A.A. 2003/2004
Appunti ed integrazioni alle esercitazioni di Reti Logiche[*]

A cura di G. Palermo, F. Ferrandi, A. Antola

Ultimo aggiornamento, 2 luglio 2004

[*]Questi appunti sono stati possibili anche per il lavoro fatto da alcuni studenti del corso di Reti Logiche A - A.A. 2003-2004

1 Minimizzazione di espressioni logiche con le proprietà dell'algebra di Boole

Esercizio 1.1 (TdE 2002/2003) - Data la funzione logica:

$$\mathsf{F} = [(\mathsf{x}'\mathsf{y} + \mathsf{x})'\mathsf{z}]\mathsf{w} + \mathsf{w}'(\mathsf{xy})' + \mathsf{xw} + \mathsf{yw}' + \mathsf{x}'\mathsf{w}$$

applicare le regole dell'algebra per semplificare la forma riducendo il numero dei prodotti ed il numero dei letterali.

Soluzione

$$\begin{array}{lcll}
\mathsf{F} & = & [(\mathsf{x}'\mathsf{y}+\mathsf{x})'\mathsf{z}]\mathsf{w} + \mathsf{w}'(\mathsf{xy})' + \underline{(\mathsf{x}+\mathsf{x}')\mathsf{w}} + \mathsf{yw}' & \textit{per distributiva} \\
\mathsf{F} & = & [(\mathsf{x}'\mathsf{y}+\mathsf{x})'\mathsf{z}]\mathsf{w} + \mathsf{w}'(\mathsf{xy})' + \underline{(1)\mathsf{w}} + \mathsf{yw}' & \textit{per inverso} \\
\mathsf{F} & = & \underline{[(\mathsf{x}'\mathsf{y}+\mathsf{x})'\mathsf{z}]\mathsf{w}} + \mathsf{w}'(\mathsf{xy})' + \underline{\mathsf{w}} + \mathsf{yw}' & \textit{per elemento neutro} \\
\mathsf{F} & = & \mathsf{w}'(\mathsf{xy})' + \underline{\mathsf{w}} + \mathsf{yw}' & \textit{per assorbimento su } [(\mathsf{x}'\mathsf{y}+\mathsf{x})'\mathsf{z}]\mathsf{w}+\mathsf{w} \\
\mathsf{F} & = & \mathsf{w} + \underline{\mathsf{w}'[(\mathsf{xy})'+\mathsf{y}]} & \textit{per distributiva} \\
\mathsf{F} & = & \underline{\mathsf{w} + [(\mathsf{xy})'+\mathsf{y}]} & \textit{per } \mathsf{a}'\mathsf{b}+\mathsf{a} = \mathsf{b}+\mathsf{a} \\
\mathsf{F} & = & \mathsf{w} + (\underline{\mathsf{x}'+\mathsf{y}'} + \mathsf{y}) & \textit{per De Morgan} \\
\mathsf{F} & = & \mathsf{w} + (\mathsf{x}' + \underline{1}) & \textit{per inverso} \\
\mathsf{F} & = & \mathsf{w} + \underline{1} & \textit{per elemento neutro} \\
\mathsf{F} & = & \underline{1} & \textit{per elemento neutro}
\end{array}$$

∎

Esercizio 1.2 (TdE 2002/2003) - Data la funzione logica:

$$\mathsf{F} = [(\mathsf{xy}')'\mathsf{wz}]\mathsf{w} + \mathsf{w}(\mathsf{x}'\mathsf{y})' + (\mathsf{w}+\mathsf{yw}')' + \mathsf{x}'\mathsf{w}$$

applicare le regole dell'algebra per semplificare la forma riducendo il numero dei prodotti ed il numero dei letterali.

Soluzione

$$
\begin{array}{lcll}
F & = & (xy')'\underline{w}z + w(x'y)' + (w + yw')' + x'w & \textit{per idempontenza} \\
F & = & (xy')'wz + w(x'y)' + \underline{w'(yw')'} + x'w & \textit{per De Morgan} \\
F & = & (xy')'wz + w\underline{(x + y')} + w'(yw')' + x'w & \textit{per De Morgan} \\
F & = & \underline{(x' + y)}wz + w(x + y') + w'(yw')' + x'w & \textit{per De Morgan} \\
F & = & (x' + y)wz + \underline{wx + wy'} + w'(yw')' + x'w & \textit{per distributiva} \\
F & = & (x' + y)wz + wx + wy' + w'\underline{(y' + w)} + x'w & \textit{per De Morgan} \\
F & = & (x' + y)wz + wx + wy' + \underline{w'y' + w'w} + x'w & \textit{per distributiva} \\
F & = & (x' + y)wz + wx + wy' + w'y' + \underline{0} + x'w & \textit{per inverso} \\
F & = & (x' + y)wz + wx + wy' + w'y' + x'w & \textit{per elemento neutro} \\
F & = & \underline{x'wz + ywz} + wx + wy' + w'y' + x'w & \textit{per distributiva} \\
F & = & ywz + wx + wy' + w'y' + \underline{x'w} & \textit{per assorbimento su } x'wz + x'w \\
F & = & ywz + \underline{w(x + x')} + wy' + w'y' & \textit{per distributiva} \\
F & = & ywz + w\underline{(1)} + wy' + w'y' & \textit{per inverso} \\
F & = & ywz + \underline{w} + wy' + w'y' & \textit{per elemento neutro} \\
F & = & ywz + w + \underline{y'(w + w')} & \textit{per distributiva} \\
F & = & ywz + w + y'\underline{(1)} & \textit{per inverso} \\
F & = & ywz + w + \underline{y'} & \textit{per elemento neutro} \\
F & = & w + \underline{y'} & \textit{per assorbimento su } ywz + w
\end{array}
$$

∎

Esercizio 1.3 (TdE 2002/2003) - Data la funzione logica:

$$F = ((x' + y)'w + xy')z + (z'(x + y) + (x + y + yw)')$$

applicare le regole dell'algebra per semplificare la forma riducendo il numero dei prodotti ed il numero dei letterali.

Soluzione

$$
\begin{array}{lcll}
F & = & ((x' + y)'w + xy')z + (z'(x + y) + (x + \underline{y})') & \textit{per assorbimento su } y + yw \\
F & = & ((x' + y)'w + xy')z + \underline{z'} + (x + y)' & \textit{per } a'b + a = b + a \\
F & = & ((x' + y)'w + xy')z + z' + \underline{x'y'} & \textit{per De Morgan} \\
F & = & (\underline{xy'}w + xy')z + z' + x'y' & \textit{per De Morgan} \\
F & = & \underline{xy'}z + z' + x'y' & \textit{per assorbimento su } xy'w + xy' \\
F & = & \underline{xy'} + z' + x'y' & \textit{per } a'b + a = b + a \\
F & = & z' + \underline{(x + x')y'} & \textit{per distributiva} \\
F & = & z' + \underline{(1)}y' & \textit{per inverso} \\
F & = & z' + \underline{y'} & \textit{per elemento neutro}
\end{array}
$$

∎

2 Minimizzazione di espressioni logiche con le mappe di Karnaugh

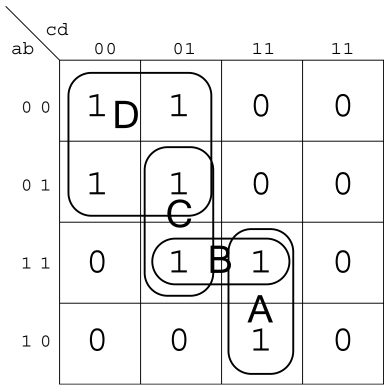
Esercizio 2.1 - Data la seguente funzione F ad una uscita, completamente specificata:

$$\mathsf{ON_{set}} = \{\mathsf{m_0}, \mathsf{m_1}, \mathsf{m_4}, \mathsf{m_5}, \mathsf{m_{13}}, \mathsf{m_{15}}, \mathsf{m_{11}}\}$$

1. Sulla mappa di Karnaugh individuare gli implicanti primi riportandone la forma algebrica e separando gli implicanti primi da quelli primi essenziali.
2. Ricavare una forma minima scegliendo una opportuna copertura della funzione sulla mappa e indicando se è unica (si assuma come funzione di costo il numero di letterali).
3. Calcolare il costo della forma minima trovata al punto precedente

Soluzione

Disegnamo la mappa di Karnaugh della funzione



1. Gli implicanti primi sono quelli indicati in figura che chiameremo rispettivamente:

   A = acd (essenziale perchè l'unico a coprire m_{11}(1011))
   B = abd
   C = bc'd
   D = a'c' (essenziale perchè l'unico a coprire m_0(0000), m_1(0001), m_4(0100))

2. Esistono due coperture minime a cardinalita' minima equivalenti per costo.

$$\mathsf{F} = \mathsf{A} + \mathsf{B} + \mathsf{D}$$

$$\mathsf{F} = \mathsf{A} + \mathsf{C} + \mathsf{D}$$

3. Il costo della/e funzione/i minima/e è

$$\mathsf{Costo} = 3 + 2 + 2 = 8$$

Esercizio 2.2 - Data la seguente funzione F ad una uscita, non completamente specificata:

$$\mathsf{ON_{set}} = \{\mathsf{0001}, \mathsf{0010}, \mathsf{0110}, \mathsf{1110}, \mathsf{1000}, \mathsf{1001}, \mathsf{1010}\} = \{m_1, m_2, m_6, m_{14}, m_8, m_9, m_{10}\}$$

$$\mathsf{DC_{set}} = \{\mathsf{0111}, \mathsf{1101}, \mathsf{1111}\} = \{m_7, m_{13}, m_{15}\}$$

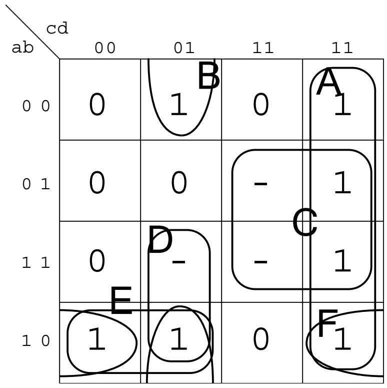
1. Sulla mappa di Karnaugh individuare gli implicanti primi riportandone la forma algebrica e separando gli implicanti primi da quelli primi essenziali.

2. Ricavare una forma minima scegliendo una opportuna copertura della funzione sulla mappa e indicando se è unica (si assuma come funzione di costo il numero di letterali).

3. Calcolare il costo della forma minima trovata al punto precedente

Soluzione

Disegnamo la mappa di Karnaugh della funzione:



1. Gli implicanti primi sono quelli indicati in figura che chiameremo rispettivamente:

   A = cd' (essenziale perchè l'unico a coprire m_2(0010))

   B = b'c'd (essenziale perchè l'unico a coprire m_1(0001))

   C = bc

   D = ac'd

   E = ab'c'

   F = ab'd'

2. Esistono due coperture minime a cardinalita' minima equivalenti per costo.

$$\mathsf{F = A + B + E}$$

$$\mathsf{F = A + B + F}$$

3. Il costo della/e funzione/i minima/e è

$$\mathsf{Costo} = 2 + 3 + 3 = 8$$

∎

3 Minimizzazione con il metodo di Quine-McCluskey per funzioni a singola/multipla uscita completamente/non completamente specificate

Esercizio 3.1 (Singola uscita) - Data la seguente funzione completamente specificata:

$$f_1 = \{m_0, m_1, m_2, m_4, m_5, m_9, m_{10}, m_{11}, m_{13}, m_{15}\}$$

1. Calcolare con il metodo di Quine-McCluskey gli implicanti primi.
2. Calcolare con il metodo di Quine McCluskey una copertura minima usando come funzione di costo il numero di letterali.

Soluzione
Fase 1 Calcolo degli implicanti primi

m_0	0000	✓
m_1	0001	✓
m_2	0010	✓
m_4	0100	✓
m_5	0101	✓
m_9	1001	✓
m_{10}	1010	✓
m_{11}	1011	✓
m_{13}	1101	✓
m_{15}	1111	✓

m_0m_1	000-	✓
m_0m_2	00-0	A
m_0m_4	0-00	✓
m_1m_5	0-01	✓
m_1m_9	-001	✓
m_2m_{10}	-010	B
m_4m_5	010-	✓
m_5m_{13}	-101	✓
m_9m_{11}	10-1	✓
m_9m_{13}	1-01	✓
$m_{10}m_{11}$	101-	C
$m_{11}m_{15}$	1-11	✓
$m_{13}m_{15}$	11-1	✓

$m_0m_1m_4m_5$	0-0-	D
$m_1m_5m_9m_{13}$	-01	E
$m_9m_{11}m_{13}m_{15}$	1-1	F

Fase 2 Calcolo della copertura minima

	m_0	m_1	m_2	m_4	m_5	m_9	m_{10}	m_{11}	m_{13}	m_{15}	Costo	
A	x		x								3	
B			x				x				3	
C							x	x			3	
D	x	x		x	x						2	✓
E		x			x	x			x		2	
F						x		x	x	x	2	✓

Gli implicanti D e F sono essenziali e vengono quindi inclusi nell'insieme di copertura. Di conseguenza le righe e le colonne (mintermini coperti) vengono eliminate dalla tabella di copertura. La tabella ridotta risulta essere:

	m_2	m_{10}	Costo	
A	x		3	
B	x	x	3	✓
C		x	3	

Applicando il criterio di dominanza di riga si possono semplificare le righe relative agli implicanti A e C(NB. I costi sono identici).

La funzione minima risulta essere:

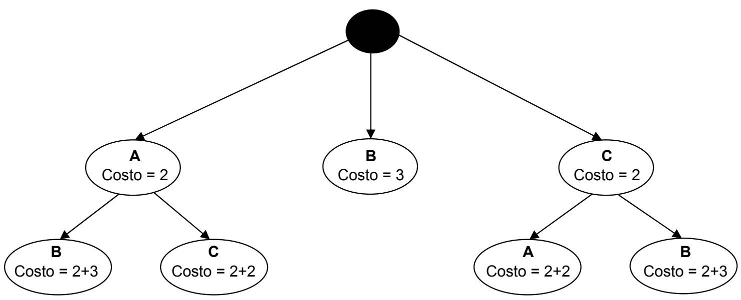
$$F = B + D + F$$

il cui costo è

$$\text{Costo} = 2 + 2 + 3 = 7$$

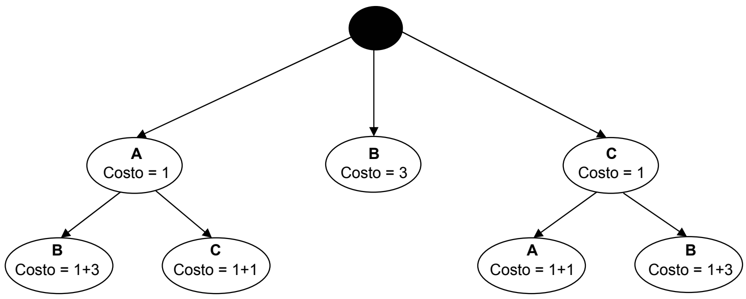
Osservazione Non sempre il calcolo della copertura minima si ottiene con la semplice applicazione dei criteri di dominanza e di essenzialità.
Se ad esempio i costi degli implicanti A, B, C sono rispettivamente 2, 3 e 2, allora, non è possibile applicare il criterio di dominanza di riga per scegliere B, ma si deve ricorrere all'applicazione dell'algoritmo di Branch&Bound.
L'albero delle decisioni si costruisce aggiungendo il primo implicante alla soluzione parziale (in questo caso A) ed eliminando dalla tabella la riga e le colonne interessate. Sucessivamente si applicano i criteri di essenzialità e dominanza e se la tabella non si riduce ad una sola riga si ripete il procedimento dall'inizio. Una volta determinata una prima soluzione completa, si risale lungo l'albero fino a raggiungere la penultima scelta fatta e da li si riparte esplorando le altre alternative.

Il costo associato ad ogni nodo è pari a quello del nodo superiore più quello dell'implicante scelto. L'albero delle decisioni ottenuto partendo dalla tabella di copertura con i nuovi costi è:



La soluzione ottima è quella con l'implicante B.

Nel caso in cui i costi dei tre implicanti A, B, C fossero stati rispettivamente 1, 3, 1, si sarebbe ottenuto il seguente albero:



In questo caso la soluzione ottima è costituita da A+C.

∎

Esercizio 3.2 (Multipla uscita) - Data la seguente funzione completamente specificata a due uscite:

$$f_1 = \{m_2, m_4, m_5, m_7\}$$

$$f_2 = \{m_3, m_4, m_5, m_7\}$$

1. Calcolare con il metodo di Quine-McCluskey gli implicanti primi.
2. Calcolare con il metodo di Quine McCluskey una copertura minima usando come funzione di costo il numero di implicanti della copertura.

Soluzione

Fase 1 Calcolo degli implicanti primi

m_2	010	10	A
m_4	100	11	✓
m_3	011	01	✓
m_5	101	11	✓
m_7	111	11	✓

m_4m_5	10-	11	B
m_3m_7	-11	01	C
m_5m_7	1-1	11	D

Osservazione Nel caso di funzioni a più uscite se un implicante primo viene incluso nella soluzione parziale di una funzione, allora il suo costo diventa uguale a zero. Questo perchè non ci sarà alcun costo aggiuntivo da parte delle altre funzioni nell'utilizzare questo implicante già preso.

FASE 2 Calcolo della copertura minima
La tabella di copertura risulta essere:

	m_2	m_4	m_5	m_7		m_3	m_4	m_5	m_7		Costo
A	X				✓						3
B		X	X		✓		X	X		✓	2
C						X			X	✓	2
D			X	X	✓			X	X		2

La forma minima risulta essere:

$$f_1 = A + B + D$$

$$f_2 = B + C$$

∎

Esercizio 3.3 - Data la seguente funzione non completamente specificata a due uscite:

$$f_1 : ON_{set} = \{m_0, m_1, m_8, m_9, m_{10}, m_{11}, m_{16}, m_{17}, m_{26}\}$$

$$f_1 : DC_{set} = \{m_2, m_5\}$$

$$f_2 : ON_{set} = \{m_0, m_1, m_5, m_8, m_{26}, m_{28}\}$$

$$f_2 : DC_{set} = \{m_9\}$$

1. Calcolare con il metodo di Quine-McCluskey gli implicanti primi.
2. Calcolare con il metodo di Quine-McCluskey una copertura minima usando come funzione di costo il numero di implicanti della copertura.

SOLUZIONE

Osservazione Questo esercizio è stato risolto seguendo il procedimento semplificato presentato nei lucidi relativi alle lezioni, dove non viene usato nell'identificatore il simbolo di `don't care` per caratterizzare gli implicanti.
Il procedimento semplificato (allo scopo di non appesantire ulteriormente la notazione di Quine-McCluskey) è equivalente nella maggior parte dei casi a quello usato dall'Ing. Palermo ad esercitazione. In alcuni casi con il procedimento semplificato vengono aggiunti degli implicanti in più che non sono primi. Nel caso particolare di questo esercizio il risultato della prima fase è identico indipendentemente dal procedimento scelto.

m_0	00000	11	✓
m_1	00001	11	✓
m_2	00010	10	✓
m_8	01000	11	✓
m_{16}	10000	10	✓
m_5	00101	11	✓
m_9	01001	11	✓
m_{10}	01010	10	✓
m_7	10001	10	✓
m_{11}	01011	10	✓
m_{26}	11010	11	A
m_{28}	11101	01	B

m_0m_1	0000-	11	✓
m_0m_2	000-0	10	✓
m_0m_1	0-000	11	✓
m_0m_{16}	-0000	10	✓
m_1m_5	00-01	11	C
m_1m_9	0-001	11	✓
m_1m_{17}	-0001	10	✓
m_2m_{10}	0-010	10	✓
m_8m_9	0100-	11	✓
m_8m_{10}	010-0	10	✓
$m_{16}m_{17}$	1000-	10	✓
m_9m_{11}	010-1	10	✓
$m_{10}m_{11}$	01011-	10	✓
$m_{10}m_{26}$	-1010	10	D

$m_0m_1m_8m_9$	0-00-	11	E
$m_0m_1m_{16}m_{17}$	-00-	10	F
$m_2m_2m_8m_{10}$	0-0-0	10	G
$m_8m_9m_{10}m_{11}$	010-	10	H

In conclusione:

- Gli implicanti primi identificati nella prima iterazione sono:

  $A =$ m_{26} 11

  $B =$ m_{28} 01

- quelli nella seconda:

  $C =$ m_1m_5 11

  $D =$ $m_{10}m_{20}$ 10

- e quelli nella terza:

  $E =$ $m_0m_1m_8m_9$ 11

  $F =$ $m_0m_1m_{16}m_{17}$ 10

  $G =$ $m_0m_2m_8m_{10}$ 10

  $H =$ $m_8m_9m_{10}m_{11}$ 10

La tabella di copertura ottenuta è:

	m_0	m_1	m_8	m_9	m_{10}	m_{11}	m_{16}	m_{17}	m_{26}		m_0	m_1	m_5	m_8	m_{26}	m_{28}		Costo
A									X						X		✓	1
B																X	✓	1
C		X										X	X				✓	1
D					X				X									1
E	X	X	X	X							X	X		X			✓	1
F	X	X					X	X		✓								1
G	X		X		X													1
H			X	X	X	X				✓								1

Riducendo la tabella e modificando i costi degli implicanti inseriti nella soluzione, si ottiene la seguente tabella di copertura ridotta:

	m_0	m_1	m_8	m_9	m_{10}	m_{11}	m_{16}	m_{17}	m_{26}	Costo
A									X	0
D									X	1

Si ricava quindi che A domina D. La forma minima della funzione a due uscite risulta essere:

$f_1 = \mathsf{F} + \mathsf{H} + \mathsf{A}$

$f_2 = \mathsf{B} + \mathsf{A} + \mathsf{B} + \mathsf{C}$

∎

4 Modello algebrico e reti multilivello

Esercizio 4.1 - Utilizzando il modello algebrico, si consideri la rete logica definita dalle seguenti espressioni:

p = bc + bde' + a'c' + ade' + a'e

q = dc' + b'ce + a'e + de + a'c'

r = eb' + adb + c'b'

s = a' + d

x = p

y = q

z = r

u = s

dove a, b, c, d, e sono gli ingressi e x, y, z, u sono le uscite.

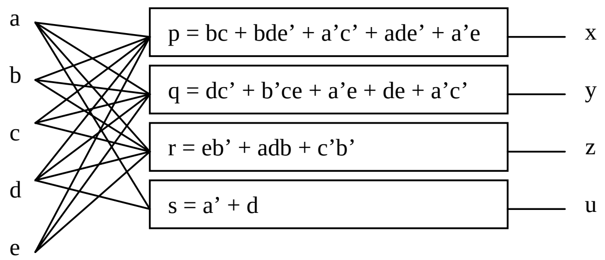
1. Si disegni il grafo associato alla rete logica e si calcoli il costo associato in termini di letterali Costo(RETE).
2. Si eseguano in sequenza le trasformazioni sotto elencante. Dopo ogni trasformazione è necessario verificare che il costo associato (letterali) alla rete trasformata non sia peggiore di quello prima della trasformazione. Se il costo risulta peggiore, la trasformazione non viene considerata e si passa alla successiva. In caso contrario la trasformazione viene considerata efficace. Nota: il calcolo del costo ad ogni passo deve essere effettuato con espressioni nella forma SOP.
   (a) Sostituire s in q
   (b) Estrarre la sottoespressione j comune a q, r e p
   (c) Decomporre p introducendo un nuovo vertice k

Soluzione
Direttamente dalle espressioni ricaviamo subito

1. Calcoliamo il costo della rete iniziale

	espressione	**costo**
p =	$bc + bde' + a'c' + ade' + a'e$	12
q =	$dc' + b'ce + a'e + de + a'c'$	11
r =	$eb' + adb + c'b'$	7
s =	$a' + d$	2
	Costo(RETE)	32

Il grafo associato alla rete iniziale è:

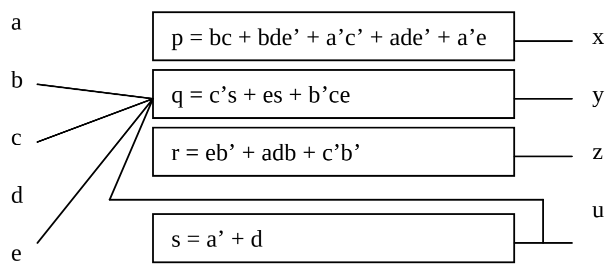


2. Eseguiamo in sequenza le operazioni richieste

   (a) Sostituire s in q.
   Iniziamo con una fattorizzazione di q per cercare eventuali *occorrenze* di s

	espressione		**costo**
$\mathsf{q} =$	$\mathsf{c'\underline{(d+a')} + e\underline{(d+a')} + b'ce}$	$\mathsf{c'\underline{s} + e\underline{s} + b'ce}$	7
	Costo(RETE)		$32 \Rightarrow 28$

   Il $\mathsf{Costo(q)} = 7$ dunque il $\mathsf{Costo(RETE)} = 28 < 32$ per cui accettiamo.
   Il grafo associato alla rete trasformata è:



Si mostrano solo le connessioni relative a Q

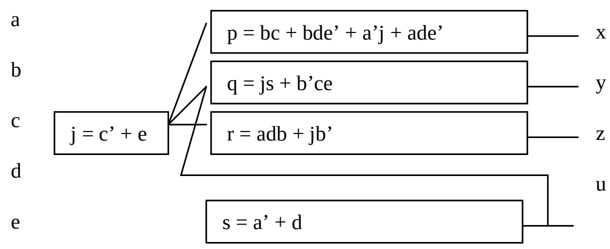
   (b) Estrarre la sottoespressione j comune a $\mathsf{q, r, p}$.
   Anche qui, fattorizziamo le tre espressioni di $\mathsf{q, r, p}$ per trovare eventuali fattori comuni

$$\left[\begin{array}{l} \mathsf{r = adb + b'(c'+e)} \\ \mathsf{q = s(c'+e) + b'ce} \\ \mathsf{p = bc + bde' + a'(c'+e) + ade'} \\ \mathsf{j = f(a,b,c,d,e)\ (dadeterminare)} \end{array}\right] \rightsquigarrow \left[\begin{array}{l} \mathsf{r = adb + b'\underline{j}} \\ \mathsf{q = s\underline{j} + b'ce} \\ \mathsf{p = bc + bde' + a'\underline{j} + ade'} \\ \mathsf{j = c' + e} \end{array}\right]$$

   Riassumendo:

	espressione	costo
$r =$	$adb + b'j$	5
$q =$	$sj + b'ce$	5
$p =$	$bc + bde' + a'j + ade'$	10
$s =$	$a' + d$	2
$j =$	$c' + e$	2
	Costo(RETE)	$28 \Rightarrow 24$

Il $\mathsf{Costo(RETE)} = 24 < 28$ per cui accettiamo.
Il grafo associato alla rete trasformata è:



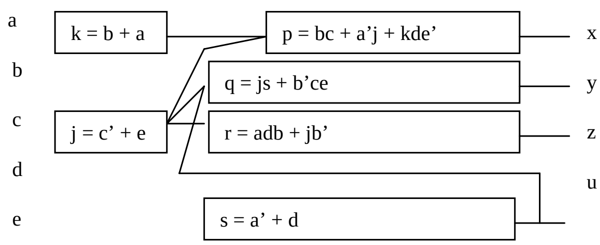
Si mostrano solo le connessioni che subiscono cambiamento

(c) Decomporre p introducendo un nuovo vertice k.
Procediamo analogamente al passo precedente

$$p = bc + bde' + a'j + ade = bc + a'j + de'\underline{(b + a)} \Leftrightarrow$$

	espressione	costo
$p =$	$bc + a'j + de'k$	7
$k =$	$b + a$	2
	Costo(RETE)	$24 \Rightarrow 23$

Il $\mathsf{Costo(RETE)} = 23 < 24$ per cui accettiamo.
Il grafo associato alla rete trasformata è:



Si mostrano solo le connessioni che subiscono cambiamento

■

Esercizio 4.2 - Si consideri la seguente funzione

$$\mathsf{f(a,b,c,d) = a'b'cd' + a'b'cd + a'bc' + a'bc'd + abcd' + abc'd + ab'cd' + ab'cd}$$

decomporre f in maniera disgiuntiva rispetto alle variabili $\mathsf{a, b}$ e operando una trasformazione locale calcolarne la variazione di costo.

SOLUZIONE
Per il *teorema di espansione di Shannon* dobbiamo pervenire ad una funzione in questa forma

$$\mathsf{f(a,b,c,d) = a'b' \cdot f(0,0,c,d) + a' \cdot bf(0,1,c,d) + ab' \cdot f(1,0,c,d) + ab \cdot f(1,1,c,d)}$$

il cui $\mathsf{Costo_{prima}(f) = 31}$

$$\begin{aligned}
f(0,0,c,d) &= cd' + cd = c(d' + d) = c \\
f(0,1,c,d) &= c' + c'd = c' \\
f(1,0,c,d) &= cd' + cd = c(d' + d) = c \\
f(1,1,c,d) &= cd' + c'd
\end{aligned}$$

Cacoliamo la nuova espressione associata al nodo e la variazione di costo

$$\mathsf{f(a,b,c,d) = a'b'c' + a'bc' + ab'c + abcd' + abc'd}$$

con $\mathsf{Costo_{dopo}(f) = 17}$. Pertanto $\Delta Costo = 14$

Osservazione La trasformazione effettuata è di tipo locale: in questo esercizio l'obiettivo è quello di ridurre i letterali della funzione associata ad un nodo della rete multilivello. E' possibile utilizzare la decomposizione disgiuntiva come trasformazione globale (v. lezione), cioè come strumento per decomporre una funzione in termini più piccoli che possono essere usati per ridurre altri nodi della rete.

■

5 Aritmetica dei calcolatori

Esercizio 5.1 (Moltiplicazioni) - Eseguire le seguenti moltiplicazioni utilizzando la rappresentazione in complemento a due

1. $(21)_{10} \cdot (21)_{10}$
2. $(-21)_{10} \cdot (21)_{10}$
3. $(21)_{10} \cdot (-21)_{10}$
4. $(-21)_{10} \cdot (-21)_{10}$

Soluzione

1. Convertiamo in complemento a due: $(21)_{10} \mapsto (010101)_2$. Poiché $(21)_{10}$ é positivo la rappresentazione in complemento a due é la stessa del binario naturale ma il numero di bit utilizzato é 6 e non 5. Il risultato della moltiplicazione é su $6+6$ bit.

$$
\begin{array}{ccccccccccccc}
 & & & & & & 0 & 1 & 0 & 1 & 0 & 1 & \cdot \\
 & & & & & & 0 & 1 & 0 & 1 & 0 & 1 & = \\
\hline
 & & & & 1 & & 1 & & 1 & & & & \\
 & & & & & & 0 & 1 & 0 & 1 & 0 & 1 & \\
 & & & & & 0 & 0 & 0 & 0 & 0 & 0 & & \\
 & & & & 0 & 1 & 0 & 1 & 0 & 1 & & & \\
 & & & 0 & 0 & 0 & 0 & 0 & 0 & & & & \\
 & & 0 & 1 & 0 & 1 & 0 & 1 & & & & & \\
 & 0 & 0 & 0 & 0 & 0 & 0 & & & & & & \\
\hline
0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 &
\end{array}
$$

2. Convertiamo in complemento a due i due fattori: $(21)_{10} \mapsto (010101)_2$, $(-21)_{10} \mapsto (101011)_2 \mapsto (-101011)_2$. Il risultato della moltiplicazione é su $6+6$ bit.

$$
\begin{array}{cccccccccccc}
 & & & & & \text{-1} & 0 & 1 & 0 & 1 & 1 & \cdot \\
 & & & & & 0 & 1 & 0 & 1 & 0 & 1 & = \\
\hline
 & & & & & \text{-1} & 0 & 1 & 0 & 1 & 1 & \\
 & & & & 0 & 0 & 0 & 0 & 0 & 0 & & \\
 & & & \text{-1} & 0 & 1 & 0 & 1 & 1 & & & \\
 & & 0 & 0 & 0 & 0 & 0 & 0 & & & & \\
 & \text{-1} & 0 & 1 & 0 & 1 & 1 & & & & & \\
0 & 0 & 0 & 0 & 0 & 0 & & & & & &
\end{array}
$$

Che si decompone nelle due matrici diagonali:

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

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

e che risulta nella sottrazione:

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

3. In questo caso si ha:

4. In questo caso si ha:

∎

Esercizio 5.2 (Algoritmo di Booth per le moltiplicazioni) - Calcolare in complemento a due il prodotto $(33)_{10} \cdot (-3)_{10}$ usando l'algoritmo di Booth.

Soluzione

Si codificano moltiplicando e moltiplicatore in complemento a due[1], quindi prima di procedere con la normale moltiplicazione si codifica secondo Booth il moltiplicatore.

Si ha:

$$\texttt{1111101} \mapsto \texttt{0000}-\texttt{11}-\texttt{1}$$

Si esegue quindi la moltiplicazione, il cui risultato sará di 14 bit, tenendo conto del fatto che se si moltiplica per 0 o per 1 si esegue la normale moltiplicazione, se invece si moltiplica per -1 si deve utilizzare l'opposto del moltiplicatore.

[1] Si faccia attenzione all'estensione del segno durante la moltiplicazione

							0	1	0	0	0	0	1
							0	0	0	0	-1	1	-1
	1	**1**	**1**	**1**	**1**	**1**	1	0	1	1	1	1	1
	0	**0**	**0**	**0**	**0**	0	1	0	0	0	0	1	
	1	**1**	**1**	**1**	1	0	1	1	1	1	1		
	0	**0**	**0**	0	0	0	0	0	0	0			
	0	**0**	0	0	0	0	0	0	0				
	0	0	0	0	0	0	0	0					
	0	0	0	0	0	0	0						
1	1	1	1	1	1	1	0	0	1	1	1	0	1

Si puó subito notare che il risultato é coerente: il primo bit vale 1, ed il risultato deve essere negativo. Per ottenere il valore decimale corrispondente, basta riportare il risultato in forma binaria, ottenendo 00000001100011, che in decimale vale 99, quindi il risultato 'e -99

∎

6 Sintesi e ottimizzazione di FSM completamente specificate

Esercizio 6.1 - Data la tabella degli stati:

	0	1
Reset	G/00	G/00
B	G/00	D/01
C	-/–	Reset/11
D	C/10	D/11
E	G/00	F/11
F	F/10	E/10
G	Reset/01	F/11

Determinare l'insieme degli stati raggiungibili.

Soluzione
Per ogni stato, si costruisce ricorsivamente l'insieme degli stati raggiungibili eliminando gli eventuali stati duplicati piú interni e gli stati giá esaminati. Si prosegue finchè successivi passaggi non aggiungono nuovi stati alla sequenza giá esaminata.
Gli stati *non* presenti nella sequenza *non* sono raggiungibili.

$$
\begin{aligned}
&\text{Reset}\{\text{G},\underline{\text{G}}\} = \text{Reset}\{\text{G}\}\\
&\text{Reset}\{\text{G}\{\underline{\text{Reset}}, \text{F}\}\} = \text{Reset}\{\text{G}\{\text{F}\}\}\\
&\text{Reset}\{\text{G}\{\text{F}\{\underline{\text{F}},\text{E}\}\}\} = \text{Reset}\{\text{G}\{\text{F}\{\text{E}\}\}\}\\
&\text{Reset}\{\text{G}\{\text{F}\{\text{E}\{\underline{\text{G},\text{F}}\}\}\}\} = \text{Reset}\{\text{G}\{\text{F}\{\text{E}\{.\}\}\}\}
\end{aligned}
$$

∎

Esercizio 6.2 - Data la seguente tabella degli stati di una FSM sincrona di Moore, minimizzarla.

	0	1	OUT
A	A	C	0
B	F	A	0
C	D	E	1
D	C	E	1
E	B	F	1
F	F	D	0

Soluzione
Costruendo la tabella (triangolare) delle implicazioni si ottiene:

- **primo passo**

B	AF CA				
C	X	X			
D	X	X	$\sim$		
E	X	X	DB EF	CB EF	
F	CD	AD	X	X	X
	A	B	C	D	E

- **secondo passo**

B	X				
C	X	X			
D	X	X	$\sim$		
E	X	X	X	X	
F	$\sim$	X	X	X	X
	A	B	C	D	E

Analizzando la tabella e propagando le non equivalenze si ottiene F $\sim$ A, C $\sim$ D. Si hanno quindi le seguenti classi di equivalenza

$$\begin{aligned} \alpha &= \{\mathrm{A}, \mathrm{F}\} \\ \beta &= \{\mathrm{B}\} \\ \gamma &= \{\mathrm{C}, \mathrm{D}\} \\ \delta &= \{\mathrm{E}\} \end{aligned}$$

da cui si ricostruisce la tabella degli stati *ridotta*

	0	1	OUT
α	α	γ	0
β	α	α	0
γ	γ	δ	1
δ	β	α	1

Nota Supponendo di scegliere α come stato di **Reset** o comunque come stato iniziale, non è possibile eliminare alcuno stato perchè non ne esistono di irraggiungibili. Infatti

$$\alpha\{\underline{\alpha}, \gamma\} = \alpha\{\gamma\}$$
$$\alpha\{\gamma\{\underline{\gamma}, \delta\}\} = \alpha\{\gamma\{\delta\}\}$$
$$\alpha\{\gamma\{\delta\{\beta, \underline{\alpha}\}\}\} = \alpha\{\gamma\{\delta\{\beta\}\}\}$$
$$\alpha\{\gamma\{\delta\{\beta\{\underline{\alpha}, \underline{\alpha}\}\}\}\} = \alpha\{\gamma\{\delta\{\beta\{.\}\}\}\}$$

▮

Esercizio 6.3 - Data la seguente specifica comportamentale in linguaggio naturale di una FSM: *la macchina asserisce un 1 in uscita ogni qual volta riconosce la seguente sequenza in ingresso*, dopo di che torna a 0

$$\texttt{1(00)}^{+}\texttt{1(00)}^{+}\texttt{1}$$

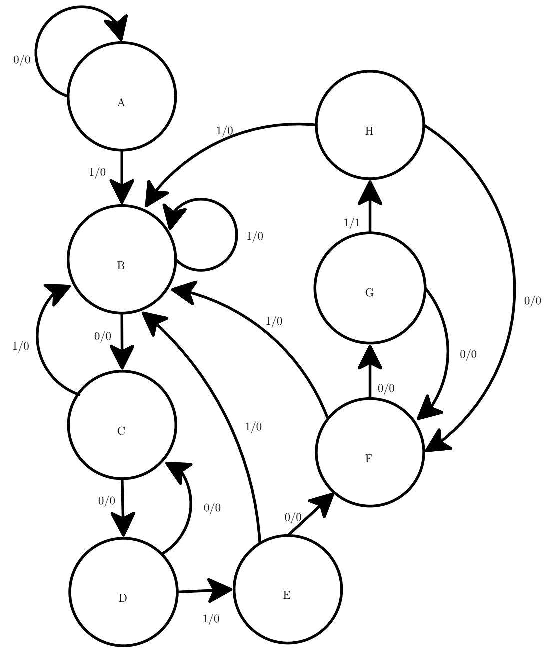
dove la notazione ($)^{+}$ indica una o piú ripetizioni della sequenza contenuta tra parentesi. Si osserva inoltre che le sequenze possono concatenarsi. Si chiede di:

1. definire il grafo delle transizioni che ne descriva il comportamento. La macchina a stati finiti e' completamente o non completamente specificata?

2. minimizzare il numero degli stati della macchina.

SOLUZIONE
Si costruisce la macchina a stati finiti di *Mealy*

1. La macchina a stati risulta definita dal seguente diagramma dove come stato iniziale si é scelto quello di nessuna sequenza riconosciuta.



Non esistono stati che possono generare una macchina non completamente specificata in quanto per ogni nodo del grafo ci sono almeno due archi uscenti e l'uscita é sempre specificata.

2. La tabella degli stati risulta essere

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

Costruendo la tabella (triangolare) delle implicazioni si ottiene:

- **primo passo**

B	AC						
C	AD	CD					
D	AC BE	BE	BE				
E	AF	CF	DF	CF EB			
F	AG	CG	DG	CG EB	GF		
G	X	X	X	X	X	X	
H	AF	CF	CF EB	CF EB	$\sim$	GF	X
	A	B	C	D	E	F	G

- **secondo passo**

B	X						
C	X	X					
D	X	X	X				
E	X	X	X	X			
F	X	X	X	X	X		
G	X	X	X	X	X	X	
H	X	X	X	X	$\sim$	X	X
	A	B	C	D	E	F	G

L'unica classe di equivalenza é E $\sim$ H.

∎

Esercizio 6.4 - Data la seguente tabella degli stati di una macchina a stati completamente specificata:

	00	01	11	10	Z
ST0	ST1	ST3	ST7	ST4	00
ST1	ST7	ST2	ST4	ST5	10
ST2	ST3	ST4	ST1	ST0	11
ST3	ST0	ST4	ST2	ST1	10
ST4	ST5	ST6	ST3	ST0	11
ST5	ST7	ST6	ST6	ST3	10
ST6	ST1	ST6	ST5	ST0	11
ST7	ST5	ST3	ST0	ST2	00

1. Eseguire la minimizzazione degli stati e realizzare la tabella degli stati della FSM minima equivalente.

Soluzione

Costruendo la tabella (triangolare) delle implicazioni si ottiene:

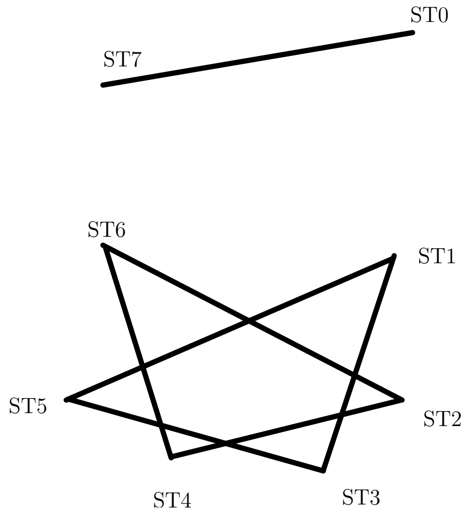
- **primo passo**

ST1	X						
ST2	X	X					
ST3	X	ST0,ST7 ST4,ST2 ST5,ST1	X				
ST4	X	X	ST5,ST3 ST4,ST6 ST1,ST3	X			
ST5	X	ST2,ST6 ST4,ST6 ST5,ST3	X	ST0,ST7 ST4,ST6 ST2,ST6 ST1,ST3	X		
ST6	X	X	ST1,ST3 ST4,ST6 ST1,ST5	X	ST1,ST5 ST5,ST3	X	
ST7	ST1,ST5 ST4,ST2	X	X	X	X	X	X
	ST0	ST1	ST2	ST3	ST4	ST5	ST6

- **secondo passo**

  Non e' possibile propagare nessuna non equivalenza quindi le coppie condizionate sono tutte equivalenti.

Grafo di equivalenza:



Le classi di equivalenza sono:
A = ST0, ST7
B = ST1, ST3, ST5
C = ST2, ST4, ST6

Tabella degli stati della FSM minima:

	00	01	11	10	Z
A	B	B	A	C	00
B	A	C	C	B	10
C	B	C	B	A	11

∎

Esercizio 6.5 (pre compitino) - Data la seguente tabella degli stati, in cui é specificato lo stato di reset (RST)

1. si determini la macchina minima equivalente tracciando la nuova tabella degli stati e il nuovo diagramma degli stati,
2. esiste una macchina piú piccola rispetto a quella identificata al passo precedente (giustificare la risposta)?
3. si determini un buon assegnamento degli stati della macchina minima (secondo i criteri visti a lezione), nell'ipotesi di utilizzo di bistabili D

ST \ i	0	1
RST	A/00	C/01
A	RST/01	B/00
B	A/00	D/01
C	D/11	C/10
D	C/11	D/10
E	B/11	C/00
F	A/00	E/01
G	RST/11	D/00

Soluzione

Poiché per la macchina é specificato lo stato di RESET si applica un'analisi di raggiungibilitá per eliminare eventuali stati non raggiungibili da questo. L'analisi puó essere fatta esaminando il diagramma degli stati o esaminando le raggiungibilitá a partire dalla tabella degli stati. Il risultato é il seguente:

$$\begin{aligned}
&\mathrm{RST}\{\mathrm{A,C}\} = \\
&\mathrm{RST}\ \{\mathrm{A}\ \{\underline{\mathrm{RST}},\mathrm{B}\},\ \mathrm{C}\{\mathrm{D},\underline{\mathrm{C}}\}\} = \mathrm{RST}\ \{\mathrm{A}\ \{\mathrm{B}\},\mathrm{C}\{\mathrm{D}\}\} = \\
&\mathrm{RST}\ \{\mathrm{A}\ \{\mathrm{B}\{\underline{\mathrm{A}},\underline{\mathrm{D}}\}\},\mathrm{C}\{\mathrm{D}\{\underline{\mathrm{C}},\underline{\mathrm{D}}\}\}\} = \mathrm{RST}\ \{\mathrm{A}\ \{\mathrm{B}\{.\}\},\mathrm{C}\{\mathrm{D}\{.\}\}\}
\end{aligned}$$

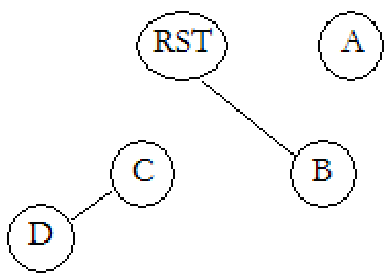
All'ultimo passo l'insieme degli stati non é cambiato rispetto al passo precedente, quindi l'analisi é terminata. L'insieme degli stati **raggiungibili** é quindi costituito da RST, A, B, C, D e gli altri stati possono essere eliminati dalla tabella iniziale.

La tabella dei soli stati raggiungibili viene esaminata per l'analisi di equivalenza (macchina completamente specificata). La tabella delle implicazioni é la seguente

A	X			
B	CD	X		
C	X	X	X	
D	X	X	X	~
	RST	A	B	C

Analizzando la tabella si verifica che non é possibile propagare le non equivalenze e quindi risulta che C $\sim$ D e RST $\sim$ B.

Tracciando il grafo delle equivalenze



si ricavano gli stati della macchina equivalente ridotta

classe di equivalenza (RST, B) = $\alpha^\star$ (questo stato nella macchina ridotta diventa lo stato di RESET, indicato con $\star$)

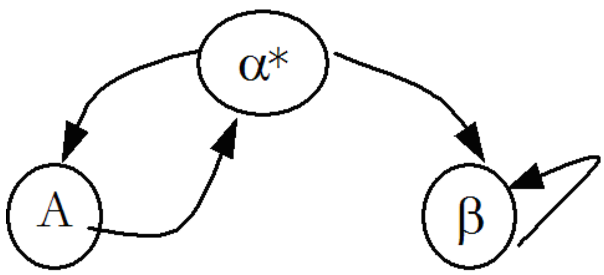
classe di equivalenza (C, D) = β

classe di equivalenza A = A

e la tabella degli stati relativa

ST \ i	0	1
$\alpha^\star$	A/00	β/01
A	$\alpha^\star$/01	$\alpha^\star$/00
β	β/11	β/10

Il diagramma degli stati della macchina minima equivalente risulta essere:



La macchina ottenuta é minima e unica perché deriva da condizioni di equivalenza e condizioni di irraggiungibilitá.

Nota: allo stesso risultato si sarebbe arrivati applicando prima l'analisi di equivalenza e quindi l'analisi di raggiungibilitá. In questo caso, gli stati non raggiungibili possono generare delle classi di equivalenza (che non comprendono stati raggiungibili) e l'analisi di raggiungibilitá puó essere effettuata a partire dalla macchina ridotta per equivalenza.

Assegnamento degli stati

La macchina minima é costituita da 3 stati, sono quindi necessari 2 bistabili (2 variabili di stato). Dalla tabella degli stati ridotta si determinano i seguenti vincoli di assegnamento:

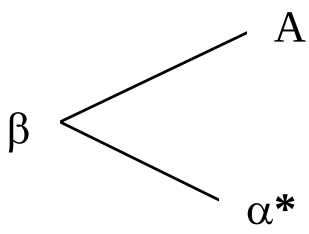
Stati che condividono lo stesso stato prossimo, per un certo ingresso (I criterio):

$$\alpha^\star, \beta$$

Stati prossimi con ingressi adiacenti (II criterio)

$$A, \beta$$

Il grafo delle adiacenze é il seguente ed é contenuto in un ipercubo, quindi tutte le adiacenze possono essere soddisfatte



Un possibile assegnamento é $\alpha^{\star} = 00$, $\beta = 01$, A $= 11$

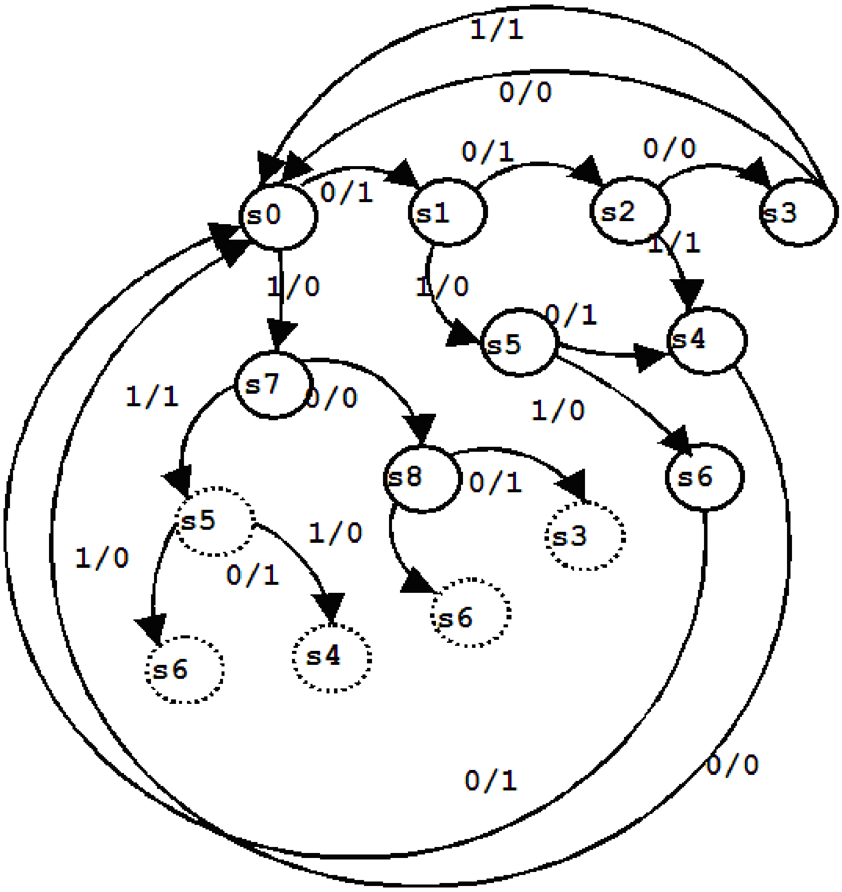
Esercizio 6.6 (pre compitino) - Data la seguente tabella di conversione dalla codifica BCD alla codifica ad eccesso tre. Si costruisca il diagramma della macchina a stati che riceve serialmente dal bit meno significativo a quello piú significativo il dato codificato in BCD e produca serialmente, sempre a partire dal bit meno significativo, il dato codificato ad eccesso tre.

1	0000	0011
6	0001	0100
4	0010	0101
9	0011	0110
3	0100	0111
8	0101	1000
5	0110	1001
10	0111	1010
2	1000	1011
7	1001	1100

SOLUZIONE

La macchina é una macchina di Mealy non completamente specificata.

La derivazione del diagramma degli stati é abbastanza complessa. Il diagramma degli stati viene tracciato secondo il principio di costruire un ramo per ogni sequenza di ingresso. Come stato iniziale si considera quello in cui si inizia a riconoscere una sequenza (lo stato iniziale é anche quello a cui ci si riporta in corrispondenza dell'ultimo ingresso di ogni sequenza). La stesura parte dal ramo associato al ricevimento della sequenza 0000. Le sequenze di ingresso vengono esaminate secondo un ordine che consente di sfruttare rami e sotto-rami giá inseriti.



Le codifiche in ingresso sono state considerate secondo il seguente ordine:

1)	0000	0011
2)	1000	1011
3)	0100	0111
4)	0010	0101
5)	0110	1001
6)	0001	0100
7)	1001	1100
8)	0101	1000
9)	0011	0110
10)	0111	1010

Si noti anche che, per ogni sequenza nel grafo, sono stati evidenziati tutti e 4 gli stati che la compongono, tratteggiando quelli giá presenti.

Il diagramma puó essere ulteriormente minimizzato considerando che S3 e S4 sono compatibili. Se viene associato un unico stato a S3 e S4 allora anche S8 é equivalente a S5.

∎

7 Criteri per l'assegnamento degli stati

Esercizio 7.1 (continua Esercizio 6.3) - Data la tabella degli stati seguente si chiede di procedere con l'assegnamento degli stati (si consiglia di applicare entrambi i due criteri principali).

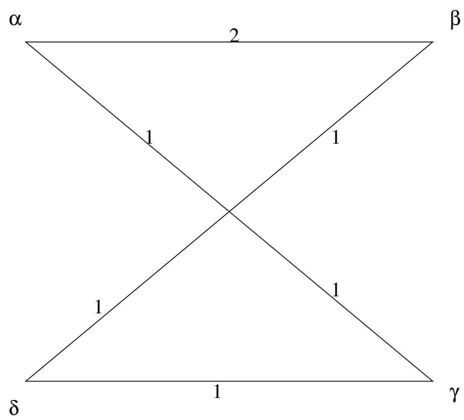
XY\I	0	1	OUT
α	α	γ	0
β	α	α	0
γ	γ	δ	1
δ	β	α	1

Soluzione

I risultati dell'applicazione dei due criteri sono riassunti nella tabella seguente:

primo criterio	secondo criterio
α, β	α, γ
β, δ	γ, δ
	β, α

Il grafo delle adiacenze risulta essere:



É possibile soddisfare tutti i vincoli di adiacenza poichè il grafo di adiacenza è contenuto in un ipercubo ed in particolare si ha che con X e Y variabili di stato l'assegnamento risulta essere:

$$\begin{array}{ccc} & & \text{x y} \\ \alpha & = & 00 \\ \beta & = & 01 \\ \delta & = & 11 \\ \gamma & = & 10 \end{array}$$

Ora si può generare la tabella delle transizioni mettendo in corrispondenza ogni stato con la sua codifica. Si ottiene

XY\I	0	1	OUT
00	00	10	0
01	00	00	0
11	01	00	1
10	10	11	1

Definita la tabella delle transizioni si effettua la sintesi delle reti combinatorie che implementano la funzione stato prossimo e la funzione di uscita ipotizzando flip-flop di tipo D.

- *Funzione d'uscita*

$$f_{OUT} = \mathsf{X}$$

- *Funzione stato prossimo relativa alla variabile X*

XY\I	0	1
00	0	1
01	0	0
11	0	0
10	1	1

$$\mathsf{X}^{\star} = \mathsf{XY}' + \mathsf{Y}'\mathsf{I}$$

- *Funzione stato prossimo relativa alla variabile Y*

XY\I	0	1
00	0	0
01	0	0
11	1	0
10	0	1

$$\mathsf{Y}^{\star} = \mathsf{XYI}' + \mathsf{XY}'\mathsf{I}$$

∎

Esercizio 7.2 (pre compitino) - Data la seguente tabella degli stati

St\I	0	1	Z
A	E	C	1
B	B	A	0
C	A	E	0
D	A	B	1
E	A	D	1

1. Si determini la macchina minima equivalente e se ne tracci la tabella degli stati
2. Si determini un buon assegnamento degli stati
3. Si sintetizzi la funzione associata all'uscita Z

SOLUZIONE

1. Costruendo la tabella (triangolare) delle implicazioni si ottiene:

B	×			
C	×	AB,AE		
D	AE,CB	×	×	
E	CD	×	×	BD
	A	B	C	D

   La macchina di Moore é *minima* e quindi la tabella degli stati é identica a quella iniziale.

2. Applicando i criteri per trovare i vincoli di adiacenza si ottiene:
   Primo criterio

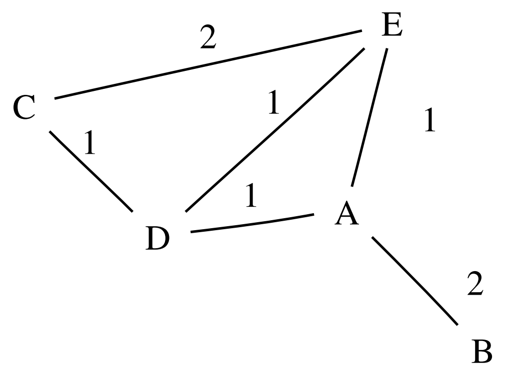
   - CD stato prossimo A con ingresso 0
   - CE stato prossimo A con ingresso 0
   - DE stato prossimo A con ingresso 0

   Secondo criterio

   - EC
   - BA
   - AE
   - AB
   - AD

   I risultati dell'applicazione dei due criteri sono riassunti nella tabella seguente. Si traccia il grafo delle adiacenze.

Vincolo	*Peso*
CD	1
CE	2
DE	1
AB	2
AE	1
AD	1



Si noti come non è possibile soddisfare tutti i vincoli, non essendo il grafo contenuto in un n-cubo. Viene eliminato l'arco AC.
Un possibile assegnamento della codifica degli stati è il seguente (è stato ricavato costruendo prima un n-cubo vuoto).

	00	01	11	10
0			C	D
1	B		E	A

3. La funzione Z (data solo dallo stato presente) è rappresentata dalla seguente tabella della verità

	Q_2	Q_1	Q_0	Z
A	1	1	0	1
B	1	0	0	0
C	0	1	1	0
D	0	1	0	1
E	1	1	1	1

da cui possiamo genereare la seguente mappa di Karnaugh

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

da cui si trova la funzione d'uscita definita come

$$Z = Q_1 Q_0' + Q_2 Q_1$$

∎

8 Classi di massima compatibilità. Riduzione di macchine a stati non completamente specificate

Esercizio 8.1 - Data la seguente tabella degli stati per una macchina (di Mealy) non completamente specificata, si chiede di procedere all'individuazione delle classi di compatibilità ed infine alla costruzione di una macchina minimizzata.

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

Soluzione
Procediamo per passi

Tabella delle implicazioni

- **primo passo**

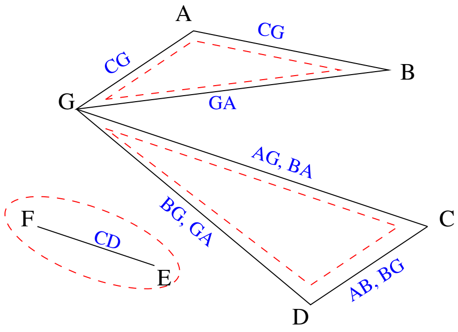
B	CG					
C	X	X				
D	X	X	AB BG			
E	CE GC	EG GC	X	X		
F	CF GD	GF GD	X	X	CD	
G	CG	GA	AB GA	GA BG	CA	DA
	A	B	C	D	E	F

- **secondo passo**

B	CG					
C	X	X				
D	X	X	AB BG			
E	X	X	X	X		
F	X	X	X	X	CD	
G	CG	GA	AB GA	GA BG	X	X
	A	B	C	D	E	F

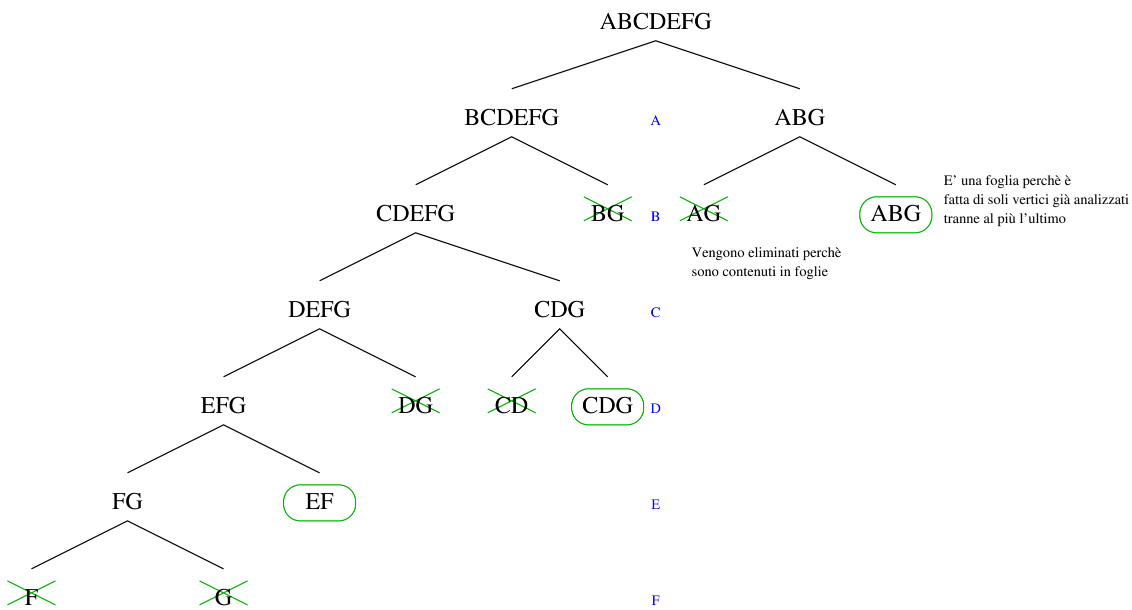
dalla quale si ottiene il seguente

Grafo di compatibilità



dal quale è immediato individuare le classi di massima compatibilità (evidenziate con tratteggio).

Nota Se non fosse così semplice ricavare le classi di massima compatibilità si può ricorrere ad un algoritmo esatto, il cui risultato è riportato qui di seguito sul medesimo esempio.

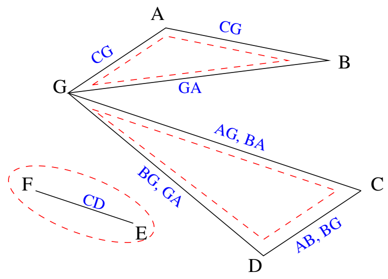


Esercizio 8.2 (continua Esercizio 8.1) - L'esercizio prosegue con la seguente richiesta: *"trovare una macchina minimale a partire dalle classi di massima compatibilitá definite nei punti precedenti."*

Soluzione

Si riportia la tabella (originale) ed il grafo sul quale si individueranno le classi di massima compatibilitá.

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



da cui le classi di massima compatibilitá risultavano essere

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

Applicazione dell'algoritmo euristico

1. **Prima Iterazione**

   (a) Inizializzo la lista vuota: $L_1 = \{\ \}$

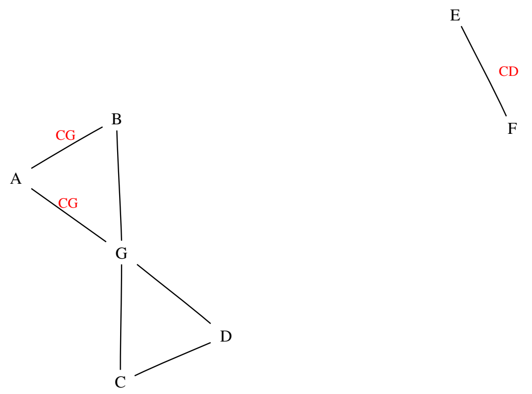
   (b) Ordino le classi: ABG, CDG, EF

      i. Scelgo quella con piú stati: **ABG**

      ii. La rimuovo e mi restano: CDG, EF

      iii. Inserisco in L_1 tutti i vincoli presenti in **ABG**: $L_1 = CG, GA$

      iv. Elimino dal grafo e da L_1 i vincoli soddisfatti da **ABG** (cioé **AB BG** e **AG**): quindi L_1 diventa uguale a CG ed il grafo diventa:



      v. Elimino dal grafo tutti i nodi, appartenenti alla classe **ABG**, che non appartengono a nessun vincolo di L_1 e/o del grafo: il grafo diventa

2. **Seconda Iterazione**

    (a) $L_1 = \{CG\}$

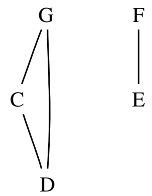
    (b) CDG, EF

        i. **CDG**

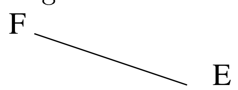
        ii. EF

        iii. $L_1 = \{CG\}$

        iv. L_1 diventa vuota, dal grafo rimuovo il vincolo soddisfatto **CD** ottenendo:



        v. rimuovo $\mathbf{C}, \mathbf{D}, \mathbf{G}$ ed il grafo diventa



3. **Terza Iterazione**

    (a) $L_1 = \{\ \}$

    (b) EF

        i. **EF**

        ii.

        iii. $L_1 = \{\ \}$

        iv. L_1 invariata, il grafo é vuoto, [END]

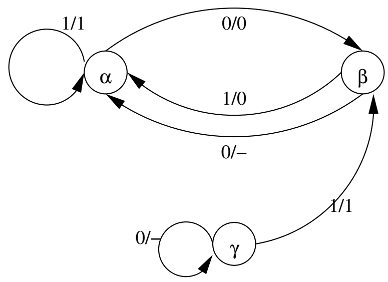
Avendo tenuto traccia delle classi via via scelte (quelle in **grassetto**) si puo' ricostruirne l'insieme delle classi di compatibilita' minimale: **ABG** , **CDG** , **EF**. Non c'é da sorprendersi del fatto che la copertura trovata sia quella costituita dalle classi di massima compatibilitá, infatti, l'algoritmo usato puo' restituire l'insieme di partenza.

Costruzione di una macchina minimale Generiamo la tabella degli stati ridotta a partire da quella originale tenendo conto delle classi α, β, γ definite

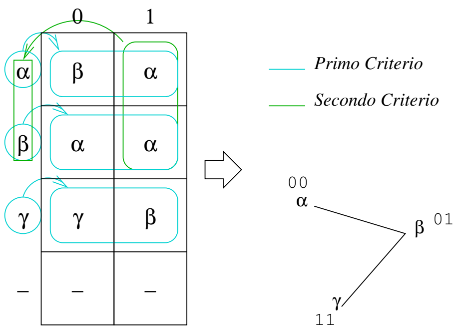
	0	1
α	$\beta/0$	$\alpha/1$
β	$\alpha/-$	$\alpha/0$
γ	$\gamma/-$	$\beta/1$
$-$	$-/-$	$-/-$

dove l'ultima riga (ininfluente) é stata aggiunta solo per rendere la tabella una "tabella di Karnaugh".

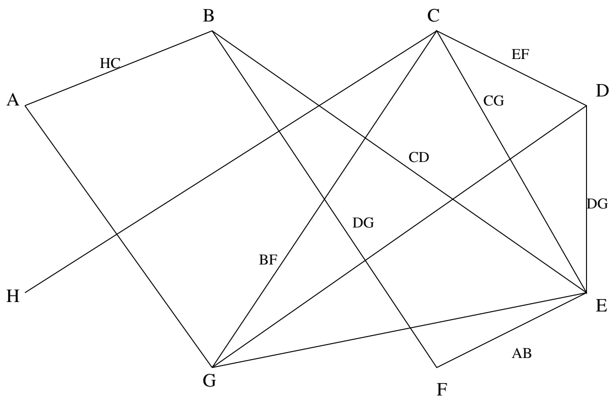
La FSM che ne deriva é la seguente



Assegnamento degli stati Applicando il *primo criterio (per colonna)* ed il *secondo criterio (per riga)* sulla tabella sopra indicata, possiamo concludere che una possibile codifica degli stati é la seguente.



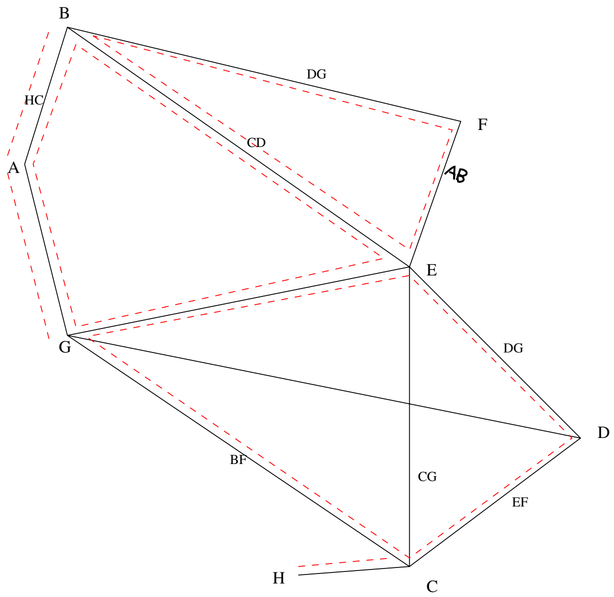
Esercizio 8.3 - Sia dato il seguente grafo di compatibilitá



determinarne l'insieme delle classi di massima compatibilitá e identificare un insieme di copertura a cardinalitá inferiore (applicando l'algoritmo euristico noto).

Soluzione

Per prima cosa spostiamo i vertici per "vedere meglio" le classi di massima compatibilitá



che risultano essere

$$\{CDGE, BFE, HC, AB, AG\}$$

Applicazione dell'algoritmo euristico

1. **Prima Iterazione**

   (a) Inizializzo la lista vuota: $L_1 = \{\ \}$

   (b) Ordino le classi: $CDGE$, BFE, HC, AB, AG

      i. Scelgo quella con piú stati: **CDGE**

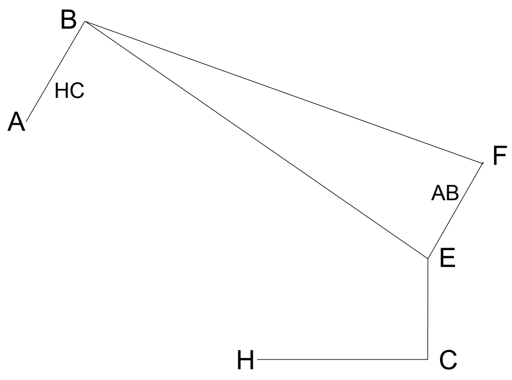
      ii. La rimuovo e mi restano: BFE, HC, AB, AG

      iii. Inserisco in L_1 tutti i vincoli in **CDGE**: $L_1 = BF$, CG, DG, EF

      iv. Elimino dal grafo e da L_1 i vincoli soddisfatti da **CDGE**: quindi L_1 diventa uguale a EF, BF

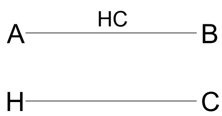
      v. Elimino dal grafo tutti i nodi appartenenti alla classe **CDGE** che non appartengono a nessun vincolo di L_1 e/o del grafo: scompaiono quindi i nodi G, D

   Al termine della prima iterazione, si ottiene che in L_1 sono ancora presenti i seguenti vincoli: $L_1 = BF$, EF, mentre il grafo risultante (e base per l'iterazione successiva) é il seguente:



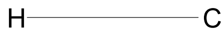
2. **Seconda Iterazione**
   (a) $L_1 = BF,\ EF$
   (b) $BFE,\ HC,\ AB,\ EC$
      i. **BFE**
      ii. $HC,\ AB,\ EC$
      iii. $L_1 = BF,\ EF,\ AB$
      iv. Elimino i vincoli soddisfatti: quindi L_1 diventa uguale a AB
      v. Elimino i nodi: vengono rimossi $F,\ E$

   Al termine della seconda iterazione, in L_1 é presente un solo vincolo: $L_1 = AB$ e il grafo risultante é il seguente:



3. **Terza Iterazione**
   (a) $L_1 = AB$
   (b) $HC,\ AB$
      i. **AB**
      ii. HC
      iii. $L_1 = AB,\ HC$
      iv. Elimino i vincoli soddisfatti: $L_1 = HC$
      v. Elimino i nodi: vengono rimossi $A,\ B$

   Al termine della terza iterazione é rimasto un solo vincolo: $L_1 = HC$ ed il grafo é il seguente:



4. **Quarta Iterazione**
   (a) $L_1 = HC$
   (b) HC
      i. **HC**
      ii. vuota
      iii. $L_1 = HC$
      iv. Elimino i vincoli soddisfatti: $L_1 =$ vuota
      v. Elimino i nodi: C, H [END]

   Al termine della quarta iterazione, non sono rimasti vincoli in L_1 ed ho considerato tutti i nodi. L'agoritmo é giunto a termine

Posso ora ricostruire l'insieme delle classi scelte: $\mathbf{CDGE}, \mathbf{BFE}, \mathbf{AB}, \mathbf{HC}$. In questo caso, l'insieme di copertura ha cardinalitá inferiore (di una classe) rispetto a quello individuato dalle classi di massima compatibilitá.

Esercizio 8.4 - Data la seguente tabella degli stati di una FSM sincrona parzialmente specificata, effettuarne la minimizzazione.

	IN=0	IN=1	OUT
A	A	C	0
B	F	A	0
C	D	E	1
D	C	E	1
E	B	-	1
F	F	D	0

Soluzione
Costruendo la tabella (triangolare) delle implicazioni si ottiene:

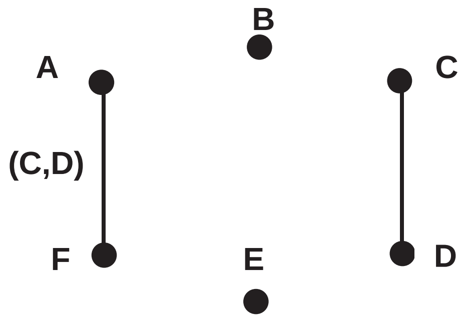
- **primo passo**

B	AF CA				
C	X	X			
D	X	X	$\vee$		
E	X	X	DB	CB	
F	CD	AD	X	X	X
	A	B	C	D	E

- **secondo passo**

B	X				
C	X	X			
D	X	X	$\vee$		
E	X	X	X	X	
F	CD	X	X	X	X
	A	B	C	D	E

Grafo di compatibilità:



Una possibile macchina ridotta compatibile con quella data é costituita da 4 stati, che corrispondono alle 4 classi di massima compatibilitá:

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

La corrispondente tabella degli stati é:

	IN=0	IN=1	OUT
α	α	δ	1
β	β	α	0
γ	β	β	0
δ	γ	-	1

∎

Esercizio 8.5 (pre compitino) - Data la seguente tabella degli stati, in cui é specificato lo stato di reset (RST):

1. Si esegua l'analisi di compatibilitá (ignorando completamente i problemi di raggiungibilitá),

2. si determinino le classi di massima compatibilitá, utilizzando un algoritmo noto

3. si indichi il limite superiore al numero degli stati nella macchina ridotta,
4. si identifichi una macchina ridotta compatibile, utilizzando un algoritmo noto, e se ne tracci la nuova tabella degli stati.

ST \ i	0	1
RST	1,0	4,0
1	RST,1	RST,0
2	6,1	5,0
3	-,-	-,0
4	5,0	9,1
5	2,0	7,0
6	3,0	7,0
7	RST,-	8,-
8	5,-	9,1
9	RST,1	7,1

Soluzione

1. Analisi di compatibilitá

   Costruendo la tabella (triangolare) delle implicazioni si ottiene:

   - **primo passo**

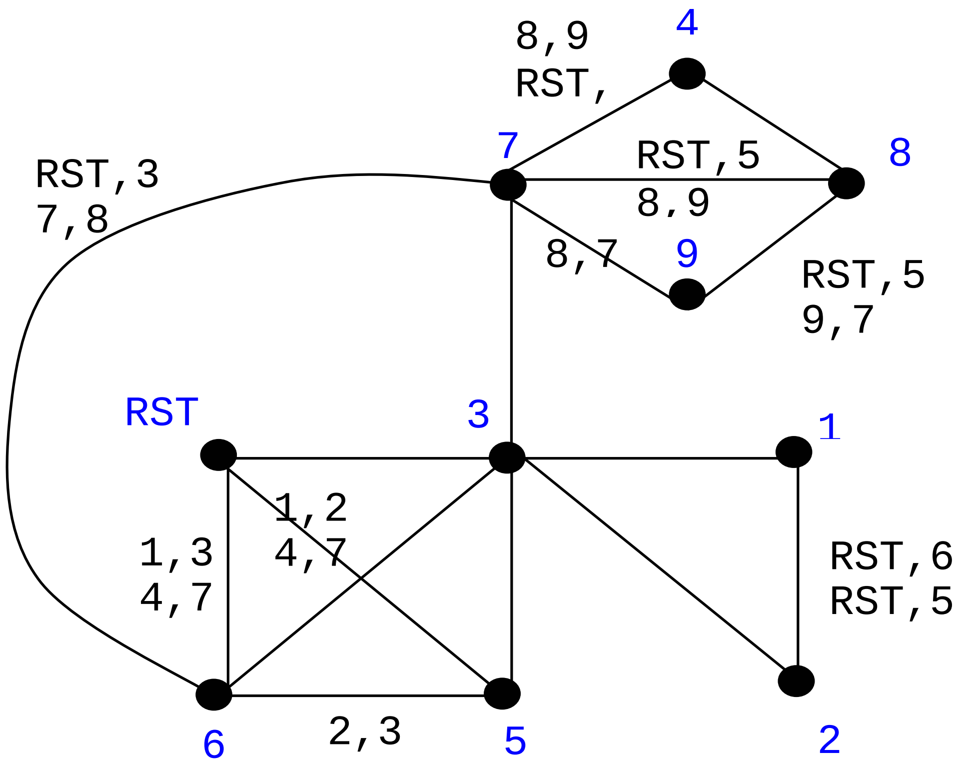
1	X								
2	X	RST,6 RST,5							
3	∨	∨	∨						
4	X	X	X	X					
5	1,2 4,7	X	X	∨	X				
6	1,3 4,7	X	X	∨	X	2,3			
7	1,RST 4,8	RST,8	6,RST 5,8	∨	5,RST 9,8	2,RST 7,8	3,RST 7,8		
8	X	X	X	X	∨	X	X	5,RST 8,9	
9	X	X	X	X	X	X	X	8,7	5,RST 9,7
	RST	1	2	3	4	5	6	7	8

   - **secondo passo**

     Analizzando la tabella e propagando le non compatibilitá si ottiene la tabella seguente

1	X								
2	X	RST,6 RST,5							
3	$\vee$	$\vee$	$\vee$						
4	X	X	X	X					
5	1,2 4,7	X	X	$\vee$	X				
6	1,3 4,7	X	X	$\vee$	X	2,3			
7	X	X	X	$\vee$	5,RST 9,8	X	3,RST 7,8		
8	X	X	X	X	$\vee$	X	X	5,RST 8,9	
9	X	X	X	X	X	X	X	8,7	5,RST 9,7
	RST	1	2	3	4	5	6	7	8

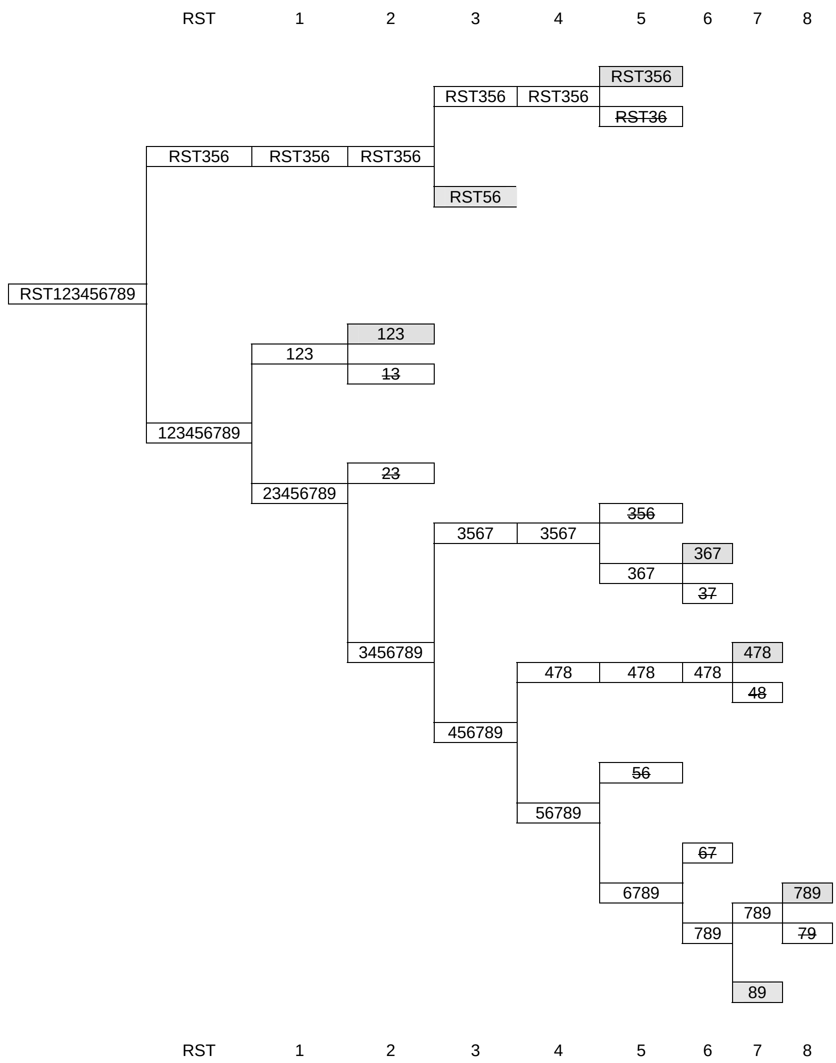
2. É possibile detrerminare le classi di massima compatibilitá derivando il grafo di compatibilitá dalla tabella delle implicazioni analizzata



Le classi di massima compatibilitá sono i sottografi completi non contenuti in altri sottografi e risultano essere:

(RST, 3, 6, 5)
(7, 3, 6)
(1, 2, 3)
(7, 4, 8)
(7, 8, 9)

L'identificazione delle classi di massima compatibilitá tramite la costruzione dell'albero é la seguente (l'albero é ruotato di 90° in senso antiorario, quindi alto equivale a destra e basso equivale a sinistra).



Si ricorda che:

- ogni livello corrisponde all'analisi di un nodo
- dato un nodo e un livello, si costruisce il suo successore di sinistra considerando tutti gli stati del progenitore tranne il nodo analizzato
- dato un nodo e un livello, si costruisce il suo successore di destra considerando tutti gli stati del progenitore giá analizzati, il nodo analizzato e gli stati successivi ad esso compatibili
- un nodo é foglia dell'albero se sono stati esaminati tutti gli stati che lo compongono, tranne al piú l'ultimo
- un nodo puó essere eliminato, se tutti i suoi stati sono compresi in un altro nodo che é una foglia
- le foglie identificano le classi di massima compatibilitá

Le classi di massima compatibilitá sono quindi le foglie dell'albero: (RST, 3, 5, 6), (1, 2, 3), (3, 6, 7), (4, 7, 8), (7, 8, 9).

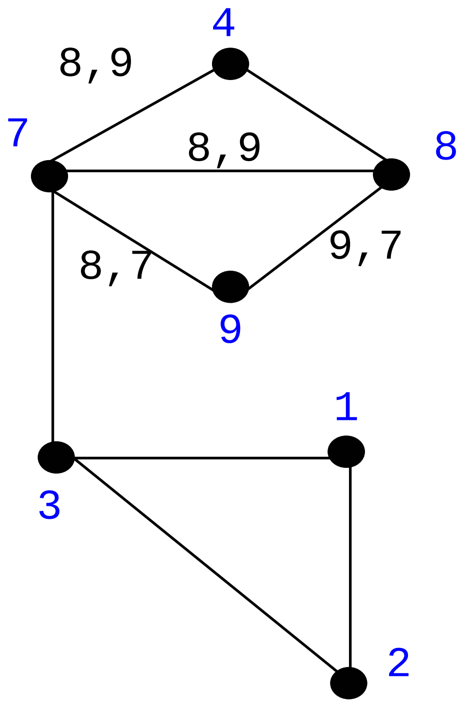
3. Il limite superiore al numero degli stati della macchina ridotta é pari al numero delle classi di massima compatibilitá, cioé 5.

4. Una possibile macchina ridotta é quella costituita considerando come stati tutte le classi di massima compatibilitá: questo porterebbe ad una macchina costituita da 5 stati.

   Si applica l'algoritmo spiegato a lezione per identificare un insieme di copertura della macchina data con cardinalitá possibilmente inferiore a 5.

   (a) **Prima Iterazione**

      i. Classi di compatibilitá da considerare:(RST, 3, 5, 6), (1, 2, 3), (3, 6, 7), (4, 7, 8), (7, 8, 9)
      ii. Lista L1: vuota
      iii. Classe scelta: (RST, 3, 5, 6)
      iv. Vincoli della classe scelta inseriti in L1: 12, 47, 23, 13
      v. Eliminazione dal grafo (disegno omesso) e da L1 dei vincoli soddisfatti dalla classe scelta. La lista L1 diventa L1: 12, 47, 23, 13
      vi. Estrazione dal grafo dei nodi della classe scelta che non compaiono né nei vincoli in L1 né in quelli presenti sul grafo: si possono estrarre il nodi RST, 6 e 5.
      vii. Grafo risultante:



(b) **Seconda Iterazione**

i. Classi di compatibilitá da considerare:(1, 2, 3), (4, 7, 8), (7, 8, 9), (3, 7)

ii. Lista L1: 12, 47, 23, 13

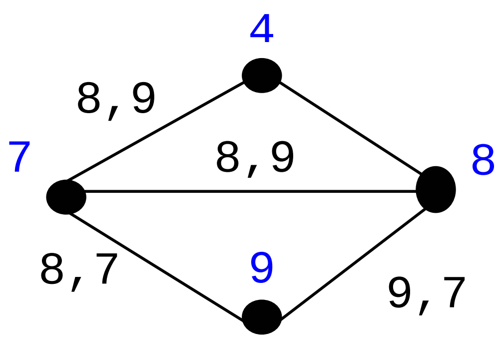
iii. Classe scelta: (1, 2, 3)

iv. Vincoli della classe scelta inseriti in L1: 12, 47, 23, 13

v. Eliminazione dal grafo (disegno omesso) e da L1 dei vincoli soddisfatti dalla classe scelta. La lista L1 diventa L1: 47

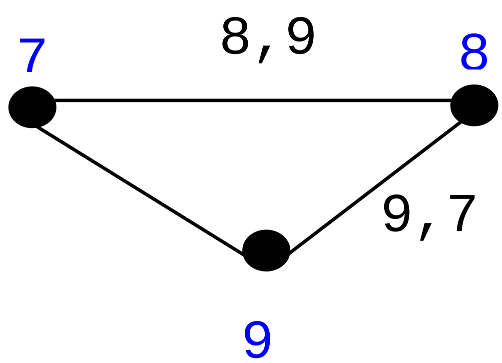
vi. Estrazione dal grafo dei nodi della classe scelta che non compaiono né nei vincoli in L1 né in quelli presenti sul grafo: si possono estrarre il nodi 1, 2 e 3.

vii. Grafo risultante:



(c) **Terza Iterazione**

i. Classi di compatibilitá da considerare: (4, 7, 8), (7, 8, 9)
ii. Lista L1: 47
iii. Classe scelta: (4, 7, 8)
iv. Vincoli della classe scelta inseriti in L1: 47, 89
v. Eliminazione dal grafo (disegno omesso) e da L1 dei vincoli soddisfatti dalla classe scelta. La lista L1 diventa L1: 89
vi. Estrazione dal grafo dei nodi della classe scelta che non compaiono né nei vincoli in L1 né in quelli presenti sul grafo: si puoéstrarre il nodo 4
vii. Grafo risultante:



(d) **Quarta Iterazione**

i. Classi di compatibilitá da considerare: (7, 8, 9)
ii. Lista L1: 89
iii. Classe scelta: (7, 8, 9)
iv. Vincoli della classe scelta inseriti in L1: 89, 97
v. Eliminazione dal grafo (disegno omesso) e da L1 dei vincoli soddisfatti dalla classe scelta. La lista L1 diventa L1: vuota

vi. Estrazione dal grafo dei nodi della classe scelta che non compaiono né nei vincoli in L1 né in quelli presenti sul grafo: si possono estrarre i nodi 7,8 e 9

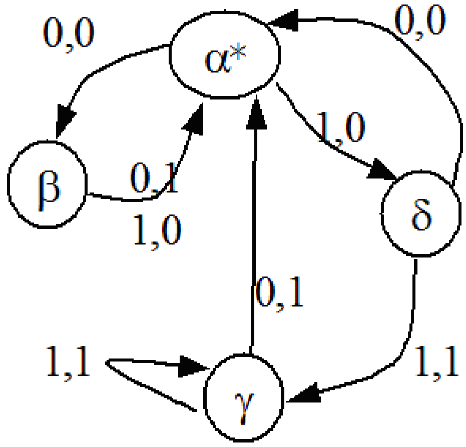
vii. Grafo risultante: vuoto

viii. L'algoritmo termina.

La copertura individuata é costituita quindi dalle seguenti 4 classi di compatibilitá:

(RST, 3, 6, 5)	=	$\alpha^\star$ (stato di reset)
(1, 2, 3)	=	β
(7, 4, 8)	=	δ
(7, 8, 9)	=	γ

Il grafo risultante é il seguente:



∎

9 Sintesi di contatori

Esercizio 9.1 - Dato il seguente ciclo di conteggio:

A	B	C
1	1	1
0	0	0
1	1	0
0	0	1
1	0	1

1. Progettare un contatore sincrono che esegue il ciclo di conteggio utilizzando bistabili di tipo JK
2. Sintetizzare inoltre le sole funzioni di eccitazione relative al bistabile C

SOLUZIONE

A	B	C	J_A	K_A	J_B	K_B	J_C	K_C
1	1	1	-	1	-	1	-	1
0	0	0	1	-	1	-	0	-
1	1	0	-	1	-	1	1	-
0	0	1	1	-	0	-	-	0
1	0	1	-	0	1	-	-	0

Sintesi 1: $J_C = A$ e $K_C = B$
Sintesi 2: $J_C = K_C = B$

∎

Esercizio 9.2 (pre compitino) - Dato il ciclo di conteggio definito nella tabella sottostante, sintetizzare un contatore che lo realizzi avendo a disposizione Filp-Flop JK (provare per esercizio anche con Flip-Flop D e T)

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

Soluzione

Si scrive per prima cosa la tabella delle eccitazioni del Filp-Flop che s'intende utilizzare (JK)

Q	Q*	J	K
0	0	0	-
0	1	1	-
1	0	-	1
1	1	-	0

ora non serve far altro che sostituire–in corrispondenza della transizione desiderata– gli ingressi al/ai Flip-Flop che la generano

A	B	C	D	J_A	K_A	J_B	K_B	J_C	K_C	J_D	K_D
0	0	1	1	0	-	1	-	-	1	-	0
0	1	0	1	0	-	-	0	1	-	-	0
0	1	1	1	1	-	-	1	-	1	-	0
1	0	0	1	-	0	0	-	1	-	-	0
1	0	1	1	-	0	1	-	-	1	-	0
1	1	0	1	-	0	-	0	1	-	-	0
1	1	1	1	-	1	-	1	-	0	-	0

Sintetizzando con le *mappe di Karnaugh* (una mappa per ogni variabile) rispetto alla variabile C. si ha rispettivamente:

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

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

e quindi

$$\begin{cases} J_C = 1 \\ K_C = A' + B' \end{cases}$$

La sintesi relativa alle altre uscite é:

$$\begin{cases} J_A = BC \\ K_A = BC \end{cases}$$

$$\begin{cases} J_B = C \\ K_B = C \end{cases}$$

$$\begin{cases} J_D = 1 \\ K_D = 0 \end{cases}$$

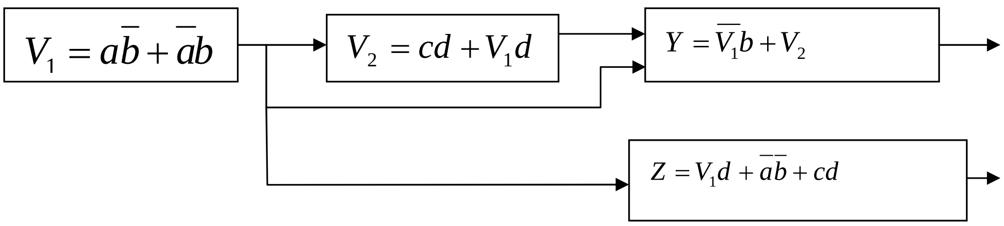
oppure

$$\begin{cases} \mathrm{J_D} = 0 \\ \mathrm{K_D} = 0 \end{cases}$$

$\blacksquare$

10 Sintesi di circuiti sequenziali con componenti programmabili: PLA, PAL, ROM

Esercizio 10.1 (pre compitino) - Data la seguente rete combinatoria multi livello con ingressi (a, b, c, d) e uscite (Y, Z)



1. Realizzare la rete combinatoria tramite PLA (si suppone di avere a disposizione tutti i termini prodotto necessari). Si indichino esplicitamente i termini prodotto del piano AND e le espressioni relative al piano OR, si disegni anche lo schema logico delle interconnessioni da programmare.

2. Realizzare la rete combinatoria tramite PAL (si suppone di avere a disposizione tutti i termini prodotto necessari) con piano OR costituito da OR a due ingressi. Si indichino esplicitamente i termini prodotto del piano AND e le espressioni relative al piano OR, si disegni anche lo schema logico delle interconnessioni da programmare.

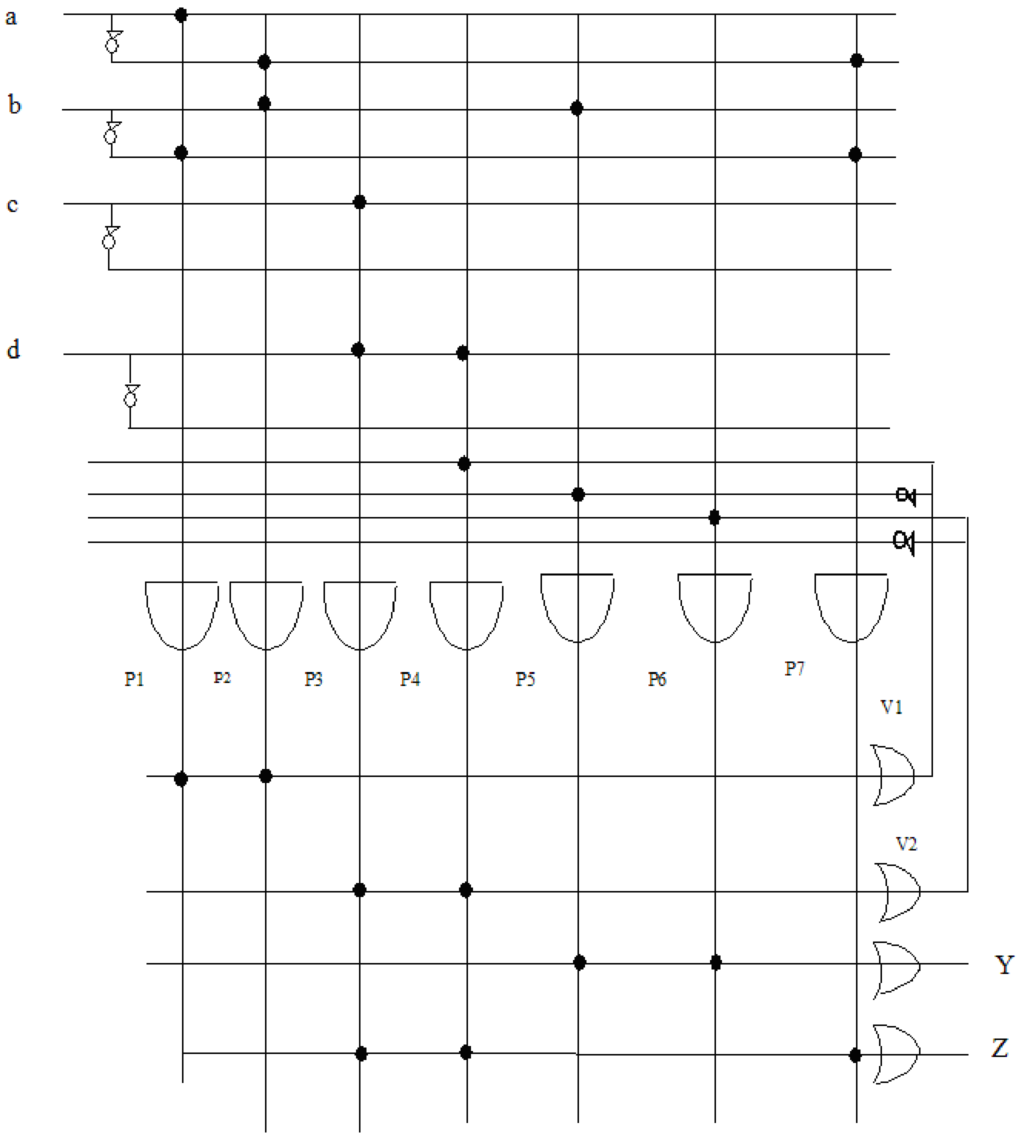
SOLUZIONE

1. Realizzazione tramite PLA

   Termini prodotto e sezione OR da realizzare:

   $P_1 = a\overline{b}$
   $P_2 = \overline{a}b$
   $P_3 = cd$
   $P_4 = V_1 d$
   $P_5 = \overline{V_1}b$
   $P_6 = V_2$
   $P_7 = \overline{a}\overline{b}$

   $V_1 = P_1 + P_2$
   $V_2 = P_4 + P_3$
   $Y = P_5 + P_6$
   $Z = P_4 + P_7 + P_3$

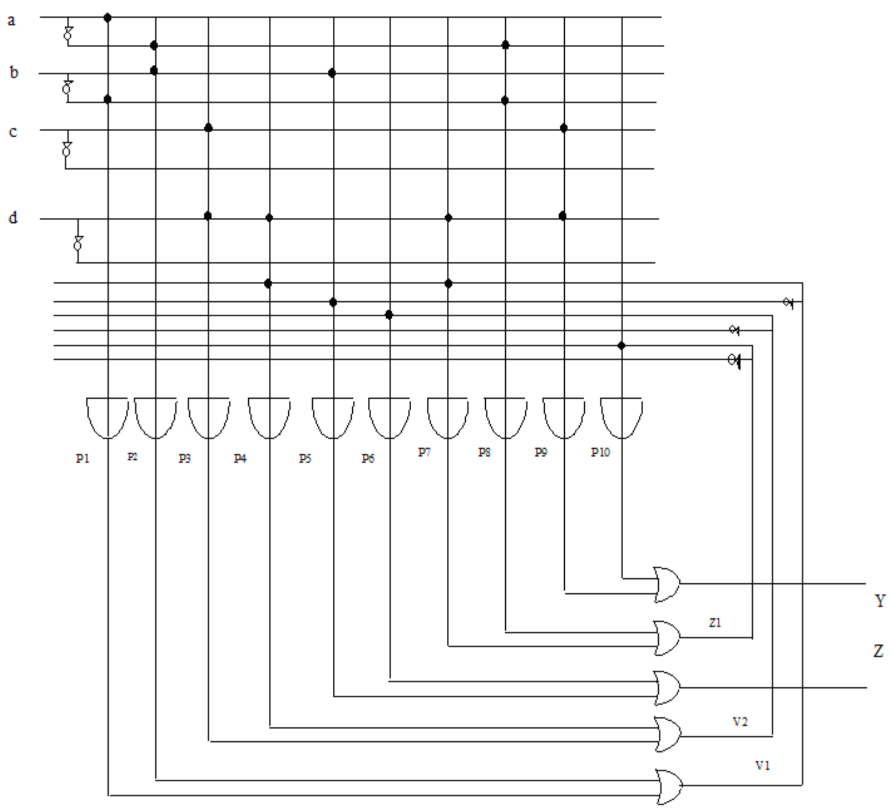

a
b
c
d
P1
P2
P3
P4
P5
P6
P7
V1
V2
Y
Z

2. Realizzazione tramite PAL

   Termini prodotto e sezione OR da realizzare

   $P_1 = a\bar{b}$
   $P_2 = \bar{a}b$
   $P_3 = cd$
   $P_4 = V_1 d$
   $P_5 = \overline{V_1} b$
   $P_6 = V_2$
   $P_7 = V_1 d$
   $P_8 = \bar{a}\bar{b}$
   $P_9 = cd$
   $P_{10} = Z_1$

   $V_1 = P_1 + P_2$
   $V_2 = P_4 + P_3$
   $Y = P_5 + P_6$
   $Z_1 = P_7 + P_8$ $Z = P_{10} + P_9$



11 VHDL

Esercizio 11.1 (pre compitino) - Data la seguente descrizione di circuito in VHDL

```
entity EXVHDL is
    port(A     : in std_logic;
         B     : in std_logic;
         clock : in std_logic;
         Z     : out std_logic);
end EXVHDL;

architecture ARC of EXVHDL is
    signal C,D,E,Z_int: std_logic;
begin
    P1: process (clock,B)
    begin
        if (clock'event and clock = '1') then
            Z_int <= D and B;
        end if;
    end process P1;

    P2: process (Z_int,B)
    begin
        if (B = '0') then
            E <= B;
        else
            E <= Z_int;
        end if;
    end process P2;

    D <= not C;
    C <= A xor E;
    Z <= Z_int;
end ARC;
```

1. Indicare se il circuito descritto é un circuito combinatorio o sequenziale con le opportune motivazioni.
2. Nel caso il circuito sia combinatorio, ricavare la tabella della veritá ad esso associata, altrimenti ricavare la tabella delle transizioni della macchina sequenziale.
3. Ricavare le forme d'onda dei segnali Z, C, D in corrispondenza dei segnali di ingresso di seguito riportati.

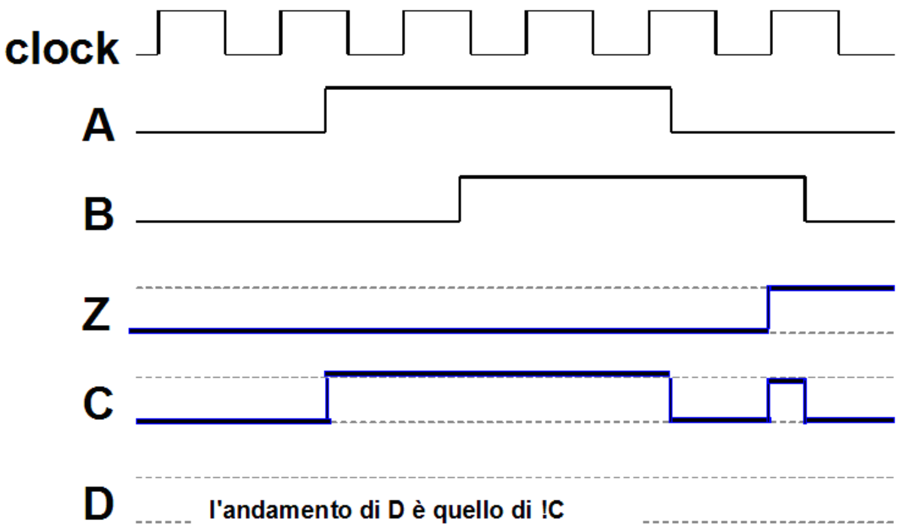
Soluzione

1. Il circuito é sequenziale e rappresenta una macchina di Moore

2. Tabella degli stati della macchina di Moore

Stato	AB=00	AB=01	AB=11	AB=10	out
S0=0	0	1	0	0	0
S1=1	0	0	1	0	1

3.



Esercizio 11.2 (pre compitino) - Data la seguente descrizione di circuito in VHDL

```
entity ESVHDL is
    port(A     : in std_logic;
         B     : in std_logic;
         clock : in std_logic;
         reset : in std_logic;
         Z     : out std_logic);
end ESVHDL;

architecture ARC of ESVHDL is
    signal C,D,E,Z_int: std_logic;
```

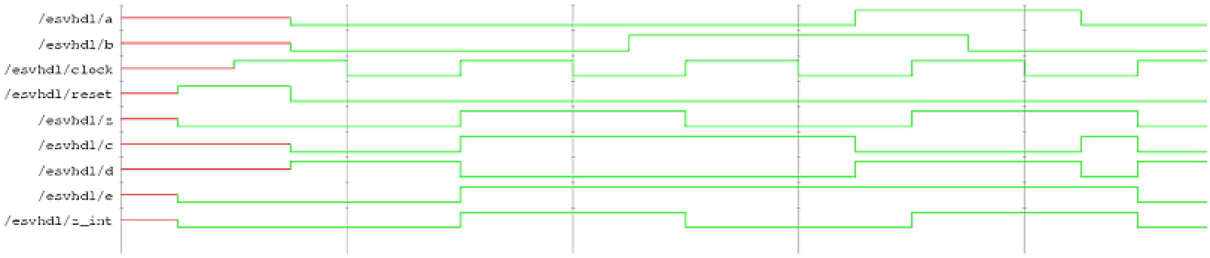
```
begin
    P1: process (clock,reset)
    begin
        if (reset = '1') then
            Z_int <= '0';
        elsif (clock'event and clock = '1') then
            Z_int <= D;
        end if;
    end process P1;

    P2: process (Z_int,B)
    begin
        if (B = '1') then
            E <= '1';
        else
            E <= Z_int;
        end if;
    end process P2;

    D <= not C;
    C <= A xor E;
    Z <= Z_int;
end ARC;
```

1. Indicare se il circuito descritto é un circuito combinatorio o sequenziale con le opportune motivazioni.
2. Ricavare le forme d'onda dei segnali out_z, C, D in corrispondenza dei segnali di ingresso di seguito riportati.
3. Nel caso il circuito sia combinatorio, ricavare la tabella della veritá ad esso associata, altrimenti ricavare la tabella delle transizioni della macchina sequenziale.

Soluzione

1. Il circuito é sequenziale e rappresenta una macchina di Moore

2.



3. Tabella degli stati della macchina di Moore

Stato	AB=00	AB=01	AB=11	AB=10	out
0	1	0	1	0	0
1	0	0	1	1	1

∎