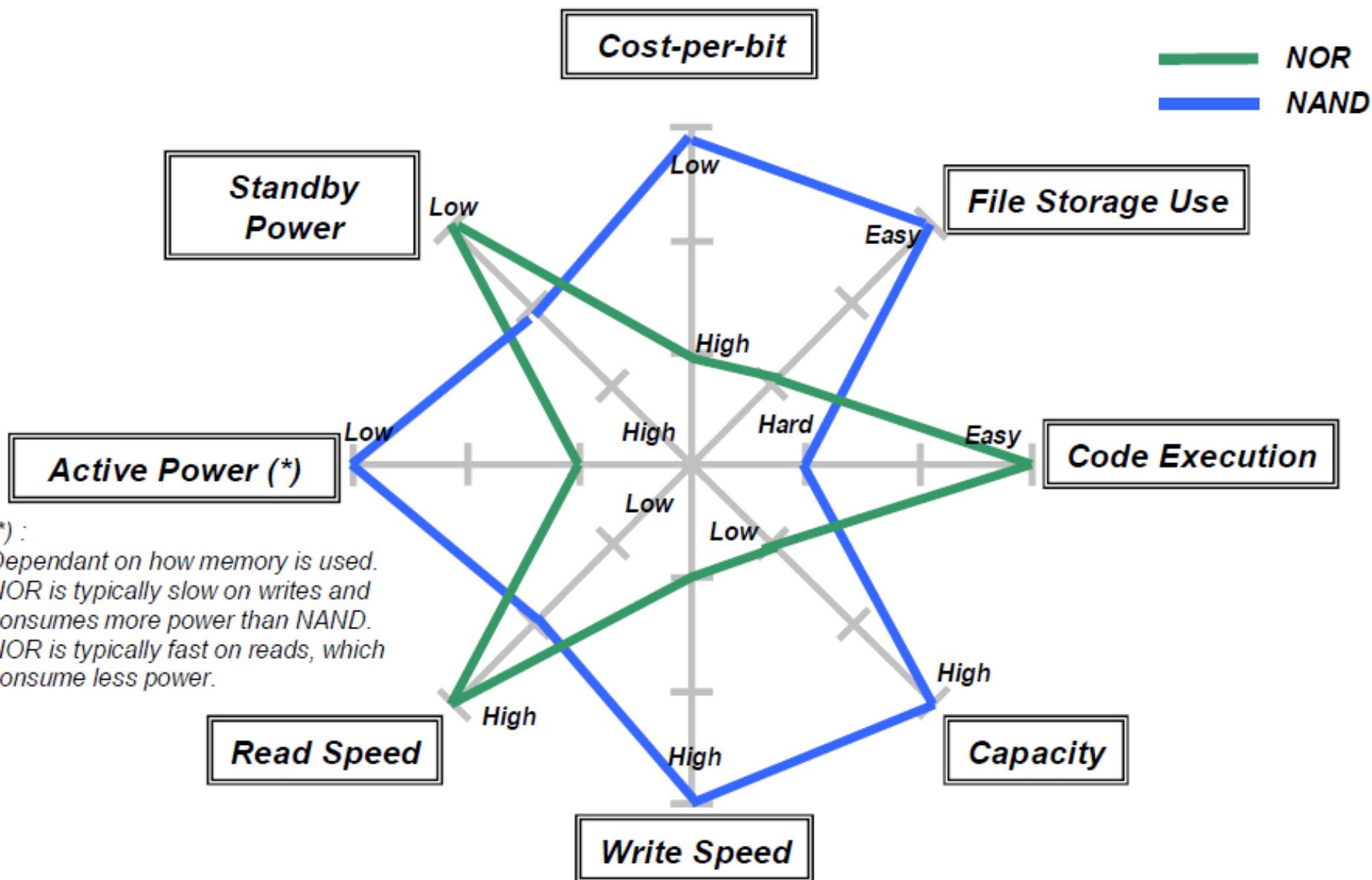
Kitérő: memória technológiák: FLASH

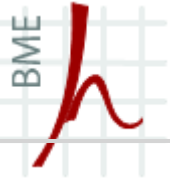
FLASH (EEPROM technológiából kifejlesztett áramkörben írható, olvasható nem felejtő memória, nagyon elterjedt)

- NOR FLASH
 - EEPROM-hoz hasonló véletlen írás olvasás jellemzi
 - **Gyors olvasás** - kód végrehajtás lehetséges belőle
 - Viszonylag kicsi kapacitás (512Kbit.. 2Gbit)
 - Viszonylag drága (Ft/bit)
- NAND FLASH
 - Viszonylag nagy tárcapacitás (256Kbit.. 512Gbit)
 - Lassú, bonyolult olvasás, írás
 - Törlés blokkokban – gyors törlés
 - **Nagy, nem felejtő adattárolók:** USB FLASH, Solid State Disk
 - Nagyon olcsó (Ft/bit)



NOR - NAND FLASH





NOR - NAND FLASH

	NAND	NOR
Advantages	Fast PROGRAMs	Random access
	Fast ERASEs	Byte PROGRAMs possible
Disadvantages	Slow random access	Slow PROGRAMs
	Byte PROGRAMs difficult	Slow ERASEs
Applications	File (disk) applications	Replacement of EPROM
	Voice, data, video recorder	Execute directly from nonvolatile memory
	Any large sequential data	



Kérdések?

KÖSZÖNÖM A FIGYELMET!