

Esercitazioni di Reti Logiche

Sintesi di Reti Combinatorie & Complementi sulle Reti Combinatorie

Zeynep KIZILTAN

Dipartimento di Scienze dell'Informazione

Universita' degli Studi di Bologna

Anno Accademico 2007/2008

Argomenti

- Procedura di sintesi dei circuiti combinatori.
- Porte NAND e NOR.
- Alcuni blocchi logici di base & loro uso nei circuiti:
 - Sommatore (Adder)
 - Decodificatore (Decoder)
 - Multiplexer
 - PLA (Array logica programmabile)

Procedura di sintesi

- L'obiettivo della sintesi è determinare il circuito che implementa un funzionamento.
- A partire dalla descrizione verbale, il processo di sintesi individua:
 - un diagramma logico;
 - opp. un insieme di equazioni booleane dalle quali il diagramma può essere ricavato.

Procedura di sintesi

Per eseguire la sintesi:

1. Identificare il numero di ingressi e uscite.
2. Costruire la tabella di verità, interpretando la specifica verbale.
3. Determinare le funzioni booleane delle uscite come funzione delle variabili di ingresso:
 - le funzione devono essere semplificate tramite le mappe di Karnaugh, sfruttando i primi implicant (essenziali), le condizioni di non-specificazione, ...
4. Disegnare il diagramma logico.

Esercitazione 1

- Progettare un circuito combinatorio per convertire un codice BCD in un codice excess-3.

Codice BCD

- Il sistema numerico binario e' il sistema piu' naturale per un computer, mentre il sistema decimale e' il piu' naturale per un uomo.
- Per permettere all'uomo e al computer di comunicare, occorre:
 - convertire i numeri decimali in binario;
 - eseguire i calcoli aritmetici in binario;
 - convertire il risultato da binario in decimale.
- Questo richiede che i numeri decimali vengano memorizzati per convertirli nel sistema binario.

Codice BCD

- Poiché i calcolatori possono gestire soltanto 0 e 1, bisogna codificare le cifre decimali soltanto con i simboli 0 e 1.
- Il codice binario decimale (BCD) è un sistema per codificare i numeri decimali:
 - associa ogni cifra decimale 0,1,2,..9 a un numero binario di 4 bit.
- Un numero con n cifre decimali codificato in BCD richiede $4n$ bit:
 - ciascun gruppo di quattro bit rappresenta una cifra decimale.
- $(185)_{10} = (0001\ 1000\ 0101)_{BCD} \neq (10111001)_2$



Binary-Coded Decimal (BCD)

Decimal Symbol	BCD Digit
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

Codice excess-3

- Cambiando il numero binario di 4 bit associato a ciascun decimale, altri sistemi di codificazione possono essere ottenuti.
- Excess-3 è un sistema per codificare i numeri decimali:
 - associa ogni cifra decimale al numero binario di BCD aumentato di 3 unità.
- $(185)_{10} = (0001\ 1000\ 0101)_{\text{BCD}} = (0100\ 1011\ 1000)_{\text{E-3}}$
 - $0100 = 0001 + 0011$
 - $1011 = 1000 + 0011$
 - $1000 = 0101 + 0011$

Convertitore di codice

1. Il numero di ingressi e uscite
 - tutti e due i codici utilizzano 4 bit.
 - ci sono 4 variabili A,B,C,D di ingresso e 4 variabili W,X,Y,Z di uscita.

Convertitore di codice

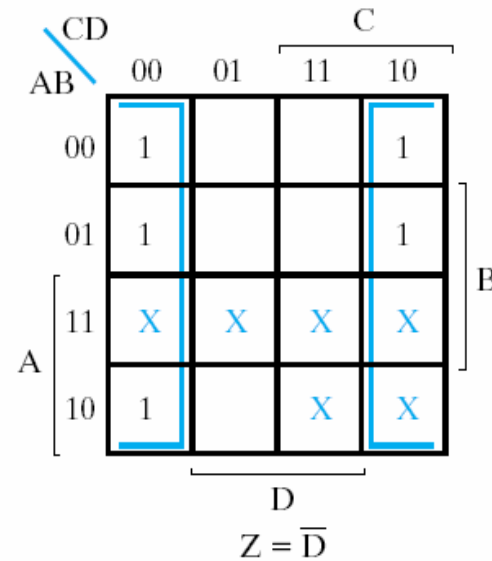
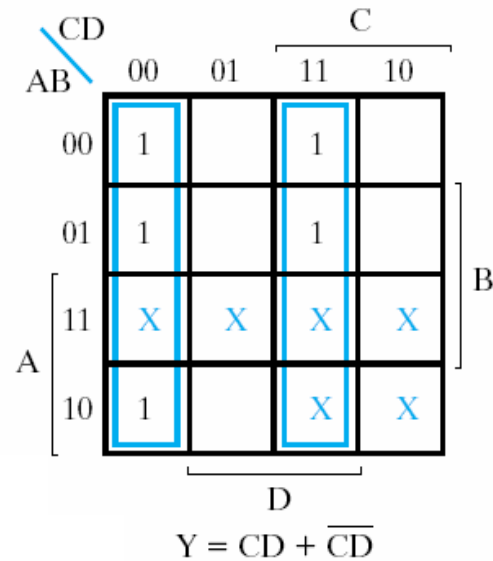
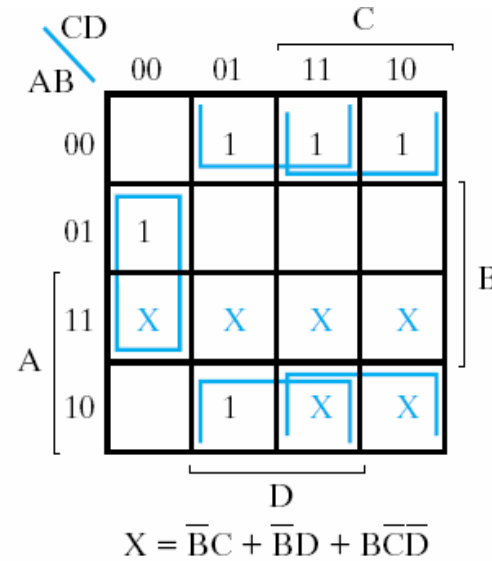
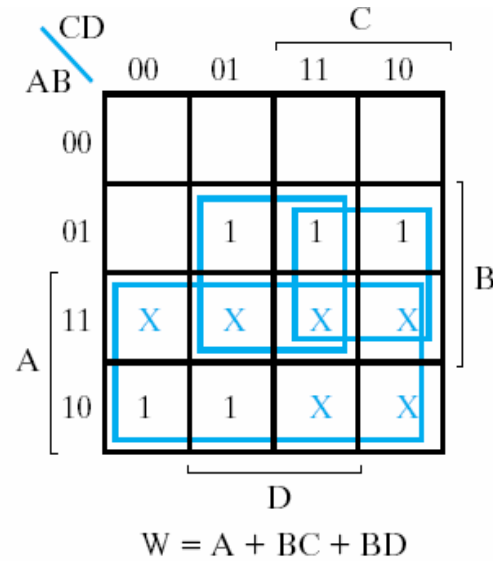
2. La tabella di verità

- Il codice E-3 è ottenuto aggiungendo $(0011)_2 = 3_{10}$ al codice BCD.
- solo 10 righe:
 - le altre 6 combinazioni dei valori non sono usate.
- esse possono essere trattate come condizioni di non-specificazione nelle mappe.

Decimal Digit	Input BCD				Output Excess-3			
	A	B	C	D	W	X	Y	Z
0	0	0	0	0	0	0	1	1
1	0	0	0	1	0	1	0	0
2	0	0	1	0	0	1	0	1
3	0	0	1	1	0	1	1	0
4	0	1	0	0	0	1	1	1
5	0	1	0	1	1	0	0	0
6	0	1	1	0	1	0	0	1
7	0	1	1	1	1	0	1	0
8	1	0	0	0	1	0	1	1
9	1	0	0	1	1	1	0	0

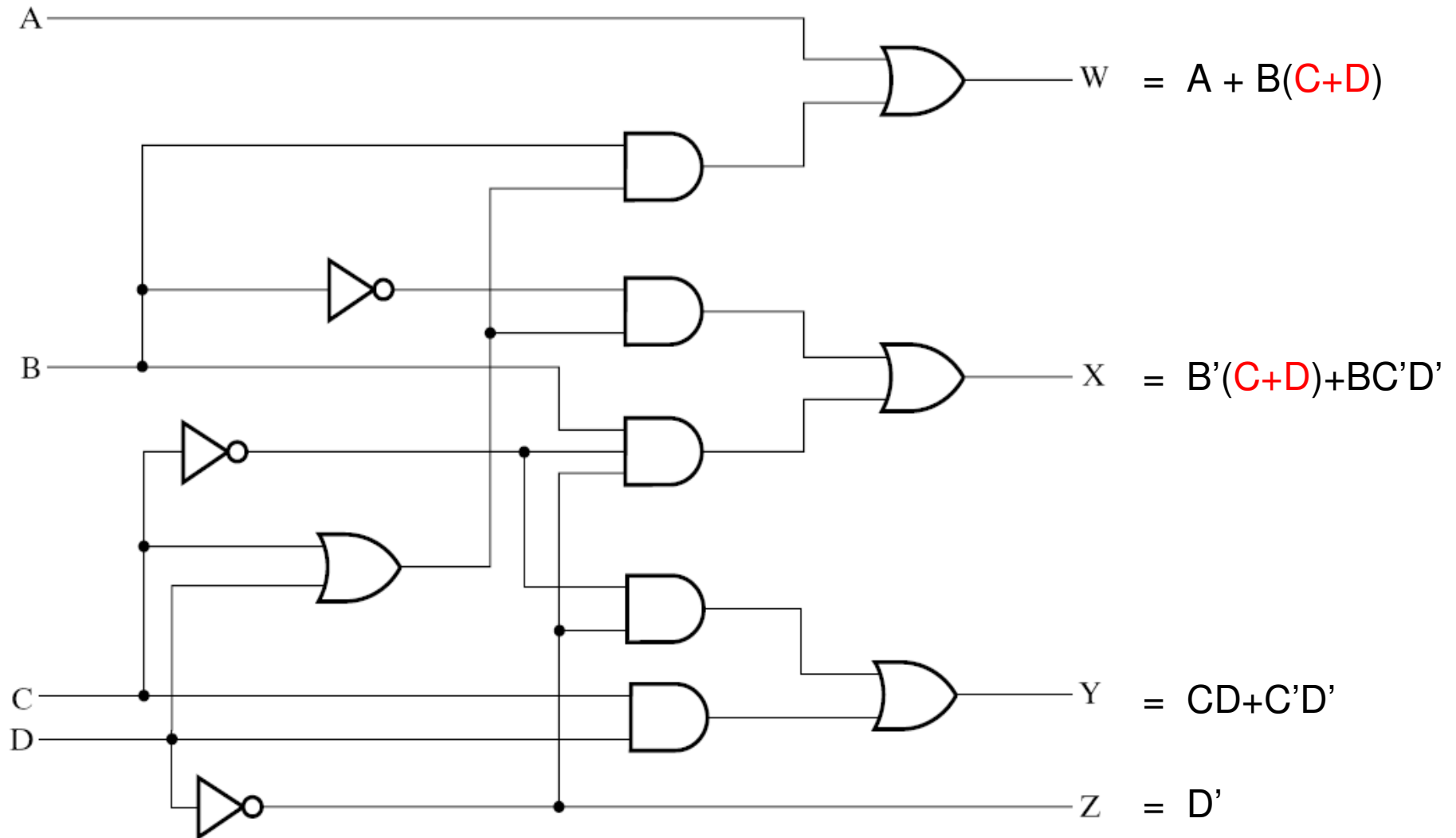
3. Funzioni booleane

- ciascuna mappa rappresenta una delle 4 uscite.



4. Il diagramma logico

- Nella implementazione, le espressioni possono essere manipolate algebricamente per condividere le porte.



Esercitazione 2

- Disegnare il diagramma logico per la funzione:

$$F = A + (B' + C)(D' + BE')$$

- utilizzando soltanto porte **NAND**,
- utilizzando soltanto porte **NOR**.

Come si realizza un circuito di soli NAND?

- Costruire una rete combinatoria per F tramite porte AND-OR.
- Convertire ogni porta AND in NAND con simbolo AND-NOT.
- Convertire ogni porta OR in NAND con simbolo NOT-OR.
- Per ogni pallino che non è bilanciato da un altro:
 - inserire una porta NOT (in forma di porta NAND);
 - oppure complementare il letterale in ingresso.

Come si realizza un circuito di soli NOR?

- Costruire una rete combinatoria per F tramite porte AND-OR.
- Convertire ogni porta OR in NOR con simbolo OR-NOT.
- Convertire ogni porta AND in NOR con simbolo NOT-AND.
- Per ogni pallino che non è bilanciato da un altro:
 - inserire una porta NOT (in forma di porta NOR);
 - oppure complementare il letterale in ingresso.

Esercitazione 3

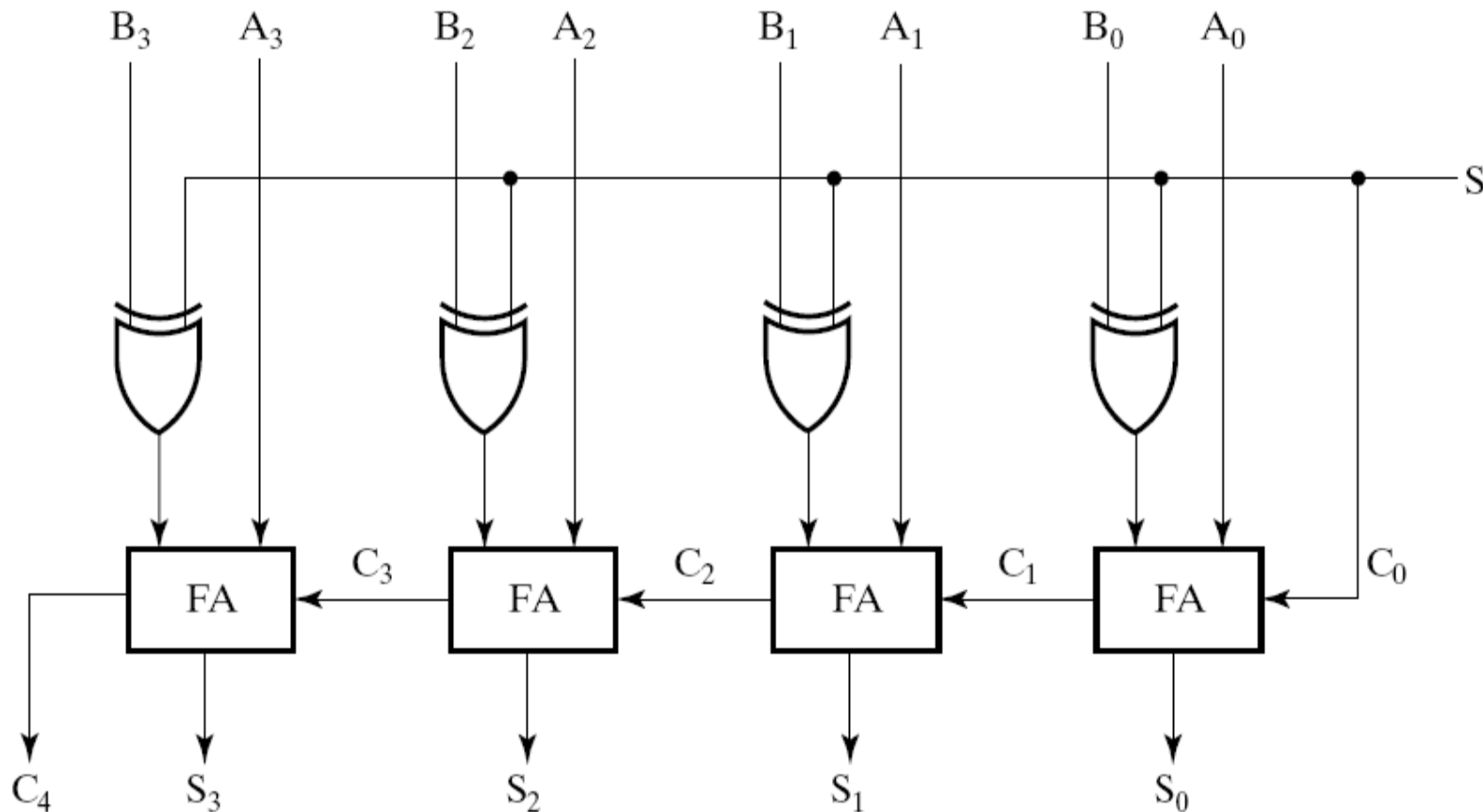
- Realizzare un circuito per la sottrazione tra 2 numeri binari di 4 bit (**senza segno**) utilizzando un sommatore (a propagazione di riporto) a 4 bit.

Sottrattore

- La sottrazione $A - B$ dei 2 numeri binari (senza segno) può essere eseguita con il sommare il complemento a 2 del sottraendo B al minuendo A .
- Il complemento a 2 può essere ottenuto sommando 1 al complemento a 1 del numero considerato.
→ $A - B = A + (\text{il complemento a 1 di } B) + 1$
! Il complemento a 1 di un numero può essere calcolato complementando, uno a uno, i bit del numero.

Sommatore/Sottrattore

- $B \oplus 1 = B'$ $B \oplus 0 = B$
- $S=0 \rightarrow A+B$
- $S=1 \rightarrow A + (\text{il complemento a 1 di } B) + 1$



Esercitazione 4

- Un circuito combinatorio è definito dalle tre funzioni booleane seguenti:

$$F_1(X, Y, Z) = X' Y' Z' + X Z$$

$$F_2(X, Y, Z) = X Y' Z' + X' Y$$

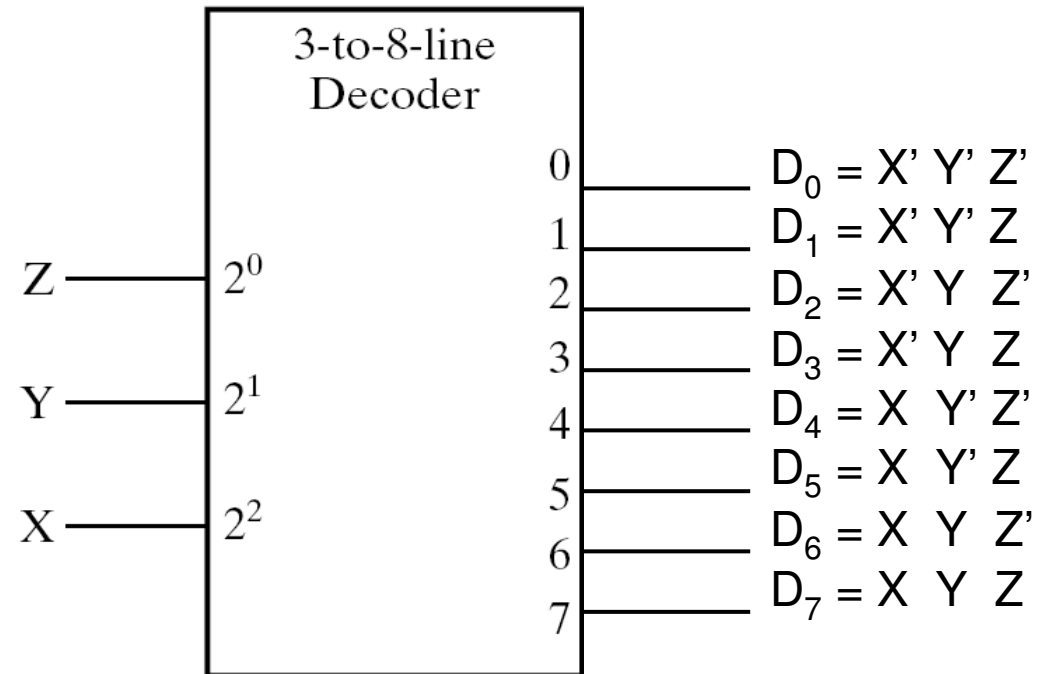
$$F_3(X, Y, Z) = X' Y' Z + X Y$$

Progettare il circuito utilizzando un decodificatore e porte esterne.

Decodificatore (Decoder)

- Un decoder e' un circuito combinatorio che genera 2^n mintermini di n variabili.
- Quindi e' molto utile nella progettazione dei circuiti:
 - qualunque funzione booleana è esprimibile come somma di mintermini.

Esercitazione 4



$$F_1(X,Y,Z) = X' Y' Z' + X Z$$

$$= \sum m(0,5,7)$$

		Y			
		00	01	11	10
X	0	1	0	0	0
	1	0	1	1	0
		Z			

$$F_2(X,Y,Z) = X Y' Z' + X' Y$$

$$= \sum m(2,3,4)$$

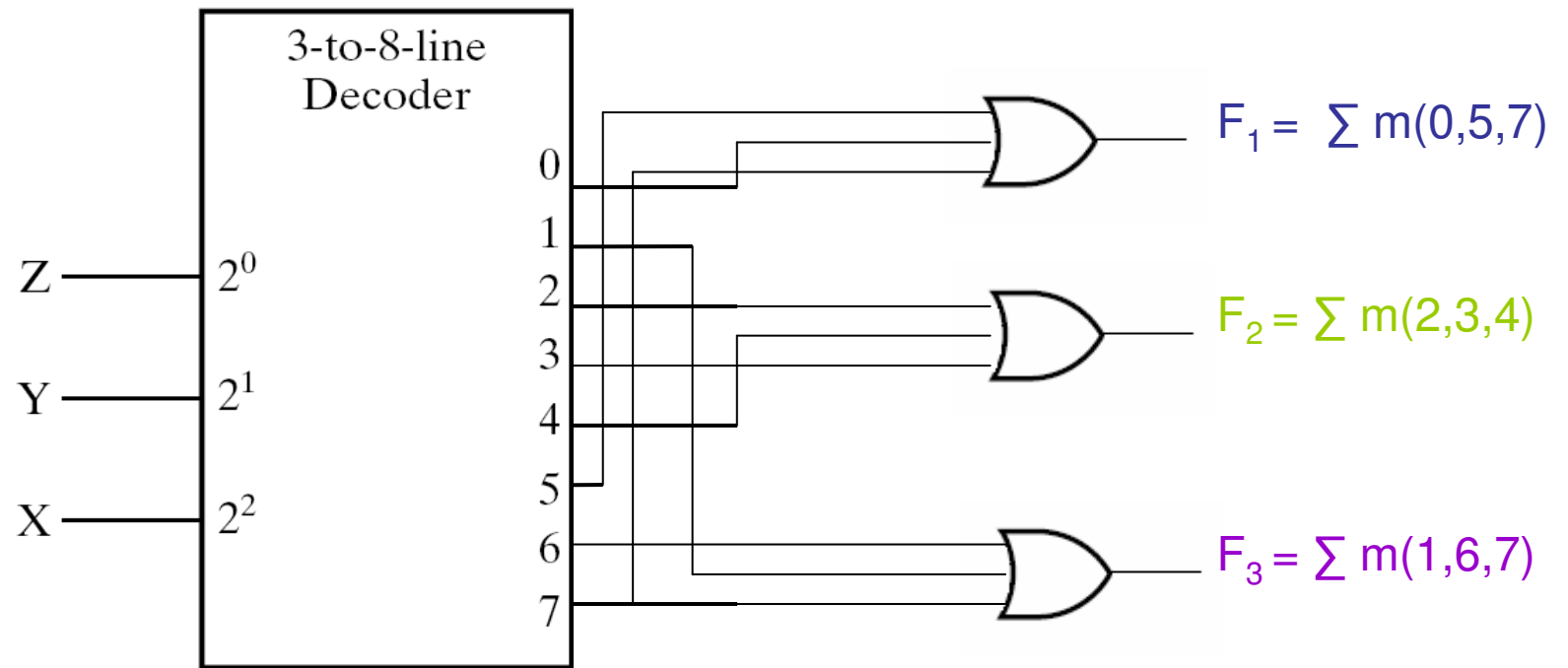
		Y			
		00	01	11	10
X	0	0	0	1	1
	1	1	0	0	0
		Z			

		Y			
		00	01	11	10
X	0	0	1	0	0
	1	0	0	1	1
		Z			

$$F_3(X,Y,Z) = X' Y' Z + X Y$$

$$= \sum m(1,6,7)$$

Esercitazione 4



Multiplexer

- Un multiplexer e' una rete combinatoria che seleziona dati binari e li dirige a una singola linea di uscita.
- La selezione è controllata da un insieme di altri ingressi (ingressi di controllo/selezione).
- Qualunque funzione booleana puo' essere implementata con un multiplexer applicando le variabili booleane alle linee di selezione.

Esercitazione 5

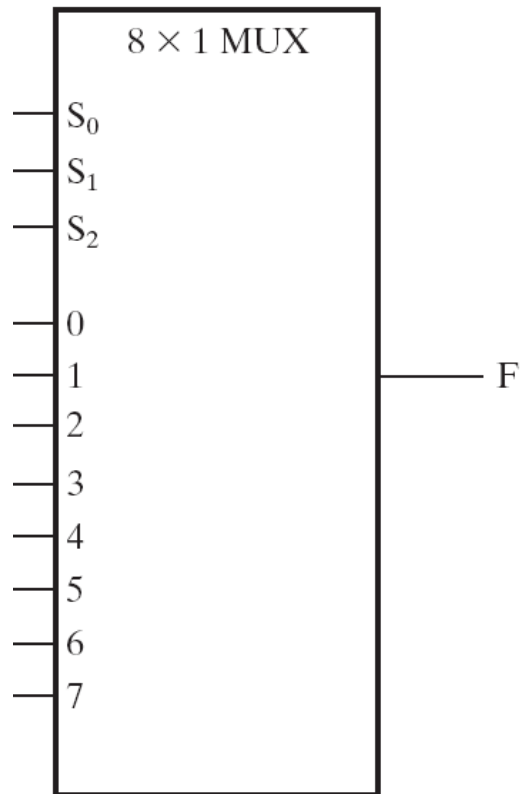
- Implementare la funzione booleana seguente con un multiplexer 8-1 e un singolo invertitore:

$$F(A,B,C,D) = \sum m(1,3,4,11,12,13,14,15)$$

Esercitazione 5

! Un 8x1 MUX ha 3 variabili di selezione.

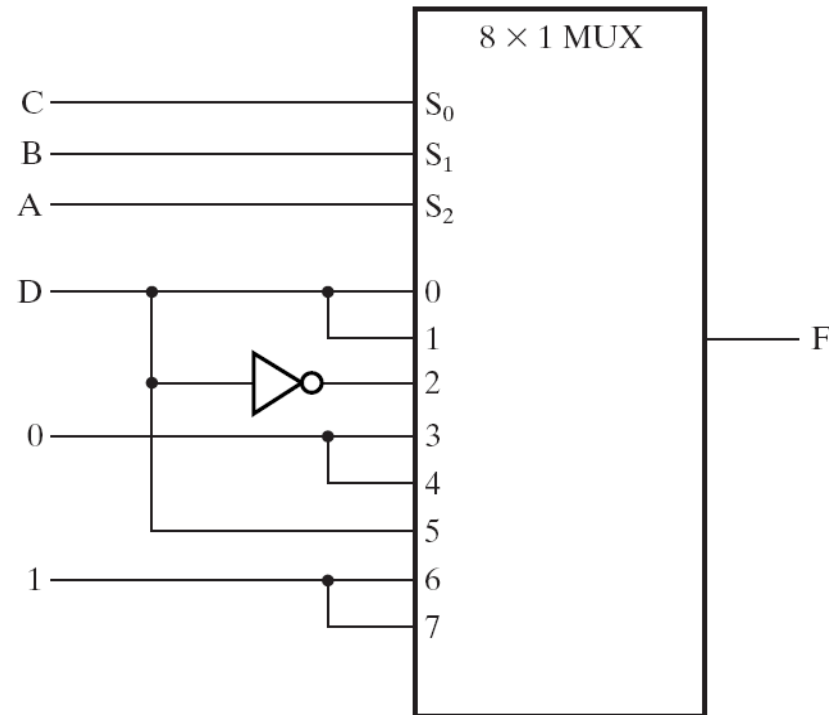
! Abbiamo 4 variabili di ingresso.



Esercitazione 5

- Applicando le prime 3 variabili alle linee di selezione S_2 , S_1 , e S_0 , la funzione puo' essere definita in termini dell'ultima variabile.

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



PLA: Array logico programmabile

- Un PLA e' una rete combinatoria costituito da:
 - un array (matrice) di porte AND che puo' essere programmato per generare alcuni termini di prodotto delle variabili di ingresso;
 - e un array di porte OR che puo' essere programmato per sommare selettivamente i termini.
- L'obiettivo è progettare, in forma di somma di mintermini, qualunque funzione.

Esercitazione 6

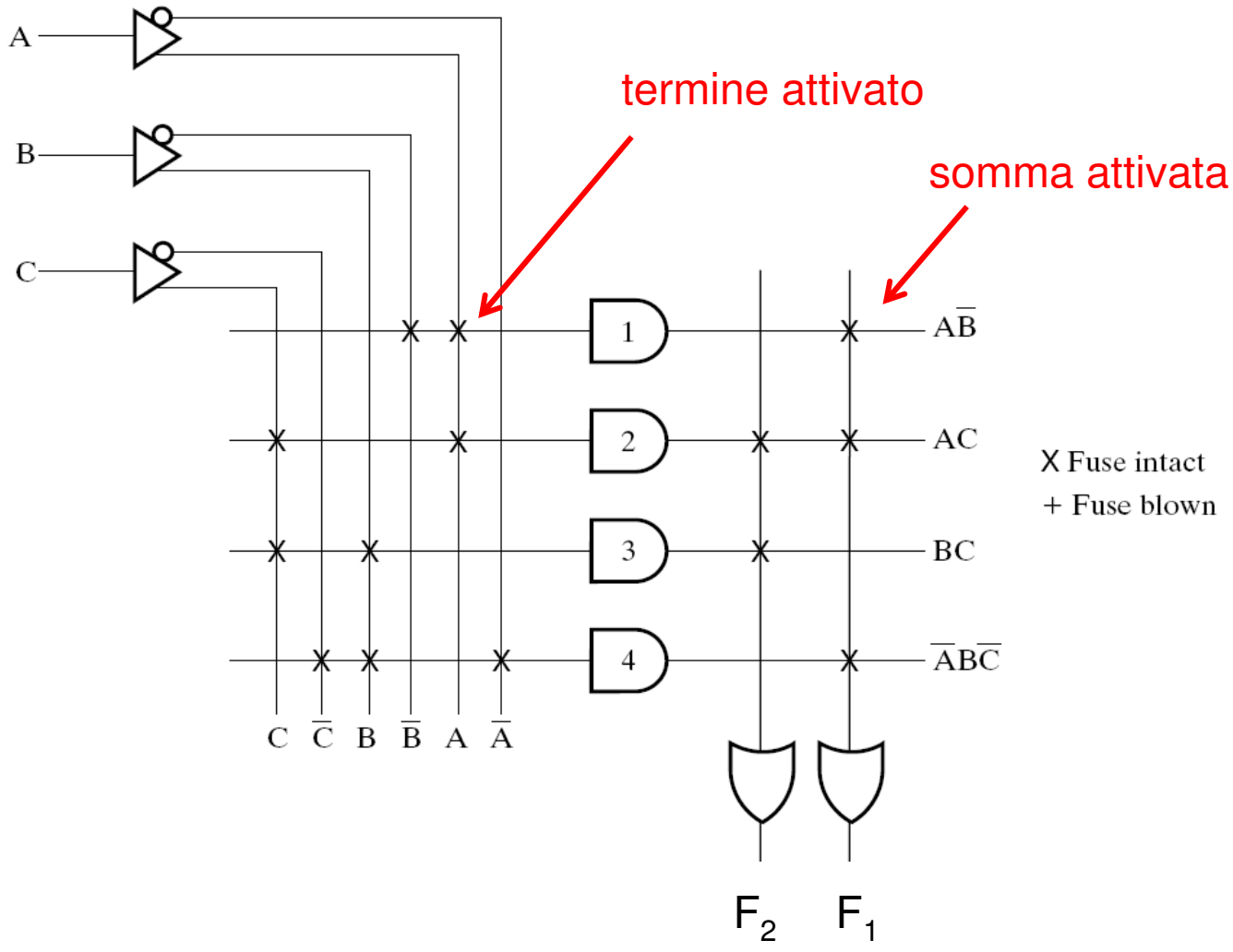
- Implementare con un PLA le funzioni booleane:

$$F_1(A,B,C) = AB' + AC + A'BC'$$

$$F_2(A,B,C) = AC + BC$$

PLA

$$F_1(A,B,C) = AB' + AC + A'BC'$$
$$F_2(A,B,C) = AC + BC$$



PLA

- La mappa puo' essere specificata in forma di tabella.

		Inputs			Outputs	
Product term		A	B	C	(T) F ₁	(C) F ₂
$A\bar{B}$	1	1	0	—	1	—
AC	2	1	—	1	1	1
BC	3	—	1	1	—	1
$\bar{A}B\bar{C}$	4	0	1	0	1	—