Lezione 5

Tecniche di semplificazione

Circuiti digitali notevoli

F. Pedersini

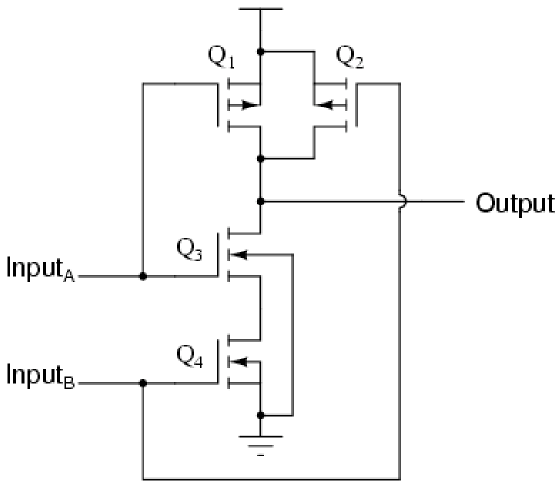
Dipartimento di Scienze dell'Informazione
Università degli Studi di Milano

- Tempi di commutazione
- Semplificazioni algebriche
- Semplificazioni con Mappe di Karnaugh
- Circuiti combinatori notevoli

Semplicità e prestazioni di un circuito

❖ Criteri di valutazione di semplicità e prestazioni:

- Semplicità (area)
  - numero di porte in totale
- Velocità (tempo di commutazione)
  - numero di porte attraversate
- Soddisfazione di altri vincoli
  - potenza dissipata,
  - facilità di debug...

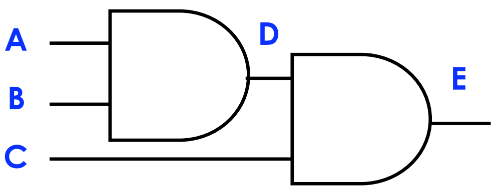


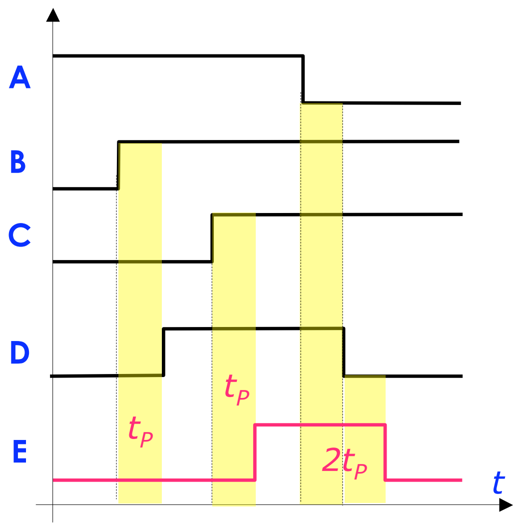
Cammino critico

❖ Ogni circuito logico è caratterizzato da un tempo di commutazione

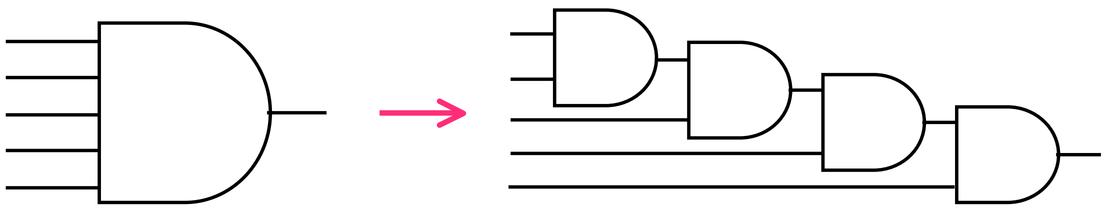
❖ CAMMINO CRITICO: <u>massimo</u> numero di porte da attraversare da ingresso a uscita

- Non si contano gli inverters (inclusi nelle porte)



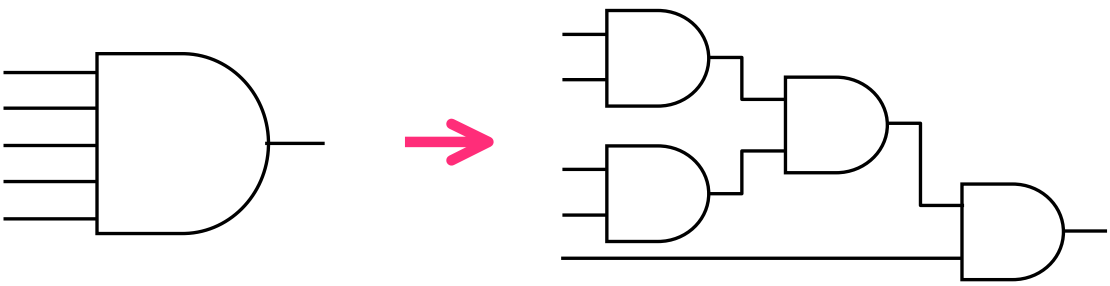


- Gli elementi costruttivi di base tipici sono porte a 2 ingressi
  - **Porta a N ingressi → N–1 porte a 2 ingressi**



Porta a **N** ingressi → Cammino Critico: **N-1**

- Ottimizzazione del cammino critico
  - Cammino critico: **ceil($\log_2$N)**



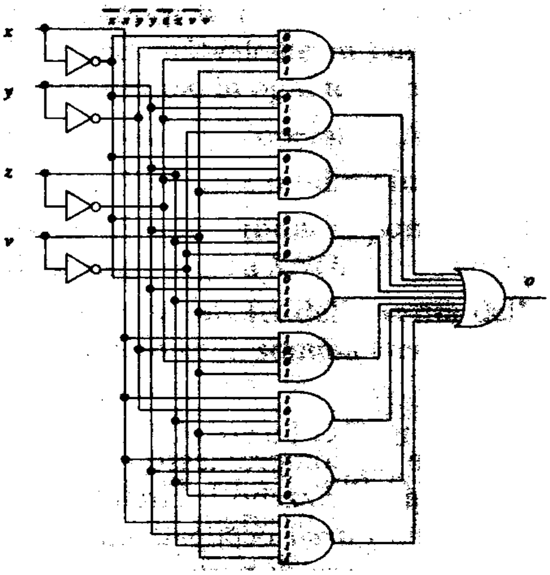
Cammino Critico: **ceil($\log_2 5$) = 3**

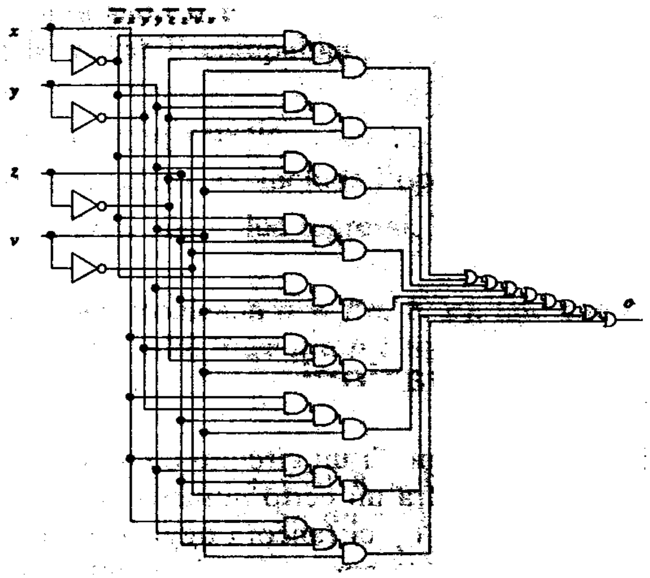
Sommario

- Tempi di commutazione
- Semplificazioni algebriche
- Semplificazioni con Mappe di Karnaugh
- Circuiti combinatori notevoli

Riduzione a circuiti con porte a 2 ingressi

$$O = \bar{x}\bar{y}\bar{z}\bar{v} + \boxed{\bar{x}y\bar{z}\bar{v} + \bar{x}y\bar{z}v} + \bar{x}yz\bar{v} + \bar{x}yzv + \bar{x}\bar{y}zv + \bar{x}yzv + xyz\bar{v} + xyzv$$





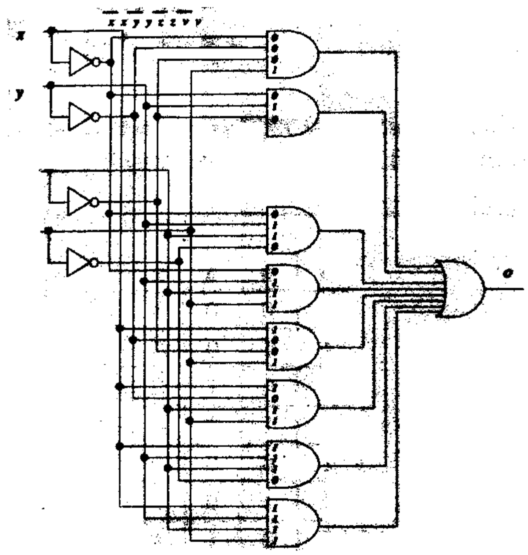
Cammino critico = 2 , N. porte = 10

Cammino critico = 11 , N. porte = 35

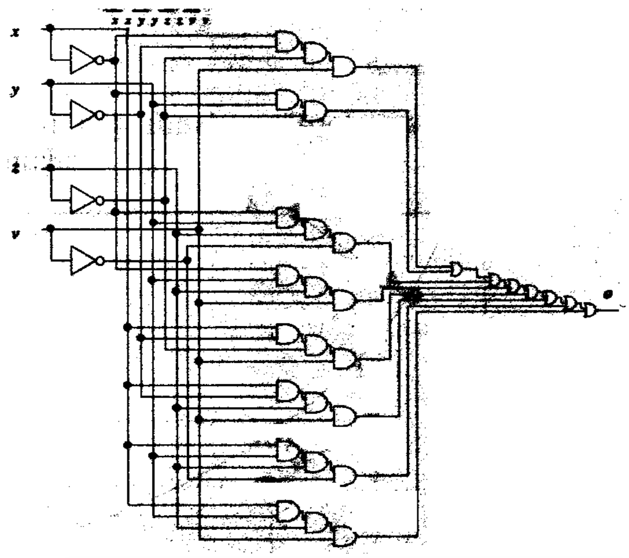
Semplificazione

- Semplificando la prima parte dell'espressione logica...

$$\bar{x}y\bar{z}\bar{v} + \bar{x}y\bar{z}v = \bar{x}y\bar{z}\left(\bar{v} + v\right) = \bar{x}y\bar{z}$$



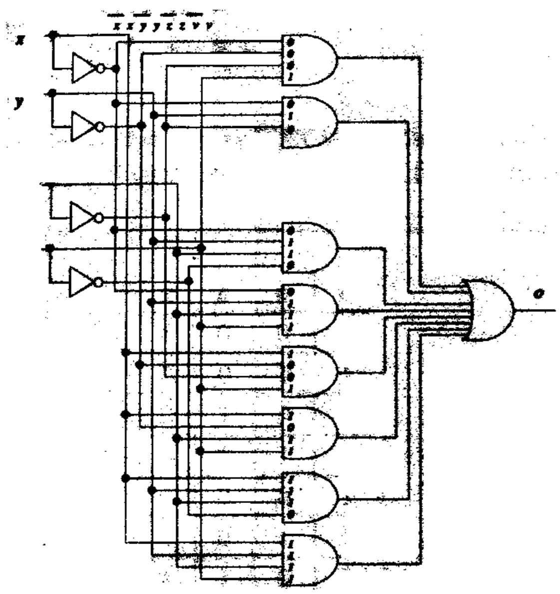
Cammino critico = 2   N. porte = 9

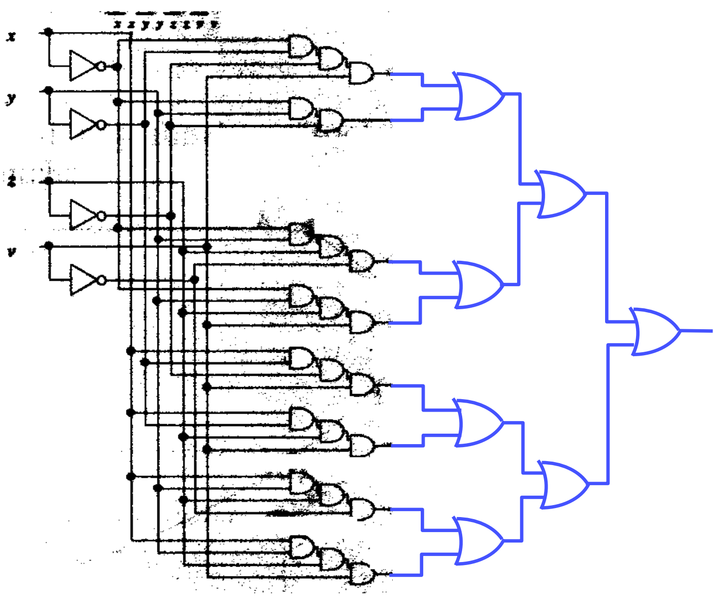


Cammino critico = 10   N. porte = 30

Ottimizzazione del cammino critico

- Collegando le porte in modo ottimizzato, si riduce significativamente il cammino critico...





Cammino critico = 6   N. porte = 30

Semplificazione

$$
\begin{aligned}
O &= \bar{x}\bar{y}\bar{z}\bar{v} + \bar{x}y\bar{z}\bar{v} + \bar{x}y\bar{z}v + \bar{x}yz\bar{v} + \bar{x}yzv + \bar{x}\bar{y}zv + \bar{x}yzv + xyz\bar{v} + xyzv \\
&= \bar{x}\bar{z}\bar{v}\left(y + \bar{y}\right) + \bar{x}y\bar{z}v + \bar{x}yz\left(\bar{v} + v\right) + \bar{x}zv\left(\bar{y} + y\right) + xyz\left(\bar{v} + v\right) = \\
&= \bar{x}\left(\bar{z}\bar{v} + zv + y\bar{z}v\right) + yz = \bar{x}\left(\bar{z}\bar{v} + v\left(z + \bar{z}y\right)\right) + yz = \\
&= \bar{x}\left(\bar{z}\bar{v} + v\left(z + y\right)\right) + yz = \\
&= \bar{x}\left(\bar{z}\bar{v} + vz + vy\right) + yz
\end{aligned}
$$

- Esercizio:
  - *disegnare il circuito logico relativo all'espressione semplificata*

Sommario

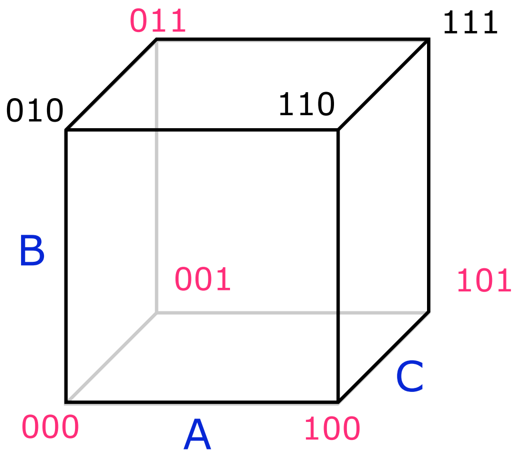
- Tempi di commutazione
- Semplificazioni algebriche
- Semplificazioni con Mappe di Karnaugh
- Circuiti combinatori notevoli

Semplificazione: mappe di Karnaugh

- Tecnica di semplificazione, a partire dalla tabella di verità
- Esempio: funzione di 3 variabili
  - Rappresentazione cubica di funzioni logiche a 3 variabili: F = *f*(a,b,c)
  - Muovendosi sui lati, la configurazione di variabili cambia di un solo bit
  - Distanza di HAMMING: ***d(v1, v2)*** = n. di bit diversi tra le sequenze

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

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

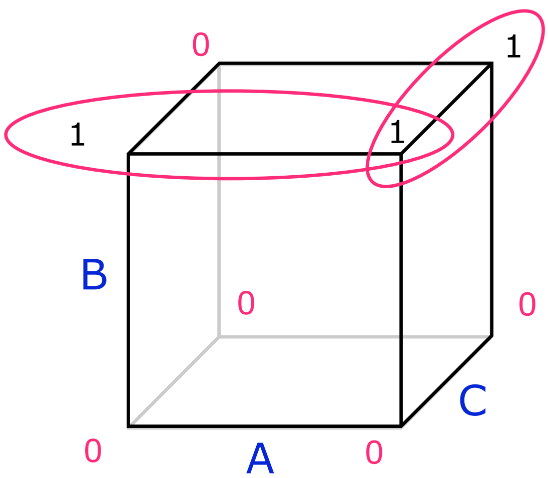


Semplificazione: mappe di Karnaugh

- **Copertura**: ricerca di tutti gli implicanti
- Se i vertici di un lato sono entrambi 1, l'implicante è indipendente dalla variabile corrispondente al lato

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

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



- Per **N>3** variabili, la rappresentazione diviene complessa…

Semplificazione: mappe di Karnaugh

❖ Rappresentazione piana della funzione:
- "srotolo" il cubo
- Codifica di Gray (codice riflesso) lungo ogni direzione

AB / C	00	01	11	10
0	000	010	110	100
1	001	011	111	101

indipendente da a: **b~c**

indipendente da c: **ab**

AB / C	00	01	11	10
0	0	1	1	0
1	0	0	1	0

F = ab + b~c

Semplificazione: mappe di Karnaugh

❖ Rappresentazione piana, utilizzabile per: **$2 \leq N \leq 4$**

A / B	0	1
0	1	0
1	1	0

F = ~a

AB / CD	00	01	11	10
00	0	1	1	0
01	0	0	1	0
11	1	1	1	1
10	0	0	1	0

F = ab + cd + b~c~d

Semplificazione: mappe di Karnaugh

- Mappa di Karnaugh: rappresentazione piana e ciclica

AB CD	00	01	11	10
00	0	1	1	0
01	0	0	1	0
11	1	0	1	1
10	0	0	1	0

F = ab + b~c~d + ~bcd

Esercizi

Data: F = AC + BC + ~A~BC + AB~C

1. *Scrivere la tabella di verità di F*
2. *Determinare la I e la II forma canonica di F*
3. *Semplificare F mediante mappa di Karnaugh*
4. *Ottenere la stessa semplificazione mediante operazioni algebriche*

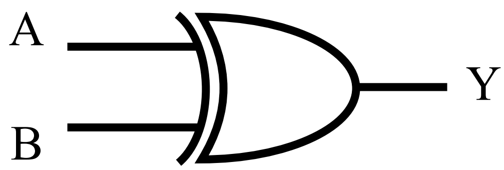
- *Calcolare la funzione F = f(A,B,C,D) tale che:*
  F=1 sse: il n. di ingressi='1' è 1 oppure 4
  1. *Calcolare la TT*
  2. *Det. la forma canonica più conveniente*
  3. *Semplificare la rappresentazione di F mediante mappe di Karnaugh*

Sommario

- Tempi di commutazione
- Semplificazioni algebriche
- Semplificazioni con Mappe di Karnaugh
- Circuiti combinatori notevoli

Operatore XOR

$$Y = A \text{ XOR } B = A \oplus B$$



- Operazione di **OR esclusivo**
  - Se **o A o B, non entrambi,** sono veri, **A XOR B** è vera

Tabella della verità

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

$$SoP: \quad Y = \overline{A}B + A\overline{B}$$

$$PoS: \quad Y = (A+B)\cdot(\overline{A}+\overline{B})$$

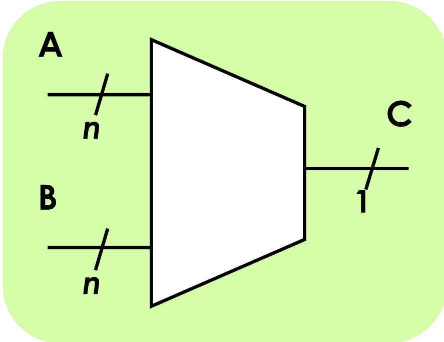
$$Y = (A+B)\cdot(\overline{A}+\overline{B}) =$$

$$= \cancel{A\overline{A}} + A\overline{B} + \overline{A}B + \cancel{B\overline{B}} = A\overline{B} + \overline{A}B \qquad c.v.d.$$

Comparatore

- Confronta parole di **n** bit
  - IN: **2** gruppi di **n** bit
  - OUT: **1** bit

  - OUT = 1 se i due IN sono <u>uguali</u>
  - OUT = 0 se diversi.



A_0	B_0	C_0
0	0	1
0	1	0
1	0	0
1	1	1

A_1	B_1	C_1
0	0	1
0	1	0
1	0	0
1	1	1

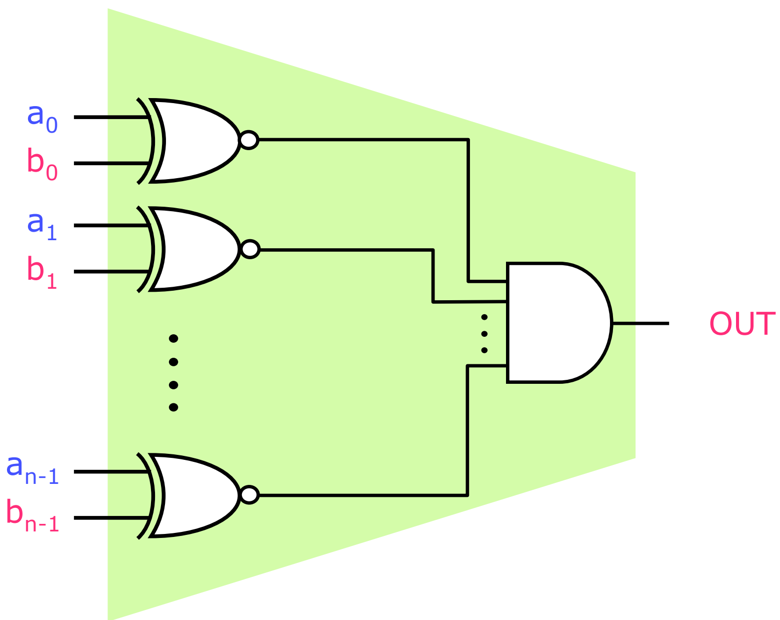
$$C_i = \overline{A_i \oplus B_i}$$

$$C = C_0 \cdot C_1 \cdot \ldots \cdot C_{n-1}$$

n porte **XOR** negate + 1 **AND** ad **n** ingressi

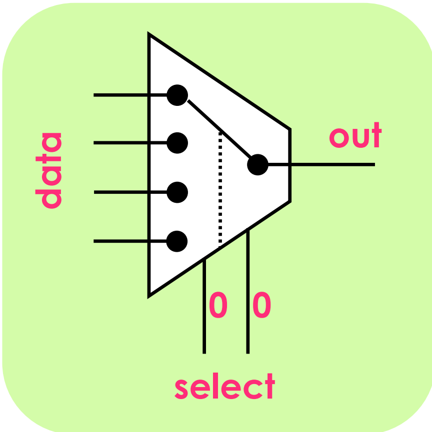
Comparatore

- Comparatore a n ingressi: schema circuitale



Multiplexer

- **Operatore di selezione**
  - **IN:** **n** linee di input (**data**)
    k linee di controllo (**select**)
  - **OUT:** **1** linea
    - Il valore fornito sulla linea di controllo viene connessa all'uscita la linea di ingresso selezionata.



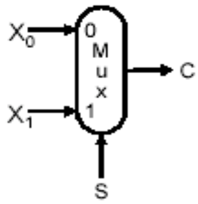
- Quante linee di controllo?

  k = ceil ($\log_2 n$)

  - Linee di input: $n = 4$
  - Linee di controllo: $k = ceil(\log_2 4) = 2$

Multiplexer binario

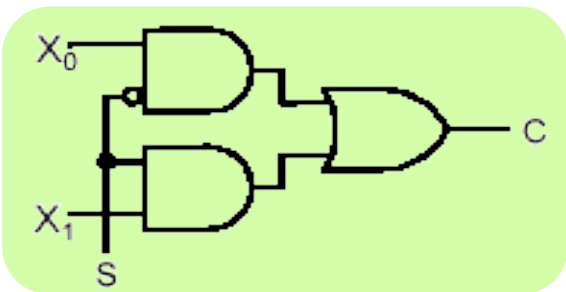
- **n = 2, k = 1**
  - Selezione S "apre" la porta opportuna
  - Circuito logico a **3 ingressi, 1 uscita**



S	X_0	X_1	C
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

SoP:

$$C = \overline{S}X_0\overline{X_1} + \overline{S}X_0X_1 + S\overline{X_0}X_1 + SX_0X_1 =$$
$$= \overline{S}X_0 + SX_1$$



Sintesi Multiplexer in forma canonica POS

$$C = (S + X_0 + X_1)(S + X_0 + \overline{X_1})(\overline{S} + X_0 + X_1)(\overline{S} + \overline{X_0} + X_1) =$$

$$\begin{cases} a = \overline{S} + X_1 \\ b = S + X_0 \end{cases}$$

$$= \left[(b + X_1)(b + \overline{X_1})\right] \cdot \left[(a + X_0)(a + \overline{X_0})\right] =$$

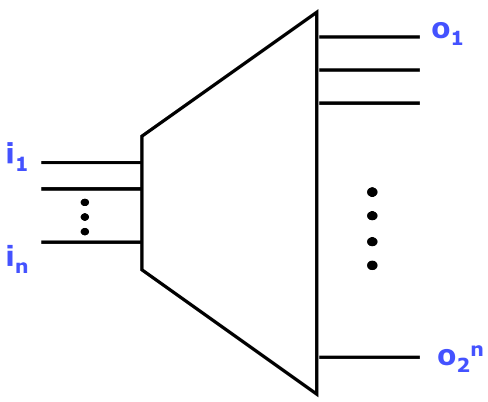
$$= \left[b + b\left(X_1 + \overline{X_1}\right) + X_1\overline{X_1}\right] \cdot [a] = ba =$$

$$= (S + X_1)(\overline{S} + X_0)$$

S	X_0	X_1	C
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

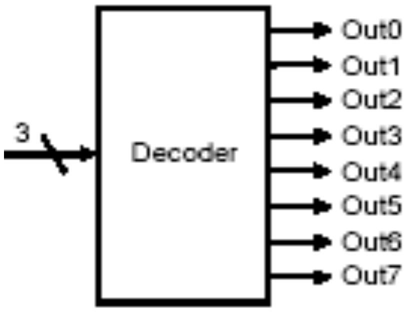
Decodificatore (decoder)

- **n** ingressi, **2^n** uscite
- Il numero espresso sugli ingressi è usato per asserire (porre a 1) la linea di uscita di tale indice
  - con 4 linee di input e 16 di output (da 0 a 15), se in ingresso arriva il valore 0110, in uscita si alza la linea di indice 5 (la sesta!)
- usato per inidirizzare la memoria



La funzione decoder

Decoder



a. A 3-bit decoder

A	B	C	U_0	U_1	U_2	U_3	U_4	U_5	U_6	U_7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

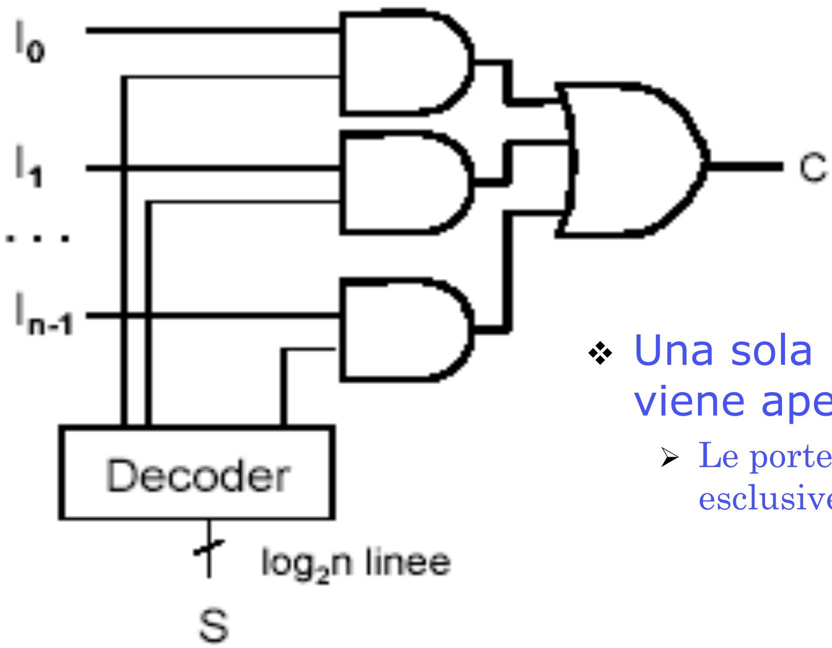
Le funzioni di uscita sono 2^n per n input:

$U_0 = \sim A \sim B \sim C$

...

$U_7 = A\ BC$

Multiplexer (Mux) a più vie.



- Una sola porta alla volta viene aperta dal segnale S.
  - Le porte sono mutuamente esclusive.

HALF Adder (1 bit)

- Somma aritmetica tra 2 bit
  - 2 ingressi: addendi: **a , b**
  - 2 uscite: somma: **s**
    riporto: **r**

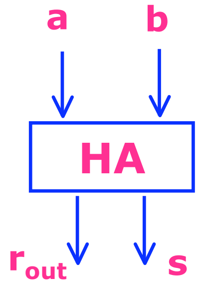
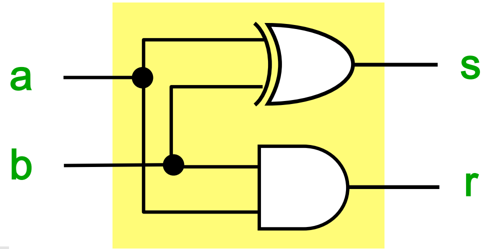


Tabella della verità			
a	b	somma	riporto
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

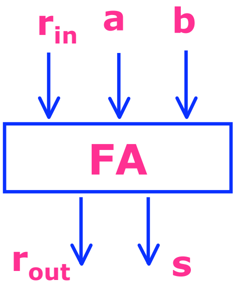
$s = a \oplus b$

$r = a \bullet b$



FULL Adder (1 bit)

- Gestisce anche il riporto in ingresso
  - 3 ingressi: a, b, r_{IN}
  - 2 uscite: s, r_{OUT}



a	b	r_{in}	r_{out}	s
0	0	0	0	0
0	1	0	0	1
1	0	0	0	1
1	1	0	1	0
0	0	1	0	1
0	1	1	1	0
1	0	1	1	0
1	1	1	1	1

SOP:

$$s = m_1 + m_2 + m_4 + m_7$$

$$r_{out} = m_3 + m_5 + m_6 + m_7$$

$$s = \bar{a} b \overline{r_{in}} + a \bar{b} \overline{r_{in}} + \bar{a} \bar{b} r_{in} + a b r_{in}$$

$$r_{out} = a b \overline{r_{in}} + \bar{a} b r_{in} + a \bar{b} r_{in} + a b r_{in}$$

Semplificando le espressioni

$$
\begin{aligned}
s &= \bar{a}b\overline{r_{in}} + a\bar{b}\overline{r_{in}} + \bar{a}\bar{b}r_{in} + abr_{in} = \\
&= (a \oplus b)\overline{r_{in}} + \left(ab + \bar{a}\bar{b}\right)r_{in} = \\
&= \boxed{(a \oplus b)\overline{r_{in}} + \overline{(a \oplus b)}r_{in} = (a \oplus b) \oplus r_{in}}
\end{aligned}
$$

$$
\begin{aligned}
r_{out} &= ab\overline{r_{in}} + \bar{a}br_{in} + a\bar{b}r_{in} + abr_{in} = \\
&= ab\left(r_{in} + \overline{r_{in}}\right) + \left(a\bar{b} + \bar{a}b\right)r_{in} = \\
&= \boxed{ab + (a \oplus b)r_{in}}
\end{aligned}
$$

Implementazione circuitale

$$
\boxed{\begin{aligned} s &= (a \oplus b)\overline{r_{in}} + \overline{(a \oplus b)}r_{in} \\ r_{out} &= ab + (a \oplus b)r_{in} \end{aligned}} = \boxed{(a \oplus b) \oplus r_{in}}
$$