

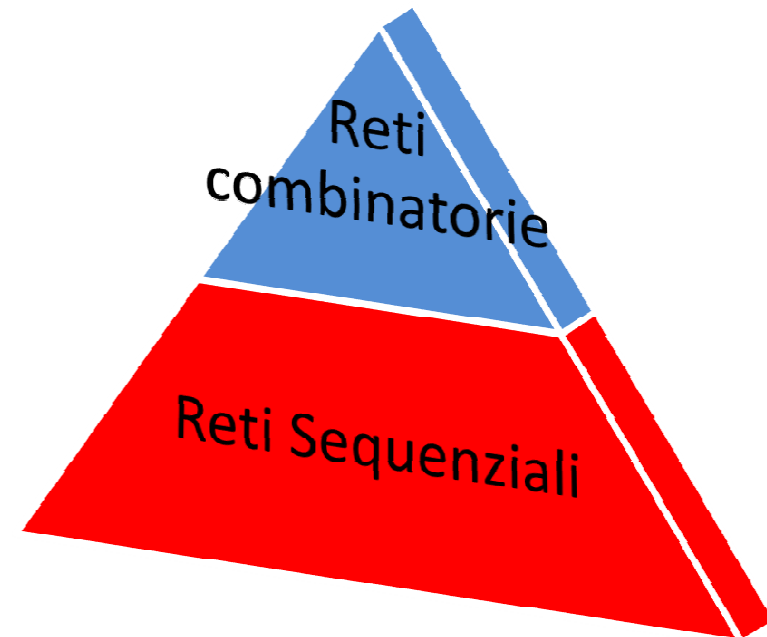


Architettura degli elaboratori

Reti combinatorie

ARGOMENTI DELLA LEZIONE

- ☐ Rete combinatoria
 - ☐ Algebra di Boole
 - ☐ Transistori
 - ☐ Porte logiche elementari
 - ☐ Esempi di reti combinatorie (decodificatore, codificatore)
 - ☐ Circuiti di calcolo elementari (addizionatore, sottrattore e comparatore)
- ☐ Rete sequenziale
 - ☐ Elementi: latch, modello di Huffman, automi a stati finiti
 - ☐ Esempi di reti sequenziali (registro contatore, registro a scorrimento)
 - ☐ Registro di memorizzazione e sistemi di interconnessione (punto a punto, multiplexer, demultiplexer, mesh, bus)





Architettura degli elaboratori

Reti combinatorie

Reti combinatorie

- ❑ Nel 1936, con la tesi di laurea *Un'analisi simbolica dei circuiti a relè e a commutatori*, il matematico statunitense Claude Elwood Shannon dimostrò che sfruttando l'Algebra di Boole potevano essere realizzate delle reti combinatorie in grado di svolgere operazioni logiche ed aritmetiche su operandi binari



Claude Elwood Shannon



Algebra di Boole

- Un insieme finito di elementi $A:\{0,1\}$, chiamato **supporto**, e due operatori indicati con **somma** \oplus e **prodotto** \otimes , aventi per dominio $A \times A$ e codominio A , definiscono un' **Algebra di Boole**

ASSIOMI

1. Commutazione di somma e di prodotto:

$$(x \oplus y) = (y \oplus x) \text{ e } (x \otimes y) = (y \otimes x)$$

2. Distributività di somma e di prodotto:

$$(x \oplus (y \otimes z)) = (x \oplus y) \otimes (x \oplus z)$$

$$(x \otimes (y \oplus z)) = (x \otimes y) \oplus (x \otimes z)$$

3. Esistenza dell'elemento neutro per la somma (0) e per il prodotto (1):

$$(x \oplus 0) = x \text{ e } (x \otimes 1) = x$$

4. Complementarietà degli elementi, ovvero ogni $x \in A$ ammette uno, e un solo, complementare $\sim x$ tale che:

$$(x \oplus \sim x) = 1 \text{ e } (x \otimes \sim x) = 0$$

5. Idempotenza:

$$(x \oplus x) = x \text{ e } (x \otimes x) = x$$



Espressione Booleana

- ❑ In modo induttivo si possono definire le **espressioni booleane** (E) come parole composte da operatori booleani, parentesi, costanti (cioè gli elementi di A) e **variabili booleane** (x_1, \dots, x_n) ovvero delle entità non note che possono assumere un qualunque valore di A.
- ❑ In particolare si ha che:
 1. gli elementi di A (le costanti) e le variabili booleane sono espressioni booleane;
 2. se E, E_1 e E_2 sono espressioni booleane allora anche $(E_1 \oplus E_2)$, $(E_1 \otimes E_2)$ e $(\sim E)$ sono espressioni booleane;
 3. non esistono altre espressioni booleane oltre a quelle generate da un numero finito di applicazioni delle regole 1 e 2.

ESEMPIO

$\sim(((x_1 \oplus x_2) \otimes x_3)$ è una espressione booleane

$\oplus x_1 \otimes)(\otimes x_2$ non è una espressione booleana



Espressione Booleana

Proprietà

- ❑ Si definisce l'**espressione duale** di E ($\sim E$) quella ottenuta scambiando le somme con i prodotti e l'elemento I con O
- ❑ Ad esempio l'espressione $E = (x_1 \oplus x_2 \oplus I) \otimes (\sim x_3 \oplus O)$ ha come duale $\sim E = (x_1 \otimes x_2 \otimes O) \oplus (\sim x_3 \otimes I)$.

Altre proprietà:

$$(E_1 \oplus E_2) \oplus E_3 = E_1 \oplus (E_2 \oplus E_3) \text{ e } (E_1 \otimes E_2) \otimes E_3 = E_1 \otimes (E_2 \otimes E_3)$$

$$(E_1 \oplus E_1) = E_1 \text{ e } (E_1 \otimes E_1) = E_1$$

$$(E_1 \oplus E_2) = E_2 \text{ se e solo se } (E_1 \otimes E_2) = E_1$$

$$(E_1 \oplus I) = I \text{ e } (E_1 \otimes O) = O$$

$$(E_1 \oplus (E_1 \otimes E_2)) = E_1 \text{ e } (E_1 \otimes (E_1 \oplus E_2)) = E_1$$

$$(E_1 \oplus (\sim E_1 \otimes E_2)) = E_1 \oplus E_2 \text{ e } (E_1 \otimes (\sim E_1 \oplus E_2)) = E_1 \otimes E_2$$

$$\sim O = I \text{ e } \sim I = O$$

$$(\sim(\sim E_1)) = E_1$$

$$(E_1 \oplus (\sim E_1)) = I \text{ e } (E_1 \otimes (\sim E_1)) = O$$

Teorema De Morgan:

$$\sim(E_1 \oplus E_2) = (\sim E_1) \otimes (\sim E_2) \text{ e } \sim(E_1 \otimes E_2) = (\sim E_1) \oplus (\sim E_2)$$



Funzione Booleana

- ❑ Una **funzione booleana** è una legge che manipola le costanti e le variabili booleane
- ❑ Formalmente una funzione booleana $f(x_1, \dots, x_n)$ di n variabili booleane è una relazione che associa ad elementi del dominio $\{0,1\}^n$ un elemento del codominio $\{0,1\}$.

$$f(x_1, x_2, x_3) = \begin{cases} 1 & \text{se } x_1 = x_2 = x_3 \\ 0 & \text{altrimenti} \end{cases}$$



Tabella Booleana

- Un altro modo per identificare una funzione booleana è la **tabella della verità** che elenca, per ognuna delle 2^n possibili assegnazioni, il corrispondente valore della funzione

x_1	x_2	x_3	$f(x_1, x_2, x_3)$
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	1



Tabella Booleana

- ❑ Definendo **letterali** le variabili nella forma semplice (o in quella complementata) presenti in un'espressione booleana; si indica, considerando una funzione f di n variabili, con **minterm** il prodotto di n letterali di f e con **maxterm** la somma di n letterali di f .
- ❑ Per la realizzazione di circuiti elettrici si fa riferimento alla **forma canonica disgiuntiva** e alla **forma canonica congiuntiva** delle espressioni. Un'espressione in forma canonica disgiuntiva è la somma di tutti e soli i minterm che coincidono ad assegnazioni per cui la funzione ha valore 1. Un'espressione in forma canonica congiuntiva, invece, è definita come il prodotto di tutti i maxterm che corrispondono ad assegnazioni per cui la funzione ha valore 0



Tabella Booleana

x_1	x_2	x_3	$f(x_1, x_2, x_3)$
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	1

Forma canonica disgiuntiva

$$(\sim x_1 \otimes \sim x_2 \otimes \sim x_3) \oplus (x_1 \otimes x_2 \otimes x_3)$$

Forma canonica congiuntiva

$$(\sim x_1 \oplus \sim x_2 \oplus x_3) \otimes (\sim x_1 \oplus x_2 \oplus \sim x_3) \otimes (\sim x_1 \oplus x_2 \oplus x_3) \otimes (x_1 \oplus \sim x_2 \oplus \sim x_3) \otimes (x_1 \oplus \sim x_2 \oplus x_3) \otimes (x_1 \oplus \sim x_2 \oplus x_3)$$



Tabella Booleana

Minimizzazione

Grazie a delle **tecniche di minimizzazione** (riduzione per assiomi e proprietà; Mappe di Karnaugh; Metodo di Quine-McCluskey) è possibile derivare un'espressione minima dalle forme canoniche.

La minimizzazione dei letterali è un nodo cruciale nella progettazione, perché riduce il numero di porte logiche e d'ingressi da dover realizzare nel momento in cui si produce il corrispondente circuito elettronico.



Porte logiche

- ❑ Una volta che è definita una funzione booleana è necessario trovare dei componenti fisici che ne consentano la fabbricazione
- ❑ In particolare è necessario utilizzare degli elementi in grado di soddisfare gli operatori di somma, prodotto e dualità booleana; ovvero si ha bisogno di **porte logiche**.

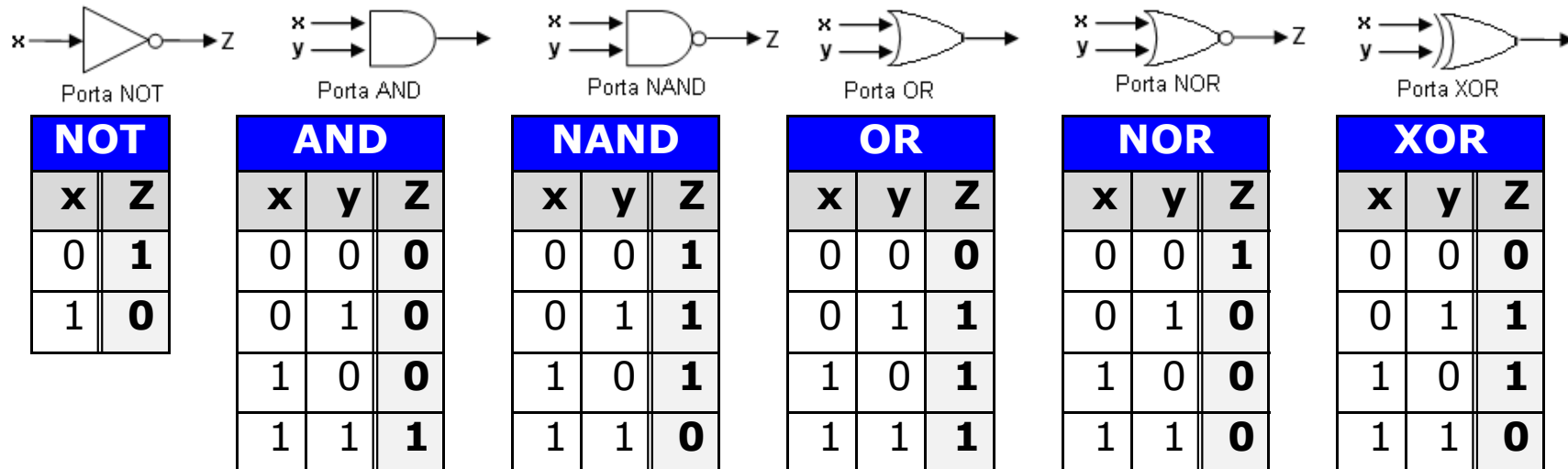


Porte logiche

- ❑ Una **porta logica** (*gate*) è un elemento di calcolo, realizzato mediante un componente elettromeccanico (relé, anni 1920-1940) o elettrico (*transistor*, dal 1950) avente un determinato numero di **linee di ingresso** (*fan-in*) ed una **linea di uscita** (*fan-out*) che, eventualmente, può essere collegata all'entrata di una o più porte (eccetto quella da cui esce)
- ❑ I segnali applicati alle linee di ingresso e di uscita sono segnali elettrici e si possono associare a essi due valori convenzionali:
 - ❑ **1** (presenza di segnale o segnale alto: [2V;5V])
 - ❑ **0** (assenza di segnale o segnale basso: [0V;1V])

Porte logiche

- ❑ Ciascuna porta logica risolve una funzione (o **tabella della verità**)
- ❑ Le **principali porte logiche** utilizzate sono: NOT, OR, NOR, AND, NAND e XOR





Porte Logiche

Tecnologie usate

- ❑ Le tecnologie microelettroniche oggi più usate per la realizzazione di porte logiche sono:

- ❑ BJT (Bipolar Junction Transistor): transistor bipolari

- ❑ TTL (transistor-transistor logic)
- ❑ ECL (emitter coupled logic)

- ❑ MOS (field-effect transistor)

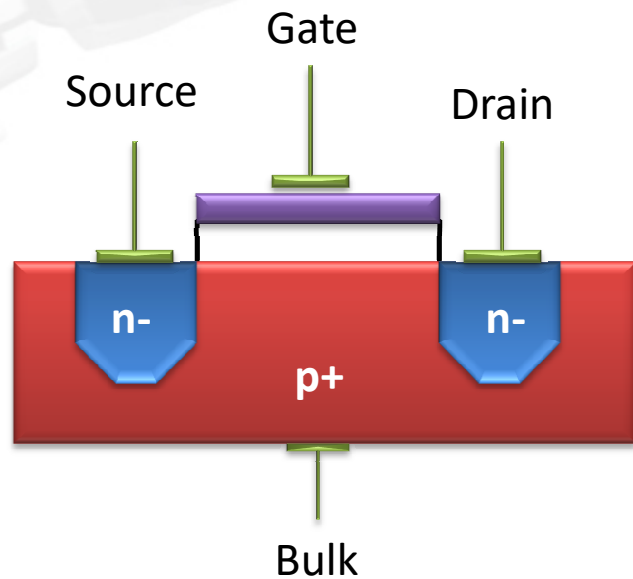
- ❑ pMOS
- ❑ nMOS
- ❑ CMOS
- ❑ BicMOS

TECNOLOGIA	PRO	CONTRO
BJT	Veloci	•Consumano energia •Dimensione superiori ai MOS
MOS	•Richiedono meno energia •Più piccole	Lente

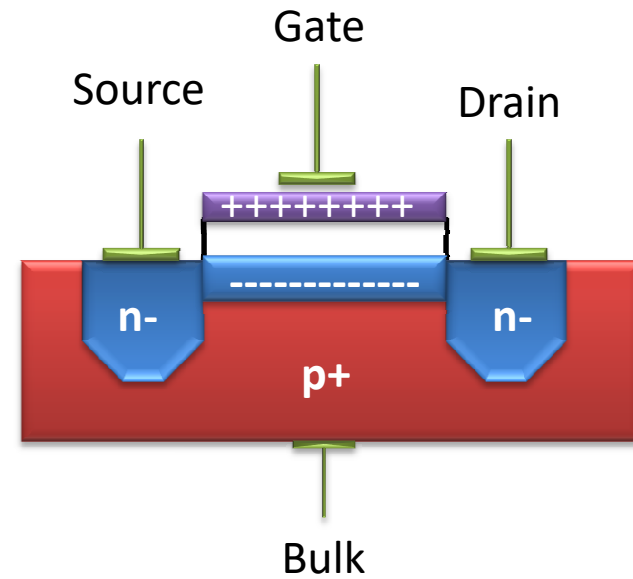
Porte Logiche

Tecnologia MOS

Applicazione di alta tensione al gate



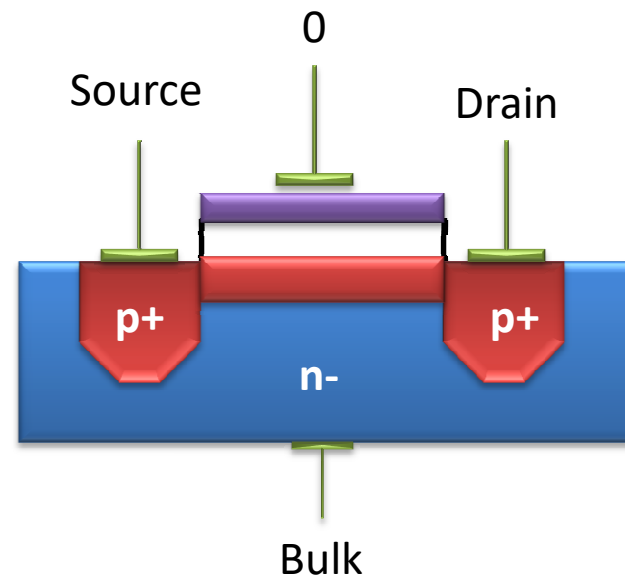
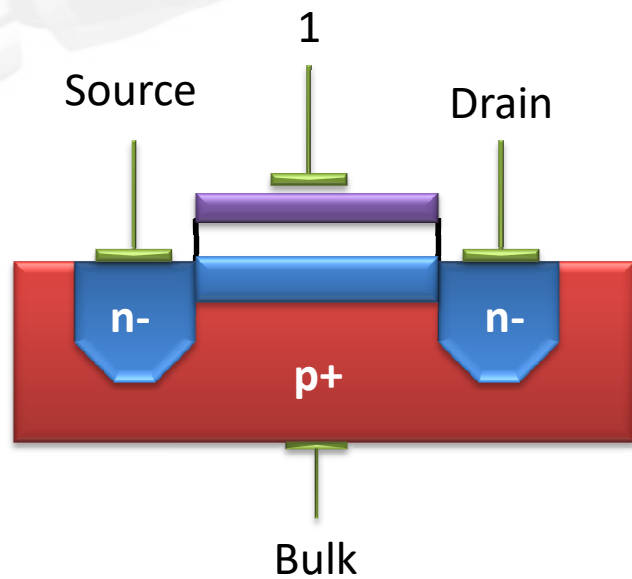
Nessun traffico di corrente



Traffico di corrente da source a drain

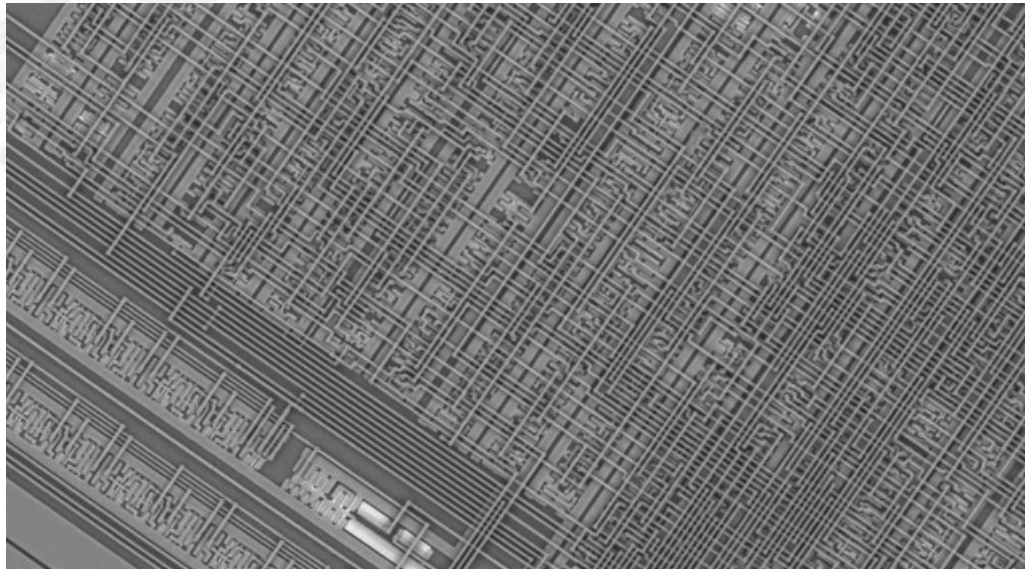
Porte Logiche

Tecnologie nMOS e pMOS

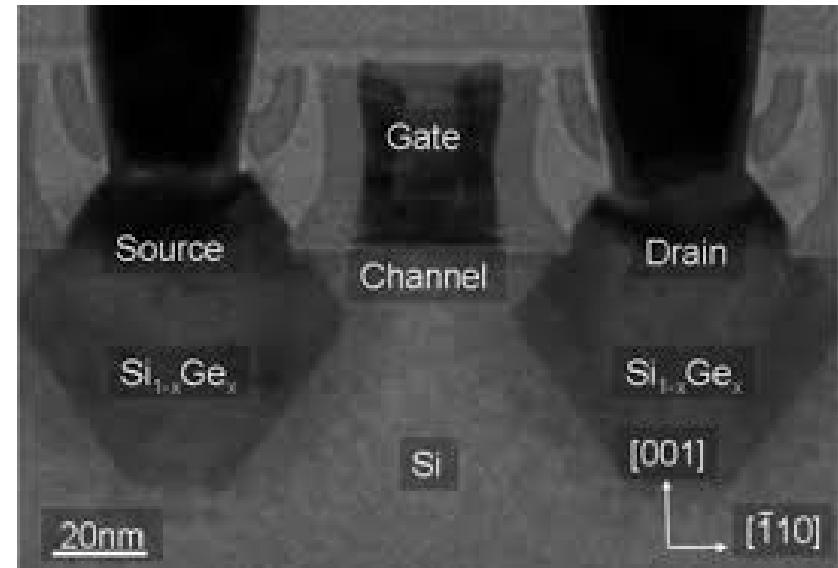


Porte Logiche

Tecnologie nMOS e pMOS



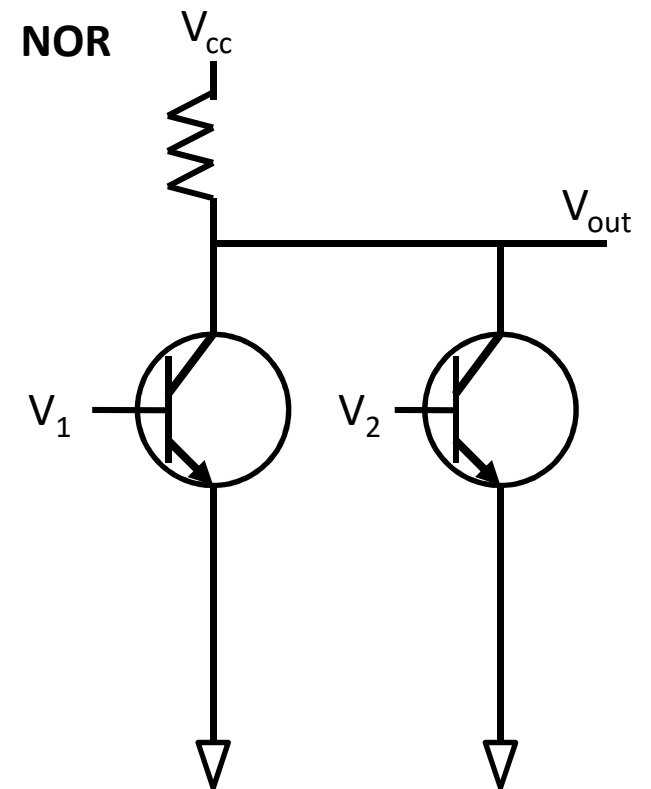
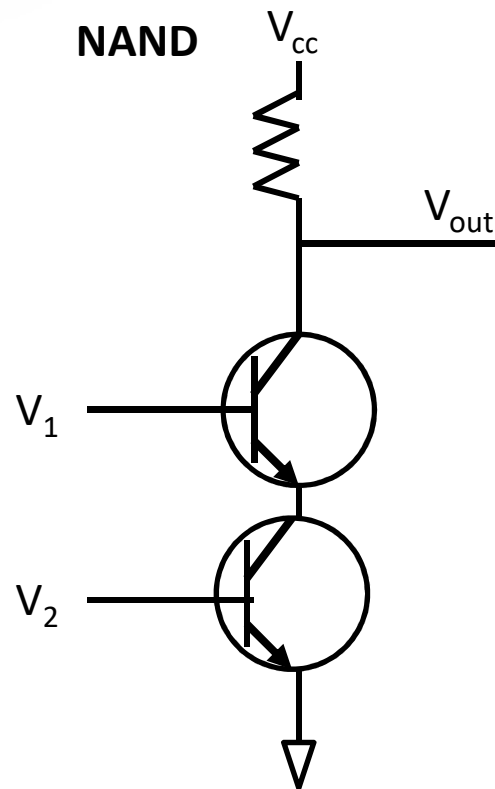
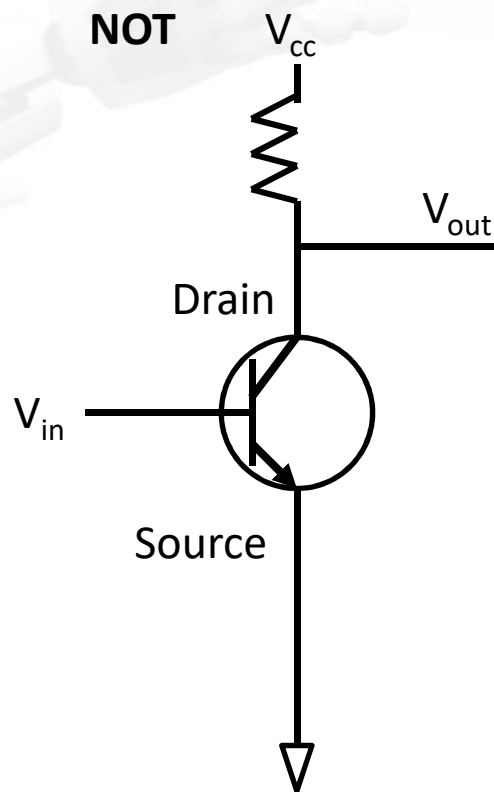
Chip



Cella nMOS

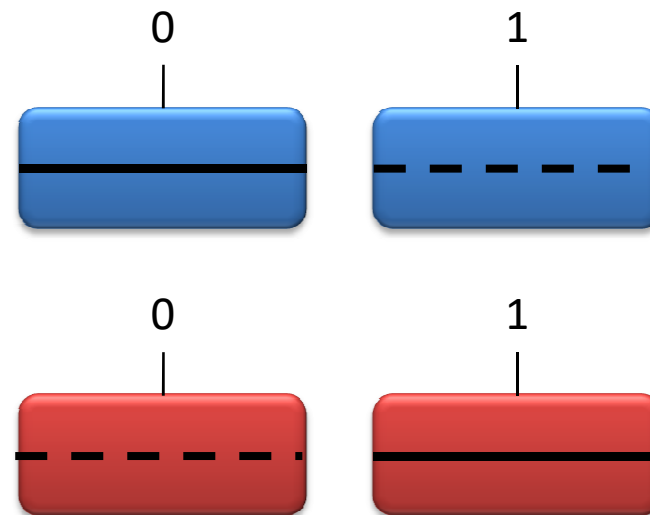
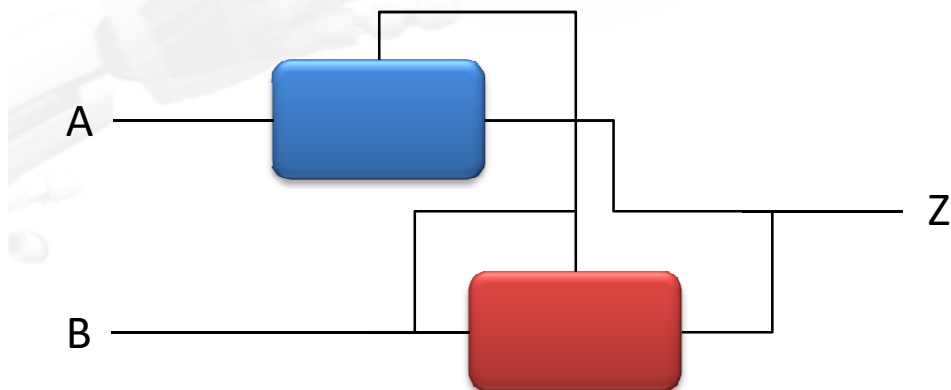
Porte Logiche

Porte logiche: schema



Porte Logiche

OR logico realizzato con transistori pMOS e nMOS

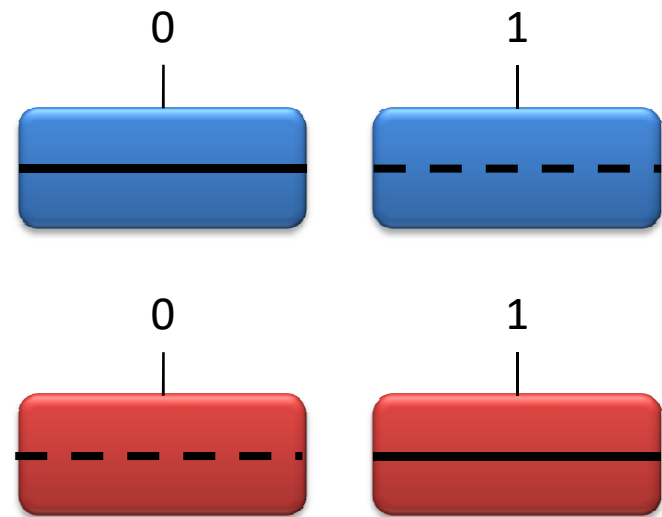
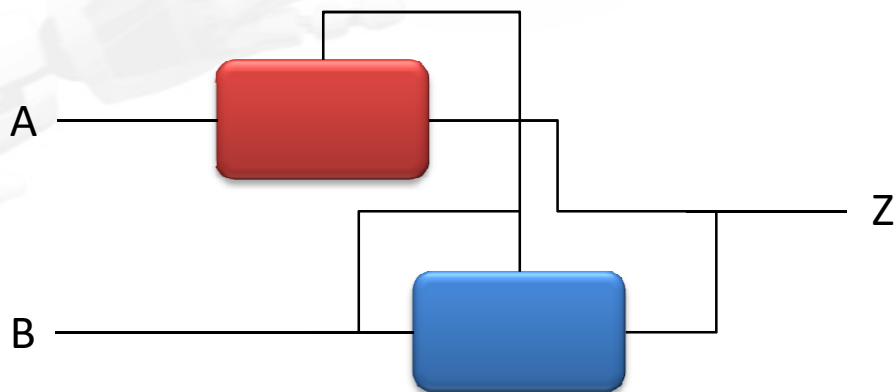


OR

Z	0	1	1	1
A	0	0	1	1
B	0	1	0	1

Porte Logiche

AND logico realizzato con transistori nMOS e pMOS



AND

Z	0	0	0	1
A	0	0	1	1
B	0	1	0	1

Porte Logiche

Considerazioni

“L'ENIAC è dotato di 18000 valvole e pesa 30 tonnellate, i calcolatori del futuro potranno avere 1000 valvole e forse peseranno solo una tonnellata e mezza”

Popular Mechanics, Marzo, 1949



	Nomenclatura	Num. Transitori
SSI	<i>Small Scale Integration</i>	meno di 10transistor
MSI	<i>Medium Scale Integration</i>	da 10 a 100transistor
LSI	<i>Large Scale Integration</i>	da 100 a 10000transistor
VLSI	<i>Very Large Scale Integration</i>	da 10000 a 100.000transistor
ULSI	<i>Ultra Large Scale Integration</i>	fino a 10.000.000 di transistor



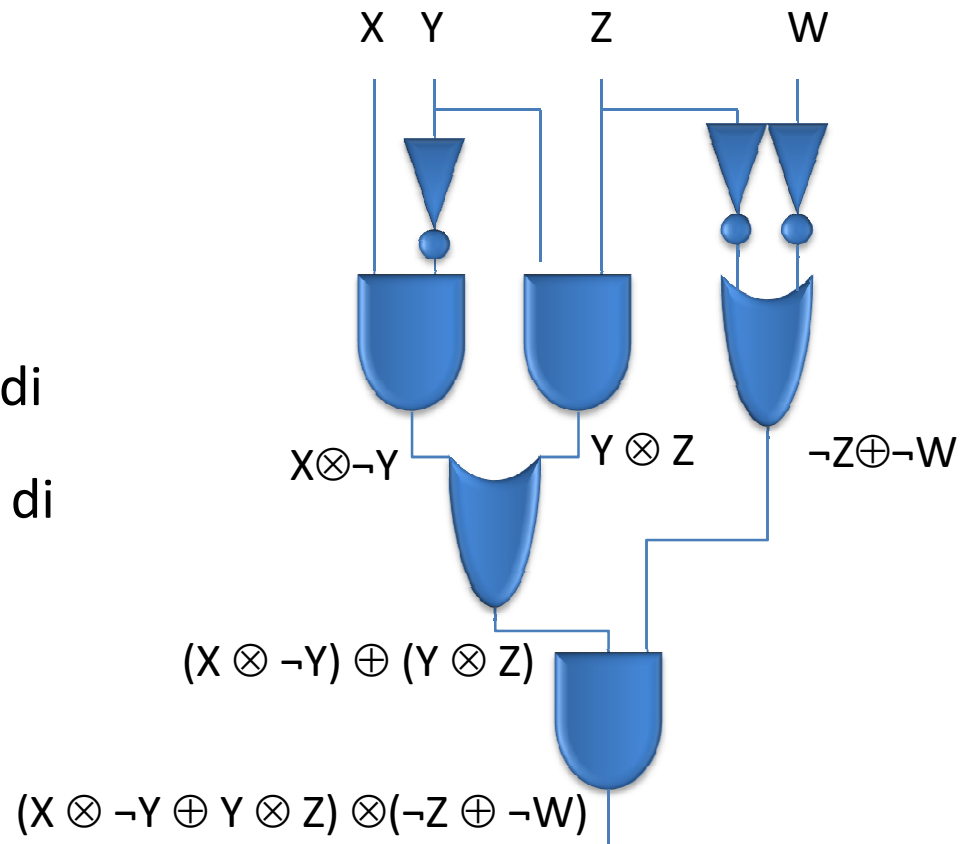
Architettura degli elaboratori

Rete combinatoria

RETE COMBINATORIA

Generalità

- ❑ Una **rete** è una interconnessione di componenti attivi, le **porte**, collegati tra loro mediante componenti passivi, **le linee**
- ❑ Una **rete combinatoria** è un circuito elettronico in grado di elaborare, in modo automatico, funzioni binarie di una o più variabili binarie





RETE COMBINATORIA

Definizione

- ❑ Il comportamento di una rete combinatoria è definito mediante un'espressione logica o una funzione booleana con codominio ad esponente anche maggiore di uno

$$Z = F(x) \quad \text{con} \quad F : \{0,1\}^n \rightarrow \{0,1\}^m$$

- ❑ Quindi una rete combinatoria è un modulo in grado di soddisfare una **tabella della verità**, che per ognuna delle 2^n combinazioni possibili relative agli n valori di ingresso (x_1, x_2, \dots, x_n) indica gli m valori di uscita (z_1, z_2, \dots, z_m) .



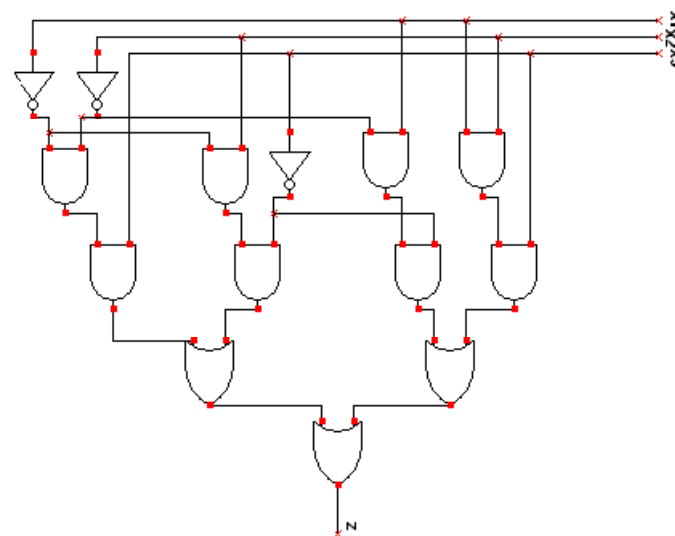
RETE COMBINATORIA

Esempio

Linee ingresso			Linee uscita
x_1	x_2	x_3	Z
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Espressione booleana in forma canonica:

$$Z = (\sim x_1 \otimes \sim x_2 \otimes x_3) \oplus (\sim x_1 \otimes x_2 \otimes \sim x_3) \oplus (x_1 \otimes \sim x_2 \otimes \sim x_3) \oplus (x_1 \otimes x_2 \otimes x_3)$$



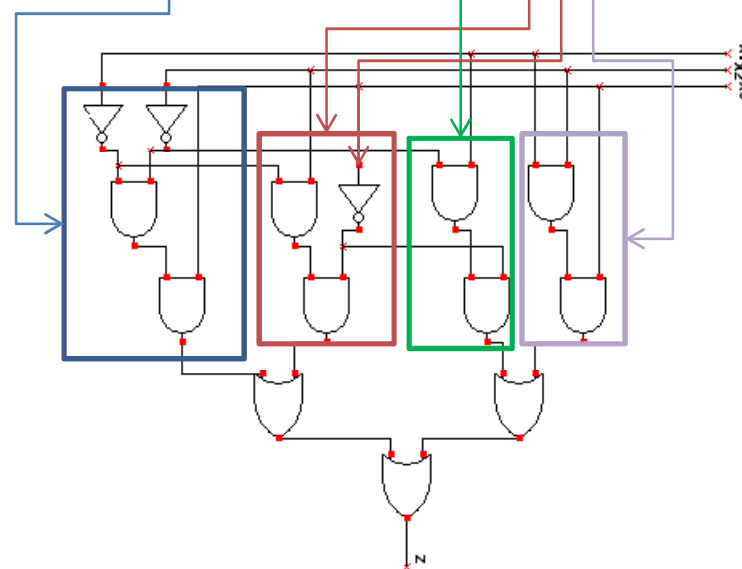
RETE COMBINATORIA

Esempio

Linee ingresso			Linee uscita
x_1	x_2	x_3	Z
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Espressione booleana in forma canonica:

$$Z = (\sim x_1 \otimes \sim x_2 \otimes x_3) \oplus (\sim x_1 \otimes x_2 \otimes \sim x_3) \oplus (x_1 \otimes \sim x_2 \otimes \sim x_3) \oplus (x_1 \otimes x_2 \otimes x_3)$$



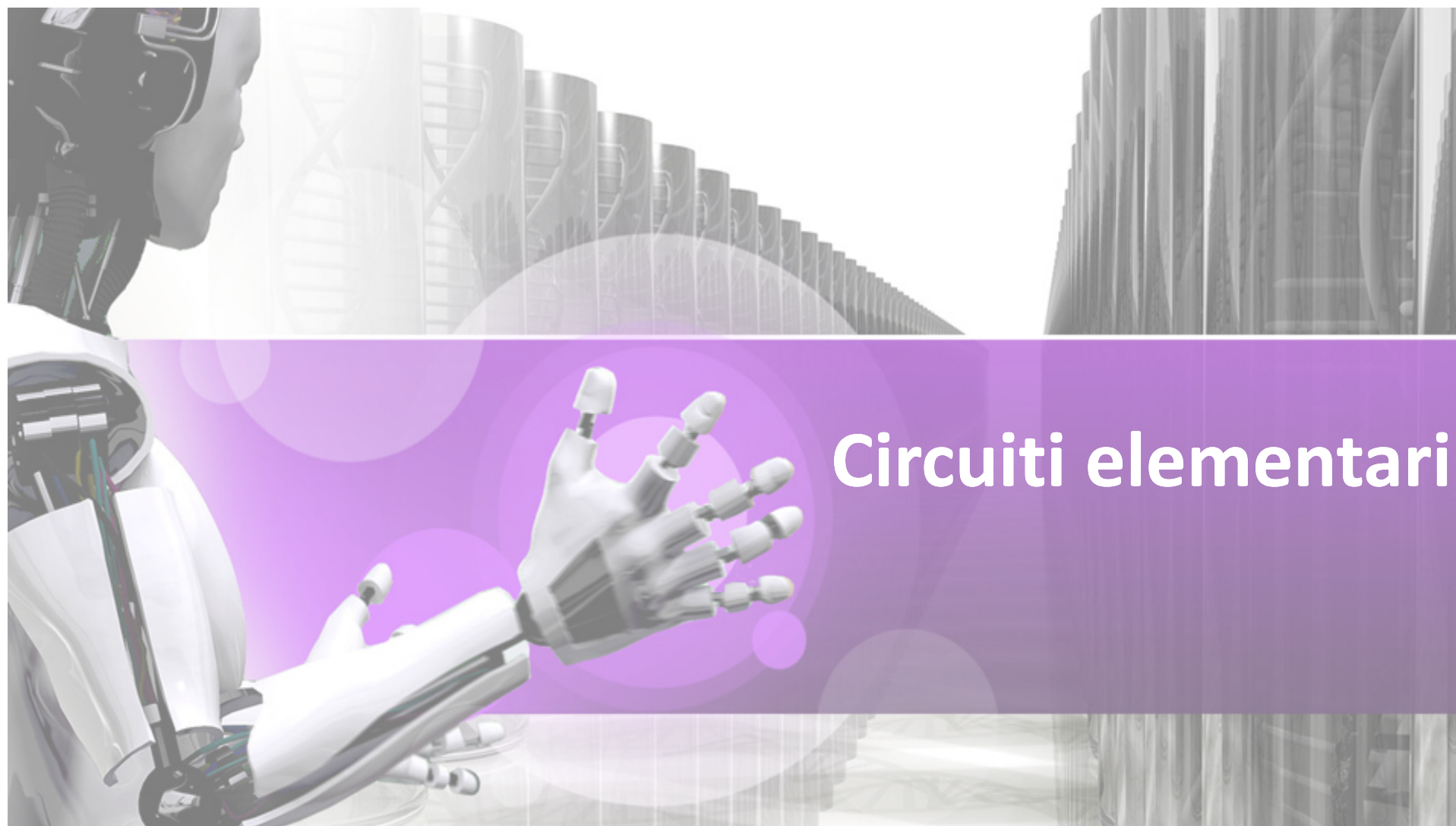


RETE COMBINATORIA

Caratteristiche progettuali

- ❑ Tra i parametri principali nel progetto di una rete combinatoria, è opportuno considerare:
 - ❑ l'**assorbimento di energia** (che stabilisce un limite complessivo al numero di porte utilizzabili)
 - ❑ il **ritardo** (che determina la velocità di calcolo)
 - ❑ La velocità di calcolo della rete combinatoria varia in base alla profondità della rete (di solito si considera un tempo costante perché la rete combinatoria ha una profondità finita)
 - ❑ il **costo** di realizzazione
 - ❑ Dipende dal numero di transistor impiegati che cambia a seconda della tecnologia usata, della funzione da soddisfare e il numero di ingressi
es.: la porta NOT è costituita da 1 transistor, NAND o NOR 2 transistor; AND e OR 3 o 4 transistor; altre porte: > 4 transistor

PORTA	TEMPO DI COMMUTAZIONE DEL SEGNALE
NAND, NOR	ΔT
AND, OR	$2\Delta T$
XOR	$4\Delta T$



Circuiti elementari

RETE COMBINATORIA

Decodificatore

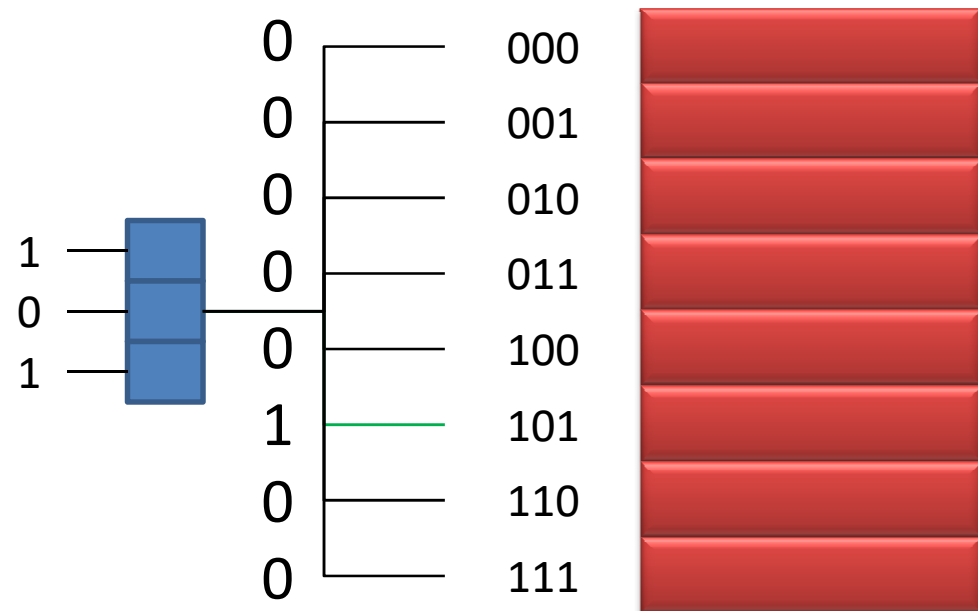
- ❑ Il **decodificatore** è una rete combinatoria che trasforma parole associate a **codifiche strette** in parole associate a **codifiche lasche** (le linee di uscita sono in numero maggiore rispetto le linee di ingresso)
- ❑ Un decodificatore è una rete combinatoria con m linee di ingresso e $n=2^m$ linee di uscita
- ❑ Logicamente il decodificatore **riconosce una stringa** (es.: una locazione di memoria o una istruzione)

DECODIFICATORE										
Linee ingresso			Linee uscita							
x_2	x_1	x_0	z_7	z_6	z_5	z_4	z_3	z_2	z_1	z_0
0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	1	0	0
0	1	1	0	0	0	0	1	0	0	0
1	0	0	0	0	0	1	0	0	0	0
1	0	1	0	0	1	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0

RETE COMBINATORIA

Decodificatore: esempio di impiego

- ❑ Il decodificatore è usato, ad esempio, per identificare una cella di memoria



Codificatore

- ❑ I **codificatori** sono una famiglia di reti combinatorie che trasformano parole codificate in una codifica lasca in parole d'uscita rappresentate in codifica stretta
- ❑ In generale un codificatore è una rete combinatoria che ha n linee di ingresso e $m = \lceil \log_2(n) \rceil$ linee di uscita; cioè vi è la produzione della codifica binaria dell'indice dell'unica linea di ingresso attiva
- ❑ Logicamente il codificatore è un **generatore di codici** (es.: dei comandi)

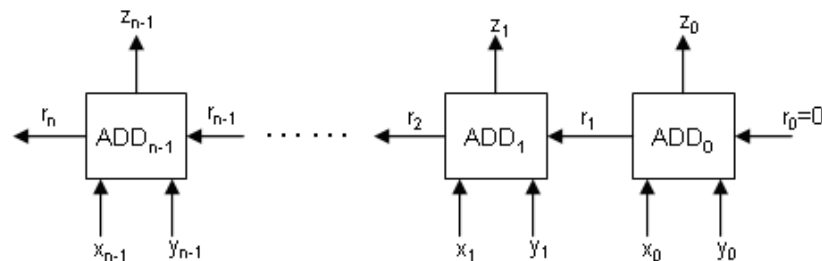
[illegible]

RETE COMBINATORIA

Addizionatore

- ❑ L'**addizionatore** è una rete combinatoria che consente l'operazione di somma tra due operandi (addendi)
- ❑ La rete combinatoria associata ad un addizionatore può essere realizzata mediante una tecnica di decomposizione. Questo perché l'addizione di due numeri binari può essere vista come la somma di due bit alla i -esima posizione (x_i e y_i) ai quali va aggiunto l'**eccedenza** (r_i) per ottenere un **risultato** (z_i) ed un eventuale **riporto** (r_{i+1}) per le **cifre successive**

ADDIZIONATORE				
x_i	y_i	r_i	z_i	r_{i+1}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

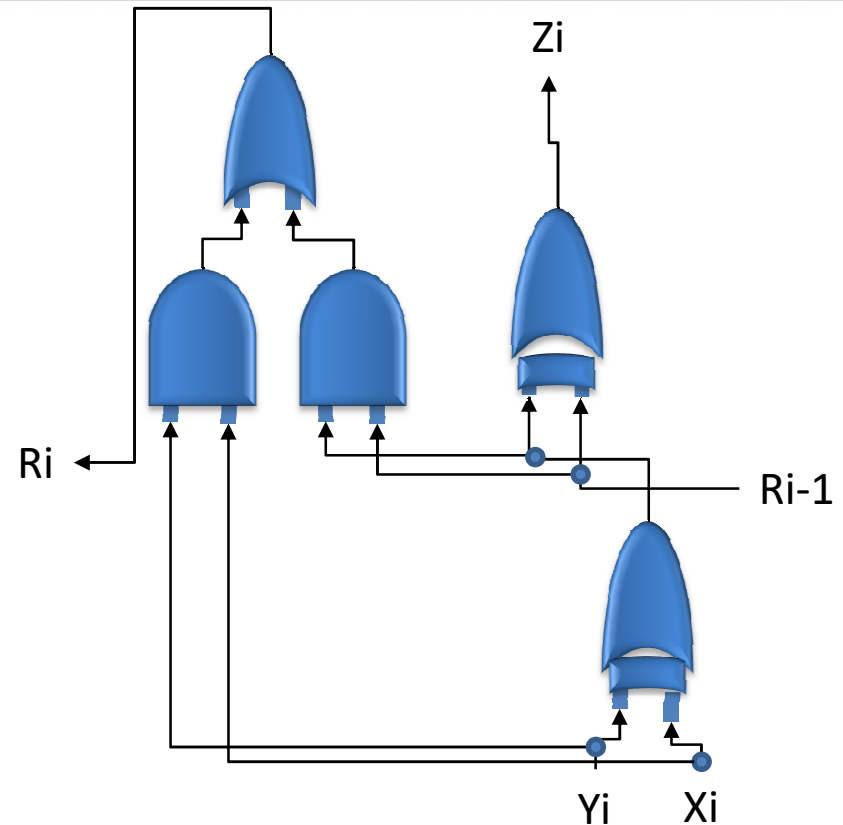
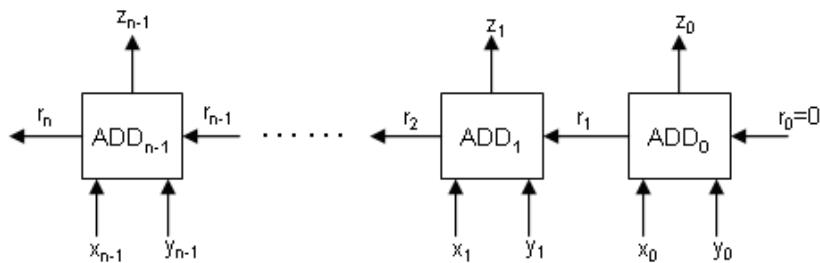


RETE COMBINATORIA

Addizionatore

ADDIZIONATORE

x_i	y_i	r_i	z_i	r_{i+1}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

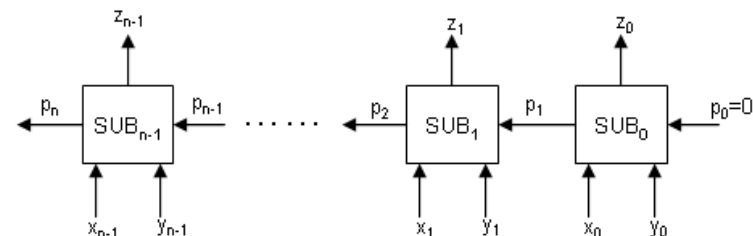


RETE COMBINATORIA

Sottrattore

- ❑ Il **sottrattore** è una rete combinatoria che permette la sottrazione tra due operandi (minuendo e sottraendo)
- ❑ Anche in questo caso è possibile fare riferimento ad una struttura modulare.
All' i -esimo bit del minuendo (x_i) va sottratto sia il sottraendo (y_i) sia il bit di prestito (p_i) della posizione precedente per poi generare il bit risultante (z_i) ed il bit del prestito (p_{i+1}) per le cifre successive

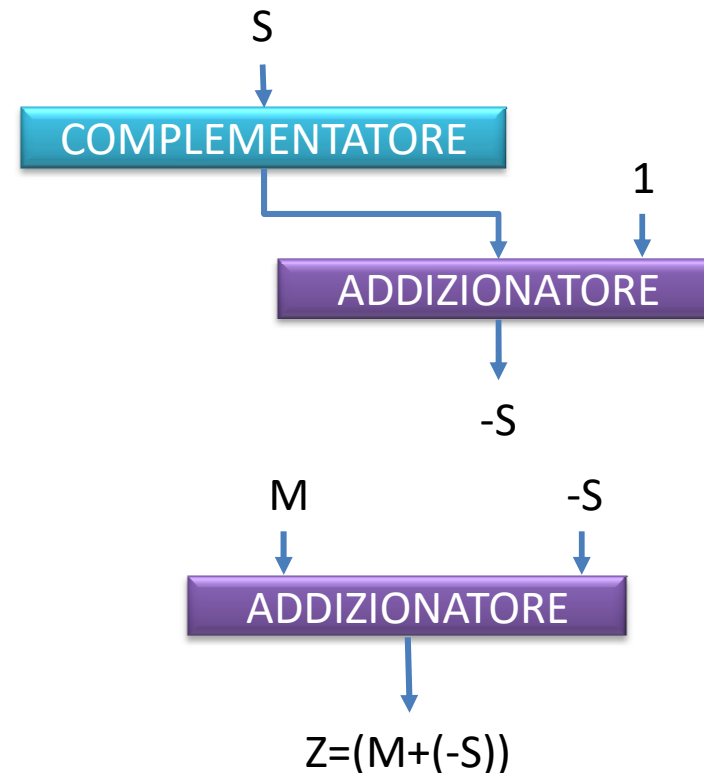
SOTTRATTORE				
x_i	y_i	p_i	z_i	p_{i+1}
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1



RETE COMBINATORIA

Sottrattore: realizzazione pratica

- ❑ Nel concreto la realizzazione di un sottrattore può essere effettuato con una circuiteria differente da quella vista
- ❑ L'operazione di sottrazione $z=m-s$ si riduce alla espressione equivalente $z=(m+(-s))$
- ❑ Si utilizza un **complementatore** ed un **addizionatore** che prende in input come addendi il numero complementato ed il valore 1
- ❑ L'operazione di sottrazione si ottiene aggiungendo il minuendo

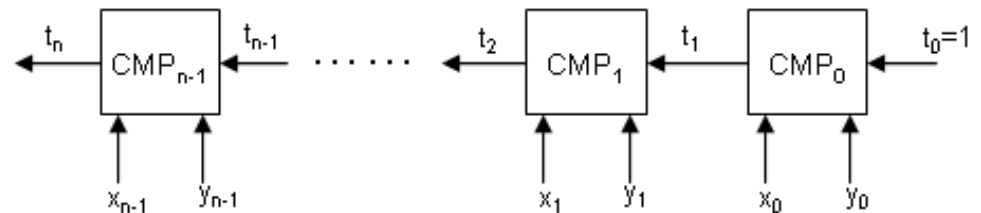


RETE COMBINATORIA

Comparatore

- ❑ Il **comparatore** è una rete combinatoria che ha una unica linea di uscita che vale 1 se il numero x , di n bit, applicato in ingresso risulta maggiore o uguale (in senso algebrico) al numero y , di n bit, anch'esso preso in ingresso, con cui si effettua il confronto
- ❑ Anche per tale componente è possibile fare riferimento ad una struttura modulare

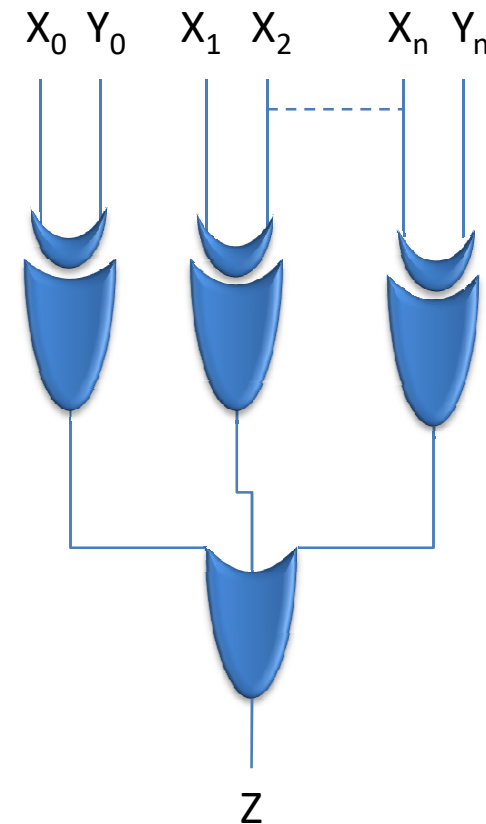
COMPARATORE			
x_i	y_i	t_i	t_{i+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1



RETE COMBINATORIA

Comparatore logico

- ❑ Altresì, invece, è possibile realizzare un **comparatore logico** utilizzando n porte XOR, a cui in ingresso sono associati gli i -esimi bit dei valori da comparare, le cui uscite sono collegate ad una porta OR
- ❑ In questo caso, infatti, è necessario stabilire solamente se le stringhe binarie sono uguali o diverse
- ❑ Un comparatore logico determina il risultato finale in tempo costante $O(1)$





Architettura degli elaboratori

*Cella di memoria
(temporanea)*



CELLA DI MEMORIA

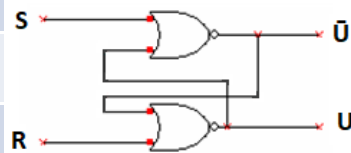
- ❑ Il componente in grado di memorizzare l'informazione è il **circuito bistabile** (*flip flop* o *latch*): una cella di memoria capace di contenere un singolo bit per un tempo determinato
- ❑ L'uso di più latch comporta la realizzazione del **registro**, che rappresenta una componente fondamentale all'interno dei calcolatori elettronici

Osservazione. La differenza tra un flip flop ed un latch è che il latch riproduce il valore del segnale d'ingresso per tutto il tempo in cui il segnale di clock assume il valore alto

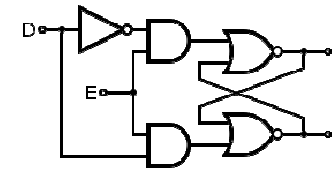
CELLA DI MEMORIA

Tipologie: RS, D, JK, T

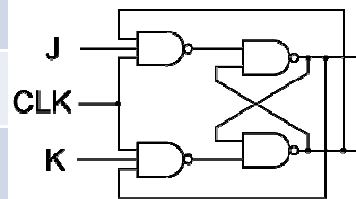
S	R	U	\bar{U}
0	0	Nessun cambiamento	
0	1	0	1
1	0	1	0
1	1	Non definito	



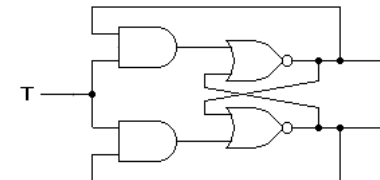
D	U	\bar{U}
0	0	1
1	1	0



J	K	U	\bar{U}
0	0	Nessun cambiamento	
0	1	0	1
1	0	1	0
1	1	Complemento del valore	



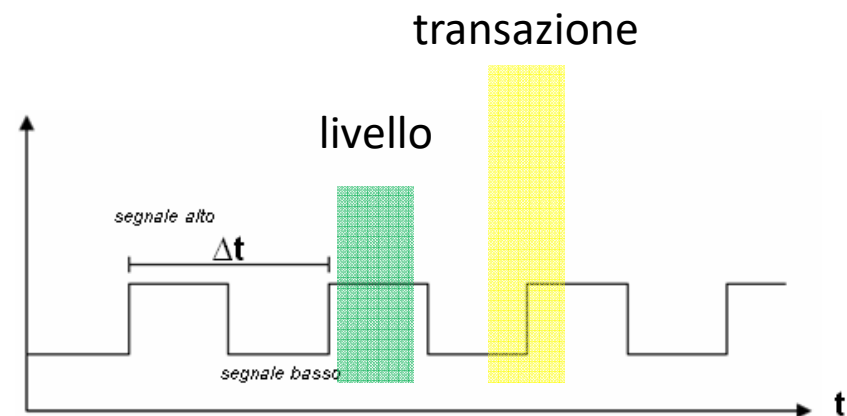
T	U	
0	U	Nessun cambiamento
1	\bar{U}	Complemento



CELLA DI MEMORIA

Commutazione dei segnali

- ❑ Nei flip flop la transizione di stato si verifica durante la transizione del clock da 0 a 1 (**fronte di salita**) oppure da 1 a 0 (**fronte di discesa**)
- ❑ Nei latch il segnale rimane in uscita per tutto l'intervallo del segnale alto





Architettura degli elaboratori

Rete sequenziale



RETE SEQUENZIALE

Generalità

- Una rete sequenziale sincrona realizza fisicamente un **automa a stati finiti** (*Finite State Automata*) che può essere descritto formalmente con una quintupla:

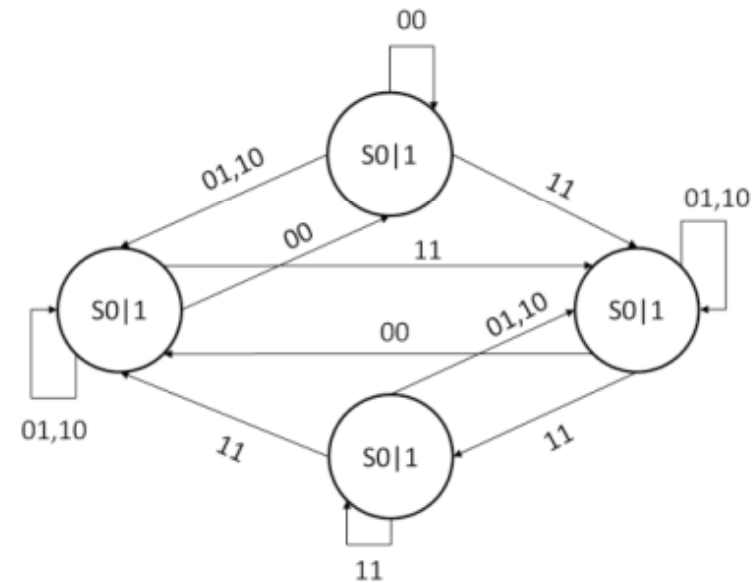
$$M = \langle S, I, O, \delta, \lambda \rangle$$

S è l'insieme degli stati possibili;
 I è l'insieme dei valori in ingresso;
 O è l'insieme dei valori in uscita;
 δ è la funzione che lega gli ingressi e lo stato corrente con lo stato successivo;
 λ è la funzione che mette in relazione lo stato e l'ingresso corrente alle uscite.

RETE SEQUENZIALE

Dall'automa di moore alla rete sequenziale: Esempio (Automa Moore)

- Un automa che riconosce la sottosequenza 010 in una successione di ingresso di sequenze di 0 ed 1 può essere descritto mediante un automa di Moore



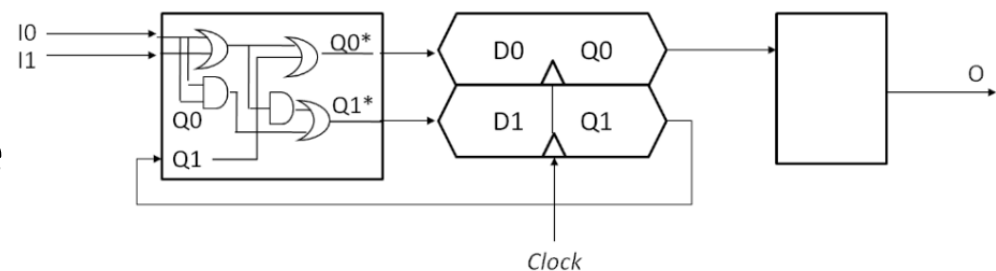
RETE SEQUENZIALE

Dall'automa di Moore alla rete sequenziale:
Esempio (tabella do transizione e circuito sequenziale)

- ❑ Da cui si deriva la tabella di transizione degli stati (vedi al lato)

Tabella di transizione degli stati (MOORE)			
Stato	δ		Λ
	0	1	0
S0	S0	S2	0
S1	S0	S1	0
S2	S3	S1	0
S3	S0	S2	1

- ❑ Dopo una fase di minimizzazione si deriva il relativo circuito sequenziale





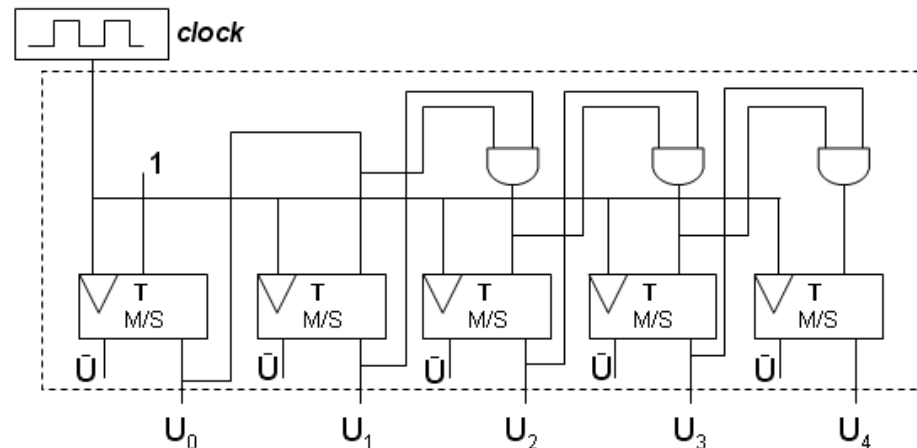
Architettura degli elaboratori

Principali reti sequenziali

RETE SEQUENZIALE

Registro Contatore

- ❑ Il **registro contatore** è un componente in grado di incrementare di 1 la rappresentazione binaria del numero in esso memorizzato ad ogni colpo di clock
- ❑ Tale funzionalità può essere ottenuta mediante l'uso di un latch T di tipo Master Slave
- ❑ La presenza della connessione tra uscite dello Slave e ingressi del Master fa sì che il flip flop T-M/S sia in grado di effettuare una commutazione del valore quando l'ingresso T è posto al valore 1 ed il segnale di clock viene variato. L'uscita rimane invece invariata se l'ingresso T assume il valore 0 durante il periodo di tempo in cui varia il valore del segnale di clock

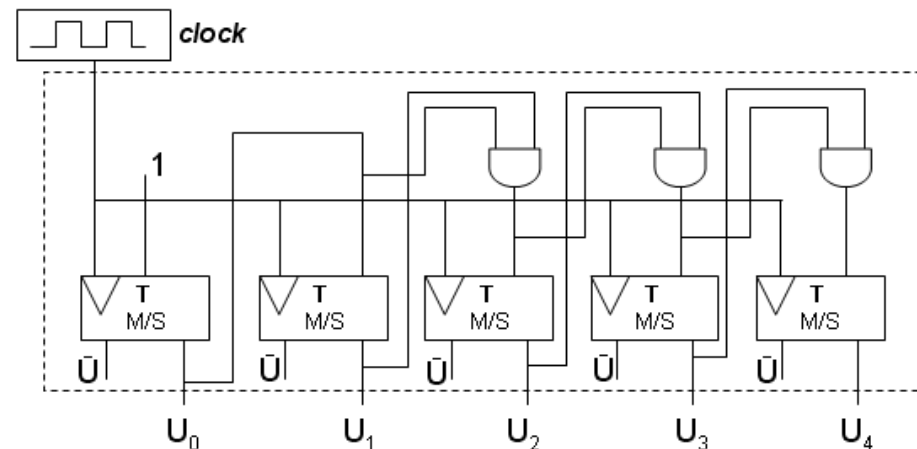


RETE SEQUENZIALE

Registro Contatore: funzionamento

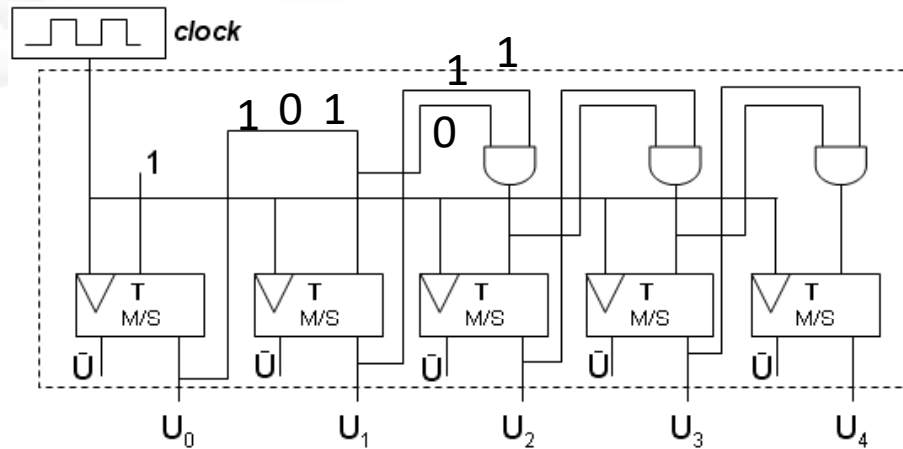
- ❑ Ad ogni fronte di salita (passaggio dal segnale basso al segnale alto), applicato all'ingresso del temporizzatore, corrisponde una commutazione del primo latch dovuta al fatto che il suo ingresso è connesso al valore costante 1
- ❑ Altresì, la condizione di commutazione per la i -esima cifra corrisponde al riporto non nullo nel caso di somma della costante 1 col valore precedentemente memorizzato

T	U	
0	U	Nessun cambiamento
1	\bar{U}	Complemento



RETE SEQUENZIALE

Registro Contatore

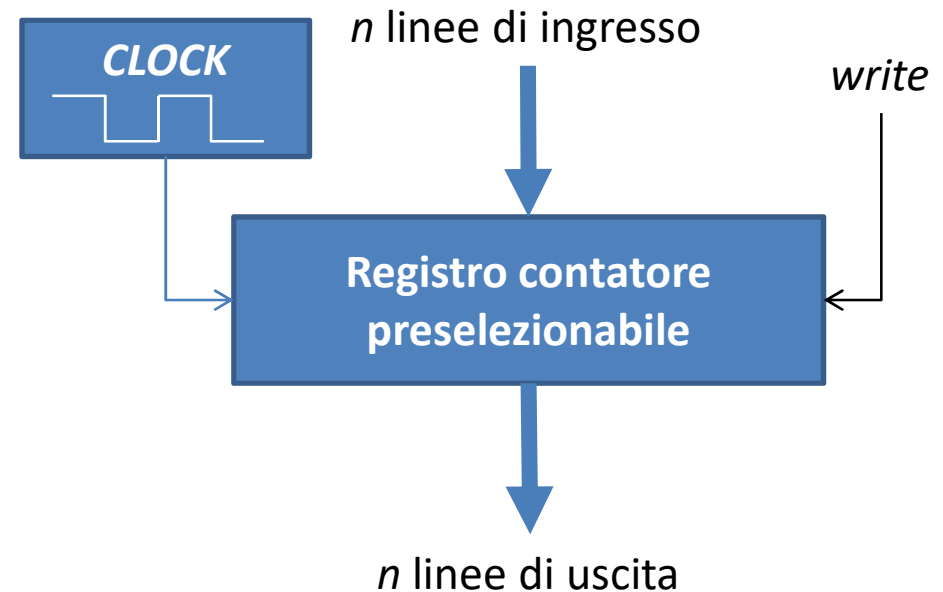


Inizio	0	0	0	0	0
Fronte di salita 0	1				
Fronte di discesa 0	0	1			
Fronte di salita 1	1	1			
Fronte di discesa 1	0	0	1		

RETE SEQUENZIALE

Registro Contatore preselezionabile

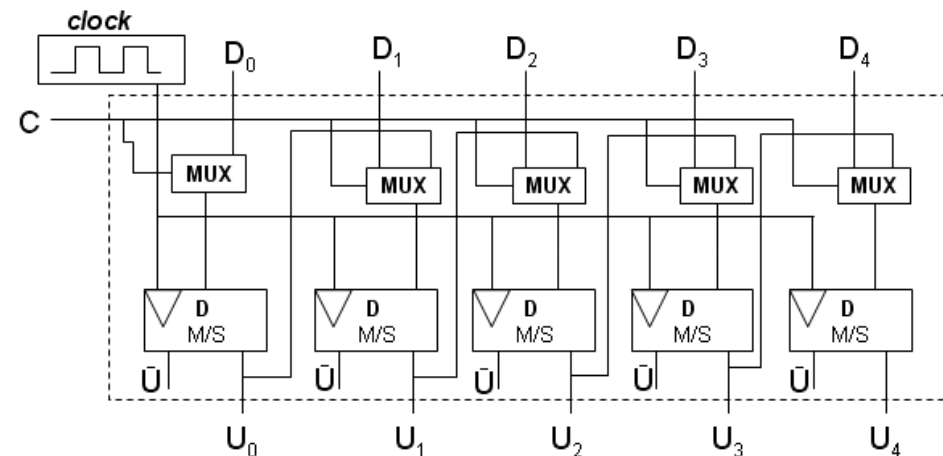
- ❑ Il registro contatore può essere anche realizzato in maniera tale che sia **preselezionabile** ovvero si offre la possibilità di “forzare” il suo contenuto con un valore differente da quello contenuto



RETE SEQUENZIALE

Registro a scorrimento

- ❑ Un **registro a scorrimento** (o *shift register*) è realizzato interponendo un multiplexer sugli ingressi di un latch di tipo D-M/S
- ❑ A seconda del valore della variabile di controllo C, ad ogni ciclo di clock il registro effettua uno scorrimento (shift) di una posizione dei bit della rappresentazione,
 - ❑ con $C=0$ c'è l'inserzione della cifra 0 nel bit meno significativo (lasciato libero dallo scorrimento). **Shift aritmetico**
 - ❑ con $C=1$ c'è la memorizzazione di un nuovo valore presente sugli ingressi D. **Shift logico**





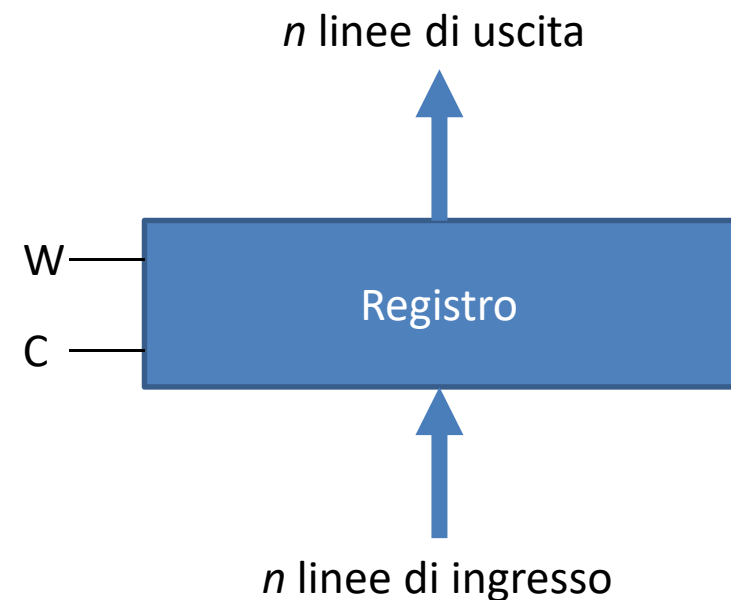
Architettura degli elaboratori

Interconnessione

RETE SEQUENZIALE

Registro di memorizzazione

- ❑ Quando si progetta un calcolatore bisogna stabilire a priori la dimensione dei registri (contatore, shift, locazioni dove conservare gli operandi)
- ❑ Le singole informazioni costituenti le parole, cioè le cifre binarie, sono memorizzate in latch e l'insieme risultante è un componente denominato **registro**
- ❑ Il modo più semplice per realizzare un registro è quello di utilizzare n celle di memoria ed almeno due linee: una (W) per selezionare simultaneamente le n celle elementari e l'altra (C) di azzeramento del registro ("pulendo" il contenuto con uno 0 per ogni cella)





INTERCONNESSIONE

Generalità

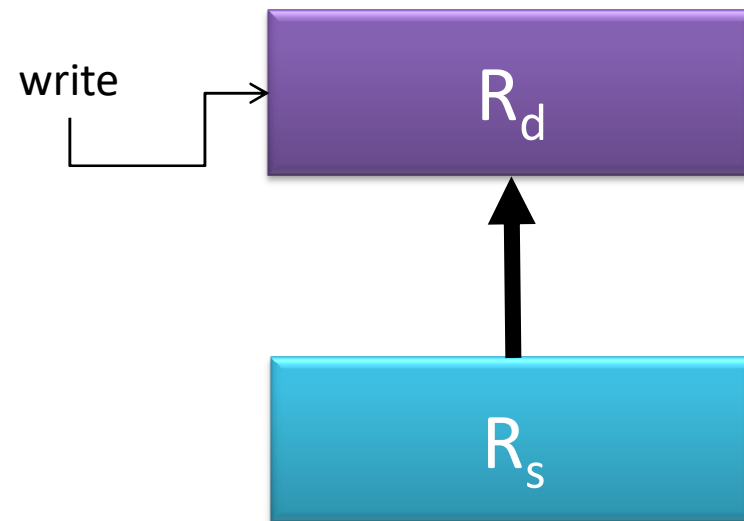
- ❑ Il termine interconnessione fa riferimento a reti che siano in grado di trasferire, o meglio **duplicare**, l'informazione contenuta nei registri
- ❑ Per raggiungere questo obiettivo è possibile prendere in considerazione diverse tecniche

	Sorgente prefissata	Sorgente variabile
Destinazione prefissata	Punto a punto	Multiplexer
Destinazione variabile	Demultiplexer	Mesh, bus,...

INTERCONNESSIONE

Punto a Punto

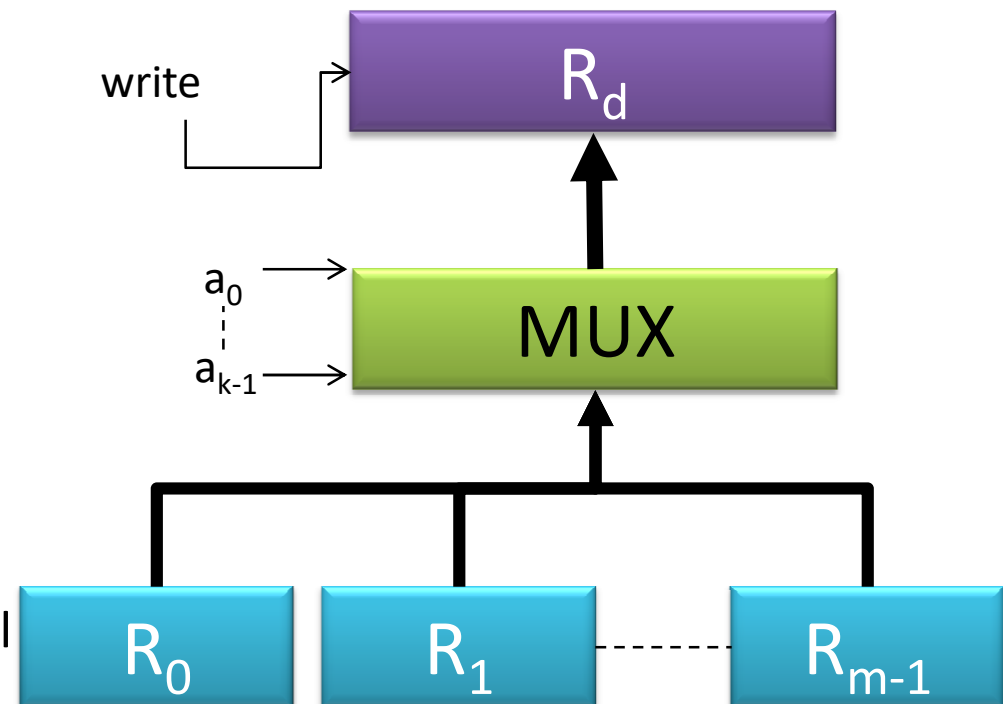
- ❑ L'interconnessione punto a punto è prevista per effettuare il trasferimento della parola contenuta in un registro sorgente R_s ad una destinazione R_d
- ❑ In questo caso tutte le n uscite di R_s sono legate alle n entrate di R_d ovviamente predisponendo una linea (*transfer* o *write*) di controllo che indica, con il valore 1, il trasferimento di informazione e con 0 la conservazione del valore corrente



INTERCONNESSIONE

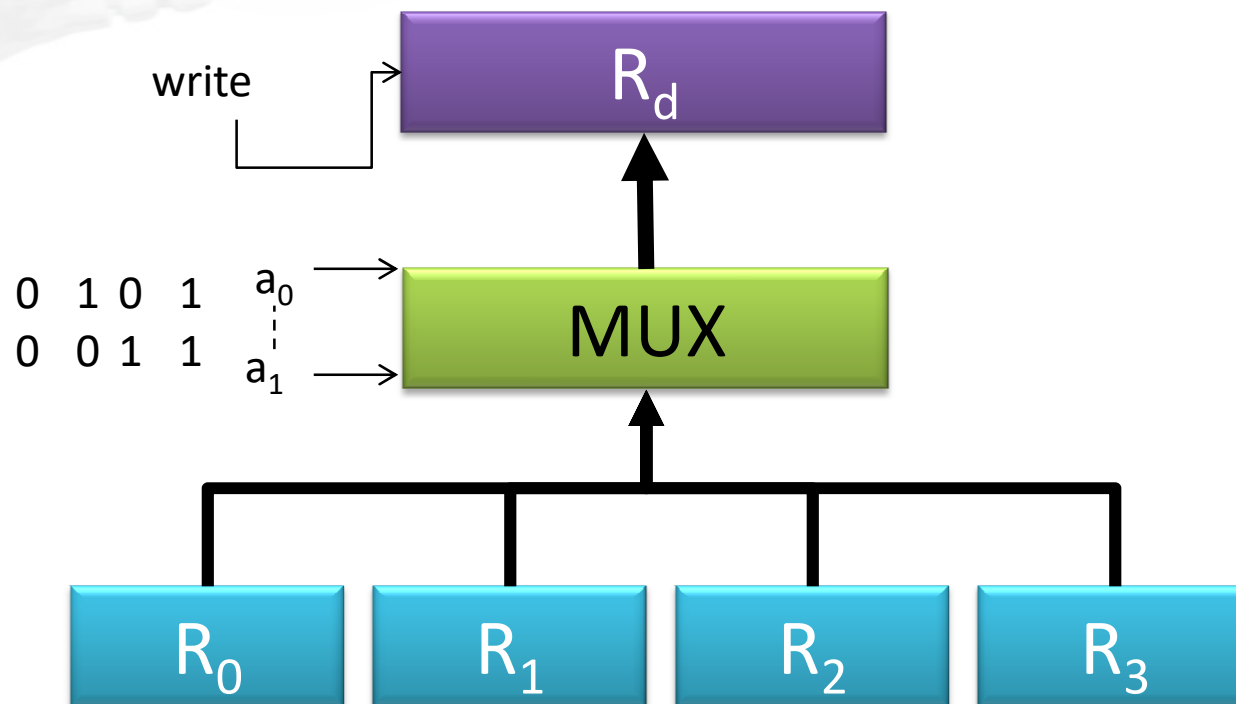
Multiplexer

- ❑ Il **multiplexer** è la rete di interconnessione atta a favorire il trasferimento tra m registri sorgenti ed un registro destinazione prefissato
- ❑ Il multiplexer ha due tipi di linee di ingresso:
 - ❑ m fasci (ciascuno di n linee) di ingresso per i registri
 - ❑ $k = \lceil \log_2(m) \rceil$ linee di ingresso (**indirizzi o address**) per codificare gli m possibili registri
- ❑ Le n linee di uscita, invece, sono usate come linee di ingresso per il registro destinazione



INTERCONNESSIONE

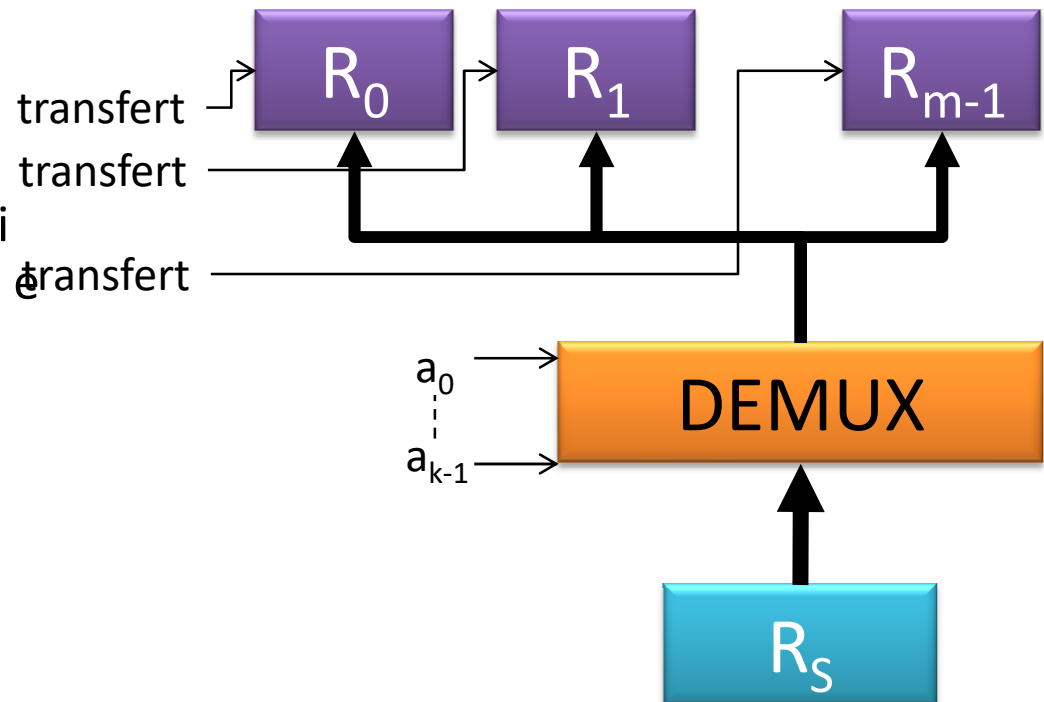
Multiplexer



INTERCONNESSIONE

Demultiplexer

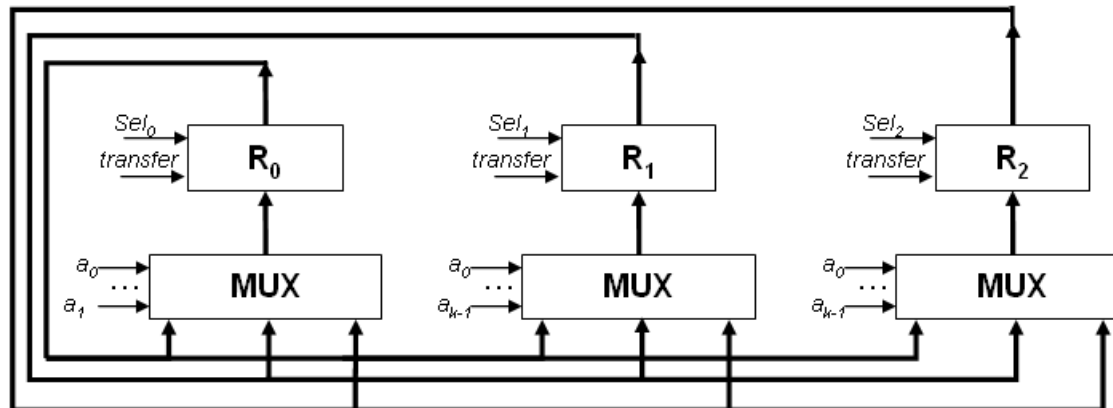
- ❑ Il **demultiplexer** è la rete di interconnessione atta a favorire il trasferimento tra uno specifico registro sorgente ed m registri destinatari
- ❑ Il demultiplexer ha come linea di ingresso la parola di lunghezza n e $k = \lceil \log_2(m) \rceil$ linee (**indirizzi**) per decodificare gli m possibili registri; mentre come linee di uscita ha m fasci (ciascuno di n linee) per raggiungere i registri destinazione



INTERCONNESSIONE

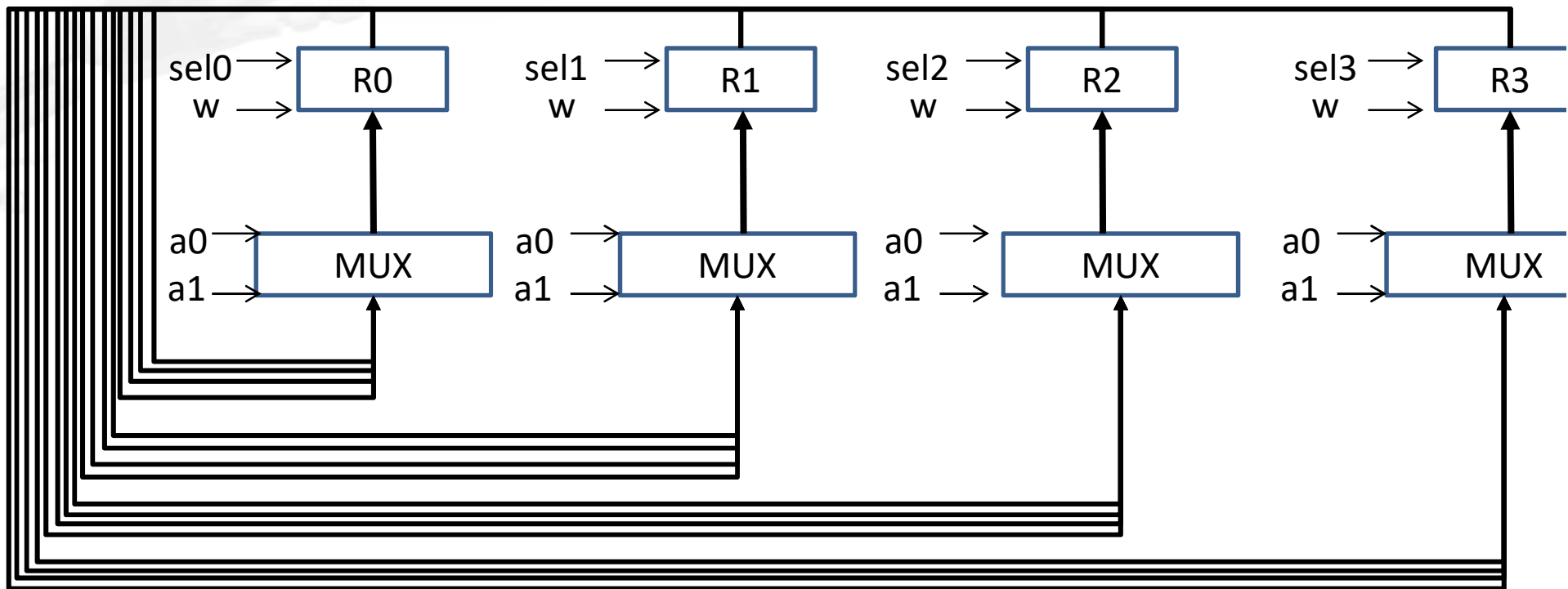
Mesh

- ❑ Le **reti mesh** sono reti in grado di interconnettere tra loro m registri, o più in generale m componenti
 - ❑ Evita una connessione punto a punto che necessita di m^2 collegamenti
- ❑ In generale si usa per ogni registro un multiplexer collegato agli m registri sorgenti
- ❑ Sebbene questa architettura consenta trasferimenti simultanei, per coppie di registri disgiunti, è una soluzione da scongiurare al crescere di m e non molto pratica (l'aggiunta di altri componenti può richiedere una nuova progettazione)



INTERCONNESSIONE

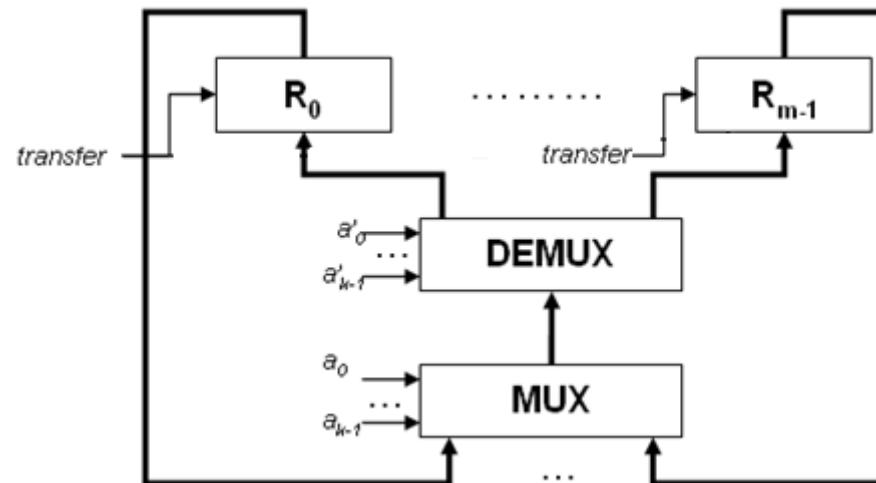
Mesh



INTERCONNESSIONE

Mesh: variante

- ❑ Qualora il numero dei registri sia elevato si può ridurre il numero di componenti di interconnessione sfruttando un multiplexer ed un demultiplexer collegati in serie
- ❑ Impostando le linee di indirizzo del multiplexer (a_0, \dots, a_{k-1}), il registro selezionato è trasferito all'uscita del multiplexer. In seguito determinando gli indirizzi del demultiplexer (a'_0, \dots, a'_{k-1}) il contenuto del registro sorgente è trascritto in quello del registro destinazione.
- ❑ Gli indirizzi del multiplexer e demultiplexer possono essere comunemente attivati nello stesso ciclo di macchina. Inoltre c'è un risparmio sui componenti, ma è impossibile effettuare trasferimenti simultanei



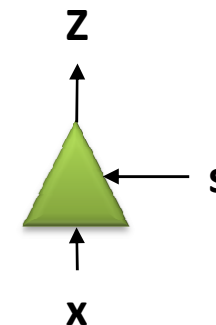
INTERCONNESSIONE

Bus: buffer a 3 stati

- ❑ Un modo economico per consentire una **interconnessione a sorgente e destinazione variabile** si può ottenere utilizzando un particolare componente il **buffer tri-state**
- ❑ Il buffer tri-state ha due linee di ingresso x e s ed una di uscita Z
- ❑ Questo componente presenta tre stati individuati da:
 - 1) nel caso in cui $s=0$ il circuito è tagliato
 - 2) con $s=1$ se $x=0$ allora $Z=0$
 - 3) con $s=1$ se $x=1$ allora $Z=1$

BUS a 3 stati

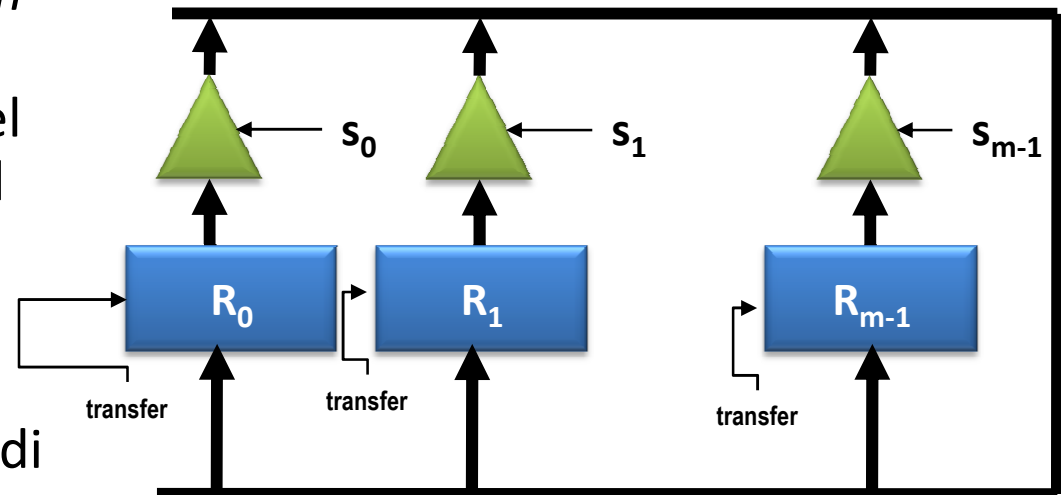
x	s	Z
0	0	\
0	1	0
1	0	\
1	1	1



INTERCONNESSIONE

Bus

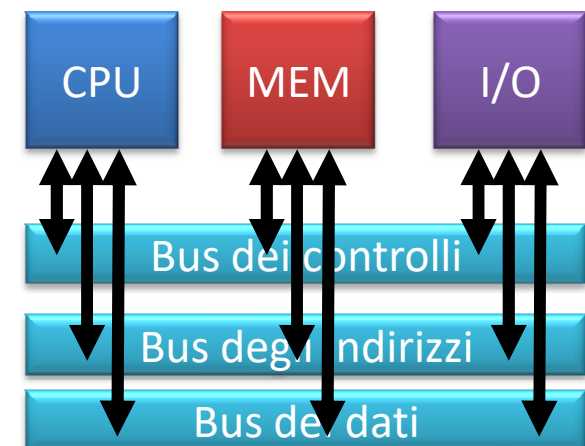
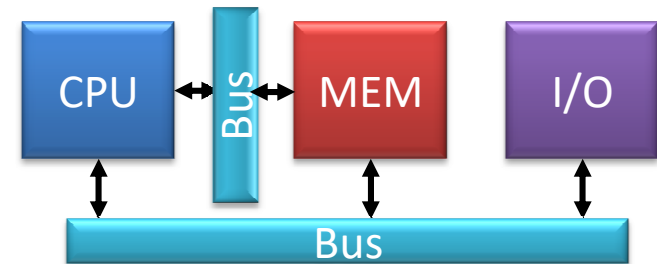
- ❑ Il buffer tri-state permette una interconnessione con m registri realizzata con un gruppo di n (dimensione del registro) linee che prende il nome di **bus**
- ❑ In questo caso per il trasferimento è sufficiente attivare la linea di ingresso di **selezione** (s) del **registro sorgente** e del **registro destinazione**

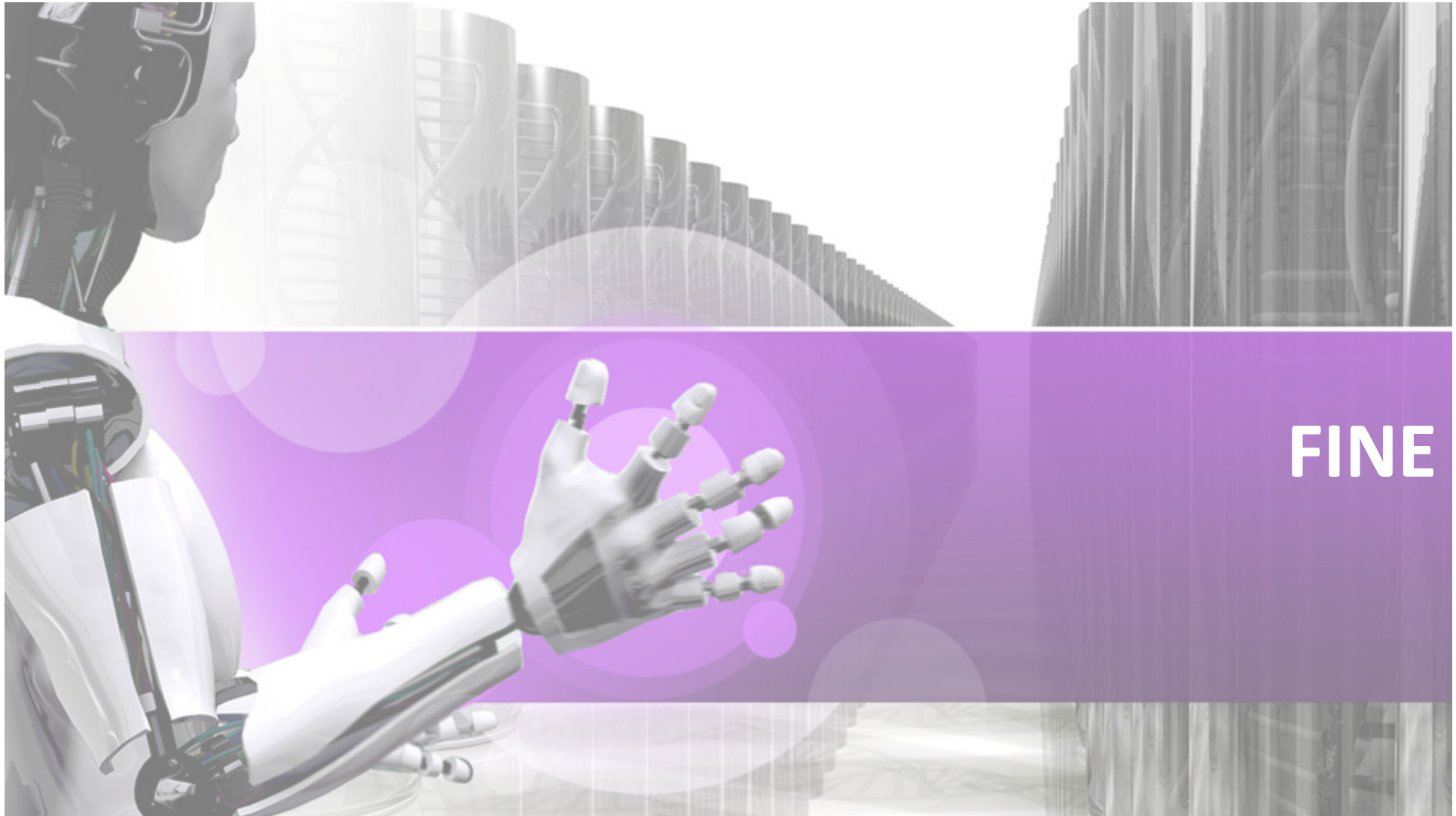


INTERCONNESSIONE

Bus

- ❑ I bus, però, pur essendo molto economici non permettono di effettuare trasferimenti simultanei durante lo stesso ciclo macchina
- ❑ Per risolvere tale problema, a volte, si utilizzano bus che collegano **gruppi di componenti** o registri logicamente indipendenti tra loro (in modo da consentire trasferimenti simultanei su ciascuno dei bus disponibili)





FINE