

# **Laboratorio di Elettronica e Tecniche di Acquisizione Dati 2024-2025**

**Elettronica digitale (2<sup>a</sup> parte)**

(cfr. <http://physics.ucsd.edu/~tmurphy/phys121/phys121.html>  
[https://en.wikipedia.org/wiki/Programmable\\_logic\\_device](https://en.wikipedia.org/wiki/Programmable_logic_device))

# Famiglie logiche

Famiglie logiche più diffuse e usate

- **CMOS** (Complementary MOS)
- **NMOS** (MOSFET a canale n)
- **TTL** (Transistor-Transistor Logic)
- **ECL** (Emitter Coupled Logic)

} transistor **FET**  
} transistor **BJT**

Le porte logiche possono essere fabbricate con le varie tecnologie in un singolo chip con stesse funzioni, compatibili

numero di porte



**SSI** small scale integration (1-10 gates)  
**MSI** medium scale integration (10-100 gates)  
**LSI** large scale integration ( $\sim 10^3$ )  
**VLSI** very large scale integration ( $\sim 10^6$ )  
**ULSI** ultra large scale integration ( $> 10^6$ )

# Famiglie logiche (le due più comuni)

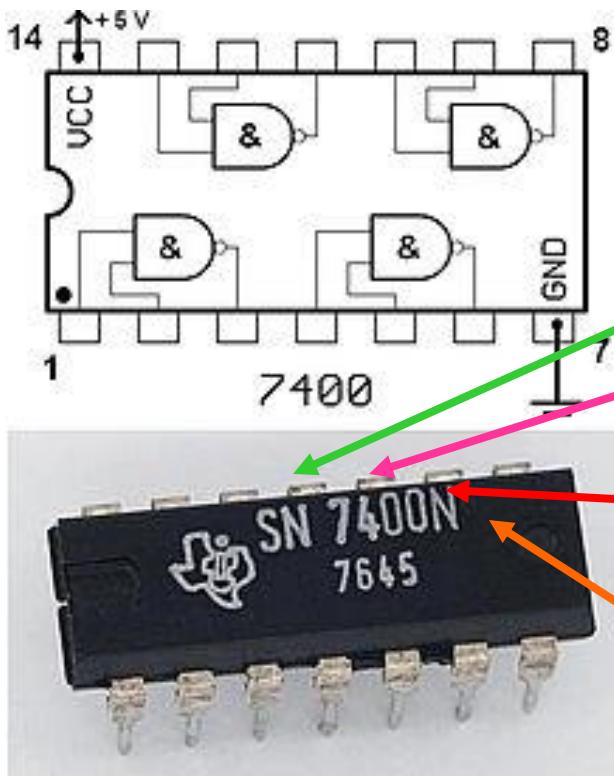
- **TTL**: Transistor-Transistor Logic, basato sul BJT
  - output: ‘1’ logico:  $V_{OH} > 3.3 \text{ V}$ ; ‘0’ logico:  $V_{OL} < 0.35 \text{ V}$
  - input: ‘1’ logico:  $V_{IH} > 2.0 \text{ V}$ ; ‘0’ logico:  $V_{IL} < 0.8 \text{ V}$
  - zona “morta” fra 0.8V e 2.0 V
- **CMOS**: Complimentary MOSFET, basato su FET
  - output: ‘1’ logico:  $V_{OH} > 4.7 \text{ V}$ ; ‘0’ logico:  $V_{OL} < 0.2 \text{ V}$
  - input: ‘1’ logico:  $V_{IH} > 3.7 \text{ V}$ ; ‘0’ logico:  $V_{IL} < 1.3 \text{ V}$
  - zona “morta” fra 1.3V e 3.7 V

L’uscita di un CMOS è TTL-compatibile

# Confronto famiglie logiche

	TTL (V)	CMOS (V)	ECL (V)
tensione massima di alimentazione	5	5	-5.2
valore massimo $V_{in}$ identificato come 0	0.8	1.3	-1.4
valore minimo $V_{in}$ identificato come 1	2.0	3.7	-1.2
valore massimo $V_{out}$ identificato come 0	0.35	0.2	-1.7
valore minimo $V_{out}$ identificato come 1	3.3	4.7	-0.9

# Nomenclatura circuiti



AA 74 AAA XXX P

due lettere indicano la casa costruttrice  
74, sempre uguale (porte logiche)

tre lettere che indicano la sottofamiglia

numeri indicano la funzione del circuito

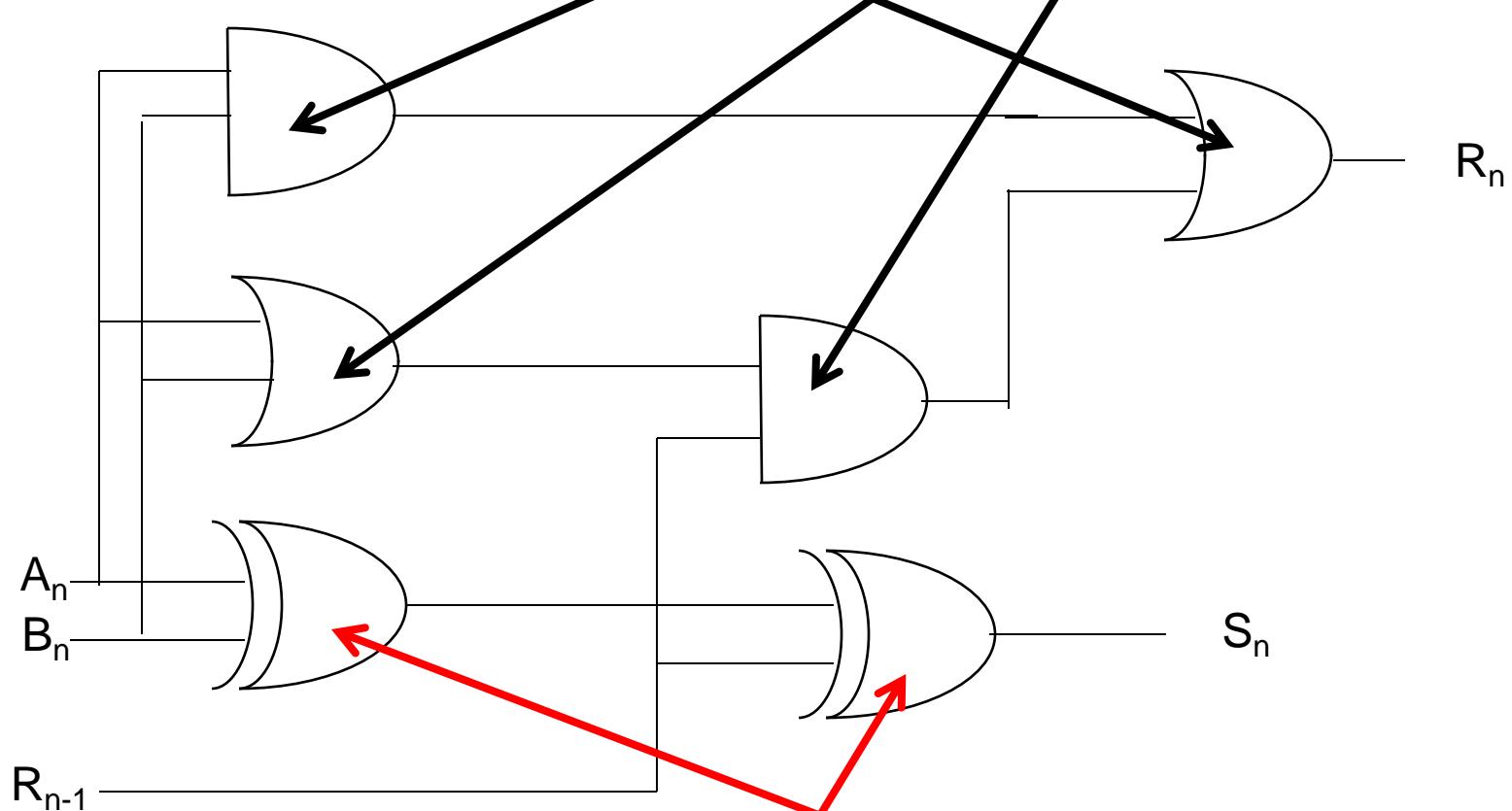
lettere che identificano il contenitore  
(packaging)

SN74ALS245N

significa che è fatto dalla Texas Instruments (SN), è una porta logica TTL con range di temperatura commerciale (74), è della famiglia “Advanced Low-power Schottky” (ALS), ed è un buffer bi-direzionale a 8 bit, in un package plastico di tipo through-hole DIP (N).

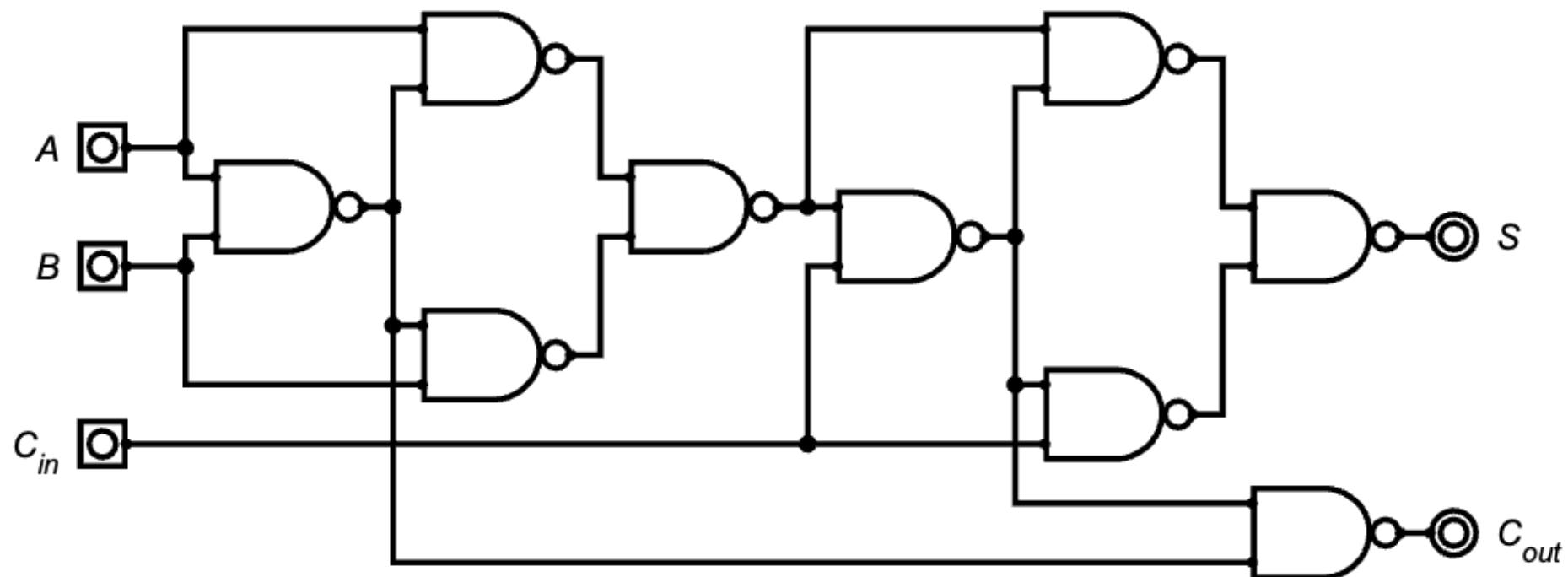
# Full Adder

$$R_n = A_n B_n + (A_n + B_n) R_{n-1}$$

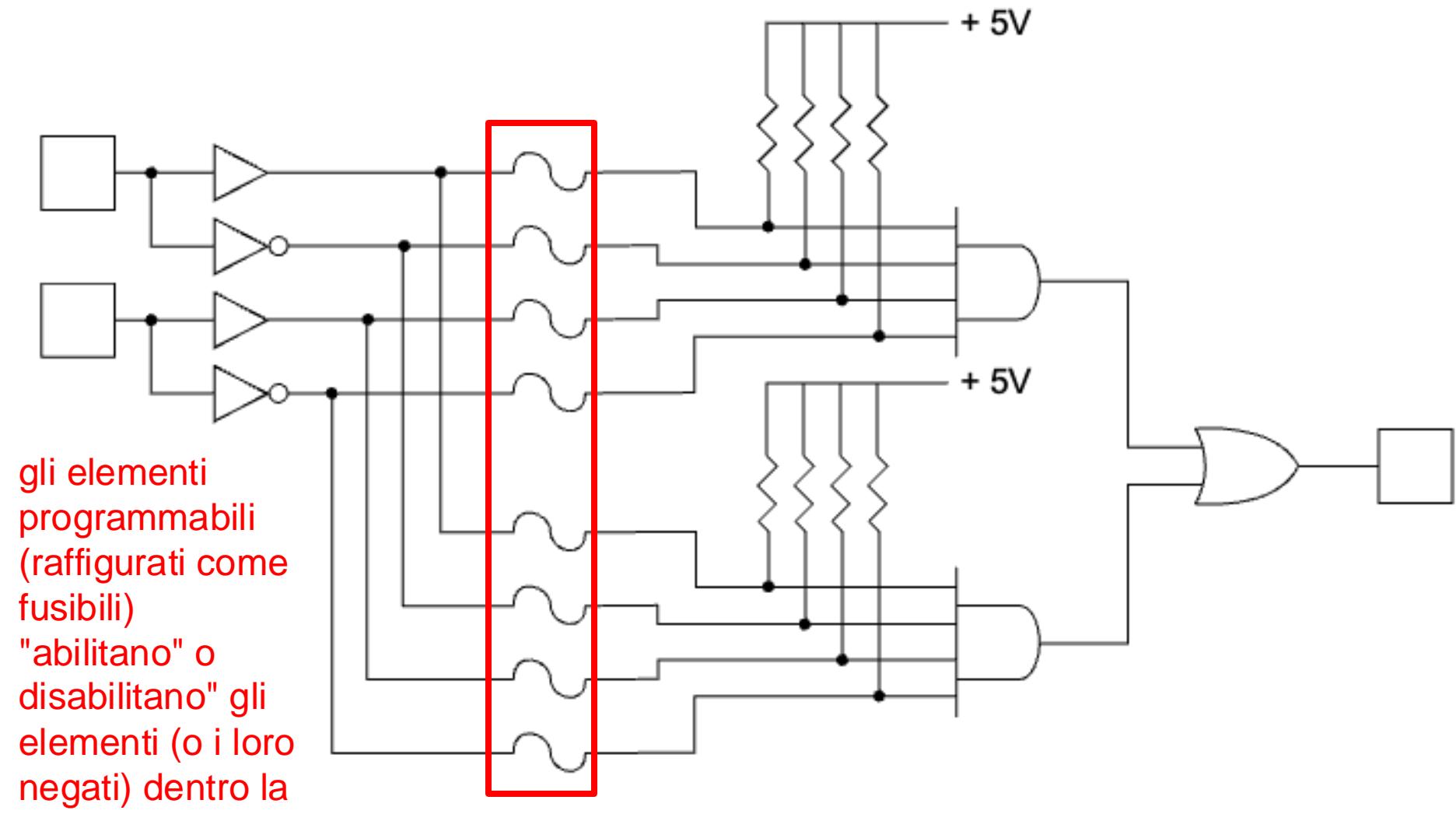


$$S_n = R_{n-1} \oplus (A_n \oplus B_n)$$

# Full Adder - NAND

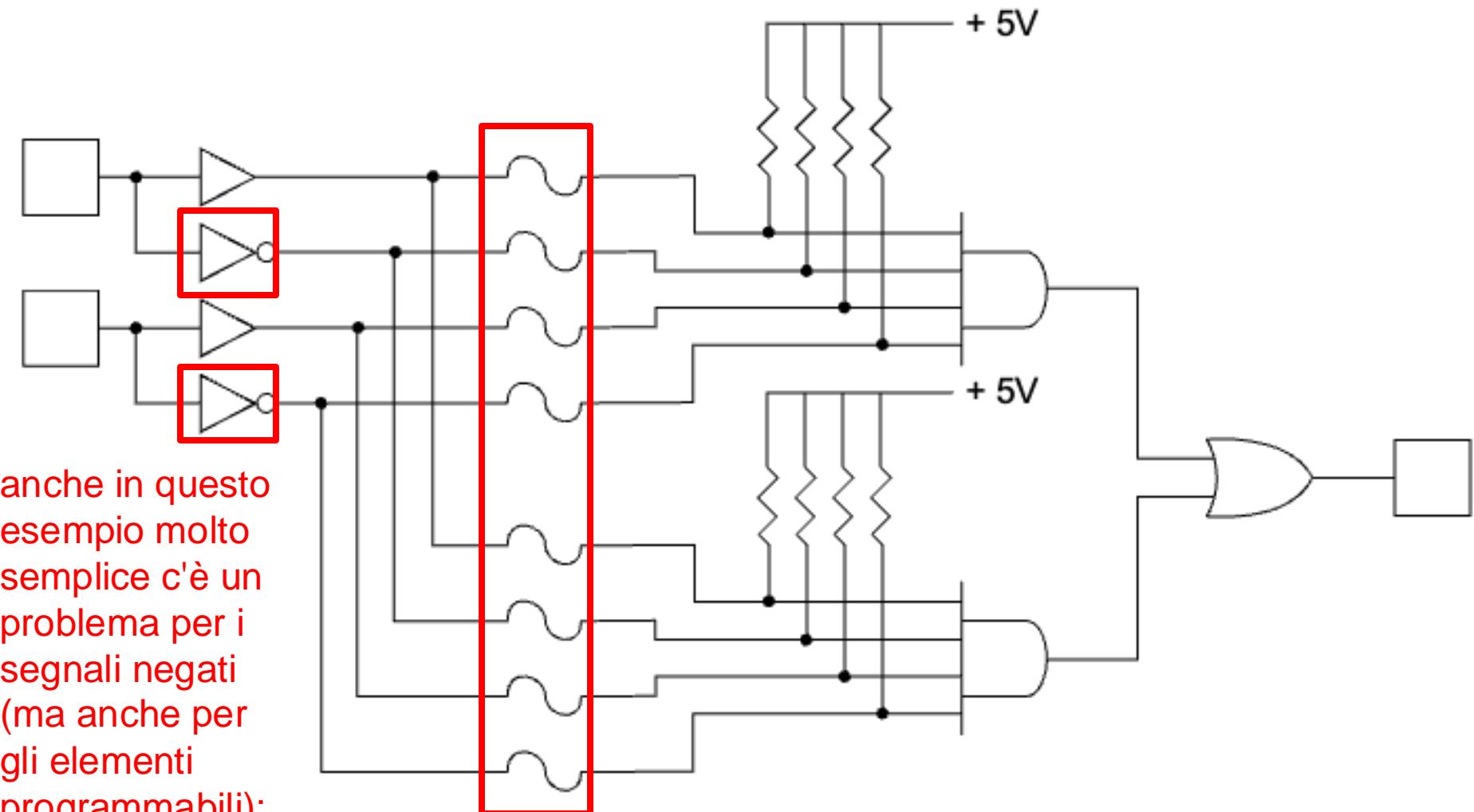


# Logica programmabile



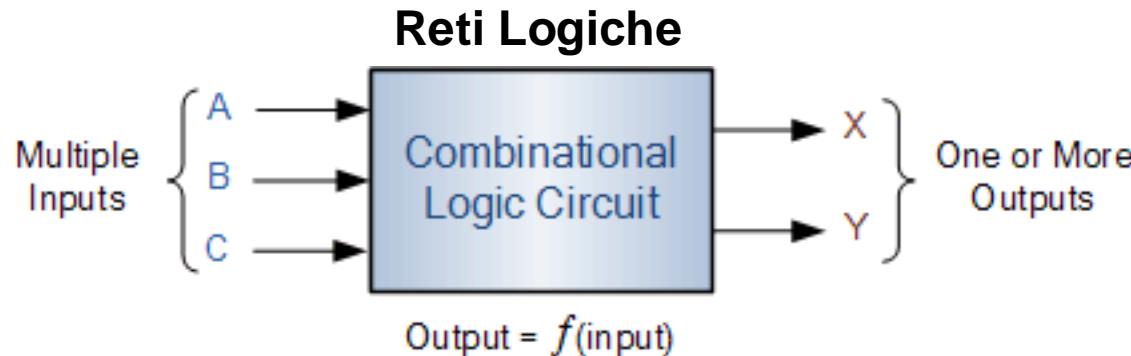
Simplified programmable logic device

# Logica programmabile

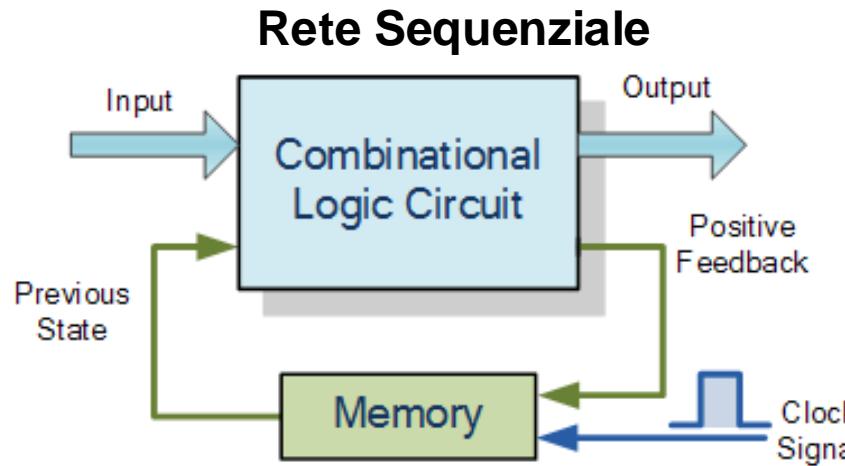


Simplified programmable logic device

# Reti Logiche e Sequenziali



- No feedback, in ogni istante l'output è funzione degli input
- Tutti i circuiti analizzati fino adesso sono reti logiche

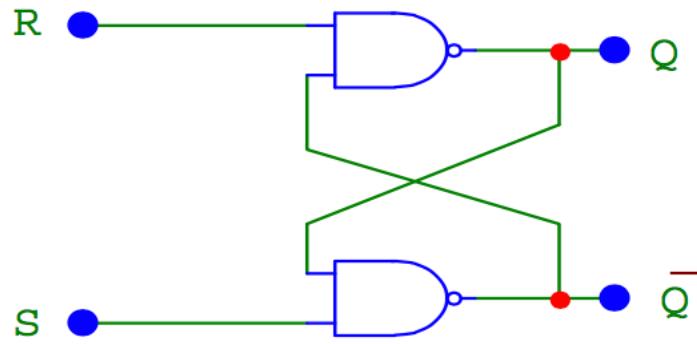


- L'output è funzione dell'input corrente e dell'output precedente
- I (possibili) cambi di output sono definiti dal segnale di **clock**
- Il circuito ha “**memoria**” dei suoi stati precedenti

# Flip Flop SR (NAND)

- **FLIP FLOP**: unità di memoria fondamentale dei circuiti digitali. Elemento di base dei circuiti sequenziali. È un circuito che immagazzina l'informazione di base, (bit, 0 o 1)

## SR Flip Flop (Set – Reset)



R	S	$Q_n$	$\bar{Q}_n$
0	0	non consentito	
0	1	1	0
1	0	0	1
1	1	$Q_{n-1}$	$\bar{Q}_{n-1}$

stato “set”

stato “reset”

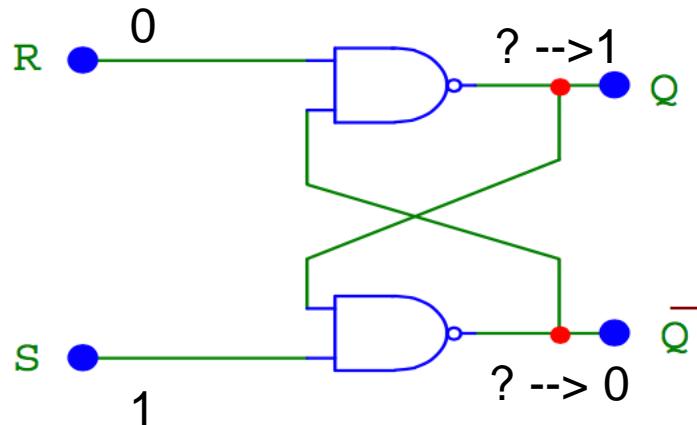
- Asincrono: il cambio di stato dell'output avviene in corrispondenza al cambio di stato degli input

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

# Flip Flop SR (NAND)

- **FLIP FLOP**: unità di memoria fondamentale dei circuiti digitali. Elemento di base dei circuiti sequenziali. È un circuito che immagazzina l'informazione di base, (bit, 0 o 1)

## SR Flip Flop (Set – Reset)



R	S	$Q_n$	$\bar{Q}_n$
0	0	non consentito	
0	1	1	0
1	0	0	1
1	1	$Q_{n-1}$	$\bar{Q}_{n-1}$

stato “set”

stato “reset”

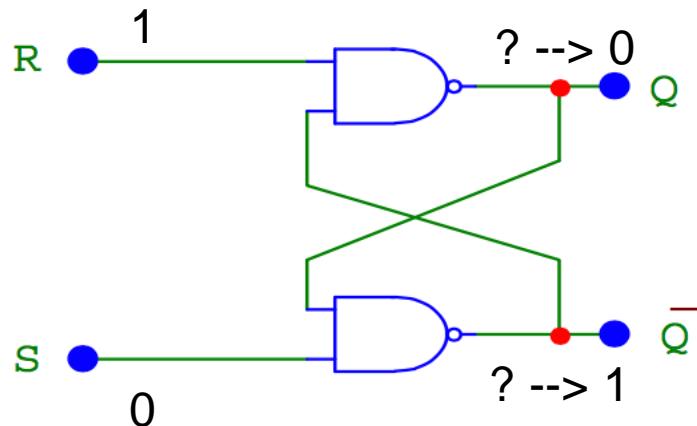
- Asincrono: il cambio di stato dell'output avviene in corrispondenza al cambio di stato degli input

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

# Flip Flop SR (NAND)

- **FLIP FLOP**: unità di memoria fondamentale dei circuiti digitali. Elemento di base dei circuiti sequenziali. È un circuito che immagazzina l'informazione di base, (bit, 0 o 1)

## SR Flip Flop (Set – Reset)



R	S	$Q_n$	$\bar{Q}_n$
0	0	non consentito	
0	1	1	0
1	0	0	1
1	1	$Q_{n-1}$	$\bar{Q}_{n-1}$

stato “set”

stato “reset”

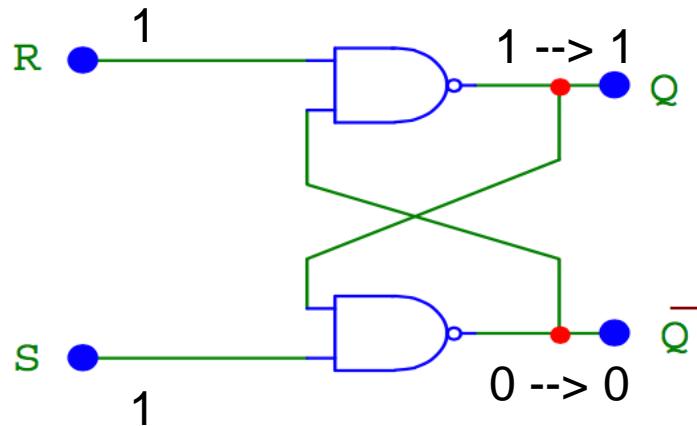
- Asincrono: il cambio di stato dell'output avviene in corrispondenza al cambio di stato degli input

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

# Flip Flop SR (NAND)

- **FLIP FLOP**: unità di memoria fondamentale dei circuiti digitali. Elemento di base dei circuiti sequenziali. È un circuito che immagazzina l'informazione di base, (bit, 0 o 1)

## SR Flip Flop (Set – Reset)



R	S	$Q_n$	$\bar{Q}_n$
0	0	non consentito	
0	1	1	0
1	0	0	1
1	1	$Q_{n-1}$	$\bar{Q}_{n-1}$

stato “set”

stato “reset”

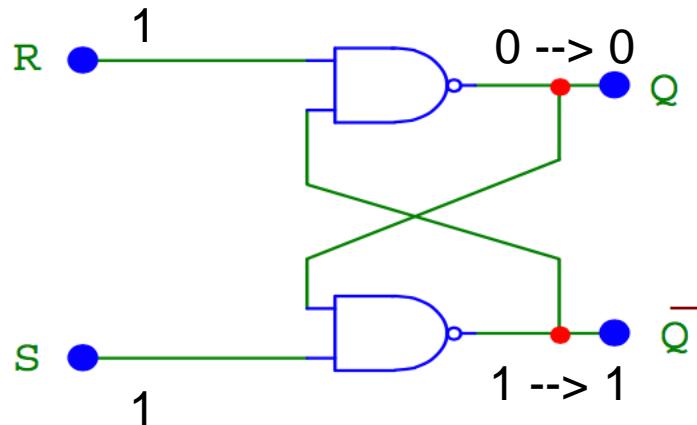
- Asincrono: il cambio di stato dell'output avviene in corrispondenza al cambio di stato degli input

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

# Flip Flop SR (NAND)

- **FLIP FLOP**: unità di memoria fondamentale dei circuiti digitali. Elemento di base dei circuiti sequenziali. È un circuito che immagazzina l'informazione di base, (bit, 0 o 1)

## SR Flip Flop (Set – Reset)



R	S	$Q_n$	$\bar{Q}_n$
0	0	non consentito	
0	1	1	0
1	0	0	1
1	1	$Q_{n-1}$	$\bar{Q}_{n-1}$

stato “set”

stato “reset”

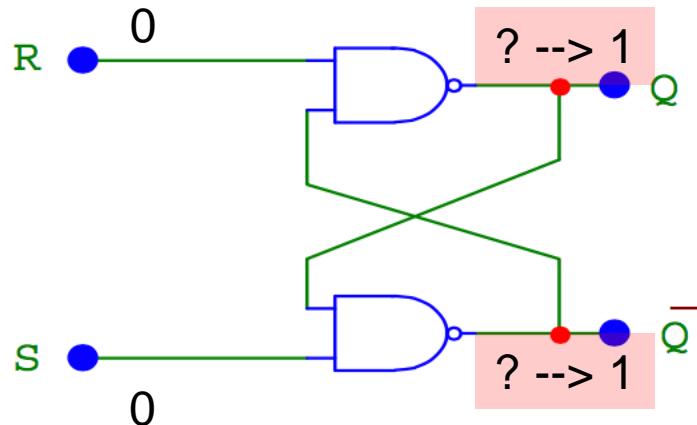
- Asincrono: il cambio di stato dell'output avviene in corrispondenza al cambio di stato degli input

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

# Flip Flop SR (NAND)

- **FLIP FLOP**: unità di memoria fondamentale dei circuiti digitali. Elemento di base dei circuiti sequenziali. È un circuito che immagazzina l'informazione di base, (bit, 0 o 1)

## SR Flip Flop (Set – Reset)



R	S	$Q_n$	$\bar{Q}_n$
0	0	non consentito	
0	1	1	0
1	0	0	1
1	1	$Q_{n-1}$	$\bar{Q}_{n-1}$

stato "set"

stato "reset"

- Asincrono: il cambio di stato dell'output avviene in corrispondenza al cambio di stato degli input
- Stato  $R=0, S=0$  proibito:  
 $Q_n = 1 \quad \bar{Q}_n = 1$  (i.e. non è vero che " $Q \neq \bar{Q}$ ") : situazione anomala, è necessario evitare che il FF sia in questo stato

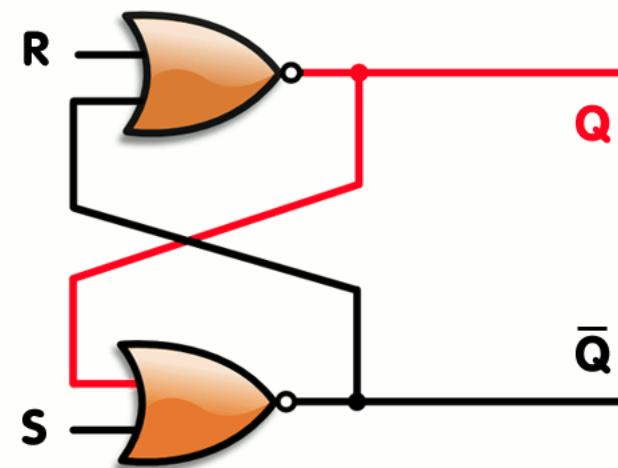
# Flip Flop SR (NOR)

## SR NOR latch [edit]

While the R and S inputs are both low, **feedback** maintains the Q and  $\bar{Q}$  outputs in a constant state, with  $\bar{Q}$  the complement of Q. If S (*Set*) is pulsed high while R (*Reset*) is held low, then the Q output is forced high, and stays high when S returns to low; similarly, if R is pulsed high while S is held low, then the Q output is forced low, and stays low when R returns to low.

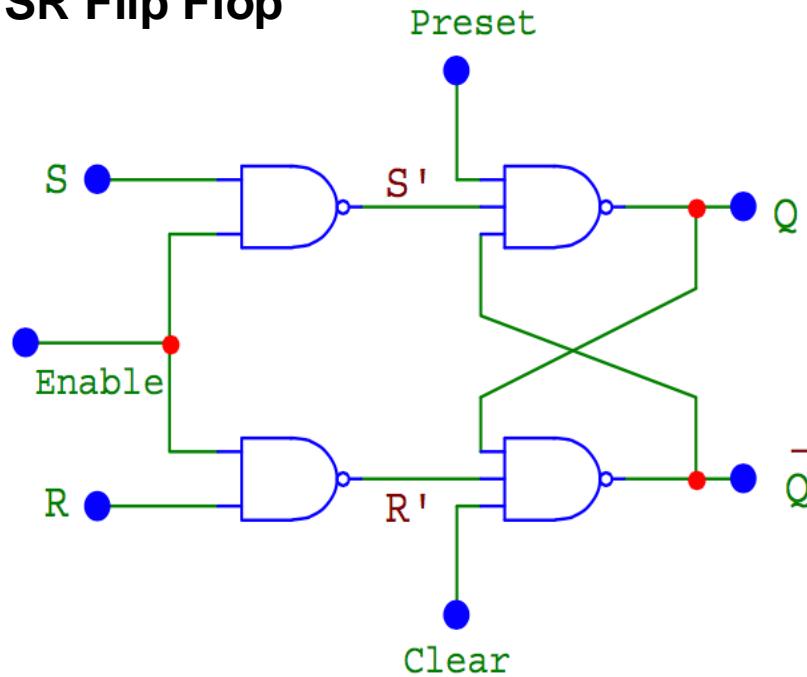
SR latch operation<sup>[3]</sup>

Characteristic table			Excitation table				
S	R	$Q_{next}$	Action	Q	$Q_{next}$	S	R
0	0	Q	Hold state	0	0	0	X
0	1	0	Reset	0	1	1	0
1	0	1	Set	1	0	0	1
1	1	X	Not allowed	1	1	X	0



# Flip Flop SR - gated

## SR Flip Flop



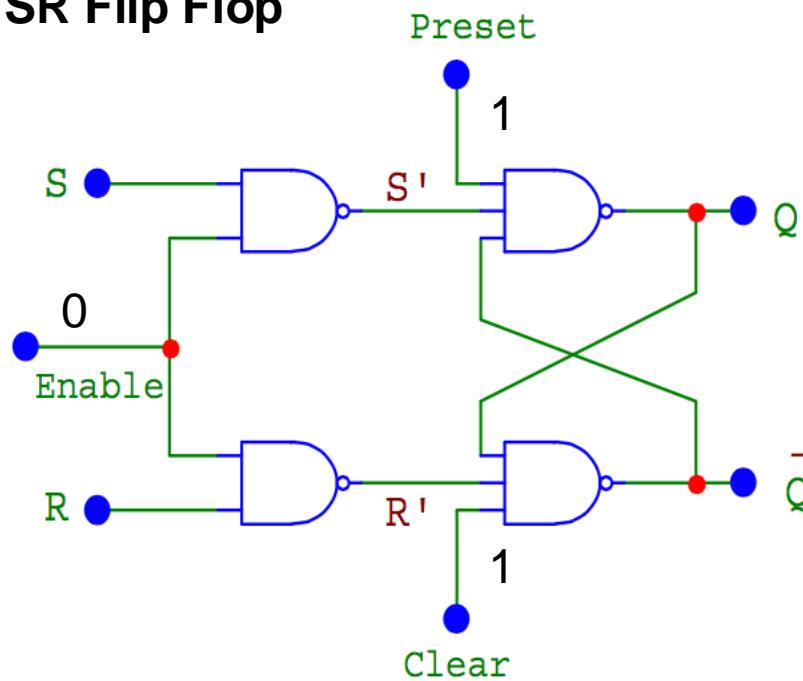
R	S	R'	S'	$Q_n$	$\bar{Q}_n$
x	x	1	1	$Q_{n-1}$	$\bar{Q}_{n-1}$
0	1	1	0	1	0
1	0	0	1	0	1
1	1			non consentito	

**ENABLE:** gate che abilita la porta:

**PRESET** e **CLEAR**: gate per definire lo stato iniziale del FF

# Flip Flop SR - gated

SR Flip Flop

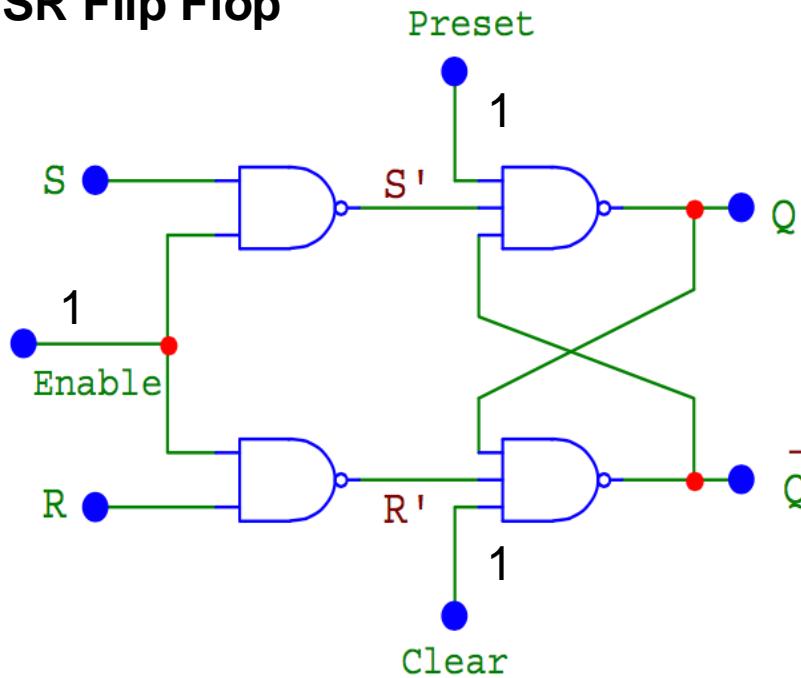


R	S	$R'$	$S'$	$Q_n$	$\bar{Q}_n$
x	x	1	1	$Q_{n-1}$	$\bar{Q}_{n-1}$
0	1	1	0	1	0
1	0	0	1	0	1
1	1			non consentito	

**ENABLE = 0:**  $S'=R'=1$ : FF Mantiene lo stato attuale, non risponde a variazioni di S e R

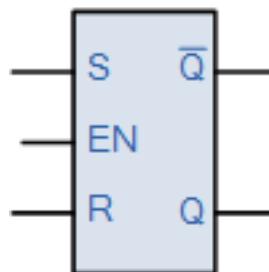
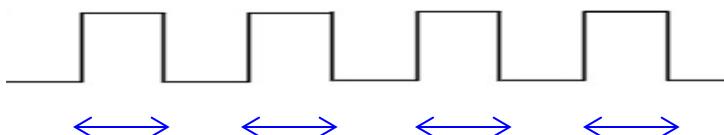
# Flip Flop SR - gated

**SR Flip Flop**



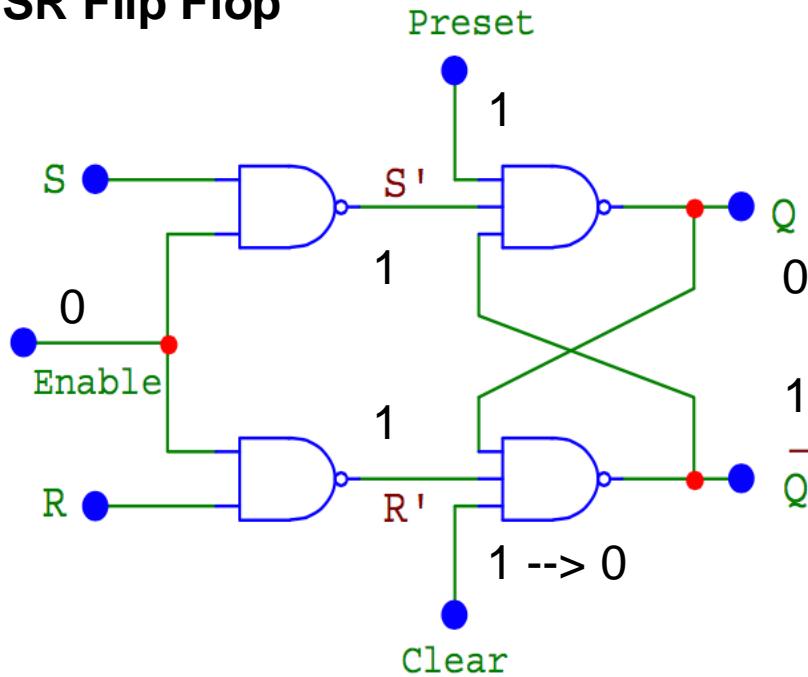
R	S	$R'$	$S'$	$Q_n$	$\bar{Q}_n$
0	0	1	1	$Q_{n-1}$	$\bar{Q}_{n-1}$
0	1	1	0	1	0
1	0	0	1	0	1
1	1			non consentito	

- Il gate EN permette di controllare quando il FF può cambiare stato: quando EN=0, l'uscita del FF **“memorizza”** l'output definito nel tempo in cui EN=1
- Sincrono:** il cambio di stato di Q e  $\bar{Q}$  avviene solamente quando il segnale di enable è positivo



# Flip Flop SR - gated

**SR Flip Flop**

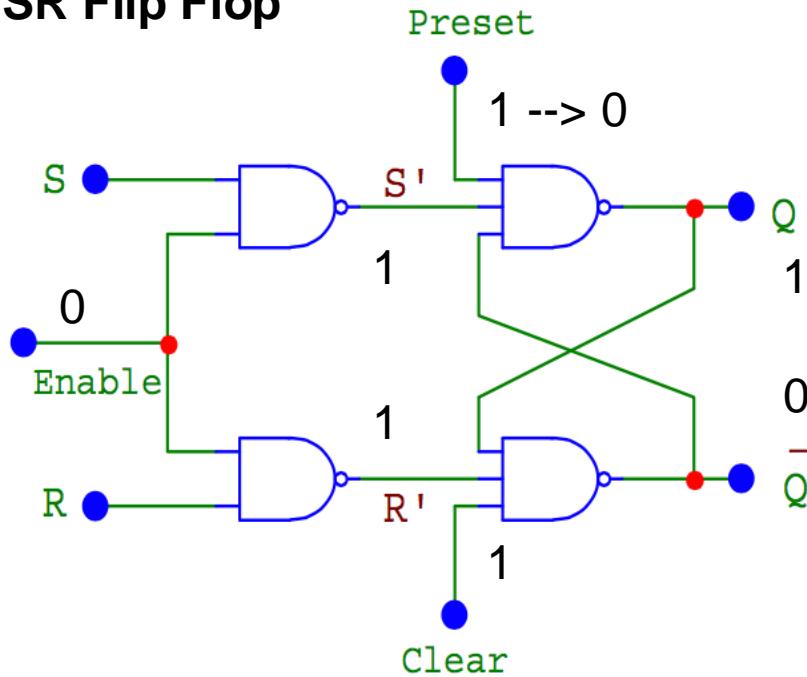


R	S	$R'$	$S'$	$Q_n$	$\bar{Q}_n$
0	0	1	1	$Q_{n-1}$	$\bar{Q}_{n-1}$
0	1	1	0	1	0
1	0	0	1	0	1
1	1			non consentito	

- Il gate EN permette di controllare quando il FF può cambiare stato: quando EN=0, l'uscita del FF **“memorizza”** l'output definito nel tempo in cui EN=1
- Sincrono:** il cambio di stato di Q e  $\bar{Q}$  avviene solamente quando il segnale di enable è positivo

# Flip Flop SR - gated

**SR Flip Flop**

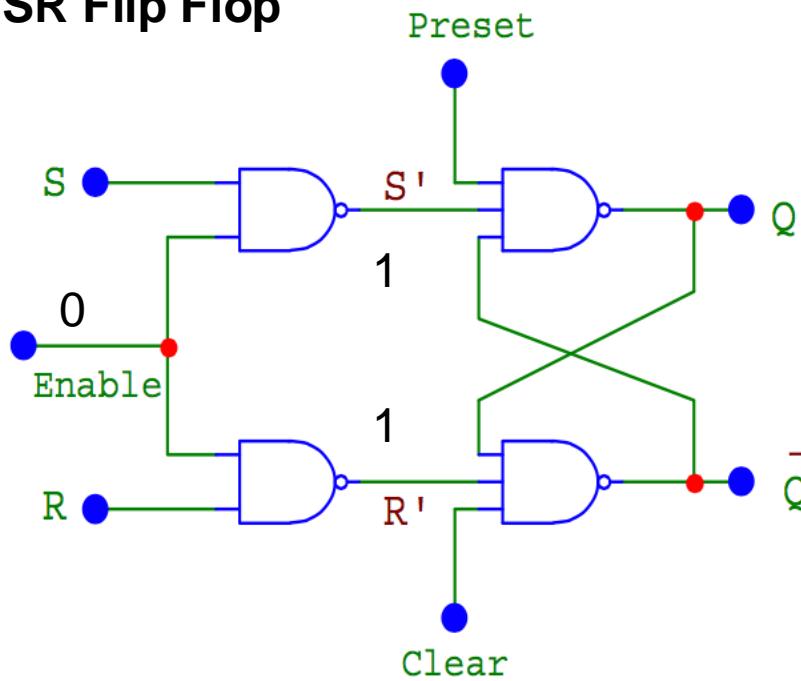


R	S	R'	S'	$Q_n$	$\bar{Q}_n$
0	0	1	1	$Q_{n-1}$	$\bar{Q}_{n-1}$
0	1	1	0	1	0
1	0	0	1	0	1
1	1			non consentito	

- Il gate EN permette di controllare quando il FF può cambiare stato: quando EN=0, l'uscita del FF **“memorizza”** l'output definito nel tempo in cui EN=1
- Sincrono:** il cambio di stato di Q e  $\bar{Q}$  avviene solamente quando il segnale di enable è positivo

# Flip Flop SR - gated

**SR Flip Flop**



R	S	R'	S'	$Q_n$	$\bar{Q}_n$
0	0	1	1	$Q_{n-1}$	$\bar{Q}_{n-1}$
0	1	1	0	1	0
1	0	0	1	0	1
1	1			non consentito	

- Il gate EN permette di controllare quando il FF può cambiare stato: quando EN=0, l'uscita del FF **“memorizza”** l'output definito nel tempo in cui EN=1
- Sincrono:** il cambio di stato di Q e  $\bar{Q}$  avviene solamente quando il segnale di enable è positivo
- Gli ingressi di **Preset** e **Clear** devono essere tenuti alti durante il funzionamento. Possono essere usati per definire lo stato iniziale del FF quando il segnale EN è basso (EN=0)

# Flip Flop D

## D Flip Flop

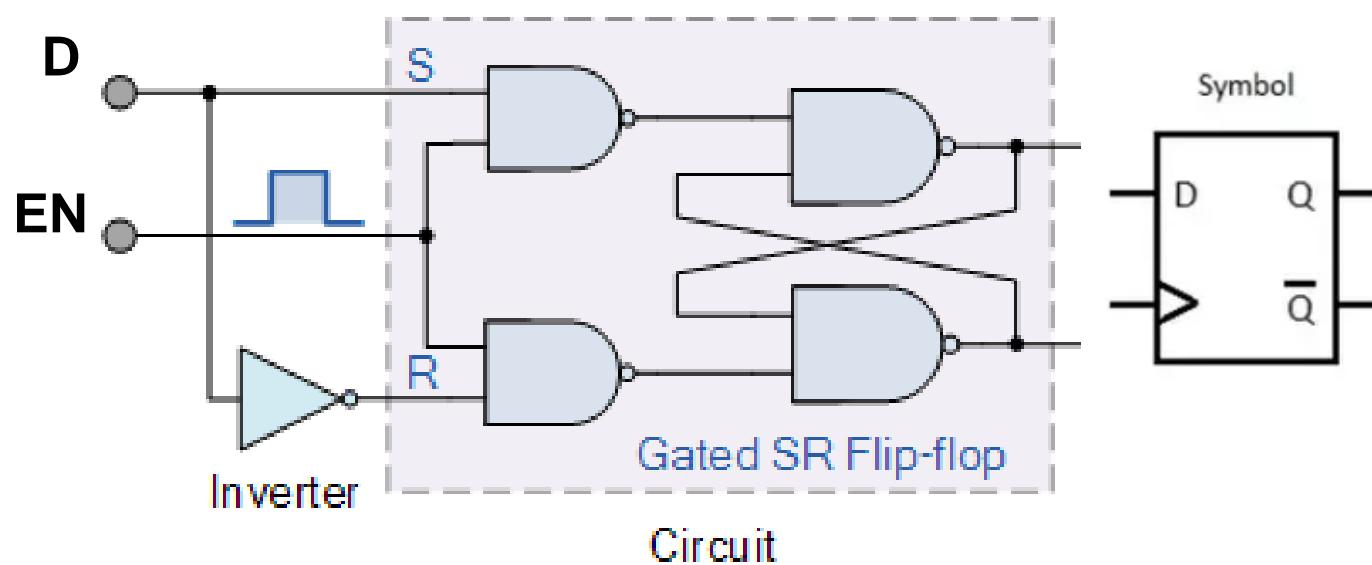


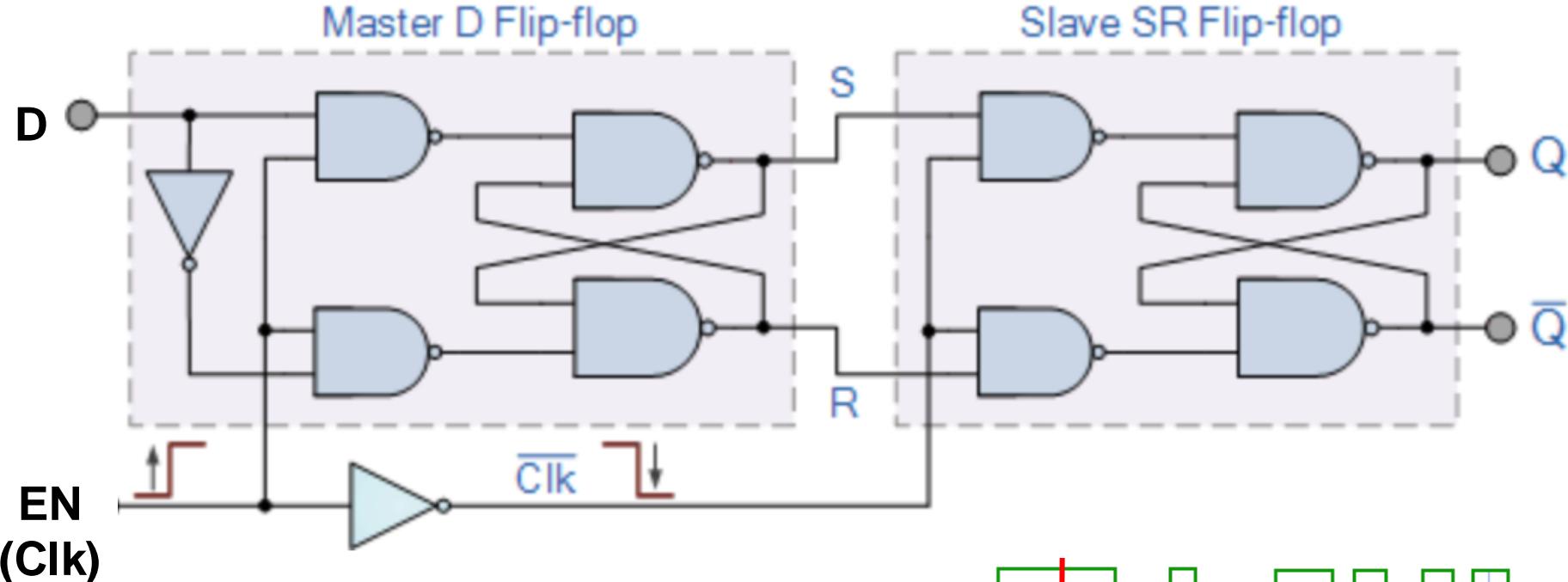
Table of truth:

clk	D	Q	$\bar{Q}$
0	0	Q	$\bar{Q}$
0	1	$\bar{Q}$	Q
1	0	0	1
1	1	1	0

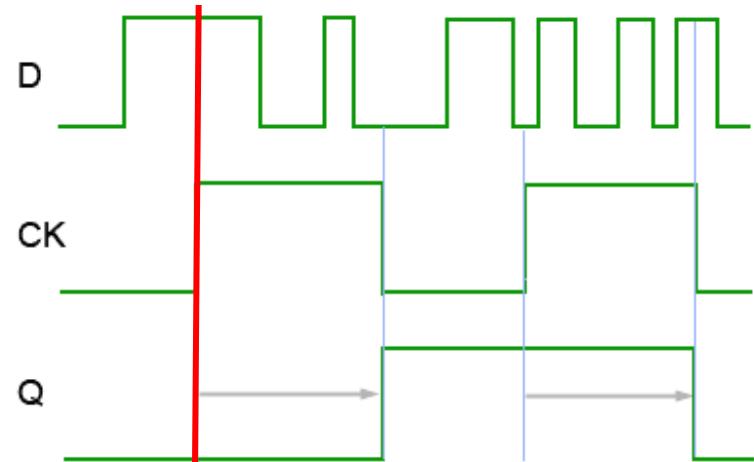
- Grazie all'invertitore, si ha solamente  $S=1$ ,  $R=0$  oppure  $S=0$ ,  $R=1 \rightarrow$  assimilabile a un unico input “D”
- Il DATO (D) viene trasferito su Q solo se il segnale di enable è alto
  - Latch FF, **Level Triggered**

# Master Slave Flip Flop D

## MASTER-SLAVE D Flip Flop



- $\text{Clk}=1 \rightarrow S \text{ e } R \text{ sono settati da } D.$
- $\text{Clk}=0 \rightarrow Q \text{ e } \bar{Q} \text{ sono settati da } S \text{ e } R.$
- $D$  è trasferito a  $Q$  in un intero ciclo di clock
- **Edge Triggered**

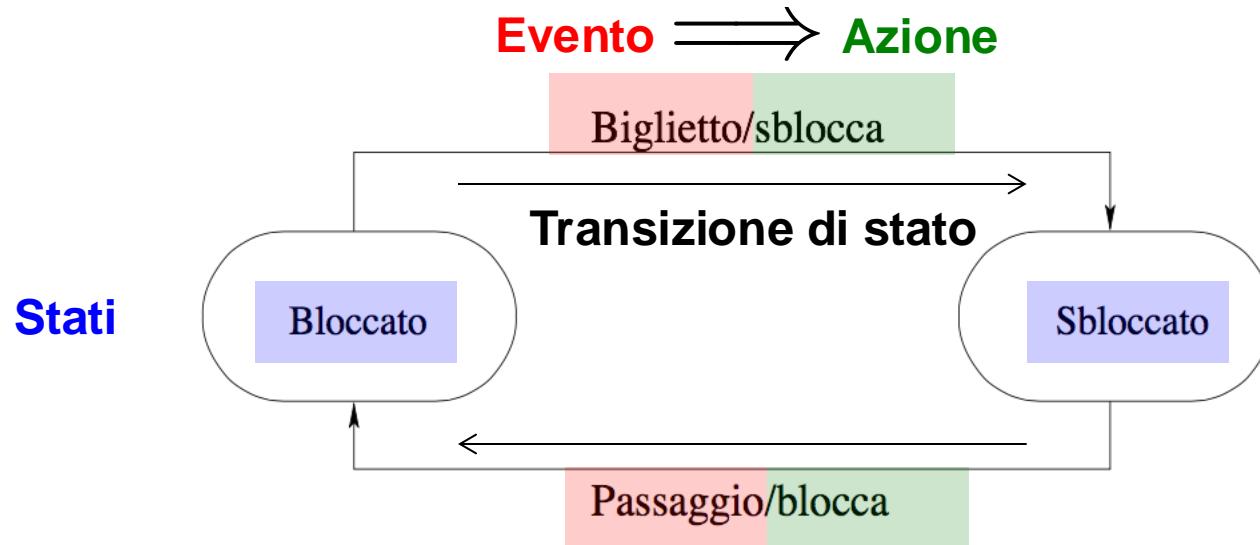


# Backup

# Sistemi Logici Complessi

porte logiche + flip flop + memorie/registri --> Sistemi logici complessi

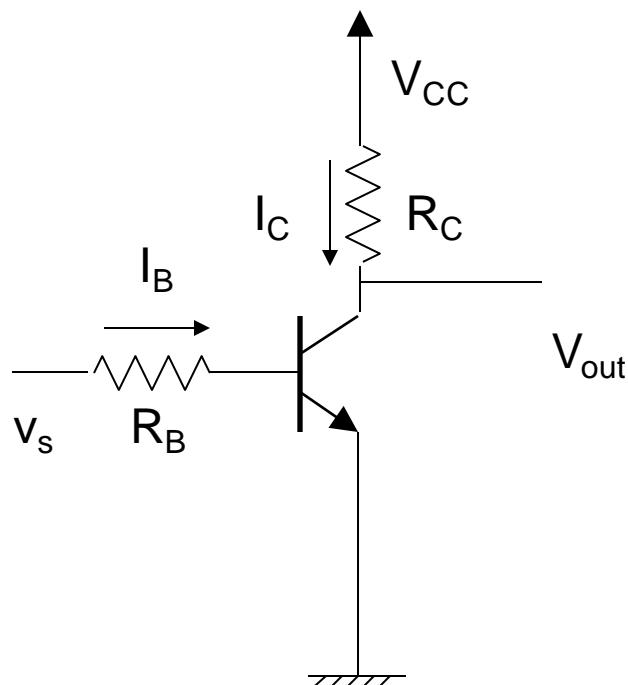
**FSM (Finite State Machine)**: sistema che può trovarsi in un numero finito di **stati** che può cambiare mediante **transizioni** triggerate da **eventi** esterni



# Invertitore (NOT)

Realizzazione: è di fatto un interruttore

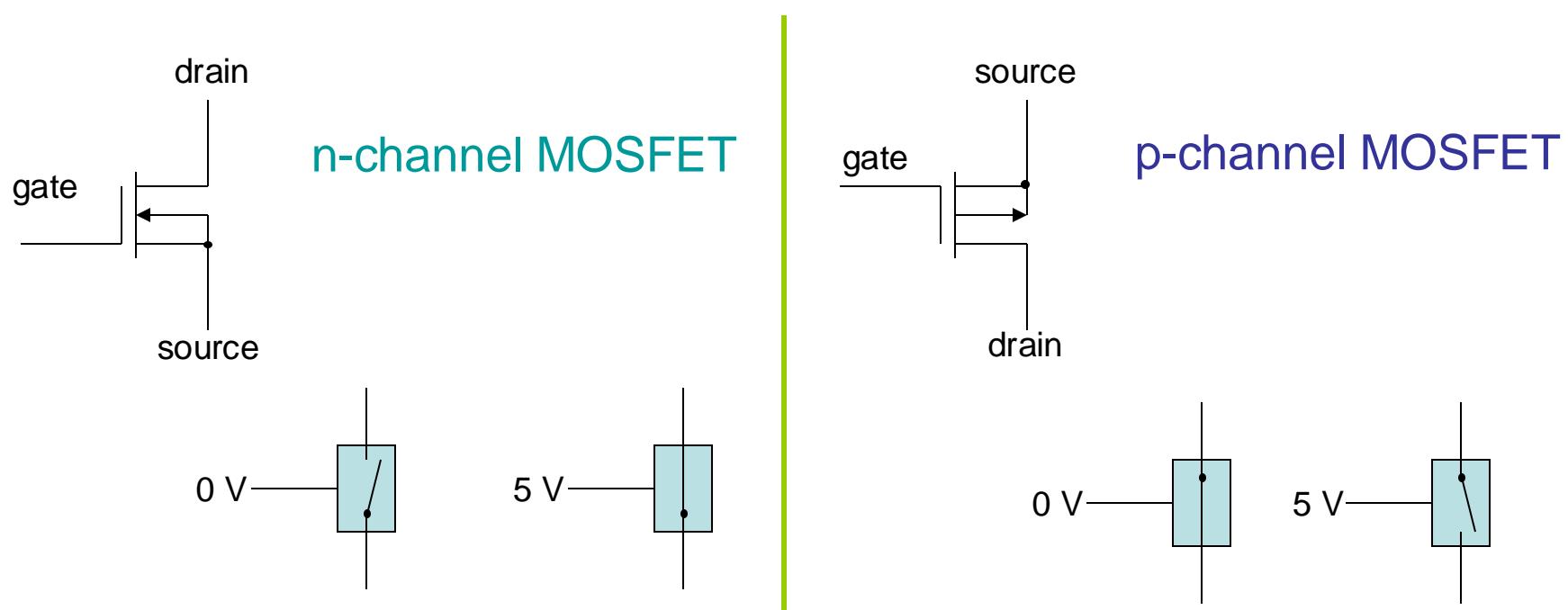
logica TTL (BJT)



- quando  $V_s$  è  $\sim 0$  il transitor è in cut-off  
→  $I_B \sim 0$   
→  $I_C \sim 0$   
→  $V_{out}$  è “pulled up” verso  $V_{CC}$
- quando  $V_s$  è “grande” il transitor va in saturazione  
→  $I_C$  è massima  
→  $V_{out} \sim 0$   
(dato che  $V_{CC} - V_{out} = R_C * I_C$ )

# Interruttori MOSFET

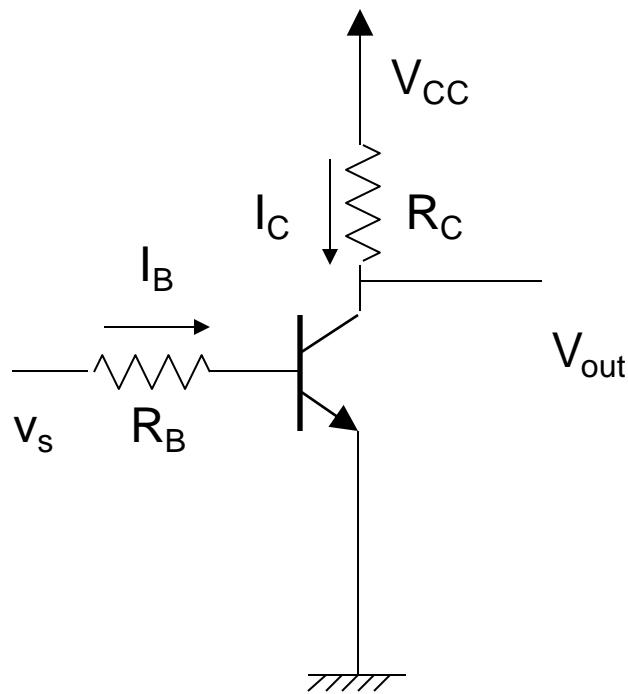
- i MOSFET, utilizzati nei circuiti di logica, agiscono come interruttori controllati con un voltaggio
  - n-channel MOSFET è chiuso (conduce) quando è applicato un voltaggio positivo (+5V), aperto quando il voltaggio è nullo
  - p-channel MOSFET è aperto quando è applicato un voltaggio positivo (+5V), chiuso (conduce) quando il voltaggio è nullo



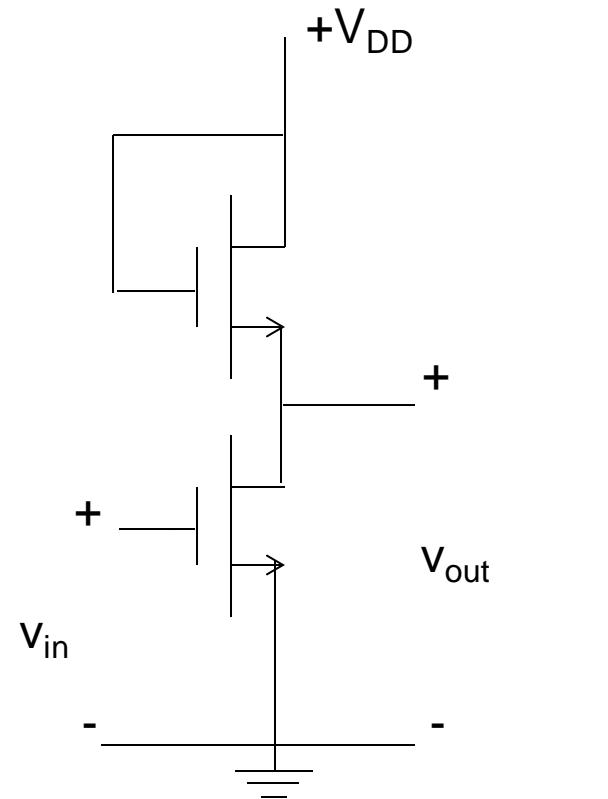
# Invertitore (NOT)

Realizzazione: è di fatto un interruttore

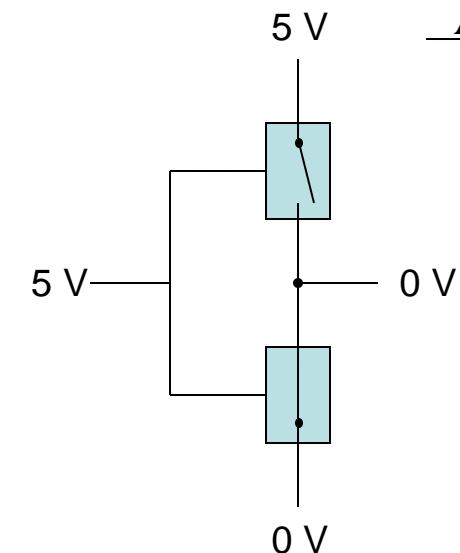
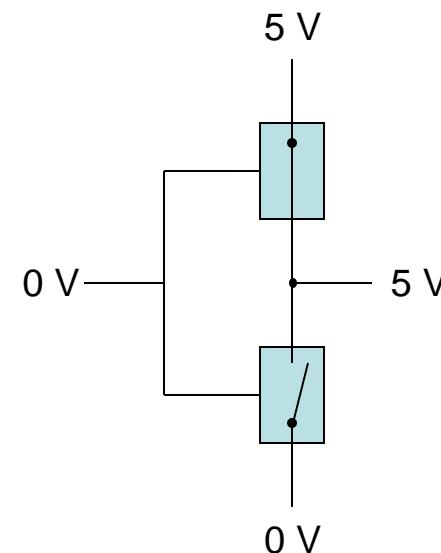
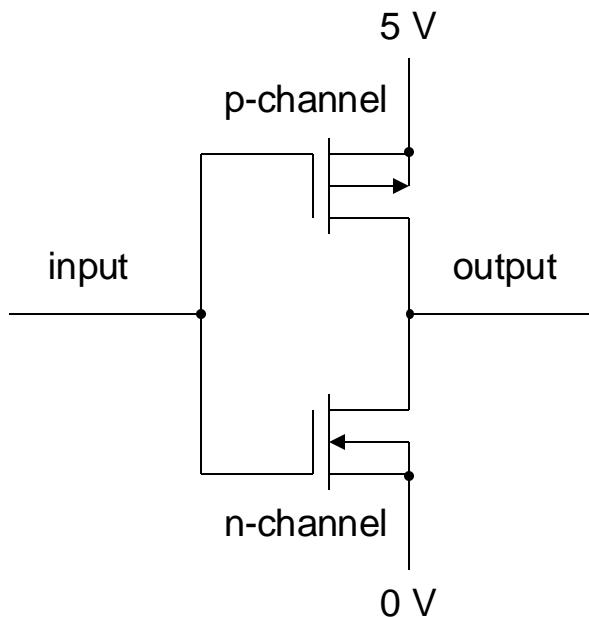
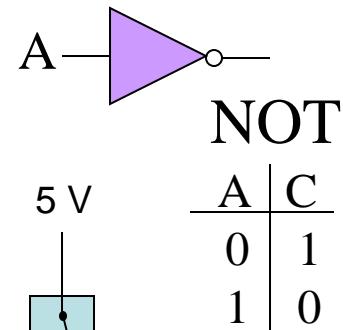
logica TTL (BJT)



logica NMOS (MOSFET)



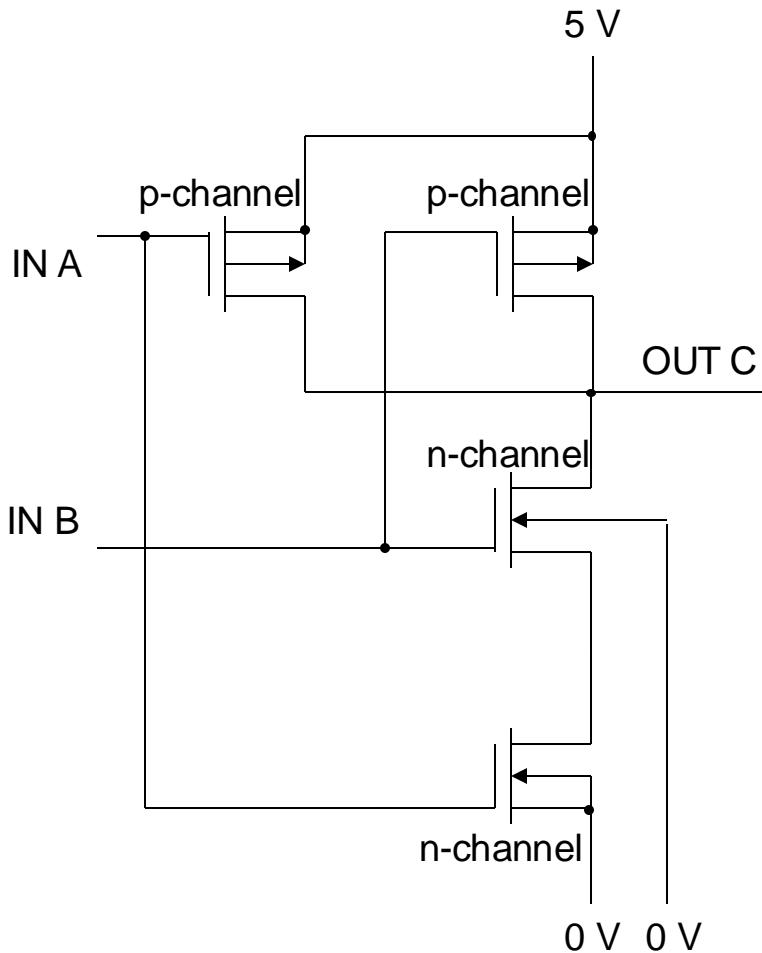
# Invertitore (NOT) MOSFET:



- 0V come input “apre” il FET in basso (n-channel) ma “chiude” quello in alto (p-channel) → l’output è a +5V
- 5V come input “chiude” il FET in basso (n-channel) ma “apre” quello in alto (p-channel) → l’output è a 0V

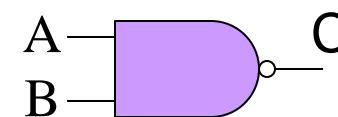
→ l’effetto netto è l’inversione logica:  $0 \rightarrow 5$ ;  $5 \rightarrow 0$

# NAND MOSFET:



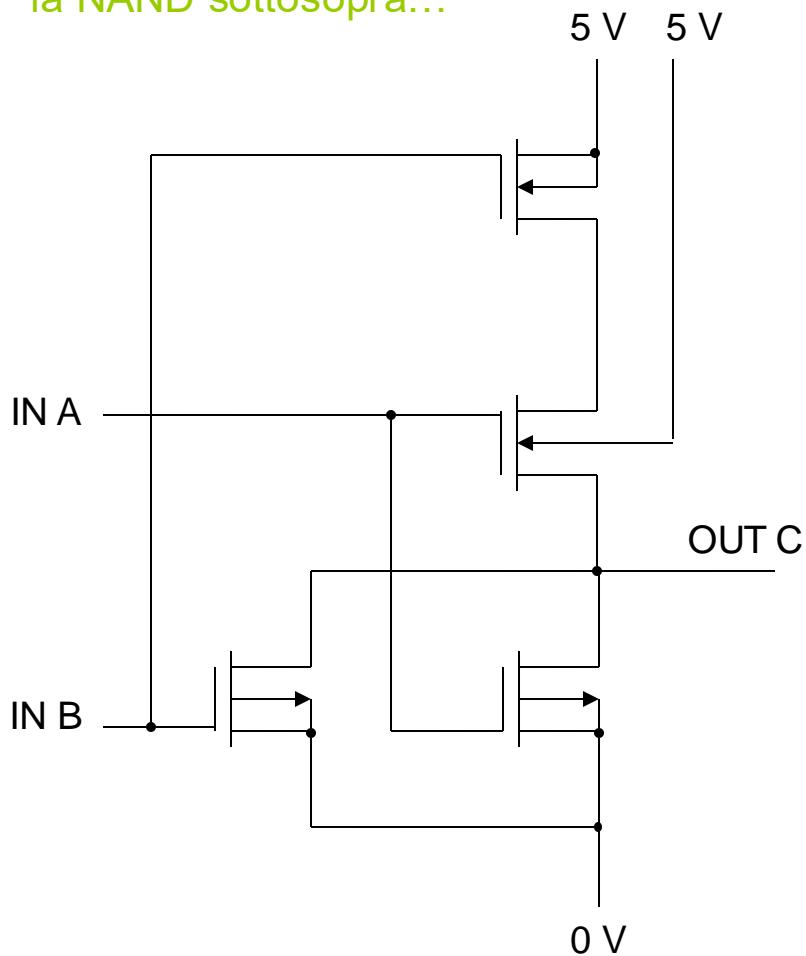
- Entrambe gli input a 0V:
  - i due FET in basso **OFF**, i due in alto **ON**  
→ uscita “alta”
- Entrambe gli input a 5V:
  - i due FET in basso **ON**, i due in alto **OFF**  
→ uscita “bassa”
- IN A a 5V, IN B a 0V:
  - alto a sinistra **OFF**, più basso **ON**
  - alto a destra **ON**, in mezzo **OFF**  
→ uscita “alta”
- IN A a 0V, IN B a 5V:
  - opposto rispetto a prima  
→ uscita “alta”

NAND		
A	B	C
0	0	1
0	1	1
1	0	1
1	1	0



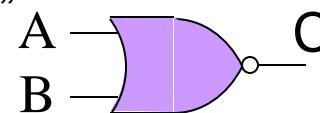
# NOR MOSFET:

la NAND sottosopra...



- Entrambe gli input a 0V:
  - i due FET in basso **OFF**, i due in alto **ON**  
 $\rightarrow$  output “alto”
- Entrambe gli input a 5V:
  - i due FET in basso **ON**, i due in alto **OFF**  
 $\rightarrow$  output “basso”
- IN A a 5V, IN B a 0V:
  - basso a sinistra **OFF**, basso destra **ON**
  - più alto **ON**, in mezzo **OFF**  
 $\rightarrow$  output “basso”
- IN A a 0V, IN B a 5V:
  - opposto rispetto a prima $\rightarrow$  output “basso”

NOR		
A	B	C
0	0	1
0	1	0
1	0	0
1	1	0



# Sottofamiglie TTL

