

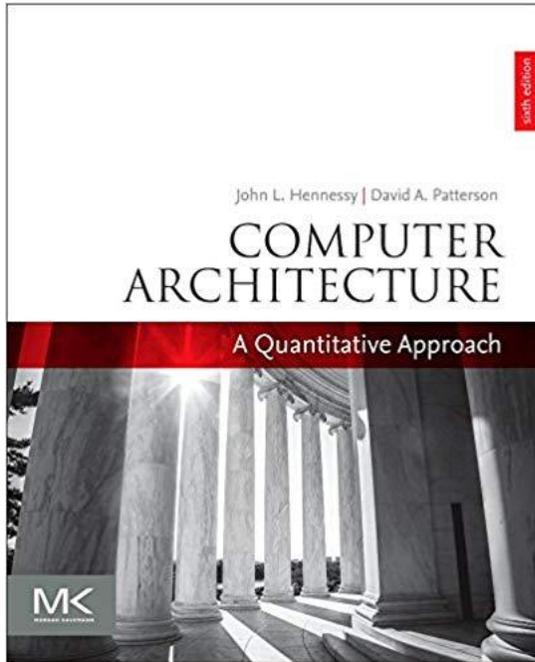
Structura și Organizarea Calculatoarelor

– Cursul 1 –

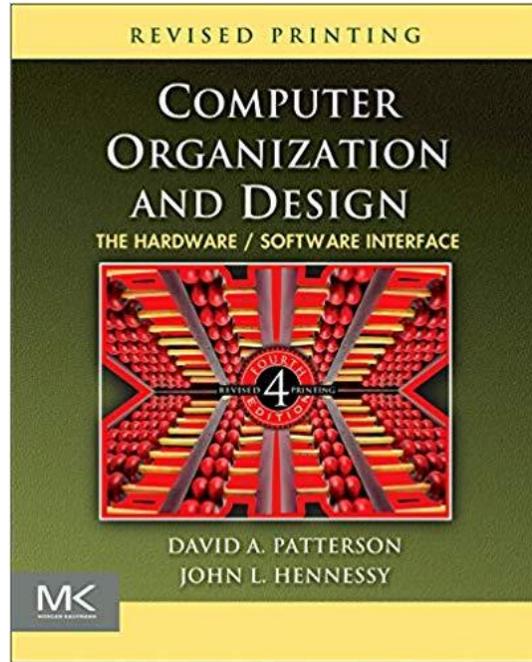
De la porți logice la procesoare

Facultatea de Automatică și Calculatoare
Politehnica București

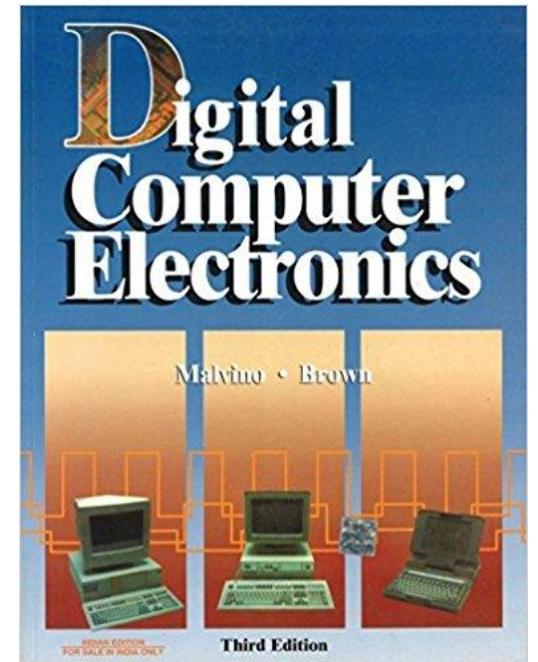
The Books



[Computer Architecture: A Quantitative Approach](#)



[Computer Organization and Design: The Hardware / Software Interface](#)



[Digital Computer Electronics](#)

Cum construiesc un calculator?

Building a Computer



What regular people think we do



What our family thinks we do



What the industry thinks we do



What society thinks we do



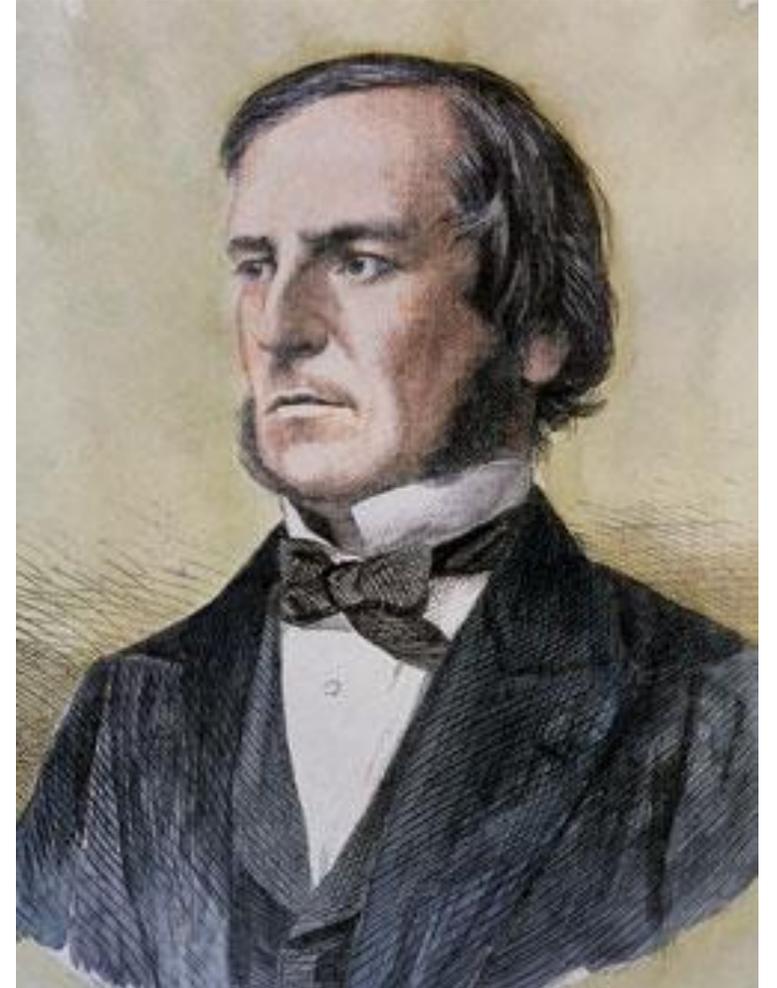
What we think we do



What we really do

George Boole, 1815-1864

- Matematician, filozof și logician englez
- “The Laws of Thought” (1854) – algebra booleană
- Părintele erei informației



Manipularea expresiilor logice

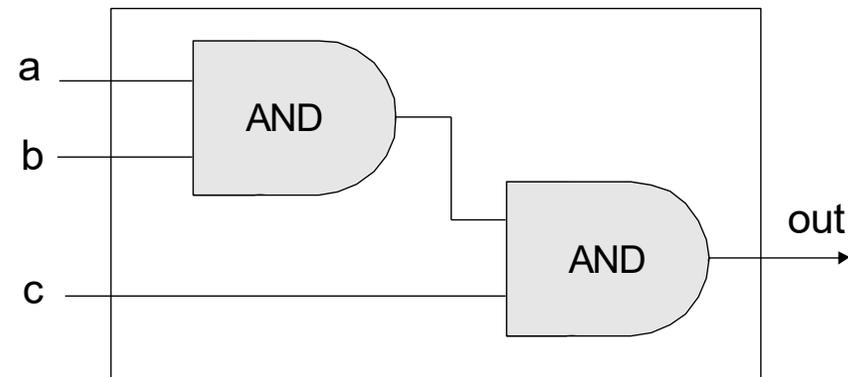
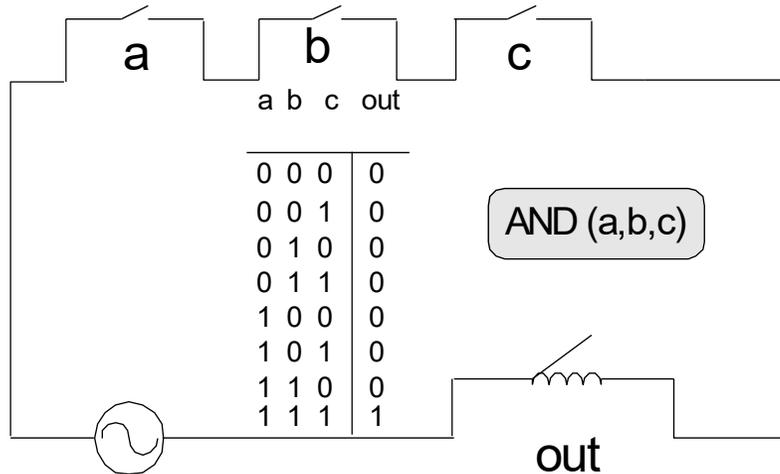
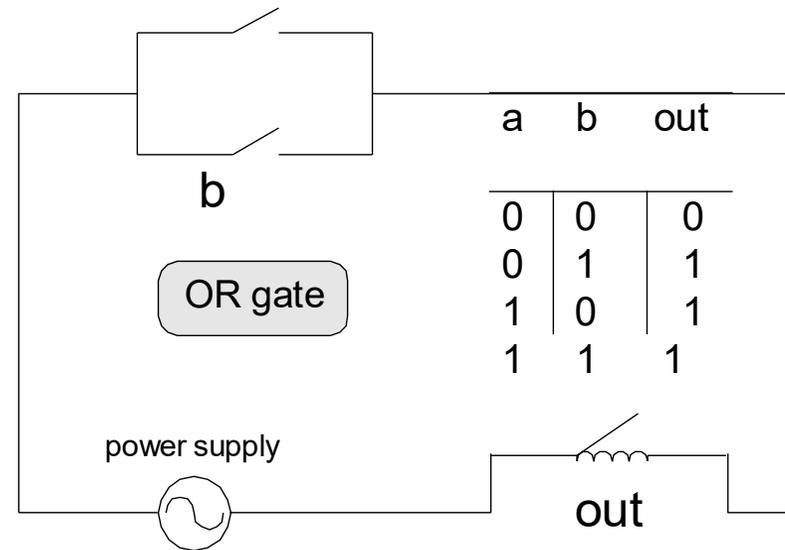
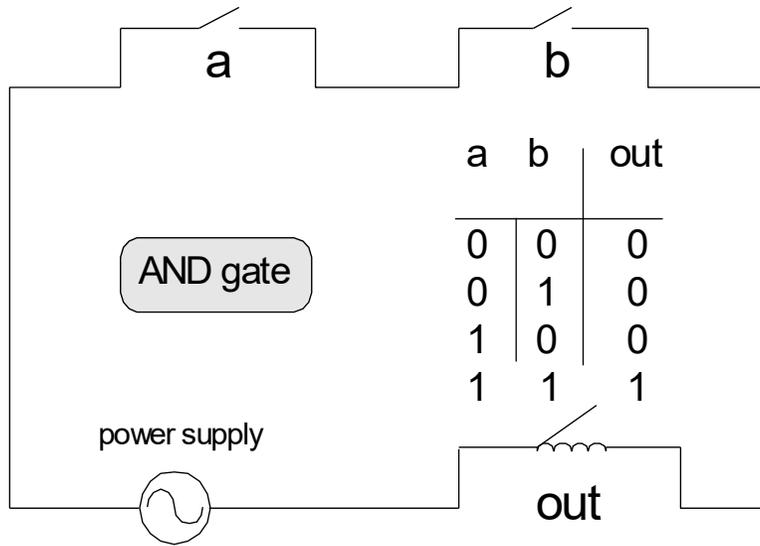
Numele legii	Forma SAU	Forma ȘI
Identitate	$x \vee 0 = x$	$x \wedge 1 = x$
Unu/Zero	$x \vee 1 = 1$	$x \wedge 0 = 0$
Idempotență	$x \vee x = x$	$x \wedge x = x$
Negare	$x \vee x' = 1$	$x \wedge x' = 0$
Comutativitate	$x \vee y = y \vee x$	$x \wedge y = y \wedge x$
Asociativitate	$(x \vee y) \vee z = x \vee (y \vee z)$	$(x \wedge y) \wedge z = x \wedge (y \wedge z)$
Distributivitate	$x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$	$x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$
DeMorgan	$(x \vee y)' = x' \wedge y'$	$(x \wedge y)' = x' \vee y'$

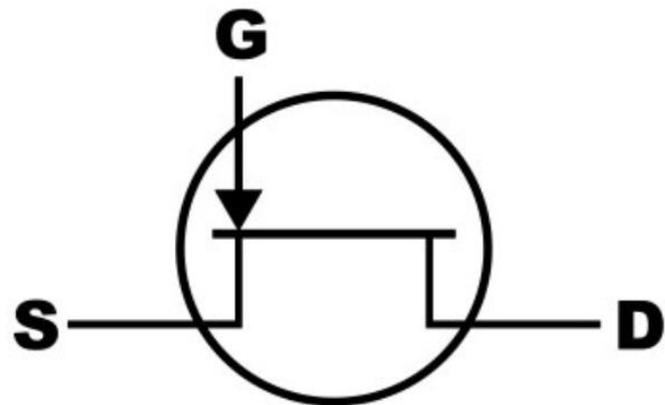
Claude Shannon, 1916-2001

- Matematician și inginer american
- "A Symbolic Analysis of Relay and Switching Circuits" (1938) – teză de master la MIT, aplică logica booleană circuitelor de calcul
- "A Mathematical Theory of Communication" (1948) – pune bazele teoriei informației

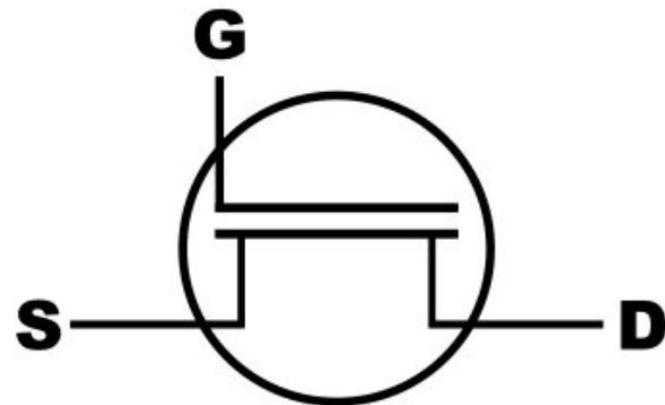


Porți logice cu rele (Shannon)

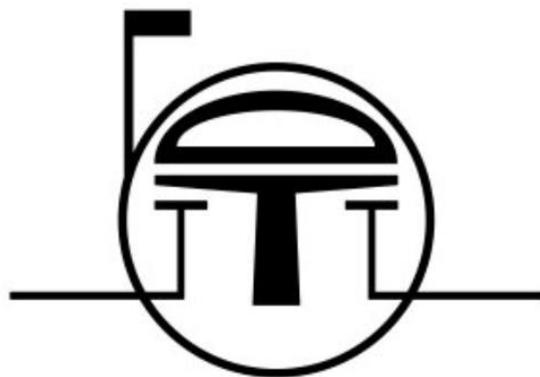




JFET



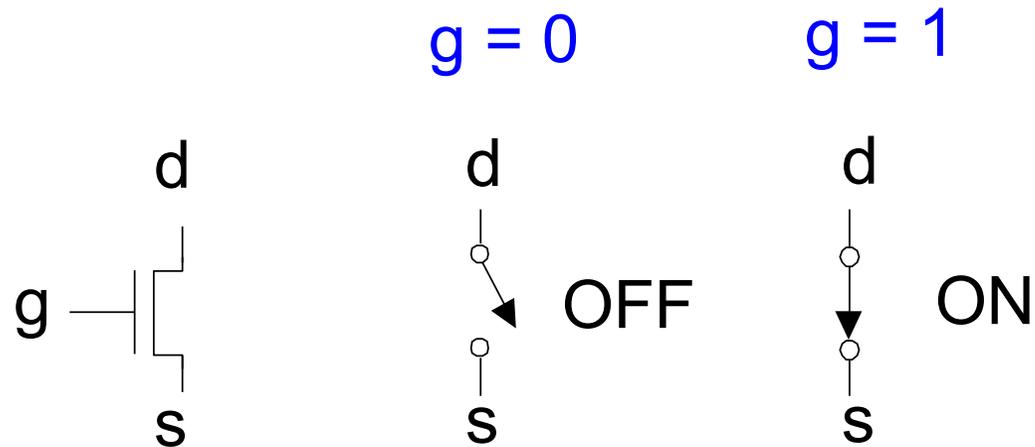
MOSFET



BOBA FETT

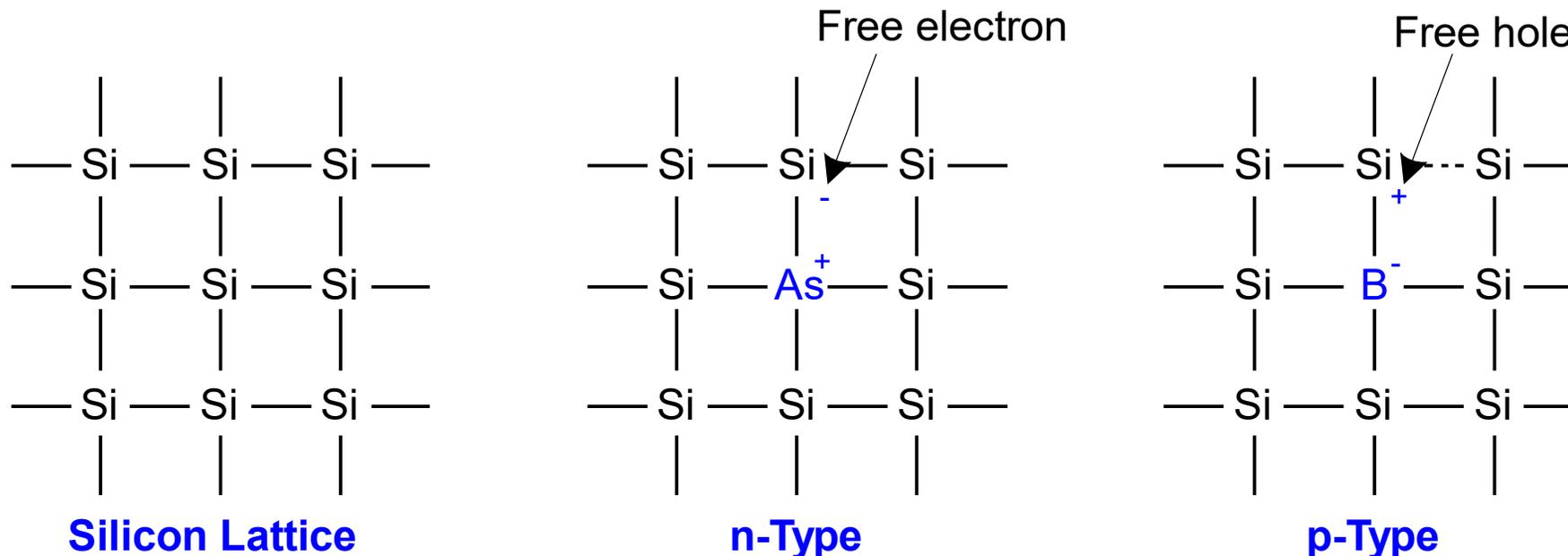
Tranzistoare

- Toate porțile logice moderne sunt construite cu tranzistoare
- Comutator controlat în tensiune cu trei porturi
 - 2 sunt conectate împreună în funcție de tensiunea aplicată pe al treilea port
 - d și s sunt conectate (ON) dacă g este 1 logic



Siliciul

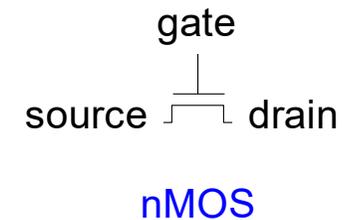
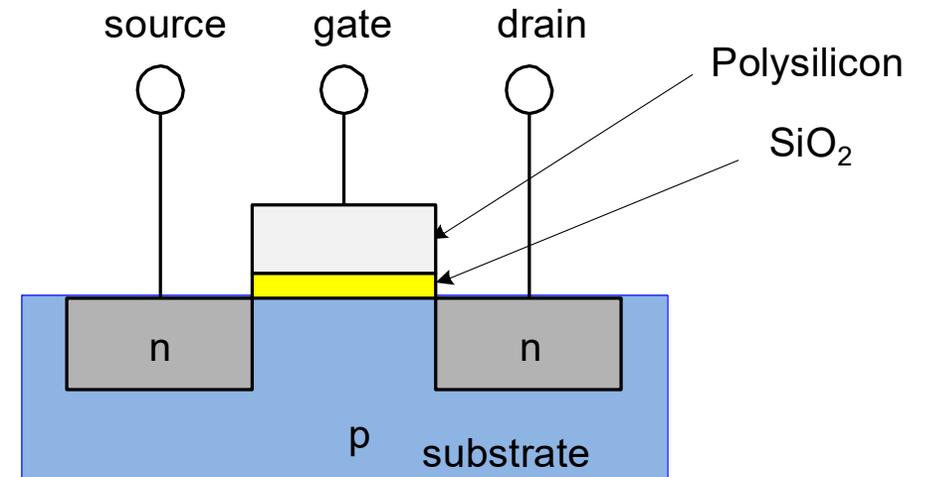
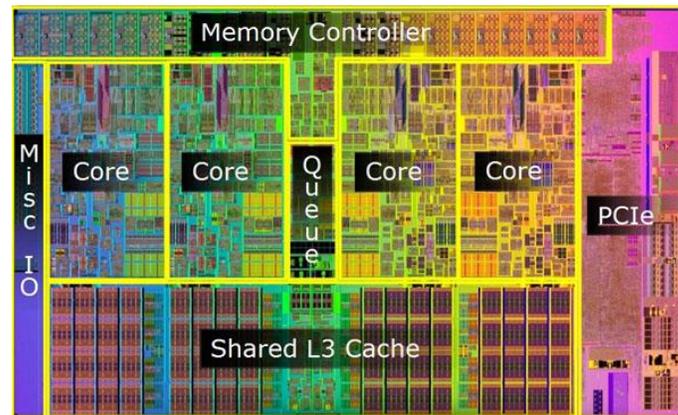
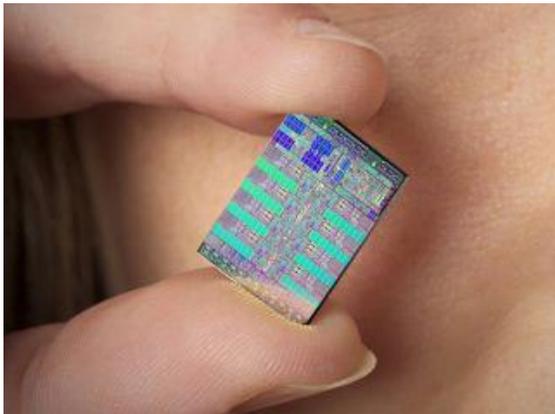
- Tranzistoare construite din siliciu, un semiconductor
- Siliciul pur nu este un conductor foarte bun (nu are sarcini libere)
- Siliciul dopat este un bun conductor (sarcini libere)
 - Tip-n (sarcini *negative* libere, electroni)
 - Tip-p (sarcini *pozitive* libere, goluri)



Tranzistoare MOS

- **Metal oxide silicon (MOS):**
 - Poartă din polisiliciu (sau **Metal**)
 - Izolator din **Oxid** (dioxid de siliciu)
 - **Siliciu** dopat

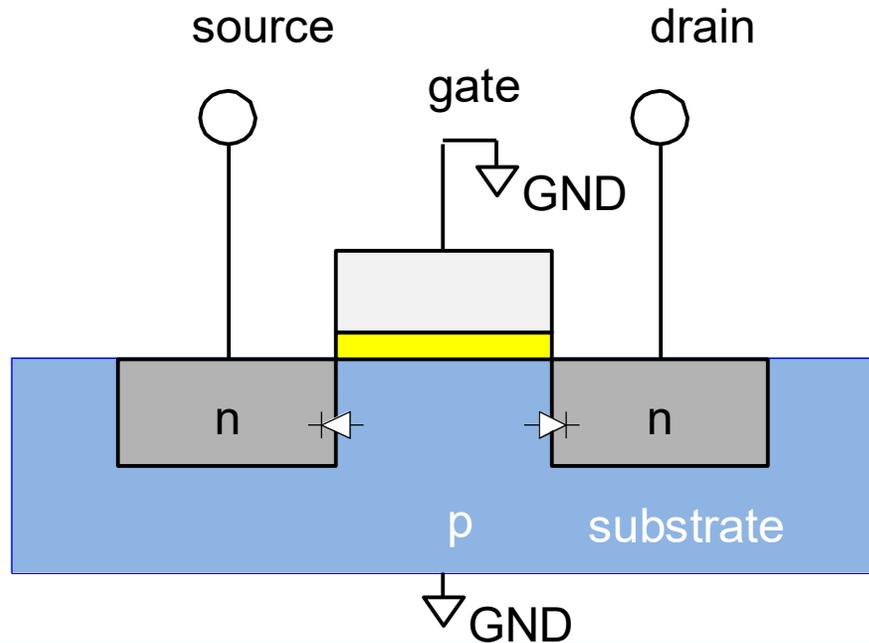
Matrița unui procesor poate conține miliarde de astfel de tranzistoare.



Tranzistoare: nMOS

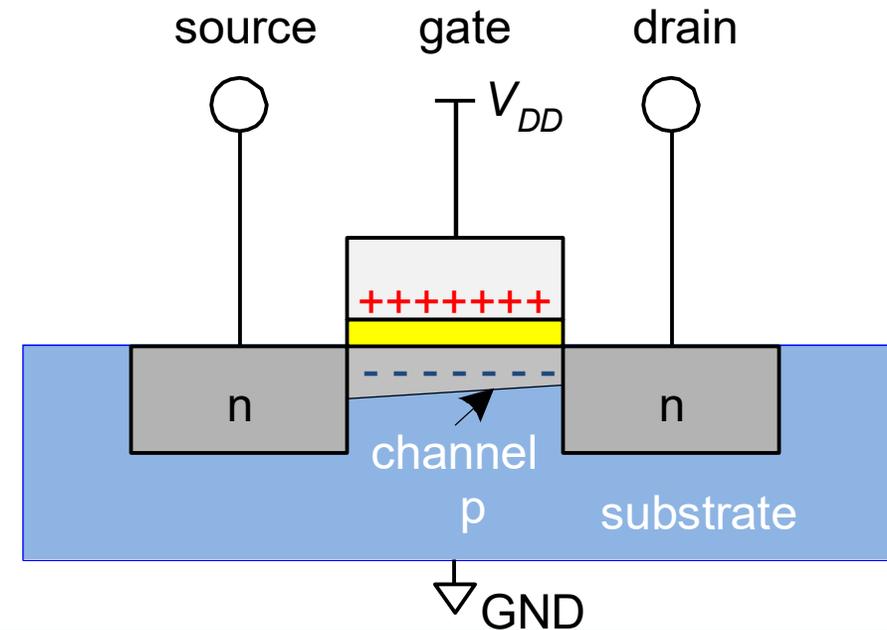
Gate = 0

OFF (nu avem conexiune între sursă și drenă)



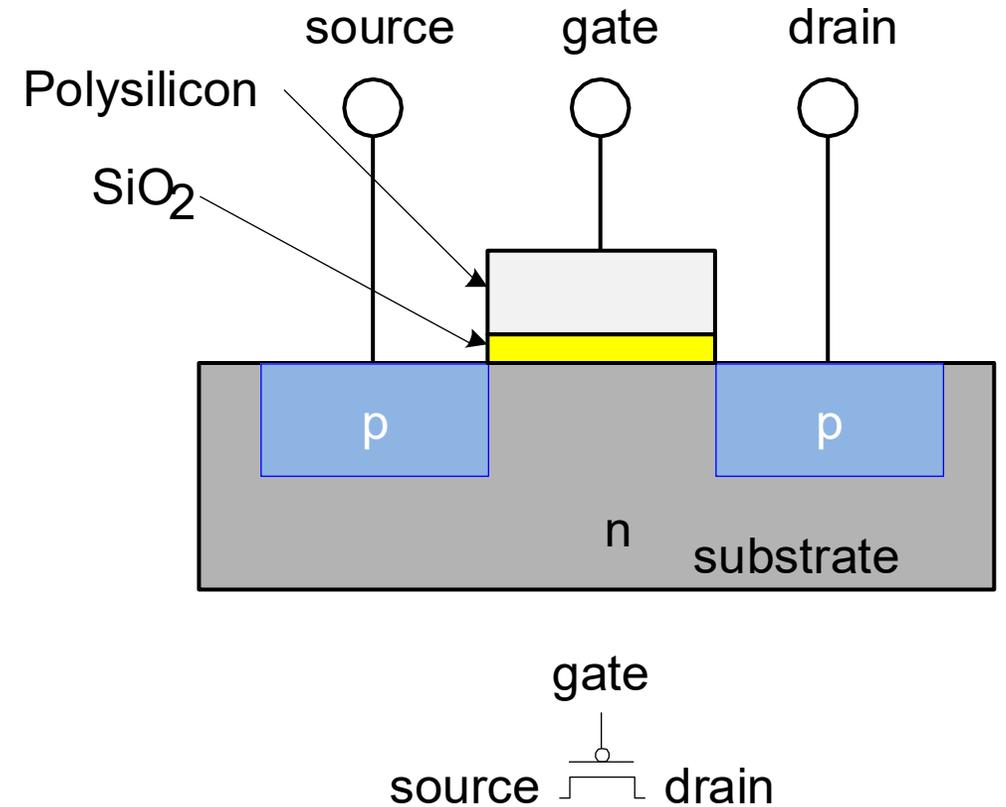
Gate = 1

ON (canal de conducție între sursă și drenă)



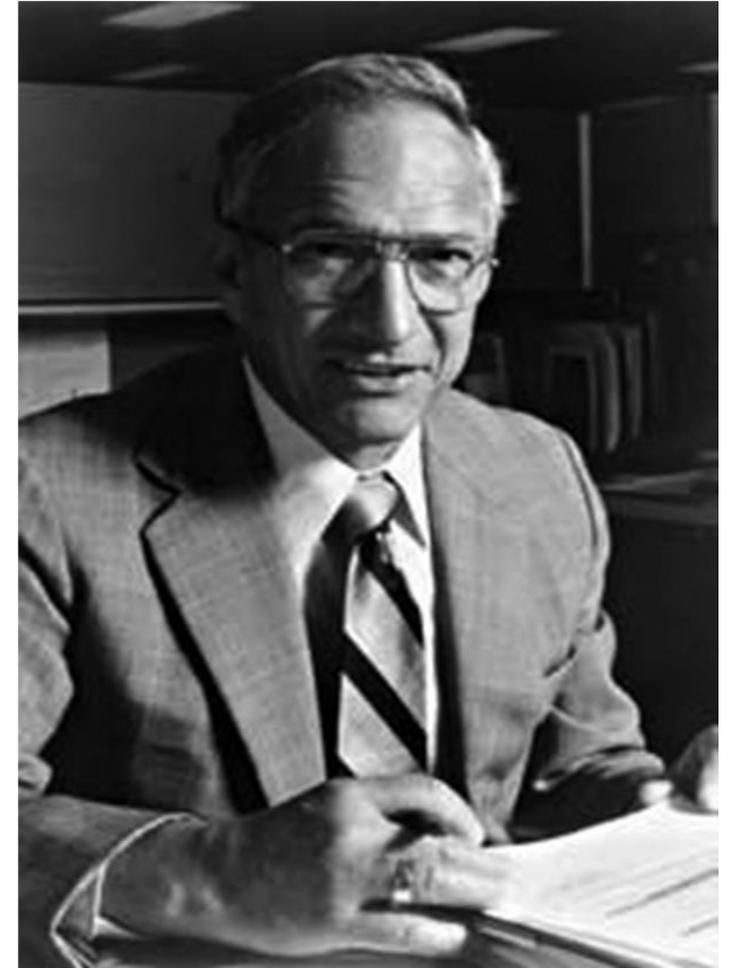
Tranzistoare: pMOS

- pMOS este opusul nMOS
 - ON când Poarta = 0
 - OFF când Poarta = 1



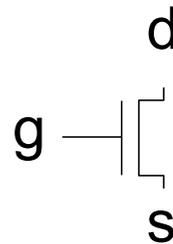
Robert Noyce, 1927-1990

- Poreclit “Primarul din Silicon Valley”
- Co-fondator Fairchild Semiconductor în 1957
- Co-fondator Intel în 1968
- Co-inventatorul circuitului integrat

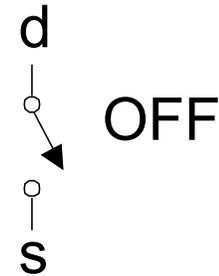


Funcționarea tranzistoarelor

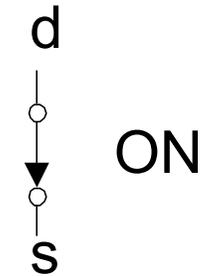
nMOS



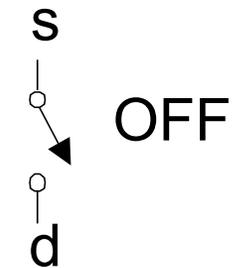
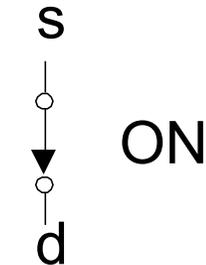
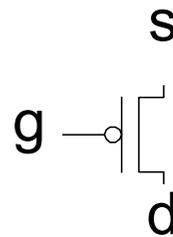
$g = 0$



$g = 1$

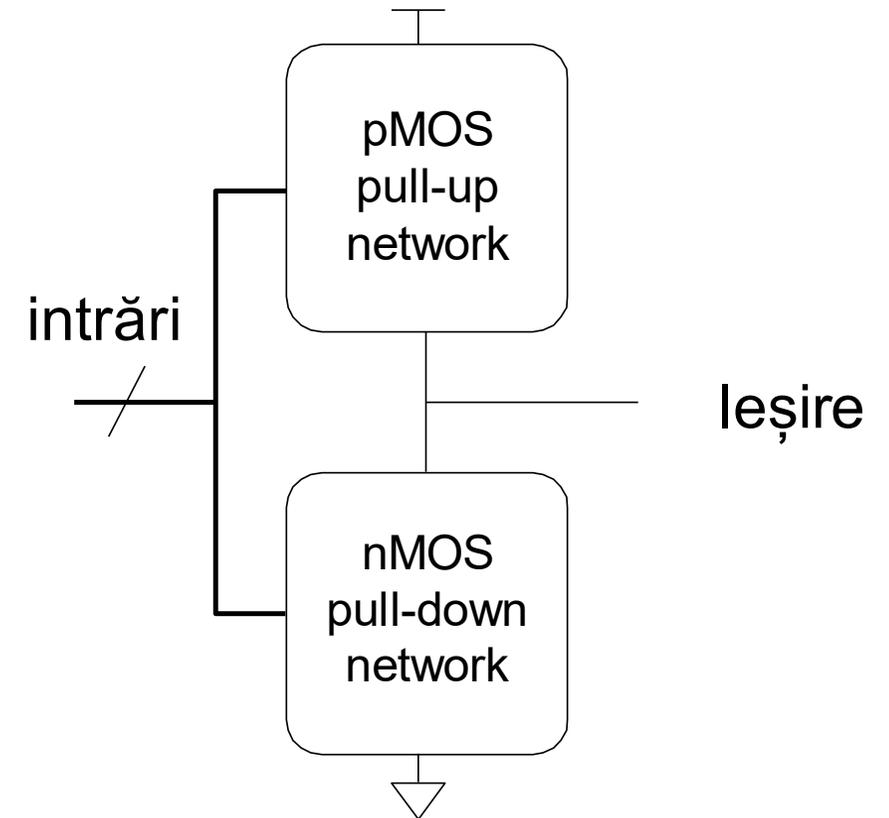


pMOS



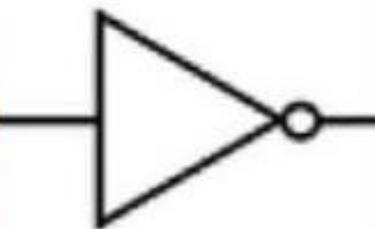
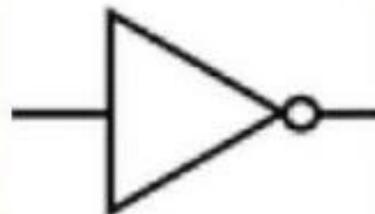
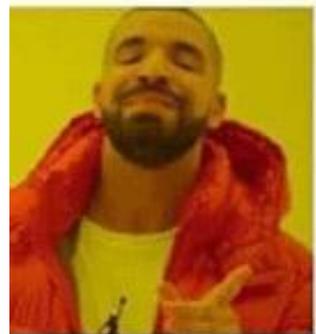
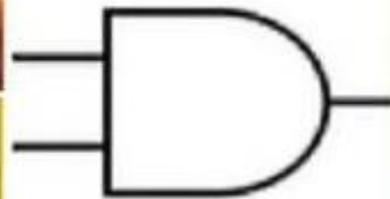
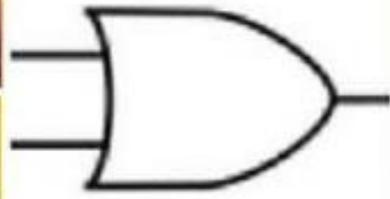
Funcționarea tranzistoarelor

- **nMOS:** lasă să treacă 0-uri, deci conectăm sursa la GND
- **pMOS:** lasă să treacă 1-uri, deci conectăm sursa la V_{DD}



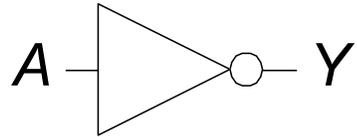
Porți logice

- **Implementează funcții logice:**
 - inversiune (NOT), AND, OR, NAND, NOR, etc.
- **O intrare:**
 - Poarta NOT, buffer
- **Două intrări:**
 - AND, OR, XOR, NAND, NOR, XNOR
- **Intrări multiple**



Porți CMOS: Poarta NOT

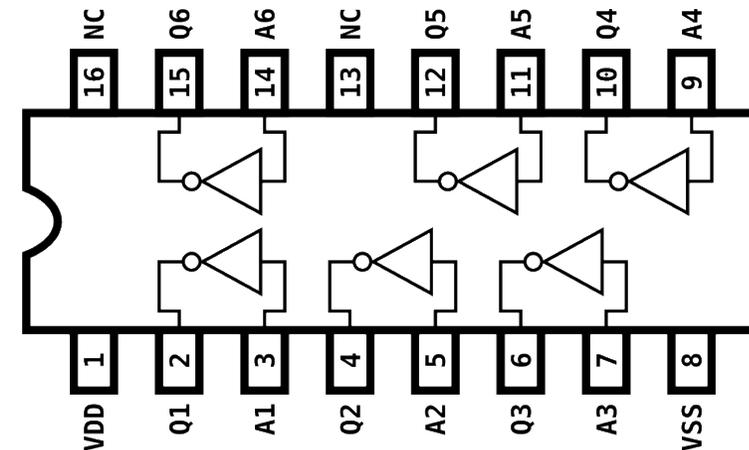
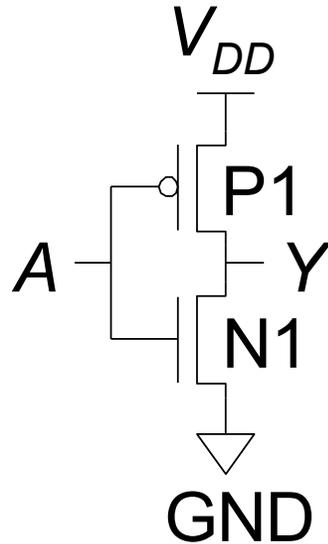
NOT



$$Y = \overline{A}$$

A	Y
0	1
1	0

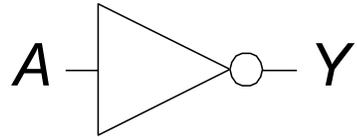
A	P1	N1	Y
0			
1			



CD4049, circuit integrat cu 6 porți logice NOT

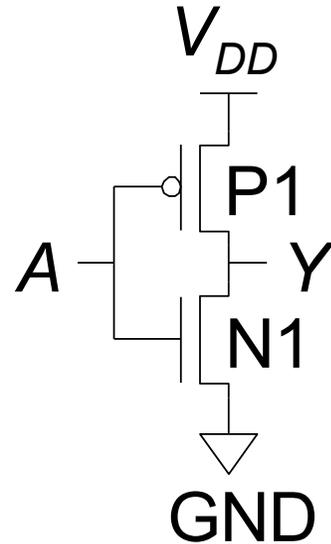
Porți CMOS: Poarta NOT

NOT

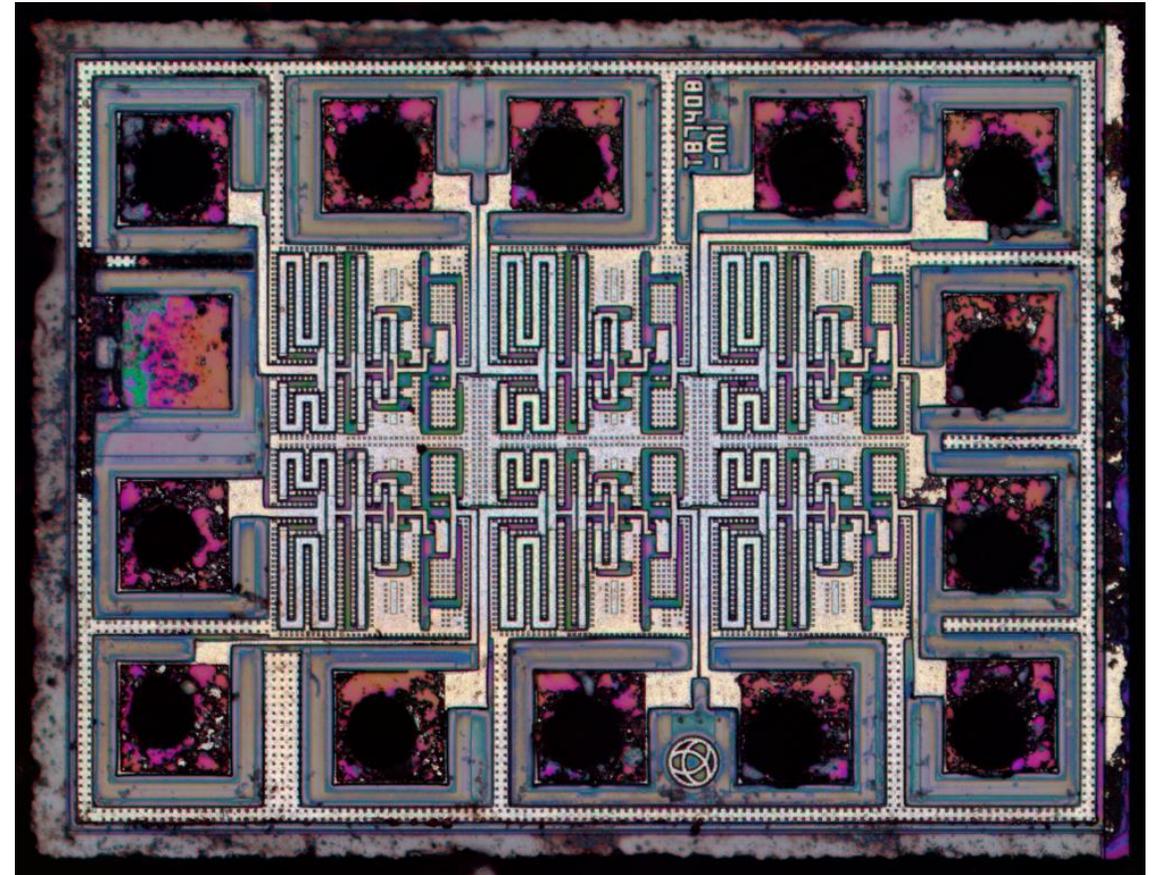


$$Y = \overline{A}$$

A	Y
0	1
1	0



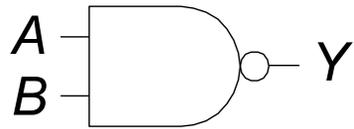
A	P1	N1	Y
0	ON	OFF	1
1	OFF	ON	0



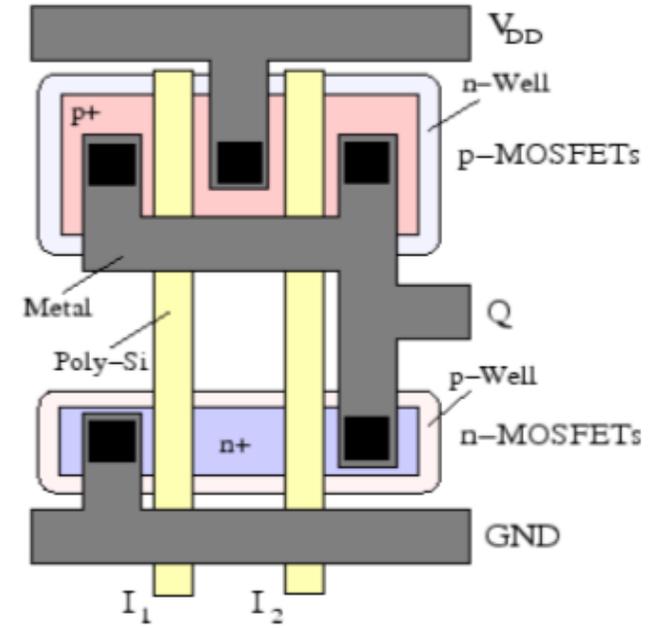
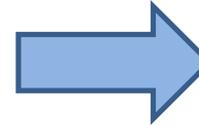
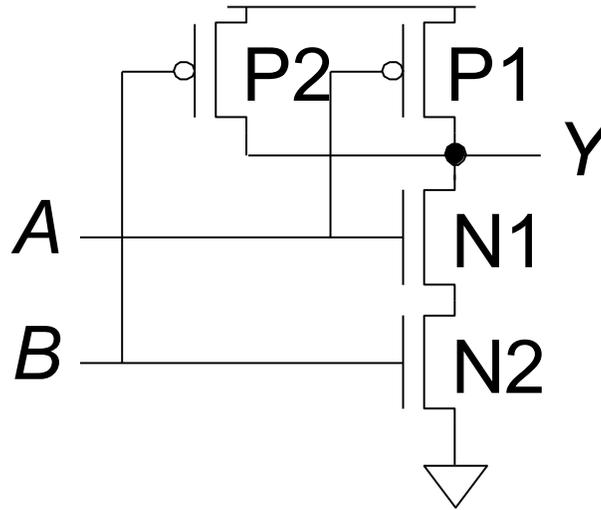
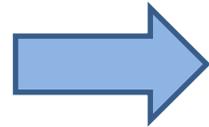
Matrița de siliciu a CD4049. Dimensiune 722x552 μ m
<https://zeptobars.com/en/read/CD4049-cmos-inverter-metal-gate>

Porți CMOS: Poarta NAND

NAND



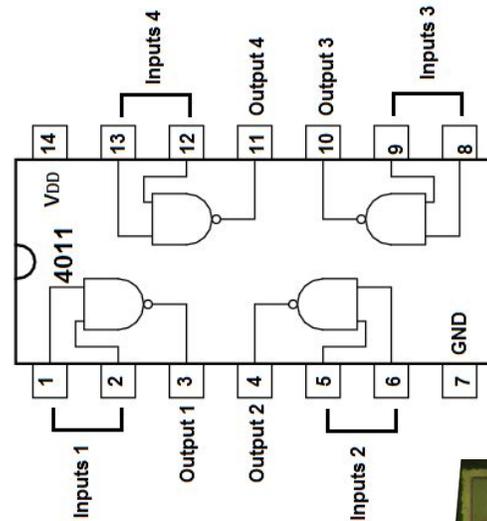
$$Y = \overline{AB}$$



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

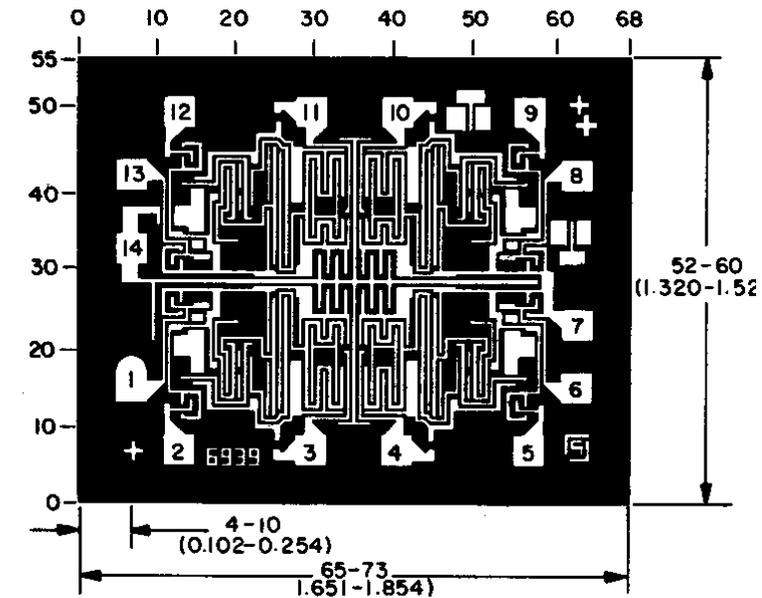
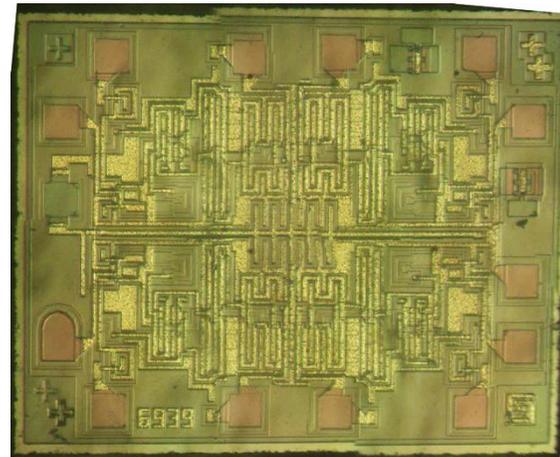
A	B	P1	P2	N1	N2	Y
0	0	ON	ON	OFF	OFF	1
0	1	ON	OFF	OFF	ON	1
1	0	OFF	ON	ON	OFF	1
1	1	OFF	OFF	ON	ON	0

Porți CMOS: Poarta NAND



Structura internă a circuitului CD4011

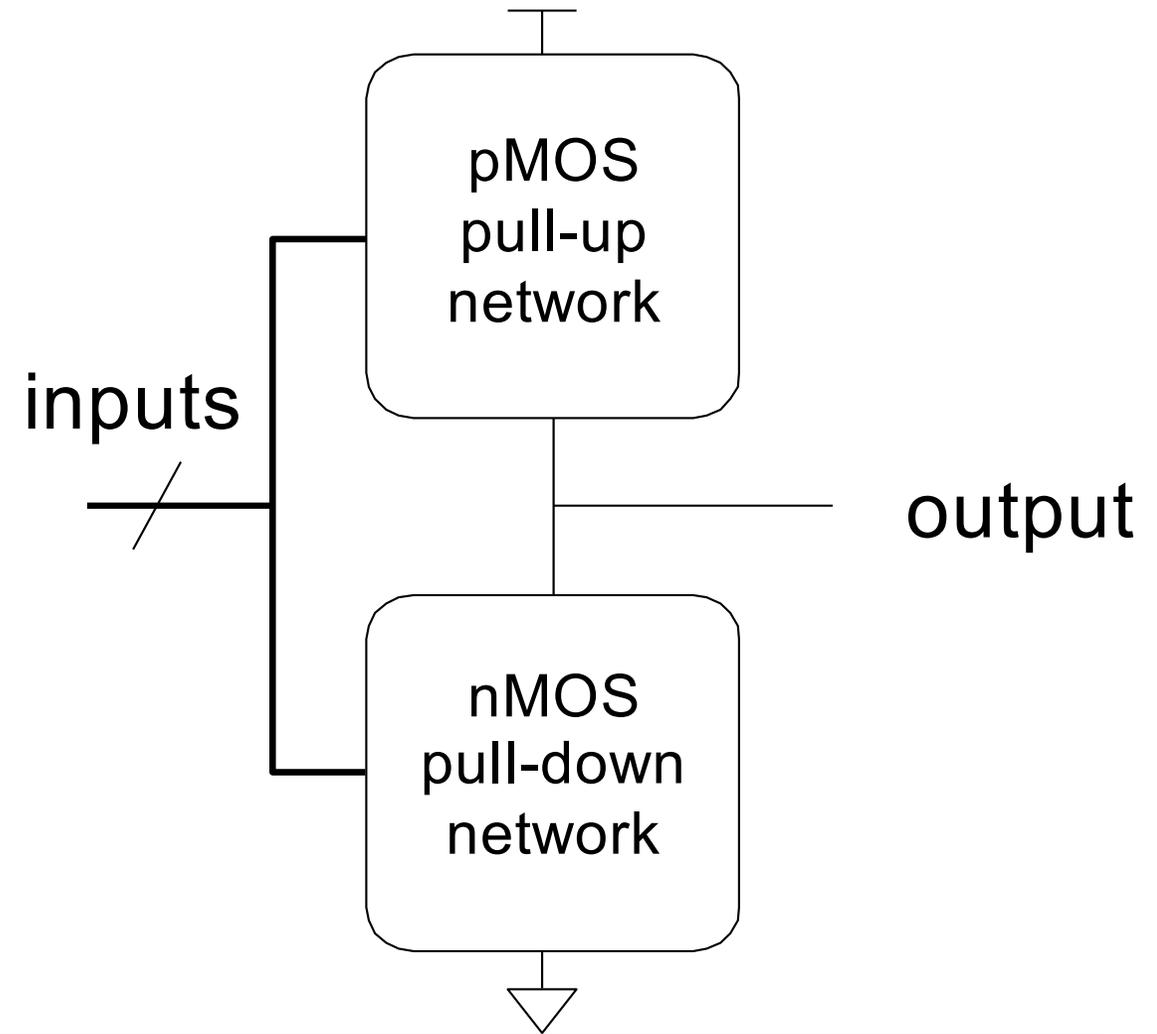
CD4011, circuit integrat cu 4 porți logice NAND



Structura unei porți CMOS

Structura internă a oricărei porți logice, indiferent de complexitate, poate fi simplificată la două structuri:

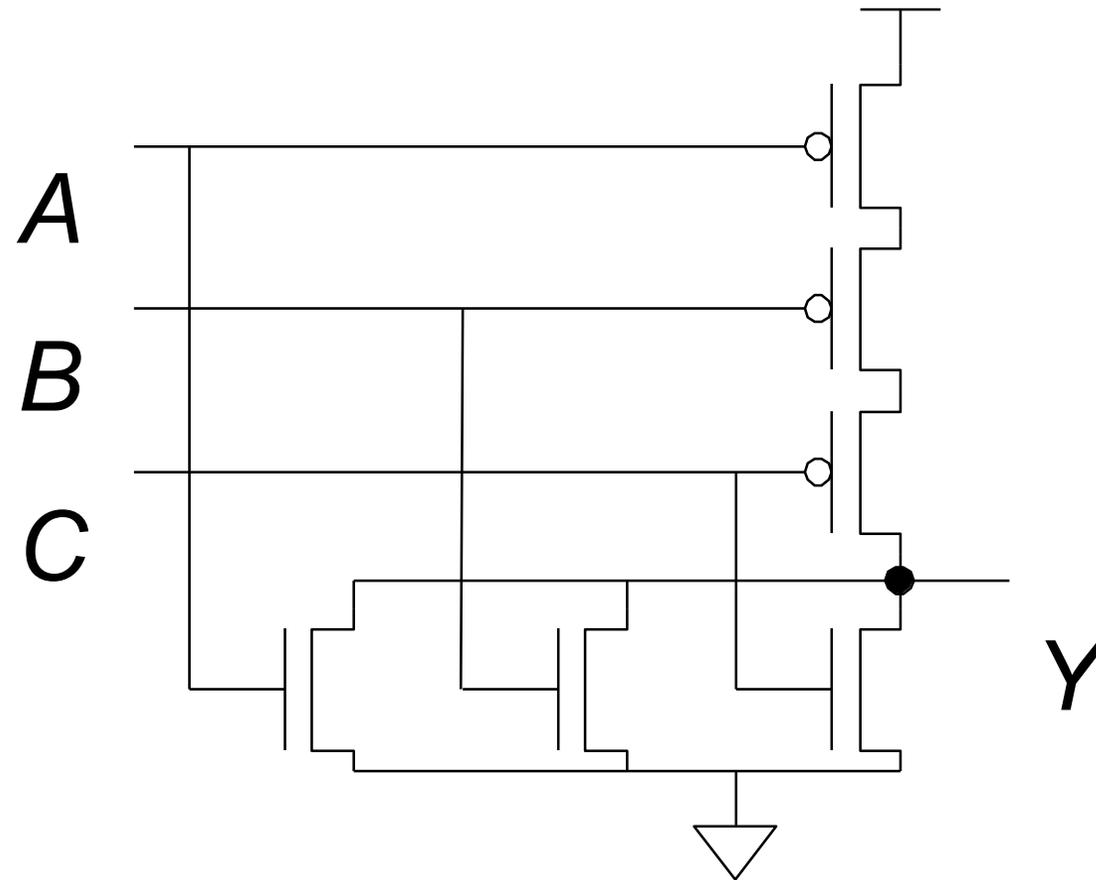
- Rețea **pull-up** -> trage ieșirea la '1' logic (V_{cc})
- Rețea **pull-down** -> trage ieșirea la '0' logic (GND)



Poarta NOR

Cum construim o poartă NOR cu trei intrări?

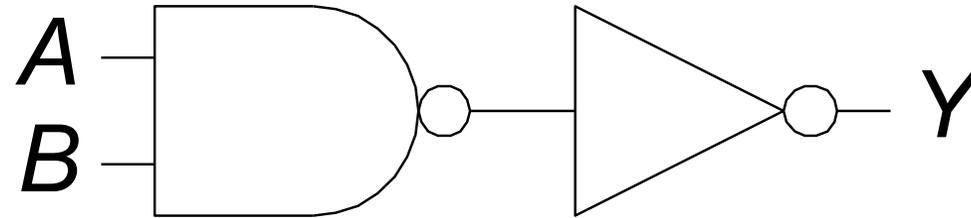
Poarta NOR3



Alte porți CMOS

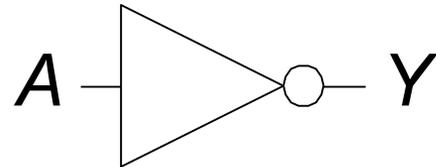
Cum construim o poartă AND cu două intrări?

Poarta AND2



Porți logice cu o intrare

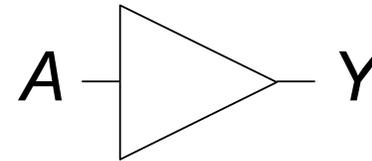
NOT



$$Y = \bar{A}$$

A	Y
0	1
1	0

BUF

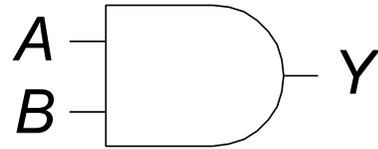


$$Y = A$$

A	Y
0	0
1	1

Porți logice cu două intrări

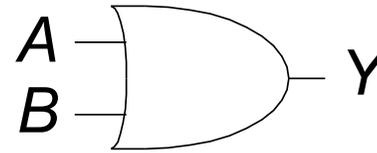
AND



$$Y = AB$$

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

OR

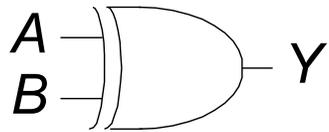


$$Y = A + B$$

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

Mai multe porți logice cu două intrări

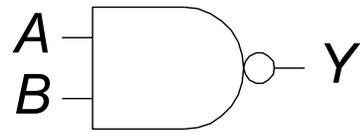
XOR



$$Y = A \oplus B$$

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

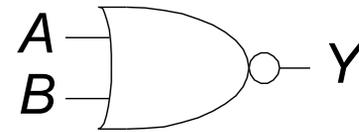
NAND



$$Y = \overline{AB}$$

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

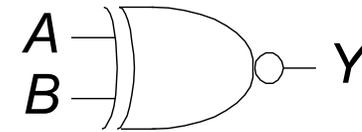
NOR



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

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

XNOR

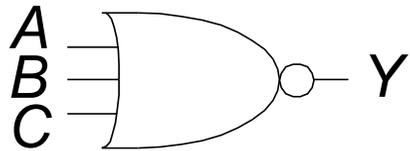


$$Y = \overline{A \oplus B}$$

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

Porți logice cu mai multe intrări

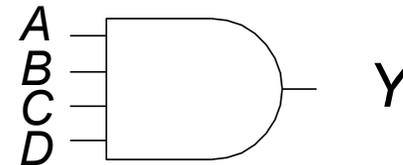
NOR3



$$Y = \overline{A+B+C}$$

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

AND4

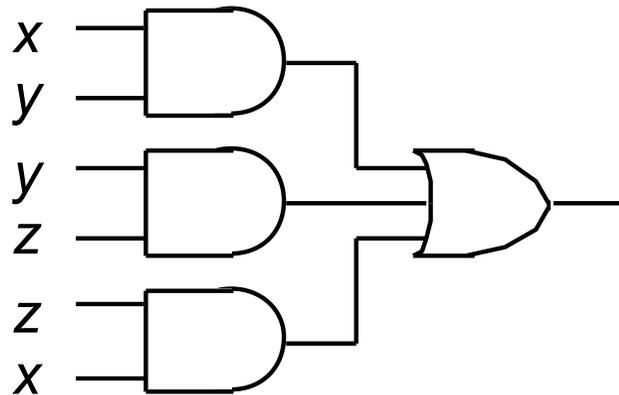


$$Y = ABCD$$

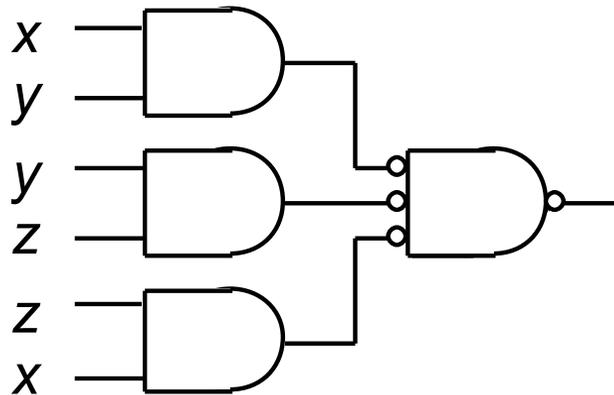
A	B	C	D	Y
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

Proiectarea rețelelor de porți logice

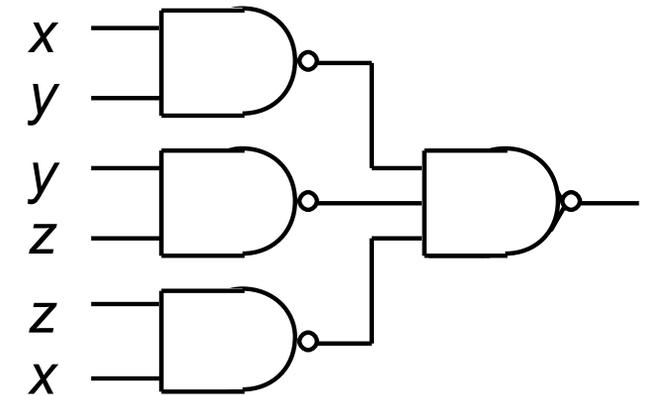
- AND-OR, NAND-NAND, OR-AND, NOR-NOR
- Optimizare logică: cost, viteză de execuție, putere disipată



(a) AND-OR circuit



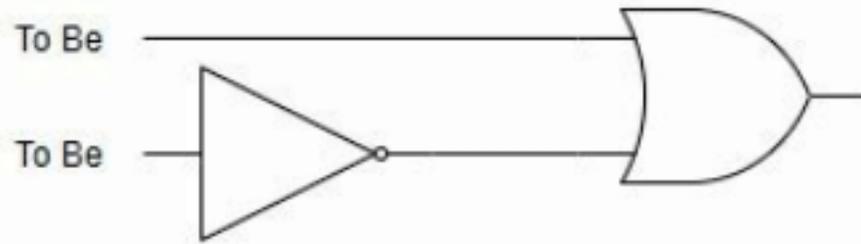
(b) Intermediate circuit



(c) NAND-NAND equivalent

Un circuit AND-OR pe 2 niveluri și două circuite echivalente

Some guy named
William



Me, an intellectual who
understands Boolean
algebra

1

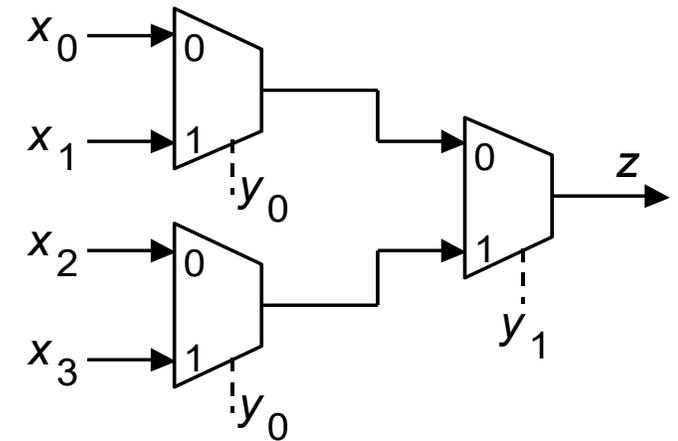
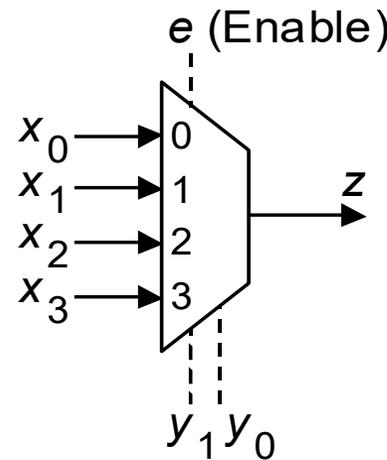
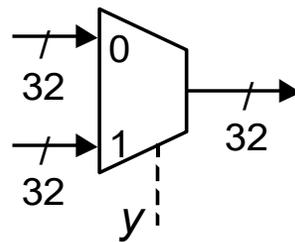
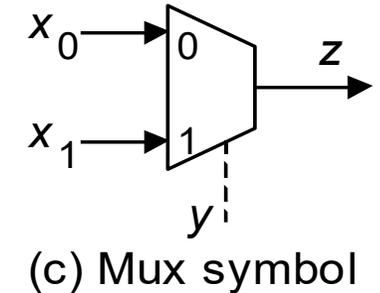
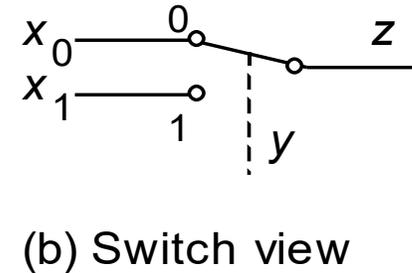
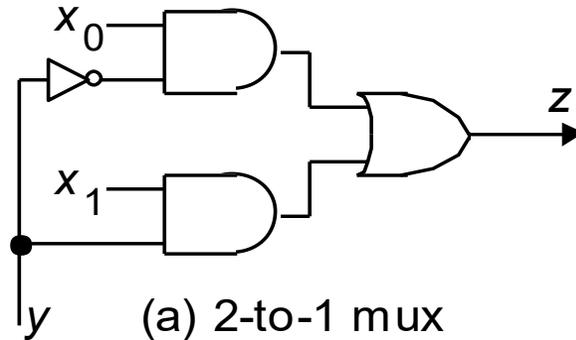
How inefficient of him

Circuite combinaționale utile

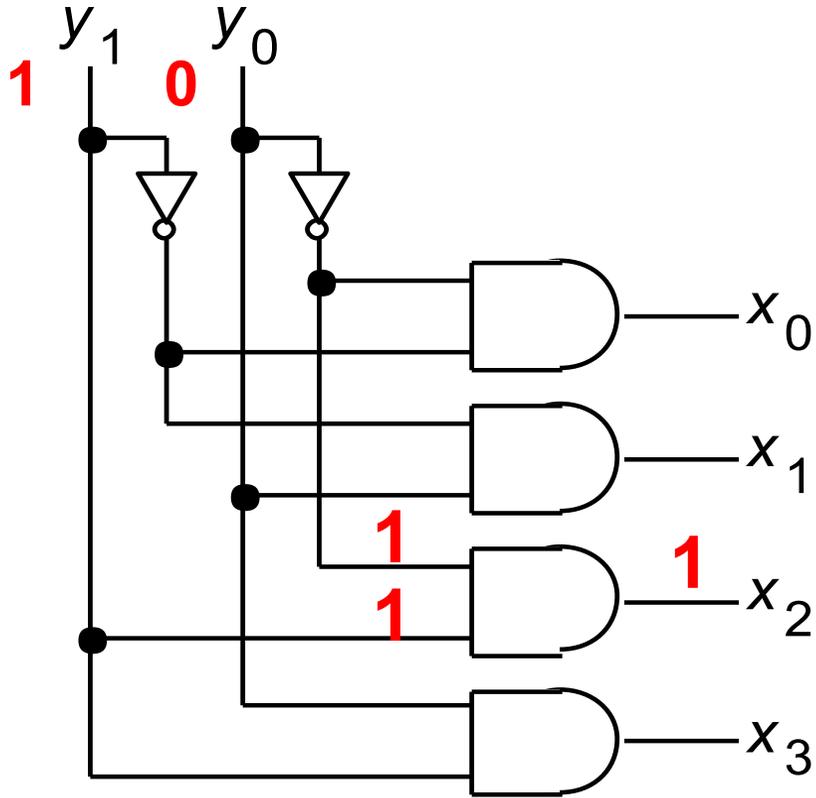
- Blocuri de nivel înalt
- Seamănă cu blocurile prefabricate folosite în construcția unei case
 - Componente aritmetice (Sumatoare, Multiplicatoare, Unități Aritmetice-Logice)
 - În continuare vom vorbi despre trei componente utile: **multiplexoare, decodificatoare/demultiplexoare și codificatoare**

Multiplexoare

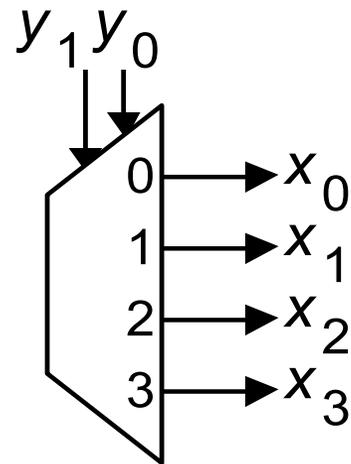
Permite selectarea uneia sau mai multor intrări și rutarea acestora la ieșire, în funcție de valoarea binară a intrărilor de selecție/adresă furnizate.



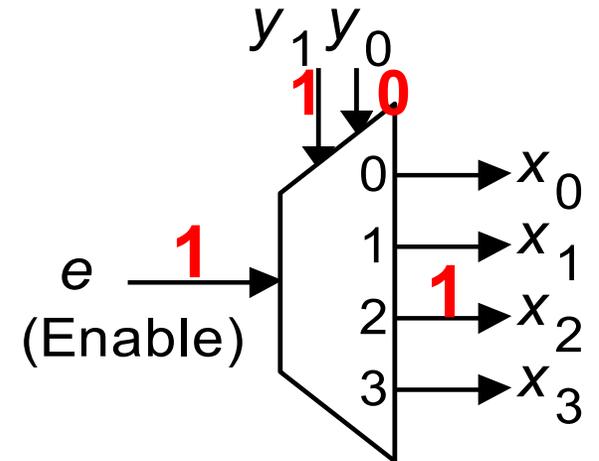
Decodificatoare/demultiplexoare



(a) 2-to-4 decoder



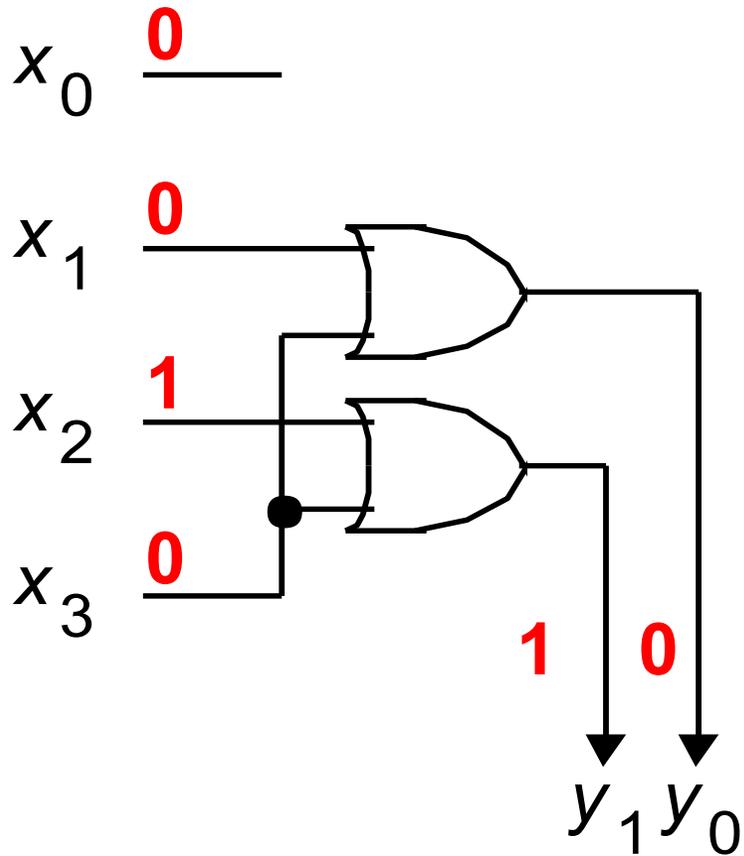
(b) Decoder symbol



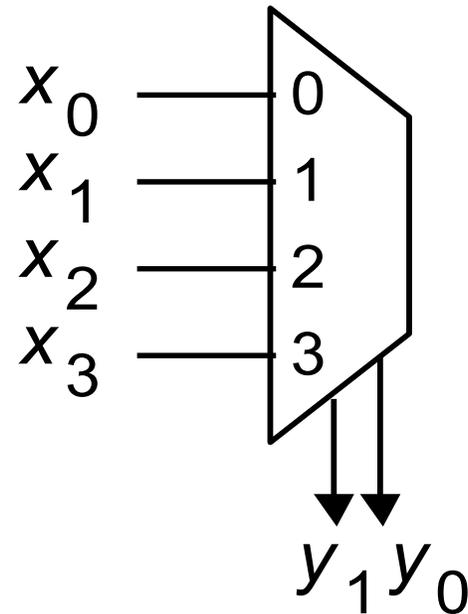
(c) Demultiplexer, or decoder with "enable"

Un decodificator permite selecția a uneia dintre cele 2^a intrări cu ajutorul a unei intrări de selecție pe a biți. Un demultiplexor (demux) selectează o ieșire în funcție de semnalele de adresă și o intrare suplimentară de Enable.

Codificatoare



(a) 4-to-2 encoder



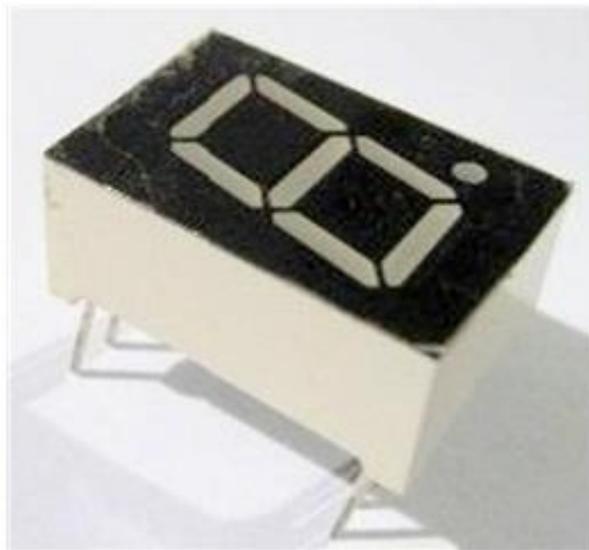
(b) Encoder symbol

Un codificator 2^a -la- a furnizează la ieșire un număr binar pe a -biți egal cu rangul valorii unice de 1 de la intrare.

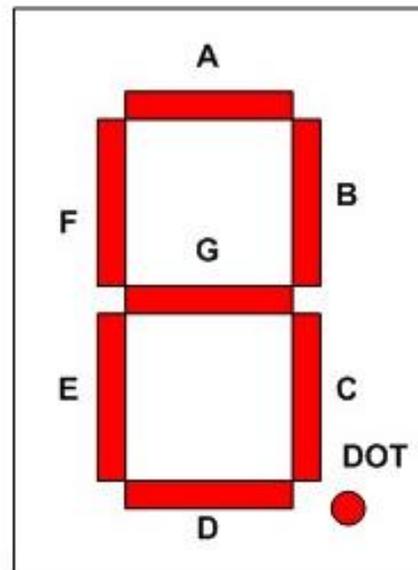
Afișarea cifrelor zecimale pe un display cu 7 segmente



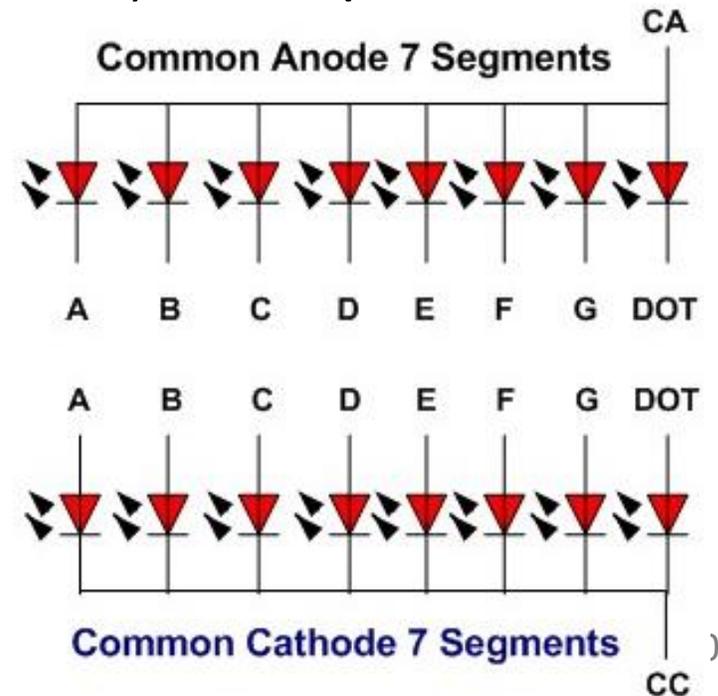
Afișaj seven-segment pentru toate cifrele zecimale. Fiecare din cele 7 segmente corespunde unui bit care poate fi activ (\Rightarrow segment aprins) sau inactiv (segment stins). În acest fel putem afișa cifre pe un ecran.



Typical 7 Segments Display

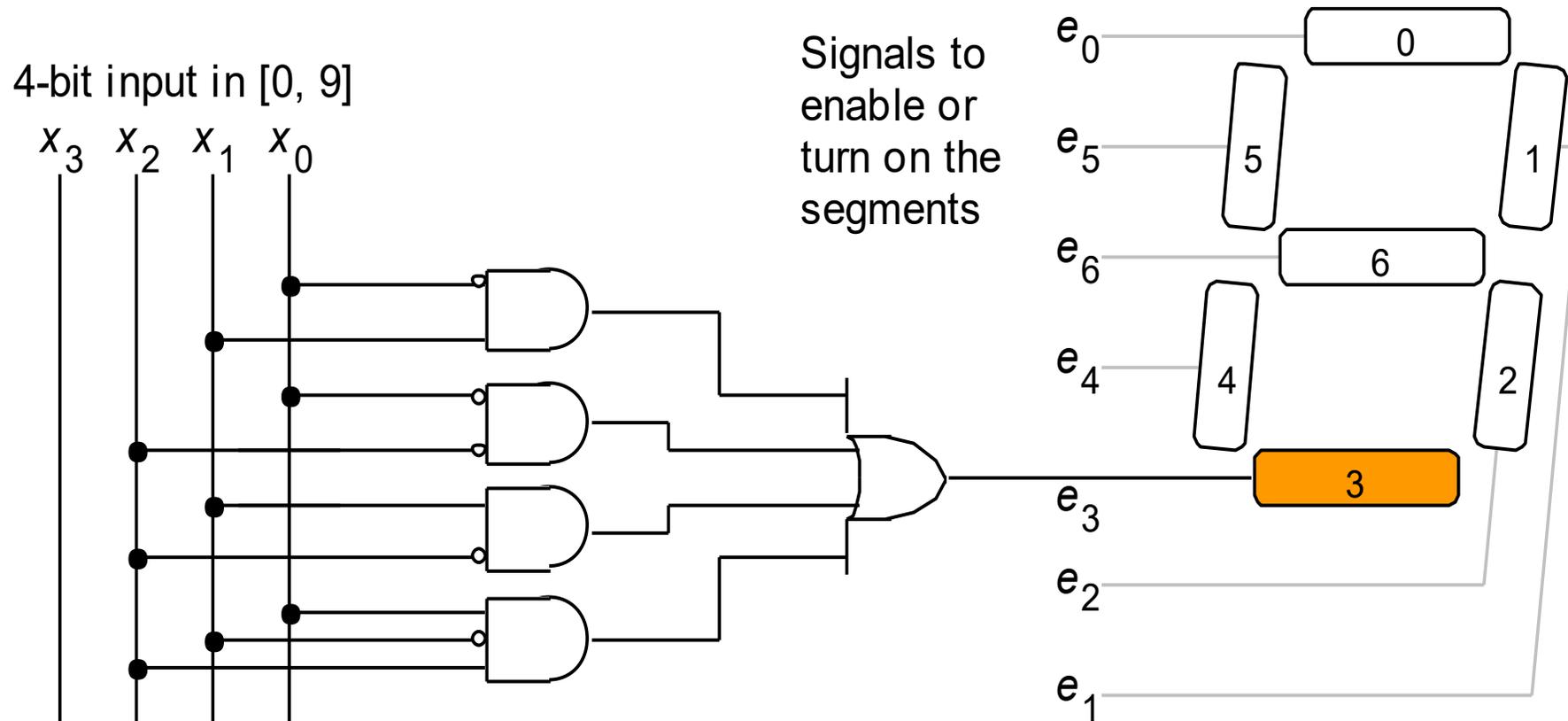


The 7 Segment's Name and the DOT



Common Cathode 7 Segments

Decodificator BCD-to-Seven-Segment

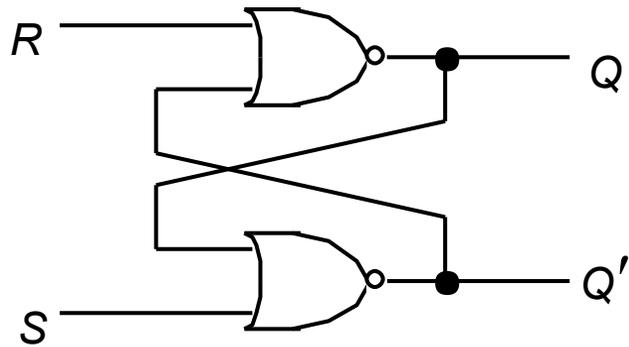


Circuitul logic care generează semnalul de Enable pentru segmentul cel mai de jos al unui digit 7-segment

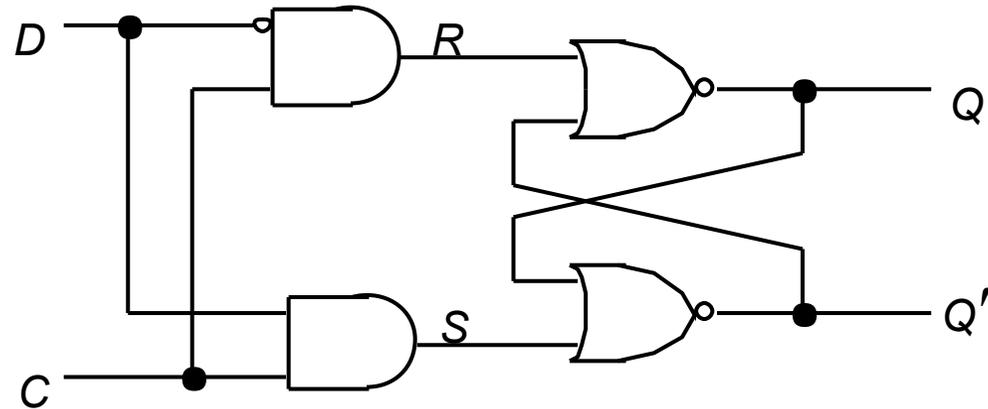
Circuite secvențiale utile

- Blocuri de nivel înalt
- Folosesc bistabile și componente combinaționale pentru a implementa **circuite cu memorie** – registre, registre de deplasare, numărătoare etc.

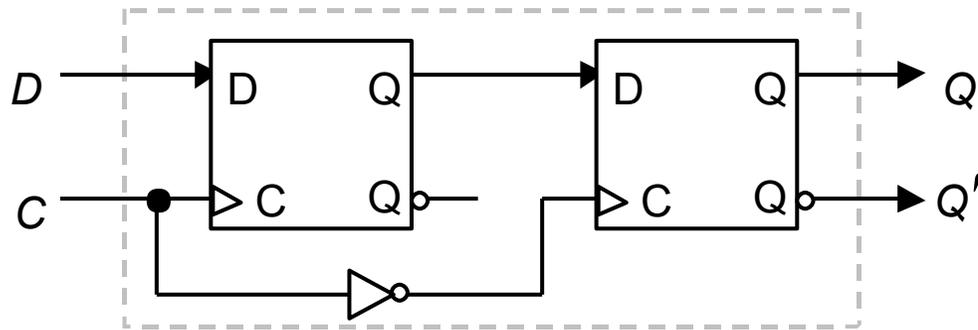
Latch-uri, Flip-Flop-uri și Registre



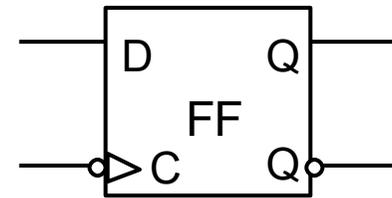
(a) SR latch



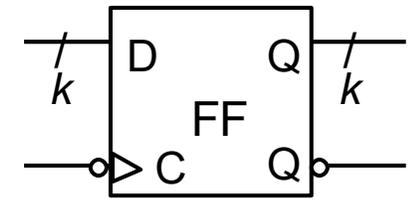
(b) D latch



(c) Master-slave D flip-flop



(d) D flip-flop symbol



(e) k -bit register

Automate FSM

Current state	----- Input -----		
	Dime	Quarter	Reset
S_{00}	S_{10}	S_{25}	S_{00}
S_{10}	S_{20}	S_{35}	S_{00}
S_{20}	S_{30}	S_{35}	S_{00}
S_{25}	S_{35}	S_{35}	S_{00}
S_{30}	S_{35}	S_{35}	S_{00}
S_{35}	S_{35}	S_{35}	S_{00}

Next state

S_{00} is the initial state
 S_{35} is the final state

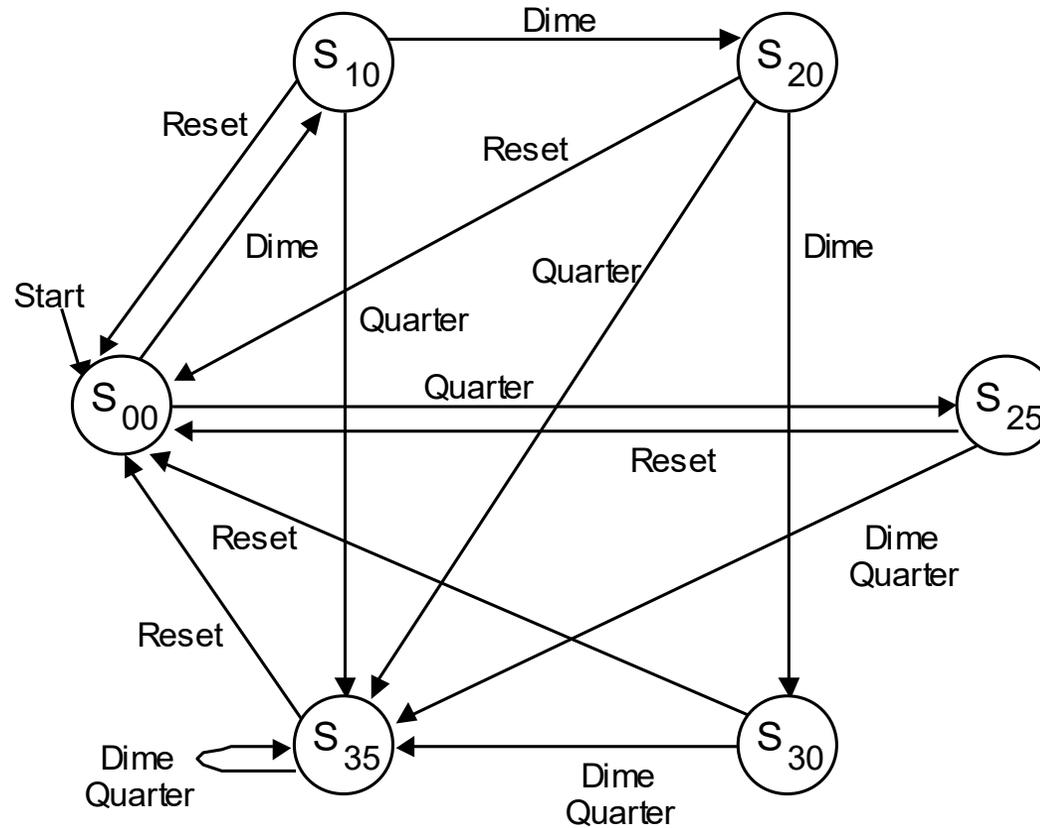
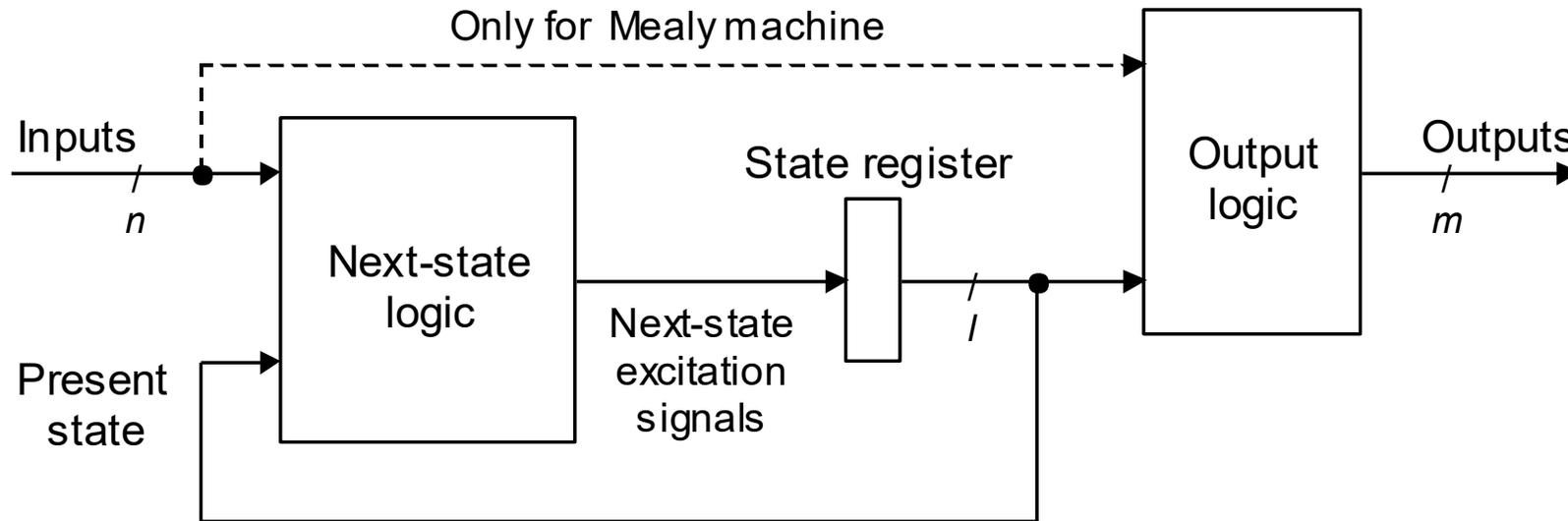


Tabela de stări și diagrama pentru un automat de băuturi răcoritoare.

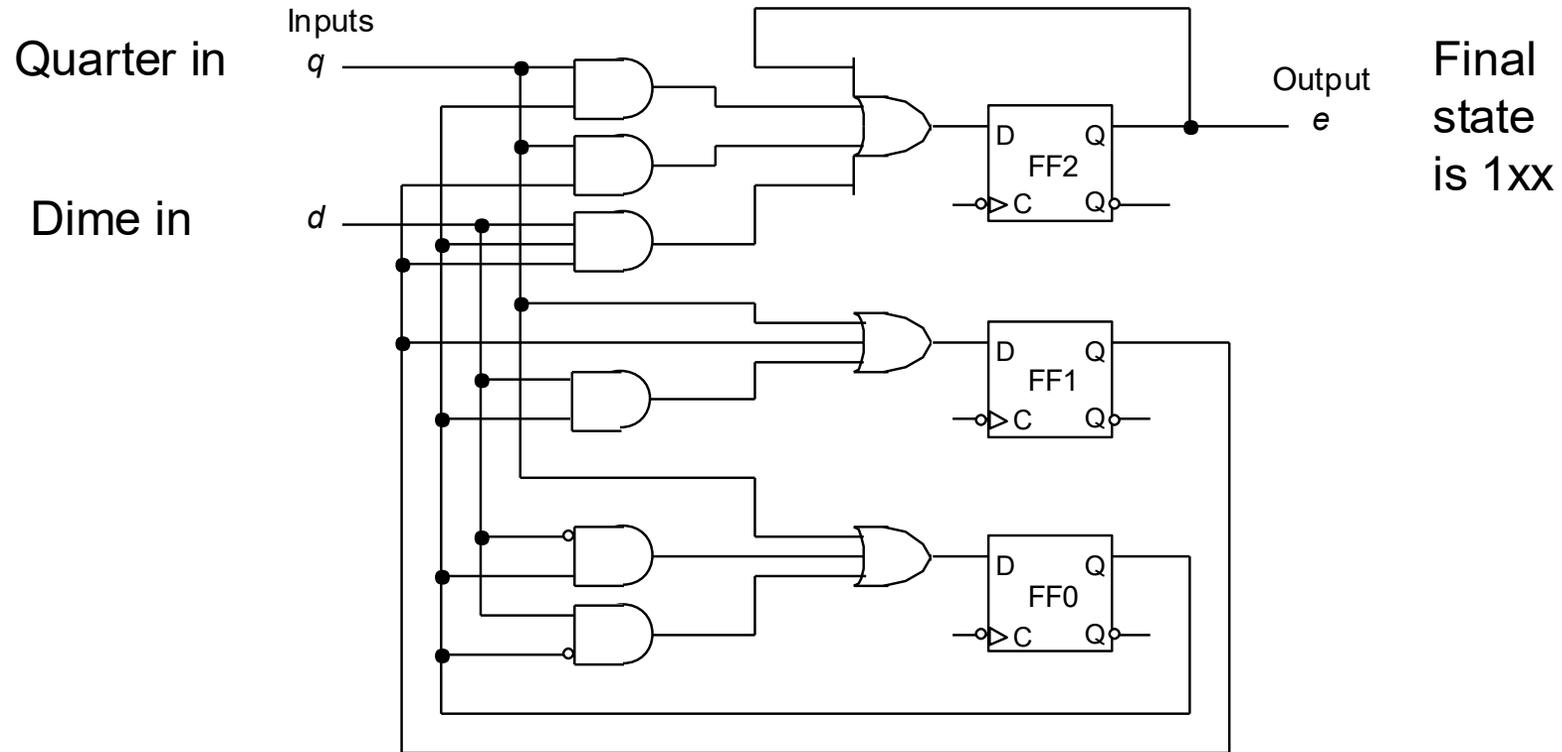
Implementarea logicii secvențiale



Realizarea hardware a automatelor secvențiale Moore și Mealy.

Proiectarea circuitelor secvențiale

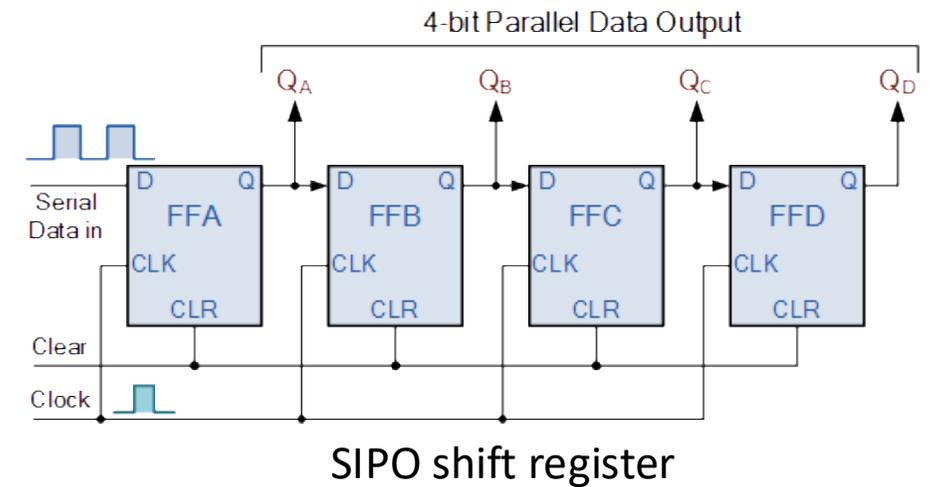
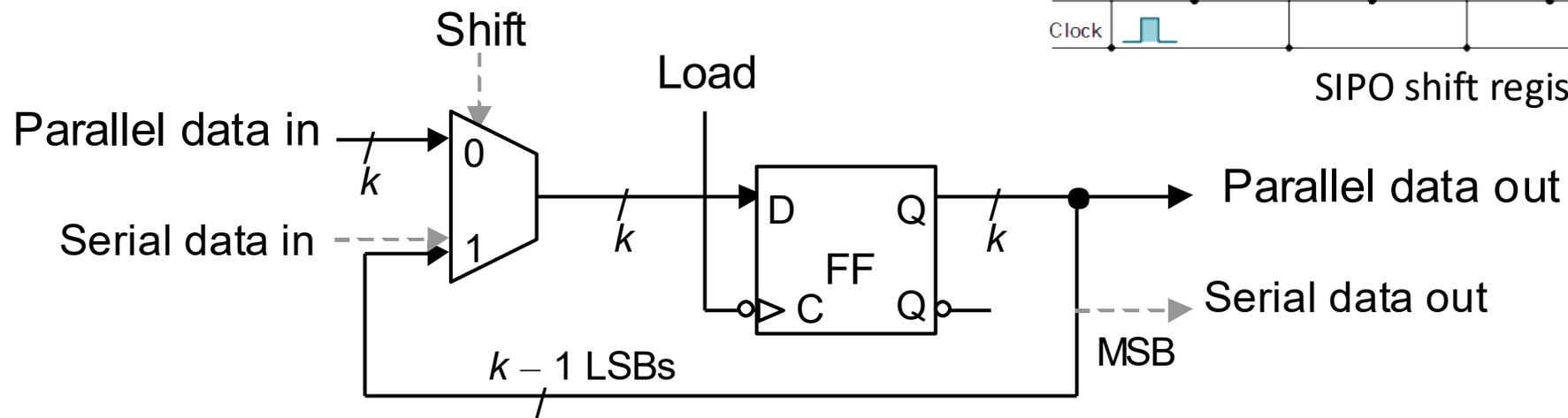
Exemplu



Implementarea hardware a automatului de răcoritoare.

Registre deplasare

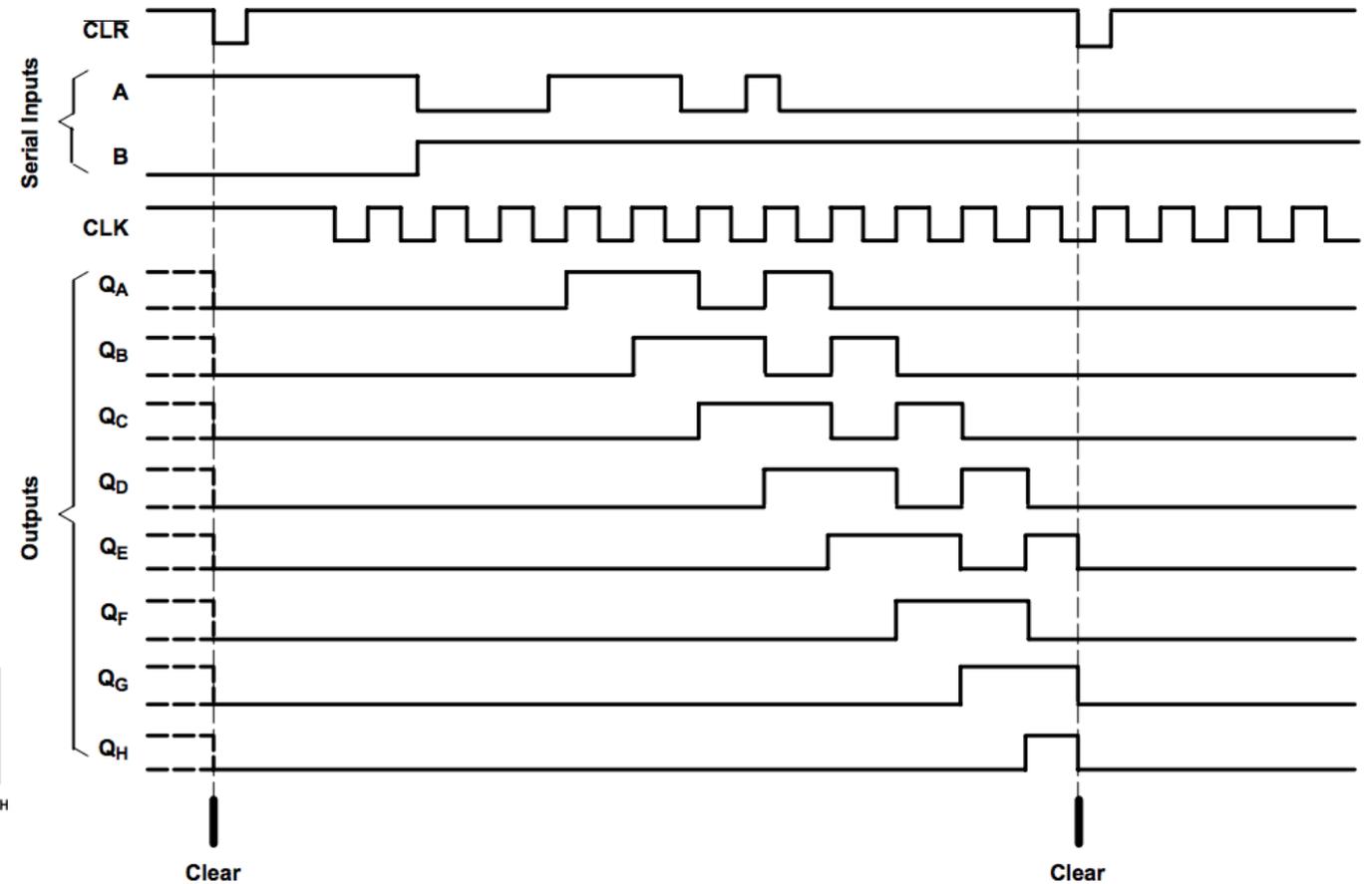
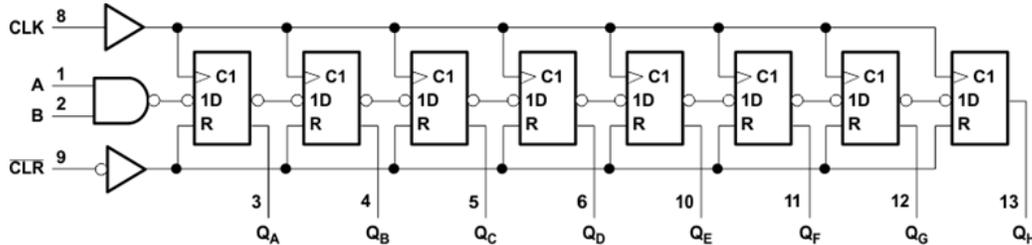
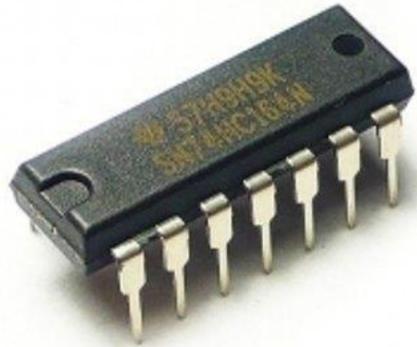
- Serial-In Serial-Out (SISO)
- Parallel-In Serial-Out (PISO)
- Serial-In Parallel-Out (SIPO)



Registru deplasare multifuncțional. Poate funcționa fie parallel-in, fie serial-in și poate furniza datele la ieșire fie paralel, fie serial.

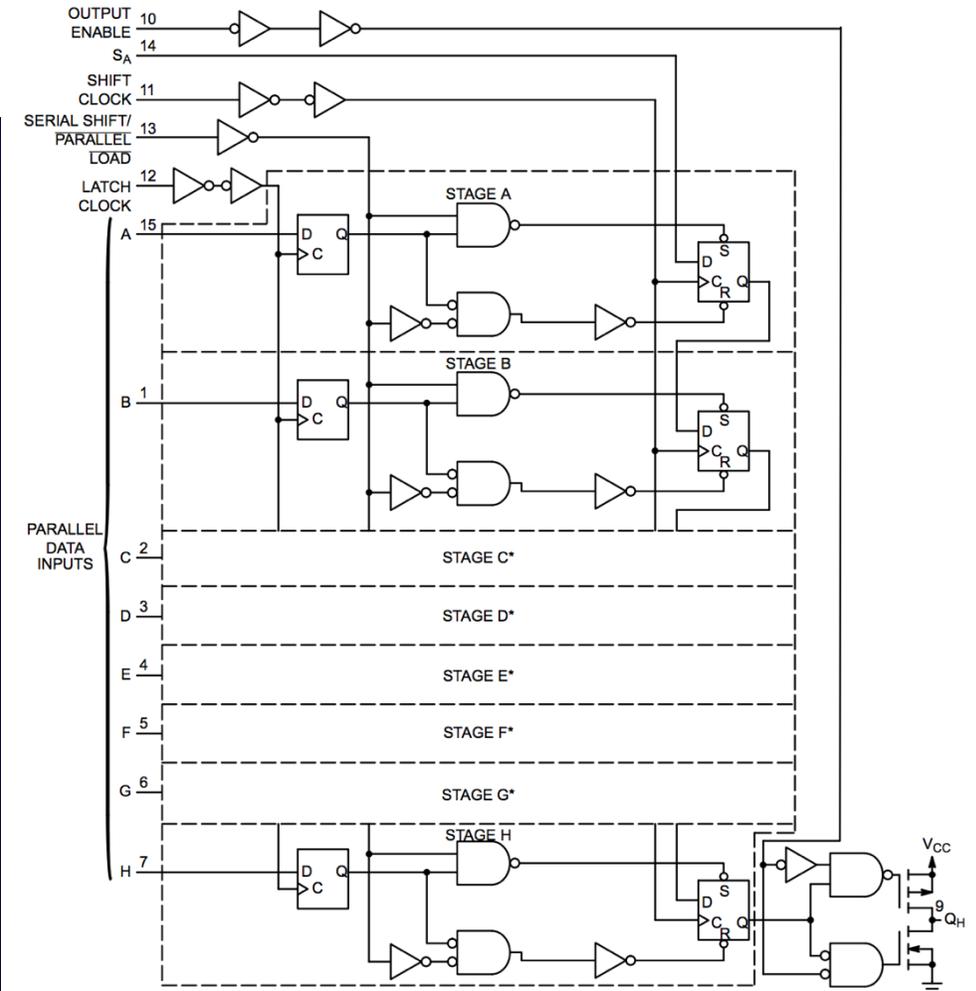
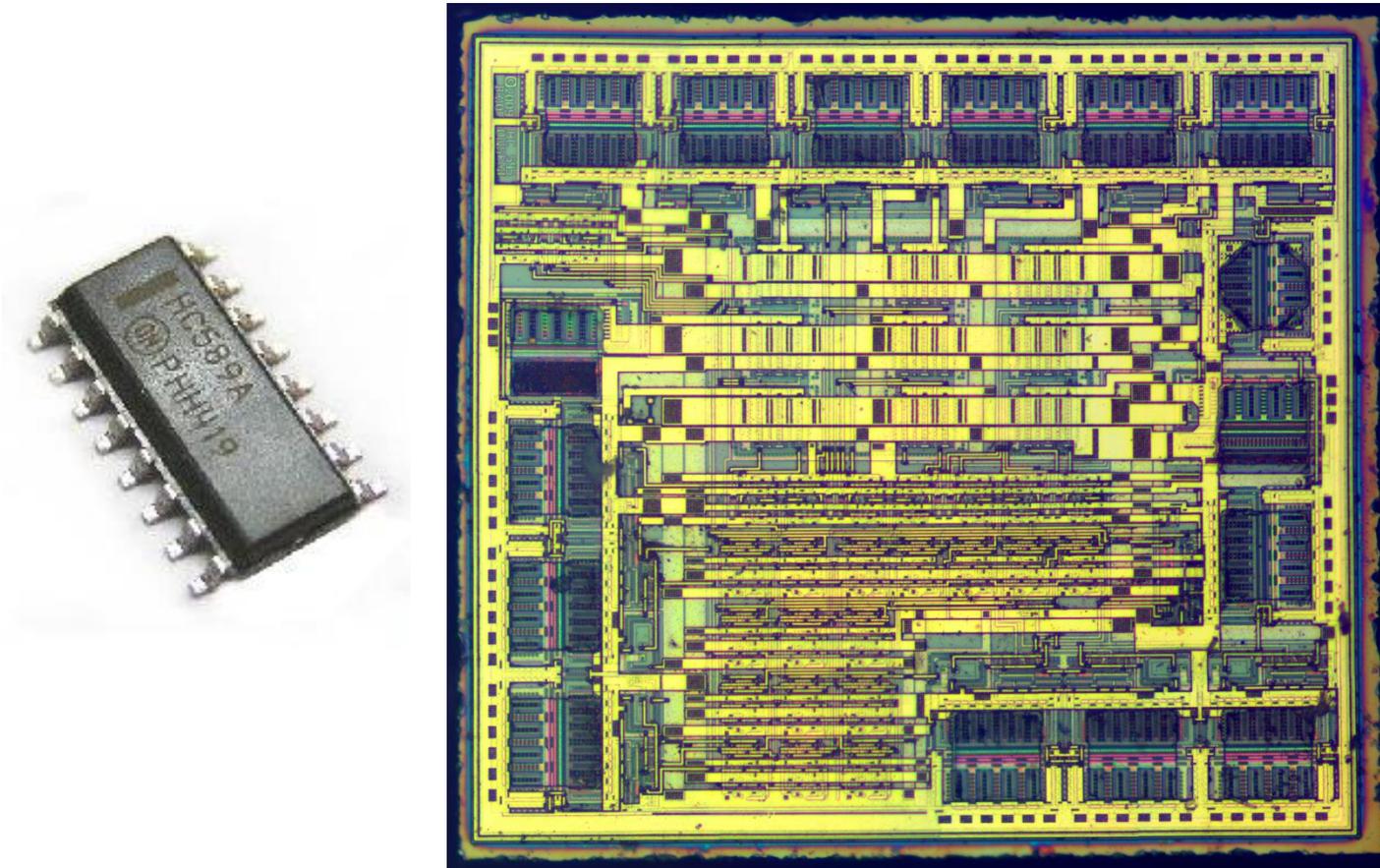
Registre deplasare - exemple

74HC164 – registru deplasare 8 biți Serial-In, Parallel-Out

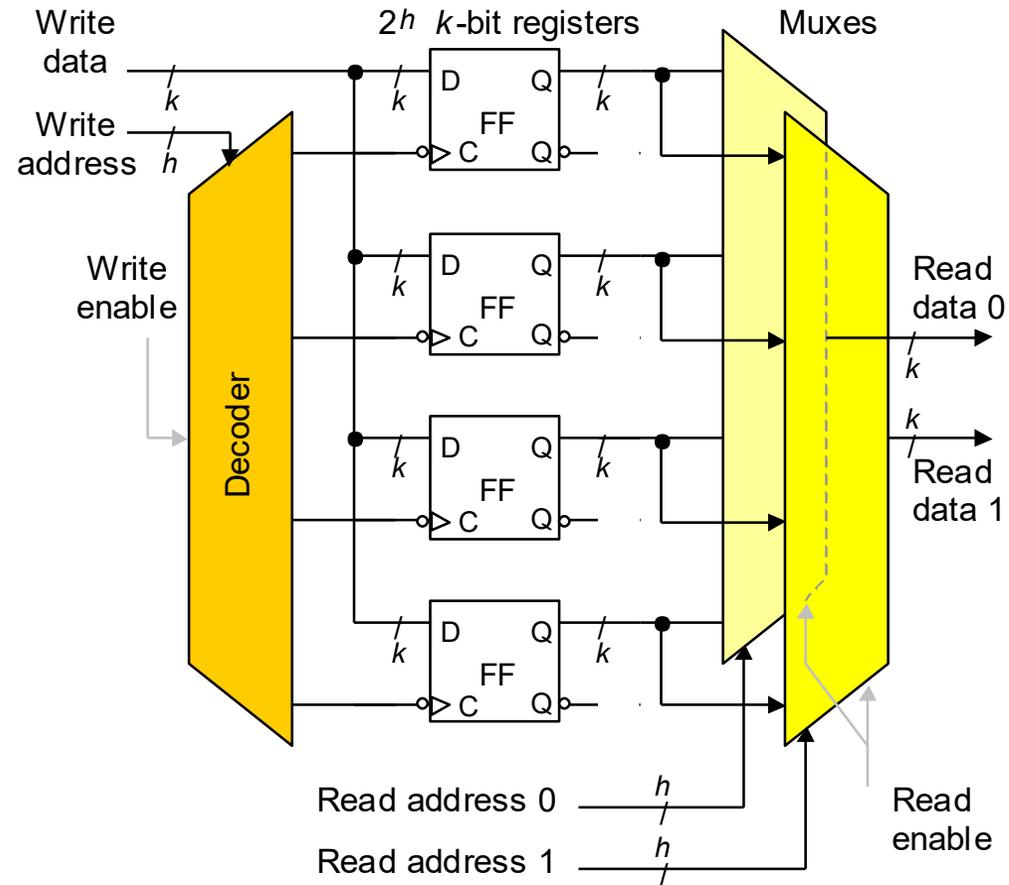


Registre deplasare - exemple

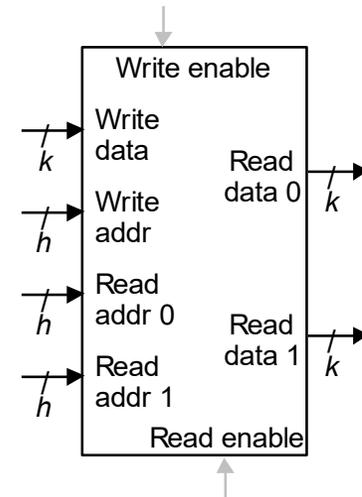
74HC589– registru deplasare 8 biți Parallel-In, Serial-Out



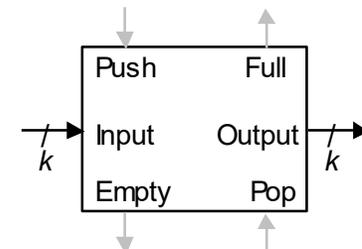
Registre și buffere FIFO



(a) Register file with random access



(b) Graphic symbol for register file

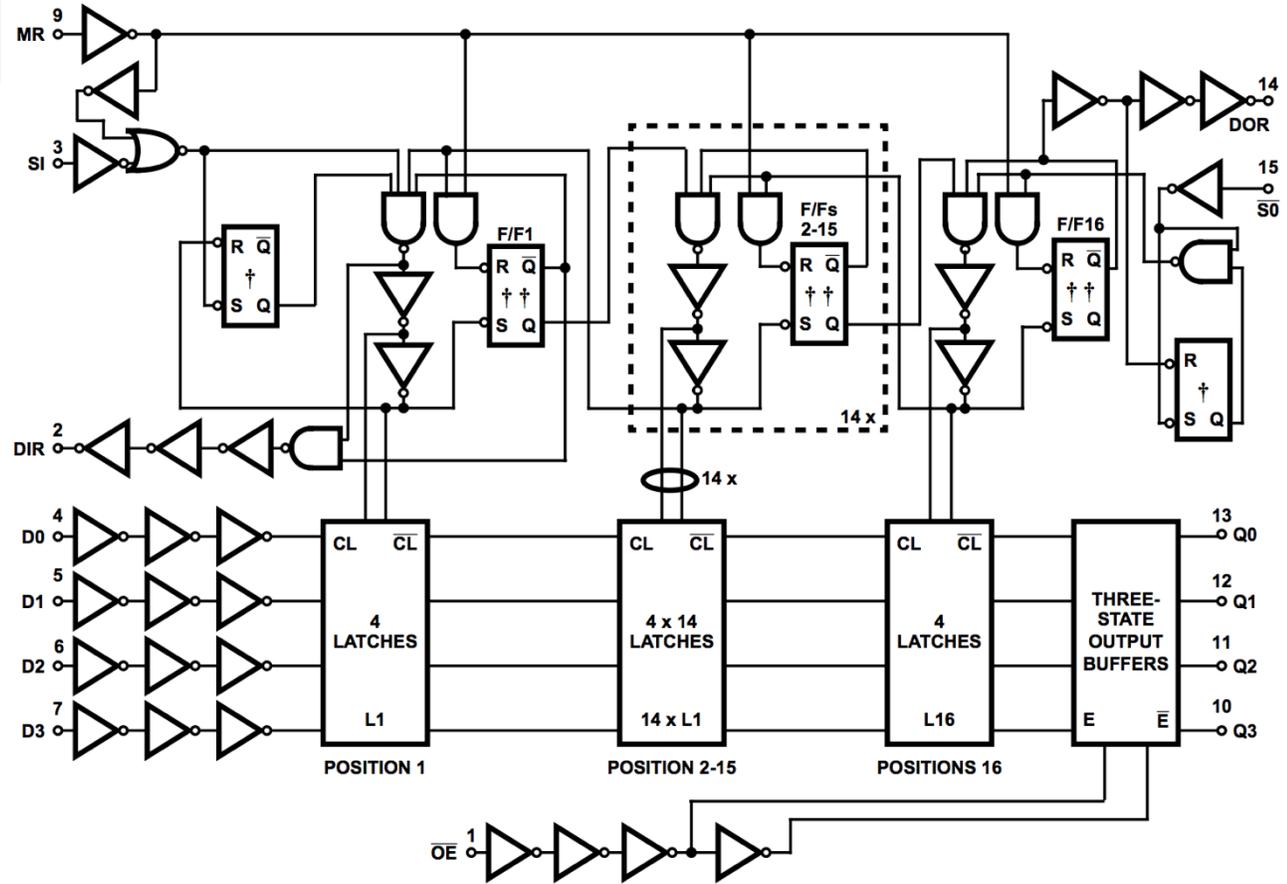
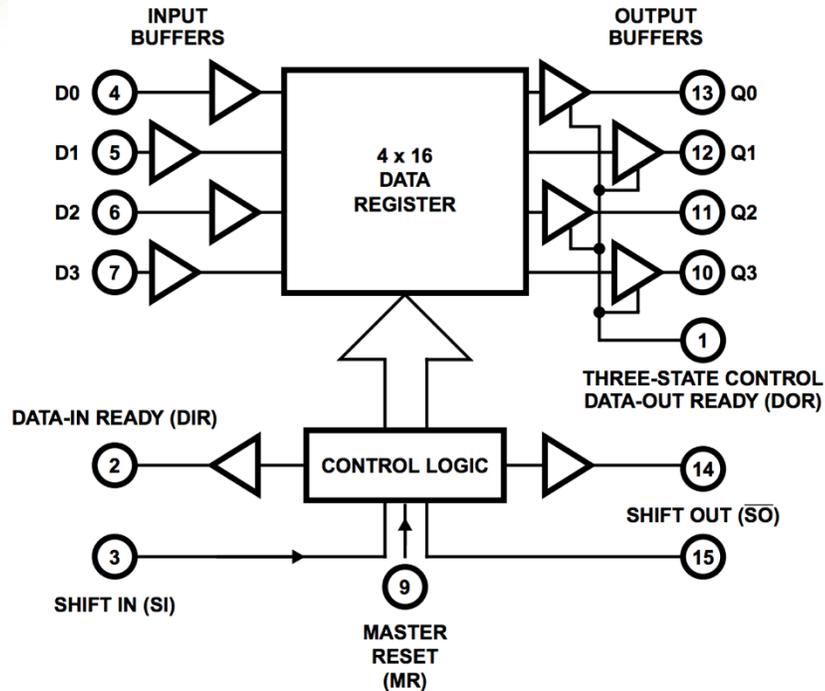


(c) FIFO symbol

Implementarea unui registru și a unui FIFO.

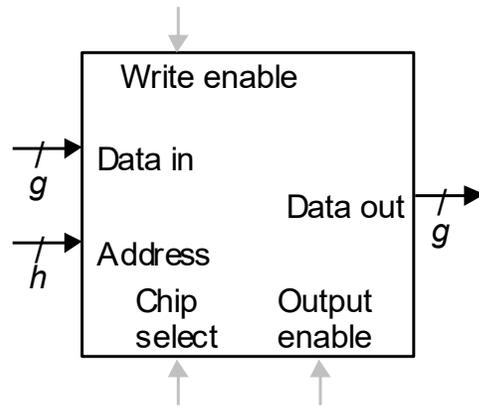
FIFO Buffer

74HC40105– registru deplasare 16 cuvinte x 4 biți

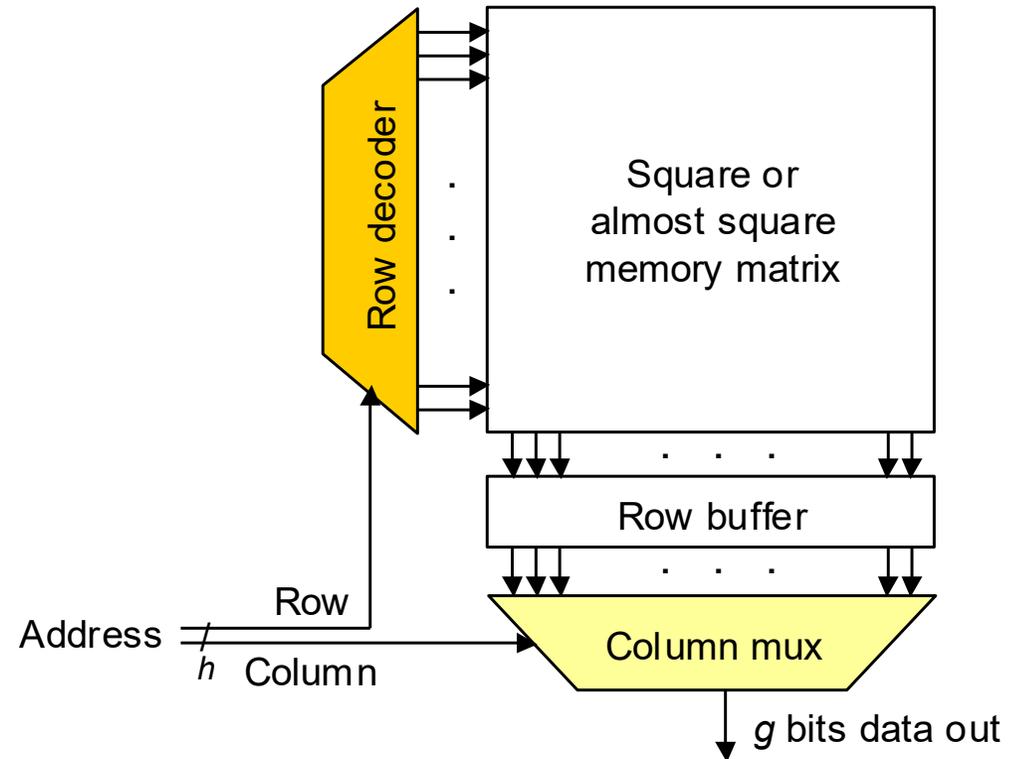


Datasheet: <http://www.ti.com/lit/ds/symlink/cd74hc40105.pdf>

Static RAM (SRAM)



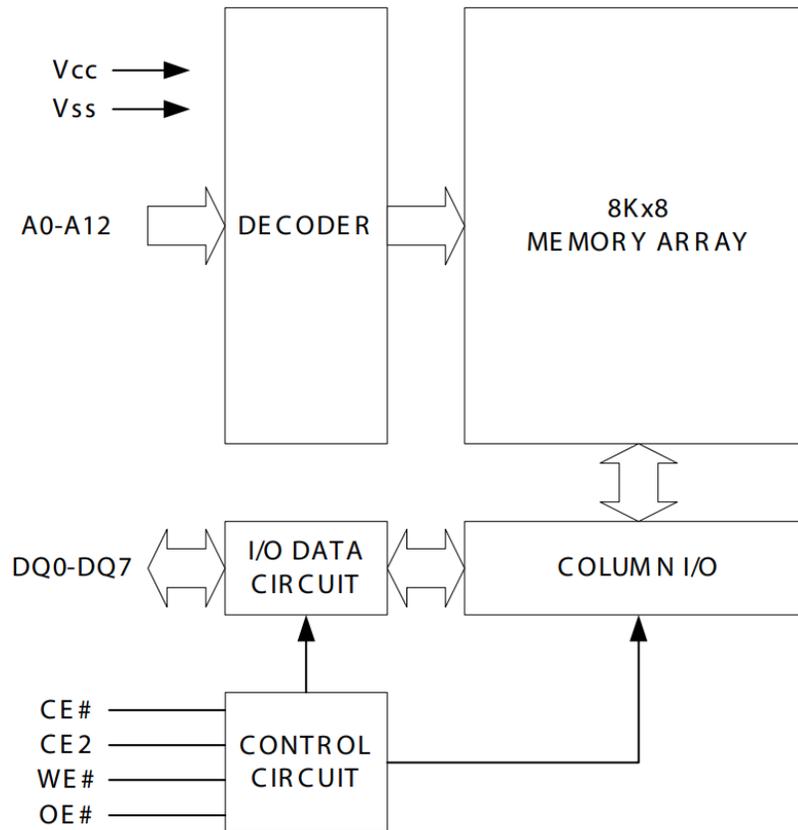
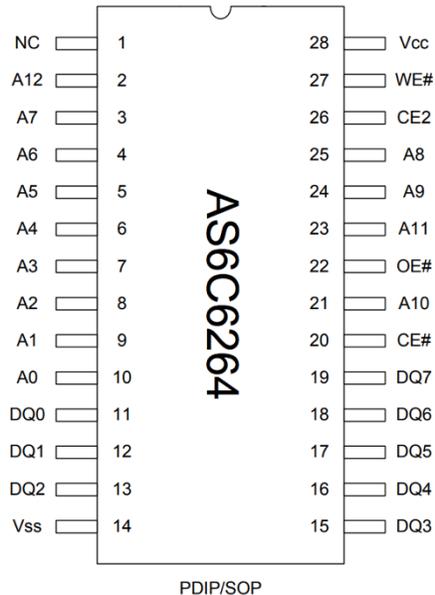
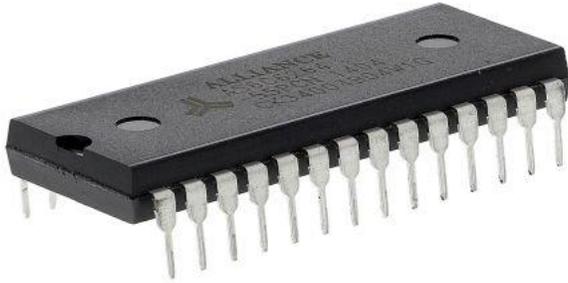
(a) SRAM block diagram



(b) SRAM read mechanism

Memoria SRAM nu este altceva decât o matrice foarte mare de registre single-port.

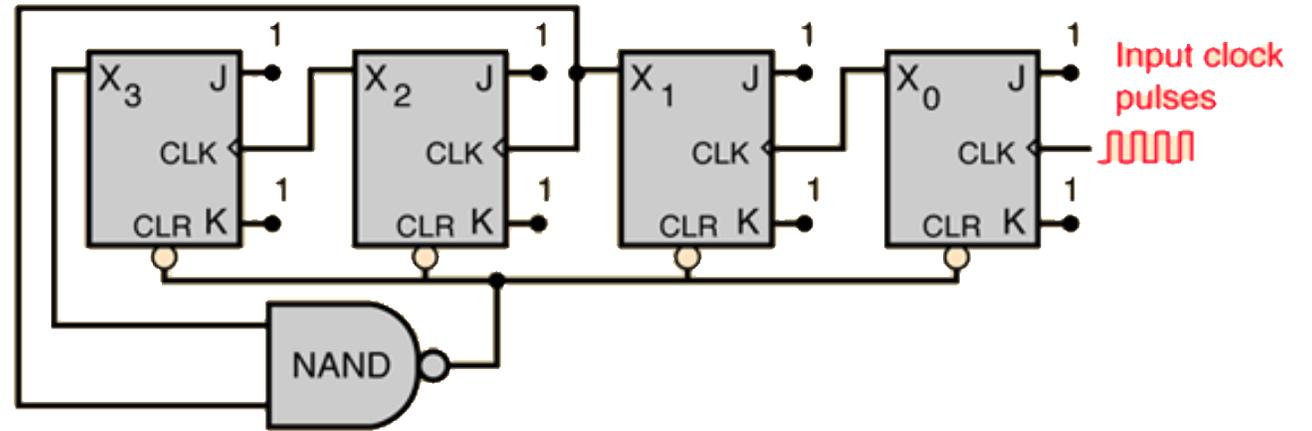
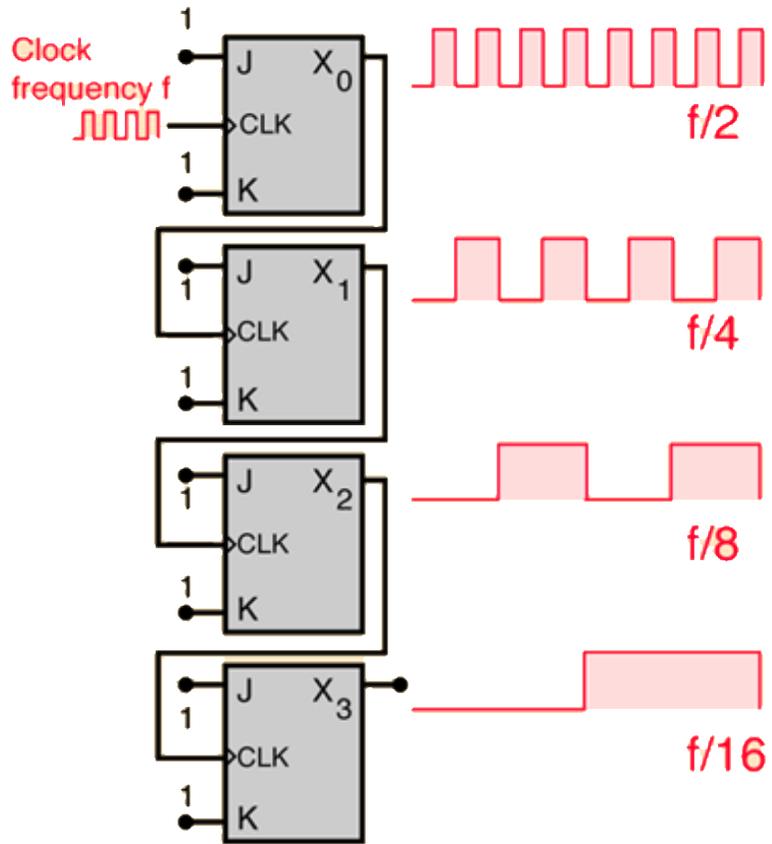
SRAM - exemplu



SYMBOL	DESCRIPTION
A0 - A12	Address Inputs
DQ0 – DQ7	Data Inputs/Outputs
CE#, CE2	Chip Enable Inputs
WE#	Write Enable Input
OE#	Output Enable Input
Vcc	Power Supply
Vss	Ground
NC	No Connection

8K X 8 BIT LOW POWER CMOS SRAM

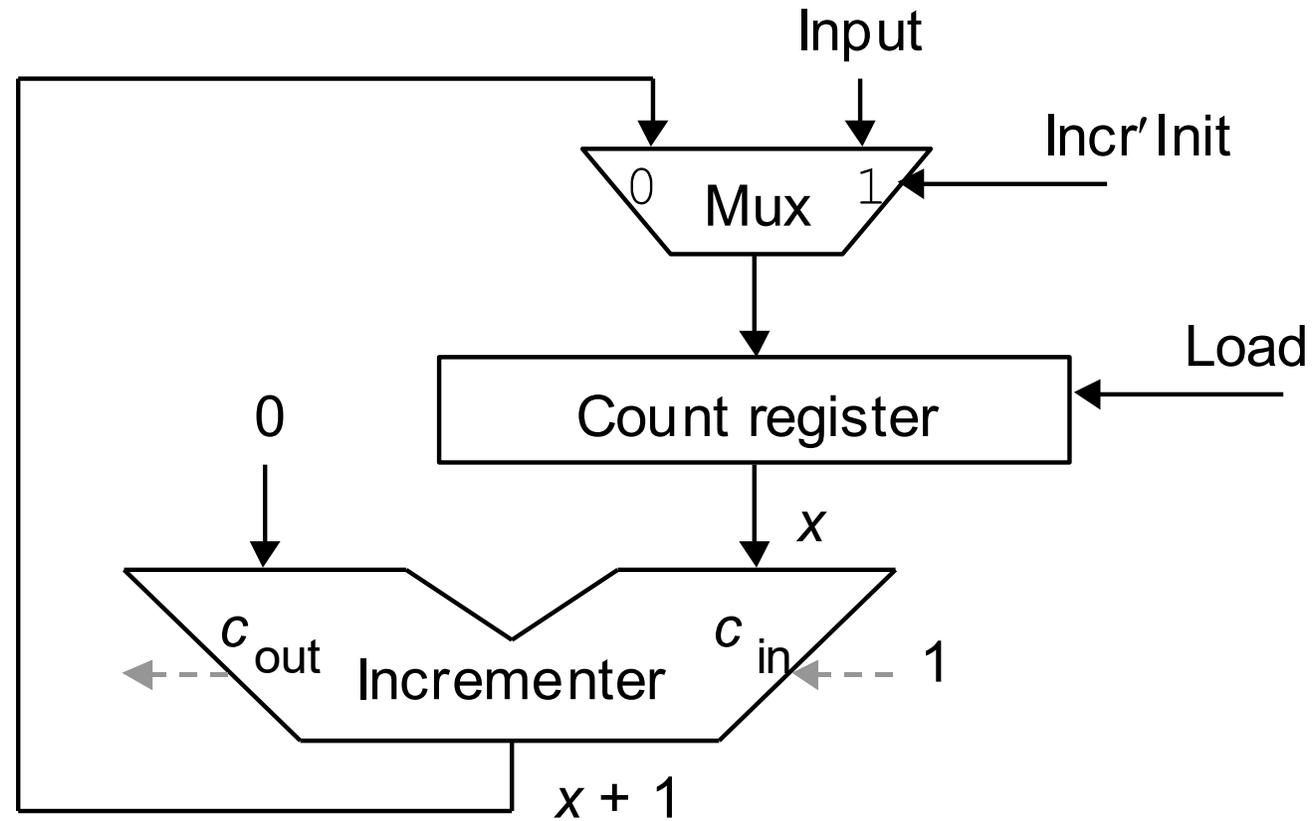
Numărătorul Binar



Numărător BCD – se resetează automat după ce ieșirile ajung la cifra 9 (1001)

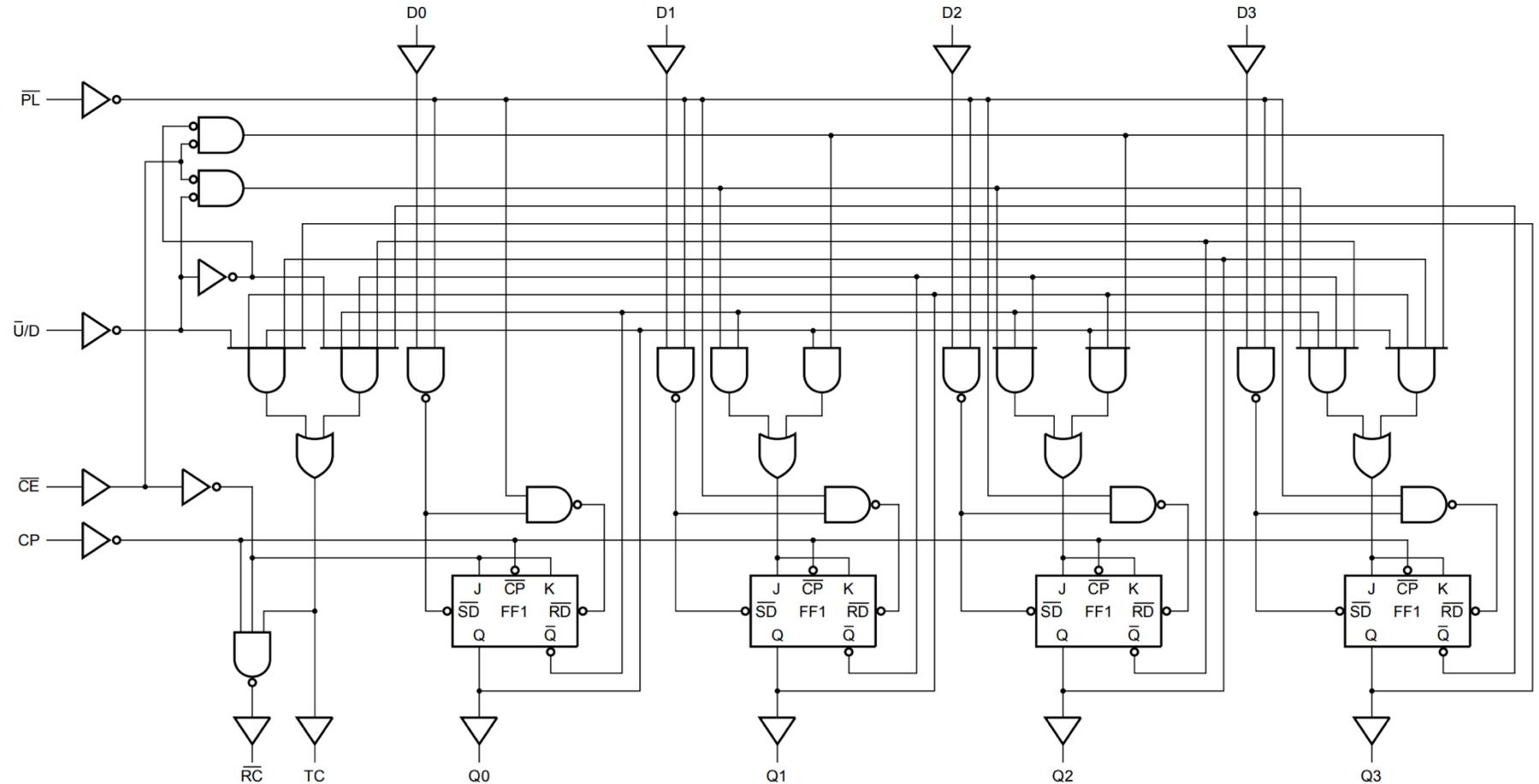
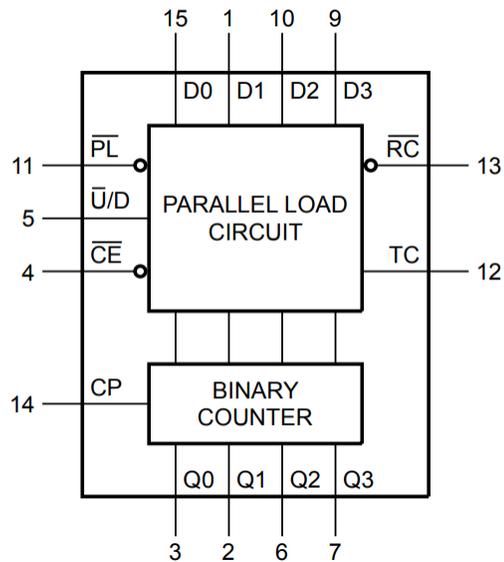
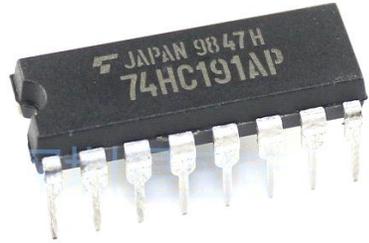
Divizor de frecvență / numărător binar

Numărătorul Binar



Numărător binar sincron cu capabilitate de inițializare.

Numărătorul Binar - exemplu



74HC191 - numărător binar presetabil cu numărare în sus sau jos



**THINKING YOU CAN
COUNT TO 10 ON YOUR
FINGERS**



**KNOWING YOU CAN
COUNT TO 1,023 ON
YOUR FINGERS**

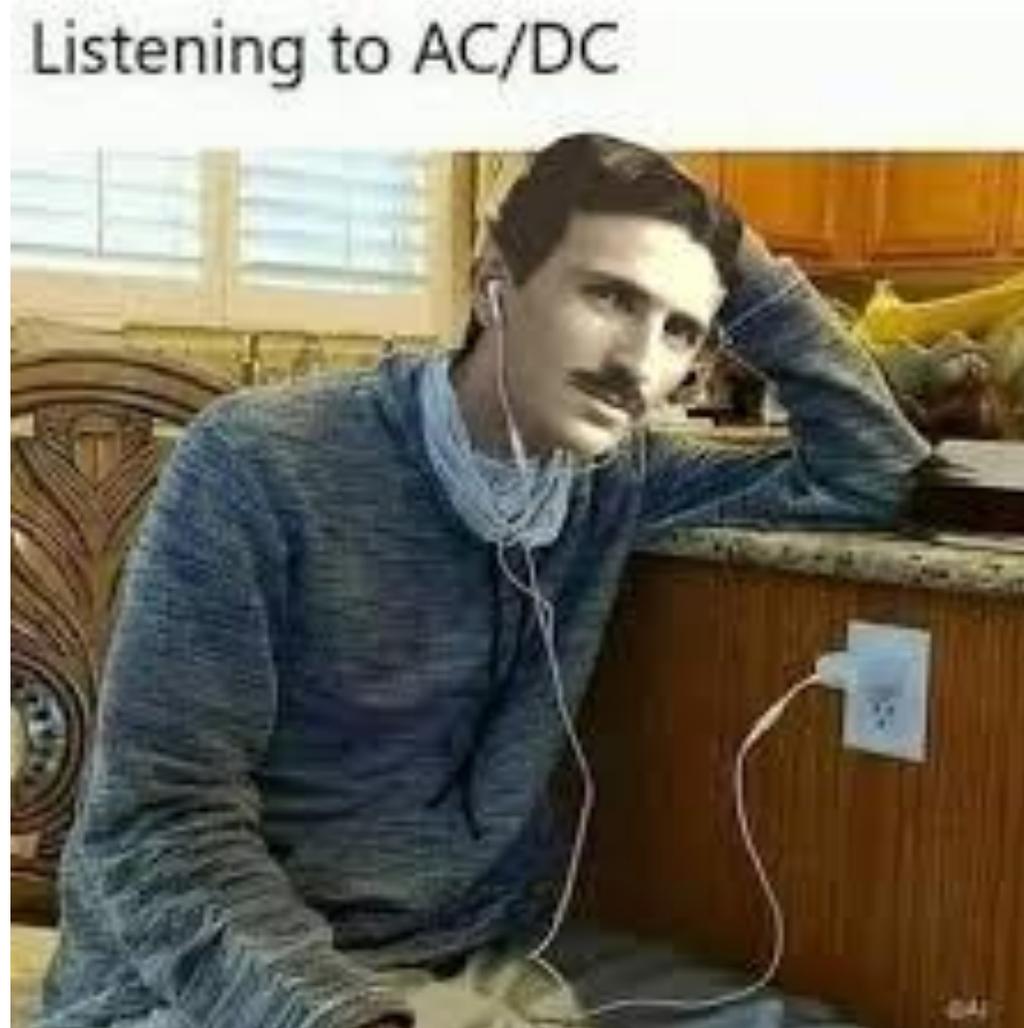
Niveluri Logice

- Valori discrete de tensiune reprezintă 1 și 0
- De exemplu:
 - 0 = *ground* (GND) sau 0 volți
 - 1 = V_{DD} sau 5 volți
- Dar 4.99 volți? Este un 0 sau un 1?
- Dar 3.2 volți?

Niveluri Logice

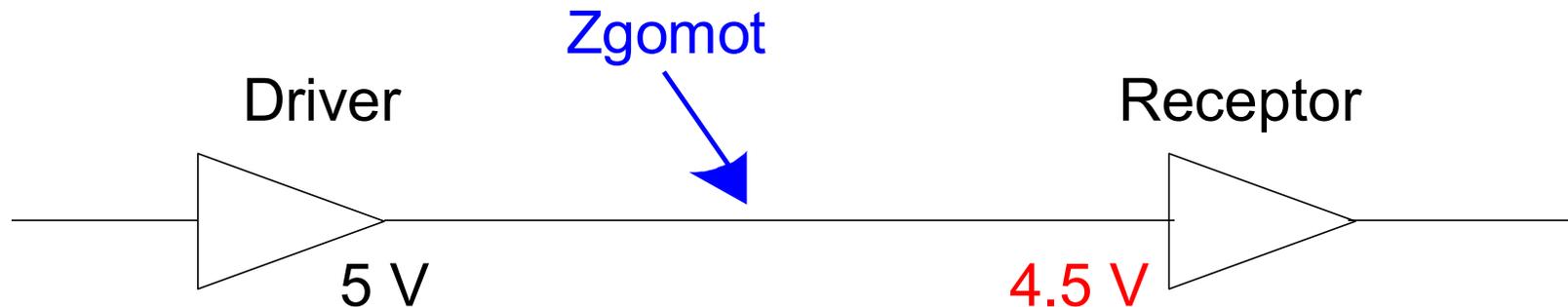
- *Domeniu* de tensiuni pentru 1 și 0
- Domenii diferite pentru intrări și ieșiri, pentru a lua în calcul și zgomotul

Ce este zgomotul?



Ce este zgomotul?

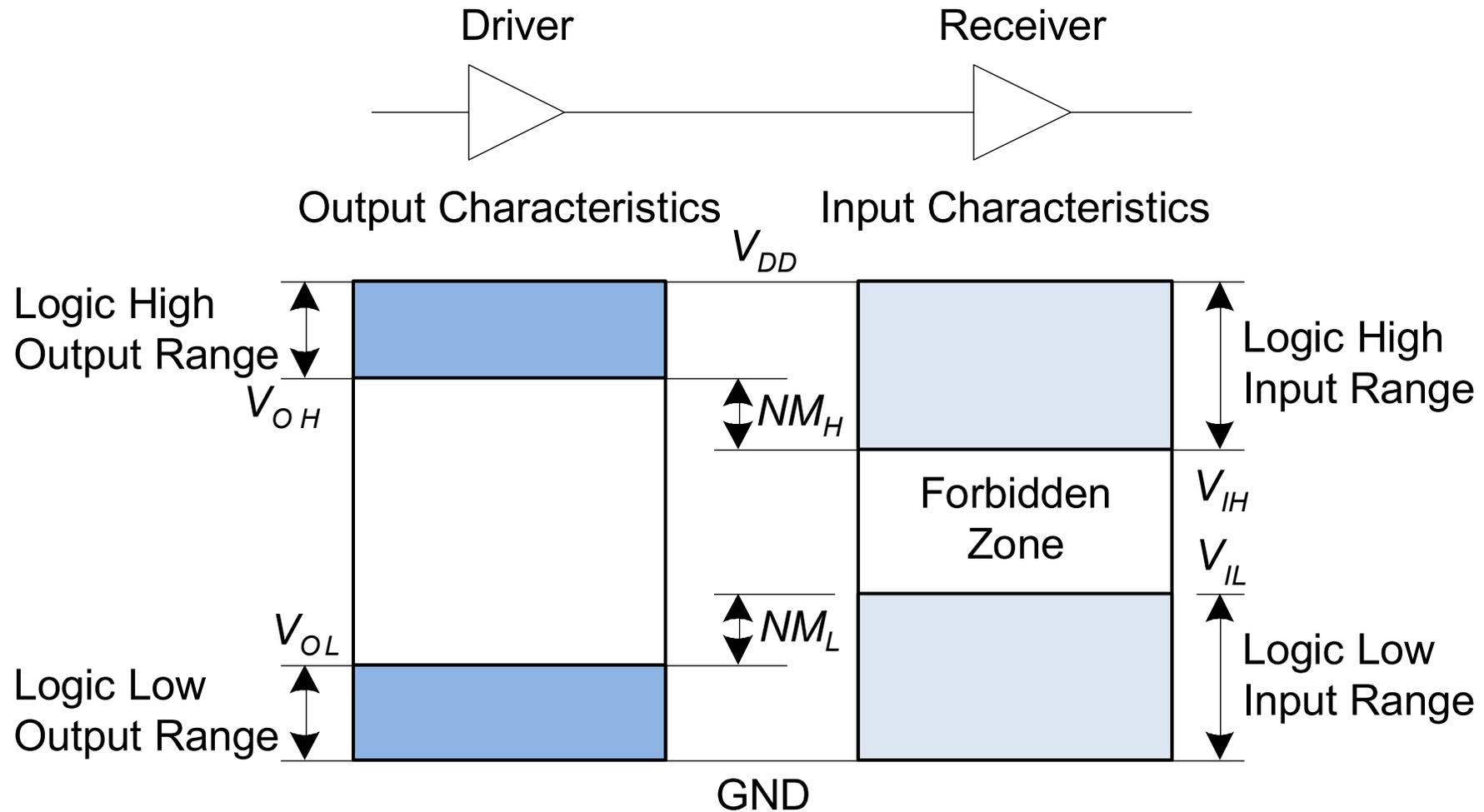
- **Orice degradează semnalul**
 - E.g., rezistența, zgomotul sursei de alimentare, cuplajele parazite de la conductorii învecinați etc.
- **Exemplu:** o poartă (driver) pune la ieșirea sa 5 V dar, din cauza rezistenței liniei lungi, receptorul primește 4.5 V



Analiza statică

- Pornind de la niște intrări logice valide, fiecare element de circuit trebuie să producă ieșiri valide din punct de vedere logic
- Trebuie să folosim o gamă restrânsă de tensiuni pentru a reprezenta valorile discrete

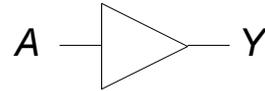
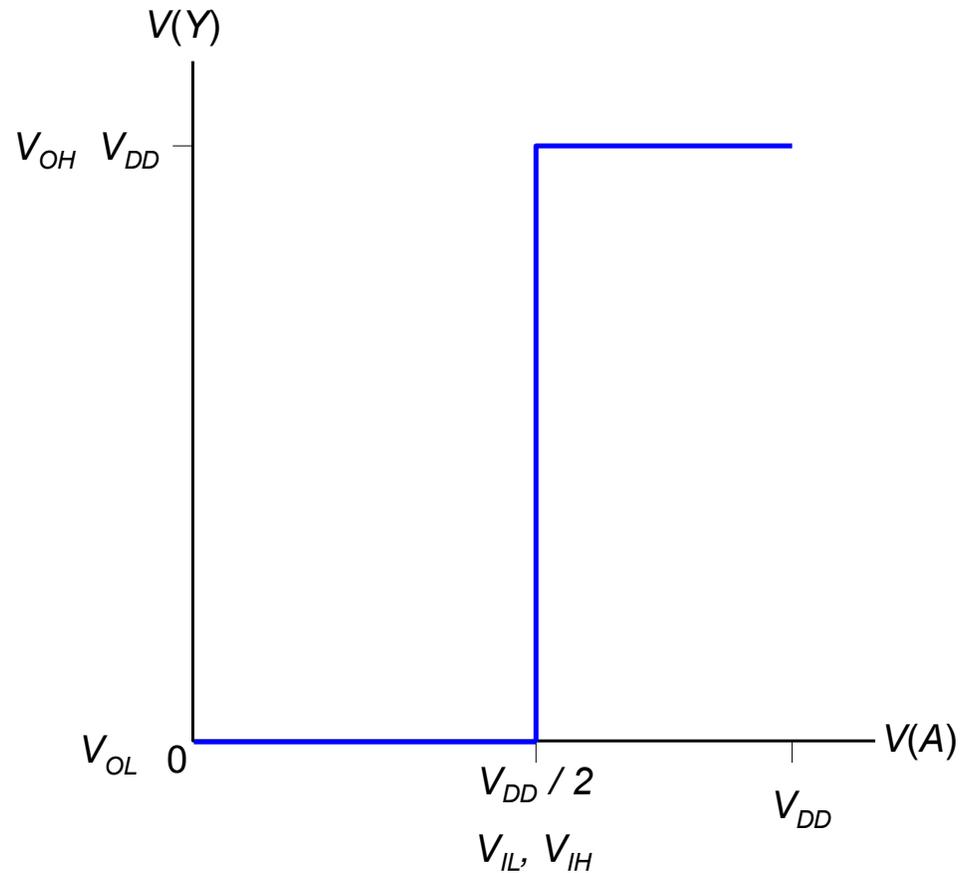
Niveluri logice



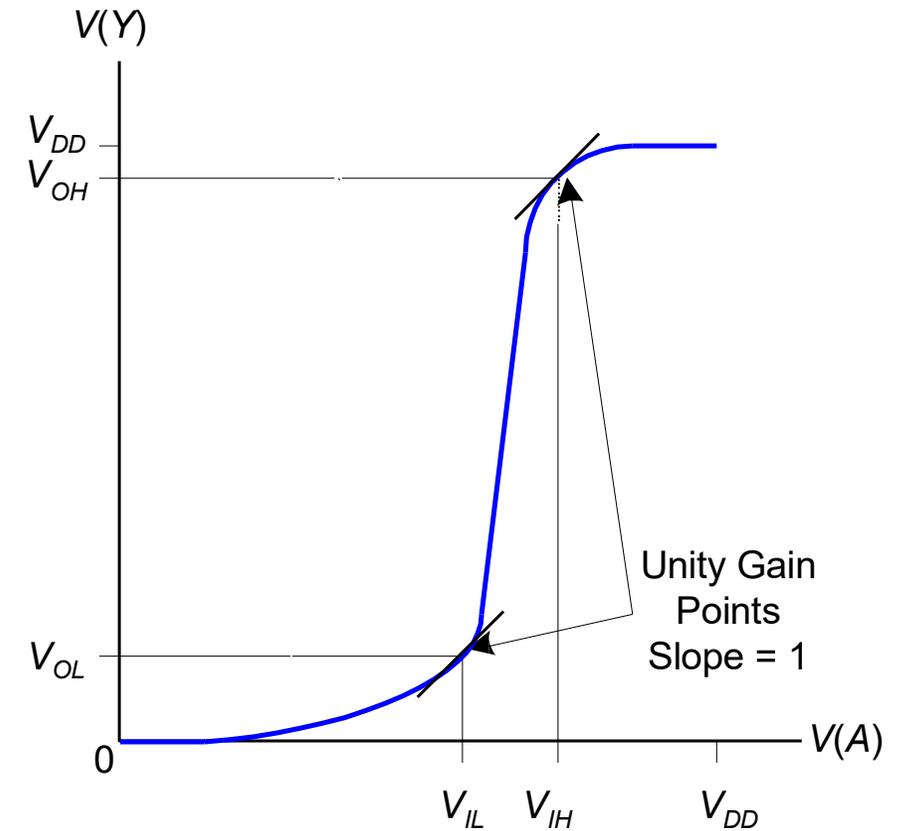
$$NM_H = V_{OH} - V_{IH} \quad NM_L = V_{IL} - V_{OL}$$

Caracteristici de transfer în c.c.

Buffer ideal:



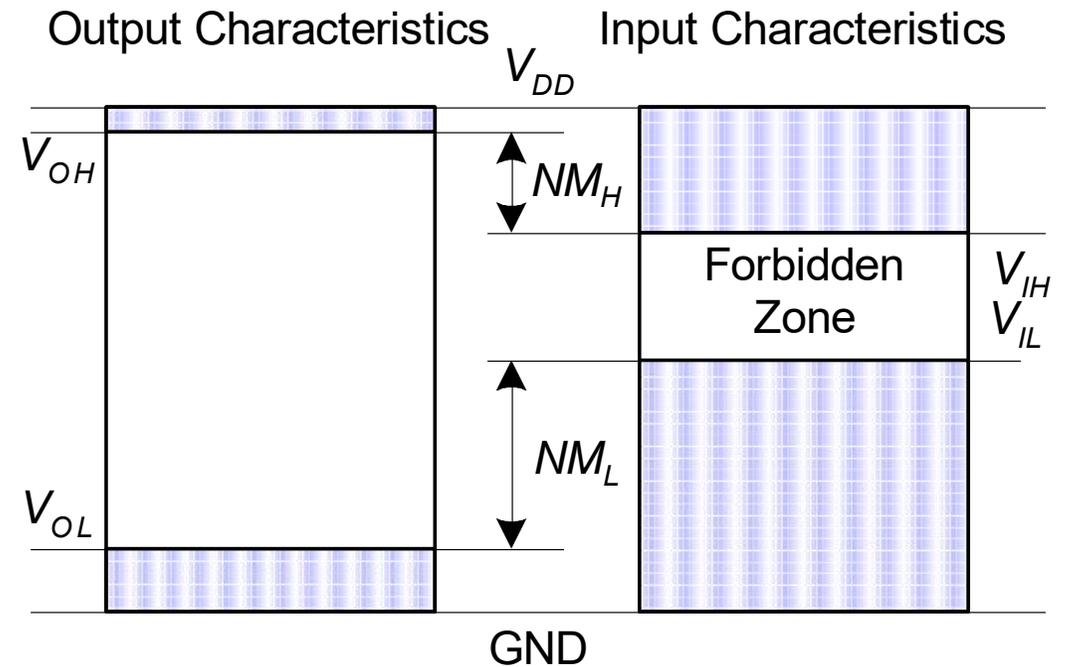
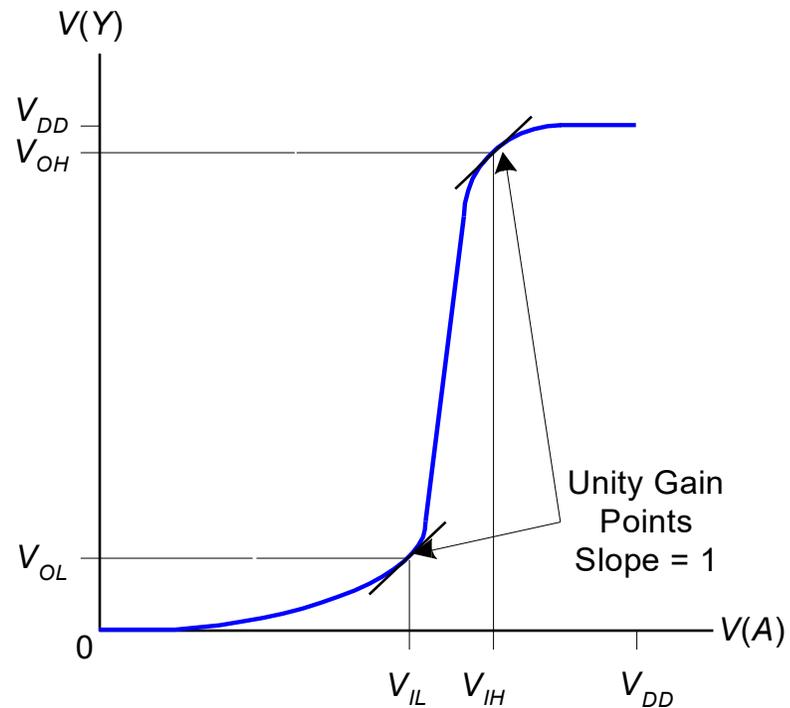
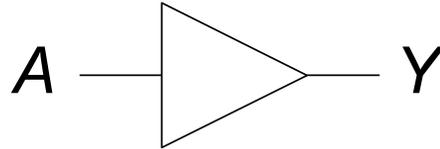
Buffer real:



$$NM_H = NM_L = V_{DD}/2$$

$$NM_H, NM_L < V_{DD}/2$$

Caracteristici de transfer în c.c.



Scalarea VDD

- In anii 1970 și 1980, $V_{DD} = 5\text{ V}$
- V_{DD} a scăzut
 - Tranzistoarele s-au micșorat
 - Consumă mai puțină putere
- 3.3 V, 2.5 V, 1.8 V, 1.5 V, 1.2 V, 1.0 V, ...
- Aveți grijă când conectați componente ce funcționează la diferite niveluri de tensiune



Teorie alternativă:

Circuitele funcționează pe bază de fum magic!

Dovada:

- Dacă iese fumul magic, circuitul nu mai funcționează!

Diferite familii de circuite logice

Familie Logică	V_{DD}	V_{IL}	V_{IH}	V_{OL}	V_{OH}
TTL	5 (4.75 - 5.25)	0.8	2.0	0.4	2.4
CMOS	5 (4.5 - 6)	1.35	3.15	0.33	3.84
LVTTL	3.3 (3 - 3.6)	0.8	2.0	0.4	2.4
LVC MOS	3.3 (3 - 3.6)	0.9	1.8	0.36	2.7

Consumul de putere

- Putere = Energia consumată pe unitatea de timp
 - Consum de putere în regim dinamic
 - Consum de putere în regim static

Puterea dinamică

- **Puterea necesară pentru a încărca capacitățile parazite ale porților**

Energia necesară pentru a încărca o capacitate C , la V_{DD} este CV_{DD}^2

- Circuitul lucrează la frecvența f : tranzistoarele comută (din 1 în 0 și vice versa) la acea frecvență
- Condensatorul este încărcat de $f/2$ ori pe secundă (descărcarea din 1 în 0 este gratis)

Puterea dinamică: $P_{\text{dynamic}} = \frac{1}{2}fCV_{DD}^2$

Puterea statică

- Puterea consumată când nici o poartă nu comută
- Căuzată de *curentul de mers în gol*, I_{DD}
(numit și curent de *leakage*)

Puterea statică: $P_{\text{static}} = I_{DD} V_{DD}$

Exemplu de calcul consum putere

- Estimați consumul de putere pentru un calculator portabil cu următoarele specificații:

- $V_{DD} = 1.2 \text{ V}$

- $C = 20 \text{ nF}$

- $f = 1 \text{ GHz}$

- $I_{DD} = 20 \text{ mA}$

$$\begin{aligned} P &= \frac{1}{2}CV_{DD}^2f + I_{DD}V_{DD} \\ &= \frac{1}{2}(20 \text{ nF})(1.2 \text{ V})^2(1 \text{ GHz}) + (20 \text{ mA})(1.2 \text{ V}) \\ &= \mathbf{14.4 \text{ W}} \end{aligned}$$

Acknowledgements

- Aceste slide-uri conțin materiale aparținând:
 - Arvind (MIT)
 - Krste Asanovic (MIT/UCB)
 - Joel Emer (Intel/MIT)
 - James Hoe (CMU)
 - John Kubiatowicz (UCB)
 - David Patterson (UCB)
 - Behrooz Parhami (UCSB)
- MIT material derived from course 6.823
- UCB material derived from course CS252