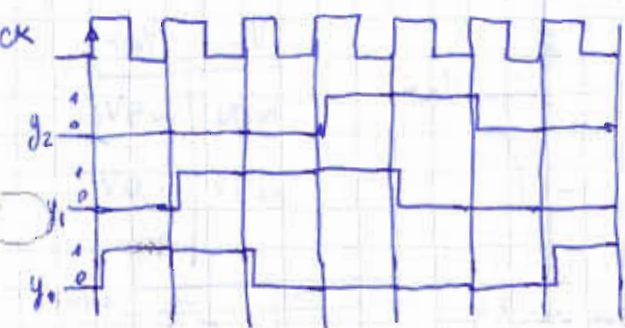
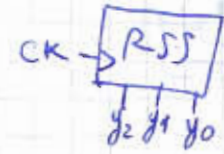


Progettare un contatore con sequenza 000-001-011-010-110-100 con FF-JK, NAND

Generare due forme d'onda.



RETE SEQUENZIALE SINCRONA



EVOLUZIONE

TABELLA TRANSIZIONI
(y2 y1 y0)ⁿ (y2 y1 y0)ⁿ⁺¹

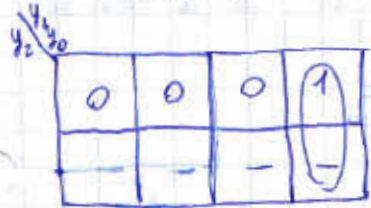
0	0	0	0	0	1
0	0	1	0	1	1
0	1	1	0	1	0
0	1	0	1	1	0
1	0	0	0	0	0
1	0	1	-	-	-
1	1	1	-	-	-
1	1	0	1	0	0

y2 \ y1 y0	00	01	11	10
0	001	011	010	110
1	000	-	-	100

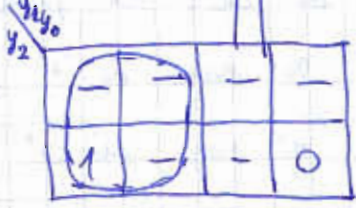
(y2 y1 y0)ⁿ⁺¹

y ⁿ → y ⁿ⁺¹	J	K
0 0	0	-
0 1	1	1
1 0	0	-
1 1	1	0

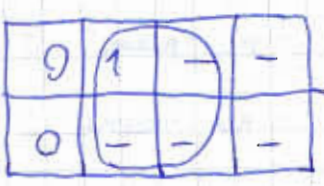
Evidenzio le configurazioni che commutano



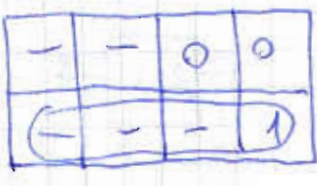
J₂ = y₁ y₀



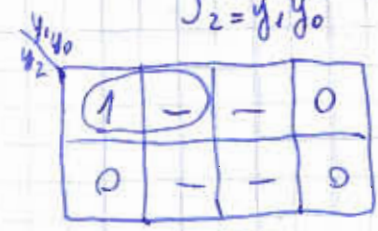
K₂ = y₁



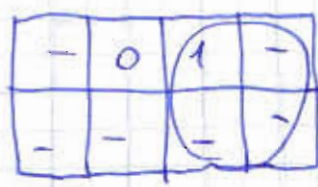
J₁ = y₀



K₁ = y₂

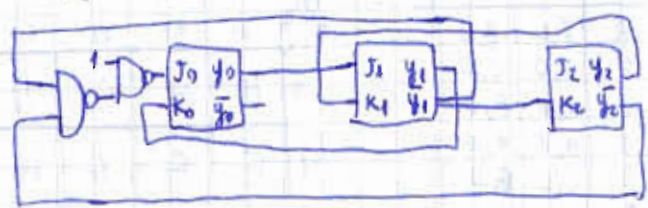


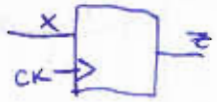
J₀ = y₂ y₁
= (y₂ ↑ y₁) ↑ 1



K₀ = y₁

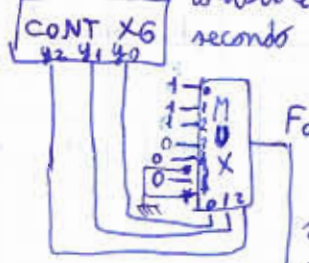
NAND ↑ J₂ = (y₁ y₀) + 0 = (y₁ ↑ y₀) ↑ 1
NOR ↓



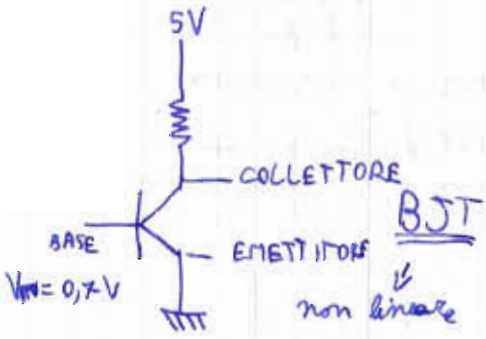
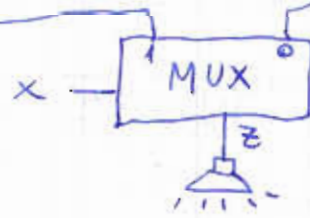
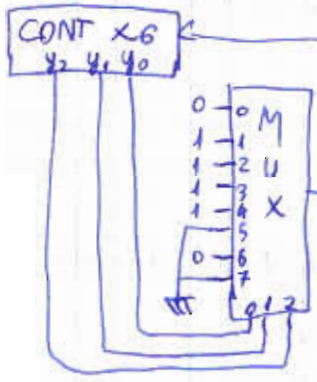


$X=0 \quad F_0 = \{1, 1, 0, 1, 0, 0\}$
 $X=1 \quad F_1 = \{0, 1, 1, 1, 0, 1\}$

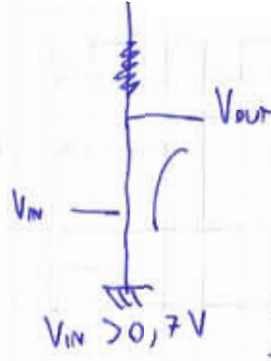
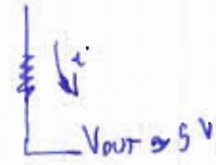
Contatore che visita i canali secondo l'ordine di prima



* stati non usati, meglio metterli a un valore definito.



non lineare $V_{IN} < 0,7V$



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



Immunità ai disturbi grazie a questa legge.

V_{IN}	V_{OUT}
0	1
1	0

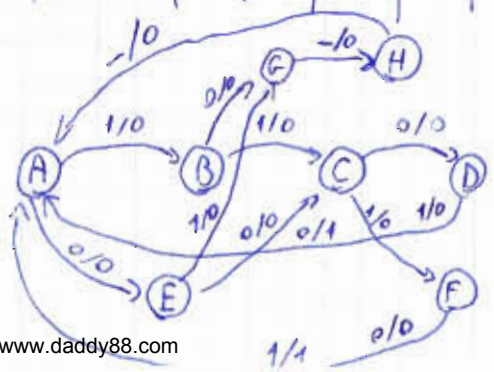
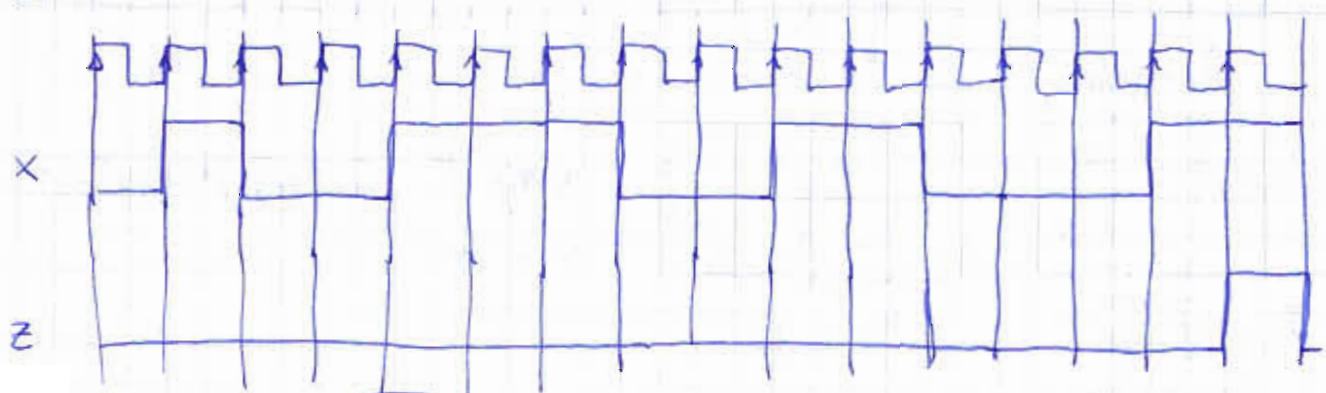
\rightarrow NOT

ES 1 del 14/09/2007

07/10/2008

$x \rightarrow \{01101110\dots\}$ $Z=1$ per 1 periodo di clock nel quarto bit n e solo se

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



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

$Y_2 Y_1 Y_0$	00	01	11	10
0	A	B	C	D
1	E	F	G	H

1) Diagramma degli stati

2) Tabella di flusso



→ ingressi
stato successivo, uscita

3) Minimizzazione degli stati

4) Tabella delle transizioni (mappe di codifica)



5) Sintesi delle funzioni

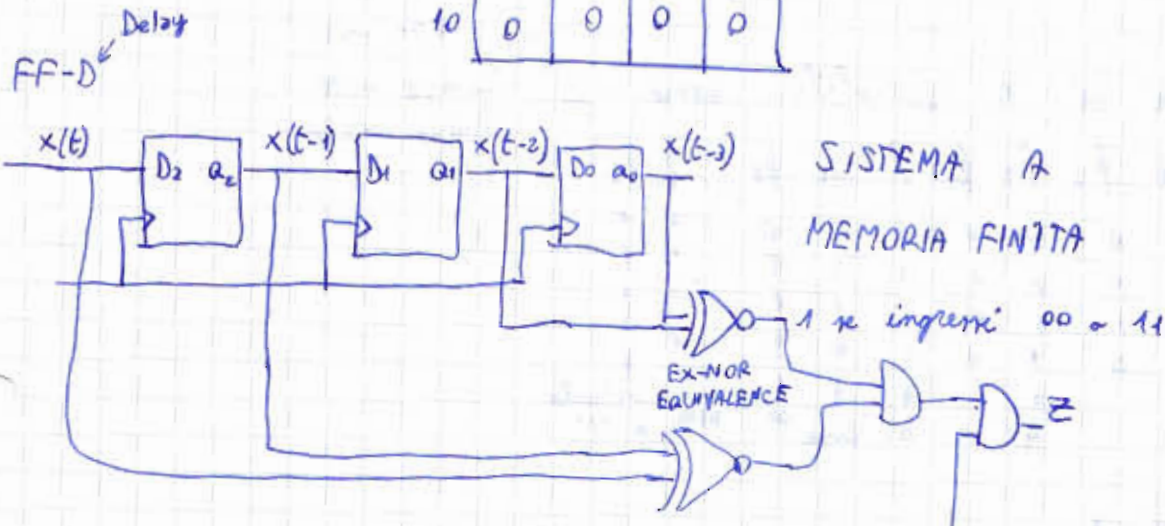
- eccitazioni FF

- uscita

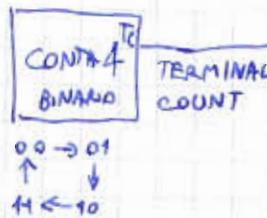
$y_2 y_1 y_0 \backslash x$	00	01	11	10
00	1	0	1	0
01	1	0	0	0
11	0	1	1	1
10	0	0	0	0

$$y_2^{(n+1)} = \bar{y}_2 \bar{x} \bar{y}_1 + y_2 x \bar{y}_1 \bar{y}_0 + x y_1 y_0 + y_2 y_1 y_0$$

6) Disegno del circuito



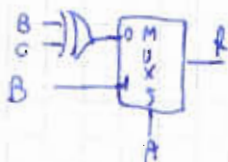
Controlla che si rispetti la sincronizzazione di controllare ogni 4 bit



ES 3

Dato una tabella di verità creare il circuito con un multiplex a due vie e porte logiche.

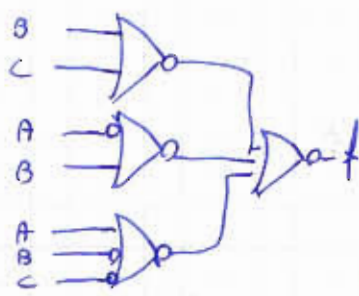
ABC	f
000	0
001	1
010	1
011	0
100	0
101	0
110	1



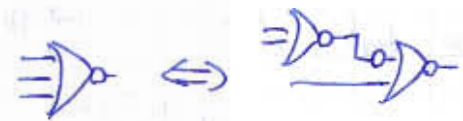
$B \backslash C$	00	01	11	10
0	0	1	0	1
1	0	0	1	-

Sintesi a NOR

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



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



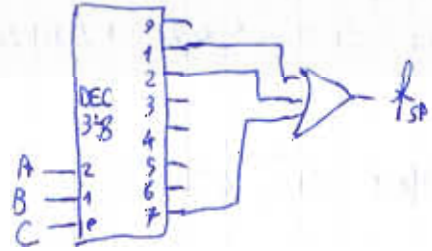
Sintesi NAND:

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

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

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

DECODER → identifica tutti i mintermi



Possibili funzioni di n variabili sono 2^{2^n} / $n=1 \rightarrow 4$, $n=2 \rightarrow 16$, $n=3 \rightarrow 256$

A	B	f0	f1	f2	f3	f4	f5	f6	f7	...	f14	f15
0	0	0	0	0	0	0	0	0	0	1	1	1
0	1	0	0	0	0	1	1	1	1	1	1	1
1	0	0	0	1	1	0	0	1	1	1	1	1
1	1	0	1	0	1	0	1	0	1	0	1	1

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

X	f0	f1	f2	f3
0	0	0	1	1
1	0	1	0	1

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

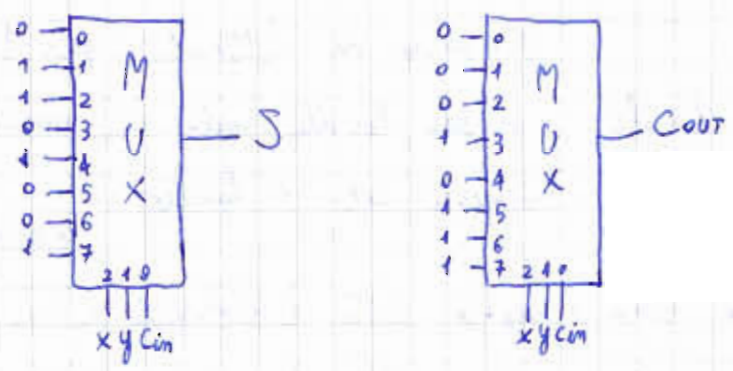
09/10/08

PROVA INTERMEDIA → 19/11/2008

SINTESI → dalla tabella di verità al circuito

ANALISI → dal circuito alle tabelle di verità.

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



es.

$$f(x_1, x_2, x_3) = x_1 \bar{x}_2 + x_3 x_2 = \bar{x}_1 (0 \cdot \bar{x}_2 + x_3 \cdot x_2) + x_1 (1 \cdot \bar{x}_2 + x_3 \cdot x_2) =$$

$$= \bar{x}_2 (x_1 \cdot 0 + x_3 \cdot 1) + x_2 (x_1 \cdot 1 + x_3 \cdot 1) =$$

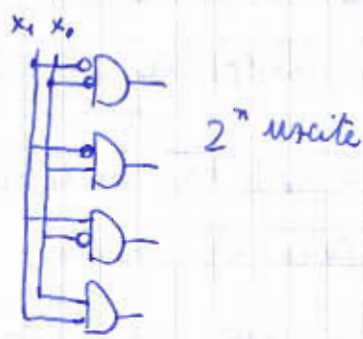
$$= \bar{x}_2 \bar{x}_1 (0) + \bar{x}_2 x_1 (1) + x_2 \bar{x}_1 (x_3) + x_2 x_1 (x_3)$$

ESPRESSIONI GENERALI SP
 ESPRESSIONI GENERALI PS
 REALIZZAZIONE CON MUX

ESPRESSIONI CANONICHE SP
 ESPRESSIONI CANONICHE PS
 REALIZZAZIONE CON DECODER

DECODER

Ha n ingressi e 2^n uscite.
 Circuito universale. Mette a disposizione tutti i mintermi della funzione.



SINTESI NOR → PDS - PS
 SINTESI NAND → SDP - SP

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

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

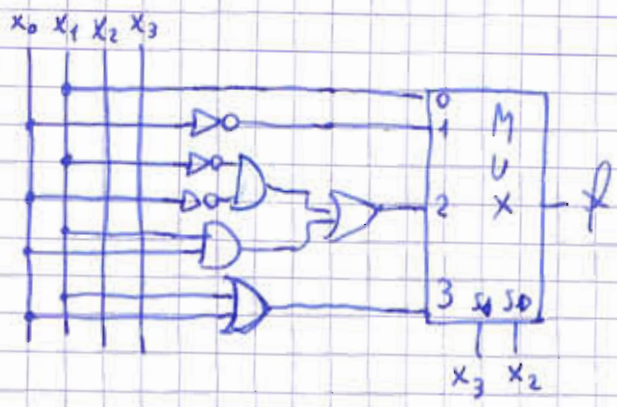
Funzioni con MUX e 4 vie + qualcosa.

$x_3 x_2$	00	01	11	10	
00	0	-	1	1	← 0
01	1	0	0	1	← 1
11	0	1	1	1	← 3
10	1	0	1	0	← 2

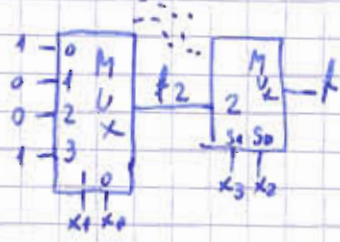


Il MUX è l'ultimo elemento del circuito. Due variabili di selezione (ad es. x_2, x_3) e 2^k funzioni di $n-k$ variabili.

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



Per fare f_2 con un MUX



ALGORITMO DI QUINE-MCCLUSKEY

Due fasi:

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

Parto dall'insieme dei mintermi e riduco.

ES.

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

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

$\bar{a} b c d$ $1-0-$ ← ne ignora
 $1-0-$ ← una che m interessa

IMPLICANTI PRIMI

$\bar{a} b d, \bar{b} \bar{c} d, \bar{e} c d, b c d, a b d, a \bar{c}$ Quelli non marcati

SECONDA FASE → tabella di coperture: righe → implicanti primi, colonne → mintermi

ci sono due criteri per semplificare la tabella:

- CRITERIO DI ESSENZIALITÀ → scegli le colonne che hanno una sola x: quel minterm andrà preso, cancellando tutte le colonne che lo copre.
- CRITERIO DI DOMINANZA → 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

ES.

Righe ordinate per costo

	1	3	7	8	9	12	13	15
$a\bar{c}$				x	x	x	x	
$\bar{b}\bar{c}d$	x				x			
abd							x	x
bcd			x					x
$\bar{a}cd$		x	x					
$\bar{a}\bar{b}d$	x	x						

IMPLICANTI PRINCIPALI = $a\bar{c}$

	1	3	7	15
r_2	x			
r_3				x
r_4			x	x
r_5		x	x	
r_6	x	x		

elimino le righe coperte da altre

	1	3	7	15
P_5			x	x
P_6	x	x		

IMPL. PRINCIPALI = $\{a\bar{c}, bcd\}$

	1	3	7	15
P_5			x	x
P_6	x	x		

IMPL. PRINC. = $\{a\bar{c}, bcd, \bar{a}\bar{b}d\}$

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

	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 → faccio delle prove

Prendo r_1 come implicante

Non prendo r_2 (considero dominati)

	C_3	C_4	C_5
r_2	x	x	
r_3		x	x
r_4	x		x

	C_3	C_4	C_5
r_2	x	x	
r_3	x	x	x
r_4	x		x

$\{r_3, r_4\}$

ancora ciclica.

METODO DI PAIRICK

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

ella fine mi dice quali colonne devo prendere...

$$= r_1 r_2 r_3 + r_1 r_2 r_4 + r_1 r_3 r_4 + r_2 r_3 r_4 + \underline{r_3 r_4} \text{ prendo la meno costosa!}$$

oppure

23/10/2008

ESEMPIO S.P. DI RUINE-MCCLUSKEY

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

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)
A 0 1 0 0*	2/6 0 - 1 0*	
6 0 1 1 0*	A/6 0 1 - 0*	
9 1 0 0 1*	6/7 0 1 1 -	
B 0 1 1 1*	9/11 1 0 - 1	
11 1 0 1 1*	7/15 - 1 1 1	
15 1 1 1 1*	11/15 1 - 1 1	

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

Implicanti primi

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

P_5 è essenziale!

$$L = \{P_5\}$$

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

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

	7	15
P_3	x	x
P_4		x
P_1	x	

P_3 ← domina

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

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

N.OP. = 4
N.IN. = 11



N.LEP. = 8

ESEMPIO P.S.

$$z = \text{TTM}_4(1, 3, 5, 8, 10, 12, 13, 14)$$

FASE 1

	a b c d	a b c d	a b c d
1	0 0 0 1 *	1/3 0 0 - 1	8/10/11/14 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 *	10/11 1 - 1 0 *	
13	1 1 0 1 *	12/13 1 1 0 -	
14	1 1 1 0 *	12/14 1 1 - 0 *	

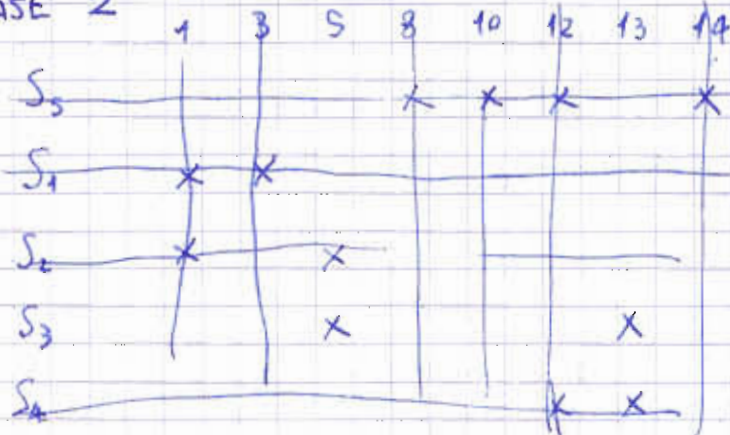
Implicati primi:

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

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

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

FASE 2



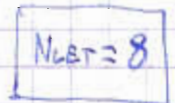
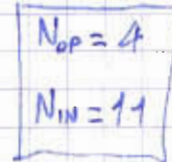
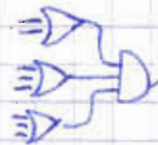
$$C = \{S_5\}$$

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

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

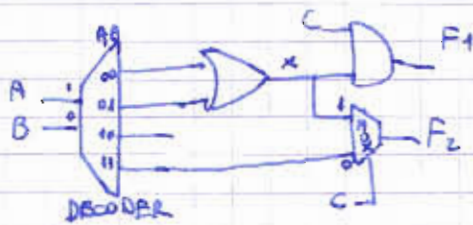
$$z = (\bar{a} + d) \cdot (a + b + \bar{d}) \cdot (\bar{b} + c + \bar{d})$$

$$= (\bar{a} \downarrow d) \downarrow (a \downarrow b \downarrow \bar{d}) \downarrow (\bar{b} \downarrow c \downarrow \bar{d})$$



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

ES. ESAME 16/2/2005



OR, NOR a 2 ingressi

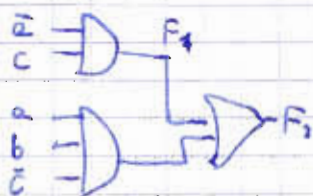


$$x = \bar{a}b + a\bar{b} = \bar{a}$$

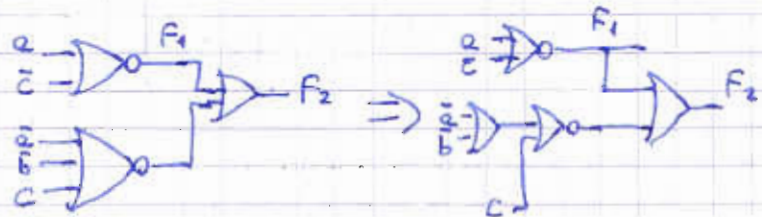
$$F_1 = \bar{a}c$$

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

il momento



applico de Morgan



Per circuiti a più uscite, non sempre la sintesi ottimale la ottengo considerando solo gli implicantti primi.

La copertura ottima di una funzione multiscelta passa attraverso gli implicantti comuni. Il metodo di Q-MC consente di minimizzare facilmente questi sistemi.

Si aggiunge un'etichetta a fianco ad ogni mintermo composto da n bit (con $n = n^{\circ}$ funzioni d'uscita) che vale 1 se tale mintermo vale 1 per quella funzione. Si fa poi l'AND tra le etichette e se è uguale all'etichetta di uno dei due, viene marcato, altrimenti no.

es. $011 \ 101$
 $001 \ 011 \Rightarrow 0-1 \ 001$ nessuno marcato

A casi per il risultato dell'AND

- tutti 0 \rightarrow scarto il mintermo
- non coincide con nessuna etichetta \rightarrow non marco nessuno
- coincide con un'etichetta \rightarrow marco il mintermo 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. \rightarrow anche ripetendoli per quelli comuni.

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

ES. Data una funzione combinatoria, sintetizzarla con MUX a 2 variabili di selezione.

$c, d = 0$

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

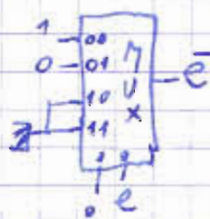
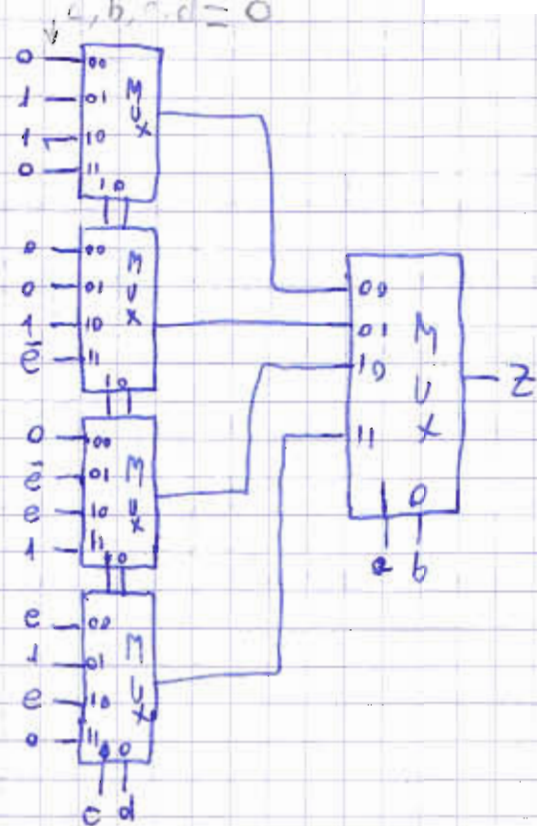
$e = 0$

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

$e = 1$

$a, b = 0$

Per ottenere \bar{e} ho bisogno di un altro multiplexer



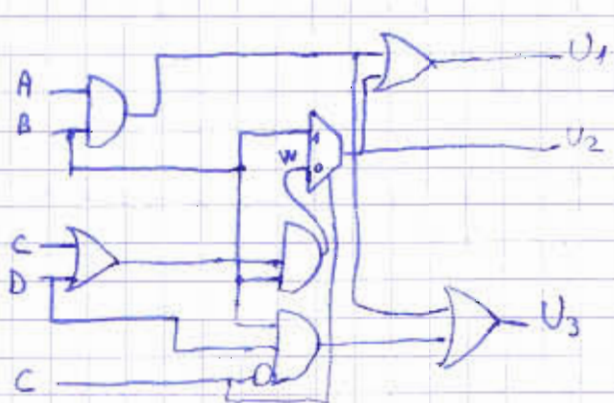
TEOREMA

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

ES. D3 agosto 2007



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

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

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

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

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

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

$$U_1 = AB + BC + BD$$

$$U_2 = BD + BC$$

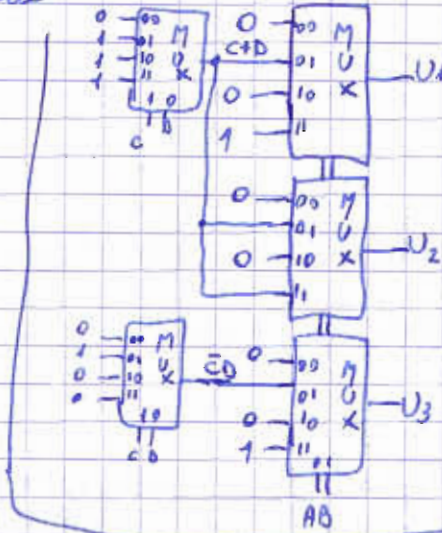
$$U_3 = AB + B\bar{C}D$$

NOP = 10 senza condizionamento

NOP = 7 con condizionamento

NOP = 6 nell'espressione non ridotta

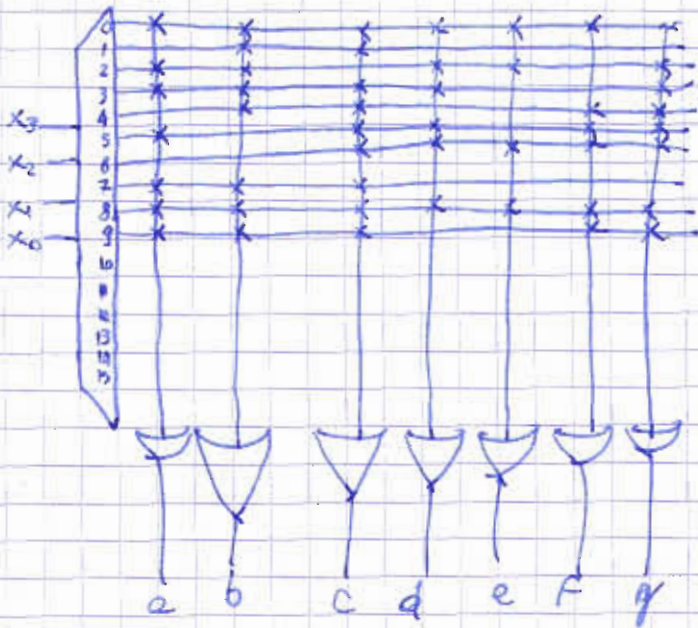
DISPLAY A 7 SEGMENTI



$X_3 X_2 X_1 X_0$	V	a	b	c	d	e	f	g
0000	0	1	1	1	1	1	1	0
0001	1	0	1	1	0	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.

Interi con decoder



RETI A 4 LIVELLI

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

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

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

$N_{OP} = 3$

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

$N_{MUX} = 6$

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

$N_{OP} = 9$

Applicando la distributiva

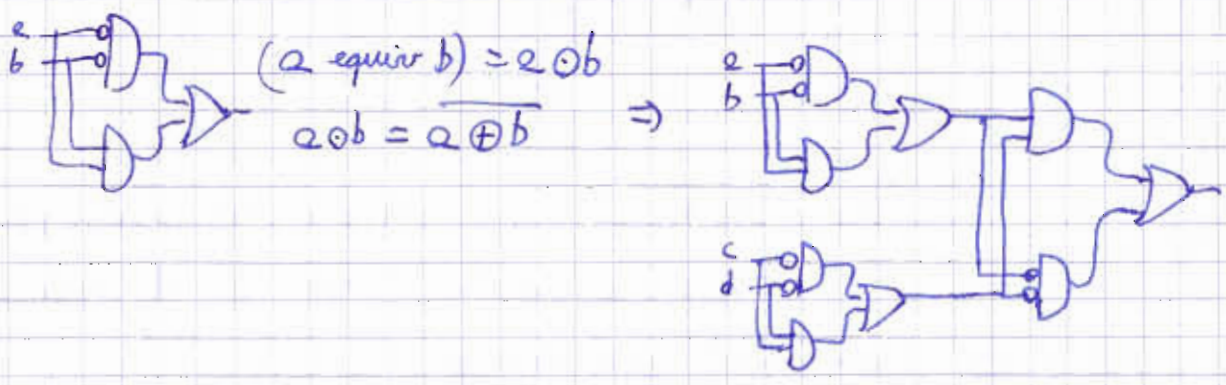
$N_{M} = 40!!$

↓

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

$$(\bar{a}\bar{b} + ab)(\bar{c}\bar{d} + cd) + (\bar{a}b + a\bar{b})(\bar{c}d + c\bar{d}) \quad N_{OP} = 15 \quad N_{M} = 30 \leftarrow \begin{matrix} n^{\circ} \text{ morsetti} \\ \text{d'ingresso} \end{matrix}$$

30/10/08



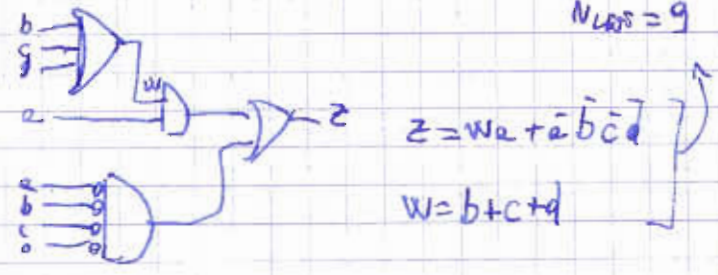
cd \ ab	00	01	11	10
00	1			
01				
11	1	1	1	1
10		1	1	1

$$f = ab + ac + ad + \bar{a}\bar{b}\bar{c}\bar{d} \quad N_{OP} = 5$$

$N_M = 14$ 2 livelli
 $N_{LSP} = 10$

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

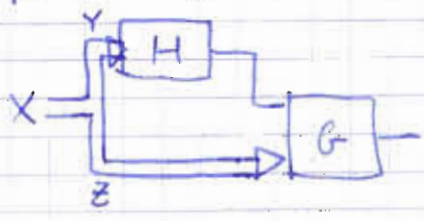
SPS $N_{OP} = 4$ 3 livelli
 $N_M = 14$
 $N_{LSP} = 9$



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 $f(x_1, \dots, x_n)$



$Y = \text{variabili vincolate } (y_1, y_m)$

$Z = \text{variabili libere } (z_1, z_r)$

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

MAPPE DI PARTIZIONE → mappa 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 la molteplicità di colonne (n° di colonne con diversa configurazione di 0 e 1) è non superiore a 2.

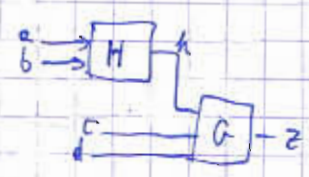
ES.

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

$Y = \{ab\}$
 $Z = \{cd\}$

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

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



↑
1° colonna: $h=1$ se variabili uguali
2° colonna: $h=0$ se variabili diverse

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

ES. 22/12/05

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

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

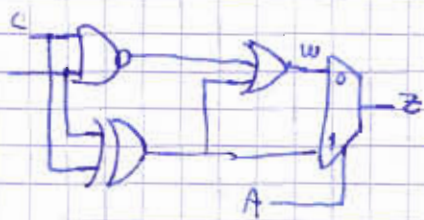
$h = B + \bar{D}$
 $h = 1$
 h
 \bar{h}

$cd \backslash h$	0	1
00	(0)	(1)
01	(1/0)	(1/1)
11	(0)	(1)
10	(1)	(0)

$f_5 = \bar{A}h + Ch + A\bar{C}\bar{h}$
 $f_5 = (A+h)(\bar{c}+h)(\bar{a}+c+h) = (A+h)\bar{c}h + (A+h)(\bar{a}+c)h = A\bar{c}h + Ah\bar{c} + A\bar{a}h + Ahc + \bar{c}h + ch + \bar{a}h + ah = A\bar{c}h + Ah\bar{c} + A\bar{a}h + Ahc + \bar{c}h + ch + \bar{a}h + ah$

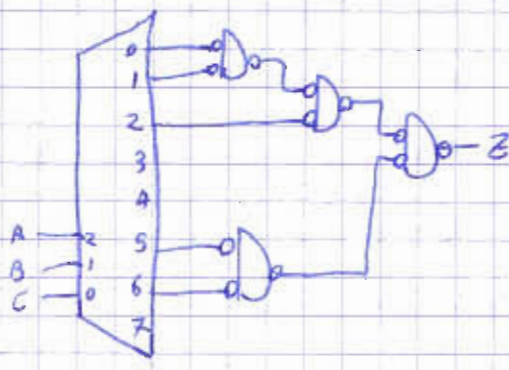
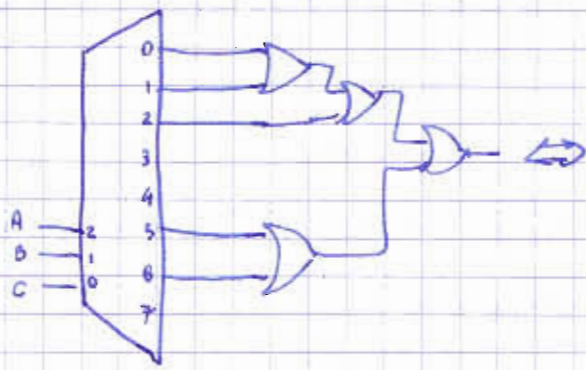
AB e CD non possono essere variabili vincolate perché 3 colonne diverse

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



$$Z = \bar{A}W + A(\bar{B}C + B\bar{C}) = \bar{A}((\bar{C}\bar{B}) + (\bar{B}C + B\bar{C})) + A\bar{B}C + AB\bar{C} = \bar{A}(\bar{B} + \bar{C} + B\bar{C} + \bar{B}C) + A\bar{B}C + AB\bar{C} = \bar{A}\bar{B} + \bar{A}\bar{C} + A\bar{B}C + AB\bar{C}$$

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



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

0. $\begin{matrix} & \bar{b}\bar{c} & \bar{b}c & b\bar{c} & bc \\ \bar{a} & 0 & 0 & 1 & 1 \\ a & 1 & 1 & 0 & 0 \end{matrix}$

IMPLICANTI
PRIMI

$I_1 = a\bar{b}$

$I_2 = b\bar{c}$

$I_3 = a\bar{c}$

1. $\begin{matrix} \text{Espansione} \\ \text{con} \\ \text{Shannon} \end{matrix}$

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

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

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

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

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

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

$+ c\bar{b}\bar{a}\cdot 0 + cba\cdot 0.$

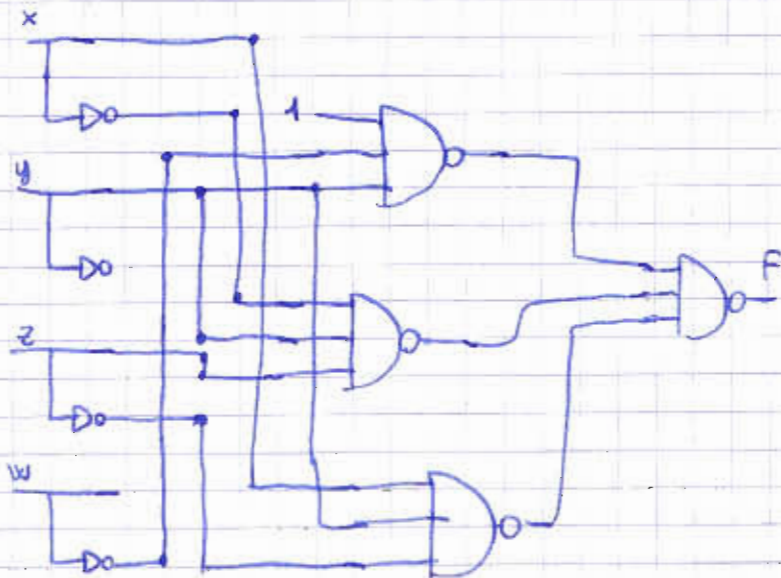
3. NAND $f = (a \uparrow \bar{b}) \uparrow (b \uparrow \bar{c})$



04/11/08

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

Devo usare solo queste porte!



$\begin{matrix} & \bar{z}\bar{w} & \bar{z}w & z\bar{w} & zw \\ \bar{x}y & 00 & 01 & 11 & 10 \\ 00 & 0 & 0 & 0 & 0 \\ 01 & 1 & 0 & 1 & 1 \\ 11 & 1 & 1 & 0 & 1 \\ 10 & 0 & 0 & 0 & 0 \end{matrix}$

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

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



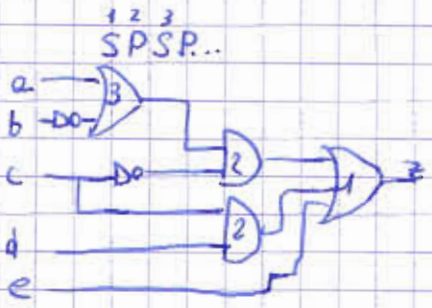
NOT con soli NAND



AND con soli NAND

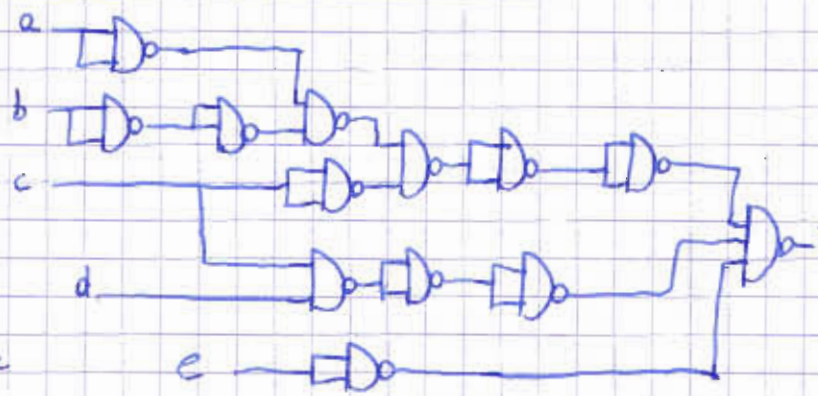


OR con soli NAND



1 2 3
SPSP...

↔

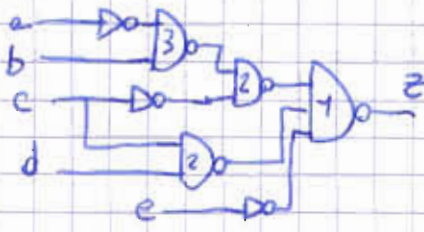


Che si semplifica avendo anche

1 NOT

Nei livelli pari i complementi rimangono

z invariati



NOT con soli NOR



OR con soli NOR



AND con soli NOR

SPSP... si convertono meglio in soli NAND

PPSP... si convertono meglio in soli NOR

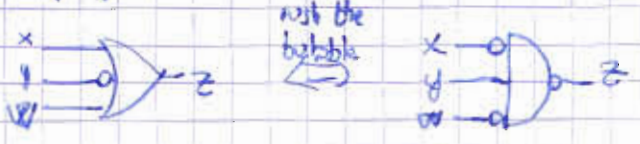
$R = x \cdot (y + (z \cdot w)) \cdot k$ aumento il livello ogni "(" e decremento ogni ")"

livello 1 2 3 4

↳ letterali dei livelli dispari vanno complementati!

$$R = \bar{x} \downarrow (y \downarrow (\bar{z} \downarrow w)) \downarrow k$$

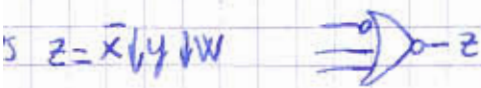
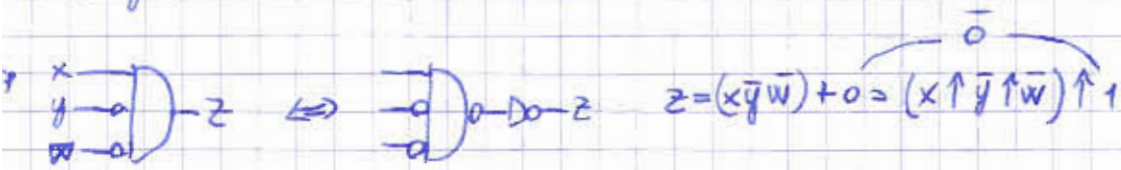
$Z = x + \bar{y} + w$ espressione SP a 4 livelli (solo la somma) $\Rightarrow Z = \bar{x} \uparrow y \uparrow w$



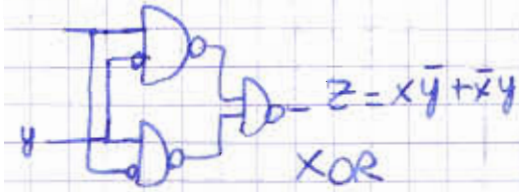
espressione PS con un solo livello di somma

$$Z = (x + \bar{y} + w) \cdot 1 = (x \downarrow \bar{y} \downarrow w) \downarrow 0$$

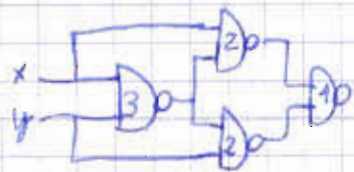
$$z = x\bar{y}\bar{w}$$



CONVERSIONE DA NAND A SP



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



$$z = (x\uparrow(x\uparrow y))\uparrow(y\uparrow(x\uparrow y)) = \text{trasforma quelle a livello dispari in normale.}$$

$$= (x \cdot (\bar{x} + \bar{y})) + (y \cdot (\bar{x} + \bar{y})) \quad \text{XOR}$$

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

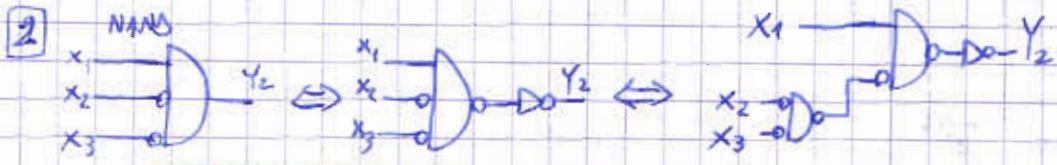
	DISPARI	PARI
NAND	OR	AND
NOR	AND	OR

Complemento solo i letterali isolati; non le parentesi.

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

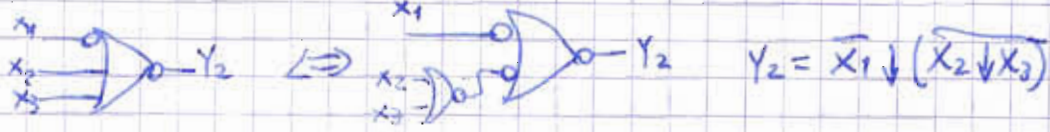
$$Y_1 = X_1 + X_2 + X_3$$

$$Y_2 = X_1 X_2 X_3$$



$$Y_2 = \overline{X_1 \uparrow (X_2 \uparrow X_3)}$$

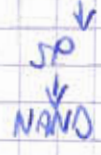
NOR PUSH THE BUBBLE



$$Y_2 = \overline{X_1} \downarrow (\overline{X_2} \downarrow X_3)$$

1) NAND: Tre modi

- 1) valutazione tramite mappa di Karnaugh NOR → PS(P...) → PS → MAPPA
- 2) NOR → PS(P..) Z = espressione PS + 0 → NAND n+1 livelli!
- 3) dualità (E⁰) = E: sostituire ogni operatore con il suo duale e ogni costante con la duale per ottenere E⁰.



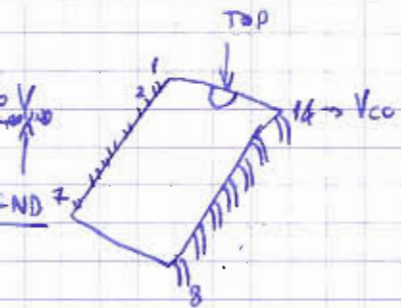
TEOREMA DI DUALITÀ

$$X^0 = X(\overline{X_1}, \overline{X_2}, \dots, \overline{X_n})$$

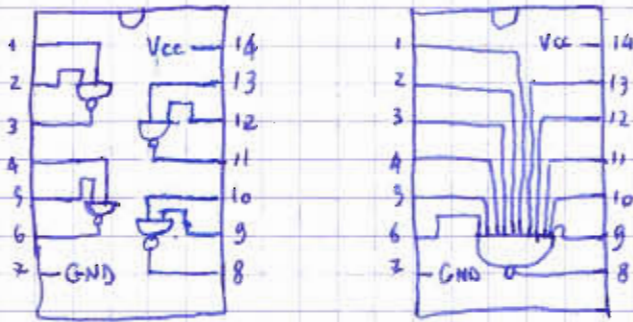
Per il teorema di dualità diventa



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



Packaged Logic

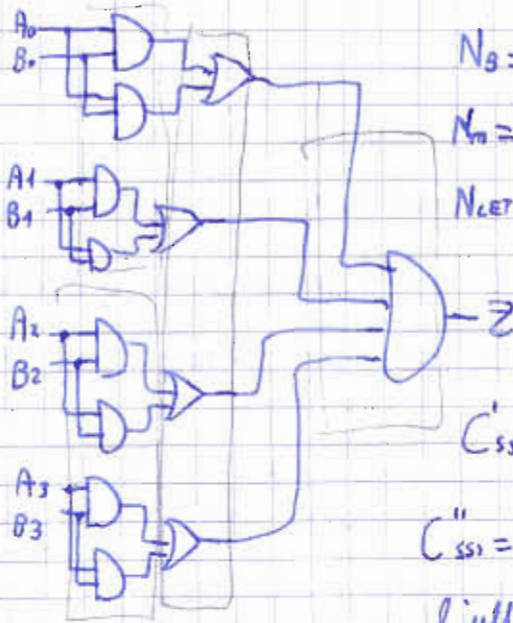
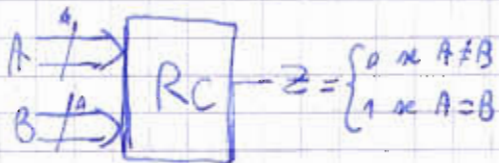


L'indice di costo
è il numero di Package.

stessa famiglia logica

stessi valori di tensione,
stessi standard,
stesso costo

Realizzare un comparatore di due operandi a 4 bit.



$N_S = 13$ (porte)

$N_M = 28$ (morsetti)

$N_{CET} = 12$

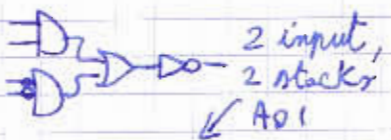
$C'_{SSI} = 4$ (packages)

$C''_{SSI} = 3 + \frac{1}{2}$ I.C. SSI.

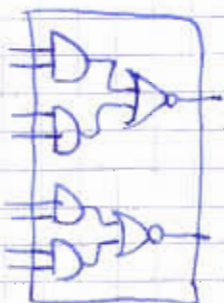
l'ultimo package lo
uso a metà (5 piedini)

Medium Scale Integration

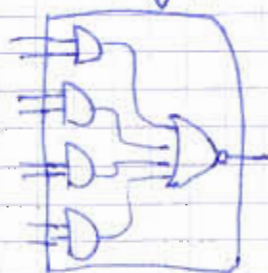
Componenti AOI (And Or Invert) → 2 o 3 livelli



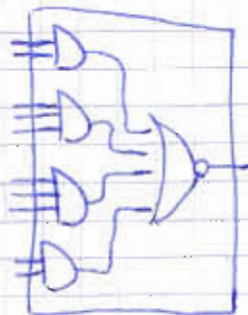
$$Z = X_1 \oplus Z_2 = X_1 \bar{X}_2 + \bar{X}_1 X_2 = \overline{X_1 \bar{X}_2 + X_1 X_2}$$



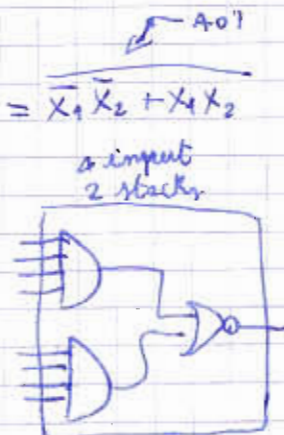
54



54



LS54



LS55

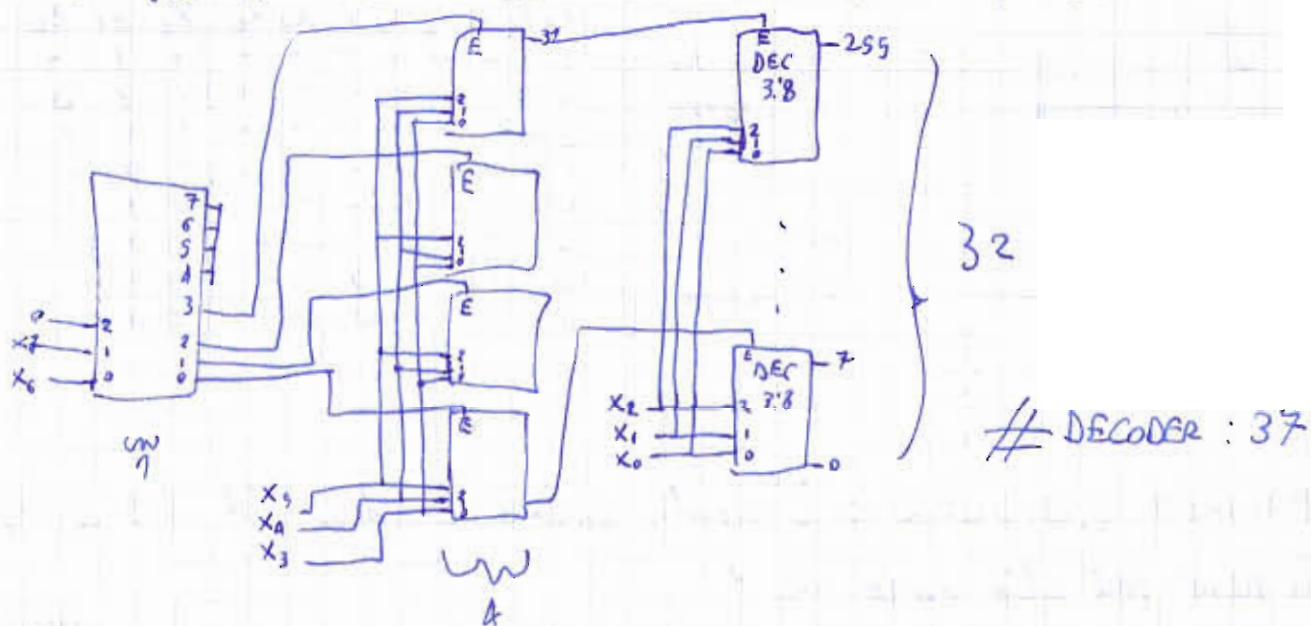
$$C''_{AOI} = 2 + \frac{1}{2} I_C$$

$$C'_{AOI} = 3 \text{ IC SSI}$$

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



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

costo materia 256 uscite = 512

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

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

MULTIPLEXER

$n_m = 2^n (n+2)$ costo: $O(n \cdot 2^n)$

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

Esempio

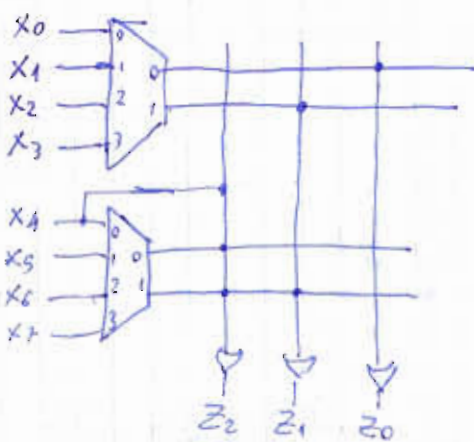
Encoder 8:3 $2^n \rightarrow n$
 $8 \rightarrow 3$

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

$$Z_2 = X_1 + X_3 + X_6 + X_7$$

$$Z_1 = X_2 + X_3 + X_6 + X_7$$

$$Z_0 = X_1 + X_3 + X_5 + X_7$$



$$Z_0' = X_1 + X_3$$

$$Z_1' = X_2 + X_3$$

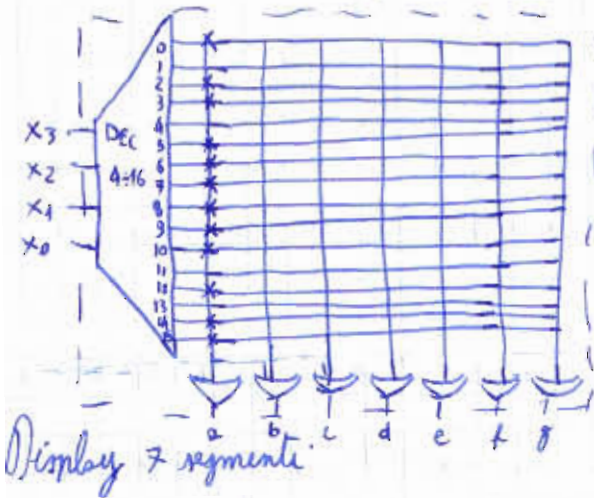
$$Z_0'' = X_5 + X_7$$

$$Z_1'' = X_6 + X_7$$

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 → l'uscita indica l'indice del bit d'ingresso di peso più alto posto ad 1.

11/11/08



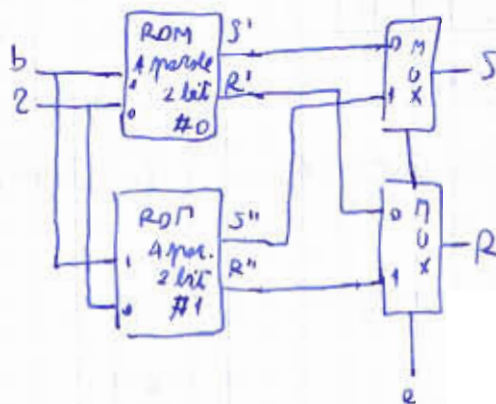
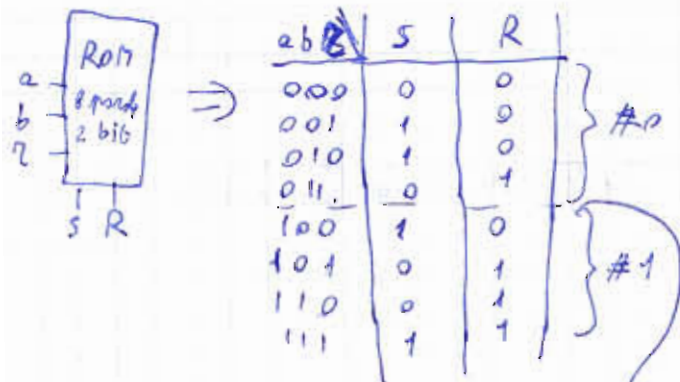
n ingressi, 2^n uscite

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

READ ONLY MEMORY - ROM

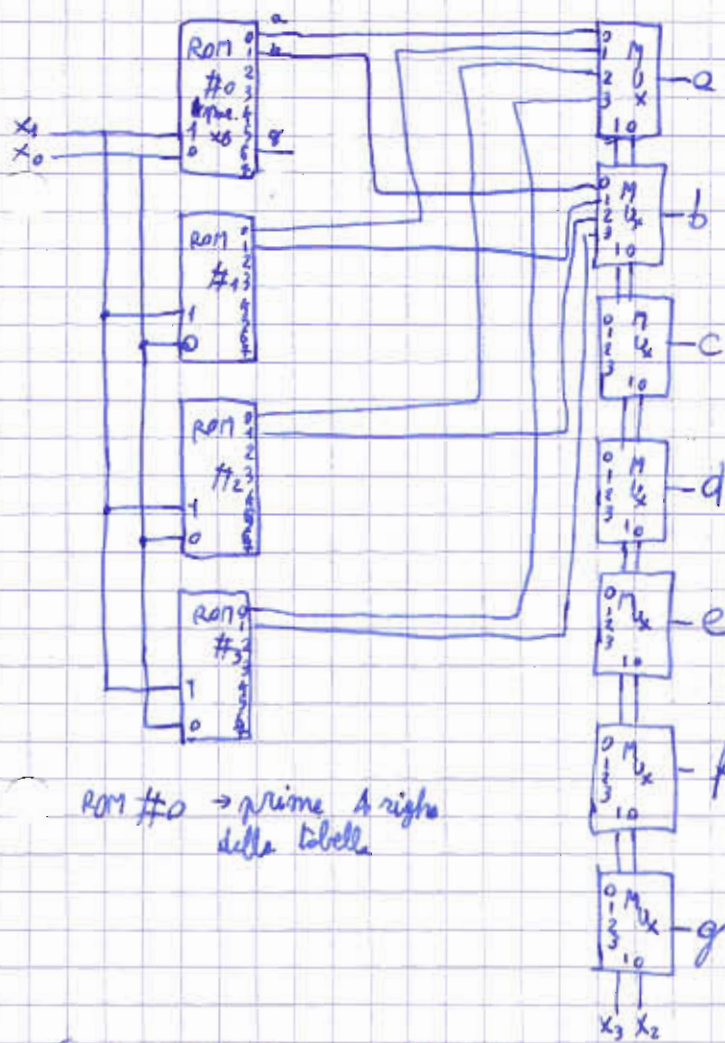
2 ROM 4 parole e 2 uscite (2 ingressi)

$n=2$ $m=2$



Primo #1 e $n=2$

Composizioni in serie.



7 mux perché
7 uscite

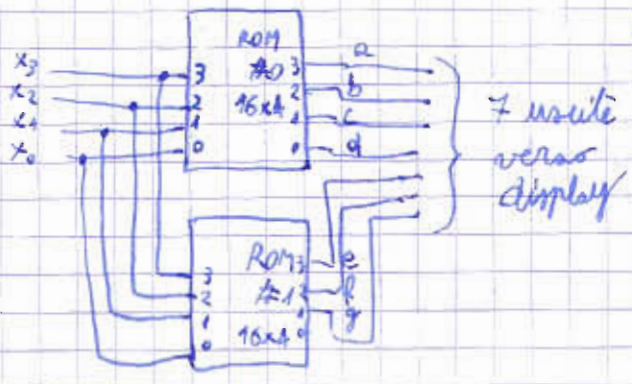
ROM #0 → prime 4 righe della tabella

CONVERSIONE IN PARALLELO

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

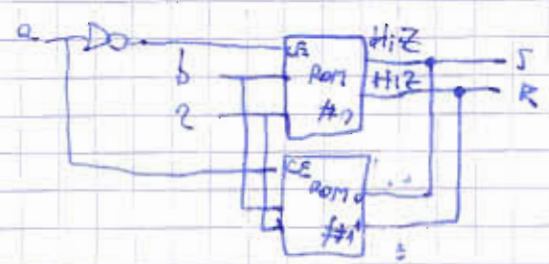
$x_3 x_2 x_1 x_0$	a	b	c	d	e	f	g
0000	1	1	1	1	1	0	0
0001	0	1	1	0	0	0	0
0010							
0011							
0100							
...							
1000	1	1	1	1	1	1	1
...							
1111	1	0	0	1	1	1	1

4 funzioni di 4 variabili



7 uscite verso display

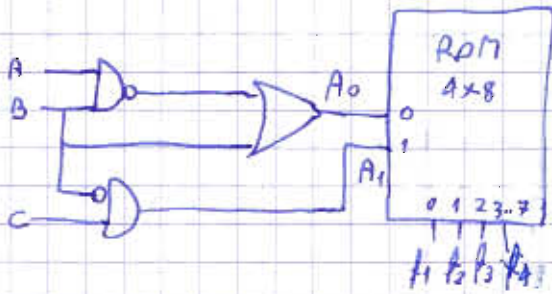
CHIP ENABLE (CE) → abilita la rom
Il FA diventa



HiZ → alta impedenza.

ES.

Contenuto ROM



A ₁ A ₀	f ₁	f ₂	f ₃	f ₄
00	0	1	0	1
01	1	0	1	0
10	0	1	0	1
11	1	0	1	0

Sintetizzare un circuito equivalente



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

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

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

$$f_2 = \bar{A}_1 \bar{A}_0 + A_1 \bar{A}_0 = \bar{A}_0$$

$$f_3 = f_1 = A_0$$

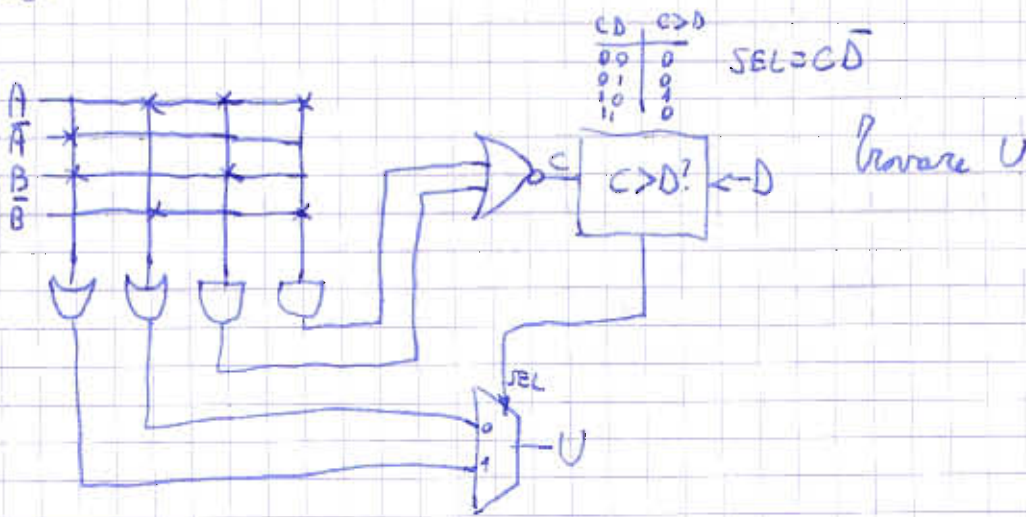
$$f_4 = f_2 = \bar{A}_0$$

$$\Rightarrow \begin{cases} f_1 = f_3 = 1 \\ f_2 = f_4 = 0 \end{cases}$$

0 — f₂, f₄

1 — f₁, f₃

ES.

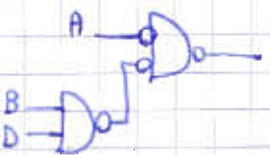


$$U = \overline{SEL} (A+B) + SEL (\bar{A} + \bar{B})$$

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

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

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



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

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

Nelle matrici programmabili $K \ll 2^n$

Una PLD non può da sola coprire tutte le funzioni in quanto ha solo K prodotti e non copre tutti i mintermi possibili nella configurazione peggiore (funzione reschiera).

PLA → programmabili sia il piano AND che quello OR

PAL → 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

Realizzare le funzioni:

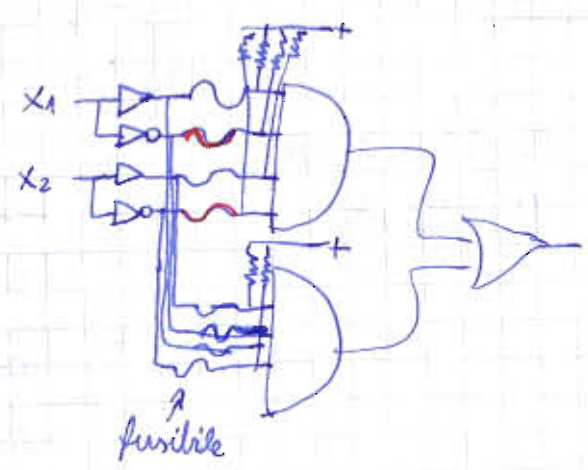
$$f_1 = X_1 X_2 \bar{X}_3 + \bar{X}_1 \bar{X}_2 X_3$$

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

su una PAL

- 3 ingressi
- 4 termini prodotti
- 2 termini somma a 2 ingressi
- 2 uscite

PAL BIPOLARI

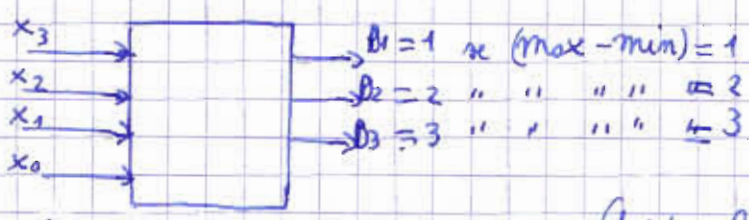
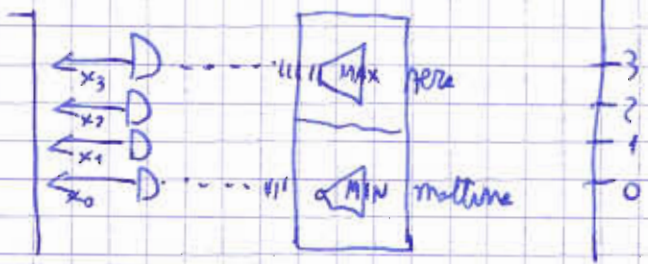


Bruciando i fusibili in rosso, \bar{X}_1 e \bar{X}_2 non entrano (circuito aperto) e gli ingressi degli AND vengono portati a 1 dal + (pull-up).

⊗ → tutti i fusibili sono lasciati

ROM → decoder al posto del piano AND.

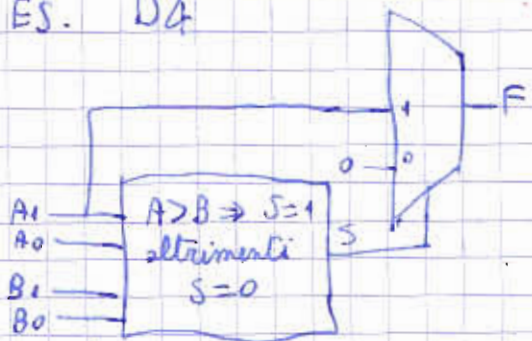
PAL 14H8 14 → n° ingressi 8 → n° uscite
 H → uscite attive su livello alto (SOP)
 www.daddy88.com Davide Valeriani



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	1	0	0	-	-	-
0	1	0	1	min 0 max 2	0	1
0	1	1	0	min 1 max 2	1	0
0	1	1	1	-	-	-
1	0	0	0	-	-	-
1	0	0	1	min 0 max 3	0	0
1	0	1	0	min 1 max 3	0	1
1	0	1	1	-	-	-
1	1	0	0	min 2 max 3	1	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

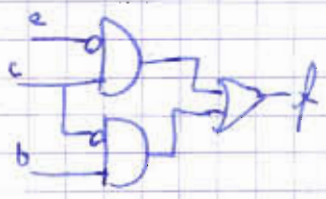
$A_1 B_1$ \ $A_0 B_0$	00	01	11	10
00	0	0	0	0
01	1	0	0	0
11	1	1	0	1
10	1	1	0	0

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

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

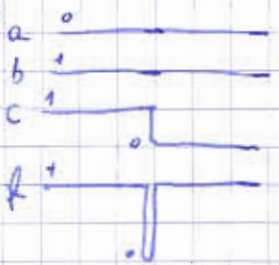
	00	01	11	10
0	0	1	1	1
1	0	0	0	1

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



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

ALFA STATICA



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

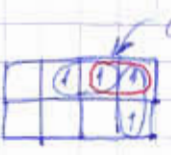
Le espressioni normali PS sono soggette ad due statiche di "0".

ALFA DINAMICA

→ quando un segnale deve cambiare, invece di farlo così lo fa così

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

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



ci è un'alea statica passando da 11 a 10

allora aggiungo gli implicanti che servono per rimuovere le alea statiche

Un'espressione PS è priva di due statiche se \nexists due maxterm adiacenti non inclusi nel medesimo implicato.



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

$$z = (b+d) \cdot (a+b) \cdot (\bar{a}+b+\bar{c}) \text{ ho 2 alea statiche}$$

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



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 C _{in}	00	01	11	10
S	0	1	0	1
C _{out}	0	0	1	0

$$S = \bar{a}_i \bar{b}_i C_{in} + \bar{a}_i b_i \bar{C}_{in} + a_i \bar{b}_i \bar{C}_{in} + a_i b_i C_{in}$$

$$N_B = 5; N_m = 3 \cdot 4 + 4 = 16$$

$$N_{LET} = 12$$

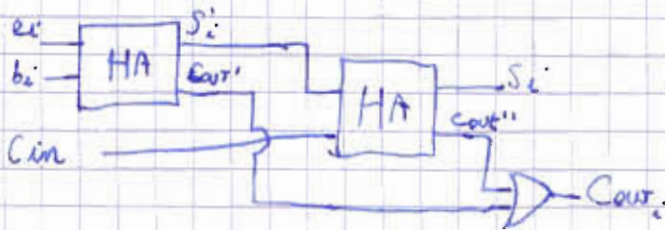
a _i b _i C _{in}	00	01	11	10
C _{out}	0	0	1	0
S	0	1	0	1

$$C_{out} = b_i C_{in} + a_i C_{in} + a_i b_i$$

$$N_B = 4; N_m = 3 \cdot 2 + 3 = 9$$

$$N_{LET} = 6$$

FA da HA



$$N_B = 5; N_m = 40;$$

$$N_{LET} = 7; N_{LIN} = 3!$$

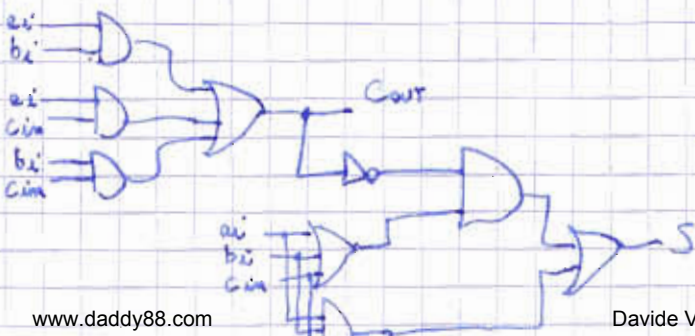
↑
← 5TA

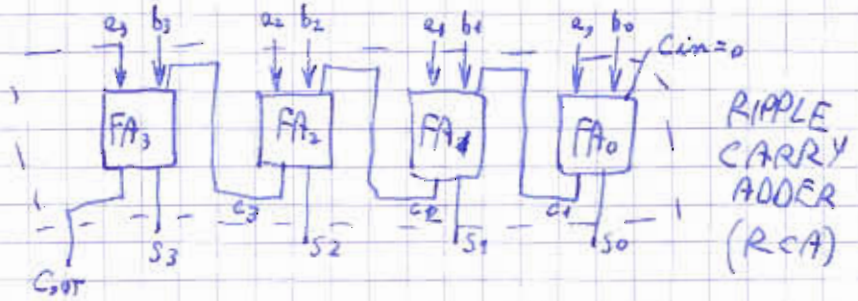
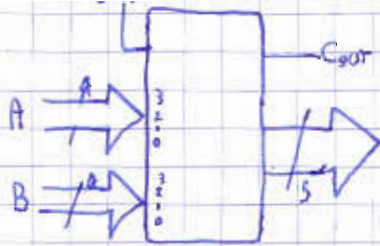
Un circuito ancora meno costoso usa la sintesi mediante mappe per C_{out} e usa un HA per la somma.

a _i b _i C _{in}	00	01	11	10
C _{out}	1	1	0	1
S	1	0	0	0

simile alla somma; per aggiungere un 1 a C_{out} aggiungo il minterm, mentre per aggiungere uno 0 aggiungo il maxterm!

$$S = (\bar{C}_{out} + a_i b_i C_{in}) \cdot (a_i + b_i + C_{in}) = \bar{C}_{out} \cdot (a_i + b_i + C_{in}) + a_i b_i C_{in}$$



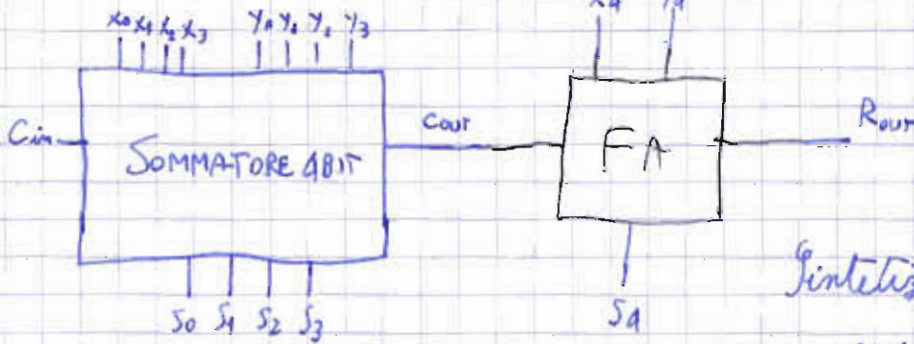


RIPPLE CARRY ADDER (RCA)

Sommatore più economico e più lento esistente. FA₀ posso sostituirlo con un HA se Cin = 0 sempre, ma ~~è~~ in commercio.

Ritardo di 2N volte una porta logica elementare (N = n° bit).

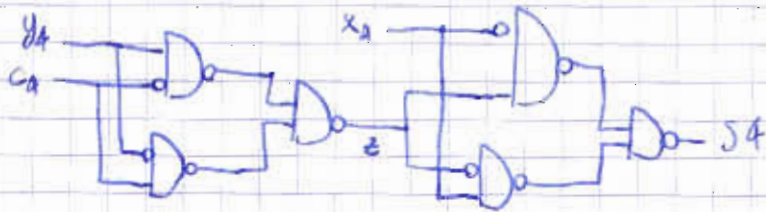
ES. ESAME



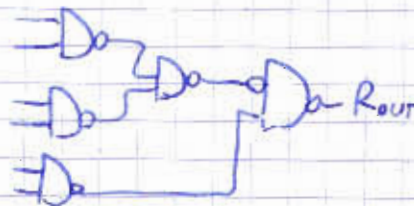
Provare circuito che somma 5 bit con soli NAND.

Sintetizzarlo quindi a soli NAND, con FAN-IN 2

$$S_A = \bar{x}_A \bar{y}_A C_A + \bar{x}_A y_A \bar{C}_A + x_A \bar{y}_A \bar{C}_A + x_A y_A C_A = \bar{x}_A (\bar{y}_A C_A + y_A \bar{C}_A) + x_A (\bar{y}_A \bar{C}_A + y_A C_A)$$



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



ALFA

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

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

1 P5 T SP

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

Non mi devo preoccupare di alfa se transito da un implicante a un DC (-)

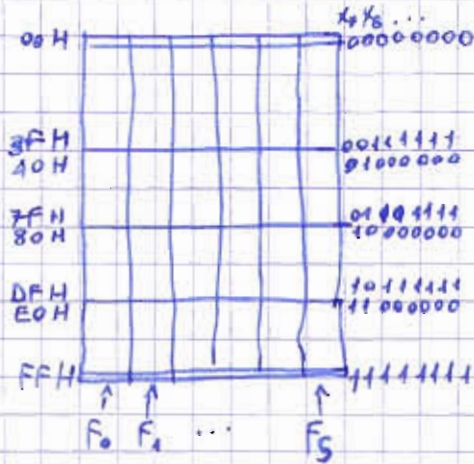
ROM

6 funzioni di 8 variabili indipendenti.

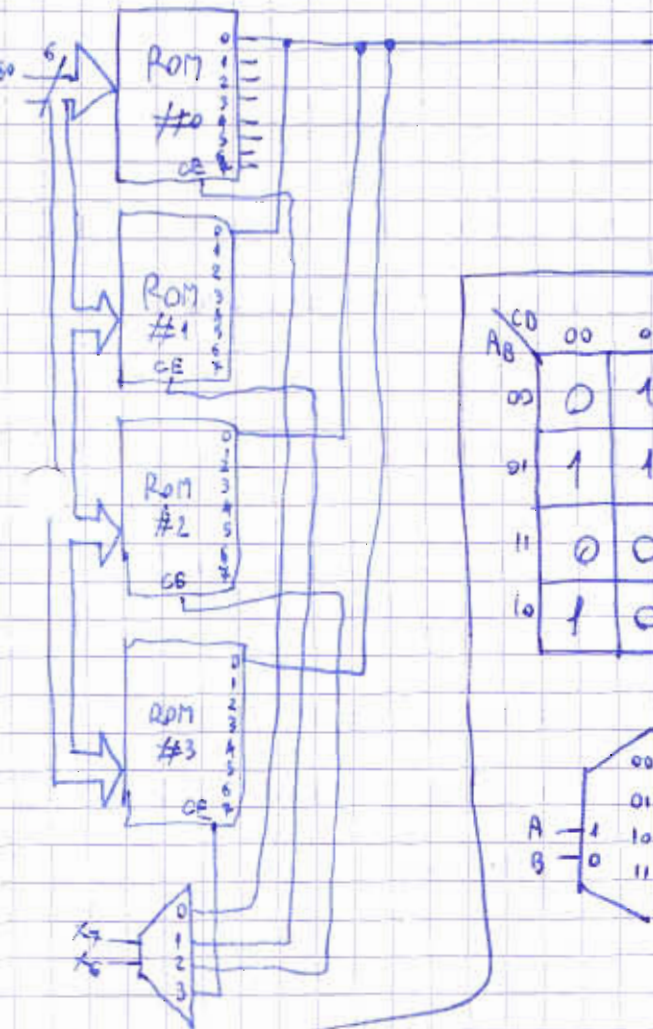
Disponibili ROM da 64 parole da 8 bit l'una dotate di CE (Chip Enable).

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

Per un decoder con 64 uscite e quindi mi servono 6 ingressi di indirizzamento.

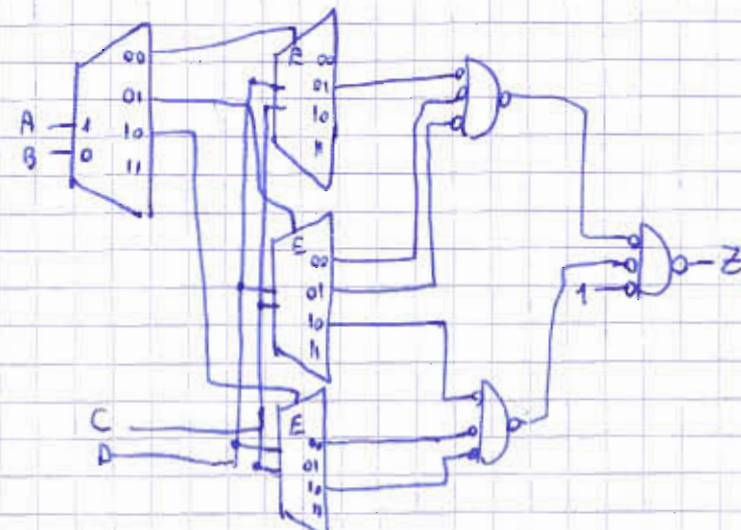


Prendendo una parola 0100001, ogni ROM porta un'uscita al suo valore corrispondente, quindi mi intereremo anche 00100001, 10100001.

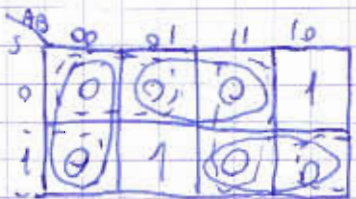


CD	00	01	11	10
Ab	00	1	-	0
01	1	1	0	0
11	0	0	0	-
10	1	0	-	1

↳ tutti 0



20/11/08



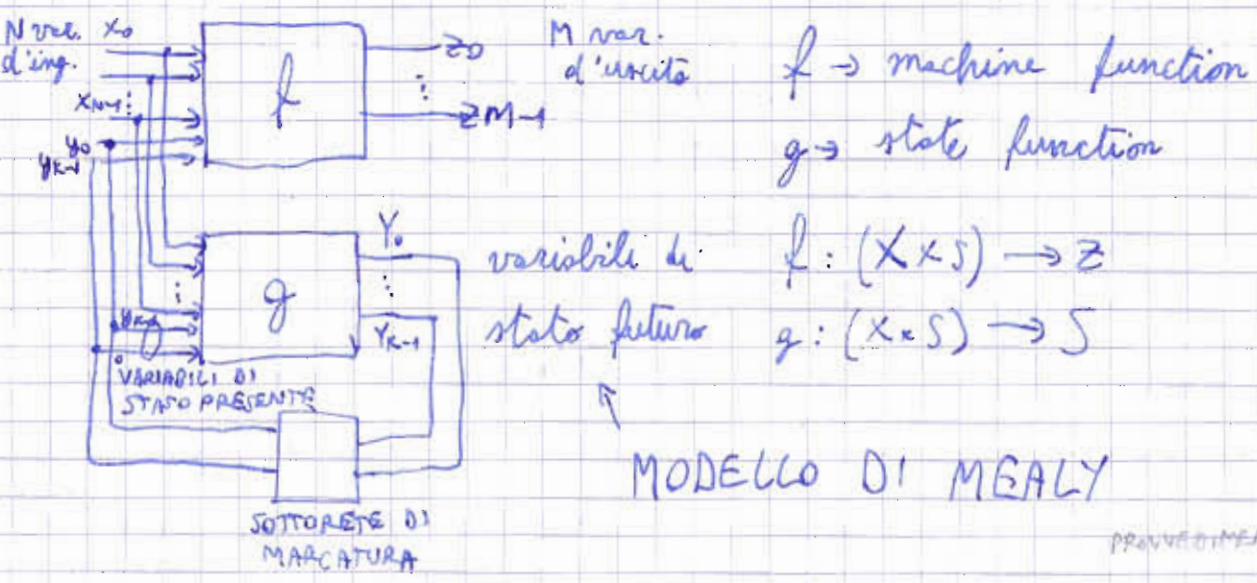
$$Z = (A+B) \cdot (S+B) \cdot (\bar{S}+\bar{A})$$

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

Prendendo C e ... risolviamo il problema delle alce

RETI SEQUENZIALI

L'uscita dipende anche dalle uscite precedenti.

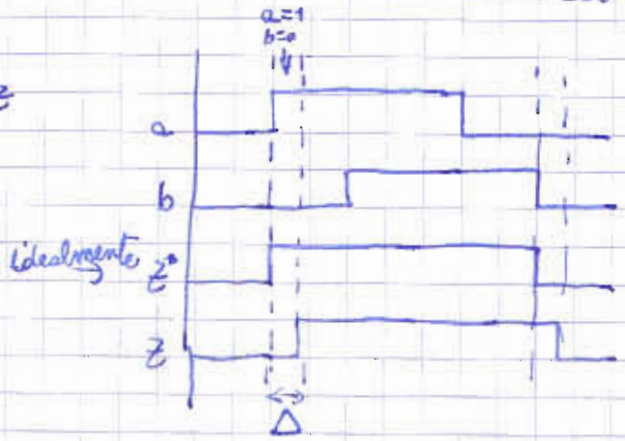


PROVVEDIMENTO GREMBIOLINO

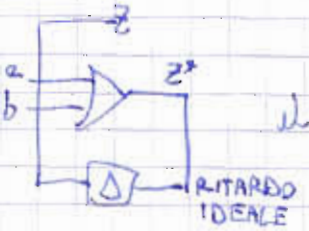
Se al posto della sottorete di marcatura ho fili libretti -> MODELLO DI MOORE

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.



se $a=1$ e $b=0$ in Δ
 ho $z=0$ quindi circuito sequenziale

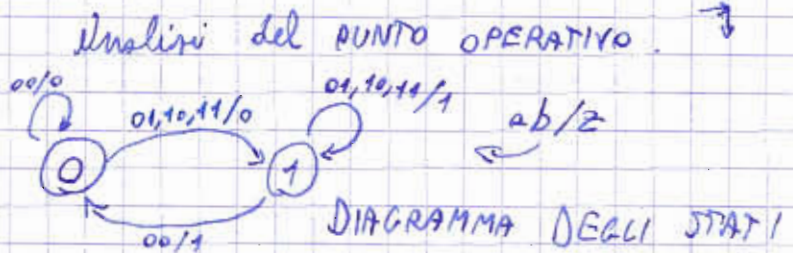


in un certo livello di precisione non esistono reti combinatorie
 in realtà Δ non è costante

$y(t) = Y(t - \Delta)$ $Y(t + \Delta) = Y(t)$ 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	TABELLA DELLE TRANSIZIONI
1	0,1	1,1	1,1	1,1	

(Y,Z)



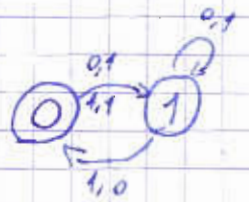
posso tagliare la retroazione

$$Z = a \uparrow y = \bar{a} + \bar{y}$$

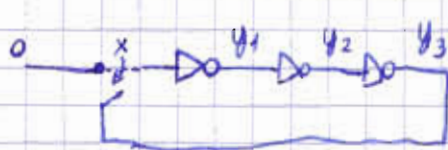
$$Y = \bar{a} + \bar{y}$$

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

	0	1	
0	1,1	1,1	(Y,Z)
1	1,1	0,0	



I ritardi rendono i circuiti sequenziali.



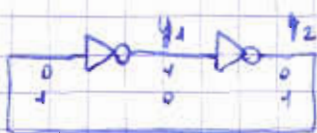
$\tau \rightarrow$ ritardo Not

	x	y ₁	y ₂	y ₃	
$t=0^-$	0	1	0	1	condizione iniziale
$t=0^+$	1	1	0	1	
$t=\tau$	1	0	0	1	
$t=2\tau$	1	0	1	1	
$t=3\tau$	0	0	1	0	
$t=4\tau$	0	1	1	0	
$t=5\tau$	0	1	0	0	variabili periodiche con periodo 6τ ma fase diversa.
$t=6\tau$	1	1	0	1	

Qualunque retroazione di un n° dispari n di Not identici produce un'oscillazione di periodo

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

OSCILLATORE AD ANELLO



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

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



per $a=0$, $z=b$
per $a=1$, $z=0$

INVERTITORE CONTROLLATO

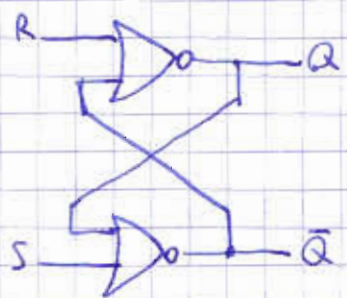


$RS=00 \rightarrow y_1$ memorizza, $y_2 = \bar{y}_1$

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

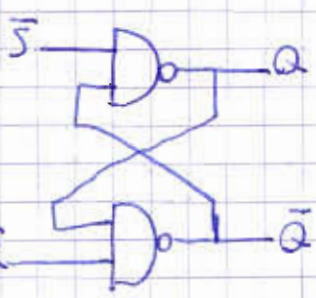
$R=0, S=1 \rightarrow y_1=1$ e $y_2=0$ commutazione o conferma

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



LATCH SR
 $\begin{matrix} e \\ t \\ e \end{matrix}$ $\begin{matrix} e \\ s \\ e \end{matrix}$

SR=11 vietata!



a	b	a ↑ b
0	0	1
0	1	1
1	0	1
1	1	0

LATCH → non dipende dall'istante di tempo.
 FLIPFLOP → dipende dal tempo.

I latch SR vengono usati come sottotipi di marcatura applicando 11 e poi 00 il latch inizia ad oscillare.

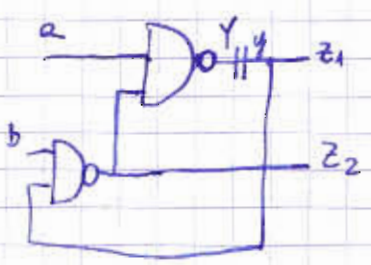
Q	Q*	S	R
0	0	0	-
0	1	1	0
1	0	0	1
1	1	-	0

TABELLA DELLE ECCITAZIONI

$Q^* = S + \bar{R}Q$ con $SR \neq 11$ EQUAZIONE CARATTERISTICA

25/11/08

COMPITINO: 19/12 ore 15.30



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

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

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

TABELLA TRANSIZIONI

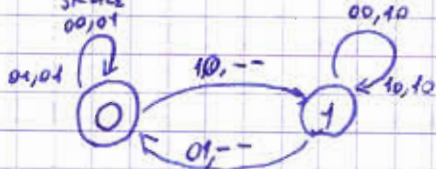
○ Y=y stato futuro = stato presente

gloriosa vittoria perché non è vero che $a=z_1$ $\bar{a}=z_2$

introduzione di "1" nella memoria (SET)

introduzione di "0" nella memoria (RESET)

SINTESE: 11 Don't Care



1) DIAGRAMMA DEGLI STATI

2) TABELLA TRANSIZIONI

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

$$Y_{sp} = S + y\bar{R}$$

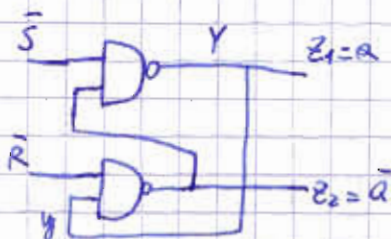
equazione caratteristica del latch SR

$$Y_{ps} = \bar{R}(S + y)$$

$$Z_{1sp} = y \quad Z_{2sp} = \bar{y} \quad \text{oppure } Z_2 = \bar{y} + R \quad \text{oppure } Z_2 = y\bar{S}$$

Y, Z1, Z2

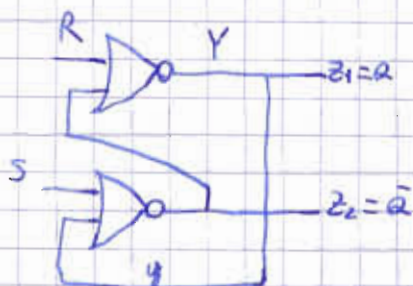
2 NAND: $Y = S + y\bar{R} = \bar{S} \uparrow (y + \bar{R})$



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

2 NOR:

$$Y_{ps} = \bar{R}(y + S) = R \downarrow (y \downarrow S)$$



$$Z_2 = \bar{y}\bar{S} = y \downarrow S$$

FLIP FLOP D



legge x sul fronte di salita

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

FLIP FLOP JK



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

opera in modo sincrono

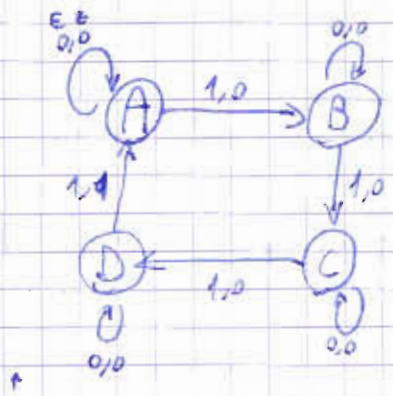
FLIP FLOP T



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

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

	$y_2 y_1$
A	00
B	01
C	10
D	11

$y_2 y_1$	E	0	1
A	00	00,0	01,0
B	01	01,0	10,0
D	11	11,0	00,1
C	10	10,0	11,0

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

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

6. MAPPA DELLE FUNZIONI DI ECCITAZIONE

$y_2 y_1$	E	0	1
00	0	0	0
01	0	1	1
11	1	1	0
10	1	1	1

$y_2 y_1$	E	0	1
00	0	0	1
01	1	1	0
11	1	1	0
10	0	1	1

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

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

TABELLA DELLE TRANSIZIONI

D_2
1ª colonna

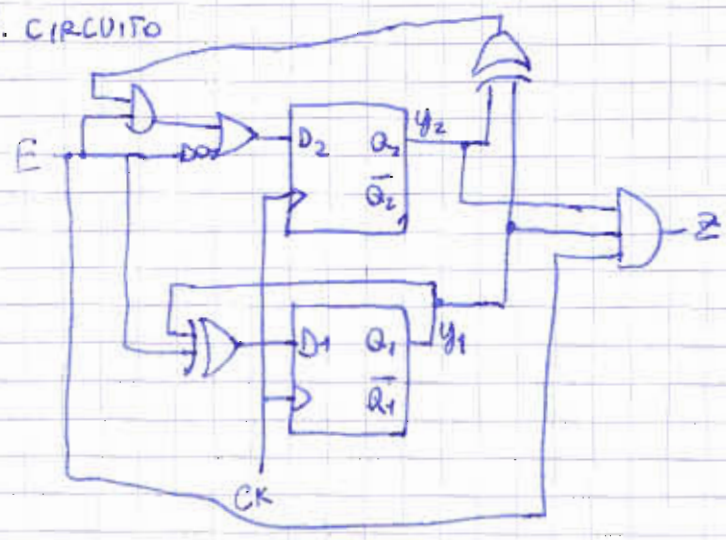
D_1
2ª colonna

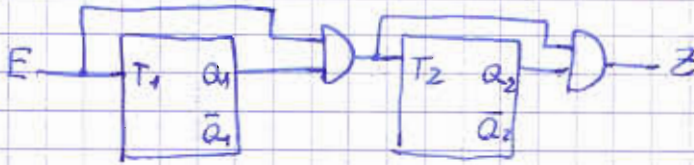
MAPPA USCITE

$y_2 y_1$	E	0	1
00	0	0	0
01	0	0	0
11	0	1	1
10	0	0	0

$$Z = E y_2 y_1$$

8. CIRCUITO





$T_1 = E$
 $T_2 = Q_1 E$
 $Z = Q_2 T_2 = Q_1 Q_2 E$

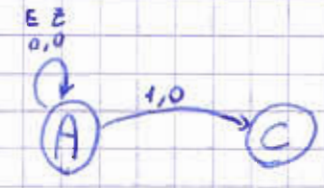
Uso la relazione del CF-T: $Q^{n+1} = (T \oplus Q)^n$
 $Q_1^{n+1} = (E \oplus Q_1)^n = E \bar{Q}_1 + \bar{E} Q_1$
 $Q_2^{n+1} = (\bar{Q}_1 E \cdot Q_2 + Q_1 E \cdot \bar{Q}_2) = \bar{Q}_1 Q_2 + \bar{E} Q_2 + Q_1 \bar{Q}_2 E$

TABELLA TRANSIZIONI

E	0	1
00	00,0	10,0
01	01,0	11,0
11	11,0	00,1
10	10,0	01,0

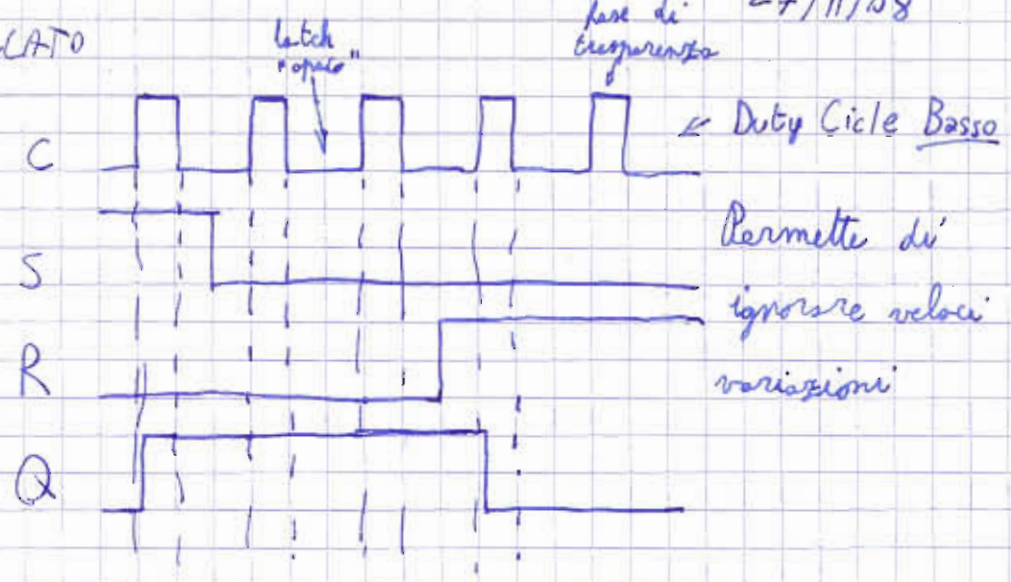
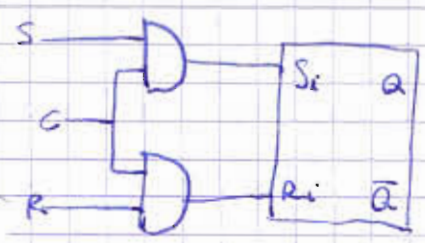
- A=00
- B=01
- C=10
- D=11

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

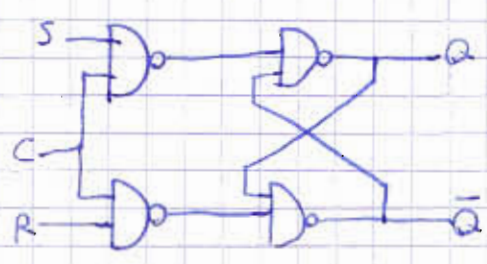


$(Q_1, Q_2)^{n+1}, Z^n$

LATCH SR CONTROLLATO

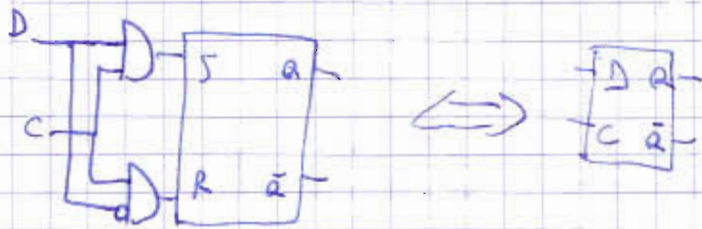


SOL: NAND

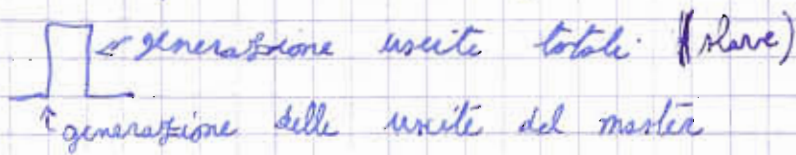


$Q^{n+1} = (S + Q \cdot \bar{R})^n$ versione sincrona → vale se C è periodico e in modo che nella fase di trasparenza non ci siano veloci variazioni
 $Q^{n+1} = \bar{C} Q^n + C(S + Q \bar{R})^n$ versione generale

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 (disattivato). Le uscite vengono elaborate sul fronte di discesa.



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



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

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

PROCEDIMENTO DI SINTESI DI RSS

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

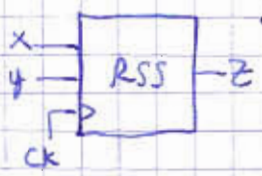
5. Tabella delle transizioni

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

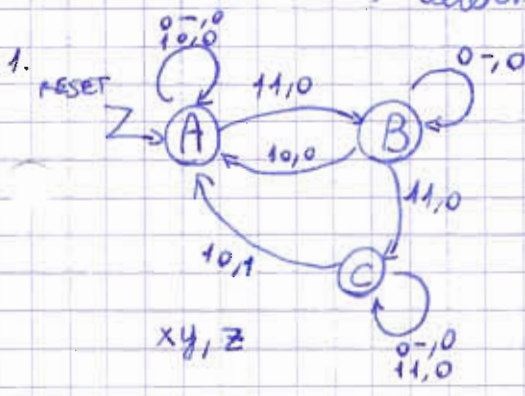
7. Mappe funzioni di eccitazione e di uscita (JK?, D?, T?)

8. Schema logico

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
 x=0 ignorato
 x=1 considerato



2.

xy	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, B=01, C=10

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

S₁S₂
S₁*S₂*, Z

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

7.

xy	00	01	11	10
00	0	0	0	0
01	0	0	1	0
11	-	-	-	-
10	1	1	1	0

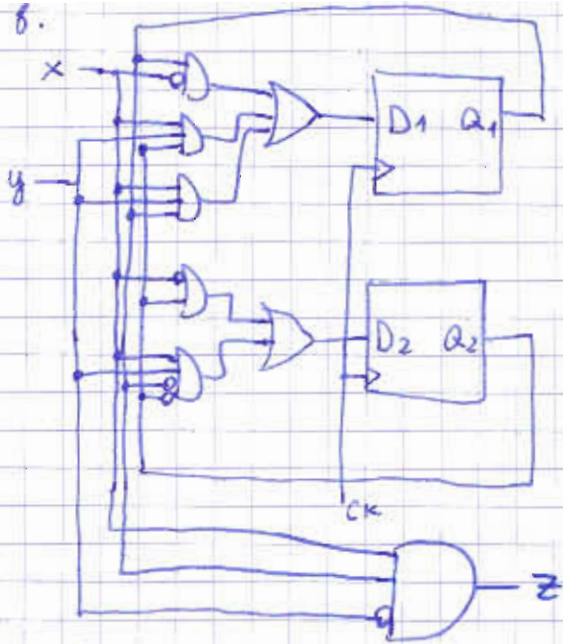
$S_1^* = D_1 = \bar{x}S_1 + xyS_2 + xyS_1$

xy	00	01	11	10
00	0	0	1	0
01	1	1	0	0
11	-	-	-	-
10	0	0	0	0

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

xy	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	-	-	-	-
10	0	0	0	1

$Z = xyS_1$



Con FF-JK

Q	J	K	Q ⁿ⁺¹
0	0	0	0 ⁿ⁺¹ → Q, J=0 ∀K
0	0	1	0 ⁿ⁺¹
0	1	0	1 ⁿ⁺¹ → Q=0, J=1 ∀K
0	1	1	1 ⁿ⁺¹
1	0	0	1 ⁿ⁺¹
1	0	1	0 ⁿ⁺¹ → Q=1, K=0 ∀J
1	1	0	1 ⁿ⁺¹ → Q=1, K=1 ∀J
1	1	1	0 ⁿ⁺¹

Q ⁿ → Q ⁿ⁺¹	J	K
0 → 0	0	0 -
0 → 1	1	1 -
1 → 0	0	- 1
1 → 1	1	- 0

ALGORITMO DI
ECCITAZIONE
DEL FF-JK

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

numeri in
grassetto
↓
cambiamenti

xy	00	01	11	10
00	0	0	0	0
01	0	0	1	0
11	-	-	-	-
10	-	-	-	-

xy	00	01	11	10
00	-	-	-	1
01	-	-	-	1
11	-	-	-	1
10	0	0	0	1

$$J_1 = xy S_2$$

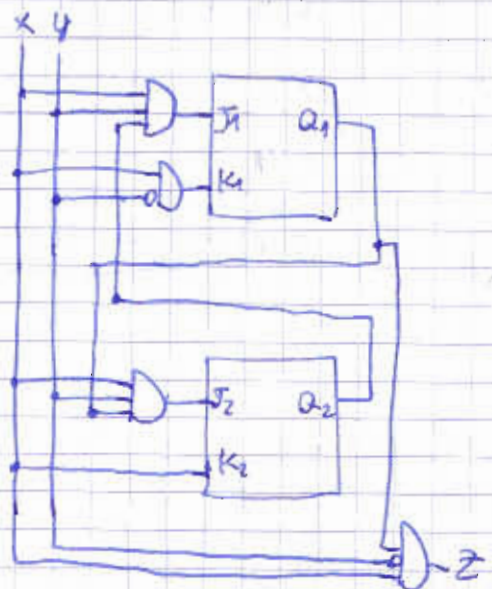
$$K_1 = x \bar{y}$$

xy	00	01	11	10
00	0	0	1	0
01	-	-	1	-
11	-	-	-	-
10	0	0	0	0

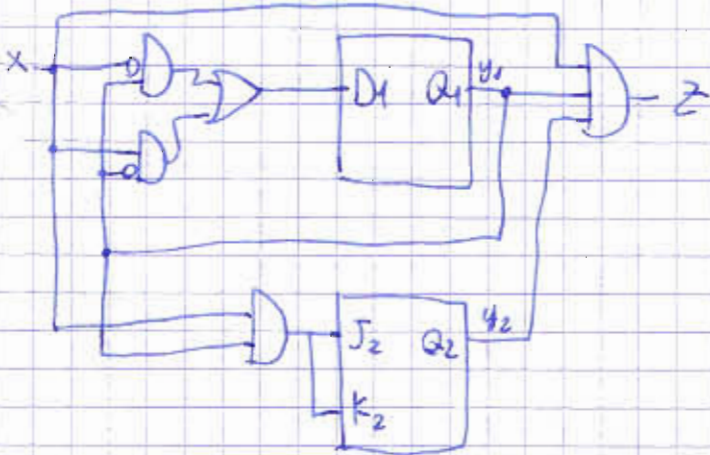
xy	00	01	11	10
00	-	-	1	-
01	0	0	1	1
11	-	-	-	-
10	-	-	-	-

$$J_2 = xy S_1$$

$$K_2 = x$$



ANALISI



$$D_1 = \bar{x}y_1 + x\bar{y}_1 \quad y_1^{n+1} = D_1^n = \bar{x}y_1 + x\bar{y}_1$$

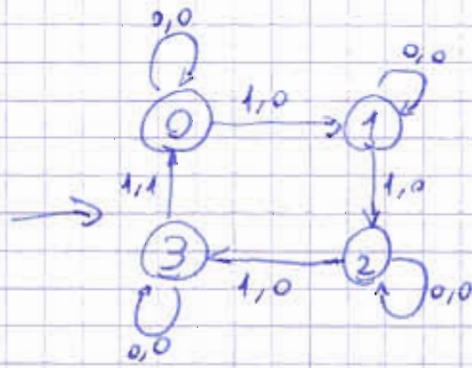
$$J_2 = xy_1 \quad y_2^{n+1} = (J_2\bar{y}_2 + \bar{K}_2y_2)^n = \bar{0}x\bar{0}y_2 + (x + \bar{y}_1)y_2 = xy_1\bar{y}_2 + (x + \bar{y}_1)y_2 = xy_1\bar{y}_2 + x\bar{y}_2 + \bar{y}_1y_2$$

$$K_2 = xy_1 \quad Z = xy_1y_2$$

$y_2 \backslash x$	0	1
0 A	00, 00	01, 0
1 B	01, 0	10, 0
3 D	11, 0	00, 1
2 C	10, 0	11, 0

y_2, y_1, Z

$S \backslash x$	0	1
0 A	A, 0	B, 0
1 B	B, 0	C, 0
3 D	D, 0	A, 1
2 C	C, 0	D, 0



2/12/08

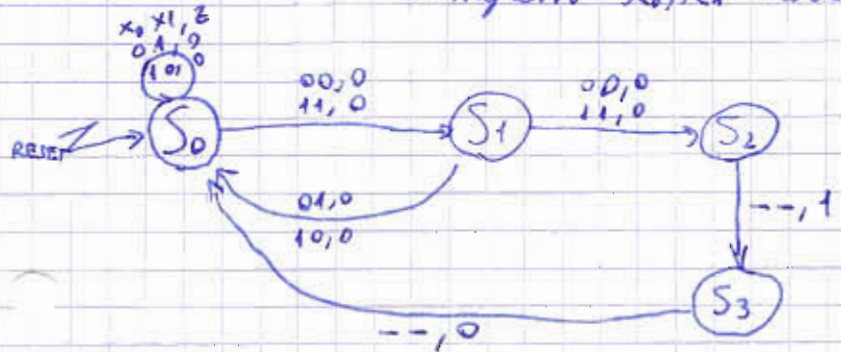
I flip flops hanno alcuni difetti: il NOT deve avere un ritardo $\rightarrow 0$, e' soggetto a problemi di aless... Nascono quindi i FF-EDGE TRIGGERED

Gli ingressi PRESET e CLEAR permettono di gestire il FF in modo asincrono, cioè indipendente dal clock, per mantenere il FF in uno stato particolare indipendentemente dagli ingressi. Entrano in forma negata

ES. ESAME



$Z(n) = 1$ se $x_0(n-1) = x_1(n-1)$ e $x_0(n-2) = x_1(n-2)$
 nell'intervallo in cui $Z=1$ e in quello successivo gli ingressi x_0, x_1 devono essere ignorati.



S_0	$S_{1,0}$	$S_{2,0}$	$S_{3,0}$	$S_{3,0}$
S_1	$S_{2,0}$	$S_{3,0}$	$S_{2,0}$	$S_{3,0}$
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

$$S_0 = 00$$

$$S_1 = 01$$

$$S_2 = 10$$

$$S_3 = 11$$

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	$11,1$	$11,1$	$11,1$	$11,1$

$$y_1^* y_0^*, Z$$

$$Z^n = y_1 \bar{y}_0$$

Utilizzo prima FF-D e poi FF-JK

CASO FF-D

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

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

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

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

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

CASO FF-JK

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

J K

$0 \rightarrow 0$	0	0	$-$
$0 \rightarrow 1$	1	1	$-$
$1 \rightarrow 0$	0	$-$	1
$1 \rightarrow 1$	1	$-$	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	$11,1$	$11,1$	$11,1$	$11,1$

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

$$J_1 = y_0 \bar{x}_1 \bar{x}_0 + y_0 x_0 x_1$$

1	1	1	1
0	0	0	0

$$K_1 = y_0$$

1	0	1	0
1	1	1	1

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

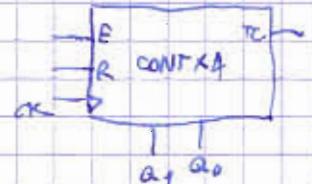
1	1	1	1
1	1	1	1

$$K_0 = 1$$

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

Il reset prende sempre.

	Enable	Reset	00	01	10	11
S^n	S^{n+1}	S^{n+1}	S^{n+1}	S^{n+1}	S^{n+1}	S^{n+1}
	S^n	0	S^{n+1}	0		



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

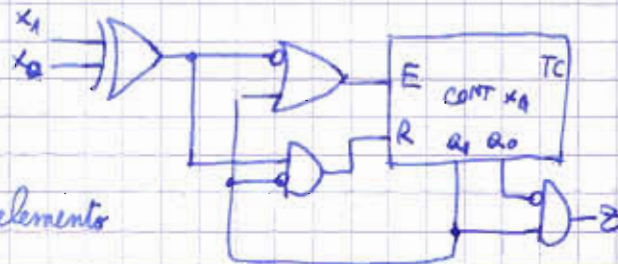
$E=1, R=0$
 $E=0, R=1$
 $E=1, R=1$

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

$$E = y_1 + X_0 X_1 + X_0 X_1$$

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

$$R = \bar{y}_1 \bar{X}_0 X_1 + \bar{y}_1 X_0 \bar{X}_1 = \bar{y}_1 (X_0 \oplus X_1)$$



Se avessi usato il RESET come elemento

determinante, potrei prendere

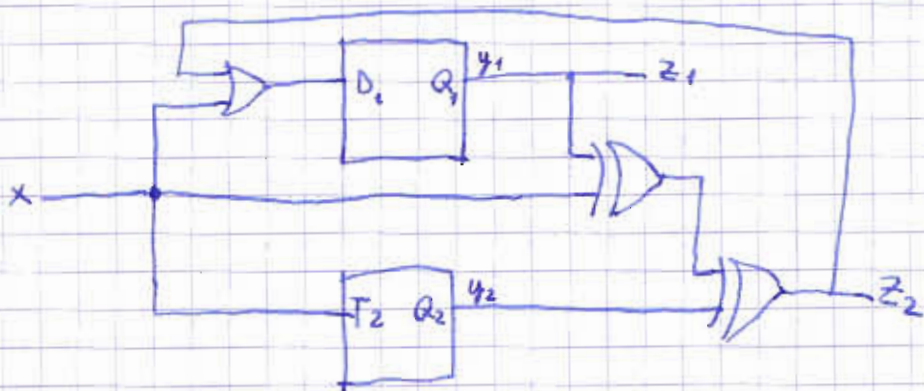
$$E=1 \text{ e } R = \bar{y}_1 (X_0 \oplus X_1)$$

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

CODIFICA ONE-HOT \rightarrow solo un FF è attivo
 uso N bit

A B C
 100
 010
 001
 Per N stati

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



Il clock è sottinteso.
EX-OR = disparità di 1

$T_2 = X$

$Z_1 = Y_1$

$Z_2 = X \oplus Y_1 \oplus Y_2 = X\bar{Y}_1\bar{Y}_2 + XY_1Y_2 + X\bar{Y}_1Y_2 + X\bar{Y}_1\bar{Y}_2$

$D_1 = X + Z_2 = X + X\bar{Y}_1\bar{Y}_2 + XY_1Y_2 + X\bar{Y}_1Y_2 + X\bar{Y}_1\bar{Y}_2 = X + \bar{Y}_1Y_2 + Y_1\bar{Y}_2 = Y_1^{n+1}$
 integrato in x oppure a x quindi inutile esplicitare, altrimenti lo avrei riscritto

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

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

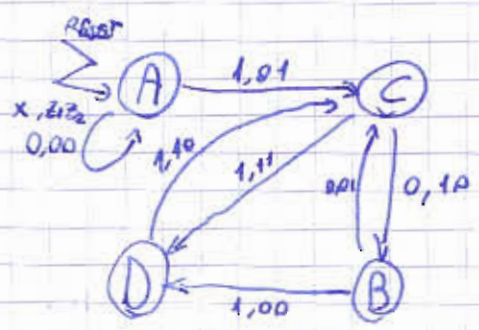
Chiamo A=00, B=01, C=11, D=10

X \ Y ₂	0	1
00	00, 00	11, 01
01	11, 01	10, 00
11	01, 10	10, 11
10	10, 11	11, 10

(Y₁Y₂)ⁿ⁺¹, (Z₁Z₂)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'automata 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 parta dallo stato iniziale in tale stato.

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

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

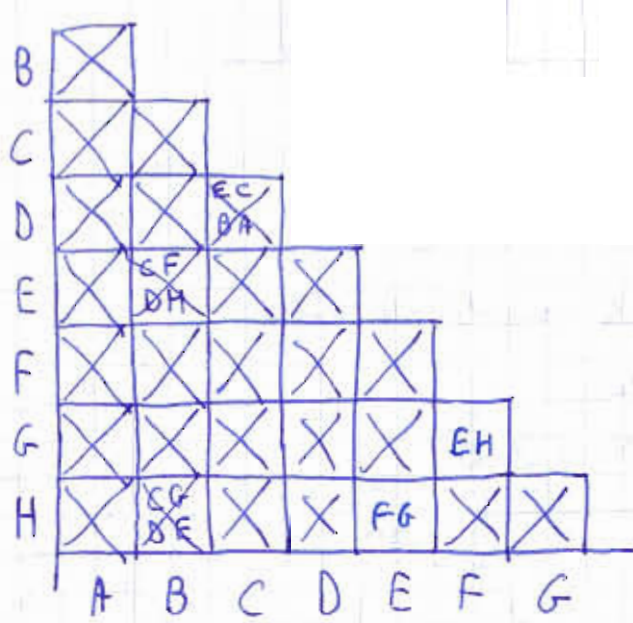
- $U(S_i, i_a) = U(S_j, i_a)$ le uscite sono uguali.
- $S^*(S_i, i_a) \sim S^*(S_j, i_a)$ gli stati futuri sono indistinguibili.

L'equivalenza di S_0 e S_3 dipende da quella di S_1, S_2 .

Casella vuota \rightarrow equivalenti Casella X \rightarrow non equivalenti.

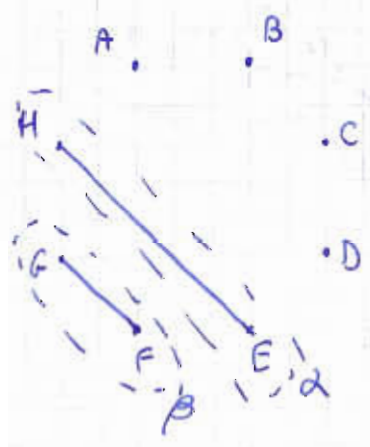
ES. D1 DEL 22/12/04

	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



EH \rightarrow FG

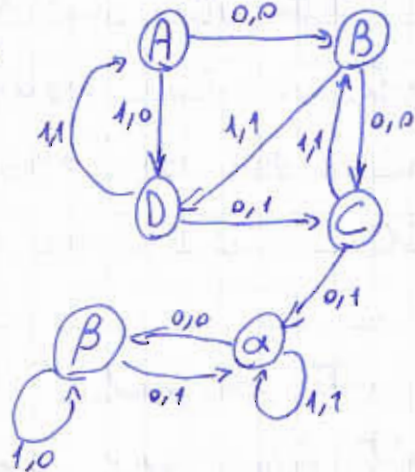
Classi massime: A, B, C, D, $\frac{EH}{\alpha}$, $\frac{FG}{\beta}$



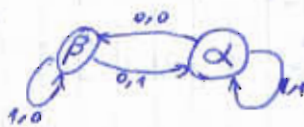
	0	1
A	B,0	D,0
B	C,0	D,1
C	α ,1	B,1
D	C,1	A,1
α	β ,0	α ,1
β	α ,1	β ,0

TABELLA DI FLUSSO MINIMA

DIAGRAMMA DEGLI STATI DELLA MACCHINA MINIMA



Assumo come stato iniziale α o β in modo da eliminare la parte ABCD e il diagramma diventa:

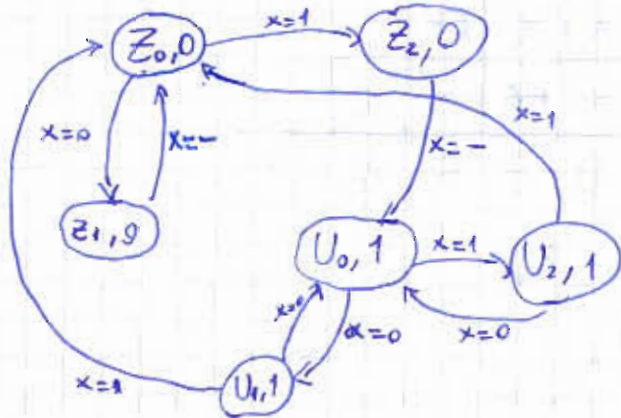


ES. ESAME 02/04



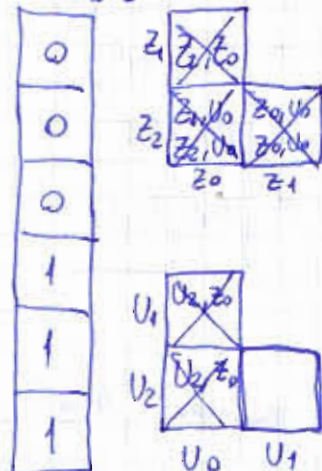
J	K	Q^{n+1}
0	0	Q^n
0	1	0
1	0	1
1	1	\bar{Q}^n

- X surge da J e K i 2 intervalli consecutivi
- Z cambia dopo due intervalli
- usare Moore (uscita associata allo stato)
- iniziare con z_0 in cui $z=0$



	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	

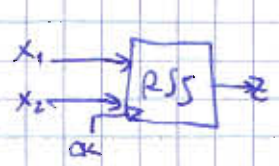
uscita z



$$u_1 \cup u_2 = K$$

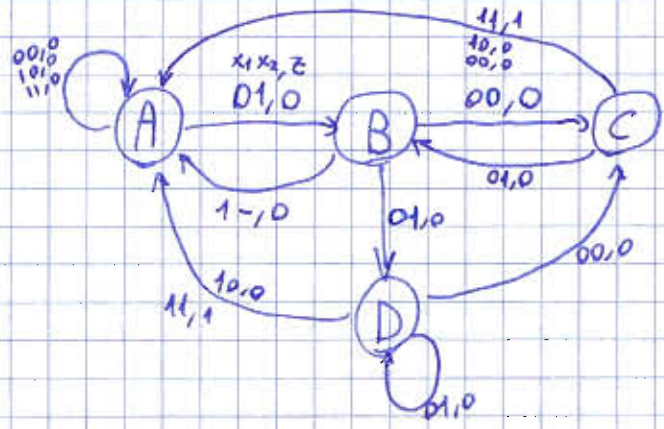
	x	0	1	
z_0	z_1	z_2		0
z_1	z_0	z_0		0
z_2	u_0	u_0		0
u_0	K	K		1
K	u_0	z_0		1

AUTOMA MINIMO



$$\begin{matrix} x_1 & x_2 \\ t-2 & 0 & 1 \\ t-1 & 0 & - \\ t & 1 & 1 \end{matrix} \quad z=1$$

fare diagramma stati (n°S=4)



	$x_1 x_2$	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

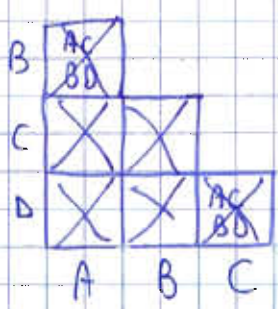


TABELLA DI FLUSSO MINIMA

$x_1 x_2$	$y_1 y_0$	00	01	11	10
A=00	00	00,0	01,0	00,0	00,0
B=01	01	11,0	10,0	00,0	00,0
C=11	11	00,0	01,0	00,1,1	00,0
D=10	10	11,0	10,0	00,1,1	00,0

$(y_1, y_0)^{n+1}, z^n$

$x_1 x_2$	$y_1 y_0$	00	01	11	10
00		0	0	0	0
01		1	1	0	0
11		0	0	0	0
10		1	1	0	0

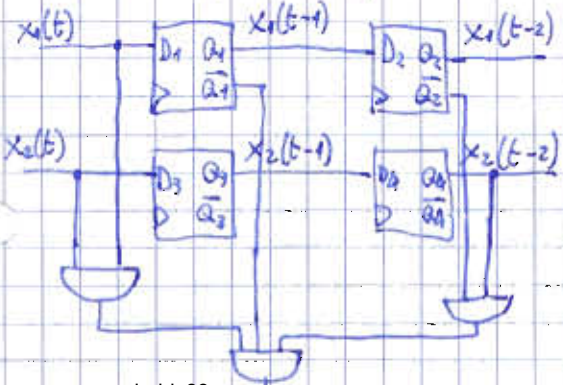
$x_1 x_2$	$y_1 y_0$	00	01	11	10
00		0	1	0	0
01		1	0	0	0
11		0	1	0	0
10		1	0	0	0

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

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

$$z = y_1 x_1 x_2$$

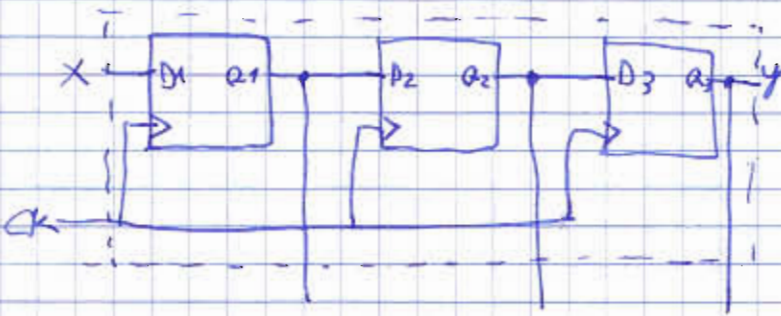
Un modo per rappresentare il circuito è



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



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

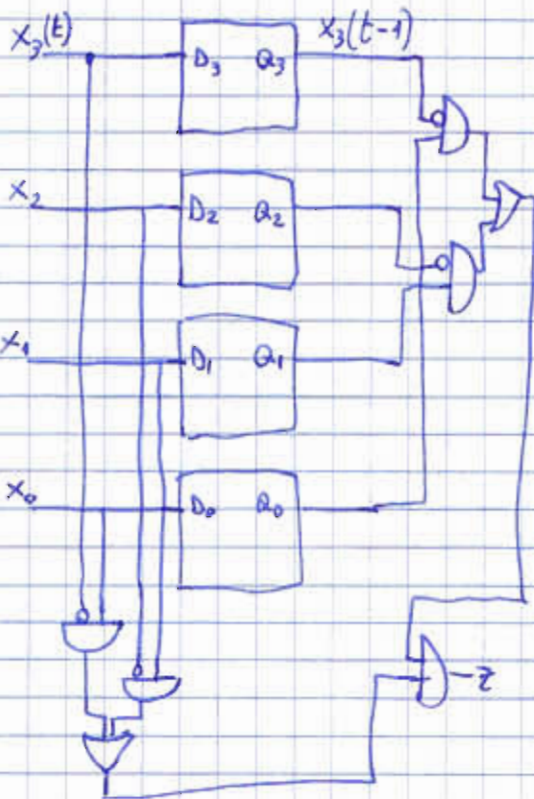
$$Q_2^{n+1} = D_2^n = Q_1^n = D_1^{n-1} = X^{n-1}$$

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



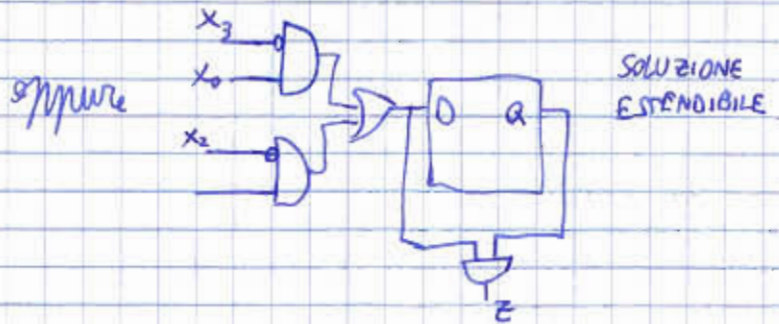
$Z=1$ se gli ultimi due numeri N e n+1 ricercati sono primi (1,2,3,5,7)
 0001, 0010, 0011, 0101, 0111

BCD \rightarrow 1° bit: segno



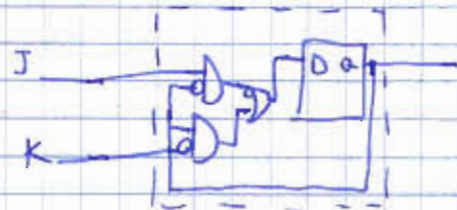
$X_3 X_2$ $X_1 X_0$	00	01	11	10
00	0	1	1	1
01	0	1	1	0
11	-	-	-	-
10	0	0	-	-

$$P = \bar{X}_3 X_0 + X_2 X_1$$



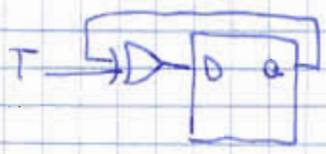
Con un FF-D sintetizzare un FF-JK

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



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

TT	0	1
0	0	1
1	1	0



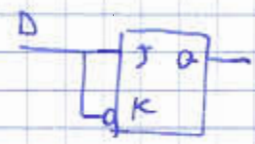
$$Q^{n+1} = D^n = \bar{Q}T + TQ = Q \oplus T$$

CON J-K : FF-D

a/D	0	1
0	0	1
1	0	1

a/D	0	1
0	0	1
1	-	-

a/D	0	1
0	-	-
1	1	0



FF-T

Q^{n+1}

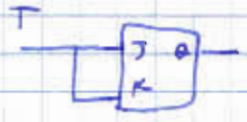
$J=D$

$K=\bar{D}$

a/T	0	1
0	0	1
1	1	0

a/T	0	1
0	0	1
1	-	-

a/T	0	1
0	-	-
1	0	1



$J=T$

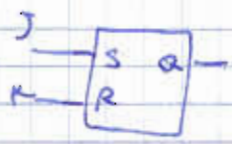
$K=T$

FF-SR

a/SR	00	01	11	10
0	0	0	-	1
1	1	0	-	1

a/SR	00	01	11	10
0	0	0	-	1
1	-	-	-	-

$J=S$



a/SR	00	01	11	10
0	0	-	-	-
1	0	1	-	0

$K=R$

16/12 NO LEZIONE

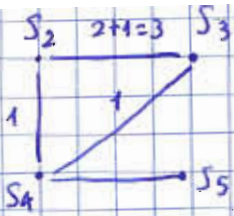
9/12/08

METODI EURISTICI X LA CODIFICA DEGLI STATI

- codificare consecutivamente 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 _{2,0}	S _{2,0}
S ₃	S _{2,0}	S _{2,0}
S ₄	S _{2,0}	S _{2,1}
S ₅	S _{2,0}	S _{2,0}

- 1) Stati con lo stesso stato futuro
 - S₂-S₃ con peso 2 (colonne =)
 - S₂-S₄ con peso 1
 - S₃-S₄ con peso 1
- 2) stati futuri (non colonne adiacenti) di uno stesso stato
 - S₂-S₃ peso 1
 - S₄-S₅ peso 1
- 3) stesso uscita in una o più colonne poco rilevante → trascurare.



S1	S2	S3	
	S5	S4	

valore = 5 (trascuro S2-S4)

MACCHINE NON COMPLETAMENTE SPECIFICATE

$S_i \sim S_j$ stati equivalenti

$S_i \approx S_j$ stati compatibili \rightarrow se s_i ha la stessa uscita (coppia) per ogni ingresso in cui s_i abbia 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-BAND (limiti inferiori, eventualmente non raggiungibile per la dimensione dell'automa minimo equivalente).

ES. ESAME

1) RAGGIUNGIBILITÀ: $A \rightarrow D \rightarrow F \rightarrow C$

B, E NON RAGGIUNGIBILI

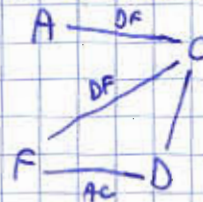
	0	1
A	D, 0	-
B	-	B, 1
C	F, -	-
D	F, 1	A, 1
E	D, -	G, 0
F	D, 1	C, -

C	DF	
D	X	\approx
F	X	DF AC
	A	C D

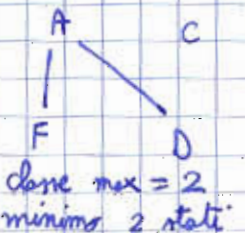
2) Coppie di stati compatibili:

AC, DF, CF, CD

GRAFO DI COMPATIBILITÀ



GRAFO INCOMPATIBILITÀ

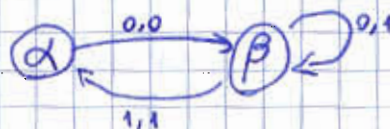


3) classi max compatibilità:

$\{\alpha\}, \{\beta\}$

	0	1
α	$\beta, 0$	-
β	$\beta, 1$	$\alpha, 1$

\Rightarrow macchina minima!



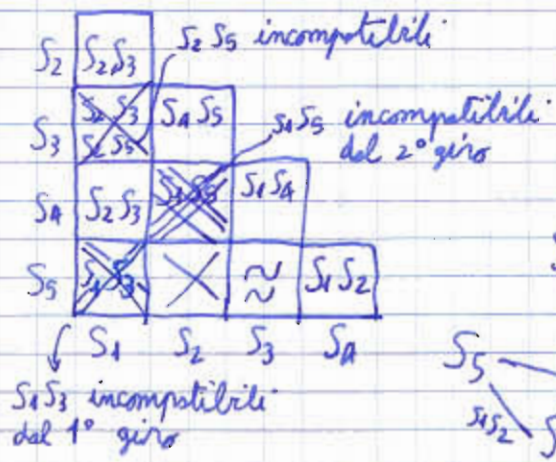
altro automa normale: $\alpha = \{A, C\}$, $\gamma = \{D, F\}$

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

stemo /
automa

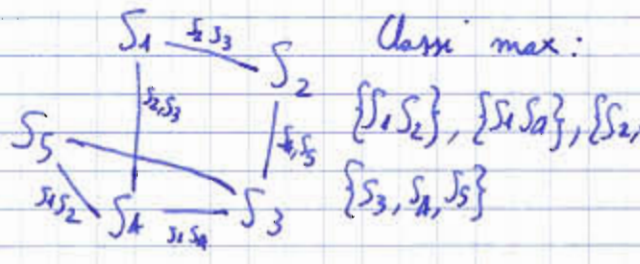
ES. ESAME

$x_1 x_2$	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	-	-	$S_4, 1$	-

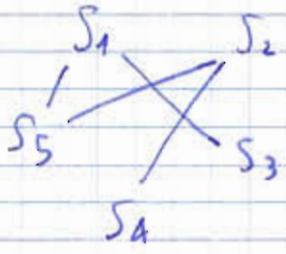


COPPIE DI STATI COMPATIBILI

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



Grfo incompatibilità:



$\max = 2$
 \hookrightarrow minimo 2 stati
 non possibile

Valutiamo il seguente insieme di classi selezionate: $\alpha = \{S_1, S_2\}$ $\beta = \{S_3, S_4, S_5\}$

$x_1 x_2$	00	01	11	10
α	?			
β				

nessuna classe contenente $S_2 S_3 \rightarrow$ non chiuso

Devo prendere $\{S_1, S_2\}$ che implica di prendere $\{S_2, S_3\}$, che implica $\{S_4, S_5\}$.

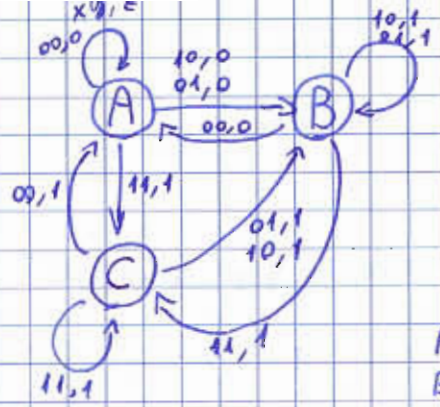
In questo modo copro tutte le classi \rightarrow macchina minima equivalente anche se non fatto da classi max.

$x_1 x_2$	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	-	$\alpha, 1$	$\alpha, 1$	-

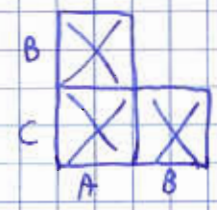
ES. SINTESI



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



xy	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



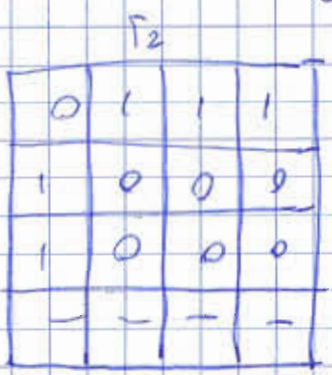
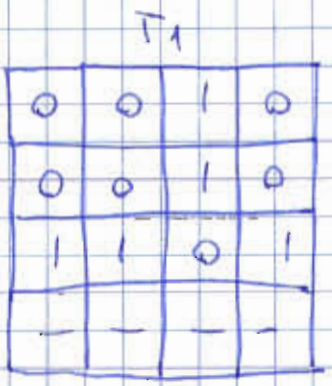
A=00
B=01
C=11

$wxyz$	00	01	11	10
00	00,0	01,0	11,1	01,0
01	00,0	01,1	11,1	01,1
11	00,1	01,1	11,1	01,1
10	-	-	-	-

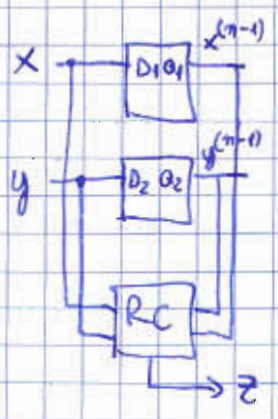
$(w_1, w_0)^{n+1}, z^n$

FF-T

$w^n \rightarrow w^{n+1}$			T_1
0	0	0	0
0	1	1	1
1	0	0	1
1	1	1	0

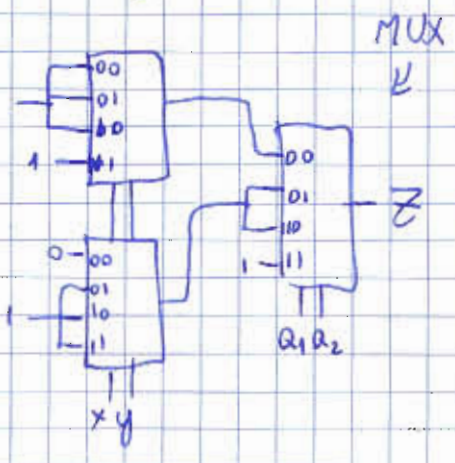


oppure



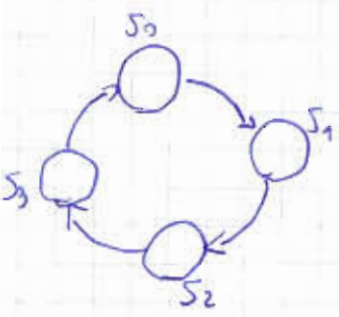
xy	00	01	11	10
00	0	0	1	0
01	0	1	1	1
11	1	1	1	1
10	0	1	1	1

Poi sintetizzo RC con porte



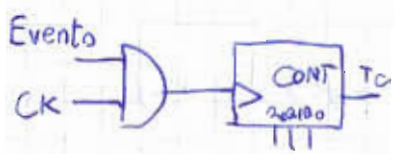
CONTATORI

↳ visita un numero finito di stati periodicamente

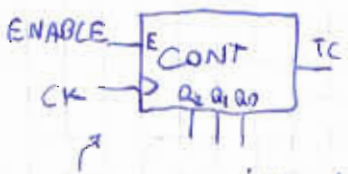


• MODULO = n° di stati visitati = 4 = BASE DI CONTEGGIO

- Serve per
- misurare il tempo
 - conteggiare degli eventi

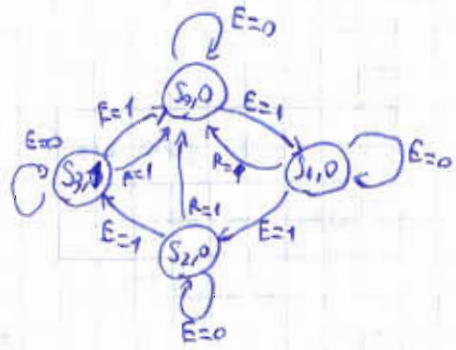


GATED-CLOCK → da non usare



↳ ingresso abilitato E=1, il contatore evolve

Circuiti non più autonomi →



• INGRESSI DI CONTROLLO →

non tutti sempre presenti

- ENABLE
- RESET
- LOAD

rimozioni

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

- RESET
- LOAD

inserzioni

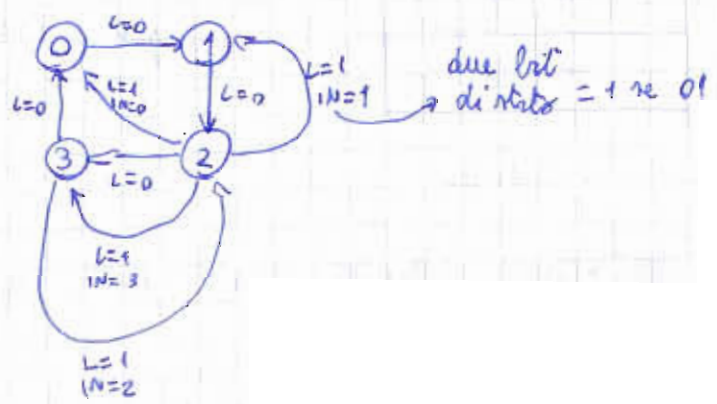
LOAD → permette di "caricare" una particolare configurazione di stato

• CODICE UTILIZZATO PER LA CODIFICA DELLO STATO

↳ a numero minimo di bit

- ↳ binario (00-01-10-11-00...)
- ↳ Gray (00-01-11-10-00...)
- ↳ BCD

↳ a numero non minimo di bit (codifica ridondante)



- ↳ Johnson
- ↳ Codici legati 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
001	011
011	100
010	111
100	010
101	001
111	001
110	000

y_2	$y_1 y_0$	00	01	11	10
0	-	011	100	111	-
1	010	-	001	-	-

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

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

y_2	$y_1 y_0$	00	01	11	10
0	-	0	1	1	-
1	010	1	1	1	-

y_2	$y_1 y_0$	00	01	11	10
0	-	1	1	0	-
1	010	1	1	1	-

$T_2 = y_1 + y_2$

$T_1 = y_0 + y_2$

y_2	$y_1 y_0$	00	01	11	10
0	-	0	1	1	-
1	010	0	0	-	-

$T_0 = y_1 \bar{y}_2$

⇒ per circuito...

CONTATORI BINARI CON FF-JK

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

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

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

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

y_i	0	1
0	0	1
1	1	0

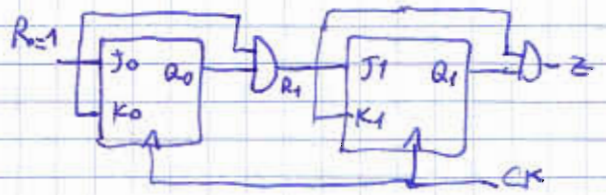
$(y_i)^{n+1}$

0	1
-	-

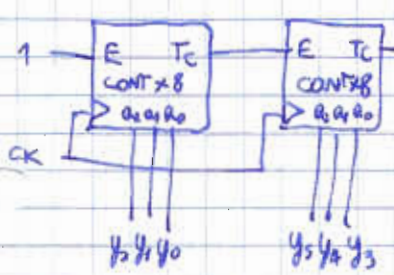
$J = R_i$

-	-
0	1

$K = R_i$



Se al posto della costante, come R_0 metto E (ingresso enable) ottengo un contatore binario dotato di enable.



ottengo un contatore di modulo $8 \cdot 8 = 64$

Introduco ora l'ingresso di RESET.

E	R	
0	0	hold
0	1	RESET
1	1	RESET
1	0	count

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

y_i^{n+1}

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

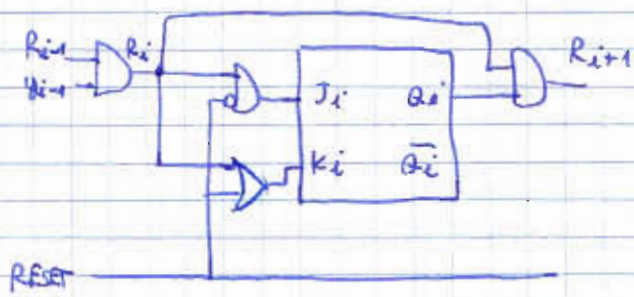
$J = R_i \bar{R}$

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

$K = R_i + R$

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

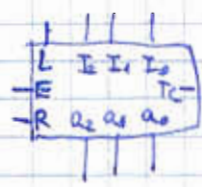
R_i = ripeto in ingresso
 R = reset



Introduco il LOAD

E	R	L	
0	0	0	HOLD
0	0	0	RESET
0	1	1	RESET
0	0	1	COUNT
1	0	0	I^n
1	1	0	RESET \rightarrow prevale su LOAD
1	1	1	RESET

$I^n \rightarrow$ configurazione specificata su ingressi esterni



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

Se considero un contatore senza reset:

$R_i I_i$	00	01	11	10	00	01	11	10
y_i	0	0	1	1	0	1	1	0
	1	1	0	0	0	1	1	0

$L=0$ $L=1$

$R_i I_i$	00	01	11	10	00	01	11	10
0	0	0	1	1	0	1	1	0
1	-	-	-	-	-	-	-	-

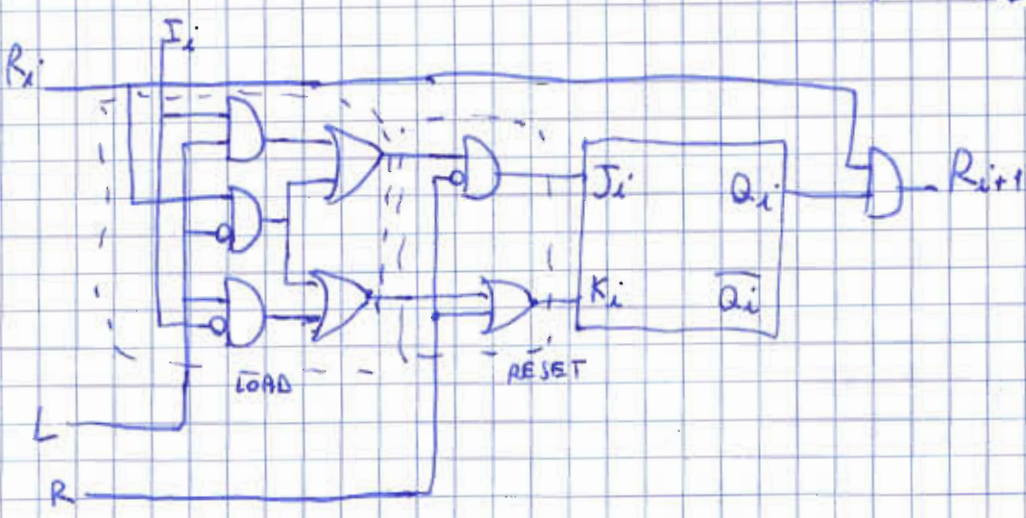
$L=0$ $L=1$

$R_i I_i$	00	01	11	10	00	01	11	10
0	-	-	1	1	-	-	-	-
1	0	0	1	1	1	0	0	1

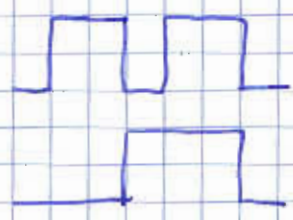
$L=0$ $L=1$

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

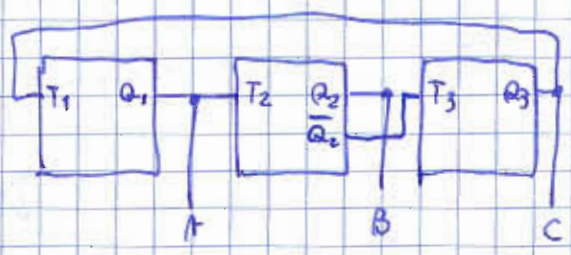
$$K_i = R_i \bar{L} + \bar{I}_i L$$



ES.DB ESAME 12/2/06



F1 periods 3
F2 periods 6



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

$$T_1 = Q_3$$

$$T_2 = Q_1$$

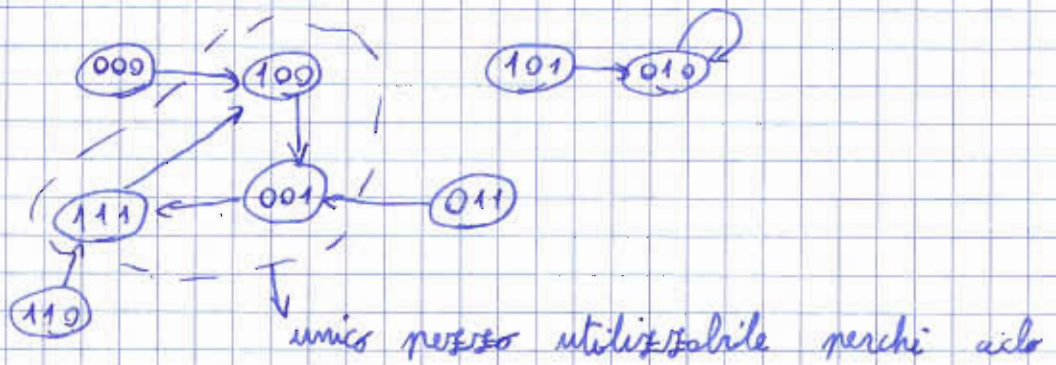
$$T_3 = \bar{Q}_2$$

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

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

$$Q_3^{n+1} = (Q_3 \oplus \bar{Q}_2)^n = Q_3 Q_2 + \bar{Q}_3 \bar{Q}_2$$

$Q_3 Q_2 Q_1$				
000	1	0	0	
001	1	1	1	
011	0	0	1	
010	0	1	0	
100	0	0	1	
101	0	1	0	
111	1	0	0	
110	1	1	1	
				$(Q_3 Q_2 Q_1)^{n+1}$



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

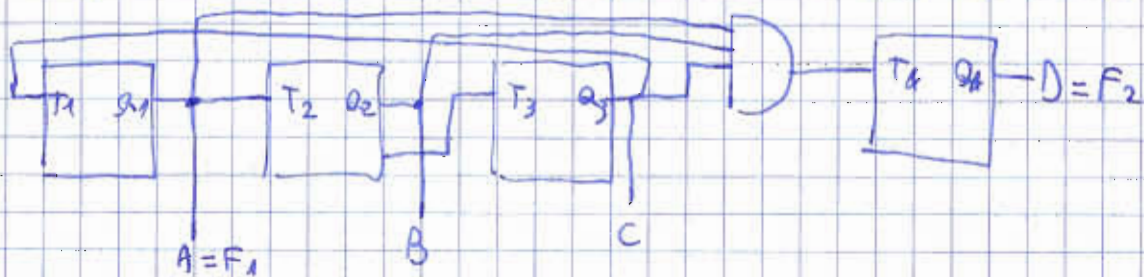
Espresso come ABC, cioè come $Q_1 Q_2 Q_3$, ottengo 001-100-111.

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.

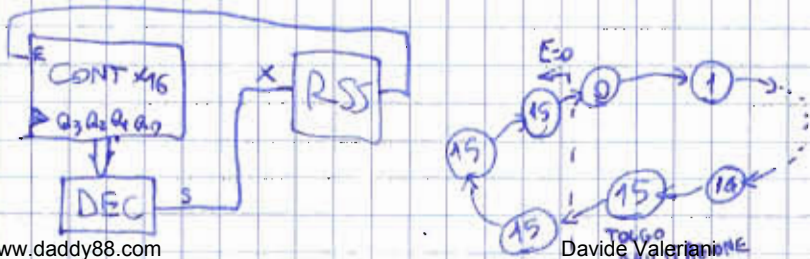
Il più semplice è E-T-Q.

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_4	0	0	1	0	0	1	0		
Q_4	0	0	0	1	1	1	0		$\dots = F_2$

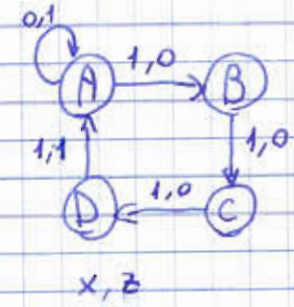
Dato un cont. x16 con enable realizzare un cont. x19.



Meady

X \ Y	0	1
A	A, 1	B, 0
B	-	C, 0
C	-	D, 0
D	A, 1	-

il contatore è fermo pertanto $X=1$

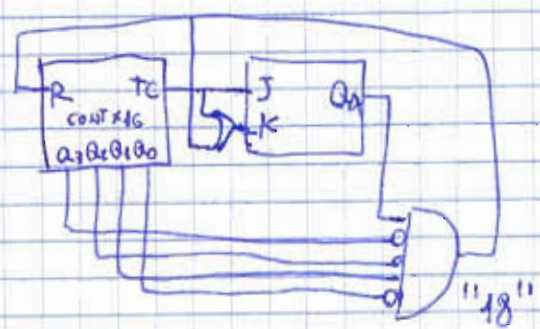


Moore

X \ Y	0	1
A	A, 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 realizzare un CONT x 19



Volendo semplificare l'AND

$Q_3 Q_2$	$Q_1 Q_0$
0 0	0 0
0 0	0 0
0 0	0 0
0 0	0 0

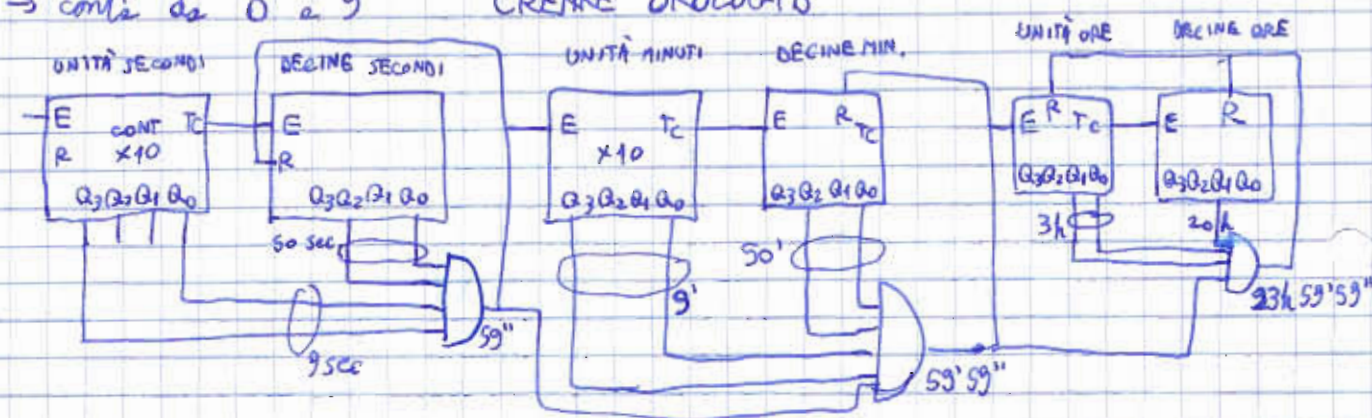
$Q_3=0$



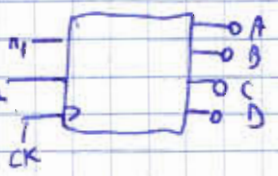
Reset = $Q_3 Q_1$ COSTO MINIMO
 Reset = $Q_3 Q_1 + Q_2 Q_4 + Q_3 Q_4$ TRANSIZIONI MINIME

BCD → conta da 0 a 9

CREARE OROLOGIO

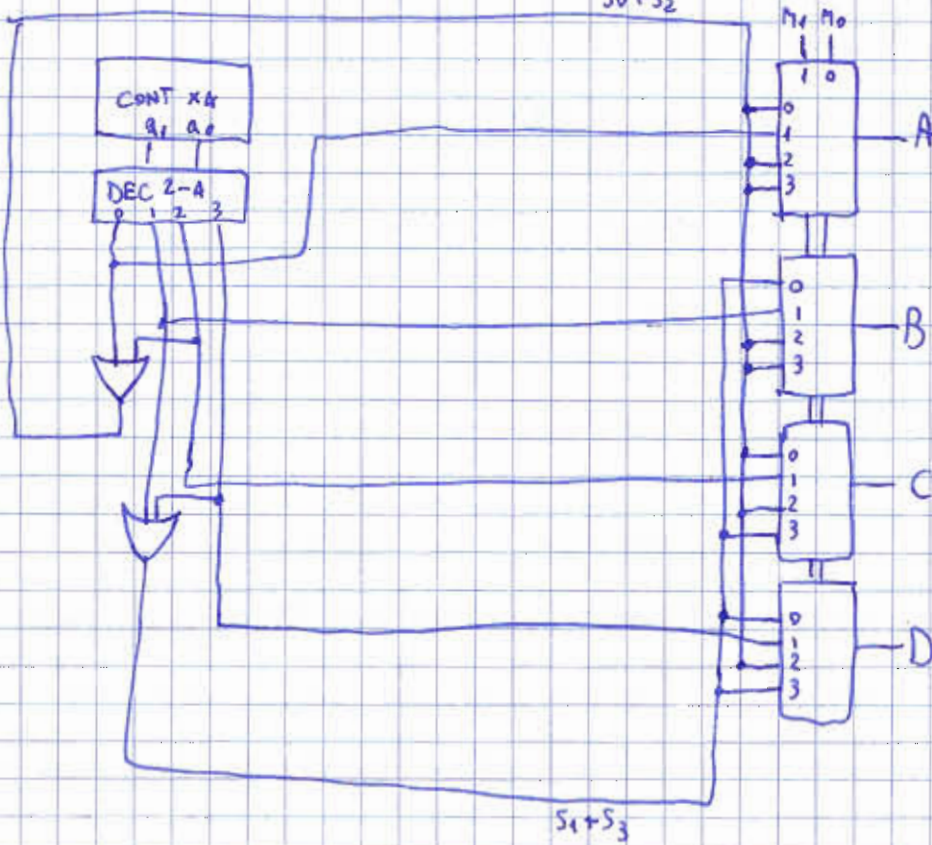


18/12/08



M, M0	00	01	10	11
a1 a0	AC	A	ABCD	AB
0 0	BD	B	-	CD
0 1	AC	C	ABCD	AB
1 0	BD	D	-	CD
1 1	-	-	-	-

Luci di Natale

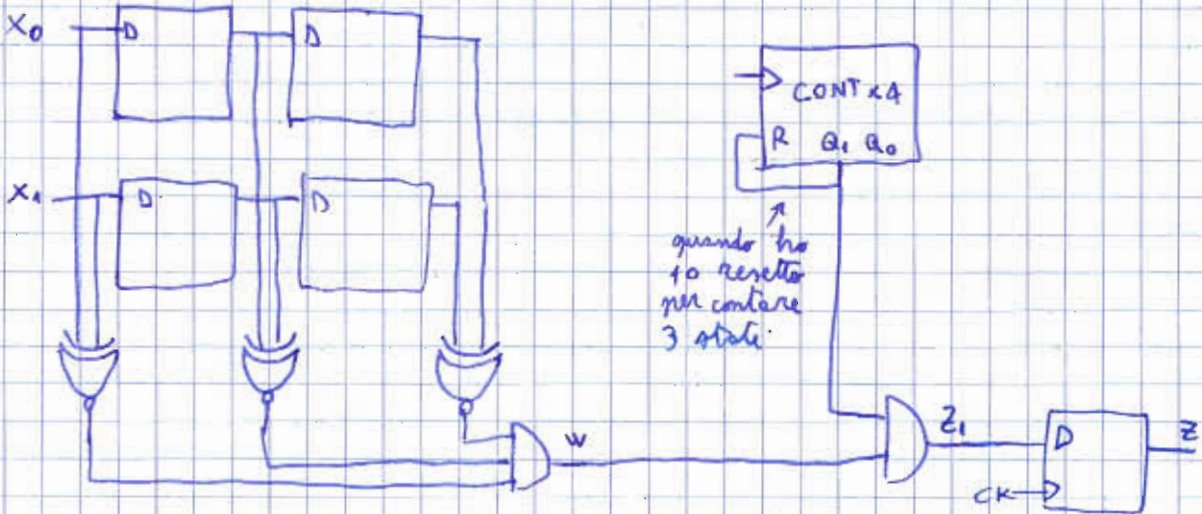


22/12/2003 DZ



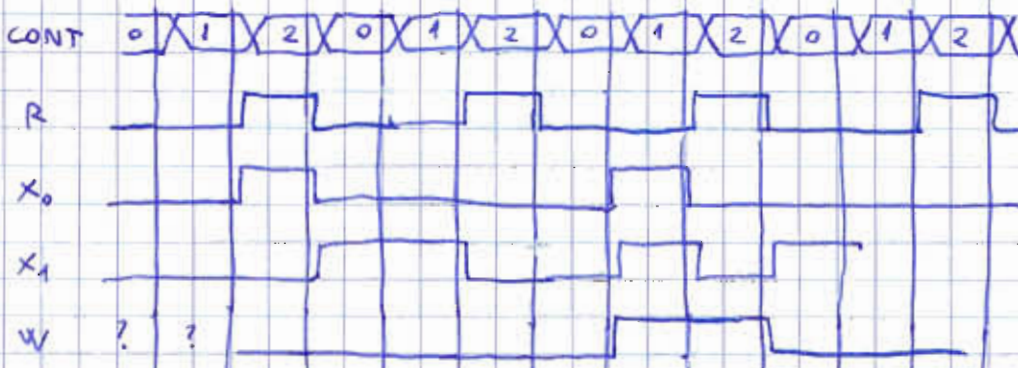
x_0 0 1 1 | 0 0 1 | 1 1 0 |
 x_1 1 0 0 | 0 1 1 | 1 1 0 |
 z 0 0 0 | 0 0 1 | 0 0 0 | 1

consideriamo prima che l'uscita vari qui.

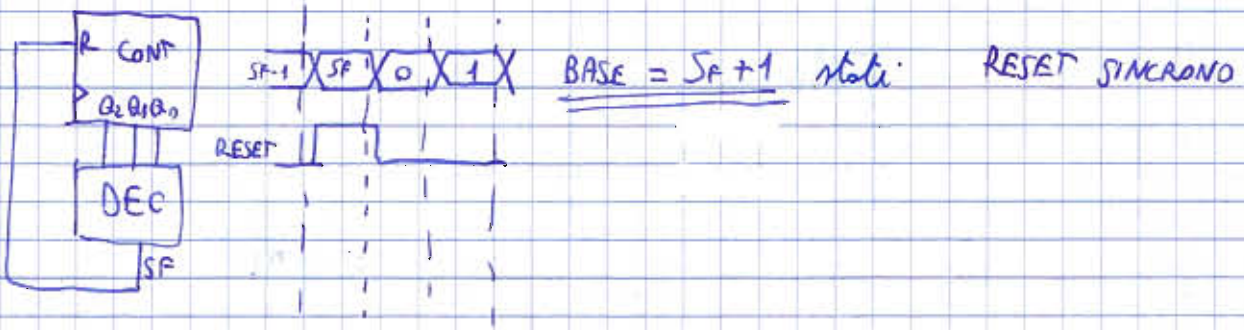


quando ho to resetto per contare 3 stati.

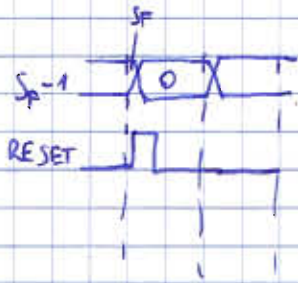
Attenzo l'uscita ritardata!



oppure mettere 3 FF-D nei registri 2 scricimento invece che solo 2. In questo modo analizzo 1 intervallo dopo



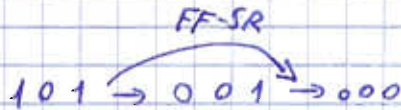
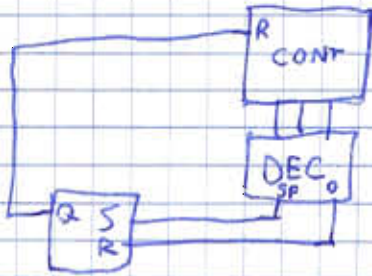
USO DI RESET ASINCRONO PER RIDUZIONE BASE



BASE = SF

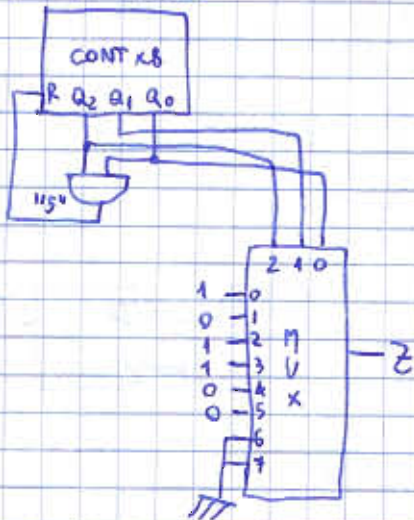
Il reset non dura un intero intervallo di clock. Luttanza, se uno dei FF è più veloce degli altri, si va in una configurazione in cui il RESET non è attivo → ERRORE

→ SOLUZIONE

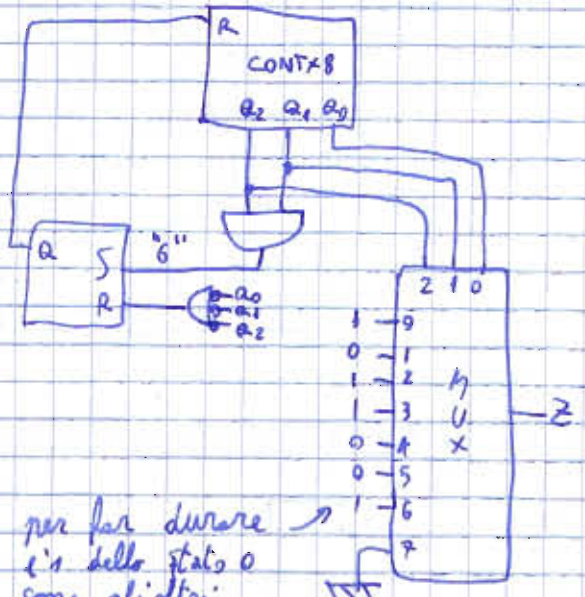


Dato un CONT x8 con RESET o/e LOAD sincroni realizzare una forma d'onda periodica 101100

MODO 1 - RESET SINCRONO



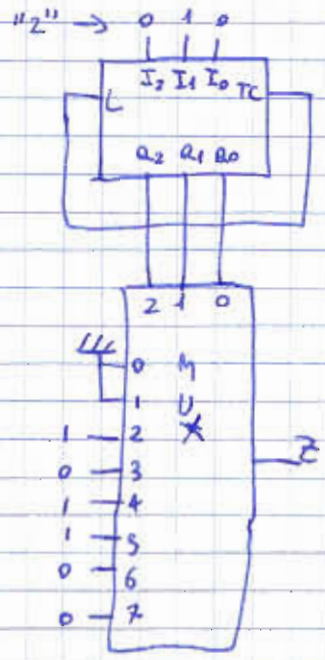
MODO 2 - RESET ASINCRONO



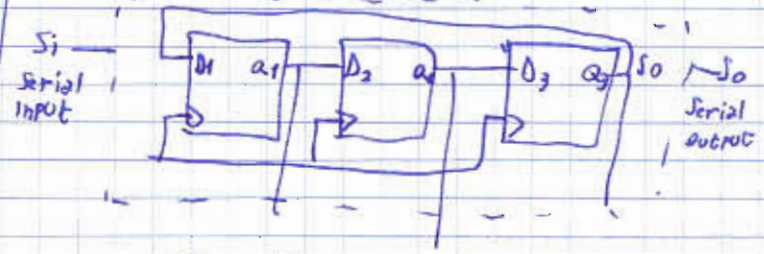
per far durare l'is dello stato 0 come gli altri ⇒ elimina i ritardi.

Modo 2 - LOAD A PARTIRE DALLO STATO FINALE

7 → 2 → 3 → 4 → 5 → 6 → 7 → 2...



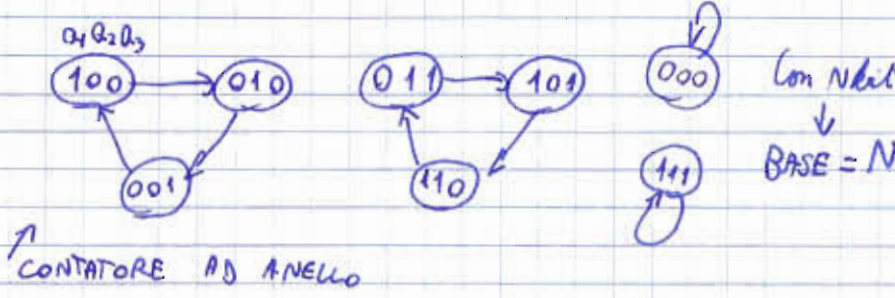
REGISTRO A SCORRIMENTO



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

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

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



CONTATORE AD ANELLO

Problema dell'inizializzazione → PRE-SET sincrono

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



$$Q_1^{n+1} = \bar{Q}_3^n$$

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

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

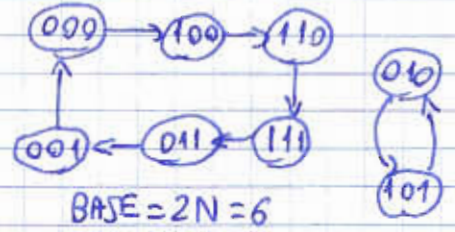
CONTATORE JOHNSON

RIEMPIMENTO/SVUOTAMENTO

↳ di 1 da sinistra
di 0 da destra

$Q_2 Q_1 Q_0$	000	001	010	011	100	101	110	111
	1	0	0	0	1	1	0	1

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



$Q_2 Q_1 Q_0$	00	01	11	10
0	S ₁	S ₆	S ₅	S ₄
1	S ₂	S ₃	S ₁	S ₃

$$S_1 = \bar{Q}_1 \bar{Q}_3 \quad S_4 = Q_1 Q_3$$

$$S_2 = Q_1 \bar{Q}_2 \quad S_5 = \bar{Q}_1 Q_2$$

$$S_3 = Q_2 \bar{Q}_3 \quad S_6 = \bar{Q}_2 Q_3$$

Ogni stato è vicino a un don't care → mi bastano AND a 2 ingressi qualunque sia la base.

ESAME 2-2006

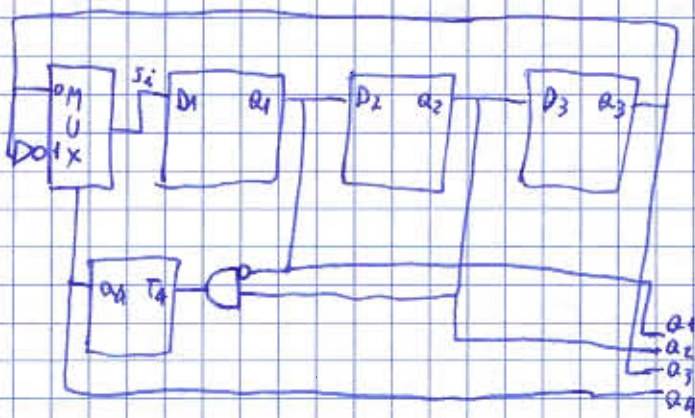
Dato uno shift register a 3 bit realizzare un contatore in base 9.

CONTEGGIO JOHNSON

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

CONTEGGIO AD ANELLO

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



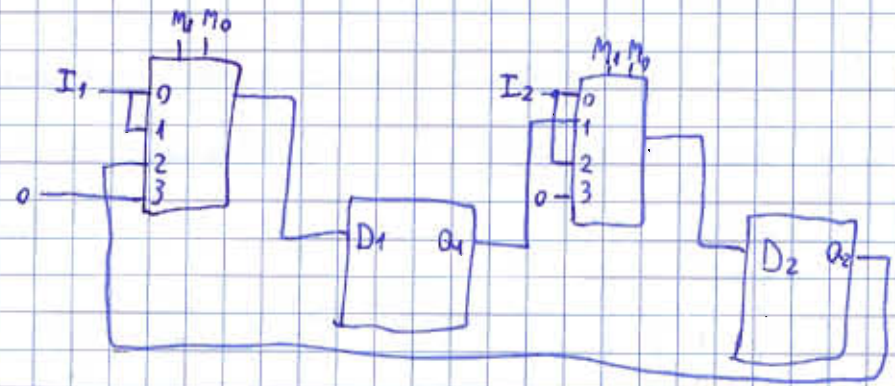
Passo da Johnson ad anello

- 000 Q_4
- 100
- 110
- 111
- 011
- 001
- 100
- 010
- 001

001 rappresenta la fine di entrambe le sequenze

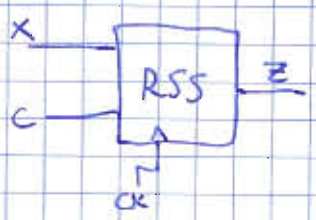
configura la situazione che lo precede \rightarrow 011 e 010 $\rightarrow Q_1=0$ e $Q_2=1$

Così gli segue lo stato dell'altro metodo



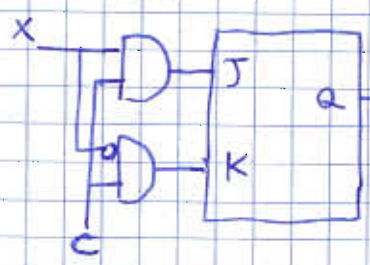
M1	M0	
0	0	REGISTRO PARALLELO
0	1	SHIFT RIGHT
1	0	SHIFT LEFT
1	1	RESET SINCRONO

CAMPIONAMENTO SINCRONO

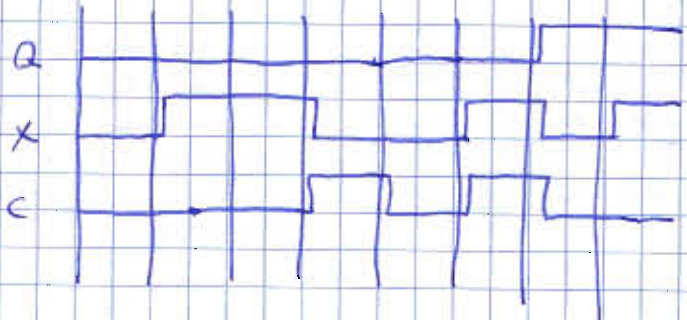


$$Z^{n+1} = X^n \quad \text{se } C^n = 1$$

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



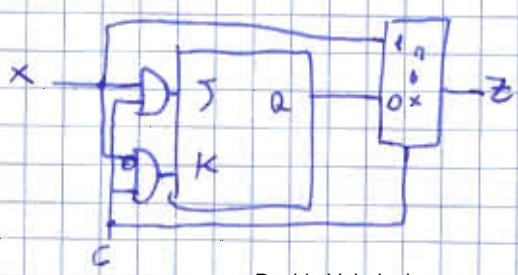
CAMPIONATORE "ALLA MEALY"



CAMPIONATORE "ALLA MEALY"

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

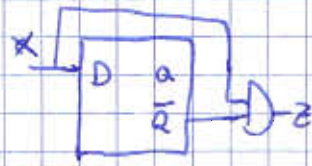
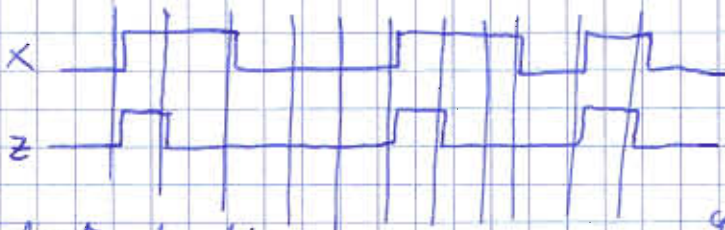
$$Z^n = Z^{n-1} \quad \text{se } C^n = 0$$



L'uscita dipende dall'ingresso corrente

DERIVATORE SINCRONO

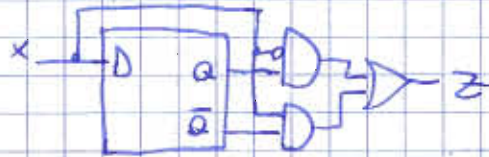
In fronte di salita



In fronte di discesa

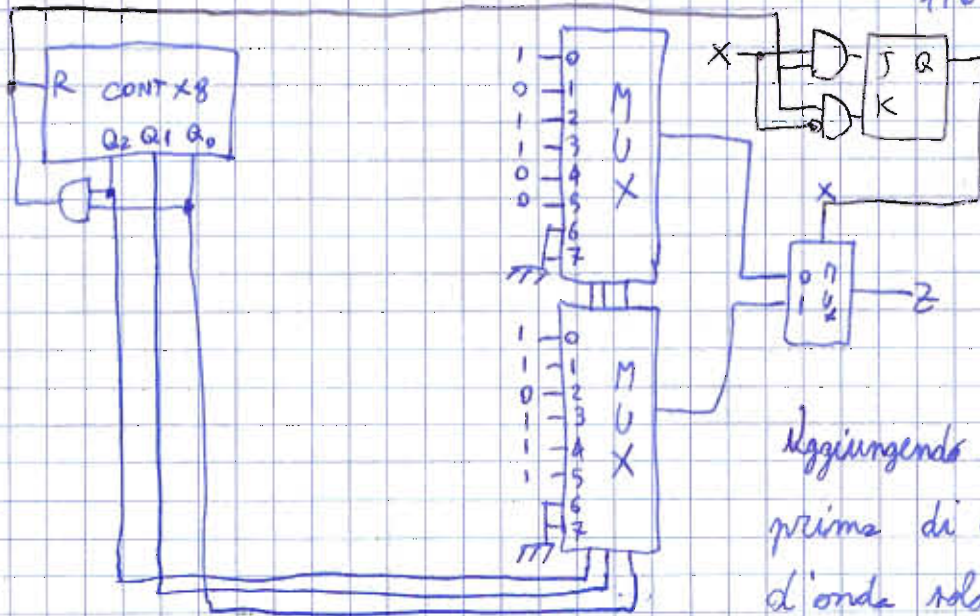


In entrambi i fronti



Circuito che realizza una delle forme d'onda

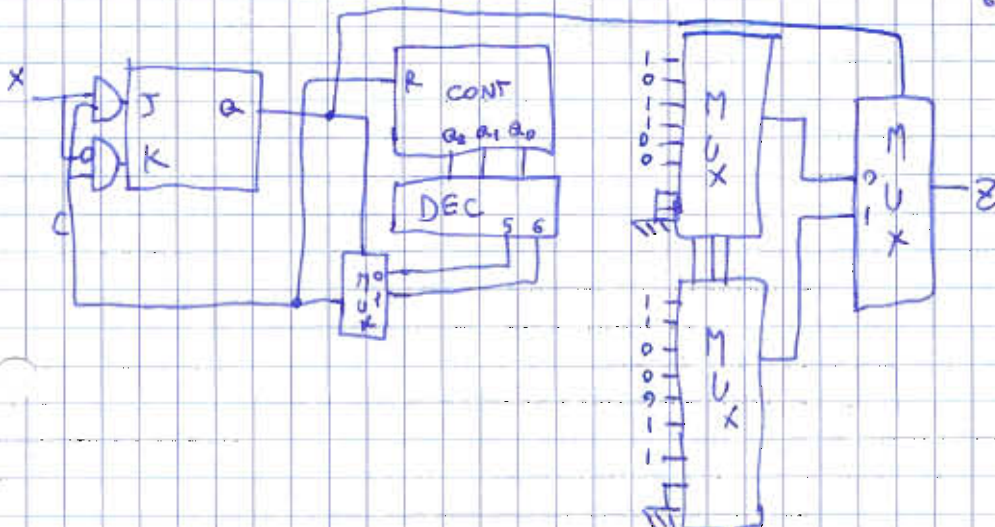
101100
110111

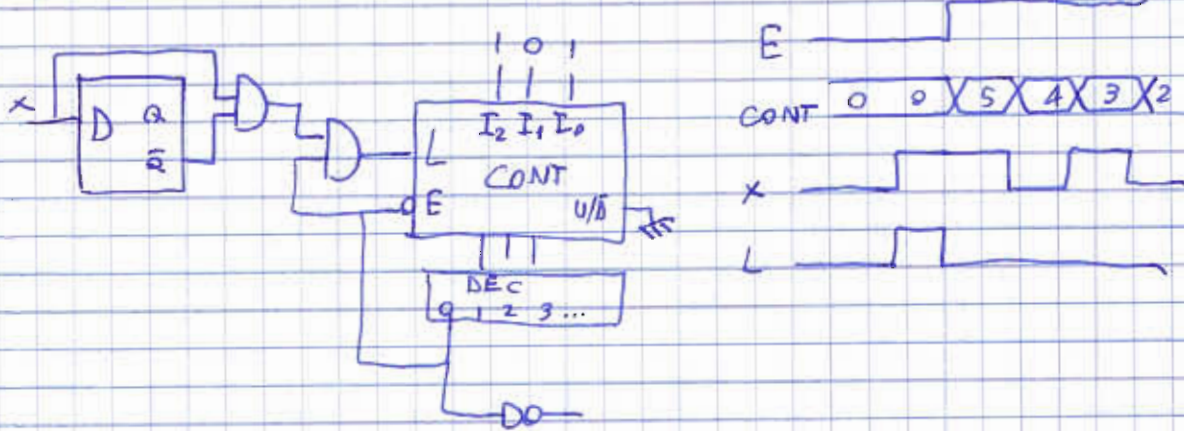


aggiungendo la parte in nero, prima di commutare le forme d'onda solo quando sono finite.

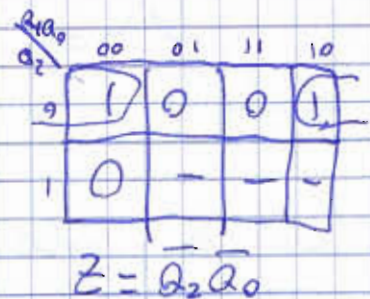
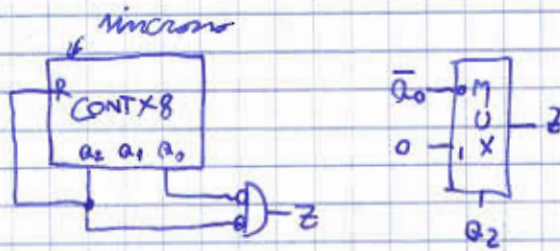
Questo funziona solo se le sequenze hanno durata.

Nel caso in cui si abbia da scegliere $\underbrace{101100}_6$ o $\underbrace{1100011}_7$

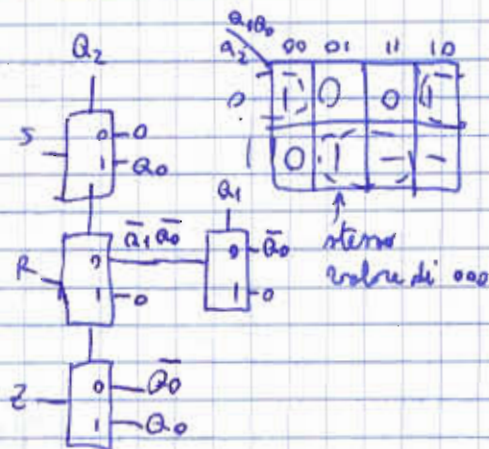
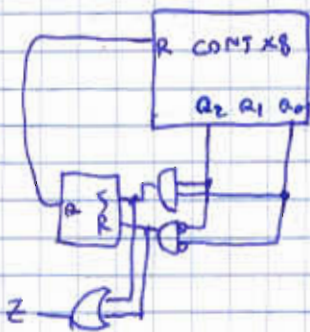




CONTATORE

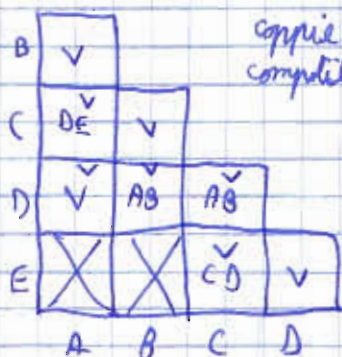


minicomb

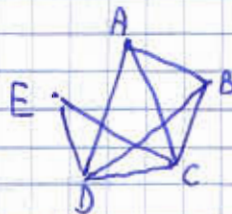


MINIMIZZAZIONE

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

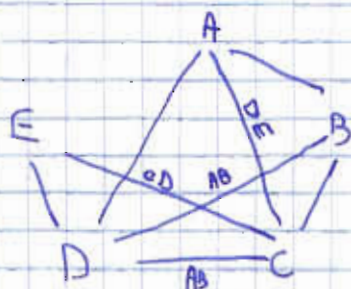


copie compatibili: AB, AC, AD, BC, BD, CD, CE, DE



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

Classi disgiunte:

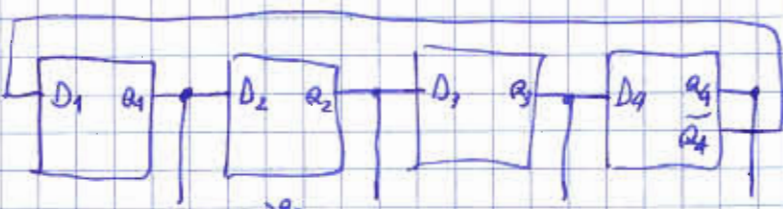


$\alpha = \{ABC\}$ $\beta = \{DE\}$

oppure

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

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



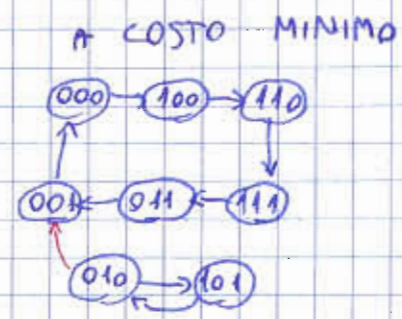
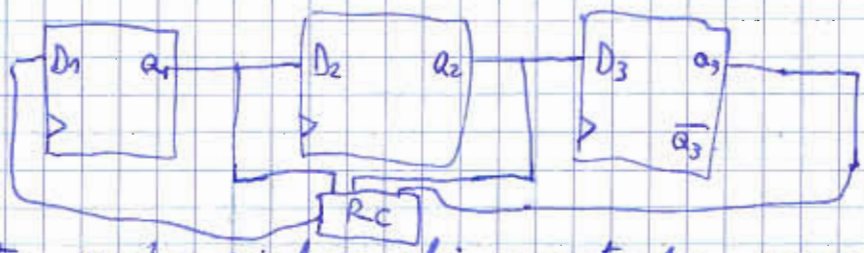
- Z0 0000
- Z1 1000
- Z2 1100
- Z3 1110
- Z4 1111
- Z5 0111
- Z6 0011
- Z7 0001
- Z8 0000

$Q_1 Q_2$	00	01	11	10
00	-	0	0	-
01	-	-	0	-
11	0	-	0	0
10	0	-	-	-

$Z_0 = \overline{Q_1} \overline{Q_2}$

$Z_1 = Q_1 \overline{Q_2}$
 $Z_2 = Q_2 \overline{Q_3}$
 $Z_3 = Q_3 \overline{Q_4}$
 $Z_4 = Q_1 Q_4$
 $Z_5 = \overline{Q_1} Q_2$
 $Z_6 = \overline{Q_2} Q_3$
 $Z_7 = \overline{Q_3} Q_4$

N FF	BASE	DECODIFICA USCITE
cont. lin.	2^N B	2^N AND a N ingressi $k = \log_2 B$ FF, B AND a K ingressi
cont. anello	N B	Gratis /
cont. Johnson	$2N$ B	$2N$ AND a 2 ingressi $B/2$ FF, B AND a 2 ingressi



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

$Q_2 Q_3$	00	01	11	10
0	100	000	001	001
1	110	-	011	111

1	0	0	0
1	-	0	1

$D_1 = \overline{Q_2} \overline{Q_3} + Q_1 \overline{Q_3}$

