

# A digitális CMOS logika alapjai

## 3. előadás

- MOS tranzisztor
- Digitális logika alapfogalmai
- CMOS alapkapuk, komplex kapuk, tárolók

# A MOS tranzisztor

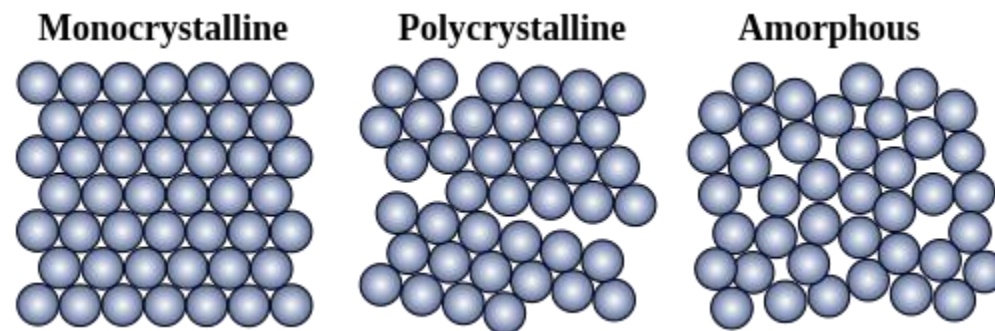
## ■ MOS: Metal-Oxide-Semiconductor

- Nevét a működést biztosító anyagstruktúráról kapta, azaz fém, a félvezető oxidja, félvezető.
- A félvezető szilícium, mivel a  $\text{SiO}_2$  (más néven kvarc) stabil anyag.
- Valójában ez az egyik fő oka annak, hogy szilíciumot használunk más félvezetők helyett
- A fém kezdetben alumínium volt, később szinte minden technológiában polikristályos szilíciumot használtak, az utóbbi években azonban újra fémből készül. (meglehetősen „trükkös” ötvözetek – ez a fém-gate-s / metal gate technológia)

## ■ Ma a vezető technológia

- 1957: az első MOS tranzisztor
- 1970: az első nagy tételben árult MOS IC (3-tranzisztoros 1 kbit DRAM, Intel)
- Ma: több milliárd MOS tranzisztor/chip
- 2005-ben állítólag több MOS tranzisztor készült, mint ahány szem rizst termesztettek. (forrás: „SEMI Annual Report’05” – nem ellenőrzött...)

## Kitérő: egykristály, polikristály, amorf anyag

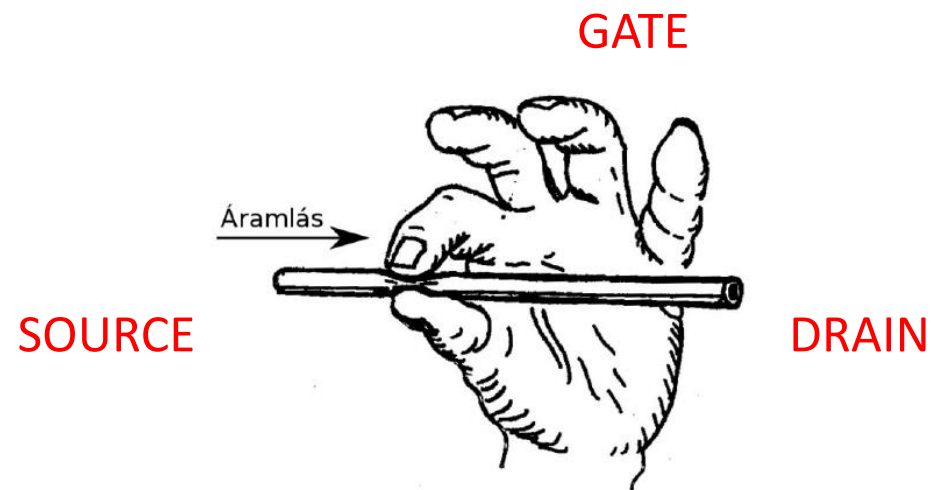


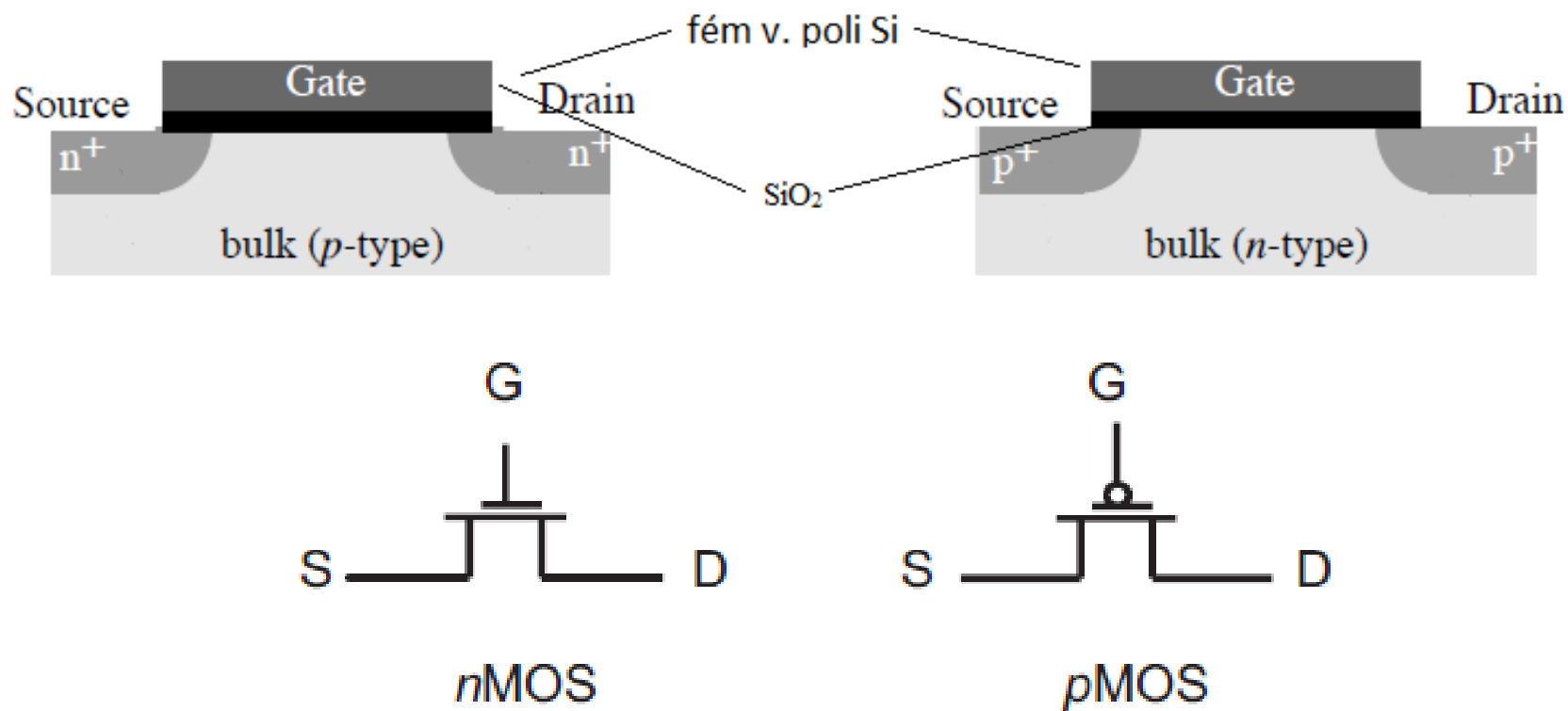
- Egykristály: hosszú távú rendezettség
- Polikristály: egykristály szemcsékből áll, különböző orientációban
- Amorf anyag: nincs, vagy csak nagyon rövidtávú a rendezettség

## A MOS tranzisztor

### ■ A működés elve:

- Egy félvezető alapon („szubsztrát”) létrehozunk két elektródát, amely között folyó áramot szabályozzuk
- A két elektróda neve „**source**” és „**drain**”
- A **source**-ból származnak a töltéshordozók, amelyeket a **drain** gyűjt össze.
- A szabályozó elektróda neve: **gate**
- Mintha egy szívószálon keresztül folyó folyadékot vezérelnénk nyomással.





- Mivel kétfajta adalékolás is létezik, kétféle kialakítás képzelhető el.
- nMOS ill. pMOS tranzisztorokról beszélünk

## A MOS tranzisztor vázlatos működése



### ▪ Tekintsük az nMOS tranzisztort!

- Alapesetben nem vezet, hiszen a source és a drain között lezárt pn átmenetek vannak.
- Ha a gate feszültsége egy bizonyos szintet meghalad, a gate alatt elektronok jelennek meg és kialakul egy csatorna, ami összeköti source-drain elektródákat.
- A jelenség neve **INVERZIÓ** ehhez szükséges feszültség a **küszöbfeszültség**.
  - (inverzió: hiába p vezetési típusú a félvezető, mégis a nagy térerősség hatására a felületen az elektronok kerülnek többségbe, mintha n típusú lenne)
- **PONTOSAN ÚGY MŰKÖDIK, MINT EGY RELÉ – de nincs mozgó alkatrész!**
- pMOS esetén fordítva

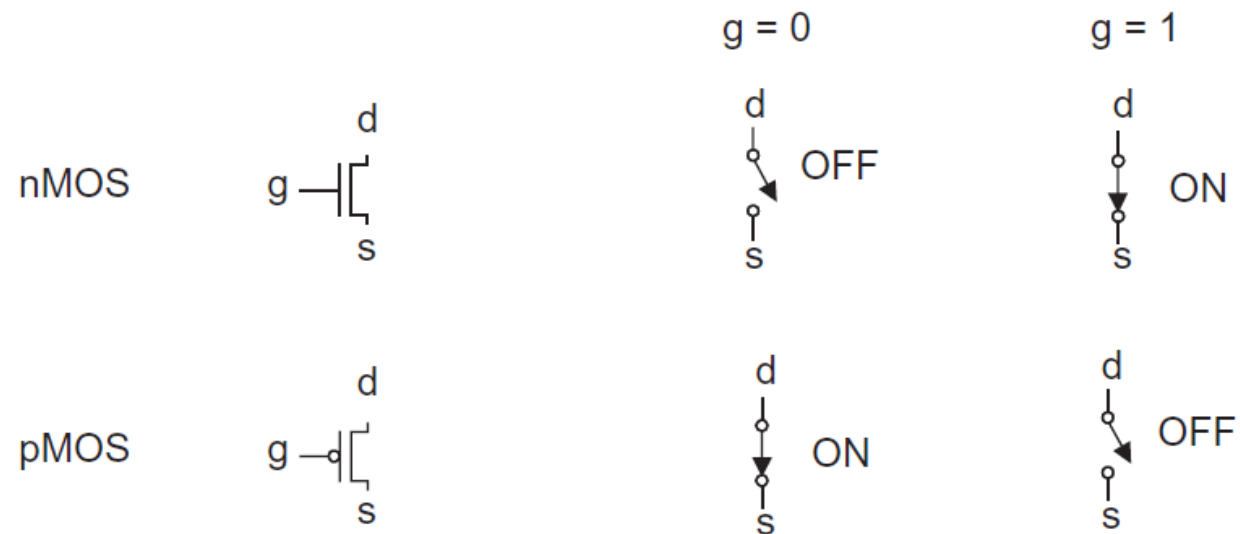
## Mire jó ez az egész?

### ■ KAPCSOLNI!

- Emlékezzünk vissza: a kételemű Boole algebra egy másik elnevezése az ún. kapcsoló algebra.
- A MOS tranzisztor egy nem teljesen ideális, de jól működő kapcsoló.
- Rögtön két változatban is rendelkezésre áll.

### ■ Vizsgáljuk meg, hogyan működik egy digitális rendszerben

- Reprezentáljuk az logikai igazat a tápfeszültséggel, a logikai hamist pedig a 0V-al.
- Feltételezzük, hogy a tápfeszültség nagyobb, mint a küszöbfeszültség.
  - valójában – ökölszabály – kb. a négyszerese.
- Feltételezzük, hogy az nMOS tranzisztor szubsztrátja földre, a pMOS tranzisztor szubsztrátja pedig a tápfeszültségre van kötve
- Nézzük meg, mi történik a kétfajta tranzisztorttal!



- A nMOS logikai 0 esetén nyitott kapcsoló, nem vezet áramot, logikai 1 esetén zárt kapcsoló, vezet.
- A pMOS logikai 0 esetén vezet, logikai 1 esetén nem vezet
- **Nem véletlen a jelölés**
  - Figyeljük meg az invertálás kis karikáját a pMOS tranzisztor gate-jén!
  - (ebben a tárgyban igyekszünk ezt az ún. „digitális” MOS szimbólumot alkalmazni, így „ránézésre” látszik, hogy melyik logikai jelszintnél mi történik)



# A CMOS

- Minden adott tehát, hogy logikai kapukat készítsünk.
- Van kétfajta tranzisztorunk, az egyik logikai magas szintű vezérlésre kapcsol, a másik logikai alacsony szintűre.
  - Ha ezeket egy áramkörben alkalmazzuk, ezek lesznek az ún. komplementer MOS áramkörök, rövidítve CMOS
- Ha ügyesen kombináljuk, tudunk készíteni
  - Invertert
  - Alapkaput
  - (itt már megállhatnánk, hiszen pl. kétbemenetű NAND kapuból minden logikai függvény megépíthető)
  - Bonyolultabb logikai függvényt is. (majd látni fogjuk, ez lesz egy hatalmas előnye a CMOS-nak!)

## „nem teljesen ideális, de jól működő kapcsoló”

### ■ Mit jelent ez?

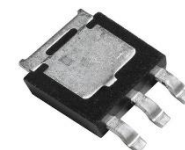
- Bekapcsolt állapotban is esik rajta feszültség, oka az ún. csatornaellenállás.
- Azaz fogyaszt ( $P = I^2 R_{DSon}$ )
- Csak addig vezet, amíg a gate-source feszültségkülönbség nagyobb, mint a küszöbfeszültség. Azaz digitális logikában gondolkodva az nMOS nem tud tápfeszültségig feltölteni, a pMOS pedig nem tud 0V-ig kisütni. Többek között ezért alkalmazzuk együtt.

### ■ Mire jó még?

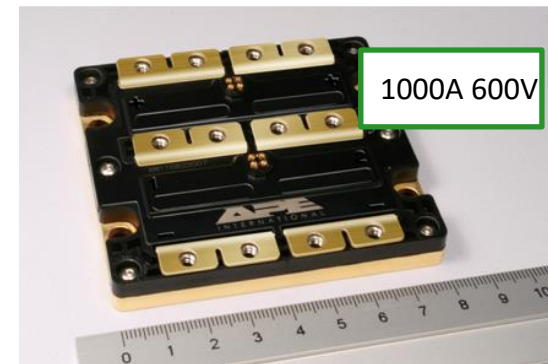
### ■ Mindenre (is)

- Bármit kapcsolni. A nA-tól a kA-ig 😊. Persze a méret más lesz.
- Inverterek, elektromos motorok meghajtása, elektromos autó, DC-DC átalakítók stb. stb. stb.
- D osztályú erősítő
- Analóg áramkörök is készülnek MOS tranzisztorral. Más tranzisztorok általában jobbak, de az hatalmas előny, hogy egybeintegrálható a digitális logikával.

100A 40V



1000A 600V





Budapesti Műszaki és Gazdaságtudományi Egyetem  
Elektronikus Eszközök Tanszéke

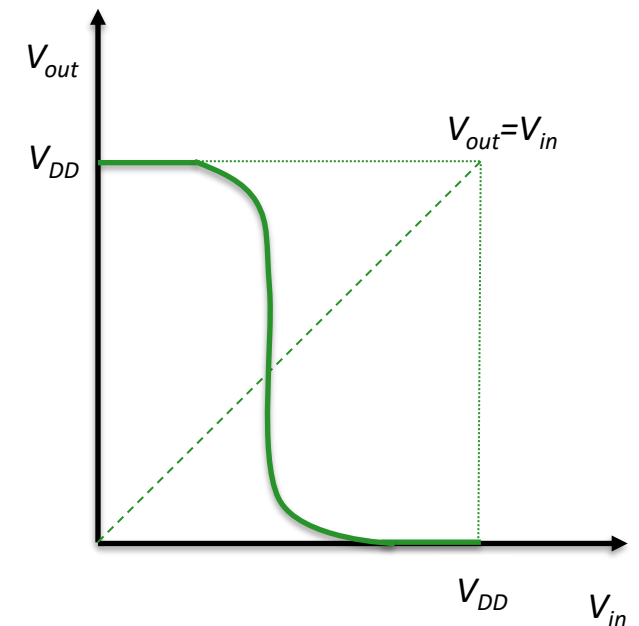
# A digitális logika alapfogalmai

# A Boole algebra áramköri megvalósítása

- A Boole algebra
  - Matematikai szemszögből egy absztrakció
    - Értékkészlete  $x \in \{0,1\}$
    - Műveletei:  $\neg \wedge \vee$
- Az értékkészlet elemeihez valamilyen könnyen feldolgozható fizikai mennyiséget rendelünk, ami általában a feszültség.
  - (lehet áram vagy feszültségkülönbség – nagysebességű logikákban)
  - A logikai 1-hez a  $V_H$ , a logikai 0-hoz a  $V_L$  feszültség szintet rendeljük.
  - Az eseten nagy részében a  $V_H$  megegyezik a tápfeszültséggel, a  $V_L$  pedig a föld.
  - A  $V_H - V_L$  mennyiséget hívjuk **swing**-nek.
  - A tápfeszültség ( $V_{DD}$ ) és a föld (0V) különbséget pedig **RAIL**-nek.
    - (honnan jön a név? – a tápfeszültség és a föld széles vezetői, mint egy sínpár húzódnak végig – a kapcsolási rajzon is és a fizikai valóságban is...)

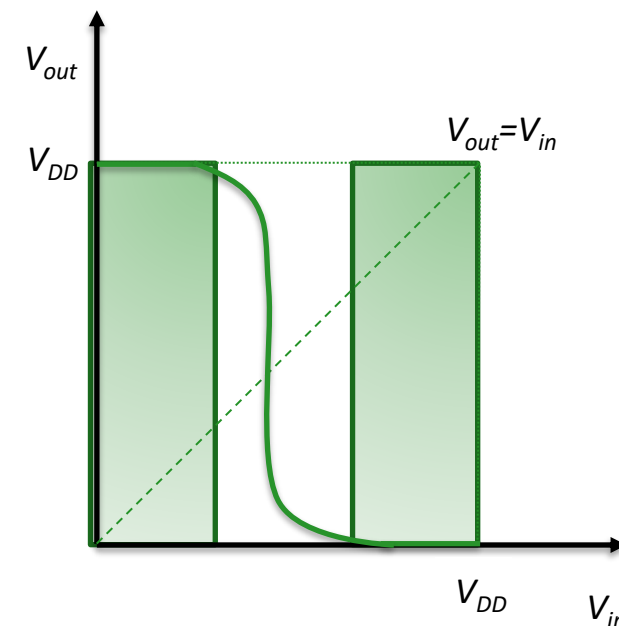
## A transzfer karakterisztika

- Az inverter az  $V_L=f(V_H)$  ill  $V_H=f(V_L)$  függvényt kell, hogy megvalósítsa.
- Ez az ún. transzfer karakterisztika
- A komparálási feszültség az a feszültség, ami felett logikai 1, ami alatt pedig logikai 0-nak tekintünk egy jelet.
  - Komparálási feszültségen:  $V_{OUT} = V_{IN}$
  - Azaz a transzfer karakterisztika és a  $45^\circ$ -os egyenes metszéspontja.



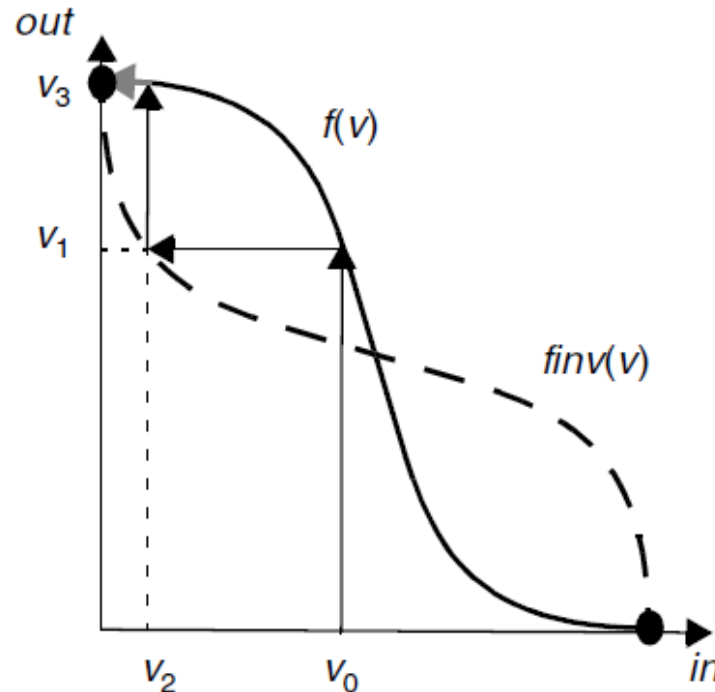
## Zaj/zavar védettség

- Széles bemeneti feszültség tartományhoz azonos kimeneti érték tartozik.
  - Azaz ha a zaj/zavar ennél kisebb, a kimeneten nem jelenik meg, **elnyomja!**
- A karakterisztika három szakaszból áll:
  - A két szélső szakasz laposan fut, azaz a bemeneten lévő feszültségváltozások csak nagyon kis változást okoznak a kimeneten.
    - Ez lehetőséget teremt arra, hogy biztonságos logikai szint tartományokat jelöljünk ki.
  - A középső szakasz meredek.
    - (nagy az erősítése, azaz  $A = \left| \frac{dV_{OUT}}{dV_{IN}} \right|$  nagy)
    - kis bemeneti megváltozásra nagy kimeneti feszültségváltozás történik



## Jel-regeneráció

- A digitális feldolgozás során a logikai jel szintje regenerálódik



- Ha egy ábrán ábrázoljuk a  $f(V)$  transzfer karakterisztikát és  $f_{inv}(V)$  inverzét, nagyon szemléletesen belátható.
- Így lesz egy „rossz”  $V_0$  jelből egyre jobb  $V_1, V_2, V_3$  feszültség.
- Így a digitális logikai kapunak **stabil** két egyensúlyi helyzete van

# Robosztusság

- A digitális logikai áramkör **ROBOSZTUS**
- A működés lényege kevésbé érzékeny (természetesen a megadott abszolút határok között) többek között
  - A bemeneten lévő zaj-zavarra
  - A tápfeszültség megváltozására
  - A környezeti hőmérsékletre
  - Az egyes alkatrészek paramétereinek véletlenszerű megváltozásának
  - Stb.





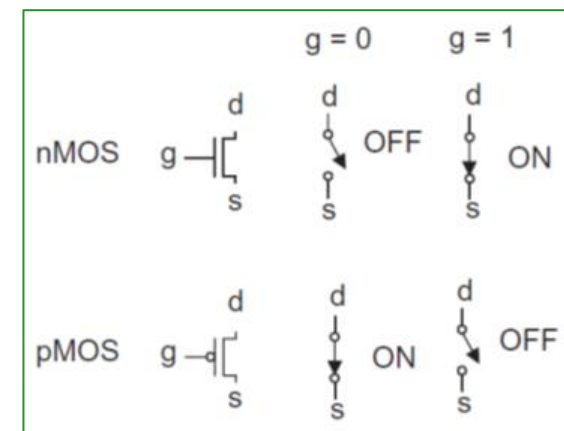
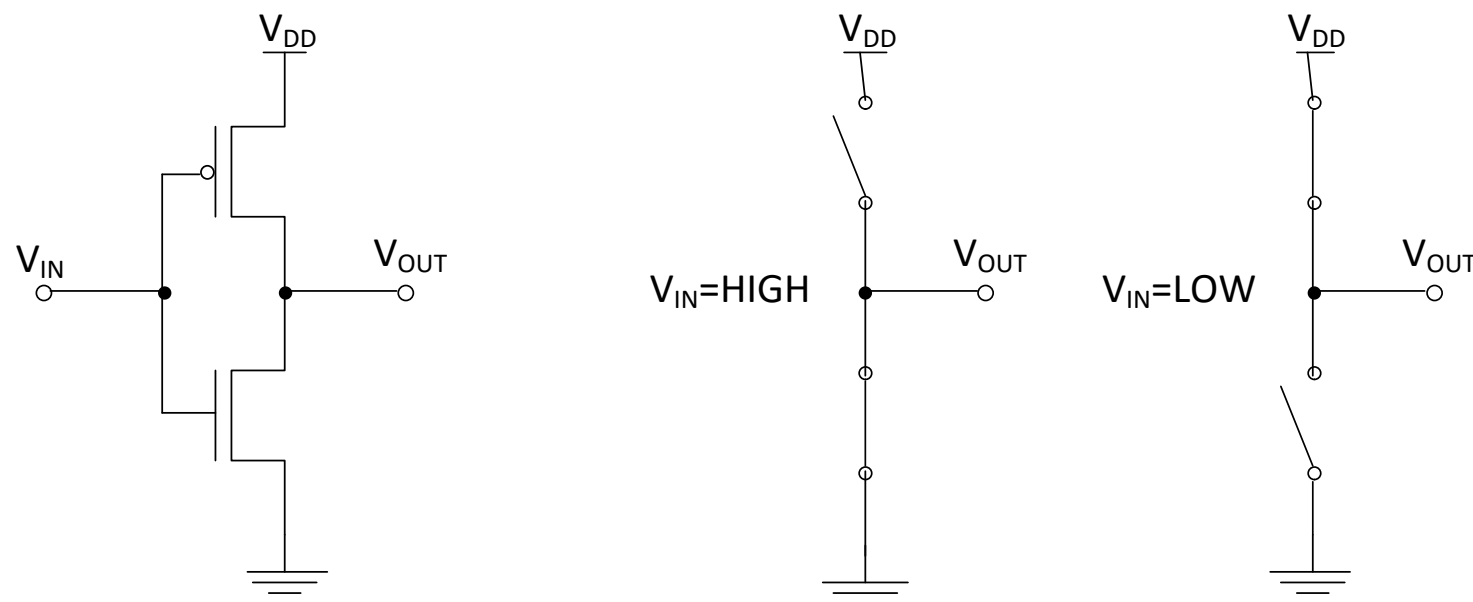
Budapesti Műszaki és Gazdaságtudományi Egyetem  
Elektronikus Eszközök Tanszéke

# CMOS logikai áramkörök

# CMOS áramkörök

- Complementary MOS
  - n és p csatornás tranzisztorokból állnak a kapuk, innen származik a név
- Manapság egyeduralkodó logikai áramkörökben
  - A logikai szintek „tiszták”:  $V_H = V_{DD}$ ,  $V_L = 0V$  (azaz **rail-to-rail**, 0-tól tápfeszültségig történik a működés)
  - A statikus (állandósult állapotbeli) áramfelvétel alacsony.
  - A jelváltozás esetén a fel és lekapcsolási késleltetések (nagyjából) megegyeznek. (szimmetrikus működésre méretezett kapuk esetén.)
  - Tápfeszültség-érzékeny, széles tápfeszültség tartományokban működik.
  - Jól integrálható, a kapuk egyszerűek. (kevés tranzisztort igényelnek)
    - Mit jelent ez?
    - Egységnyi területre jóval több kapu fér, mint más technológiákban
    - Nem véletlen, hogy – a nagyon speciális, pl. nagyfrekvenciás területek kivételével minden más logikai áramkört kiszorított.

## A CMOS inverter kapcsolási rajza



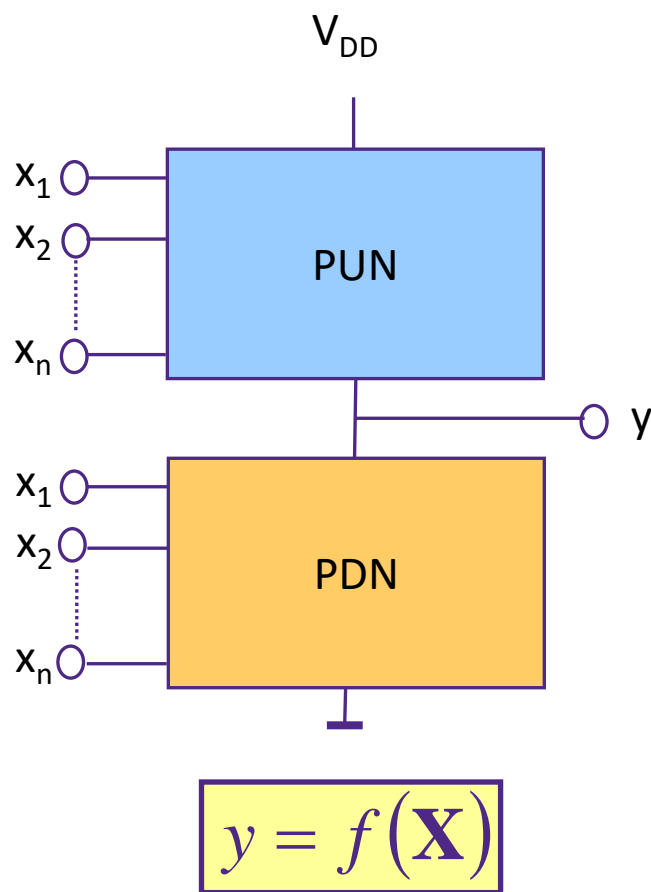
- Egy n és egy p csatornás MOS tranzisztorból áll.
  - Állandósult állapotban a két tranzisztor közül csak az egyik vezet, a másik mindig lezár.
- Azaz, mint egy olyan kapcsoló, ami a kimenetre a bemeneti jel szintjétől függően vagy a tápfeszültséget, vagy a földet kapcsolja.



Budapesti Műszaki és Gazdaságtudományi Egyetem  
Elektronikus Eszközök Tanszéke

# Statikus CMOS alapkapuk

# CMOS kapu felépítése



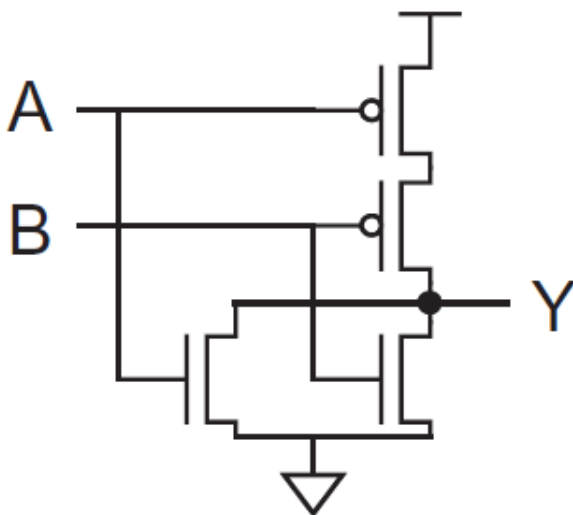
- A kapuk esetében egy p csatornás tranzisztorokból álló „pull up” (PUN) ill. n csatornás tranzisztorokból álló „pull down” hálózat (PDN) alkotja a kaput, mindkét hálózat annyi tranzisztorból áll, ahány bemenete van a függvénynek
- pull up network
  - p-csatornás tranzisztorok
  - rövidzár, ha  $f(\mathbf{X})=1$
  - szakadás, ha  $f(\mathbf{X})=0$
- pull down network
  - n-csatornás tranzisztorok
  - rövidzár, ha  $f(\mathbf{X})=0$
  - szakadás, ha  $f(\mathbf{X})=1$

## CMOS kapu felépítése

- Kapcsoló jellegű működés:
  - ÉS jellegű kapcsolatot két tranzisztor(vagy hálózatrész) soros kapcsolása ad.
  - VAGY jellegű kapcsolatot pedig két tranzisztor (vagy hálózatrész) párhuzamos kapcsolása ad.
- A PUN és a PDN működése ellentétes. Amikor az egyik vezet, a másiknak szakadást kell adnia és fordítva
  - Reciprok hálózat
  - (Ez bonyolultan hangzik. Arról van szó, hogy amit az egyik hálózatban sorba kapcsoltunk, a reciprok hálózatban párhuzamosan kell kapcsolni és fordítva. Ld. a következő dián)

## CMOS NOR kapu

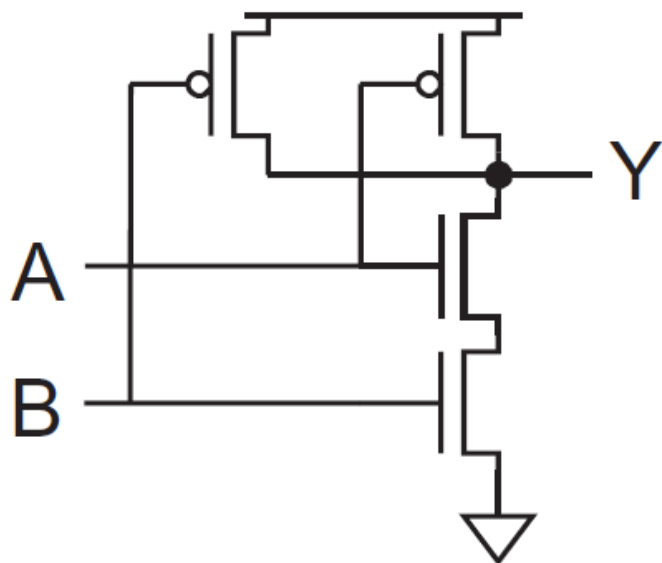
- A pull down network két párhuzamosan kötött nMOS tranzisztorból áll, a pull up pedig két sorbakapcsolt pMOS-ból.
- Ha mindkét bemenet 0, akkor mindkét pMOS vezet és egyik nMOS sem vezet. Így a kimenet 1.
- Ha A vagy B bemenet igaz, valamelyik nMOS tranzisztor vezet, a hozzátartozó pMOS viszont lezár, így a kimenet 0 lesz.



$$Y = \overline{A + B} = \bar{A} \cdot \bar{B}$$

## CMOS NAND kapu

- A pull-down network most két, sorba kötött nMOS tranzisztorból áll, a pull-up pedig két párhuzamosan kötött pMOS-ból.



$$Y = \overline{AB} = \bar{A} + \bar{B}$$



## Komplex kapuk

- Tranzisztor szinten tudunk bonyolultabb (nem alapvető) logikai függvényeket megvalósítani.
- Általában maximum 4 bemenettel, az *és* ill. *vagy* függvényeket kombináljuk.
- Például:
  - OAI21
    - $Y = \overline{(A + B)C}$
  - AOI22
    - $Y = \overline{AB + CD}$
- Általában  $n$  bemenet *és/vagy* kombinációja  $2n$  tranzisztor segítségével megvalósítható.
  - Ha feltételezzük, hogy a bemenet pozitív és negatív változata is rendelkezésre áll
- Mivel egy kapu, a késleltetés a többszintű realizációhoz képest kedvezőbb.
- A többszintű realizációhoz képest kevesebb tranzisztort tartalmaz.

## Példa

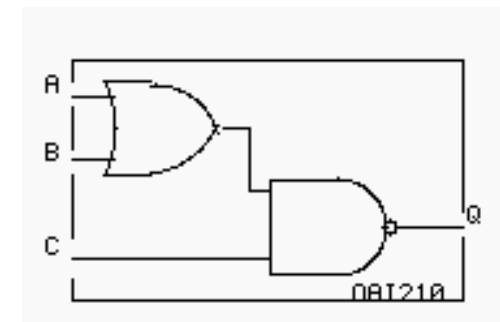
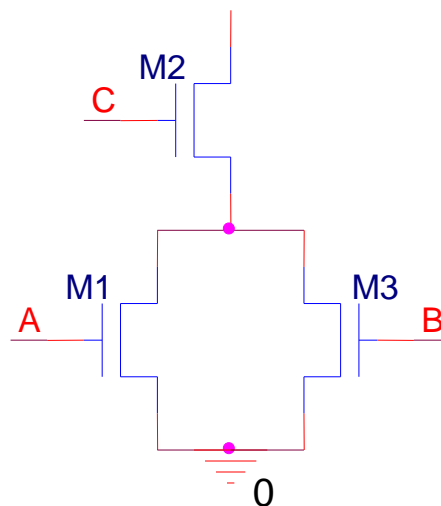
- Tervezzük meg az  $Y = \overline{(A + B)C}$  függvényt megvalósító komplex kaput!

- Első lépés a PDN (pull-down network) megtervezése, ezt n csatornás tranzisztorokkal kell megvalósítani.

A negált logikai függvény minden 0 értékéhez

a kimenet és föld között áramutat kell biztosítani, a függvényben szereplő összegnek párhuzamosan, a szorzatoknak sorba kapcsolt hálózatrészek felelnek meg.

- Itt tartunk:



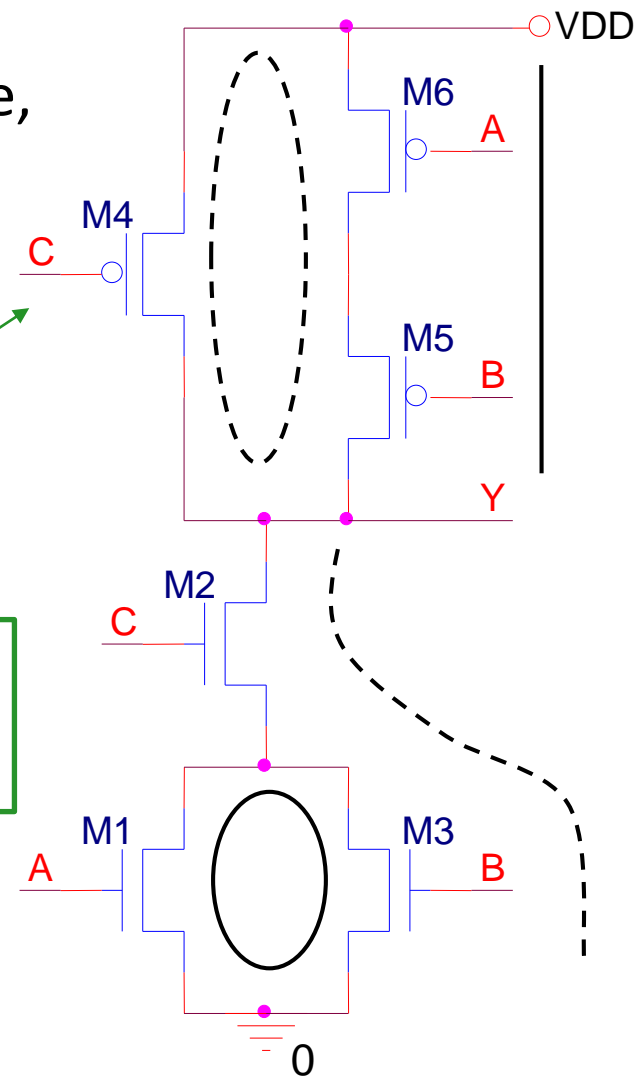
## Példa, folytatás

- A második lépés PUN (pull-up network) megtervezése, ezt a p vezetékes tranzisztorokkal kell megvalósítani.
  - A pull-up network a negált függvény minden 1 értékéhez a tápfeszültség és a föld között áramutat kell, hogy biztosítson.

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



Vigyázat! A PUN is ugyanazt a vezérlést kapja, tehát semmiképpen sem negáltat! (a pMOS logikai 0-nál vezet!)



## Összehasonlítás

### ■ A kétszintű változat

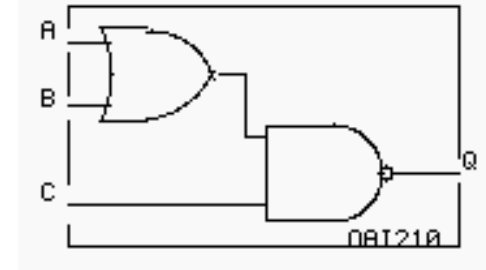
- OR nincs, tehát NOR + inverter és egy NAND kapu
- Összesen  $4 + 2 + 4 = 10$  tranzisztor
- Az (intrinsic) késleltetés A->Q ill. B->Q irányban kb. háromszorosa a C->Q iránynak

### ■ Ha nem foglalkozunk az invertálással

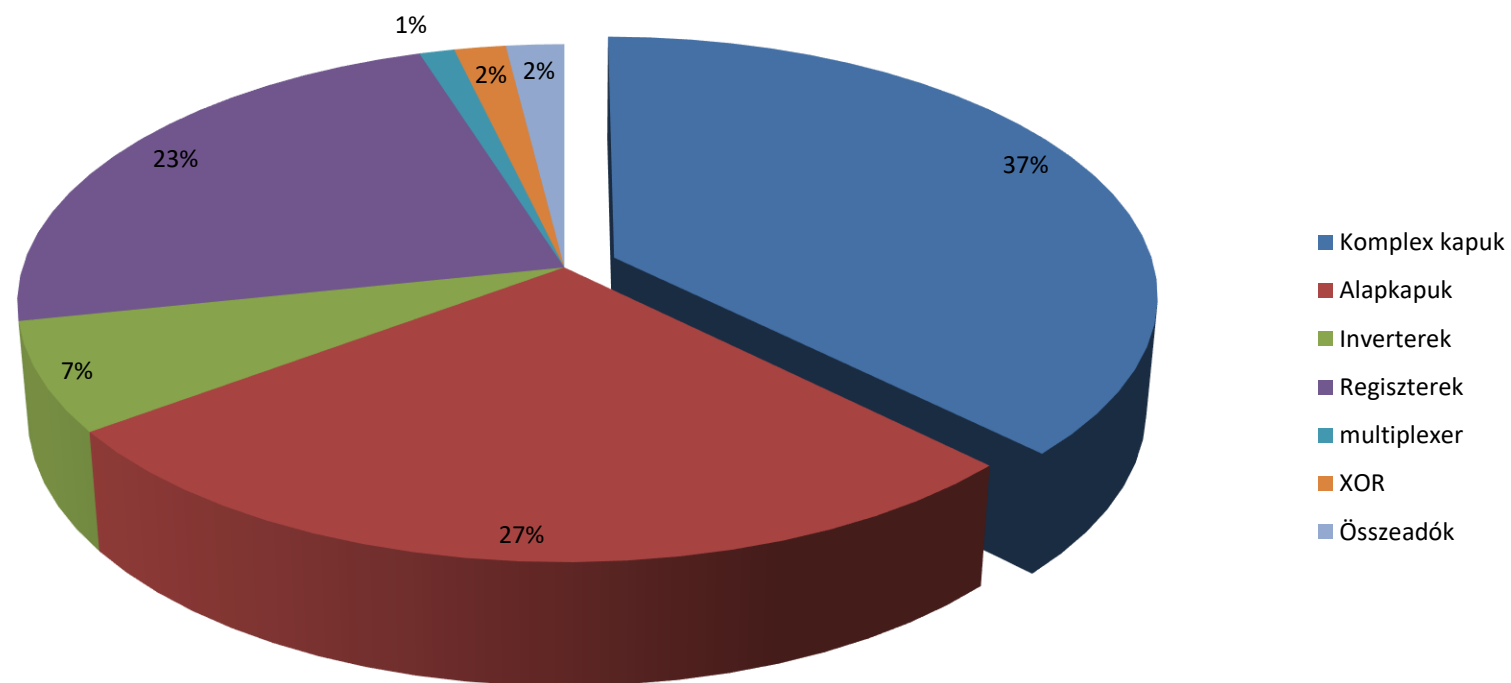
- $Q = \overline{(A + B)C} = \overline{A + B} + \overline{C}$
- Két darab kétbemenetű NOR kapu, 8 tranzisztor
- A késleltetés A->Q ill. B->Q irányban kb. kétszerese a C->Q iránynak

### ■ Komplex kapu

- 6 tranzisztor
- A késleltetések közel egyformák (ha jól van méretezve a kapu)



## 6502 processzor mag, kb. 12000 tranzisztor



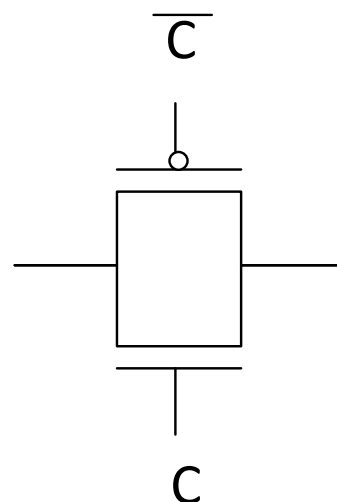


Budapesti Műszaki és Gazdaságtudományi Egyetem  
Elektronikus Eszközök Tanszéke

# A CMOS transzfer kapu

## CMOS tranzfer kapu

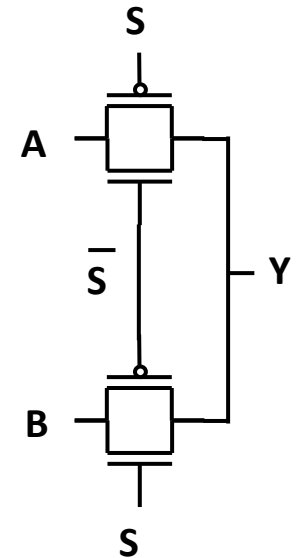
- A jelfolyam útjába helyezett kapcsoló.
- CMOS kivitelben egy n és egy p típusú tranzisztort kapcsolnak össze, a vezérlő jelek egymás inverzei.
  - A tranzisztorról eddig tanultak alapján erre nem lenne szükség, de a valós tranzisztor karakterisztikái miatt ezt így kell csinálni
- Tranzfer kapuk alkalmazásával tovább egyszerűsíthetők az áramkörök.



C=0 -> szakadás  
C=1 -> rövidzár

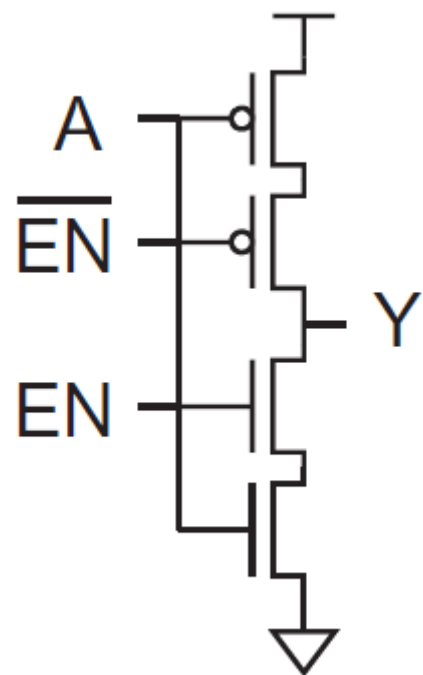
## Tranzfer kapuk alkalmazása

- Bizonyos funkciók tranzfer kapuk alkalmazásával jóval egyszerűbben, kevesebb tranzisztortal valósíthatóak meg.
- Tipikusan a kiválasztó jellegű funkciók ilyenek.
  - Sok függvény ide vezet.
  - Programozható logikák egy részében a kiválasztás így történik. (**ld. FPGA**)
- Tekintsünk például egy kétbemenetű multiplexert!
  - $Y = A\bar{S} + BS$
- Komplex kapuval 8 tranzistor szükséges, a tranzfer kapus pedig csak 4 tranzisztorból áll!





## Órajel vezérelt CMOS (Clocked CMOS, C<sup>2</sup>MOS)



- Háromállapotú (tri-state kapu)
  - Buszok meghajtására
- EN=0 – a kimenet lebeg
- EN=1 – a kimenet a bemenet inverze.

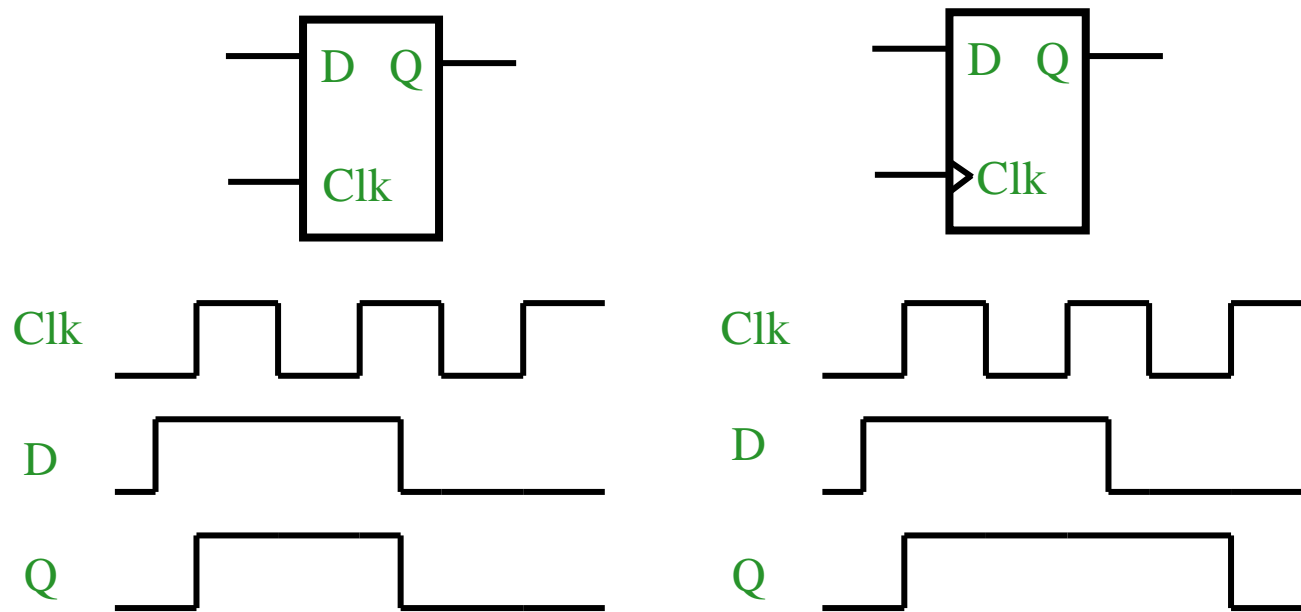
## CMOS tárolók

Latch

Flip-flop / regiszter

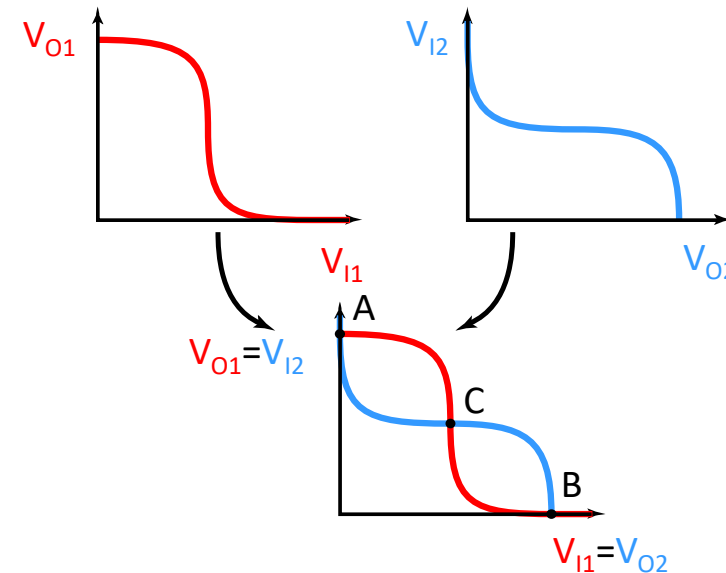
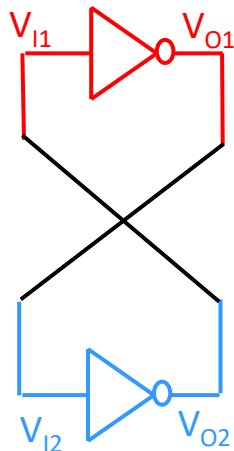
## Latch és flip-flop

- Sok helyen ekvivalens a két fogalom. Ebben az anyagban így használjuk:
  - latch: az engedélyezett latch átlátszó, a bemeneti változás (késleltetés után a kimenetre jut), tehát **SZINTVEZÉRELT**
  - flip-flop: a beírás az órajel fel vagy lefutó élére történik, tehát **ÉLVEZÉRELT**



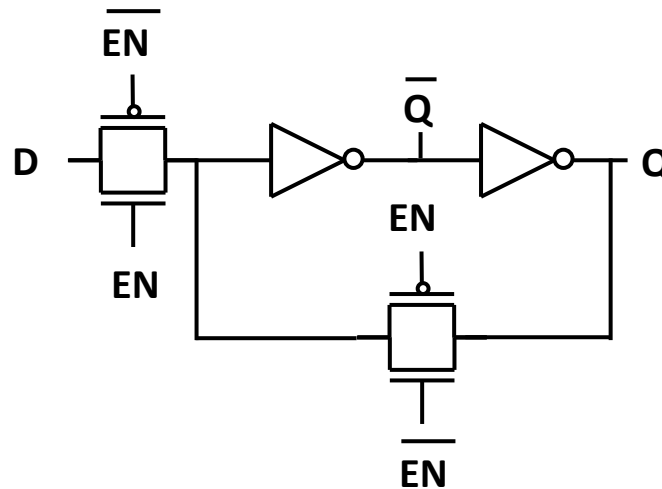
# A tárolás alapelve

- Két állapotú (bistabil) áramkörök alapja: két, gyűrűbe kapcsolt inverter
  - Ennek az elrendezésnek két stabil állapota van (A, B)
    - (ezen kívül egy metastabil, (C) ekkor minden feszültség a komparálási feszültség, de az inverter karakterisztika gondoskodik róla, hogy ebben az állapotban ne maradjon meg – ezt csak szimulátorban lehet előállítani.)
- Ahhoz, hogy tárolóként lehessen használni, írhatóvá kell tenni.
  - Az invertálási funkciót megtartva az inverterek helyett negált kimenetű logikai kapukat használunk, amelyeknek a további bemenetei lehetővé teszik a beállítást.



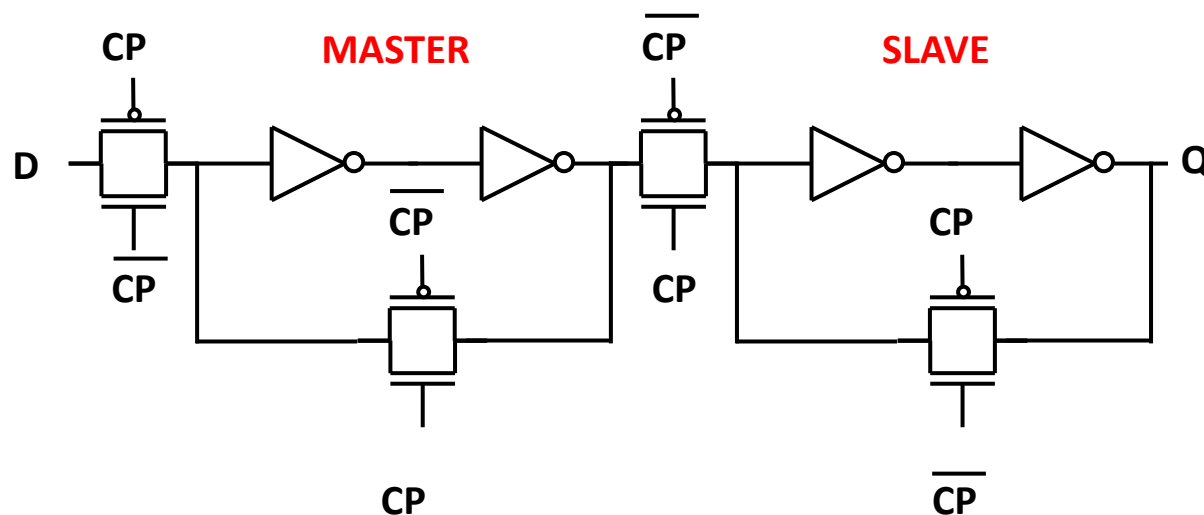
## D-latch transzfer kapuval

- A beírás és a visszacsatoló ág egy-egy transzfer kapuval van ellenfázisban vezérelve.
  - EN=1 alatt transzparens működés,  $Q = D$ , mivel a beíró ág transzfer kapuja vezet, a visszacsatoló ágban elhelyezett transzfer gate viszont zárt.
  - EN=0 alatt a kimenet visszaíródik a bemenetre, a transzfer kapuk most ellentétesen vezetnek: a visszacsatoló ág „él”, a beíró ág elzárt.
  - Összesen nyolc tranzisztorral megvalósítható és nincs szükség a D negáltjára.
  - Figyeljük meg, hogy a bemenetről „átengedő” és a visszacsatoló transzfer kapu **ellenfázisban** működik!



## D-flipflop

- A master-slave flip-flop két sorbakötött, ellenütemű órajellel vezérelt latch
  - CP alacsony szintjén az első tároló átlátszó.
  - CP felfutó élére a master nemátlátszó lesz és a tartalom a slave-be íródik



## Források, ajánlott irodalom, érdekességek

- Digitális alapáramkörök szimulációja
- Komplex kapu szimulációja: AOI21, OAI21
- Multiplexerek
- Latch szimulációja