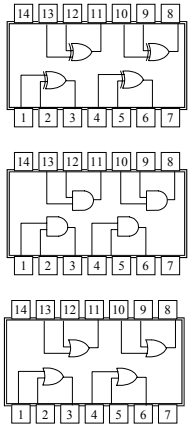
Esercitazione N. 7

Progetto di un sommatore per numeri di due bit



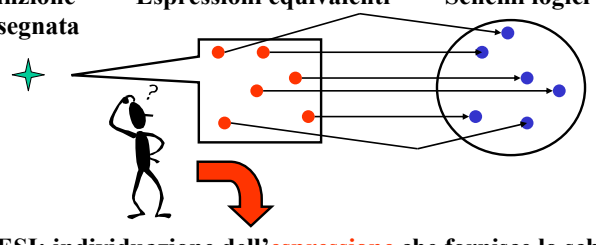
Capitolo 5
Reti combinatorie

5.1 - Il problema della sintesi
5.2 - Reti di costo minimo
5.3 – Il metodo delle mappe
5.4 – Reti programmabili

5.1
Il problema della sintesi

Il problema della sintesi

Funzione assegnata — **Espressioni equivalenti** — **Schemi logici**



SINTESI: individuazione dell'espressione che fornisce lo schema "migliore" per la realizzazione della funzione assegnata.

- **Rapidità di progetto**
- **Massima velocità**
- **Massima flessibilità**
- **Minima complessità**

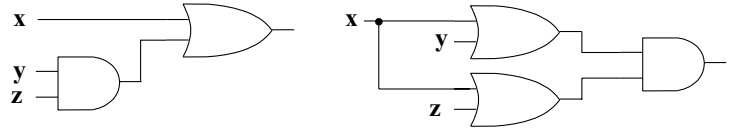
5.2 Reti di costo minimo

Complessità e velocità

Indicatori : N_{gate} = numero di gate,
N_{conn} = numero di connessioni
N_{casc} = numero di gate disposti in cascata sul più lungo percorso di elaborazione

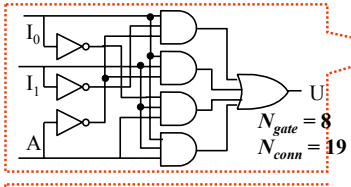
- Complessità $\Rightarrow$ funzione crescente di N_{gate} , N_{conn}
- Velocità di elaborazione $\Rightarrow$ funzione decrescente di N_{casc}

Esempio:



- Le due reti sono equivalenti (E3).
- Hanno la stessa velocità di elaborazione.
- La rete di sinistra è meno complessa.

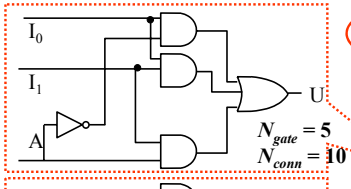
Di nuovo il selettore a due vie



$U = A'.I_0.I_1' + A'.I_0.I_1 + A.I_0'.I_1 + A.I_0.I_1$

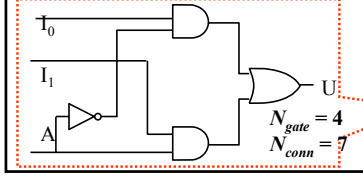
E4: (x + x = x)

$U = A'.I_0.I_1' + A'.I_0.I_1 + A.I_0'.I_1 + A.I_0.I_1 + A'.I_0.I_1 + A.I_0.I_1$



E3, E9: (xy' + xy = x)

$U = A'.I_0 + I_1.I_0 + A.I_1$



E11: (xy + x'z + yz = xy + x'z

$U = A'.I_0 + A.I_1$

Schemi logici di "costo minimo"

IPOTESI:
- ingressi in forma vera e complementata
- fan-in grande quanto serve

Rete combinatoria di costo minimo (tipo SP e tipo PS) - Schema logico che realizza una funzione qualsiasi con
1. **non più di 2 gate in cascata tra ingressi e uscita**
2. **minimo numero di gate**
3. **minimo numero di ingressi per gate.**

N.B. - Il numero di gate e/o di connessioni della rete di costo minimo di tipo SP è in generale diverso da quello della rete di costo minimo di tipo PS che realizza la stessa funzione.

Espressioni minime

Espressione minima (SP/PS) - Descrizione algebrica di una rete di costo minimo: espressione normale (SP/PS) formata dal minimo numero possibile di "termini" (prodotti/somme) aventi ciascuno il minimo numero possibile di "letterali" (variabili in forma vera o complementata).

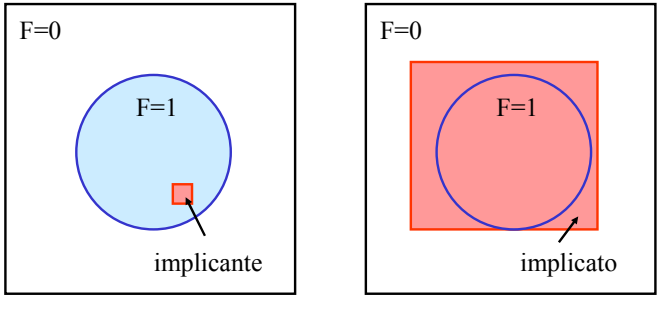
N.B - E' possibile che più espressioni normali dello stesso tipo siano minime (abbiano cioè eguali valori di N_{gate} e N_{conn}).

Implicanti e implicati primi

Implicante di una funzione - Termine prodotto di n o meno variabili che assume valore 1 per configurazioni per cui anche la funzione vale 1.
Implicante primo - Implicante da cui non è possibile eliminare alcun letterale senza perdere la sua proprietà.
Implicato di una funzione - Termine somma di n o meno variabili che assume valore 0 per configurazioni per cui anche la funzione vale 0.
Implicato primo - Implicato da cui non è possibile eliminare alcun letterale senza perdere la sua proprietà.

N.B. - Gli implicanti (implicati) primi di una funzione si individuano a partire dalla prima (seconda) forma canonica applicando in ogni maniera possibile e finché è possibile **E9.**

Implicanti e Implicati



Condizioni necessarie

Espressione irridondante - Espressione normale SP o PS da cui non può essere eliminato alcun termine senza invalidare l'equivalenza con l'espressione stessa.

T12) L'espressione minima SP è una somma irridondante di implicanti primi.

T13) L'espressione minima PS è un prodotto irridondante di di implicati primi.

N.B - **E9** e **E11** sono gli "strumenti" che consentono di passare per manipolazione algebrica dall'espressione canonica a quella minima.

Metodi per la determinazione dell'espressione minima

Metodi algoritmici
(Quine-Mc Cluskey, Petrick)
consentono di trattare funzioni con un numero qualsiasi di variabili e vengono tipicamente eseguiti da un calcolatore.

Metodo grafico
(Mappe di Karnaugh)
consente di trattare agevolmente funzioni fino a 6 variabili.
e viene eseguito manualmente.

5.3
Il metodo delle mappe

Mappe di Karnaugh

Mappe

Mappa di Karnaugh - Rappresentazione bidimensionale della tabella della verità di una funzione di 2,3,4 variabili, i cui valori sono stati elencati sui bordi in maniera che due configurazioni consecutive differiscano per il valore di un solo bit.

Esempi:

a \ b	0	1
0	0	1
1	1	1

Somma di 2 variabili

a \ br	00	01	11	10
0	0	0	1	0
1	0	1	1	1

Riporto del Full Adder

ab \ cd	00	01	11	10
00	0	1	0	1
01	1	0	1	0
11	0	1	0	1
10	1	0	1	0

Parità su 4 variabili

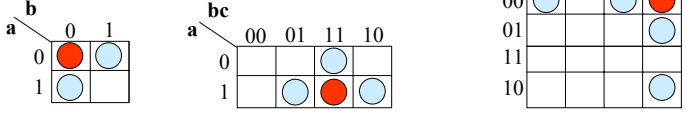
Adiacenza tra celle

Coppia di celle adiacenti su mappe di Karnaugh - Due celle le cui coordinate differiscono per un solo bit.
In una mappa che descrive una funzione di *n* variabili ogni cella ha *n* celle adiacenti.

Regola grafica per l'adiacenza - Sono adiacenti celle aventi un lato in comune o poste all'estremità di una stessa riga o colonna.

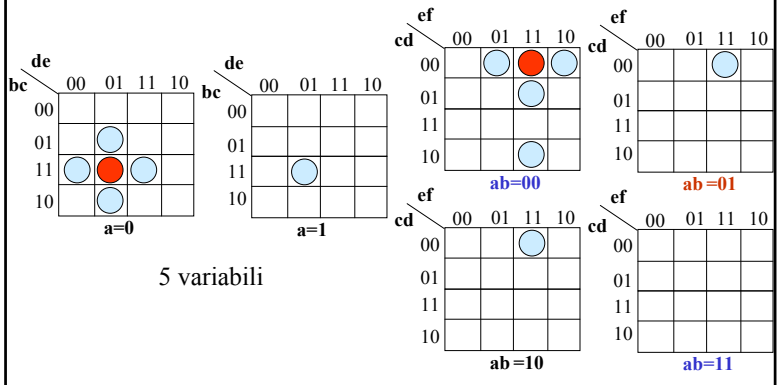
cella scelta come esempio

celle adiacenti



2 variabili 3 variabili 4 variabili

Estensione delle mappe a 5 e a 6 variabili



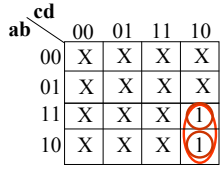
5 variabili

6 variabili

Ulteriore regola di adiacenza - Sono adiacenti celle che occupano la stessa posizione in sotto-mappe adiacenti.

Ricerca degli implicanti e degli implicati

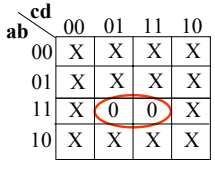
Manipolazione algebrica per via grafica (1)



canonica SP:
ab'cd' + abcd' + ...

E9

normale SP: acd' + ...



canonica PS:
(a'+b'+c+d')(a'+b'+c'+d') ...

E9

normale PS: (a'+b'+d')...

Due termini di una espressione canonica (SP o PS) corrispondenti a configurazioni che individuano celle adiacenti sono equivalenti ad un unico termine con un letterale in meno.

Manipolazione algebrica per via grafica (2)

ab \ cd	00	01	11	10
00	X	X	X	X
01	X	1	1	X
11	X	1	1	X
10	X	X	X	X

canonica SP: a'bc'd + a'bcd + abc'd + abcd + ...

E9 E9

normale SP: a'bd + abd + ...

E9

normale SP: bd + ...

Quattro mintermini corrispondenti a configurazioni che individuano un "raggruppamento" di 4 celle a 2 a 2 adiacenti sono equivalenti ad un unico termine con due letterali in meno.

Manipolazione algebrica per via grafica (3)

La proprietà è vera anche per quattro maxtermini

ab \ cd	00	01	11	10
00	X	X	X	X
01	X	X	X	X
11	X	X	0	0
10	X	X	0	0

canonica PS:

(a'+b'+c'+d').(a'+b'+c'+d).(a'+b+c'+d').(a'+b+c'+d) .(...

E9 E9

normale PS: (a'+b'+c') · (a'+b+c') .(...

E9

normale PS: (a' + c').(…

Individuazione dei termini primi

Raggruppamenti rettangolari

Raggruppamento Rettangolare (RR) di ordine p - Insieme di 2^p celle di una mappa all'interno del quale ogni cella ha esattamente p celle adiacenti.

RR ed implicanti -Un RR di ordine p costituito da celle contenenti valore 1, ed eventualmente condizioni di indifferenza, individua un implicante della funzione. Nel prodotto compaiono le sole (n-p) variabili che rimangono costanti nel RR, in forma vera se valgono 1, in forma complementata se valgono 0.

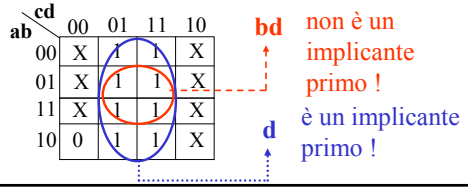
RR ed implicati -Un RR di ordine p costituito da celle contenenti valore 0, ed eventualmente condizioni di indifferenza, individua un implicato della funzione. Nella somma compaiono le sole (n-p) variabili che rimangono costanti nel RR, in forma vera se valgono 0, in forma complementata se valgono 1.

Raggruppamenti, Implicanti e Implicati

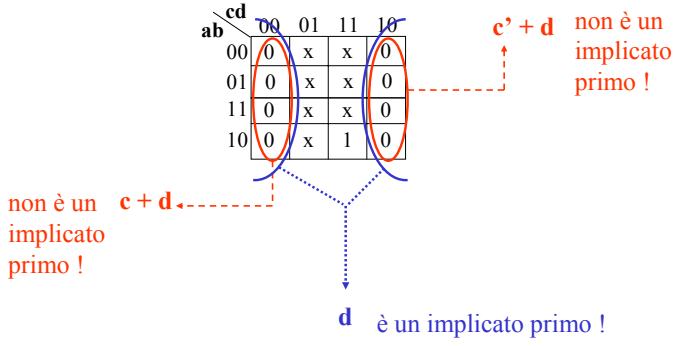
RR di dimensione massima ed implicanti primi - Un RR formato da celle contenenti valore "1" o "-" e non interamente incluso in un RR di ordine superiore individua un implicante primo.

RR di dimensione massima ed implicati primi - Un RR formato da celle contenenti valore "0" o "-" e non interamente incluso in un RR di ordine superiore individua un implicato primo (RR).

Esempio (caso SP):

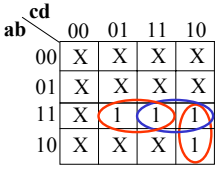


Esempio (caso PS)



Individuazione grafica dei termini ridondanti

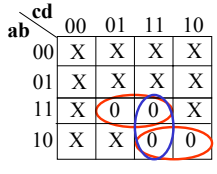
Un RR le cui celle sono tutte incluse in altri RR può non essere preso in considerazione per E11.



normale SP:
acd' + abc + abd + ...

E11

normale SP: acd' + abd + ...



normale PS:
(a'+b'+d')(a'+c'+d')(a'+b+c') ...

E11

normale PS: (a'+b'+d')(a'+b+c') ...

Individuazione della copertura minima

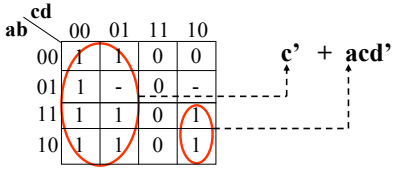
Copertura minima

Copertura di una funzione su una mappa - Insieme di RR la cui unione racchiude tutte le celle contenenti o valore 1 (copertura degli uni) o valore 0 (copertura degli zeri), ed eventualmente celle con valore indifferente.

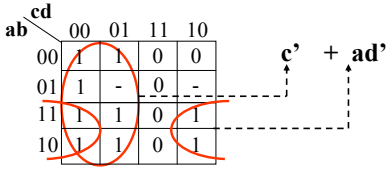
Coperture ed espressioni normali - Una copertura degli uni (zeri) individua una espressione normale SP (PS) che descrive, nel suo dominio, la funzione assegnata tramite la mappa. Gli implicanti (implicati) che appaiono nell'espressione sono individuati dai raggruppamenti componenti la copertura.

Copertura minima - Copertura costituita dal minimo numero possibile di RR di dimensione massima e corrispondente alla espressione minima.

Coperture ed espressioni (1)

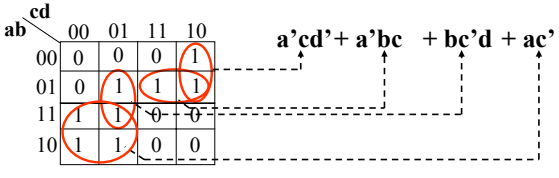


Uno dei due RR non è di dimensione massima (acd' non è un implicante primo): l'espressione non è minima.

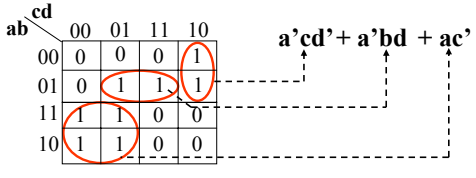


L'espressione è minima !

Coperture ed espressioni (2)

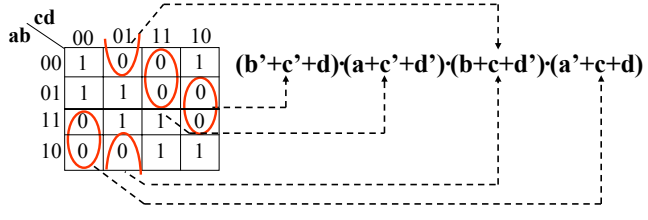


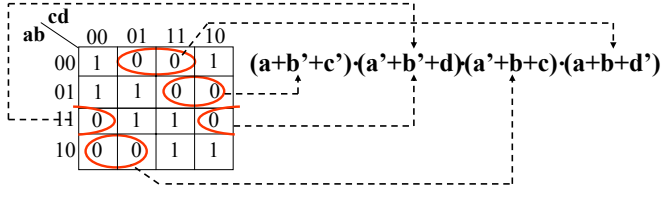
Somma irridondante di implicanti primi, ma non espressione minima



Espressione minima

Coperture ed espressioni (3)





Due espressioni minime di tipo PS

Coperture ed espressioni (4)

ab \ cd	00	01	11	10
00	1	1	1	1
01	1	1	1	1
11	-	1	-	1
10	1	1	-	1

La funzione f(a, b, c, d) è identicamente uguale a 1

a \ bc	00	01	11	10
0	0	1	0	0
1	1	0	1	0

a' b'c + ab'c' + abc

L'espressione minima SP è l'espressione canonica

ab \ cd	00	01	11	10
00	0	1	1	0
01	1	1	1	-
11		1	1	1
10	0	1	1	0

PS: b + d

SP: b + d

Le coperture minime PS ed SP portano alla stessa espressione

Individuazione grafica dell'espressione minima (1)

A partire dalla mappa che descrive la funzione occorre determinare la copertura minima e da questa la corrispondente espressione minima. Il procedimento è per sua natura non sistematico e presuppone l'abilità di chi lo esegue.

È tuttavia possibile delineare una sequenza di passi che consentono di individuare con facilità la copertura minima:

1) Si decide se cercare l'espressione di tipo SP o PS e ci si predispone di conseguenza a coprire gli uni o gli zeri.

ab \ cd	00	01	11	10
00	0	0	0	1
01	0	1	-	-
11	1	1	0	0
10	1	1	0	0

1) scegliamo SP

Individuazione grafica dell'espressione minima (2)

2) Si cerca di individuare tra le celle da coprire una cella che possa essere racchiusa in un solo RR e lo si traccia di dimensione massima, annotando il termine corrispondente. Se la funzione è incompleta il RR può contenere anche condizioni di indifferenza.

ab \ cd	00	01	11	10
00	0	0	0	1
01	0	1	-	-
11	1	1	0	0
10	1	1	0	0

1) scegliamo SP
2) a'cd'

Individuazione grafica dell'espressione minima (3)

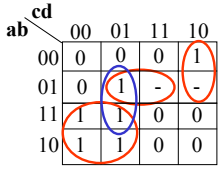
3) Si ripete fino a quando è possibile il passo 2, tenendo conto della possibilità di coprire anche celle incluse in RR già tracciati.

ab \ cd	00	01	11	10
00	0	0	0	1
01	0	1	-	-
11	1	1	0	0
10	1	1	0	0

1) scegliamo SP
2) a'cd'
3) ac'

Individuazione grafica dell'espressione minima (4)

4) Si prendono in considerazione le celle ancora da coprire e se ne sceglie a colpo d'occhio la copertura migliore, tenendo conto come al solito della possibilità di coprire celle già coperte e condizioni di indifferenza.

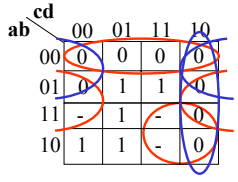


1) scegliamo SP
2) a'cd'
3) ac'
4) a'bd *oppure* bc'd

5) Si ripete il passo 4 fino a soddisfare la condizione di copertura. Si scrive infine l'espressione minima.

5) a'cd' + ac' + { a'bd / bc'd }

Individuazione grafica della espressione minima (5)



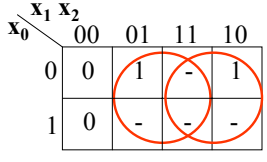
1) scegliamo PS
2) a+b
3) b'+d *oppure* a + d
4) a'+ c' *oppure* c'+d
5) (a+b)· { b'+d / a +d } · { a'+c' / c'+d }

(a+b)·(a +d) ·(c'+d)

Sintesi minima di un encoder

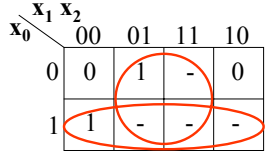
x_2	x_1	x_0	z_1	z_0
0	0	0	0	0
1	0	0	1	1
0	1	0	1	0
0	0	1	0	1
1	1	0	-	-
1	0	1	-	-
0	1	1	-	-
1	1	1	-	-

z_1



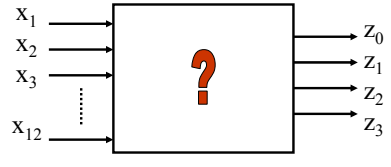
$z_1 = x_1 + x_2$

z_0



$z_0 = x_0 + x_2$

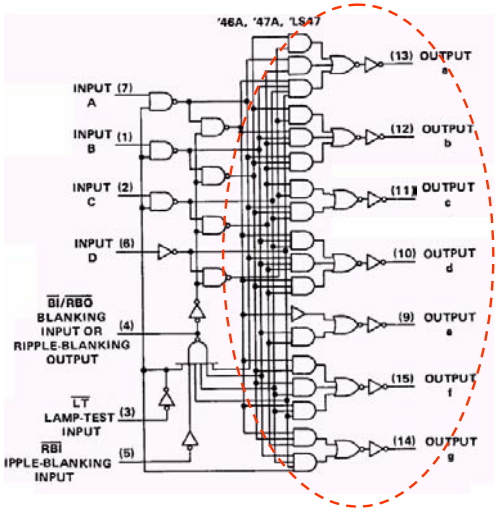
Encoder 12:4



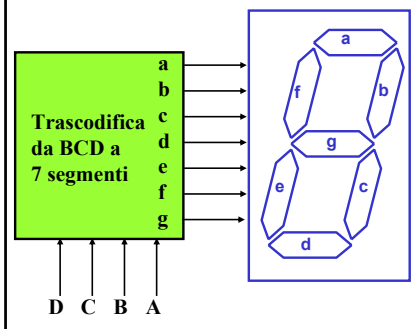
$z_0 = x_1 + x_3 + x_5 + x_7 + x_9 + x_{11}$
$z_1 = x_2 + x_3 + x_6 + x_7 + x_{10} + x_{11}$
$z_2 = x_4 + x_5 + x_6 + x_7 + x_{12}$
$z_3 = x_8 + x_9 + x_{10} + x_{11} + x_{12}$

Il circuito di trascodifica BCD-7 segmenti

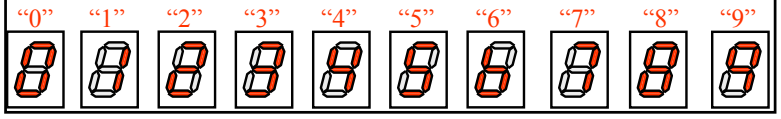
SN 7446



Sintesi

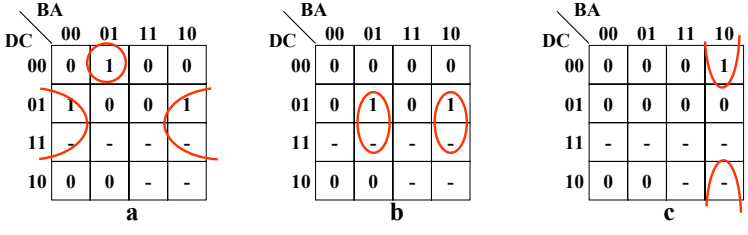


D	C	B	A	a	b	c	d	e	f	g
0	0	0	0	0	0	0	0	0	0	1
0	0	0	1	1	0	0	1	1	1	1
0	0	1	0	0	0	1	0	0	1	0
0	0	1	1	0	0	0	0	1	1	0
0	1	0	0	1	0	0	1	1	0	0
0	1	0	1	0	1	0	0	1	0	0
0	1	1	0	1	1	0	0	0	0	0
0	1	1	1	0	0	0	1	1	1	1
1	0	0	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	1	1	0	0



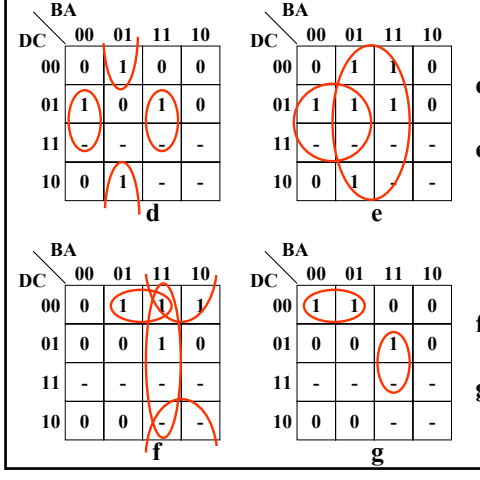
Progetto della rete di costo minimo (1)

N.B. - eventuali alee statiche non disturbano la visione!



a = D'C'B'A + CA' **b = CB'A + CBA'** **c = C'BA'**

Progetto della rete di costo minimo (2)



d = CB'A' + C'B'A + CBA

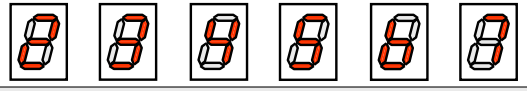
e = A + CB'

f = D'C'A + BA + C'B

g = D'C'B' + CBA

Risposta della rete di costo minimo a configurazioni non previste dal codice BCD

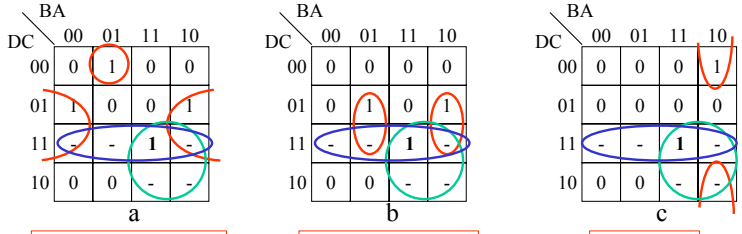
DCBA	1010	1011	1100	1101	1110	1111
a	0	0	1	0	1	0
b	0	0	0	1	1	0
c	1	0	0	0	0	0
d	0	0	1	0	0	1
e	0	1	1	1	0	1
f	1	1	0	0	0	1
g	0	0	0	0	0	1



☹ la rete di costo minimo non consente la rilevazione di alcuna configurazione di ingresso "illecita"

Progetto della rete in grado di rilevare le configurazioni di ingresso illecite (1)

Alle configurazioni illecite devono corrispondere sul display simboli diversi da quelli previsti per le configurazioni lecite; in particolare il display deve essere spento per la configurazione DCBA = 1111. Quest'ultima specifica richiede di ri-sintetizzare solo le funzioni a, b, c.



$a = D'C'B'A + CA'$ $b = CB'A + CBA'$ $c = C'BA'$

$a_1 = a + DC$ $a_2 = a + DB$ $b_1 = b + DC$ $b_2 = b + DB$ $c_1 = c + DC$ $c_2 = c + DB$

Progetto della rete in grado di rilevare le configurazioni di ingresso illecite (2)

In dipendenza delle espressioni selezionate per le funzioni a, b, c, si ottengono così 8 reti (R_1, R_2, ..., R_8), tutte della medesima complessità, caratterizzate dallo stesso comportamento sia per le configurazioni di ingresso previste dal codice BCD, sia per la configurazione DCBA = 1111 (display spento).

$R_1 : a_1\ b_1\ c_1\ d\ e\ f\ g$ $R_5 : a_2\ b_1\ c_1\ d\ e\ f\ g$

$R_2 : a_1\ b_1\ c_2\ d\ e\ f\ g$ $R_6 : a_2\ b_1\ c_2\ d\ e\ f\ g$

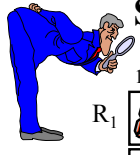
$R_3 : a_1\ b_2\ c_1\ d\ e\ f\ g$ $R_7 : a_2\ b_2\ c_1\ d\ e\ f\ g$

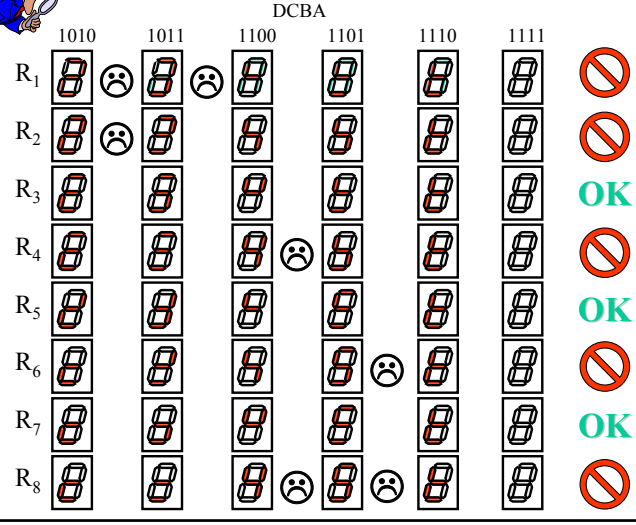
$R_4 : a_1\ b_2\ c_2\ d\ e\ f\ g$ $R_8 : a_2\ b_2\ c_2\ d\ e\ f\ g$

Il comportamento è lo stesso anche per la configurazione DCBA = 1110, dal momento che i segmenti a, b, c sono comunque spenti, qualunque siano le espressioni selezionate.

Il simbolo corrispondentemente visualizzato è: 

Scelta della rete "ottimale" (1)





Scelta della rete "ottimale" (2)

Le soluzioni R_1, R_2, R_4, R_6 e R_8 vanno scartate, in quanto non consentono la rilevazione di tutte le configurazioni di ingresso illecite.

Le soluzioni R_3, R_5 e R_7 vanno bene, in quanto consentono la rilevazione di tutte le configurazioni di ingresso illecite, peraltro con simboli tutti diversi fra loro.

Quale scegliere allora, visto che hanno tutte la medesima complessità e velocità di elaborazione ???

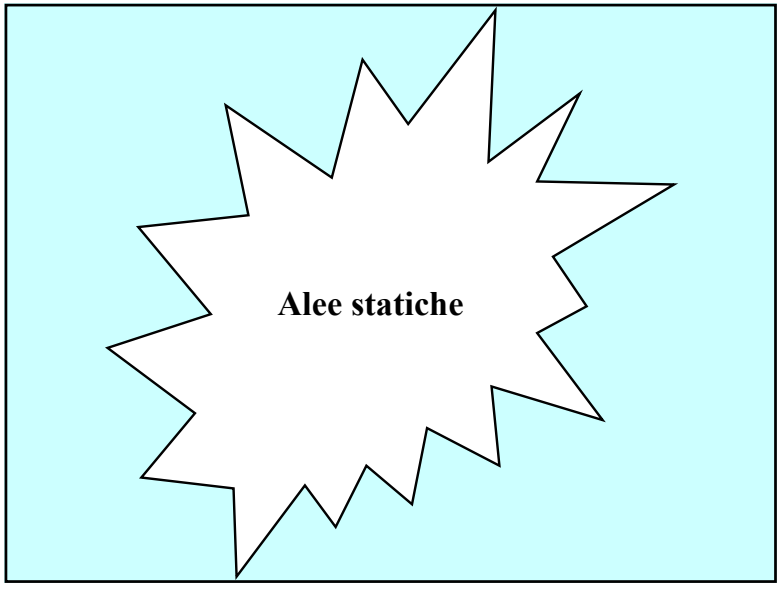
La soluzione R_7, che richiede un minore consumo di energia (17 segmenti globalmente accesi, anziché 18) per visualizzare le configurazioni illecite. Questa è la soluzione adottata nei circuiti integrati SN 7446A, 7447A !!!

Esercitazione N.8

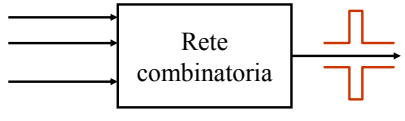
1. Tracciare sulla mappa la funzione di 4 variabili

   $$F = \Sigma_4\, m(1, 3, 5, 7, 8, 9, 12, 13)$$

2. Individuare graficamente tutti gli implicanti e gli implicati primi.
3. Scrivere le espressioni corrispondenti.
4. Evidenziare quali dei RR tracciati in 2. non sono essenziali per la copertura.
5. Scrivere le espressioni minime SP e PS della F.



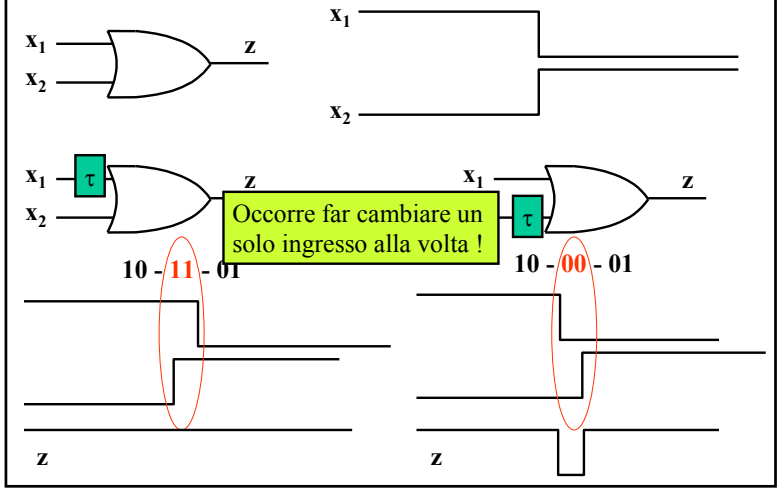
Alea statica



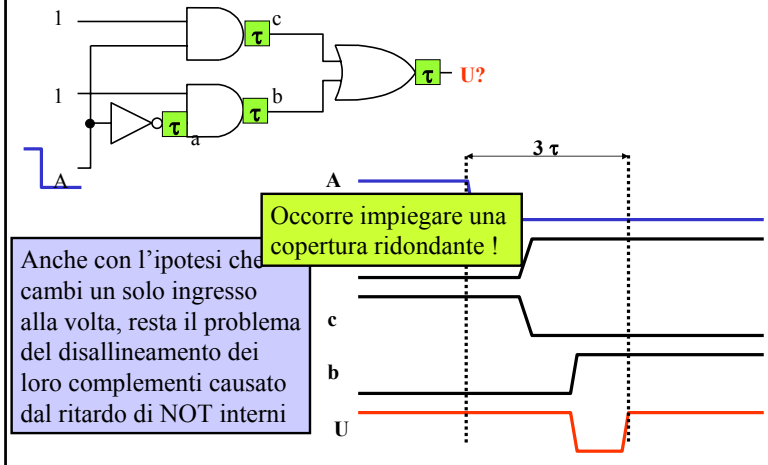
Alea statica: i ritardi di propagazione possono determinare una temporanea variazione dell'uscita di una rete combinatoria, in risposta ad una sequenza di due ingressi per cui avrebbe invece dovuto rimanere costante .

Comportamento apparentemente aleatorio
inaccettabile su una retroazione diretta

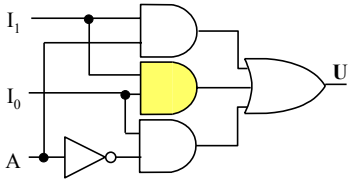
Alea statica nel OR



Alea statica nel SELETTORE



Il SELETTORE: la copertura "ridondante"

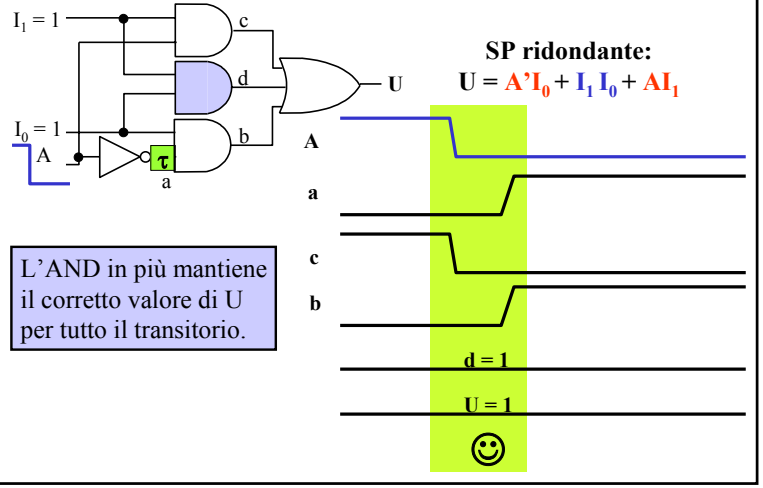


A \ I1 I0	00	01	11	10
0	0	1	1	0
1	0	0	1	1

SP ridondante:
$U = A'I_0 + I_1 I_0 + AI_1$

SP minima:
$U = A'I_0 + AI_1$

SELETTORE: rete non minima



Eliminazione a priori delle alee statiche

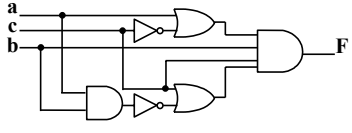
T14) Una r. c. completamente specificata ed i cui segnali d'ingresso cambiano di valore uno solo alla volta non presenta alea statica se è descritta da un'espressione normale formata da tutti i termini primi

Per eliminare a priori le alee statiche in una rete combinatoria incompletamente specificata è necessario e sufficiente scegliere una copertura in cui ogni coppia di 1 (o di 0) contenuta in celle adiacenti sia racchiusa in almeno un RR.

Analisi con le mappe

Uso delle mappe in sede di analisi (1)

1) Si scrive l'espressione associata allo schema e la si manipola fino ad ottenere una espressione normale:



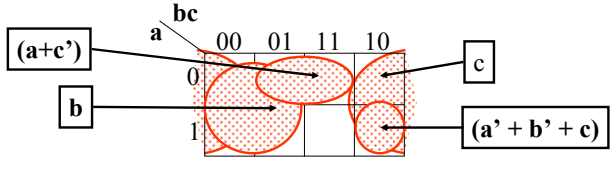
$F=(a+c')\cdot b\cdot c\cdot((a\cdot b)'+c)$

E10

$F=(a+c')\cdot b\cdot c\cdot(a'+b'+c)$

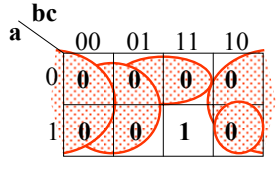
normale PS

2) Si predispone una mappa di dimensioni adeguate e si tracciano sulla mappa i RR che corrispondono ai termini :



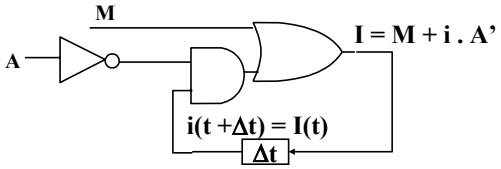
Uso delle mappe in sede di analisi (2)

3) Nelle celle coperte da un RR si indica il valore 1 se l'espressione normale è SP, 0 se è PS; nelle celle non coperte da RR si inserisce 0 nel caso SP, 1 nel caso PS:



N.B. - La valutazione di una espressione individua sempre una funzione completa !

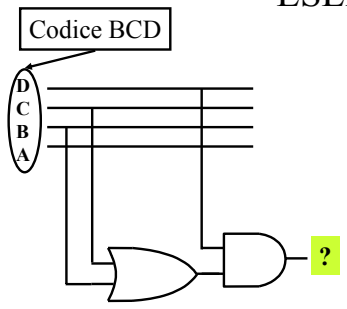
Valutazione dello stato futuro del relè con autoritenuta



i \ M A	00	01	11	10
0	0	0	1	1
1	1	0	1	1

$$I = M + i \cdot A'$$

ESERCIZIO



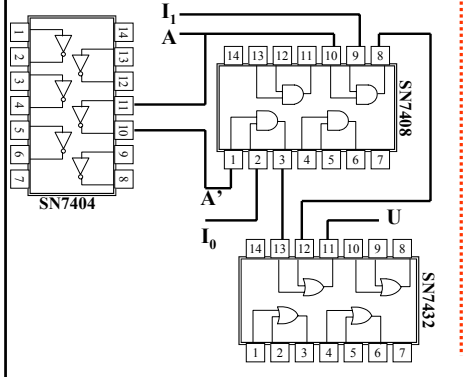
D.(C+B)

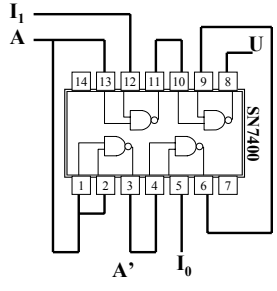
DC \ BA	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	1	1	1	1
10	0	0	1	1

Configurazioni errate per il codice BCD

NAND e NOR

Realizzazione con NAND SSI di un selettore a due vie





$$U = A' \cdot I_0 + A \cdot I_1$$

$$U = ((A \uparrow A) \uparrow I_0) \uparrow (A \uparrow I_1)$$

Sintesi con NAND

La sintesi "a NAND" può essere effettuata trasformando un'espressione normale SP che descrive la funzione assegnata in una nuova espressione contenente esclusivamente operatori "↑":

$F = a \cdot b + c' \cdot d + e \cdot f' + g$

↓ definizione dell'operatore ↑

$F = (a \uparrow b)' + (c' \uparrow d)' + (e \uparrow f')' + g$

↓ E10 (II^a legge di De Morgan)

$F = \big((a \uparrow b) \cdot (c' \uparrow d) \cdot (e \uparrow f') \cdot g'\big)'$

↓ definizione dell'operatore ↑

$F = (a \uparrow b) \uparrow (c' \uparrow d) \uparrow (e \uparrow f') \uparrow g'$

N.B. : stesso numero di operatori!

Algoritmo per la sintesi a NAND

1) Si parte da un'espressione SP, SPS, SPSP... e si introducono gli operatori "." e le parentesi non indicati esplicitamente.

2) Si sostituisce il simbolo "↑" ad ogni simbolo "."

3) Si sostituisce il simbolo "↑" ad ogni simbolo "+" e si complementano le variabili e le costanti affiancate a tale simbolo senza l'interposizione di una parentesi.

4) Si disegna lo schema logico corrispondente all'espressione trovata. Se l'espressione di partenza è a più di due livelli si cerca l'eventuale presenza di NAND con ingressi identici e li si sostituisce con uno solo (sfruttando il fan-out >1 del gate corrispondente).

N.B. - La trasformazione dell'espressione minima SP individua l'espressione minima a NAND.

Esempio: sintesi a NAND di un EX-OR

$U = a\,b' + a'b$

$U = a\,b' + a'b + a'a + b'b$

$U = a\,(a' + b') + b\,(a' + b')$ ***SPS !***

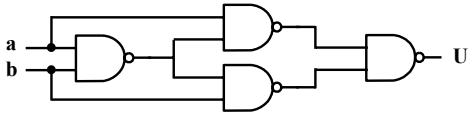
↓ *passo 1*

$U = (a \cdot (a' + b')) + (b \cdot (a' + b'))$

↓ *passi 2 e 3*

$U = (a \uparrow (a \uparrow b)) \uparrow (b \uparrow (a \uparrow b))$

↓ *passo 4*



Sintesi con NOR

La sintesi "a NOR" può essere effettuata trasformando un'espressione normale PS che descrive la funzione assegnata in una nuova espressione contenente esclusivamente operatori "↓":

$F = (a' + b' + c) \cdot (d' + e) \cdot f' \cdot g$

↓ definizione dell'operatore ↓

$F = (a' \downarrow b' \downarrow c)' \cdot (d' \downarrow e)' \cdot f' \cdot g$

↓ E10 (I^a legge di De Morgan)

$F = \big((a' \downarrow b' \downarrow c) + (d' \downarrow e) + f + g'\big)'$

↓ definizione dell'operatore ↓

$F = (a' \downarrow b' \downarrow c) \downarrow (d' \downarrow e) \downarrow f \downarrow g'$

Algoritmo per la sintesi a NOR

1) Si parte da un'espressione PS, PSP, PSPS... e si introducono gli operatori "·" e le parentesi non indicati esplicitamente.

2) Si sostituisce il simbolo "↓" ad ogni simbolo "+"

3) Si sostituisce il simbolo "↓" ad ogni simbolo "·" e si complementano le variabili e le costanti affiancate a tale simbolo senza l'interposizione di una parentesi.

4) Si disegna lo schema logico corrispondente all'espressione trovata. Se l'espressione di partenza è a più di due livelli si cerca l'eventuale presenza di NOR con ingressi identici e li si sostituisce con uno solo (sfruttando il fan-out >1 del gate corrispondente).

N.B. - La trasformazione dell'espressione minima PS individua l'espressione minima a NOR.

Esempio: sintesi a NOR di un "equivalence"

$U = (a + b') \cdot (a' + b)$

$U = (a + b') \cdot (a' + b) \cdot (a' + a) \cdot (b' + b)$

$U = (a + a'b') \cdot (b + a'b')$ ***PSP !***

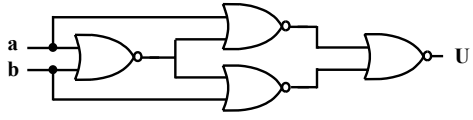
passo 1

$U = (a + (a' \cdot b')) + (b + (a' \cdot b'))$

passi 2 e 3

$U = (a \downarrow (a \downarrow b)) \downarrow (b \downarrow (a \downarrow b))$

passo 4



Esercitazione N. 9

DOMANDA N. 1 – Selettore con NOR a 2 ingressi

DOMANDA N. 2

AND a tre ingressi
- con NAND a 2 ingressi
- con NAND a 2 ingressi
- con NAND & NOR a 2 ingressi

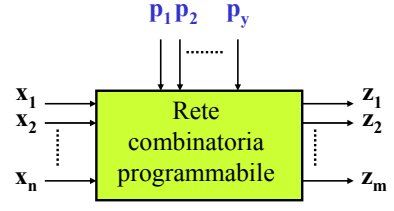
OR a tre ingressi
- con NAND a 2 ingressi
- con NOR a 2 ingressi
- con NAND & NOR a 2 ingressi

5.4
Reti programmabili

La programmazione del hardware

Le reti combinatorie programmabili

Rete combinatoria programmabile - Rete combinatoria in grado di presentare diverse relazioni ingresso/uscita singolarmente selezionabili mediante l'attribuzione di una determinata configurazione di valori ad un gruppo di segnali detti bit di programmazione.



$$z_i = F_i(p_1,p_2,\ldots,p_y,x_1,x_2,\ldots,x_n) = F_P(x_1,x_2,\ldots,x_n)$$

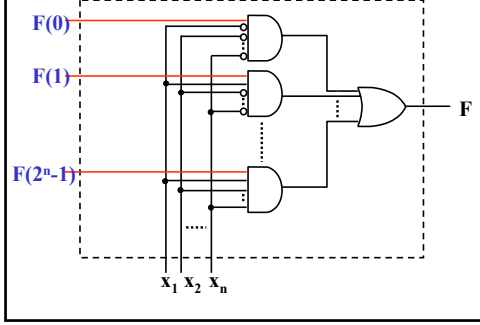
Memorie a sola lettura

Il MUX come rete programmabile

$$F(x_1,x_2,\ldots x_i,\ldots x_n) = \sum_{i=0}^{2^n-1} m(i)\cdot F(i)$$

Espressione generale SP

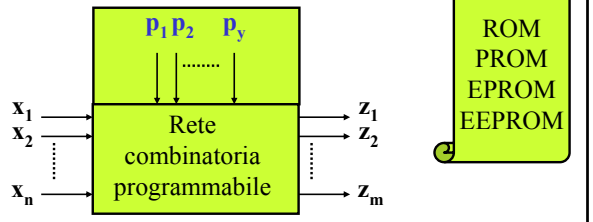
F(i): bit di programmazione → MUX



Al crescere di n cresce esponenzialmente il n° dei pin da utilizzare per la programmazione.
I MUX disponibili ne hanno al più 16:
16 + 4 + 2 +2 = 24

Read Only Memory

Memoria a sola lettura - Dispositivo integrato contenente
- **la realizzazione di una o più espressioni generali SP**
- **i relativi bit di programmazione.**

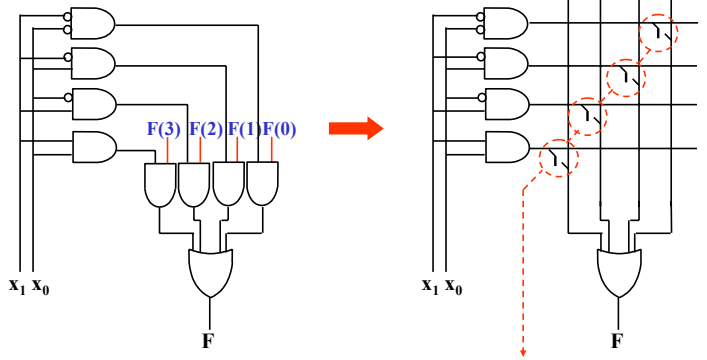


ROM
PROM
EPROM
EEPROM

Esempio*: *n= 24 piedini
4 tensioni di alimentazione
16 bit di indirizzo
4 funzioni di 16 variabili!
$4 \times 2^{16} = 262.204$ ***segnali interni di programmazione***

Struttura di una ROM (1)

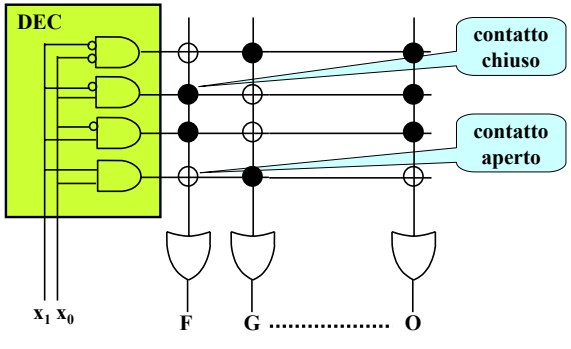
Una diversa realizzazione del MUX (proprietà associativa **)** → **I contatti al posto dei segnali di programmazione**



F(i)=0/1 → Contatto aperto/chiuso

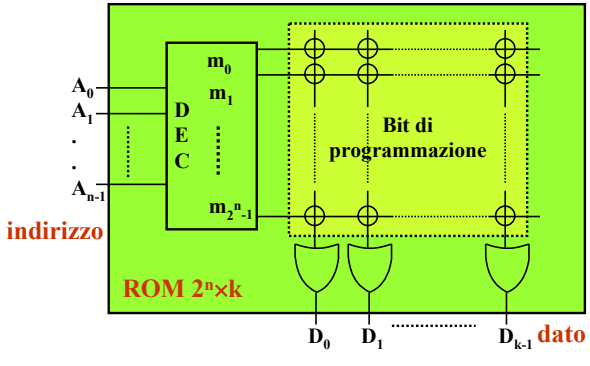
Struttura di una ROM (2)

- **Rappresentazione "compatta" della struttura di una ROM :**



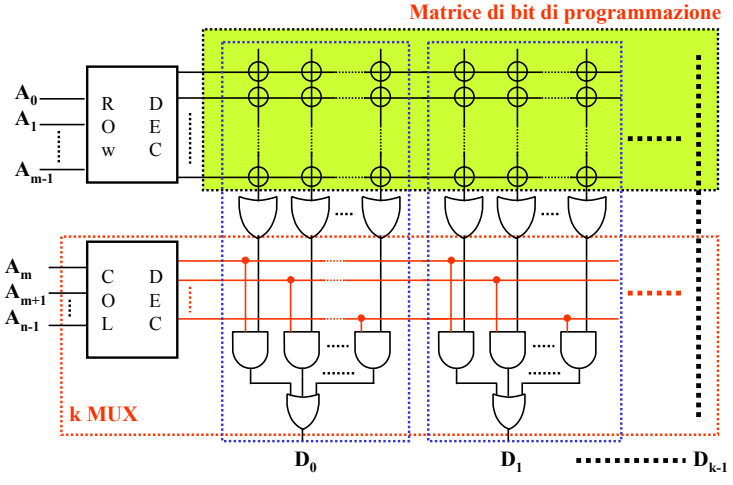
***fan-out* del DEC: realizzazione integrata di più funzioni**

Le ROM come circuiti di memoria



Ogni configurazione delle variabili di ingresso può essere vista come l'indirizzo di un dato formato dai bit che sono stati programmati nella riga corrispondente della matrice.

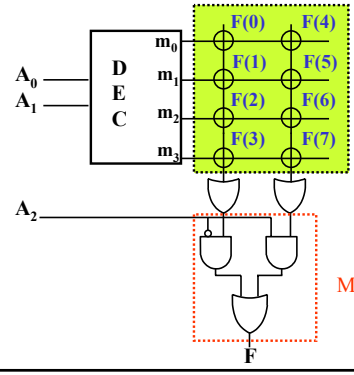
Selezione a due dimensioni



Caso di studio

$F(A_2,A_1,A_0) = A_2'A_1'A_0'\,F(0) + A_2'A_1'A_0\,F(1) + A_2'A_1A_0'\,F(2) + A_2'A_1A_0\,F(3) + A_2A_1'A_0'\,F(4) + A_2A_1'A_0\,F(5) + A_2A_1A_0'\,F(6) + A_2A_1A_0\,F(7)$

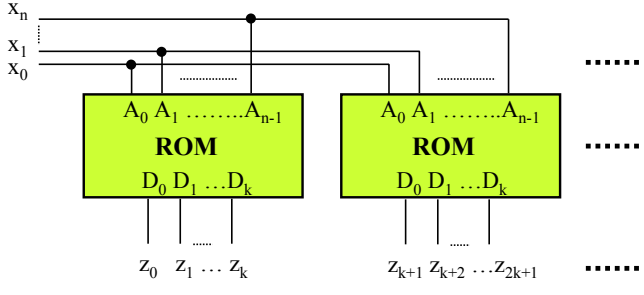
$= A_2'(A_1'A_0'\,F(0) + A_1'A_0\,F(1) + A_1A_0'\,F(2) + A_1A_0\,F(3)) + A_2\,(A_1'A_0'\,F(4) + A_1'A_0\,F(5) + A_1A_0'\,F(6) + A_1A_0\,F(7))$



Bit di programmazione

Selezione lineare:
8 AND a 3 ingressi
Selez. a due dimensioni:
6 AND a 2 ingressi

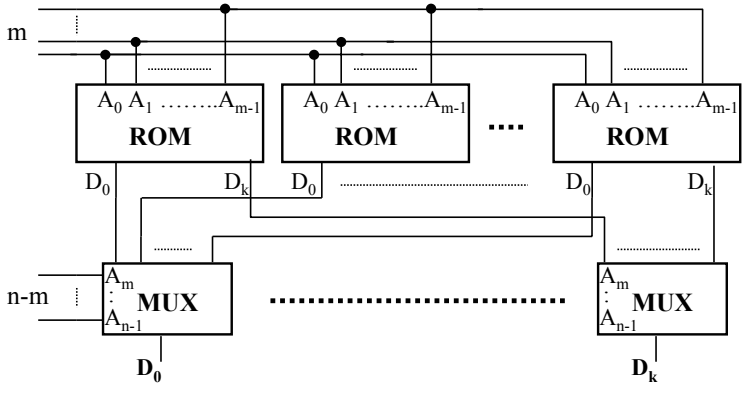
Estensione del numero di uscite



Collegamento in parallelo di più ROM

Estensione del numero di ingressi

Selezione a 2 dimensioni



N.B. - Al primo livello occorrono $2^{(n-m)}$ ROM

ROM, PROM, EPROM, E²PROM

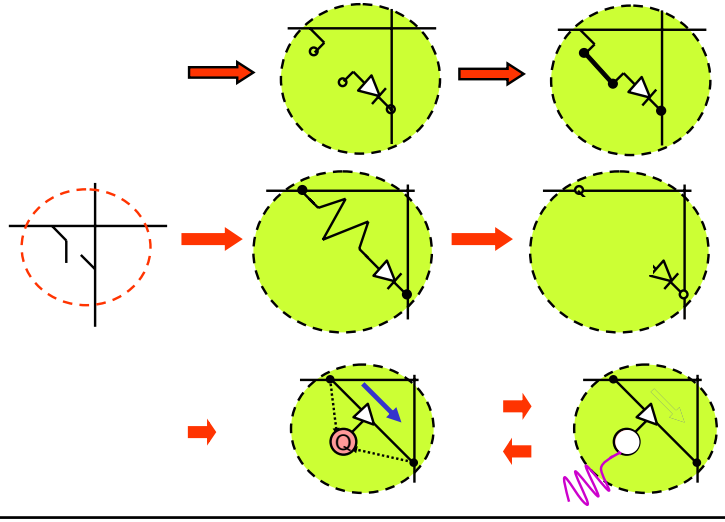
Memorie non volatili a sola lettura

Memorizzazione di informazioni che devono permanere quando il sistema non è alimentato e che non non cambiano durante il funzionamento .

Esempi: trascodifica bit map di caratteri ASCII, BIOS del PC, smart card

Tipo	Proprietà programmazione	uso
ROM	una volta	n° di copie grande
PROM	una volta	n° di copie piccolo
EPROM	più volte	prototipi
EEPROM	più volte	personalizzazione

ROM, PROM, EPROM



Memorie a sola lettura cancellabili elettricamente

EEPROM (*Electrically Erasable* PROM) : si programmano e cancellano byte-per-byte tramite segnali elettrici e senza rimuovere il dispositivo dalla piastra stampata.

FLASH-EPROM: si programmano/cancellano elettricamente direttamente sulla piastra. La cancellazione è più veloce rispetto alle EEPROM: con un'unica operazione è possibile cancellare l'intero dispositivo oppure uno o più "settori".

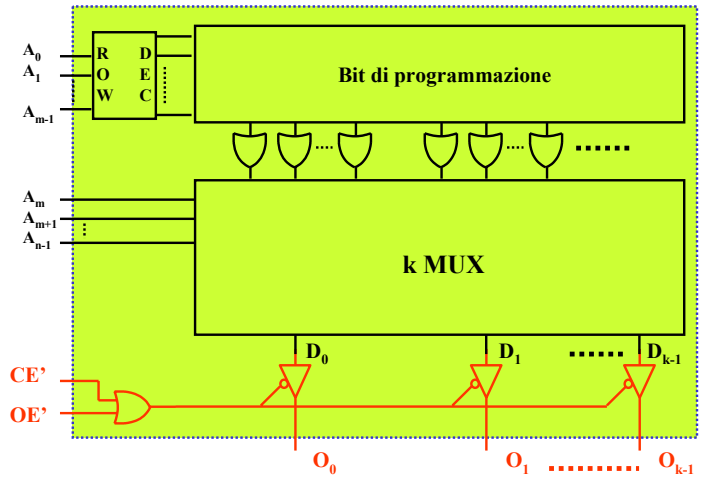
TMS29F010
131072 BY 8-BIT
FLASH MEMORY
SMJS840A – NOVEMBER 1997 – REVISED JUNE 1998

memory sector architecture

	A16	A15	A14	Address Range
Sector 0	0	0	0	00000h – 03FFFh
Sector 1	0	0	1	04000h – 07FFFh
Sector 2	0	1	0	08000h – 0BFFFh
Sector 3	0	1	1	0C000h – 0FFFFh
Sector 4	1	0	0	10000h – 13FFFh
Sector 5	1	0	1	14000h – 17FFFh
Sector 6	1	1	0	18000h – 1BFFFh
Sector 7	1	1	1	1C000h – 1FFFFh

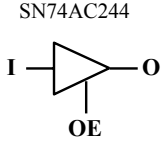
Amplificatori a tre stati (d'uscita)

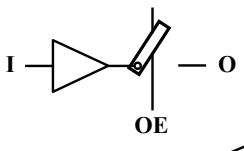
Stadio di uscita di una ROM



Amplificatore a 3 stati d'uscita

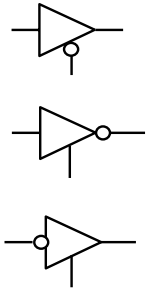
SN74AC244

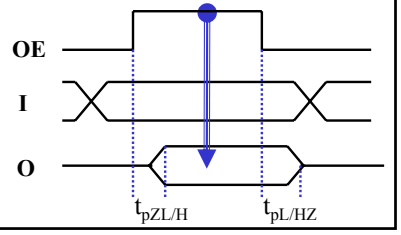




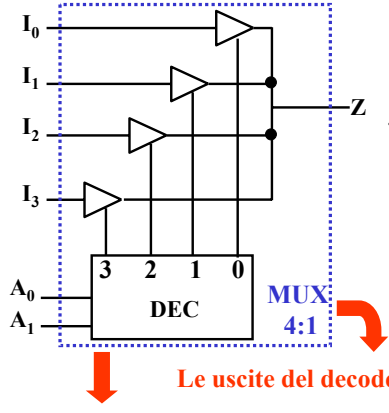
OE	I	O
H	L	L
H	H	H
L	X	Z

Lo stato elettrico del segnale d'uscita è indefinito o fluttuante (Terzo Stato, Stato di Alta Imp.)





MUX con amplificatori 3-state (1)



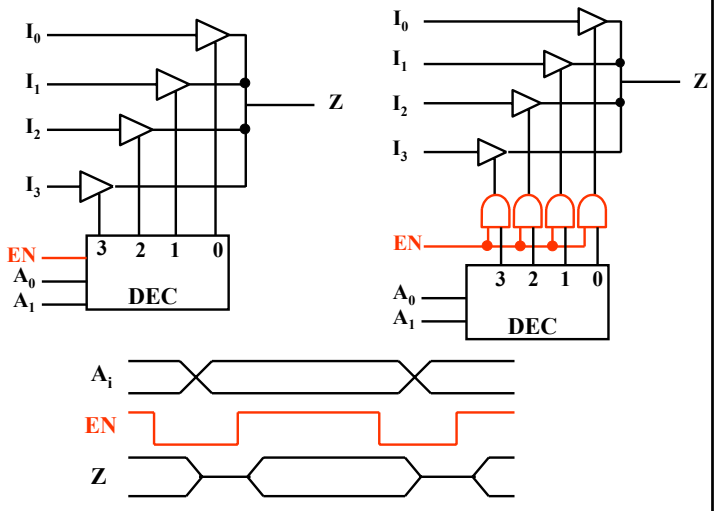
In ogni istante di tempo non deve esserci più di un 3-state abilitato !

Situazione di corto circuito (*conflitto elettrico*) con possibili malfunzionamenti del sistema

Le uscite del decoder non variano simultaneamente

Nei 3-state il tempo di risposta all'abilitazione (t_{pZH}, t_{pZL}) è inferiore a quello necessario per il passaggio nel terzo stato (t_{pHZ}, t_{pLZ})

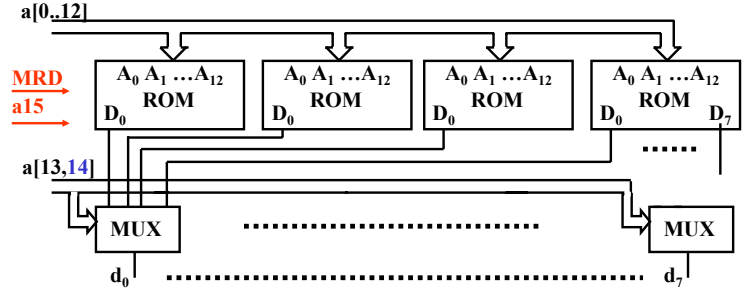
MUX con amplificatori 3-state (2)



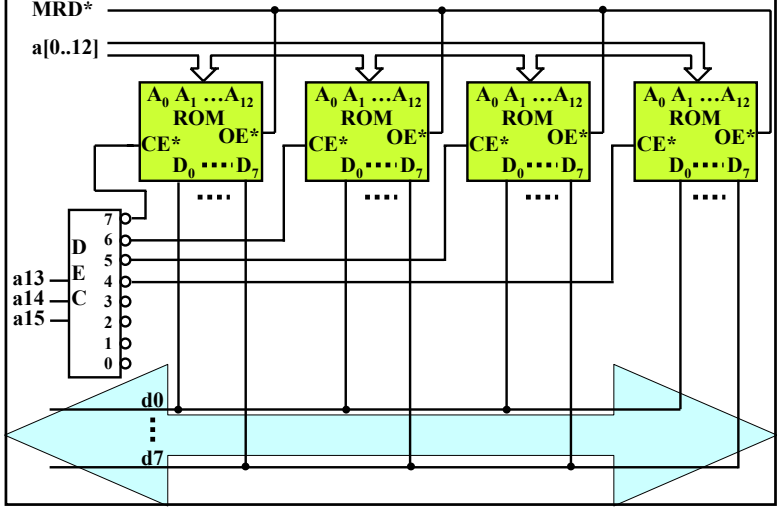
Progetto di un banco di ROM (1)

Supponiamo di voler connettere 32K byte di ROM ad una CPU con 16 bit di indirizzo ($A_0..A_{15}$) ed 8 bit di dato e di avere a disposizione dispositivi ROM da 8K x 8 . Supponiamo inoltre che la CPU veda il banco di ROM nella parte alta del suo spazio di indirizzamento (8000H - FFFFH, cioè A_{15}=1).

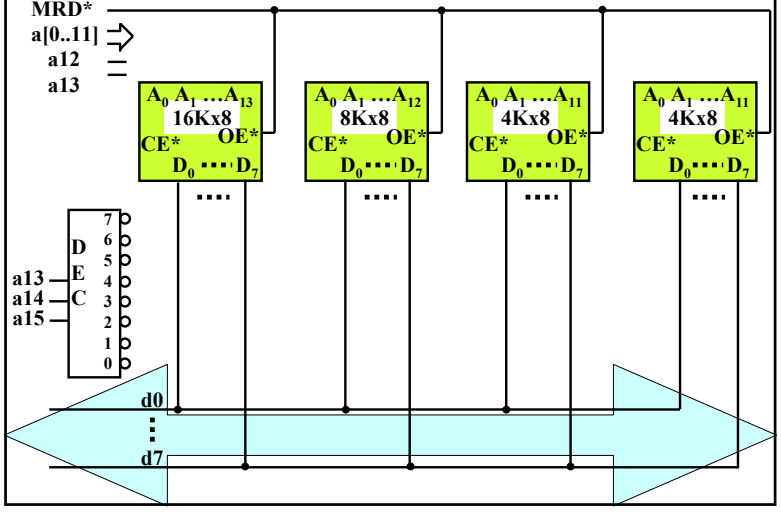
Estensione del numero degli ingressi



Progetto di un banco di ROM (1)

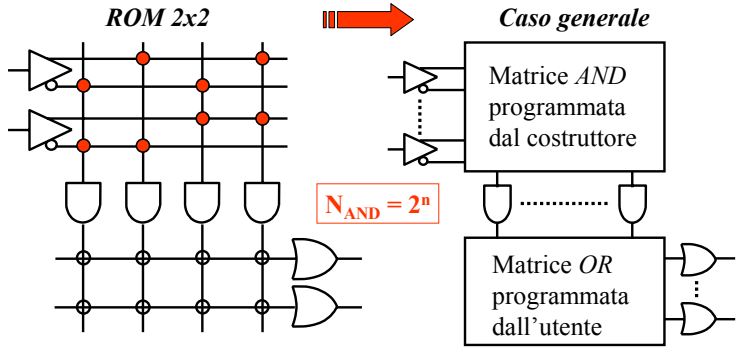


Esercitazione N. 10



PLA e PAL

Rappresentazione di una ROM in termini di matrici AND e OR

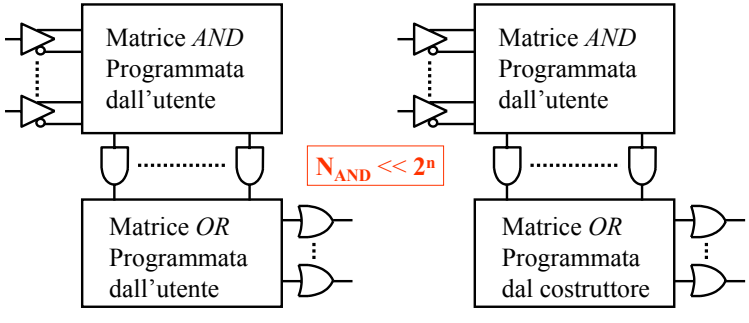


N.B. - Ogni AND realizza un potenziale mintermine e può essere impiegato per la programmazione di ciascuna uscita (*espressione generale SP*).

PLA e PAL

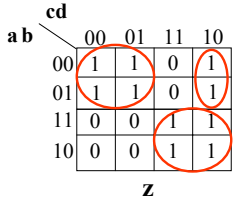
PLA: Programmable Logic Array

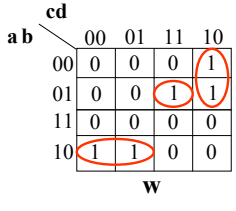
PAL: Programmable Array Logic

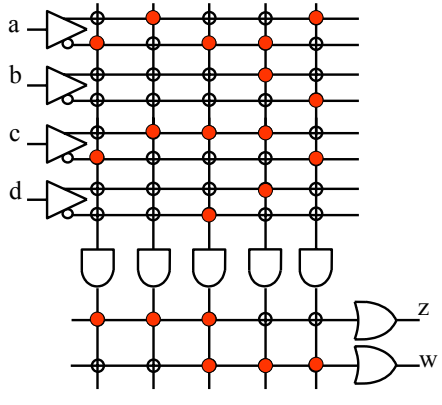


N.B. - Ogni AND realizza un implicante (*espressione normale SP*).
Nelle PLA gli implicanti possono essere "comuni" a più uscite.

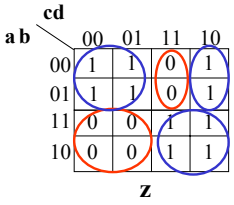
Sintesi con PLA

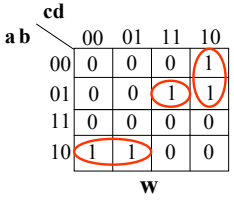


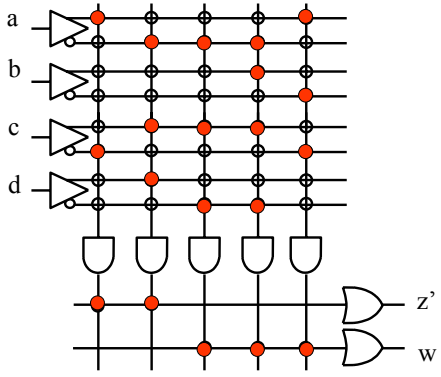




Sintesi con PAL

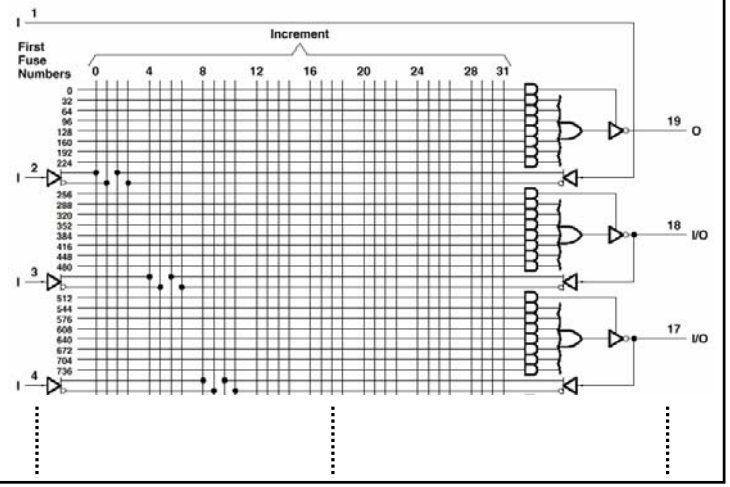






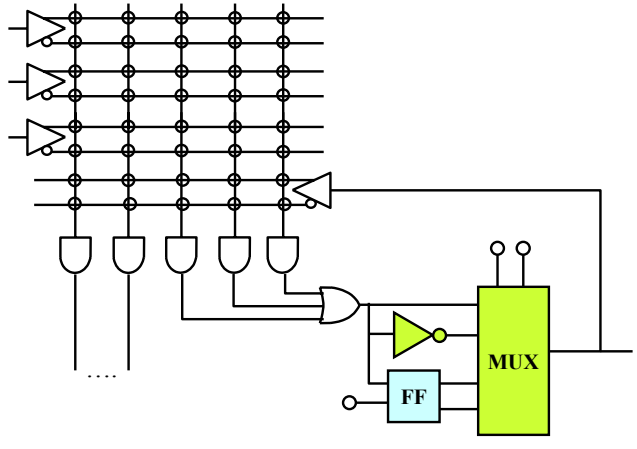
N.B. - Quando non si dispone di un numero sufficiente di AND può essere utile realizzare la funzione complemento.

PAL con I/O programmabile (16L8)

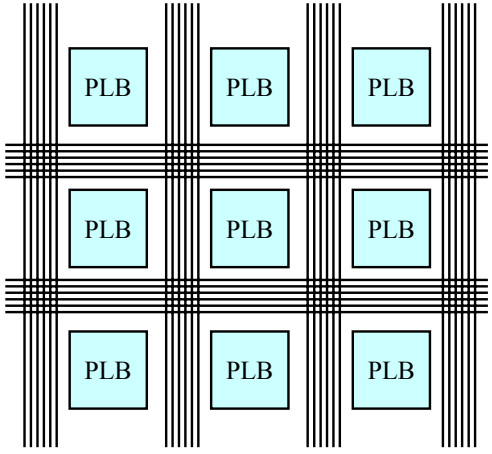


PLD, CPLD e FPGA

PLD



CPLD, FPGA



CAD (XILINX)

Get specification and define inputs and outputs

↓

Design entry via schematic, state-machine, HDL editor

↓

Functional simulation of the design

↓

Fit to CPLD or FPGA

↓

Timing simulation of the mapped design

↓

Download to the board

↓

Debug the design

Esercitazione N. 11