Reti Logiche

October 4, 2019

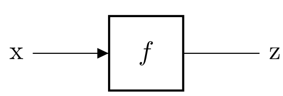
23 Sep 2019, 16:14

1 Propietá delle reti loiche

- Intrconnessione
- Scomposizione Uno schema logico puó essere rappresentato come composizione di reti piú semplici, (si arresta quando si arriva a *reti logiche elemnetari*)
- Elementari Descrivono causa effetto

Nei sistemi pratici la storia passata del sistema deve essere rappresentata attraverso un insieme finito di segnali. (i.e. il numero passato di stati della rete deve essere finito) (slide 36. 01_Modelli)
La funzione di uscita della rete dipende da ingresso corrente e stato interno della rete.

- Funzione di uscita o funzione macchina: $z(t) = f(x(t), y(t))$ (tabella, lookup-table)
- Next step function: $Y(t) = g(x(t), y(t))$, $g(x, y)$ chiamata **Funzione di stato**
- Funzione di marcatura dello stato: $y(t + \Delta t) = Y(t)$ o $y(t) = Y(t - \Delta t)$



Nelle reti logiche combinatorie manca la dipendena dal tempo (slide 41)
Reti sequenziali corrispondono al modello generale, contengono retroazione e ritardi
Rete senza ingressi provenienti dall'eseterno viene chiamata: rete autonoma. ($y(t) = y(t + 1)$: la rete é costante). Altra alternativa é un comportamento ciclico.
Se: $z = f(x)$ e $Y = g(x, y)$: rete combinatoria, dipende solo da z
Rete sequenziale con ingresso non attaccato a g é rete combinatoria (sostituire con figura)
Rete sequenziale con uscita non attaccata a g é rete combinatoria (sostituire con figura)
Sintesi multiuscita pg. 29 slide _2_RetiCombinatorie_I_parte
Numero funzioni di n variabili $F(n) = 2^{2^n}$, slide 30

x_1	x_2	x_3	f
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Metodo somma di prodotti: prendo solo righe con 1, le scrivo come prodotti e le sommo

$$\left.\begin{array}{l} f_3 = \overline{x_1}x_2x_3 \\ f_5 = x_1\overline{x_2}x_3 \\ f_6 = x_1x_2\overline{x_3} \\ f_7 = x_1x_2x_3 \end{array}\right\} f = \overline{x_1}x_2x_3 + x_1\overline{x_2}x_3 + x_1x_2\overline{x_3} + x_1x_2x_3$$

Applicbile a qualsiasi tabella combinatoria (Espressione canonica *sum of products* SP)

1.1 Espressione canonica Prodotto di Somme: PS

Data una tabella della veritá, consideriamo gli zeri della funzione (OFF-set). Per ogni zero possiamo costruire un termine somma in cui compaiono come variabili dipendenti tutte le variabili della funzione di partenza. Le variabili compaiono in forma vera se compaiono come zeri, vengono prese in forma complementata se compaiono come 1 nella riga della tabella.

$$\left.\begin{array}{l} s_0 = x_1x_2x_3 \\ s_1 = x_1x_2\overline{x_3} \\ s_2 = x_1\overline{x_2}x_3 \\ s_4 = \overline{x_1}x_2x_3 \end{array}\right\} S = (x_1x_2x_3)(x_1x_2\overline{x_3})(x_1\overline{x_2}x_3)(\overline{x_1}x_2x_3)$$

Si riescono a sintetizzare reti piú ottimizzate attraverso le mappe di Karnugh o alg. Quiaf-McCluskeyt
N letterari: si applica su soltanto al max 2 livelli. Conta il numero di morsetti del 1^o livello (somma: N_{let} = 12, prodotto 12, altro 6)

2 24 Sep 2019, 9:02

Prendendo in considerazione l'esempio del Full-Adder (slide 4)

2.1 Espressione canonica ***SoP*** delle funzioni S_0 e C_1

$S_0 = \overline{x_0 y_0} c_1 + \overline{x_0} y_0 \overline{c_0} + x_0 \overline{y_0 c_0} + x_0 y_0 c_0 = m_1 + m_2 + m_4 + m_7 = \sum m_3(1,2,4,7)$
$C_1 = \overline{x_0} y_0 c_0 + x_0 \overline{y_0} c_0 + x_0 y_0 \overline{c_0} + x_0 y_0 c_0 = m_3 + m_5 + m_6 + m_7 = \sum m_3(3,5,6,7)$
Un modo piú preciso sarebbe: $S(x_0, y_0, c_0)$, esplicitando l'ordine in cui vengono prese le variabili
$\sum m_3$ indica la sommatoria dei mintermini di 3 variabili

2.2 Espressione canonica ***PoS***

$S_0(x_0, y_0, c_0) = (x_0 + y_0 + c_0)(x_0 + \overline{y_0} + \overline{c_0})(\overline{x_0} + y_0 + \overline{c_0})(\overline{x_0} + \overline{y_0} + c_0) = M_0 M_3 M_5 M_6 = \prod M_3(0,3,5,6)$

2.3 Teorema di Shannon

Ogni espressione logica di n variabili puó essere espressa come:

$$f(x_1, x_2, \ldots, x_n) = x_1 \cdot f(0, x_2, \ldots, x_n) + x_1 \cdot f(1, x_2, \ldots, x_n)$$

Dall'espressione di una funzione riusciamo a riottenere la somma dei mintermini, utilizzando il th. di Shannon:

$$\begin{aligned}
z &= x_1 x_2 + x_1 x_3 + x_2 x_3 = \\
&= \overline{x_1}(0 \cdot x_2 + 0x_3 + x_2 x_3) + x_1(1x_2 + 1x_3 + x_2 x_3) = \\
&= \overline{x_1}(x_2 x_3) + x_1(x_2 + x_3 + x_2 x_3) = \\
&= \overline{x_2}(x_1(0x_3) + x_1(0 + x_3 + 0x_3)) + x_2(\overline{x_1}(1x_3) + x_1(1 + x_2 + 1x_3)) = \\
&= \overline{x_2}(x_1 x_3) + x_2(\overline{x_1} x_3 + x_1) = \\
&= \overline{x_3}(\overline{x_2} x_1 0 + x_2 \overline{x_1} 0 + x_2 x_1) + x_3(\overline{x_2} x_1 1 + x_2 \overline{x_1} 1 + x_2 x_1) = \\
&= \overline{x_2} x_2 x_1 + x_3 \overline{x_2} x_1 + x_3 x_2 \overline{x_1} + x_3 x_2 x_1 = \\
&= m_3 + m_5 + m_6 + m_7
\end{aligned}$$

Le leggi di de morgan dimostrano che le famiglie (And-Not) e (Or-Not) sono funzionalmente complete.
DECODER: ad n ingresi corrispondono 2^n uscite
Espressione generale SP: $f(x_{n-1}, \ldots, x_1, x_0) = \sum_{i=0,2^n-1} m(i) f(i)$
Espressione generale PS: $f(x_{n-1}, \ldots, x_1, x_0) = \prod_{i=0,2^n-1} (M(i) + f(i))$
Il decoder sta in corrispondenza con le espressioni canoniche, il multiplexer sta un corrispondenza con le espressioni generali.

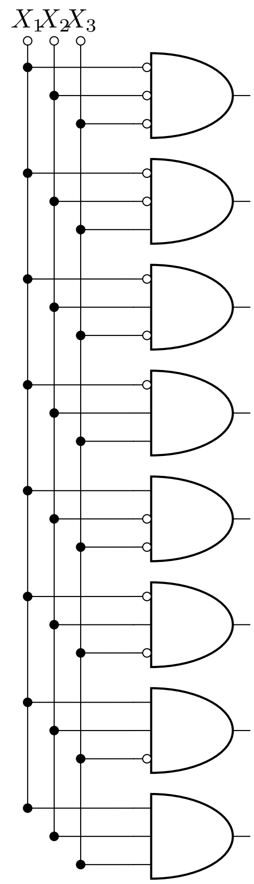


Figure 1: Decoder 3:8 ad AND

3 25 Sep 2019, 14:46

- Il segnale RC (RC_0 e RC_1) dipendono da $n-1$ variabili. e non da n variabili.
- RC_0 e RC_1 potrebbero essere semplificabili

Una funzione di n variabili puó essere espressa come una combinazione di due funzioni di n-1 variabili attraverso un Multiplexer.

$$f(x_1, x_2, \ldots, x_n) = \overline{x_1}\,\overline{x_2} f(0, 0, \ldots, x_n) + \overline{x_1} x_2 f(0, 1, \ldots, x_n) + x_1 \overline{x_2} f(1, 0, \ldots, x_n) + x_1 x_2 f(1, 1, \ldots, x_n)$$

Funzione semplificata di due variabili con Th. di Shannon.

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

$$f_0 = \overline{C}D$$
$$f_1 = 1$$
$$f_3 = C + \overline{D}$$
$$f_2 = \overline{C}D$$

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

$$f_0 = B + \overline{C}D$$
$$f_1 = B\overline{D} + BC + \overline{B}\,\overline{C}D$$

Supponendo di dover realizzare la funzione con un multiplexer a 4 vie, utilizzando esclusivamente un mux e non altri componenti:
Nel caso in cui fossimo partiti dalle variabili CD:

$$f_0 = B$$
$$f_1 = \overline{A} + \overline{B} = \overline{A \cdot B}$$
$$f_2 = B$$
$$f_3 = B$$

- Espressione canonica SP
  - AND per mintermini + OR
  - Decoder + or
- Espressione canonica PS
  - OR per Maxtermini + AND
  - Decoder a NAND + AND
- Espressione generale SP e PS: MuX
- Espressione parziali: (1 MUx + altro)

a	b	c	Sum	Count
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

Count $= \overline{a}bc + a\overline{b}c + ab\overline{c} + abc = \overline{a}bc + a\overline{b}c + ab$

$$\begin{cases} N_{op} = 5 \\ N_m = 3 \cdot 4 + 4 = 16 \end{cases} \rightarrow \begin{cases} N_{op} = 4 \\ N_m = 11 \end{cases}$$

Definiamo come **distanza di Hamming** ($\mathcal{D}_H$) il numero di bit che differiscono in 2 configurazioni binarie
Es: $\mathcal{D}_H(110, 111) = 1$
Se $\mathcal{D}_H(C_i, C_j) = 1 \Rightarrow C_i \,\text{e}\, C_j$ sono **adiacenti**.

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

Raggruppamento rettangolare di 2 caselle 2^1 caselle ($2^k = 2^1$) (Raggruppamento di ordine 1)
$C_{out} = I_1 + I_2 + I_3 = ac + ab + bc \qquad (N_{op} = 4, N_m = 9)$

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

$E = 0$

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

$E = 1$

$Z(a, b, c, d, e)$
'-' rappresenta una condizione di indifferenza.
Nei raggruppamenti prendo in considerazione le variabili che rimangono costanti

$$Z(a, b, c, d, e) = b\overline{e} + \overline{c}d\overline{e} + b\overline{c}\overline{d} + a\overline{c}d + ab$$

$N_{lett} = 13$, ingressi del primo livello
$N_{op} = 6$, numero mintermini + 1
$N_m = 18$, tutte le lettere che compaiono + mintermini

3.1 Sintesi prendendo in considerazione gli zeri

Uno 0 di una funzione é un maxtermine, ed esprime un solo 0. L'obbiettivo é costruire termini somma (Implicati)

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

E= 0

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

E= 1

$$Z = (B+\overline{C})(A+\overline{D}+\overline{E})(\overline{C}+D+\overline{E})(B+D) = (B \downarrow \overline{C}) \downarrow (A \downarrow \overline{D} \downarrow \overline{E}) \downarrow (\overline{C} \downarrow D \downarrow \overline{E}) \downarrow (B \downarrow D)$$

$N_{op} = 5$
$N_m = 14$
Sintesi di $Z(A, B, C, D, E)$

- mediante soli MUX a 4 vie
- 1 porta logica + MUX a 4 vie

4 lezione 30 Sep 2019, 14:38

Es:

X_3X_2 \ X_0X_1	00	01	11	10
00	0	1	1	1
01	0	1	1	0
11	0	1	0	0
10	0	0	1	0

X_4= 0

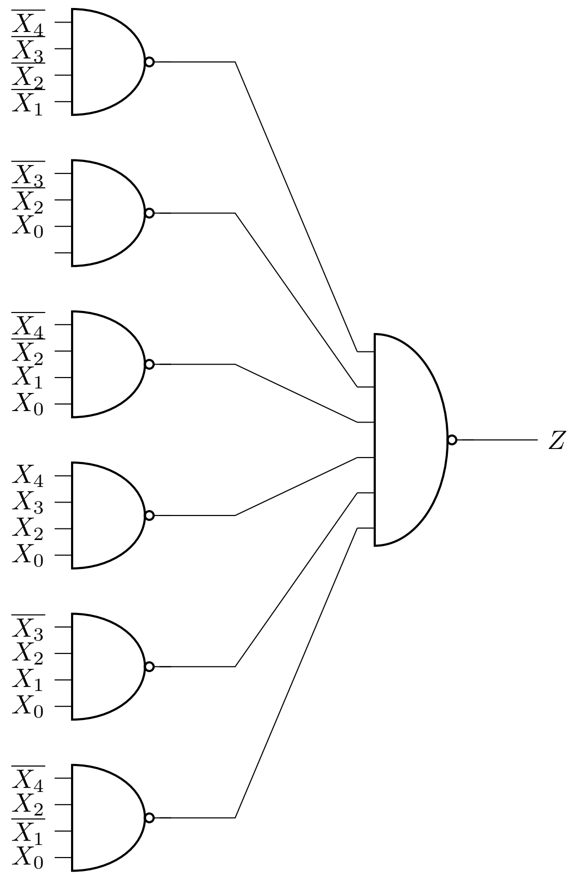
X_3X_2 \ X_0X_1	00	01	11	10
00	0	1	1	0
01	0	0	1	0
11	0	1	1	0
10	0	0	0	0

X_4= 1

$$Z = \overline{x_4 x_3 x_2 x_3 x_2} x_0 + \overline{x_3 x_2} x_0 + \overline{x_4 x_2} x_1 x_0 + + x_4 x_3 x_2 x_0 + \overline{x_3} x_2 x_1 x_0 + \overline{x_4} x_2 \overline{x_1} x_0$$

$$N_b = 7 \qquad N_m = 29$$

Operatore NAND: $x_1 \uparrow x_2 \uparrow \ldots \uparrow x_n = \overline{x_1 \cdot x_2 \cdot \ldots \cdot x_n}$
Operatore NOR: $x_1 \downarrow x_2 \downarrow \ldots \downarrow x_n = \overline{x_1 + x_2 + \ldots + x_n}$
(Non vale la proprietà associativa)
(Completezza funzionale NAND lezione 5 slide 6)
(Completezza funzionale NAND lezione 5 slide 7)



È possibile convertire un qualsiasi circuito logico utilizzando soli NAND.
Costo del circuito semplificato: $N_m = 29$, esattamente uguale a quello con operatori booleni, il costo N_m e $N_b \coloneqq N_{op}$ è esattamente lo stesso

4.1 Come passare da SP a NAND

Lezione 5 slide 8

1. Involuzione: doppia negazione
2. Trasformo dell'involuzione con De Morgan

Es.

$$\begin{aligned} Z &= \bar{a} + a\bar{b} + a\bar{b}\bar{c} + ab\bar{c} \\ &= \overline{\overline{(\bar{a} + a\bar{b} + a\bar{b}\bar{c} + ab\bar{c})}} \\ &= \overline{a \cdot (a \uparrow \bar{b}) \cdot (a \uparrow \bar{b} \uparrow \bar{c}) \cdot (a \uparrow b \uparrow \bar{c})} \\ &= a \uparrow (a \uparrow \bar{b}) \uparrow (a \uparrow \bar{b} \uparrow \bar{c}) \uparrow (a \uparrow b \uparrow \bar{c}) \end{aligned}$$

(Slide 10 punto 4 ultimo b è negato)
NOTA: MAI sintetizzare una rete NOR partendo da SP

4.2 Trasformazione a NAND in rete a più livelli

1. Inserire le parentesi sottintese
2. Numerare i livelli
3. Trasformare gli operatori AND e OR in NAND
4. Complementare eventuali letterari isolati che costituiscono variabili di funzioni OR (tutte a livello pari)

4.3 Trasformazione a NOR in rete a più livelli

1. Inserire le parentesi sottintese
2. Numerare i livelli
3. Trasformare gli operatori AND e OR in NOR
4. Complementare eventuali letterari isolati che costituiscono variabili di funzioni AND (tutte a livello dispari)

$$Z = (b \uparrow (\overline{d} \uparrow e)) \uparrow (\overline{c} \uparrow (1 \uparrow (\overline{d} \uparrow c))) \uparrow a$$

Operatori a livello dispari: Somme, pari: prodotti

$$= (b \cdot (d + \overline{e})) + (\overline{c} \cdot (0 + (\overline{d} \cdot e))) + \overline{a} =$$
$$= bd + b\overline{e} + \overline{c}\overline{d}e + \overline{a}$$

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

$e = 0$

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

$e = 1$

$$Z = (e + \overline{a} + b)(\overline{e} + \overline{c} + d + \overline{a})(\overline{a} + b + \overline{d}) = (e \downarrow \overline{a} \downarrow b) \downarrow (\overline{e} \downarrow \overline{c} \downarrow d \downarrow \overline{a}) \downarrow (\overline{a} \downarrow b \downarrow \overline{d})$$

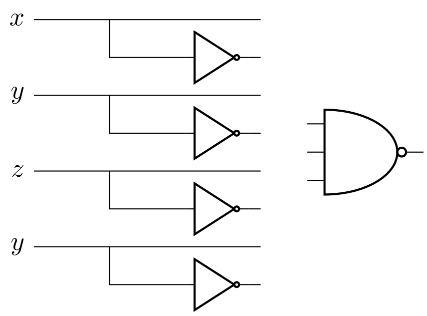
4.4 Esercizio prova passata d'esame

Realizzare la funzione sotto riportata. Utilizzando solamente i componenti a disposizione (4 nand + variabili in forma negata)

$$F = xy\overline{z} + \overline{x}y\overline{x}\overline{w} + xyz\overline{w} + \overline{x}yz$$

$zw \backslash xy$	00	01	11	10
00	0	0	0	0
01	1	0	1	1
11	1	1	0	1
10	0	0	0	0

$$F = y\overline{w} + \overline{x}yz + xy\overline{z} = (y \uparrow \overline{w}) \uparrow (\overline{x} \uparrow y \uparrow z) \uparrow (x \uparrow y \uparrow \overline{z})$$



4.4.1 Realizzare F con soli MUX a 4 vie (2 var di selezione)

4.4.2 Usando un MUX a 3 variabili di selezione + altro

5 Lezione 01 Oct 2019, 9:08

5.1 Conversione da realizzazione NAND a NOR e viceversa

Conversione da SP a NAND-NAND

$$Z = a\overline{b} \Leftrightarrow Z = (a\overline{b}) + 0 = (a \uparrow \overline{b}) \uparrow 1$$

Conversione PS a NOR-NOR

$$Z = a \cdot b = \overline{a} \downarrow \overline{b}$$

$$Z = \overline{a} + b = a \uparrow \overline{b}$$

$$Z = (\overline{a} + b) \cdot 1 = (\overline{a} \downarrow b) \downarrow 0$$

Metodo **Standard**:

1. Da schama ad espressione NAND
2. espressione NAND a espressione booleana (SP)
3. Valutazione tramite mappa K
4. Resintesi PS

5. Da PS a NOR (Schema NOR)

$$Z = ((x \uparrow y) \uparrow x) \uparrow ((x \uparrow y) \uparrow y)$$
$$Z = (\overline{x} + \overline{y}) \cdot x + (\overline{x} + \overline{y}) \cdot y = x\overline{y} + \overline{x}y$$

$X \backslash Y$	0	1
0	0	1
1	1	0

$$Z = (x + y)(\overline{x} + \overline{y}) = (x \downarrow y) \downarrow (\overline{x} \downarrow \overline{y})$$

5.2 Proprietà di Dualità

Da ogni realizzazione booleana se ne ricava un'altra, duale, sostituendo AND $\leftrightarrow$ OR (Espressione duale) (Non si toccano i complementi)
Espressione duale viene indicata con E^D
NOTA:

$$E(x, y, z) \to E^D(x, y, z) \neq E(x, y, z)$$

5.3 Teorema di dualità

Se ho un'espressione $Z = f(x_1, x_2, \ldots, x_n)$, allora $Z^D = \overline{f}(\overline{x_1}, \overline{x_2}, \ldots, \overline{x_n})$.
Inoltre la proprietà di dualità soddisfa l'involuzione: $(Z^D)^D = Z$

5.4 Conversione NAND $\to$ NOR (o viceversa) mediante dualità

- Ricavo espressione duale
- Applico teorema dualità

$$Z^D = ((x \downarrow y) \downarrow x) \downarrow ((x \downarrow y) \downarrow y)$$
$$Z = (Z^D)^D = \overline{((\overline{x} \downarrow \overline{y}) \downarrow \overline{x}) \downarrow (\overline{x} \downarrow \overline{y}) \downarrow \overline{y})}$$

5.5 Esercizio esame

Data la funzione delle seguente variabili $Z(x_4, x_3, x_2, x_1, x_0) = \sum m_5(1, 2, 3, 4, 5, 11, 13, 17, 23, 24, 25, 27, 31) + d_5(6, 10, 12, 18, 20, 21, 22, 30)$

1. Trascrivere la funzione sulla mappa K
2. Ricavare l'espressione NOR ottima priva di Alee e riportare il costo in N_{lett}
3. Usare esclusivaente NOR a 3 ingressi e NOT (schema logico, costo N letterali)
4. Ricavare una realizzazione di Z basata su un MUX a 3 var di selezione + altro

$X_4 = 0$

X_2X_3 \ X_0X_1	00	01	11	10
00	0	1	1	1
01	1	1	0	-
11	-	1	0	0
10	0	0	1	-

$X_4 = 1$

X_2X_3 \ X_0X_1	00	01	11	10
00	0	1	0	-
01	-	-	1	-
11	0	0	1	-
10	1	1	1	0

$$Z = (x_4 + \overline{X_2} + \overline{X_1})(x_4 + \overline{X_3} + X_2 + X_1)(\overline{X_4} + \overline{X_2} + X_1)(\overline{X_4} + X_3 + x_2 + \overline{X_1})(X_3 + X_2 + X_1 + X_0)(\overline{X_4} + \overline{X_3} + X_0))$$
$$= (x_4 \downarrow \overline{X_2} \downarrow \overline{X_1}) \downarrow (x_4 \downarrow \overline{X_3} \downarrow X_2 \downarrow X_1) \downarrow (\overline{X_4} \downarrow \overline{X_2} \downarrow X_1) \downarrow (\overline{X_4} \downarrow X_3 \downarrow x_2 \downarrow \overline{X_1}) \downarrow (X_3 \downarrow X_2 \downarrow X_1 \downarrow X_0) \downarrow (\overline{X_4} \downarrow \overline{X_3} \downarrow X_0))$$

Elenco componenti utilizzati: $N_{lett} = 21$

- 3 NOR a 3 ingressi
- 3 NOR a 1 ingressi
- 1 NOR a 6 ingressi

Elenco componenti punto 3:

- 6 NOR a 3 ingrressi
- 3 NOR a 2 ingrressi
- 2 NOR a 3 ingrressi
- 1 NOR a 2 ingrressi
- 5 NOT

Costo in letterari (NOTA: la rete non è più a 2 livelli, per calcolare il costo lo calcolo 'a pezzi di 2 livelli'):
$N_{lett} = N_{lett_a} + N_{lett_b} + N_{lett_c} = 9 + 12 + 4 = 25$

- Partire sempre dal MUX che pilota l'uscita

6 Lezione 02 Oct 2019, 14:41

6.1 Analisi di reti combinatorie con porte logiche anche multilivello

Utilizzo delle espresisoni variabili

$$
\begin{aligned}
F_1 &= T \cdot \overline{D} + \overline{c} \\
F_2 &= F_1 + w\overline{A}D \\
T &= a \uparrow \overline{b} = \overline{a} + b \\
W &= (a \uparrow \overline{b}) + c \\
\hline
F_1 &= (\overline{a} + b) \cdot \overline{D} + \overline{c} = \overline{AD} + B\overline{D} + \overline{c} \\
F_2 &= \overline{AD} + B\overline{D} + \overline{C}\overline{A}D + \overline{A}CD
\end{aligned}
$$

$CD \backslash AB$	00	01	11	10
00	1	1	0	1
01	1	1	0	1
11	1	1	0	1
10	1	1	0	0

$CD \backslash AB$	00	01	11	10
00	1	1	1	1
01	1	1	1	1
11	1	1	0	1
10	1	1	0	0

$$F_1 = (\overline{C} + \overline{D})(\overline{A} + B + \overline{C}) = (\overline{C} \downarrow \overline{D}) \downarrow (\overline{A} \downarrow B \downarrow \overline{C})$$

$$F_2 = (\overline{C} + \overline{D} + \overline{A})(\overline{A} + B + \overline{C}) == (\overline{C} \downarrow \overline{D} \downarrow \overline{A}) \downarrow (\overline{A} \downarrow B \downarrow \overline{C})$$

$N_{lett} = 3 + 3 + 4 = 10$

6.2 Analisi di reti combinatorie con componenti MSI

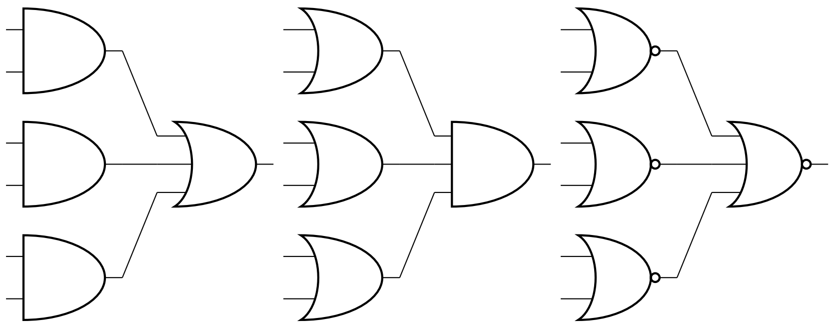
$$
\begin{gathered}
Z = \overline{A}f_0 + Af_1 \\
f_0 = BC \oplus \overline{B}C = \overline{(BC)}\overline{B}C + BC\overline{(\overline{B}C)} = \ldots = C \\
f_1 = \overline{BCD} + \overline{(B\overline{C})A} = \overline{BCD} + \overline{B} + C + \overline{A} = \overline{B} + C + \overline{A}
\end{gathered}
$$

$$Z = \overline{A}C + A(\overline{A} + \overline{B} + C) = C + A\overline{B}$$

$C \backslash AB$	00	01	11	10
0	0	1	1	0
1	1	1	1	0

$$F = (\overline{B} + C)(A + C)$$

6.3 Sintesi con espressione a due livelli



And-Or , Or-And, Nor-Nor

X_3X_2 \ X_1X_0	00	01	11	10
0	1	0	1	1
1	0	0	1	0

$$Z = \overline{a}\overline{c} + bc = (\overline{a} \uparrow \overline{c}) \uparrow (b \uparrow c)$$

X_3X_2 \ X_1X_0	00	01	11	10
0	1	0	1	1
1	0	0	1	0

$$Z = (\overline{a} + c)(b + \overline{c}) = (\overline{a} \downarrow c) \downarrow (b \downarrow \overline{c})$$

X_3X_2 \ X_1X_0	00	01	11	10
0	0	1	0	0
1	1	1	0	1

$$Z_2 = a\overline{c} + \overline{b}c$$

X_3X_2 \ X_1X_0	00	01	11	10
00	1	-	-	1
01	0	1	0	0
11	1	1	-	0
10	-	1	0	1

Espressioni a due livelli generalizzate

SP: $z = \overline{bd} + a\overline{c} + bd$
NAND-NAND: $Z = (\overline{b} \uparrow d) \uparrow (a \uparrow \overline{c}) \uparrow (b \uparrow d)$
OR-AND: $Z = (\overline{b} + \overline{c})(\overline{c} + \overline{d})(a + \overline{b} + d)$
NOR-NOR: $Z = (\overline{b} \downarrow c) \downarrow (\overline{c} \downarrow \overline{d}) \downarrow (a \downarrow \overline{b} \downarrow d)$
AND -NOR: $Z = (bc) \downarrow (cd) \downarrow (\overline{a}v\overline{d})$
NAND-AND: $Z = (b \uparrow c)(c \uparrow d)(\overline{a} \uparrow b \uparrow d)$
OR NAND: $Z = (b + d) \uparrow (\overline{b} + \overline{d} \uparrow (\overline{a} + c)$
NOR-OR: $Z = (b \downarrow d) + (\overline{b} \downarrow \overline{d}) + (\overline{a} \downarrow c)$

6.4 Alee

X_3X_2 \ X_1X_0	00	01	11	10
0	1	0	1	1
1	0	0	1	0

$Z = \overline{ac} + bc$
Per passare dalla configurazione 010 a 111 ci si impega del tempo. In un breve lasso di tempo le variabili che cambiano assumono valori diversi (es. 010 $\rightarrow$ 011 $\rightarrow$ 111)
Nel caso in: 010 $\rightarrow$ 011 $\rightarrow$ 111, Z rimane costante. Nel caso 010 $\rightarrow$ 110 $\rightarrow$ 111, Z potrebbe presentare un glitch (guarda mappa k). Lo scenario viene detto di *Multiple Input Change*.
Questi ‘glitch’ vengono chiamati **Alee**.

6.5 Single input change

X_3X_2 \ X_1X_0	00	01	11	10
0	1	0	1	1
1	0	0	1	0

$abc = 011 \rightarrow abc = 010$
Se $\tau_1 + \tau_3 > \tau_4$: allora
Alea Statica: Possibilità che un segnale che deve restare costante abbia un breve transitorio al livello opposto (= possibilità di glitch)
Alea Statica di 1: Il segnale dovrebbe restare costante ad 1
Alea Statica di 0: il segnale deve restare costante a 0
Alea Dinamica: il segnale che deve cambiare (da 0 a 1 o da 1 a 0) ma lo fa a rischio di più variazioni (c'è una serie di rimbalzi).

6.6 Teoremi

- T1: I circuiti logici derivati da espressioni SP normali sono privi di alee statiche "0"
- T2: I circuiti logici derivati da espressioni PS normali sono privi di alee statiche "1"
- Corollario: Valgono anche per espressioni a due livelli generalizzate
- T3: Rilevazione alee statiche
  - (SP) Un espressione contiene un'alea statica se ci sono 2 mintermini adiacenti non coperti dal medesimo implicante.
  - (PS) Un espressione contiene un'alea statica se ci sono 2 maxterm adiacenti non coperti dal medesimo implicato.

X_3X_2 \ X_1X_0	00	01	11	10
0	1	0	1	1
1	0	0	1	0

  Un'espressione normale SP non contiene prodotti in cui compare due volte la stessa variabile.

  Un'espressione normale PS non contiene somme in cui compare due volte la stessa variabile.

- T4: Le espressioni SP e PS normali sono prive di alee dinamiche.

L'alea dinamica può nascere nelle reti a più (di 2) livelli per effetto di alee statiche nei livelli a monte. La cura consiste nell'eliminare le alee statiche tra coppie di livelli.