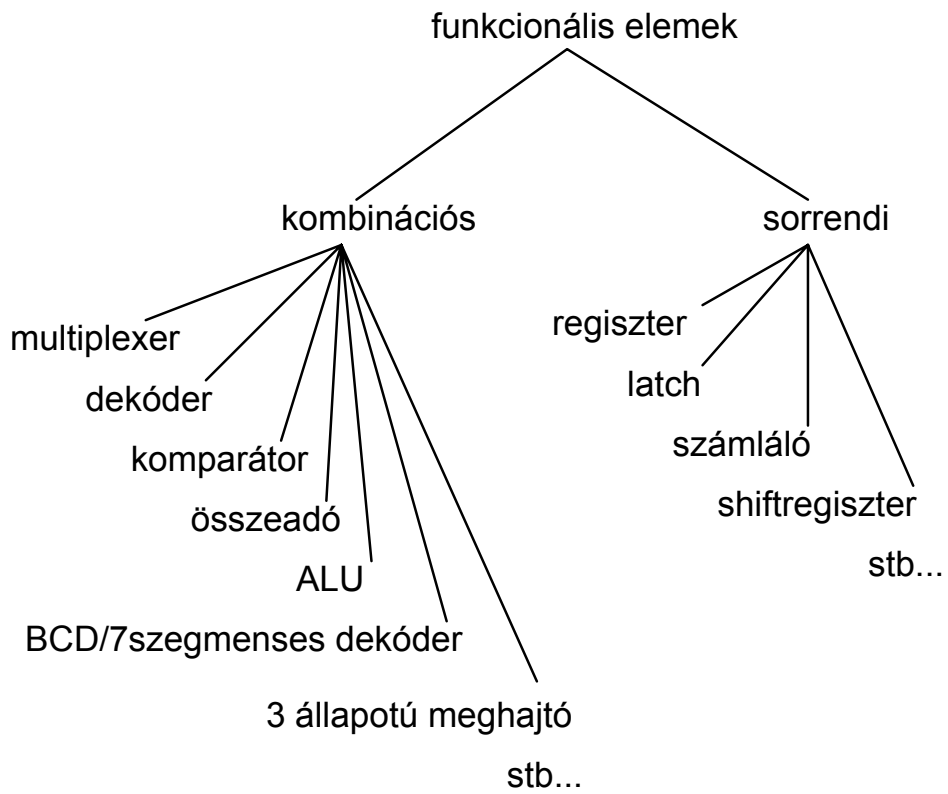


Funkcionális elemek

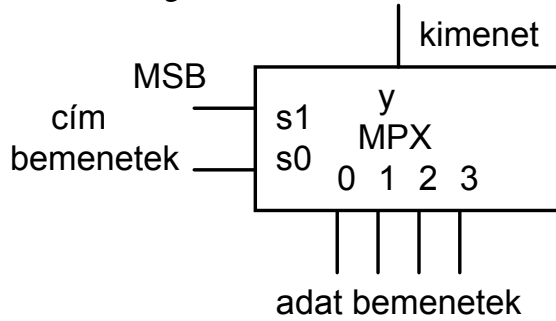
© Benesóczky Zoltán 2004

A jegyzetet a szerzői jog védi. Azt a BME hallgatói használhatják, nyomtathatják tanulás céljából.
Minden egyéb felhasználáshoz a szerző beleegyezése szükséges.



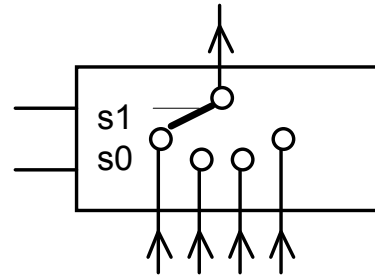
Multiplexer

Rajzele:



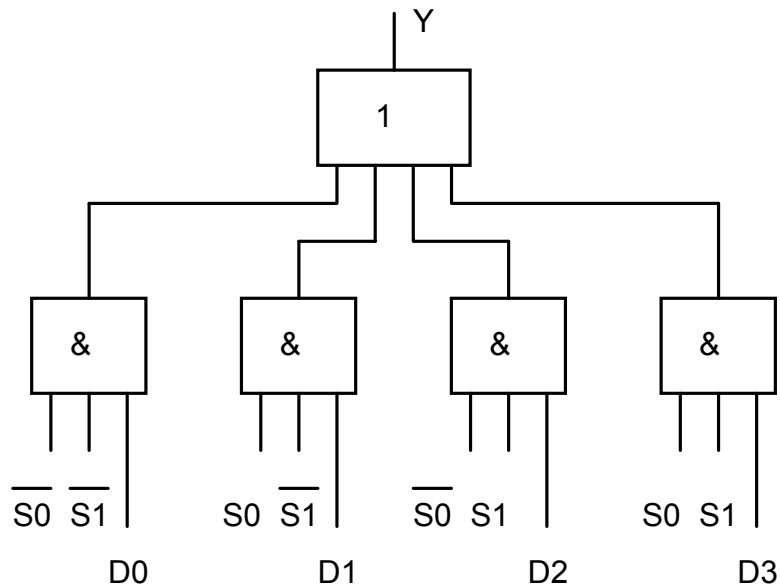
a)

Működése:

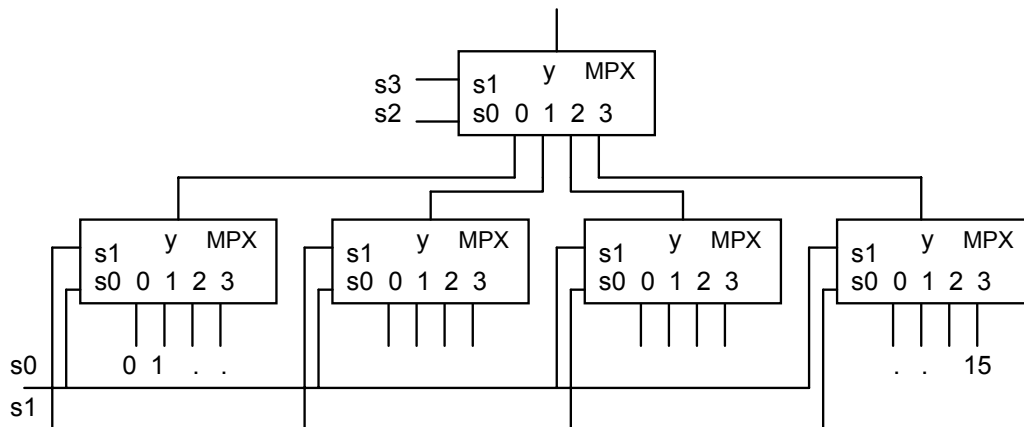


b)

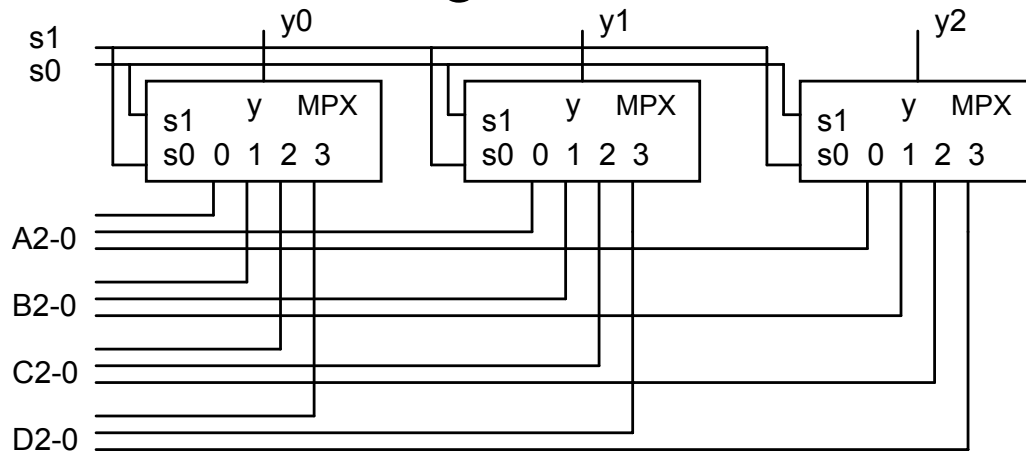
Belső felépítése:



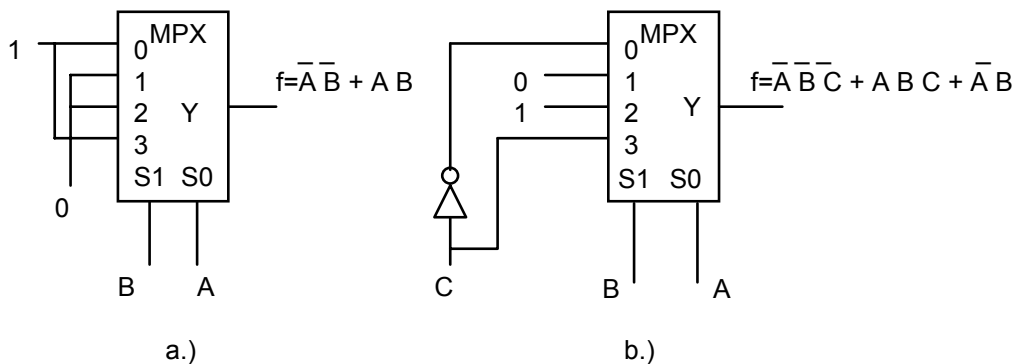
Csatorna számának növelése:



Csatorna szélességének bővítése:



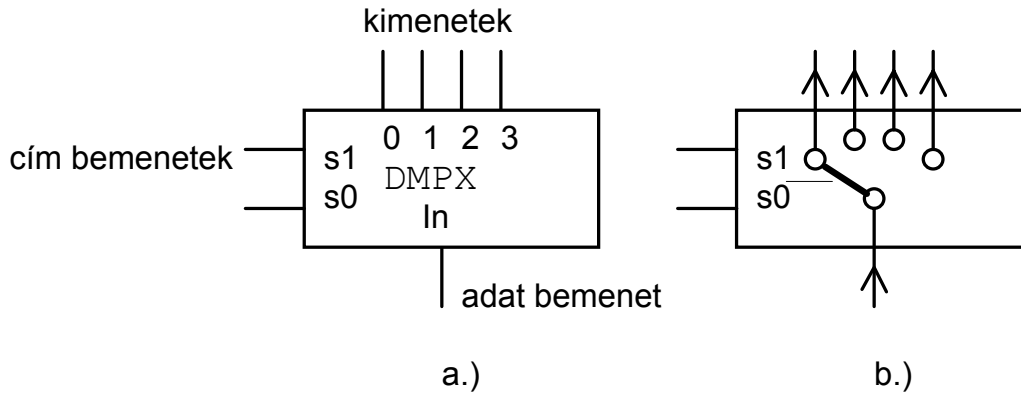
Multiplexer mint univerzális KH:



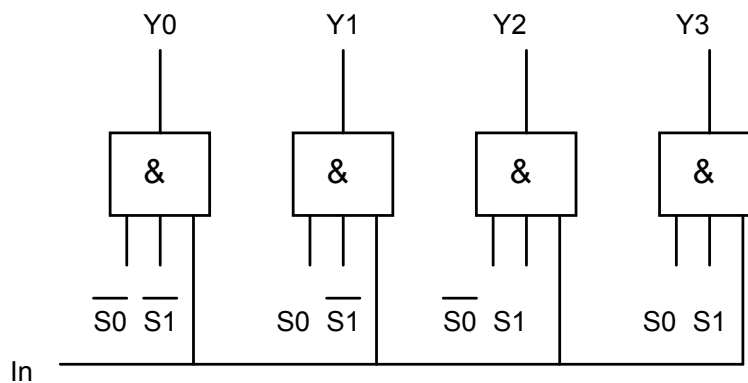
Dekóder/demultiplexer

A dekóder rajzjele:

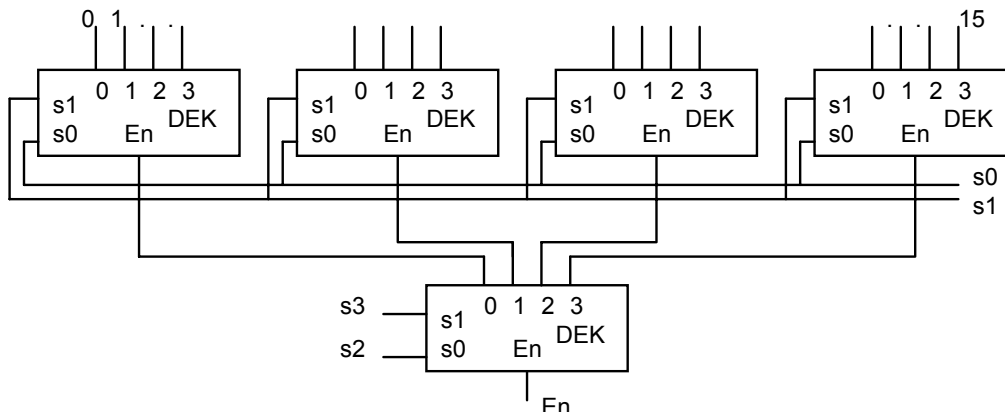
Működése:



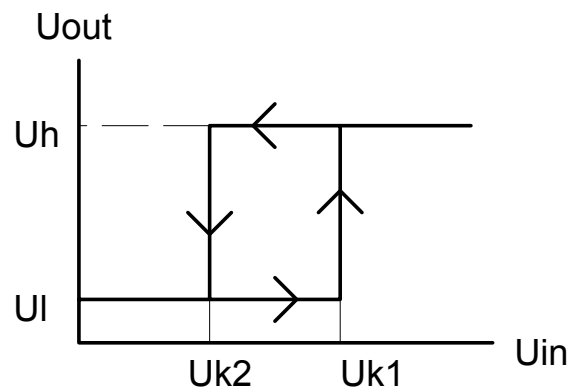
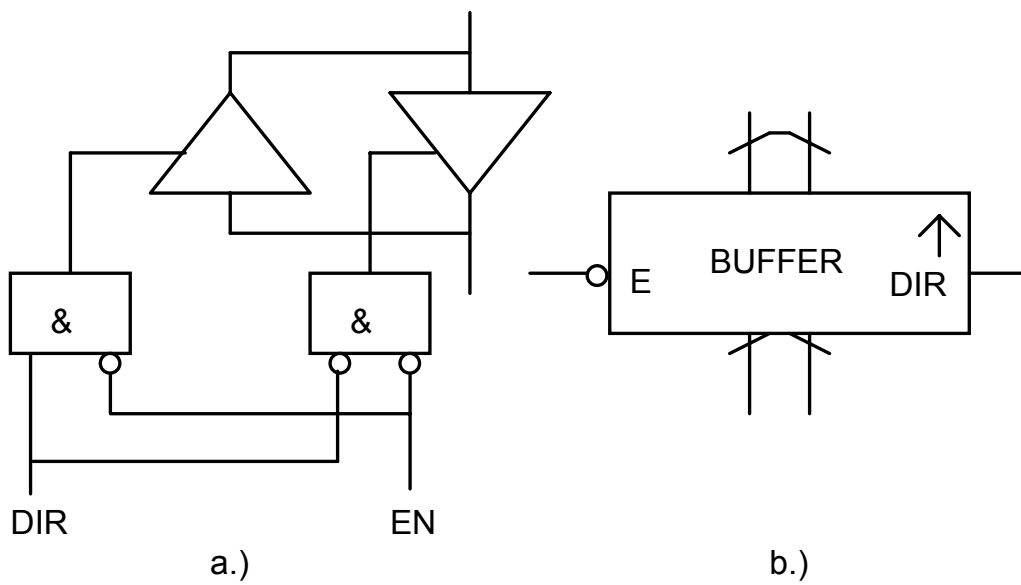
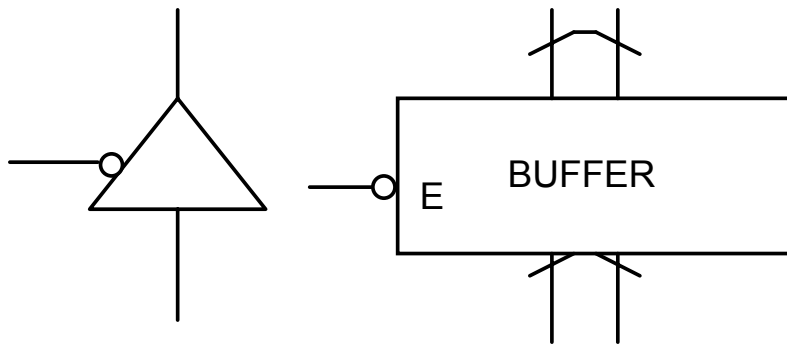
A dekóder belső felépítése:



A dekóder bővítése:

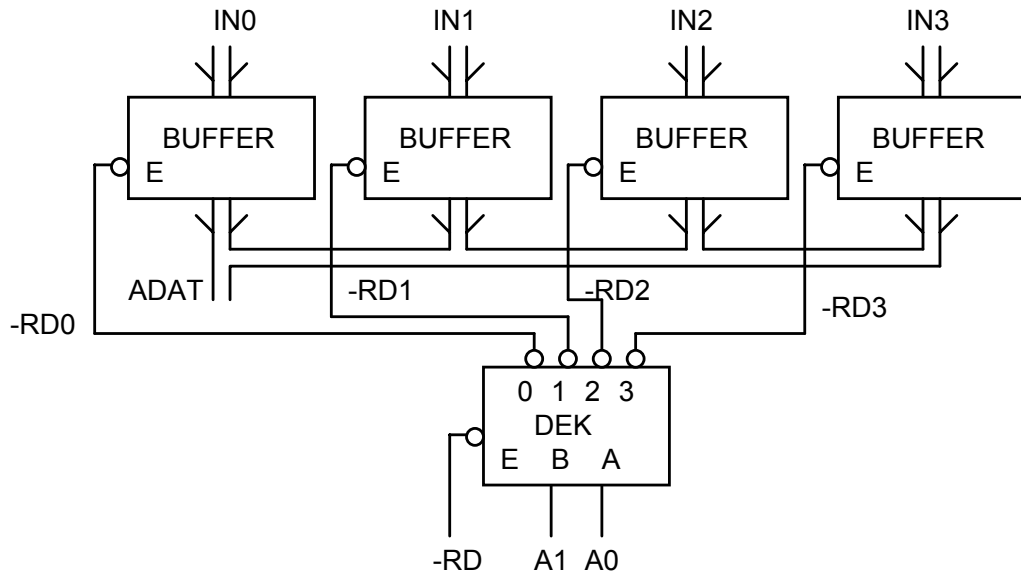


3 állapotú meghajtó

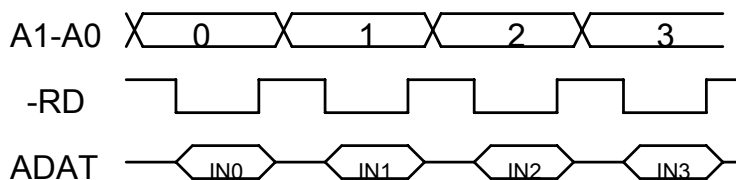
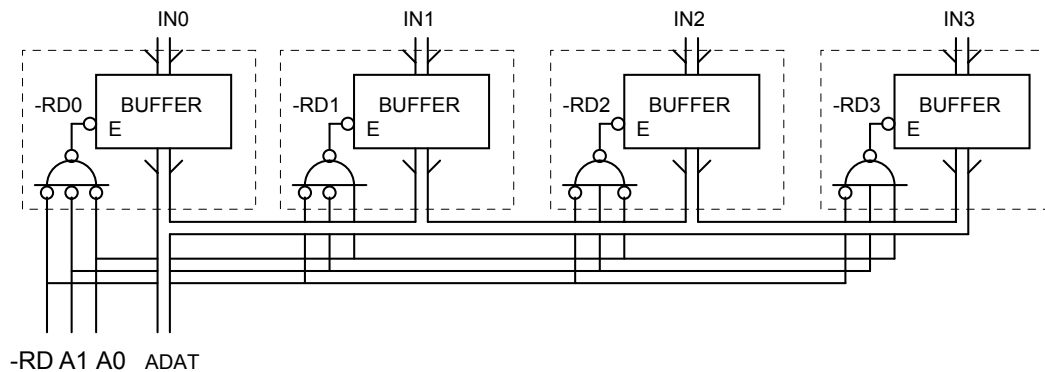


Multiplexer 3 állapotú meghajtóval

Centralizált felépítés:

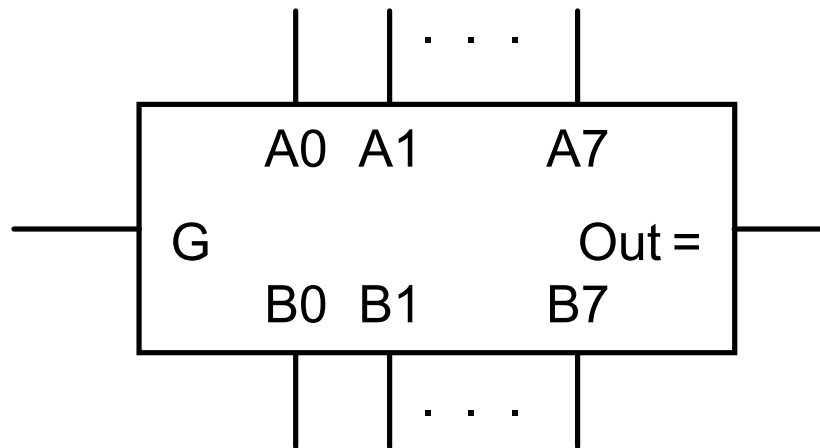
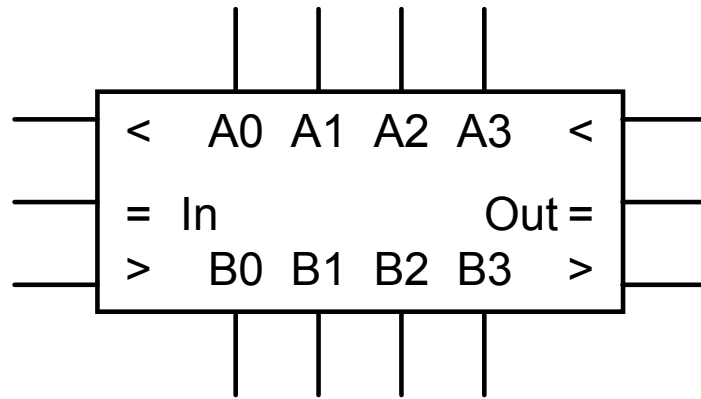


Decentralizált felépítés:



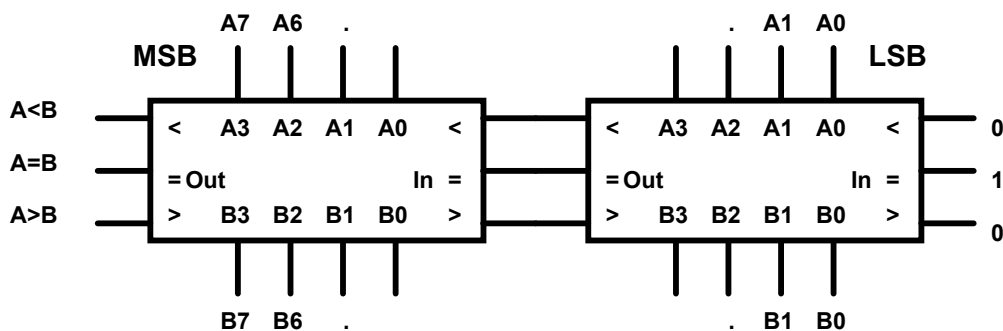
Komparátor

Rajzjele:



Hagyományos kaszkádosítás

A legelső komparátor kaszkádosító bemeneteit úgy kell beállítani, mintha egy előző komparátor egyenlőséget jelezne.



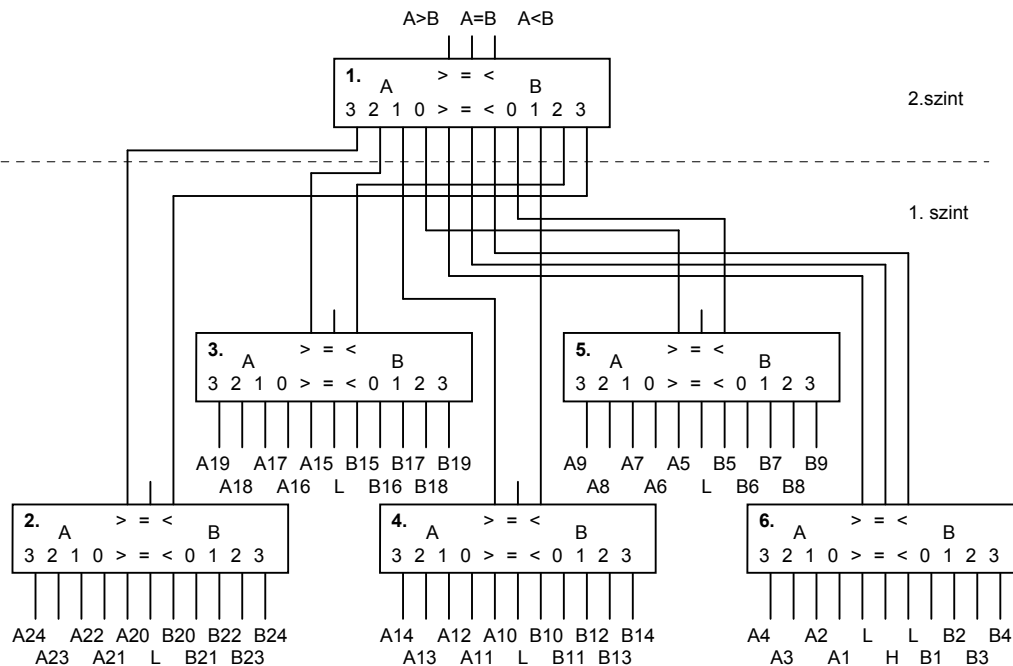
Komparátorral a 2-es komplementben ábrázolt számokat az előjel bit invertálása után tudjuk összehasonlítani (offset kóddá transzformálás).

Gyorsított kaszkádosítás

A hagyományos kaszkádosítás hátránya, hogy a kaszkádosított egységek késletetési ideje összeadódik.

Itt 4 bites komparátorokkal gyors 24 bites komparátor.

A kapcsolás olyan komparátort tételez fel, amelynek $<$ és $>$ bementére egyszerre 1-et kapcsolva, annak $<$ és $>$ kimenete egyforma logikai értéket ad, ha $A=B$.



Összeadó

Funkciója:

Az összeadó két n bites (előjel nélküli abszolútértékes vagy 2-es komplementes ábrázolású) bináris szám összeadása.

$0+0=0$ $C=0$, $0+1=1+0=1$ $C=0$,

$1+1=0$ $C=1$

P1: **1001**

+1011

10100

Paramétere:

- szószélesség (hány bites): n

Bemenetei:

- összeadandó számok (A, B) 2 x n bit

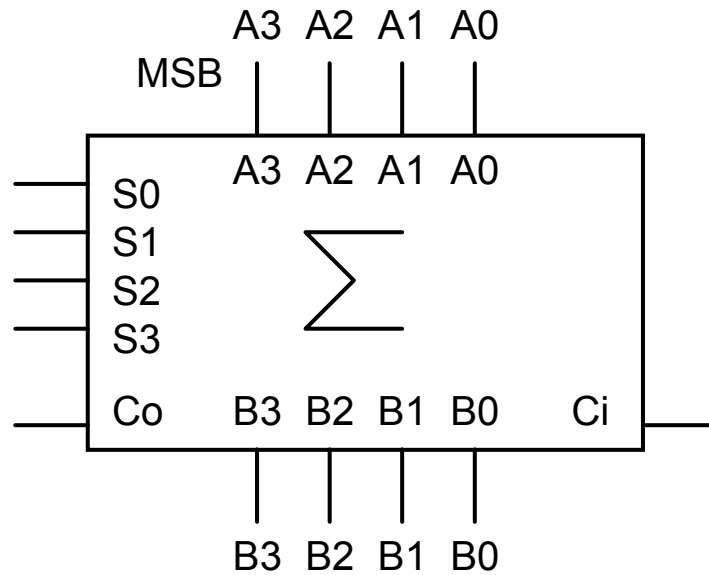
C_i átvitel bemenet

Kimenetei:

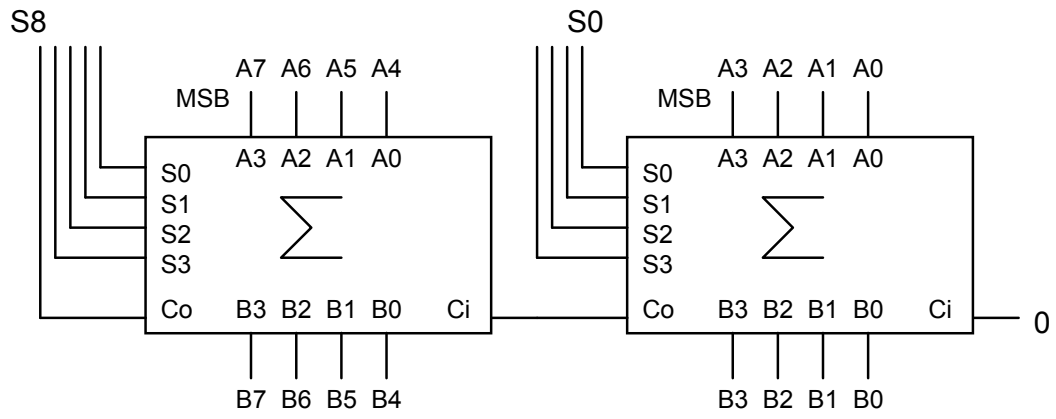
- eredmény S (SUM) n bit

- Co átvitel

Rajzjele:

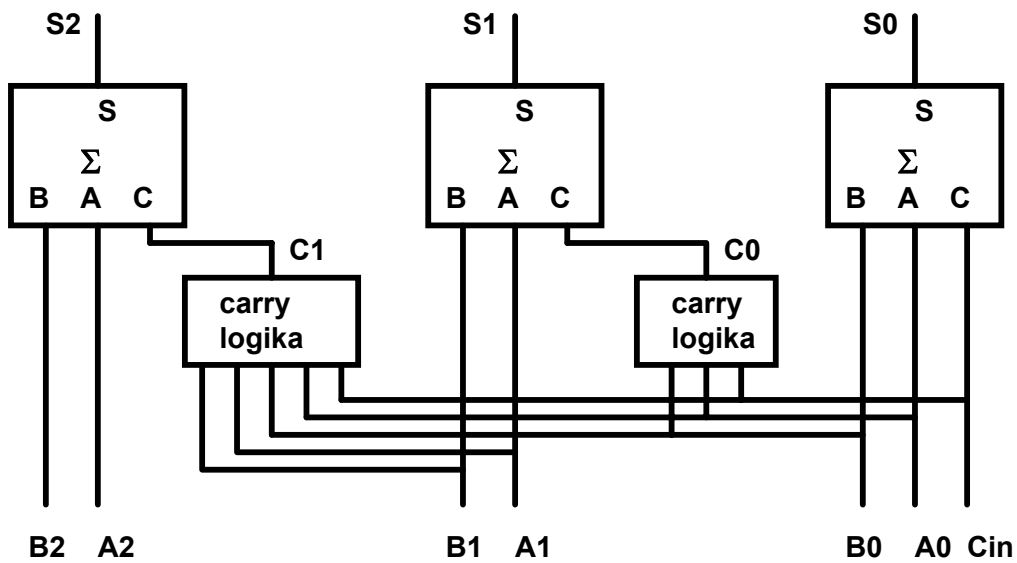


Két 4 bites összeadó kaszkádosítása
A legelső összeadó Cin bemenetére 0-át
kell kötni.



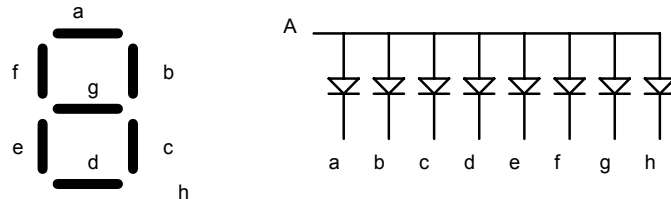
Sok bit esetén az átvitel soros terjedése nagy késleltetést okoz.

Elkerülése: átvitel gyorsítással

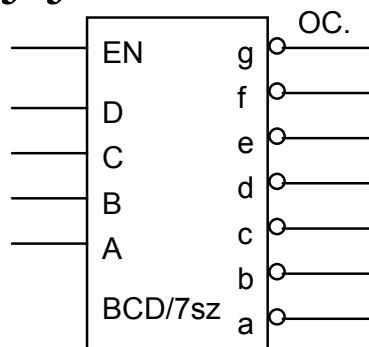


BCD/7szegmenses dekóder

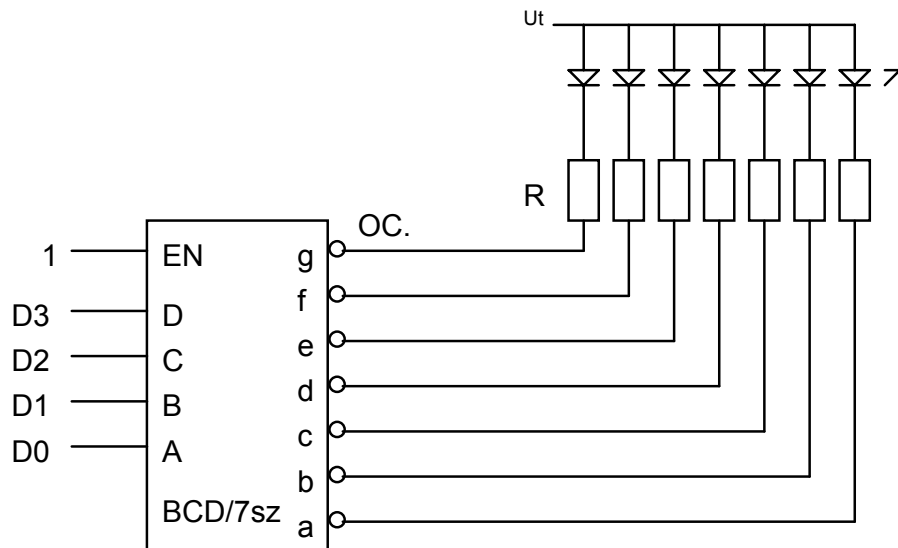
A meghajtandó LED kijelző:



A dekóder rajzjele:

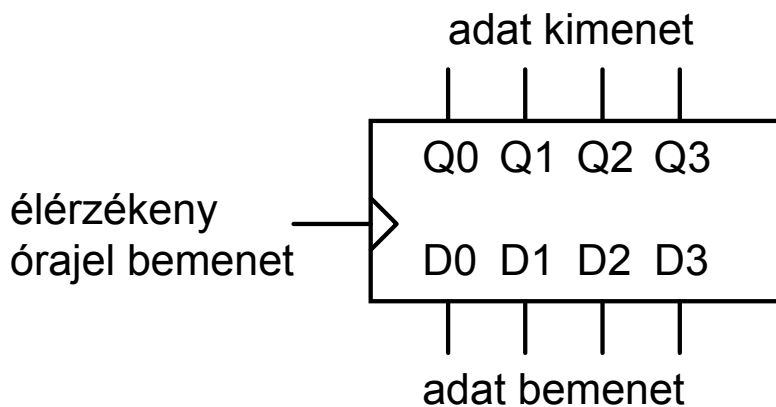


Alkalmazása:

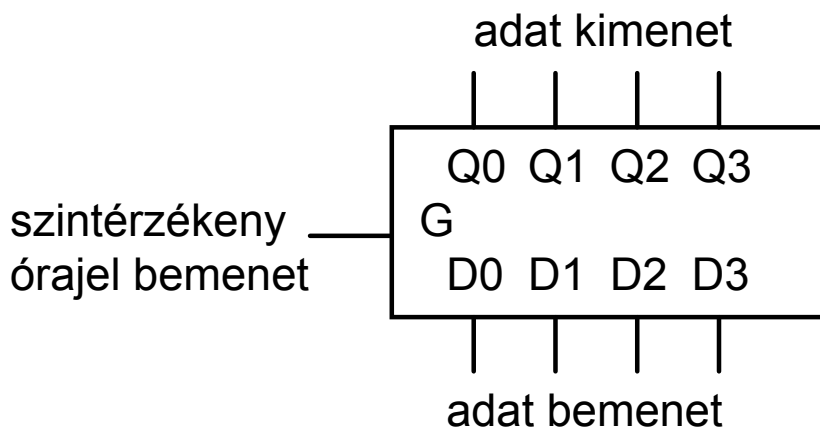


Sorrendi funkcionális elemek (tároló tulajdonságú elemek)

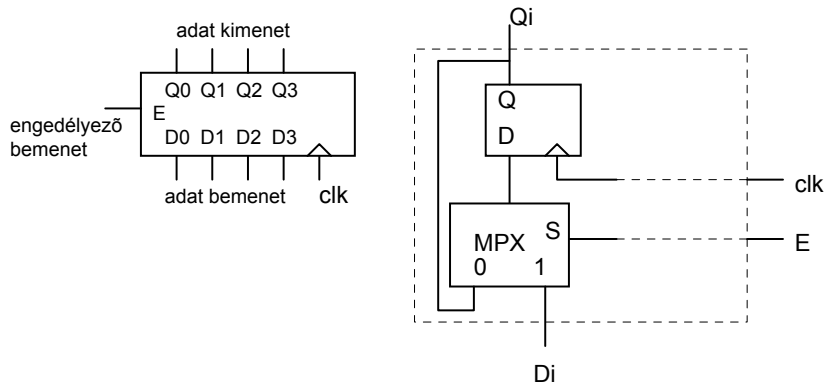
Regiszter (közös órajelű D flip-flopok)



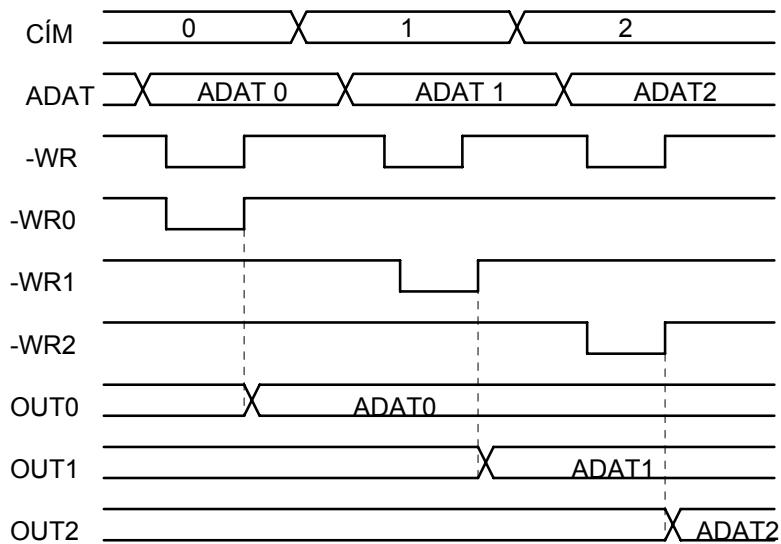
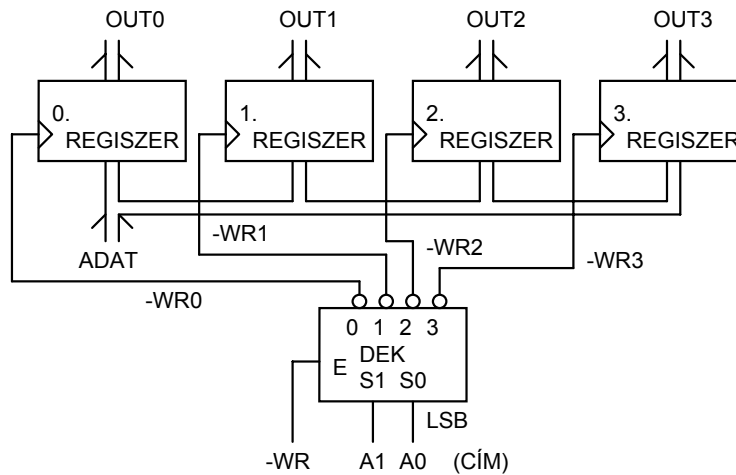
Latch (közös G-jű D-G flip-flopok)



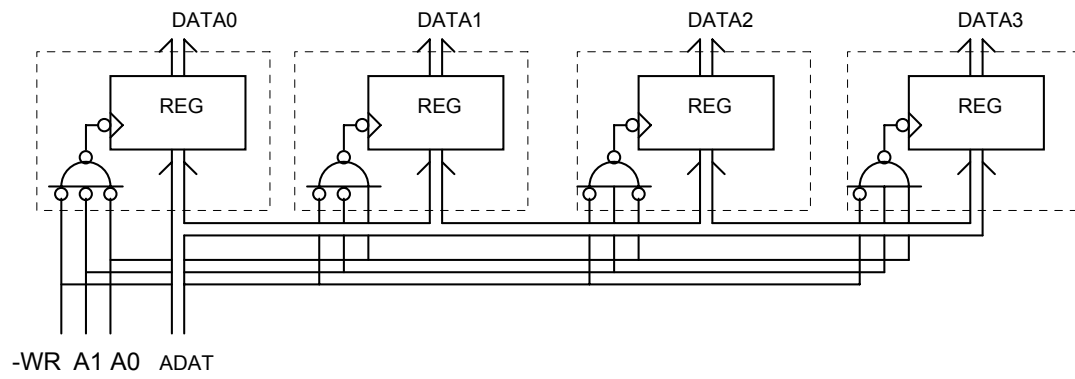
Engedélyezhető regiszter



Regiszterek beírása buszról (centralizált felépítés):



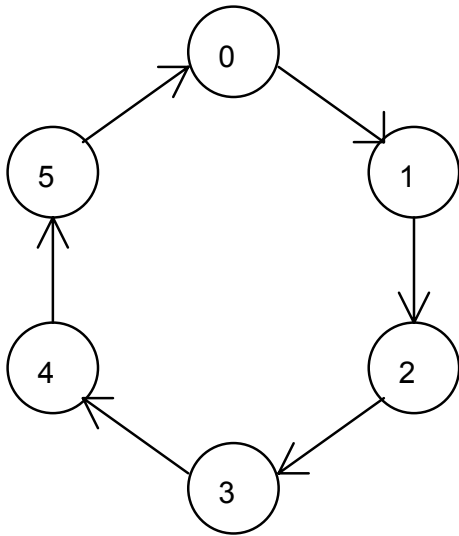
Decentralizált felépítés:



Számlálók

A számlálók (számláló üzemmódjára jellemző) állapotgráfja gyűrű alakú.

Pl. 6-os számláló állapotgráfja:

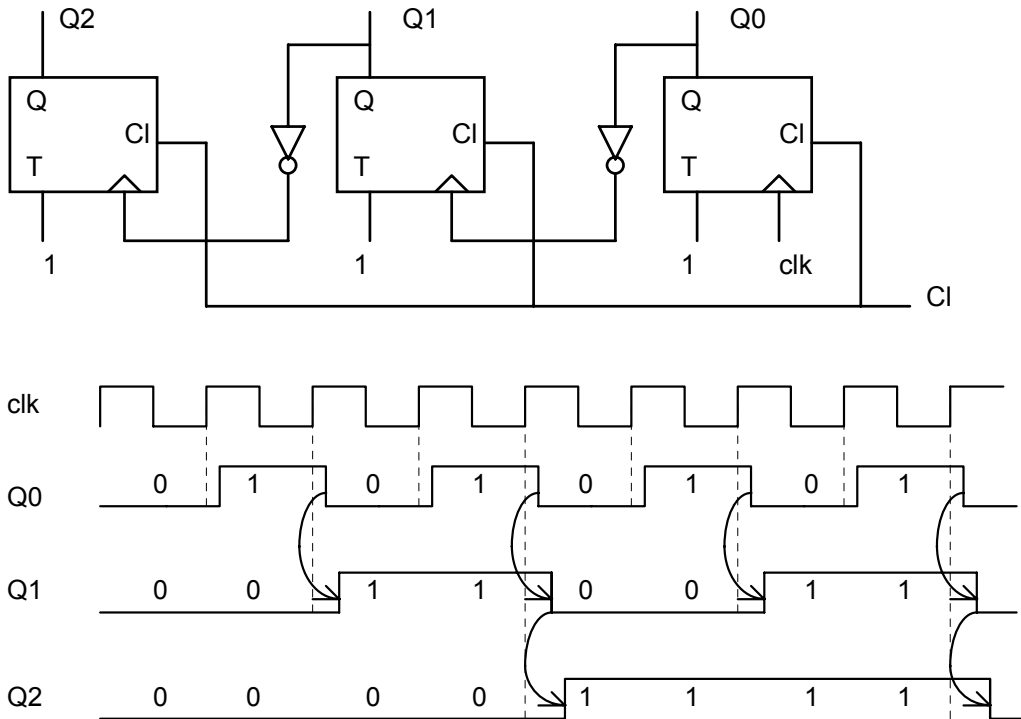


A számláló az órajeleit számolja. A **modulusa**, a ciklus hossza.

Modulus alapján: bináris, decimális, egyéb (12-es, 6-os stb.)

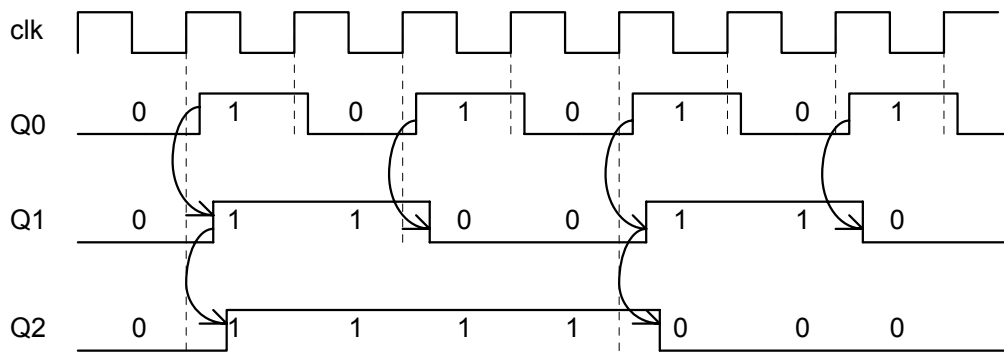
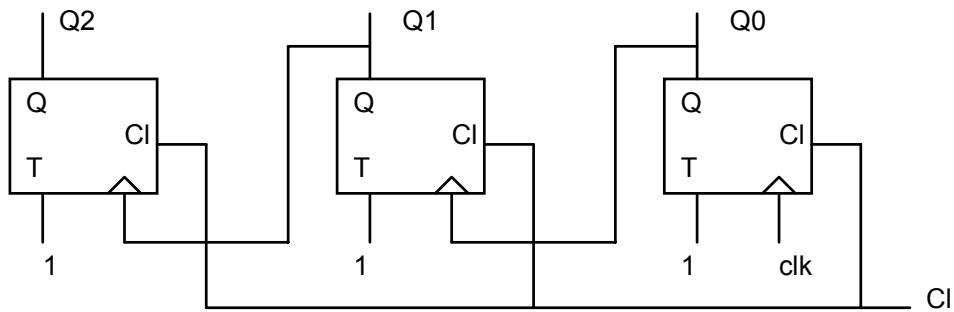
Aszinkron számláló

Aszinkron bináris *felfele* számláló,
aszinkron törléssel:



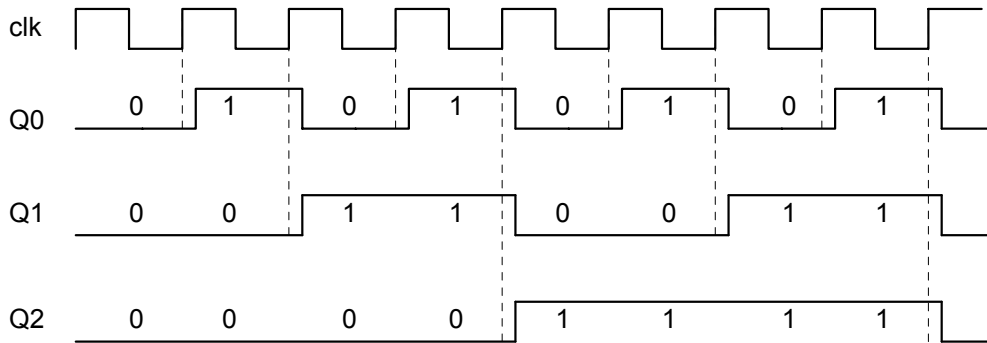
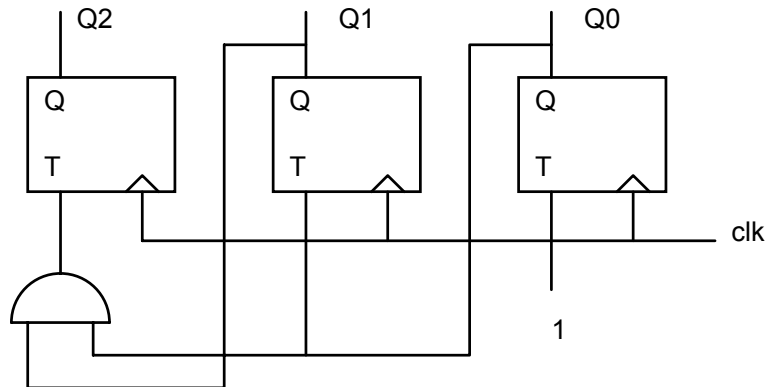
Az egyes kimenetek egyre jobban késnek
az órajelhez képest.

Aszinkron bináris *lefele* számláló, aszinkron törléssel:



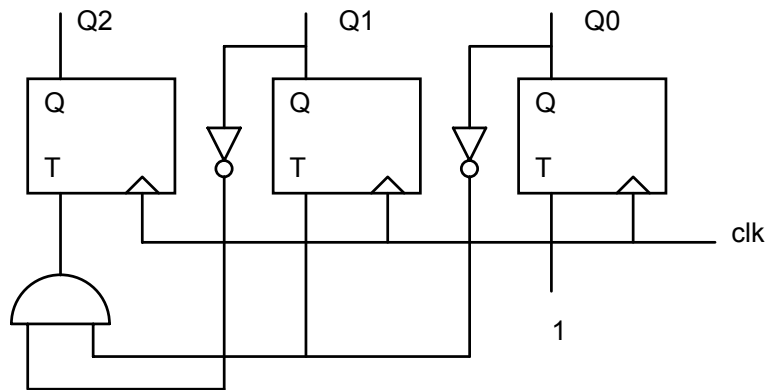
Szinkron számláló

Szinkron bináris *felfele* számláló belső felépítése:

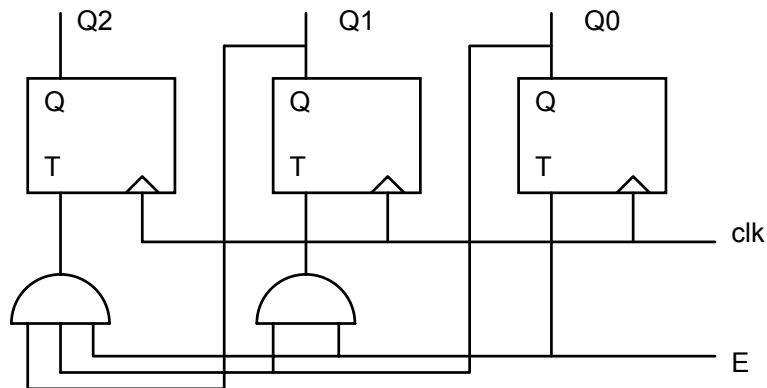


A kimenetek egyszerre váltanak, késleltetésük kicsi az órajelhez képest.

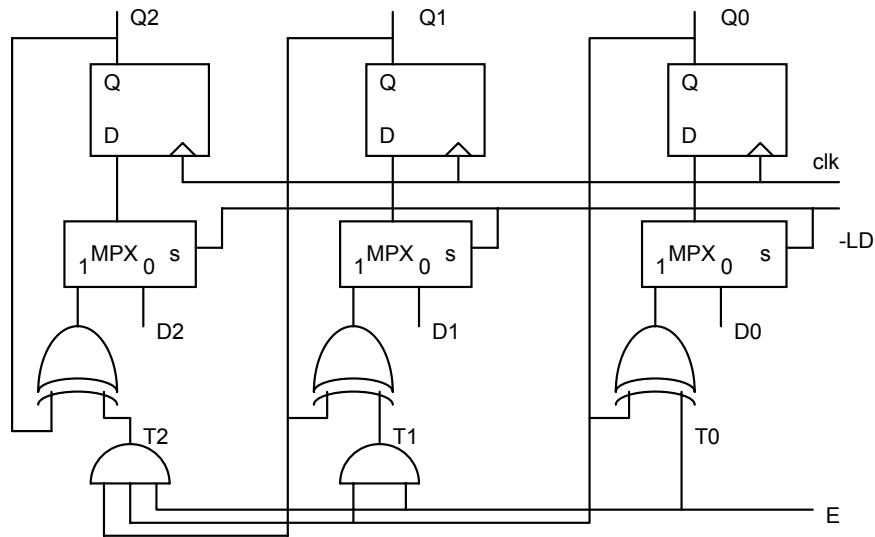
Szinkron bináris *lefele* számláló belső felépítése:



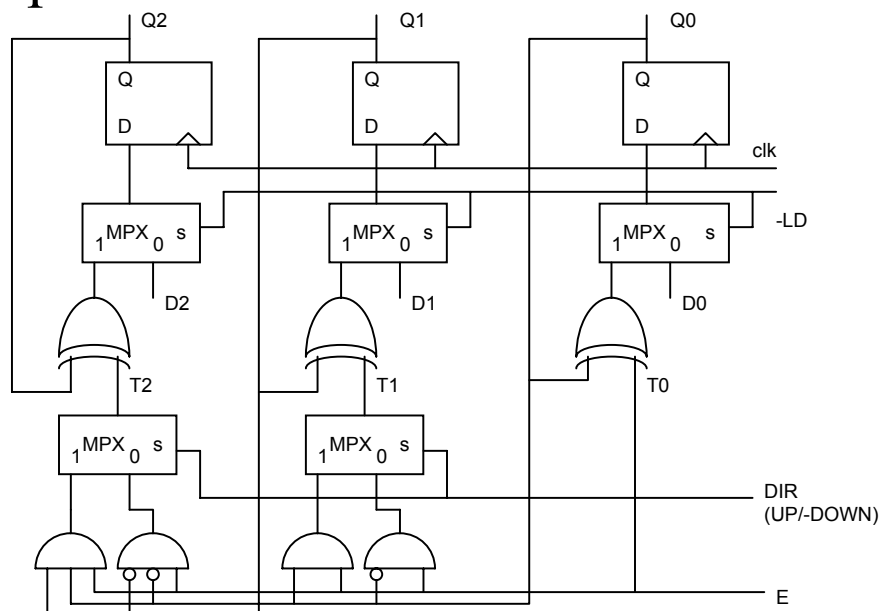
Szinkron engedélyezhető bináris felfele számláló belső felépítése:



Szinkron bináris, engedélyezhető és tölthető felfele számláló funkcionális felépítése:



Szinkron bináris, engedélyezhető és tölthető fel-le számláló funkcionális felépítése:



Számlálók vezérlő jelei lehetnek:

CL: törlés

LD: betöltés

En: engedélyezés (csak szinkron esetben)

DIR: számlálási irány

Számlálók kimeneti jelei lehetnek:

CY(carry): felfele számláló végállapota

Bináris számlálónál

$$CY = Q_0.Q_1.Q_2.....$$

BO(borrow): lefele számláló végállapota

$$BO = /Q_0./Q_1./Q_2.....$$

Max/min: fel-le számláló végállapotai

Bináris számlálónál

$$M/m = DIR.Q_0.Q_1.Q_2..+ /DIR./Q_0./Q_1./Q_2..$$

RCY: ripple carry

$$RCY = E.CY$$

RCO: ripple clock

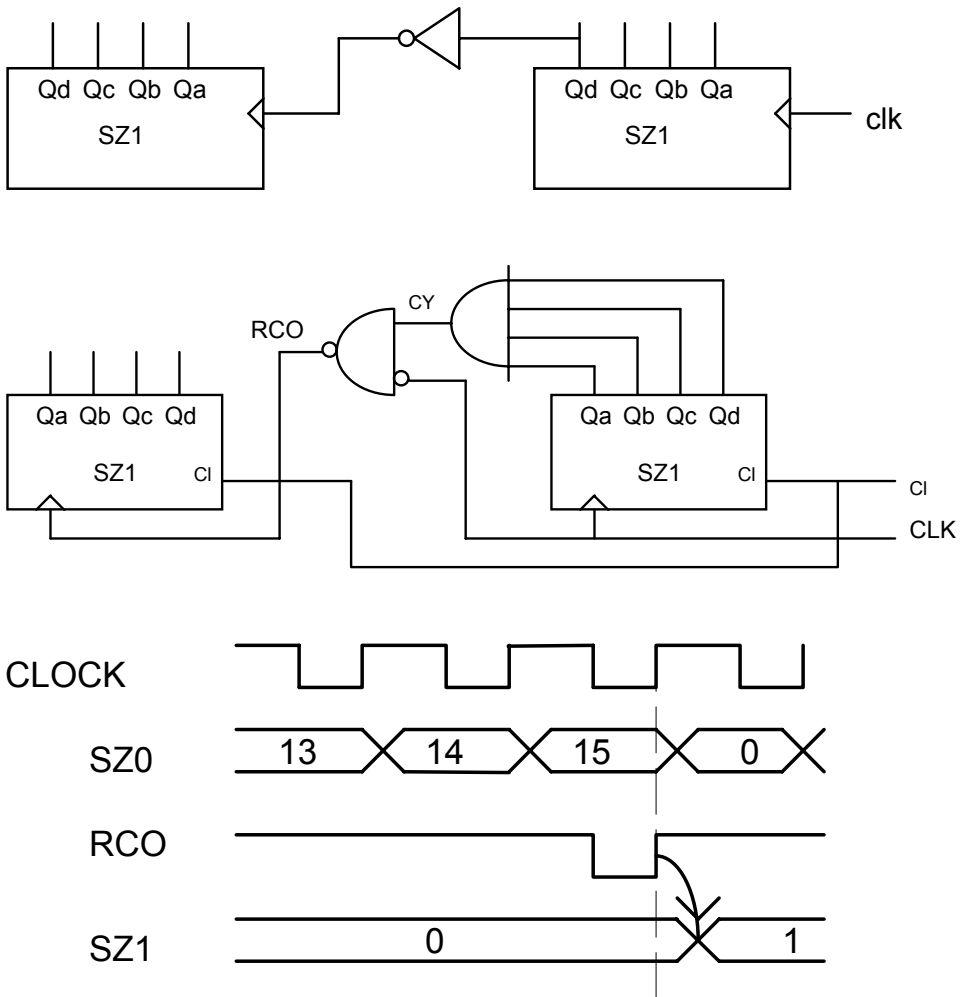
$$RCO = E./(CY./CLK)$$

Számlálók kaszkádosítása (modulus növelés)

3db 4 bites bináris számláló működése

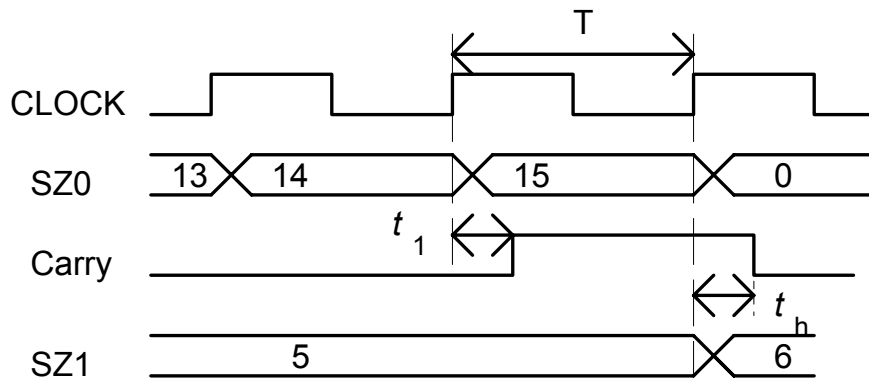
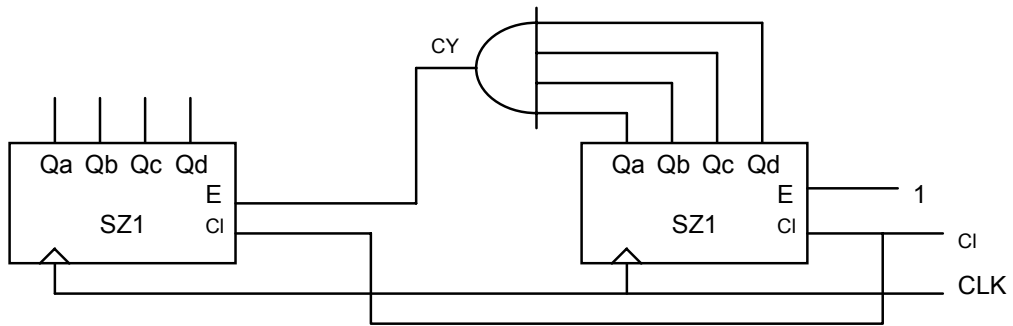
3.	2.	1.	0.
0000	1111	1111	1111
0001	0000	0000	0000

Aszinkron kaszkádosítások:

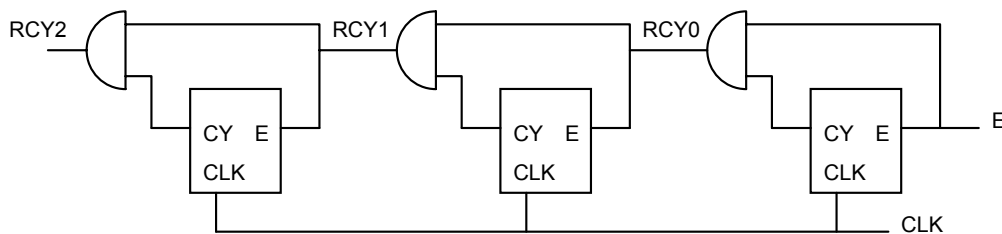


A kaszkádosított számlálók modulusai összeszorzódnak.

Szinkron kaszkádosítások:



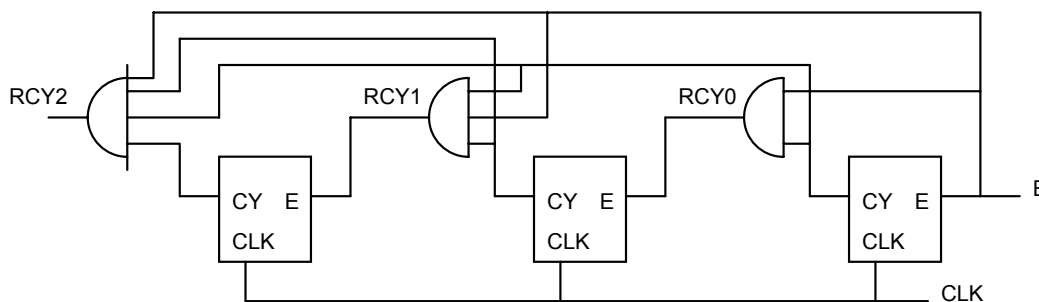
A carry soros terjesztése:



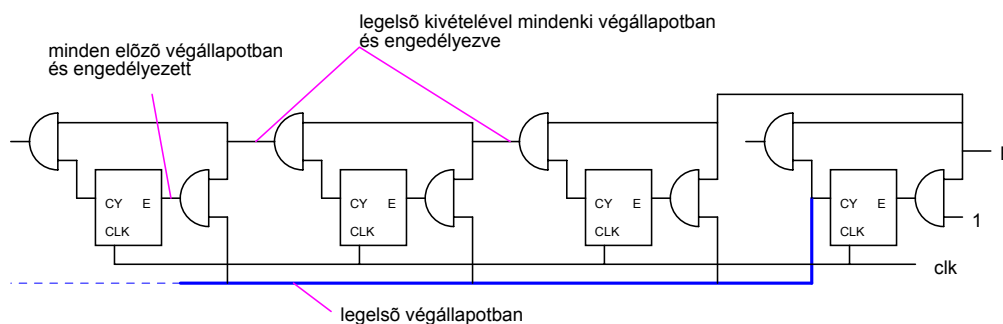
Az engedélyezés az MSB felé egyre jobban késik az órajelhez képest.

Gyorsított kaszkádosítások

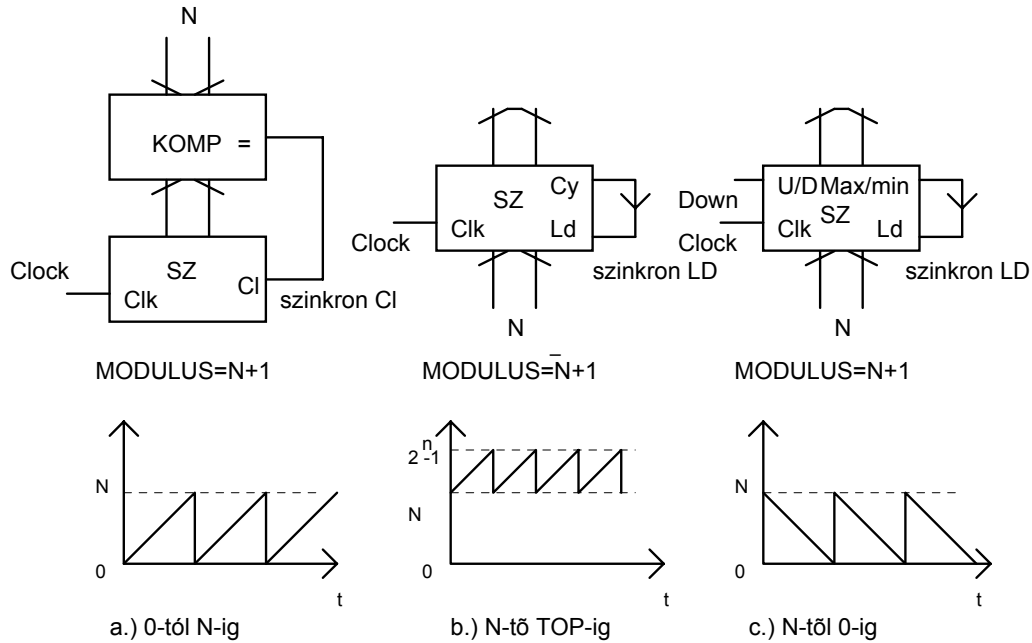
A CY párhuzamos előállításával:



A legelső CY előrecsatolásával:

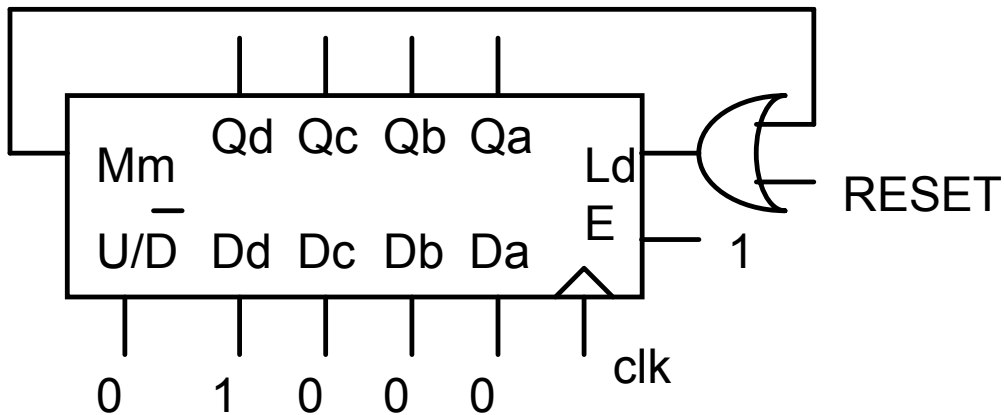


Moduluscsökkentési eljárások



Mintapéldák:

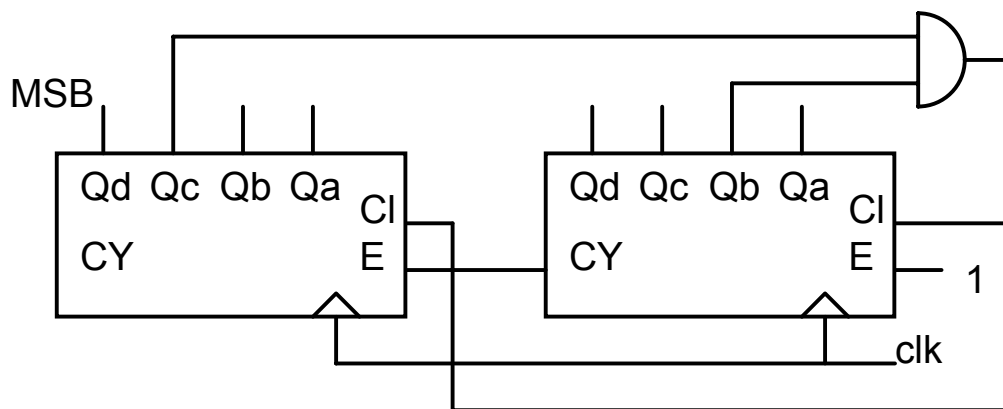
Készítsünk 16-os szinkron tölthető lefele számlálóból 9-es modulusút. A kódolása 8-0.



Készítsünk 16-os szinkron törlésű, engedélyezhető felfele számlálóból 67-os modulusút. A kódolása legyen 0-66.

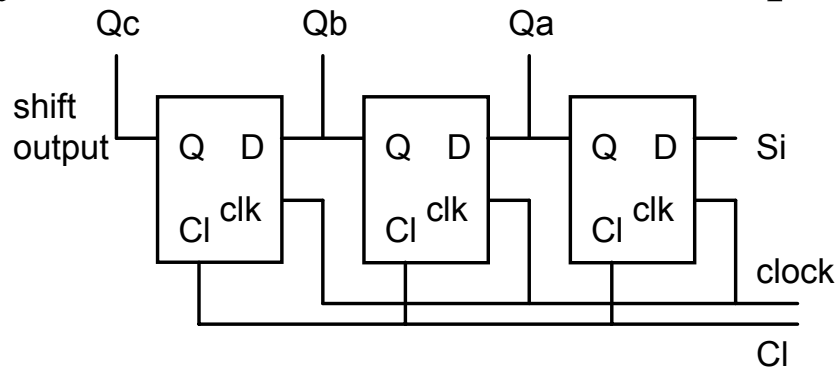
$$66_D = 00100\ 0010_B$$

Kaszkádosítunk, majd modulust csökkentünk.

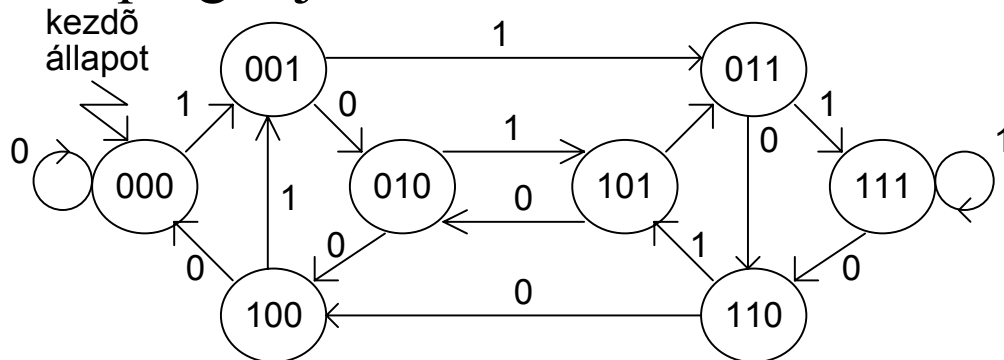


A shiftregiszter

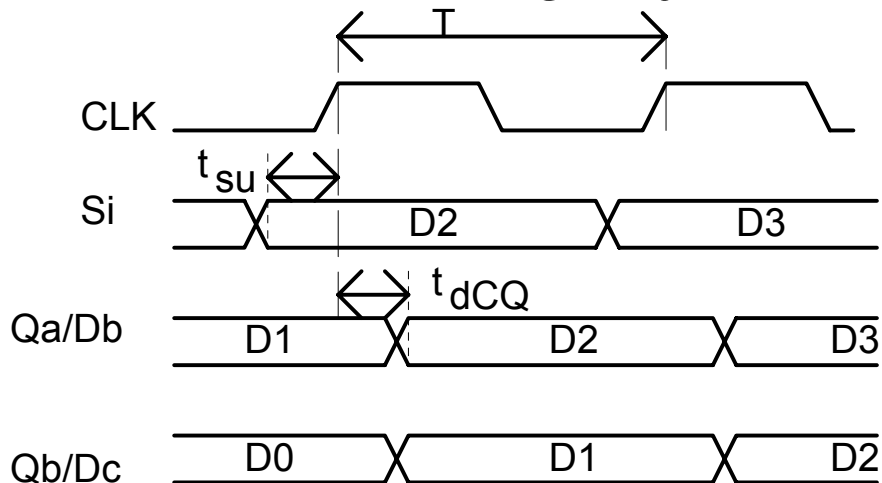
Egy balra shiftelő SHR belső felépítése:



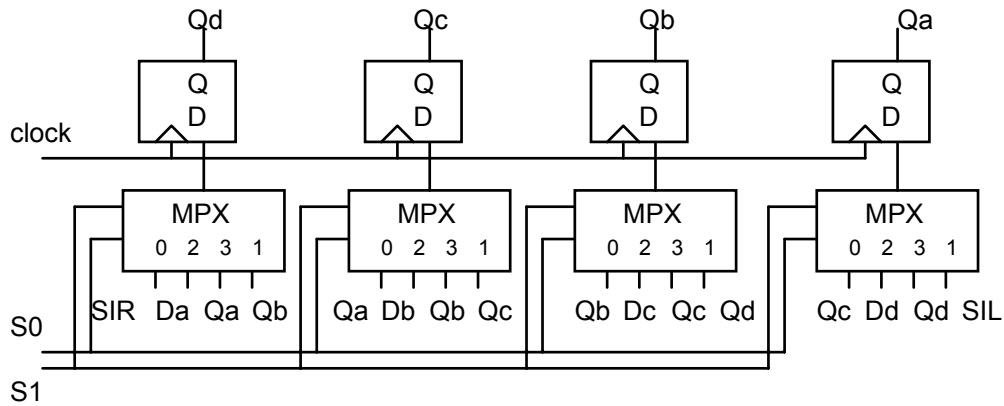
Állapotgráfja:



Működésének idődiagramja:



Tölthető, engedélyezhető, jobbra-balra shiftelő shiftregiszter belső felépítése:



S1S0

00 jobbra shift (SHR)

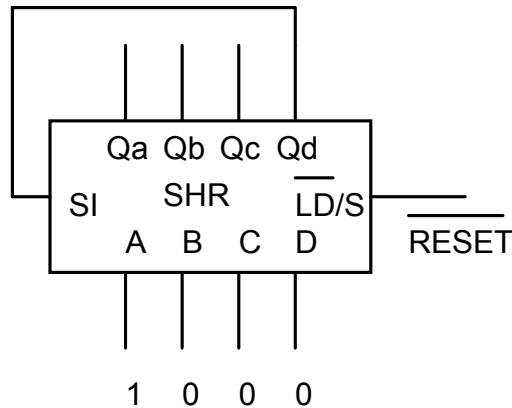
01 balra shift (SHL)

10 betölt (LOAD)

11 tart (HOLD)

Shiftregiszter mint számláló

Gyűrűs számláló:

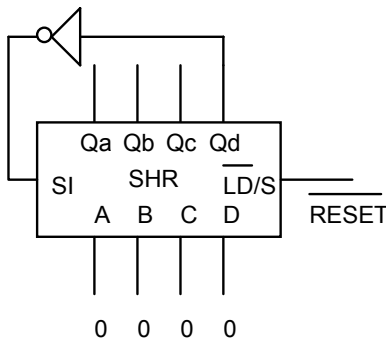


A számláló kódolása Qa, Qb, Qc, Qd:

1000, 0100, 0010, 0001

N bites SHR esetén a modulusa: N

Johnson számláló:

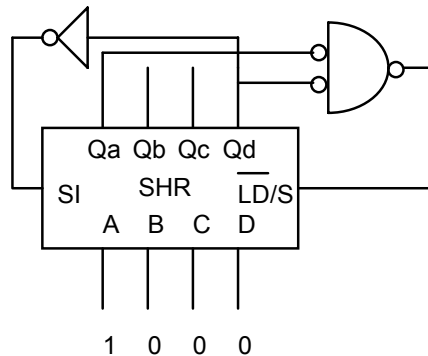


A számláló kódolása Qa, Qb, Qc, Qd:

0000, 1000, 1100, 1110, 1111, 0111, 0011, 0001

Modulusa N bites SHR esetén: 2N

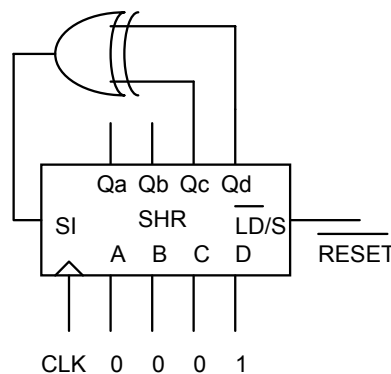
Önkorrigáló Johnson számláló



Néhány órajelen belül tetszőleges állapotból beletalál a normál ciklusba.

Pl: 1010, 1101, 0110, 1000,...

Álvéletlen generátor MOD 2 visszacsatolt shiftregiszterrel

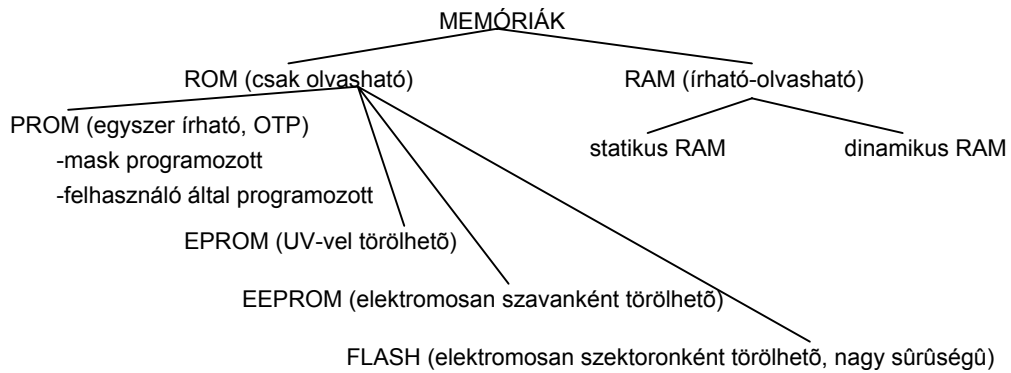


Kódolása:

0001, 1000, 0100, 0010, 1001, 1100, 0110,
1011, 0101, 1010, 1101, 1110, 1111, 0111,
0011

Modulusa N bites SHR és maximális ciklushosszú visszacsatolás esetén: $2^N - 1$

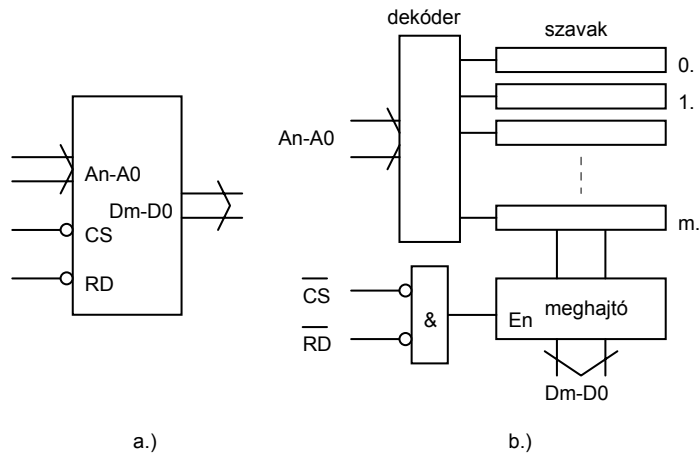
Memória elemek:



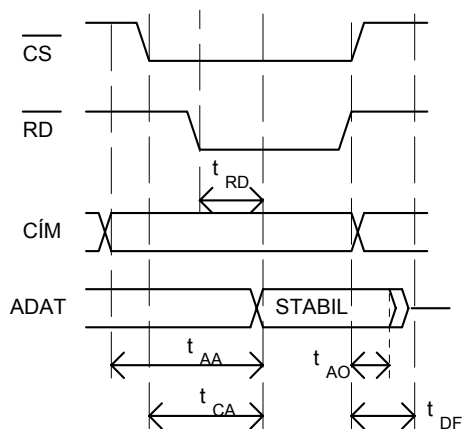
Jellemzők:

szószélesség, byte szám, hozzáférési idő

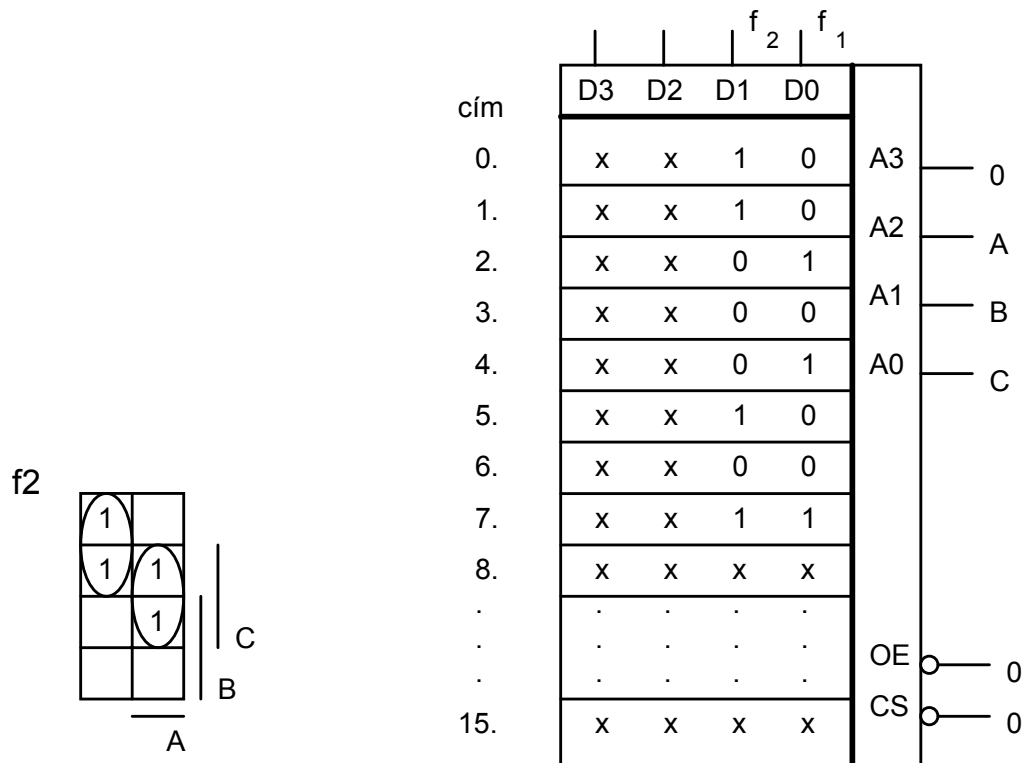
ROM



Olvasás idődiagramja:



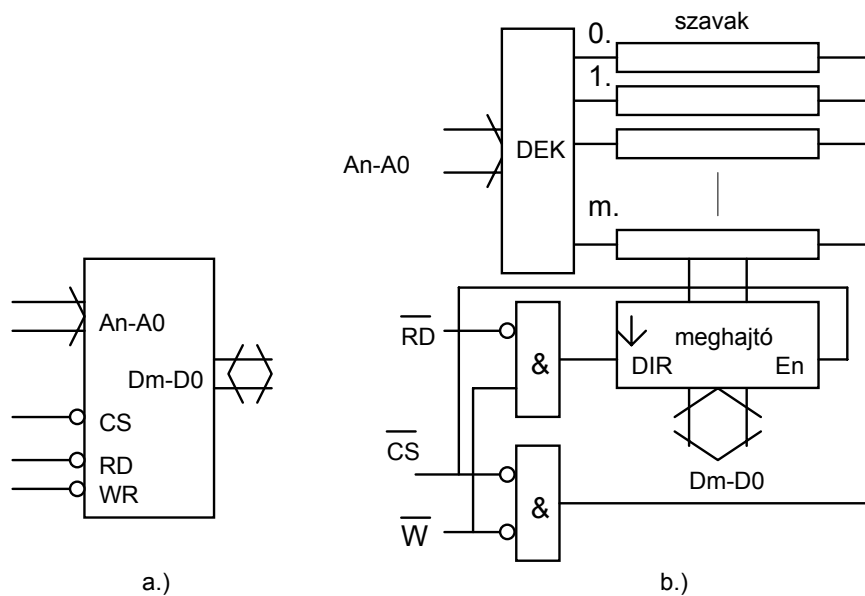
ROM mint inverzális KH:



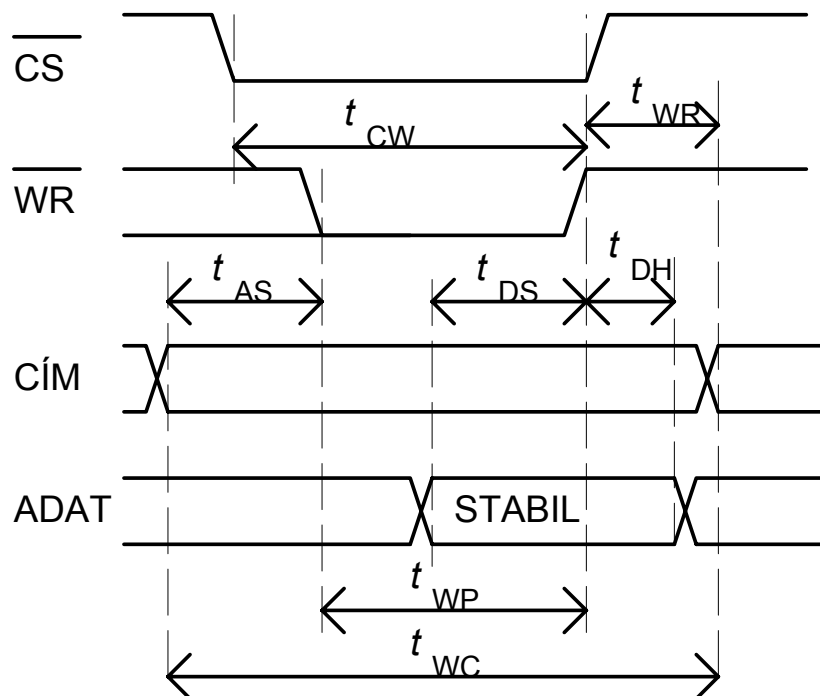
$$f_2 = \overset{\text{a.)}}{\overline{A}\overline{B}} + AC = \overset{\text{b.)}}{\overline{A}\overline{B}\overline{C}} + ABC + \overline{A}\overline{B}C + \overline{A}\overline{B}C$$

A tartalom az igazságtábla.

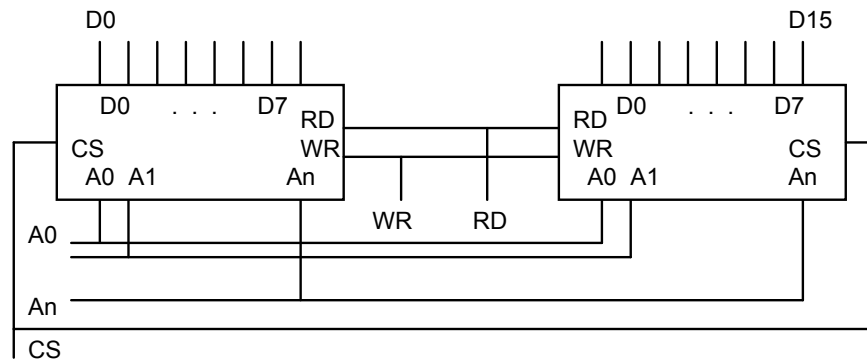
RAM



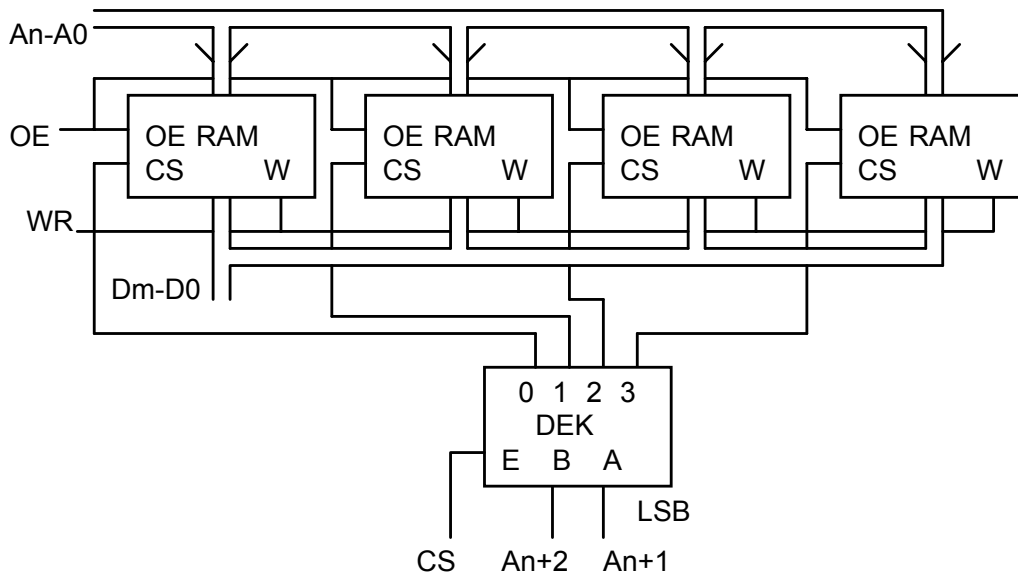
Statikus RAM írási ciklusa:



Memóriák szélességének növelése

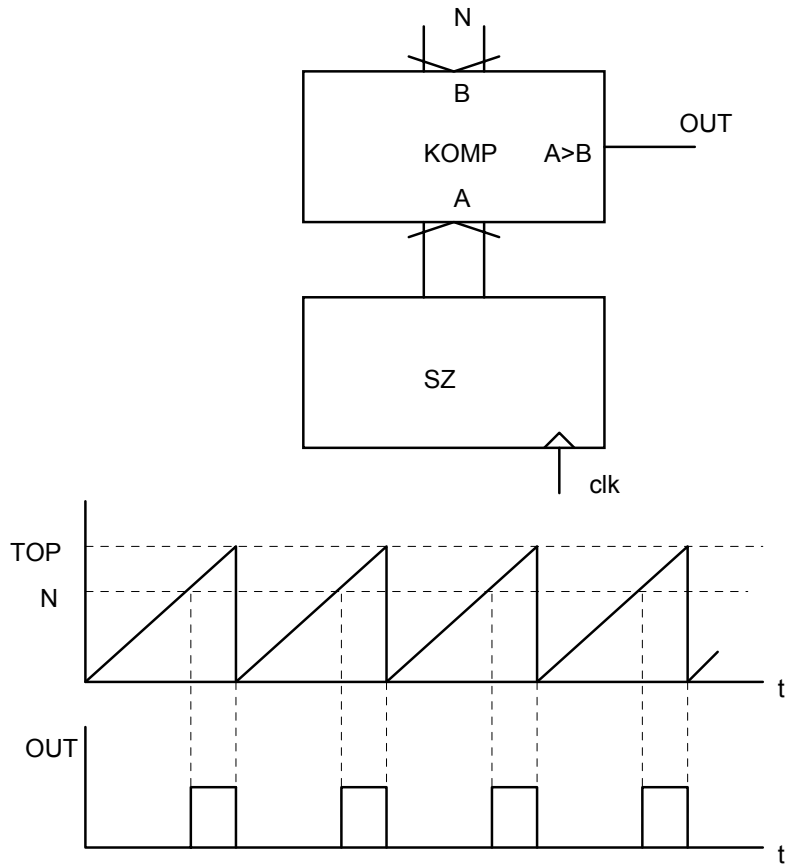


Memóriák kapacitásának növelése

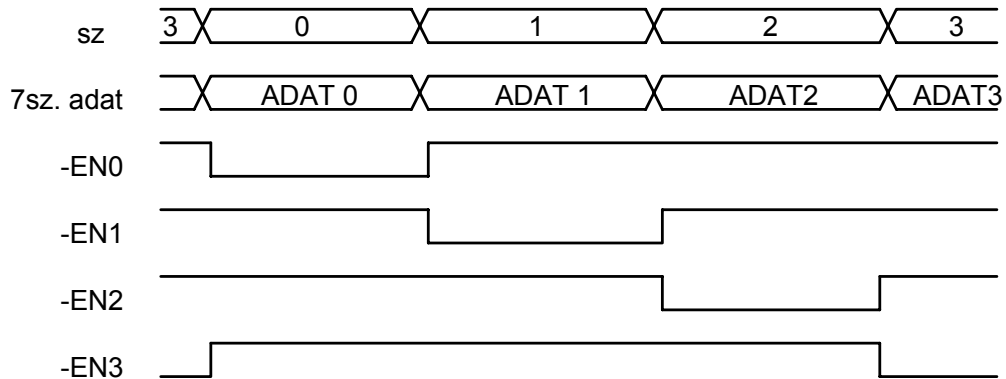
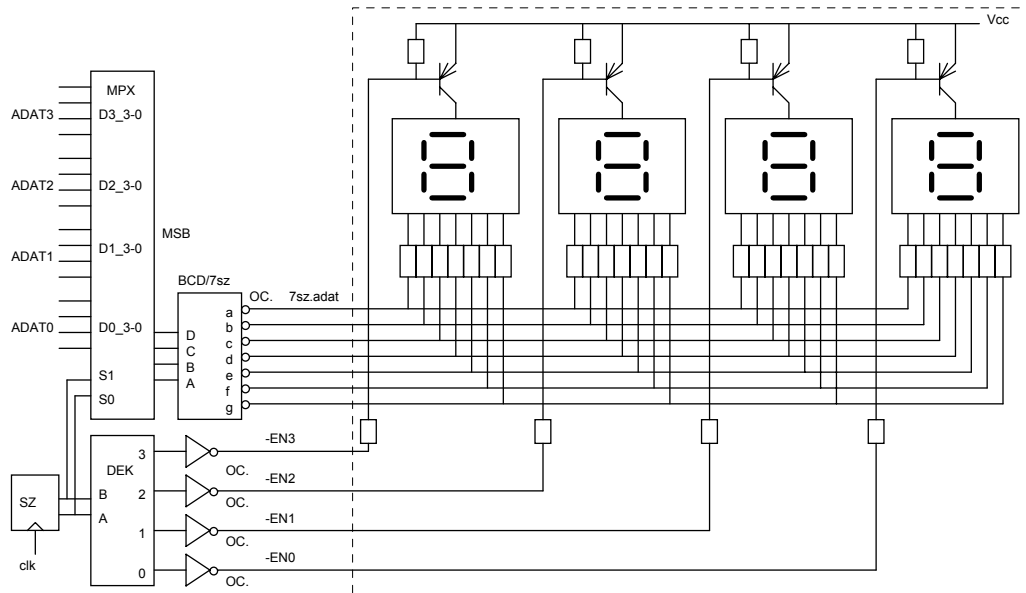


Példák funkcionális elemekkel felépített logikákra:

PWM (változtatható kitöltési tényezőjű jel) előállítás.



Időmultiplexált kijelző megvalósítása



Mátrix billentyűzet kezelése

