ÓBUDAI EGYETEM

7.hét: A sorrendi hálózatok elemei II.

Bevezetés

- Tárolók

Bevezetés

- Regiszterek
- Számlálók
- Memóriák

Regiszter

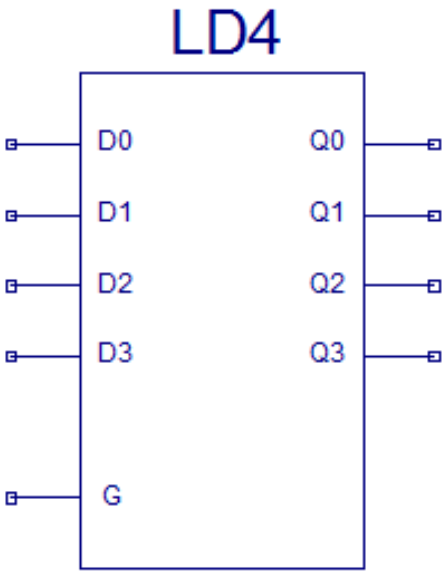
- DEFINÍCIÓ
- Tárolóegységek összekapcsolásával, egyszerű bemeneti kombinációs hálózattal kiegészítve

Regiszter

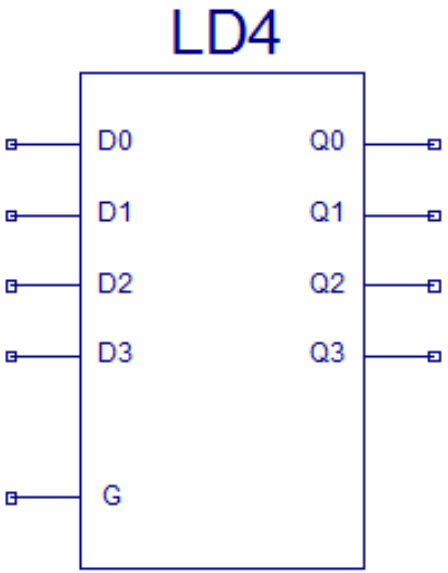
- DEFINÍCIÓ
- Tárolóegységek összekapcsolásával, egyszerű bemeneti kombinációs hálózattal kiegészítve

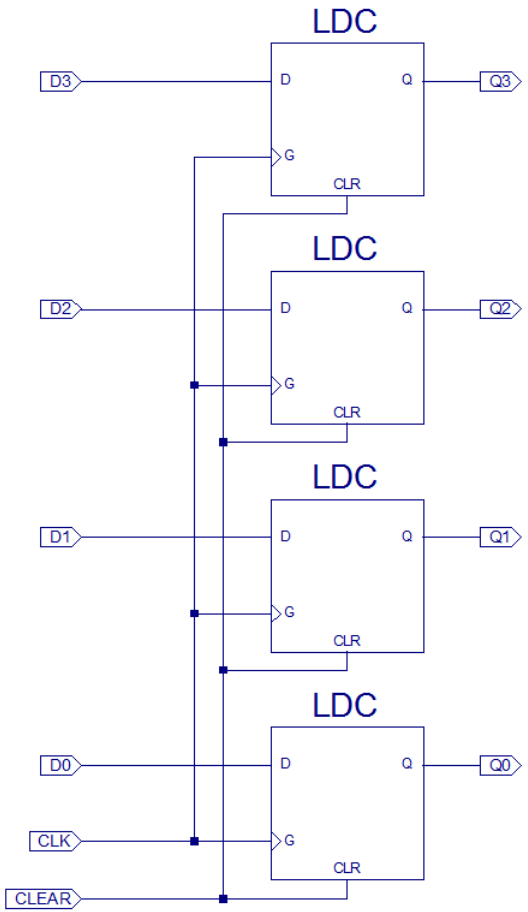
- FELADAT
- Átmeneti tárolás
- Léptetés
- Vezérlési feladatok
- Soros-párhuzamos, párhuzamos-soros átalakítás

Regiszter - tárolás



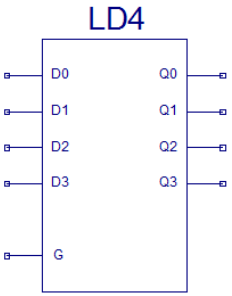
Regiszter - tárolás



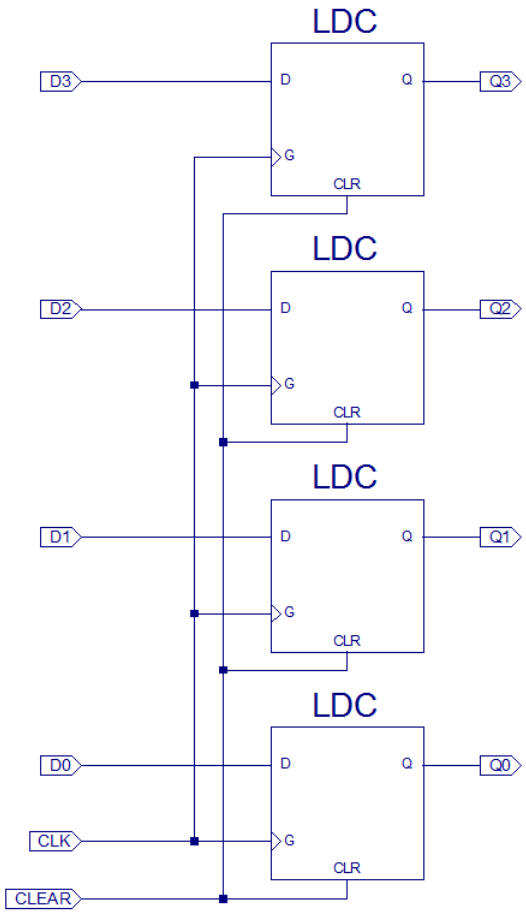


Regiszter - tárolás

- Átmeneti tárolás
  - Azonos órajelű D tárolókból épül fel
  - Több bites adatok átmeneti tárolására
    - Vezérlő információk
    - Műveletek operandusainak és eredményének tárolása




```
signal d : STD_LOGIC_VECTOR(3 downto 0);
signal q : STD_LOGIC_VECTOR(3 downto 0);
signal clk: STD_LOGIC;
…
process
Begin
    if (clear = '1') then
        q <= "0000";
    elsif (clk`event and clk = '1') then
       q <= d;
    end if;
end process;
```



Regiszter – léptetés

- Azonos órajelű D tárolókból épül fel
- A tárolók kimenete egy másik tároló bemenetére csatlakozik
- Az órajel hatására az információ az egyik tárolóból a másikba íródik

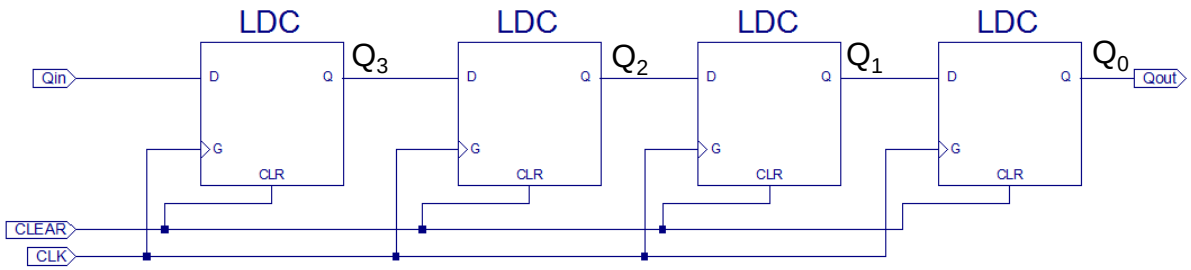
Regiszter – léptetés

- Azonos órajelű D tárolókból épül fel
- A tárolók kimenete egy másik tároló bemenetére csatlakozik
- Az órajel hatására az információ az egyik tárolóból a másikba íródik

- Léptetés
  - Jobbra
  - Balra
  - Két irányba

Regiszter – léptetés

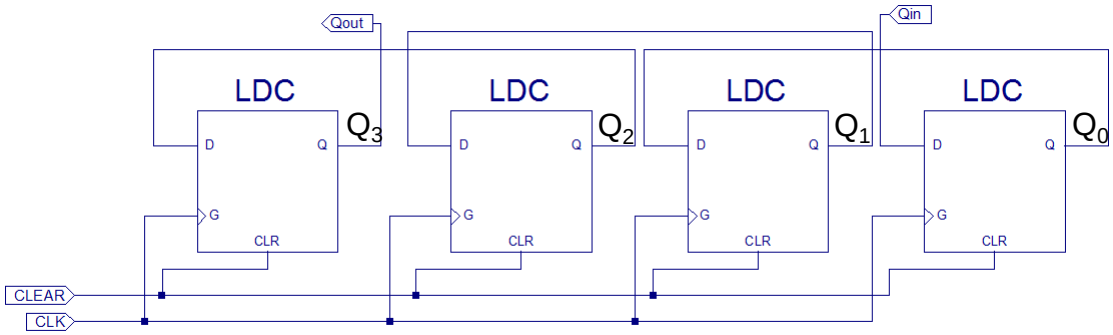
- Azonos órajelű D tárolókból épül fel
- A tárolók kimenete egy másik tároló bemenetére csatlakozik
- Az órajel hatására az információ az egyik tárolóból a másikba íródik
- Léptetőregiszter, shift-regiszter
- **Léptetés jobbra**




```
Begin
   if (clear = '1') then
      q <= "0000";
   elsif (clk`event and clk = '1') then
      q(2 downto 0) <= q(3 downto 1);
      q(3) <= qin;
   end if;
end;
```

Regiszter - léptetés

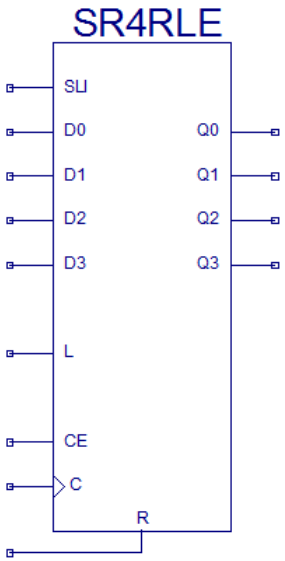
- Azonos órajelű D tárolókból épül fel
- A tárolók kimenete egy másik tároló bemenetére csatlakozik
- Az órajel hatására az információ az egyik tárolóból a másikba íródik
- Léptetőregiszter, shift-regiszter
- **Léptetés balra**

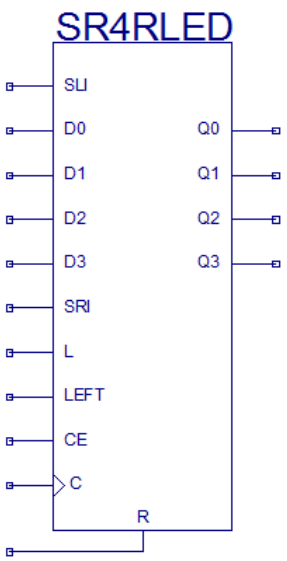



```
Begin
   if (clear = '1') then
      q <= "0000";
   elsif (clk`event and clk = '1') then
      q(3 downto 1) <= q(2 downto 0);
      q(0) <= qin;
   end if;
end;
```

Regiszter - léptetés

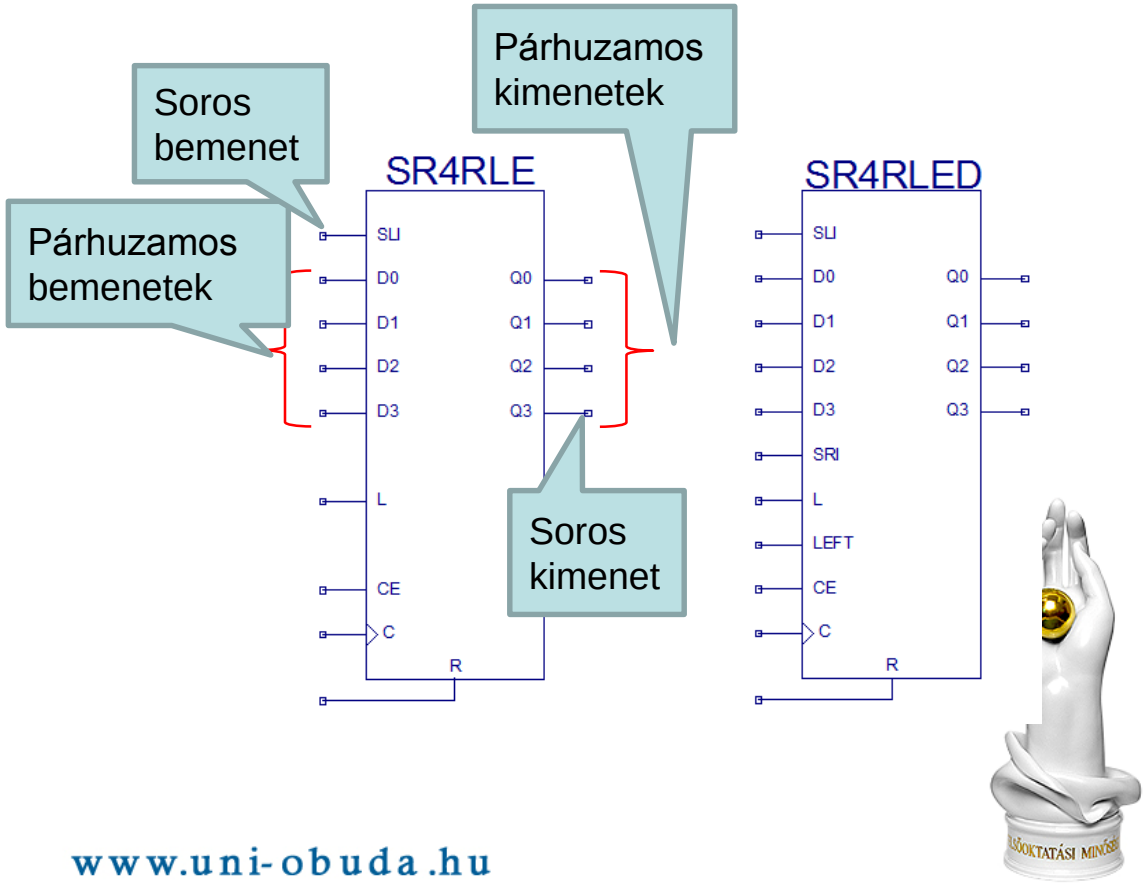
- Jobbra, balra léptetés





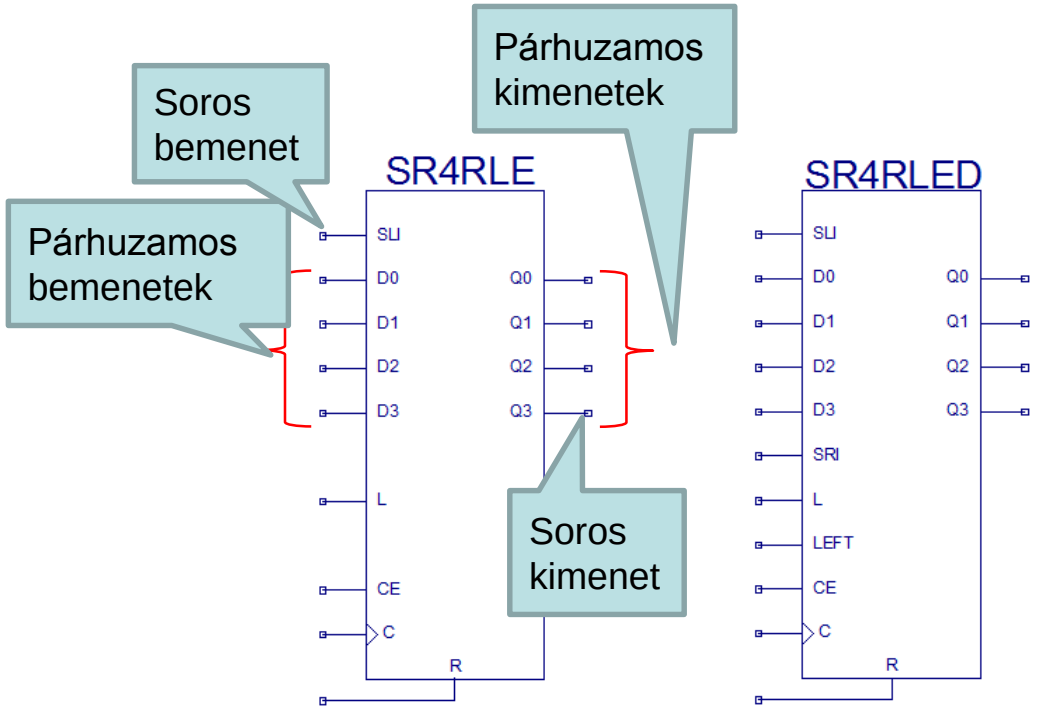
Regiszter - léptetés

- **Jobbra, balra léptetés**



Regiszter - léptetés

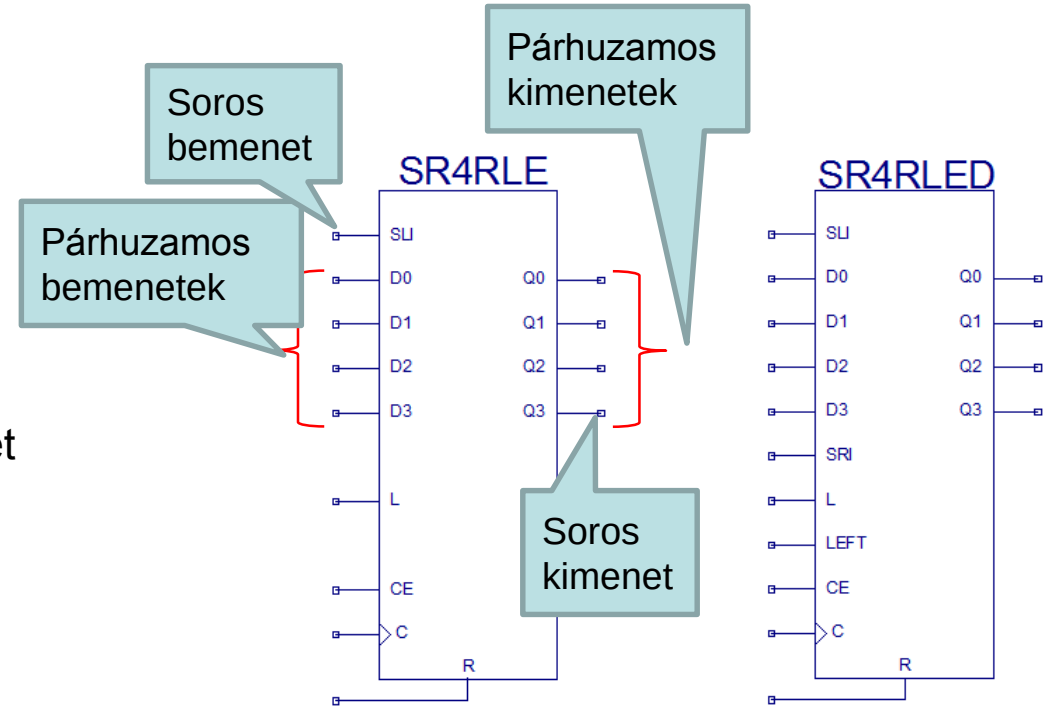
- **Jobbra, balra léptetés**
- Párhuzamos be- kimenet
- Soros be- kimenet
- Alkalmas soros/párhuzamos átalakításra



Regiszter - léptetés

- **Jobbra, balra léptetés**
- Párhuzamos be- kimenet
- Soros be- kimenet
- Alkalmas soros/párhuzamos átalakításra

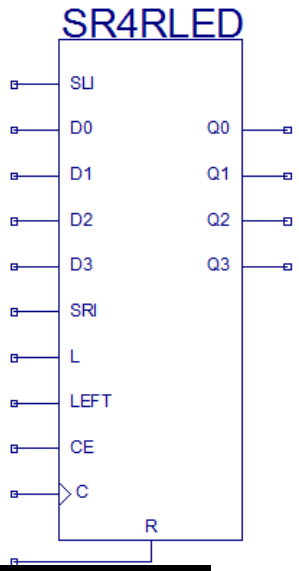
SLI: baloldali soros bemenet
SRI: jobboldali soros bemenet
D: párhuzamos bemenetek
Q: párhuzamos kimenetek
CE: órajel engedélyező bemenet
C: órajel
L: beírás engedélyezés
LEFT: balra/jobbra léptetés
R: szinkron törlés



Regiszter - léptetés

- Jobbra, balra léptetés
- Párhuzamos be- kimenet
- Soros be- kimenet
- Alkalmas soros/párhuzamos átalakításra

SLI: baloldali soros bemenet
SRI: jobboldali soros bemenet
D: párhuzamos bemenet
Q: párhuzamos kimenetek
CE: órajel engedélyező bemenet
C: órajel
L: beírás engedélyezés
LEFT: balra/jobbra léptetés
R: szinkron törlés



Inputs								Outputs		
R	L	CE	LEFT	SLI	SRI	D3 : D0	C	Q0	Q3	Q2 : Q1
1	X	X	X	X	X	X	↑	0	0	0
0	1	X	X	X	X	D3 : D0	↑	D0	D3	Dn
0	0	0	X	X	X	X	X	No Change	No Change	No Change
0	0	1	1	SLI	X	X	↑	SLI	q2	qn-1
0	0	1	0	X	SRI	X	↑	q1	SRI	qn+1
qn-1 or qn+1 = state of referenced output one setup time prior to active clock transition										

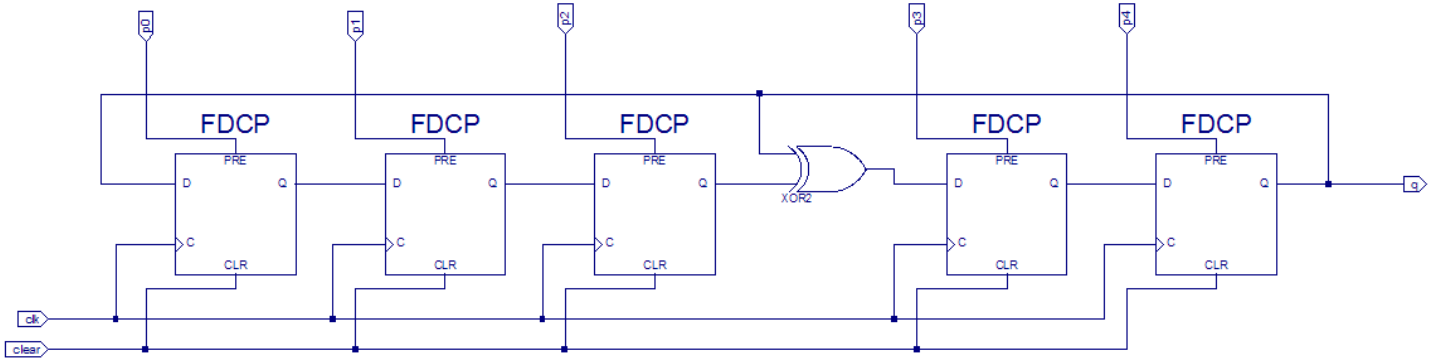
Regiszter - LFSR

- **Álvéletlen szám generátor léptetőregiszterrel (LFSR (Linear Feedback Shift Register)**
  - Bitminta generálás
  - Titkosítás
  - Hibavédelem

- Ha a regiszterek tartalma 0 ez az állapot marad
- Nem nulla kezdőállapot után véges hosszúságú periodikus jelet állít elő a kimeneten
- A periódus hossz maximum 2n-1 (n a regiszterek száma)

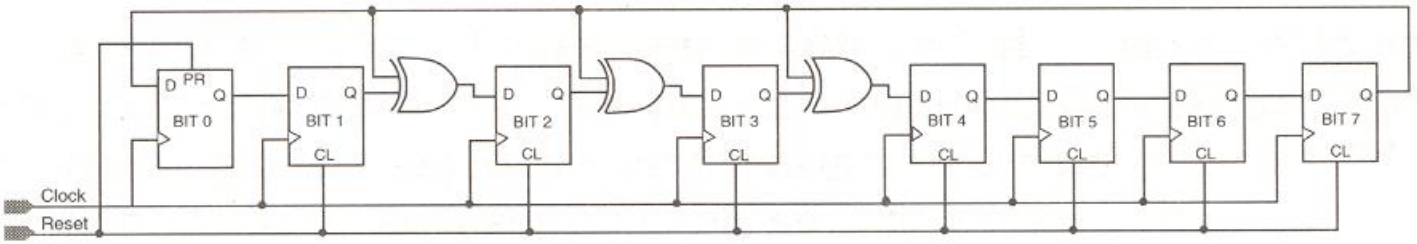
Regiszter - LFSR

- **Álvéletlen szám generátor léptetőregiszterrel (LFSR (Linear Feedback Shift Register)**
  - Bitminta generálás
  - Titkosítás
  - Hibavédelem

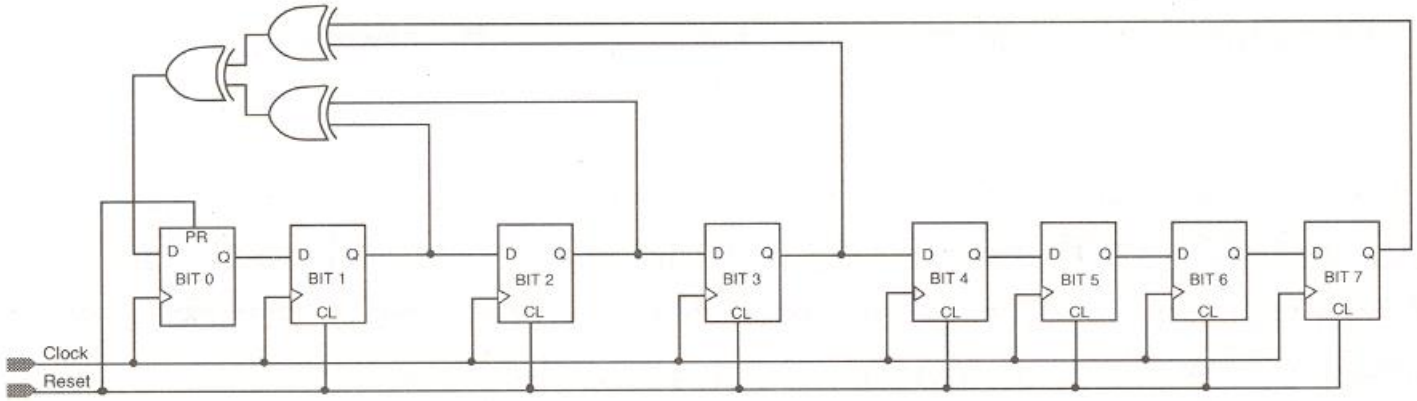


- Ha a regiszterek tartalma 0 ez az állapot marad
- Nem nulla kezdőállapot után véges hosszúságú periodikus jelet állít elő a kimeneten
- A periódus hossz maximum 2n-1 (n a regiszterek száma)

Regiszter – LFSR



(a) One-to-many



Note. Uses XOR gates therefore all 0s not in sequence so not reset to all 0s.

(b) Many-to-one

Regiszter – LFSR

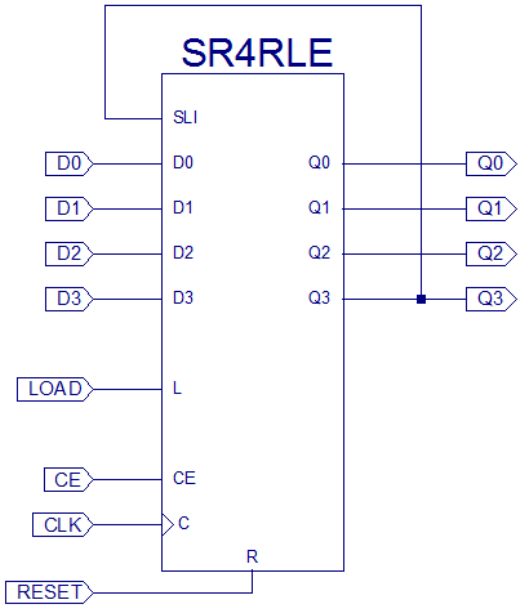
n	XNOR from	n	XNOR from	n	XNOR from	n	XNOR from
3	3,2	45	45,44,42,41	87	87,74	129	129,124
4	4,3	46	46,45,26,25	88	88,87,17,16	130	130,127
5	5,3	47	47,42	89	89,51	131	131,130,84,83
6	6,5	48	48,47,21,20	90	90,89,72,71	132	132,103
7	7,6	49	49,40	91	91,90,8,7	133	133,132,82,81
8	8,6,5,4	50	50,49,24,23	92	92,91,80,79	134	134,77
9	9,5	51	51,50,36,35	93	93,91	135	135,124
10	10,7	52	52,49	94	94,73	136	136,135,11,10
11	11,9	53	53,52,38,37	95	95,84	137	137,116
12	12,6,4,1	54	54,53,18,17	96	96,94,49,47	138	138,137,131,130
13	13,4,3,1	55	55,31	97	97,91	139	139,136,134,131
14	14,5,3,1	56	56,55,35,34	98	98,87	140	140,111
15	15,14	57	57,50	99	99,97,54,52	141	141,140,110,109
16	16,15,13,4	58	58,39	100	100,63	142	142,121
17	17,14	59	59,58,38,37	101	101,100,95,94	143	143,142,123,122
18	18,11	60	60,59	102	102,101,36,35	144	144,143,75,74
19	19,6,2,1	61	61,60,46,45	103	103,94	145	145,93
20	20,17	62	62,61,6,5	104	104,103,94,93	146	146,145,87,86
21	21,19	63	63,62	105	105,89	147	147,146,110,109
22	22,21	64	64,63,61,60	106	106,91	148	148,121

Számlálók

- Összeadók
- Számlálók
- kivonók
- Számábrázolás HEXA

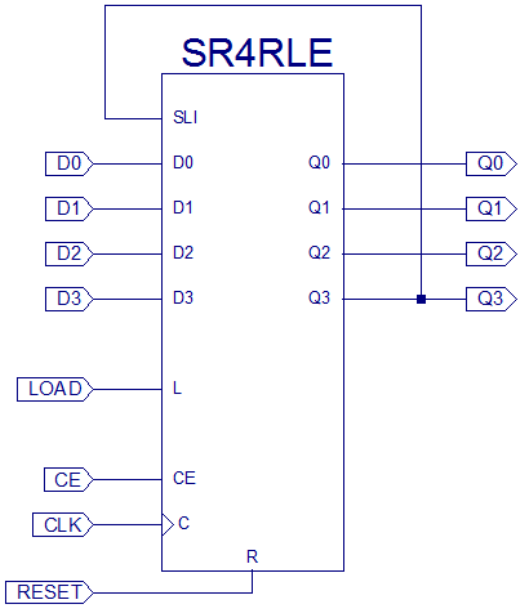
Számlálók

- Gyűrűs-számláló



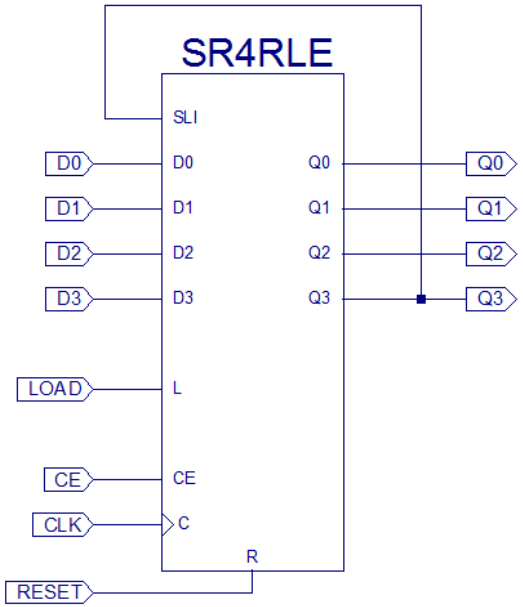
Számlálók

- **Gyűrűs-számláló**
- A LOAD bemenettel a D3-D0 bemeneteket 0001 alaphelyzetbe állítjuk
- Az órajel engedélyezése után minden órajel ciklusban az 1-es továbblép a következő helyi értékre
- A visszacsatolás miatt 4 ciklus után újra kezdődik a folyamat
- A 2n (16) lehetséges állapotból csak n (4) valósul meg (12 tiltott állapot)



Számlálók

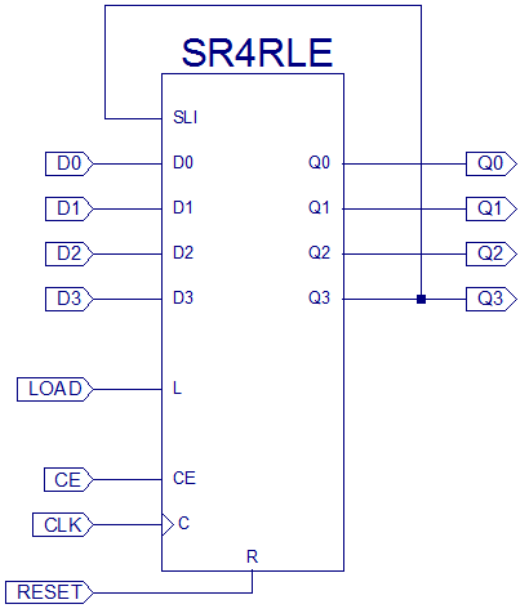
- **Gyűrűs-számláló**
- A LOAD bemenettel a D3-D0 bemeneteket 0001 alaphelyzetbe állítjuk
- Az órajel engedélyezése után minden órajel ciklusban az 1-es továbblép a következő helyi értékre
- A visszacsatolás miatt 4 ciklus után újra kezdődik a folyamat
- A 2n (16) lehetséges állapotból csak n (4) valósul meg (12 tiltott állapot)



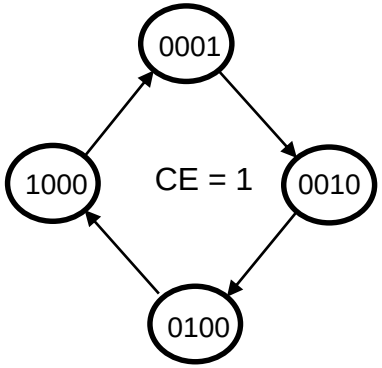
Q_3	Q_2	Q_1	Q_0
0	0	0	1
0	0	1	0
0	1	0	0
1	0	0	0
0	0	0	1
0	0	1	0
…			

Számlálók

- **Gyűrűs-számláló**
- A LOAD bemenettel a D3-D0 bemeneteket 0001 alaphelyzetbe állítjuk
- Az órajel engedélyezése után minden órajel ciklusban az 1-es továbblép a következő helyi értékre
- A visszacsatolás miatt 4 ciklus után újra kezdődik a folyamat
- A 2n (16) lehetséges állapotból csak n (4) valósul meg (12 tiltott állapot)



Q_3	Q_2	Q_1	Q_0
0	0	0	1
0	0	1	0
0	1	0	0
1	0	0	0
0	0	0	1
0	0	1	0
…			

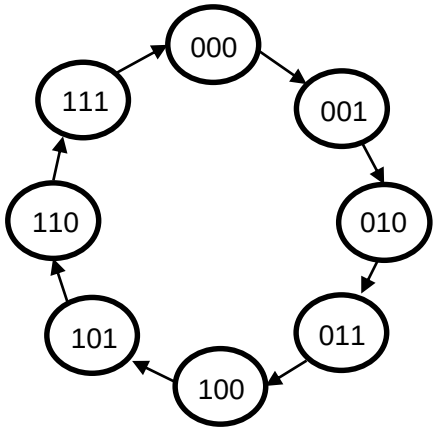


Bináris felfelé számláló

- Pl. 0-tól 7-ig (000b – 111b) egyesével számlál minden órajel ciklusban (3-bites)
- A hálózat $2^3 = 8$ állapotot vehet fel, nincs tiltott állapot

Bináris felfelé számláló

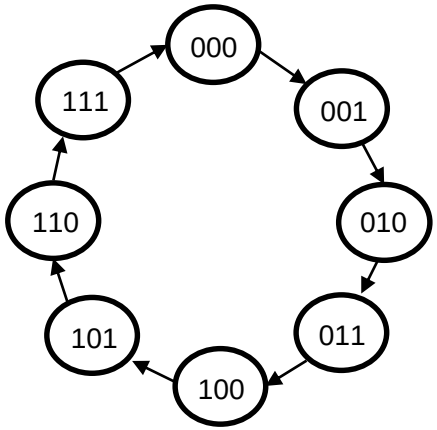
- Pl. 0-tól 7-ig (000b – 111b) egyesével számlál minden órajel ciklusban (3-bites)
- A hálózat $2^3 = 8$ állapotot vehet fel, nincs tiltott állapot



Bináris felfelé számláló

- Pl. 0-tól 7-ig (000b – 111b) egyesével számlál minden órajel ciklusban (3-bites)
- A hálózat $2^3 = 8$ állapotot vehet fel, nincs tiltott állapot

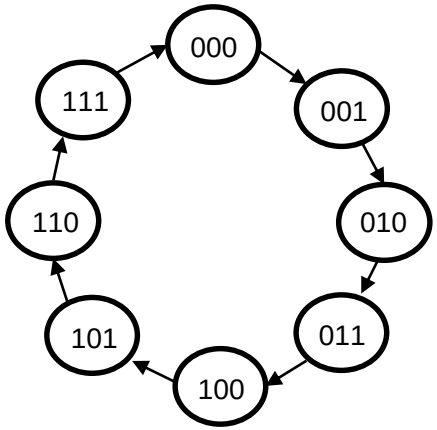
Q_2	Q_1	Q_0	
0	0	0	(0)
0	0	1	(1)
0	1	0	(2)
0	1	1	(3)
1	0	0	(4)
1	0	1	(5)
1	1	0	(6)
1	1	1	(7)

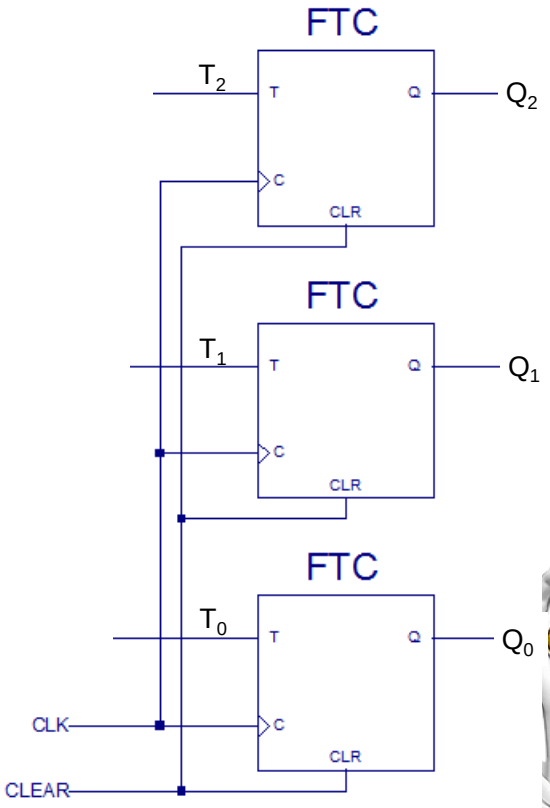


Bináris felfelé számláló

- Pl. 0-tól 7-ig (000b – 111b) egyesével számlál minden órajel ciklusban (3-bites)
- A hálózat $2^3 = 8$ állapotot vehet fel, nincs tiltott állapot

Q_2	Q_1	Q_0	
0	0	0	(0)
0	0	1	(1)
0	1	0	(2)
0	1	1	(3)
1	0	0	(4)
1	0	1	(5)
1	1	0	(6)
1	1	1	(7)

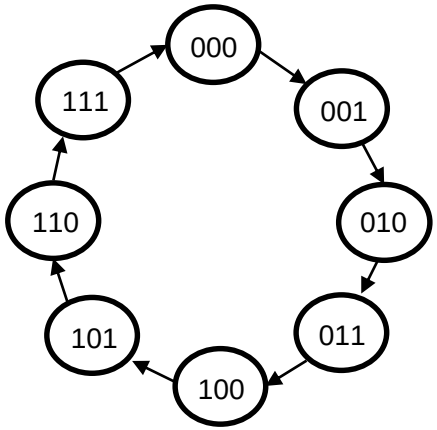


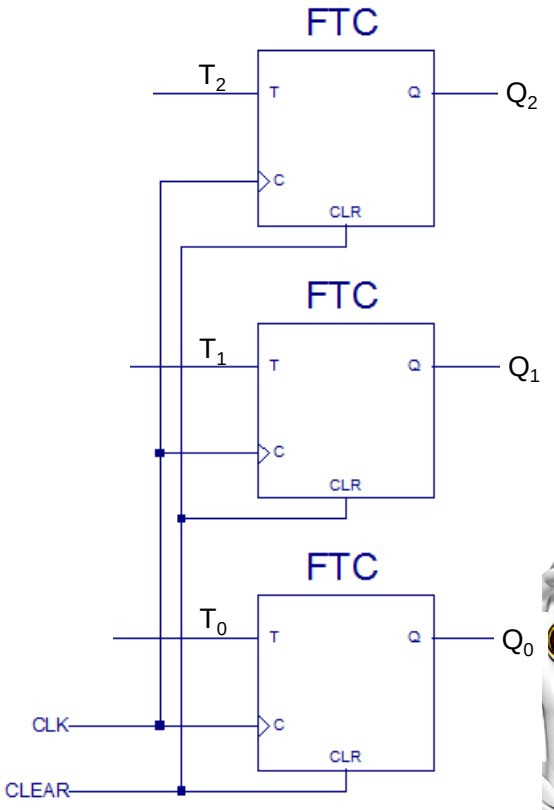


Bináris felfelé számláló

- Pl. 0-tól 7-ig (000b – 111b) egyesével számlál minden órajel ciklusban (3-bites)
- A hálózat $2^3 = 8$ állapotot vehet fel, nincs tiltott állapot

Q_2	Q_1	Q_0	
0	0	0	(0)
0	0	1	(1)
0	1	0	(2)
0	1	1	(3)
1	0	0	(4)
1	0	1	(5)
1	1	0	(6)
1	1	1	(7)





- A Q_0 bit minden órajelre billen
  - T-tároló T = 1 állandó bemenettel
- A Q_1 bit akkor billen a következő órajelre ha $Q_0 = 1$
  - T-tároló T = Q_0 állandó bemenettel
- A Q_2 bit akkor billen a következő órajelre ha $Q_1Q_0 = 1$
  - T-tároló T = (Q_0 ÉS Q_1) állandó bemenettel

Bináris felfelé számláló

- Pl. 0-tól 7-ig (000b – 111b) egyesével számlál minden órajel ciklusban (3-bites)
- A hálózat $2^3 = 8$ állapotot vehet fel, nincs tiltott állapot
- VHDL:

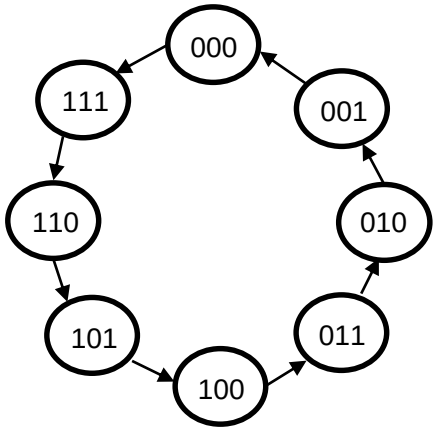
```
use IEE.STD_LOGIC_ARITH.ALL;
…
signal q : STD_LOGIC_VECTOR(2 downto 0);
signal clk: STD_LOGIC;
signal clear: STD_LOGIC;
…
process
Begin
  if (clear = '1') then
    q <= "000";
  elsif (clk`event and clk = '1') then
     q <= q+1;
  end if;
end process;
```

Bináris lefelé számláló

- Pl. 7-től 0-ig (111b – 000b) egyesével számlál minden órajel ciklusban (3-bites)
- A hálózat $2^3 = 8$ állapotot vehet fel, nincs tiltott állapot

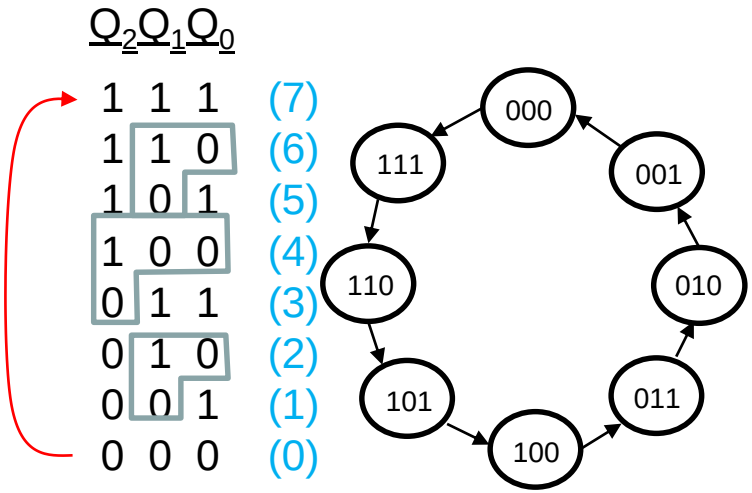
Bináris lefelé számláló

- Pl. 7-től 0-ig (111b – 000b) egyesével számlál minden órajel ciklusban (3-bites)
- A hálózat $2^3 = 8$ állapotot vehet fel, nincs tiltott állapot



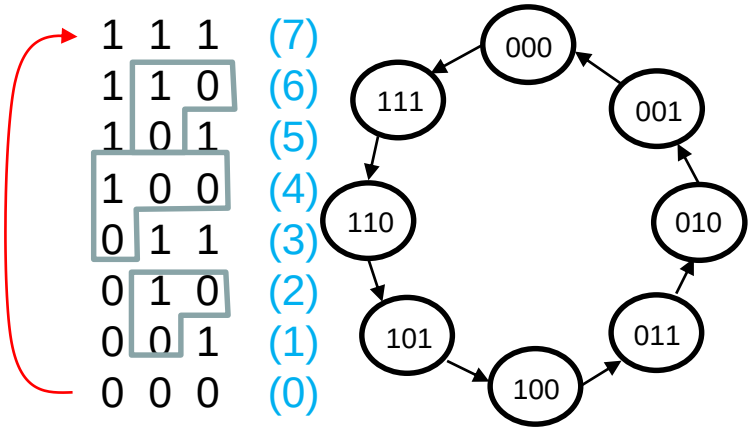
Bináris lefelé számláló

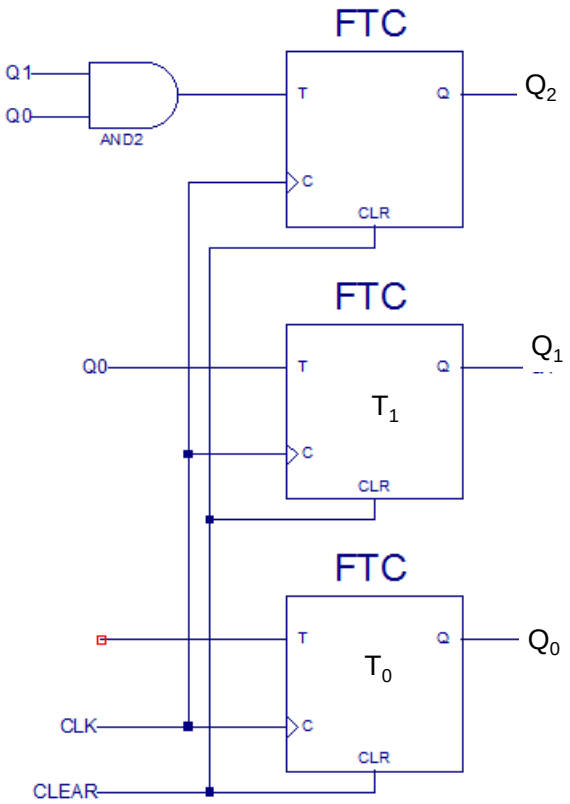
- Pl. 7-től 0-ig (111b – 000b) egyesével számlál minden órajel ciklusban (3-bites)
- A hálózat $2^3 = 8$ állapotot vehet fel, nincs tiltott állapot



Bináris lefelé számláló

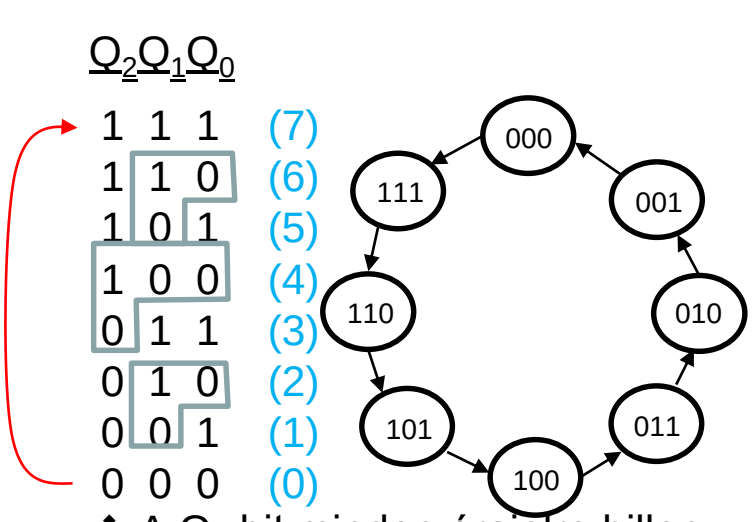
- Pl. 7-től 0-ig (111b – 000b) egyesével számlál minden órajel ciklusban (3-bites)
- A hálózat $2^3 = 8$ állapotot vehet fel, nincs tiltott állapot

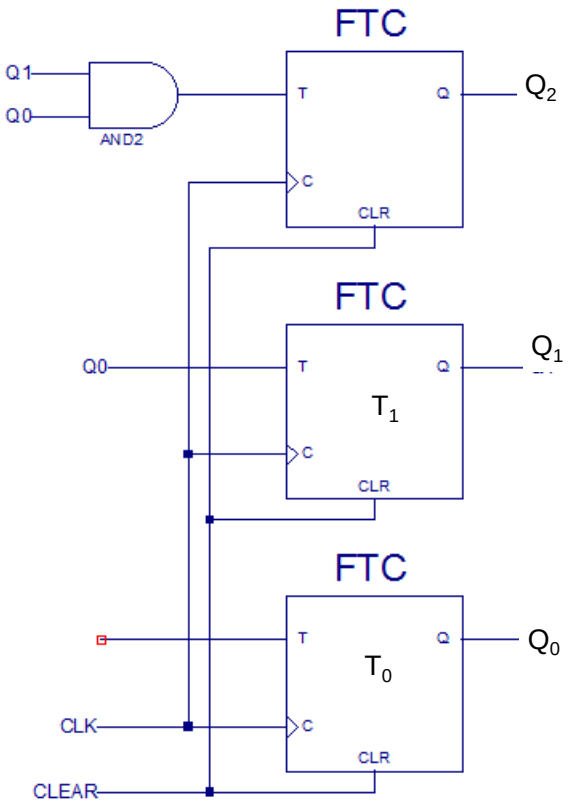




Bináris lefelé számláló

- Pl. 7-től 0-ig (111b – 000b) egyesével számlál minden órajel ciklusban (3-bites)
- A hálózat $2^3 = 8$ állapotot vehet fel, nincs tiltott állapot





- A Q_0 bit minden órajelre billen
  - T-tároló T = 1 állandó bemenettel
- A Q_1 bit akkor billen a következő órajelre ha $Q_0 = 0$
  - T-tároló T = Q_0 állandó bemenettel
- A Q_2 bit akkor billen a következő órajelre ha $Q_1Q_0 = 00$
  - T-tároló T = (Q_0 ÉS $\overline{Q_1}$) állandó bemenettel

Bináris lefelé számláló

- Pl. 7-től 0-ig (111b – 000b) egyesével számlál minden órajel ciklusban (3-bites)
- A hálózat $2^3 = 8$ állapotot vehet fel, nincs tiltott állapot
- VHDL:

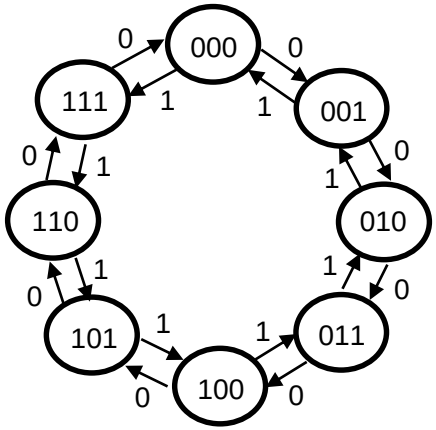
```
use IEE.STD_LOGIC_ARITH.ALL;
…
signal q : STD_LOGIC_VECTOR(2 downto 0);
signal clk: STD_LOGIC;
signal clear: STD_LOGIC;
…
process
Begin
  if (clear = '1') then
    q <= "000";
  elsif (clk`event and clk = '1') then
     q <= q-1;
  end if;
end process;
```

Bináris fel – le számláló

- Az számlálási irányt megadó X bemenet függvényében
  - Pl. 7-től 0-ig (111b – 000b) egyesével számlál minden órajel ciklusban (3-bites)
  - Pl. 0-tól 7-ig (000b – 111b) egyesével számlál minden órajel ciklusban (3-bites)

Bináris fel – le számláló

- Az számlálási irányt megadó X bemenet függvényében
  - Pl. 7-től 0-ig (111b – 000b) egyesével számlál minden órajel ciklusban (3-bites)
  - Pl. 0-tól 7-ig (000b – 111b) egyesével számlál minden órajel ciklusban (3-bites)



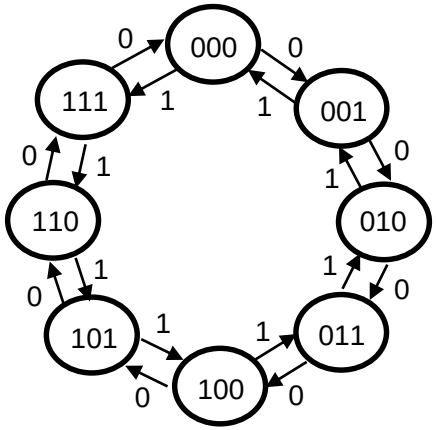
Bináris fel – le számláló

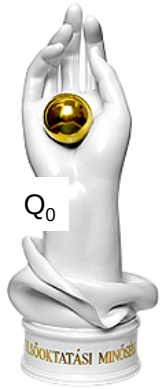
- Az számlálási irányt megadó X bemenet függvényében
  - Pl. 7-től 0-ig (111b – 000b) egyesével számlál minden órajel ciklusban (3-bites)
  - Pl. 0-tól 7-ig (000b – 111b) egyesével számlál minden órajel ciklusban (3-bites)

$$T_2 = Q_1 \cdot Q_0 \cdot \bar{X} + \bar{Q}_1 \cdot \bar{Q}_0 \cdot X$$

$$T_1 = Q_0 \cdot \bar{X} + \bar{Q}_0 \cdot X$$

$$T_0 = 1$$





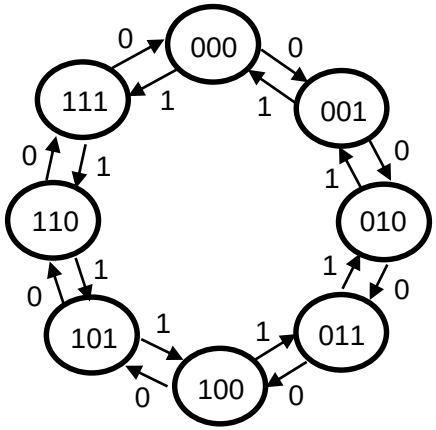
Bináris fel – le számláló

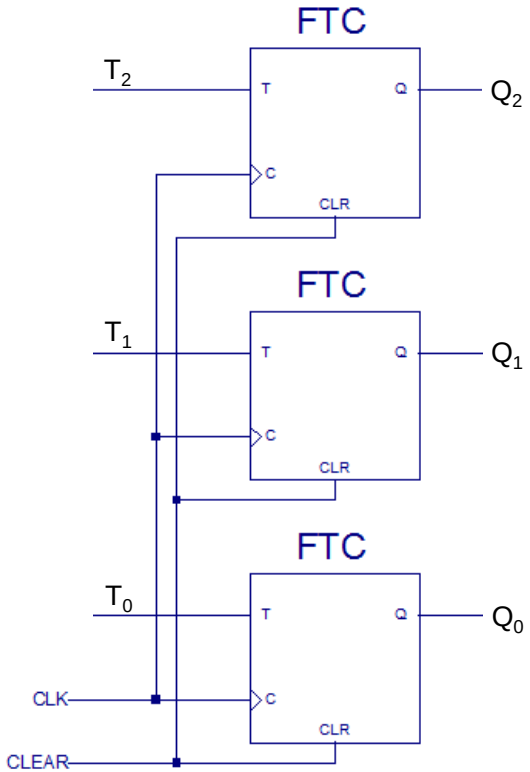
- Az számlálási irányt megadó X bemenet függvényében
  - Pl. 7-től 0-ig (111b – 000b) egyesével számlál minden órajel ciklusban (3-bites)
  - Pl. 0-tól 7-ig (000b – 111b) egyesével számlál minden órajel ciklusban (3-bites)

$$T_2 = Q_1 \cdot Q_0 \cdot \bar{X} + \bar{Q}_1 \cdot \bar{Q}_0 \cdot X$$

$$T_1 = Q_0 \cdot \bar{X} + \bar{Q}_0 \cdot X$$

$$T_0 = 1$$



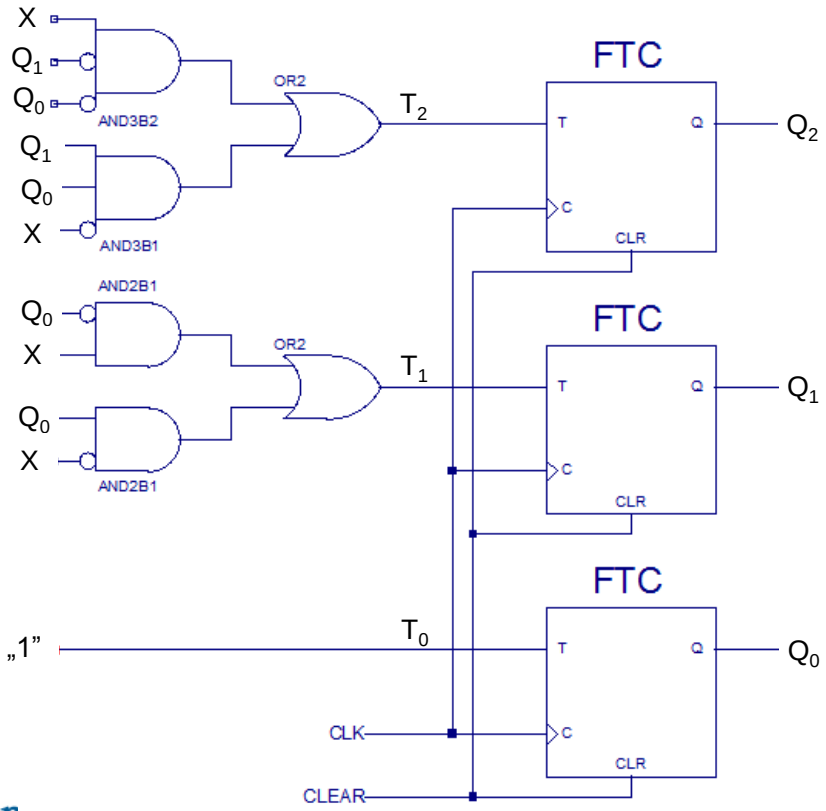


Bináris fel – le számláló

$$T_2 = Q_1 \cdot Q_0 \cdot \bar{X} + \bar{Q}_1 \cdot \bar{Q}_0 \cdot X$$

$$T_1 = Q_0 \cdot \bar{X} + \bar{Q}_0 \cdot X$$

$$T_0 = 1$$



Bináris fel – le számláló

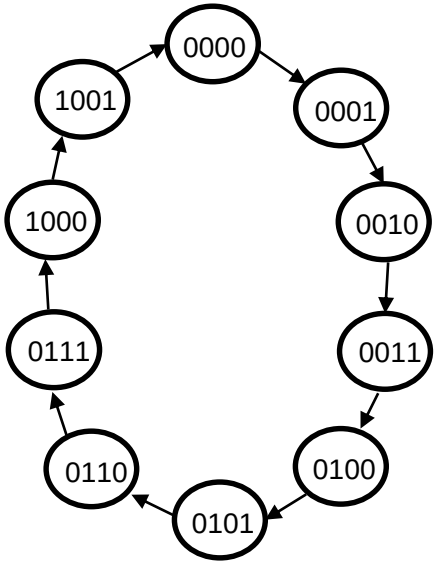
```
use IEE.STD_LOGIC_ARITH.ALL;
…
signal q : STD_LOGIC_VECTOR(2 downto 0);
signal x: STD_LOGIC;
signal clk: STD_LOGIC;
signal clear: STD_LOGIC;
…
process
Begin
  if (clear = '1') then
    q <= "000";
  elsif (clk`event and clk = '1') then
     if (x = '0') then q <= q+1;
    else q <= q-1;
    end if;
  end if;
end process;
```

Decimális felfelé számláló

- Pl. 0-tól 9-ig (0000b – 1001b) egyesével számlál minden órajel ciklusban (4-bites BCD kód)
- A hálózat 2^4 = 16 állapotot vehet fel, ebből 6 tiltott állapot
- 1001b-ig úgy működik mint egy bináris számláló, de 1001b után 0000b kell következzen
- A bemeneti kombinációs hálózat a bináris számlálótól különböző lesz

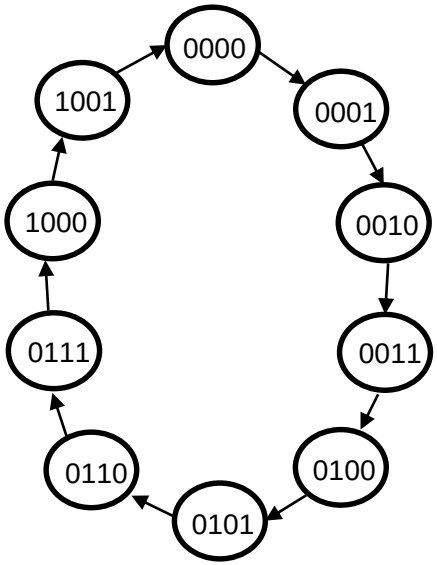
Decimális felfelé számláló

- Pl. 0-tól 9-ig (0000b – 1001b) egyesével számlál minden órajel ciklusban (4-bites BCD kód)
- A hálózat $2^4 = 16$ állapotot vehet fel, ebből 6 tiltott állapot
- 1001b-ig úgy működik mint egy bináris számláló, de 1001b után 0000b kell következzen
- A bemeneti kombinációs hálózat a bináris számlálótól különböző lesz



Decimális felfelé számláló

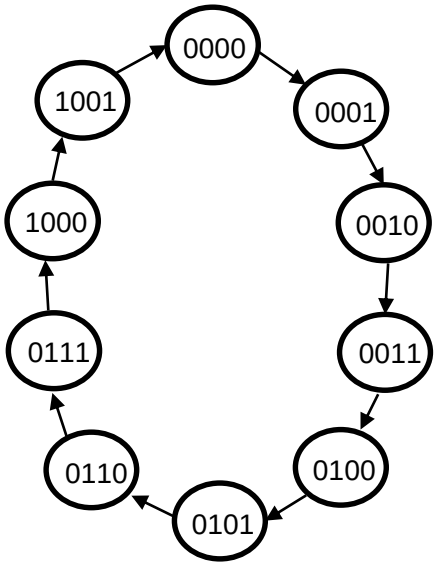
- Pl. 0-tól 9-ig (0000b – 1001b) egyesével számlál minden órajel ciklusban (4-bites BCD kód)
- A hálózat $2^4 = 16$ állapotot vehet fel, ebből 6 tiltott állapot
- 1001b-ig úgy működik mint egy bináris számláló, de 1001b után 0000b kell következzen
- A bemeneti kombinációs hálózat a bináris számlálótól különböző lesz



Q_3	Q_2	Q_1	Q_0	
0	0	0	0	(0)
0	0	0	1	(1)
0	0	1	0	(2)
0	0	1	1	(3)
0	1	0	0	(4)
0	1	0	1	(5)
0	1	1	0	(6)
0	1	1	1	(7)
1	0	0	0	(8)
1	0	0	1	(9)

Decimális felfelé számláló

- Pl. 0-tól 9-ig (0000b – 1001b) egyesével számlál minden órajel ciklusban (4-bites BCD kód)
- A hálózat $2^4 = 16$ állapotot vehet fel, ebből 6 tiltott állapot
- 1001b-ig úgy működik mint egy bináris számláló, de 1001b után 0000b kell következzen
- A bemeneti kombinációs hálózat a bináris számlálótól különböző lesz



Q_3	Q_2	Q_1	Q_0	
0	0	0	0	(0)
0	0	0	1	(1)
0	0	1	0	(2)
0	0	1	1	(3)
0	1	0	0	(4)
0	1	0	1	(5)
0	1	1	0	(6)
0	1	1	1	(7)
1	0	0	0	(8)
1	0	0	1	(9)

$$T_0 = 1$$

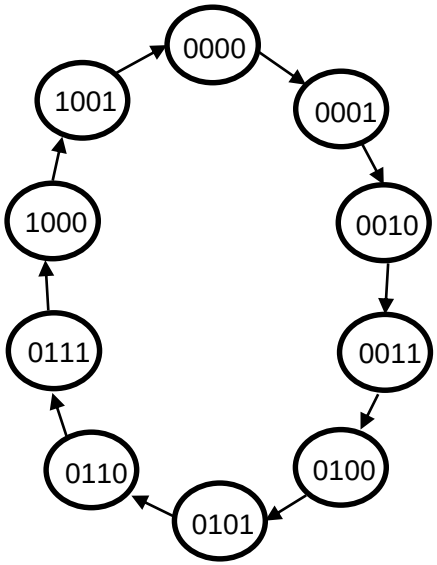
$$T_1 = Q_0 \cdot \overline{Q_0 \cdot Q_3}$$

$$T_2 = Q_0 \cdot Q_1$$

$$T_3 = Q_0 \cdot Q_1 \cdot Q_2 + Q_0 \cdot Q_3$$

Decimális felfelé számláló

- Pl. 0-tól 9-ig (0000b – 1001b) egyesével számlál minden órajel ciklusban (4-bites BCD kód)
- A hálózat $2^4 = 16$ állapotot vehet fel, ebből 6 tiltott állapot



$$T_3 = Q_0 \cdot Q_1 \cdot Q_2 + Q_0 \cdot Q_3$$

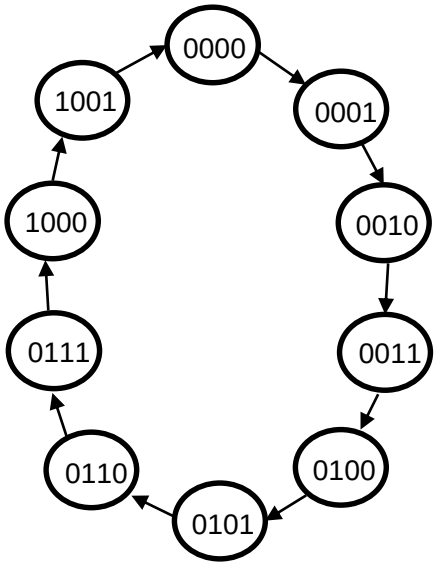
$$T_2 = Q_0 \cdot Q_1$$

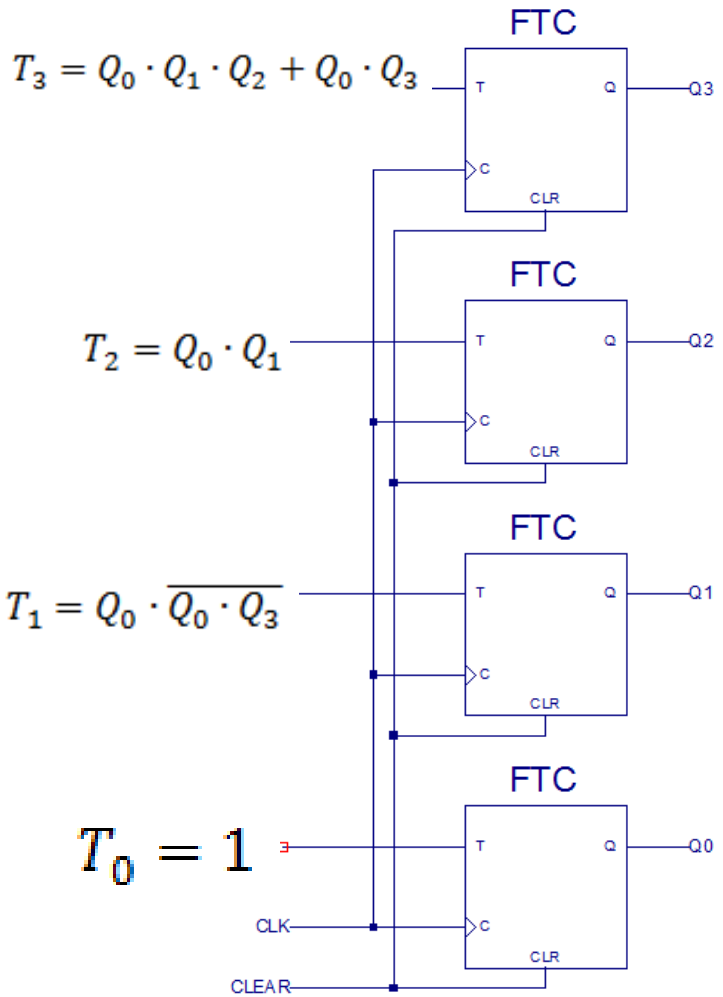
$$T_1 = Q_0 \cdot \overline{Q_0 \cdot Q_3}$$

$$T_0 = 1$$

Decimális felfelé számláló

- Pl. 0-tól 9-ig (0000b – 1001b) egyesével számlál minden órajel ciklusban (4-bites BCD kód)
- A hálózat 2^4 = 16 állapotot vehet fel, ebből 6 tiltott állapot



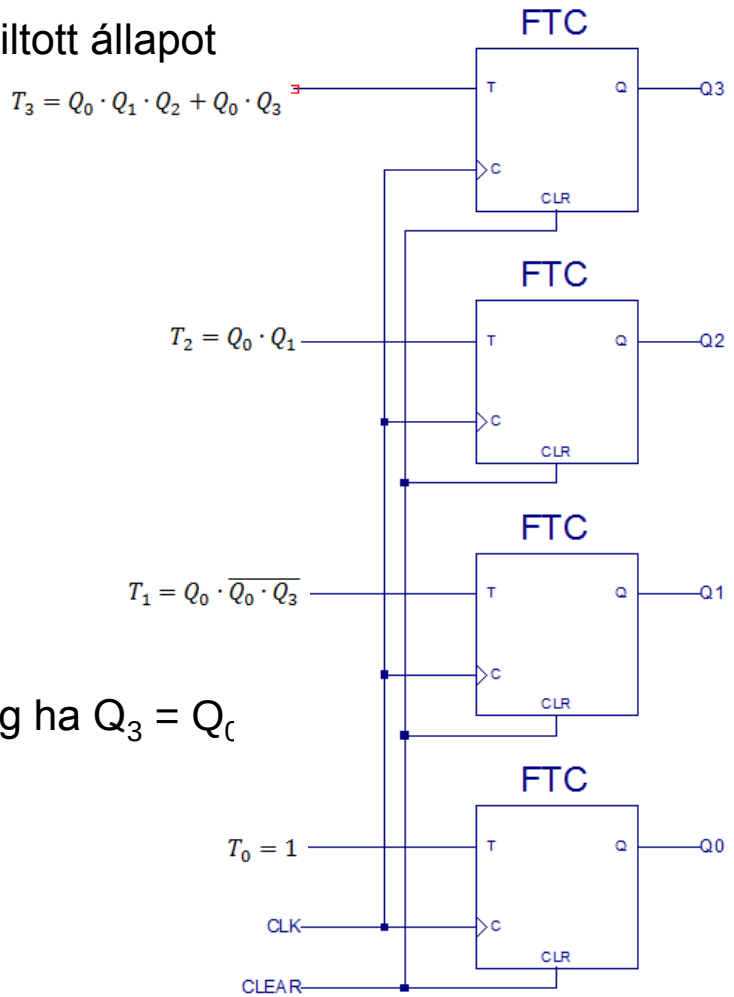


Decimális felfelé számláló

- Pl. 0-tól 9-ig (0000b – 1001b) egyesével számlál minden órajel ciklusban (4-bites BCD kód)
- A hálózat $2^4 = 16$ állapotot vehet fel, ebből 6 tiltott állapot

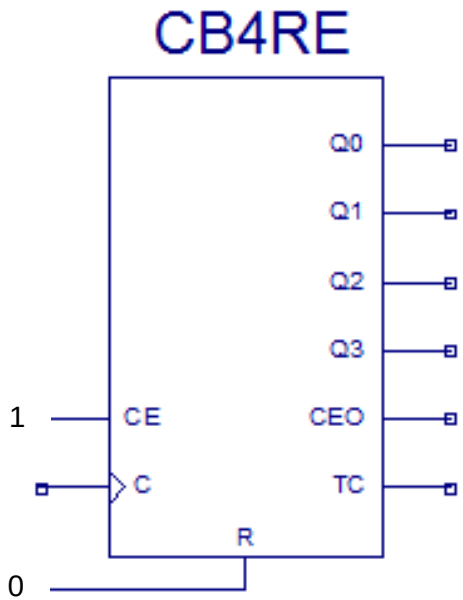
- Q_3 kimenet billen ha
  - $Q_0 = Q_1 = Q_2 = 1$ vagy
  - $Q_3Q_2Q_1Q_0 = 1001$
- Q_2 kimenet billen ha
  - $Q_0 = Q_1 = 1$
- Q_1 kimenet billen ha
  - $Q_0 = 1$ és
  - $Q_3Q_2Q_1Q_0 \neq 1001$

- Az ”1001” állapot jelzéséhez BCD kódban elég ha $Q_3 = Q_0$



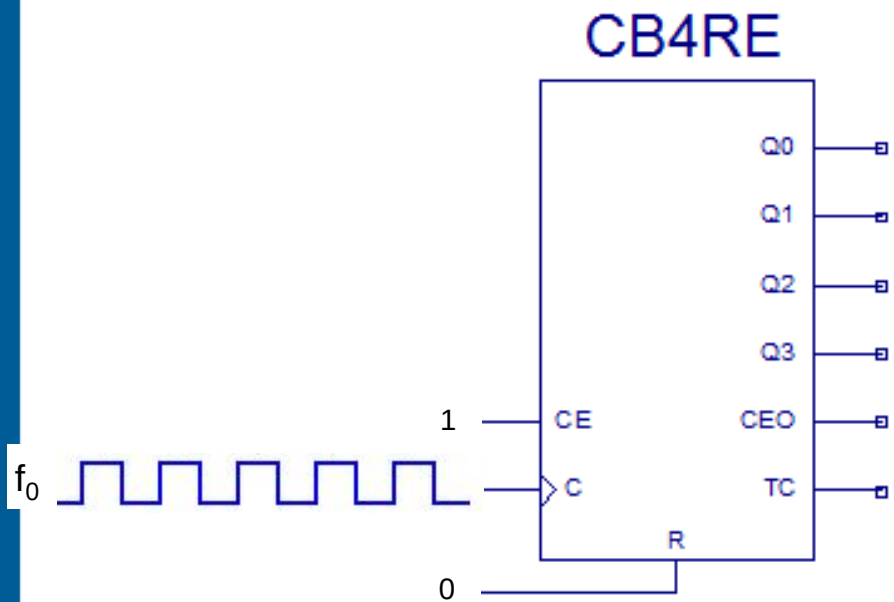
Frekvenciaosztás számlálóval

- n bites számláló frekvenciaosztása: 2^n



Frekvenciaosztás számlálóval

- n bites számláló frekvenciaosztása: 2^n



f_0

$f_0/2$

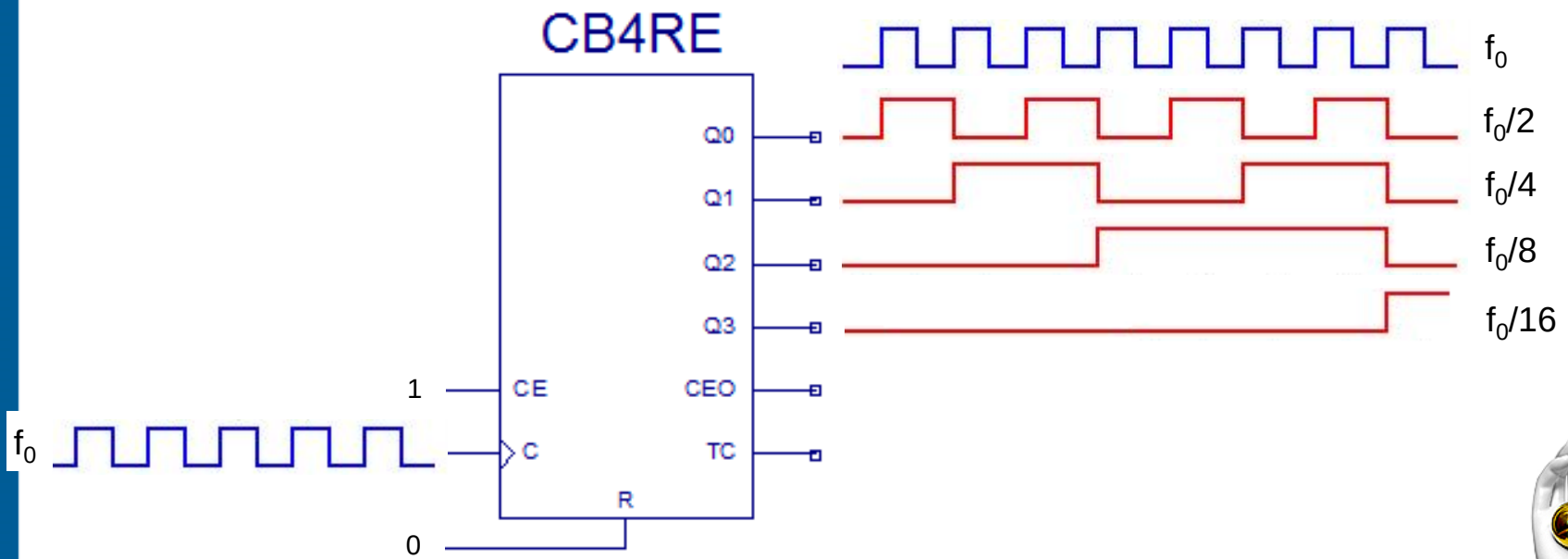
$f_0/4$

$f_0/8$

$f_0/16$

Frekvenciaosztás számlálóval

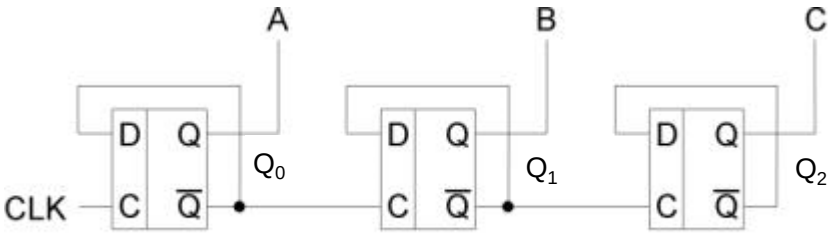
- n bites számláló frekvenciaosztása: 2^n



Aszinkron bináris felfelé számláló

- Pl. 0-tól 7-ig (000b – 111b) egyesével számlál minden órajel ciklusban (3-bites)
- A hálózat $2^3 = 8$ állapotot vehet fel, nincs tiltott állapot

Q_2	Q_1	Q_0	
0	0	0	(0)
0	0	1	(1)
0	1	0	(2)
0	1	1	(3)
1	0	0	(4)
1	0	1	(5)
1	1	0	(6)
1	1	1	(7)

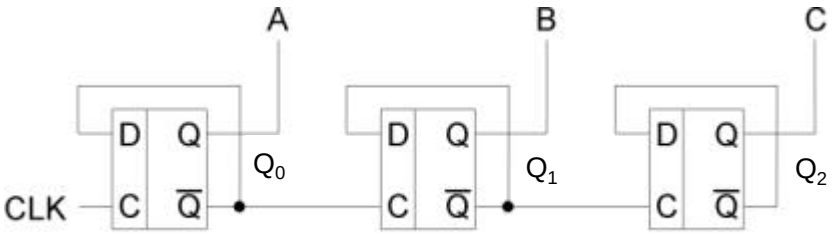


- A T tárolók kimenete a következő tároló órajel bemenetére csatlakozik
- A tárolók kimeneti jelének frekvenciája fele a bemeneti órajel frekvenciájának
- Előny:
  - Nem kellenek kiegészítő kapuáramkörök
- Hátrány:
  - A késleltetések miatt a tárolók nem egyszerre billennek
  - Az órajel változásakor rövid időre határozatlan kimeneti jel

Aszinkron bináris felfelé számláló

- Pl. 0-tól 7-ig (000b – 111b) egyesével számlál minden órajel ciklusban (3-bites)
- A hálózat $2^3 = 8$ állapotot vehet fel, nincs tiltott állapot

Q_2	Q_1	Q_0	
0	0	0	(0)
0	0	1	(1)
0	1	0	(2)
0	1	1	(3)
1	0	0	(4)
1	0	1	(5)
1	1	0	(6)
1	1	1	(7)

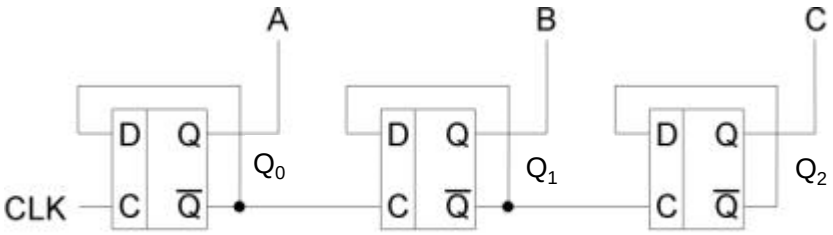


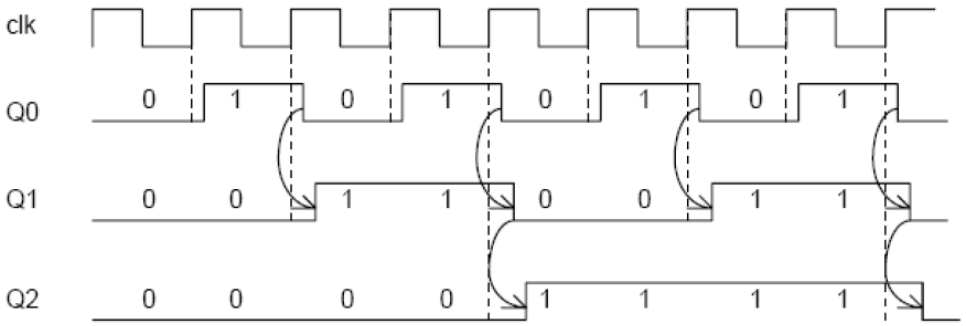
- A T tárolók kimenete a következő tároló órajel bemenetére csatlakozik
- A tárolók kimeneti jelének frekvenciája fele a bemeneti órajel frekvenciájának
- Előny:
  - Nem kellenek kiegészítő kapuáramkörök
- Hátrány:
  - A késleltetések miatt a tárolók nem egyszerre billennek
  - Az órajel változásakor rövid időre határozatlan kimeneti jel

Aszinkron bináris felfelé számláló

- Pl. 0-tól 7-ig (000b – 111b) egyesével számlál minden órajel ciklusban (3-bites)
- A hálózat $2^3 = 8$ állapotot vehet fel, nincs tiltott állapot

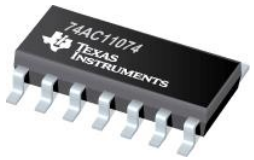
Q_2	Q_1	Q_0	
0	0	0	(0)
0	0	1	(1)
0	1	0	(2)
0	1	1	(3)
1	0	0	(4)
1	0	1	(5)
1	1	0	(6)
1	1	1	(7)





Diszkrét áramkörök

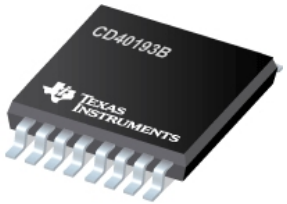
- D flip-flop
  - 74AC11074



- J-K tároló
  - CD54AC112

- Léptető regiszter
  - CD4015

- Számláló
  - CD40193

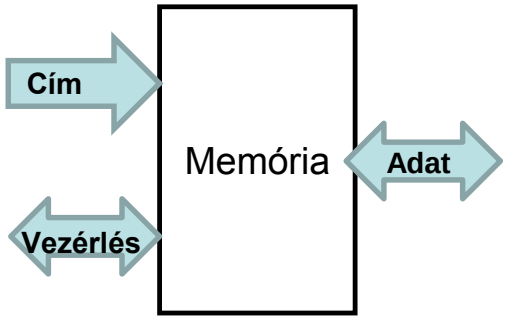


Texas Instruments

Memóriák

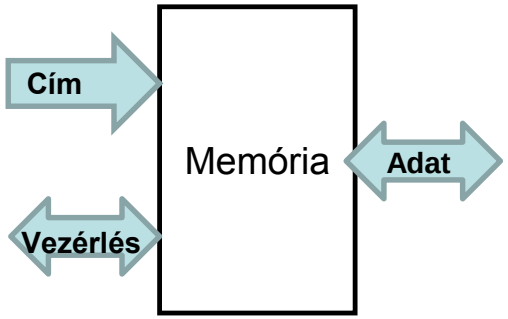
Memóriák

- Memóriák (tárolók)
  - Nagyobb mennyiségű információ átmeneti vagy tartós tárolására szolgáló egységek



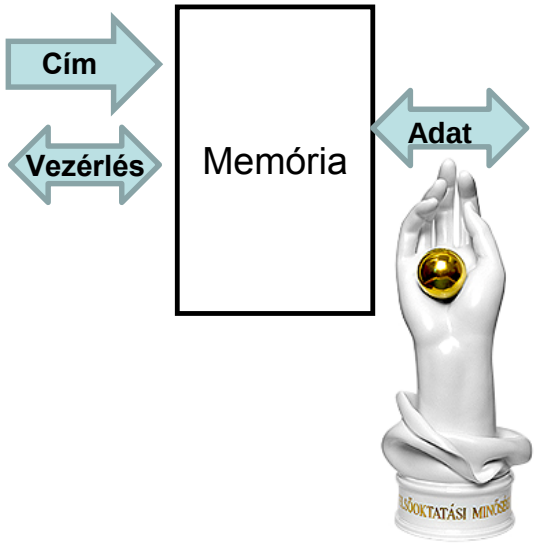
Memóriák

- Memóriák (tárolók)
  - Nagyobb mennyiségű információ átmeneti vagy tartós tárolására szolgáló egységek
    - **Regiszterek**
      - Néhány bit, vagy bitcsoport (4, 8, 16, 32 …stb.) tárolására



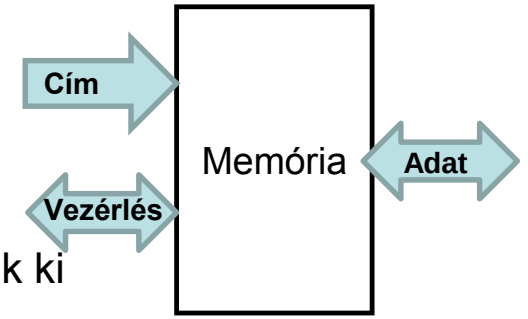
Memóriák

- Memóriák (tárolók)
  - Nagyobb mennyiségű információ átmeneti vagy tartós tárolására szolgáló egységek
    - Regiszterek
      - Néhány bit, vagy bitcsoport (4, 8, 16, 32 …stb.) tárolására
    - Több regiszterből nagyobb tárolókapacitású tárolók építhetők
      - Kiegészítő és vezérlő egységek szükségesek
        - A tárolt információ célszerű kezeléséhez
        - Más áramkörökkel való együttműködés, illeszthetőség
        - foglaltság, készenlét stb…



Memóriák

- Memóriák (tárolók)
  - Nagyobb mennyiségű információ átmeneti vagy tartós tárolására szolgáló egységek
    - Regiszterek
      - Néhány bit, vagy bitcsoport (4, 8, 16, 32 …stb.) tárolására
    - Több regiszterből nagyobb tárolókapacitású tárolók építhetők
      - Kiegészítő és vezérlő egységek szükségesek
        - A tárolt információ célszerű kezeléséhez
        - Más áramkörökkel való együttműködés, illeszthetőség
        - foglaltság, készenlét stb…



- **Memóriák működése**
  - A „**Cím" bemenetre** érkező információval jelöljük ki
    - » az „Adat" csatlakozóra érkező, tárolandó
    - » az „Adat" csatlakozón távozó, kiolvasandó
    - információ memórián belüli helyét (memória rekeszt)
  - A „**Vezérlés"** csatlakozásokon keresztül
    - adhatunk utasításokat : beírás, kiolvasás stb…
    - vagy nyerhetünk információkat a memória működésére vonatkozólag : foglaltság, készenlét stb…

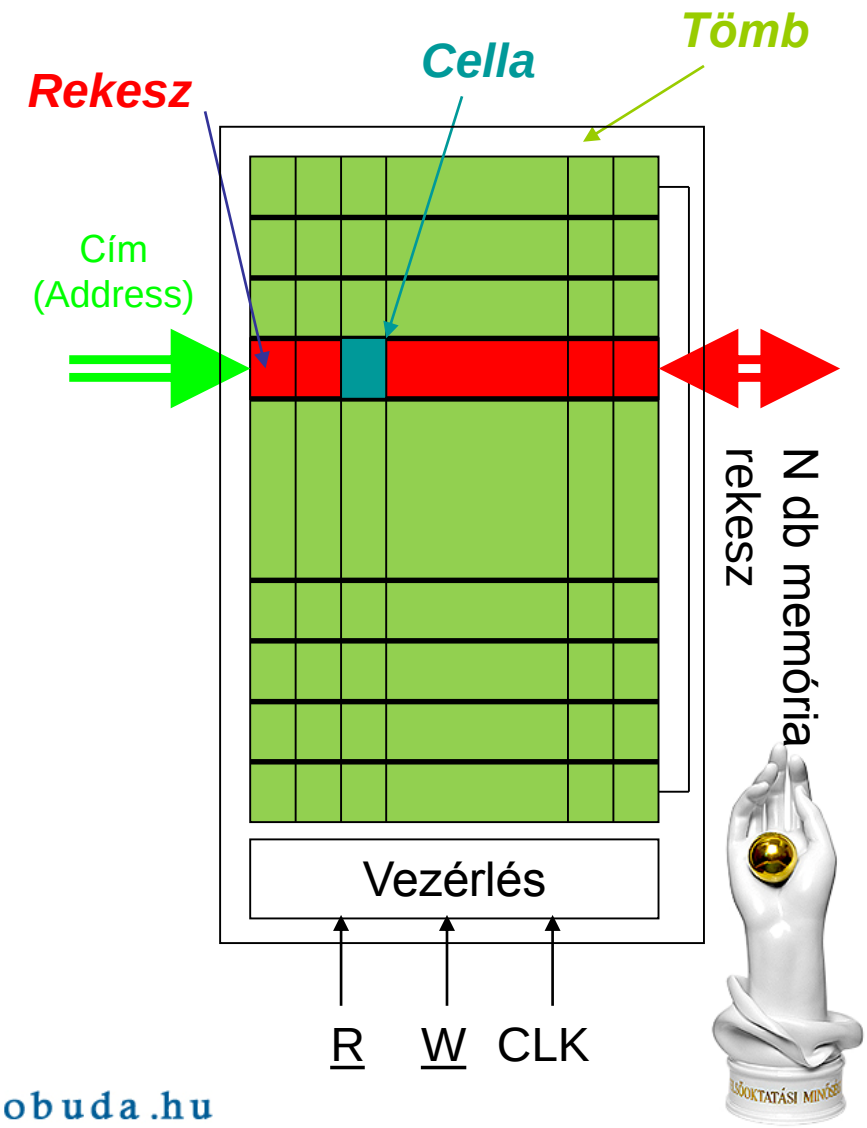
Memóriák

- Működés
  - Általános memória felépítése

ÓBUDAI EGYETEM

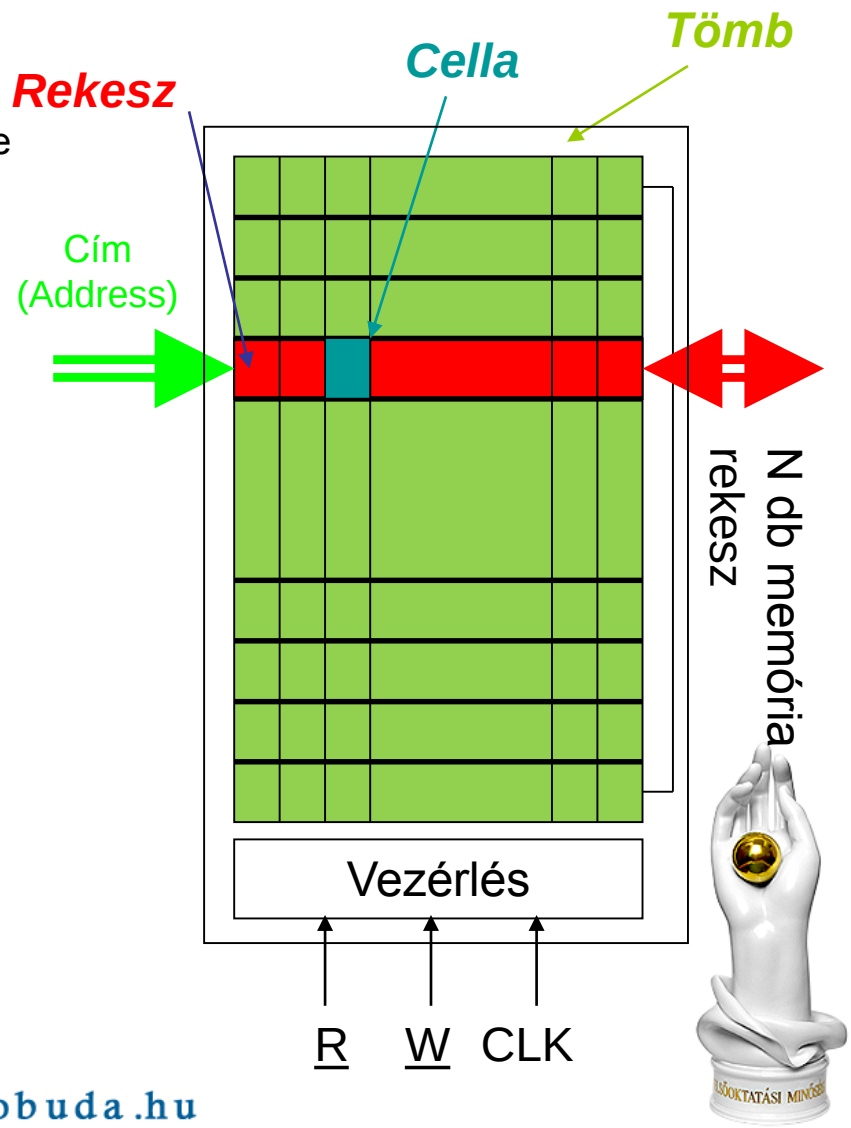
Memóriák

- Működés
  - Általános memória felépítése



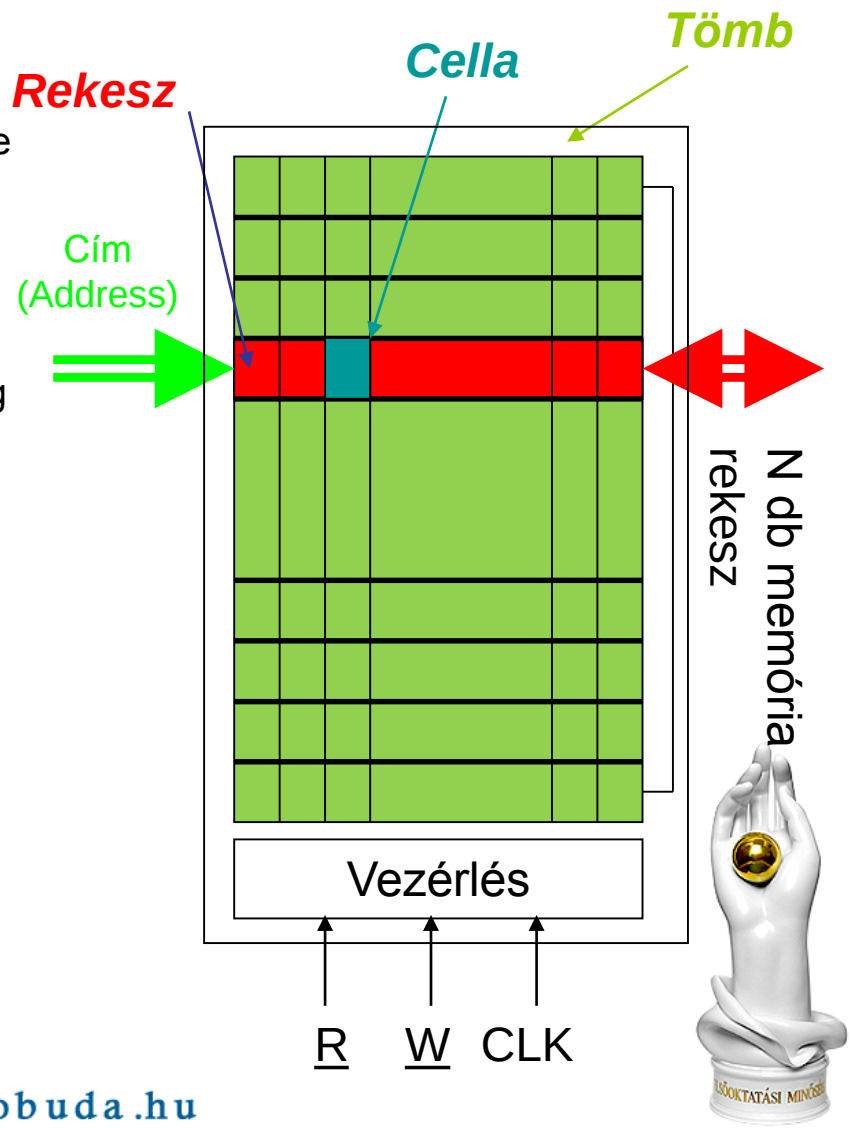
Memóriák

- Működés
  - Általános memória felépítése
    - 1 db cella 1 bit tárolására képes
    - 1 rekesz (sor) K db cellából épül fel. 1 sor mérete K bit. Ettől függően léteznek:
      - Byte szervezésű (K = 8)
      - Szó szervezésű (K = 16)
      - Duplaszó szervezésű (K = 32)



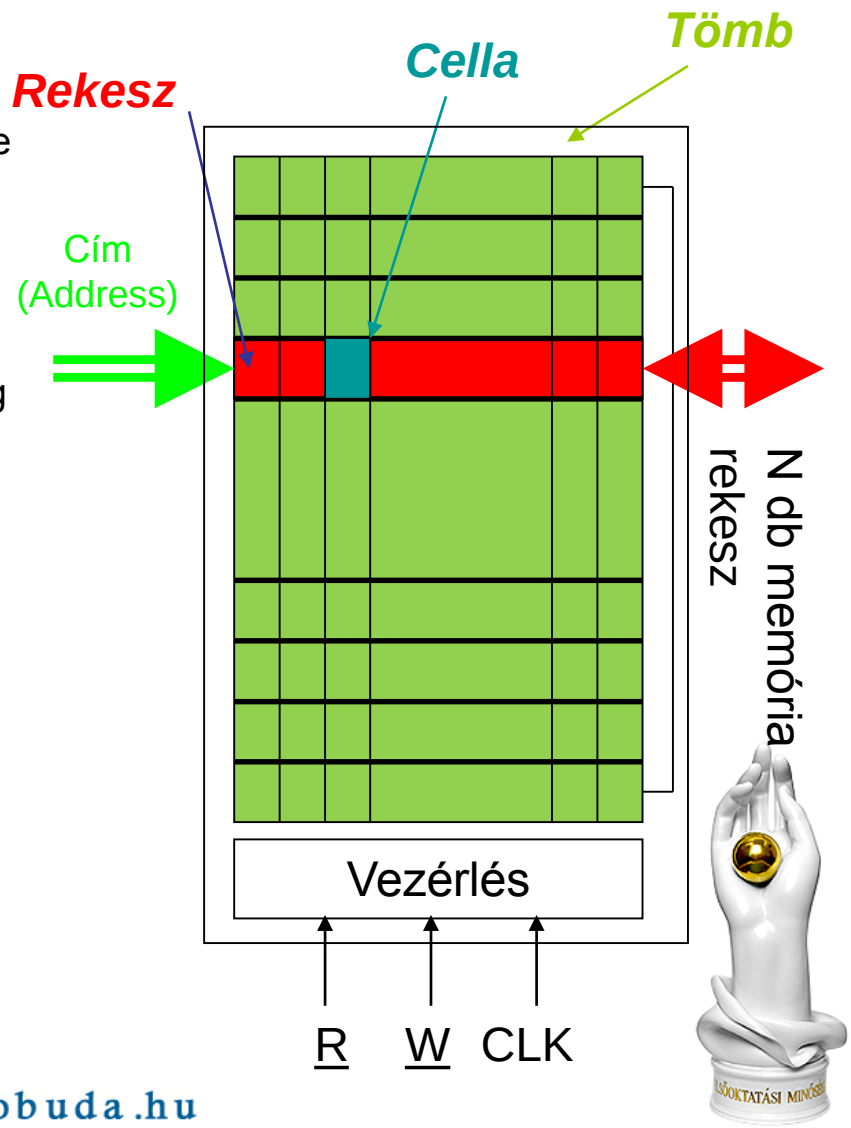
Memóriák

- Működés
  - Általános memória felépítése
    - 1 db cella 1 bit tárolására képes
    - 1 rekesz (sor) K db cellából épül fel. 1 sor mérete K bit. Ettől függően léteznek:
      - Byte szervezésű (K = 8)
      - Szó szervezésű (K = 16)
      - Duplaszó szervezésű (K = 32)
    - A párhuzamos hozzáférésű memóriáknál az adatbusz hozzávezetéseinek számát K adja meg



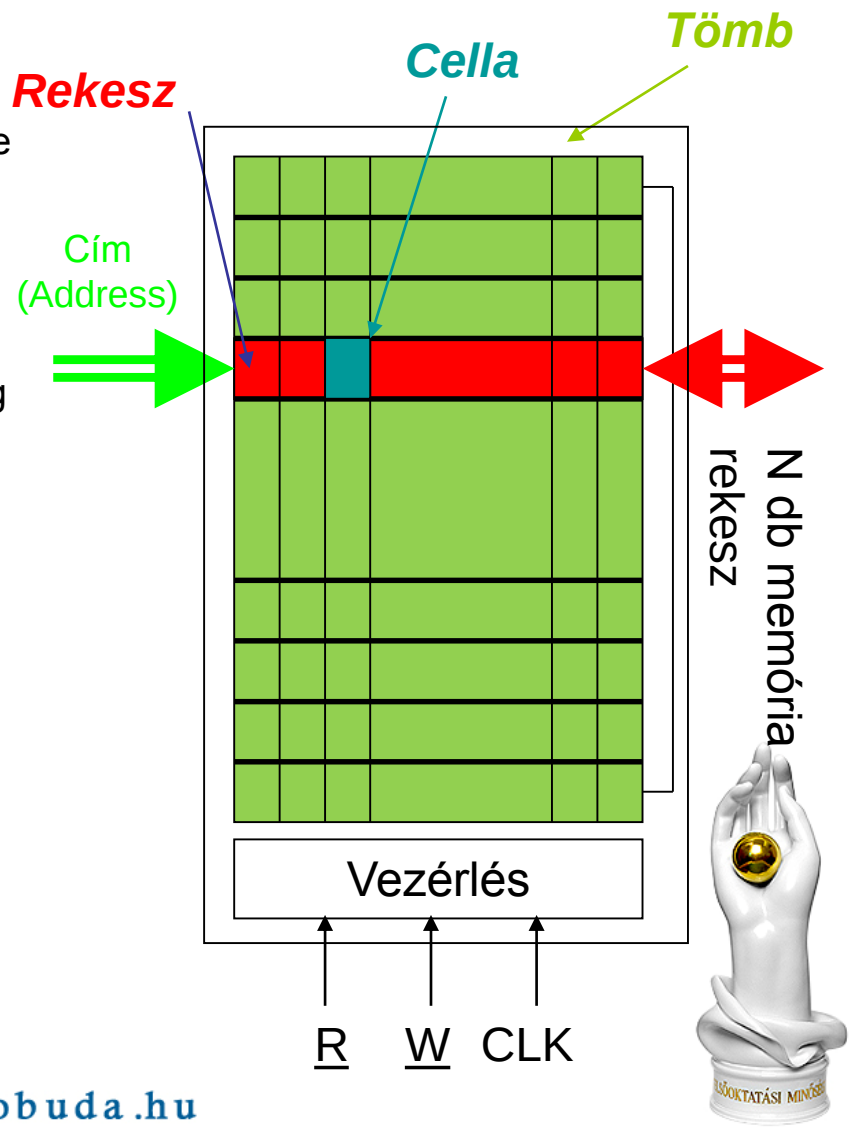
Memóriák

- Működés
  - Általános memória felépítése
    - 1 db cella 1 bit tárolására képes
    - 1 rekesz (sor) K db cellából épül fel. 1 sor mérete K bit. Ettől függően léteznek:
      - Byte szervezésű (K = 8)
      - Szó szervezésű (K = 16)
      - Duplaszó szervezésű (K = 32)
    - A párhuzamos hozzáférésű memóriáknál az adatbusz hozzávezetéseinek számát K adja meg



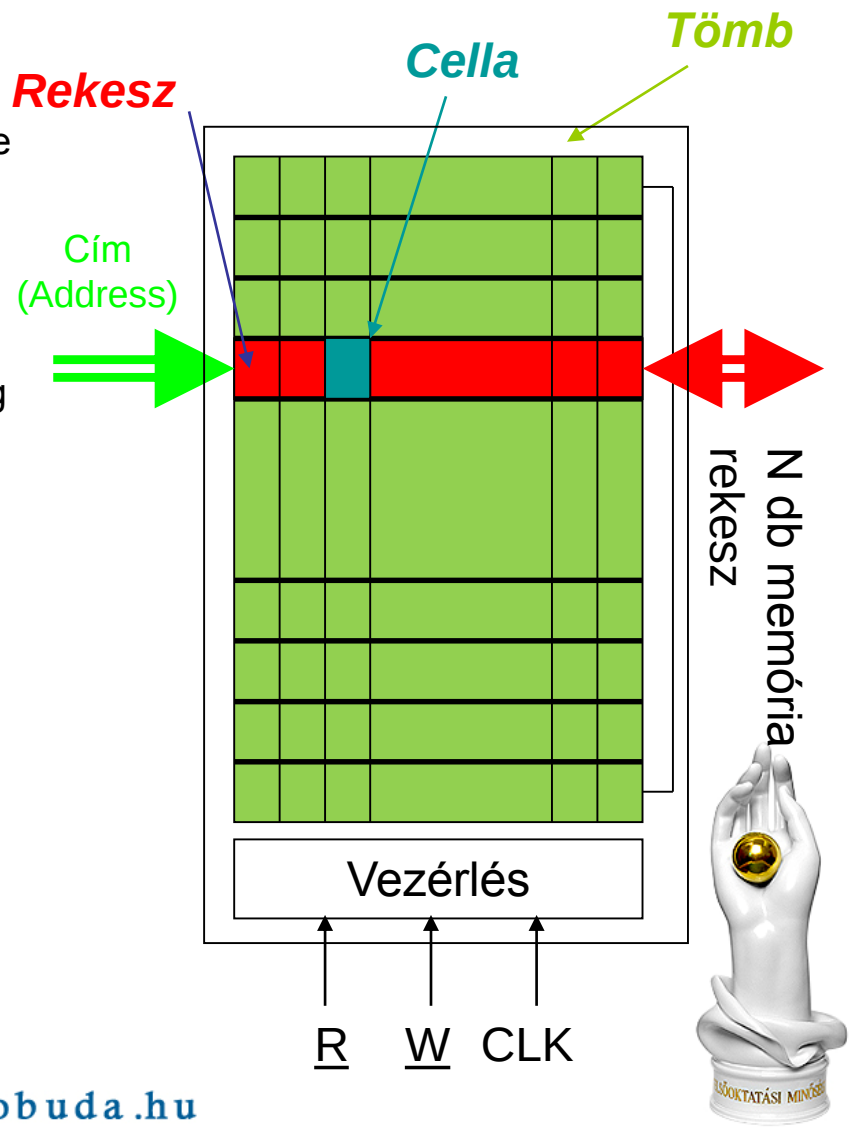
Memóriák

- Működés
  - Általános memória felépítése
    - 1 db cella 1 bit tárolására képes
    - 1 rekesz (sor) K db cellából épül fel. 1 sor mérete K bit. Ettől függően léteznek:
      - Byte szervezésű (K = 8)
      - Szó szervezésű (K = 16)
      - Duplaszó szervezésű (K = 32)
    - A párhuzamos hozzáférésű memóriáknál az adatbusz hozzávezetéseinek számát K adja meg
    - A memória tömb N = 2^L db rekeszből épül fel
      - A párhuzamos címzésű memóriáknál a címbusz hozzávezetéseinek száma L



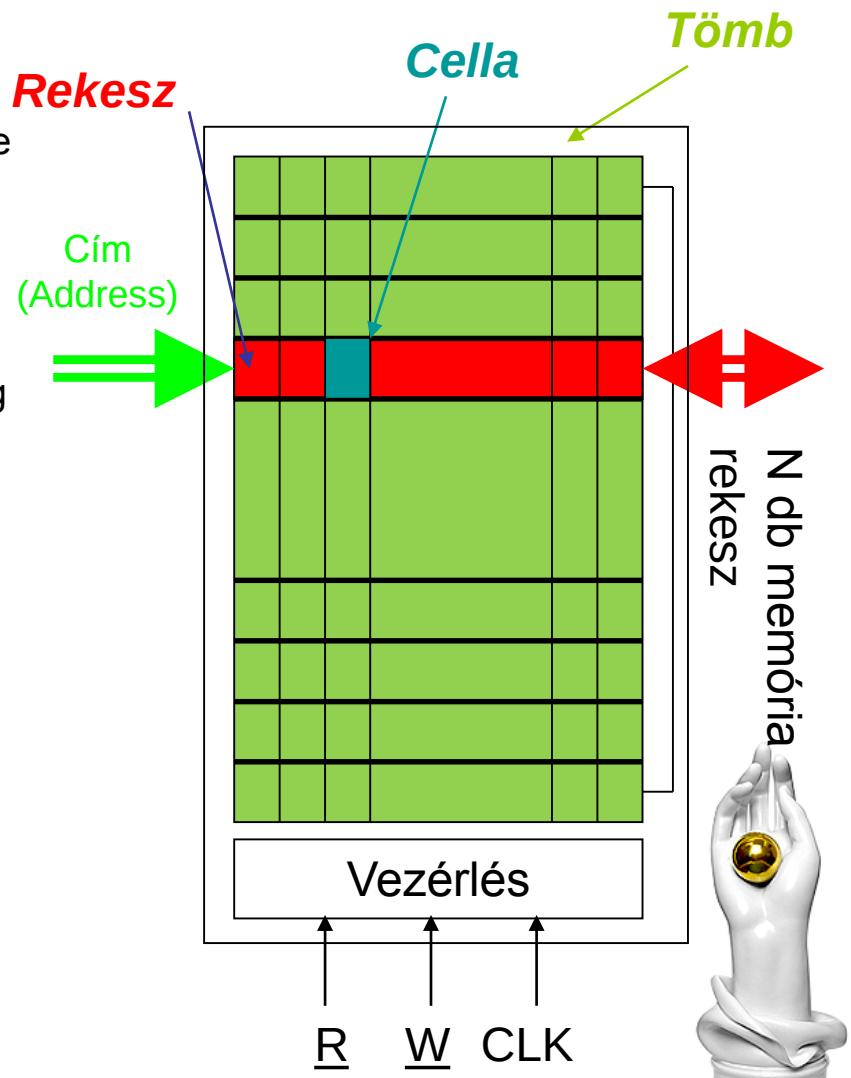
Memóriák

- Működés
  - Általános memória felépítése
    - 1 db cella 1 bit tárolására képes
    - 1 rekesz (sor) K db cellából épül fel. 1 sor mérete K bit. Ettől függően léteznek:
      - Byte szervezésű (K = 8)
      - Szó szervezésű (K = 16)
      - Duplaszó szervezésű (K = 32)
    - A párhuzamos hozzáférésű memóriáknál az adatbusz hozzávezetéseinek számát K adja meg
    - A memória tömb N = 2^L db rekeszből épül fel
      - A párhuzamos címzésű memóriáknál a címbusz hozzávezetéseinek száma L
    - Memória tárolókapacitása
      - tárolt bitek száma = 1 rekesz mérete * rekeszek darabszáma = K * N bit



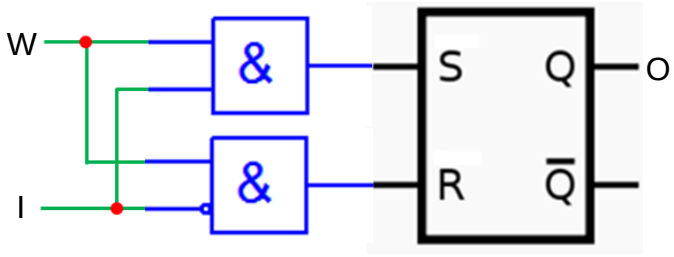
Memóriák

- Működés
  - Általános memória felépítése
    - 1 db cella 1 bit tárolására képes
    - 1 rekesz (sor) K db cellából épül fel. 1 sor mérete K bit. Ettől függően léteznek:
      - Byte szervezésű (K = 8)
      - Szó szervezésű (K = 16)
      - Duplaszó szervezésű (K = 32)
    - A párhuzamos hozzáférésű memóriáknál az adatbusz hozzávezetéseinek számát K adja meg
    - **A memória tömb N = 2^L db rekeszből épül fel**
      - A párhuzamos címzésű memóriáknál a címbusz hozzávezetéseinek száma L
    - **Memória tárolókapacitása**
      - tárolt bitek száma = 1 rekesz mérete * rekeszek darabszáma = K * N bit
    - **Vezérléstől függően 3 féle állapot**:
      - Írás: $\underline{R} = 1, \underline{W} = 0$
      - Olvasás : $\underline{R} = 0, \underline{W} = 1$
      - Üresjárat / tárolás: $\underline{R} = 1, \underline{W} = 1$



Memóriák

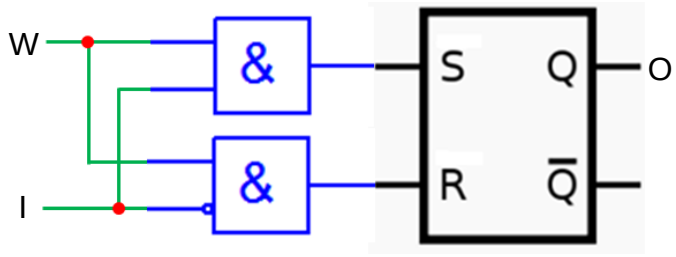
- Cella
  - RS tároló + logika



R	S	Q^{n+1}
0	0	Q^n
0	1	1
1	0	0
1	1	X

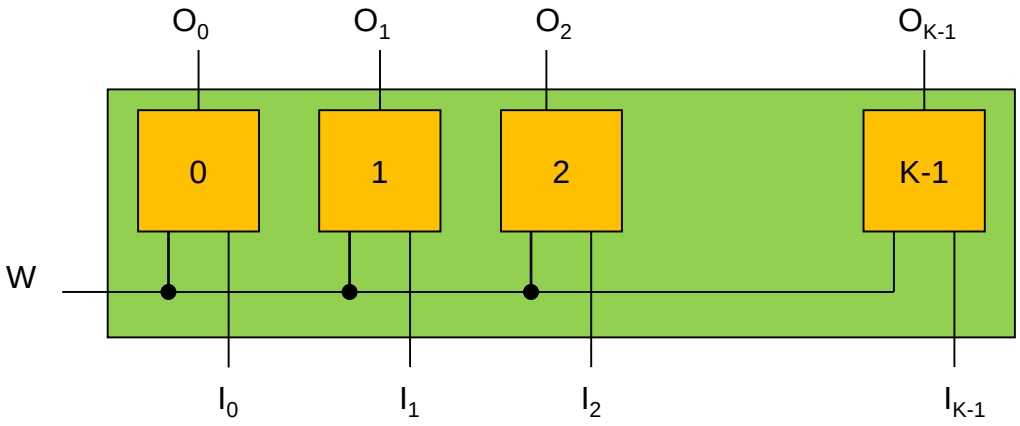
Memóriák

- REKESZ
  - Általános memória felépítése
    - Cella
      - RS tároló + logika



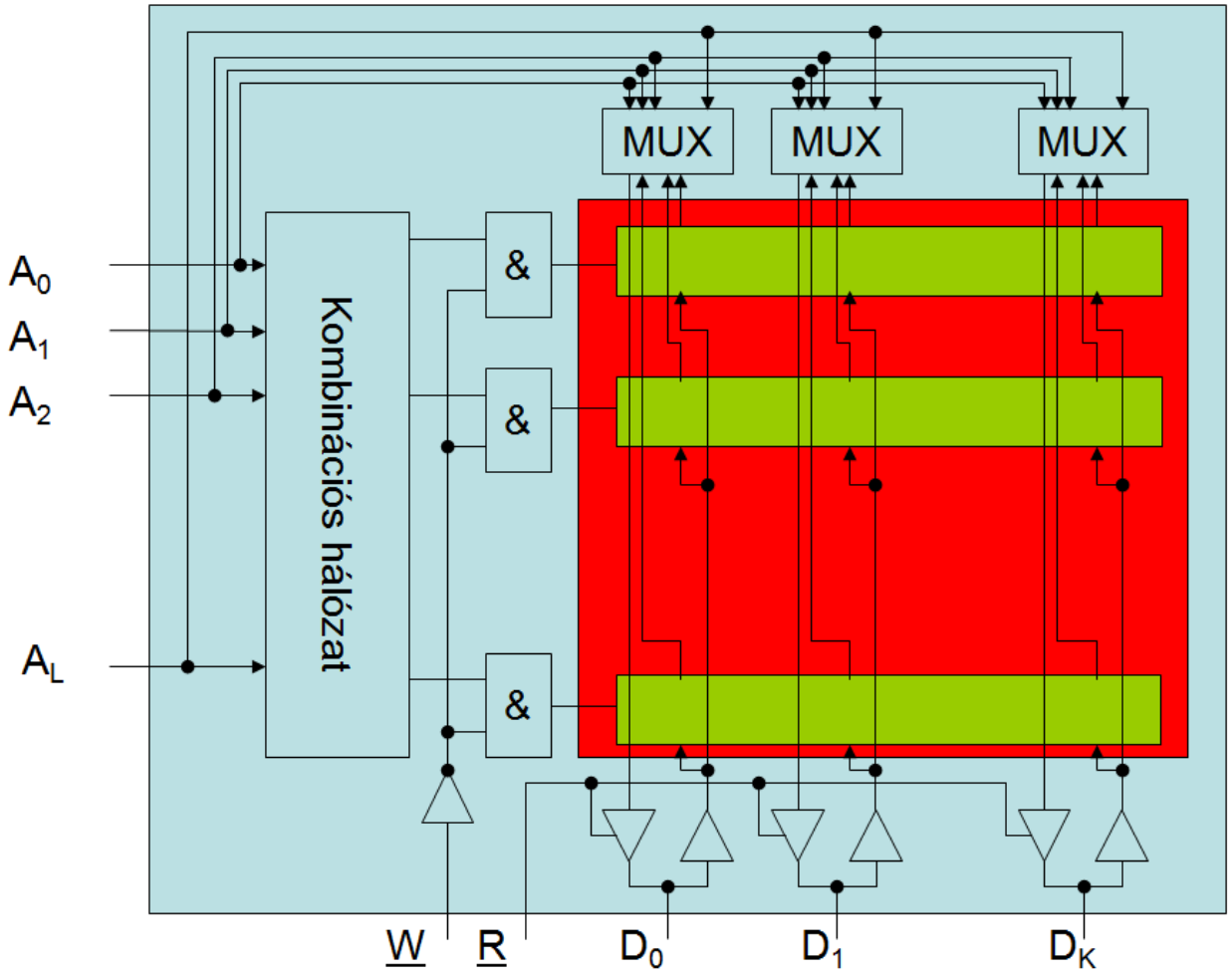
R	S	Q^{n+1}
0	0	Q^n
0	1	1
1	0	0
1	1	X

    - Rekesz
      - Regiszterhez hasonlóan, tároló cellák párhuzamosan kapcsolva



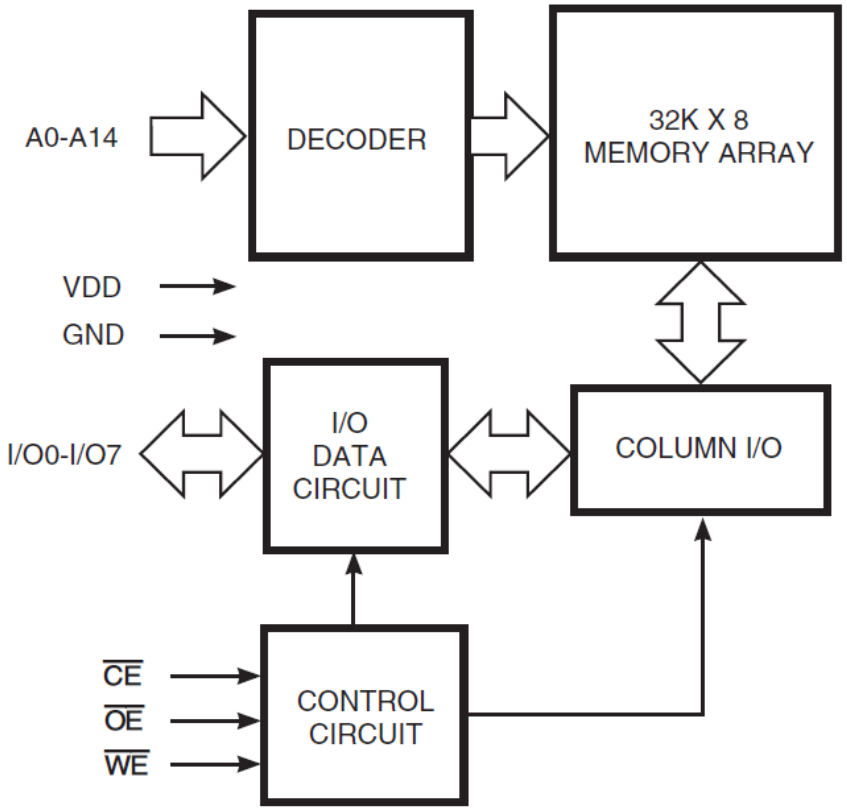
Memóriák

- Rekeszek memória tömbbé szervezése



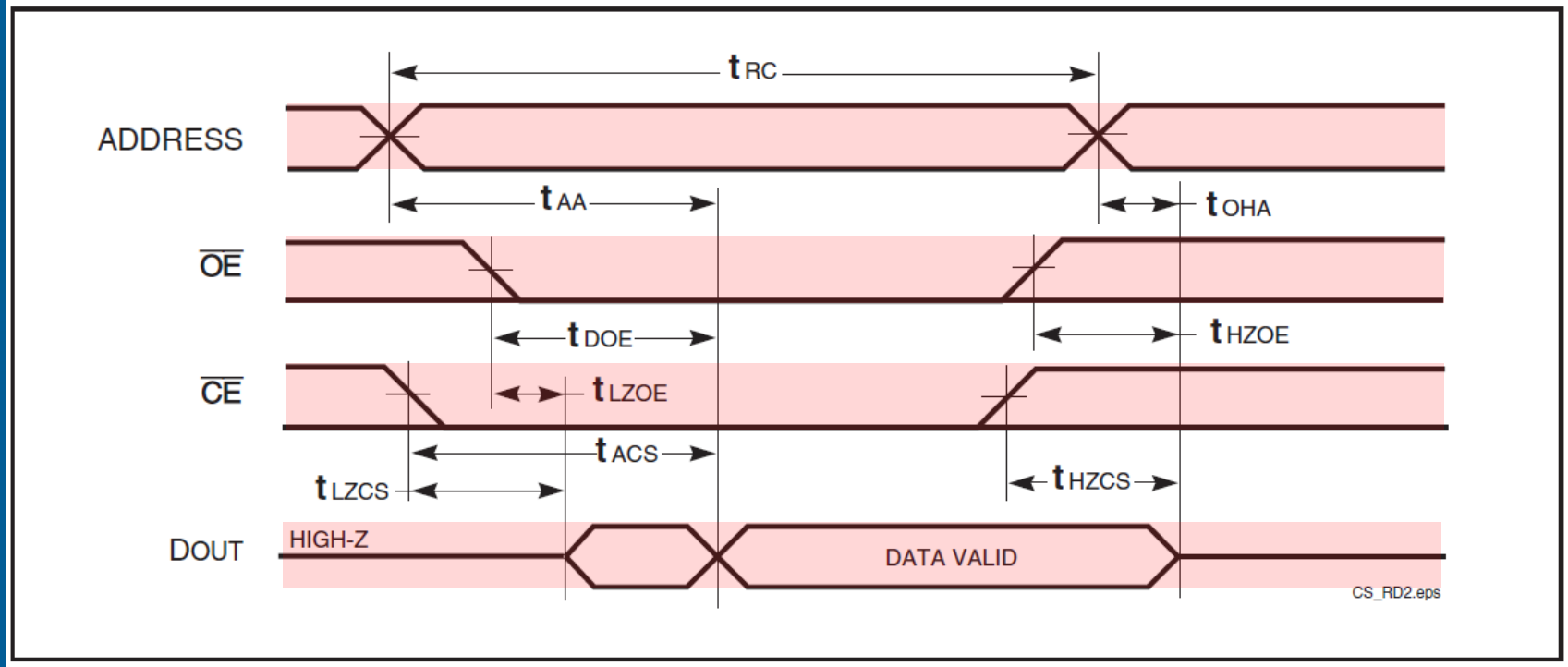
Memóriák

- Rekeszek memória tömbbé szervezése
  - IS65C256AL



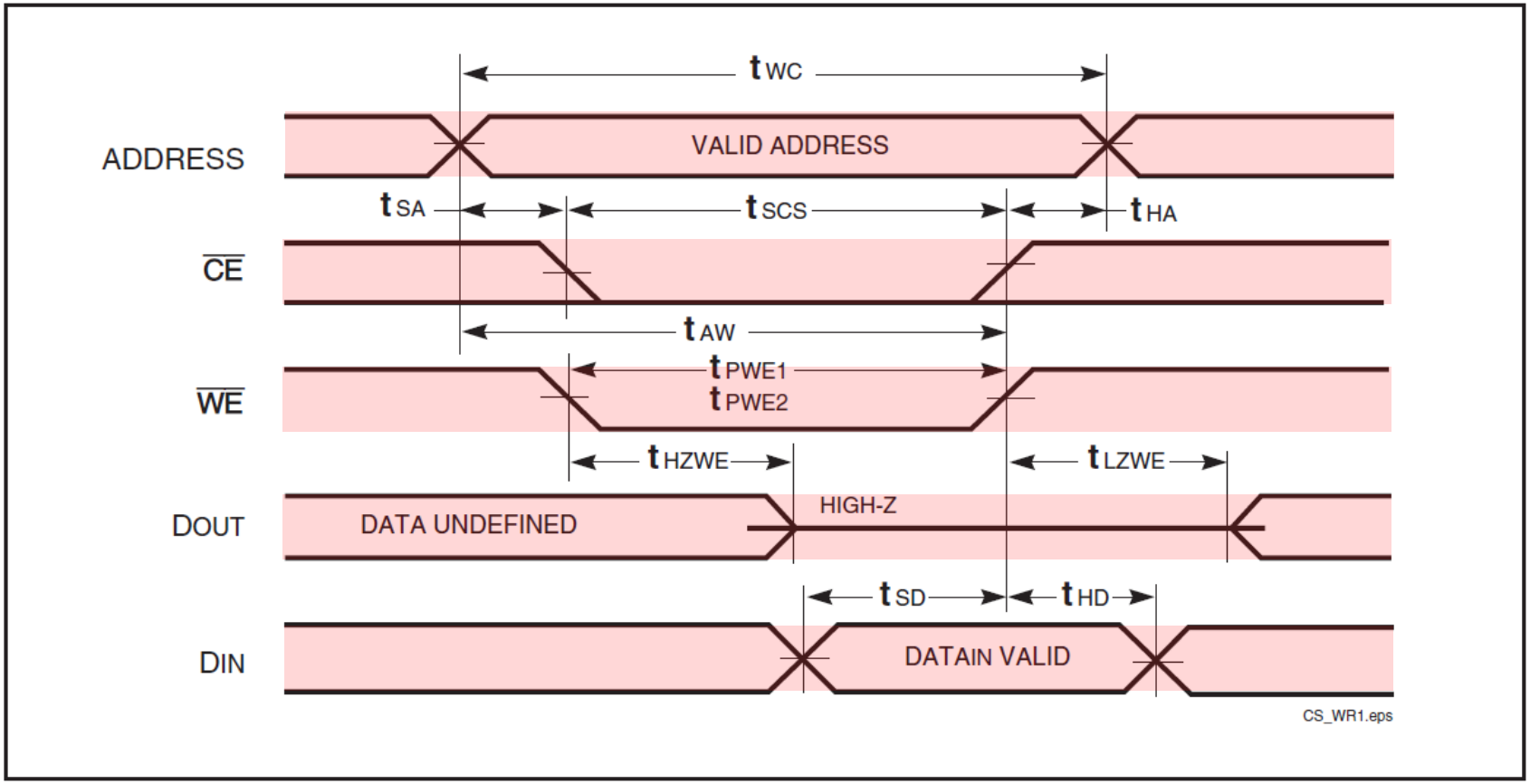
Memóriák

- Működés
  - Olvasás, párhuzamos független adat és címvezetékekkel
    - CE: Chip Enable
    - OE : Output Enable



Memóriák

- Működés
  - Írás, párhuzamos független adat és címvezetékekkel
    - CE: Chip Enable
    - WE : Write Enable



Memóriák csoportosítása

- A megcímzett rekesz hozzáférési módja szerint
- Az információ beírhatósága szempontjából
- A tárolás időbeli módja

Memóriák csoportosítása

- A megcímzett rekesz hozzáférési módja szerint
  - **Tetszőleges (véletlen) hozzáférésű memória (RAM: Random Access Memory)**
    - Bármely adat a címtől függetlenül azonos idő alatt elérhető
  - **Soros hozzáférésű memória (SAM: Serial Access Memory)**
    - Az adat címétől függő elérési idő
    - Pl. Mágnesszalagos tárolás
  - **Asszociatív memória (CAN : Content Addressable Memory)**
    - Megadja, hogy az adott információ a memória mely címén található

Memóriák csoportosítása

- Az információ beírhatósága szempontjából
  - Csak olvasható memória

    (ROM: Read Only Memory)
  - Módosítható memória

    (RWM: Read Write Memory)

    RAM

Memóriák csoportosítása

- A tárolás időbeli módja
  - Statikus (pl.: SRAM)
    - Tápfeszültség esetén az információt korlátlan ideig megőrzi
  - Dinamikus (pl.: DRAM)
    - A memória tartalma időnként frissítésre szorul (ez hátrány)
    - (De) nagy tároló kapacitás érhető el

Memóriák

- Pl.:
  - 2^{16} bit tárolása (64 Kibit), 2^{16} cella
  - 8 bit (byte) szervezés: 8 KiByte, 2^{13} rekesz
  - Párhuzamos szervezéssel
    - 8 adatvezeték
    - 13 cím vezeték
  - Soros szervezéssel csökkenteni lehet a kivezetések számát
    - Egy 32 bites 2GiByte-os memóriának
      - 32 adatvezetés
      - 29 címvezeték
    - Párhuzamos memória + soros-párhuzamos átalakítás

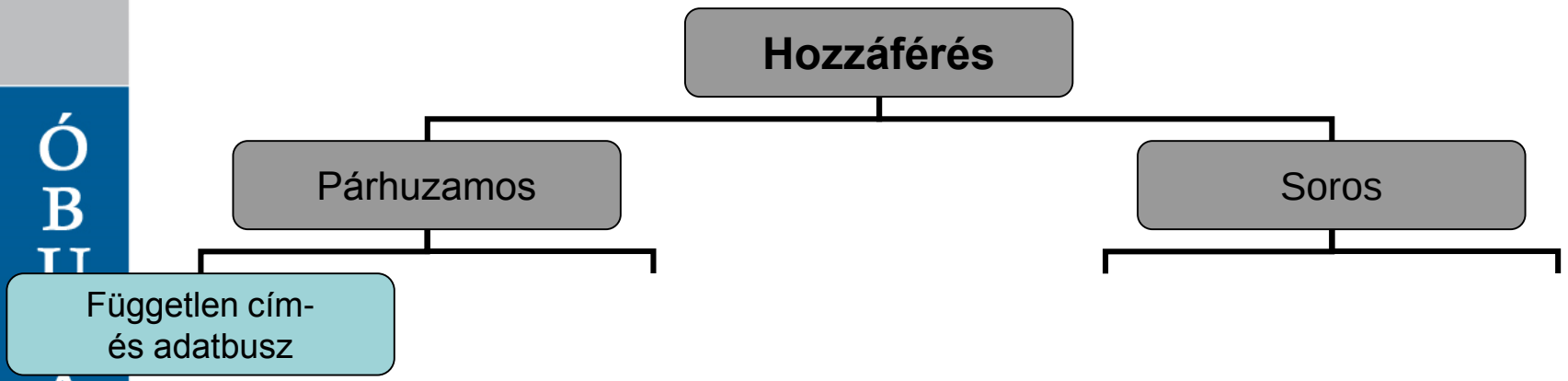
Memóriák

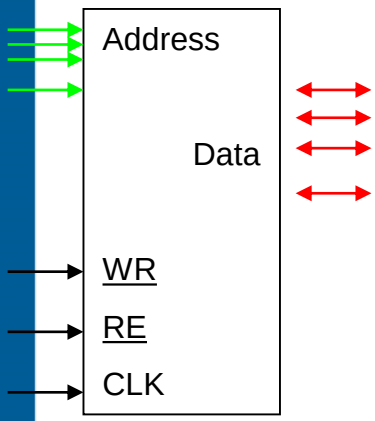
512M x 32 (K =29, L=32) tömb esetén ***M****: chip lábainak száma* ***Q****: 1 művelethez szükséges órajel periódusok száma*

Memóriák

512M x 32 (K =29, L=32) tömb esetén ***M****: chip lábainak száma* ***Q****: 1 művelethez szükséges órajel periódusok száma*

Memóriák



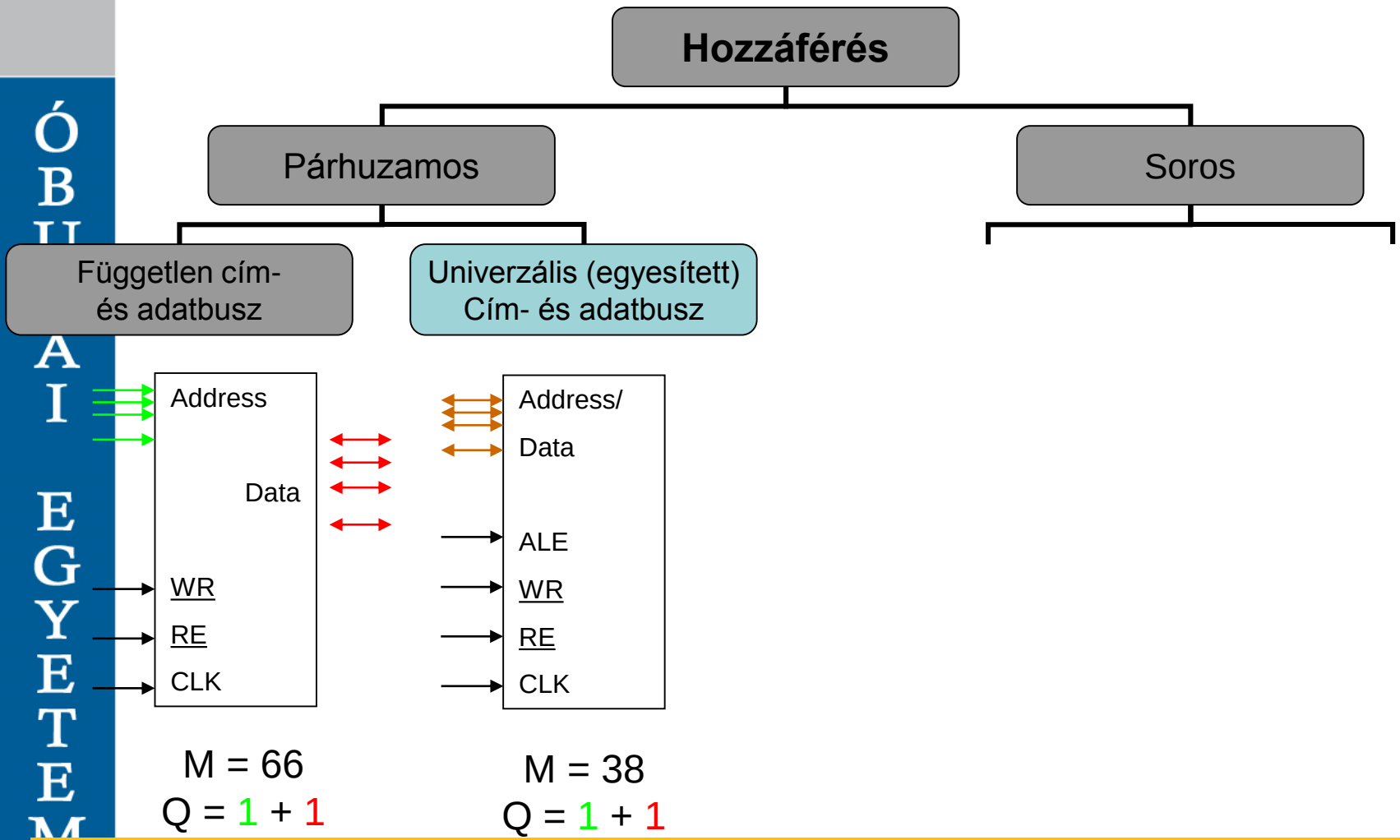


M = 66

Q = 1 + 1

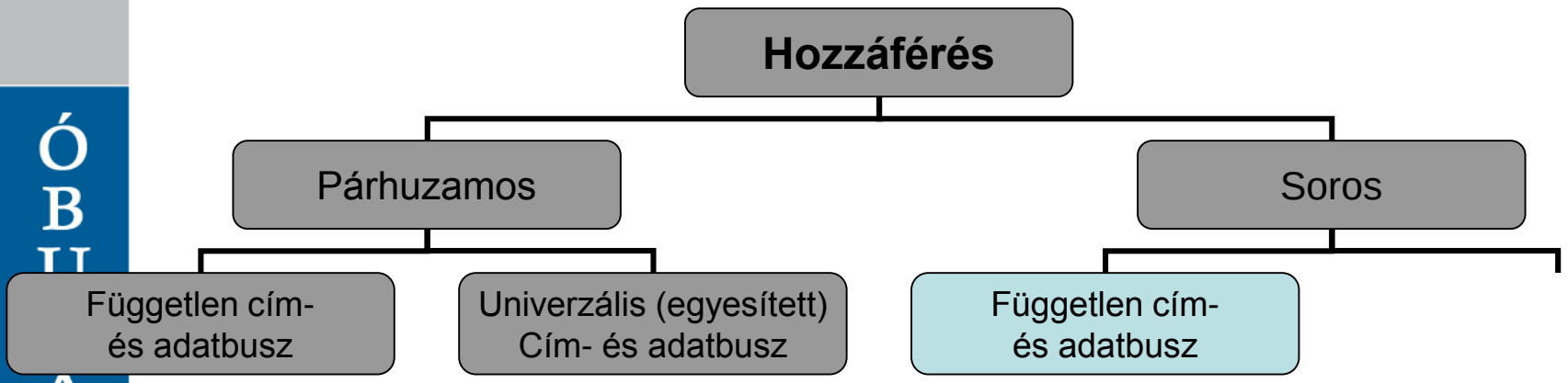
512M x 32 (K =29, L=32) tömb esetén ***M****: chip lábainak száma* ***Q****: 1 művelethez szükséges órajel periódusok száma*

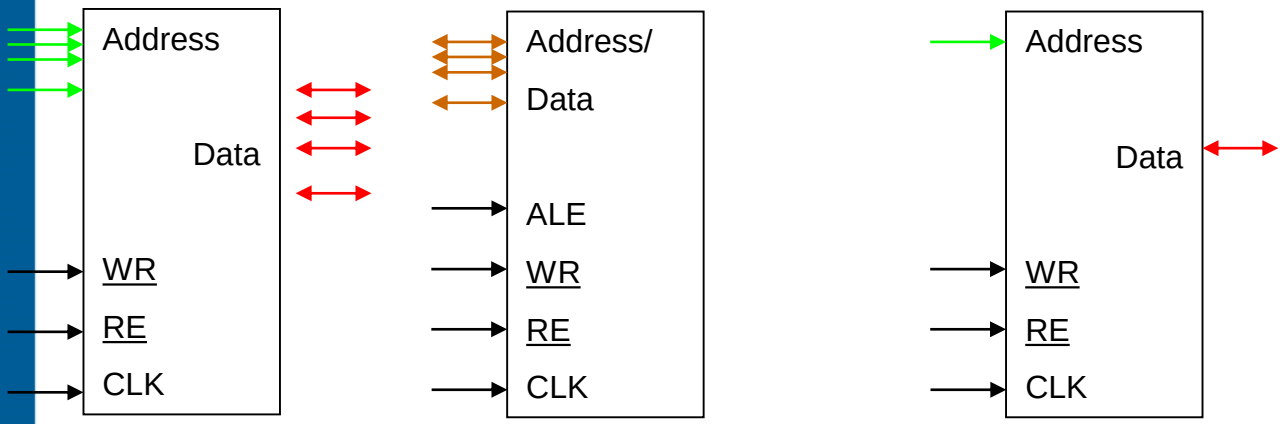
Memóriák



512M x 32 (K =29, L=32) tömb esetén ***M****: chip lábainak száma* ***Q****: 1 művelethez szükséges órajel periódusok száma*

Memóriák





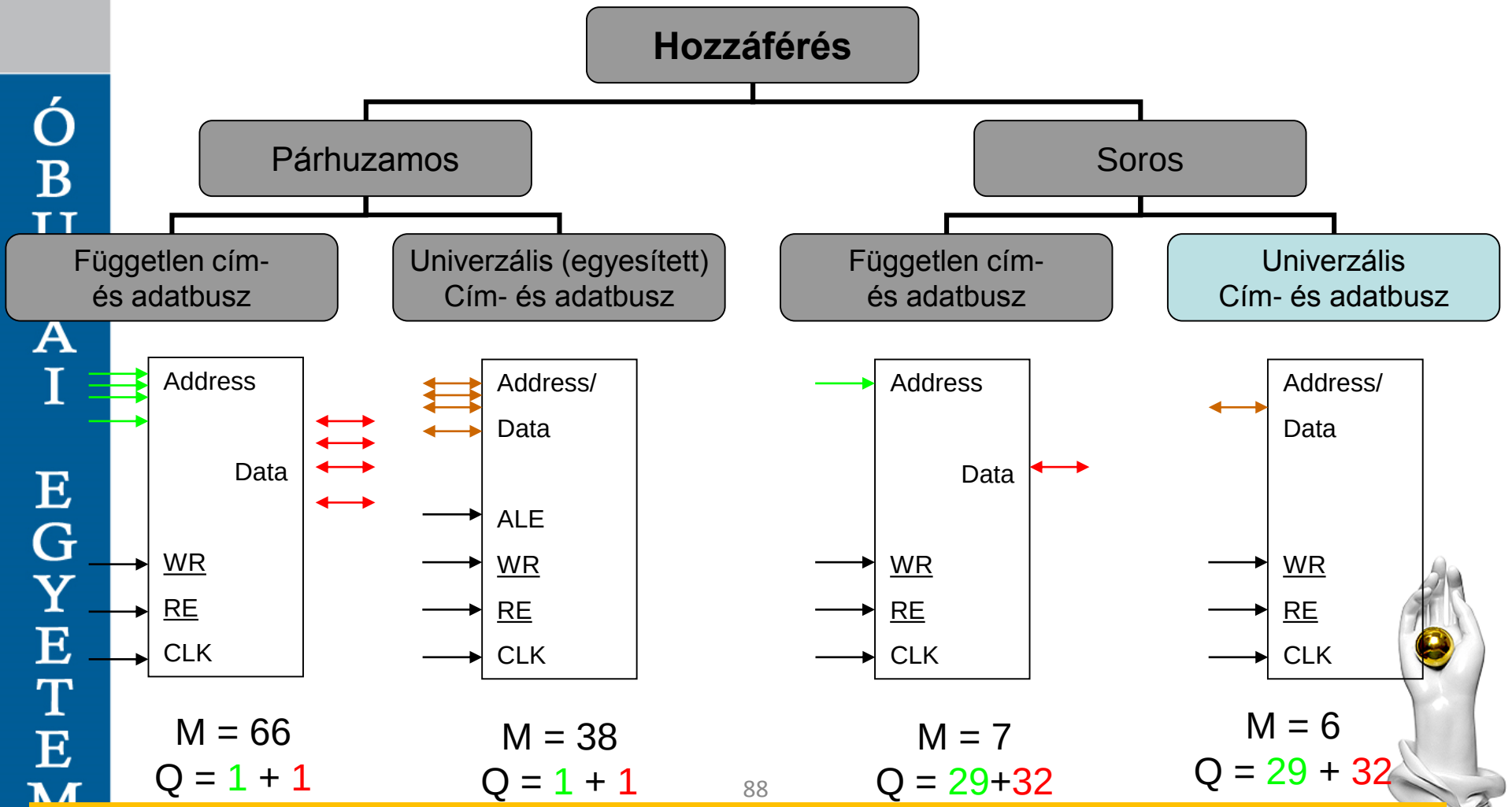
M = 66
Q = 1 + 1

M = 38
Q = 1 + 1

M = 7
Q = 29+32

512M x 32 (K =29, L=32) tömb esetén ***M****: chip lábainak száma* ***Q****: 1 művelethez szükséges órajel periódusok száma*

Memóriák



M = 66
Q = 1 + 1

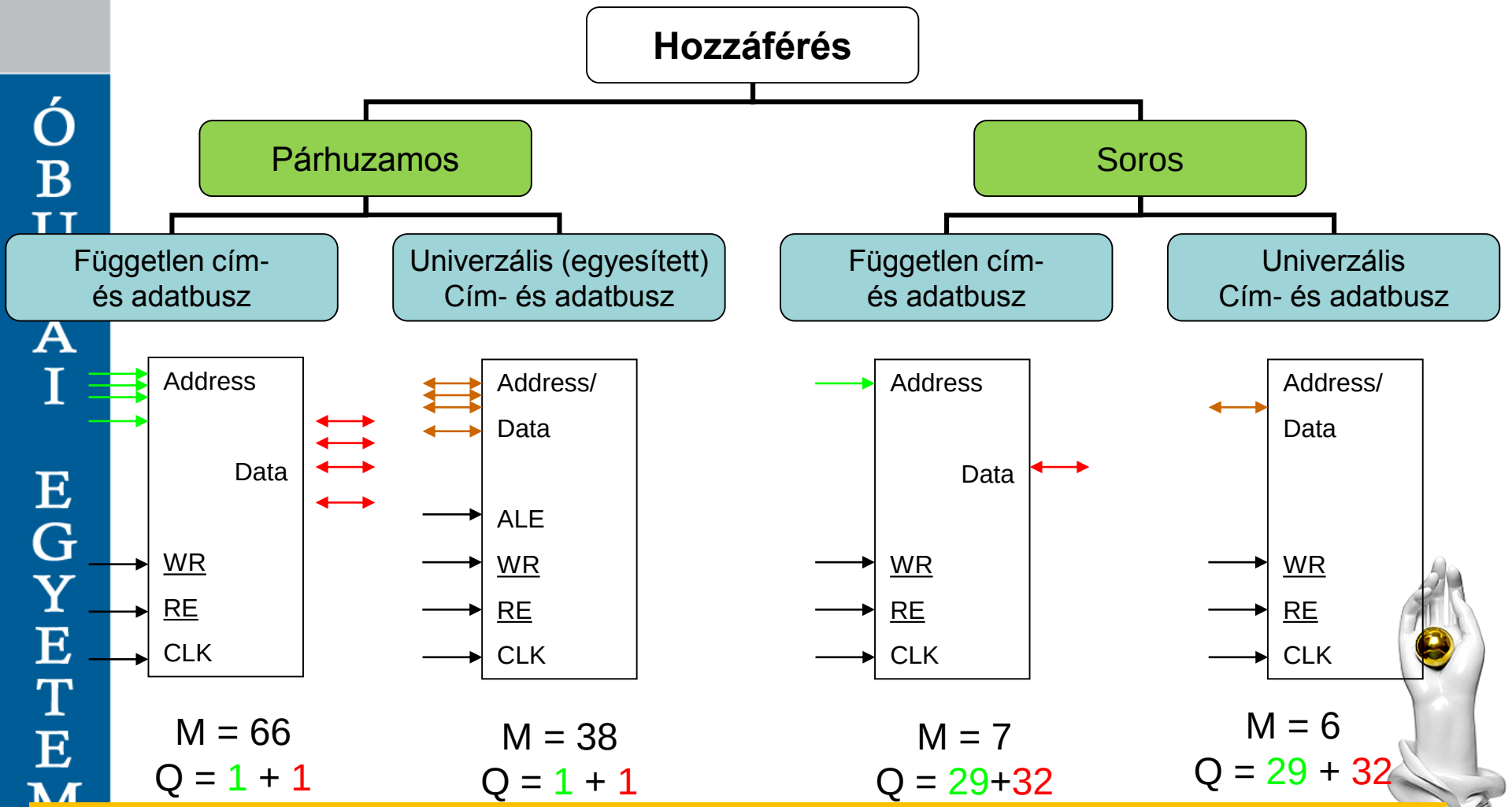
M = 38
Q = 1 + 1

M = 7
Q = 29+32

M = 6
Q = 29 + 32

512M x 32 (K =29, L=32) tömb esetén ***M****: chip lábainak száma* ***Q****: 1 művelethez szükséges órajel periódusok száma*

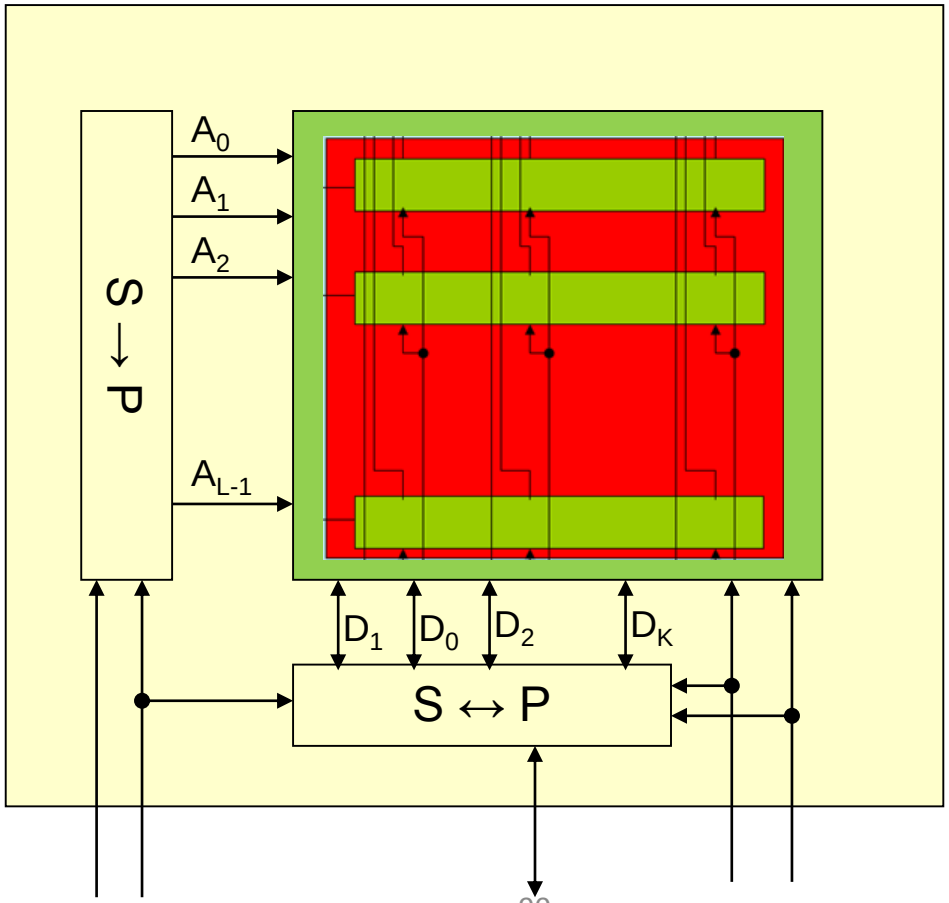
Memóriák



512M x 32 (K =29, L=32) tömb esetén ***M****: chip lábainak száma* ***Q****: 1 művelethez szükséges órajel periódusok száma*

Memóriák

- Soros hozzáférésű memória
  - Párhuzamos memóriából, soros-párhuzamos, párhuzamos-soros átalakítókkal

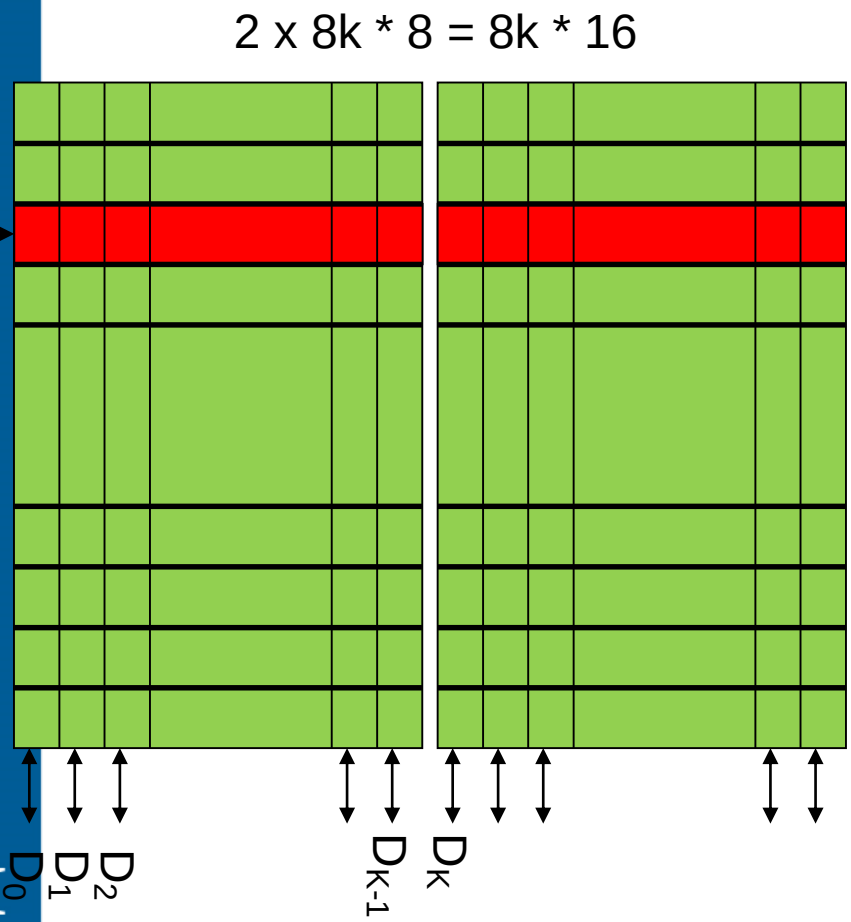


Memóriák bővítése

- Szóhossz bővítés
- Kapacitás bővítés

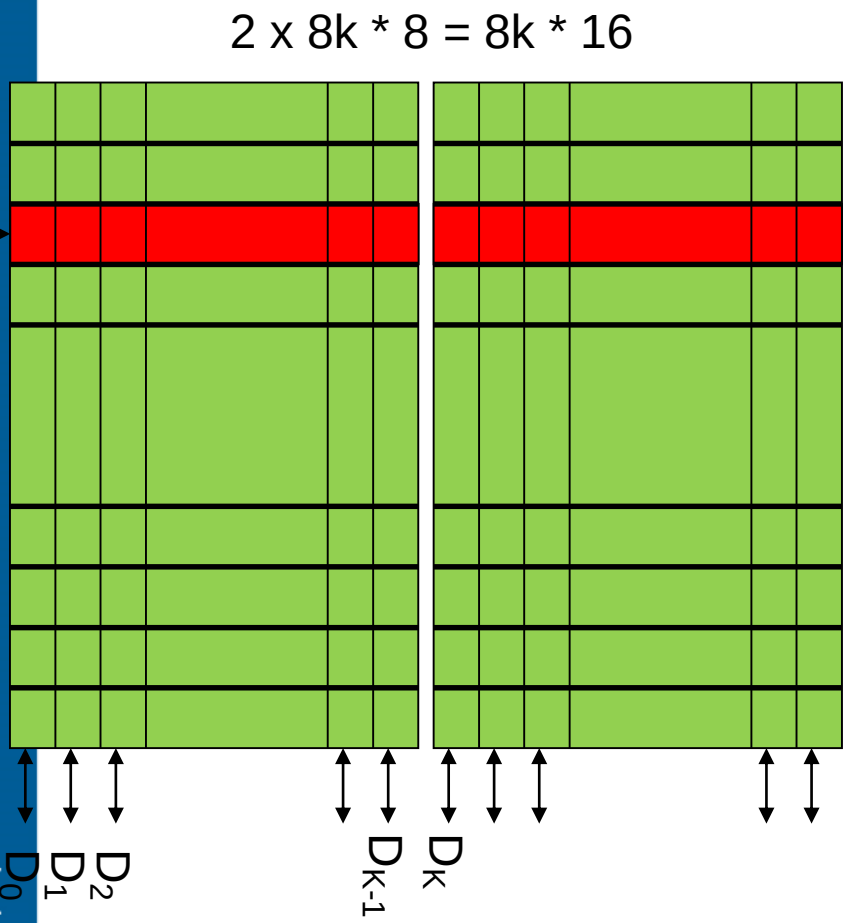
Memóriák bővítése

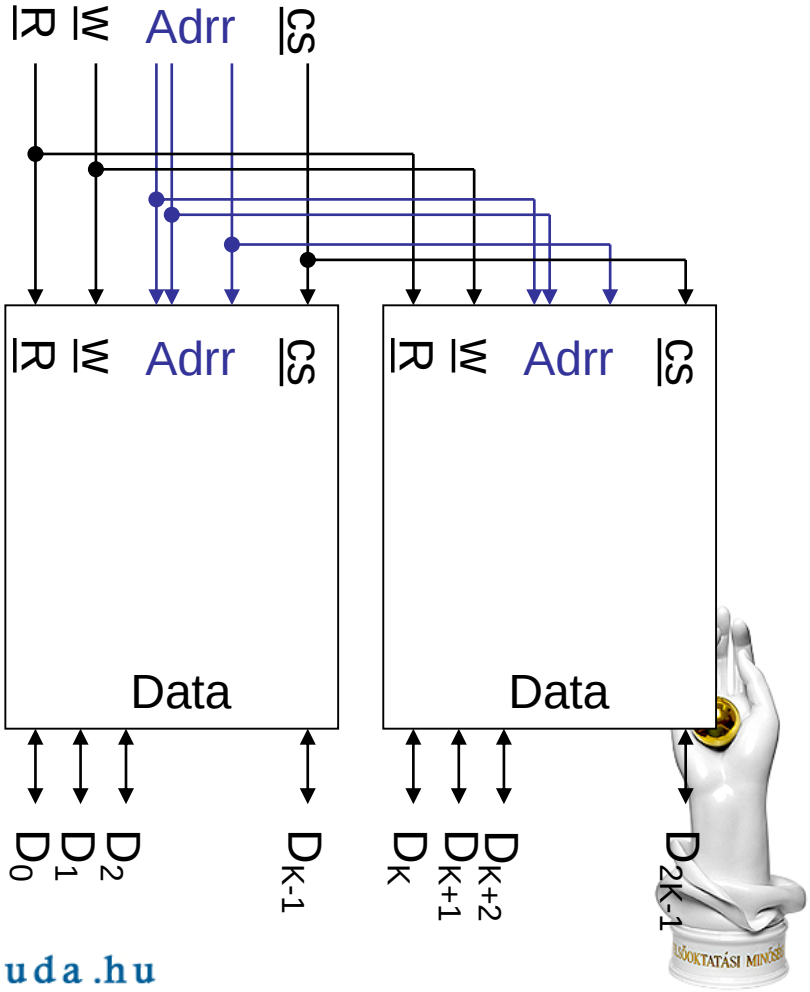
- Szóhossz bővítés



Memóriák bővítése

- Szóhossz bővítés

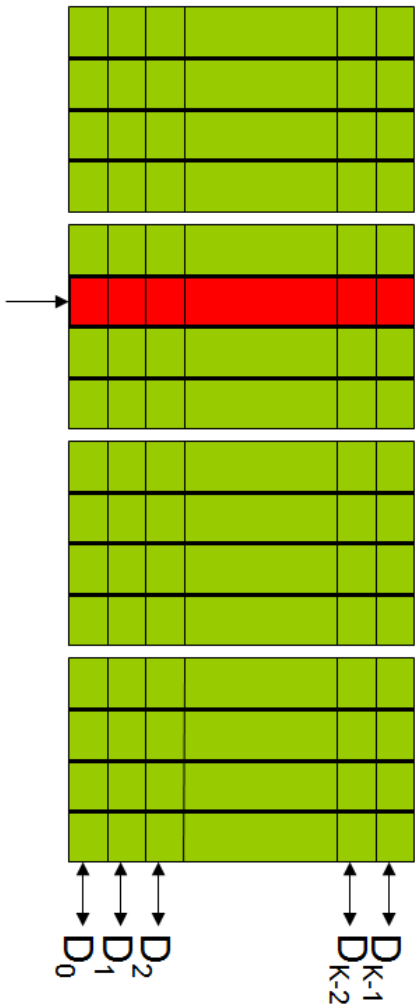




Memóriák

- Kapacitás bővítése

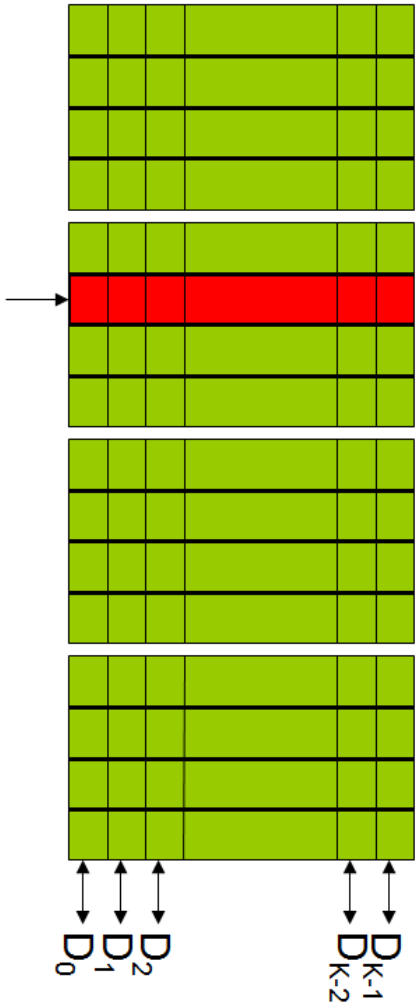
4 x 8k * 8 = 32k * 8

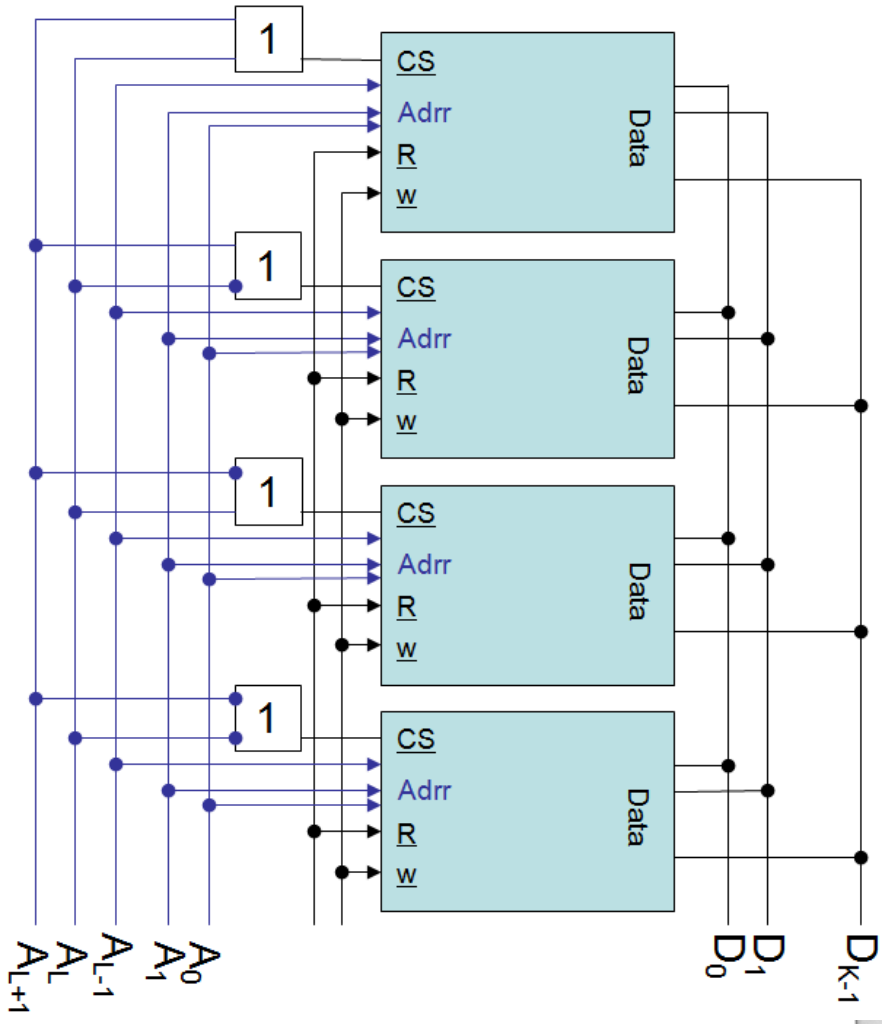


Memóriák

- Kapacitás bővítése

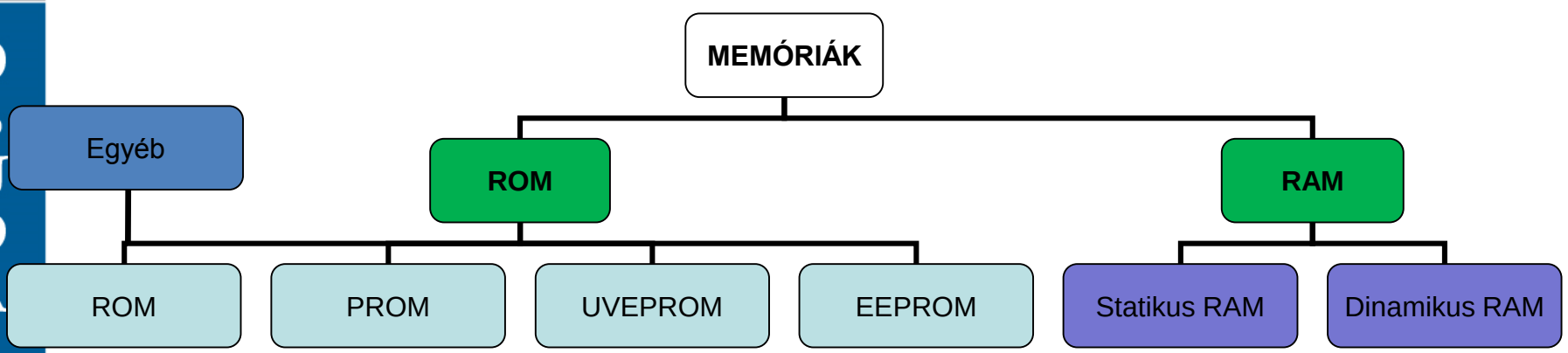
4 x 8k * 8 = 32k * 8





Memóriák

- Írás,olvasás, törlés



- Írás a gyárban
- Törlés nem lehetséges

- 1x felhasználó által is írható (beégethető)
- Törlés nem lehetséges

- Felhasználó által is írható
- Törlés 10”-es UV-s levilágítással

- Felhasználó által is írható
- Elektromosan törölhető

Nem illékony

- Tápfesz nélkül elveszti a tartalmát
- Gyors
- Nem kell frissíteni

- Tápfeszültség alatt is néha frissíteni kell
- Lassú

Illékony