Számítógép architektúrák

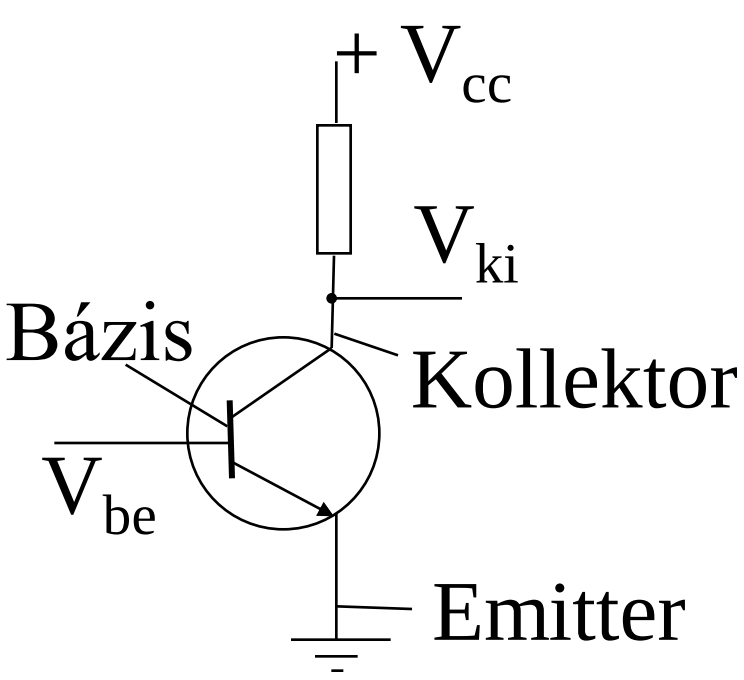
- **Számítógépek felépítése**
- **Digitális adatábrázolás**
- **Digitális logikai szint**
- **Mikroarchitektúra szint**
- **Gépi utasítás szint**
- **Operációs rendszer szint**
- **Assembly nyelvi szint**
- **Probléma orientált (magas szintű) nyelvi szint**
- **Perifériák**

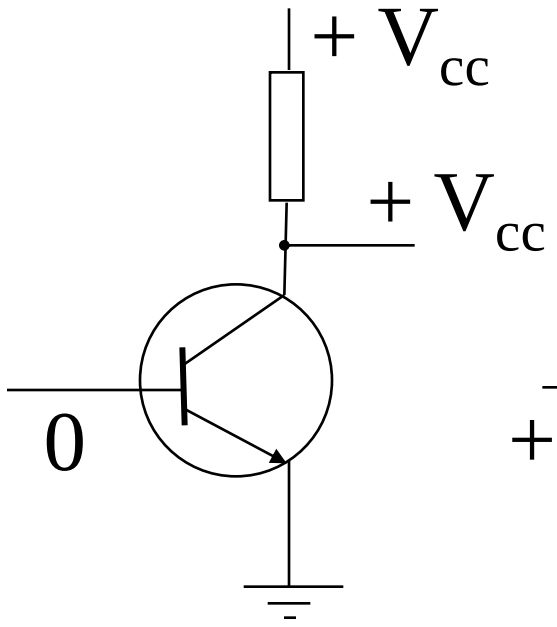
Digitális logikai szint

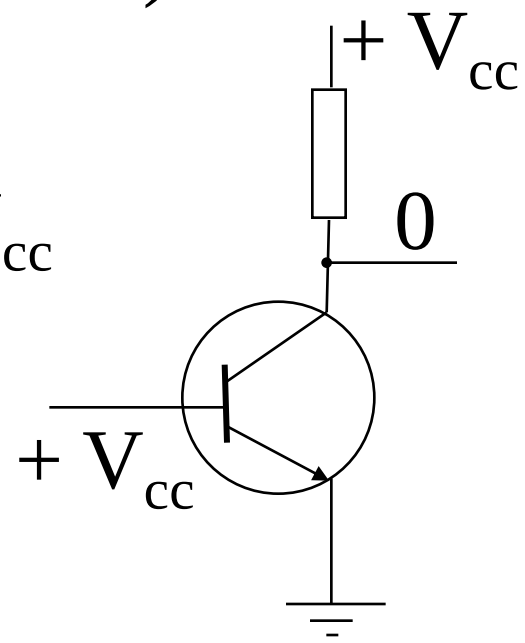
Digitális áramkör: két érték – általában
0-1 Volt között az egyik (pl. 0, hamis),
2-5 Volt között a másik (1, igaz).
Más feszültségeket nem engednek meg.

Kapu (gate): kétértékű jelek valamilyen függvényét tudja meghatározni.
Kapcsolási idő néhány ns
(nanoszekundum = 10^{-9} s)

NEM (*NOT*) kapu (3.1-2. ábra)







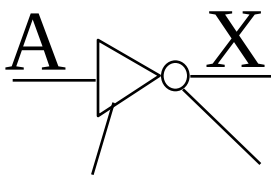
***NEM* (*NOT*)** kapu, inverter

Tranzisztor

Igazság tábla:

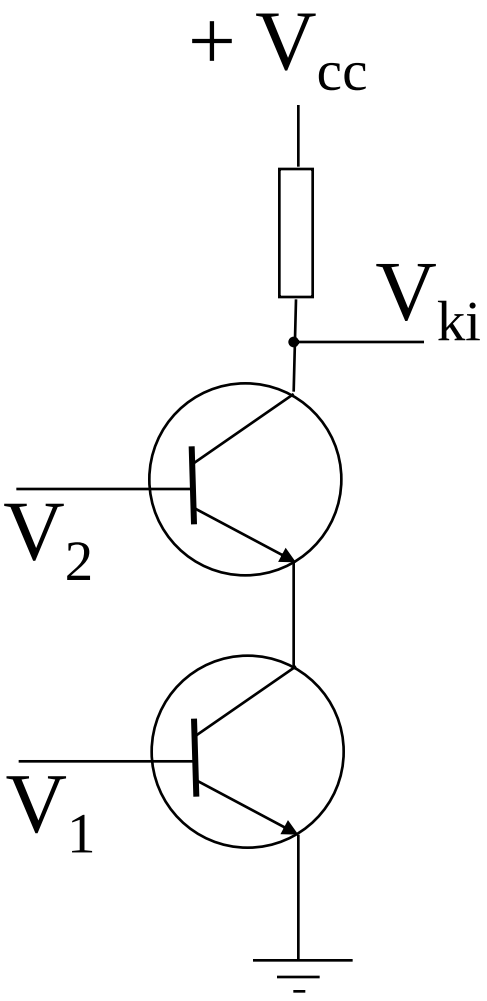
A	X
0	1
1	0

Szimbolikus jelölése:



erősítő **Inverziós gömb**

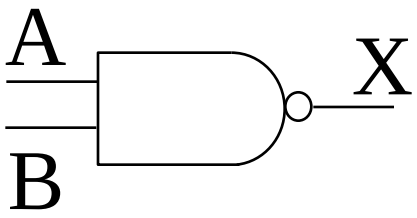
NEM-ÉS (*NAND*) kapu (3.1-2. ábra)



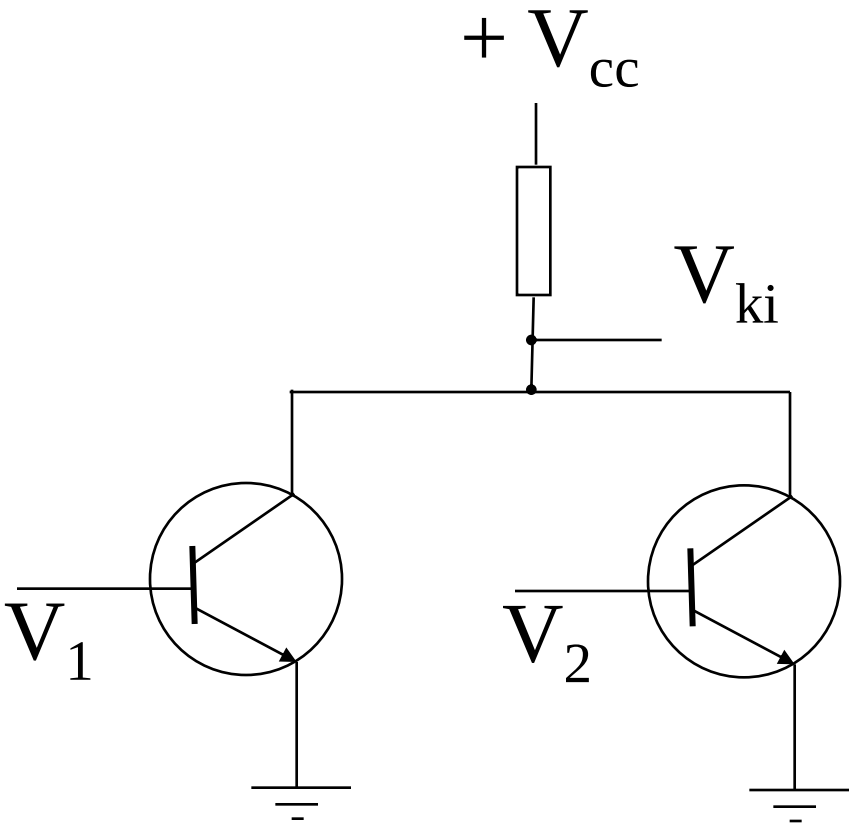
Igazság tábla:

A	B	X
0	0	1
0	1	1
1	0	1
1	1	0

Szimbolikus jelölése



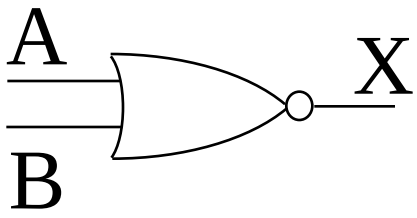
NEM-VAGY (*NOR*) kapu (3.1-2. ábra)



Igazság tábla:

A	B	X
0	0	1
0	1	0
1	0	0
1	1	0

Szimbolikus jelölése

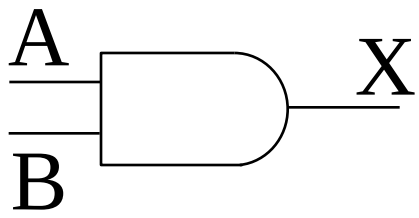


ÉS (*AND*) kapu (3.2. ábra)

Igazság tábla:

A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

Szimbolikus jelölése

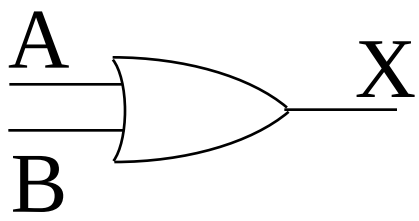


VAGY (*OR*) kapu (3.2. ábra)

Igazság tábla:

A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

Szimbolikus jelölése



Boole-algebra

Olyan algebra, amelynek változói és függvényei csak a 0, 1 értéket veszik fel,
a műveletei:

- ***ÉS*** (konjunkció),
- ***VAGY*** (diszjunkció),
- ***NEM*** (negáció).

Igazságtábla: olyan táblázat, amely a változók összes lehetséges értéke mellett megadja a függvény vagy kifejezés értékét.

Pl. 3 változós többségi függvény (**3.3. ábra**):
értéke 1, ha legalább két argumentuma 1

Igazság tábla:

A	B	C	M
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Boole-algebrai alakja:

$$M = \overline{A}BC + A\overline{B}C + AB\overline{C} + ABC$$

A fölülvonás a ***NEM*** *(negáció)*,

az egymás mellé írás az ***ÉS***,

a + a ***VAGY*** művelet jele.

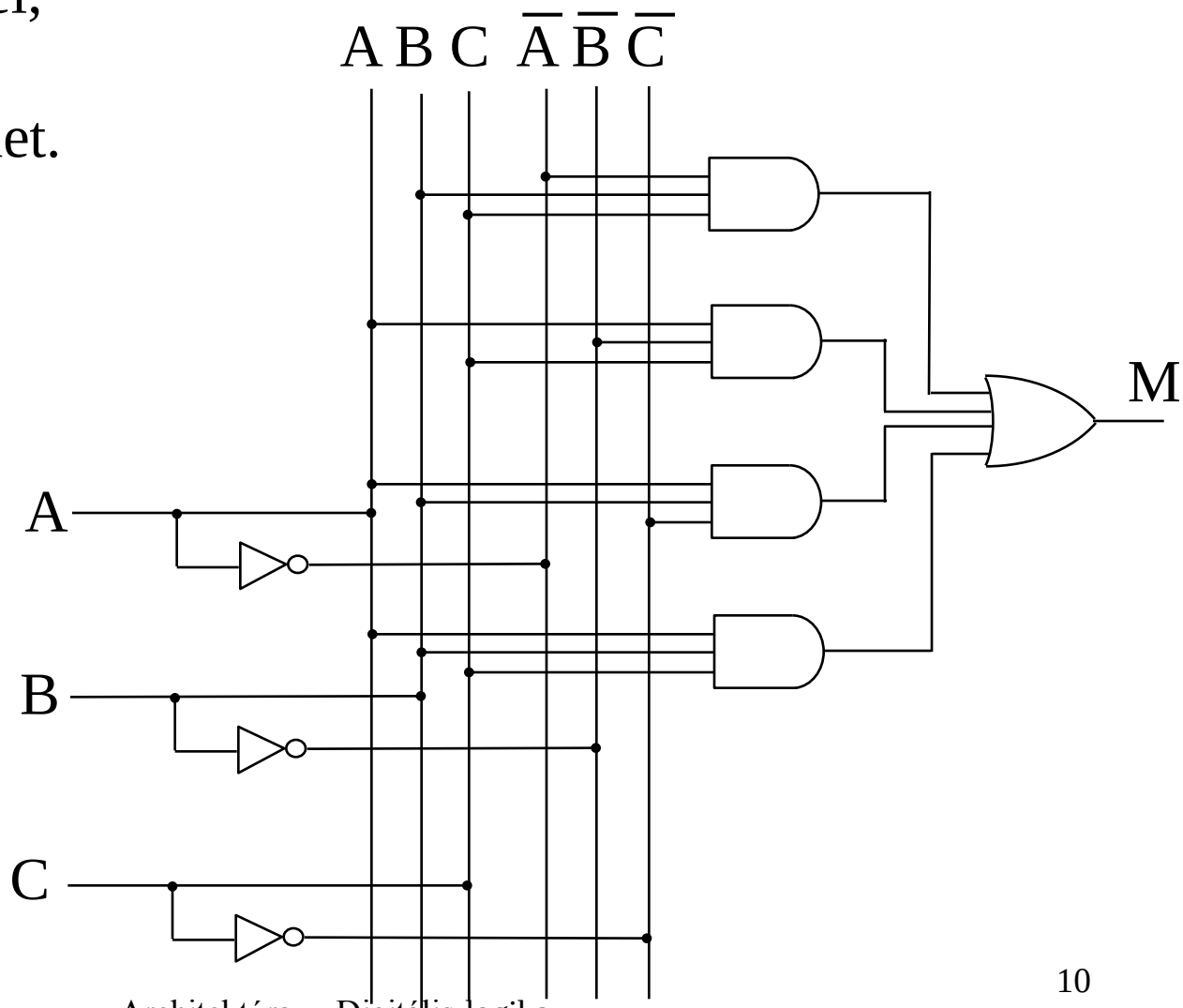
Diszjunktív normálforma.

Boole-függvény megvalósításának lépései (3.3. ábra):

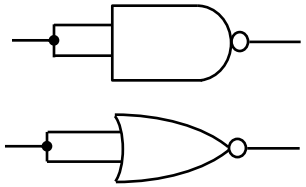
- igazságtábla,
- negált értékek,
- ***ÉS*** kapuk bemenetei,
- ***ÉS*** kapuk,
- ***VAGY*** kapu, kimenet.

$$M = \overline{A}BC + A\overline{B}C + AB\overline{C} + ABC$$

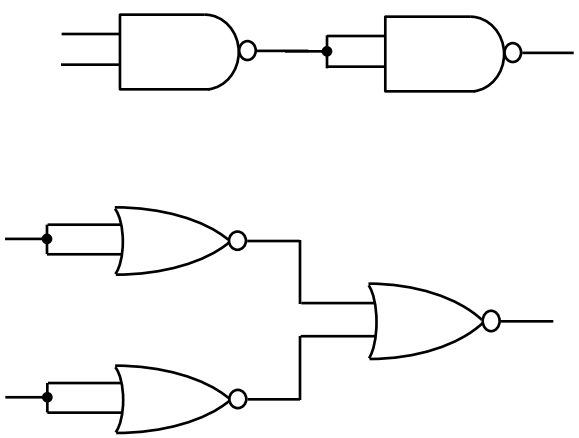
A	B	C	M
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1



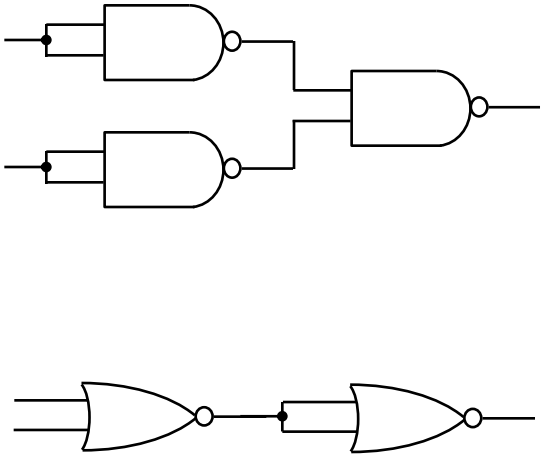
NAND és *NOR* előnye: teljesség (**3.4. ábra**)



NOT



AND



OR

Definíció: Akkor mondjuk, hogy két Boole-függvény **ekvivalens**, ha az összes lehetséges bemenetre a két függvény azonos kimenetet ad.

Két Boole-függvény ekvivalenciája könnyen ellenőrizhető az igazság táblájuk alapján.

Pl.: AB + AC és A(B + C) ekvivalens (**3.5. ábra**).

Az első függvény megvalósításához két ***ÉS*** és egy ***VAGY*** kapura van szükség, a másodikhoz elegendő egy ***ÉS*** és egy ***VAGY*** kapu.

Néhány azonosság (3.6. ábra)

Szabály	***ÉS*** forma	***VAGY*** forma
Identitás	$1A = A$	$0+A=A$
Null	$0A = 0$	$1+A=1$
Idempotens	$AA=A$	$A+A=A$
Inverz	$A\overline{A}=0$	$A+\overline{A}=1$
Kommutatív	$AB=BA$	$A+B=B+A$
Asszociatív	$(AB)C=A(BC)$	$(A+B)+C=A+(B+C)$
Disztribúciós	$A+BC=(A+B)(A+C)$	$A(B+C)=AB+AC$
Abszorpciós	$A(A+B)=A$	$A+AB=A$
De Morgan	$\overline{AB}=\overline{A}+\overline{B}$	$\overline{A+B}=\overline{A}\,\overline{B}$

Disztribúciós szabály:

$$A+BC=A+(BC)=(A+B)(A+C)$$

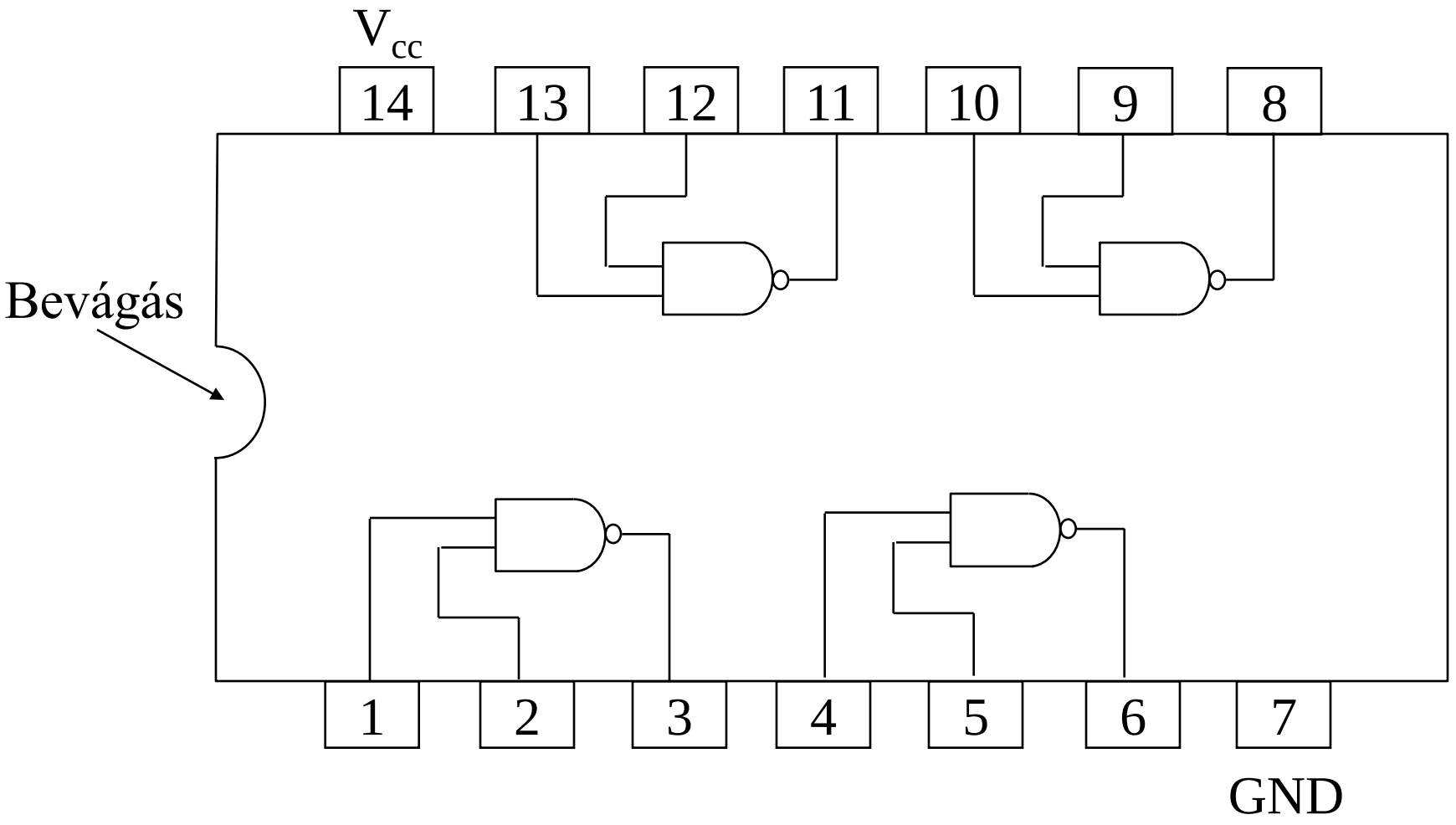
Jelölje az ***ÉS*** műveletet $\wedge$, a ***VAGY*** műveletet $\vee$, akkor

$$A \vee (B \wedge C) = (A \vee B) \wedge (A \vee C)$$

Alapvető digitális logikai áramkörök

Integrált áramkör (**IC**, Integrated Circuit, chip, lapka) 5x5 mm^2 szilícium darab kerámia vagy műanyag lapon (tokban), lábakkal (pins). Négy „alaptípus”:

- **SSI** (Small Scale Integrated 1-10 kapu),
- **MSI** (Medium Scale ..., 10-100 kapu),
- **LSI** (Large Scale..., 100-100 000 kapu),
- **VLSI** (Very Large Scale ..., > 100 000 kapu).



3.10. ábra SSI lapka négy ***NAND*** kapuval

$\mathbf{V_{cc}}$: Tápfeszültség, **GND**: föld.

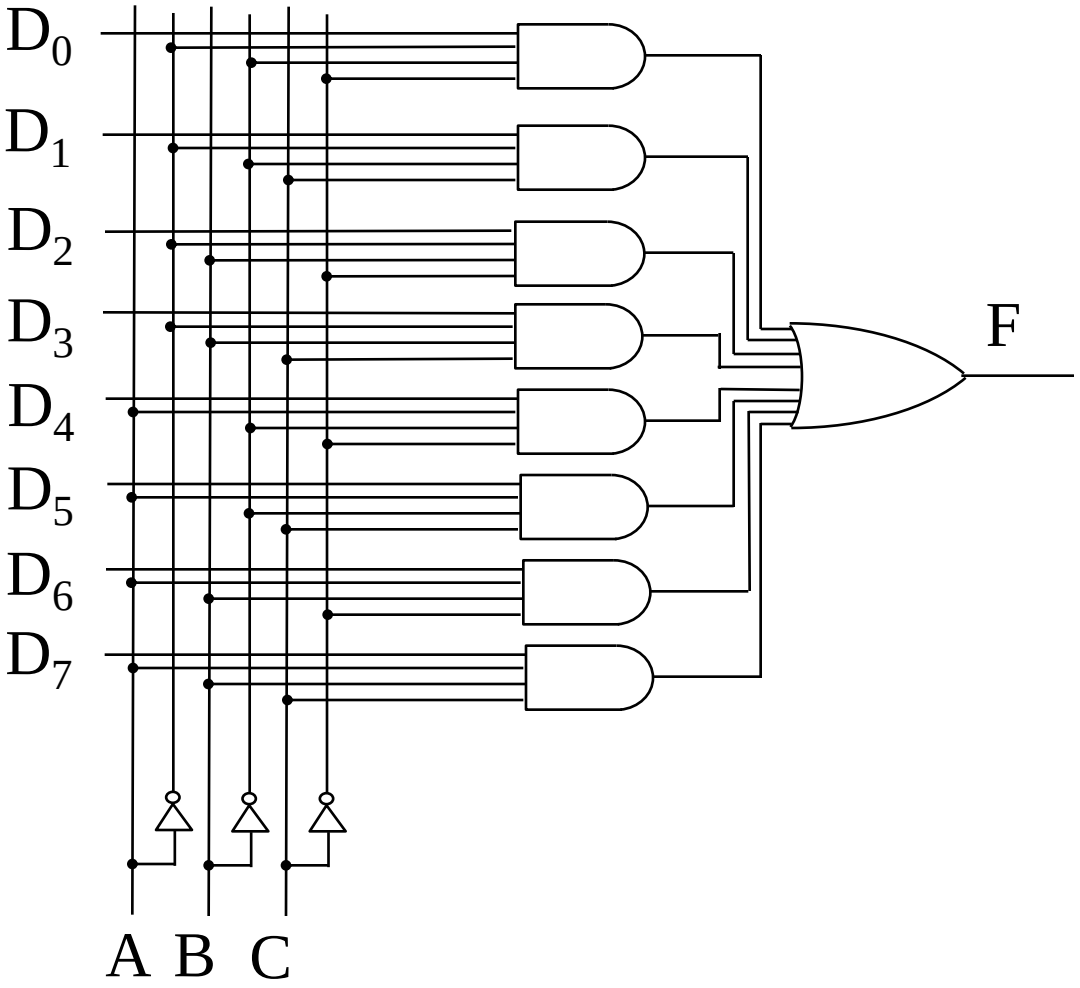
Kívánalom: sok kapu – kevés láb

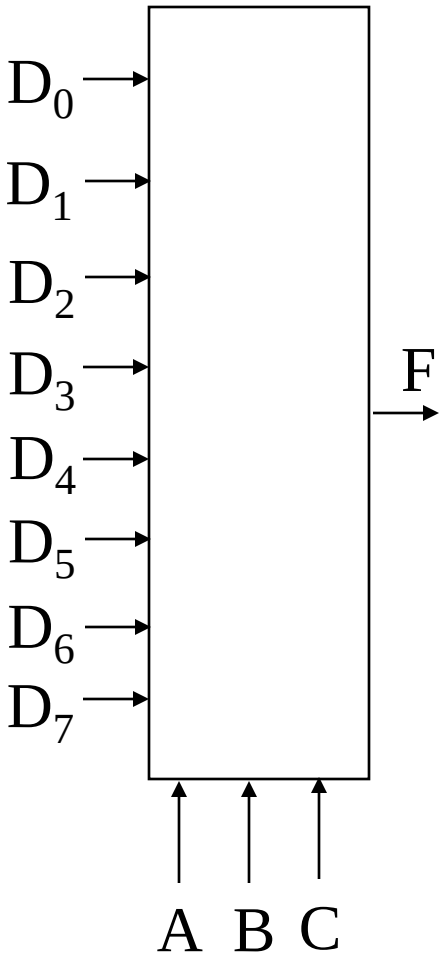
Kombinációs áramkörök

Definíció:

A kimeneteket egyértelműen meghatározzák a pillanatnyi bemenetek.

- **Multiplexer: *n*** vezérlő bemenet, ***2^n^*** adatbemenet, ***1*** kimenet. Az egyik adatbemenet kapuzott (gated) a kimenetre (**3.11-12. ábra**).





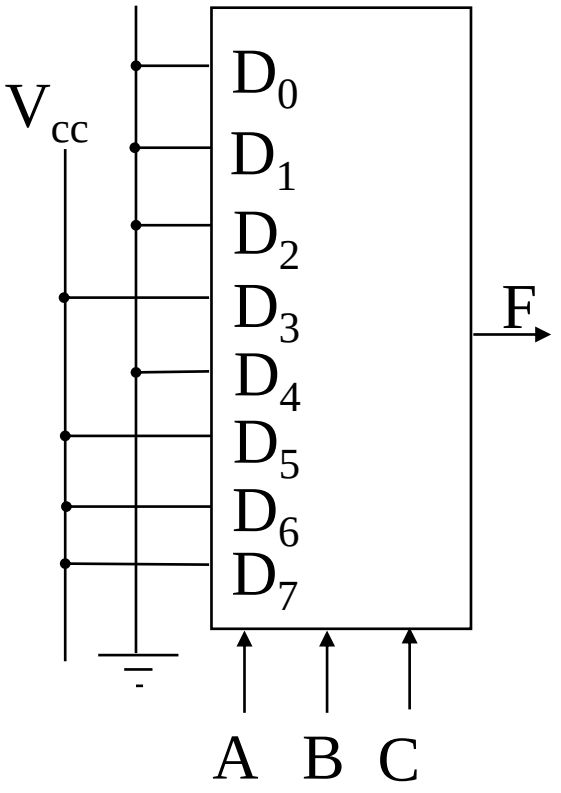
Sematikus rajza

n vezérlő bemenetű multiplexerrel tetszés szerinti ***n*** változós Boole-függvény megvalósítható az adatbemenetek megfelelő választásával. Pl. a 3 változós többségi függvény:

3.12. ábra

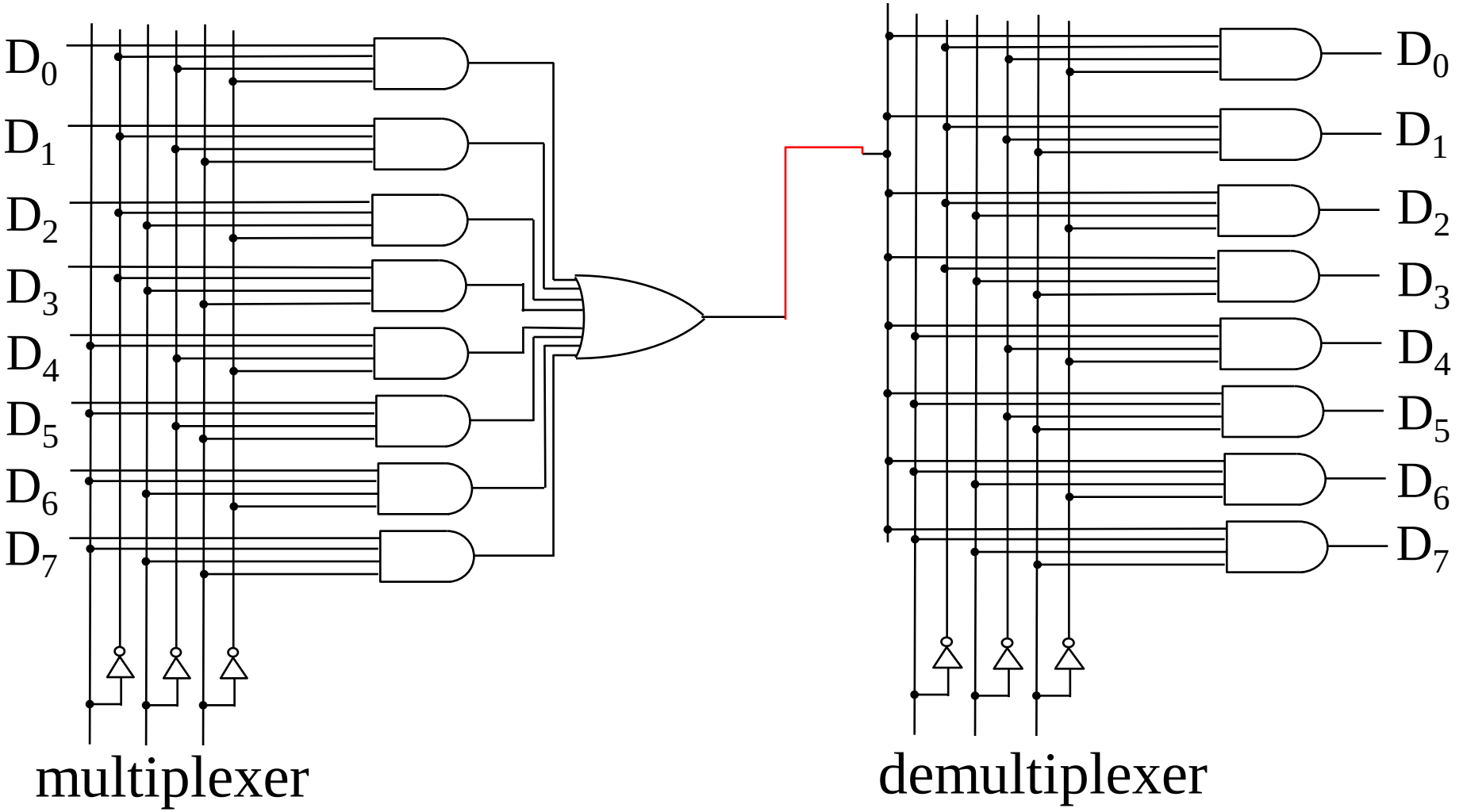
Igazság tábla:

A	B	C	M
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

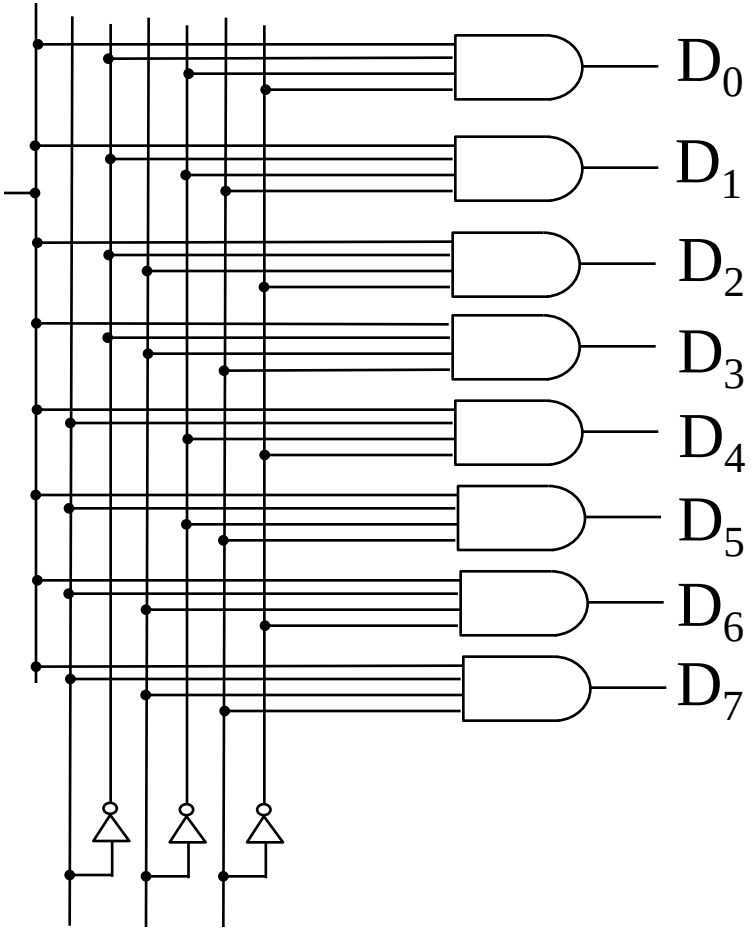


Párhuzamos-soros átalakítás:
vezérlő vonalakon rendre: 000, 001, … 111.

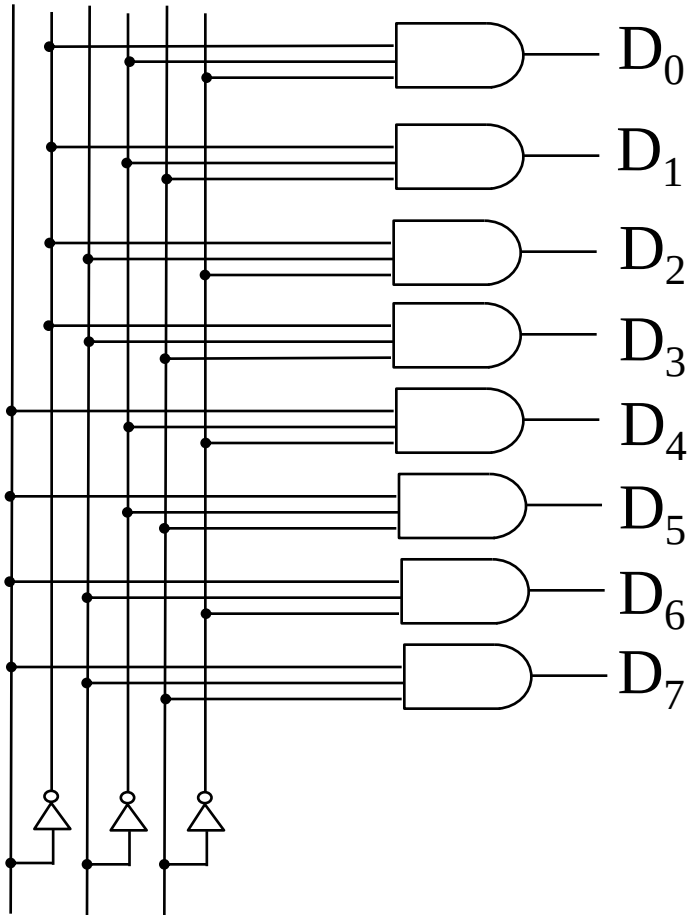
- **Demultiplexer:** egy egyedi bemenetet irányít az ***n*** vezérlő bemenet értékétől függően a 2^n kimenet egyikére



- **Dekódoló: *n*** bemenet, ***2^n^*** kimenet. Pontosan egy kimeneten lesz 1 (**3.13. ábra**). Demultiplexerrel: a bemenetet igazra állítjuk.



demultiplexer



dekódoló

- **Összehasonlító (comparator): (3.14. ábra)**

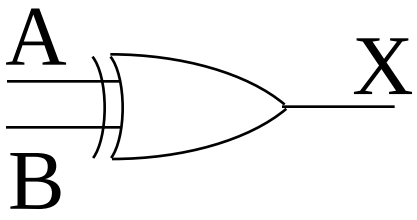
KIZÁRÓ VAGY kapu

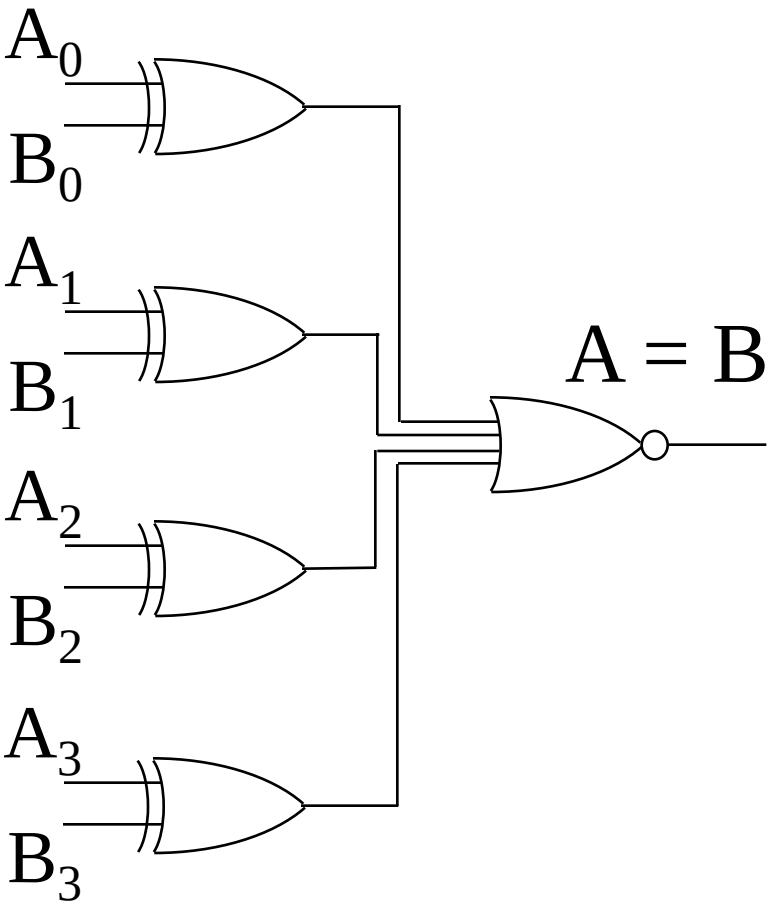
(***XOR eXclusive OR***)

Igazság tábla:

A	B	X
0	0	0
0	1	1
1	0	1
1	1	0

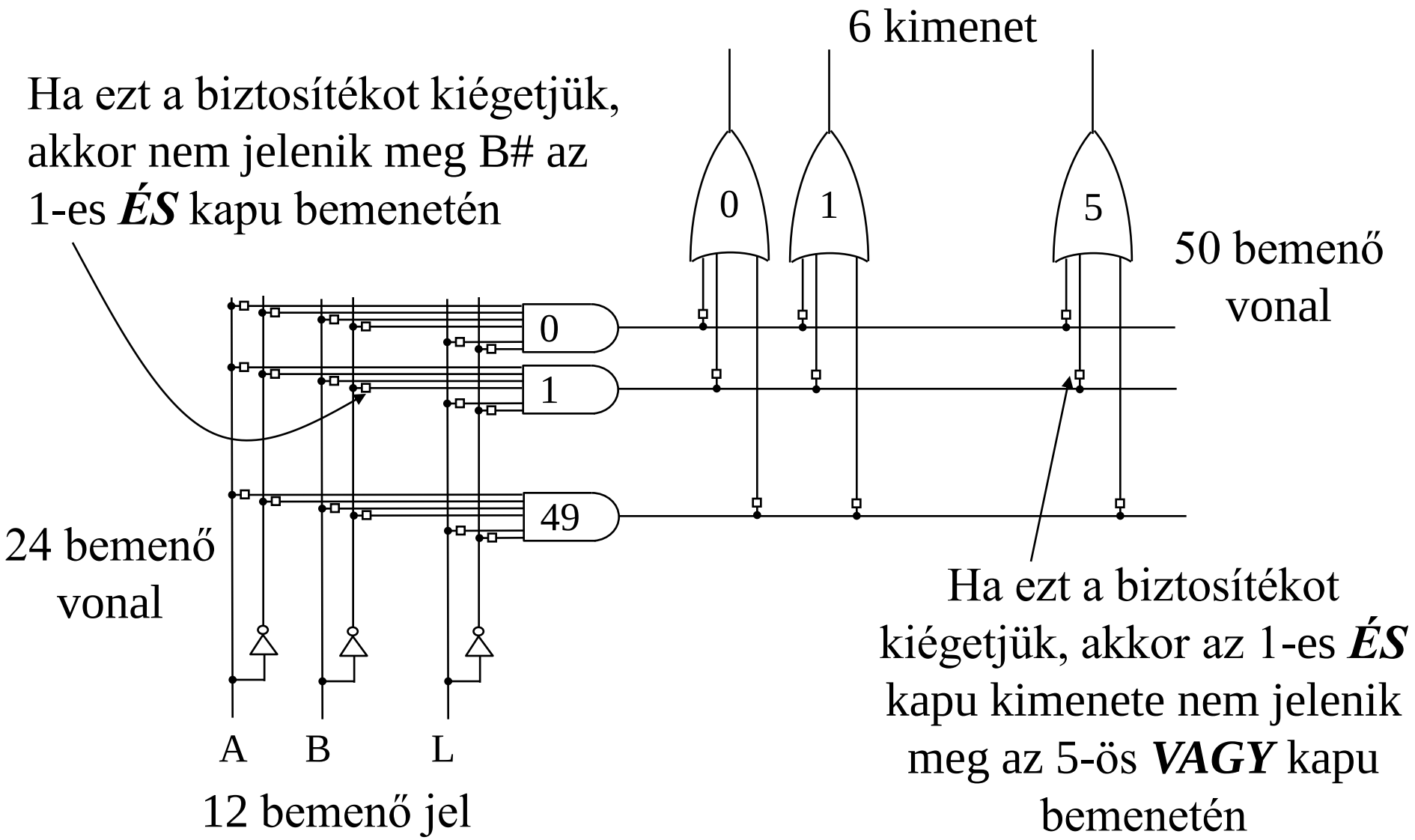
Szimbolikus jelölése





4 bites összehasonlító

- **Programozható logikai tömbök: PLA (3.15. ábra)** (Programmable Logic Array).

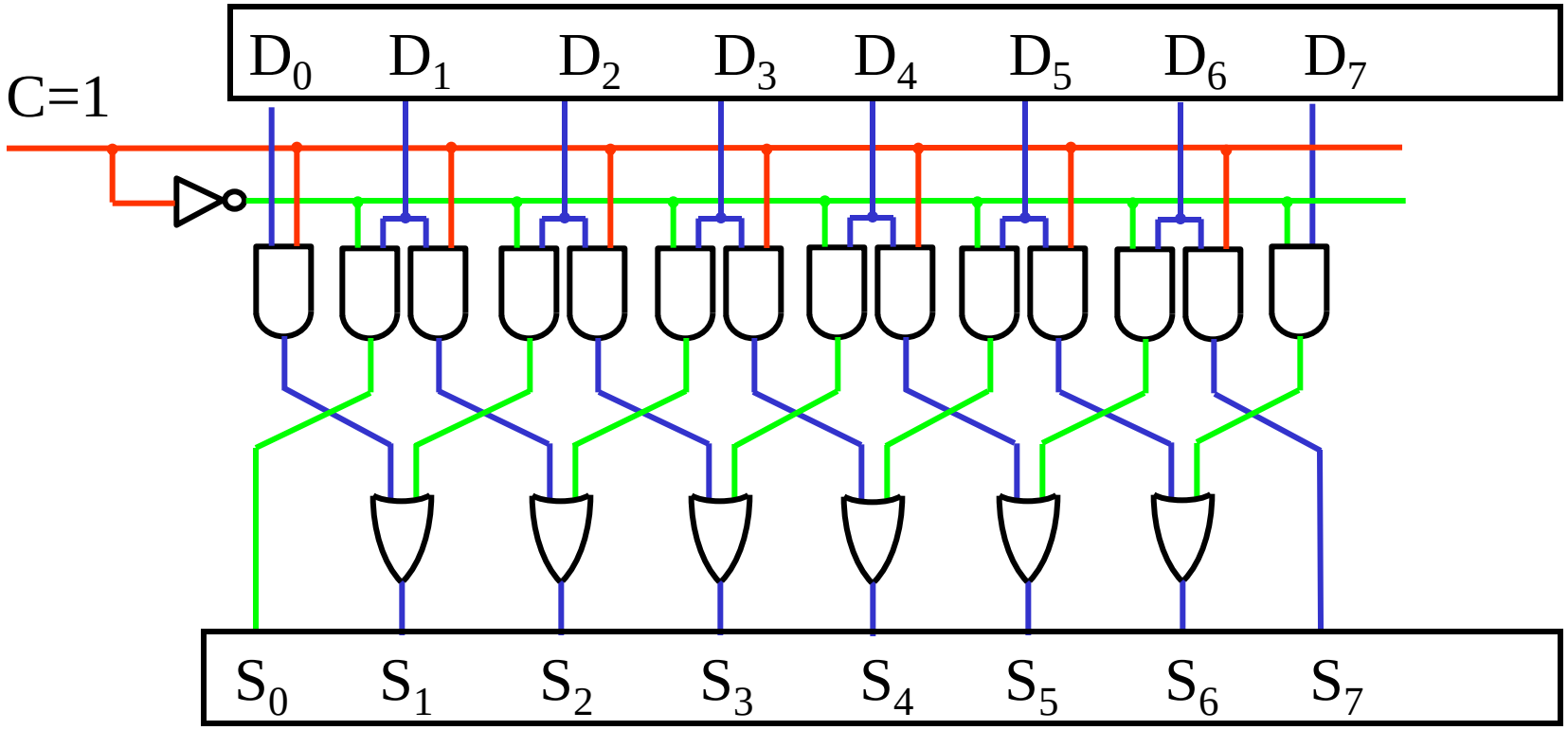


Aritmetikai áramkörök

A kombinációs áramkörökön belül külön csoportot alkotnak.

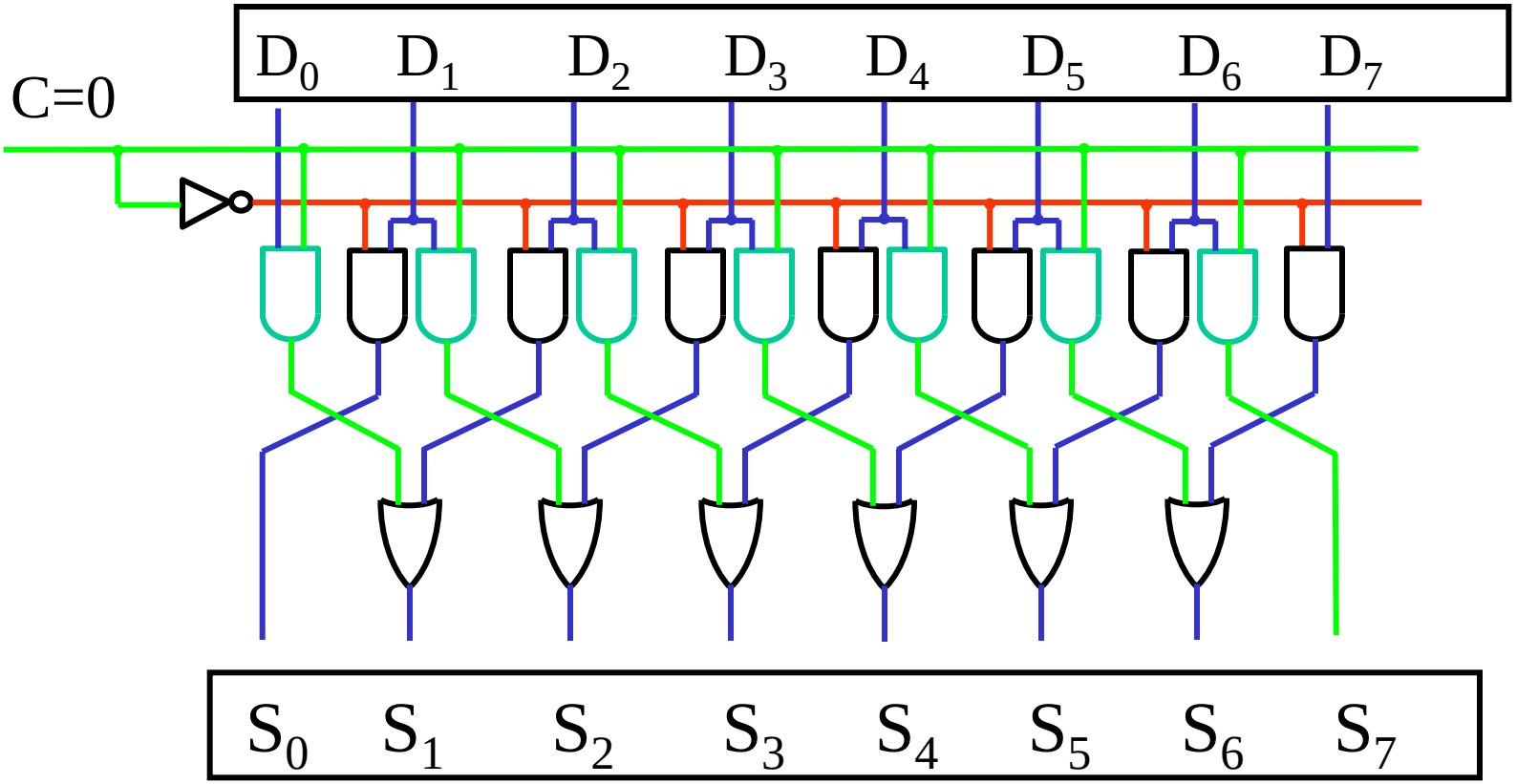
Léptető (shifter): 3.16. ábra, C=1: jobbra léptet.

(Igaz, Hamis, Adat)

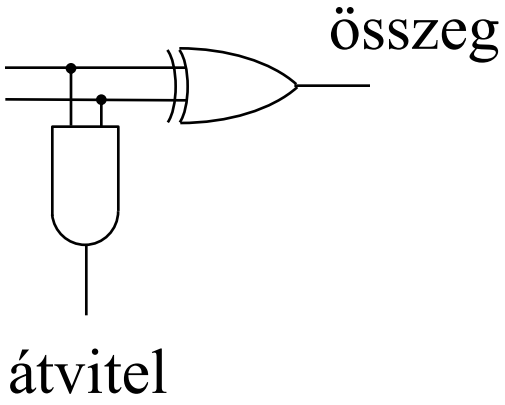


Léptető (shifter): 3.16. ábra, C=0: balra léptet.

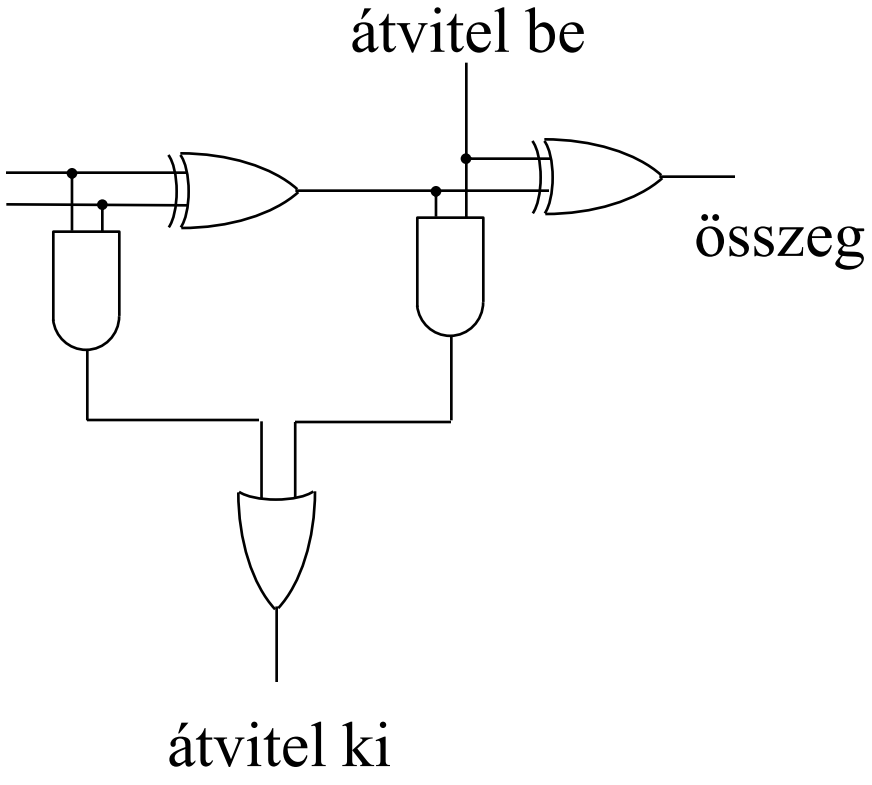
(Igaz, Hamis, Adat)



Összeadók:

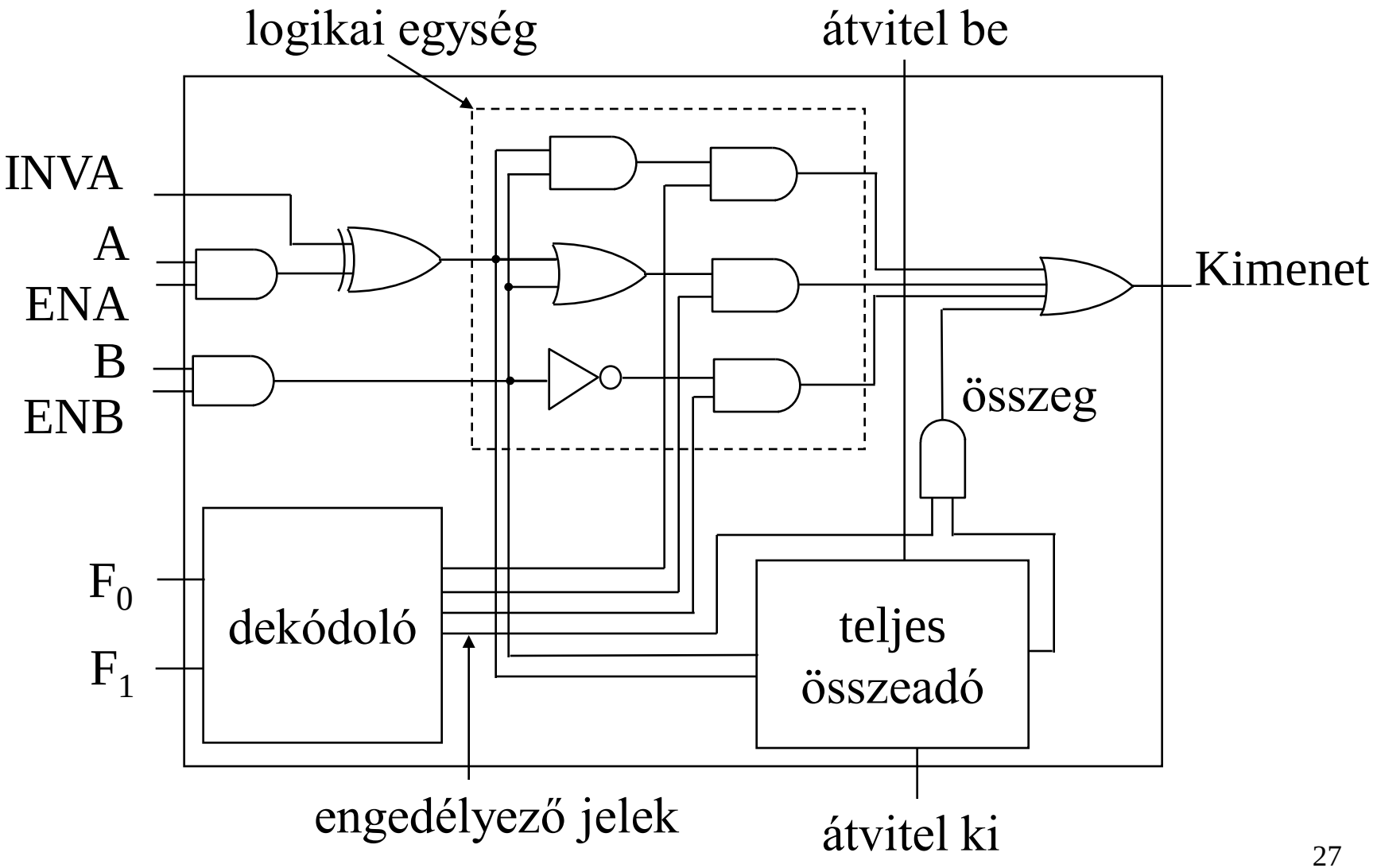


Fél-összeadó
(half adder, **3.17. ábra**)

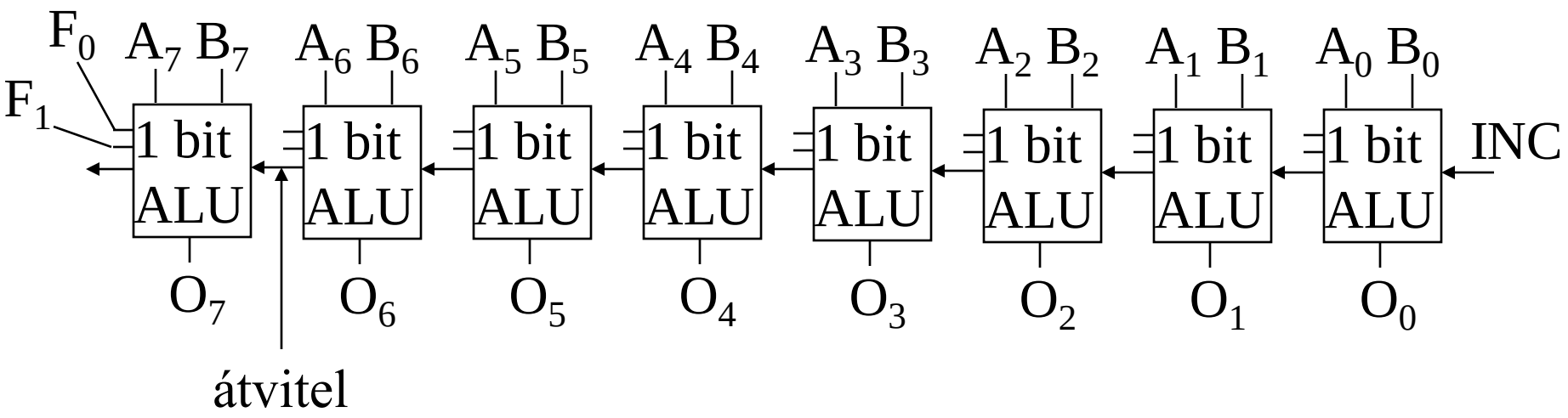


Teljes-összeadó
(full adder, **3.18. ábra**)

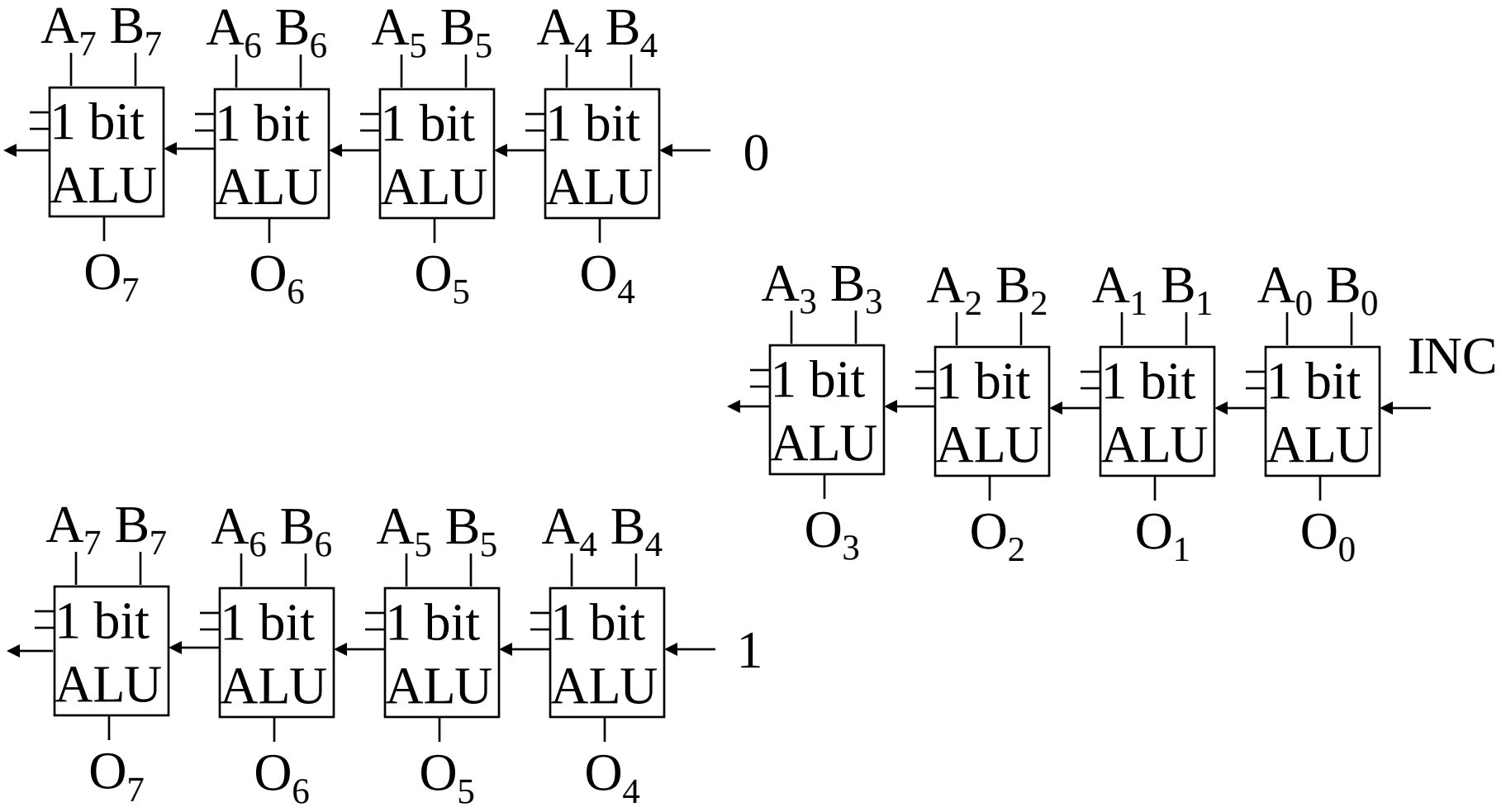
Aritmetikai-logikai egység: bitszelet (bit slice, **3.19. ábra**),
F0, **F1** -től függően ***ÉS***, ***VAGY***, ***NEGÁCIÓ*** vagy +



- átvitel továbbterjesztő összeadó (ripple carry adder):



- átvitel kiválasztó összeadó (carry select adder) eljárás:



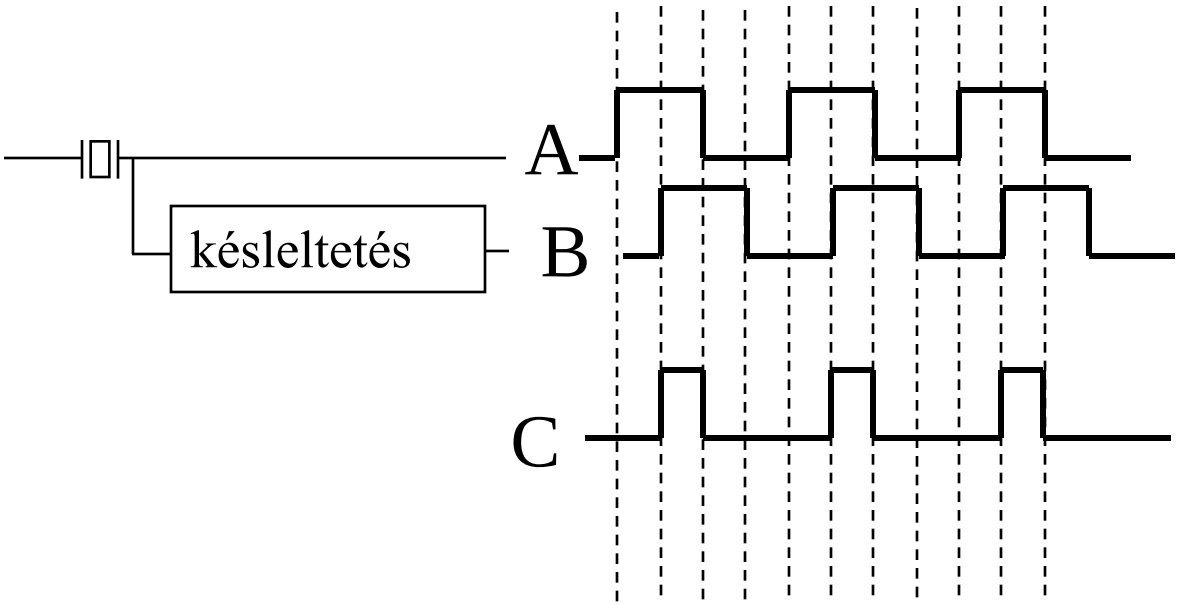
Nem kombinációs áramkörök

Óra (clock, 3.21. ábra): ciklusidő (cycle time).

Pl.: 500 MHz - 2 nsec.

Finomabb felbontás késleltetéssel.

Aszimmetrikus óra.



Memória: „Emlékszik” az utolsó beállításra.

Tároló: Szint vezérelt (level triggered).

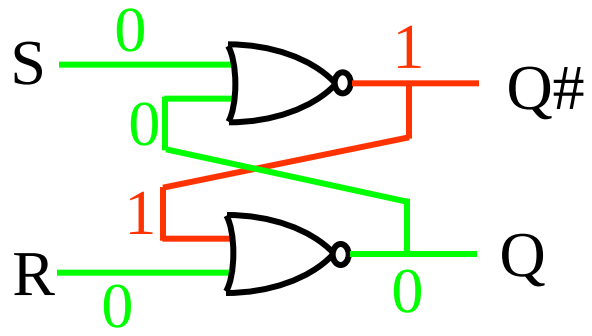
SR tároló (Set Reset latch, **3.22. ábra**).

Stabil állapot: a két kimenet **0**, **1** vagy **1**, **0**.

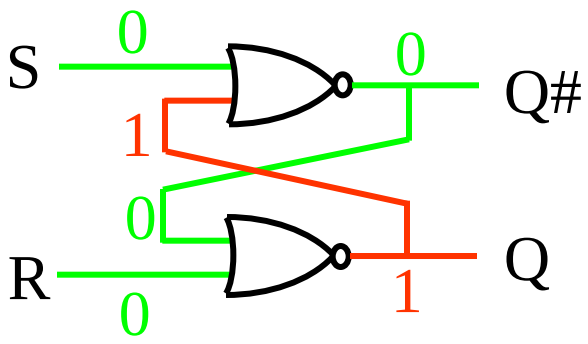
S (set), **R** (reset) bemenet. (Q# = Q)

A	B	NOR
0	0	1
0	1	0
1	0	0
1	1	0

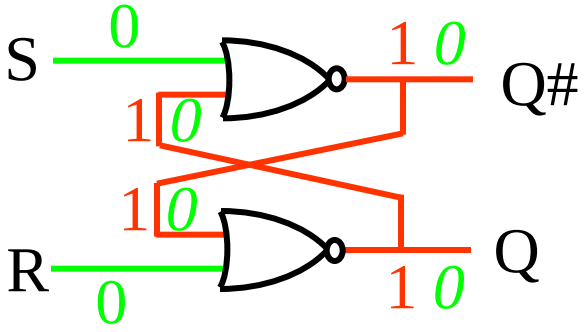
0 állapot:

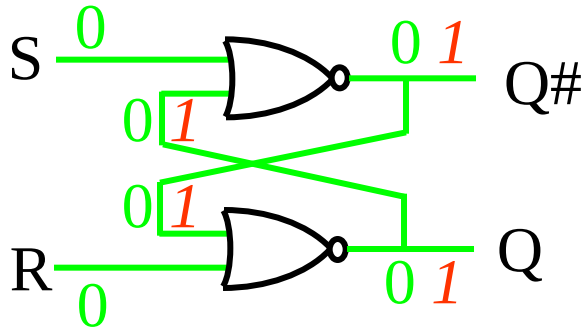


1 állapot:



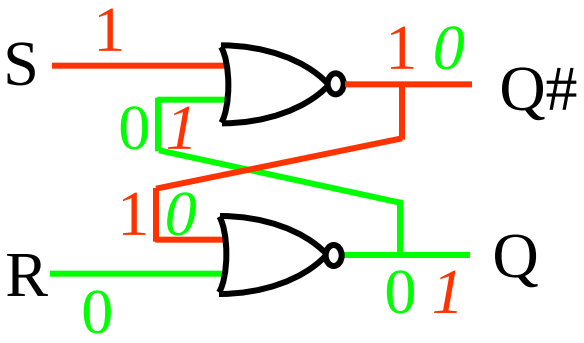
Nem stabil állapotok (pl. clock):





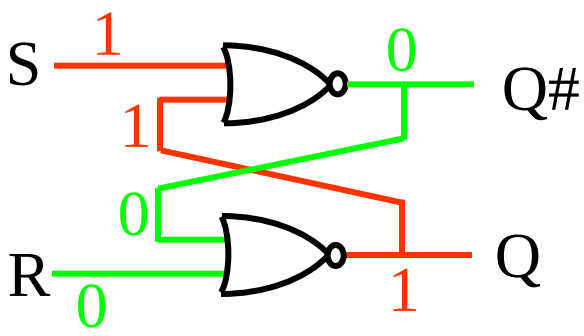
1-be állítás (Set):

0-ról:



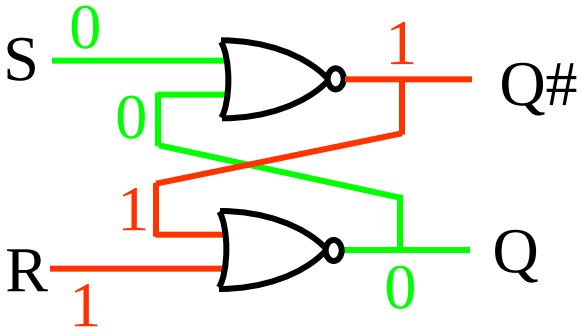
A	B	NOR
0	0	1
0	1	0
1	0	0
1	1	0

1-ről:

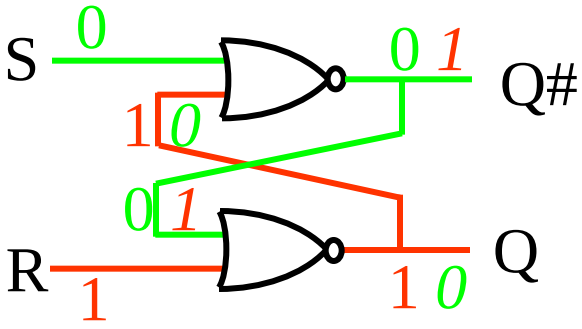


0-ba állítás (Reset):

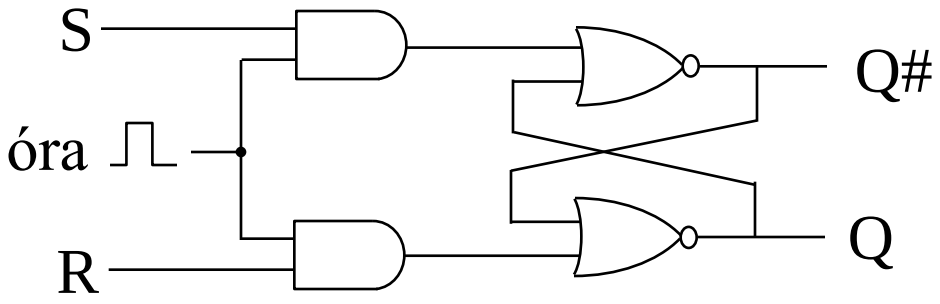
0-ról:



1-ről:

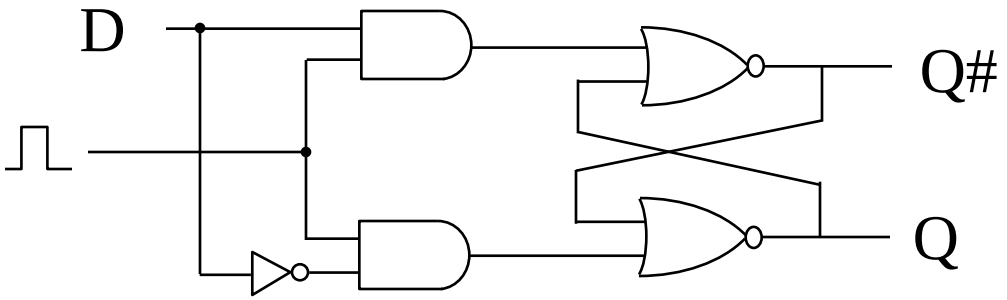


Időzített (clocked) **SR** tároló (**3.23. ábra**).

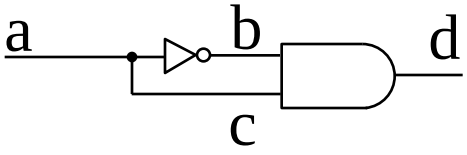


Mindkét **SR** tároló indeterminisztikussá válna,
ha **S = R = 1** egyszerre fordulna elő.

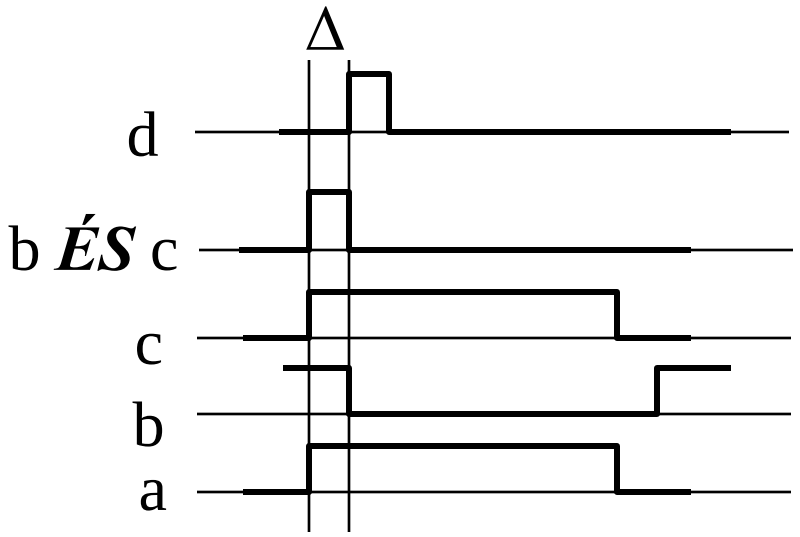
Megoldás: **Időzített D**-tároló (**3.24. ábra**).



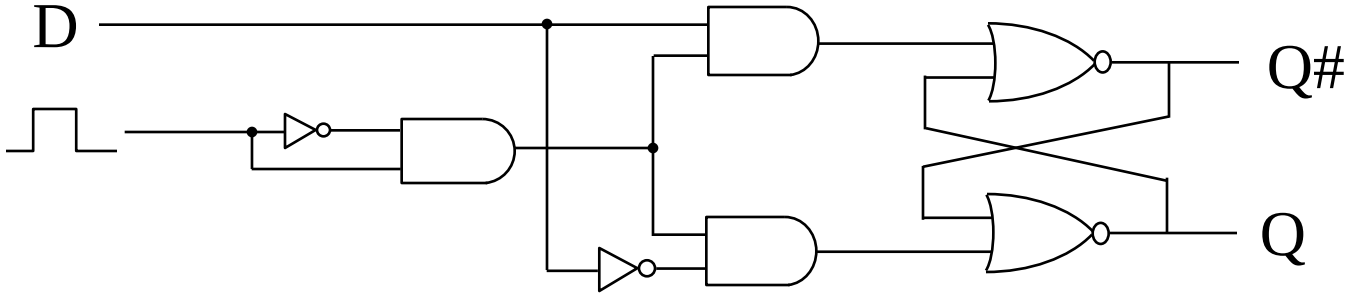
Pulzusgenerátor 3.25. ábra.



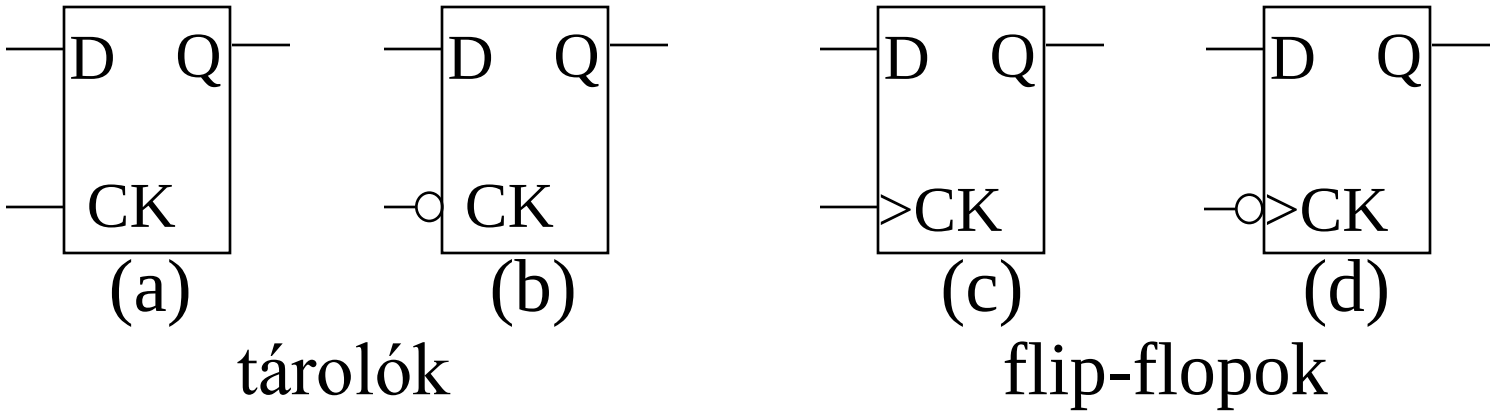
Az inverternek van egy pici (1-10 ns) késleltetése (Δ).



Flip-flop: élvezérelt (edge triggered), **D** flip-flop: **3.26. ábra**.



3.27. ábra: Tárolók és flip-flopok



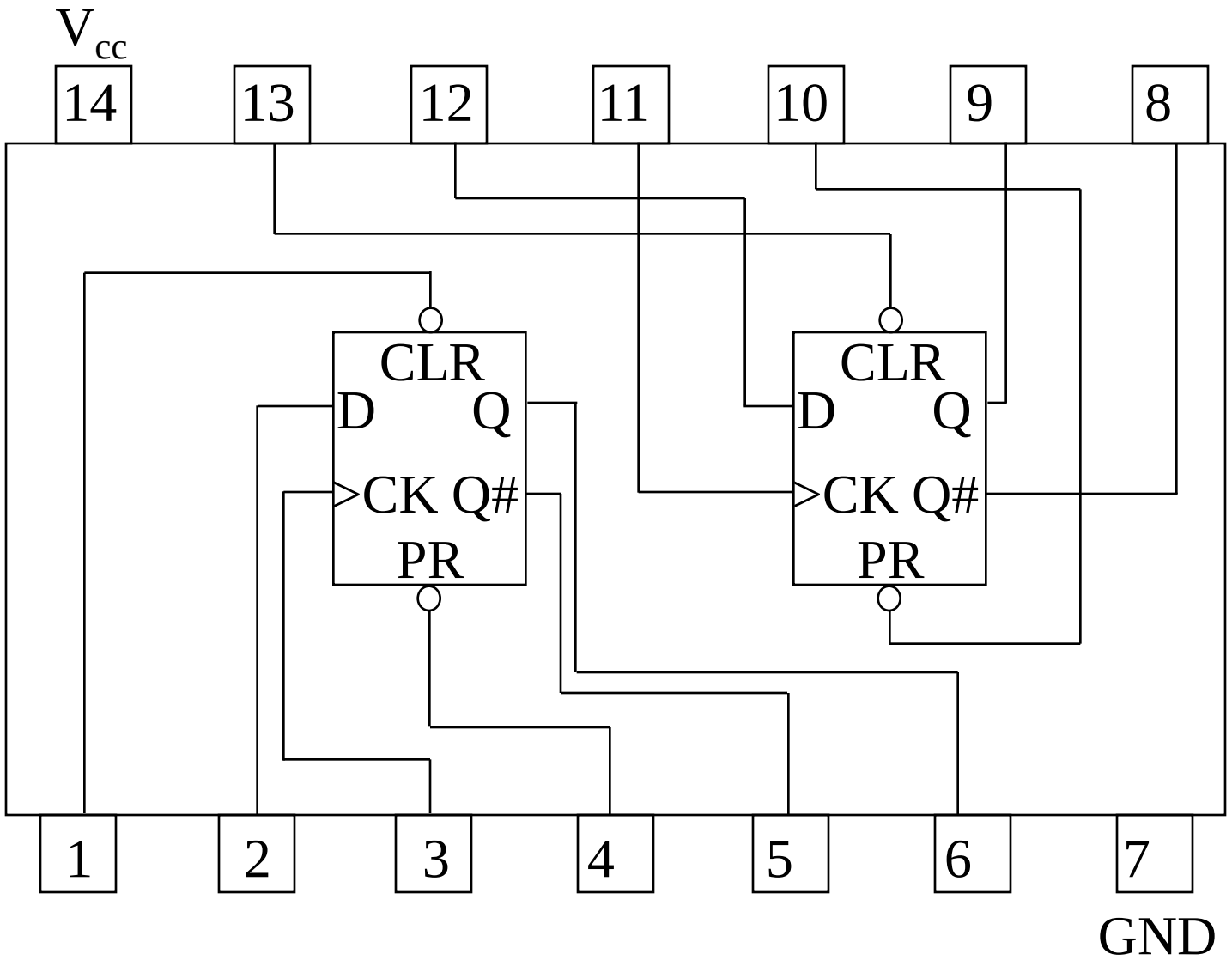
CK: órajel

(a) **CK=1**, (b) **CK=0** szint esetén írja be **D**-t,
(c) **CK** emelkedő, (d) **CK** lefelé menő élénél.

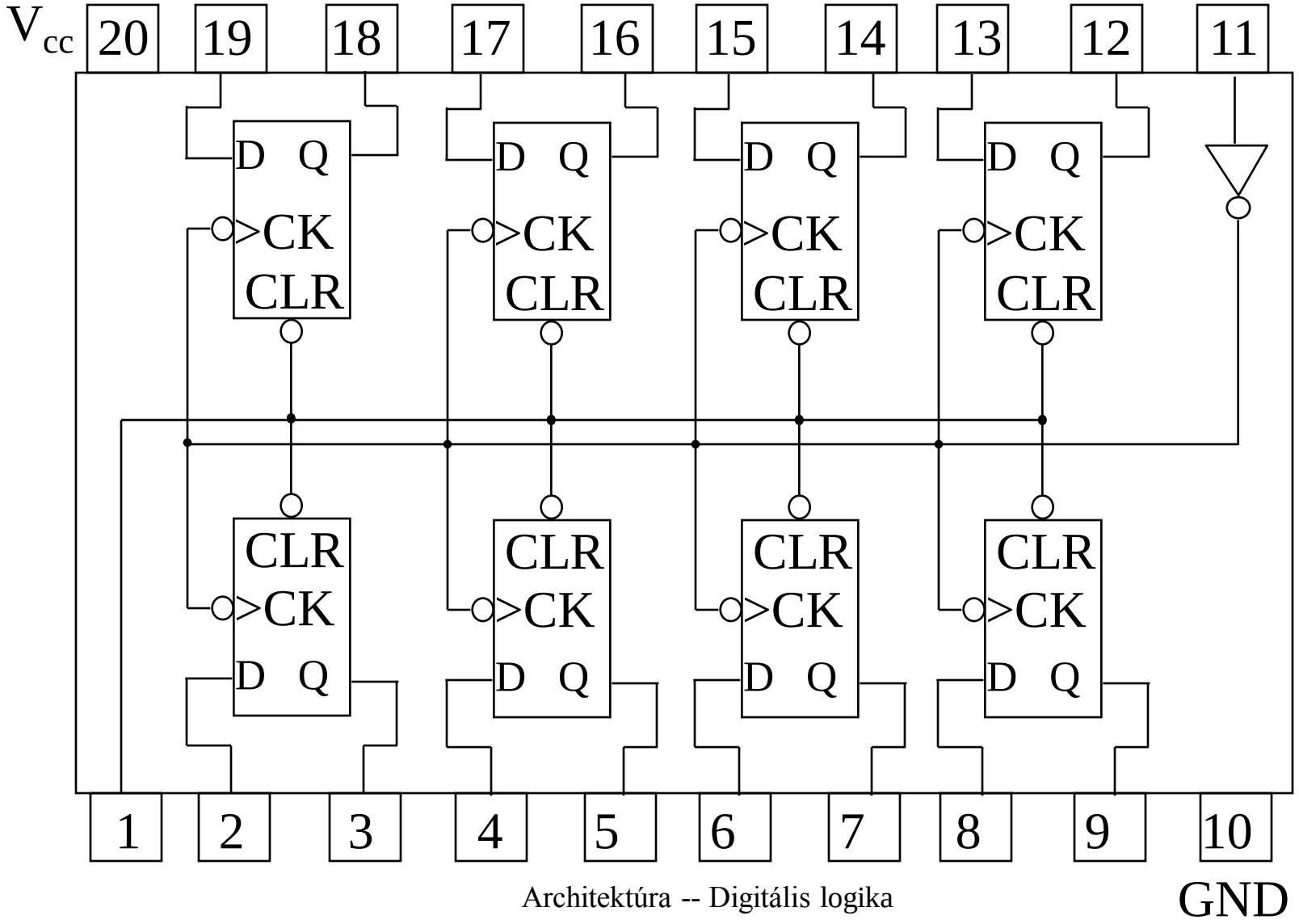
Sokszor **S** (set, **PR** preset), **R** (reset,**CLR** clear) bemenet,

illetve **Q#** kimenet is van.

3.28. ábra: (a) 2 független D flip-flop,



3.28. ábra: (b) közös **CK**-val és **CLR**-rel vezérelt 8 bites D flip-flop: (**regiszter**)



Memória szervezése

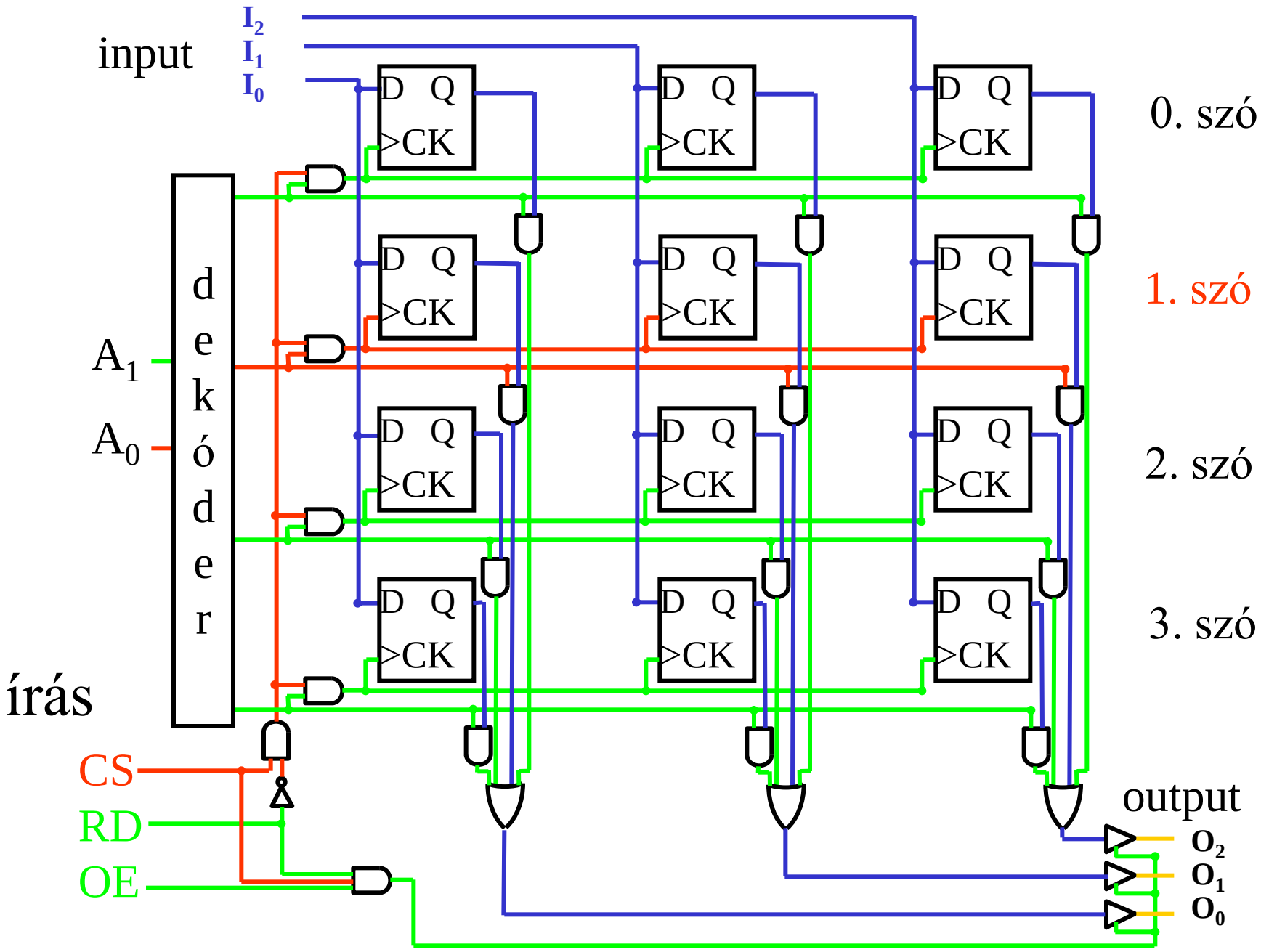
Elvárás: szavak címezhetősége.

3.29. ábra: Négy db három bites szó. Bemenetek:
három a vezérléshez,
- **CS** (Chip Select): lapka választás,
- **RD** (ReaD): 1: olvasás, 0: írás választása,
- **OE** (Output Enable): kimenet engedélyezése.

kettő a címzéshez (dekódoló),
három a bemenő adatoknak

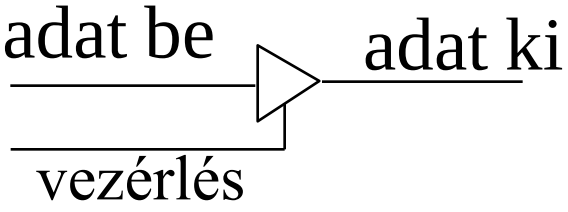
és

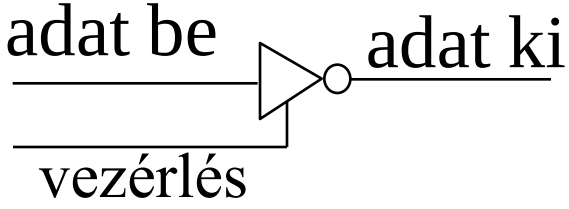
három adat kimenet.


input
I_2
I_1
I_0
D Q
>CK
0. szó
1. szó
2. szó
3. szó
A_1
A_0
dekódoder
írás
CS
RD
OE
output
O_2
O_1
O_0

Memória szervezése

Az igazi memóriáknál a bemenet és kimenet közös (kevesebb lábra van szükség): Nem invertáló és invertáló pufferek (ezek három állapotú eszközök, **tri-state device**, **3.30. ábra**).



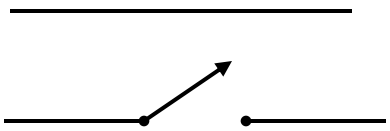


nem invertáló puffer

Ha a vezérlő jel

magas

alacsony

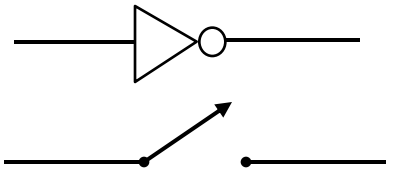


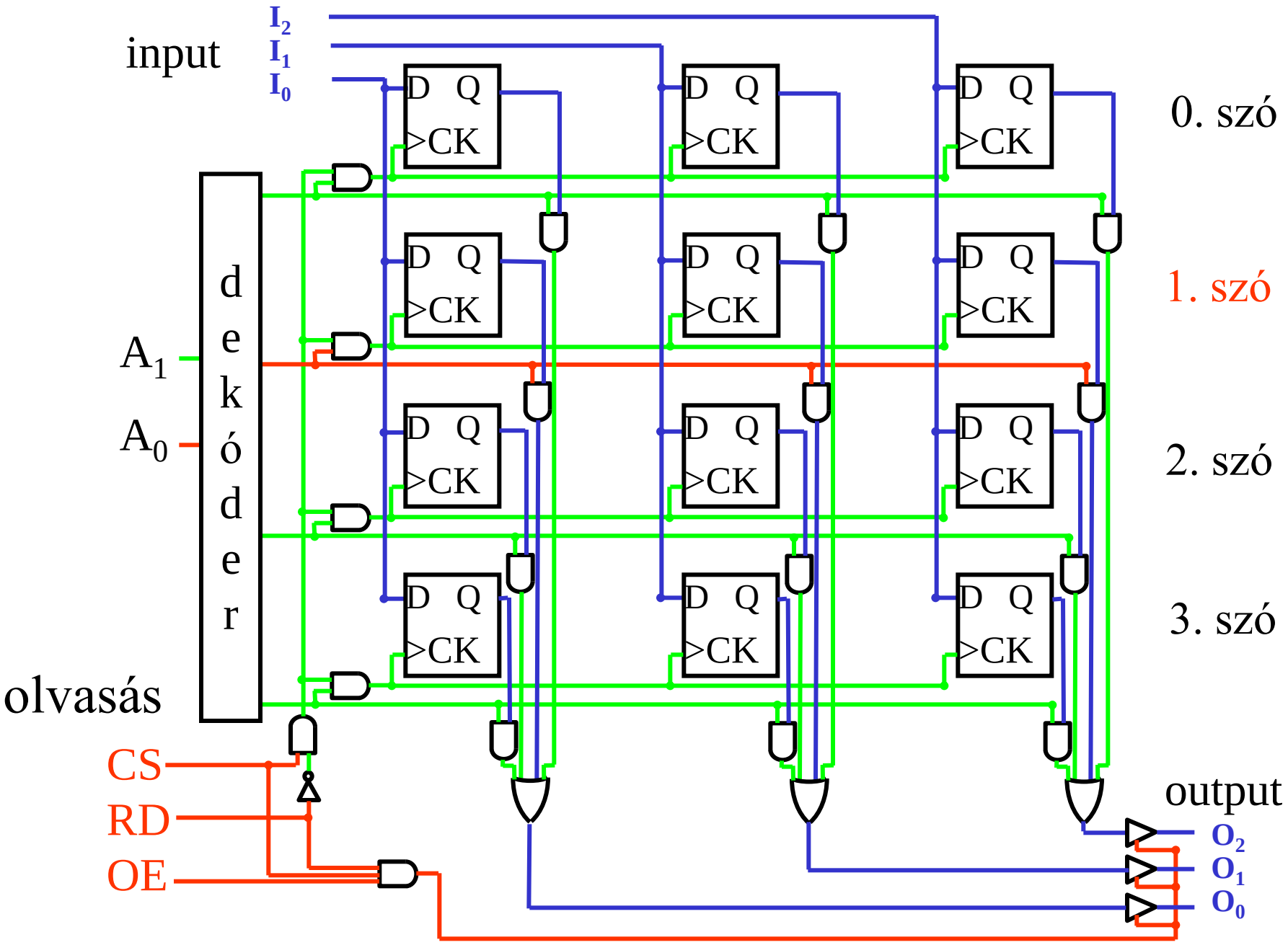
invertáló puffer

Ha a vezérlő jel

magas

alacsony

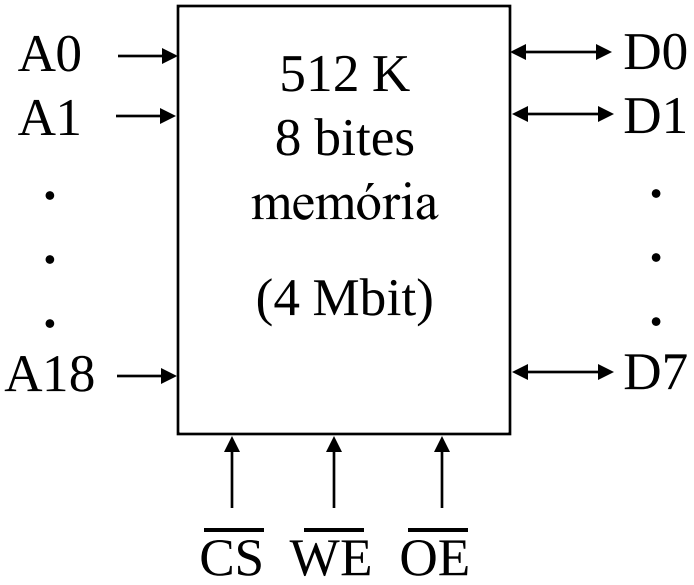



input
I_2
I_1
I_0
D Q
>CK
0. szó
1. szó
2. szó
3. szó
A_1
A_0
dekódoder
olvasás
CS
RD
OE
output
O_2
O_1
O_0

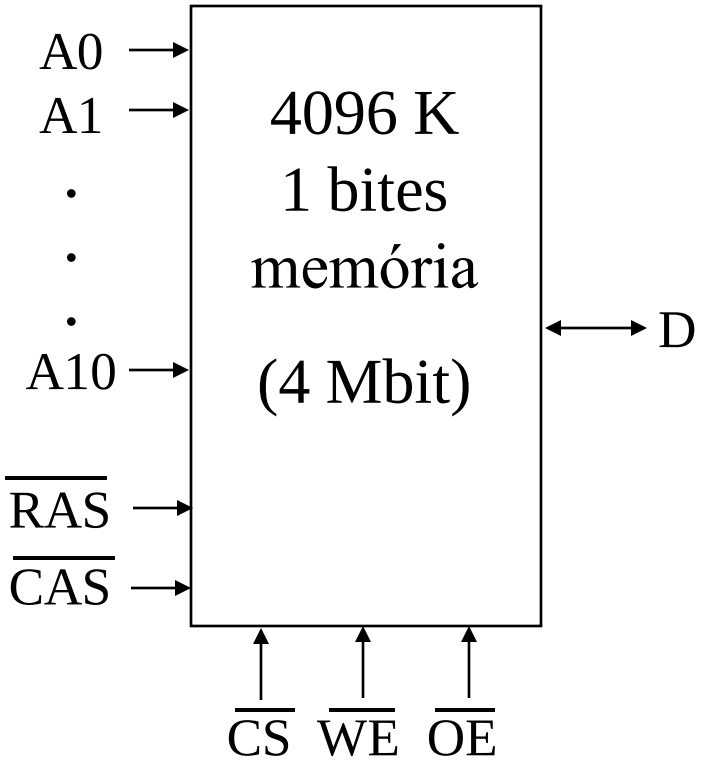
Memórialapkák

Előnyös, ha a szavak száma **2 hatvány**.

4 Mbit-es memória kétféle szervezése: **3.31. ábra.**



19 cím, 8 adat vonal



11 cím, 1 adat vonal

Row **A**ddress **S**trobe

Column **A**ddress **S**trobe

Memórialapkák

A jel (bemenet) **beállított** (asserted) vagy **negált**.

CS beállított: **1**, de **CS#** beállított: **0**

a) **512 K** bájtos elrendezés: **19** cím, **8** adat vonal.

b) **2048*2048** bites elrendezés: **11** cím, **1** adat vonal: Bit kiválasztás sor- (**RAS**: Row Address Strobe) és oszlopindex **CAS** (Column ...) segítségével. Gyakran alkalmazzák nagyobb memóriáknál, bár a két cím megadása lassíthat.

Nagyobb memóriáknál **1**, **4**, **8**, **16** bites kimeneteket is használnak.

RAM (Random Access Memory)

- **Statikus RAM** (**SRAM**). **D** flip-flop elemekből épül fel. Amíg áram alatt van, tartja a tartalmát.
  Elérési idő: néhány nsec (cache-nek jók).
- **Dinamikus RAM** (**DRAM**): minden bit egy tranzisztor és egy kondenzátor: néhány msec-onként frissíteni kell, de nagyobb adatsűrűség érhető el. Elérési idő: néhány tíz nsec (főmemóriák).
  - régi: **FPM** (Fast Page Mode) sor-, oszlopcím.
  - újabb: **EDO** (Extended Data Output) lehet új memóriahivatkozás, mielőtt az előző befejeződik.
- **SDRAM** (Synchronous **DRAM**). A központi óra vezérli. Blokkos átvitel.
  Újabban: **DDR** (Double Data Rate). Az órajel föl- és lefutó élénél is van adatátvitel.

ROM (Read-Only Memory)

ROM: gyárilag kialakított tartalom.

PROM (Programmable **ROM**): a tartalom biztosítékok kiégetésével alakul ki (a **PLA**-khoz hasonlóan, **3.15. ábra**).

EPROM (**Erasable PROM**): a biztosítékok speciális fénnyel kiolvaszthatók és „kijavíthatók”.

EEPROM: elektromos impulzusokkal.

Flash memória: törlés és újraírás csak blokkonként. Kb. 100 000 használat után „elkopnak”.
Ilyen van a legtöbb MP3 lejátszóban, digitális fényképezőgépben …

Gyorsító tár (cache – 2.16. ábra)

A processzorok mindig gyorsabbak a memóriáknál.

A **CPU** lapkára integrálható memória gyors, de kicsi.

Feloldási lehetőség: a központi memória egy kis részét (gyorsító tár) a **CPU** lapkára helyezni: Amikor egy utasításnak adatra van szüksége, akkor először itt keresi, ha nincs itt, akkor a központi memóriában.

Lokalitási elv: Ha egy hivatkozás a memória ***A*** címére történik, akkor a következő valószínűleg valahol ***A*** közelében lesz (ciklus, mátrix manipulálás, …).

Ha ***A*** nincs a gyorsító tárban, akkor az ***A***-t tartalmazó (adott méretű) blokk (gyorsító sor - cache line) kerül beolvasásra a memóriából a gyorsító tárba.

Találati arány (***h***): az összes hivatkozás mekkora hányada szolgálható ki a gyorsító tárból.

Hiba arány: *1-h*.

Ha a gyorsító tár elérési ideje: ***c***,

a memória elérési ideje: ***m***, akkor az

átlagos elérési idő = ***c + (1- h) m***.

A gyorsító tár mérete: nagyobb tár – drágább.

A gyorsító sor mérete: nagyobb sor – nagyobb a sor betöltési ideje is. Ugyanakkora tárban kevesebb gyorsító sor fér el.

Osztott (külön utasítás és adat) **gyorsító tár előnyei:**

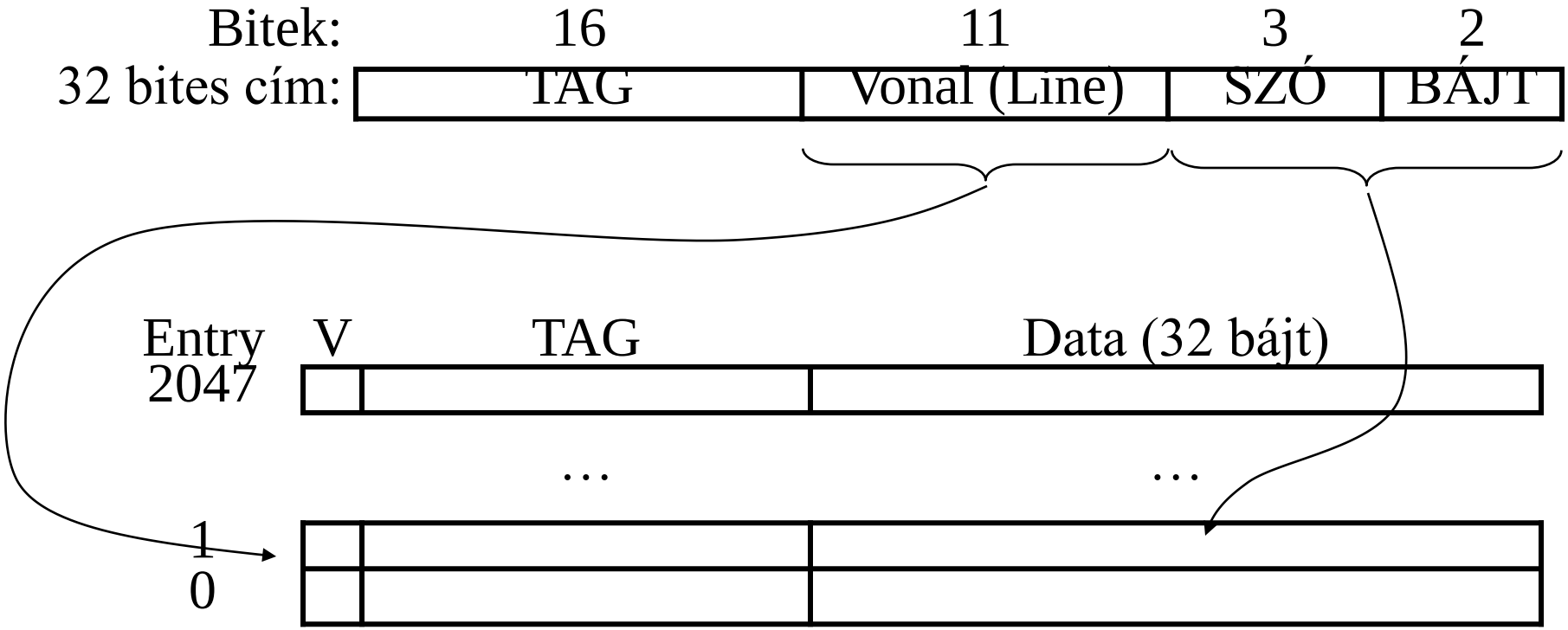
- Egyik szállítószalag végzi az utasítás, másik az operandus előolvasást.
- Az utasítás gyorsító tárat sohasem kell visszaírni (az utasítások nem módosulnak).

Egyesített gyorsító tár: nem lehetséges párhuzamosítás.

Hierarchia:

- elsődleges, a **CPU** lapkán,
- másodlagos, a **CPU**-val egy tokban,
- külön tokban.

Direkt leképezésű gyorsító tár működése: (4_38_abrahoz)



Ha a gyorsító tár **Vonal** által mutatott sorában **V=1** (valid), és a **TAG** megegyezik a címben lévő **TAG**-gel, akkor az adat bent van a gyorsító tárban (ebben a sorban).

Halmazkezelésű (csoportasszociatív) gyorsító tár

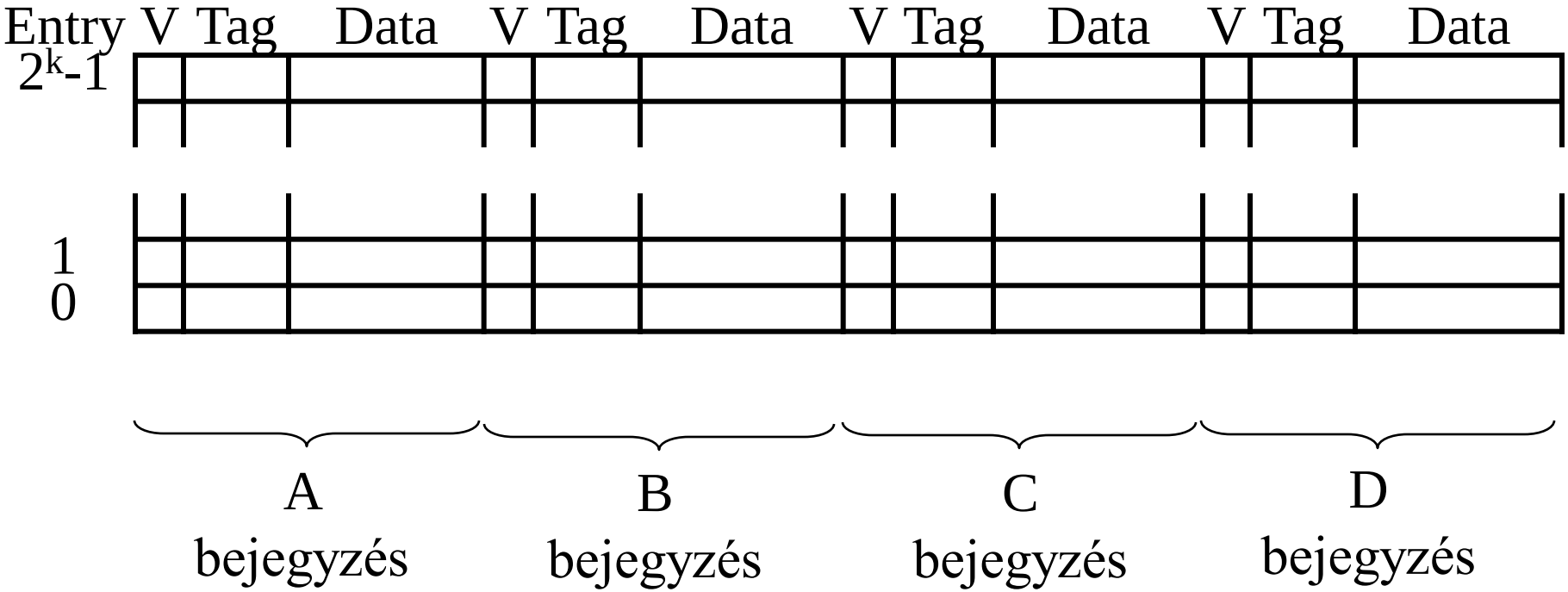
Ha egy program gyakran használ olyan szavakat, amelyek távol vannak egymástól, de ugyanoda képződnek le a gyorsító tárban, akkor sűrűn kell cserélni a gyorsító sort.

Ha minden címhez ***n*** bejegyzés van, akkor ***n*** utas halmazkeresésű gyorsító tárról beszélünk.

Ritka a több, mint 4 utas kezelés.

LRU (Least Recently Used) algoritmus:
gyorsító sor betöltése előtt a legrégebben használt bejegyzés kerül ki a gyorsító tárból.

Halmaz kezelésű gyorsító tár (4.39. ábra)



Ha a gyorsító tár **Vonal** által mutatott sorában az **A**, **B**, **C** és **D** bejegyzések egyikében **V=1** (valid), és a hozzá tartozó **TAG** megegyezik a címben lévő **TAG**-gel, akkor az adat bent van a gyorsító tárban (ebben a bejegyzésben).

Memóriába írás

Stratégiák:

Írás áteresztés (**write through**): az írás a memóriába történik. Ha a cím a gyorsítóban van, oda is be kell írni, különben el kellene dobni a gyorsító sort.

Késleltetett írás (**write deferred**, **write back**): ha a cím bent van a gyorsító tárban, akkor csak a gyorsító tárba írunk, a memóriába csak gyorsító sor cserénél.

Ha a cím nincs a gyorsító tárban, akkor előtte betölthetjük: **írás allokálás** (**write allocation**) – többnyire ezt alkalmazzák késleltetett írás esetén.

Memória hierarchia (2.18. ábra)

Elérési idő:

néhány nanosec

>100 msec

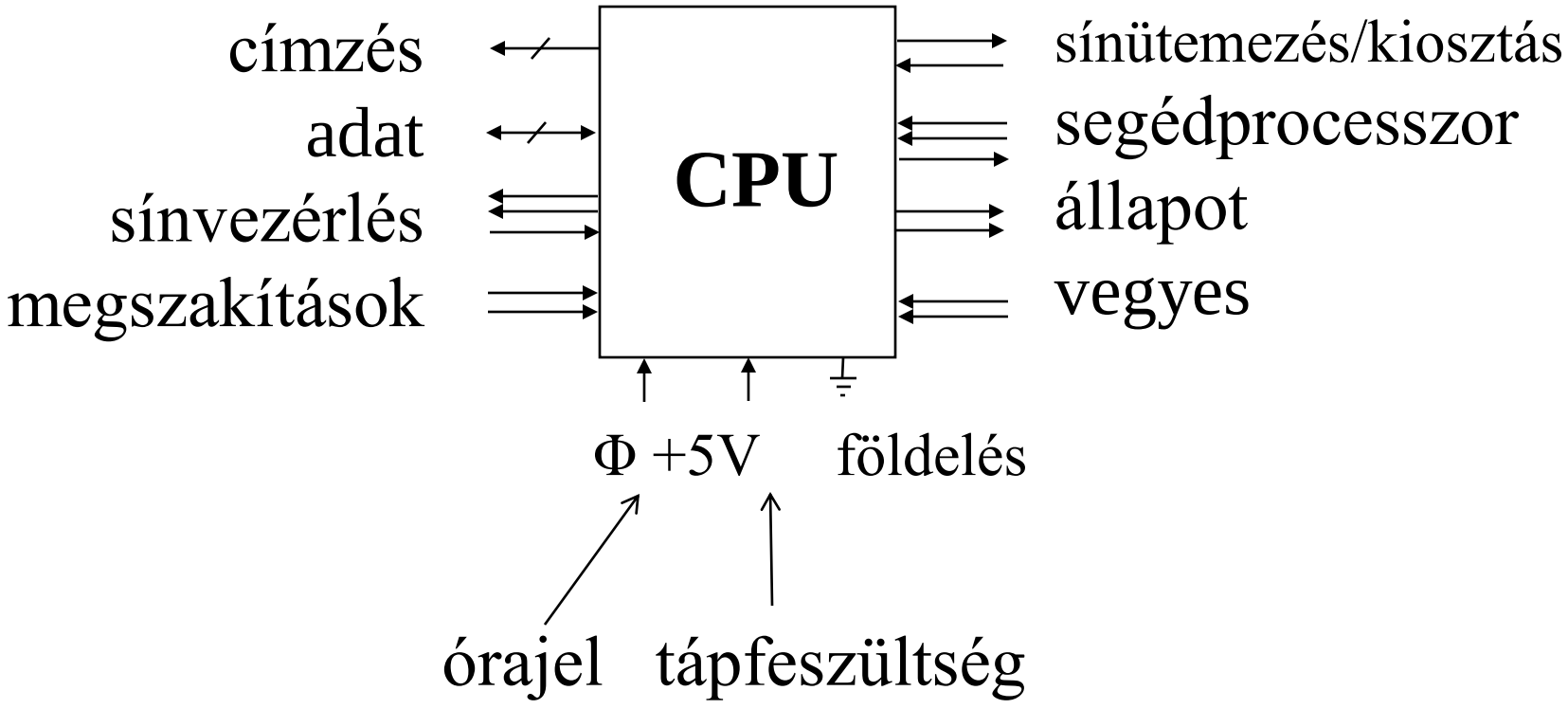
Kapacitás:

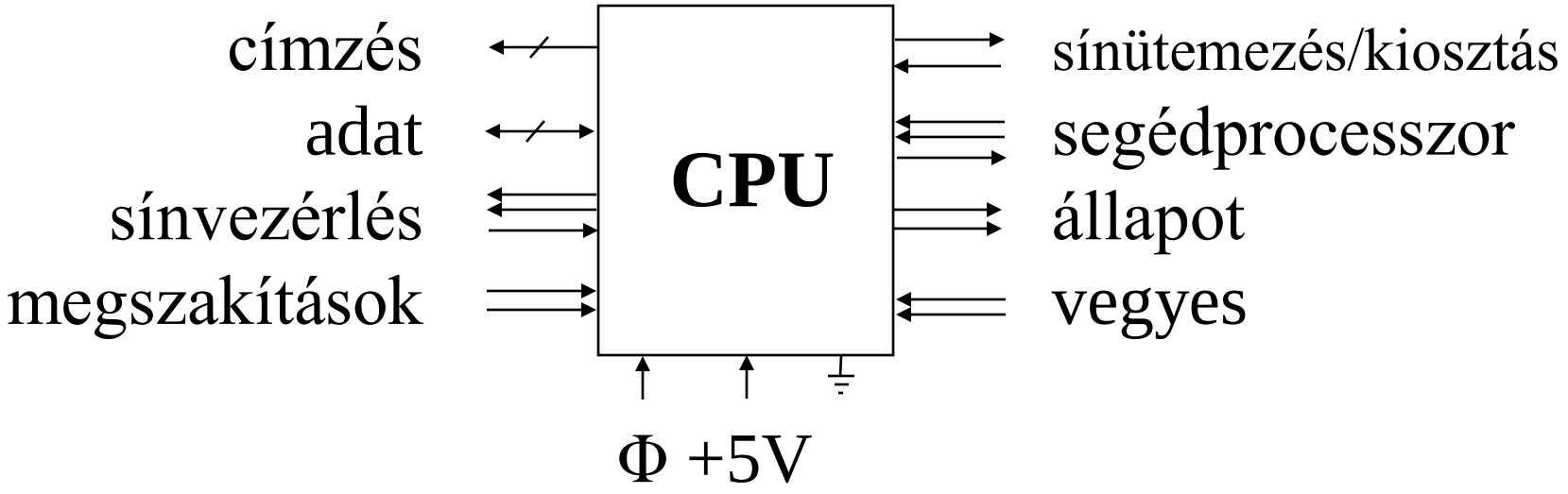
néhány bájt

néhány száz GB

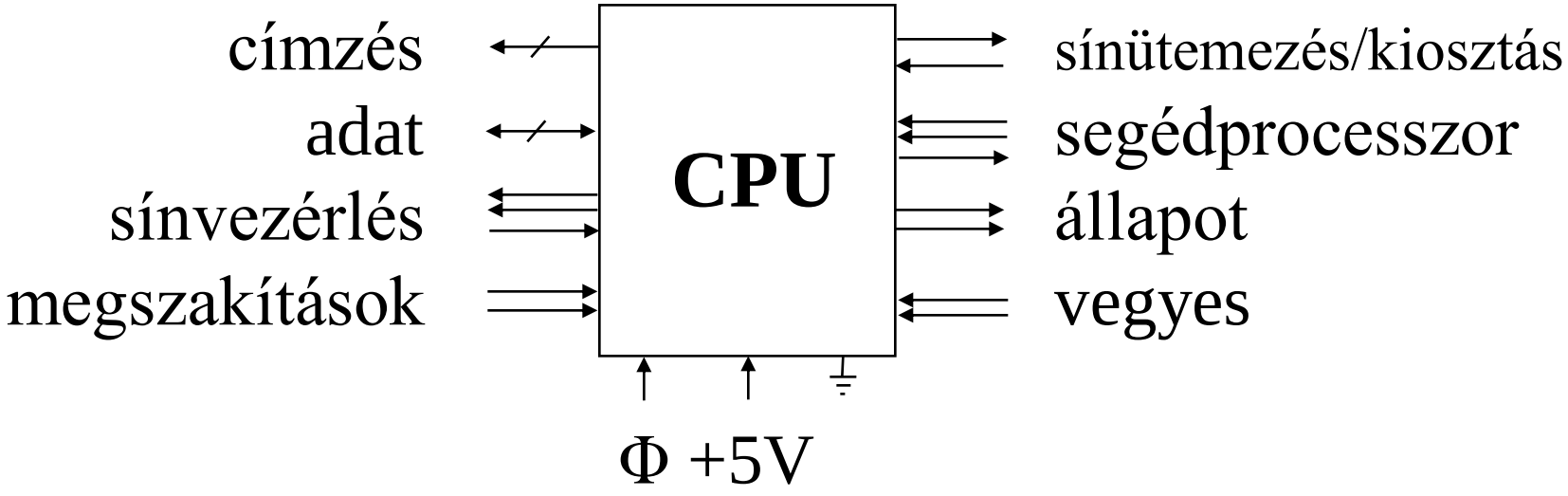
CPU (Central Processing Unit)

Általában egyetlen lapkán van. **Láb**akon keresztül kommunikál a többi egységgel (**3.34. ábra**).



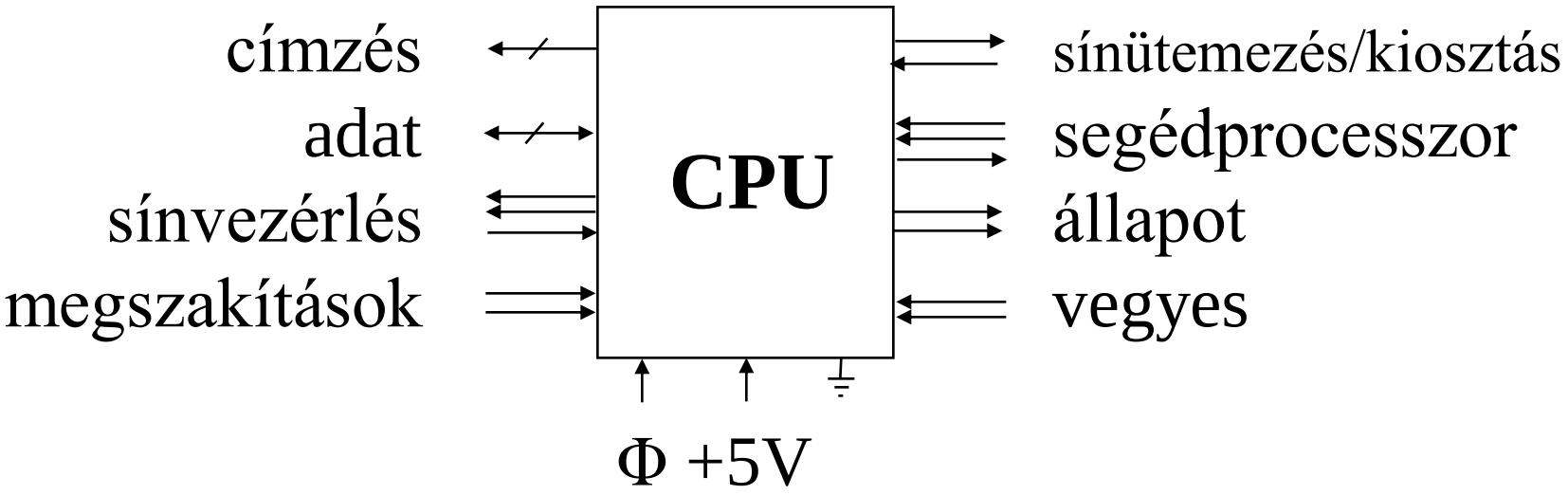


Lábak (**pins**) három típusa: **cím**, **adat**, **vezérlés**. Ezek párhuzamos vezetékeken, az un. **sín**en keresztül kapcsolódnak a **memória**, az **I/O** egységek hasonló lábaihoz.



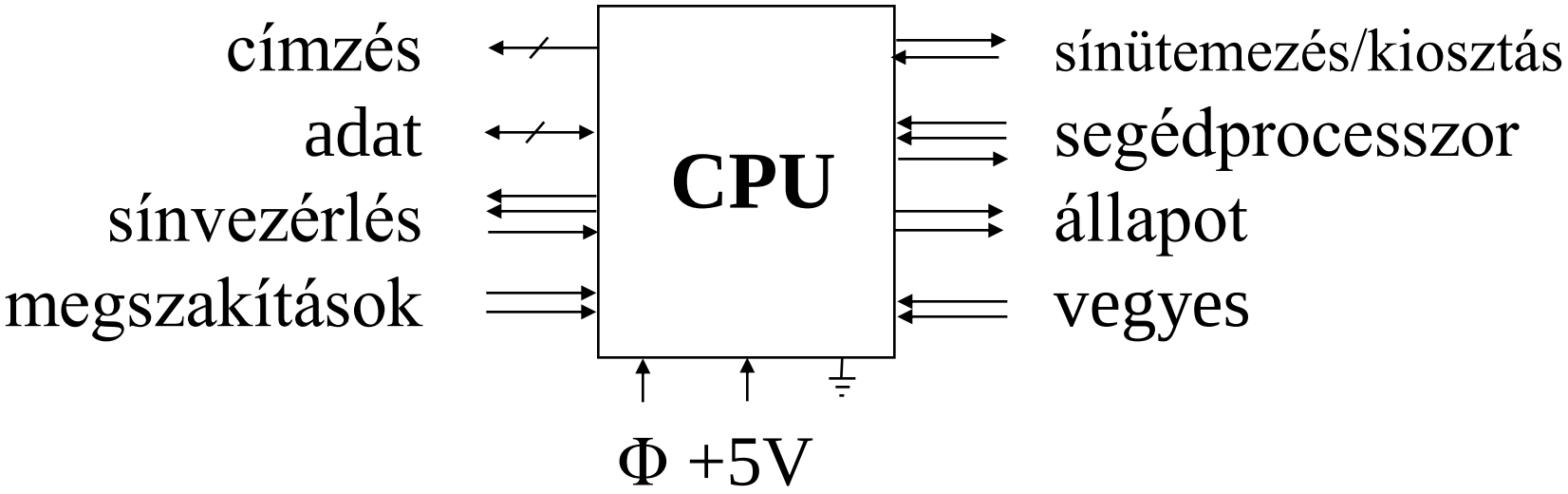
Lényeges a cím- és adatlábak száma (**3.34. ábra**):

- Ha *m* címláb van, akkor 2^m memóriarekesz érhető el (tipikus *m* = 16, 20, 32, 64).
- Ha *n* adatláb van, akkor egyszerre *n* bit olvasható illetve írható (tipikus *n* = 8, 16, 32, 36, 64).



Óra, áram (3.3 v. 5V), föld, továbbá **vezérlőlábak**:

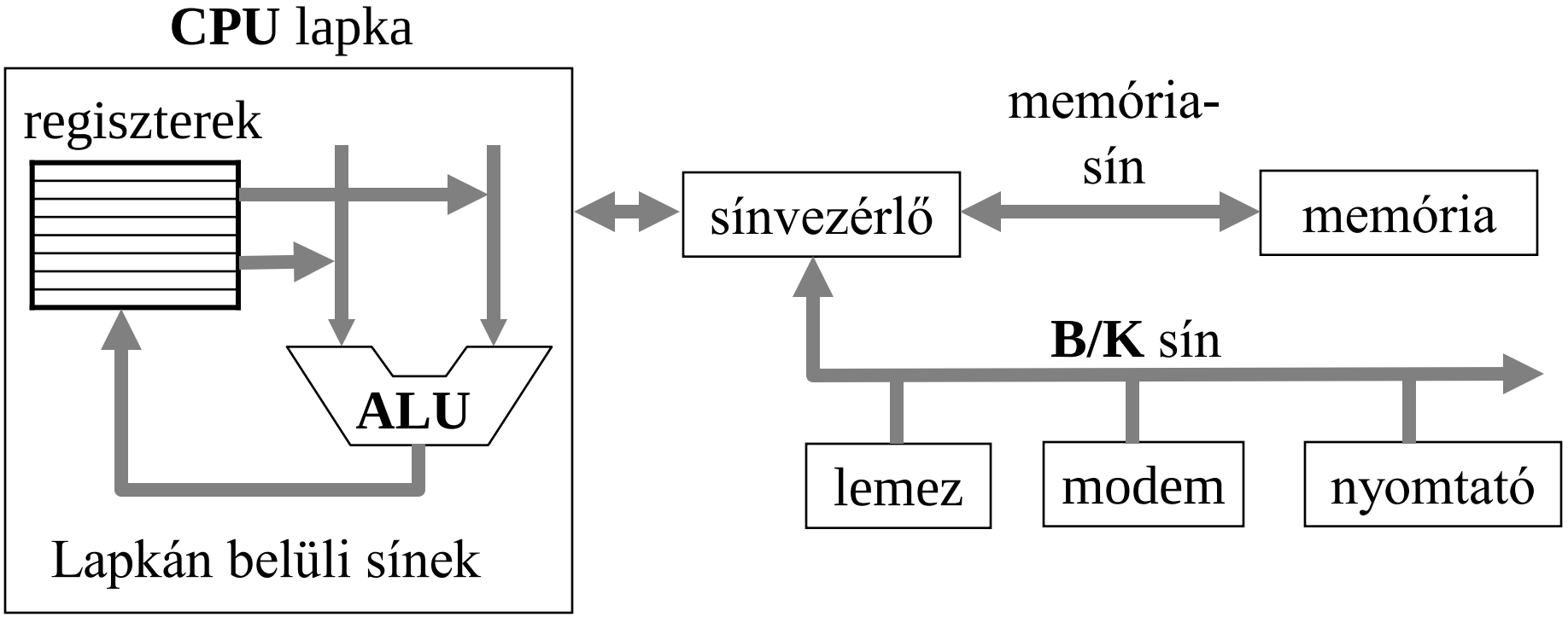
- sín vezérlés (bus control): mit csináljon a sín,
- megszakítások,
- sín kiosztás (ütemezés, egyeztetés – bus arbitration): kinek dolgozzon a sín,
- segéd processzor vezérlése, jelzései,
- állapot,
- egyebek.



Pl. **utasítás betöltése**:

- A **CPU** kéri a sín használat jogát,
- Az utasítás címét a cím lábakra teszi,
- vezérlő vonalon informálja a memóriát, hogy olvasni szeretne,
- a memória a kért szót az adat vonalakra teszi, kész jelzést tesz egy vezérlő vonalra,
- a **CPU** végrehajtáshoz átveszi az utasítást.

Sín (bus): Korai személyi számítógépeknél egyetlen (külső) rendszersín, manapság legalább kettő van: egy belső és egy külső (**I/O**), **3.35. ábra**.



Sínprotokoll: a sín működésének + a csatlakozások mechanikai, elektronikus definíciója

Mesterek (**masters**): aktív (kezdeményező) berendezések (**CPU**, lemez vezérlő).

Szolgák (slaves): passzív (végrehajtó) berendezések (lemez vezérlő, **CPU**), **3.35. ábra**.

Ez a szereposztás tranzakciónként eltérő lehet.

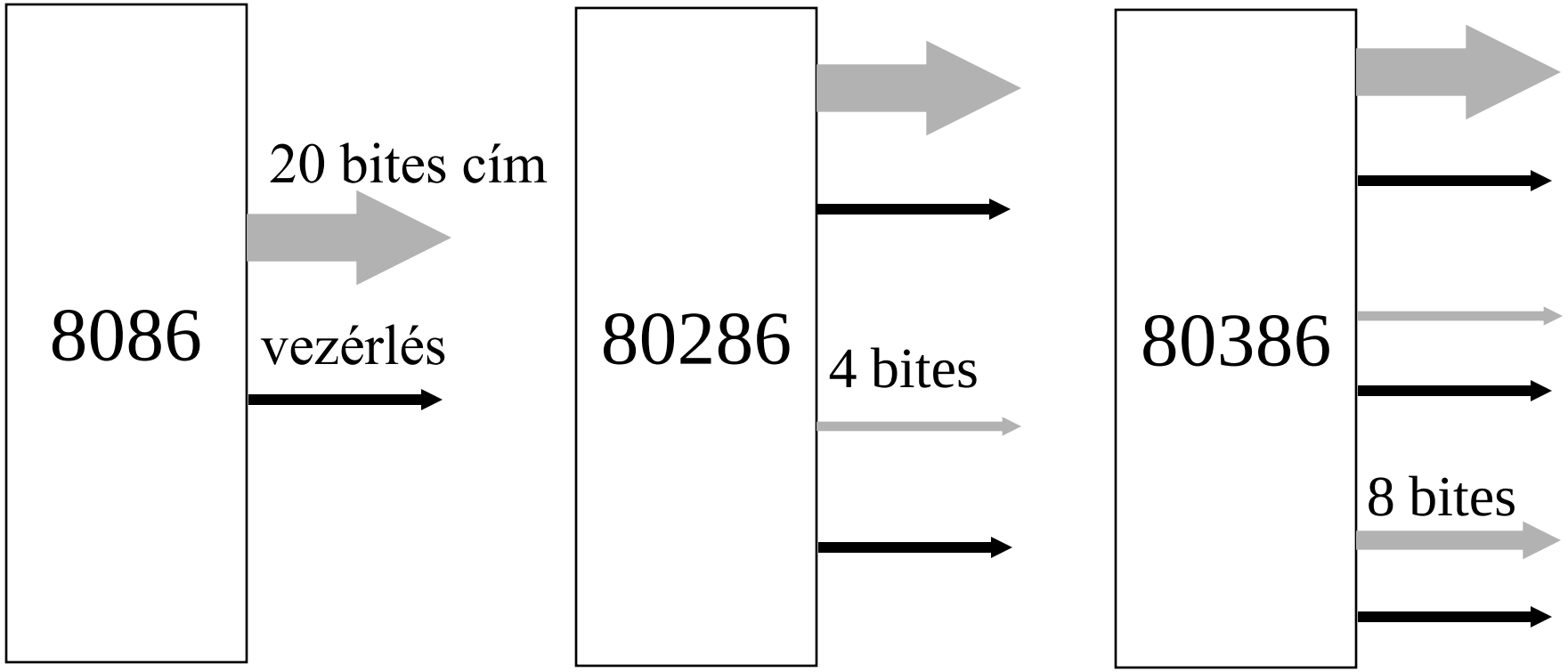
Mester	Szolga	példa
CPU	Segéd proc.	**CPU** felkínálja az utasítást
Segéd proc.	**CPU**	Segéd proc. kéri az operandusokat

A memória sohasem lehet mester!

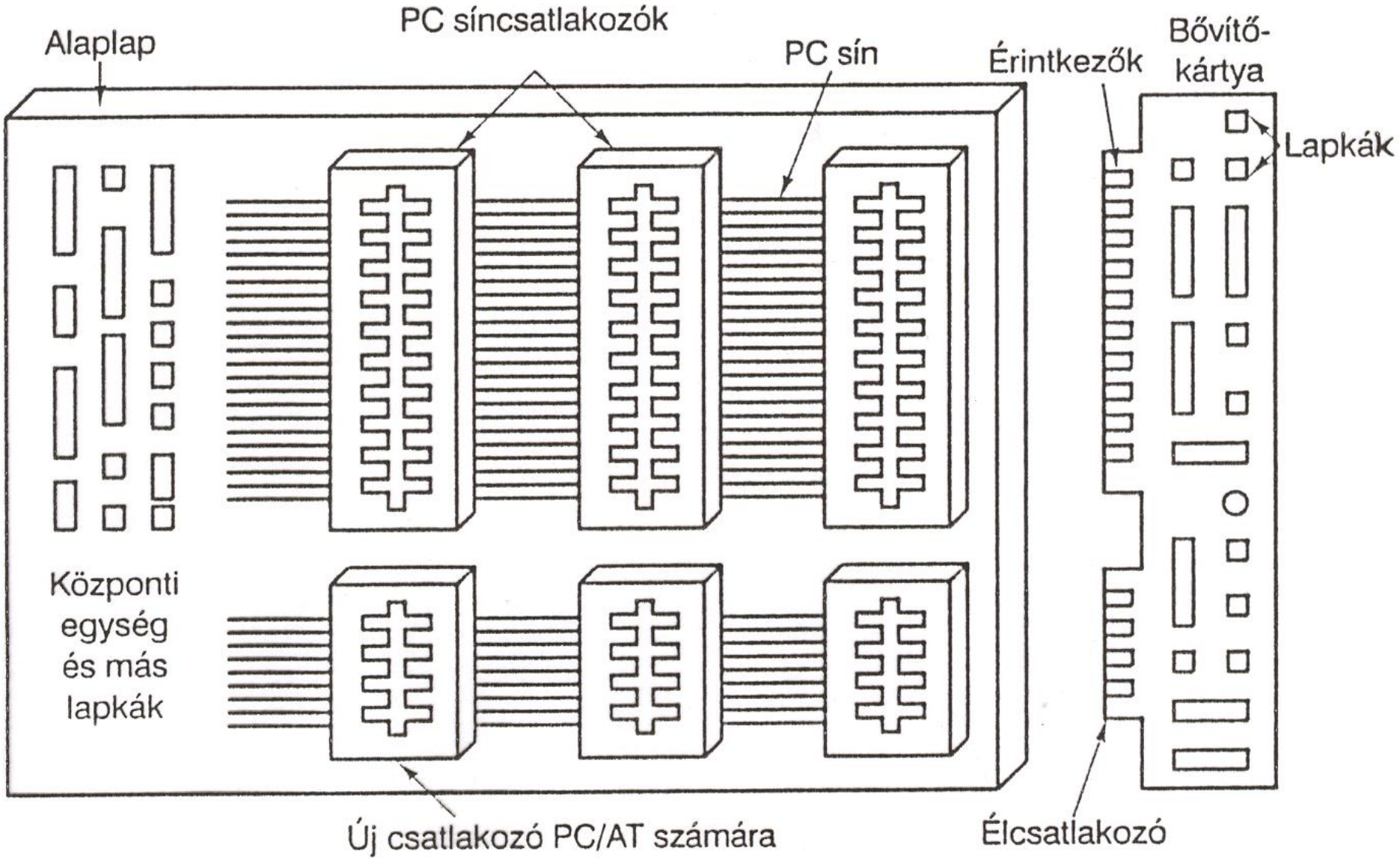
A sínhez kapcsolódó lapkák lényegében erősítők.
Mester – **sín vezérlő** (**bus driver**) – sín.
Sín – **sín vevő** (**bus receiver**) – szolga.
Mester–szolgáknál: **sín adó-vevő** (**bus transceiver**).
A csatlakozás gyakran **tri-state device** vagy **open collector** – **wired-OR** segítségével történik.

Sávszélesség: (továbbítható bitek száma) / sec.
Sávszélesség növelése:
Gyorsítás: probléma a sín aszimmetria (skew), kompatibilitás.
Sínszélesség: szélesebb sín → drágább, kompatibilitás.

Sínszélesség (pl. **IBM PC**: **3.37., 3.51. ábra**).



3.37. ábra. A cím szélességének növekedése az elmúlt időszakban



3.51. ábra. *A PC/AT sín két komponense, az eredeti PC és az új rész*

Alaplap (motherboard, parentboard, **3.51. ábra**)

Rajta van a **CPU**, sín(ek), ezen illesztő helyek (slots) a memória és a **beviteli/kiviteli (Input/Output – I/O)** eszközök számára (**3.51., 2.28. ábra**).

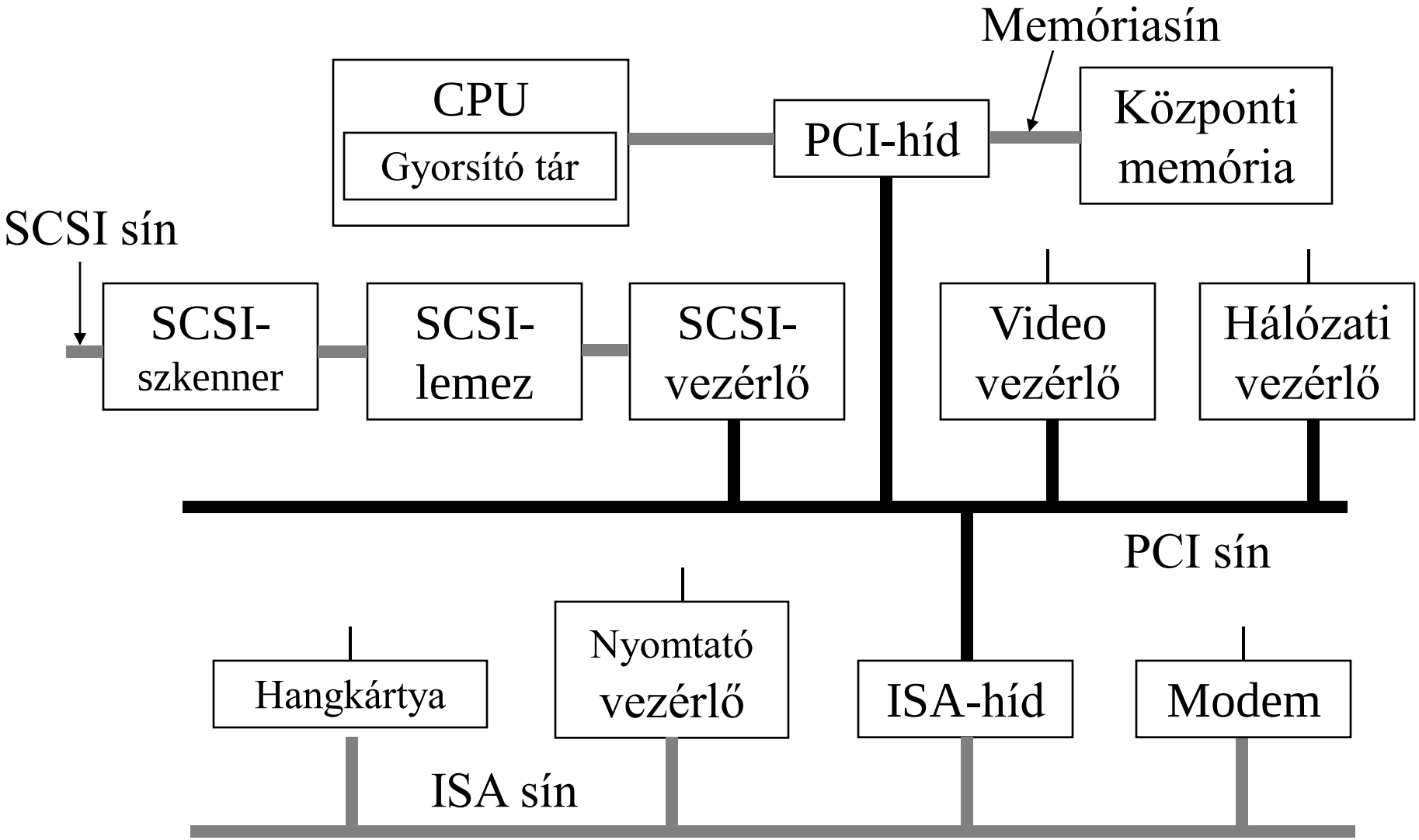
I/O eszköz: maga az eszköz + vezérlő (controller) külön kártyán vagy az alaplapon (**2.29. ábra**).

Gyorsabb **CPU** gyorsabb sínt igényel!

Kívánság: PC cseréjénél megmaradhasson a régi perifériák egy része: az új gépben is kell a régi sín!

Sínek szabványosítása.

Egy gépen belül több sín is használható: **2.30. ábra**.



2.30. ábra. *Egy tipikus modern PC PCI, SCSI és ISA sínnel*

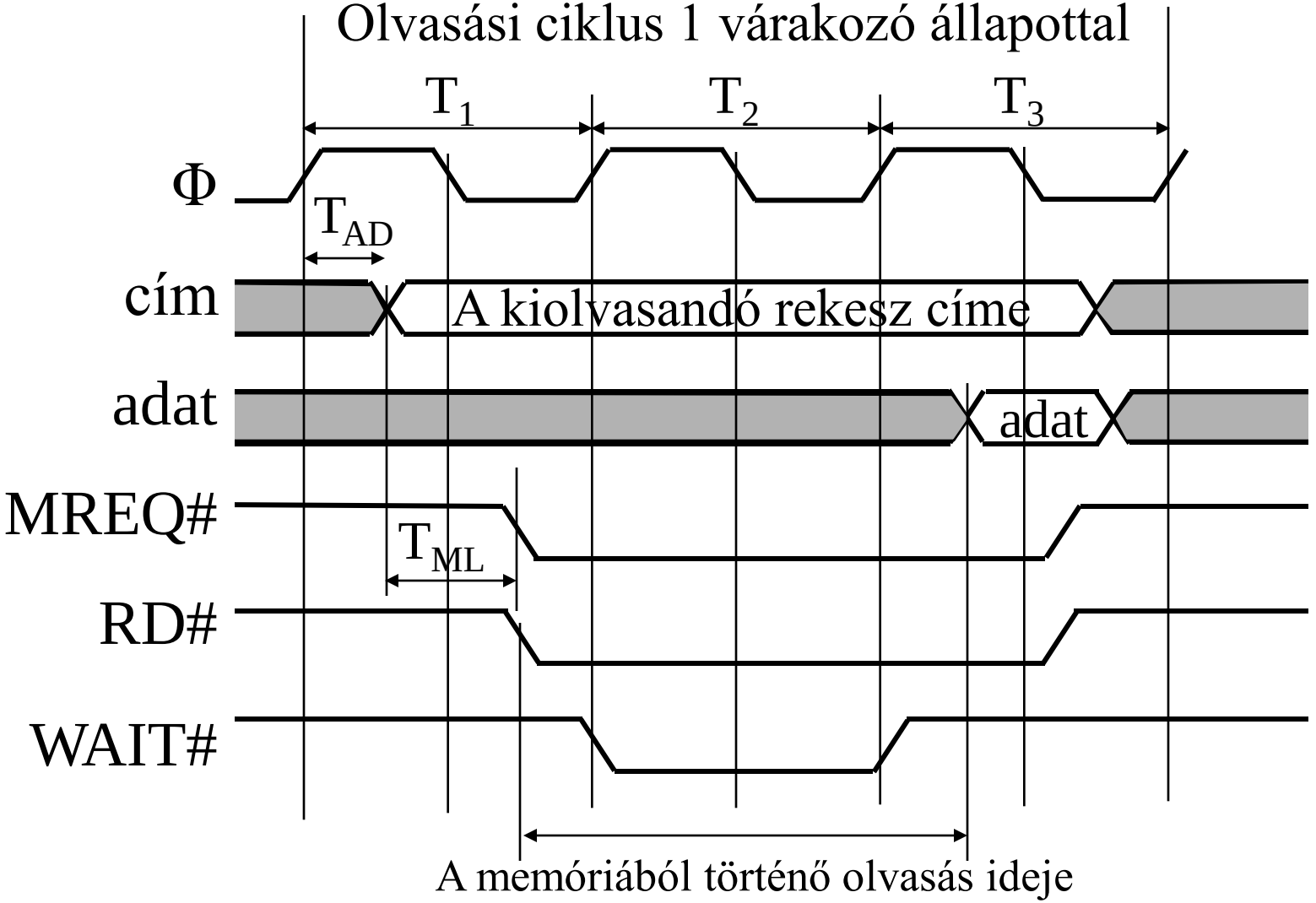
Sokszorozott (multiplexed) **sín**: pl. először a cím van a sínen, majd az adat (ugyanazokon a vezetékeken). Ilyenkor a sín szélesség lényegesen csökken (olcsóbb, kevesebb láb szükséges a sínhez való csatlakozáshoz), csökken a sáv szélesség is, de nem olyan mértékben.
Általában bonyolultabb a sín protokoll.

Sínek időzítése

Szinkron sín: 5 – 100 MHz-es órajel van a sín egy vezetékén. Minden síntevékenység az órajelhez van igazítva.

Síntevékenységek: cím megadása, vezérlőjelek (**MREQ#**, **RD#**, **WAIT#**), adat megérkezése, …
(3.38. ábra)

Jelölés	Tevékenység	min	max	idő
T_{AD}	Cím megérkezési ideje a sínre		11	ns
T_{ML}	Cím a sínen van MREC# előtt	6		ns
…	…	…	…	…



Kicsit hosszabb válasz idő esetén
még egy várakozó ciklusra lenne szükség.

Minden sínművelet a ciklusidő (sín ciklus) egész számú többszöröséig tart:
pl. 2.1 ciklusidő helyett 3 ciklusidő kell.

A leglassabb eszközhöz kell a sín sebességét igazítani, a gyors eszköz is lassan fog működni.

Aszinkron sín:

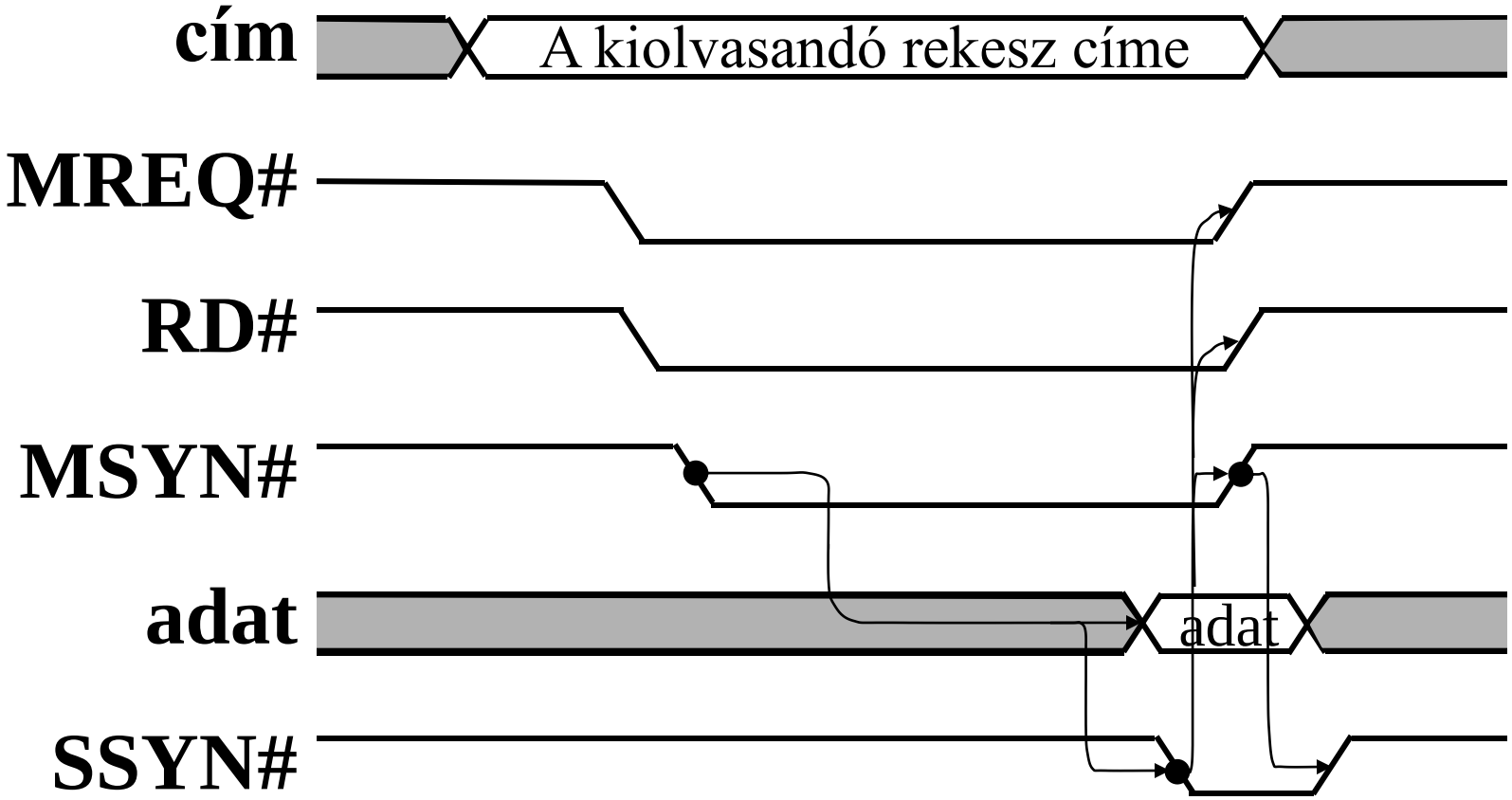
Minden eseményt egy előző esemény okoz!
Nincs órajel, **WAIT**.

MSYN# (kérés - Master SYNchronization),
SSYN# (kész - Slave SYNchronization).

Ugyanazon a sínen gyors és lassú mester - szolga pár is lehet.

Aszinkron sín működése (3.39. ábra)

Akkor indulhat újabb tranzakció, ha **SSYN#** negált.



Ugyanazon a sínen gyors és lassú mester - szolga pár is lehet.

Teljes kézfogás (full handshake):

Akkor indulhat, ha **SSYN#** negált!

- Mester: kívánságok beállítása, majd **MSYN#**, vár,
- Szolga: látja **MSYN#**-t: dolgozik, majd **SSYN#**, vár,
- Mester: látja **SSYN#** -t (kész), dolgozik, ha kell, majd negálja **MSYN#** -t,
- Szolga: látja **MSYN#** negálását, negálja **SSYN#** -t.

Ugyanazon a sínen gyors és lassú mester - szolga pár is lehet.

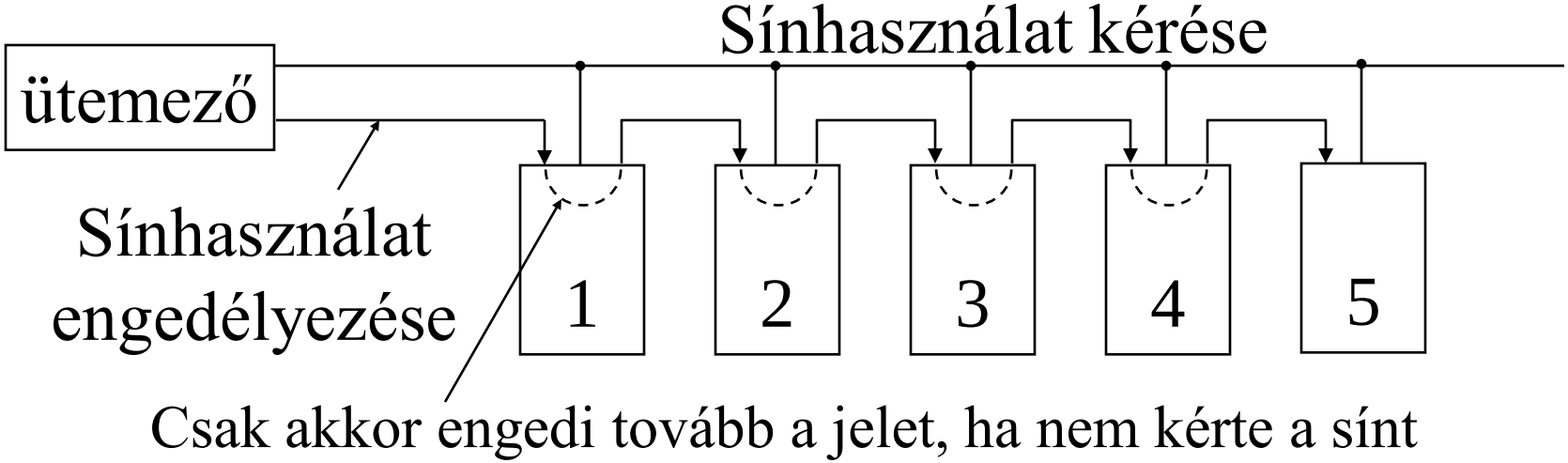
Sínütemezés (kiosztás)

Ha egyszerre többen is igénylik a sínt (**CPU**, **I/O** vezérlő), akkor a **sínütemező** (bus arbiter) dönt.

Általában **I/O** elsőbbséget kap (ciklusloPás).

Sínütemezés (kiosztás – bus arbitration)

- Centralizált (**3.40. (a) ábra**): (margaréta) láncolás (daisy chaining), egy vagy többszintű lehet.



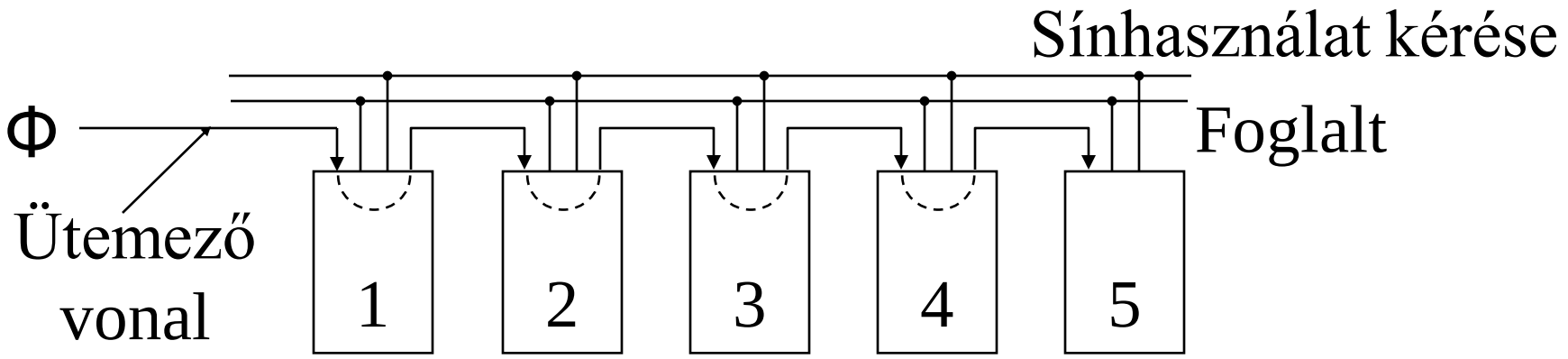
Ha van ***kérés*** és a ***sín szabad***: ***sín foglalási engedély***.

Néha további vezeték van az engedély fogadásának jelzésére (újabb sín kérés kezdődhet a sín használata közben).

- decentralizált

  - pl. 16 prioritású: 16 eszközhöz 16 kérés vonal, minden eszköz minden kérés vonalat figyel, tudja, hogy a saját kérése volt-e a legmagasabb prioritású.

  - **3.41. ábra**: ha nem ***foglalt*** és ***be***, akkor lefoglalhatja a sínt (***ki*** negálása, ***foglalt*** beállítása).



Sín műveletek

Az eddigiek **közönséges sín műveletek** voltak.

Blokkos átvitel (3.42. ábra): A kezdő memória címen kívül az adat sínre kell tenni a mozgatandó adatok számát. Esetleges várakozó ciklusok után ciklusonként egy adat mozgatása történik.

Megszakítás kezelés: később tárgyaljuk részletesen.

Több processzoros rendszerekben:
olvasás – módosítás – visszaírás ciklus: szemafor.

Példák sínekre

Az első **IBM PC** (**3.37. ábra**) 62 vonalas (vezeték, line), 20 címnek, 8 adatnak + **DMA**, megszakítás …

PC/AT szinkron sín (**3.51. ábra**): további 36 vezeték (címnek összesen 24, adatnak 16, …).

Microchannel (**IBM OS/2** gépekhez), szabadalmak

ISA (Industry Standard Architecture) lényegében 8.33 MHz-es **PC/AT** sín (sávszélesség: 16.7 MB/s).

EISA (Extended **ISA**) 32 bitesre bővített **ISA** (sávszélesség: 33.3 MB/s).

Színes TV-hez 135 MB/s sávszélesség kellene (1024*768 pixel, 3 bájt*2, 30 kép/sec).
lemez → memória → képernyő

PCI (Peripheral Component Interconnect): 32 bites adat átvitel (33,3 MHz, sávszélesség: 133 MB/s) szabadon felhasználható licensz.
Multiplexelt cím- és adatkivezetések.

Új változatai: 64 bites adat, 66 MHz, 528 MB/s.

Problémák:
- a memóriához lassú,
- nem kompatíbilis az **ISA** bővítőkártyákkal.

Megoldás (**3.52.** vagy **2.30. ábra**): több sín
Belső sín, **PCI** híd, **PCI** sín, **ISA** híd, **ISA** sín.

Általános soros sín (USB)

Igény: bármikor könnyen lehessen perifériát kapcsolni a géphez, ne kelljen szétszedni a gépet, újra boot-olni, ne kelljen áramellátásról gondoskodni, …
Plug 'n Play (csatlakoztasd és működik) perifériák.
Sokféle perifériát lehessen azonos módon csatlakoztatni, akár a gép működése közben, hardver ismeretek nélkül.

USB (Universal Serial Bus - általános soros sín):
Négy vezeték: adatok (2), tápfeszültség (1), föld (1).

USB 1.0 **1,5 Mbps** (billentyűzet, egér,…)

USB 1.1 **12 Mbps** (nyomtató, fényképezőgép,…)

USB 2.0 **480 Mbps** (DVD lejátszó,…)

A központi elosztó (**root hub**)
1 ms-onként üzenetekkel (**frame, 3.54. ábra**)
kommunikál az eszközökkel.

A frissen csatlakoztatott eszköz címe 0.
Ha a központi elosztó tudja fogadni az eszközt,
akkor egyedi címet (1-127) ad neki (**konfigurálja**).

Frame – keret

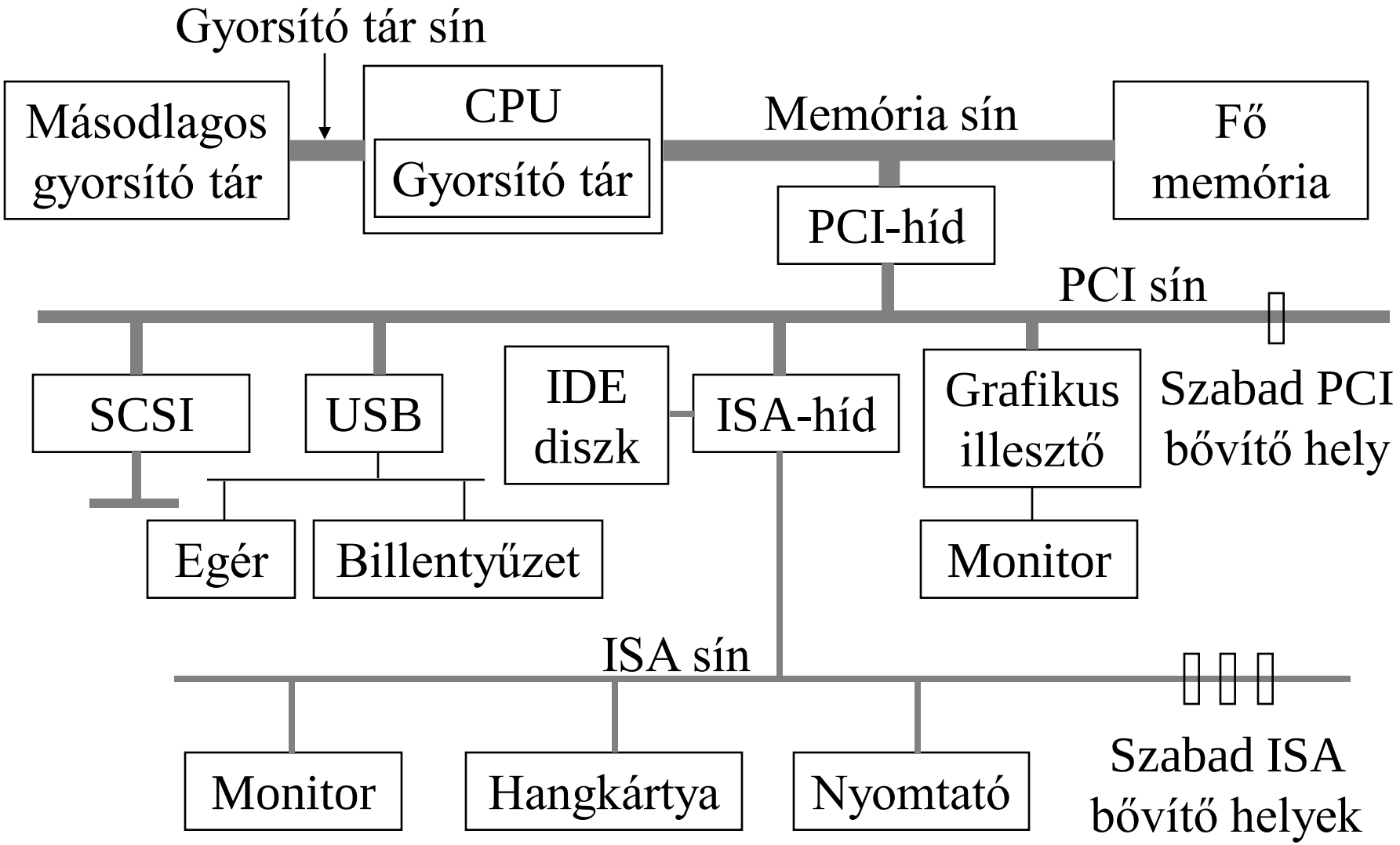
Egy vagy több csomagból áll.

Az egyes csomagok haladhatnak a központból az eszközök felé vagy fordítva. A haladási irány egy kereten belül is változhat.

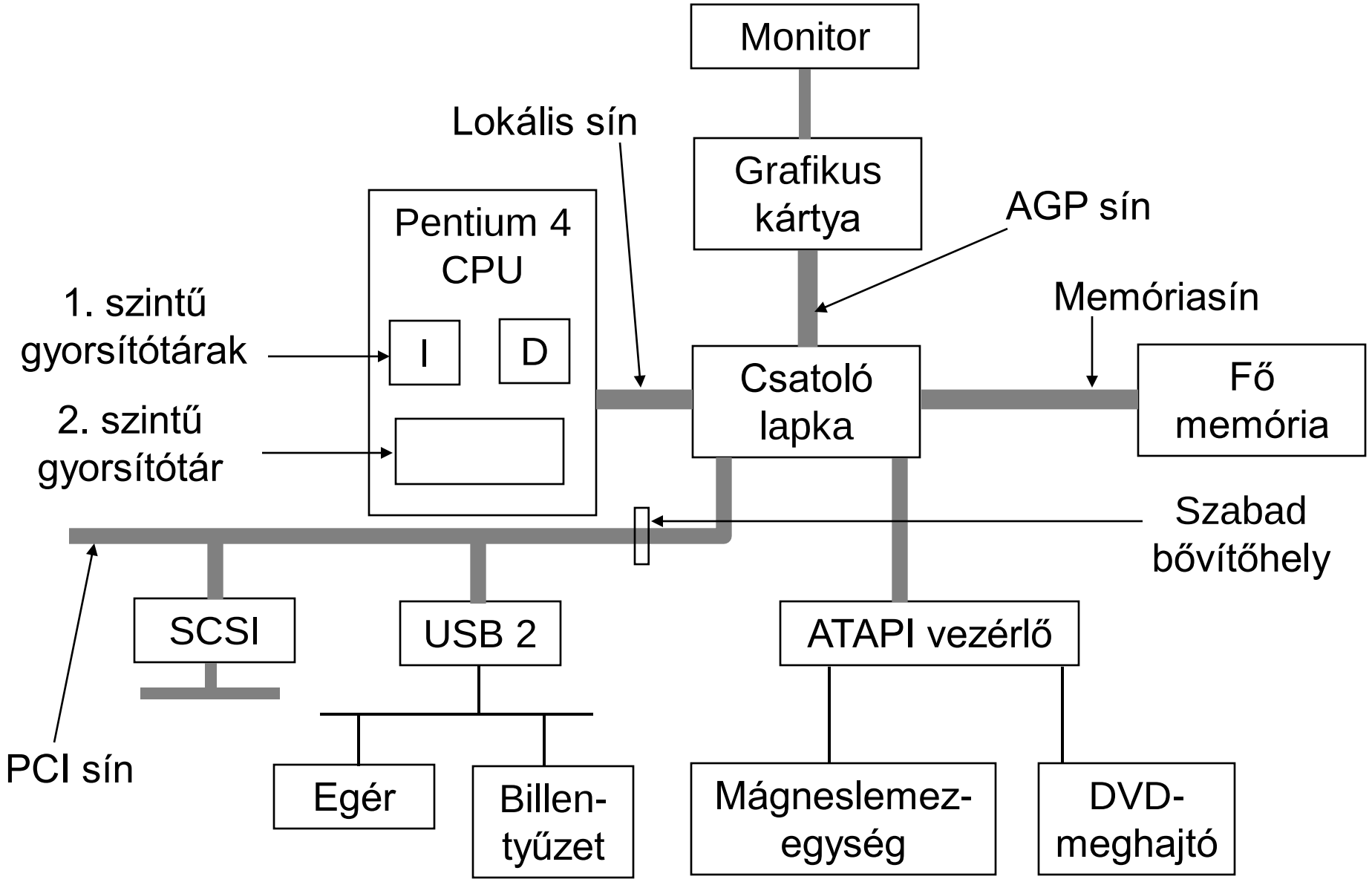
Az első csomag mindig **SOF**:
Start Of Frame – keret kezdet, szinkronizálja az eszközöket.

A keret lehet

- **Control** – vezérlő:
  - Eszköz konfigurálás,
  - Parancs,
  - Állapot lekérdezés.
- **Isochronous** – izoszinkron: valós idejű eszközök használják, pl. telefon. Hiba esetén nem kell ismételni az üzenetet.
- **Bulk** – csoportos: nagy tömegű adat átvitelére szolgál.
- **Interrupt** – megszakítás: Az **USB** nem támogatja a megszakítást, helyette pl. 50 ms-enként lekérdezhető az eszköz állapota.

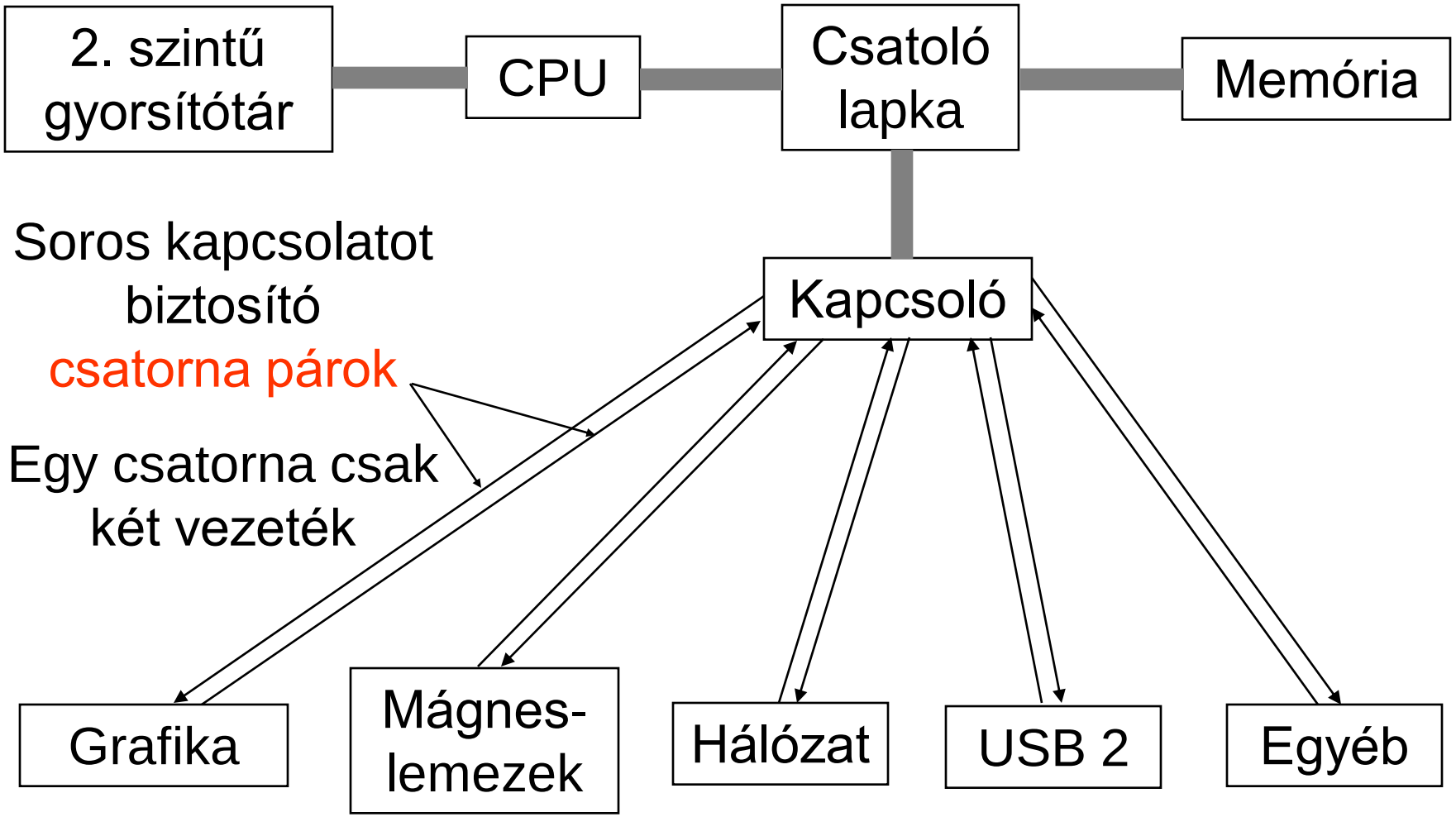


3.52. ábra. *Egy korai Pentium rendszer architektúrája*



3.53. ábra. *Egy modern Pentium 4 rendszer sín struktúrája*

PCI Express

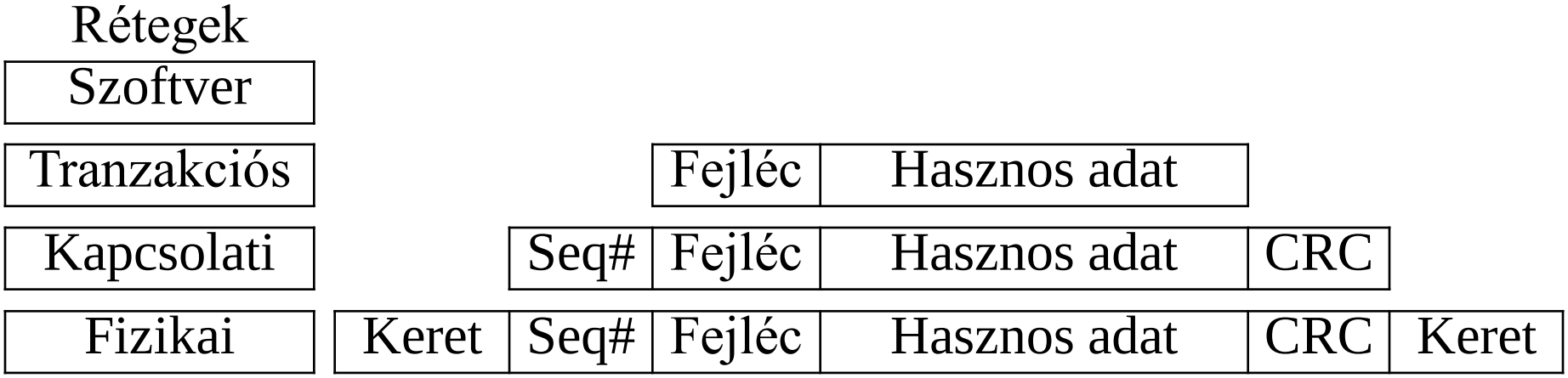


3.57. ábra. *Egy tipikus PCI Express rendszer vázlata*

Hagyományos sín	PCI Express
Több leágazású sín	Központosított kapcsoló
Széles, párhuzamos sín	Keskeny, közvetlen soros kapcsolat
Bonyolult mester – szolga kapcsolat	Kicsi, csomagkapcsolt hálózat
	CRC kód: nagyobb megbízhatóság
	A csatlakozó kábel > 50 cm lehet
	Az eszköz kapcsoló is lehet
	Meleg csatlakoztatási lehetőség
	Kisebb csatlakozók: kisebb gép

- Nem kell nagy bővítőkártyával csatlakozni a sínhez
- A winchester a monitorba is kerülhet

Egy csatorna hasznos sávszélessége minimum 2 Gbps, de bíznak benne, hogy hamarosan 10 Gbps



3.58. ábra. *A PCI Express protokollrendszer*
A csomagok formátuma

Fejléc cím, magas/alacsony prioritás, …
Seq# az üzenet sorszáma
CRC ciklikus redundanciakód (Cyclic Redundancy Check)

Ha a számított és kapott CRC megegyezik, akkor nyugtázza, különben újra kéri az adatot.

Input, output (I/O) utasítások (I8086/88)

A külvilággal történő információ csere **port**-okon (kapukon) keresztül zajlik. A kapu egy memória cím, az információ csere erre a címre történő írással, vagy erről a címről való olvasással történik. Egy-egy cím vagy cím csoport egy-egy perifériához kötődik. A központi egység oldaláról a folyamat egységesen az **IN** (input) és az **OUT** (output) utasítással történik.

A perifériától függ, hogy a hozzá tartozó port 8 vagy 16 bites. A központi egységnek az **AL, AX** illetve **EAX** regisztere vesz részt a kommunikációban. A port címzése 8 bites közvetlen adattal vagy a **DX** regiszterrel történik (65536 port).

Példa MASM kóddal:

```
IN    AL/AX/EAX,port
      ; AL/AX ⇐ egy byte/word a port-ról
OUT   port,AL/AX/EAX
      ; port ⇐ egy byte/word AL/AX-ből
```