Digitális technika 2.

BMEVIIIAA02

előadás

2020/21 tavaszi félév

Aritmetika

Miről volt szó

Aszinkron számlálók

- A következő fokozat órajele az előző fokozat kimenete
- Számlálási ciklus csökkentése
  - Impulzus formálás

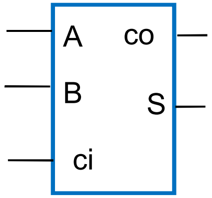
Szinkron számlálók

- Szinkron sorrendi hálózat
- Moore modell
- Z = y
- Egy fokozat akkor lép, ha az összes alacsonyabb helyiérték 1-es
- 4 bites számláló
  - Engedélyezés, törlés, töltés, kaszkádosítás támogatása
- Számlálási ciklus módosítása
  - Szinkron törlés / aszinkron törlés

Bináris összeadás

```
 10110
+01010
------
100000
 111100 ← Átvitel
```

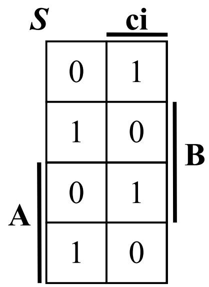
1 bites teljes összeadó



A,B: operandusok
ci: átvitel az előző helyi értékről

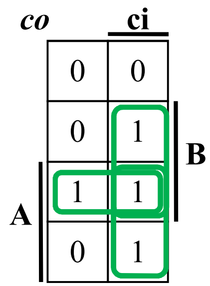
S: összeg
co: átvitel a következő helyi érték felé

A	B	ci	S	co
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1



$S = A \oplus B \oplus ci$

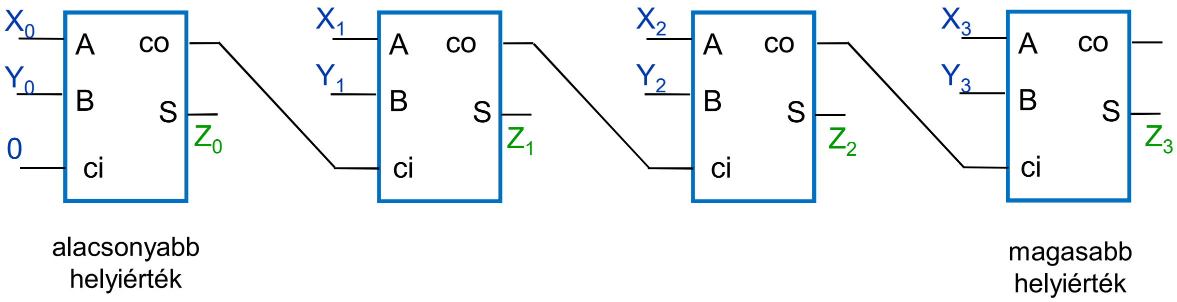
$S^{3}_{1,3}$



$co = A \cdot B + A \cdot ci + B \cdot ci$

Összeadás - kaszkádosítás

4 bites összeadó: $Z_{3..0} = X_{3..0} + Y_{3..0}$

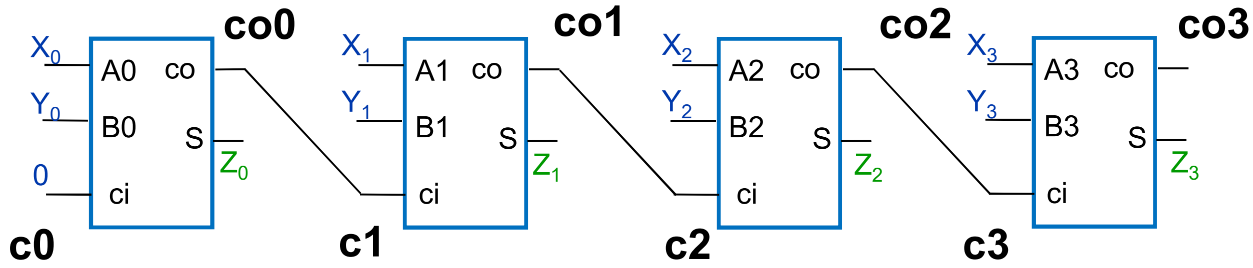


Mikor érvényes az eredmény?

n bites összeadó → $n * \Delta t_{tö}$

Aritmetika - összeadás

Összeadás – gyors átvitelképzés



$$co = A \cdot B + A \cdot ci + B \cdot ci = \underbrace{A \cdot B}_{\substack{\text{G} \\ \text{generate}}} + ci \cdot \underbrace{(A + B)}_{\substack{\text{P} \\ \text{propagate}}}$$

$$co0 = A_0 \cdot B_0 + A_0 \cdot c0 + B_0 \cdot c0 = \boldsymbol{G_0 + P_0 \cdot c0}$$

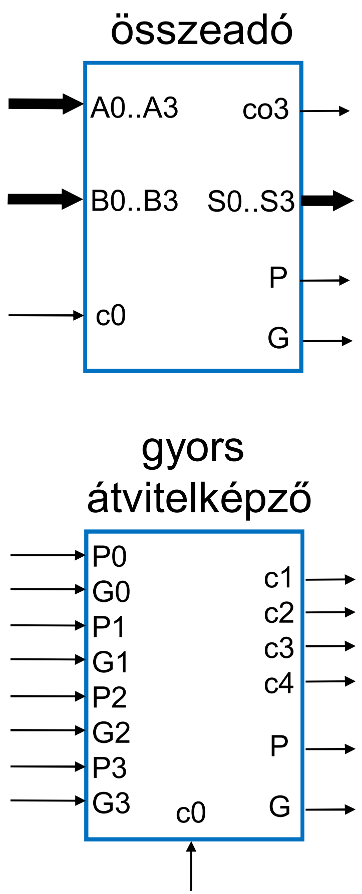
$$co1 = A_1 \cdot B_1 + A_1 \cdot c1 + B_1 \cdot c1 = G_1 + P_1 \cdot c1 = G_1 + P_1 (G_0 + P_0 \cdot c0) = \boldsymbol{G_1 + P_1 \cdot G_0 + P_1 \cdot P_0 \cdot c0}$$

.
.
.

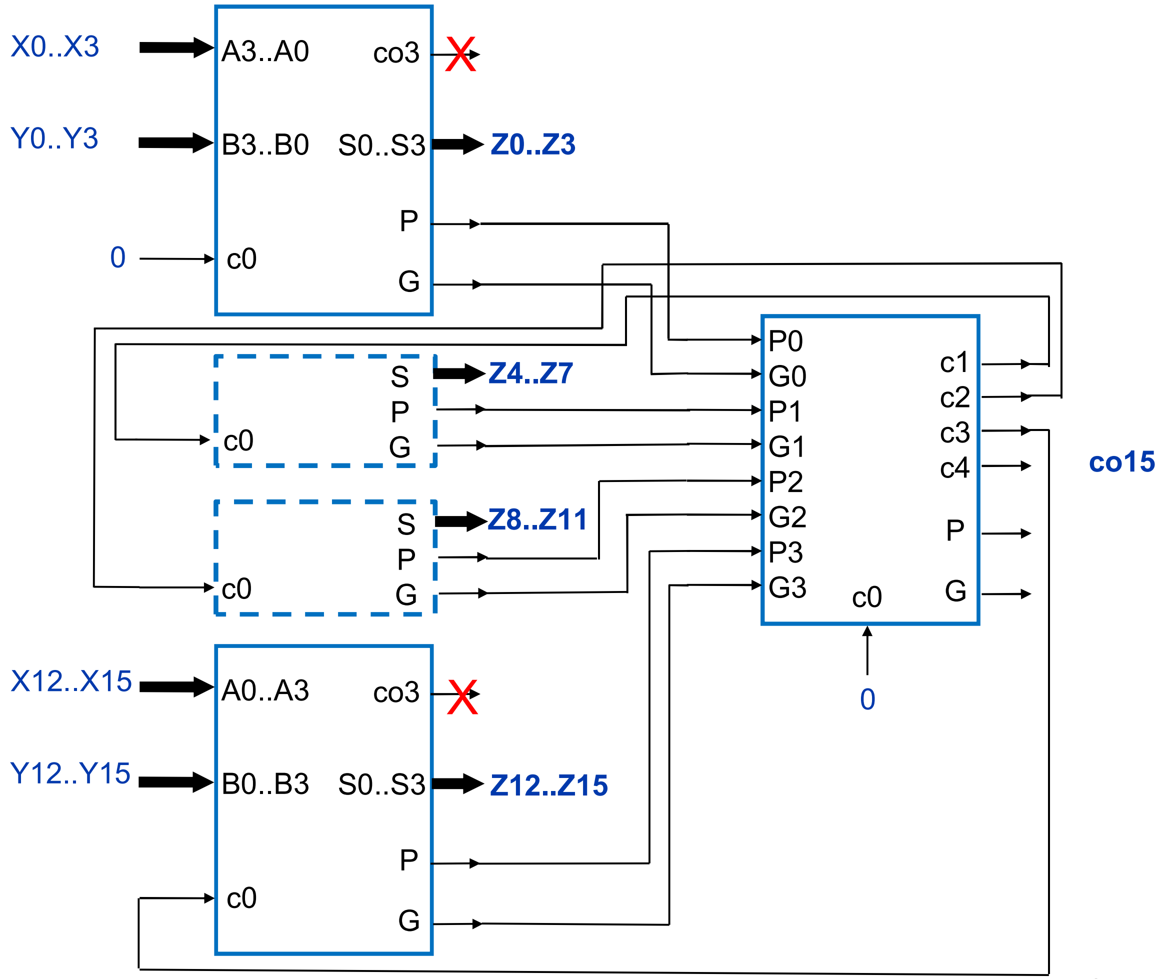
$$co_i = Ai \cdot Bi + Ai \cdot ci + Bi \cdot ci = G_i + Pi \cdot ci = \underbrace{\boldsymbol{G_i + P_i \cdot G_{i-1} + P_i \cdot P_{i-1} \cdot G_{i-2} + \ldots + P_i \cdot P_{i-1} \cdot \ldots \cdot P_0 \cdot c0}}_{\text{3 szintű hálózat}}$$

Aritmetika - összeadás

Gyors átvitelképzés (carry-look-ahead)

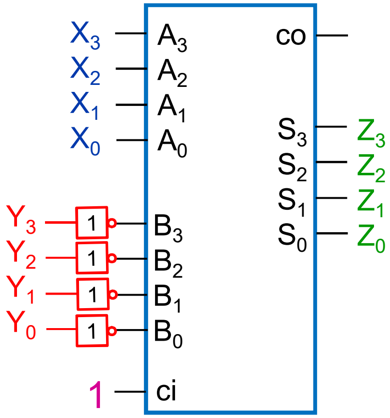


16 bites összeadó (carry-look-ahead)



Aritmetika - kivonás

$Z = X - Y = X + (-Y)$ (-Y) → kettes komplemes



Kettes komplemens képzés

$-Y = \overline{Y} + 1$

Aritmetika

Aritmetikai túlcsordulás: az eredmény már nem ábrázolható
Különböző előjelű operandusok esetén **nem** léphet fel

4 bites kettes komplemens: -8 … +7

6: **0110** 3: **0011** -6: **1010** -3: **1101**

6+(-3) = 3		(-6)+3 = (-3)		6+3 = 9		(-6)+(-3) = -9	
0110	**6**	1010	**-6**	**0**110	**6**	**1**010	-6
+ 1101	**-3**	+ 0011	**3**	+ **0**011	**3**	+ **1**101	-3
0011	**3**	1101	**-3**	**1**001	**-7**	**0**111	**7**

túlcsordulás: azonos előjelű operandusok esetén az eredmény előjele nem egyezik meg az operandusok előjelével

túlcsordulás ≠ átvitel !!

Aritmetika

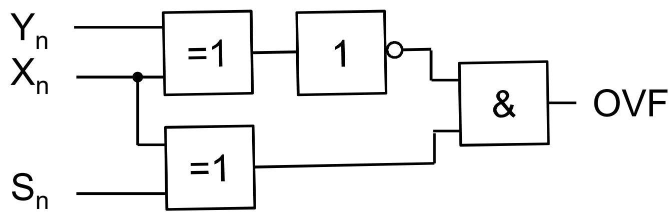
Aritmetikai túlcsordulás: overflow (OVF)

kettes komplemes előjel: a legmagasabb helyiérték

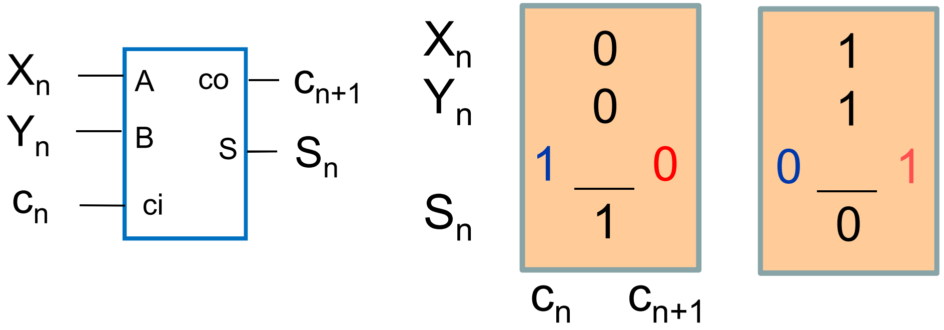
$$OVF = \overline{(X_n \oplus Y_n)} \cdot (X_n \oplus S_n)$$

az operandusok előjele megegyezik

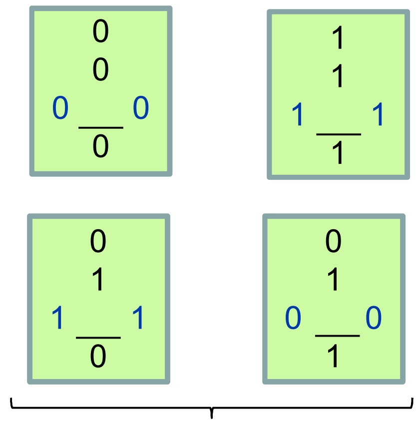
az eredmény előjele különbözik



Ha a legmagasabb helyiértéket előállító összeadó átvitel bemenete és átvitel kimenete elérhető:



$$OVF = c_n \oplus c_{n+1}$$

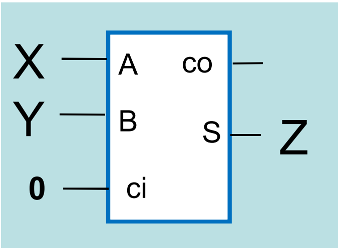


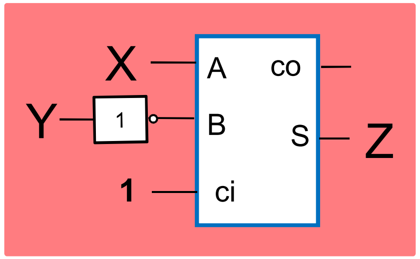
nincs túlcsordulás

Összeadó/kivonó

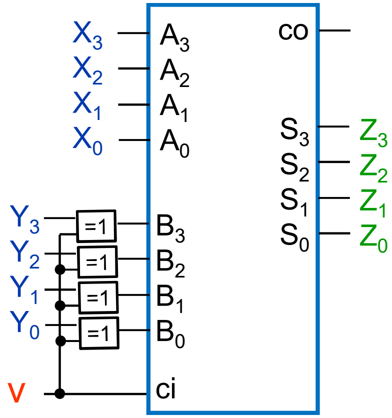
$Z = X + Y$, ha $v=0$ $Z = X - Y$, ha $v=1$

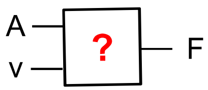
$-Y = \overline{Y} + 1$





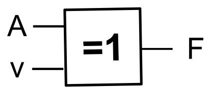
4 bites
összeadó





$F = A$, ha $v = 0$

$F = \overline{A}$, ha $v = 1$



4 bites
összeadó/kivonó

Előjel kiterjesztés

X: n bites pozitív → n+1 bites pozitív

$X_{n-1}\ X_{n-2} \ldots X_1\ X_0$ → $\mathbf{0}\ X_{n-1}\ X_{n-2} \ldots X_1\ X_0$

a szám pozitív kell, hogy maradjon

X: n bites kettes komplemens → n+1 bites kettes komplemens

$X_{n-1}\ X_{n-2} \ldots X_1\ X_0$ → $\mathbf{X_{n-1}}\ X_{n-1}\ X_{n-2} \ldots X_1\ X_0$

előjel

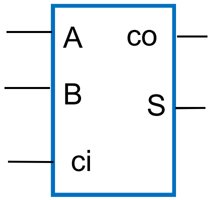
a szám előjele nem változhat meg

-1

4 bites kettes komplemens	`1111`
5 bites kettes komplemens	`11111`
…	
n bites kettes komplemens	`11...11`

Aritmetika – BCD összeadás

BCD összeadó



A,B: 0…9
S: 0…9

Használjunk bináris összeadót

maximális érték

```
    1001  A:9
    1001  B:9
+      1  ci:1
   10011  19
```

	Bináris eredmény					BCD eredmény					
	co	**S3**	**S2**	**S1**	**S0**	**c0**	**S3**	**S2**	**S1**	**S0**	
0	0	0	0	0	0	0	0	0	0	0	**0**
	.	.	.	.	.	.	.	.	.	.	
9	0	1	0	0	1	0	1	0	0	1	**9**
10	0	1	0	1	0	1	0	0	0	0	**16**
11	0	1	0	1	1	1	0	0	0	1	**17**
	.	.	.	.	.	.	.	.	.	.	
16	1	0	0	0	0	1	0	1	1	0	**22**
	.	.	.	.	.	.	.	.	.	.	
19	1	0	0	1	1	1	1	0	0	1	**25**

+6

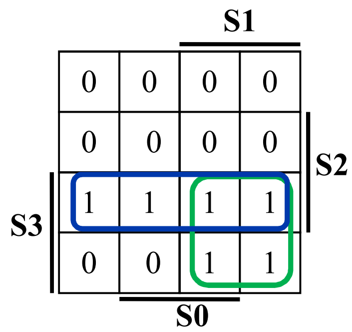
bináris eredmény <= 9 → BCD eredmény = bináris eredmény
bináris eredmény >9 → BCD eredmény = bináris eredmény + 6

Aritmetika – BCD összeadás

BCD átvitel előállítása

	co	S3	S2	S1	S0	C_{BCD}
0	0	0	0	0	0	0
	.	.	.	.	.	0
9	0	1	0	0	1	0
10	0	1	0	1	0	1
11	0	1	0	1	1	1
	.	.	.	.	.	1
15	0	1	1	1	1	1
16	1	0	0	0	0	1
	.	.	.	.	.	1
19	1	0	0	1	1	1

S3..S0 > 9



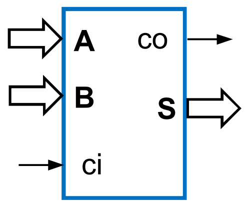
$S3 \cdot S2 + S3 \cdot S1$

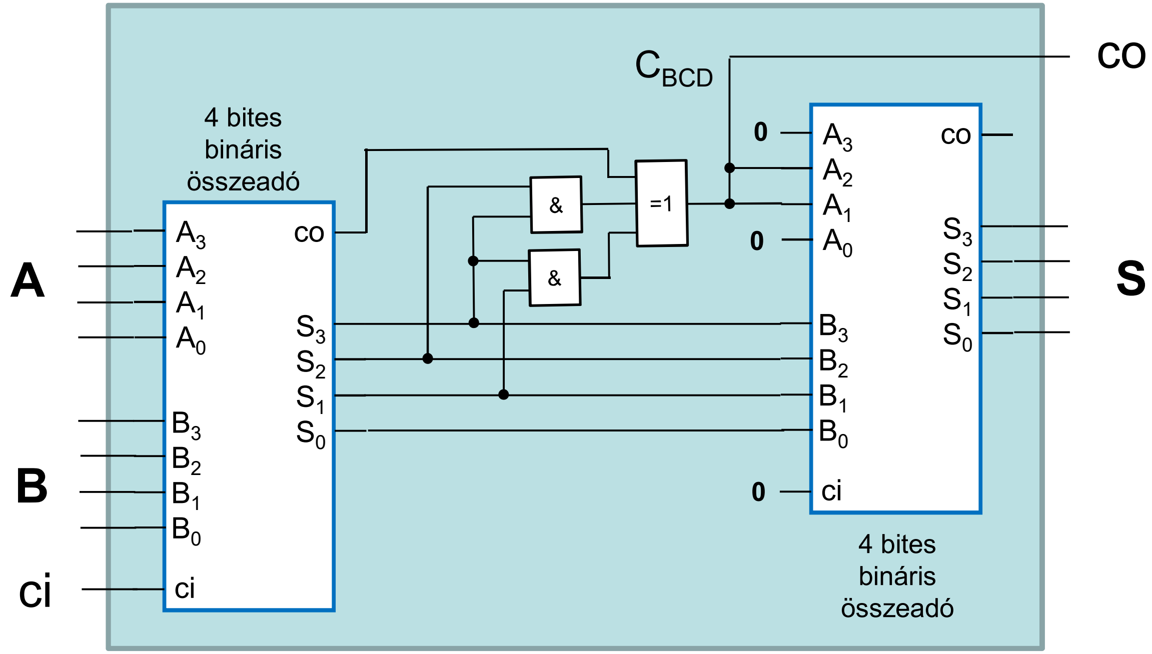
co = 1

$$C_{BCD} = co + S3 \cdot S2 + S3 \cdot S1$$

Aritmetika – BCD összeadás

BCD összeadó → A, B, S: BCD számok





$C_{BCD} = co + S3 \cdot S2 + S3 \cdot S1$

6: 0110

Aritmetika – szorzás

Decimális szám szorzása „kézzel”

```
123 * 456
492
 615
  738
56088
```

Mit kell tudni?
- Egyjegyű számmal szorozni
- Összeadni

Mekkora lesz az eredmény?

999 * 999 = 998001
6 jegyű szám

Két n jegyű szám szorzata legfeljebb 2n jegyű

Aritmetika – szorzás

Bináris számok szorzása

Egyjegyű bináris szorzás

X	Y	Z
0	0	0
0	1	0
1	0	0
1	1	1

ÉS

4 bites szorzás → eredmény 8 bites

```
      (13)     (11)
      1101  *  1011
      ----
      1101   ]  1
     1101        ]  2
    0000             ]  3
   1101
   --------
   10001111  (143)
```

Hány bites összeadó kell?

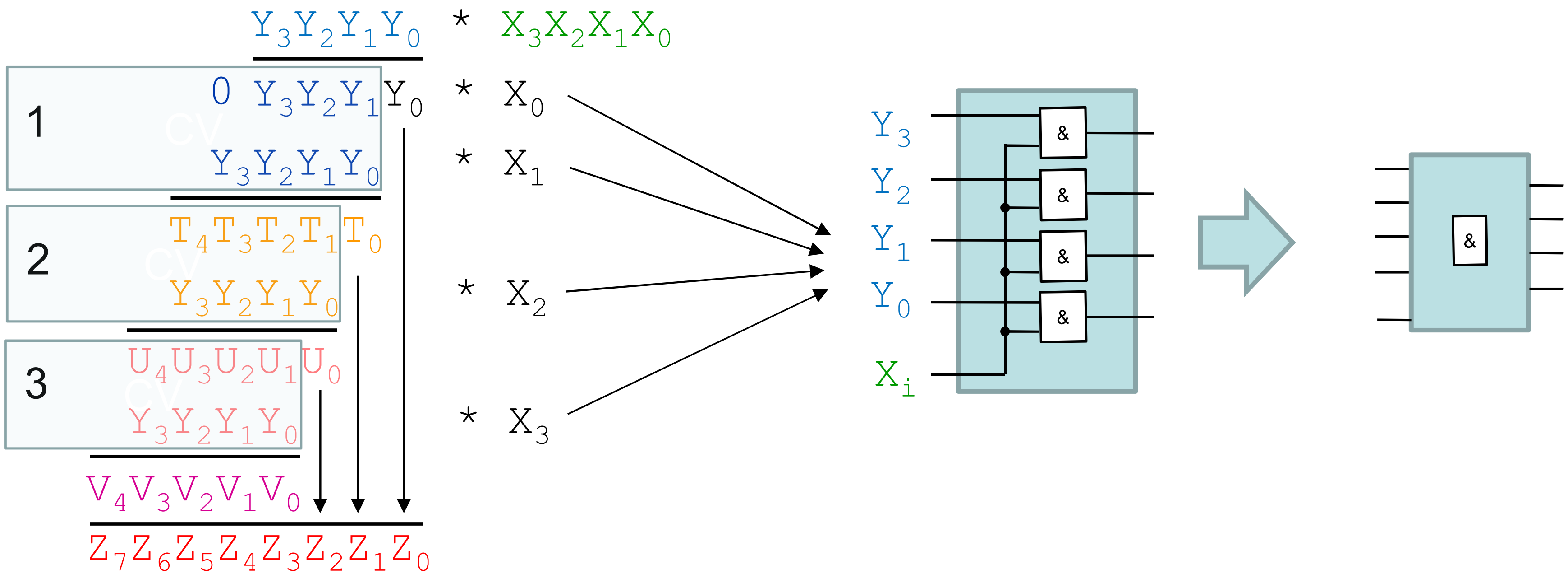
Egyszerre elegendő 4 bitet összeadni

Hány összeadó kell?

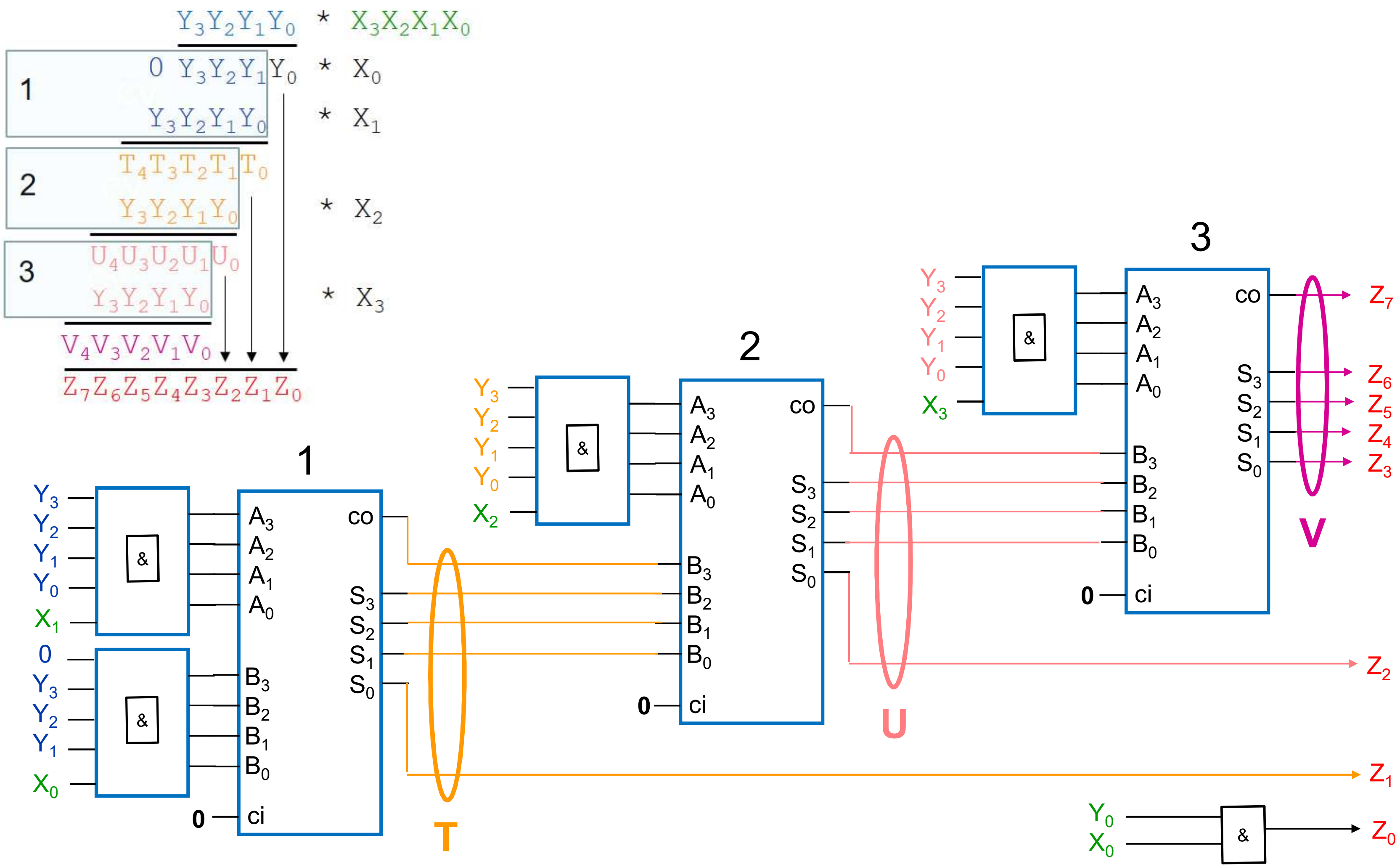
4 operandus → 3 összeadó

Aritmetika – szorzás

4 bites szorzás

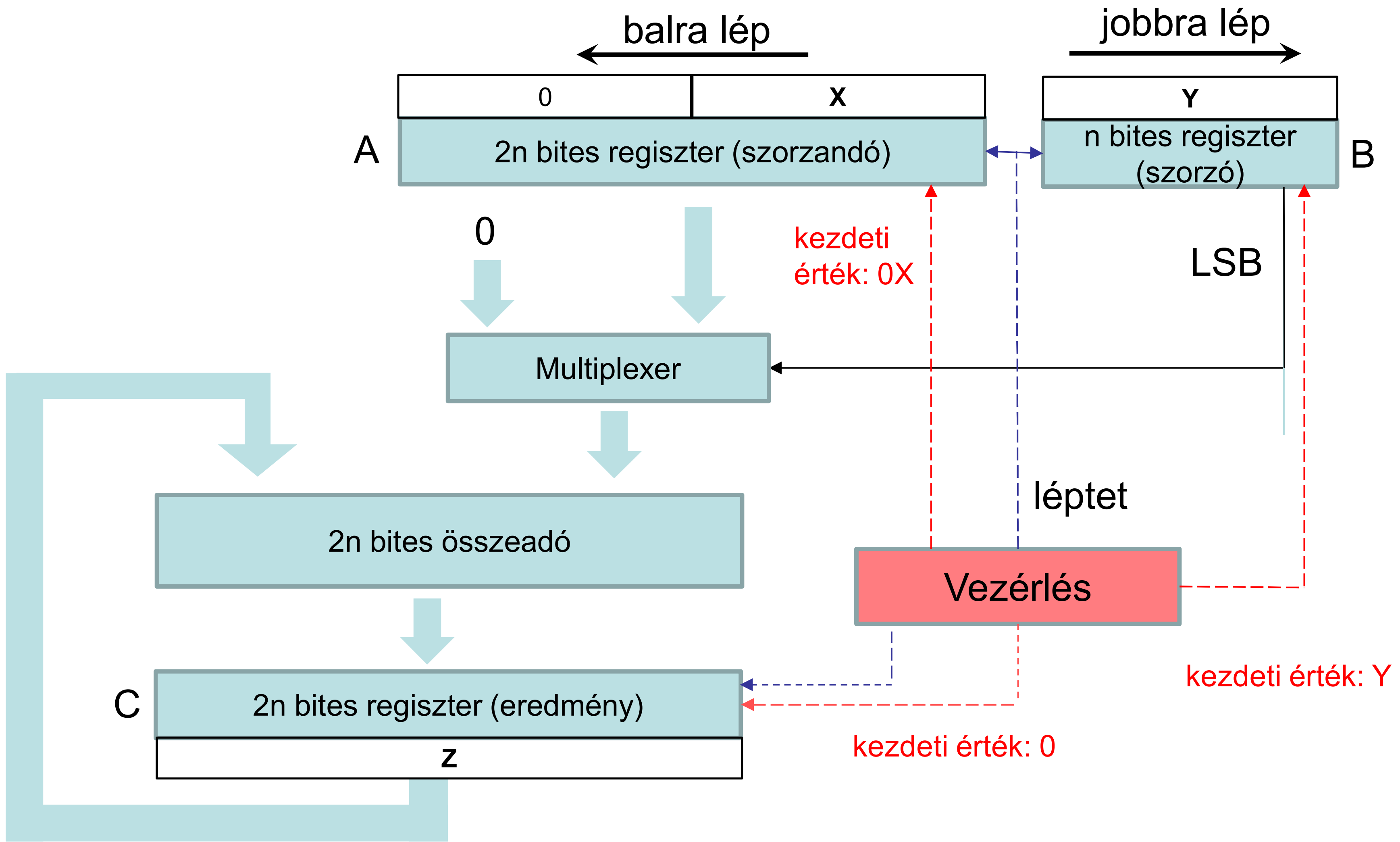


Aritmetika – szorzás



Aritmetika – szorzás

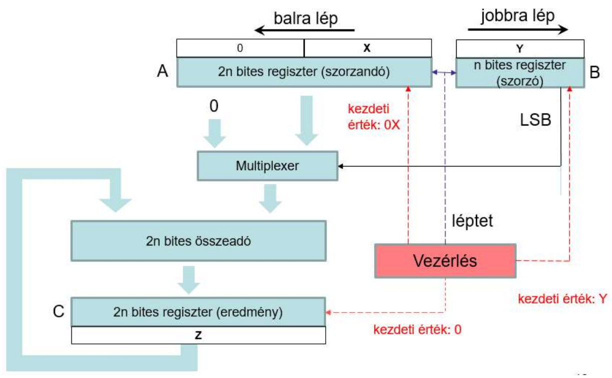
Bináris számok szorzása – sorrendi hálózattal Z = X * Y



Aritmetika – szorzás

(A) (B) (C)

1010 * 0101 = 00110010 (10 * 5 = 50)



	A	B	C	
1	00001010	0101	00000000 00001010	kezdeti értékadás $C = C + A * B_{LSB}$
2	00010100	0010	 00001010	Léptetés $C = C + A * B_{LSB}$
3	00101000	0001	 00110010	Léptetés $C = C+ A * B_{LSB}$
4	01010000	0000	 00110010	Léptetés $C = C + A * B_{LSB}$

00000000 C
00001010 $A*B_{LSB}$
00001010

00001010 C
00000000 $A*B_{LSB}$
00001010

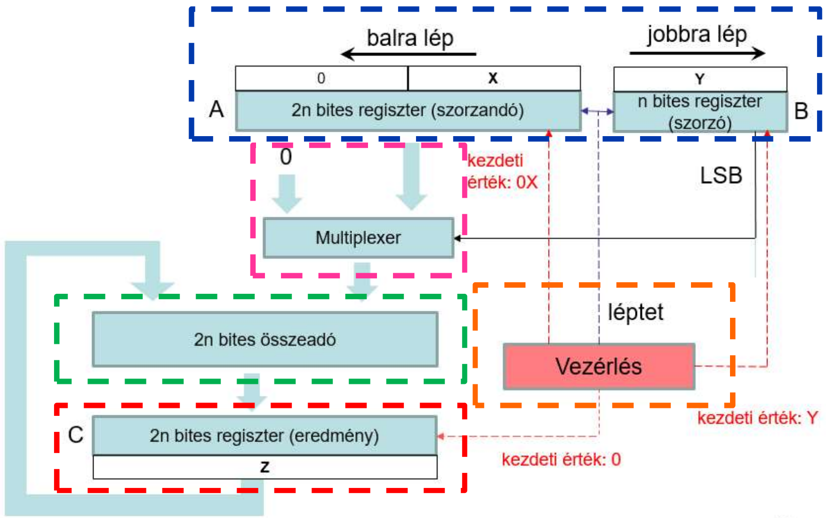
00001010 C
00101000 $A*B_{LSB}$
00110010

00110010 C
00000000 $A*B_{LSB}$
00110010

Aritmetika – szorzás

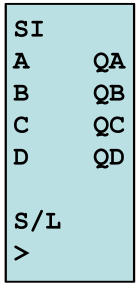
4 BITES SZORZÓ

Építőelemek

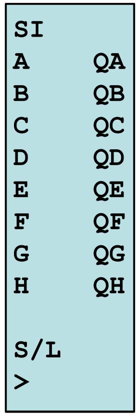


léptető regiszter

4 bites

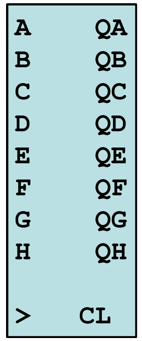


8 bites



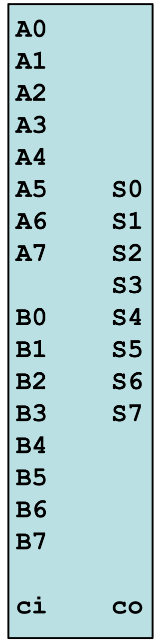
regiszter

8 bites



összeadó

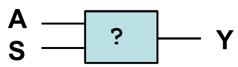
8 bites



S/L	>	művelet
0	↑	lép
1	↑	tölt

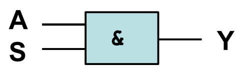
CL	>	művelet
1	x	aszinkron törlés
0	↑	tárolás

Multiplexer



A	S	Y
x	0	0
A	1	A

ÉS kapu



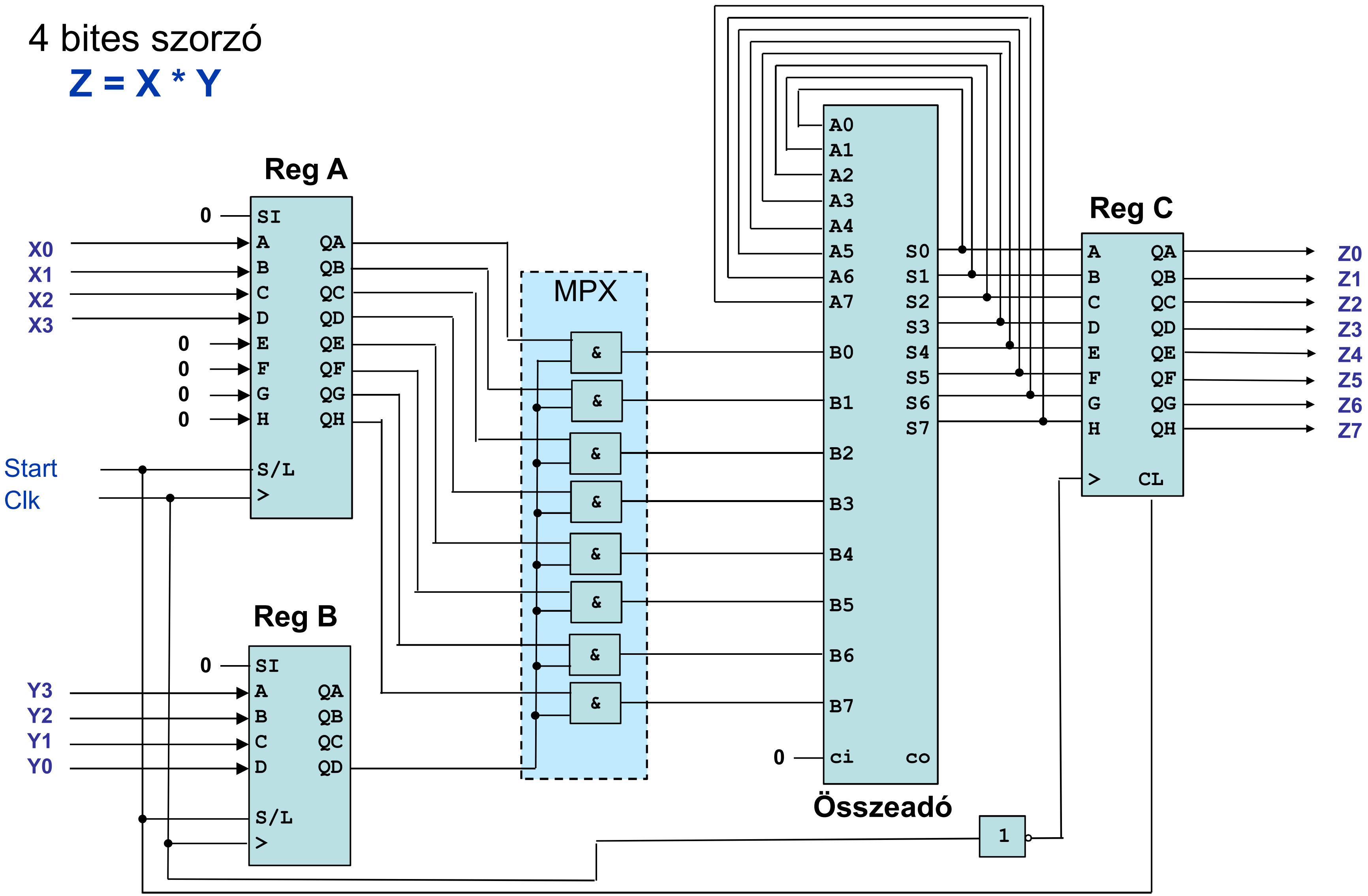
Vezérlés

Léptető regiszterek (A,B): lépés felfutó élre
Regiszter (C): tárolás lefutó élre

Aritmetika – szorzás

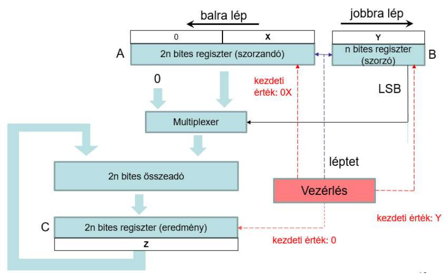
4 bites szorzó

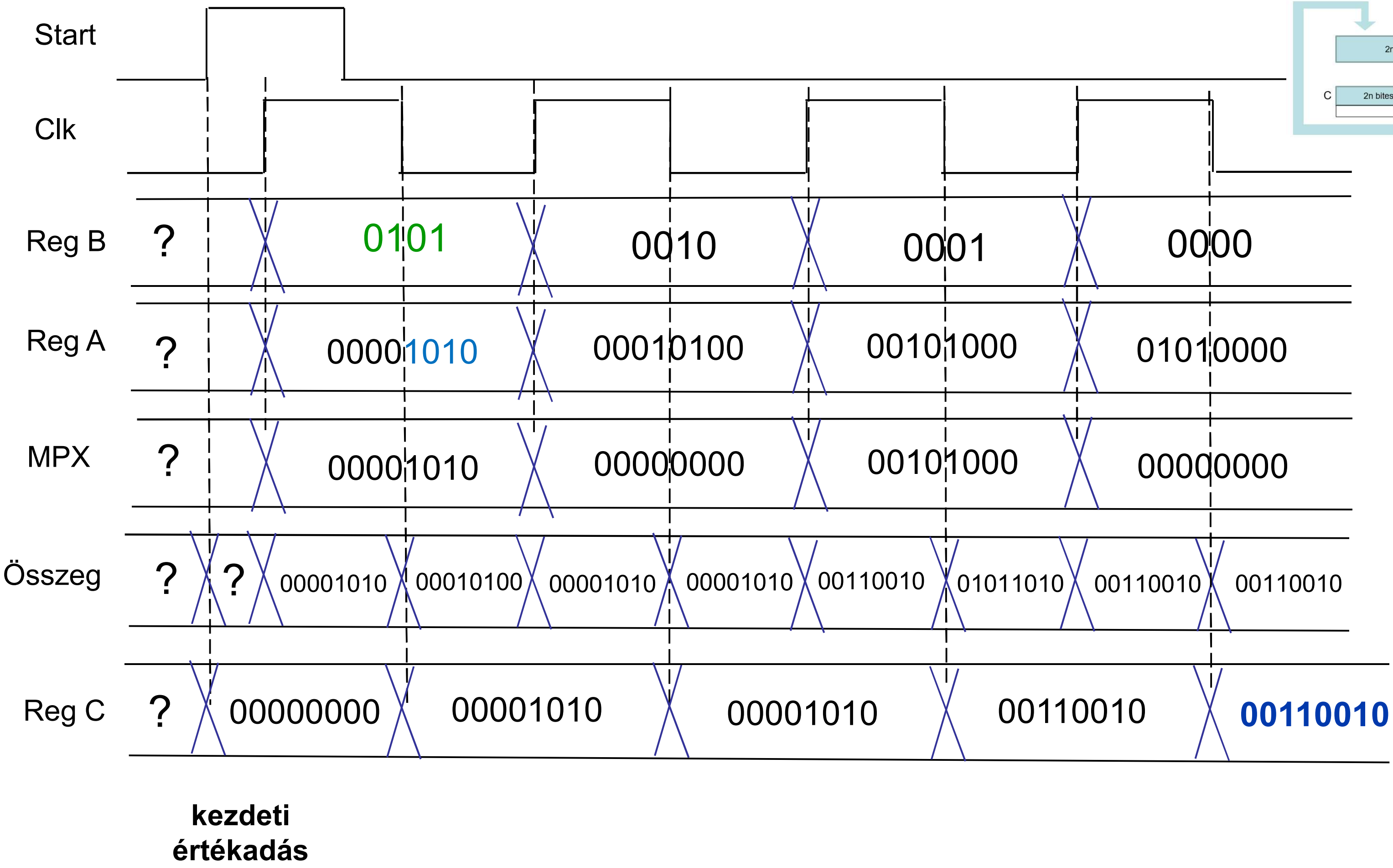
Z = X * Y



Aritmetika – szorzás

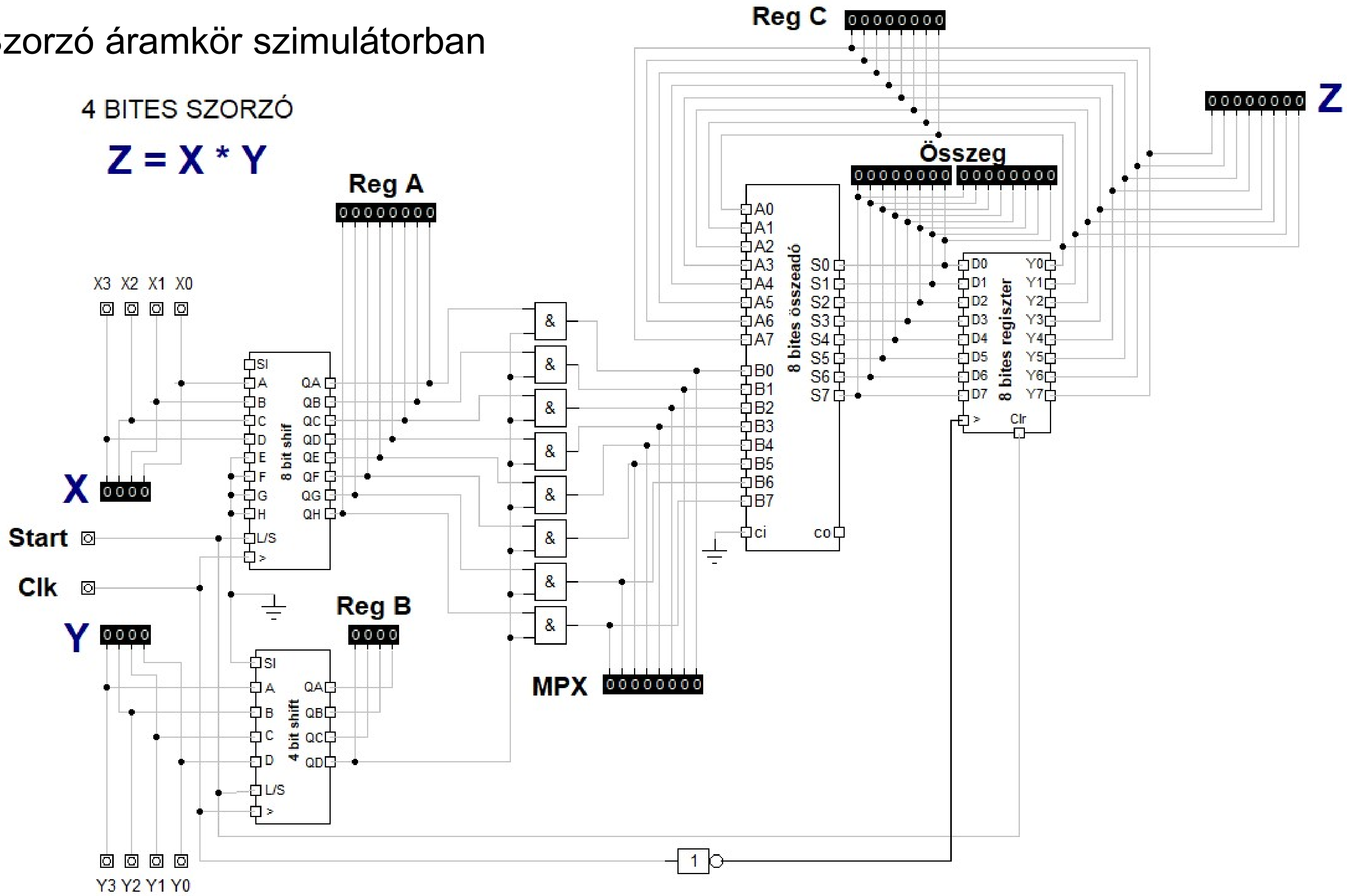
1010 * 0101 = **00110010**





Aritmetika – szorzás

Szorzó áramkör szimulátorban



■ ■ ■