



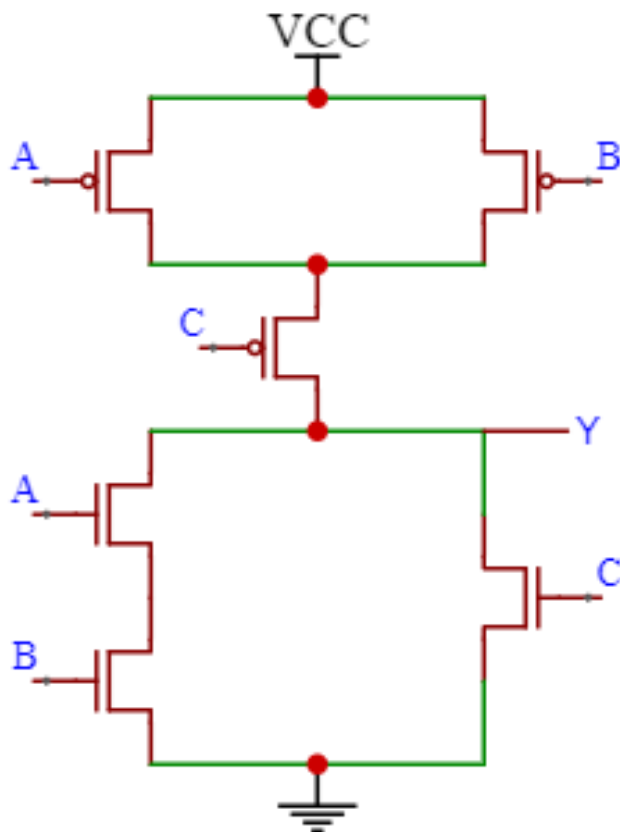
# Elektronika alapjai

## 4. Gyakorlat – CMOS áramkörök

Összeállította:

Ress Sándor, Jani Lázár, Krammer Olivér, Straubinger Dániel

### 1. Feladat (A) - Milyen logikai függvényt valósít meg az alábbi CMOS kapu?



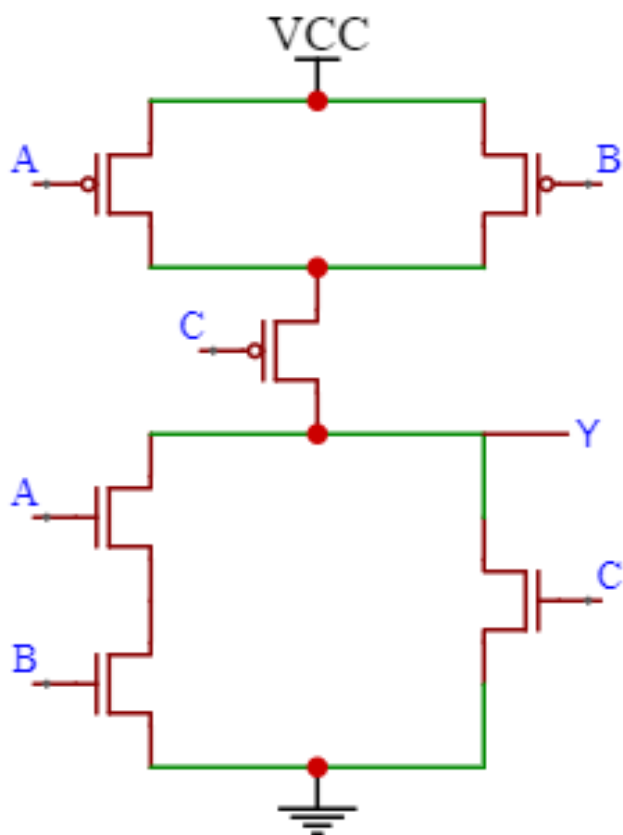
#### Complementary metal-oxide-semiconductor

- pMOS tranzisztorok source-a a tápra van kötve
- pMOS tranzisztorok logikai 0 (0V) esetén vezetnek
- nMOS tranzisztorok source-a a földre van kötve
- nMOS tranzisztorok logikai 1 (tápfeszültség) esetén vezetnek

A kimenetet a

- pMOS tranzisztorokból álló **felhúzó hálózat** (*pull-up network*),
- És az nMOS tranzisztorokból álló **lehúzó hálózat** (*pull-down network*) állítja be
- Statikus CMOS kapuk esetén minden bemenő kombináció esetén **kizárólag** a felhúzó vagy a lehúzó hálózat vezet

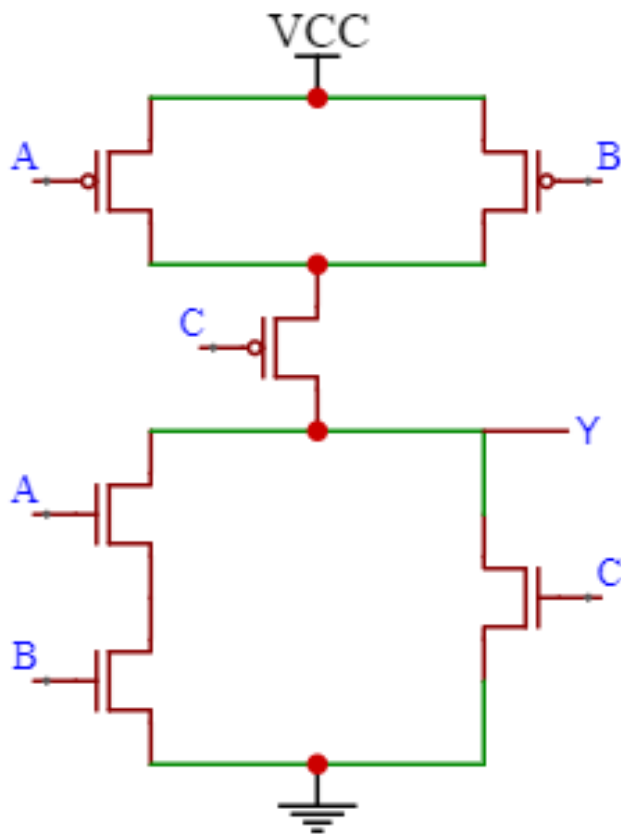
### 1. Feladat (A) - Milyen logikai függvényt valósít meg az alábbi CMOS kapu?



Írjuk fel az igazságtáblát!

A	B	C	Y
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

## 1. Feladat (A) - Milyen logikai függvényt valósít meg az alábbi CMOS kapu?



Írjuk fel az igazságtáblát!

A	B	C	Y
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

A kimenet logikai nulla, ha:

- az A és B-hez tartozó NMOS tranzisztor egyszerre vezet, azaz a logika függvény A ÉS B
- a C-hez tartozó nMOS tranzisztor vezet

Logikai kifejezés:  $Y = \overline{AB + C}$

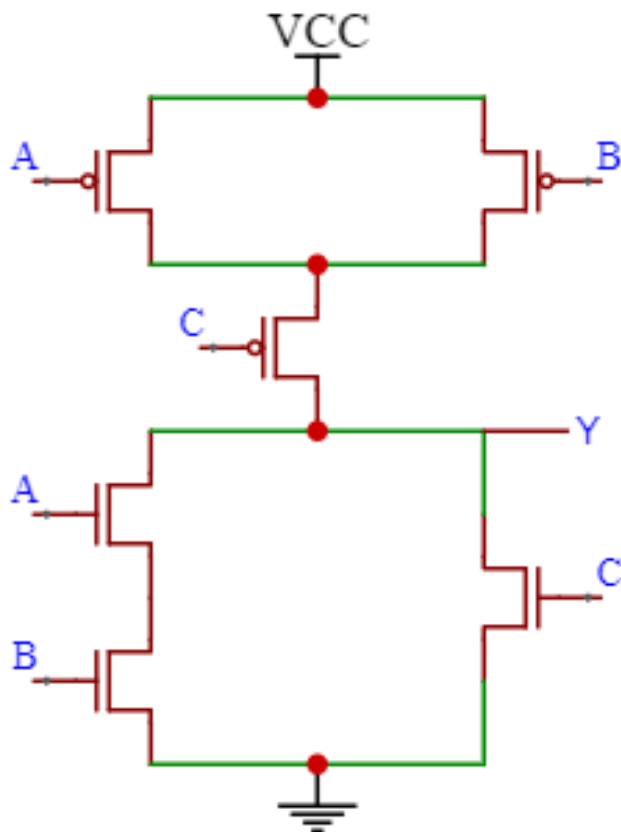
Pull-up network ellenőrzése (De Morgan átalakítás):

$$Y = \overline{AB + C} = \overline{AB} \cdot \overline{C} = (\overline{A} + \overline{B})\overline{C}$$

A kimenet akkor lesz logikai 1, ha

- C=0, és
- A vagy B nulla

**1. Feladat (B)** - Hány tranzisztorral tudnánk megvalósítani ezt a kaput, "hagyományos" (NAND, NOR, INV) alapkapuk használatával?



Logikai kifejezés átalakítása De Morgan-t alkalmazva:

$$Y = \overline{AB + C} = \overline{AB} \cdot \overline{C} = \overline{\overline{\overline{AB}} \cdot \overline{C}} = \overline{\overline{AB} + C}$$

Szükséges kapuk:

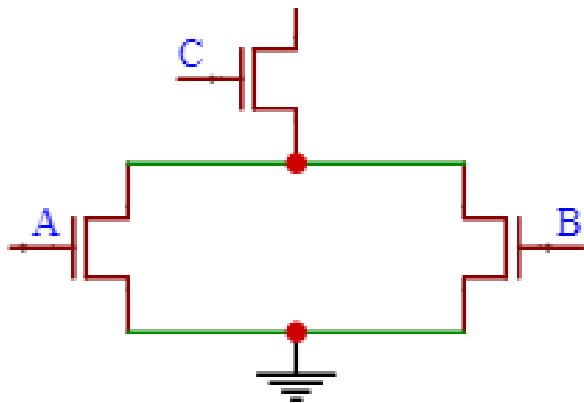
- 2 bemenetű NAND  $\overline{AB}$  logikai kifejezéshez
- Inverter  $\overline{C}$  előállításához
- 2 bemenetű NOR

[A kapu interaktív kapcsolási rajza](#)

**1. Feladat (C)** - Tervezze át a kaput, hogy az  $Y = \overline{(A + B)C}$  függvényt valósítsa meg!

A pull-down network közvetlenül meghatározható a logikai kifejezésből:

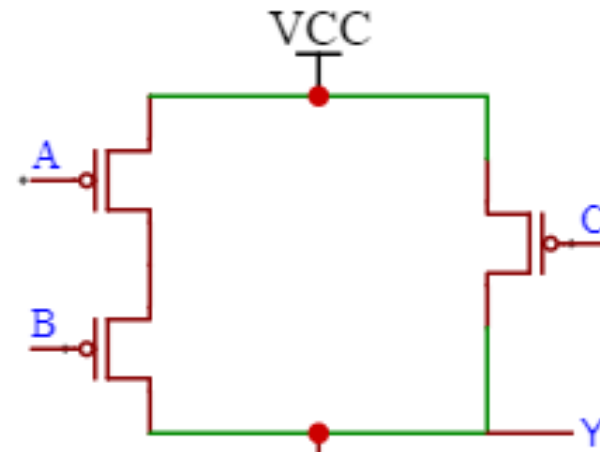
- ÉS kapcsolat sorba kötött tranzisztorokat,
- VAGY kapcsolat párhuzamosan kötött tranzisztorokat jelent.



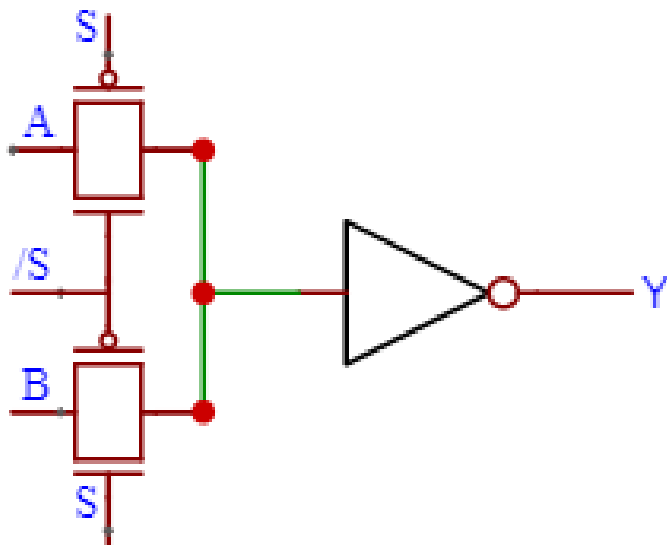
A pull-up network megvalósításához két módszer is alkalmazható:

- Pull-down network komplementere
  - nMOS helyett pMOS,
  - párhuzamosan kapcsolt tranzisztorokat sorosan,
  - sorosan kapcsolt tranzisztorokat párhuzamosan kell kötni
- De Morgan azonosságokkal

$$Y = \overline{(A + B)C} = \overline{\overline{\overline{A + B} + \overline{C}}} = \overline{\overline{A} \cdot \overline{B} + \overline{C}}$$



**2. Feladat** - Milyen logikai függvényt valósít meg az alábbi kapcsolás? A /S rajztechnikai okokból az S jel negáltját jelenti.



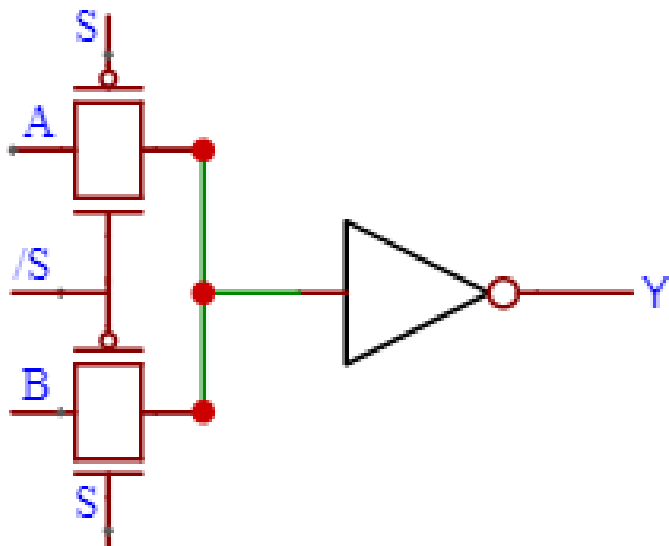
Működés:

- S=0 esetén a felső transzferkapu átengedi az A jelet, az alsó transzferkapu zárva van.
- S=1 esetén a felső tranzisztor zárt, az alsó átengedi a B jelet.
- Inverter bemenetére a transzfer kapuk kimenete kapcsolódik.

Összefoglalva:

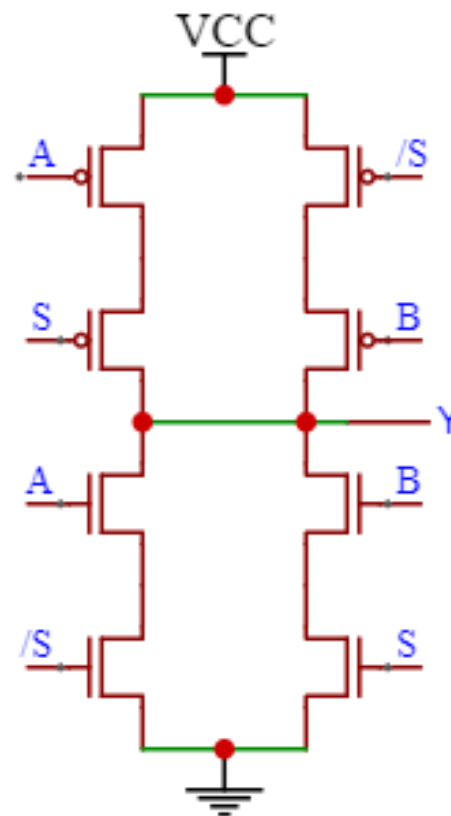
$$Y = \overline{A\bar{S}} + BS$$

**2. Feladat** - Milyen logikai függvényt valósít meg az alábbi kapcsolás? A /S rajztechnikai okokból az S jel negáltját jelenti.



A megvalósított logikai kifejezés:

$$Y = \overline{A\bar{S}} + BS$$

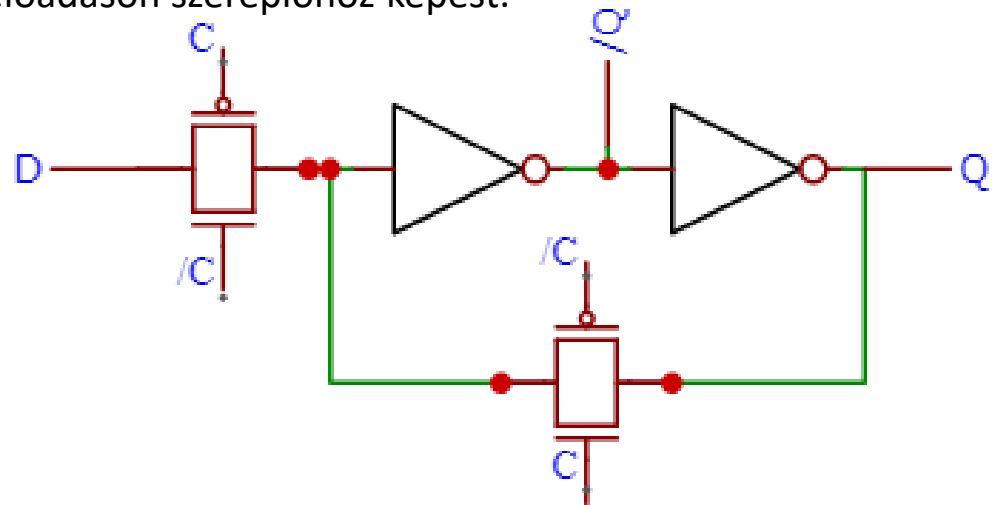
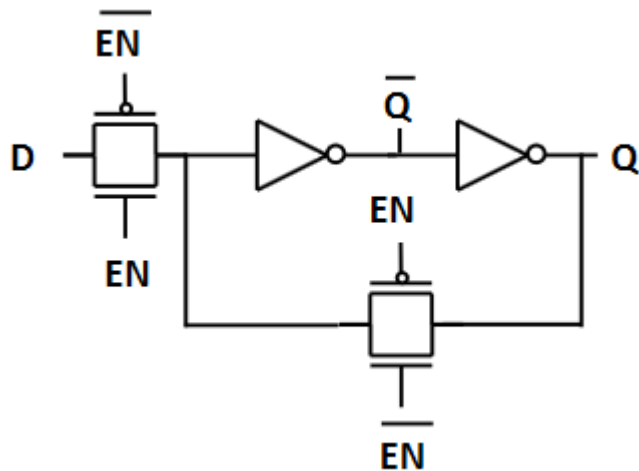


A két megvalósítás megtekinthető itt



### 3. Feladat - Hogyan kellene átalakítani az előadáson látott latch-et úgy, hogy az órajel alacsony szintjén legyen átlátszó?

Fel kell cserélni az órajelek polaritását az előadáson szereplőhöz képest.



Ha az órajel magas, akkor az első transzfer kapu leválasztja a bemenetet, a második pedig létrehozza a két inverter között a visszacsatolást, azaz a 2. inverter kimenetét az első inverter bemenetére kapcsolja. Ez a rendszer stabil, amíg tápfeszültség van, megőrzi az állapotát.

Az első transzfer kapu órajel alacsony szintjén vezet, ekkor a visszacsatoló transzfer kapu leválaszt. Ekkor a bemenet a két inverteren keresztül íródik a kimenetre, azaz a latch átlátszó.

[A működés interaktívan](#)

**4. Feladat (1)** - Egy otthon router átlagos teljesítménye 5W. Mennyibe kerül a havi üzemeltetése, ha 1kWh kb. 40Ft? Mennyibe kerül egy átlagosan 70W-os PC alapú szerver 0-24 otthoni üzemeltetése havonta?

$$E = P \cdot t$$

- Egy óra alatt az elfogyasztott energia 0,005 kWh. Egy átlagos 30 napos hónappal számítva  $0,005 \cdot 24 \cdot 30 = 3,6$  kWh, ez kb. 144 forintot jelent.
- A PC-s szerver által elfogyasztott energia 0,07 kWh, 30 nap alatt  $0,07 \cdot 24 \cdot 30 = 50,4$  kWh, ez 2000 Ft havonta, ami már szemmel látható.

**4. Feladat (2)** - Egy CMOS technológiával készült SoC órajele 1GHz, tápfeszültsége 3V. A rendszer így teljesen feltöltött akkumulátorról 12 órát működik. Az órajelet felére csökkentjük.

a) Meddig fog teljesen feltöltött akkumulátorról működni?

$$P \sim fV_{DD}^2 \qquad E_0 = P_0 \cdot t_0 \qquad E_0 = \frac{P_0}{2} \cdot t_1 \rightarrow t_1 = 2 \cdot t_0$$

Mivel a frekvencia a fele lesz, így az SoC fogyasztása is felére csökken. Így kétszer annyi ideig fog a rendszer akkumulátorról működni, azaz 24 órát.

b) Egy taszk az eredeti rendszeren 100s-ig futott és 1kJ energiát használt fel. Mennyi ideig fog futni a módosított rendszeren és mennyi energiát használ fel?

A teljesítményt felére csökkentjük, de a taszk a fele akkora órajellel kétszerannyi ideig fog futni, tehát 200s-ig és ugyanúgy 1kJ energiát használ el.

**4. Feladat (2)** - Egy CMOS technológiával készült SoC órajele 1GHz, tápfeszültsége 3V. A rendszer így teljesen feltöltött akkumulátorról 12 órát működik. Az órajelet felére csökkentjük.

c) Egy folyamatos, felügyelet jellegű taszkat futtatunk, amely biz. események bekövetkezésekor 10ms múlva tud reagálni. Hogyan változik ez a reakcióidő a felére csökkentett órajelű rendszeren?

$$t_0 = \frac{N}{f_0} \rightarrow t_1 = \frac{N}{f_0/2} = 2 \cdot t_0$$

A feleakkora órajel miatt kétszer annyi idő lesz, mire reagál, azaz 20ms

d) Hogyan alakulnak ezek az értékek, ha a tápfeszültséget is kétharmadára csökkentjük?

$$P \sim fV_{DD}^2 \rightarrow P' \sim \frac{f}{2} \left( \frac{2}{3} V_{DD} \right)^2 = \frac{1}{2} \cdot \frac{4}{9} \cdot fV_{DD}^2 \rightarrow \frac{P'}{P} = \frac{2}{9}$$

Azaz 4,5x több ideig fog működni, 54 órát. Az órajelet felére csökkentettük, azaz a taszk kétszer annyi ideig fog futni, 200 másodpercig, de a teljesítményigény 2/9 részére csökken, így a felhasznált energia:  $E = 1kJ \cdot \frac{2}{9} = 444J$ . A reakcióidő ugyanúgy 20ms lesz, mint az előző esetben.

**4. Feladat (3)** - Egy rendszerben a mikroprocesszor magfeszültsége 3GHz-en 1,1V. A rendszert kétprocesszorossá szereljük át és 1,5GHz frekvencián működtetjük, 700mV tápfeszültségről. Feltételezzük, hogy a processzor fogyasztásának nagy részét a töltéspumpálás okozza.

$$P_1 \sim f_1 V_{DD1}^2 \quad P_2 \sim 2 \cdot f_2 V_{DD2}^2 \quad \frac{P_2}{P_1} = 2 \cdot \frac{1,5}{3} \cdot \frac{0,7^2}{1,1^2} = 0,404$$

Tehát a szükséges elektromos teljesítményigény ~40% csökken, míg a számítástechnikai értelemben vett teljesítménye (performance) változatlan. Azaz a kétprocesszoros, kisebb frekvencián működtetett rendszer energiahatékonysága (performance/W) két és félszeres, feltéve, hogy a taszkokat hatékonyan el tudjuk osztani a processzorok között.

**5. Feladat** - Egy kétbemenetű CMOS NAND/NOR kapu bemenete  $p$  valószínűséggel változik meg. Mekkora valószínűséggel változik meg a kapu kimenete?

NAND kapu

A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

A valószínűség ugyanaz lesz, mert mindkét kapu igazságtáblájában az 1 kimenethez csupán egy kombináció tartozik.

A kimenet kizárólag az **A=1, B=1** kombináció esetén 1. Azt kell meghatározni, hogy a többi kombinációból minnek kell változnia, hogy ez a kombináció legyen.

Nézzük végig minden lehetséges bemenetre!

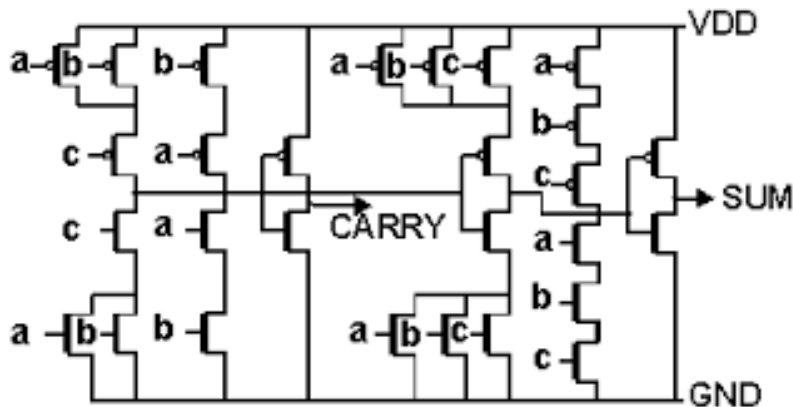
- **A=0, B=0:** mindkét bemenetnek váltania kell, hogy megváltozzon a kimenet, azaz ekkor a kimenet megváltozásának valószínűsége  $p^2$
- **A=0, B=1:** A-nak 1-be kell váltania, B-nek maradnia kell 1-ben, ennek a valószínűsége  $p(1-p)$
- **A=1, B=0:** A-nak változatlanul kell lennie, B-nek 1-be kell váltania, azaz a valószínűség  $(1-p)p$
- **A=1, B=1:** Akármelyik megváltozik, a kimenet is változni fog. Annak a valószínűsége, hogy egyik sem változik  $(1-p)^2$ . Ezt a teljes valószínűségből kivonva megkapjuk, hogy mennyi a kimenet váltásának valószínűsége  $1-(1-p)^2$

A fenti eseteket összegezni kell  $\frac{1}{4}$ -es súllyal:

$$p(ki) = \frac{1}{4} (p^2 + p(1-p) + (1-p)p + 1 - (1-p)^2) = p - \frac{p^2}{2}$$

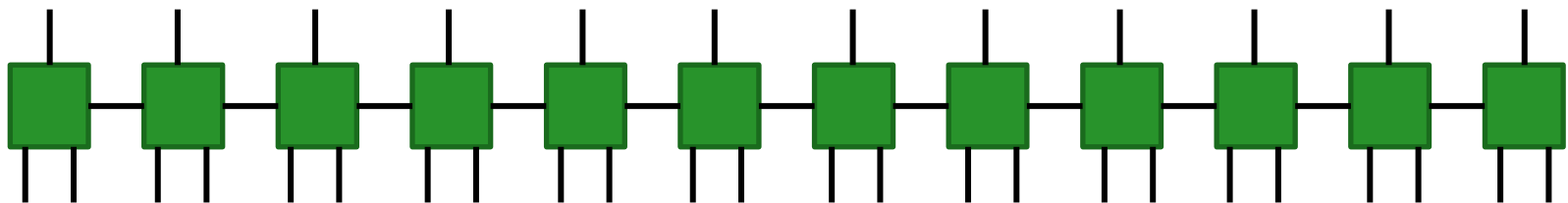
**6. Feladat** - Egy teljes összeadó esetén a carry 100ps, az összeg pedig a carry elkészülése után további 100ps után készül el. Mekkora lesz egy 12 bites, ripple carry összeadó legnagyobb késleltetése?

Teljes összeadó (full adder)



A ripple carry összeadó teljes összeadó áramköröket fűz fel. A **kritikus út** (legnagyobb késleltetésű jelút) tartalmazni fogja a carry láncot. Mivel a bitek számával lineárisan nő a kritikus út hossza, ezért sokbites összeadók esetén ez az architektúra nem előnyös.

Az összeg (SUM) a kiszámolt carry bitet is felhasználja az eredmény előállításához, így az MSB késleltetése az alábbiak szerint alakul



A késleltetés  $t_{pd} = 12 \cdot 100ps + 100ps = 1300ps$  lesz.