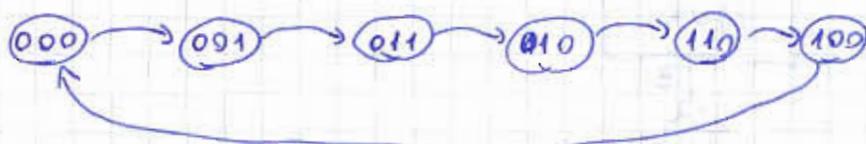
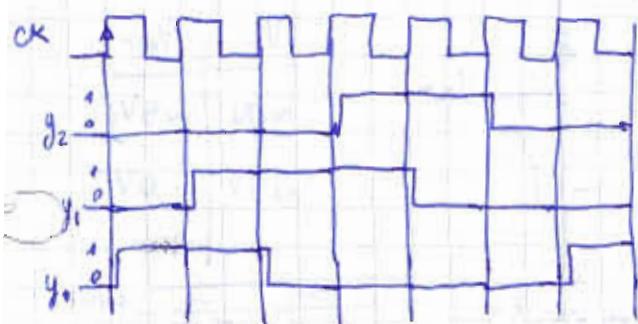
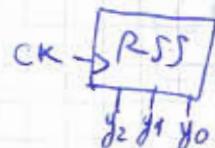


Progettare un contatore con sequenza 000-001-011-010-110-100 con FF-JK, NAND

1) Generare due forme d'onda.



RETE SEQUENZIALE SINCRONA



EVOLUZIONE

TABELLA TRANSIZIONI
 $(y_2 y_1 y_0)^n \rightarrow (y_2 y_1 y_0)^{n+1}$

y^n	y^{n+1}	J	K
0 0 0	0 0 1	0	0
0 0 1	0 1 1	1	1
0 1 1	0 1 0	0	1
0 1 0	1 1 0	1	0
1 0 0	0 0 0	0	0
1 0 1	-	-	-
1 1 1	-	-	-
1 1 0	1 0 0	-	-

Evidenzia le configurazioni che commutano

y_2	y_1	y_0	00	01	11	10
0	001	011	010	110	-	-
1	000	-	-	100	-	-

$(y_2 y_1 y_0)^{n+1}$

$y^n \rightarrow y^{n+1}$	J	K
0 0	0	0
0 1	1	1
0 0	0	1
1 1	1	0

y_2	y_1	y_0	-	-
-	1	-	-	-
1	-	0	-	-

y_2	y_1	y_0	-	-
0	1	-	-	-
0	-	-	-	-

$$J_2 = y_1 \bar{y}_0$$

$$K_2 = \bar{y}_1$$

$$J_1 = y_0$$

$$K_1 = y_2$$

1	-	-	0
0	-	-	0

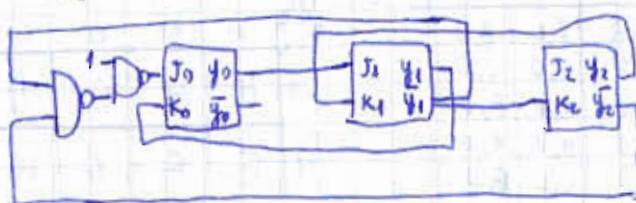
-	0	1	-
-	-	-	-

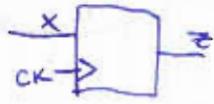
$$J_0 = \bar{y}_2 \bar{y}_1$$

$$K_0 = y_1$$

$$= (\bar{y}_2 \uparrow \bar{y}_1) \uparrow \uparrow$$

$$\begin{array}{l} \text{NAND } \uparrow \\ J_2 = (y_1 \bar{y}_0) + 0 = (y_1 \bar{y}_0) \uparrow \uparrow \\ \text{NOR } \downarrow \end{array}$$

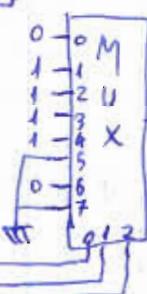
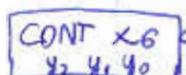




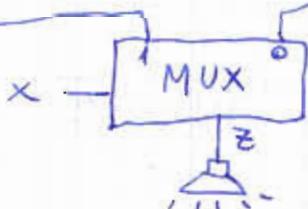
$$x=0 \quad F_0 = \{1, 1, 0, 1, 0, 0\}$$

$$x=1 \quad F_1 = \{0, 1, 1, 1, 0, 1\}$$

contatore che visita i censi secondo l'ordine di prima

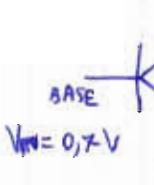


F_1

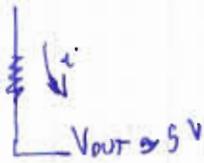


* stati non usati, meglio metterli a un valore definito

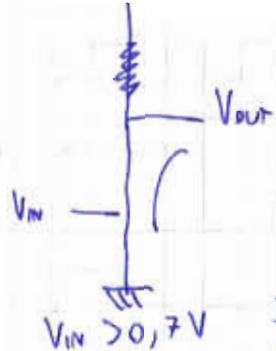
5V



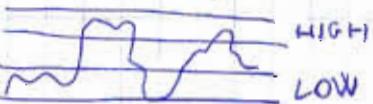
COLLETTORE
EMETTORE
BJT
non lineare



$V_{IN} < 0.7V$



V_{IN}	V_{OUT}
NON	$\sim 5V$
$\sim 5V$	$\sim 0V$



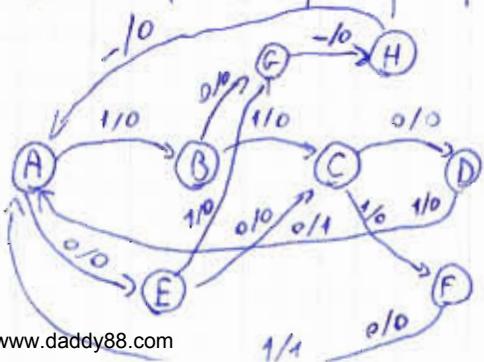
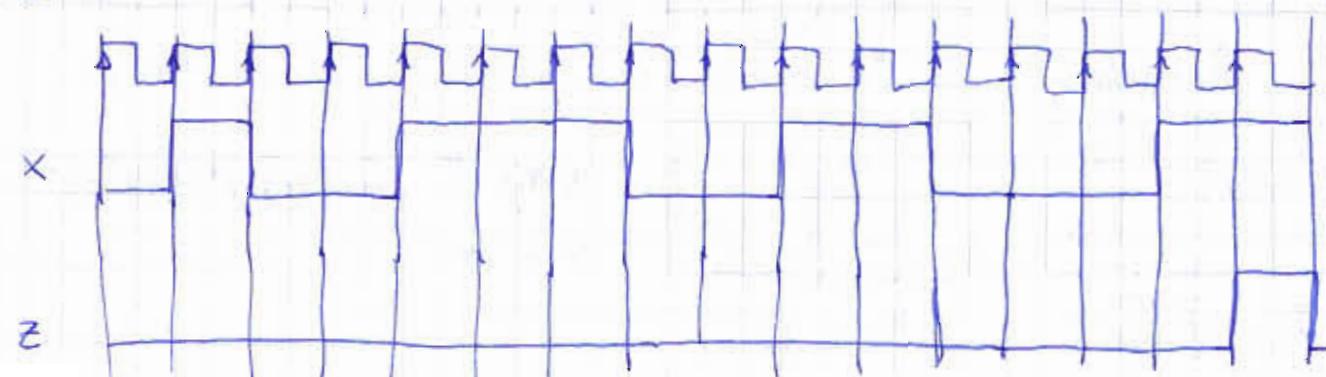
V_{IN}	V_{OUT}
0	1

07/10/2008

E/S 1 del 14/09/2007

$x \rightarrow 101101110\dots$ $z=1$ per 1 periodo di clock nel quarto bit n° e solo se

- nei primi 2 bit, n° pari di "1" è
- nei secondi 2 bit, n° pari di "0"



A	0	1
B	E, 0	B, 0
C	G, 0	C, 0
D	D, 0	F, 0
E	A, 1	A, 0
F	C, 0	B, 0
G	A, 0	A, 1
H	H, 0	H, 0

	00	01	11	10
0	A	B	C	D
1	E	F	G	H

1) Diagramma degli stati



2) Tabelle di flusso

	0	1
0	0	1
1	B, 1	C, 1
2		
3		

stato successivo, uscite

3) Minimizzazione degli stati

4) Tabela delle transizioni (mappe di codifica)

X ₃	0	1
X ₂	000	001
X ₁	000	001
X ₀	000	001

	0	1
y ₂ y ₁ y ₀	x	
A 000	100, 0	001, 0
B 001	111, 0	011, 0
C 011	010, 0	101, 0
D 010	000, 1	000, 0
E 100	011, 0	111, 0
F 101	000, 0	000, 1
G 111	110, 0	110, 0
H 110	000, 0	000, 0

5) Sintesi delle funzioni

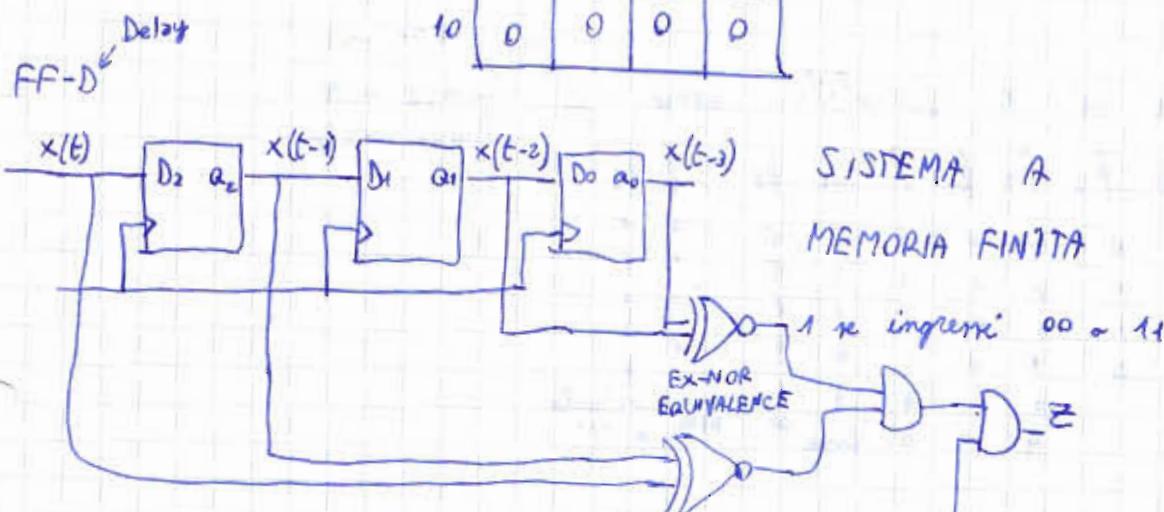
- eccitazioni FF

- uscite

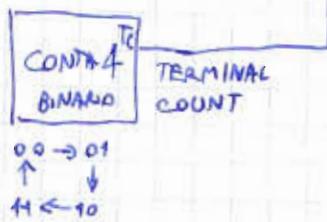
y ₂ x	00	01	11	10
00	(1)	0	(1)	0
01	(1)	0	0	0
11	0	(1)	(1)	1
10	0	0	0	0

$$y_2^{(nn)} = \bar{y}_2 \bar{x} \bar{y}_1 + y_2 \bar{x} y_1 \bar{y}_0 + x y_1 y_0 + y_2 y_1 y_0$$

6) Disegno del circuito



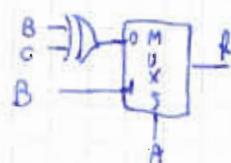
Controlla che si rispetti la sincronizzazione di controllore ogni 4 bit



ES 3

Osta una tabella di verità creare il circuito con due moltiplicatori e porte logiche.

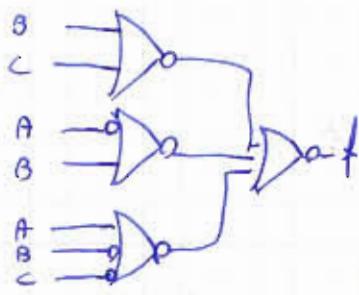
ABC	f
000	0
001	1
010	1
011	0
100	0
101	0
110	-



	00	01	11	10
0	(0)	1	(0)	1
1	(0)	(0)	1	-

Sintesi a NOR

$$f = \overline{ABC} \cdot (\bar{A} + B) \cdot (A + \bar{B} + \bar{C}) = (\bar{B} \cdot \bar{C}) \cdot (\bar{A} + B) \cdot (A + \bar{B} + \bar{C})$$



chiedere NOR con
FAN-IN = 2, cioè
n° max di ingressi
uguale a 2.

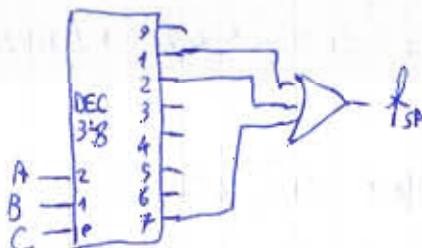
$$\overline{D_o} \Leftrightarrow \overline{\overline{D_o}} = D_o$$

Sintesi NAND: $f = \bar{A}\bar{B}C + B\bar{C} + AB = (\bar{A} \uparrow \bar{B} \uparrow C) \uparrow (B \uparrow \bar{C}) \uparrow (A \uparrow B)$

Espressione canonica SP $f_{SP} = \bar{A}\bar{B}C + \bar{A}B\bar{C} + ABC$ prendo gli 1

Espressione " PJ $f_{PJ} = (A+B+C) \cdot (A+\bar{B}+\bar{C}) \cdot (\bar{A}+B+C) \cdot (\bar{A}+\bar{B}+\bar{C})$ prendo gli 0.

DECODER \rightarrow identifica tutti i minterms



Possibili funzioni di n variabili sono 2^{2^n}

- $n=1 \rightarrow 4$
- $n=2 \rightarrow 16$
- $n=3 \rightarrow 256$

A	B	f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}
0	0	0	0	0	0	0	0	0	1	1	1	1	1
0	1	0	0	0	1	1	1	1	1	1	1	1	1
1	0	0	0	1	1	0	0	1	1	1	1	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1

costante "0" AND A B Ex-OR OR NAND costante "1"

X	f_0	f_1	f_2	f_3
0	0	0	1	1
1	0	1	0	1

cost. X NOT cost "1"
"0" (buffer)

09/10/08

PROVA INTERMEDIA \rightarrow 19/11/2008

SINTESI \rightarrow dalla tabella di verità al circuito

ANALISI \rightarrow del circuito alle tabelle di verità.

MINTERM \rightarrow ONSETMAXTERM \rightarrow OFF-SETRETE COMBINATORIA \rightarrow circuito privo di retroazione

CRITERIO DI COSTO 1

1) numero di porte logiche

2) numero di morsetti di ingresso totali

CRITERIO DI COSTO 2

1) numero di letterali; per le reti a due livelli conta il numero di ingressi del 1° livello

16/10/08

MINTERM \rightarrow prodotto che contiene in modo vero o negato tutte le variabili di una funzioneMAXTERM \rightarrow somma " " " " " " " " " " " " " "1^a FORMA CANONICA \rightarrow somma dei minterm che generano uscita 1 (SP) $\sum m(1,3,..)$ 2^a FORMA CANONICA \rightarrow prodotto dei maxterm che generano uscita 0 (PS) $\prod M(2,4,5,..)$ LEGGI J1 DE MORGAN $\rightarrow \overline{A \cdot B} = \overline{A} + \overline{B}$ $\overline{A+B} = \overline{A} \cdot \overline{B}$

$$\overline{D_0 \Leftrightarrow D_1} \quad \overline{D_0 \Leftrightarrow \overline{D_1}}$$

$$\overline{D_1} \Leftrightarrow \overline{D_0} \quad \overline{D_1} \Leftrightarrow \overline{\overline{D_0}}$$

TEOREMI DI ESPANSIONE DI SHANNON

$$f(x_1, x_2, \dots, x_n) = \bar{x}_1 \cdot f(0, x_2, \dots, x_n) + x_1 \cdot f(1, x_2, \dots, x_n)$$

$$f(x_1, x_2, \dots, x_n) = (\bar{x}_1 + f(1, x_2, \dots, x_n)) \cdot (x_1 + f(0, x_2, \dots, x_n))$$

es.

$$z(x_0, x_1) = x_0 x_1 + \bar{x}_1 \stackrel{T.S.}{=} \bar{x}_0 (0 \cdot x_1 + \bar{x}_1) + x_0 (1 \cdot x_1 + \bar{x}_1)$$

Consente di separare le variabili, isolerne una

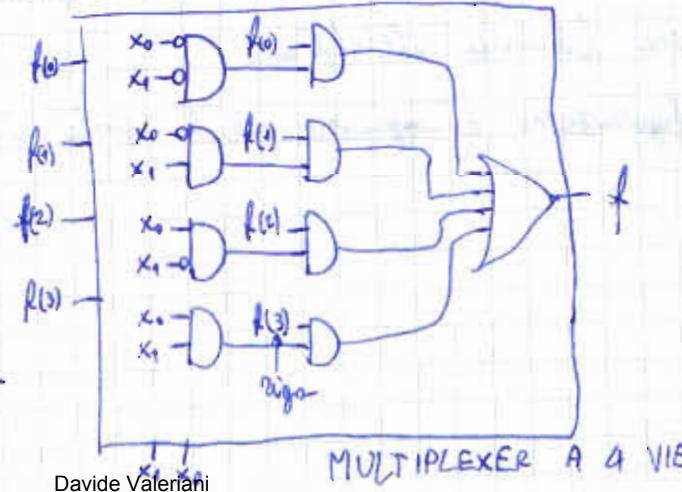
espressione da valutare interamente

Date le tabelle di verità

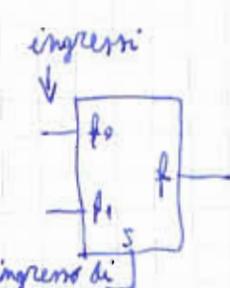
x	y	C _{in}	S	Cout
00	0	0	0	0
00	1	1	0	0
01	0	1	0	0
01	1	0	1	0
10	0	1	0	0
10	1	0	1	1
11	0	0	1	1
11	1	1	1	1

$$S = \bar{x}_0 \bar{C}_{in} \cdot 0 + \bar{x}_0 \bar{C}_{in} \cdot 1 + \bar{x}_1 \bar{y} \bar{C}_{in} \cdot 1 + \bar{x}_1 \bar{y} \bar{C}_{in} \cdot 0 + \\ + x_0 \bar{C}_{in} \cdot 1 + x_0 \bar{C}_{in} \cdot 0 + x_1 \bar{y} \bar{C}_{in} \cdot 0 + x_1 \bar{y} \bar{C}_{in} \cdot 1$$

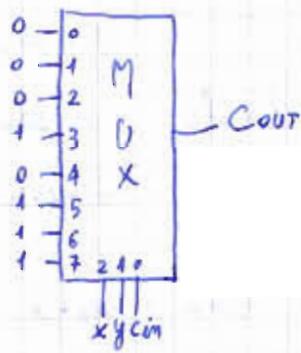
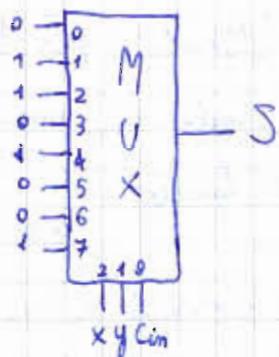
ESPRESSONE GENERALE DAL TEOREMA DI SHANNON



MULTIPLEXER



Il multiplexer ha n variabili di selezione o indirizzamento, 2^n ingressi dati o di programmazione e 1 uscita.



es.

$$\begin{aligned}
 f(X_1, X_2, X_3) &= X_1 \bar{X}_2 + X_3 X_2 = \bar{X}_1 (0 \cdot \bar{X}_2 + X_3 \cdot X_2) + X_1 (1 \cdot \bar{X}_2 + X_3 \cdot X_2) = \\
 &= \bar{X}_1 (\bar{X}_1 (0 \cdot 0 + X_3 \cdot 0) + X_1 (1 \cdot 0 + X_3 \cdot 0)) + X_2 (\bar{X}_1 (0 \cdot 1 + X_3 \cdot 1) + X_1 (1 \cdot 1 + X_3 \cdot 1)) = \\
 &= \bar{X}_1 \bar{X}_1 (0) + \bar{X}_1 X_1 (1) + X_2 \bar{X}_1 (X_3) + X_2 X_1 (X_3)
 \end{aligned}$$

ESPRESSIONI
GENERALI SP

REALIZZAZIONE
CON MUX

ESPRESSIONI
GENERALI PS

ESPRESSIONI
CANONICHE SP

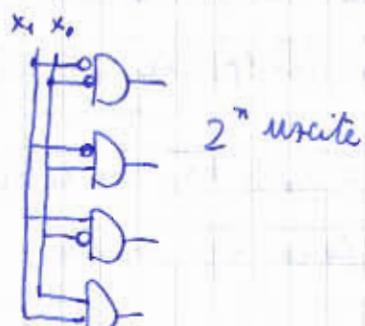
REALIZZAZIONE
CON DECODER

ESPRESSIONI
CANONICHE PS

DECODER

Se n ingressi è 2^n uscite.

Circuito universale. Mette a disposizione tutti i minterm della funzione.



SINTESI NOR \rightarrow PDS - PS

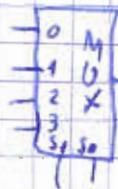
SINTESI NAND \rightarrow SDP - SP

DISTANZA DI HAMMING \rightarrow numero di bit che cambia nel passare da una configurazione binaria all'altra.

DCset \rightarrow configurazioni d'ingresso per le quali il valore dell'uscita è non specificato.

$x_3 \backslash x_2$	00	01	11	10
00	0	-	1	1
01	-	0	0	1
11	0	1	1	1
10	1	0	1	0

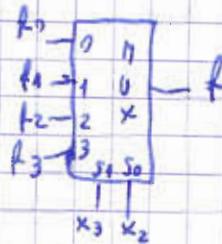
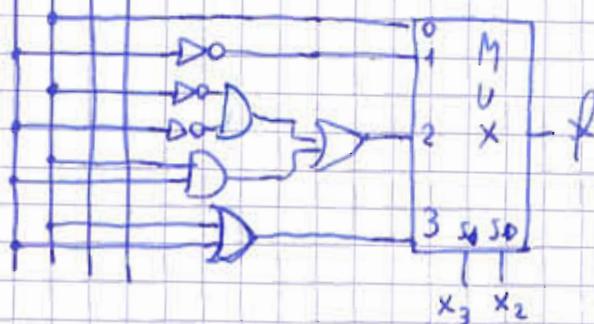
Ginteri' con MUX e 4 varie + qualcosa.



Il MUX è l'ultimo elemento del circuito. Due variabili di selezione (ad es. x_2, x_3) e 2^k funzioni di $n-k$ variabili.

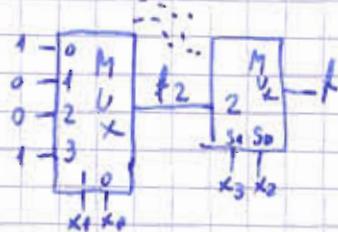
$$f_0 = x_1 \quad f_1 = \bar{x}_0 \quad f_3 = x_1 + x_0 \quad f_2 = \bar{x}_1 \bar{x}_0 + x_1 x_0$$

$x_0 \ x_1 \ x_2 \ x_3$



Per fare f_2 con un

MUX



ALGORITMO DI QUINE-MCCLUSKEY

Due fasi:

- ricerca degli implicanti primi → applico mense $a \cdot z + \bar{a} \cdot z = z(a+\bar{a}) = z$.
- ricerca della copertura ottima

Ricerca dell'insieme dei mintermi e riduzione.

ES.

$$f(a,b,c,d) = \sum [1, 3, 7, 8, 9, 12, 13, 15]$$

\overline{ab}	00	01	11	10
00	0	1	1	0
01	0	0	1	0
11	1	1	1	0
10	1	1	0	0

$a \ b \ c \ d$	$a \ b \ c \ d$
1 0 0 0 1 *	1/3 0 0 -1 ←
8 1 0 0 0 *	1/9 -0 0 +1 ←
3 0 0 1 1 *	8/9 1 0 0 -*
9 1 0 0 1 *	3/7 0 -1 1 ←
12 1 1 0 0 *	9/13 1 -0 1 *
7 0 1 1 1 *	12/13 1 1 0 -*
13 1 1 0 1 *	7/15 -1 1 1 ←
- - - -	13/15 1 1 -1 ←
15 1 1 1 1 *	- - - -

$a \ b \ c \ d$
1 - 0 - me ignoro
1 - 0 - una che non interessa

IMPICANTI PRIMI

$\bar{a}\bar{b}d, \bar{b}\bar{c}d, \bar{c}\bar{d}, bcd, abd, a\bar{c}$ Quelli non marcati.

SECONDA FASE \rightarrow tabella di coperture: righe \rightarrow implicanti primi, colonne \rightarrow minterm

Ci sono due criteri per semplificare la tabella:

- CRITERIO DI ESSENZIALITÀ \rightarrow sceglie le colonne che hanno una sola X: quel minterm andrà preso, cancellando tutte le colonne che la riga copre.
- CRITERIO DI DOMINANZA \rightarrow elimina le righe coperte già da altre righe o le colonne che coprono altre colonne.

Se nonostante queste semplificazioni otteniamo una tabella non siamo di fronte a una tabella CICLICA

FES.

	1	3	7	8	9	12	13	15	Righe ordinate per costo
$\bar{a}\bar{c}$	-	-	-	-	X	X	X	-	IMPLICANTI PRINCIPALI = $\{\bar{a}\bar{c}\}$
$\bar{a}b\bar{c}d$	X	-	-	-	-	-	-	-	
$a\bar{b}d$.	-	-	X	X	X	X	-	
$b\bar{c}d$	-	X	-	-	-	-	-	-	
$\bar{a}\bar{c}d$	X	X	-	-	-	-	-	-	
$\bar{a}\bar{b}\bar{d}$	X	X	-	-	-	-	-	-	

	1	3	7	15
P_2	X	-	-	-
P_3	-	-	X	-
P_4	-	-	X	X
P_5	-	X	-	-
P_6	X	X	-	-

$$f = \bar{a}\bar{c} + b\bar{c}d + \bar{a}\bar{b}\bar{d}$$

	1	3	7	15	IMPL. PRINCIPALI
P_4	X	-	-	-	$\{\bar{a}\bar{c}, b\bar{c}d\}$
P_5	-	-	X	X	
P_6	X	X	-	-	
P_5	-	X	-	-	
P_6	-	X	-	-	

	C_1	C_2	C_3	C_4	C_5
r_1	X	X	-	-	-
r_2	-	-	X	X	-
r_3	-	X	-	X	X
r_4	X	-	X	-	X

TABELLA CICLICA \rightarrow faccio delle prove

Prendo r_1 come implicante

Non prendo r_1 (considero dominato)

	C_1	C_2	C_3
r_2	X	X	-
r_3	-	X	X
r_4	X	-	X

	C_1	C_2	C_3	C_4	C_5
r_2	X	X	-	-	-
r_3	-	X	X	-	-
r_4	X	-	X	X	X

encora adice

METODO DI PATRICK

$$P = (r_1 + r_4) \cdot (r_1 + r_3) \cdot (r_2 + r_4) \cdot (r_2 + r_3) \cdot (r_3 + r_4) \leftarrow \text{colonne ...}$$

alla fine mi dice quali colonne devo prendere...

$$= r_1 r_2 r_3 + r_1 r_2 r_4 + r_1 r_3 r_4 + r_2 r_3 r_4 + \underline{r_3 r_4} \quad \text{prendo le meno costose!} \\ \text{oppure}$$

23/10/2008

ESEMPIO S.P. DI RUINE-MCCCLUSKEY

$$z = \sum m_a (0, 2, 4, 6, 7, 9, 11, 15)$$

0 0 0 0 0 *	0/2 0 0 - 0 *	0/2/4/6 0 -- 0
2 0 0 1 0 *	0/4 0 - 0 0 *	0/4/2/6 0 -- 0
1 0 1 0 0 *	2/6 0 - 1 0 *	
6 0 1 1 0 *	1/6 0 1 - 0 *	
9 1 0 0 1 *	6/7 0 1 1 -	
11 0 1 1 1 *	3/11 1 0 - 1 ..	
15 1 1 1 1 4	7/15 1 1 1 1	

$$z = \bar{a}bc + a\bar{b}d + bcd + acd + \bar{a}\bar{d}$$

Implicanti primi

P_5 è essenziale!

$$\mathcal{L} = \{P_5\}$$

	0	2	4	6	7	9	11	15
$P_5 \bar{a}\bar{d}$		*	*	*	*			
$P_2 a\bar{b}d$						*	*	
$P_3 bcd$					x		x	
$P_4 acd$						x	x	
$P_1 \bar{a}bc$					x	x		

	7	9	11	15
P_2		*	*	
P_3	x		x	
P_4		x	x	
P_1	x			1

$$\mathcal{L} = \{P_5, P_2\}$$

$$7 \quad 15$$

$P_3 \quad x \quad x \leftarrow \text{domina}$

$$\mathcal{L} = \{P_5, P_2, P_3\}$$

$$z = a\bar{b}d + bcd + \bar{a}\bar{d}$$

$$\text{N.O.P.} = 4$$

$$\text{N.I.N.} = 11$$



$$\text{N.L.E.T.} = 8$$

ESEMPIO P.S.

$$Z = \prod_{i=1}^8 M_i (1, 3, 5, 8, 10, 12, 13, 14)$$

FASE 1

	$a b c d$	$a b c d$	$a b c d$
1	0 0 0 1 *	1/3 0 0 - 1	8/10/11/12 1 - - 0
8	- 1 0 0 0 *	1/5 0 - 0 1	(8/12/10/14 1 - - 0)
3	0 0 1 1 *	8/10 1 0 - 0 *	
5	0 1 0 1 *	8/12 1 - 0 0 *	
10	1 0 1 0 *	5/13 - 1 0 1	
12	1 1 0 0 *	10/14 1 - 1 0 *	
13	1 1 0 1 *	12/13 1 1 0 -	
14	1 1 1 0 *	12/14 1 + - 0 *	

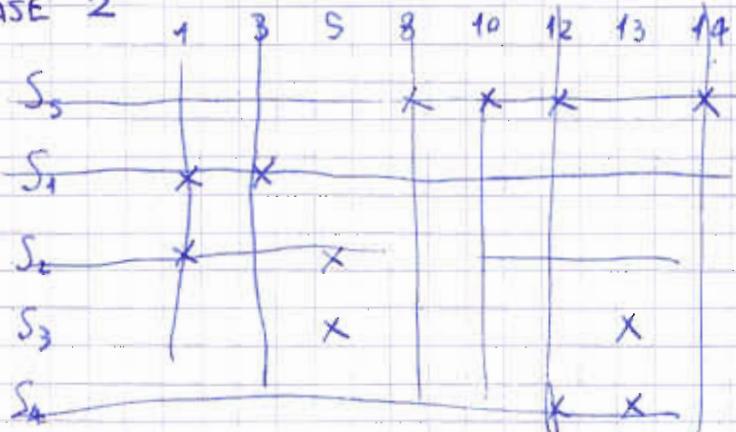
Implicati primi

$$S_1 = a + b + \bar{d} \quad S_4 = \bar{a} + \bar{b} + c$$

$$S_2 = a + c + \bar{d} \quad S_5 = \bar{a} + d$$

$$S_3 = \bar{b} + c + \bar{d}$$

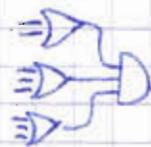
FASE 2



$$L = \{S_5\}$$

$$C = \{S_5, S_1\} \quad S_3 \text{ domina}$$

$$C = \{S_5, S_1, S_3\}$$



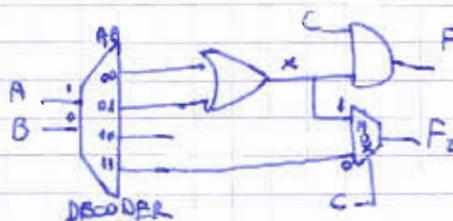
$$\begin{array}{|l|l|} \hline N_{op} & = 4 \\ \hline N_{in} & = 11 \\ \hline \end{array}$$

$$\begin{array}{|l|} \hline N_{tot} = 8 \\ \hline \end{array}$$

$$\begin{aligned} Z &= (\bar{a} + d) \cdot (a + b + \bar{d}) \cdot (\bar{b} + c + \bar{d}) \\ &= (\bar{a} + d) \downarrow (a \downarrow b \downarrow \bar{d}) \downarrow (\bar{b} \downarrow c \downarrow \bar{d}) \end{aligned}$$

Nel caso in cui alcuni valori non sono specificati (DON'T CARE) li riporto nella FASE 1 per minimizzare ma NON nella FASE 2.

EJ. ESAME 16/2/2005

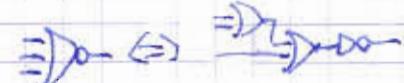


OR, NOR a 2 ingressi

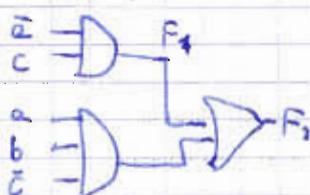
$$x = \bar{a}b + \bar{a}\bar{b} = \bar{a}$$

$$F_1 = \bar{a}c$$

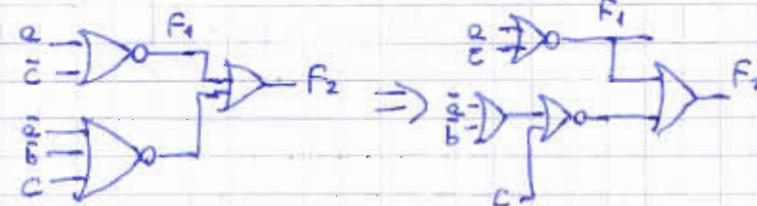
$$F_2 = (ab)\bar{c} + c \cdot \bar{a} = ab\bar{c} + \bar{a}c$$



il momento



applico de Morgan



Per circuiti a più uscite, non sempre le sintesi ottimali si ottengono considerando solo gli implicati primi.

La copertura ottima di una funzione multivariata passa attraverso gli implicati primi. Il metodo di Q-MC consente di minimizzare facilmente questi sistemi. Si aggiunge un'etichetta a fianco ad ogni minterm composto da n bit (con $n = n^*$ funzioni d'uscita) che vale 1 se tale minterm vale 1 per quella funzione. Si fa poi l'AND fra le etichette e se è uguale all'etichetta di uno

011 101
001 011 $\Rightarrow_{M_3} 0-1$ 001 nessuno marcato
Le due, viene marcato, altrimenti n

A casi per il risultato dell'AND

- tutti 0 \rightarrow scarto il minterm
- non coincide con nessuna etichetta \rightarrow non marco nessuno
- coincide con un'etichetta \rightarrow marco il minterm con quell'etichetta
- coincide con entrambe le etichette \rightarrow marco entrambi

Costruisco la tabella di copertura separando (nelle colonne) gli ONSET di una funzione da quelli dell'altra, non riportando quelle delle condizioni di indifferenza.
 \hookrightarrow anche ripetendoli per quelle comuni.

La dominanza tra colonne si può applicare solo all'interno delle stesse funzioni. L'essenzialità va applicata per funzione.

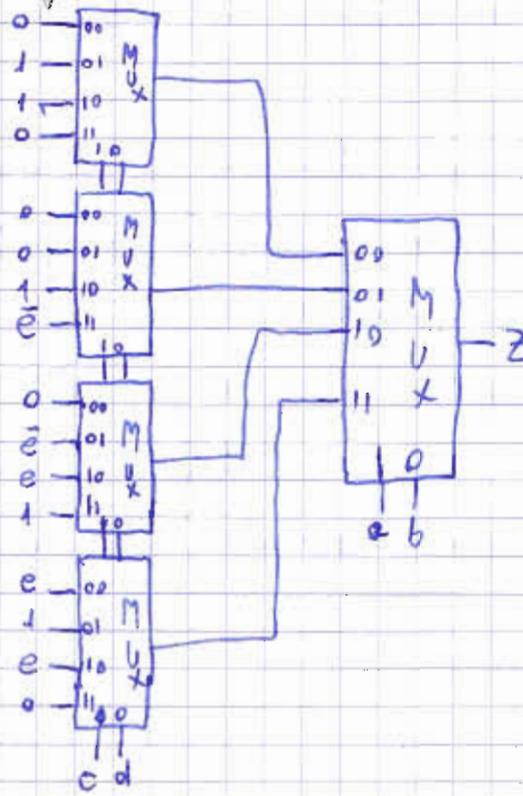
E5. Data una funzione combinatoria, sintetizzarla con MUX + 2 variabili di selezione.

$$C_1 C_0 = 0$$

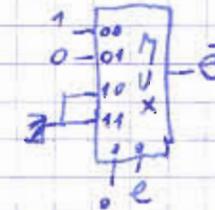
ab	00	01	11	10	ab	00	01	11	10	a, b = 0
00	0	1	-	1	00	0	1	0	-	
01	0	0	1	1	01	0	-	0	1	
11	0	-	0	0	11	1	1	-	1	
10	0	1	1	0	10	0	0	1	1	

$$e = 0$$

$$e = 1$$



Per ottenere e ho bisogno di un altro multiplexer



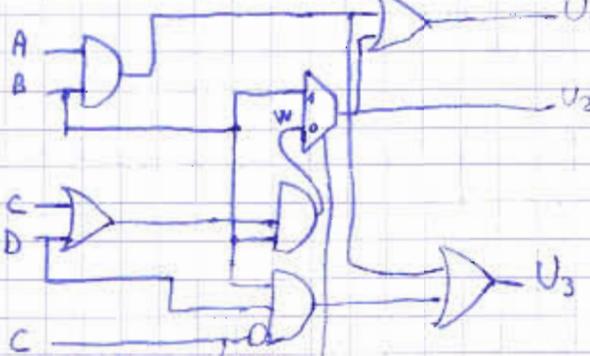
TEOREMA

Dato una funzione di n variabili ed un MUX ad $n-1$ variabili di selezione, nel caso peggiore servirà un NOT per completare le sintesi.

Dato una funzione di n variabili ed un MUX ad $n-2$ variabili di selezione, nel caso peggiore serviranno 16 componenti.

28/10/08

E.S. D3 agosto 2007



$$U_2 = BC + \bar{C}W = BC + \bar{C}(B \cdot (C+D)) = BC + B\bar{C}D$$

$$U_1 = AB + U_2 = AB + BC + B\bar{C}D$$

$$U_3 = AB + B\bar{C}\bar{D}$$

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

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

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

$$U_1 = AB + BC + BD$$

$$U_2 = BD + BC$$

$$U_3 = AB + B\bar{C}\bar{D}$$

$N_{op} = 10$ senza condizionamento

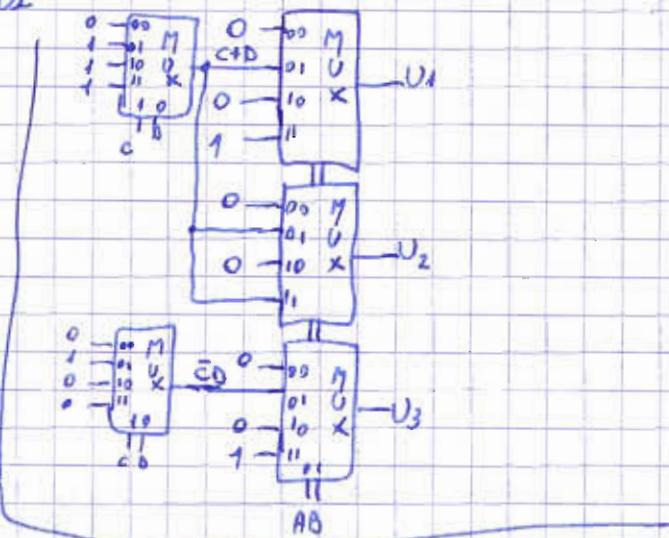
$N_{op} = 7$ con condizionamento

$N_{op} = 6$ nell'espressione non ridotta

DISPLAY A 7 SEGMENTI,

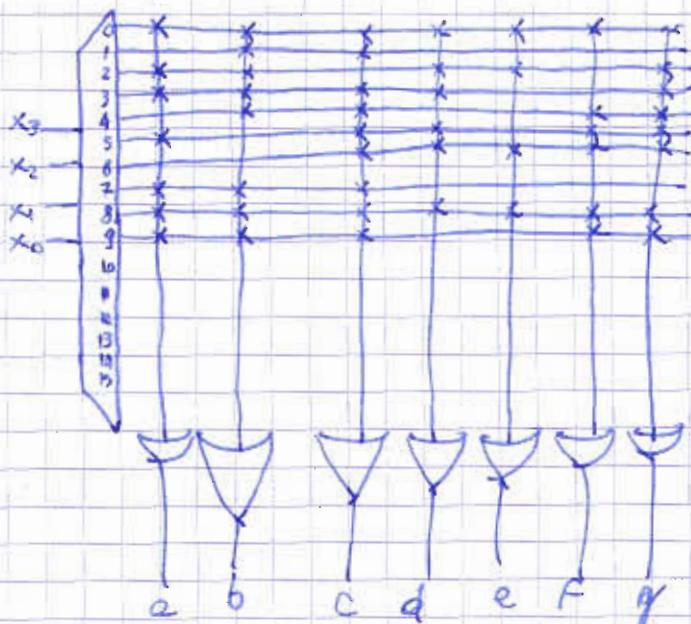


$x_3x_2x_1x_0$	U	a	b	c	d	e	f	g
0000	0	1	1	1	1	1	0	
0001	1	0	1	1	0	0	0	
0010	2	1	1	0	1	1	0	1
0011	3	1	1	1	1	0	0	1
0100	4	0	1	1	0	0	1	1
0101	5	1	0	1	1	0	1	1
0110	6	0	0	1	1	1	1	1
0111	7	1	1	1	0	0	0	0
1000	8	1	1	1	1	1	1	1
1001	9	1	1	1	0	0	1	1



Quanti MUX a 2 var. servono nel caso peggiore
19 : 7 per le uscite e 12 per le configurazioni
non banali degli ingressi.

Pintesi con decoder



RETI A 7 LIVELLI

x_3	x_2	x_1	x_0	00	01	11	10
00	00	01	11	1	1	1	
01	1	1	1	1	1	1	1
11	1	1	1	1	1	1	1

$$f(a, b, c, d) = b\bar{c} + ad + \bar{c}d + ab = \bar{c}(b+d) + a(b+d) = (a+\bar{c})(b+d)$$

$$N_{op} = 4 + 1 = 5$$

$$N_{op} = 3$$

$$N_{mux} = 4 \times 2 + 1 \times 4 = 12$$

$$N_{mux} = 6$$

	00	01	11	10
00	1	1		
01		1	1	
11	1	1		
10	1	1	1	

$N_{op} = 9$ applicando la distributiva

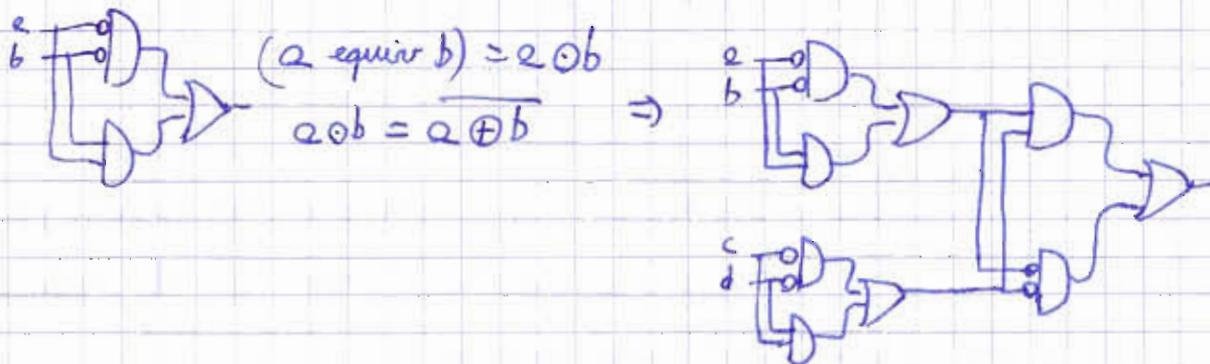
$N_{max} = 40!!$

b)

$$\bar{a}\bar{b}\bar{c}\bar{d} + \bar{a}\bar{b}cd + \bar{a}b\bar{c}d + \bar{a}bc\bar{d} + ab\bar{c}\bar{d} + abc\bar{d} + \bar{a}\bar{b}\bar{c}d + ab\bar{c}\bar{d} = \dots$$

$$(\bar{a}\bar{b} + ab)(\bar{c}\bar{d} + cd) + (\bar{a}b + ab)(\bar{c}d + cd) \quad N_{op} = 15 \quad N_{max} = 30 \leftarrow \begin{matrix} n^{\text{o}} \text{ morselli} \\ \text{d'ingresso} \end{matrix}$$

30/10/08



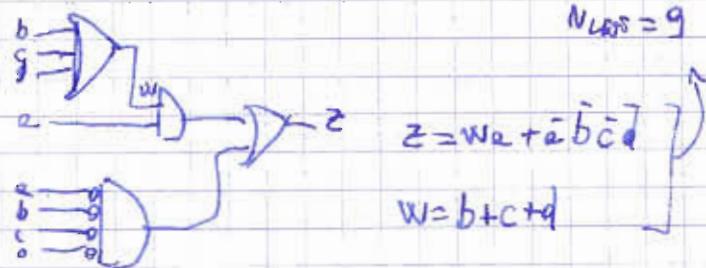
	00	01	11	10
00	1			
01	1	1	1	1
11				
10	1	1	1	1

$$f = ab + ac + ad + \bar{a}\bar{b}\bar{c}\bar{d}$$

$N_{op} = 5$
 $N_n = 14$ 2 livelli
 $N_{Lors} = 10$

$$f = a(b+c+d) + \bar{a}\bar{b}\bar{c}\bar{d} \quad SPS$$

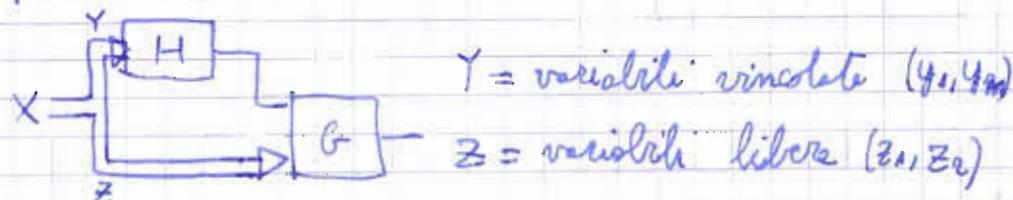
$N_{op} = 4$
 $N_n = 14$ 3 livelli
 $N_{Lors} = 9$



SCOMPOSIZIONE SEMPLICE DISGIUNTIVA

Data una funzione booleana $f(x_1, \dots, x_n)$ di n variabili:

- partizioniamo l'insieme delle variabili X in due sottoinsiemi disgiunti Y e Z
- identifichiamo (se possibile) due funzioni G e H tali che $f(x_1, \dots, x_n) = H(Y) + G(Z)$



Non per ogni partizione degli ingressi è disponibile una scomposizione semplice disgiuntiva.

MAPPE DI PARTIZIONE \rightarrow mappe di Karnaugh in cui le configurazioni delle variabili vincolate Y contraddistinguono le colonne.

Ogni riga della mappa di partizione è una funzione di Y.

La partizione è possibile se le molteplicità di colonne (n° di colonne con diversa configurazione di 0 e 1) è non superiore a 2.

ES:

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

$$Y = \{a, b\}$$

$$Z = \{c, d\}$$

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

$$h \rightarrow H$$

$$f \rightarrow G$$

$$z = \bar{c}\bar{d}h + \bar{c}d\bar{h} + cd\bar{h} + c\bar{d}h$$

$$\leftarrow h = \bar{a}b + ab \quad \text{due tipi di colonne}$$

$$2^{\text{a}} \text{ colonna}: h=0 \text{ se variabili diverse}$$

$$1^{\text{a}} \text{ colonna}: h=1 \text{ se variabili uguali}$$

ES. 22/12/05

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

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

$$h = \bar{b} + \bar{d}$$

$$h = 1$$

$$h = 0$$

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

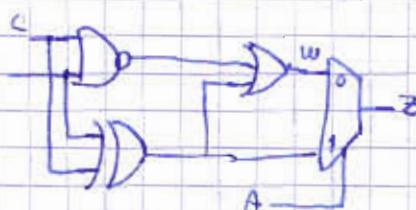
$$F = \bar{A}h + Ch + A\bar{C}h$$

$$F = (\bar{A} + C)(\bar{C} + h)(\bar{A} + C + h) =$$

$$= (\bar{A} + h)(\bar{C} + h)(\bar{A} + C + h)$$

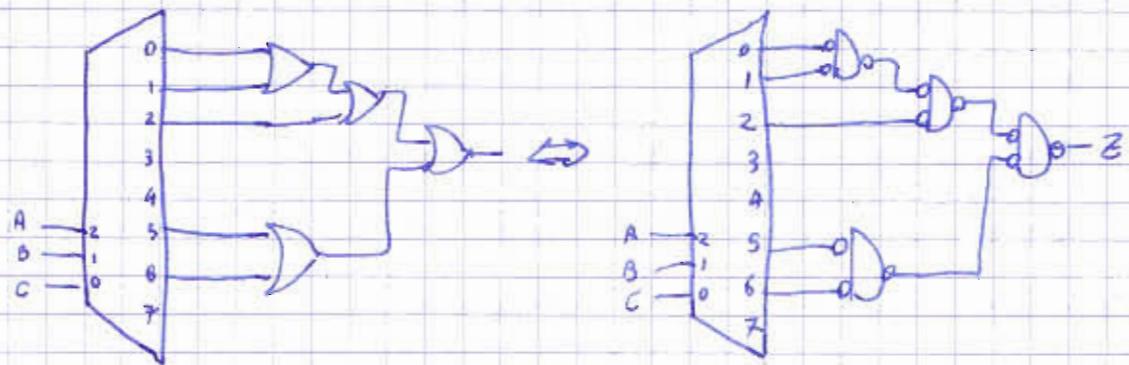
AB e CD non possono essere variabili vincolate perché 2 colonne diverse

Potremo prendere $Y = \{A, B\}$ mettendo 1 in due - per avere 3 colonne uguali



$$z = \bar{A}w + A(\bar{B}\bar{C} + \bar{B}C) = \bar{A}((\bar{C} \cdot \bar{B}) + (\bar{B}\bar{C} + \bar{B}C)) + A\bar{B}\bar{C} + A\bar{B}C = \\ = \bar{A}(\bar{B} + \bar{C} + \bar{B}\bar{C} + \bar{B}C) + A\bar{B}\bar{C} + A\bar{B}C = \bar{A}\bar{B} + \bar{A}\bar{C} + A\bar{B}\bar{C} + A\bar{B}C$$

A\B	00	01	11	10
0	1	1	0	1
1	0	1	0	1



$$\text{Dotta } f = ab + bc$$

	00	01	11	10
0	0	0	1	1
1	1	1	1	1

IMPICANTI
BASICI

$$I_1 = ab$$

$$I_2 = b\bar{c}$$

$$I_3 = a\bar{c}$$

- Expansion

$$f = \bar{a}(a\bar{b} + b\bar{c}) + a(\bar{b}\bar{b} + b\bar{c}) =$$

$$= \bar{a}(b\bar{c}) + a(\bar{b} + b\bar{c}) =$$

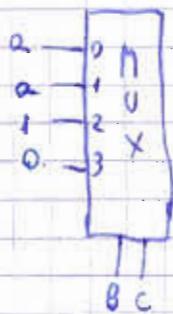
$$= \bar{b}(\bar{a}(a\bar{c}) + a(\bar{a} + a\bar{c})) + b(\bar{a}(1\bar{c}) + a(\bar{1} + a\bar{c})) =$$

$$= \bar{b}[(\bar{a}\cdot 0) + (\bar{a}\cdot 1)] + b[(\bar{a}\cdot \bar{c}) + (a\cdot \bar{c})] =$$

$$= \bar{c}(\bar{b}(\bar{a}\cdot 0 + a\cdot 1) + b(\bar{a}\cdot \bar{a} + a\cdot 1)) + c(\bar{b}(\bar{a}\cdot 0 + a\cdot 1) + b(\bar{a}\cdot \bar{a} + a\cdot 1)) =$$

$$= \bar{a}\bar{b}\bar{c}\cdot 0 + \bar{c}\bar{b}a\cdot 1 + \bar{c}b\bar{a}\cdot 1 + \bar{c}ba\cdot 1 + cb\bar{a}\cdot 0 + cb\bar{a}\cdot 1 +$$

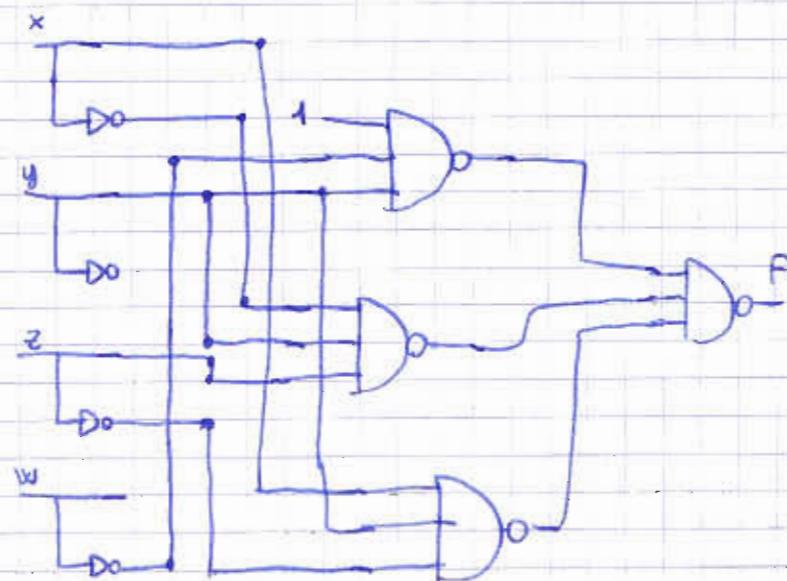
$$+ cb\bar{a}\cdot 0 + cb\bar{a}\cdot 0.$$



04/11/08

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

Dovrò usare solo queste porte!



x	y	w	00	01	11	10
0	0	0	0	0	0	0
1	0	0	1	1	1	1

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

$$= (y \uparrow \bar{w}) \uparrow (\bar{x} \uparrow y \uparrow z) \uparrow (x \uparrow y \uparrow \bar{z})$$



NOT con
solì NAND

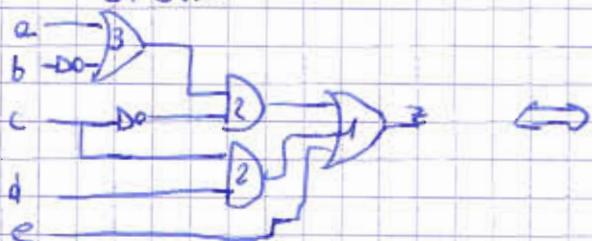


AND con soli
NAND



OR con soli
NAND

^{1 2 3}
SPSP...



b

c

d

e



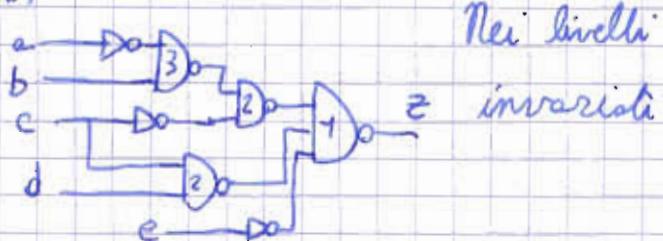
y

z

w

Che si semplifica avendo anche

i NOT

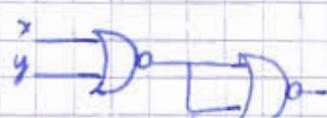


Nei livelli pari i complementi rimangono

invariati



NOT con
solì NOR



OR con soli
NOR



AND con soli
NOR

SPSP... si convertono meglio in soli NAND

PSPS... si convertono meglio in soli NOR

$R = x \cdot (y + (z \cdot w)) \downarrow K$ aumenta il livello ogni "(" e decremente ogni ")".

livelli 1 2 3 4

I letterali dei livelli dispari vanno complementati!

$$R = \overline{x} \downarrow (y \downarrow (\overline{z} \downarrow \overline{w})) \downarrow K$$

$$Z = x \uparrow \overline{y} \uparrow w \quad \text{espressione SP a 4 livelli (solo la somma)} \Rightarrow Z = \overline{x} \uparrow y \uparrow \overline{w}$$



espressione PS con un solo livello di somma

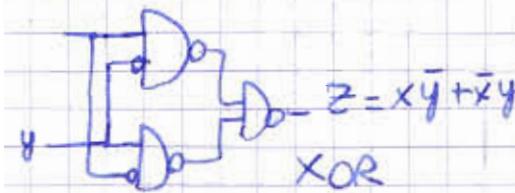
$$Z = (x \uparrow \overline{y} \uparrow w) \downarrow o$$

$$z = x \bar{y} \bar{w}$$

$$\begin{array}{c} x \\ | \\ y \rightarrow a \\ | \\ \overline{w} \rightarrow b \end{array} \rightarrow z \Leftrightarrow \begin{array}{c} \overline{a} \\ | \\ \overline{a} \end{array} \rightarrow z \quad z = (\bar{x} \bar{y} \bar{w}) + 0 = (\bar{x} \uparrow \bar{y} \uparrow \bar{w}) \uparrow 1$$

$\Rightarrow \overline{a} \rightarrow z$

CONVERSIONE DA NAND A SP



La conversione da SPS... a NAND produce un livello in più. Da fare solo se richiesto. Lo stesso vale per la conversione da PSP... a NOR.

$$\begin{array}{c} x \\ | \\ y \end{array} \rightarrow \begin{array}{c} \overline{2} \\ | \\ \overline{3} \\ | \\ \overline{2} \end{array} \rightarrow z = \underset{2}{(x \uparrow (x \uparrow y))} \uparrow \underset{1}{(y \uparrow (x \uparrow y))} = \underset{3}{\text{Transformo quelle a livello}} \\ \text{dispari in normale} \\ = (x \cdot (\bar{x} + \bar{y})) + (y \cdot (\bar{x} + \bar{y})) \quad \text{NOR}$$

NAND

LIVELLO DISPARI	OPERATORI OR - LETTERALI COMPLEMENTATI
LIVELLO PARI	OPERATORI AND - LETTERALI DIRETTI

NOR

LIVELLO DISPARI	OPERATORI AND - LETTERALI COMPLEMENTATI
LIVELLO PARI	OPERATORI OR - LETTERALI DIRETTI

	DISPARI	PARI
NAND	OR	AND
NOR	AND	OR

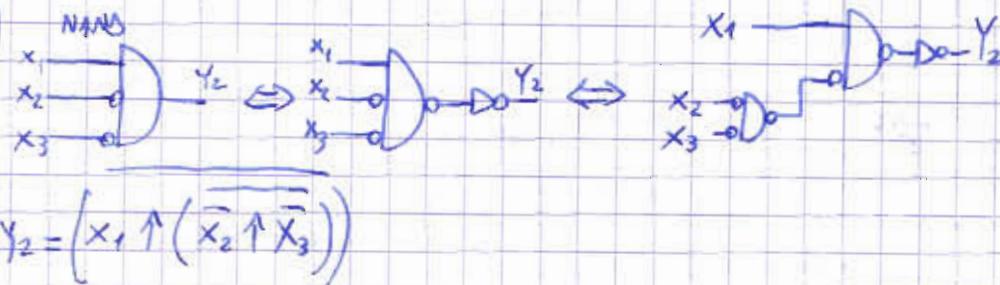
Complemento solo i letterali isolati, non le parentesi.

Realizzare con NAND e con NOR (4 circuiti) a 2 livelli

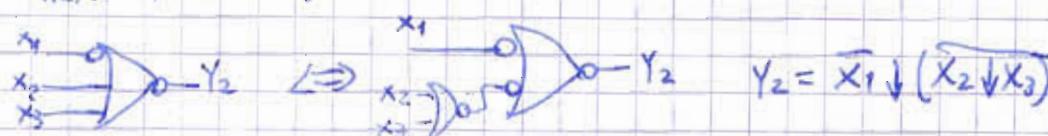
$$Y_1 = X_1 + X_2 + \bar{X}_3$$

$$Y_2 = X_1 \bar{X}_2 \bar{X}_3$$

2)

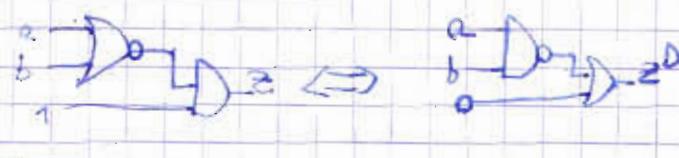


NOR PUSH THE BUBBLE



1) NAND - tre modi:

- 1) valutazione tramite mappa di Karnaugh $\text{NOR} \rightarrow \text{PS(P...)} \rightarrow \text{PS} \rightarrow \text{MAPPA}$
- 2) $\text{NOR} \rightarrow \text{PS(P..)}$ $Z = \text{espressione PS} + 0 \rightarrow \text{NAND}$ $n+1$ livelli! $\downarrow \text{SP}$
- 3) dualità $(E^D) = E$: sostituendo ogni operatore con il suo duale $\downarrow \text{NAND}$
e ogni costante con la duale per ottenere E^D .

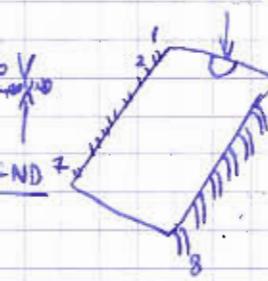


TEOREMA DI DUALITÀ
 $X^D = \overline{X(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n)}$

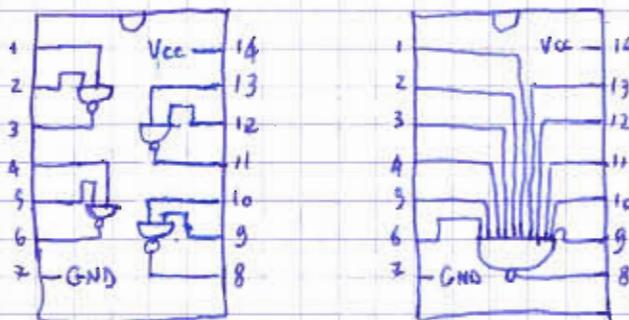
Per il teorema di dualità diventa



TIPO	ELEMENTI DA RAGGRUPPARE	CORRISPONDENZA
AND - OR	+, -	diretta
NAND - NAND	1, -	"
NAND - AND	0, -	"
AND - NOR	0, -	"
OR - AND	0, -	negata
NOR - NOR	0, -	"
NOR - OR	1, -	"



Packaged Logic



L'indice di costo

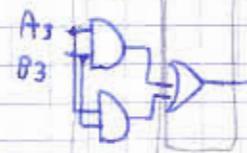
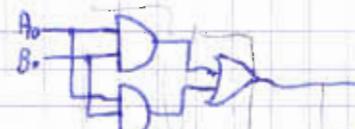
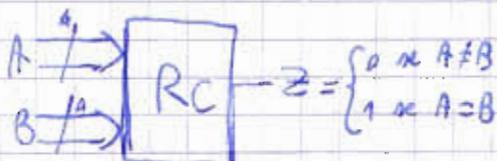
è il numero di Package.

stessa famiglia logica

stessi valori di tensione,

stesso standard,
stesso costo

Realizzare un comparatore di due operandi a 4 bit.



$N_0 = 13 \text{ (porti)}$

$N_m = 28 \text{ (moselli)}$

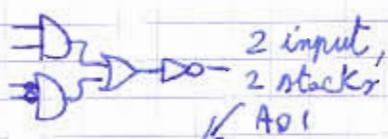
$N_{NET} = 12$

$C_{SSI} = 4 \text{ (package)}$

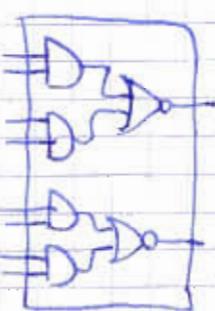
$C''_{SSI} = 3 + \frac{1}{2} \text{ IC SSI}$

l'ultimo package lo
uso a metà (5 piedini)

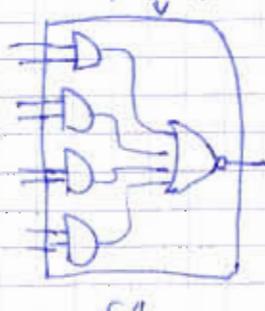
Medium Scale Integration

Componenti: AOI (And Or Invert) \rightarrow 2 o 3 livelli2 input,
2 stacks
 \checkmark AOI

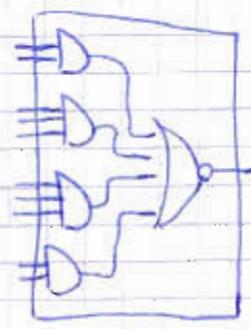
$Z = X_1 \oplus X_2 = X_1 X_2 + \overline{X_1} X_2 = \overline{X_1 X_2} + X_1 X_2$

2 input
4 stacks

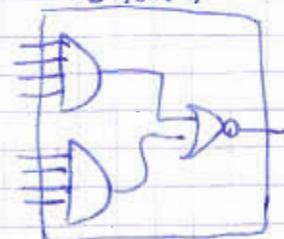
54



55



LS55A

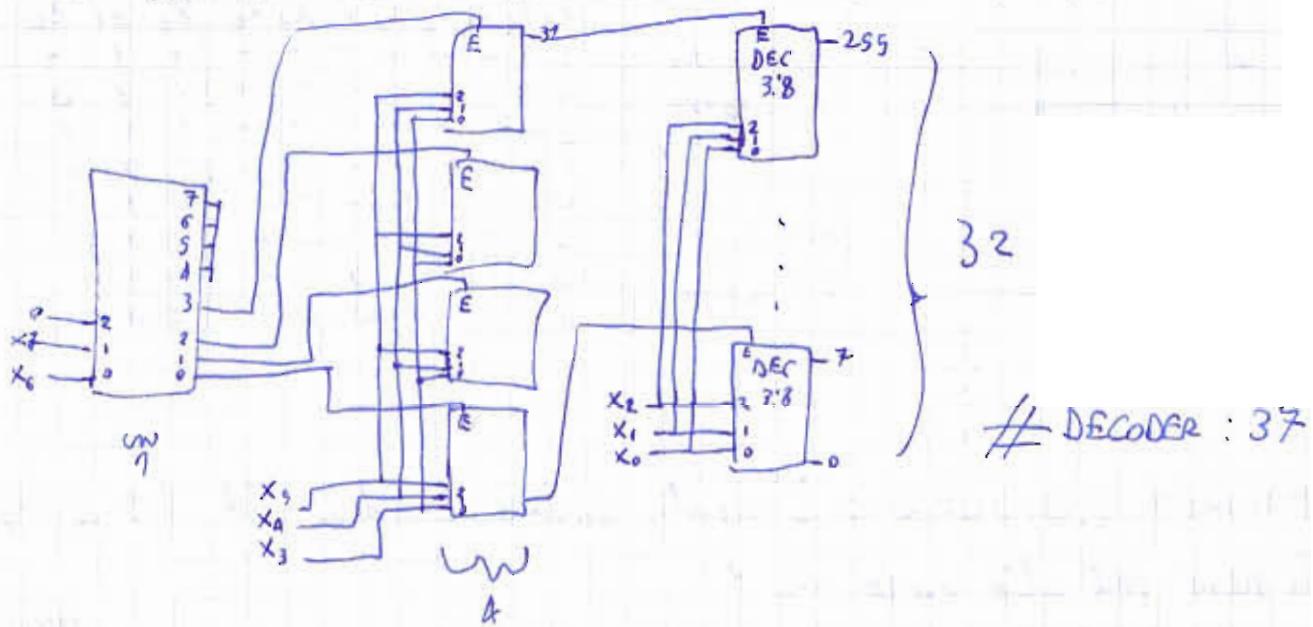
 \downarrow
 $A0I$ a input
2 stacks

$C''_{AOI} = 2 + \frac{1}{2} \text{ IC}$

$C_{AOI} = 3 \text{ IC SSI}$

- La sintesi di reti combinatorie può avvenire mediante:
- operatori elementari (AND, OR, NOT, NAND, NOR)
 - componenti mono-funzionali (CONVERTITORE, COMPARATORE)
 - componenti multi-funzionali (PAL, PLA, ALU)
 - componenti universali (DECODER+OR, MULTIPLEXER)

Dati a.B. decoder 3:8 sintetizzare un decoder da 8:256



$$\text{costo decoder } 4:16 = 2^4 \cdot 2^2 = 64$$

$$\text{costo matrice } 256 \text{ uscite} = 512$$

Il più grande beneficio lo si ottiene passando da un decoder a un livello a uno o due livelli.

La soluzione ottimale prevede di ottenere le stesse 256 uscite con il minor numero di ingressi. Tuttavia però cercare il decoder più quadrato possibile (n° ingressi = n° uscite).

MULTIPLEXER

$$n_m = 2^n(n+2) \quad \text{costo: } O(n \cdot 2^n)$$

ENCODER \rightarrow ingressi: 2^n e 1 uscita. Gintest a 1 livello di OR.

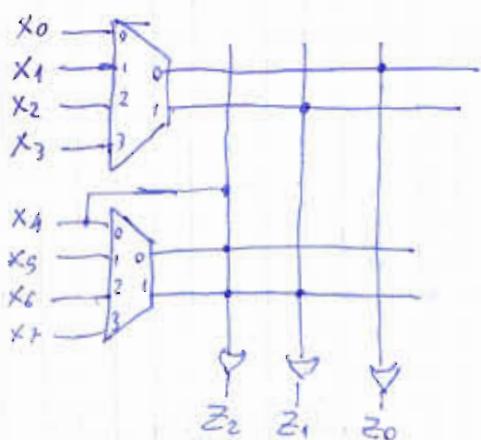
Esempio

Encoder 8:3

$$2^n \rightarrow n$$

$$8 \rightarrow 3$$

X_0	X_1	X_2	X_3	X_4	X_5	X_6	X_7	Z_2	Z_1	Z_0
1	0	0	0	0	0	0	0	0	0	$Z_2 = X_1 + X_3 + X_6 + X_7$
0	1	0	0	0	0	0	0	0	1	
0	0	1	0	0	0	0	0	0	1	$Z_1 = X_2 + X_3 + X_6 + X_7$
0	0	0	1	0	0	0	0	0	1	
0	0	0	0	1	0	0	0	1	1	$Z_0 = X_1 + X_3 + X_5 + X_7$
0	0	0	0	0	1	0	0	0	0	
0	0	0	0	0	0	1	0	1	1	
0	0	0	0	0	0	0	1	1	1	



$$Z_0' = X_1 + X_3$$

$$Z_1' = X_2 + X_3$$

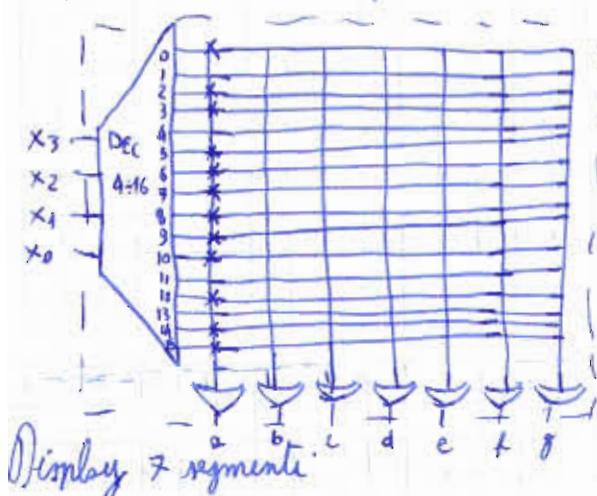
$$Z_0'' = X_5 + X_7$$

$$Z_1'' = X_6 + X_7$$

X_7	X_6	X_5	X_4	X_3	X_2	X_1	X_0	Z_2	Z_1	Z_0
1	-	-	-	-	-	-	-	1	1	1
0	1	-	-	-	-	-	-	1	1	0
0	0	1	-	-	-	-	-	1	0	1
0	0	0	1	-	-	-	-	1	0	0
0	0	0	0	1	-	-	-	0	1	1
0	0	0	0	0	1	-	-	0	1	0
0	0	0	0	0	0	1	-	0	1	1
0	0	0	0	0	0	0	1	0	0	0

PRIORITY ENCODER \rightarrow l'uscita indica l'indice del bit d'ingresso di peso più alto posto ad 1.

11/11/08



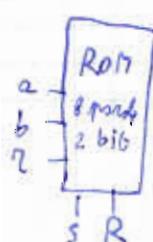
Display 7 segmenti

2 ROM a parole \times 2 uscite (2 ingressi)

n ingressi, 2^n uscite

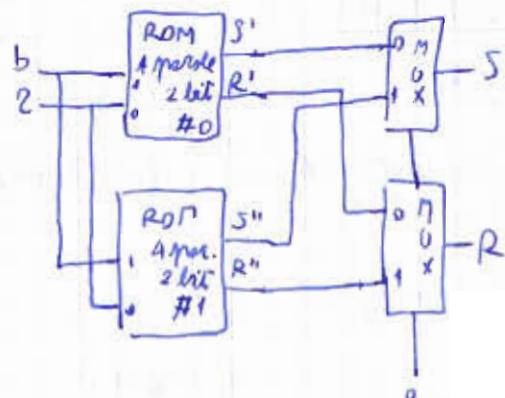
m OR con fan-in elevato $0(2^n)$

READ ONLY MEMORY - ROM



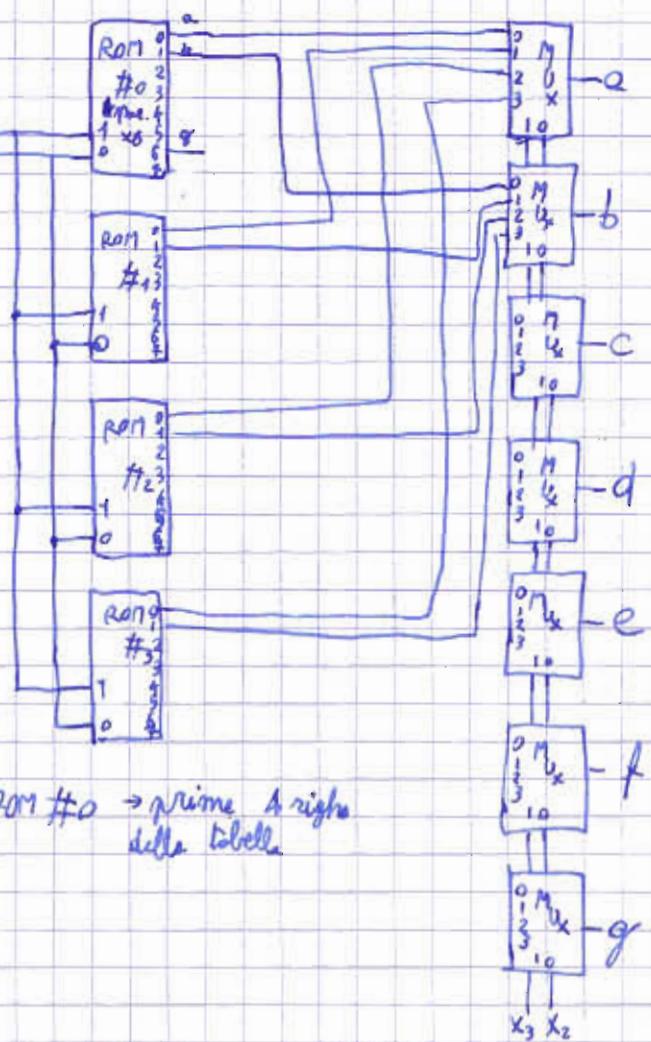
a	b	c	2b	5	R
0	0	0	000	0	0
0	0	1	001	1	0
0	1	0	010	1	0
0	1	1	011	0	1
1	0	0	100	1	0
1	0	1	101	0	1
1	1	0	110	0	1
1	1	1	111	1	1

$n=2$ $m=2$



Branche $\#1$ se $a=1$

conversione in serie.



7 mux perché
7 uscite

ROM #0 → prime 4 righe
della tabella

CONVERSIONS IN PARALLELO

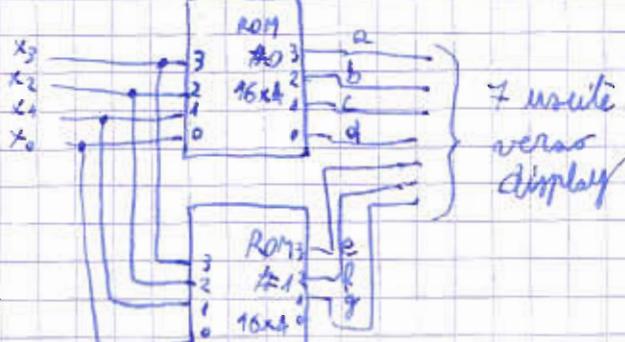
Disponiamo di: ROM 16x4 (16 parole, 4 uscite). Cerco la soluzione convertitrice BCD/HEX - 7 segmenti

	a	b	c	d	e	f	g
0000	1	1	1	1	1	0	
0001	0	1	1	0	0	0	
0010							1
0011							
0100							
⋮							
1000	1	1	1	1	1	1	
⋮							
1111	1	0	0	1	1	1	

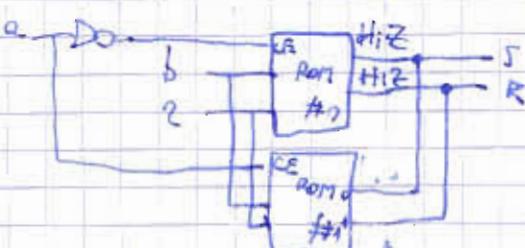
4 funzioni
di 4 versatilità

CHIP ENABLE (CE) → abilita la rom

Il FA. diventa

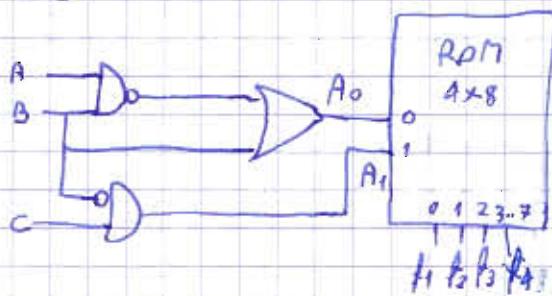


7 uscite
verso display



HIZ → alta impedenza.

ES.



Contenuto ROM

$A_1, A_0 \backslash f_1 f_2 f_3 f_4$	00	01	10	11
00	0 1 0 1	0 0 1 1		
01	1 0 1 0	1 0 0 0		
10	0 1 0 1	1 0 0 0		
11	1 0 1 0	1 0 0 0		

Intetizzare un circuito equivalente



$$A_0 = A \uparrow B + B = \bar{A} + \bar{B} + B = 1$$

$$A_1 = \bar{B}C$$

$$f_1 = \bar{A}_1 A_0 + A_1 A_0 = A_0$$

$$f_2 = \bar{A}_1 \bar{A}_0 + A_1 \bar{A}_0 = \bar{A}_0 \quad \Rightarrow \quad f_1 = f_3 = 1$$

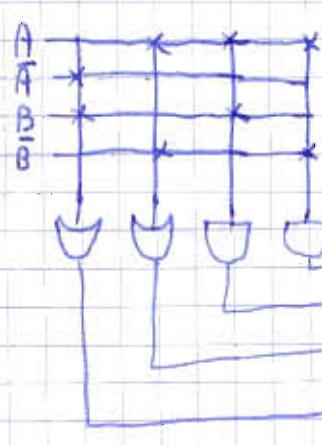
$$f_3 = f_4 = A_0$$

$$f_4 = f_2 = \bar{A}_0$$

$$0 \rightarrow f_2, f_4$$

$$+ \rightarrow f_1, f_3$$

ES.



$C > D$	00	01	11
00	0		
01	0	1	
11	0	1	0

$$SEL = CD$$

$$C > D ? \leftarrow D$$

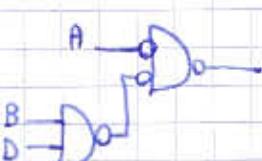
Provare U

$$U = \overline{SEL}(A+B) + SEL(\bar{A}+B)$$

$$C = \overline{AB} + \overline{AB} = \bar{A}$$

$$SEL = \bar{A}\bar{D}$$

$$U = \bar{A}\bar{D}(A+B) + \bar{A}D(\bar{A}+B) \Rightarrow U = (A+B)(A+\bar{B}) + \bar{A}\bar{D}(\bar{A}+B) = A+A\bar{B}+AD+\bar{B}D+\bar{A}\bar{D}+\bar{A}B\bar{D}$$



$A \setminus D$	00	01	11	10
0	1	1	0	1
1	1	1	1	1

$$U = A + \bar{B} + \bar{D}$$

Nelle matrici programmatili $K \ll 2^n$

Una PLA non può da sola coprire tutte le funzioni in quanto ha solo K prodotti e non copre tutti i minterm possibili nella configurazione peggiore (fusione recubica).

PLA \rightarrow programmabile sia il piano AND che quello OR

PAL \rightarrow programmabile solo il piano AND, più veloce ma meno potente e meno costoso

Realizzazione delle funzioni:

$$f_1 = X_1 X_2 + X_1 \bar{X}_3 + \bar{X}_1 \bar{X}_2 X_3$$

$$f_2 = X_1 X_2 + X_1 X_3 + \bar{X}_1 \bar{X}_2 X_3$$

con una PLA

- 3 ingressi: X_1, X_2, X_3

- 4 prodotti: $X_1 X_2, X_1 \bar{X}_3, \bar{X}_1 \bar{X}_2 X_3, X_1 X_3$

- 2 somme: $X_1 X_2 + X_1 \bar{X}_3 + \bar{X}_1 \bar{X}_2 X_3, X_1 X_2 + X_1 X_3 + \bar{X}_1 \bar{X}_2 X_3$

- 2 uscite: f_1, f_2

PAL BIPOLAR

Realizzare le funzioni

$$f_1 = X_1 X_2 \bar{X}_3 + \bar{X}_1 X_2 X_3$$

$$f_2 = \bar{X}_1 \bar{X}_2 + X_1 X_2 X_3$$

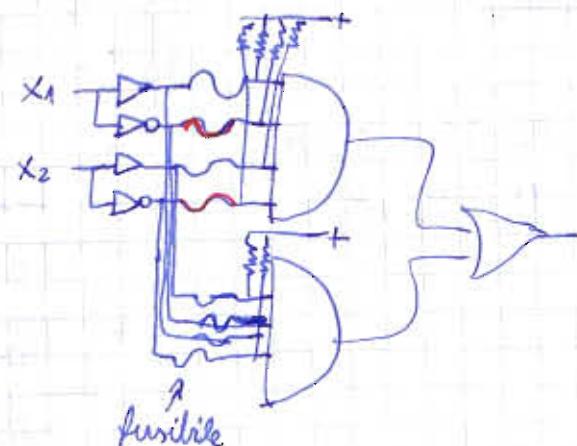
su una PAL

3 ingressi

4 termini prodotto

2 termini somma = 2 ingressi

2 uscite



Bruciando i fusibili in rosso, \bar{X}_1 e \bar{X}_2 non entrano (circuito aperto) e gli ingressi degli AND vengono portati a 1 dal + pull-up.

\rightarrow tutti i fusibili sono lasciati.

- ROM \rightarrow decoder al posto del piano AND.

PAL 14H8

$1 \leftarrow n^{\circ}$ ingressi

$8 \rightarrow n^{\circ}$ uscite

24 piedini: 14 ingressi, 8 uscite, 2 piedini di alimentazione.

PAL 16L8

↳ uscita a livello basso, invertito
20 piedini perché alcuni piedini usati come I/O



E	T	
0	0	tutti aperti
0	1	tutti accesi
1	1	tutti accesi funs. normale
1	0	

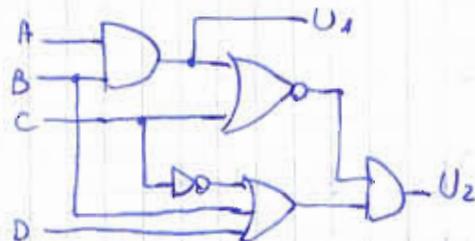
x ₁	x ₂	x ₃	x ₄	00	01	11	10
00	1	0	1	1	0	1	1
01	0	1	1	0	1	1	0
11	-	-	-	-	-	1	1
10	1	1	0	1	1	0	-

$$e = \bar{x}_2 \bar{x}_3 + x_1 x_0 + x_2 x_0 + x_3$$

$$\bar{e} = \bar{x}_3 \bar{x}_2 \bar{x}_1 x_0 + x_2 \bar{x}_0$$

ESERCIZIO D'ESAME

Dato una rete logica trovare una rete equivalente sulla PLA rotta.
Riportare anche la programmazione della PLA, evidenziando i minterm selezionati tramite delle "X", gli ingressi e le uscite.

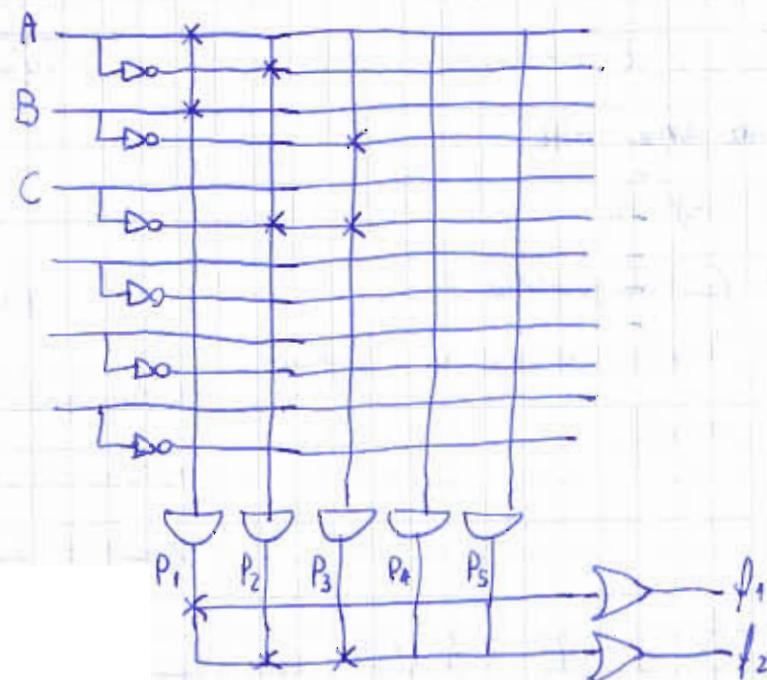


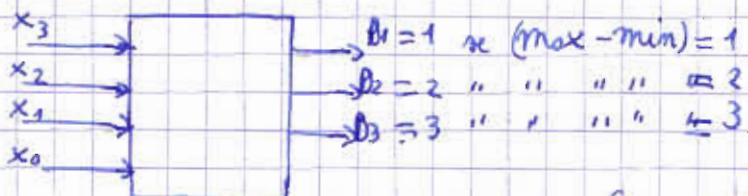
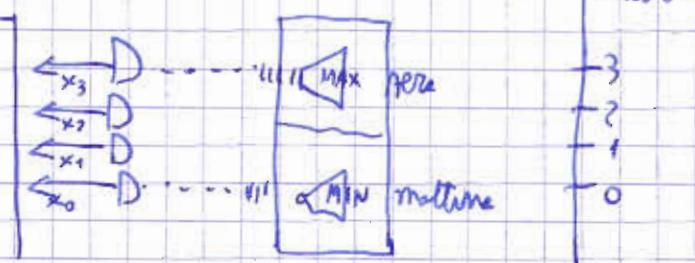
$$U_1 = A \cdot B$$

$$\begin{aligned} U_2 &= (\bar{A}B + C) \cdot (B\bar{C} + D) = (\bar{A}B \cdot \bar{C}) \cdot (B + \bar{C} + D) = \\ &= (\bar{A} + \bar{B}) \cdot \bar{C} \cdot (B + \bar{C} + D) = (\bar{A}\bar{C} + \bar{B}\bar{C})(B + \bar{C} + D) = \\ &= \bar{A}\bar{B}\bar{C} + \bar{A}\bar{C} + \bar{A}\bar{C}D + B\bar{B}\bar{C} + \bar{B}\bar{C} + \bar{B}\bar{C}D = \bar{A}\bar{C} + \bar{B}\bar{C} \end{aligned}$$

$$P_1 = AB \quad P_2 = \bar{A}\bar{C} \quad P_3 = \bar{B}\bar{C}$$

$$U_1 = P_1 \quad U_2 = P_2 + P_3$$



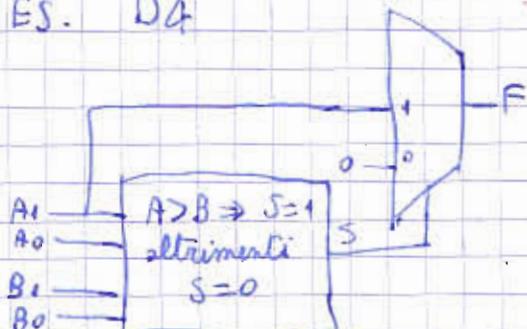


x_3	x_2	x_1	x_0	D_1	D_2	D_3
0	0	0	0	-	-	-
0	0	0	1	-	-	-
0	0	1	0	-	-	-
0	0	1	1	min 0 max 1	1 0 0	0
0	1	0	0	-	-	-
0	1	0	1	min 0 max 2	0 1 0	0
0	1	1	0	min 1 max 2	1 0 0	0
0	1	1	1	-	-	-
1	0	0	0	min 0	-	-
1	0	0	1	min 0 max 3	0 0 1	0
1	0	1	0	min 1 max 3	0 1 0	0
1	0	1	1	min 2	-	-
1	1	0	0	min 2 max 3	1 0 0	0
1	1	0	1	-	-	-
1	1	1	0	-	-	-
1	1	1	1	-	-	-

Chiede la distanza tra i galleggianti.

MIN e MAX

ES. D4



- Trovare una rappresentazione di F
- Confrontare costi di PdS e SdP con criterio dei letterali
- Trovare implementazione di F con MUX

$B_1 B_0$	$A_1 A_0$	00	01	11	10
B_1	0	0	1	1	0
B_0	0	1	0	0	1
		0	1	1	0
		1	0	0	1
		1	1	0	0

$$S' = \bar{B}_1 \bar{B}_0 A_0 + \bar{B}_0 A_1 A_0 + \bar{B}_1 A_1$$

$$S'' = (A_0 + A_1) \cdot (\bar{B}_0 + A_1) \cdot (\bar{B}_1 + A_1) \cdot (\bar{B}_0 + \bar{B}_1) / (\bar{B}_1 + A_0)$$

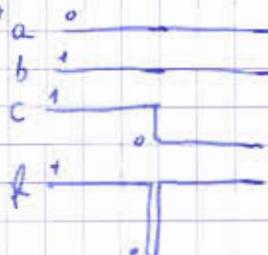
	00	01	11	10
0	0	1	1	1
1	0	0	0	1

$$f = \bar{a}c + b\bar{c}$$



Se C passa da 1 a 0, la f non dovrebbe cambiare, ma a causa del ritardo del NOT ha un leggero colpo a 0. Questo effetto si chiama

ALEA STATICIA



Le espressioni normali SP sono soggette ad alee statiche di "1".

Le espressioni normali PS sono soggette ad alee statiche di "0".

ALEA DINAMICA \rightarrow quando un segnale deve cambiare, invece di farlo così lo fa così .

Le espressioni normali SP/PS a 2 livelli sono prive di alee dinamiche.

Un'espressione SP è priva di alee statiche se non esistono due mintermi adiacenti non inclusi nel medesimo implicante.

00	11	10
01	10	-

c'è un'alea statica persistente \Rightarrow aggiungo gli implicanti che servono per rimuovere le alee statiche 0

Un'espressione PS è priva di alee statiche se le due maxterm adiacenti non inclusi nel medesimo implicante.

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

0	0	0
0	0	0

$$\bar{a}(b+d) \cdot (\bar{a}+b) \cdot (\bar{a}+b+\bar{c}) \text{ ha 2 alee statiche}$$

a	b	Cin	S	Cout
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



FULL ADDER

HALF ADDER

18/11/08

a	b	Cin	S	Cout
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



a	b	S	Cout
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



a _i	b _i	Cin	00	01	11	10
0	0	0	0	1	0	1
1	0	0	1	0	1	0

$$S = \overline{a_i} \overline{b_i} \cdot \text{Cin} + \overline{a_i} b_i \cdot \overline{\text{Cin}} + a_i \overline{b_i} \cdot \overline{\text{Cin}} + a_i b_i \cdot \text{Cin}$$

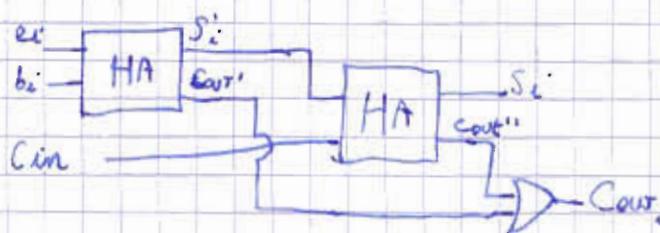
$$N_B = 5; N_m = 3 \cdot A + 4 = 16 \quad N_{LEI} = 12$$

a _i	b _i	Cin	00	01	11	10
0	0	0	0	0	1	0
1	0	0	1	0	1	0

$$\text{Cout} = b_i \cdot \text{Cin} + a_i \cdot \text{Cin} + a_i \cdot b_i$$

$$N_B = 4; N_m = 3 \cdot 2 + 3 = 9 \quad N_{LEI} = 6$$

FA da HA



$$N_B = 5; N_m = 40; \quad N_{LEI} = 7 \quad N_{IV} = 3!$$

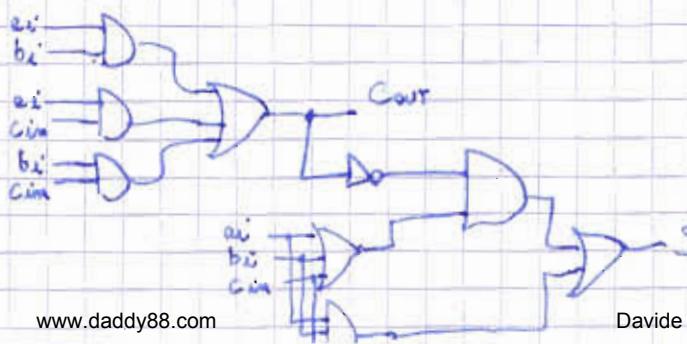
\uparrow
 $\ll 5+1$

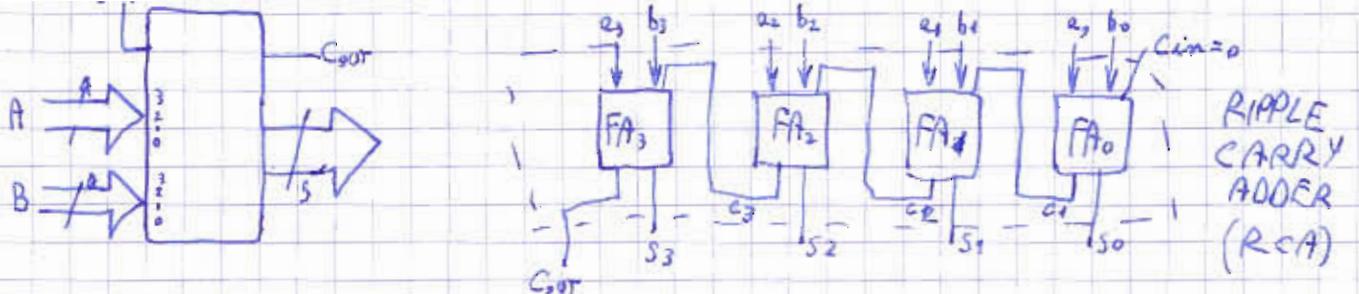
Un circuito ancora meno costoso usa la sintesi mediante mappe per Cout e usa un HA per la somma.

a _i	b _i	Cin	00	01	11	10
0	0	0	1	1	0	1
0	1	0	1	0	0	0

simile alla somma; per aggiungere un 1 a Cout aggiungo il minterm, mentre per aggiungere uno 0 aggiungo il maxterm!

$$S = (\overline{\text{Cout}} + a_i b_i \cdot \text{Cin}) \cdot (\overline{a_i} + \overline{b_i} + \text{Cin}) = \overline{\text{Cout}} \cdot (\overline{a_i} + \overline{b_i} + \text{Cin}) + a_i b_i \cdot \text{Cin}.$$

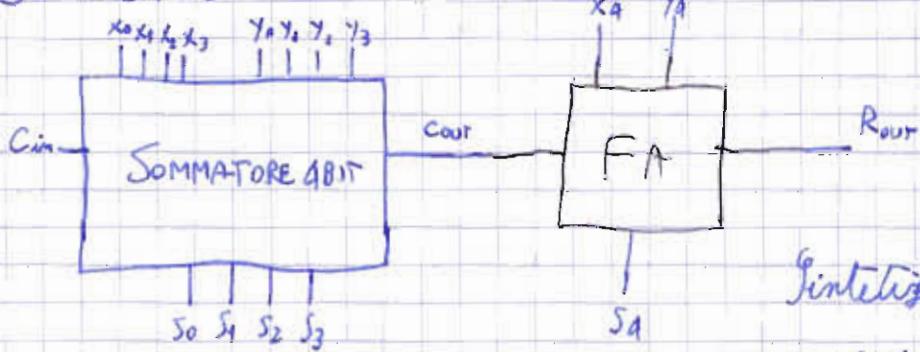




Sommatore più economico e più lento esistente. FA₀ posso sostituirlo con un HA se $C_{in} = 0$ sempre, ma non è in commercio.

Ritardo di $2N$ volte una porta logica elementare ($N = n^{\circ}$ bit).

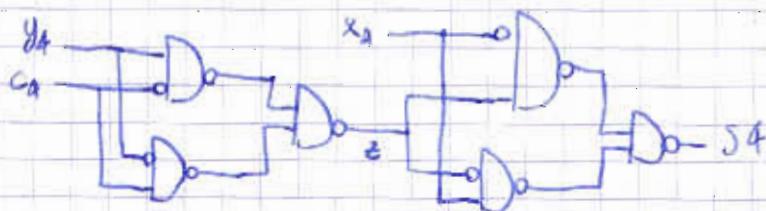
ES. ESAME



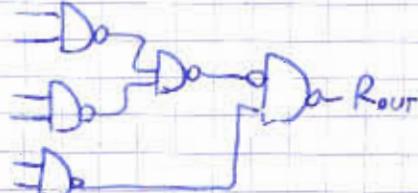
Provare circuito che somma 5 bit con solo NAND.

Sintetizzare quindi a solo NAND con FAN-IN 2

$$S_A = \bar{x}_A \bar{y}_A c_A + \bar{x}_A y_A \bar{c}_A + x_A \bar{y}_A \bar{c}_A + x_A y_A c_A = \bar{x}_A (\underbrace{\bar{y}_A c_A + y_A \bar{c}_A}_z) + x_A (\bar{y}_A \bar{c}_A + y_A c_A)$$



$$R_{out} = x_A y_A + x_A c_A + y_A c_A$$



ALEA

ab	00	01	11	10
00	0	1	0	1
01	0	1	0	1
11	1	1	0	1
10	0	0	1	1

$$f = \bar{c}\bar{d} + abc + ab\bar{c} + \bar{a}\bar{c}\bar{d}$$

L
P5 SP

$$2 \text{ alee } f' = f + b\bar{c}d + ab\bar{d}$$

Non mi devo preoccupare di alee se transitano da un implicant a un DC (-)

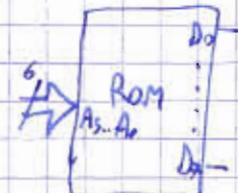
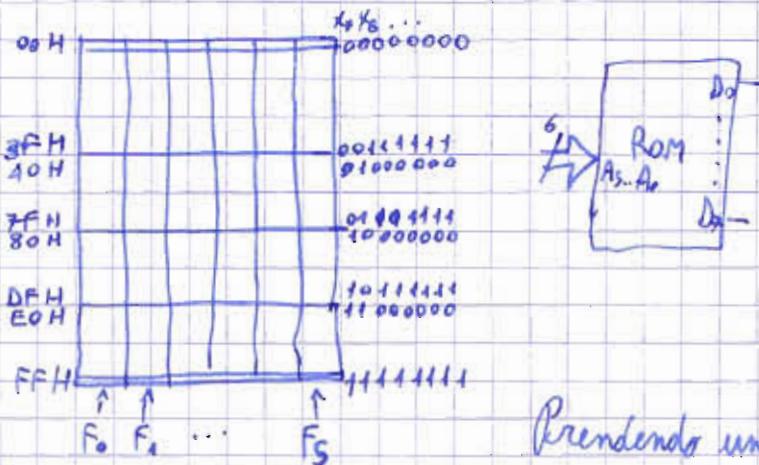
ROM

6 funzioni di 8 variabili indipendenti.

Disponibili ROM da 64 parole de 8 bit l'una dotate di CE (Chip Enable).
+ uscite ROM

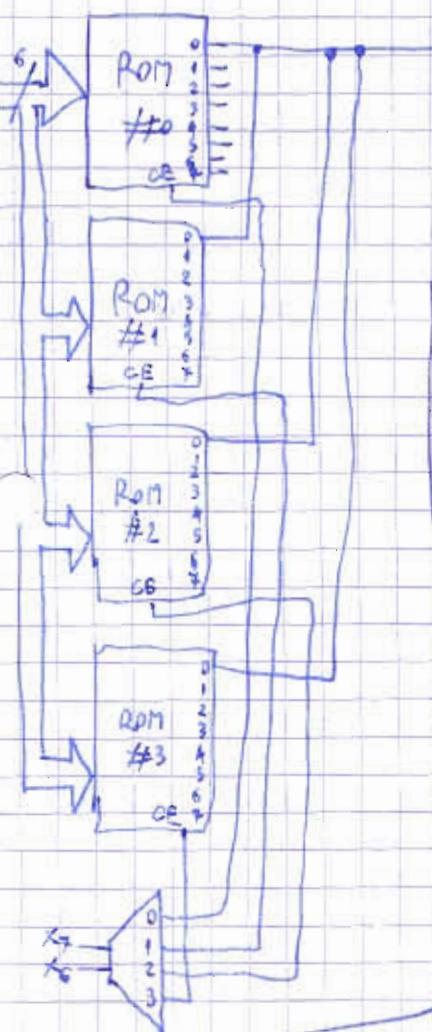
Le 8 var indip. possono assumere $2^8 = 256$ configurazioni.

Iscriviti un decoder con 6A uscite e quindi mi servono 6 ingressi di indirizzamento.



Prendendo una parola 01100001 , ogni ROM porta dir uscita il suo valore corrispondente, quindi mi interesseranno anche $00100001, 10100001$.

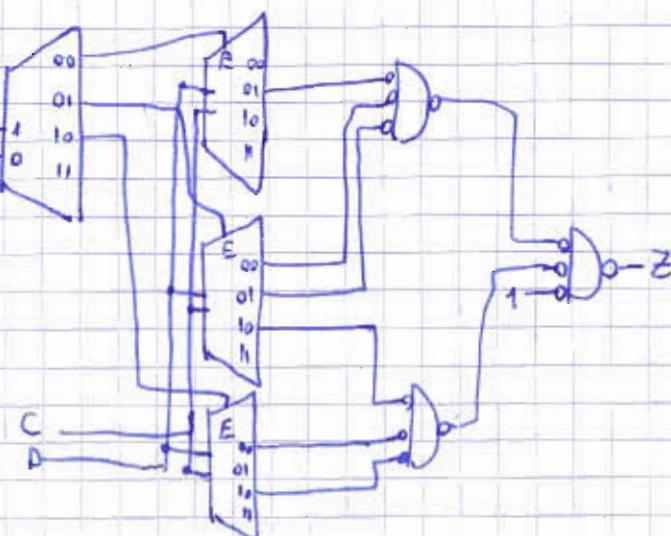
33



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

20/11/08

& tutto 0



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

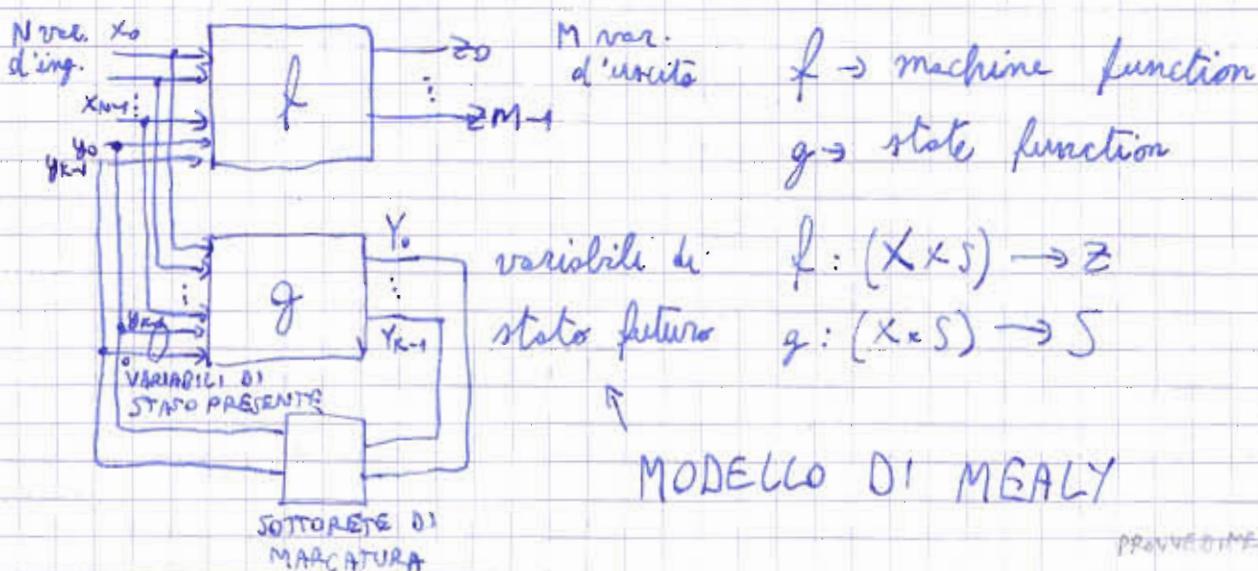
$$Z = (A+B) \cdot (S+B) \cdot (\bar{S}+\bar{A})$$

$$Z = (S+A) \cdot (\bar{A}+\bar{B}) \cdot (\bar{S}+B)$$

Prendendo C e : risolvo il problema delle aliie

RETI SEQUENZIALI

L'uscita dipende anche dalle uscite precedenti.

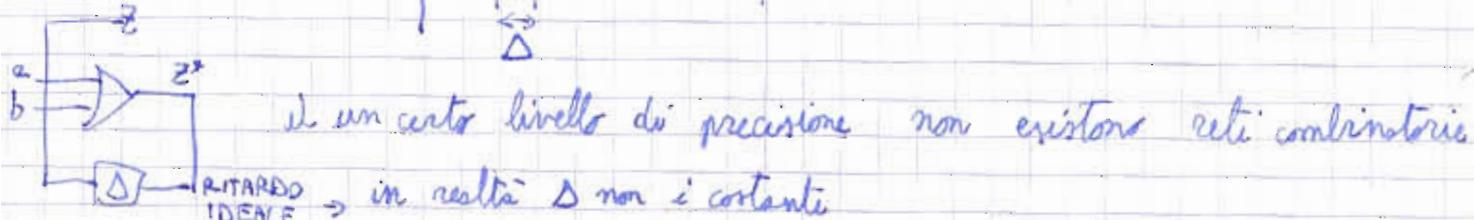
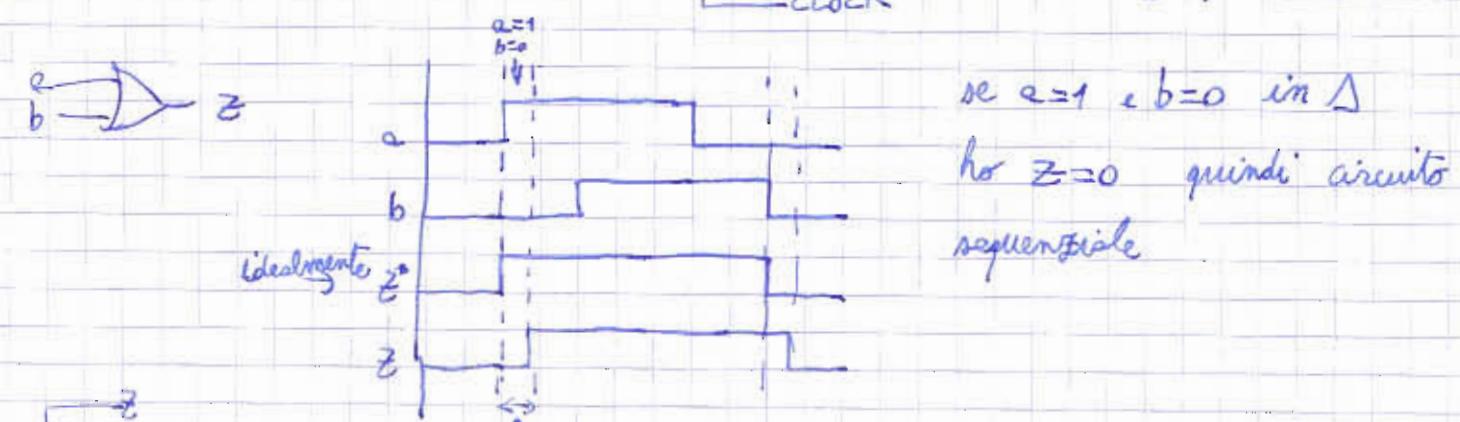
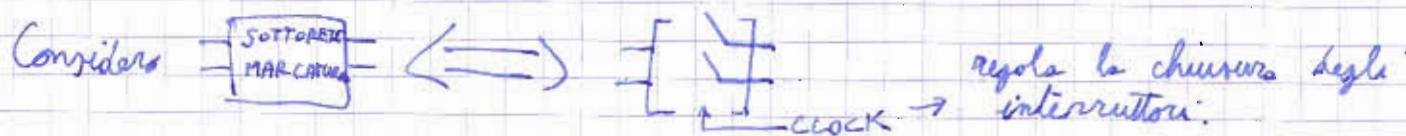


PROVVEDIMENTO FREQUENTATO

Se al posto delle sottosezioni di marcatura ho fili diretti \rightarrow MODELLO DI MEALY

Se y_{k+1}, \dots, y_0 non entrano in g ho un circuito combinatorio, così come se non entrano in f (circuito non ottimizzato).

Se X non entra in g , i due circuiti evolvono separatamente.



$y(t) = Y(t - \Delta)$ Tagliando i ritardi, avremo una rete combinatoria da studiare nel solito modo.

	00	01	11	10
0	0,0	1,0	1,0	1,0
1	0,1	1,1	1,1	1,1
(Y, Z)				

TABELLA DELLE TRANSIZIONI

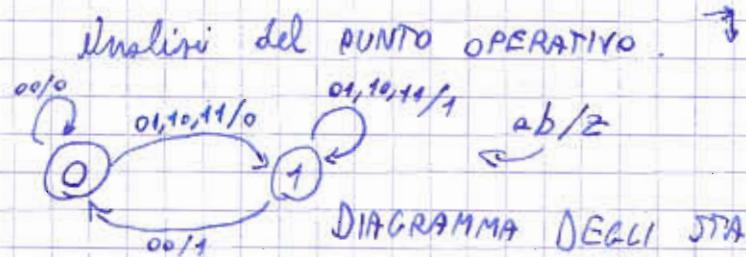
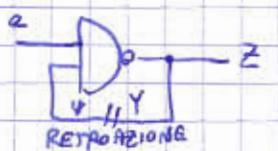


DIAGRAMMA DEGLI STATI

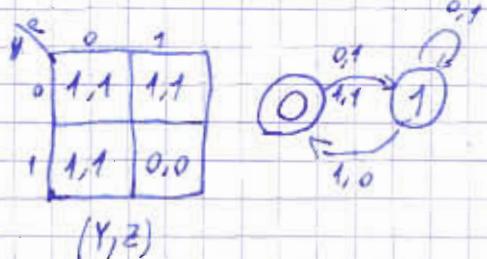


possiamo cogliere la retroazione

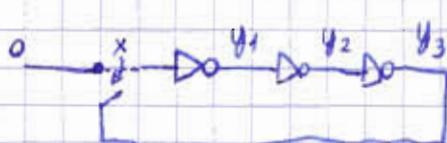
$$Z = a^T y = \bar{a} + \bar{y}$$

$$Y = \bar{a} + \bar{y}$$

$$y(t+\Delta) = Y(t)$$



I ritardi rendono i circuiti sequenziali.



τ è il ritardo Not

$$t=0^- \quad x \quad y_1 \quad y_2 \quad y_3$$

$$t=0^+ \quad 1 \quad \boxed{1} \quad 0 \quad 1$$

$$t=\tau \quad 1 \quad 0 \quad \boxed{0} \quad 1$$

$$t=2\tau \quad 1 \quad 0 \quad 1 \quad \boxed{1}$$

$$t=3\tau \quad 0 \quad \boxed{0} \quad 1 \quad 0$$

$$t=4\tau \quad 0 \quad 1 \quad \boxed{1} \quad 0$$

$$t=5\tau \quad 0 \quad 1 \quad 0 \quad \boxed{0}$$

$$t=6\tau \quad 1 \quad 1 \quad 0 \quad 1$$

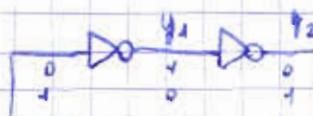
condizione instabile

variabili periodiche
con periodo 6τ ma
fase diversa.

OSCILLATORE AD

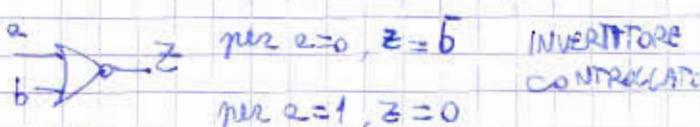
ANELLO

$$T = 2 \cdot n \cdot \tau$$

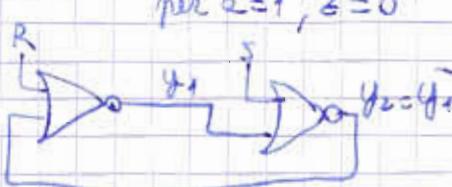


Tutte le configurazioni sono stabili: $y_2 = \bar{y}_1$

Memorizza il valore di un bit che però non sa come inserire.



INVERTITORE
CONTROLLATO

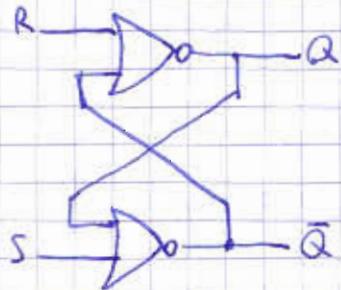


$RS = 00 \rightarrow y_1 \text{ memorizza}, y_2 = \bar{y}_1$

$R=1, S=0 \rightarrow y_1=0 \text{ e } y_2=1 \text{ commutazione o conferma}$

$R=0, S=1 \rightarrow y_1=0 \text{ e } y_2=1 \text{ commutazione o conferma}$

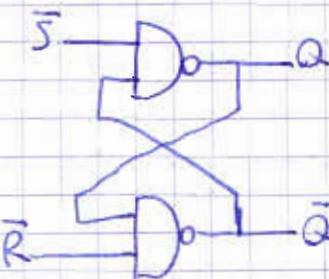
$R=1, S=1 \rightarrow VIETATA!$



LATCH SR

S
 R
 Q
 \bar{Q}

$SR=11$ vietata!



a	b	$a \uparrow b$
0	0	1
0	1	1
1	0	1
1	1	0

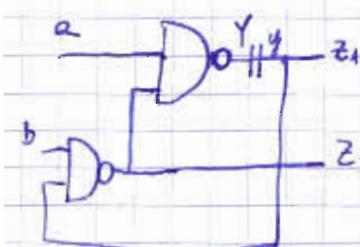
LATCH \rightarrow non dipende dell'istante di tempo.
FLIPFLOP \rightarrow dipende del tempo.

I latch SR vengono usati come sottoruti di marcatina.
Applicando 11 e poi 00 il latch inizia ad oscillare.

Q	Q^*	S	R	$Q^* \rightarrow$ uscita futura
0	0	0	-	
0	1	1	0	TABELLA DELLE ECCITAZIONI
1	0	0	1	$Q^* = S + \bar{R}Q$ con $SR \neq 11$
1	1	-	0	EQUAZIONE CARATTERISTICA

29/11/08

COMPITINO: 19/12 ore 15.30



$$Y = a \uparrow (b \uparrow y) = \bar{a} + by$$

$$z_1 = Y \quad z_2 = b \uparrow y = \bar{b} + \bar{y} \quad y(t+\Delta) = Y(t)$$

$a\bar{b}$	00	01	11	10
0	1, 01	1, 01	0, 01	0, 01
1	1, 11	0, 10	0, 10	0, 11

TABELLA
TRANSIZIONI

$Y = y$ stato futuro = stato presente

colonna iniziale
per cui non è vero

che $a = z_1$

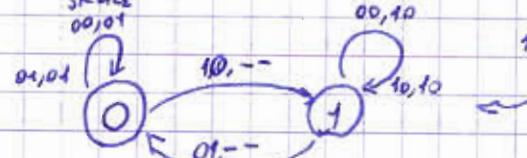
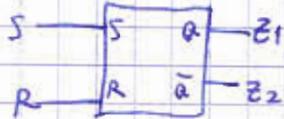
$\bar{a} = z_2$

introduzione di
"1" nella memoria
(SET)

introduzione di
"0" nella memoria (RESET)

"1" nella memoria
(SET)

SINTESI: 1) Dont Care



1) DIAGRAMMA DEGLI STATI

2) TABELLA TRANSIZIONI

y	00	01	11	10
0	0,00	0,01	--	1,--
1	1,10	0,--	--	1,10

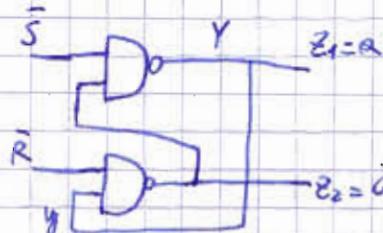
$Y_{SP} = Y + \bar{R}$

$$Y_{PS} = \bar{R}(S+Y)$$

equazione caratteristica del latch SR

$$Z_{1,SP} = Y \quad Z_{2,SP} = \bar{Y} \quad \text{oppure} \quad Z_1 = \bar{Y} + R \Rightarrow Z_1 = \bar{Y}S$$

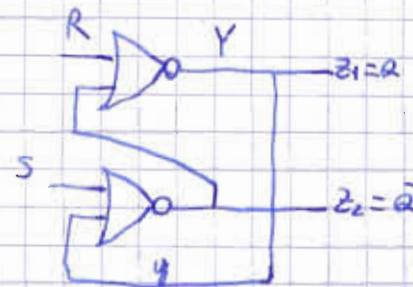
2 NAND: $Y = S + Y\bar{R} = \bar{S}\bar{A}(Y + \bar{R})$



$$Z_2 = \bar{Y} + R = Y \bar{R}$$

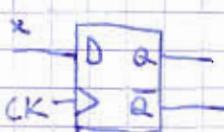
2 NOR:

$$Y_{PS} = \bar{R}(Y+S) = R\bar{Y}(Y+\bar{S})$$



$$Z_2 = \bar{Y}\bar{S} = Y\bar{S}$$

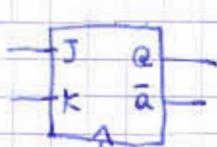
FLIP FLOP D



legge x sul fronte di solleto

$$Q^{n+1} = D^n \quad \text{Porta in uscita l'ingresso}$$

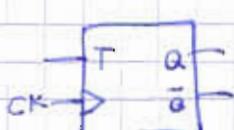
FLIP FLOP JK



$$Q^{n+1} = (J\bar{Q} + \bar{K}Q)^n$$

opera in $\rightarrow CK$
modo sincrono

FLIP FLOP T



$$Q^{n+1} = (T \oplus Q)^n = (\bar{T}Q + \bar{Q})^n$$

- Progettare un circuito sincrono con un'uscita Z e un ingresso E tale che
- se $E=1$ il circuito attiva l'uscita Z ogni 4 intervalli di clock
 - se $E=0$ il circuito si ferma con uscita $=0$.
1. DIAGRAMMA DEGLI STATI
-
2. TABELLA DI FLUSSO
- | E | 0 | 1 |
|-----|------|------|
| A | A, 0 | B, 0 |
| B | B, 0 | C, 0 |
| C | C, 0 | D, 0 |
| D | D, 0 | A, 1 |
3. CODIFICA STATI
- | y_2y_1 | A | B | C | D |
|----------|----|----|----|----|
| 00 | 00 | 01 | 10 | 11 |
| 01 | 01 | 10 | 11 | - |
| 10 | 11 | 00 | - | - |
| 11 | - | - | - | - |

4. TABEGLIA DELLE TRANSIZIONI

y_2y_1	0	1
A	00, 0	01, 0
B	01, 0	10, 0
C	11	11, 0
D	10	10, 0

$(y_2y_1)^{n+1}, Z^n$

$$Q^{n+1} = D^n$$

5. TECHNOLOGY MAPPING (es. FF-D, porte logiche)

$$\begin{aligned} D_2 &= \bar{E}y_2 + y_2\bar{y}_1 + E\bar{y}_2y_1 = \\ &= \bar{E}y_2 + E(y_1 \oplus y_2) \end{aligned}$$

$$D_1 = \bar{E}y_1 + E\bar{y}_1 = E \oplus y_1$$

6. MAPPA DELLE FUNZIONI DI ECCITAZIONE

y_2y_1	0	1
00	0	0
01	0	1
11	1	0
10	1	1

y_2y_1	0	1
00	0	1
01	1	0
11	1	0
10	0	1

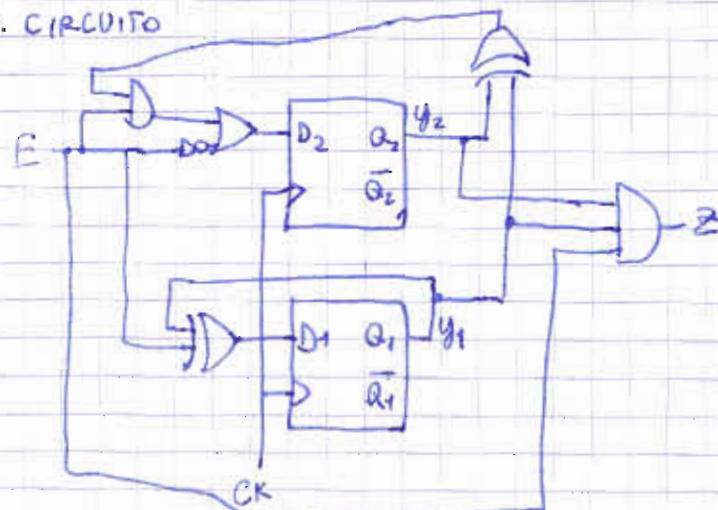
D_2
1^a colonna 2^a colonna

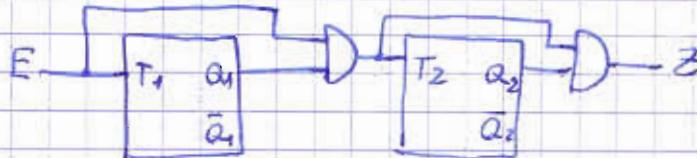
7. MAPPA USCITE

y_2y_1	0	1
00	0	0
01	0	0
11	0	1
10	0	0

$$Z = E y_2 y_1$$

8. CIRCUITO





$$T_1 = E$$

$$T_2 = Q_1 E$$

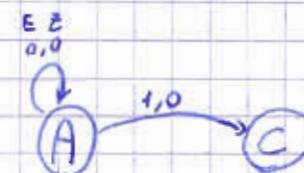
$$B = Q_2 T_2 = Q_1 Q_2 E \quad Q_1^{n+1} = (E \oplus Q_1)^n = E \bar{Q}_1 + \bar{E} Q_1$$

$$Q_2^{n+1} = (\bar{Q}_1 E \cdot Q_2 + Q_1 E \cdot \bar{Q}_2) = \bar{Q}_1 Q_2 + \bar{E} Q_2 + Q_1 \bar{Q}_2 E$$

TABELLA TRANSIZIONI

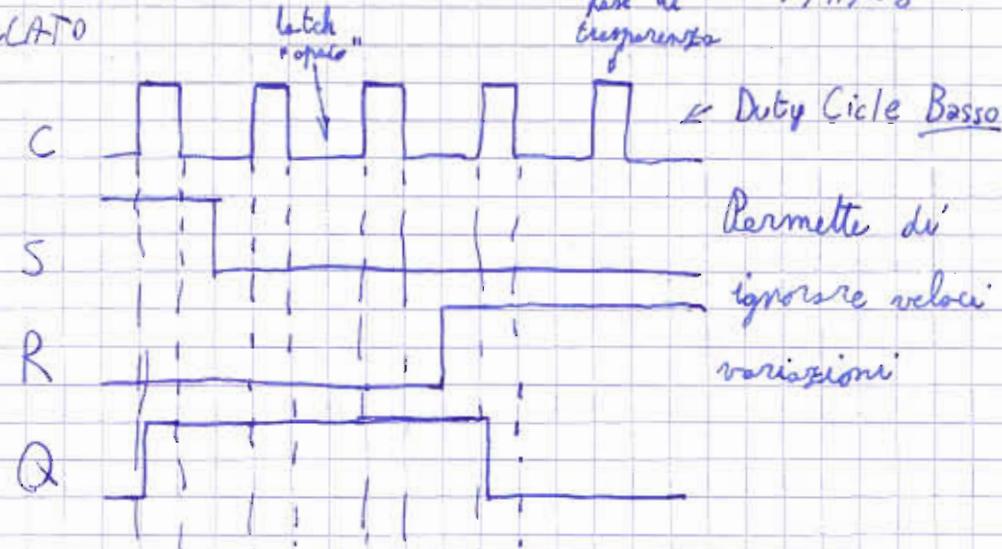
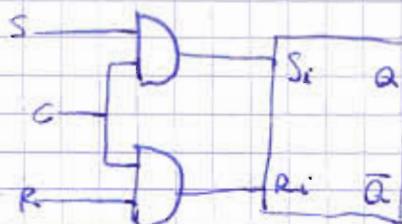
Q_1, Q_2	0	1
00	00, 0	10, 0
01	01, 0	11, 0
11	11, 0	00, 1
10	10, 0	01, 0

S	0	1
A	A, 0	C, 0
B	B, 0	D, 0
C	D, 0	A, 1
D	C, 0	B, 0

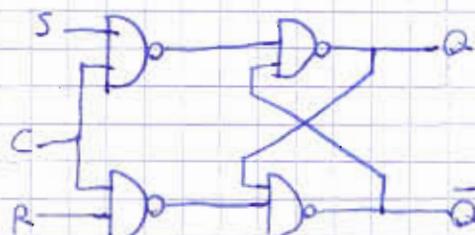


$$(Q_1 Q_2)^{n+1}, Z$$

LATCH SR CONTROLLATO



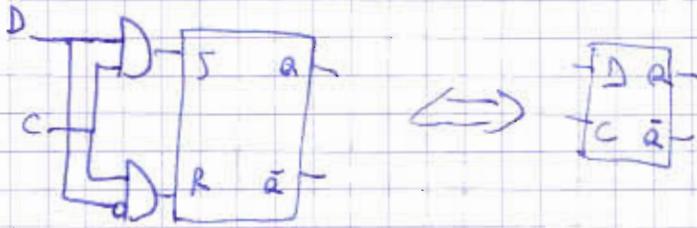
SOLI NAND



$$Q^{n+1} = (S + Q \cdot \bar{R})^n \text{ versione sincrona} \rightarrow \text{velo se } C \text{ è periodico e in modo che nella fase di trasparenza non ci siano veloci variazioni}$$

$$Q^{n+1} = \bar{C} Q^n + C(S + Q \bar{R})^n \text{ versione generale}$$

LATCH D CONTROLLATO



Nel FF MASTER SLAVE, quando $C=1$, il master elabora gli ingressi e le uscite si mantengono costanti; quando $C=0$, lo slave elabora i segnali di uscita precedenti del master e li elabora, creando l'uscita, mentre il master continua ad avere in uscita le uscite precedenti (dinattivato). Le uscite vengono elaborate sul fronte di discesa.

generazione uscite totali (slave)

generazione delle uscite del master

Il fronte di discesa campiona l'informazione (NEGATIVE EDGE-TRIGGERED). Se il DO lo metto nel master ottengo un POSITIVE EDGE-TRIGGERED.



ALEA ESSENZIALE \rightarrow dovuta al ritardo del NOT per cui può accadere che l'ingresso si muova in uscita perché entrambi i latch sono in fase trasparenza.

CATTURA DEGLI UNI (ONES CATCHING) \rightarrow se durante la fase di trasparenza ho veloci transizioni degli ingressi, il sistema errore (alea statica di "0" su "1")

PROCEDIMENTO DI SINTESI DI RSS

0. Interpretazione della specifica a parole. Si intarsi con forme d'onda Mealy? Moore?
1. Diagramma degli stati
2. Tabelle di flusso
3. Ottimizzazione numero degli stati
4. Codifica degli stati (possibile ottimizzazione).

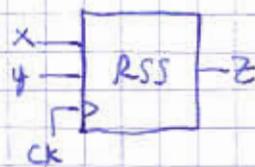
5. Tabella delle transizioni

6. Facoltà della tecnologia: tipo di FF, tecnologia combinatoria

7. Mappa funzioni di eccitazione e di uscita (JK?, D?, T?)

8. Schemi logici

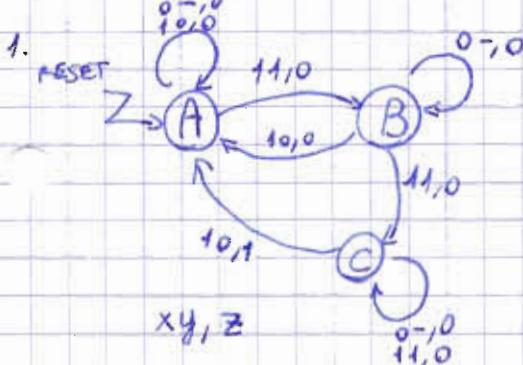
ES. 22/12/08 N° 2



- uscita inizialmente = 0

- $Z=1$ quando su y si verifica la sequenza 110

- x determina se y deve essere considerato 1 $\xrightarrow{x=0}$ ignorato $\xleftarrow{x=1}$ considerato



2.

$x\ y$	00	01	11	10
A	A,0	A,0	B,0	A,0
B	B,0	B,0	C,0	A,0
C	C,0	C,0	C,0	A,1

4. $A=00$ 5. $S_1 \ y$

$S_1 \ S_2$	00	01	11	10
A	00,0	00,0	01,0	00,0
B	01,0	01,0	10,0	00,0
C	—	—	—	—
$S_1 \ S_2$	10,0	10,0	10,0	00,1
$S_1^* \ S_2^*, Z$	—	—	—	—

6. Utilizzo FF-D e porte AND-OR-NOT

7. $S_1 \ y$

$S_1 \ S_2$	00	01	11	10
00	0	0	0	0
01	0	0	1	0
11	—	—	—	—
10	1	1	1	0

$$S_1^* = D_1 = \bar{x}S_1 + xyS_2 + xyS_1$$

$S_2 \ y$

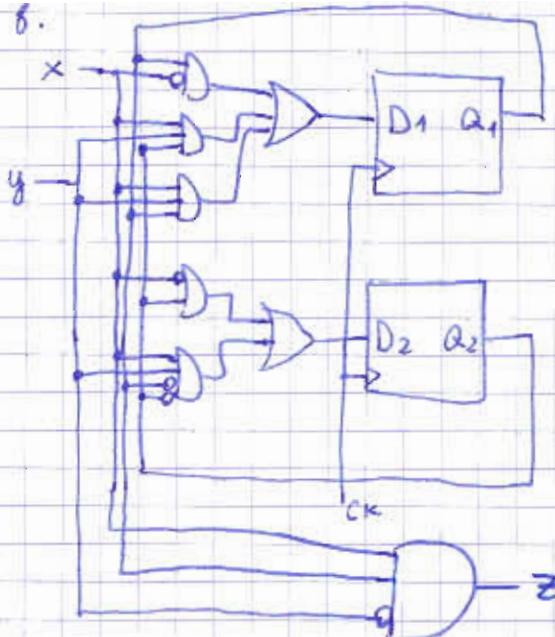
$S_1 \ S_2$	00	01	11	10
00	0	0	1	0
01	1	1	0	0
11	—	—	—	—
10	0	0	0	0

$S_2 \ y$

$S_1 \ S_2$	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	—	—	—	—
10	0	0	0	1

$$S_2^* = D_2 = \bar{x}S_2 + xy\bar{S}_1\bar{S}_2$$

$$Z = \bar{x}yS_1$$



S_1	S_2	00	01	11	10
00	00	00,0	00,0	01,0	00,0
01	01	01,0	01,0	10,0	00,0
11	—	—	—	—	—
10	10	10,0	10,0	10,0	00,1

numeri in
grassetto

Cambiamento

Con FF-JK

$Q_{JK} | Q^{n+1}$

0 0 0	0 ⁻ , $Q, J=0 \wedge K$	$0 \rightarrow 0$
0 0 1	0 ⁻	
0 1 0	1 ⁻ $\rightarrow Q=0, J=1 \wedge K$	$0 \rightarrow 1$
0 1 1	1 ⁻	
1 0 0	1 ⁻ $\rightarrow Q=1, K=0 \wedge J$	$1 \rightarrow 0$
1 0 1	0 ⁻ $\rightarrow Q=1, K=0 \wedge J$	
1 1 0	1 ⁻ $\rightarrow Q=1, K=1 \wedge J$	
1 1 1	0 ⁻ $\rightarrow Q=1, K=1 \wedge J$	$1 \rightarrow 0$

$Q^n \rightarrow Q^{n+1}$

$0 \rightarrow 0$	0	0 -
$0 \rightarrow 1$	1	1 -
$1 \rightarrow 0$	0	-1
$1 \rightarrow 1$	1	-0

ALGORITMO DI
ECCITAZIONI
DEL FF-JK

S_1	S_2	00	01	11	10
00	00	0	0	0	0
01	01	0	0	1 ⁻	0
11	—	—	1 ⁻	—	—
10	—	—	—	—	—

$$J_1 = xy S_2$$

S_1	S_2	00	01	11	10
00	—	—	—	1 ⁻	—
01	—	—	—	—	—
11	—	—	—	—	—
10	0	0	0	0	1

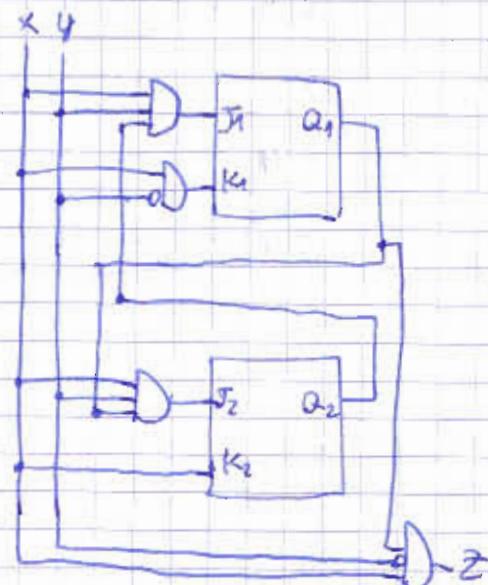
$$K_1 = x\bar{y}$$

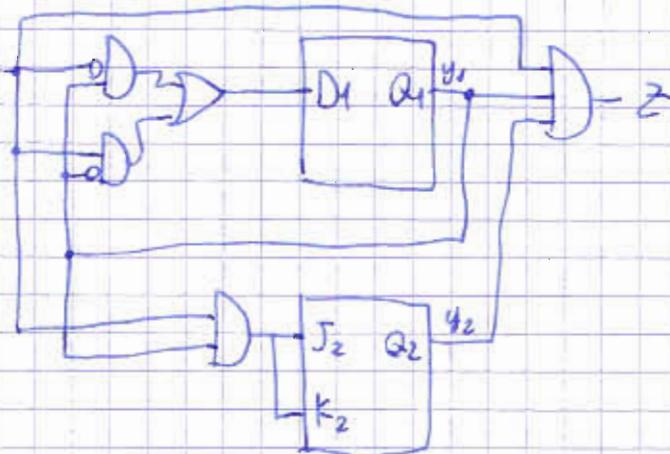
S_1	S_2	00	01	11	10
00	0	0	1 ⁻	1	0
01	—	—	1 ⁻	—	—
11	—	—	—	—	—
10	0	0	0	0	0

$$J_2 = x\bar{y} S_1$$

S_1	S_2	00	01	11	10
00	—	—	—	1 ⁻	—
01	0	0	1	1	1
11	—	—	1	—	—
10	—	—	—	—	—

$$K_2 = x$$

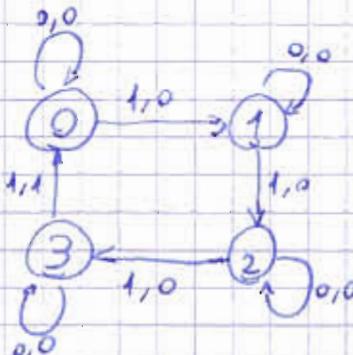




$y_2 y_1$	0	1
0 A	00, 001, 0	
1 B	01, 010, 0	
2 C	11, 000, 1	
3 D	10, 011, 0	

$y_2 y_1$	0	1
0 A	A, 0	B, 0
1 B	B, 0	C, 0
2 C	D, 0	A, 1
3 D	C, 0	D, 0

$$\begin{aligned}
 D_1 &= \bar{x}y_1 + x\bar{y}_1 & y_1^{n+1} &= D_1 = \bar{x}y_1 + x\bar{y}_1 \\
 J_2 &= x y_1 & y_2^{n+1} &= (J_2 \bar{y}_2 + \bar{K}_2 y_2)^n = \\
 K_2 &= x y_1 & & \text{as mentioned} \\
 Z &= x y_1 y_2 & Z &= x y_1 \bar{y}_2 + (\bar{x} + \bar{y}_1) \cdot y_2 = \\
 & & & = x y_1 \bar{y}_2 + \bar{x} y_2 + \bar{y}_1 y_2 =
 \end{aligned}$$

 $y_2 y_1, Z$

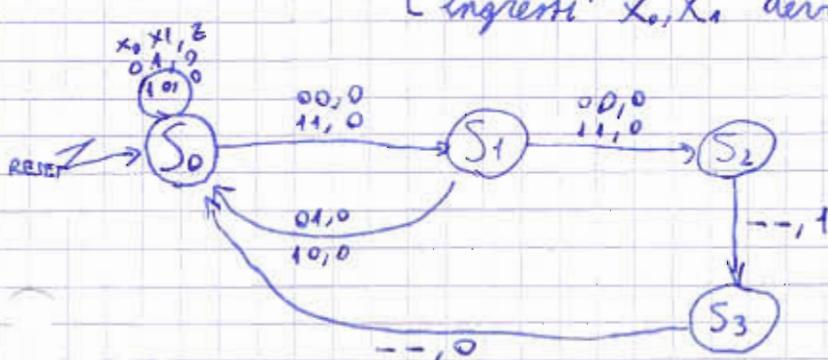
2/12/08

I flip flops hanno alcuni difetti: il Not deve avere un ritardo $\rightarrow 0$, e' soggetto a problemi di ales... Per questo quindi i FF-EDGE TRIGGERED. Gli ingressi PRESET e CLEAR permettono di gestire il FF in modo estremamente indipendente dal clock, per mantenere il FF in uno stato particolare indipendentemente dagli ingressi. Entrano in forma negata.

ES. ESAME



$Z(n) = 1 \text{ se } x_0(n-1) = x_1(n-1) \text{ e } x_0(n-2) = x_1(n-2)$
 nell'intervalli in cui $z=1$ e in quello successivo gli ingressi x_0, x_1 devono essere ignorati.



	x_1	00	01	11	10
S_0	$S_{1,0}$	$S_{0,0}$	$S_{1,0}$	$S_{0,0}$	
S_1	$S_{2,0}$	$S_{0,0}$	$S_{2,0}$	$S_{0,0}$	
S_2	$S_{3,1}$	$S_{3,1}$	$S_{3,1}$	$S_{3,1}$	
S_3	$S_{0,0}$	$S_{0,0}$	$S_{0,0}$	$S_{0,0}$	

CODIFICA STATI

$$S_0 = 00$$

$$S_1 = 01$$

$$S_2 = 10$$

$$S_3 = 11$$

	y_1	00	01	11	10
y_0	00	01,0	00,0	01,0	00,0
	01	10,0	00,0	10,0	00,0
	11	00,0	00,0	00,0	00,0
	10	11,1	11,1	11,1	11,1

$$y_1^*, y_0^*, z$$

$$z^n \equiv y_1 \bar{y}_0 \quad \text{Utilizzo prima FF-D e poi FF-JK}$$

CASO FF-D

	x_1	00	01	11	10
y_1, y_0	00	0	0	0	0
	01	1	0	1	0
	11	0	0	0	0
	10	1	1	1	1

	x_1	00	01	11	10
y_1, y_0	00	1	0	11	0
	01	0	0	0	0
	11	0	0	0	0
	10	1	1	11	1

$$D_1 = y_1 \bar{y}_0 + \bar{y}_1 y_0 \bar{x}_0 \bar{x}_1 + \bar{y}_1 y_0 x_0 x_1$$

$$D_0 = y_1 \bar{y}_0 + \bar{y}_0 \bar{x}_0 \bar{x}_1 + \bar{y}_0 x_0 x_1$$

CASO FF-JK

	x_1	J	K	
$0 \rightarrow 0$	0	0	-	
$0 \rightarrow 1$	1	1	-	
$1 \rightarrow 0$	0	-	1	
$1 \rightarrow 1$	1	-	0	

	x_1	00	01	11	10
y_1, y_0	00	01,0	00,0	01,0	00,0
	01	10,0	00,0	10,0	00,0
	11	00,0	00,0	00,0	00,0
	10	11,1	11,1	11,1	11,1

	x_1	00	01	11	10
y_1, y_0	00	0	0	0	0
	01	1	1	0	11
	11	1	1	1	1
	10	-	-	-	-

-	-	-	-
-	-	-	-
1	1	1	1
1	1	1	1
0	0	0	0

$$K_1 = y_0$$

1	0	1	0
-	-	-	-
1	1	1	1
1	1	1	1

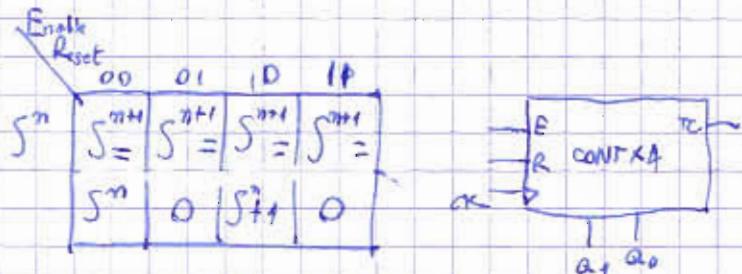
$$J_0 = y_1 \bar{x}_0 \bar{x}_1 + x_0 x_1$$

-	-	-	-
1	1	1	1
1	1	+	1
-	-	-	-

$$K_0 = 1$$

Stesso esercizio utilizzando un contatore binario di modulo 4 dotato di ENABLE e RESET.

Il reset prende sempre.



In $S_1 (y_1, y_0 = 01)$, se $x_0 \neq x_1$ deve essere attivato il reset per tornare a S_0 .

In $S_0 (y_1, y_0 = 00)$, se $x_0 = x_1$ deve valere $E = 1$. Se $x_0 \neq x_1$,

In $S_1 (y_1, y_0 = 01)$, se $x_0 = x_1$ deve essere $E = 1$ e $R = 0$

In $S_2 (y_1, y_0 = 10)$, in tutti i casi $E = 1$, $R = 0$

In $S_3 (y_1, y_0 = 11)$, in tutti i casi

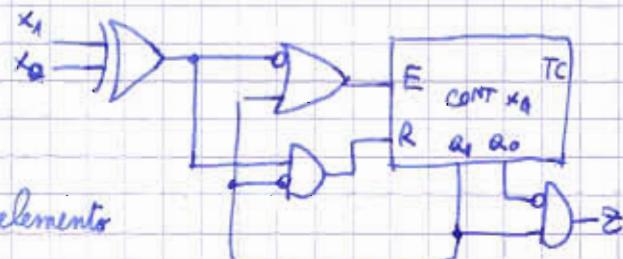
- $E = 1, R = 0$
- $E = 0, R = 1$
- $E = 1, R = 1$

y_1, y_0	00	01	11	10
00	0	1	0	1
01	0	1	0	1
11	-	-	-	-
10	0	0	0	0

$$R = \bar{y}_1 \bar{x}_0 x_1 + \bar{y}_1 x_0 \bar{x}_1 = \bar{y}_1 (x_0 \oplus x_1)$$

y_1, y_0	00	01	11	10
00	1	0	1	0
01	1	-	1	-
11	1	1	1	1
10	1	1	1	1

$$E = y_1 + \bar{x}_0 x_1 + x_0 \bar{x}_1$$



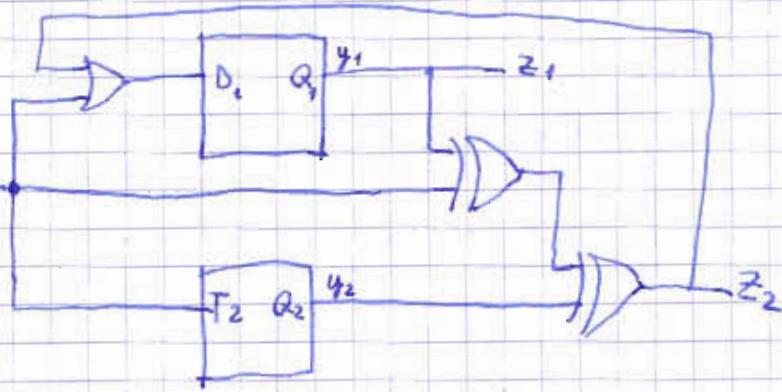
Se avessi usato il RESET come elemento determinante, potessi prendere

$$E = 1 \quad e \quad R = \bar{y}_1 (x_0 \oplus x_1)$$

Fare es. 22/12/04 su contatore modulo 4 con enable e reset.

CODIFICA ONE-HOT \rightarrow solo un FF è attivo $A = 000$, $B = 010$, $C = 001$. Per N stati. N bit

DISTANZA MINIMA \rightarrow ad esempio $S_0 = 00$, $S_1 = 01$, $S_2 = 11$, $S_3 = 10$ (bit che cambiano = 1).



Il clock è not intero.
EX-OR = disparità di 1

$$T_2 = X$$

$$Z_1 = Y_1$$

$$Z_2 = X \oplus Y_1 \oplus Y_2 = X\bar{Y}_1\bar{Y}_2 + X\bar{Y}_1Y_2 + \bar{X}\bar{Y}_1Y_2 + \bar{X}Y_1\bar{Y}_2$$

$$D_1 = X + Z_2 = X + \cancel{X\bar{Y}_1\bar{Y}_2} + \cancel{X\bar{Y}_1Y_2} + \cancel{\bar{X}\bar{Y}_1Y_2} + \cancel{\bar{X}Y_1\bar{Y}_2} = X + \bar{Y}_1Y_2 + Y_1\bar{Y}_2 = Y_1^{n+1}$$

integrale in X opposto a X quindi
inutile esplicitare,
altrimenti le avrei
svolte

$$Y_1^{n+1} = D_1^n$$

$$Y_2^{n+1} = (T_2 \oplus Y_2)^n = X\bar{Y}_2 + \bar{X}Y_2$$

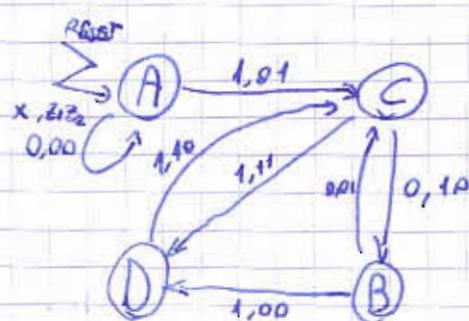
Chiamo $A=00$, $B=01$, $C=01$, $D=10$

$\bar{Y}_1 \bar{Y}_2$	0	1
00	00, 00	11, 01
01	11, 01	10, 00
11	01, 10	10, 11
10	10, 11	11, 10

$$(Y_1 Y_2)^{n+1}, (Z_1 Z_2)^n$$

X	0	1
A	A, 00	C, 01
B	C, 01	B, 00
C	B, 10	D, 11
D	D, 11	C, 10

Disegno il diagramma degli stati



4/12/08

Per le funzioni (macchine) completamente specificate, l'automa minimo equivalente è unico.

IDENTIFICAZIONE DEGLI STATI

, equivalenti per macchine completamente specificate
'compatibili' per macchine non .. "

STATO NON RAGGIUNGIBILE \rightarrow stato in cui non esiste alcuna sequenza di transizioni di stato che porta dallo stato iniziale in tale stato.

STATI INDISTINGUIBILI \rightarrow se sottoposti agli stessi ingressi producono le stesse uscite, per ogni sequenza d'ingressi.

REGOLA DI PAULL - UNGER \rightarrow due stati $s_i, s_j \in S$ sono indistinguibili se e solo se \forall sequenza d'ingressi i_a :

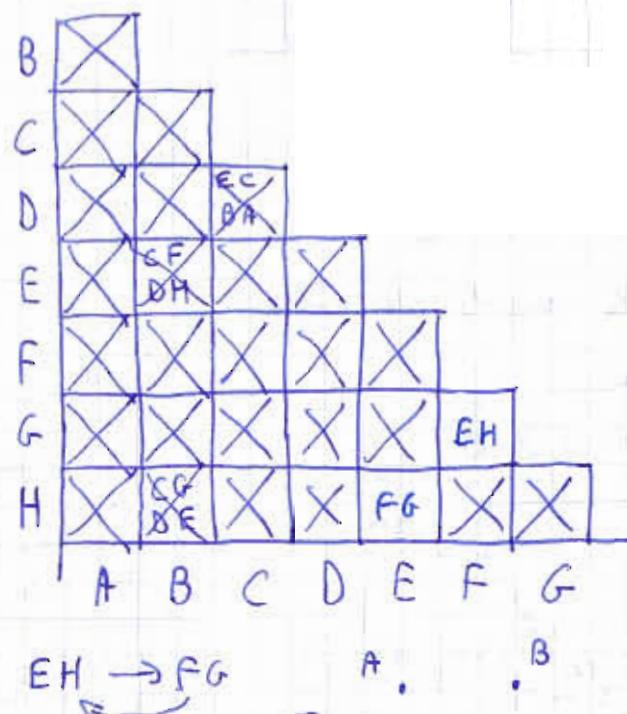
- $U(s_i, i_a) = U(s_j, i_a)$ le uscite sono uguali.
- $S^*(s_i, i_a) \sim S^*(s_j, i_a)$ gli stati futuri sono indistinguibili

L'equivalenza di s_0 e s_3 dipende da quella di s_1, s_2 .

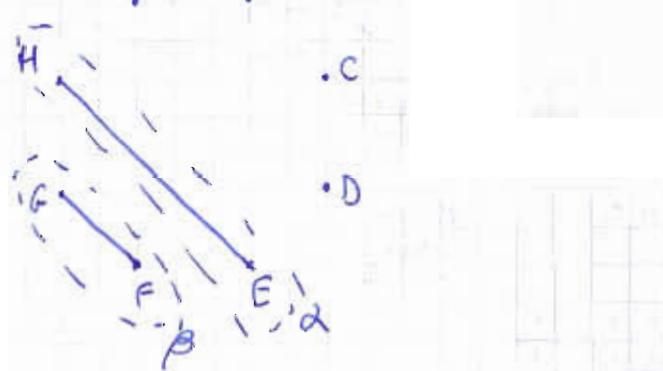
Cassella vuota \rightarrow equivalenti Cassella \times \rightarrow non equivalenti

ES. D1 DEL 22/12/04

	0	1
A	B, 0	D, 0
B	C, 0	D, 1
C	E, 1	B, 1
D	C, 1	A, 1
E	F, 0	H, 1
F	E, 1	G, 0
G	H, 1	F, 0
H	G, 0	E, 1

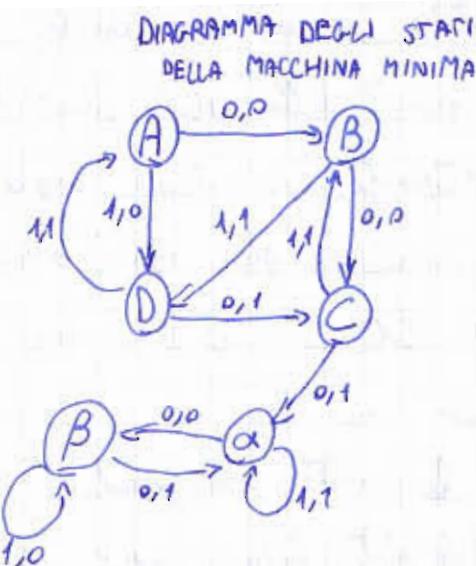


Classi massime: A, B, C, D, $\frac{EH}{\alpha}$, $\frac{FG}{\beta}$

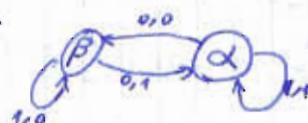


	0	1
A	B, 0	D, 0
B	C, 0	D, 1
C	A, 1	B, 1
D	C, 1	A, 1
α	B, 0	$\alpha, 1$
β	$\beta, 1$	B, 0

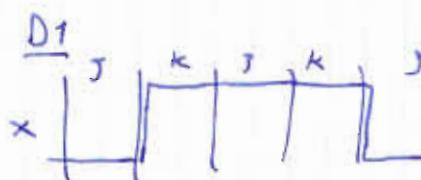
TABELLA
DI
FLUSSO
MINIMA



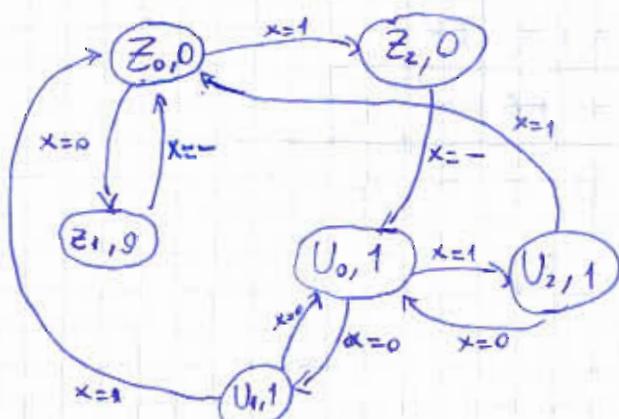
Assumo come stato iniziale $\alpha \cdot \beta$ in modo da eliminare la parte ABCD e il diagramma diventa:



ES. ESAME 02/04



- x funge da J e K in 2 intervalli consecutivi
- Z cambia dopo due intervalli
- usare Moore (usato associato allo stato)
- inizio con z_0 in cui $z=0$

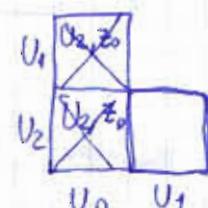
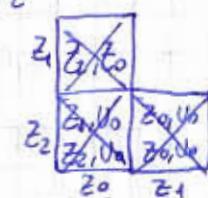


x	0	1
Z_0	$Z_1, 0$	$Z_2, 0$
Z_1	$Z_0, 0$	Z_0
Z_2	$U_0, 1$	U_0
U_0	$U_1, 1$	U_2
U_1	$U_0, 1$	Z_0
U_2	$U_0, 1$	Z_0
K	$Z_0, 0$	1

AUTOMA
MINIMO

x	0	1
Z_0	Z_1	Z_2
Z_1	Z_0	Z_0
Z_2	U_0	U_0
U_0	U_1	U_2
U_1	U_0	Z_0
U_2	U_0	Z_0

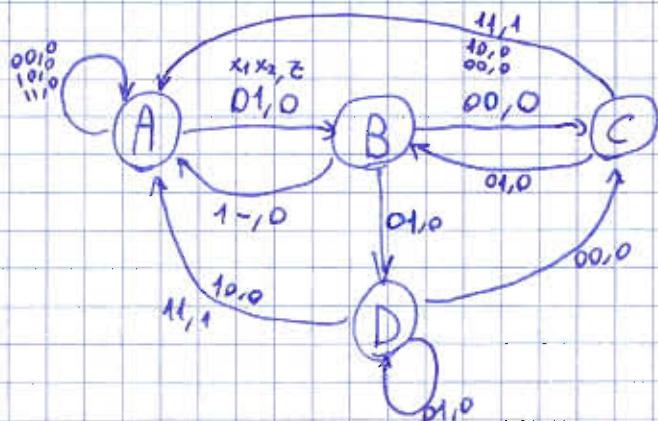
usato z
0
0
0
1
1



$$|U_1 \cup U_2| = K$$



	x_1	x_2
$t-2$	0	1
$t-1$	0	-
t	1	1

fare diagramma stato ($t \leq 4$)

	x_1x_2	00	01	11	10
A	$A,0$	$B,0$	$A,0$	$A,0$	
B	$C,0$	$D,0$	$A,0$	$A,0$	
C	$A,0$	$B,0$	$A,1$	$A,0$	
D	$C,0$	$D,0$	$A,1$	$A,0$	

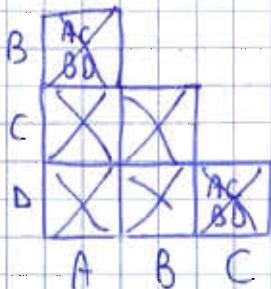


TABELLA DI FLUSSO MINIMA

$y_{1,0}$	00	01	11	10
00	0	0	0	0
01	1	1	0	0
11	0	0	0	0
10	1	1	0	0

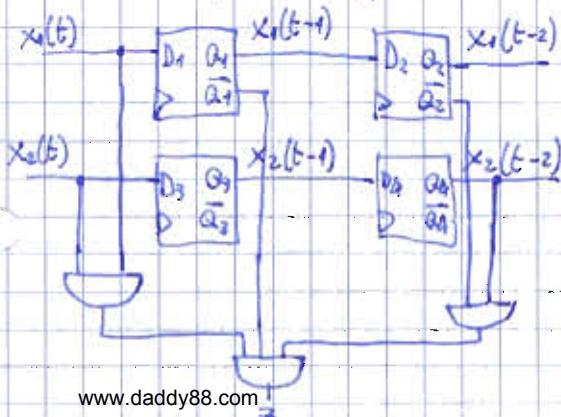
$y_{1,0}$	00	01	11	10
00	0	1	0	0
01	1	0	0	0
11	0	1	0	0
10	1	0	0	0

$D_1 = \bar{x}_1 \bar{y}_0 y_0 + \bar{x}_1 y_0 y_0$

$D_0 = \bar{x}_1 x_2 \bar{y}_1 y_0 + \bar{x}_1 \bar{x}_2 \bar{y}_1 y_0 + \bar{x}_1 x_2 y_1 y_0 + \bar{x}_1 \bar{x}_2 y_1 \bar{y}_0 = (y_1 \oplus y_0 \oplus x_1)$

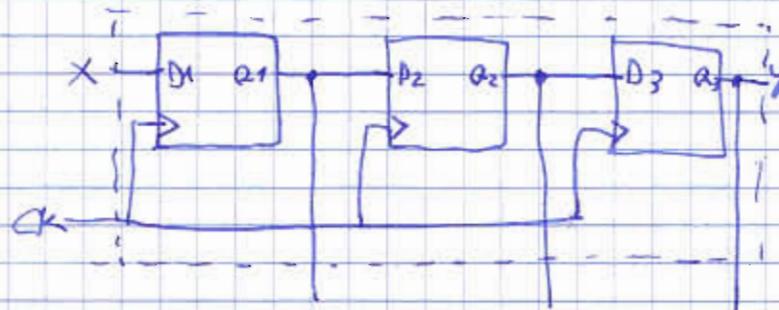
$Z = y_1 x_1 x_2$

Un modo per rappresentare il circuito è

SISTEMA A i valori vecchi degli ingressi
MEMORIA → non mi interessano più:
FINITA

realizzazione diretta: più FF, meno logica comb.

REGISTRO A SCORRIMENTO - SHIFT REGISTER



$$Q_1^{n+1} = D_1^n = X^n$$

$$Q_2^{n+1} = D_2^n = Q_1^{n-1} = D_1^{n-1} = X^{n-1}$$

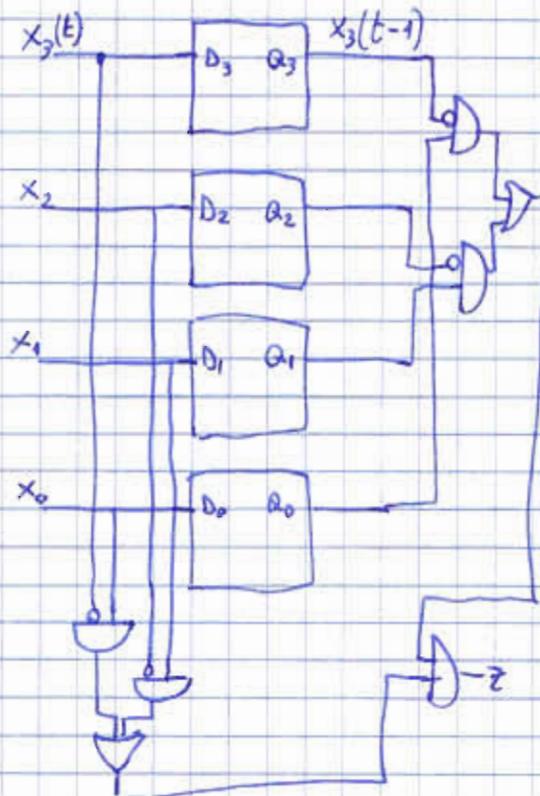
$$Q_3^{n+1} = D_3^n = Q_2^{n-2} = X^{n-2}$$



$\exists = 1$ se gli ultimi due numeri Nego scritti sono primi (1, 2, 3, 5, 7)

0001, 0010, 0011, 0101, 0111

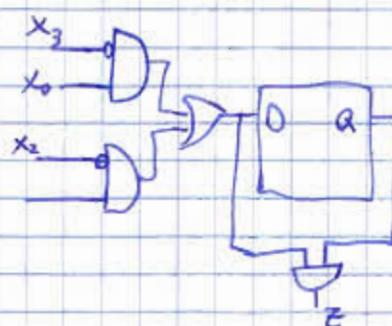
$BCD \rightarrow 1^{\circ}$ bit : signo



$x_1 x_2$	00	01	11	10
00	0	1	1	1
01	0	1	1	0
11	—	—	—	—
10	0	0	—	—

$$P = \overline{x_3}x_0 + \overline{x_2}x_1$$

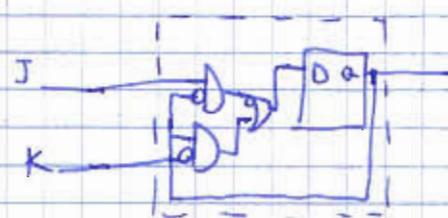
Spring



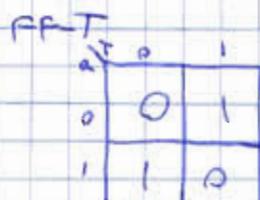
SOLUZIONE ESTENDIBILE

Con un FF-D sintetizzare un FF-JK

J_k	00	01	11	10
0	0	0	(1)	(1)
1	(1)	0	0	1



$$Q^{n+1} = D^n = J\bar{Q} + \bar{k}Q$$



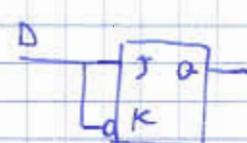
$$Q^{n+1} = \bar{D}^n = \bar{Q}\bar{T} + \bar{T}Q > Q\oplus T$$

CON J-K : FF-D

a^T	0	0	1
a^D	0	0	1
Q^{n+1}	0	1	1

a^T	0	0	1
a^D	0	0	1
$J=D$	-	-	-

a^T	0	0	1
a^D	0	0	1
$K=\bar{D}$	1	0	0



FF-T

a^T	0	0	1
a^D	0	0	1
T	1	1	0

a^T	0	0	1
a^D	0	0	1
T	-	-	-

a^T	0	0	1
a^D	0	0	1
$K=T$	1	0	1



$J=T$ $K=T$

FF-SR

a^S	00	01	11	10
a^R	0	0	-	1
R	1	1	0	-1

a^S	00	01	11	10
a^R	0	0	-	1
S	-	-	-	-

$J=S$



a^S	00	01	11	10
a^R	0	0	-	-
R	1	0	-	0

$K=R$

9/12/08

16/12 NO LEZIONE

METODI EURISTICI X LA CODIFICA DEGLI STATI

- codifica consecutivi gli stati che hanno gli stessi stati futuri
- " " " " " che sono stati futuri dello stesso stato

x	0	1
S_1	$S_{3,0}$	$S_{2,0}$
S_2	$S_{1,0}$	$S_{4,0}$
S_3	$S_{4,0}$	$S_{3,0}$
S_4	$S_{1,0}$	$S_{9,1}$
S_5	$S_{1,0}$	$S_{8,0}$

i) Stati con lo stesso stato futuro

$S_2 - S_3$ con peso 2 (colonne =)

$S_2 - S_4$ con peso 1

$S_3 - S_4$ con peso 1

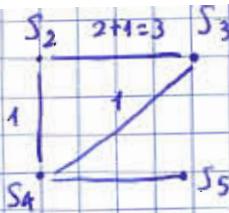
ii) Stati futuri (non colonne adiacenti) di uno stesso stato

$S_2 - S_3$ peso 1

$S_4 - S_5$ peso 1

iii) stesso uscita in una o più colonne

poco rilevante \rightarrow trascurare.



S_1	S_2	S_3
S_5	S_4	

valore = 5 (tangente S_2-S_4)

MACCHINE NON COMPLETAMENTE SPECIFICATE

$S_i \sim S_j$ stati equivalenti

$S_i \approx S_j$ stati compatibili \rightarrow se si ha la stessa uscita (coppia) per ogni ingresso in cui si altra uscita specificata.

CLASSE MASSIMA \rightarrow tutti i membri sono compatibili tra loro.

TEOREMI

L'insieme delle classi massime definisce un automa equivalente a quello di partenza non necessariamente minimo (a volte è più grande)

La più grande classe di incompatibilità definisce un LOWER-BOUND (limite inferiore, eventualmente non raggiungibile per la dimensione dell'automa minimo equivalente).

ES. ESAME

1) RAGGIUNGIBILITÀ: $A \xrightarrow{\cdot} D \xrightarrow{\cdot} F \xrightarrow{\cdot} C$

\circ	1	
A	D, 0	-,-
B	-,-	B, 1
C	F, -	-,-
D	F, 1	A, 1
E	D, -	C, 0
F	D, 1	C, -

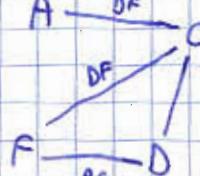
B, E NON RAGGIUNGIBILI

C	DF	
D	X	\approx
F	X	DF AC

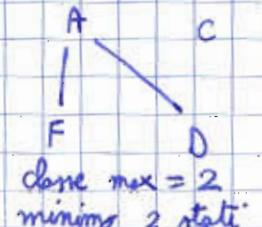
2) Coppie di stati compatibili

AC, DF, CF, CD

GRAFO DI COMPATIBILITÀ



GRAFO INCOMPATIBILITÀ

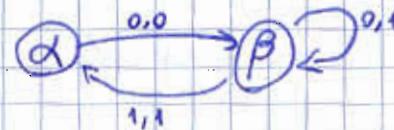


3) classi max compatibilità

$\{A, C\}$, $\{C, D, F\}$
 α β

α	0	1
β	0, 0	-,-
β	B, 1	$\alpha, 1$

\Rightarrow macchina minima!



altro automa possibile: $\alpha = \{A, C\}$, $\gamma = \{D, F\}$

α	$\gamma, 0$	$-,-$
γ	$\alpha, 1$	$\alpha, 1$

stesso /
automa .

ES. ESAME

$x_1 x_2$	00	01	11	10
S_1	$S_2, 0$	$-,-$	$S_3, -$	$S_2, 0$
S_2	$S_3, 0$	$S_3, 1$	$S_2, 0$	$-,-$
S_3	$S_3, 0$	$S_4, 1$	$-,-$	$S_3, 0$
S_4	$-,-$	$S_4, 1$	$S_2, -$	$-,-$
S_5	$-,-$	$-,-$	$S_4, 1$	$-,-$

$S_2 S_5$ incompatibili

$S_3 S_5$ incompatibili
dal 2° giro

$S_4 S_5$

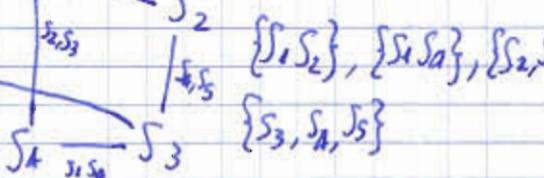
$S_5 S_3$ incompatibili
dal 1° giro

COPPIE DI STATI COMPATIBILI

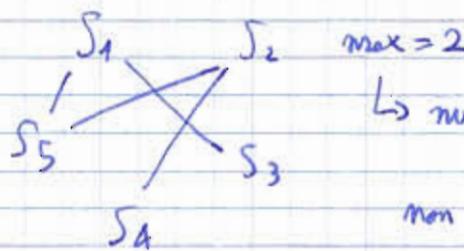
$S_1 S_2, S_4 S_4, S_2 S_3, S_3 S_4, S_3 S_5,$

$S_4 S_5$

Classi max:



Grapho incompatibilità:



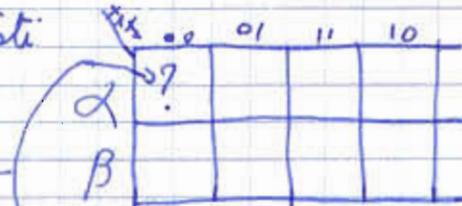
max = 2

↳ minimo 2 stati

non possibile

Volutiamo il seguente insieme di classi

selezionate: $\alpha = \{S_1, S_2\}$ $\beta = \{S_3, S_4, S_5\}$



nessuna classe contiene $S_2 S_3 \rightarrow$ non chiuso

Dovrò prendere $\{\overset{\alpha}{S_1}, \overset{\alpha}{S_2}\}$ che implica di prendere $\{\overset{\beta}{S_2}, \overset{\beta}{S_3}\}$, che implica $\{\overset{\gamma}{S_4}, \overset{\gamma}{S_5}\}$.

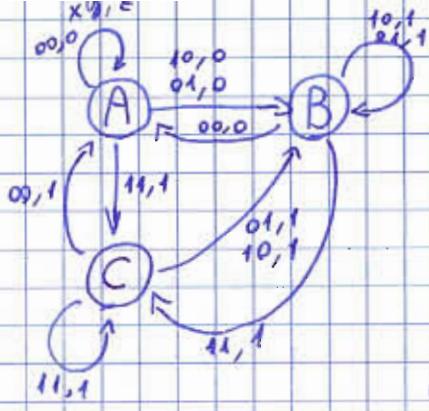
In questo modo copro tutte le classi \rightarrow macchina minima equivalente anche se non fatto da classi max.

$x_1 x_2$	00	01	11	10
α	$\beta, 0$	$\gamma, 1$	$\beta, 0$	$\alpha/\beta, 0$
β	$\beta, 0$	$\gamma, 1$	$\alpha/\beta, 0$	$\gamma, 0$
γ	$-,-$	$\alpha, 1$	$\alpha, 1$	$-,-$

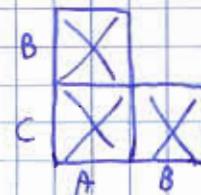
ES. SINTESI



$z=1$ quando tra gli ultimi 4 bit ricevuti (2 cicli) vi sono almeno due "1".



	00	01	11	10
A	A, 0	B, 0	C, 1	B, 0
B	A, 0	B, 1	C, 1	B, 1
C	A, 1	B, 1	C, 1	B, 1



$$\begin{array}{l} A=00 \\ B=01 \\ C=11 \end{array}$$

	00	01	11	10
w ₁ , w ₂	00, 0	01, 0	11, 1	01, 0
00	00, 0	01, 1	11, 1	01, 1
01	00, 1	01, 1	11, 1	01, 1
11	-	-	-	-
10	-	-	-	-

$(w_1, w_2)^{n+1}, z^n$

FF-T

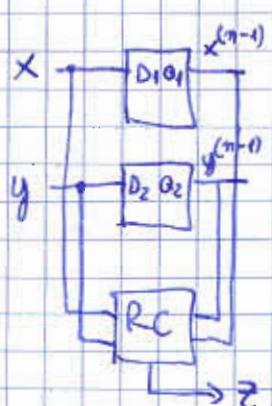
	00	01	11	10
W^n → W^{n+1}	0	0	0	0
T ₁	0	1	1	0
T ₂	1	0	0	1
T ₃	1	0	0	1
T ₄	1	1	0	0

0	0	1	0
0	0	1	0
1	1	0	1
-	-	-	-

0	1	1	1
1	0	0	0
1	0	0	0
-	-	-	-

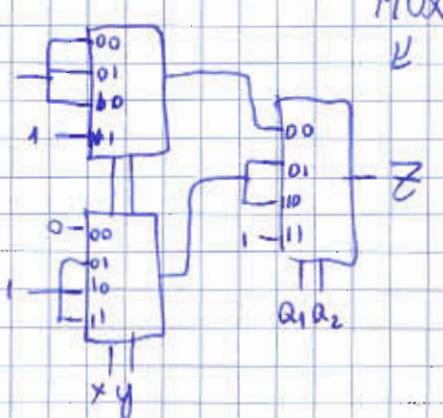
...

oppure



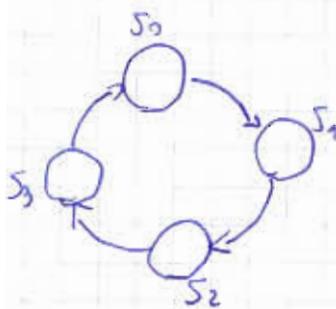
	00	01	11	10
00	0	0	1	0
01	0	1	1	1
11	1	1	1	1
10	0	1	1	1

Rai: rientrare RC con porte



CONTATORI

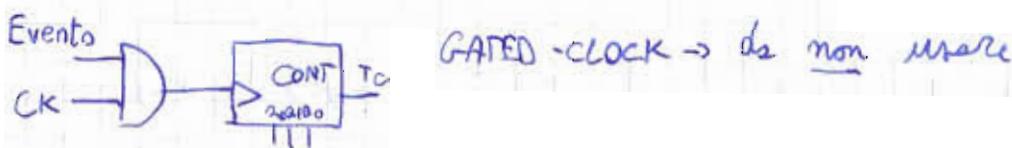
↪ visita un numero finito di stati periodicamente



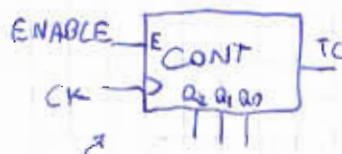
• MODULO = n° di stati visitati = 4 = BASE DI CONTEGGIO

Serve per

- misurare il tempo
- conteggiare degli eventi

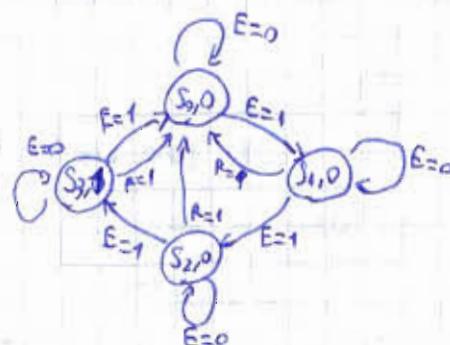


GATED-CLOCK → da non usare



↪ ingresso abilitato E=1, il contatore evolve

Circuito non più autonomo →



• INGRESSI DI CONTROLLO →

- ENABLE
- RESET } sincroni
- LOAD }
- RESET } asincroni
- LOAD }

non tutti sempre presenti

RESET → riporta il contatore nello stato 0. Ha priorità sull'enable.

LOAD → permette di "caricare" una particolare configurazione di stato

• CODICE UTILIZZATO PER LA CODIFICA DELLO STATO

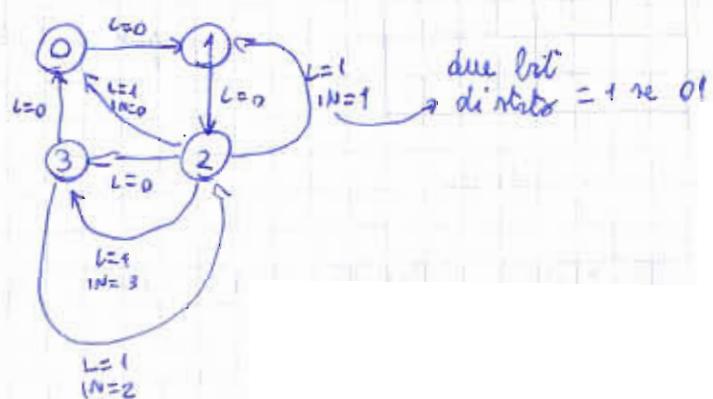
↪ numero minimo di bit

↪ binario (00-01-10-11-00...)

↪ Gray (00-01-11-10-00...)

↪ BCD

↪ numero non minimo di bit (codifica ridondante)



due bit di stato = 1 ne 01

↳ Johnson

↳ codice legato alla parità

↳ codice arbitrario

• TIPO DI FF UTILIZZATO

→ D

→ JK } più utilizzati

→ T

Progettare un contatore in base 5 che visita i seguenti stati
001 - 011 - 100 - 010 - 111 (codice non ridondante) usando FF-T

Non chiede nessun ingresso di controllo E!

	$y_2 y_1 y_0$	$(y_2 y_1 y_0)^{n+1}$
000	000	000
001	011	011
011	100	100
010	111	111
100	010	010
101	011	011
111	001	001
110	000	000

	$y_1 y_0$	00	01	11	10
0	-	011	100	111	
1	010	-	001	-	

$(y_2 y_1 y_0)^{n+1}$

	$y_1 y_0$	00	01	11	10
0	-	0	1	1	
1	1	-	1	1	

$$T_2 = y_1 + y_2$$

	$Q^n \rightarrow Q^{n+1}$	7
0	0	0
0	1	1
1	0	0
1	1	1

	$y_1 y_0$	00	01	11	10
0	-	1	1	0	
1	1	-	1	1	

$$T_1 = y_0 + y_2$$

→ per circuito...

	$y_1 y_0$	00	01	11	10
0	-	0	1	1	1
1	0	-	0	-	

$$T_0 = y_1 \bar{y}_2$$

CONTATORI BINARI CON FF-JK

$$(y_3 y_2 y_1 y_0)^{n+1} = (y_3 y_2 y_1 y_0)^n + \frac{1}{R_0}$$

$$y_i^{n+1} = \begin{cases} y_i^n & \text{se riportato in ingresso } R_i = 0 \\ \bar{y}_i^n & \text{se } R_i = 1 \end{cases}$$

$$= (y_i \cdot \bar{R}_i + \bar{y}_i \cdot R_i)^n = (y_i \oplus R_i)^n$$

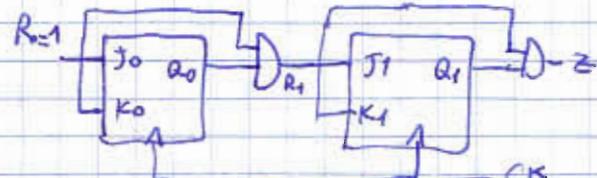
$$R_i = Y_{i-1} \cdot R_{i-1}, \quad R_0 = 1 \quad \rightarrow \text{con } J_i = K_i = R_i$$

R_i	0	1
0	0	1
1	1	0

$(R_i)^{n+1}$

0	11-
-	11-

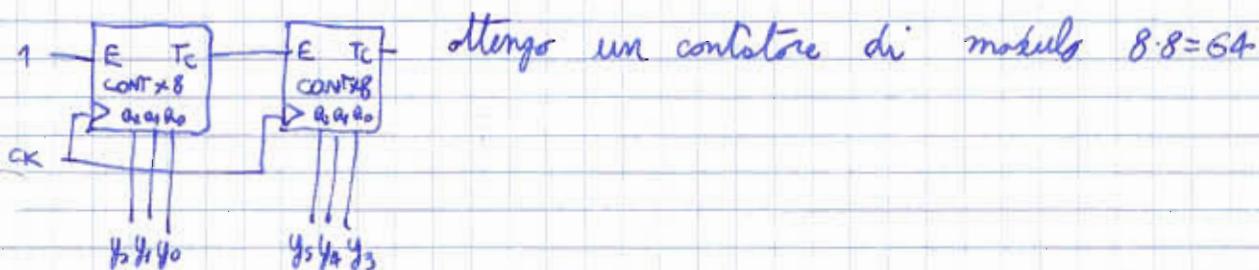
-	11-
0	111



$$J = R_i$$

$$K = R_i$$

Se al posto delle costanti, come R_i metto E (ingresso enable) ottengo un contatore binario dotato di enable.



Introduco ora l'ingresso di RESET.

ER	
00	hold
01	RESET
11	count
10	count

$Q_i \rightarrow Q_{i+1}$	J	K
0 0	0	0
0 1	1	1
1 0	0	-
1 1	1	0

R_i	00	01	11	10
Y _i	00	01	01	10
Y_{i+1}	00	01	01	10

$$R_i = \text{riporto in ingresso}$$

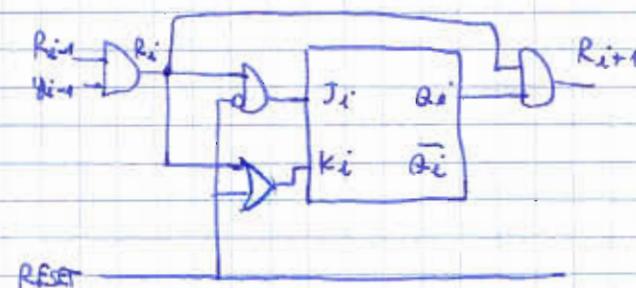
$R \rightarrow \text{reset}$

R_i	00	01	11	10
Q_i	0	0	0	1
\bar{Q}_i	1	1	1	0

$$J = R_i \cdot \bar{R}$$

R_i	00	01	11	10
Q_i	-	0	0	-
\bar{Q}_i	0	1	1	1

$$K = R_i + R$$



Introduco il LOAD

ERL	
000	HOLD
010	RESET
011	RESET
001	COUNT
100	I*
110	RESET \rightarrow prendere su LOAD
111	RESET

I^* \rightarrow configurazione specificata su ingressi esterni

1	1	1	1
L	E	I ₁	I ₂
R	a ₂	a ₃	a ₀
C	T _C	-	-

Nel caso in cui $L=0$, la tabella è come quella del contatore senza load.

Se considero un contatore senza reset:

R_i	00	01	11	10	00	01	11	10
0	0	0	1	1	0	1	1	0
1	1	1	0	0	0	1	1	0

$L=0$

$L=1$

R_i	00	01	11	10	00	01	11	10
0	0	0	1	1	0	1	1	0
1	-	-	(1)	(1)	-	(1)	-	-

$L=0$

$L=1$

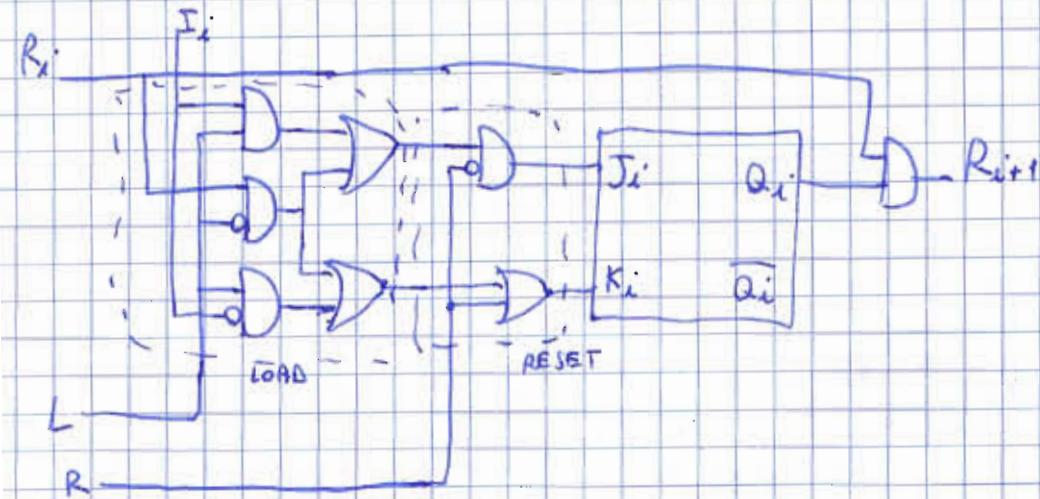
$$J_i = R_i \bar{L} + I_i L$$

R_i	00	01	11	10	00	01	11	10
0	-	-	(1)	(1)	-	-	-	-
1	0	0	(1)	(1)	1	0	0	1

$L=0$

$L=1$

$$K_i = R_i \bar{L} + I_i L$$



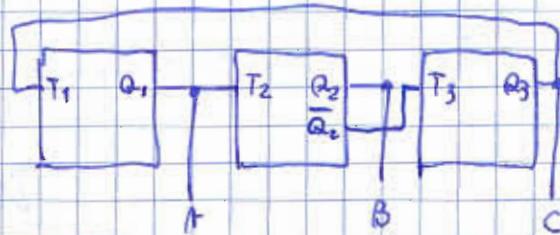
ES.D3 ESAME 12/2/06



F_1 periodo 3



F_2 periodo 6



$$y_i^{n+1} = (y_i \oplus T_i)^n$$

$$T_1 = Q_3$$

$$Q_1^{n+1} = (Q_1 \oplus Q_3)^n = Q_1 \bar{Q}_3 + \bar{Q}_1 Q_3$$

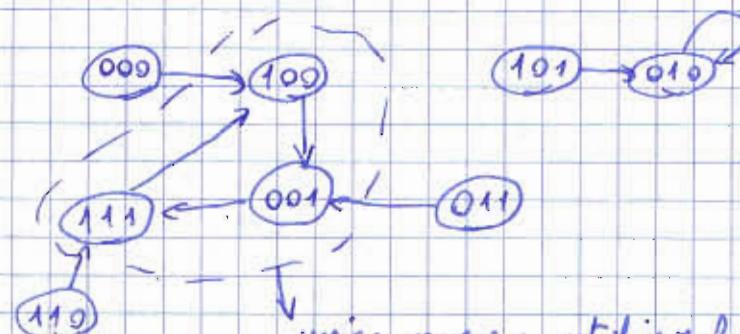
$$T_2 = Q_1$$

$$Q_2^{n+1} = (Q_2 \oplus Q_1)^n = Q_2 \bar{Q}_1 + \bar{Q}_2 Q_1$$

$$T_3 = \bar{Q}_2$$

$$Q_3^{n+1} = (Q_3 \oplus \bar{Q}_2)^n = Q_3 \bar{Q}_2 + \bar{Q}_3 \bar{Q}_2$$

$Q_3\ Q_2\ Q_1$	0 0 0	1 0 0	0 0 0
0 0 1	1 1 1	1 1 1	1 1 1
0 1 1	0 0 1	0 0 1	0 0 1
0 1 0	0 1 0	0 1 0	0 1 0
1 0 0	0 0 1	0 0 1	0 0 1
1 0 1	0 1 0	0 1 0	0 1 0
1 1 1	1 0 0	1 0 0	1 0 0
1 0 0	1 1 1	1 1 1	1 1 1



unico passo utilizzabile perché solo

Il circuito realizza un conteggio in base 3 100-001-111 ($Q_3\ Q_2\ Q_1$).

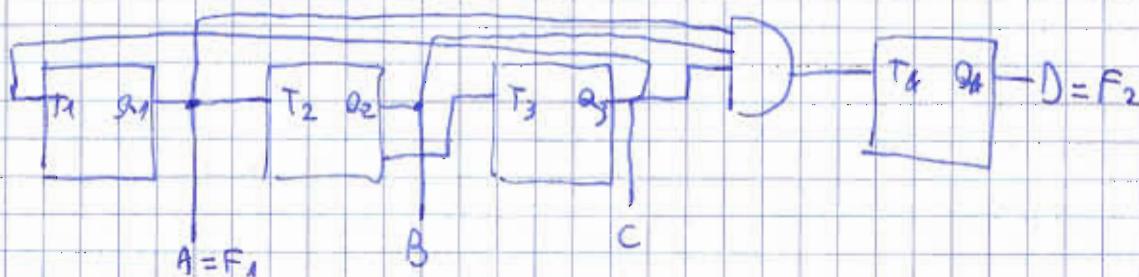
Espresso come ABC, cioè come $Q_1\ Q_2\ Q_3$, ottengo 001-100-111.

Questo mi permette di fare il conteggio per il periodo = 3. Per il periodo = 6, ho bisogno di un contatore in base 6, che non ho.

Mi basta però un contatore in base 2 da mettere in cascata.

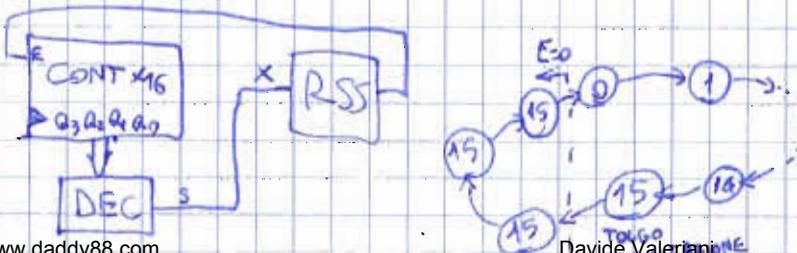
Il più semplice è $E = T_a$.

Seleziono lo stato 111 come stato di enable. Il circuito diventa:



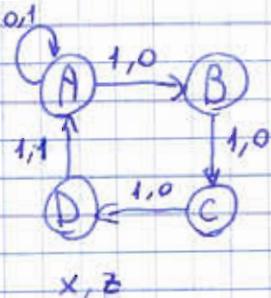
Q_1	0	1	1	0	1	1	0	$\dots = F_1$
Q_2	0	0	1	0	0	1	0	\dots
Q_3	1	0	1	1	0	1	1	$\dots = F_1$
T_a	0	0	1	0	0	1	0	
Q_4	0	0	0	1	1	1	0	$\dots = F_2$

Dato un cont. x 16 con enable realizzare un cont. x 19.



Mealy

$s \times$	0	1
A	A, 1 B, 0	B, 0
B	-	C, 0
C	-	D, 0
D	-	A, 1



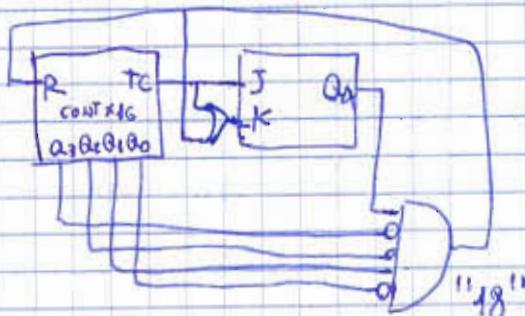
il contatore è fermato
pertanto $x=1$

Moore

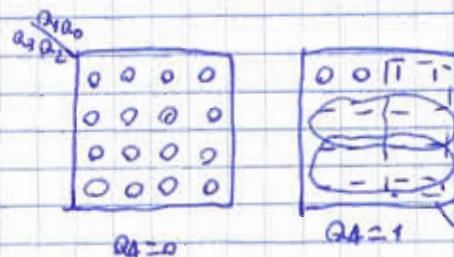
$s \times$	0	1
A	A, 1 B, 1	B, 1
B	C, 0	-
C	D, 0	-
D	A, 0	-

il contatore si
ferma a 0 per
3 intervalli

Dato un CONT x 16 con RESET realizzarne un CONT x 19



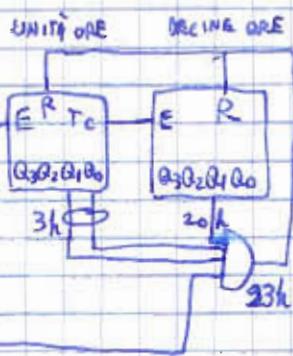
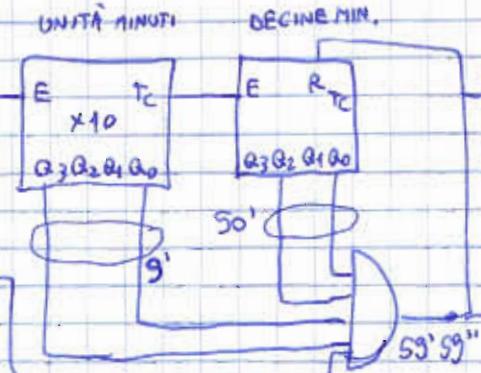
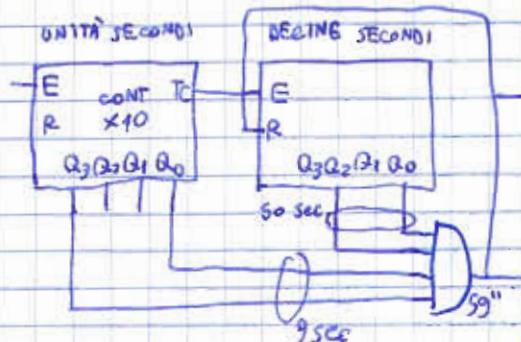
Vogendo semplificare l'AND



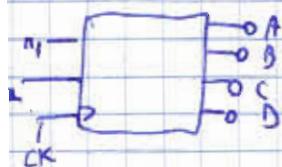
$$\text{Reset} = Q_4 Q_1 + Q_2 Q_4 + Q_3 Q_4 \quad \text{TRANSITO MINIMO}$$

BCD → conta da 0 a 9

CREARE OROLOGIO

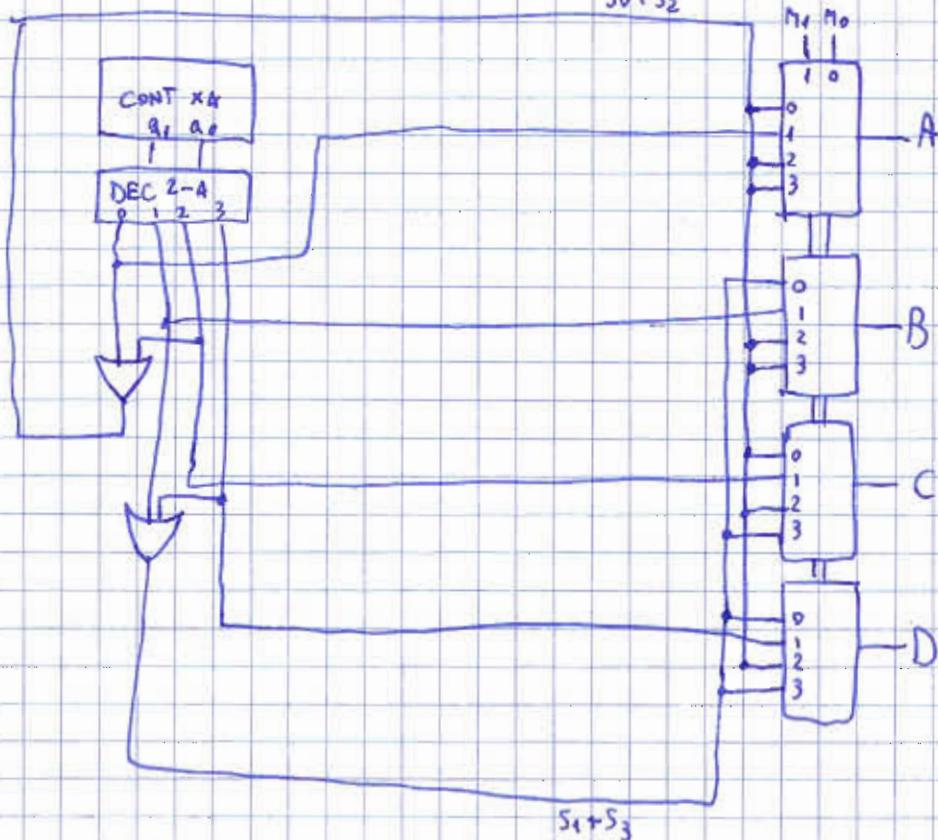


18/12/08



$M_1 M_0$	00	01	10	11
$Q_3 Q_2 Q_1 Q_0$	AC	A	ABCD	AB
00	BD	B	-	CD
01	AC	C	ABCD	AB
10	BD	D	-	CD
11				

Luca di Molale

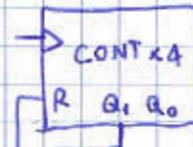
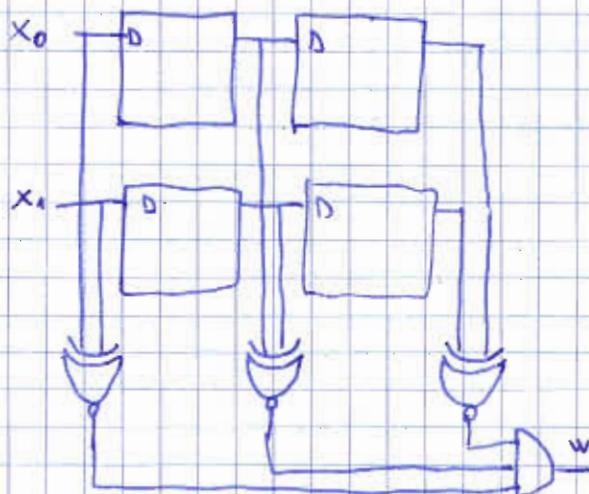


22/12/2003 D2



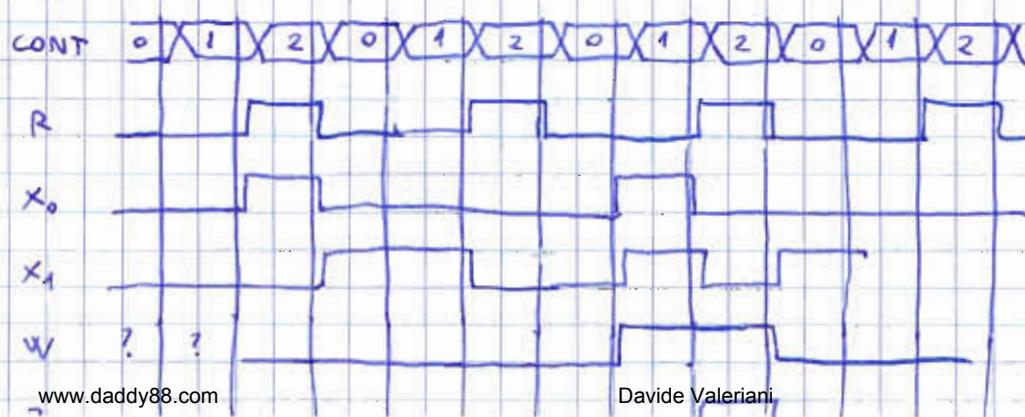
$$\begin{array}{l} X_0 \quad 0 \quad 1 \quad 1 \quad 0 \quad 0 \quad 1 \quad 1 \quad 1 \quad 0 \\ X_1 \quad 1 \quad 0 \quad 0 \quad 1 \quad 0 \quad 1 \quad 1 \quad 1 \quad 0 \\ Z \quad 0 \quad 0 \quad 0 \quad , \quad 0 \quad 0 \quad 1 \quad , \quad 0 \quad 0 \quad 0 \quad 1 \end{array}$$

consideriamo perciò che
l'uscita varia qui.

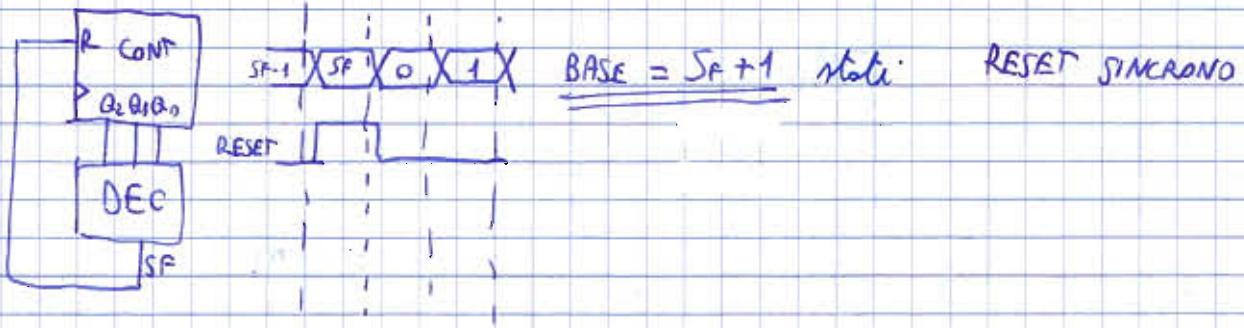


quando ho
to resettato
nel contatore
3 stato

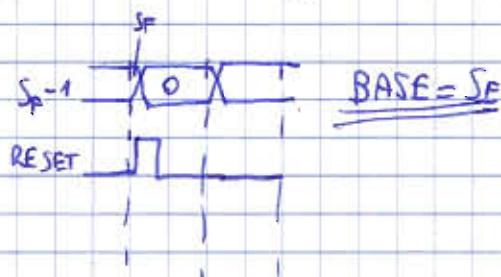
ottengo
l'uscita
ritardata!
↓



oppure mettere
3 FF-D nel
registro z
scorrimento
invece che
nolo 2.
In questo modo
analizzarlo i
intervalli dopo

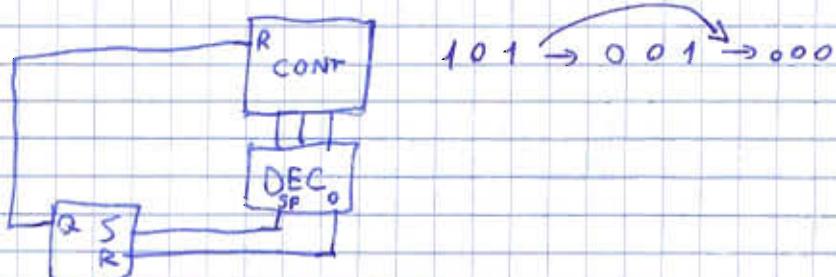


USO DI RESET ASINCRONO PER RIDUZIONE BASE



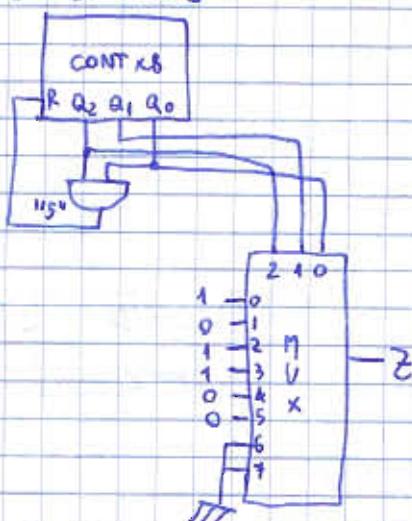
Il reset non dura un intero intervallo di clock. Tuttavia, se uno dei FF è più veloce degli altri, si va in una configurazione in cui il RESET non è attivo \rightarrow ERRORE

\rightarrow SOLUZIONE

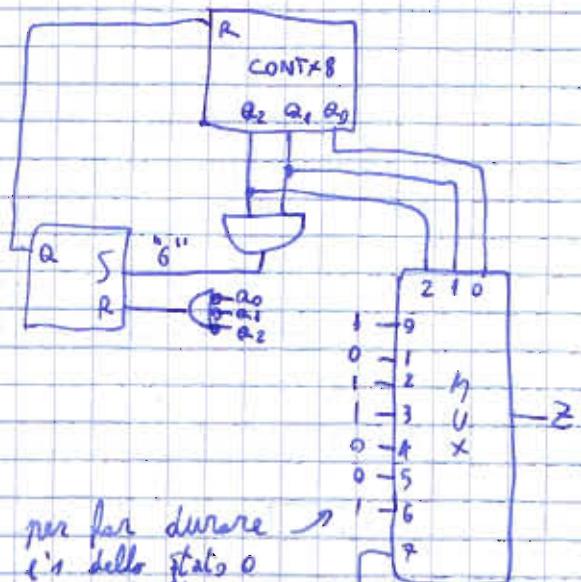


Dato un CONT X8 con RESET o/e LOAD sincroni realizzare una forma d'onda periodica 101100

MODO 1 - RESET SINCRONO



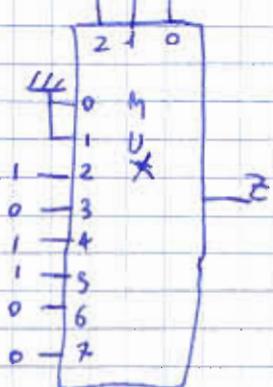
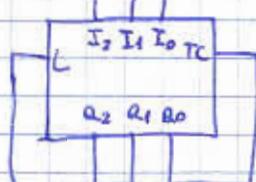
MODO 2 - RESET ASINCRONO



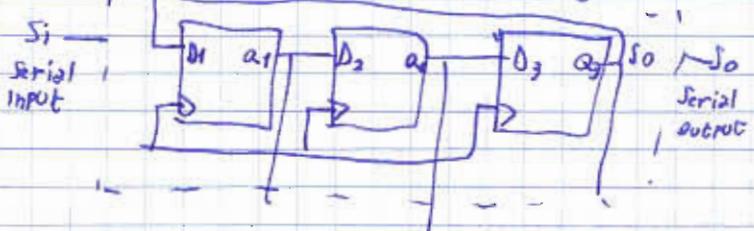
MODO 2 - LOAD A PARTIRE DALLO STATO FINALE

7 → 2 → 3 → 4 → 5 → 6 → 7 → 2...

"110" → 0 1 0



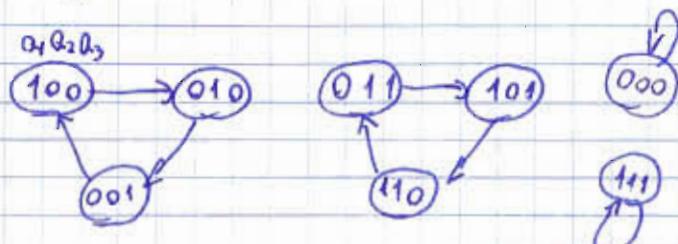
REGISTRO A SCORRIMENTO



$$Q_1^{n+1} = D_1^n = Q_3$$

$$Q_2^{n+1} = D_2^n = Q_1$$

$$Q_3^{n+1} = D_3^n = Q_2$$



con Nbit
↓
BASE = N

CONTATORE AD ANELLO

Problema dell'inizializzazione → PRE-SET asincrono

In questo modo, il contatore che parte da 000 viene portato = 1 nel primo FF-D alla sequenza one-hot!



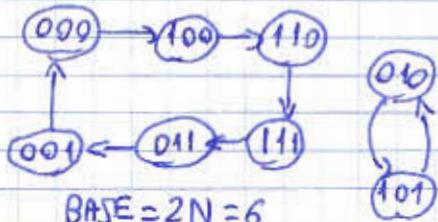
CONTATORE JOHNSON o

RIEMPIMENTO / SVUOTAMENTO

↳ da 1 da sinistra
di 0 da destra

Q ₂ Q ₁ Q ₀	000	001	011	100	101	110	111
000	1	0	0	0	0	0	0
001	0	0	0	1	0	0	0
011	0	0	1	0	1	0	0
100	1	0	1	0	0	1	0
101	0	1	0	0	1	0	0
110	0	1	1	0	0	0	1
111	1	1	1	1	1	1	1

$$(Q_2 Q_1 Q_0)^{n+1}$$



$$\text{BASE} = 2N = 6$$

Q ₂ Q ₁ Q ₀	00	01	11	10
00	S ₁	S ₆	S ₅	S ₄
01	S ₅	S ₂	S ₄	S ₃
11	S ₄	S ₃	S ₂	S ₁

$S_1 = \bar{Q}_1 \bar{Q}_3$ $S_4 = Q_1 Q_3$ Ogni stato è vicino a un
 $S_2 = Q_1 \bar{Q}_2$ $S_5 = \bar{Q}_1 Q_2$ don't care → mi bastano
 $S_3 = Q_2 \bar{Q}_3$ $S_6 = \bar{Q}_2 Q_3$ AND a 2 ingressi qualunque
 sia la base.

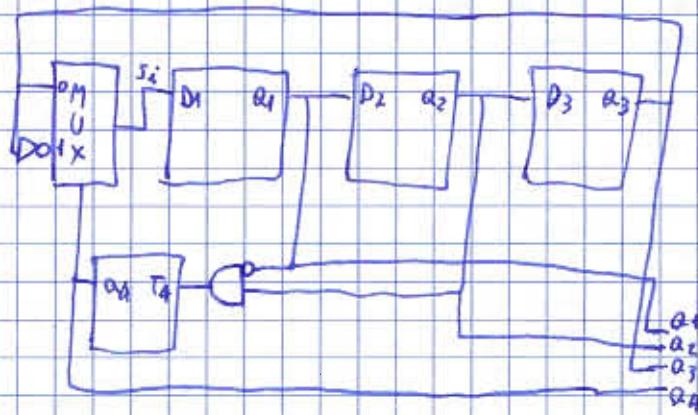
ESAME 2-2006

Dato uno shift register a 3 bit realizzare un contatore in base 3.

CONTEGGIO
JOHNSON

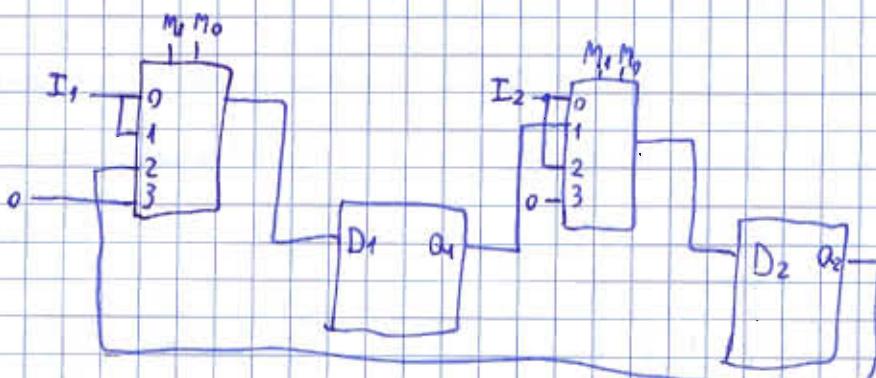
0	0	0
1	0	0
1	1	0
1	1	1
0	1	1
0	0	1

CONTEGGIO
AD ANELLO



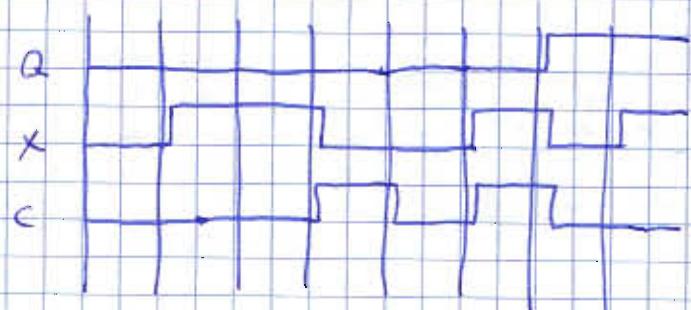
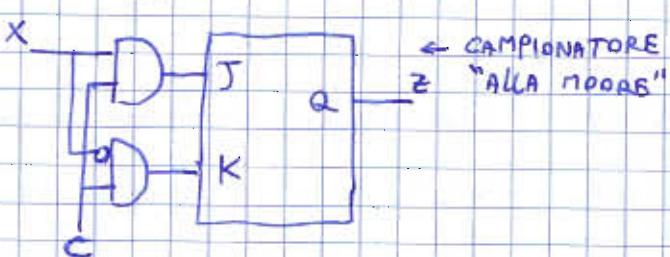
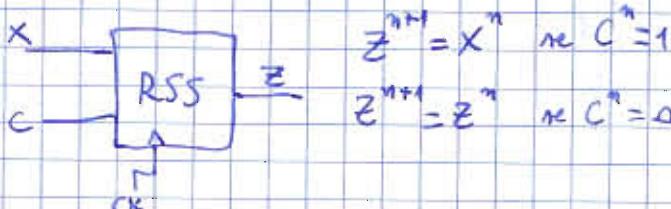
Ritorno da Johnson ad anello

000	1	001	001 rappresenta la fine
100	1	di entrambe le sequenze	
110	1		
111	1		
→ 011	1	configura la situazione che lo precede → 011 = 010 → Q ₁ =0 e Q ₂ =1	
001	0		
100	0		
010	0	Così gli segue lo stato dell'altro	
001	1	metodo	



M ₁ M ₀	
0 0	REGISTRO PARALLELO
0 1	SHIFT RIGHT
1 0	SHIFT LEFT
1 1	RESET SINCRONO

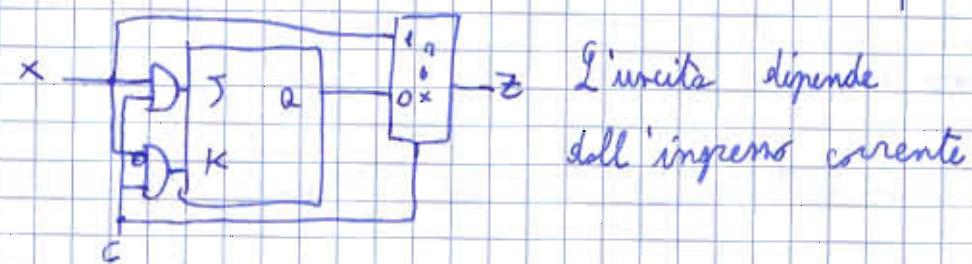
CAMPIONAMENTO SINCRONO



CAMPIONATORE "ALLA MEALY"

$$Z^n = X^n \text{ e } C^n = 1$$

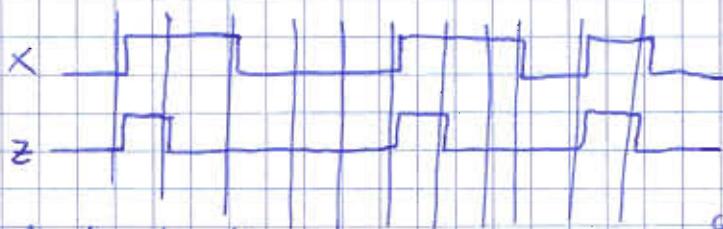
$$Z^{n+1} = Z^{n+1} \text{ e } C^n = 0$$



L'uscita dipende
dall'ingresso corrente

DERIVATORE SINCRONO

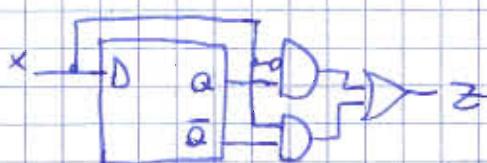
In fronte di salita



In fronte di discesa

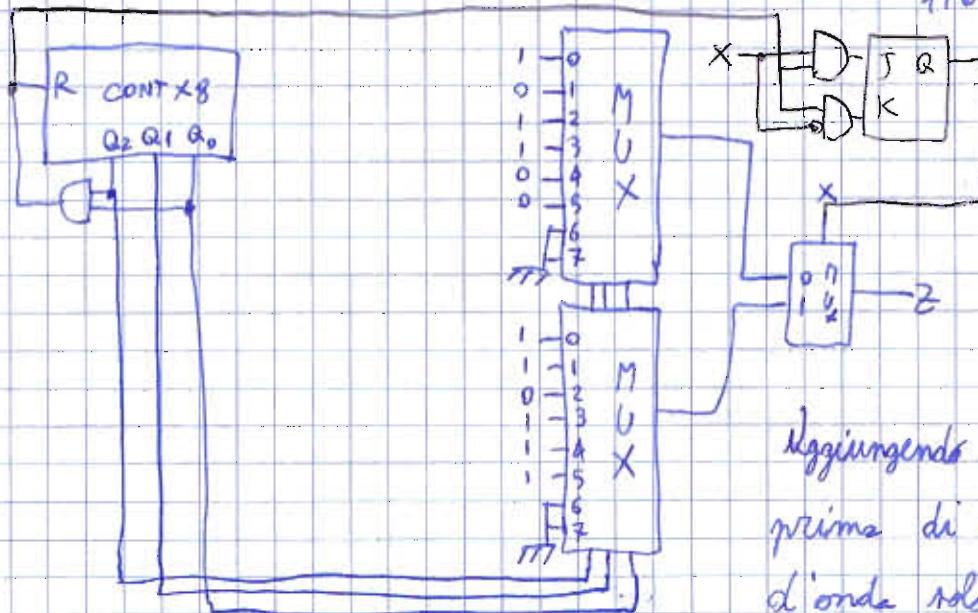


In entrambi i fronti



Circuito che realizza una delle forme d'onda

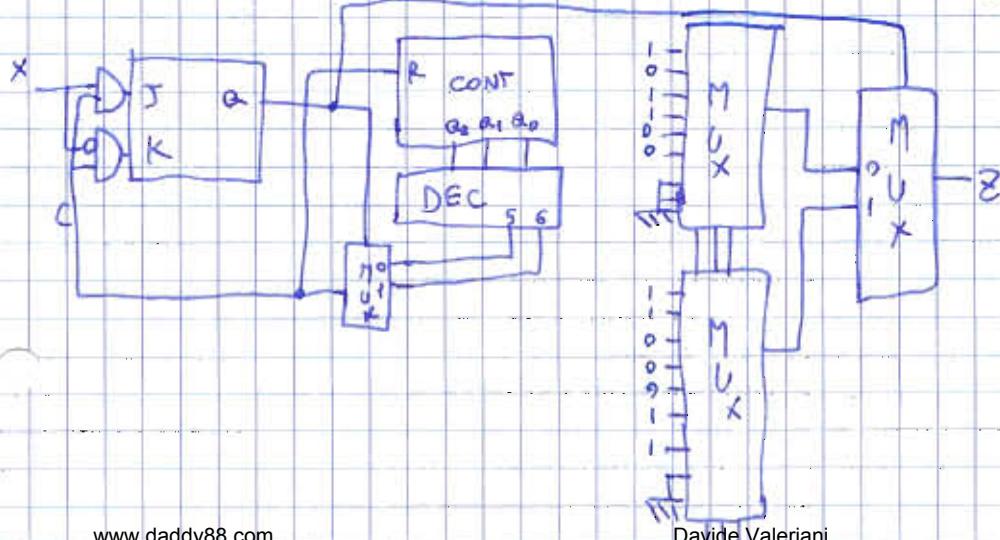
101100
110111

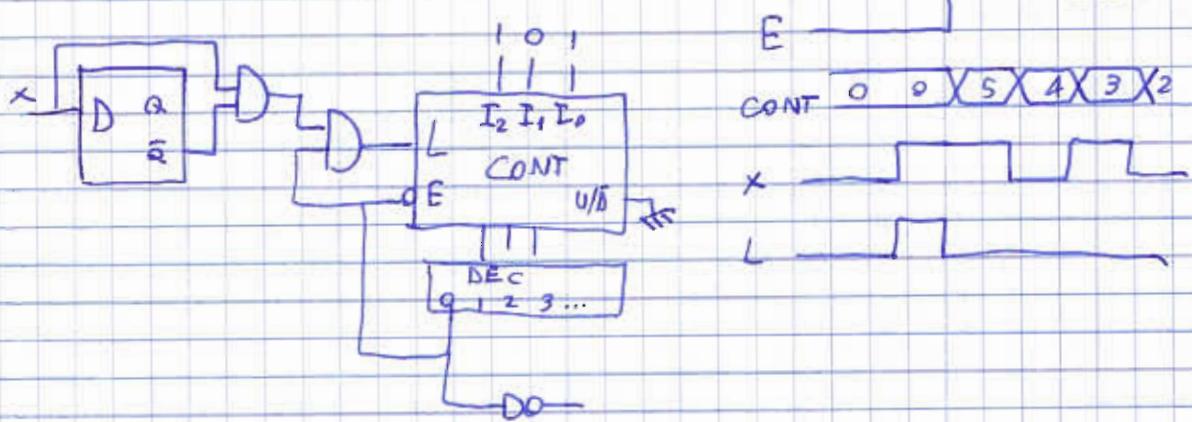


Aaggiungendo la parte in nero, prima di commutare le forme d'onda solo quando sono finite.

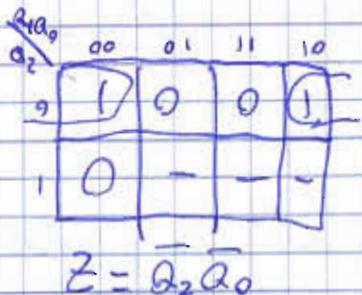
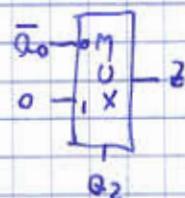
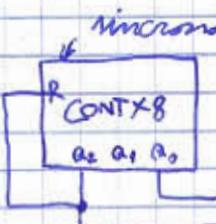
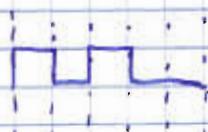
Questo funziona solo se le sequenze hanno = durata.

Nel caso in cui si abbia da scegliere $\underbrace{101100}_6$ o $\underbrace{110111}_7$

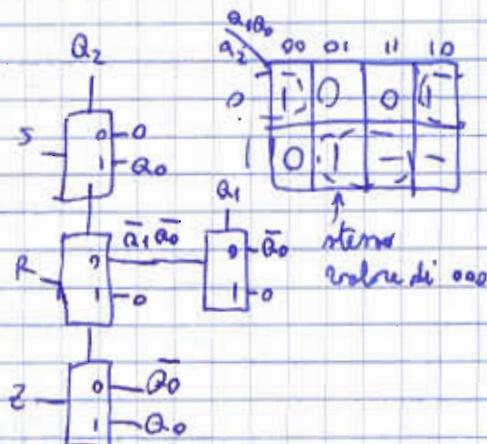
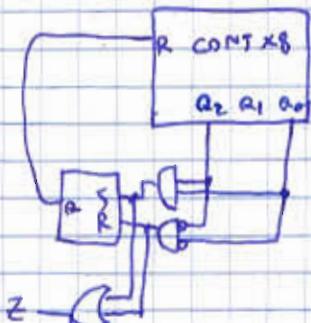




CONTENORE



wincing



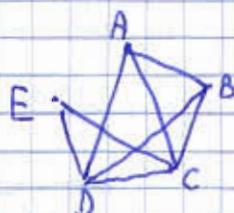
$$Z = \bar{Q}_2 \bar{Q}_0 + Q_2 Q_0$$

MINIMIZAZIONE

	0	1
A	-1, 0	0, -1
B	0, 1	-1, 0
C	0, -1	-1, 0
D	1, -1	-1, -1
E	-1, 1	0, -1

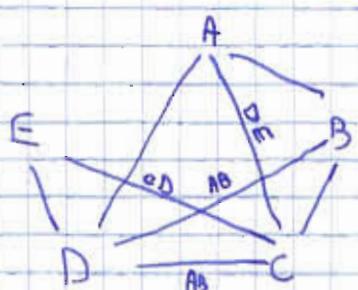
B	V				couple compatible
C	V	AB	V		
D	V	AB	AB		
E	X	X	CD	V	
	A	A	C	D	

copyie AB, AC, AD, BC, BD, CD, CE, DE
compatibili:



$$\alpha = \{A, B, C, D\} \quad \beta = \{C, D, E\}$$

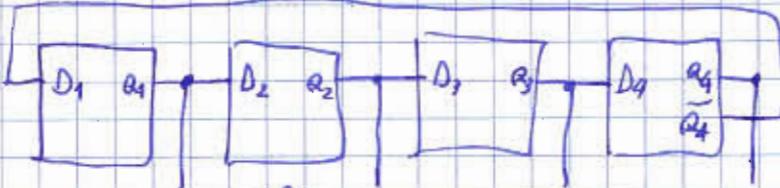
Casi dirigente:



$$\text{or } \gamma = \{ABC\} \text{ or } \delta = \{DEF\}$$

opposite

$$\gamma = \{AB\} \quad \delta = \{CDE\}$$



Contatore Johnson in base
 $2 \cdot 4 = 8$.

z_0	Q_0	Q_1	Q_2	Q_3	Q_4
z_1	0000	0000	0000	0000	0000
z_2	1000	1100	0100	0010	0001
z_3	1100	0110	0010	0001	0000
z_4	1110	0010	0001	0000	0000
z_5	1111	0001	0000	0000	0000
z_6	0111	-	-	-	-
z_7	0011	-	-	-	-
z_8	0001	-	-	-	-
z_9	0000	-	-	-	-

a_1	a_2	Q_0	Q_1	Q_2	Q_3	Q_4
0	0	0000	0000	0000	0000	0000
0	1	1100	0110	0010	0001	0000
1	0	0110	0010	0001	0000	0000
1	1	0010	0001	0000	0000	0000
		-	-	-	-	-
		1111	0001	0000	0000	0000
		1110	0010	0001	0000	0000
		0111	-	-	-	-
		0011	-	-	-	-
		0001	-	-	-	-
		0000	-	-	-	-

$$Z_1 = Q_1 \bar{Q}_2$$

$$Z_2 = Q_2 \bar{Q}_3$$

$$Z_6 = \bar{Q}_2 Q_3$$

$$Z_3 = Q_3 \bar{Q}_4$$

$$Z_7 = \bar{Q}_3 Q_4$$

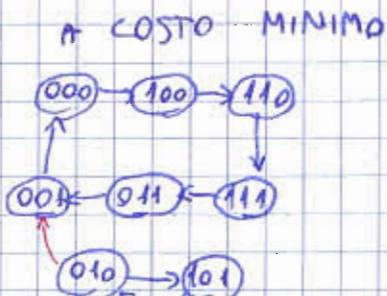
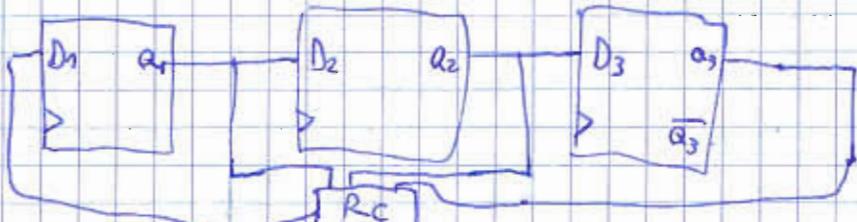
$$Z_4 = Q_1 Q_2$$

$$Z_5 = \bar{Q}_1 Q_2$$

$$Z_0 = Q_1 \bar{Q}_4$$

N FF	BASE	DECODIFICA USCITE
cont. brim.	2^N	2^N AND a N ingressi
B	$K = \log_2 B$	FF, B AND a K ingressi

cont. anello	N	Gratis
	B	/
cont. Johnson	$2N$	$2N$ AND a 2 ingressi
	B	$\frac{B}{2}$ FF, B AND a 2 ingressi



In questo modo risolvo gli eventuali errori modificando un solo stato.

a_1	a_2	Q_0	Q_1	Q_2	Q_3
0	0	0000	0000	0000	0000
1	0	1100	0110	0010	0001

1	0	0	0
1	-	0	1

$$D_1 = \bar{Q}_2 \bar{Q}_3 + Q_1 Q_2$$

