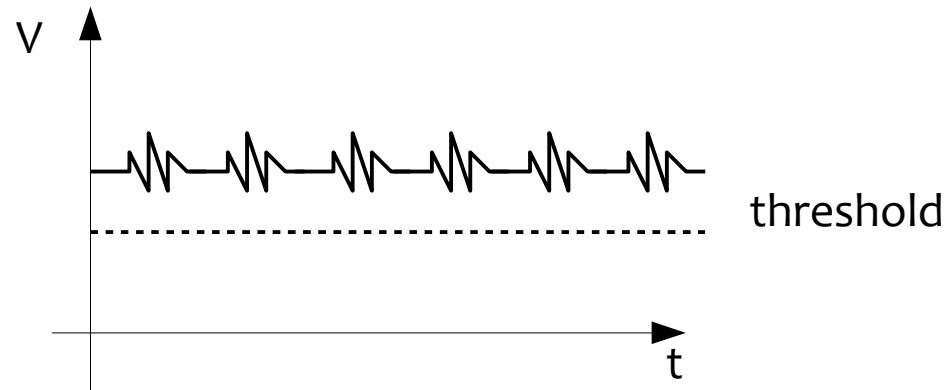


LABORATORIO DI ELETTRONICA ELETTRONICA DIGITALE

ELETTRONICA DIGITALE

- Elettronica Analogica: V (=tensione) variabile continua
- Elettronica Digitale: V può assumere solo 2 valori (=stati)
on/off, true/false, high/low, 1/0
- Questi 2 stati sono definiti dal fatto che la tensione del circuito che la implementa sia sopra/sotto un certo valore
 - Vantaggio: un segnale analogico è soggetto a fluttuazioni dovute al rumore che è sempre presente nei circuiti elettronici



- Limite: segnali che variano in modo continuo possono essere rappresentati solo con 2 livelli → soluzione: rappresentazione dei numeri in forma binaria

RAPPRESENTAZIONE DEI NUMERI

- Decimale (base 10):

- ogni cifra può assumere 10 valori $[0, \dots, 9]$
- ogni cifra è il coefficiente di una potenza di 10

$$139_{10} = 1 \times 10^2 + 3 \times 10^1 + 9 \times 10^0$$

- Binaria (base 2):

- ogni cifra può assumere 2 valori $[0, 1]$
- ogni cifra è il coefficiente di una potenza di 2

$$1001_2 = 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 9_{10}$$

Addizione/Sottrazione: regole simili (riporto/presitito)

$$\begin{aligned} \text{– es. } 1001_2 + 1111_2 &= 11000_2 \\ 11000_2 - 1001_2 &= 1111_2 \end{aligned}$$

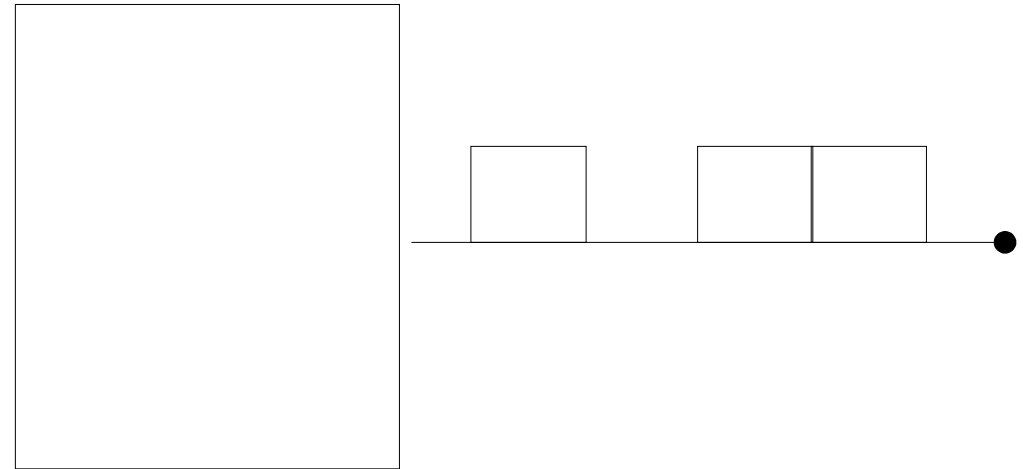
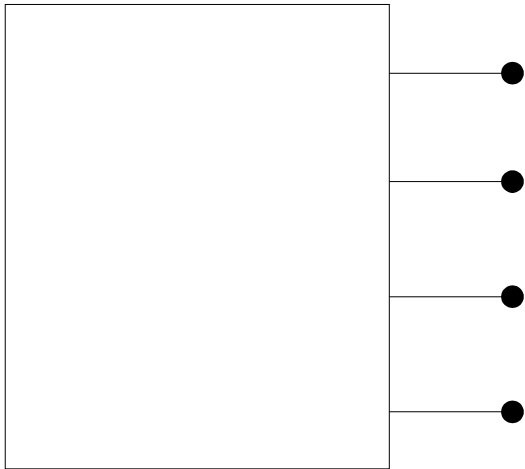
BIT & BYTE

- BIT = 1 cifra binaria (Binary digIT)
n bit $\rightarrow 0 \dots (2^n - 1)$
- 1 BYTE = 8 bit

| decimale | binario |
|----------|-----------|
| 0 | 0000 0000 |
| 1 | 0000 0001 |
| 2 | 0000 0010 |
| 3 | 0000 0011 |
| 4 | 0000 0100 |
| 5 | 0000 0101 |
| 6 | 0000 0110 |
| 7 | 0000 0111 |
| 8 | 0000 1000 |
| 9 | 0000 1001 |
| 10 | 0000 1010 |
| 11 | 0000 1011 |
| 12 | 0000 1100 |
| 13 | 0000 1101 |
| 14 | 0000 1110 |
| 15 | 0000 1111 |

NUMERI BINARI E CIRCUITI

- Due metodi per rappresentare gli n bit in un circuito
 - parallelo: ad un certo istante ciascun bit è rappresentato nel circuito da una V high/low
 - sequenziale/serie: i bit sono comunicati inviando una sequenza temporale di V high/low sulla stessa linea



Bisogna stabilire

- numero di bit che formano la sequenza
- quando inizia
- durata di 1 bit

FUNZIONI LOGICHE

- Funzioni logiche: definite tramite la tavola di verità (*truth table*) che fornisce lo stato delle m uscite (OUT) per ciascuna combinazione possibile degli n ingressi (IN)

| IN ₁ | IN ₂ | ... | IN _n | OUT ₁ | ... | OUT _m |
|-----------------|-----------------|-----|-----------------|------------------|-----|------------------|
| 0 | 0 | ... | 0 | 1 | ... | 0 |
| 0 | 0 | ... | 1 | 1 | ... | 1 |
| ... | ... | ... | ... | ... | ... | ... |
| 1 | 1 | ... | 1 | 0 | ... | 1 |

- combinatorie: ad un certo istante la configurazione delle uscite dipende solo dalla configurazione degli ingressi in quell'istante
- sequenziali: ad un certo istante la configurazione delle uscite dipende dalla configurazione degli ingressi in quell'istante e dalla configurazione precedente

FUNZIONI LOGICHE ELEMENTARI (*Gate*)

- AND

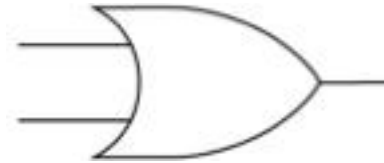
$$OUT = A \cdot B$$



| A | B | OUT |
|---|---|-----|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

- OR

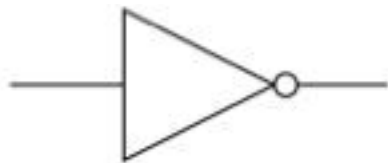
$$OUT = A + B$$



| A | B | OUT |
|---|---|-----|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

- NOT (inverter)

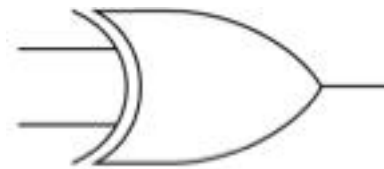
$$OUT = \bar{A}$$



| A | OUT |
|---|-----|
| 0 | 1 |
| 1 | 0 |

- EXOR

$$OUT = A \oplus B$$



| A | B | OUT |
|---|---|-----|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

- bit adder (modulo 2)
- controllo parità di una sequenza di bit

FUNZIONI LOGICHE ELEMENTARI (*Gate*)

- NAND

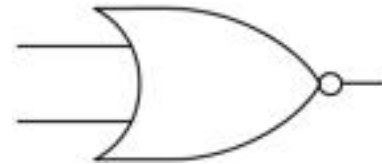
$$OUT = \overline{A \cdot B}$$



| A | B | OUT |
|---|---|-----|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

- NOR

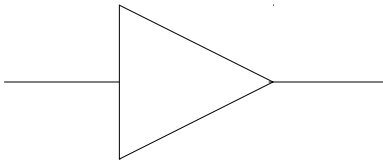
$$OUT = \overline{A + B}$$



| A | B | OUT |
|---|---|-----|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

- buffer

$$OUT = A$$



| A | OUT |
|---|-----|
| 0 | 0 |
| 1 | 1 |

Serve per “ripulire” il segnale che può degradarsi durante la propagazione all'interno del circuito

ALGEBRA DI BOOLE

- Regole con cui si manipolano le variabili booleane

- Definizioni

$$0 + A = A$$

$$1 + A = 1$$

$$A + A = A$$

$$A + \bar{A} = 1$$

$$0 \cdot A = 0$$

$$1 \cdot A = A$$

$$A \cdot A = A$$

$$A \cdot \bar{A} = 0$$

$$\overline{\bar{A}} = A$$

$$A \oplus B = (A \cdot \bar{B}) + (\bar{A} \cdot B)$$

A,B,C: variabili booleane (=0,1)

- Proprietà

$$(A + B) + C = A + B + C \quad \text{Associativa}$$

$$(A \cdot B) \cdot C = A \cdot B \cdot C$$

$$A + B = B + A \quad \text{Commutativa}$$

$$A \cdot B = B \cdot A$$

$$A \cdot (B + C) = (A \cdot B) + (A \cdot C) \quad \text{Distributiva}$$

$$A + (B \cdot C) = (A + B) \cdot (A + C)$$

$$A + (A \cdot B) = A \quad \text{Assorbitiva}$$

$$A \cdot (A + B) = A$$

$$\overline{A + B} = \bar{A} \cdot \bar{B} \quad \text{De Morgan}$$

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

Possibilità di produrre una qualunque funzione logica con un solo tipo di gate.
In pratica: NAND oppure NOR

IMPLEMENTAZIONE FISICA DEI GATE LOGICI

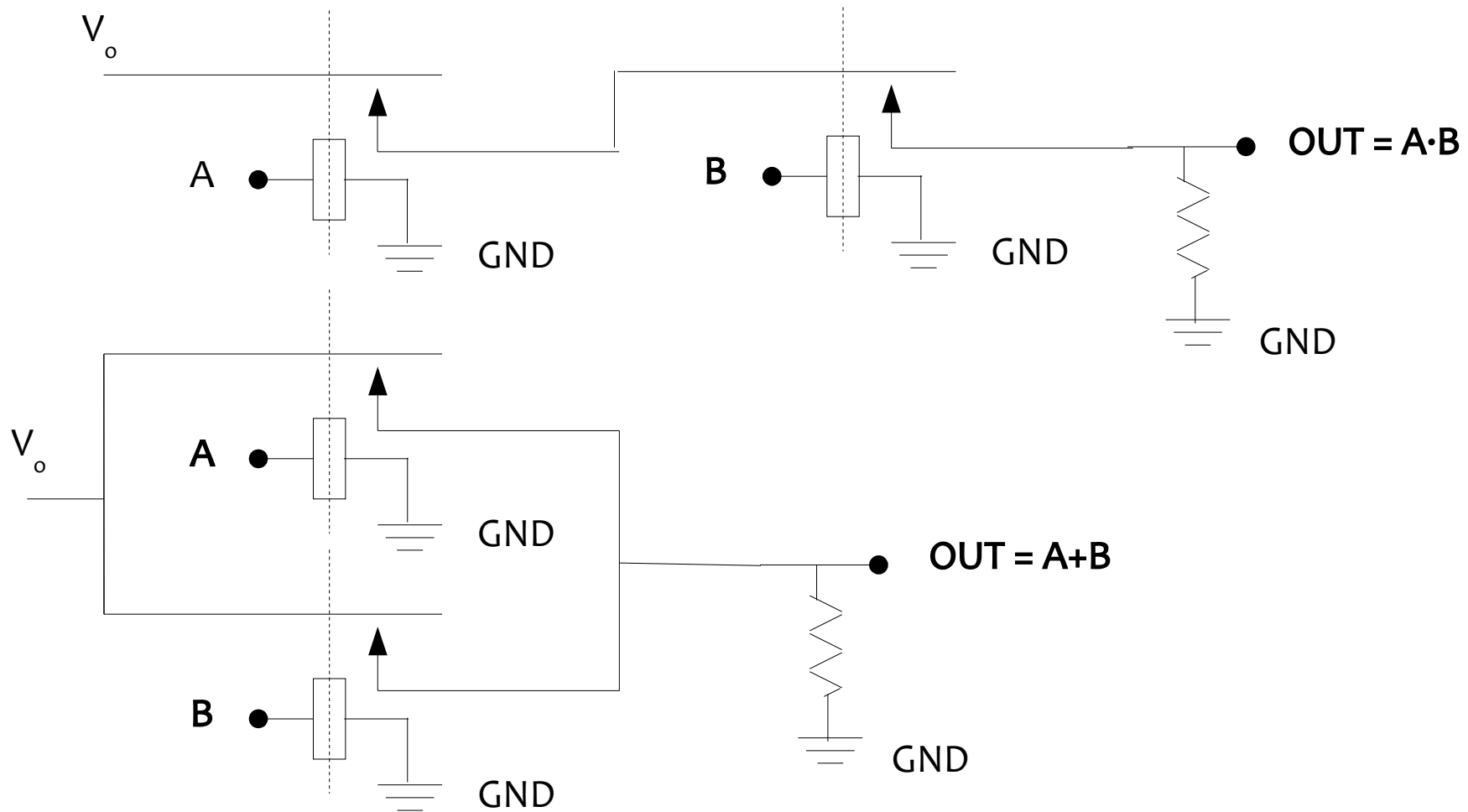
- Circuiti che in condizioni stazionarie hanno solo 2 stati stabili
- Si associa a ciascun livello di tensione un'affermazione logica

| LOGICA POSITIVA | | |
|-------------------|---|-------|
| V_{low} | 0 | false |
| V_{high} | 1 | true |

| LOGICA NEGATIVA | | |
|-------------------|---|-------|
| V_{high} | 0 | false |
| V_{low} | 1 | true |

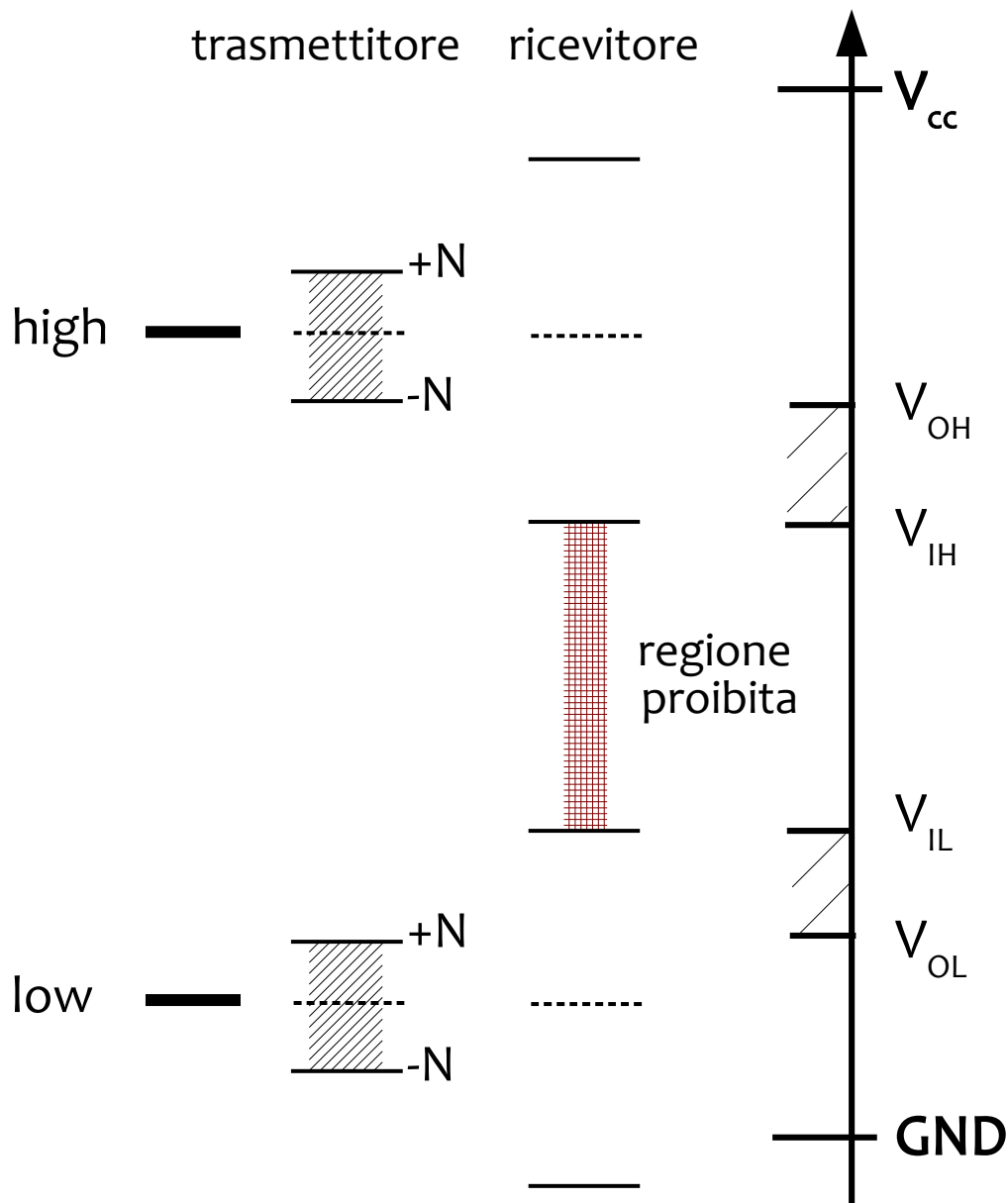
IMPLEMENTAZIONE FISICA DEI GATE LOGICI

- Gate logici realizzati come combinazione di interruttori controllabili da una tensione



In pratica: interruttori \rightarrow transistor in saturazione o in interdizione

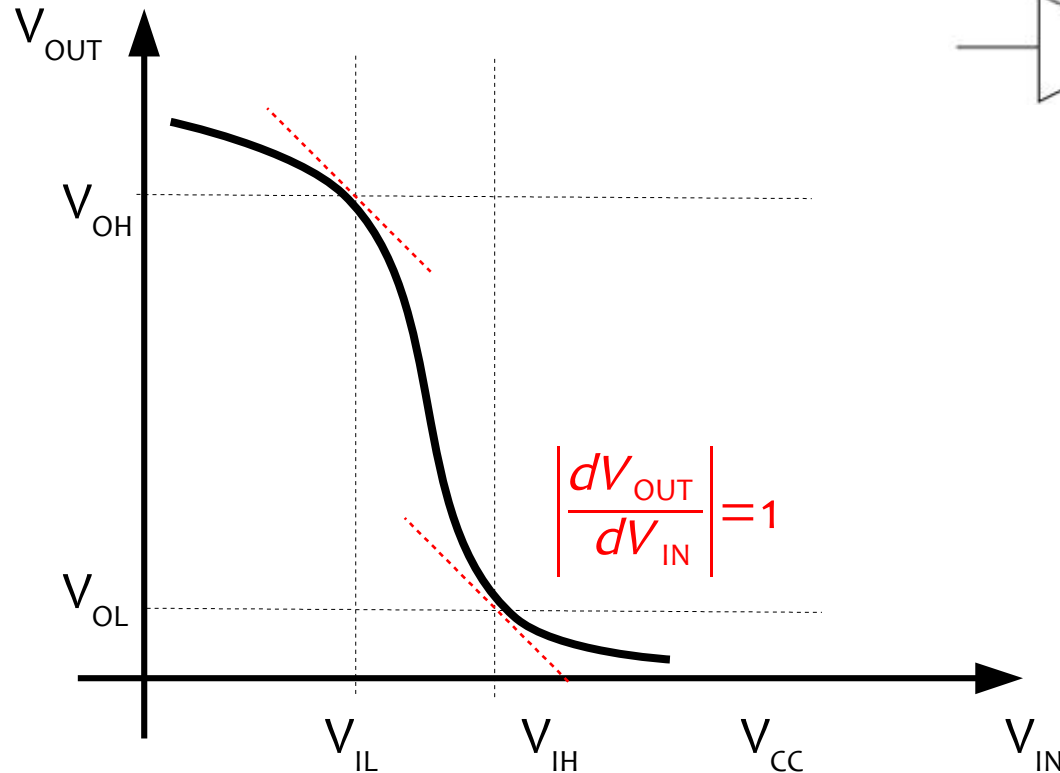
MARGINE DI RUMORE (*Noise Margin*)



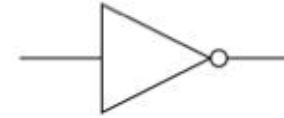
- Trasmissione di un segnale digitale
 - informazione codificata in un insieme di valori prefissati ad es. “low” e “high”
 - segnale trasmesso è comunque affetto da un errore $\pm N$
 - il ricevitore accetta segnali in un range di valori più ampio di $\pm N$
 - per evitare incertezze nella decodifica si introducono delle regioni proibite
 - in pratica il segnale è compreso tra **GND** e V_{CC}
 - noise margin: $V_{IL} - V_{OL}$ e $V_{OH} - V_{IH}$

VTC (Voltage-Transfer Characteristic)

- Esempio: gate “NOT”



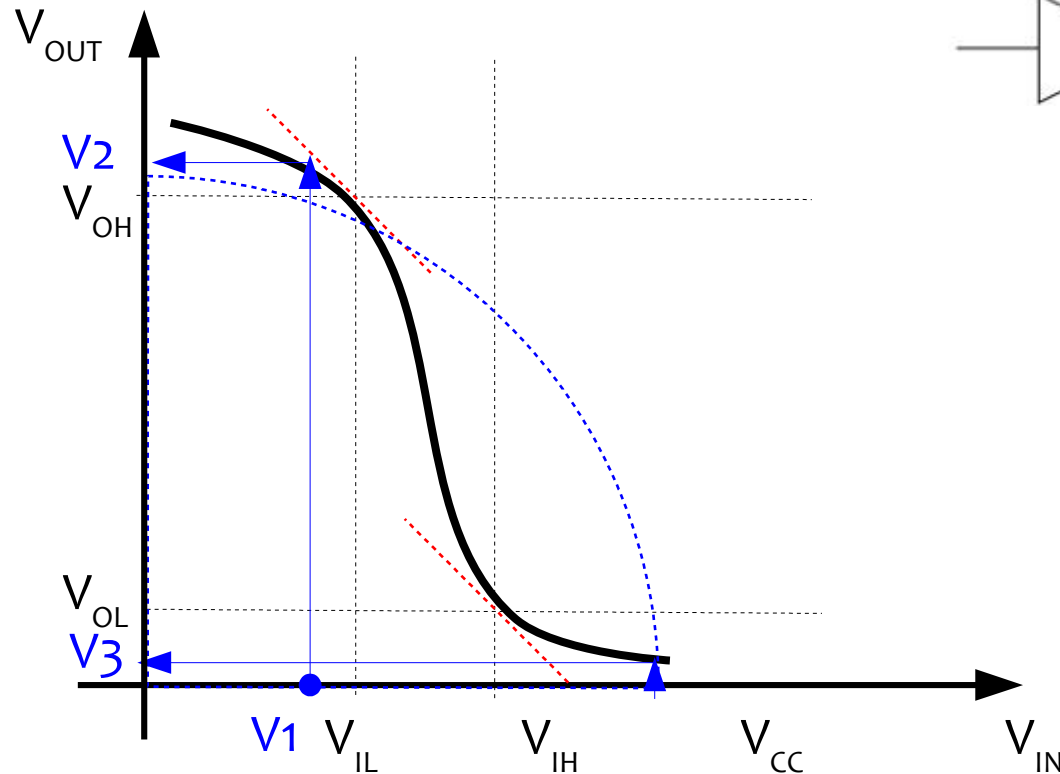
$$OUT = \bar{A}$$



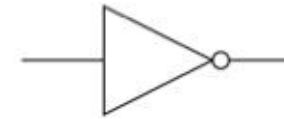
| A | OUT |
|---|-----|
| 0 | 1 |
| 1 | 0 |

VTC (Voltage-Transfer Characteristic)

- Esempio: gate “NOT”

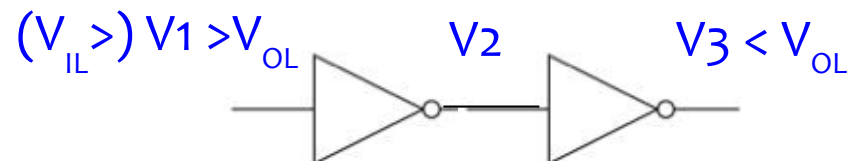


$$OUT = \bar{A}$$



| A | OUT |
|---|-----|
| 0 | 1 |
| 1 | 0 |

- Poiché $|dV_{OUT}/dV_{IN}| > 1$ l'inverter è in grado di rigenerare un segnale



CIRCUITI LOGICI INTEGRATI (*IC = Integrated Circuit*)

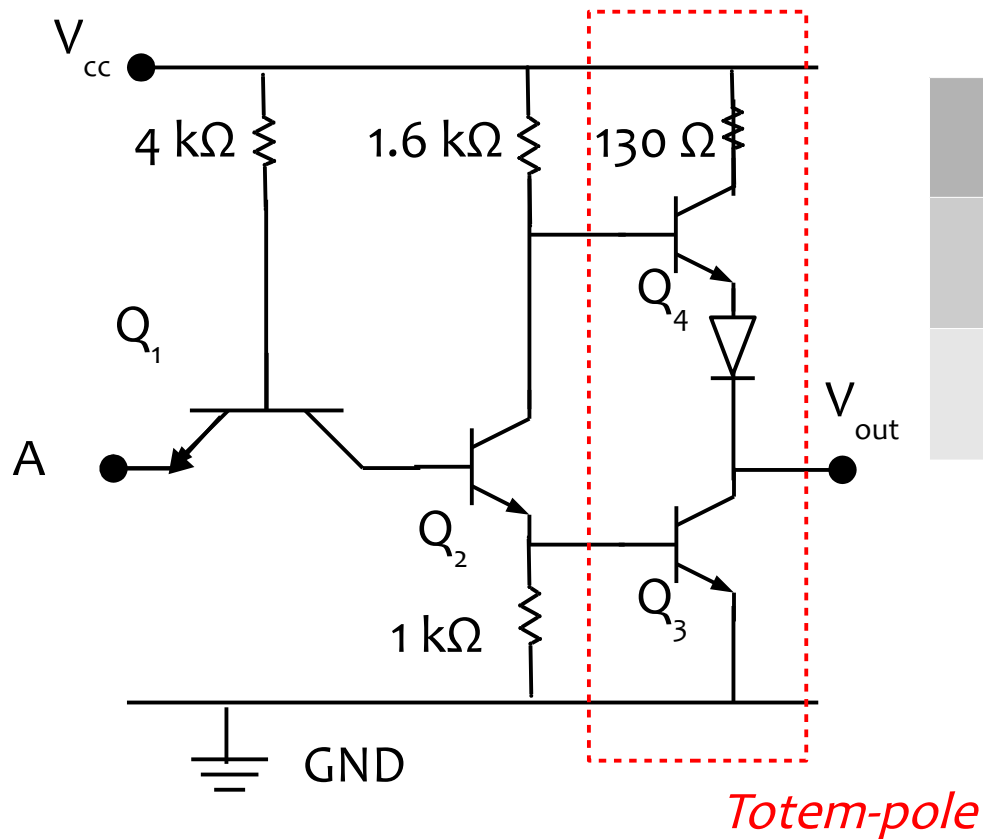
- Varie famiglie di IC
 - TTL (*Transistor-Transistor Logic*): transistor BJT
 - ECL (*Emitter-Coupled Logic*): transistor BJT
 - CMOS: transistor FET (e.g. MOSFET)

Differiscono per livello di integrazione, velocità/tempo di propagazione, dissipazione, livello di rumore (*noise immunity*), fan-out

- TTL:
 - Interdizione $V_{CE(off)} = 4 \pm 1 \text{ V}$
 - Saturazione $V_{CE(sat)} = 0.2 \pm 0.2 \text{ V}$

NOT TTL

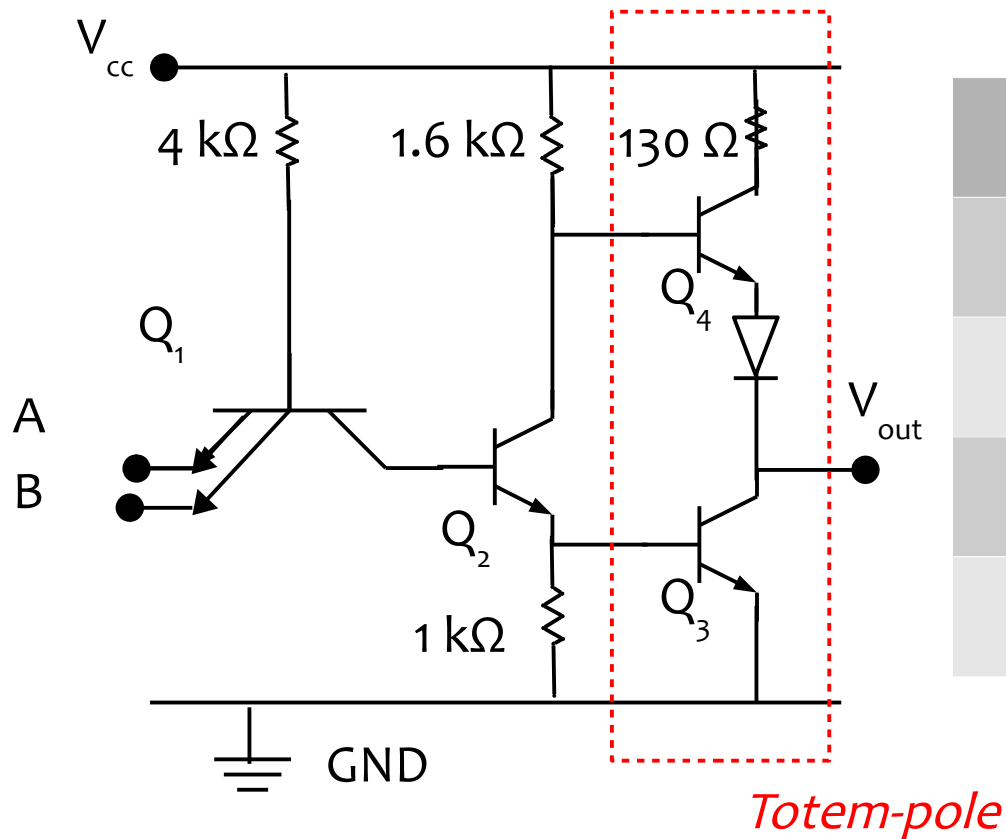
- Logica positiva
1 \rightarrow high $\rightarrow V_{cc} = +5\text{ V}$
0 \rightarrow low $\rightarrow \text{GND} = 0\text{ V}$



| A | Q1 | Q2 | Q3 | Q4 | OUT |
|------|--------|-----|-----|-----|------|
| 0 V | BC rev | Off | Off | On | +5 V |
| +5 V | BC fwd | On | On | Off | 0 V |

NAND TTL

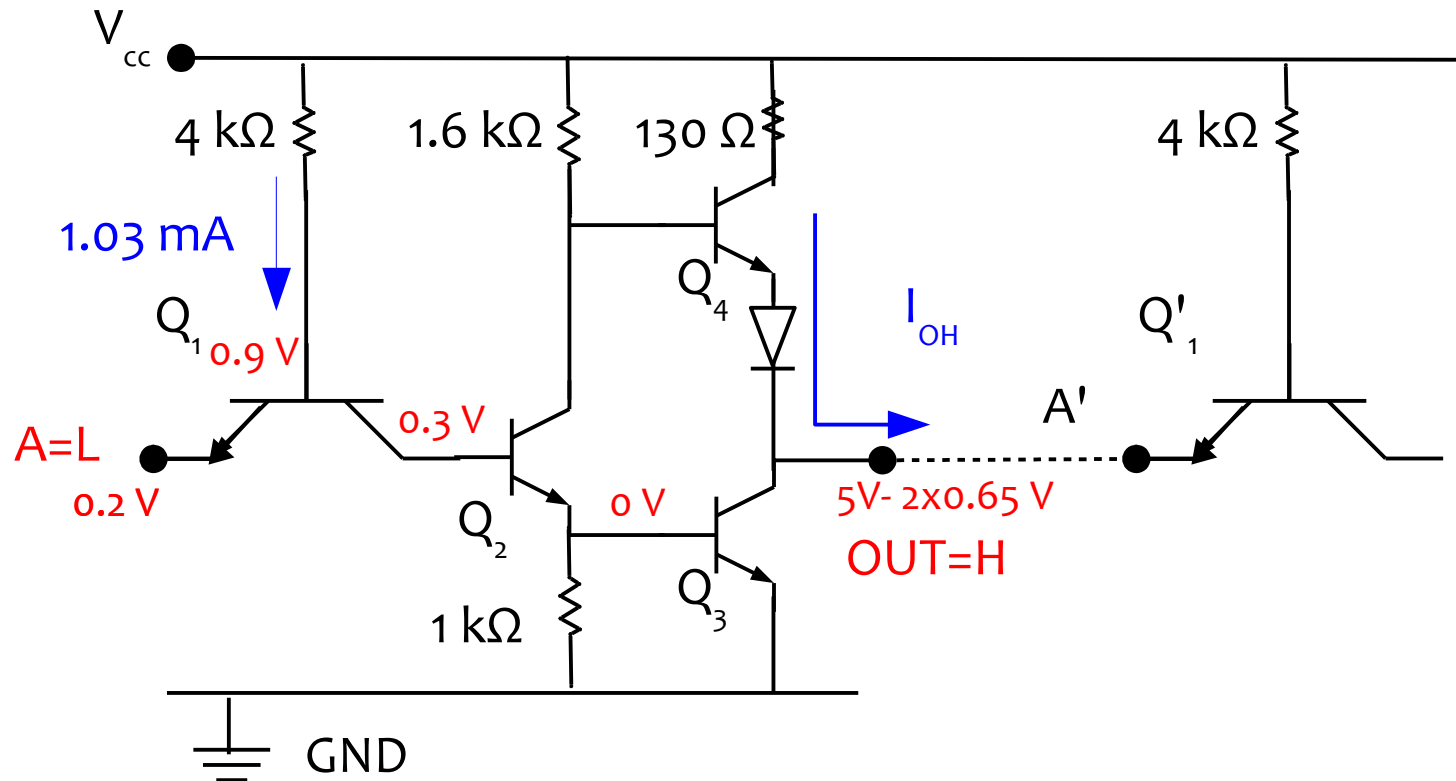
- Logica positiva
1 \rightarrow high $\rightarrow V_{cc} = +5\text{ V}$
0 \rightarrow low $\rightarrow \text{GND} = 0\text{ V}$



| A | B | Q1 | Q2 | Q3 | Q4 | OUT |
|------|------|--------|-----|-----|-----|------|
| 0 V | 0 V | BC rev | Off | Off | On | +5 V |
| 0 V | +5 V | BC rev | Off | Off | On | +5 V |
| +5 V | 0 V | BC rev | Off | Off | On | +5 V |
| +5 V | +5 V | BC fwd | On | On | Off | 0 V |

FAN-OUT NOT TTL

- Capacità dell'uscita di un gate TTL di pilotare N ingressi di gate TTL

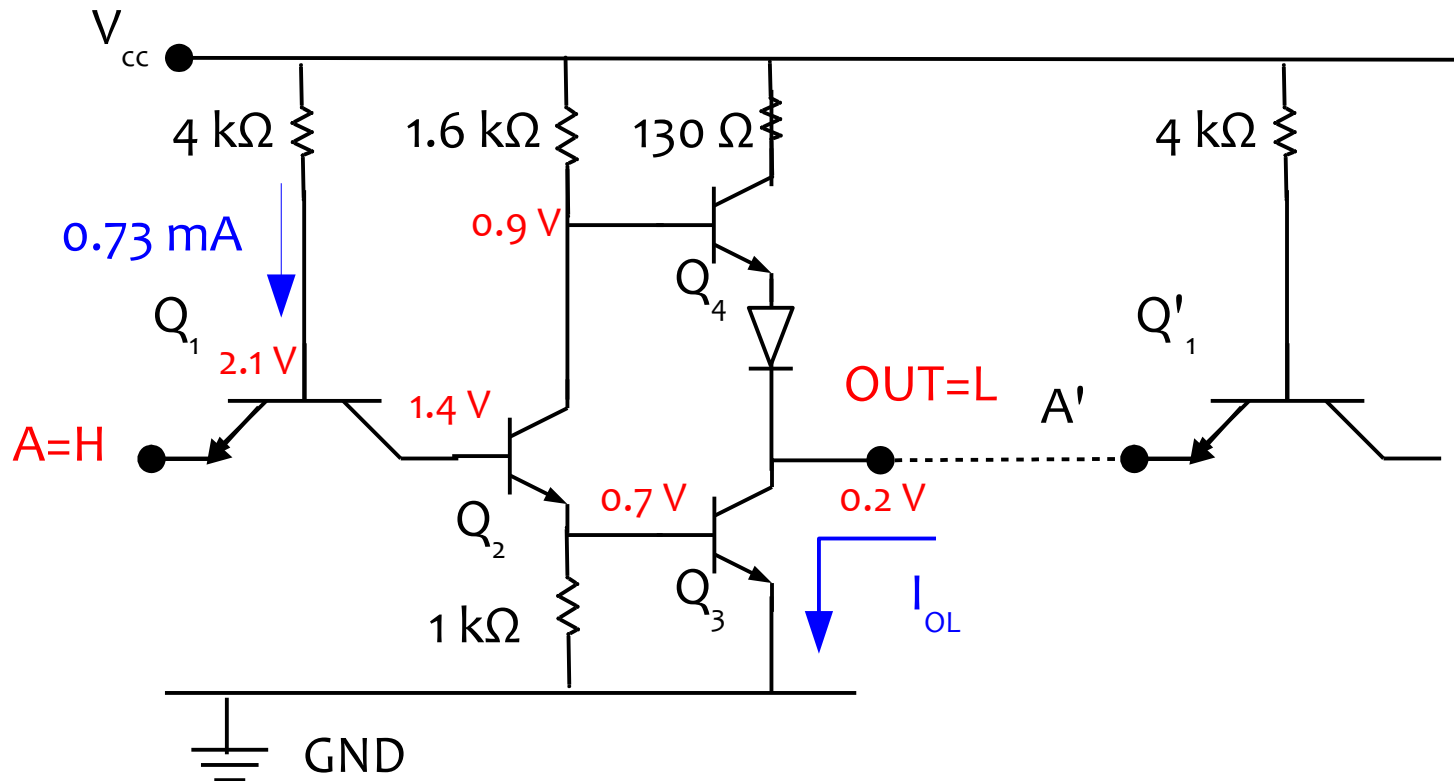


| | | |
|-----|----|------|
| Q3 | Q4 | OUT |
| Off | On | +5 V |

- OUT=H: Q_4 deve fornire (*source*) la I_{IH} richiesta dal gate successivo (Q'_1)
Ad es. $I_{IH} \sim 15 \mu A$ se $\beta_R = 1/50$

FAN-OUT NOT TTL

- Capacità dell'uscita di un gate TTL di pilotare N ingressi di gate TTL



| Q3 | Q4 | OUT |
|----|-----|-----|
| On | Off | 0 V |

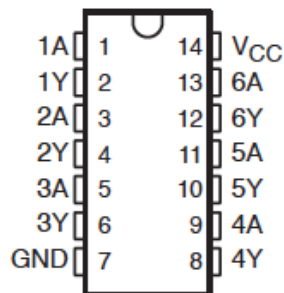
- OUT=L: Q_3 deve assorbire (*sink*) la $I_{IL} \sim -1$ mA prodotta dal gate successivo (Q'_1)

$$\text{Fan-out} = I_{OL}(\text{max}) / I_{IL}(\text{max}) \quad \text{tipicamente} \sim 10$$

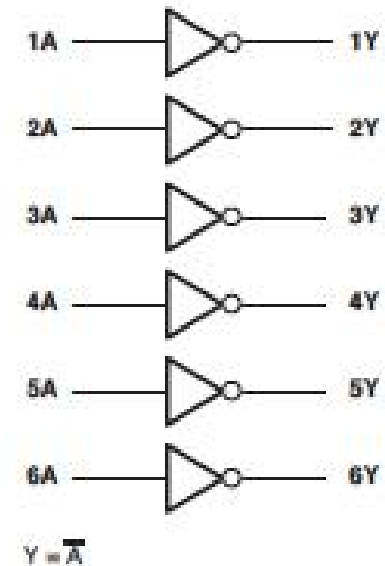
SN5404, SN54LS04, SN54S04, SN7404, SN74LS04, SN74S04 HEX INVERTERS

SDLS029C – DECEMBER 1983 – REVISED JANUARY 2004

SN5404 . . . J PACKAGE
SN54LS04, SN54S04 . . . J OR W PACKAGE
SN7404, SN74S04 . . . D, N, OR NS PACKAGE
SN74LS04 . . . D, DB, N, OR NS PACKAGE
(TOP VIEW)



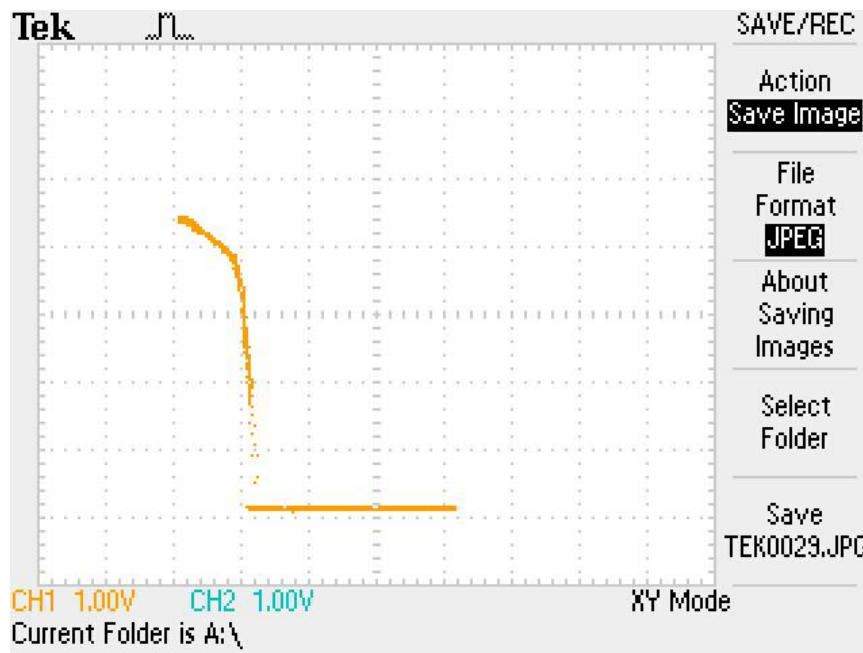
logic diagram (positive logic)



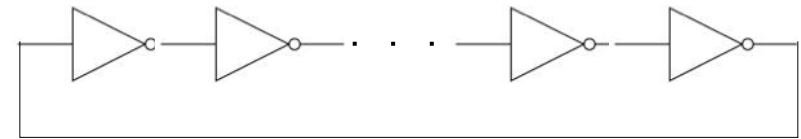
- In laboratorio: usare un IC 74LS04 per determinare
 - Voltage Transfer Characteristics
 - comportamento di un anello di NOT (*ring of inverters*)

- VTC

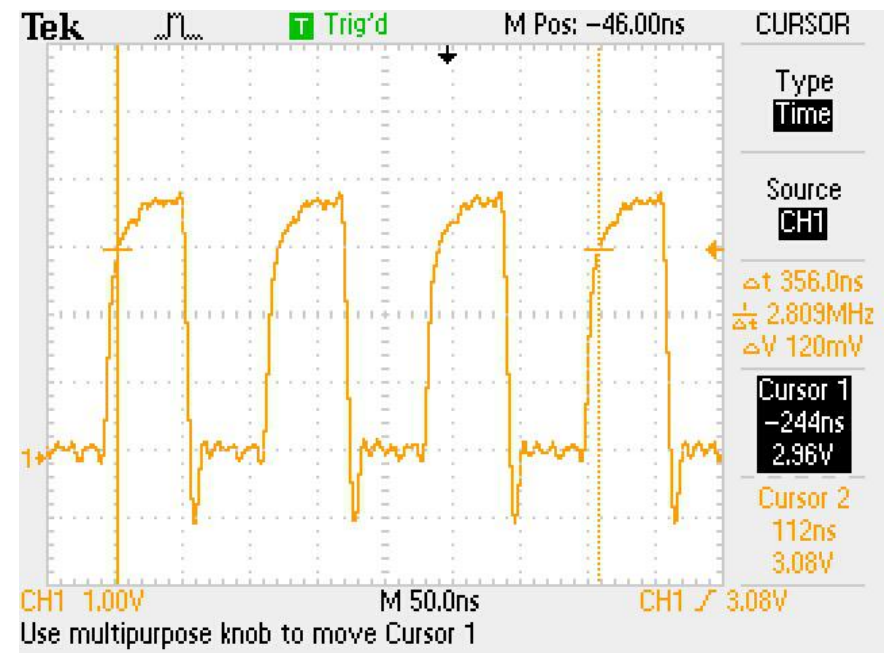
- segnale in ingresso: deve variare in modo continuo tra i livelli low ed high
- oscilloscopio in modalità XY



- Anello di n gate NOT



- spiegare il comportamento di un'uscita nei casi $n=7$, $n=9$, $n=11$



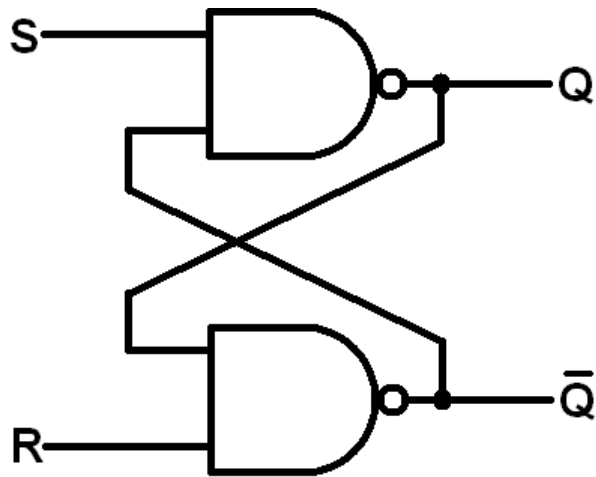
REGISTRI

- L'uscita delle funzioni logiche combinatoriali viste finora riflette lo stato degli ingressi in quell'istante

Registri (*Information Register*): funzioni logiche sequenziali che servono a mantenere/registrazione l'informazione sullo stato precedente degli ingressi → memoria

R-S (*Reset-Set*) FLIP-FLOP

- Detto anche MULTIVIBRATORE BISTABILE



| S | R | Q | \bar{Q} |
|---|---|---|-----------|
| 0 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |

comportamento
impredicibile

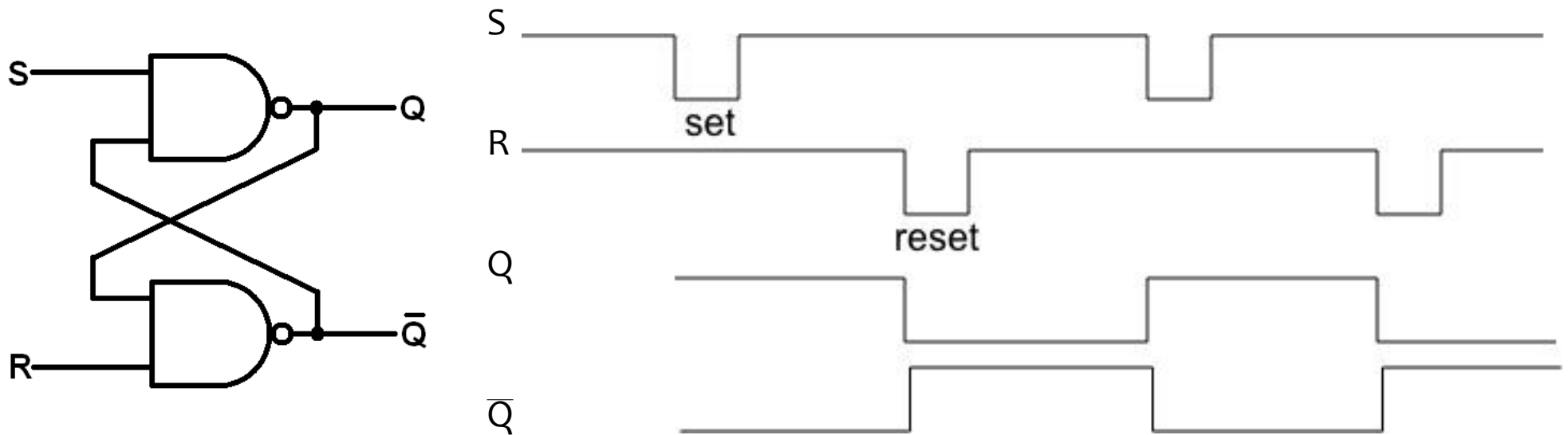
stati stabili

memoria

- Caso $(R,S)=(0,0)$: nel caso di transizioni degli ingressi $(0,0) \rightarrow (1,1)$ lo stato presente sulle uscite è impredicibile
 $(Q,\bar{Q})=(1,0)$ oppure $(Q,\bar{Q})=(0,1)$ entrambi possibili

R-S (*Reset-Set*) FLIP-FLOP

- È possibile determinare in quale dei due stati “stabili” si trovi il circuito mandando temporaneamente low R oppure S



Memoria: lo stato attuale delle uscite (Q, \bar{Q}) ricorda quale degli ingressi (R,S) è stato low per ultimo

R-S (*Reset-Set*) FLIP-FLOP

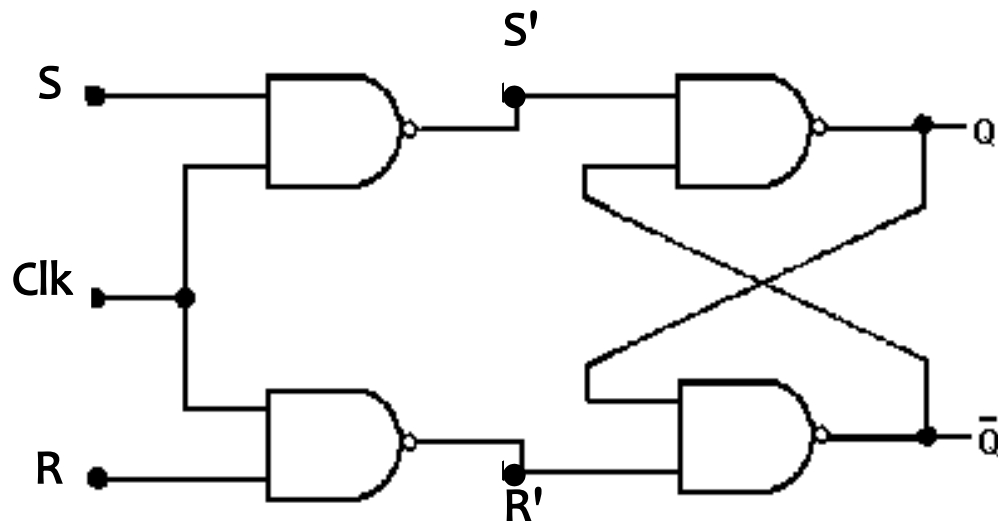
- È possibile determinare in quale dei due stati “stabili” si trovi il circuito mandando temporaneamente low R oppure S

| | | S | R | Q | \bar{Q} | |
|--------|-------|---|---|---|-----------|---------|
| Caso 1 | Start | 1 | 1 | 0 | 1 | “set” |
| | End | 0 | 1 | 1 | 0 | |
| Caso 2 | Start | 1 | 1 | 1 | 0 | |
| | End | 0 | 1 | 1 | 0 | |
| Caso 3 | Start | 1 | 1 | 1 | 0 | “reset” |
| | End | 1 | 0 | 0 | 1 | |
| Caso 4 | Start | 1 | 1 | 0 | 1 | |
| | End | 1 | 0 | 0 | 1 | |

Memoria: lo stato attuale delle uscite (Q, \bar{Q}) ricorda quale degli ingressi (R,S) è stato low per ultimo

LOGICA SINCRONIZZATA: CLOCKED R-S FLIP-FLOP

- R-S flip-flop: le uscite cambiano non appena cambiano gli ingressi
→ necessità di controllare l'istante in cui cambia lo stato delle uscite



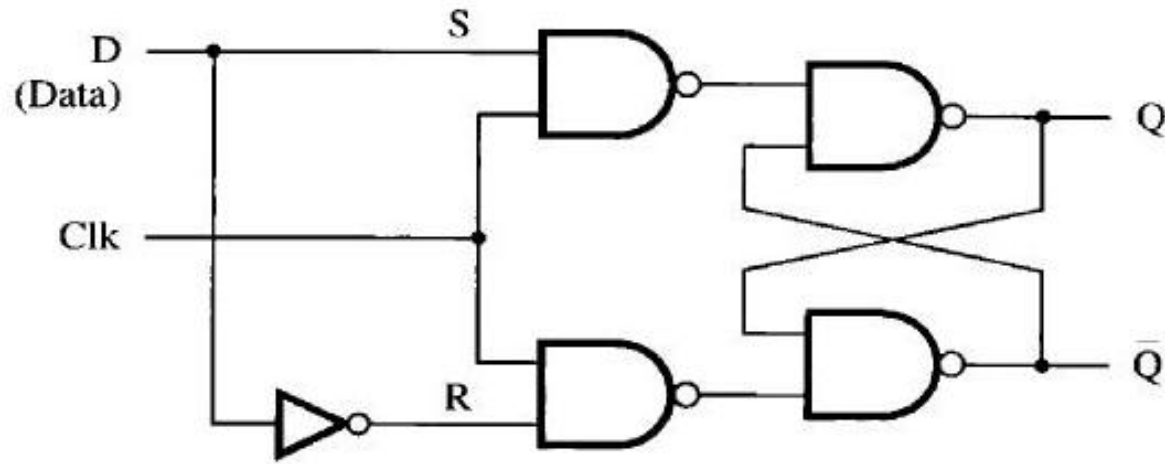
| S | R | Q_{n+1} |
|---|---|-----------|
| 0 | 0 | Q_n |
| 0 | 1 | 0 |
| 1 | 0 | 1 |
| 1 | 1 | non usato |

(R', S') possono cambiare solo quando il clock $Clk=1$ → il clock “abilita” gli ingressi del flip-flop → gli ingressi sono riportati sulle uscite solo quando $Clk=1$ e mantenuti fintanto che $Clk=0$

ovvero Clk fornisce una separazione tra ingressi (R, S) ed uscite (Q, \bar{Q})

Nota: in questo caso normalmente $(R, S)=(0,0)$ e lo stato delle uscite viene modificato quando uno degli ingressi è temporaneamente high

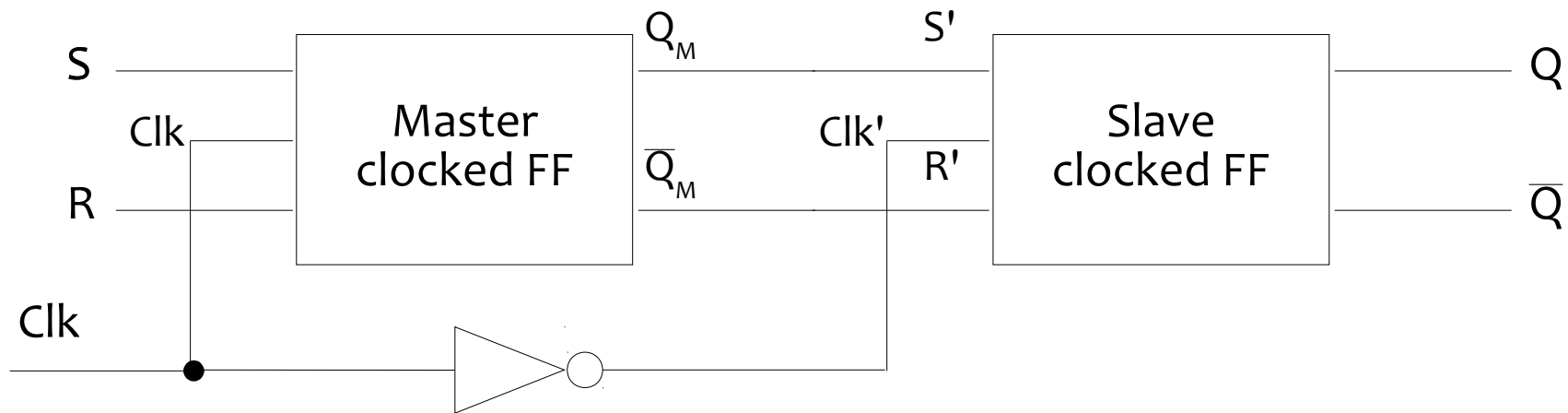
D FLIP-FLOP (*latch*)



| D | Clk | Q_{n+1} |
|---|-----|-----------|
| 0 | 1 | 0 |
| 1 | 1 | 1 |
| X | 0 | Q_n |

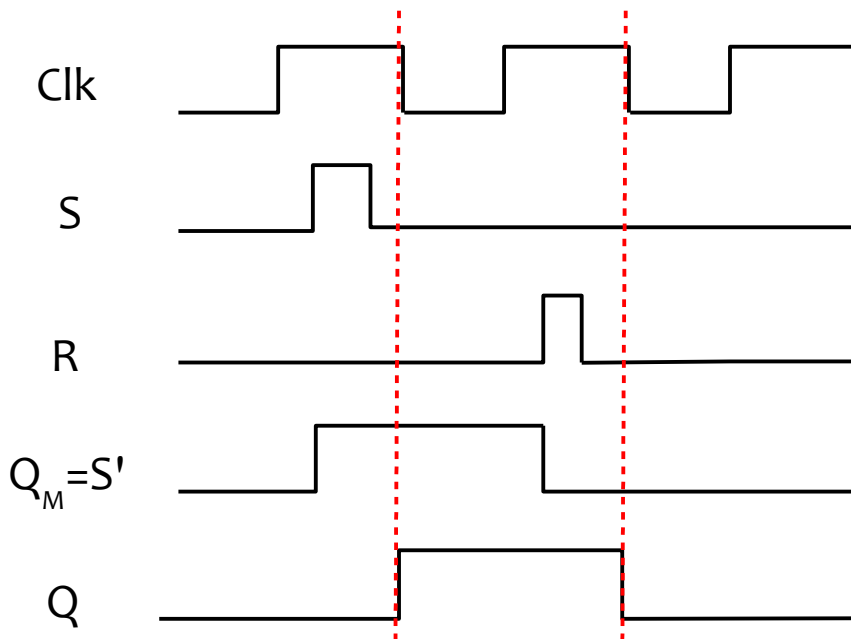
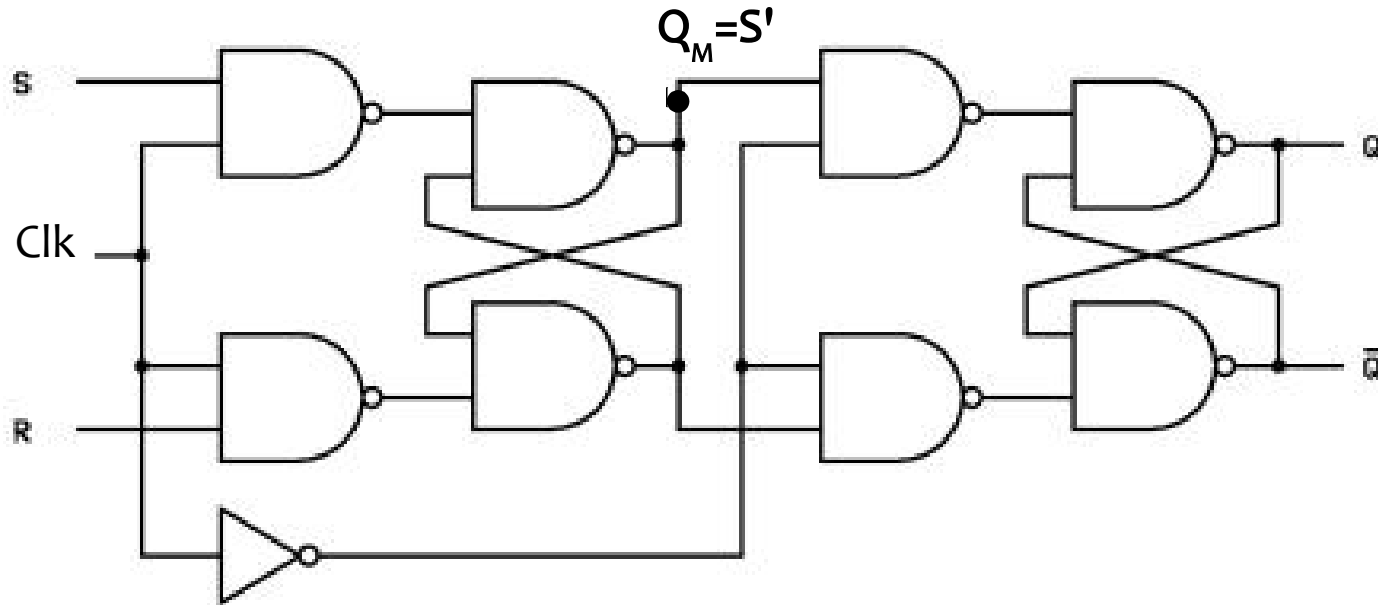
MASTER-SLAVE FLIP-FLOP

- Clocked flip-flop: se gli ingressi cambiano n-volte mentre $\text{Clk}=1$ anche le uscite cambiano n-volte
- MS flip-flop è composto da due clocked flip-flop in cui il clock è in opposizione di fase (NOT)



- $\text{Clk} = 1$: lo stato degli ingressi (R,S) del master flip-flop è trasferito alle uscite (Q_M, \overline{Q}_M); siccome $\text{Clk}' = 0$ le uscite (Q, \overline{Q}) dello slave flip-flop non cambiano
- $\text{Clk} = 0$: le uscite (Q_M, \overline{Q}_M) del master FF non cambiano ma $\text{Clk}' = 1 \rightarrow$ il loro stato è trasferito alle uscite (Q, \overline{Q}) dello slave flip-flop

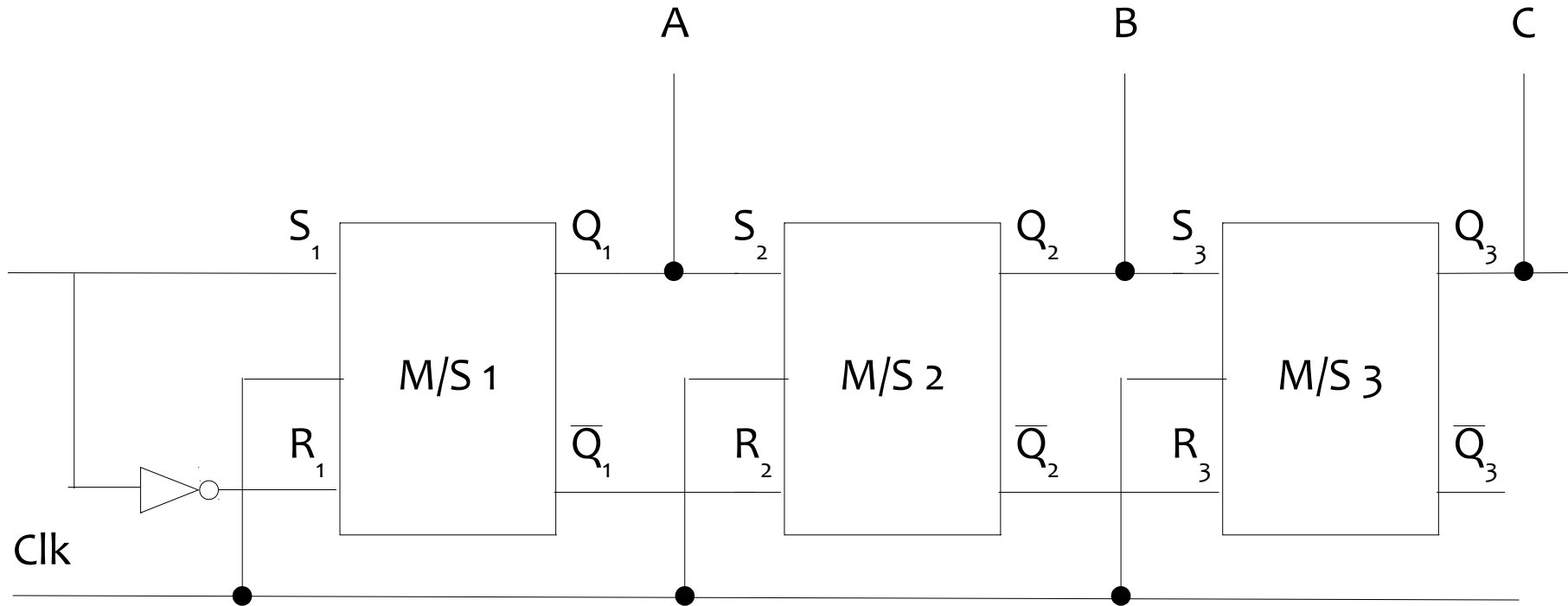
MASTER-SLAVE FLIP-FLOP



- Gli ingressi (R,S) sono trasferiti alle uscite (Q,Q̄) nelle transizioni high-low (*negative edge*) del clock (Clk)

REGISTRO DI SCORRIMENTO (*Shift Register*)

- Serie di Master-Slave flip-flop con il clock in comune



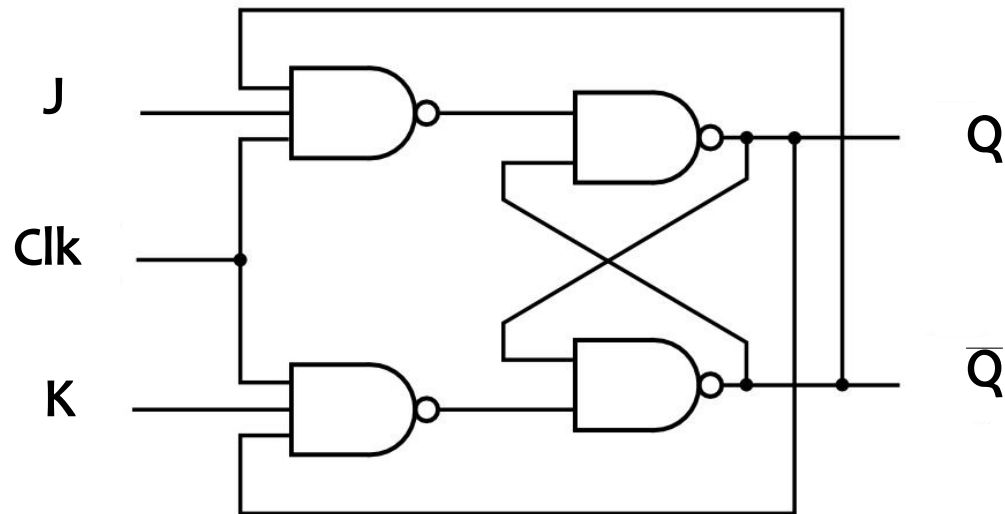
– Esempio:

| | S_1 | | S_2 | S_3 | A | B | C |
|------------|-------|--|-------|-------|---|---|---|
| Inizio | 1 | | 0 | 0 | 0 | 0 | 0 |
| Dopo 1 Clk | 0 | | 1 | 0 | 1 | 0 | 0 |
| Dopo 2 Clk | 0 | | 0 | 1 | 0 | 1 | 0 |
| Dopo 3 Clk | 0 | | 0 | 0 | 0 | 0 | 1 |

Usato per conversione da serie a parallelo

J-K FLIP-FLOP

- È una variante per evitare gli stati imprevedibili, ad es. $(R,S)=(0,0)$ nel flip-flop R-S oppure $(R,S)=(1,1)$ nei “clocked” e MS flip-flop



| J | K | Q_n | \bar{Q}_n | Q_{n+1} |
|---|---|-------|-------------|-------------|
| 0 | 0 | | | Q_n |
| 1 | 0 | 0 | 1 | 1 |
| | | 1 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 |
| | | 1 | 0 | 0 |
| 1 | 1 | | | \bar{Q}_n |

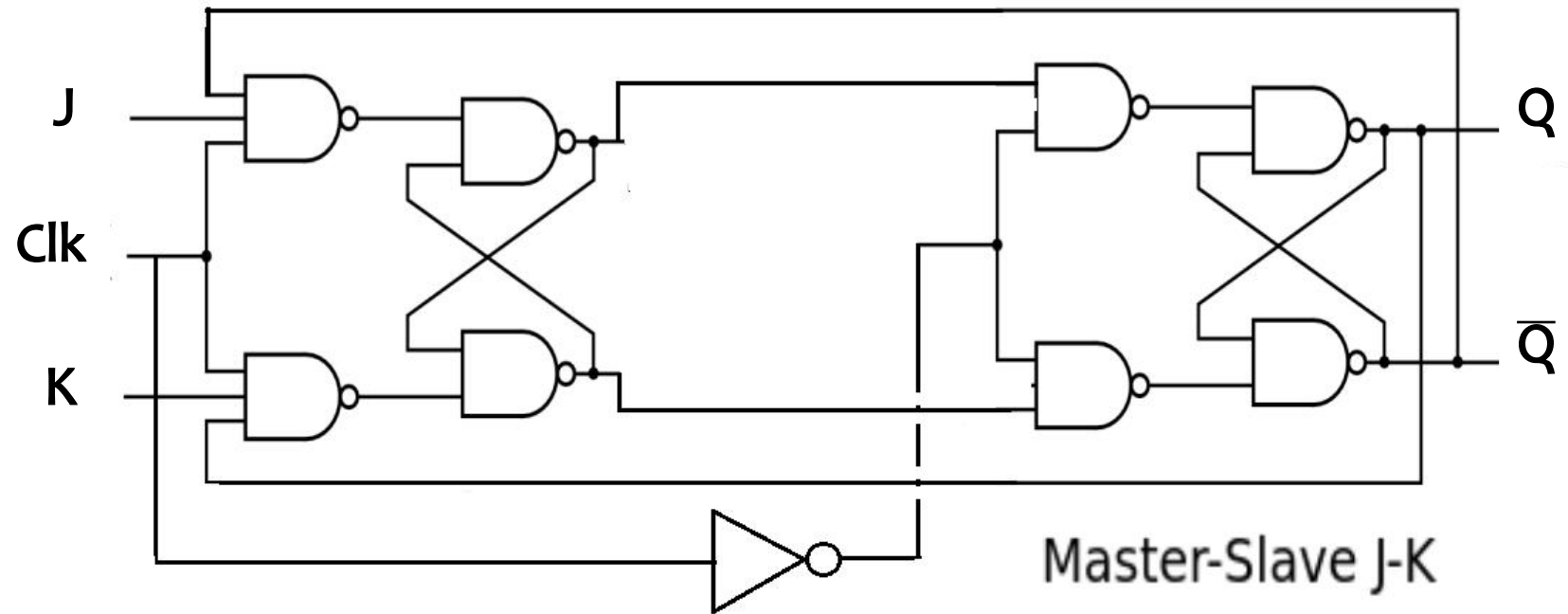
“no change” (IN disabilitati)

“set”

“reset”

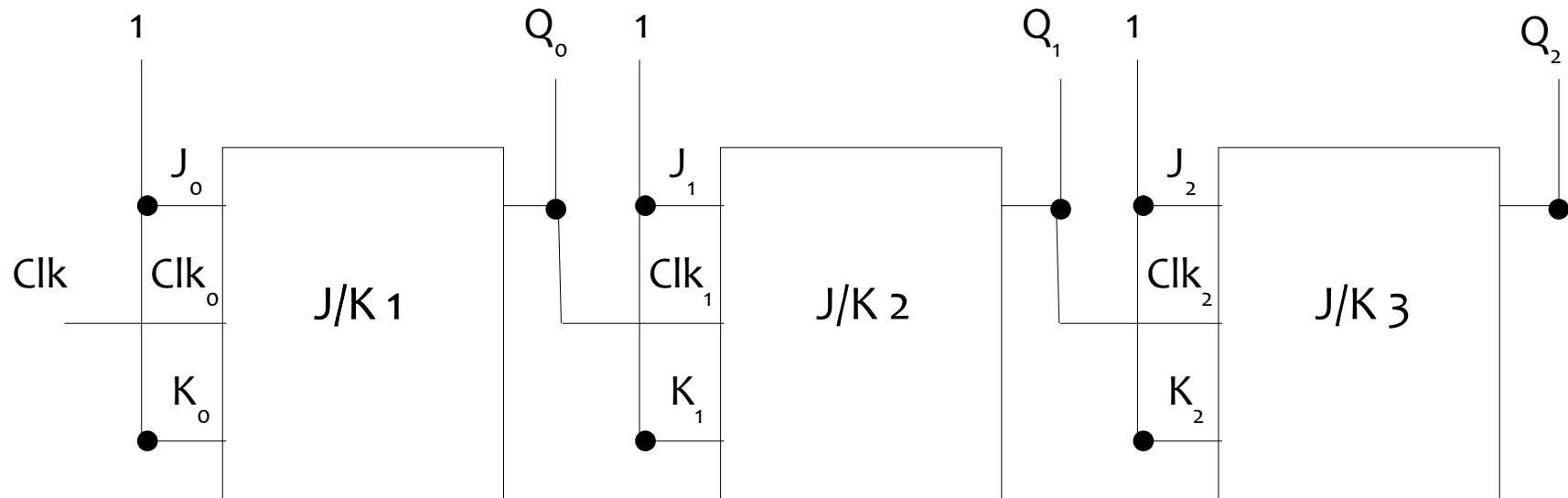
“toggle” (IN controllati da OUT)

MASTER SLAVE J-K FLIP-FLOP



CONTATORI BINARI

- Circuito sequenziale che ritorna nello stato iniziale dopo un certo numero di cicli di clock
Es: “count-up”



| | Q ₂ | Q ₁ | Q ₀ | Q ₂ | Q ₁ | Q ₀ | Q ₂ | Q ₁ | Q ₀ | Q ₂ | Q ₁ | Q ₀ | Q ₂ | Q ₁ | Q ₀ |
|------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| Dopo 1 Clk | 0 | 0 | 1 | | | | | | | | | | | | |
| Dopo 2 Clk | | | | 0 | 1 | 0 | | | | | | | | | |
| Dopo 3 Clk | | | | | | | 0 | 1 | 1 | | | | | | |
| Dopo 4 Clk | | | | | | | | | | 1 | 0 | 0 | | | |
| Dopo 5 Clk | | | | | | | | | | | | | 1 | 0 | 1 |

MATERIALE EXTRA

SOTTRAZIONE E RAPPRESENTAZIONE DI INTERI NEGATIVI

- Sottrazione

Dati A, B interi positivi: $A - B = A + (-B)$

- Rappresentazione di interi negativi usando n bit

1. segno-modulo

- MSB per segno (0 \rightarrow “+”, 1 \rightarrow “-”)
- n-1 bit restanti per modulo
ad es.: n=3 bit

| | |
|----|-----|
| +3 | 011 |
| +2 | 010 |
| +1 | 001 |
| 0 | 000 |
| 0 | 100 |
| -1 | 101 |
| -2 | 110 |
| -3 | 111 |

problema: al valore 0₁₀ corrispondono
due codici binari 100 e 000

2. complemento a 2

- $-B \rightarrow -B + 2^n$
ad es.: n=3 bit, $2^3=8=1000_2$

| | |
|----|-----|
| +3 | 011 |
| +2 | 010 |
| +1 | 001 |
| 0 | 000 |
| -1 | 111 |
| -2 | 110 |
| -3 | 101 |
| -4 | 100 |

prescrizione: $bit_k \rightarrow \overline{bit_k}$ e si somma 1

SOTTRAZIONE E RAPPRESENTAZIONE DI INTERI NEGATIVI

- Proprietà della rappresentazione “complemento a 2”
 - non è una rappresentazione posizionale
 - con n bit si rappresentano 2^n interi $[-2^{n-1}, 2^{n-1}-1]$
 - 0_{10} ha un'unica rappresentazione con tutti “0”
 - -1_{10} si rappresenta con tutti “1”
 - il complemento a 2 del complemento a 2 di X è X : $(X_{2\text{-compl}})_{2\text{-compl}} = X$
- Sottrazione

Dati A, B interi positivi rappresentati con n bit:

$$A - B = \text{mod}(A + (-B) + 2^n - 2^n, 2^n) = A + B_{2\text{-compl}}$$

- $\text{mod}(x, 2^n)$ in quanto solo n bit a disposizione
- somme di numeri dello stesso segno sono corrette se il risultato ha lo stesso segno degli addendi

ALGEBRA DI BOOLE

- Dimostrazione proprietà assorbitiva

$$A + A \cdot B = A \cdot 1 + A \cdot B = A \cdot (1 + B) = A \cdot (1) = A$$

$$A \cdot (A + B) = A \cdot A + A \cdot B = A + A \cdot B = A$$

- Dimostrazione delle Formule di De Morgan

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

Avendo definito $X = \overline{A} \cdot \overline{B}$ si verifica che la validità di De Morgan implica $X \cdot \overline{X} = 0$

$$(\overline{A} \cdot \overline{B}) \cdot \overline{(\overline{A} \cdot \overline{B})} = (\overline{A} \cdot \overline{B}) \cdot \overline{(\overline{A} + \overline{B})} = (\overline{A} \cdot \overline{B}) \cdot (A + B) = \overline{A} \cdot \overline{B} \cdot A + \overline{A} \cdot \overline{B} \cdot B = 0$$

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

Avendo definito $X = \overline{A} + \overline{B}$ si verifica che la validità di De Morgan implica $X \cdot \overline{X} = 0$

$$(\overline{A} + \overline{B}) \cdot \overline{(\overline{A} + \overline{B})} = (\overline{A} + \overline{B}) \cdot \overline{(\overline{A} \cdot \overline{B})} = (\overline{A} + \overline{B}) \cdot (A \cdot B) = \overline{A} \cdot A \cdot B + \overline{B} \cdot A \cdot B = 0$$

FORMULE DI DE MORGAN

- Prescrizione

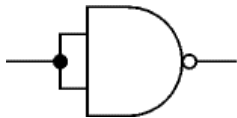
1. $AND \leftrightarrow OR$

2. $A \leftrightarrow \bar{A}, B \leftrightarrow \bar{B}, \dots$

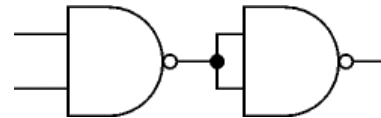
3. $(\text{espressione}) = \overline{(\text{espressione})}$

- In pratica:

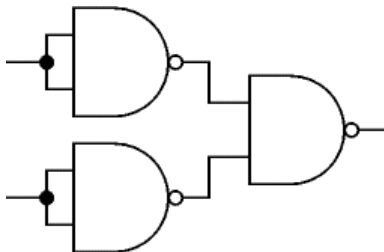
– NOT fatto con NAND



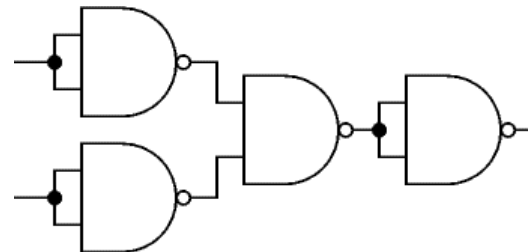
– AND fatto con NAND



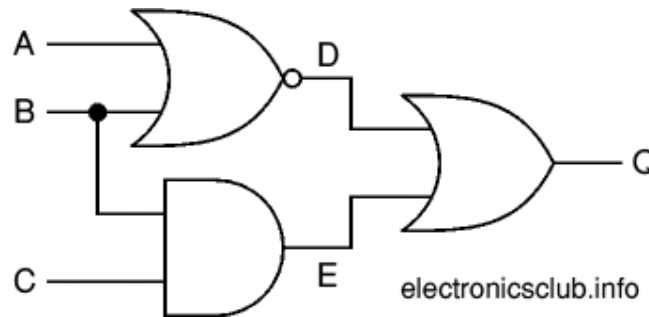
– OR fatto con NAND



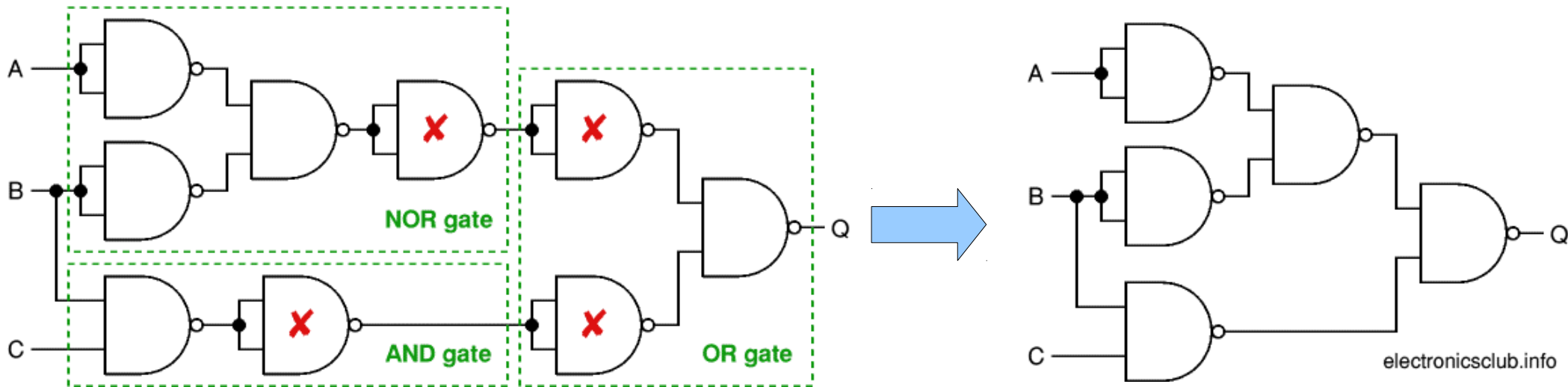
– NOR fatto con NAND



ESEMPIO DI CIRCUITO REALIZZATO SOLO CON NAND



Circuito iniziale



Circuito finale

X coppie di NOT consecutivi si eliminano

FORME CANONICHE

- Definizioni:

- minterm (m_i): AND di tutte le variabili logiche con risultato 1
- maxterm (M_i): OR di tutte le variabili logiche con risultato 0

- Sum-Of-Products*

$$F(A,B,C,\dots) = \sum_i F_i \cdot m_i \quad \text{contano solo i termini con } F_i=1$$

- Products-Of-Sums*

$$F(A,B,C,\dots) = \prod_i (F_i + M_i) \quad \text{contano solo i termini con } F_i=0$$

- Esempio:

| A | B | C | OUT=F |
|---|---|---|-------|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |

Sum-Of-Products

$$F = \overline{A} \cdot \overline{B} \cdot C + A \cdot \overline{B} \cdot C + A \cdot B \cdot C$$

Product-Of-Sums

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

MAPPE DI KARNAUGH

- Metodo per semplificare funzioni logiche complesse
 1. Si individuano le condizioni che danno $F=1$
 2. Si costruisce la mappa con colonne/righe contigue che differiscono solo di una cifra per volta
 3. Si identificano gruppi di 2, 4, 8, 16 celle contigue (orizzontali o verticali) il cui contenuto è "1" e si mette in OR la rispettiva logica

Esempio:

| A | B | C | F |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |

| | AB=00 | 01 | 11 | 10 |
|-----|-------|----|----|----|
| C=0 | 0 | 0 | 0 | 0 |
| =1 | 0 | 1 | 1 | 1 |

sum-of-products

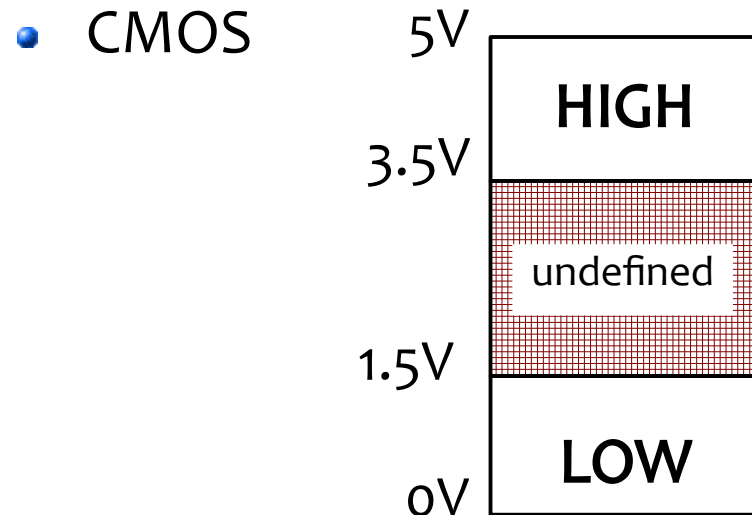
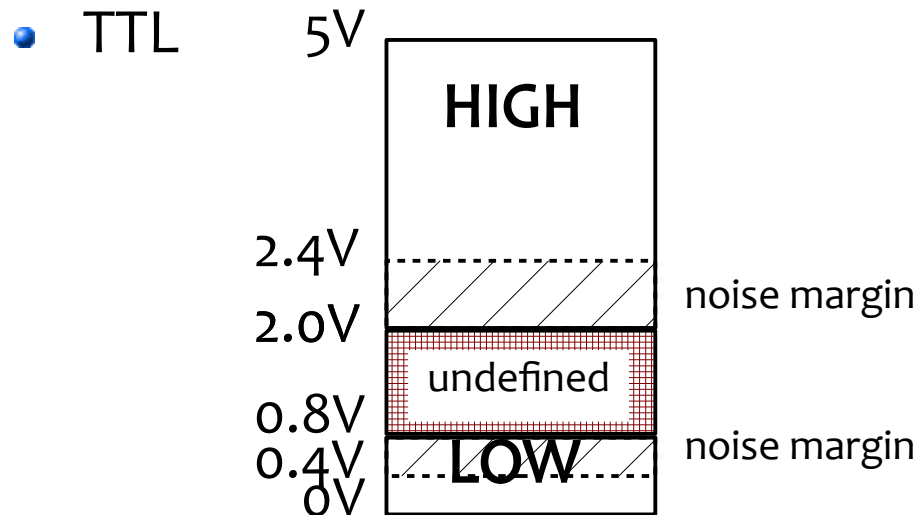
$$F = \bar{A} \cdot B \cdot C + A \cdot \bar{B} \cdot C + A \cdot B \cdot C$$



$$F = B \cdot C + A \cdot C$$

FAMIGLIE DI ICs

| Family | Basic gate | Fanout | Pd (mW/gate) | Noise immunity | Prop. delay (ns/gate) | Clock (MHz) |
|---------|------------|--------|--------------|----------------|-----------------------|-------------|
| TTL | NAND | 10 | 10 | VG | 10 | 35 |
| TTL-H | NAND | 10 | 22 | VG | 6 | 50 |
| TTL-L | NAND | 20 | 1 | VG | 33 | 3 |
| TTL-LS | NAND | 20 | 2 | VG | 9.5 | 45 |
| TTL-S | NAND | 10 | 19 | VG | 3 | 125 |
| TTL-AS | NAND | 40 | 10 | VG | 1.5 | 175 |
| TTL-ALS | NAND | 20 | 1 | VG | 4 | 50 |
| ECL 10K | OR-NOR | 25 | 40-55 | P | 2 | >60 |
| ECL100K | OR-NOR | ?? | 40-55 | P | 0.75 | 600 |
| MOS | NAND | 20 | 0.2-10 | G | 300 | 2 |
| 74C | NOR/NAND | 50 | 0.01/1 | VG | 70 | 10 |
| 74HC | NOR/NAND | 20 | 0.0025/0.6 | VG | 18 | 60 |
| 74HCT | NOR/NAND | 20 | 0.0025/0.6 | VG | 18 | 60 |
| 74AC | NOR/NAND | 50 | 0.005/0.75 | VG | 5.25 | 100 |
| 74ACT | NOR/NAND | 50 | 0.005/0.75 | VG | 4.75 | 100 |



FAMIGLIE DI ICs

Serie

- TTL
 - “ “ standard
 - S Schottky
 - LS low power Schottky
 - AS advanced Schottky
 - F fast advanced Schottky
- CMOS
 - C standard
 - HC high speed
 - AC advanced performance
 - F fast performance

↓

PPVVSSXXP

Produttore

- S Signetics
- DM National Semiconductor
- SN Texas Instruments

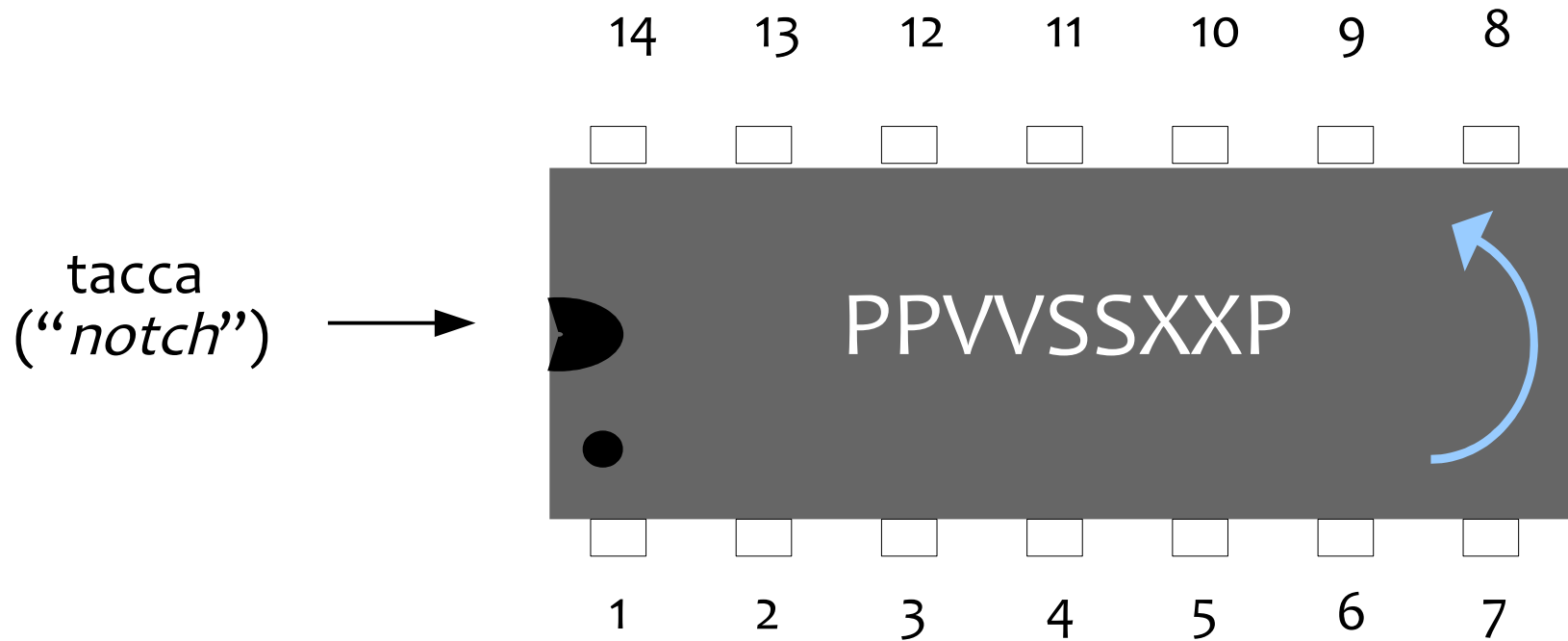
Package

- N Plastic Dual in Line (DIP)
- J Ceramic Dual In Line (CDIP)
- W Ceramic flat-pack
- D Surface mounted small outline plastic package

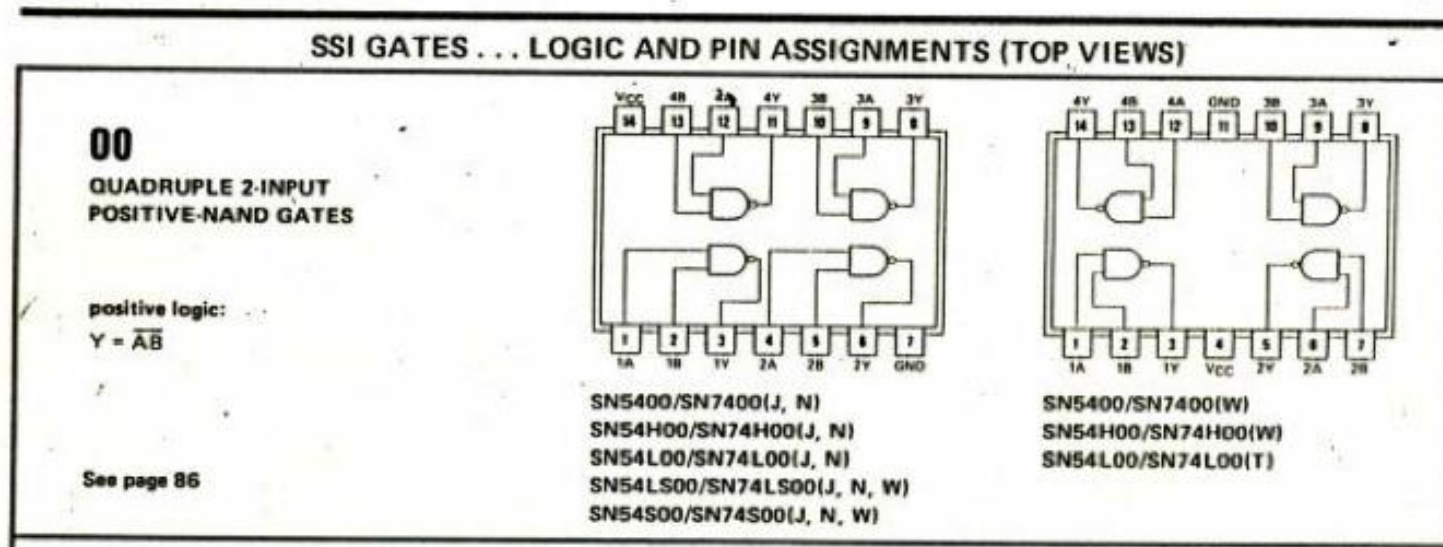
Versione

- 54 applicazioni militari (-55 °C, + 125 °C)
- 74 commerciale (0 °C, +70 °C)

NUMERAZIONE PIN ICs

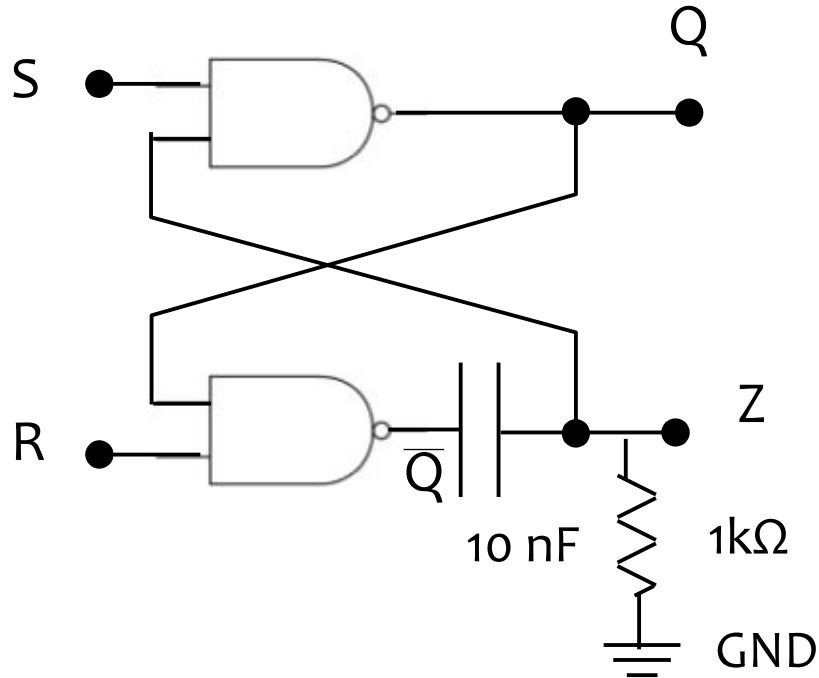


- In laboratorio: usare un IC 7400 per costruire un flip-flop R-S



- Verificare la tabella di verità
 - 1) con pulsanti e LED
 - 2) con pulsanti e oscilloscopio
 - 3) con generatore di funzioni (onda quadra) e oscilloscopio

MULTIVIBRATORE MONOSTABILE

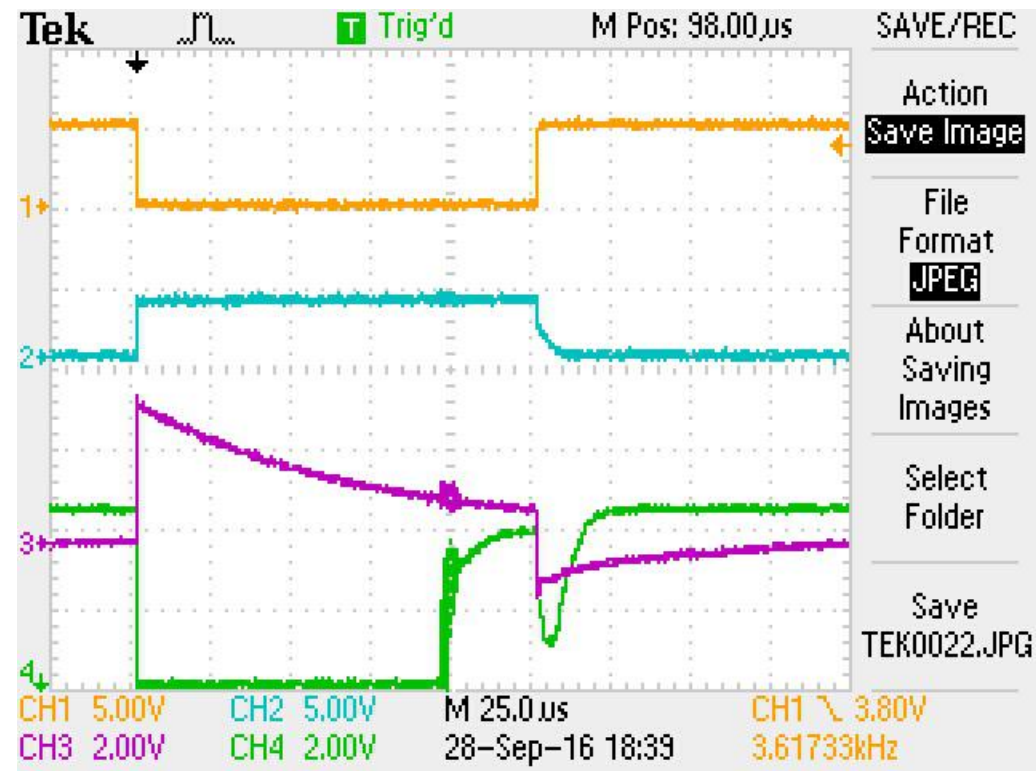
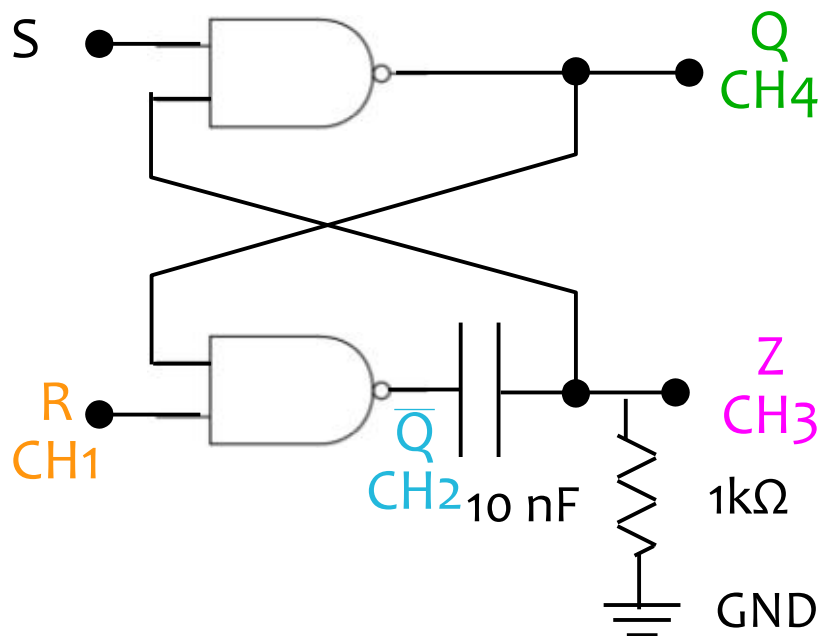


| S | R | Q | \bar{Q} | C |
|---|---|---|-----------|---------|
| 1 | 1 | 1 | 0 | scarico |

- Simile ad un flip-flop R-S ma con l'uscita di un NAND filtrata da un condensatore
- In laboratorio/1: verificare il comportamento monostabile analizzando le uscite Q e Z quando R transisce $1 \rightarrow 0 \rightarrow 1$ e osservare la tensione di commutazione
 - R \rightarrow generatore di funzioni
 - Q, Z \rightarrow oscilloscopio

NB: scegliere $T \approx 2 \tau_{RC}$ e verificare il comportamento al variare della simmetria dell'onda quadra ($T_{low} \ll \tau_{RC}$ oppure $T_{low} \gg \tau_{RC}$)

MULTIVIBRATORE MONOSTABILE

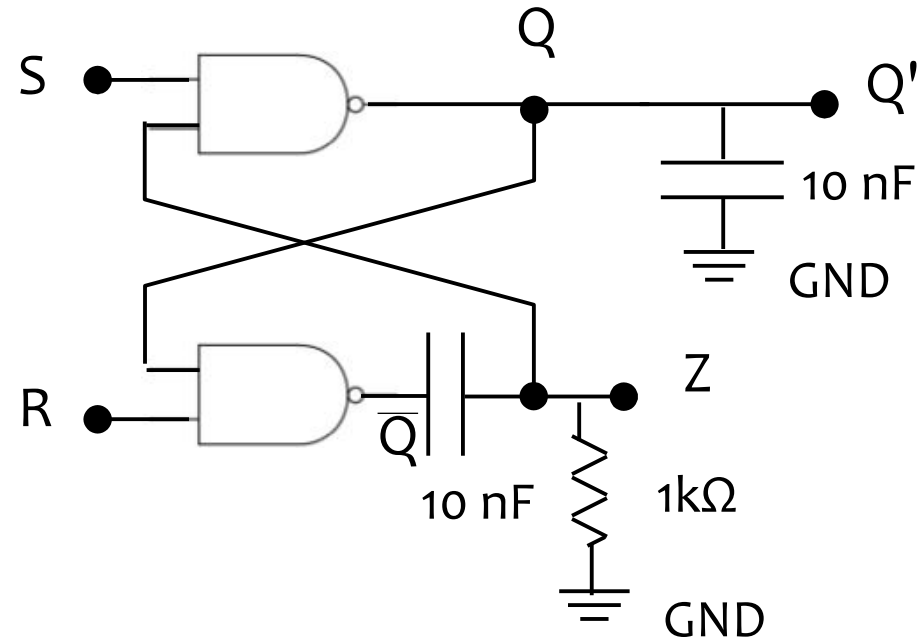
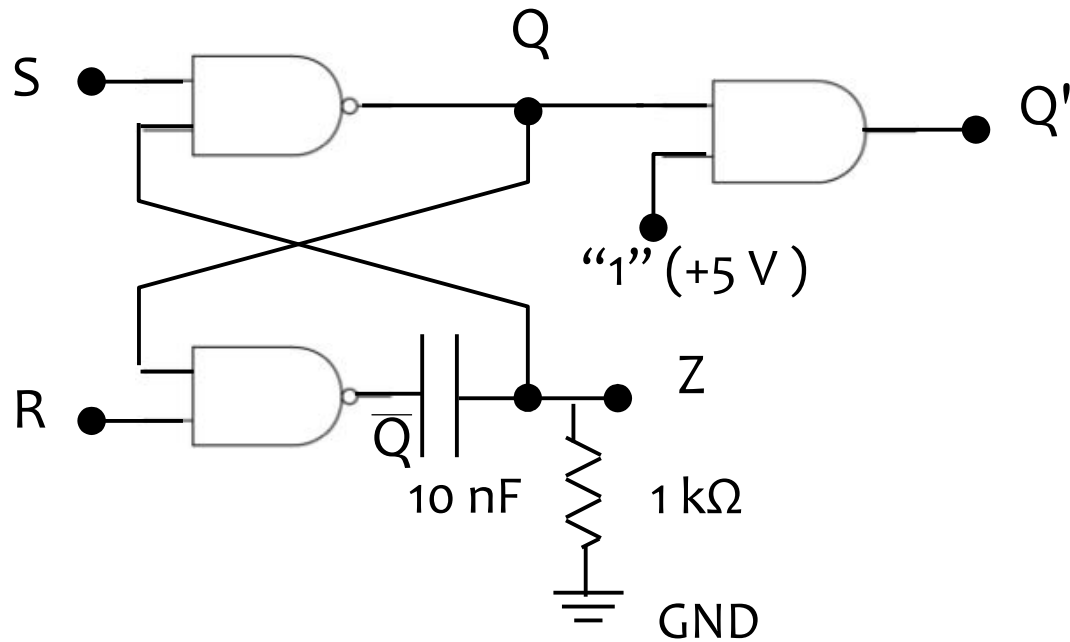


- In laboratorio/1: verificare il comportamento monostabile analizzando le uscite Q e Z quando R transisce $1 \rightarrow 0 \rightarrow 1$ e osservare la tensione di commutazione

- R \rightarrow generatore di funzioni
- Q, Z \rightarrow oscilloscopio

NB: scegliere $T \approx 2 \tau_{RC}$ e verificare il comportamento al variare della simmetria dell'onda quadra ($T_{low} \ll \tau_{RC}$ oppure $T_{low} \gg \tau_{RC}$)

- In laboratorio/2: correggere la distorsione legata alla commutazione di R



- In laboratorio/3: usare il multivibratore monostabile per produrre un impulso di larghezza predefinita ("gate")

