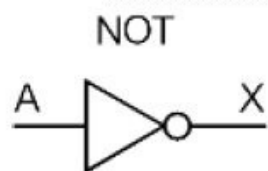


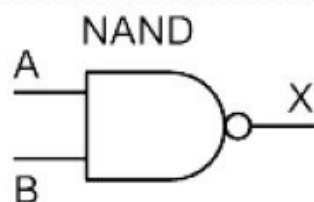


Cenni circuiti, reti combinatorie, reti sequenziali

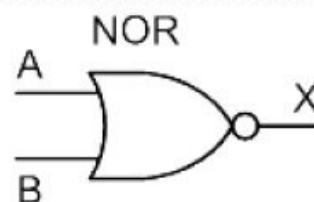
Porte logiche di base



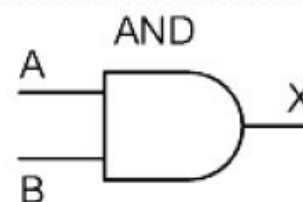
A	X
0	1
1	0



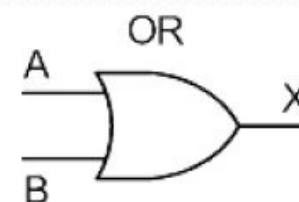
A	B	X
0	0	1
0	1	1
1	0	1
1	1	0



A	B	X
0	0	1
0	1	0
1	0	0
1	1	0

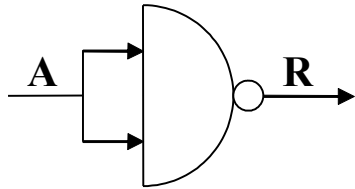


A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

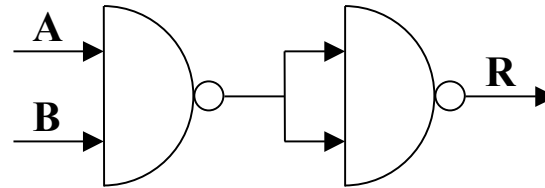


A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

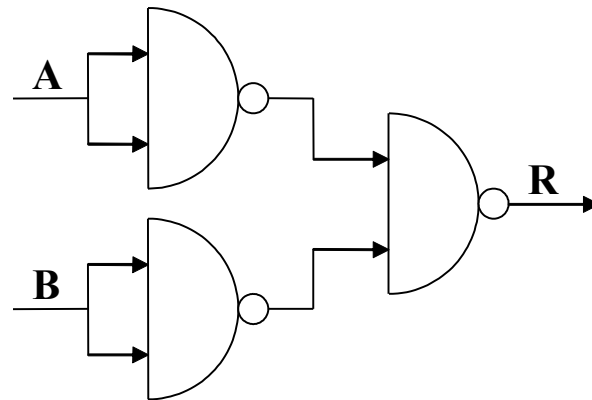
NOT



AND



OR



Quindi NAND o NOR sono complete → circuiti con solo porte NAND o solo porte NOR.



Reti combinatorie

- Rete combinatoria: insieme di porte logiche connesse il cui output in un certo istante è funzione solo dell'input in quell'istante
- N input binari e m output binari
- Ad ogni combinazione di valori di ingresso corrisponde **una ed una sola** combinazione di valori di uscita

Reti combinatorie (segue)

- Vediamo alcuni esempi di circuiti:
 - ✓ I segnali sono discretizzati e di solito assumono solo due stati:

0 / **FALSO** / [0..1] Volt



1 / **VERO** / [2..5] Volt

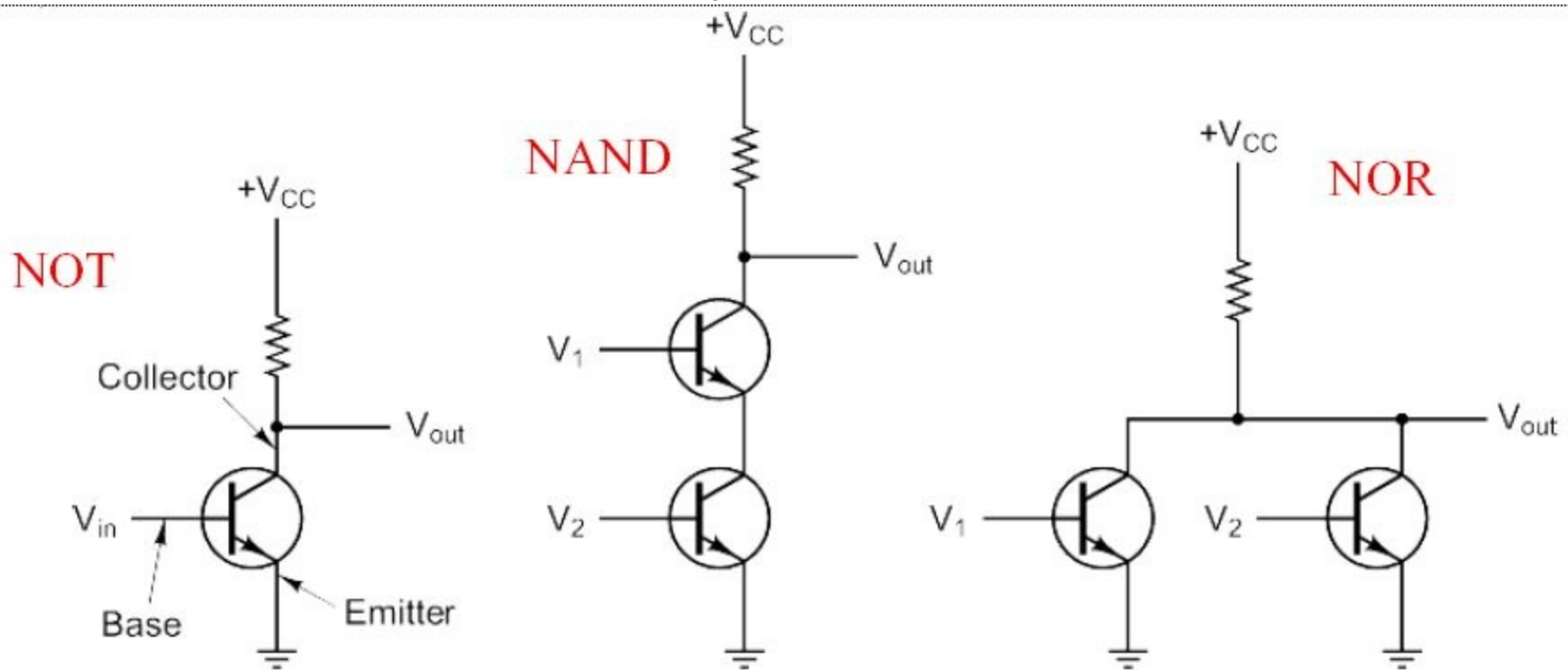


- ✓ I circuiti più complessi sono realizzati attraverso la combinazione di circuiti semplici (porte logiche)

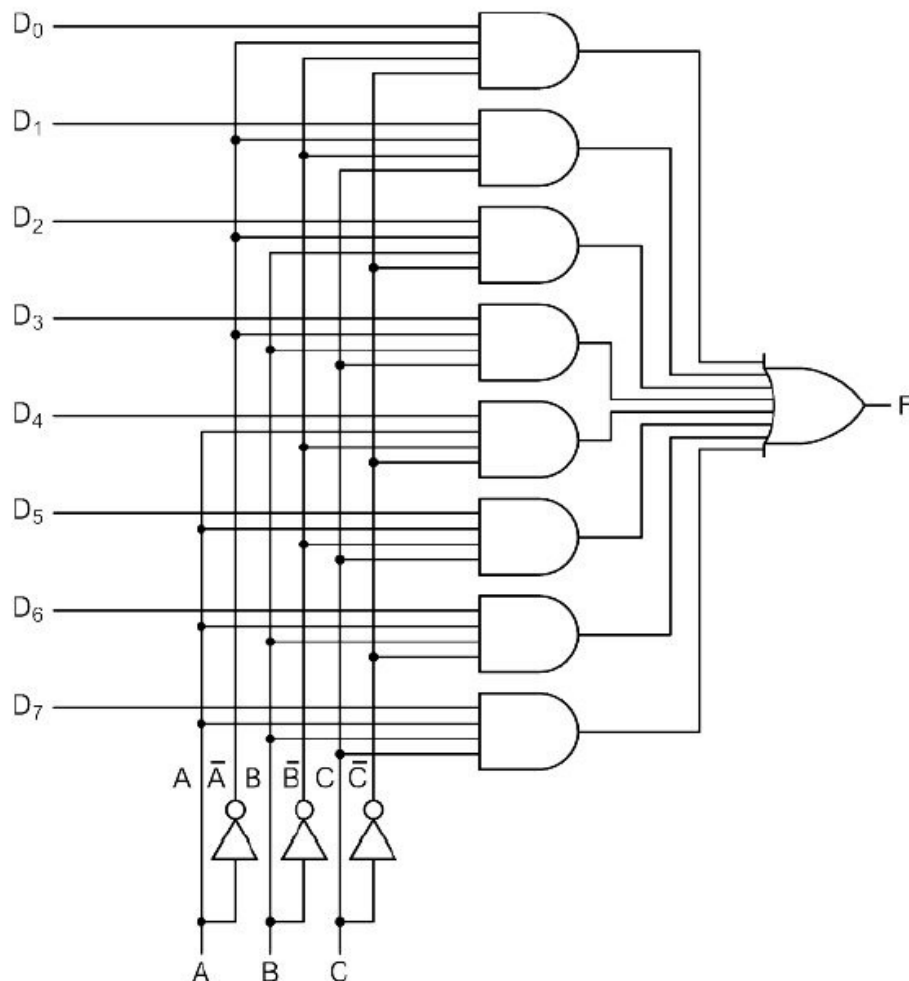
Reti combinatorie (segue)

■ Porte Logiche:

- ✓ Sono realizzate tramite transistor (sono in pratica interruttori automatici)

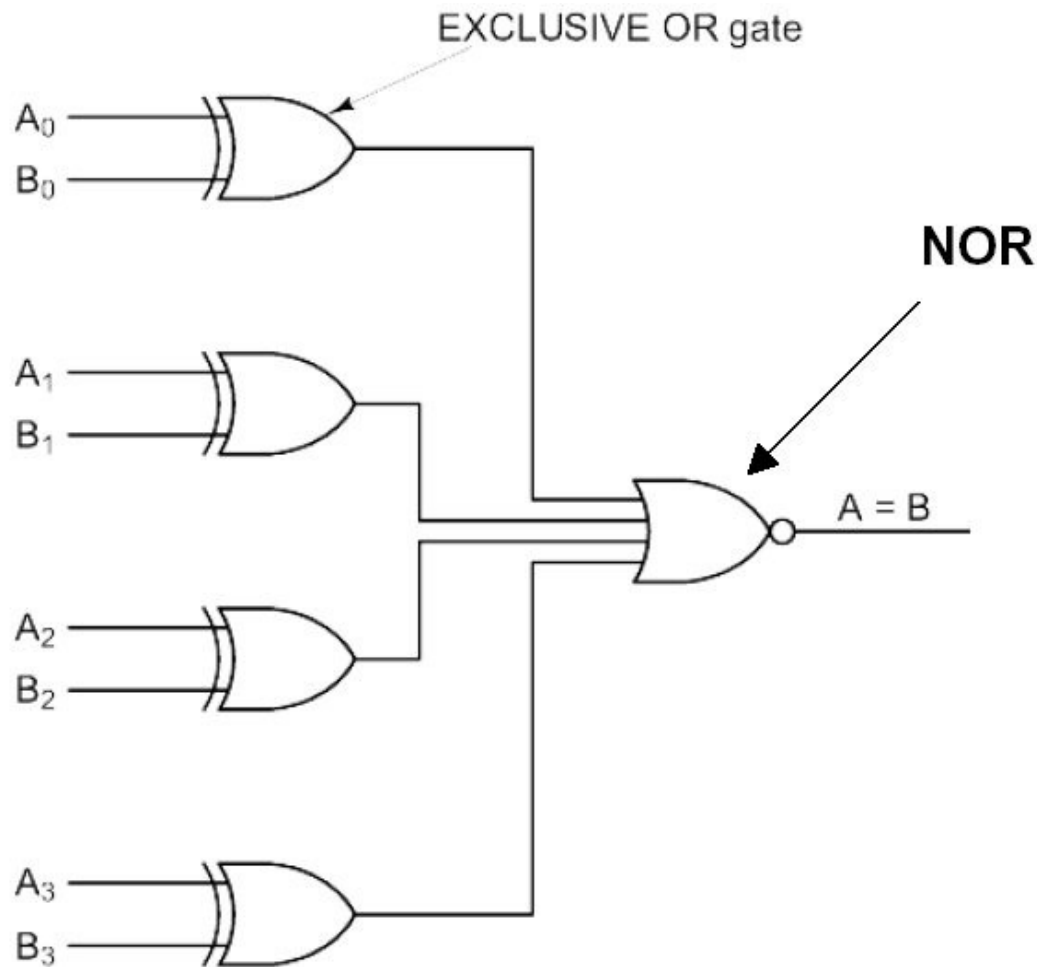


Multiplexer (o selettore) 2^n a 1



- Solo uno degli ingressi viene trasferito all'output
- n ingressi di controllo: indicano l'ingresso da trasferire
 - ✓ 2^n linee di input
($D_0 - D_7$)
 - ✓ n linee di controllo
(A, B, C)
 - ✓ 1 linea di output (F)
- ✓ Per ogni combinazione degli ingressi di controllo, $2^n - 1$ delle porte AND hanno uscita 0, l'altra fa uscire l'ingresso

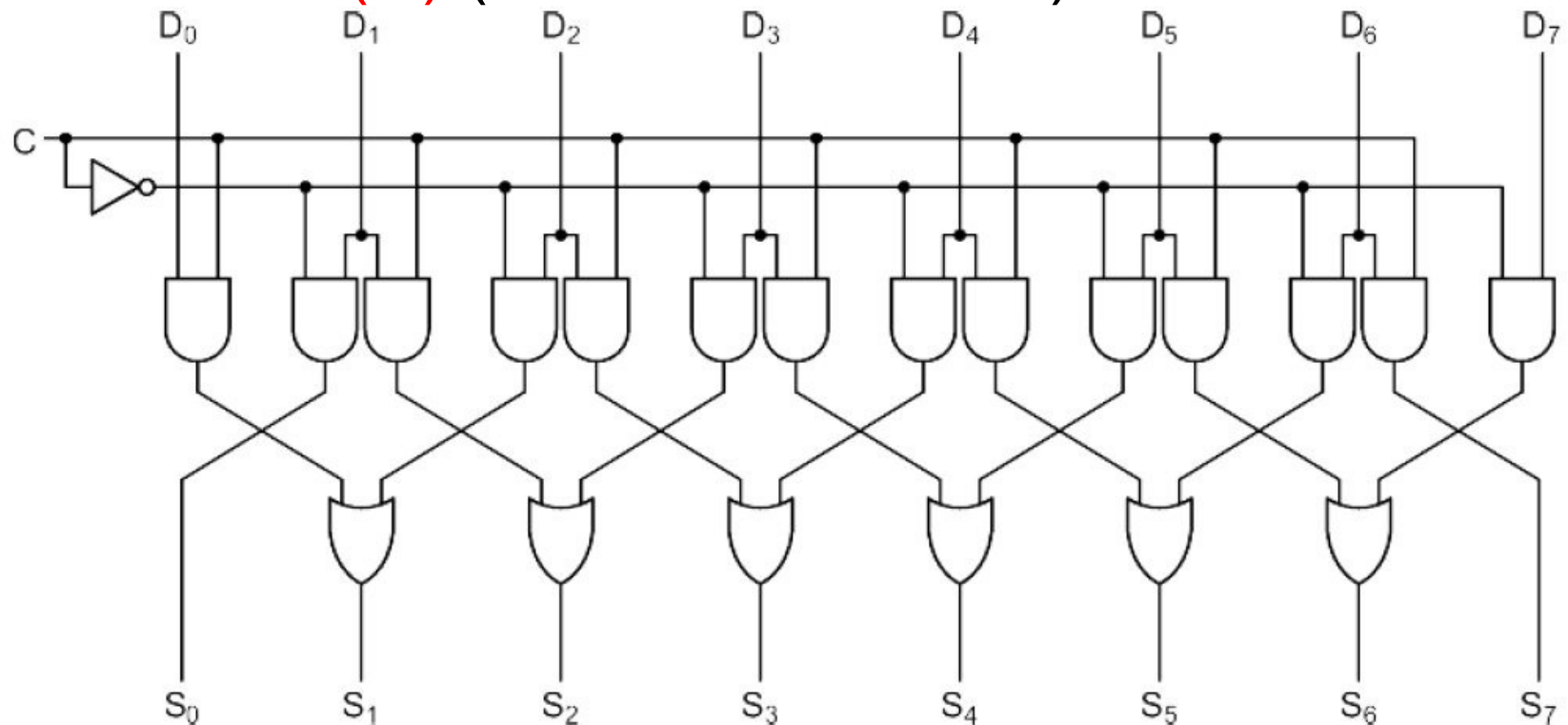
Comparatore a più bit



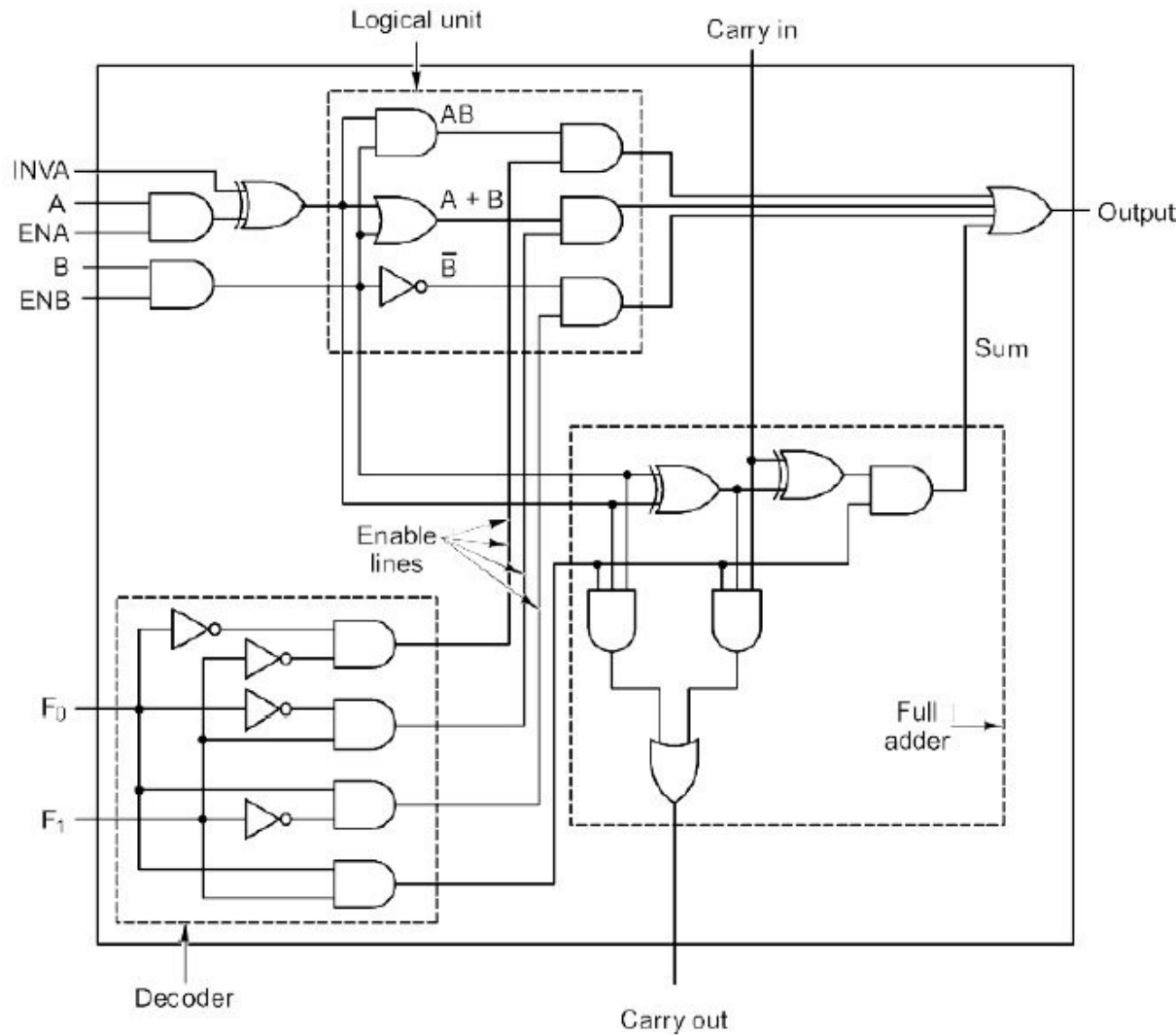
- ✓ Comparatori ad 1 bit vengono collegati tramite una porta NOR
- ✓ L'output vale 1 solo se tutti gli output dei singoli comparatori ad 1 bit valgono 0
- ✓ $(A_i = B_i)$ per ogni i , cioè $A = B$

Traslatore (shifter)

- ✓ Trasla i bit in ingresso (**D**) di una posizione, a sinistra o a destra a seconda del valore del bit di controllo (**C**) ($C=1$ shift a destra)



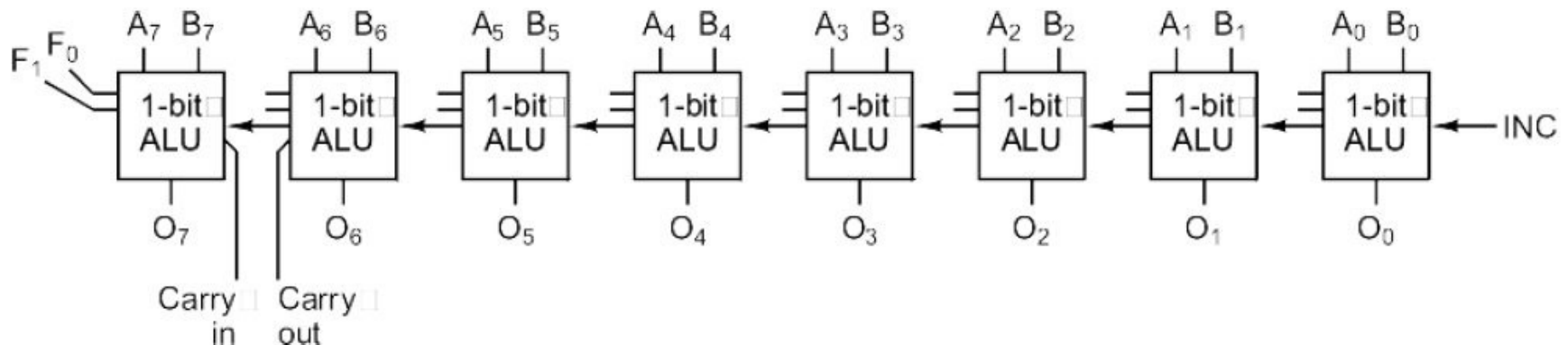
ALU a 1 bit



- ALU ad 1bit che realizza 4 operazioni (selezionate da F_0 e F_1)
- AB , A or B , $\text{not}(B)$, $A+B$
- ENA , ENB : per forzare a 0 gli input A e B
- $INVA$, $INVB$: per invertire gli input

ALU a n bit

- ✓ Si ottiene concatenando n ALU ad 1 bit
- ✓ F_0 e F_1 collegati a tutte le ALU
- ✓ Riporto intermedio propagato da una ALU alla successiva
- ✓ INC (corrispondente al carry in della ALU "0") permette di sommare 1 al risultato in caso di addizione





Reti combinatorie

- Utili per implementare la ALU e la connessione tra parti della CPU
- Non sono in grado di memorizzare uno stato, quindi non possono essere usate per implementare la memoria
- Per questo servono le reti sequenziali
 - L'output dipende non solo dall'input corrente, ma anche dalla storia passata degli input



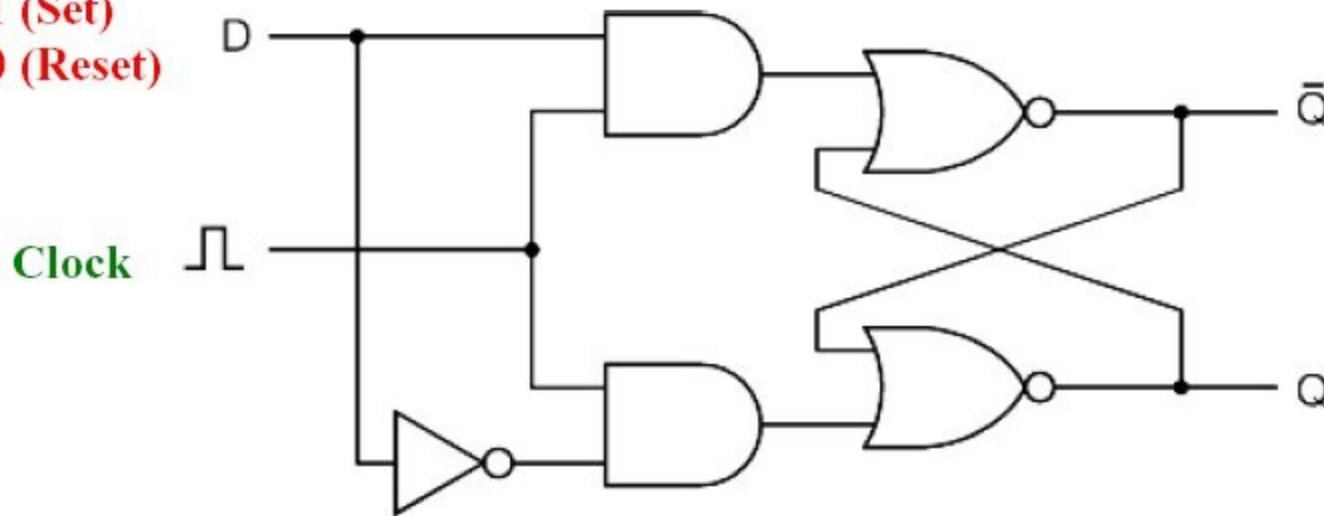
Flip flop

- Forma più semplice di una rete sequenziale
- Tanti tipi, ma due proprietà per tutti:
 - Bistabili:
 - Possono trovarsi in uno di due stati diversi
 - In assenza di input, rimangono nello stato in cui sono
 - Memoria per un bit
 - Due output
 - Uno è sempre il complemento dell'altro

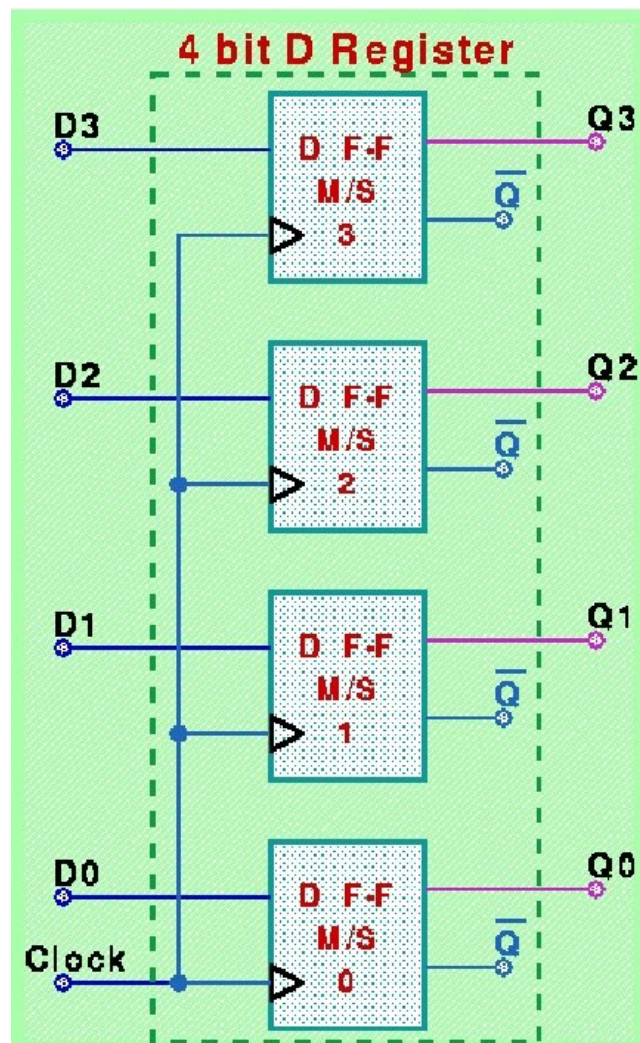
Flip flop D

- Un solo input (D)
- Usa segnale di clock per stabilizzare l'output (sincronizzazione)
- Quando clock = 0, gli output dei due AND sono 0 (stato stabile)
- Quando clock = 1, gli input sono uno l'opposto dell'altro → $Q = D$

D = 1 (Set)
D = 0 (Reset)



Registro di tipo D



- è il circuito sincrono più semplice che realizza un registro
- Memorizzazione (store): dati presentati in ingresso e clock da 0 a 1 (uscita riproduce ingresso)
- Mantenimento (hold): clock da 1 a 0 (poi costante); l'uscita rimane invariata indipendentemente dal valore degli ingressi