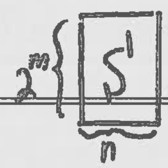
Riepilogando: vogliamo costruire un banco di memoria.

- problema della rappresentazione dei dati
- logica di funzionamento (ARCHITETTURA) e di

ORGANIZZAZIONE (prendere la parola di indirizzi e giocare sulla sua ripartizione per poter usare i chip di memoria in maniera diversa

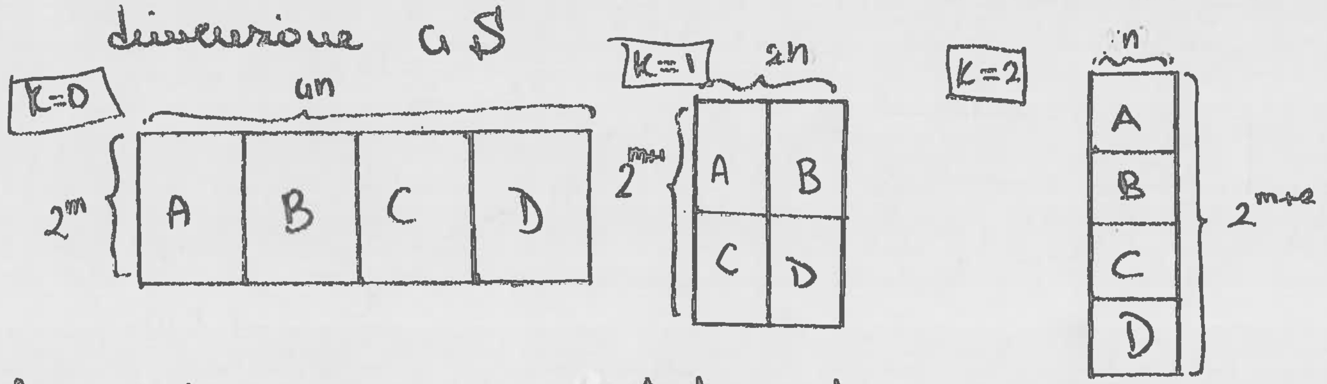
Dato un chip (modulo) di memoria di dimensione

$S' = 2^m \times n$



, posso sfruttare schemi logici

ovvero per arrivare ad avere una memoria di dimensione 4S

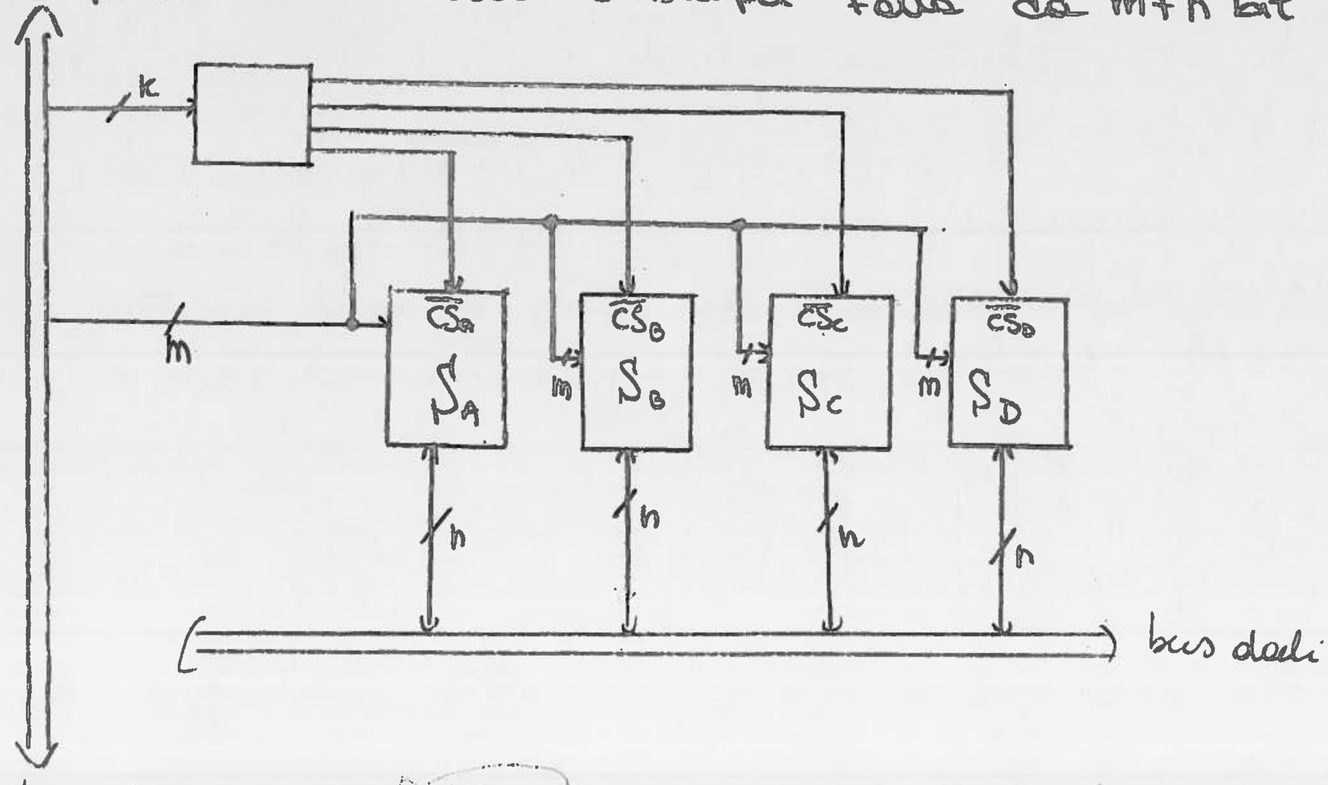


$A_{m-1} \dots A_0$
parola di indirizzo

$A_m A_{m-1} \dots A_0$
parola di indirizzo

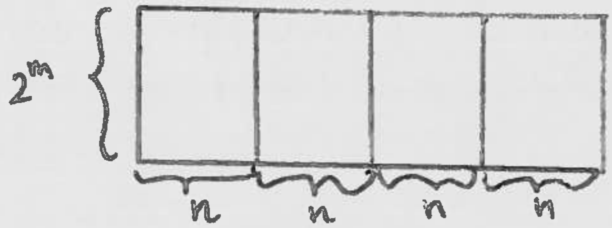
$A_{m+1} A_m \dots A_1 A_0$
parola di indirizzo

La parola di indirizzo è sempre fatta da m+K bit



bus indirizzi (m+K) bit) dipende dall'organizzazione scelta

• per $K=0$ ogni chip di memoria funziona simultaneamente per dare il suo contributo. In questo caso il decoder è una mena a mena

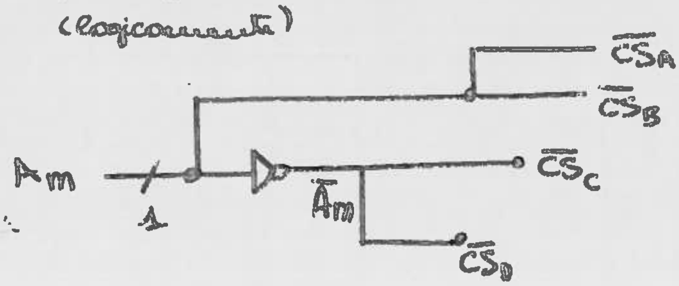


$\forall i \ \overline{CS_i} = 0$

• per $K=2$ funziona solo un modulo alla volta

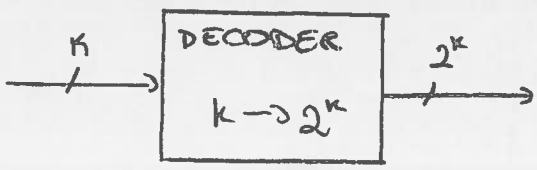
• per $K=1$ i moduli di memoria funzionano due alla volta (logicamente)

MSB =	0	A e B
	1	C e D



Nel caso $1 \to 2$ ~~bom~~

In generale, definiamo DECODER un circuito combinatorio dotato di K ingressi e 2^k uscite, ognuna delle quali ha valore 0 o 1 logico di funzionamento → solo un'uscita assume un valore diverso da tutte le altre, e quest'uscita corrisponde al codice binario presentato all'ingresso

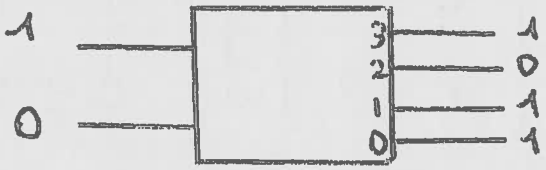


$N(x_1,x_0)$		x_1	x_0	z_1	z_2	z_3	z_4	z_1'	z_2'	z_3'	z_4'	
(0)	1	0	0	1	1	1	0	0	0	0	1	
(1)	2	0	1	1	1	0	1	0	0	1	0	(*) a
(2)	3	1	0	1	0	1	1	0	1	0	0	
(3)	4	1	1	0	1	1	1	1	0	0	0	

$z_1 \dots z_4$: logica negativa — $z_1' \dots z_4'$: logica positiva

Il decoder riconosce l'occorrenza del codice binario in ingresso e mette in uscita la colonna corrispondente

es (log. negativa)



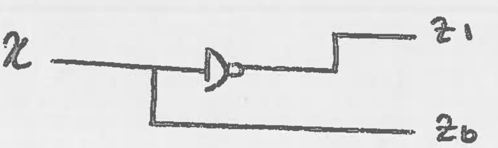
uscita 2 asserita bassa

è la colonna z_2 a testa in giù

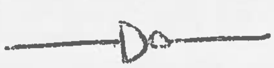
In logica positiva l'unica uscita diversa dalle altre vale 1, ma la logica ~~del~~ di funzionamento del decoder è la stessa

es. $z_i' = \overline{z_i}$ in (*) a

Decoder 1→2 in logica negativa



Il decoder 2→4, a seconda dell'organizzazione scelta, dovrà contenere qualcosa di più complesso di una porta NOT



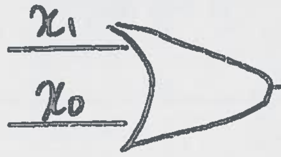
la sua tabella associata sarà del tipo

x_0	x_1	$f(x_1, x_0)$
0	0	f_0
0	1	f_1
1	0	f_2
1	1	f_3

tutte le possibili funzioni logiche implementabili con due variabili sono

	x_1	x_0	z_3	z_2	z_1	z_0	z'_3	z'_2	z'_2	z'_1						
0	0	0	1	1	1	0	0	0	0	1	0	1	0	1	0	1
1	0	1	1	1	0	1	0	0	1	0	0	1	0	1	1	0
2	1	0	1	0	1	1	0	1	0	0	0	1	1	0	1	0
3	1	1	0	1	1	1	1	0	0	0	0	1	1	0	0	1
			NAND			OR	AND			OR	15	0			XOR	
															10	

• OR funzione che assume valore 1 quando almeno uno dei suoi valori in ingresso è 1



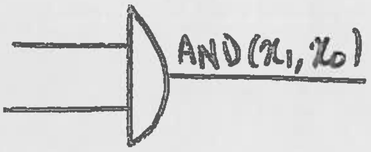
$OR(x_1, x_0) = x_1 + x_0$ (o el) SOMMA LOGICA

corrisponde $x_1 \vee x_0$

• AUT-AUT è un OR esclusivo. assume valore 1 se e solo se uno degli ingressi è 1 e l'altro è 0

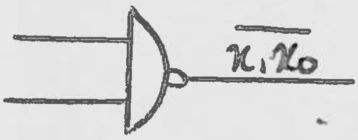
$x_1 \oplus x_0$

• AND prodotto logico. E' 1 $\Leftrightarrow$ $x_1 = x_0 = 1$, 0 altrimenti

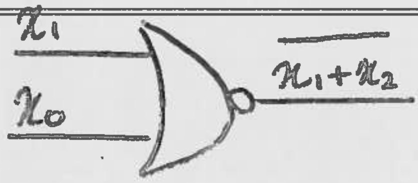


$= x_1 x_0$

• NAND negazione del prodotto logico



• NOR negazione della somma logica

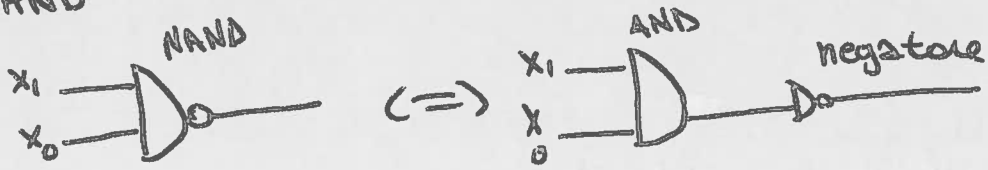


E' possibile costruire una qualsiasi rete logica a partire da AND, OR e NOT, con un numero di variabili qualsiasi (anche >2).

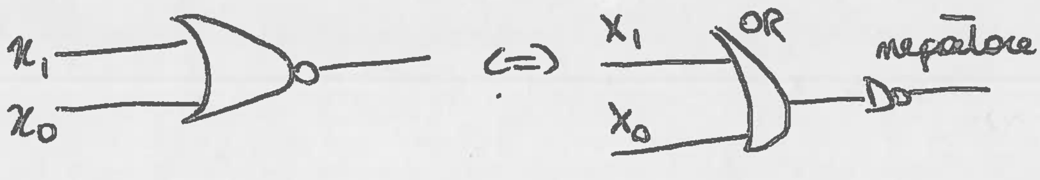
NOT  è un operatore unario

<u>Oss.</u>

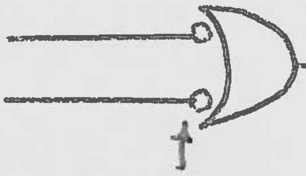
NAND



NOR



Pertanto posso costruire una qualsiasi rete logica a partire da NAND e NOR



$?(x_1, x_0) = \bar{x}_1 + \bar{x}_0$

x_1, x_0 vengono negate prima di essere inserite

Che porta è?

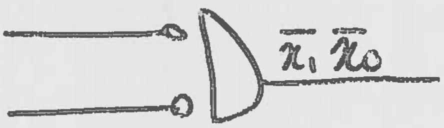
x_1	x_0	$\bar{x}_1 + \bar{x}_0$
0	0	1
0	1	1
1	0	1
1	1	0

è una NAND

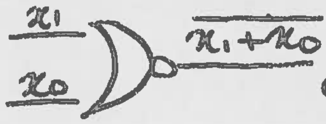
dunque

$$\overline{x_1 x_0} = \bar{x}_1 + \bar{x}_0$$

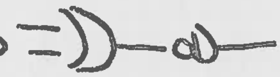
Analogamente



$\Leftrightarrow$



$\Leftrightarrow$



x_1	x_0	$\bar{x}_1 \bar{x}_0$
0	0	1
0	1	0
1	0	0
1	1	0

è una NOR

$$\bar{x}_1 \bar{x}_0 = \overline{x_1 + x_0}$$

Leggi di De Morgan

$$\overline{x_1 + x_2 + \dots + x_m} = \bar{x}_1 \cdot \dots \cdot \bar{x}_m$$

$$\overline{x_1 x_2 \dots x_m} = \bar{x}_1 + \dots + \bar{x}_m$$

$$f(x_1,x_0) = (\bar{x}_1\bar{x}_0)\overset{0}{f_0} + (\bar{x}_1 x_0)\overset{1}{f_1} + (x_1\bar{x}_0)\overset{0}{f_2} + (x_1 x_0)\overset{0}{f_3}$$

dunque l'unico termine della ~~somma~~ che conta e' f_1

- Se $f_0 = 1$, $f_1 = f_2 = 0$, $f_3 = 1$ nella somma di prodotti non contiamo il secondo e il terzo adottando

elementi neutri $\begin{cases} 0 + x = 0 \\ 1 \cdot x = 1 \end{cases}$

---

elementi forzanti $\begin{cases} 0 \cdot x = 0 \\ 1 + x = 1 \end{cases}$

dunque $f(x_1,x_0) = \bar{x}_1\bar{x}_0 \cdot 1 + x_1 x_0 \cdot 1 = \overline{x_1 \oplus x_2} = NXOR(x_1, x_0)$

- data la tabella della funzione logica

x_1	x_0	$f(x_1,x_0)$
0	0	1
0	1	1
1	0	0
1	1	0

l'espressione canonica di f come somma di prodotti e'

$$f(x_1,x_0) = \bar{x}_1\bar{x}_0 + \bar{x}_1 x_0 = \bar{x}_1 \underbrace{(\bar{x}_0 + x_0)}_{=1} = \bar{x}_1$$

Os. i) $\bar{\bar{x}} = x$

ii)

x	$x+\bar{x}$
0	1
1	1

iii) $x + \cdots + x = x$

$x \cdot \cdots \cdot x = x$

$$f(x_1, x_0) = \bar{x}_1\bar{x}_0 f_0 + \bar{x}_1 x_0 f_1 + x_1\bar{x}_0 f_2 + x_1 x_0 f_3$$

$$\overline{f(x_1, x_0)} = \bar{x}_1\bar{x}_0 \bar{f}_0 + \bar{x}_1 x_0 \bar{f}_1 + x_1\bar{x}_0 \bar{f}_2 + x_1 x_0 \bar{f}_3$$

$$\Rightarrow f(x_1, x_0) = \overline{\overline{f(x_1, x_0)}} = \overline{\bar{x}_1\bar{x}_0 \bar{f}_0 + \bar{x}_1 x_0 \bar{f}_1 + x_1\bar{x}_0 \bar{f}_2 + x_1 x_0 \bar{f}_3} =$$

$$= \overline{(\bar{x}_1\bar{x}_0 \bar{f}_0)} \cdot \overline{(\bar{x}_1 x_0 \bar{f}_1)} \cdot \overline{(x_1\bar{x}_0 \bar{f}_2)} \cdot \overline{(x_1 x_0 \bar{f}_3)} =$$

$$= (x_1 x_0 f_0) \cdot (x_1 \bar{x}_0 f_0) \cdot (\bar{x}_1 x_0 f_2)$$

De Morgan $= (x_1 + x_0 + f_0) \cdot (x_1 + \bar{x}_0 + f_1) \cdot (\bar{x}_1 + x_0 + f_2) \cdot (\bar{x}_1 + \bar{x}_0 + f_3)$

prodotto di somme

---

esempio

x_1	x_0	$f(x_1, x_0)$
0	0	1
0	1	1
1	0	0
1	1	1

PS $(\bar{x}_1 + x_0)$

considero i valori di x_0, x_1 t.c. $f(x_1, x_0) = 0$

SP $\bar{x}_1\bar{x}_0 + \bar{x}_1 x_0 + x_1 x_0$