

Reti Logiche

Appello del 25 ottobre 2007 – Seconde prove

(Rev. 02, 2007-11-12)

(D2) La *derivata* di una funzione di commutazione $f(x_1, x_2, \dots, x_k, \dots, x_{n-1}, x_n)$ rispetto alla variabile x_k è definita come

$$\partial f / \partial x_k = f(x_1, x_2, \dots, 0, \dots, x_{n-1}, x_n) \oplus f(x_1, x_2, \dots, 1, \dots, x_{n-1}, x_n)$$

Data la funzione $f(x_1, x_2, x_3, x_4) = \bar{x}_3 x_4 + x_1 (\bar{x}_3 \oplus x_4) + \bar{x}_1 \bar{x}_4 (x_2 \oplus x_3)$, determinare delle espressioni algebriche per le sue derivate rispetto a ciascuna delle variabili.

È possibile ottenere i risultati richiesti mediante pure manipolazioni algebriche; ad esempio, per la determinazione di $\partial f / \partial x_1$, si comincia col calcolo dei *cofattori* rispetto a x_1 :

$$\begin{aligned} f(0, x_2, x_3, x_4) &= \bar{x}_3 x_4 + \bar{x}_4 (x_2 \oplus x_3) \\ f(1, x_2, x_3, x_4) &= \bar{x}_3 x_4 + (\bar{x}_3 \oplus x_4) \end{aligned}$$

dopo di che l'espressione di $\partial f / \partial x_1$ ottenuta dalla definizione di derivata

$$\begin{aligned} \frac{\partial f}{\partial x_1} &= f(0, x_2, x_3, x_4) \oplus f(1, x_2, x_3, x_4) = \\ &= [\bar{x}_3 x_4 + \bar{x}_4 (x_2 \oplus x_3)] \oplus [\bar{x}_3 x_4 + (\bar{x}_3 \oplus x_4)] \end{aligned}$$

può essere semplificata mediante manipolazioni e riduzioni algebriche.

Tuttavia gli stessi risultati possono essere ottenuti in modo molto più semplice e rapido partendo dalla tavola di verità della funzione $f(x_1, x_2, x_3, x_4)$ (Tav. I) dove per convenienza ciascun

Tav. I Tavola di verità della funzione.

	x_1	x_2	x_3	x_4	f
(0)	0	0	0	0	0
(1)	0	0	0	1	1
(2)	0	0	1	0	1
(3)	0	0	1	1	0
(4)	0	1	0	0	1
(5)	0	1	0	1	1
(6)	0	1	1	0	0
(7)	0	1	1	1	0
(8)	1	0	0	0	1
(9)	1	0	0	1	1
(10)	1	0	1	0	0
(11)	1	0	1	1	1
(12)	1	1	0	0	1
(13)	1	1	0	1	1
(14)	1	1	1	0	0
(15)	1	1	1	1	1

mintermine è identificato dal corrispondente numero decimale.

A partire da questa tavola, si possono facilmente determinare le tavole di verità dei cofattori relativi alla variabile x_1 :

Tav. II Tavola di verità di $\partial f/\partial x_1$ ottenuta dalle tavole di verità dei cofattori per x_1 .

$x_2 x_3 x_4$	$f_{x_1=0}$	$f_{x_1=1}$	$\frac{\partial f}{\partial x_1}$ $f_{x_1=0} \oplus f_{x_1=1}$
0 0 0	0	1	1
0 0 1	1	1	0
0 1 0	1	0	1
0 1 1	0	1	1
1 0 0	1	1	0
1 0 1	1	1	0
1 1 0	0	0	0
1 1 1	0	1	1

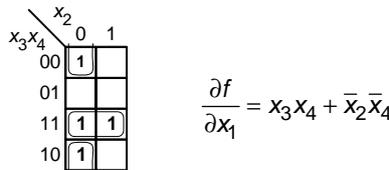


Fig. 1 Espressione algebrica di $\partial f/\partial x_1$ ottenuta mediante mappa di Karnaugh.

- la tavola di verità di $f(0, x_2, x_3, x_4) = f_{x_1=0}$ si ottiene dalla Tav. I estraendone le righe per le quali $x_1 = 0$, ossia le righe 0, 1, 2, 3, 4, 5, 6, 7;
- la tavola di verità di $f(1, x_2, x_3, x_4) = f_{x_1=1}$ si ottiene dalla Tav. I estraendone le righe per le quali $x_1 = 1$, ossia le righe 8, 9, 10, 11, 12, 13, 14, 15.

Dalle due tavole di verità così risultanti si determina poi immediatamente, mediante XOR riga per riga, la tavola di verità di $\partial f/\partial x_1$. Il procedimento è sintetizzato nella Tav. II.

Per ottenere infine un'espressione algebrica semplificata di $\partial f/\partial x_1$, è sufficiente riportarne la tavola di verità su una mappa di Karnaugh e ricavare un'espressione in somma di prodotti (Fig. 1).

Lo stesso procedimento può essere adottato per la determinazione di $\partial f/\partial x_2$ (Fig. 2):

- dalla Tav. I si ricava la tavola di verità di $f(x_1, 0, x_3, x_4) = f_{x_2=0}$ estraendone le righe per le quali $x_2 = 0$, ossia le righe 0, 1, 2, 3, 8, 9, 10, 11;
- dalla Tav. I si ricava la tavola di verità di $f(x_1, 1, x_3, x_4) = f_{x_2=1}$ estraendone le righe per le quali $x_2 = 1$, ossia le righe 4, 5, 6, 7, 12, 13, 14, 15;
- dalle due tavole di verità così ottenute si calcola, mediante operazione di XOR, la tavola di verità di $\partial f/\partial x_2$;
- si riporta quest'ultima tavola di verità in una mappa di Karnaugh e si determina un'espressione algebrica semplificata per $\partial f/\partial x_2$.

Per quanto riguarda $\partial f/\partial x_3$ (Fig. 3):

- dalla Tav. I si ricava la tavola di verità di $f(x_1, x_2, 0, x_4) = f_{x_3=0}$ estraendone le righe per le quali $x_3 = 0$, ossia le righe 0, 1, 4, 5, 8, 9, 12, 13;
- dalla Tav. I si ricava la tavola di verità di $f(x_1, x_2, 1, x_4) = f_{x_3=1}$ estraendone le righe per le quali $x_3 = 1$, ossia le righe 2, 3, 6, 7, 10, 11, 14, 15;

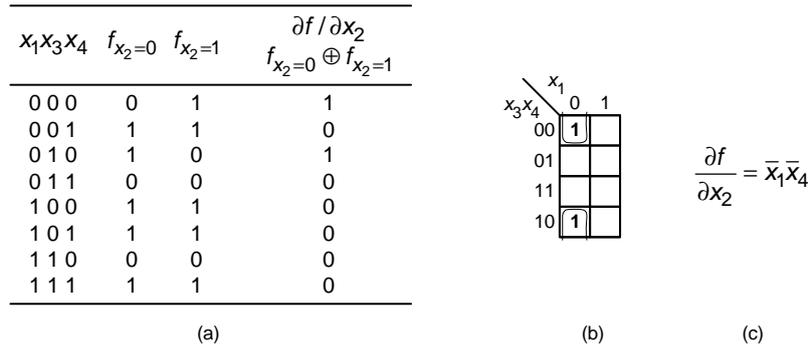


Fig. 2 Calcolo di $\partial f/\partial x_2$: (a) tavola di verità dei cofattori e della derivata, (b) mappa di Karnaugh, (c) espressione algebrica finale.

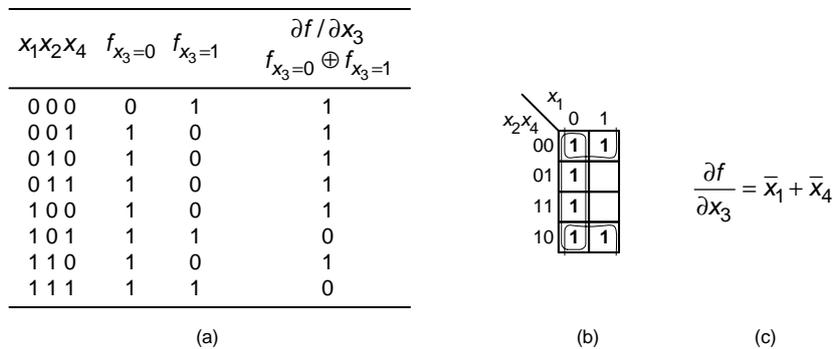


Fig. 3 Calcolo di $\partial f/\partial x_3$: (a) tavola di verità dei cofattori e della derivata, (b) mappa di Karnaugh, (c) espressione algebrica finale.

- dalle due tavole di verità così ottenute si calcola, mediante operazione di XOR, la tavola di verità di $\partial f/\partial x_3$;
- si riporta quest'ultima tavola di verità in una mappa di Karnaugh e si determina un'espressione algebrica semplificata per $\partial f/\partial x_3$.

Infine, per la determinazione di $\partial f/\partial x_4$ (Fig. 4):

- dalla Tav. I si ricava la tavola di verità di $f(x_1, x_2, x_3, 0) = f_{x_4=0}$ estraendone le righe per le quali $x_4 = 0$, ossia le righe 0, 2, 4, 6, 8, 10, 12, 14;
- dalla Tav. I si ricava la tavola di verità di $f(x_1, x_2, x_3, 1) = f_{x_4=1}$ estraendone le righe per le quali $x_4 = 1$, ossia le righe 1, 3, 5, 7, 9, 11, 13, 15;
- dalle due tavole di verità così ottenute si calcola, mediante operazione di XOR, la tavola di verità di $\partial f/\partial x_4$;
- si riporta quest'ultima tavola di verità in una mappa di Karnaugh e si determina un'espressione algebrica semplificata per $\partial f/\partial x_4$.

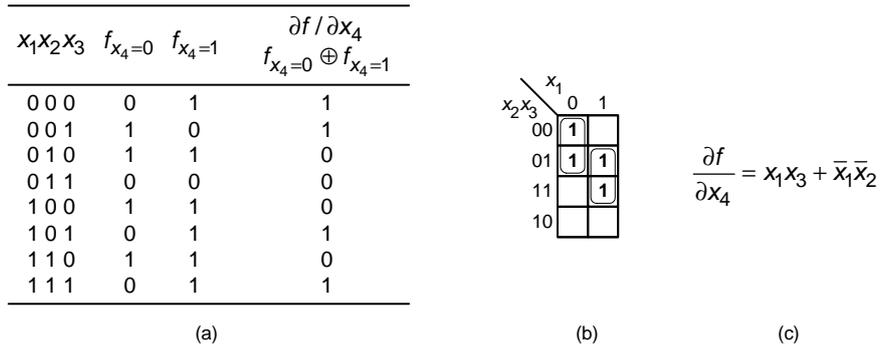


Fig. 4 Calcolo di $\partial f/\partial x_4$: (a) tavola di verità dei cofattori e della derivata, (b) mappa di Karnaugh, (c) espressione algebrica finale.

In conclusione, le derivate della funzione data possono essere espresse algebricamente come:

$$\begin{aligned} \frac{\partial f}{\partial x_1} &= x_3x_4 + \bar{x}_2\bar{x}_4 \\ \frac{\partial f}{\partial x_2} &= \bar{x}_1\bar{x}_4 \\ \frac{\partial f}{\partial x_3} &= \bar{x}_1 + \bar{x}_4 \\ \frac{\partial f}{\partial x_4} &= x_1x_3 + \bar{x}_1\bar{x}_2 \end{aligned}$$

(D3) Progettare una rete iterativa che accetta in ingresso un numero binario $X = x_{n-1}x_{n-2}\dots x_1x_0$ in complemento a 2 e produce in uscita il suo valore assoluto $Y = |X|$.

La rete deve avere due distinte modalità di comportamento, a seconda del segno di X , ossia del valore di x_{n-1} :

- se $X \geq 0$, ossia se $x_{n-1} = 0$, dovrà produrre $Y = X$;
- se $X < 0$, ossia se $x_{n-1} = 1$, dovrà produrre $Y = -X$.

(Come si ricorderà, in una rappresentazione in complemento a 2 l'opposto di un numero X si ottiene sommando aritmeticamente 1 al complemento a 1 di X : $-X = \overline{X} + 1$.)

Una possibile struttura, che fa uso di celle *Full Adder* (FA), appare in Fig. 5. Il bit di segno x_{n-1} determina la modalità di comportamento:

- se $x_{n-1} = 0$, ciascun bit di X viene avviato direttamente, tramite i multiplexer, al Full Adder; poiché il secondo operando del Full Adder è sempre 0 e il carry-in generale è forzato a 0 (essendo $x_{n-1} = 0$), nessuna cella genera carry-out e tutte le celle ricevono carry-in pari a 0, cosicché l'uscita è $Y = X + 0 = X$;
- se $x_{n-1} = 1$, ciascun bit di X viene complementato e avviato al Full Adder, mentre il carry-in dell'addizionatore viene forzato a 1 (essendo $x_{n-1} = 1$), in modo che l'uscita è $Y = \overline{X} + 1 = -X$.

Il carry-out generale rimane inutilizzato. Si noti come l'uscita Y , che è per definizione un numero positivo, abbia il bit di segno y_{n-1} forzato direttamente a 0; in tal modo la rete viene ad essere costituita da $n - 1$ celle anziché da n .

Si cominci ad osservare che il blocco invertitore/multiplexer realizza la funzione

$$x_k \overline{x_{n-1}} + \overline{x_k} x_{n-1} = x_k \oplus x_{n-1}$$

di conseguenza, la rete può essere modificata come in Fig. 6.

Inoltre, dal momento che ciascun Full Adder ha un operando fisso a 0, esso può evidentemente essere sostituito con un *Half Adder* (HA) che ha struttura più semplice. La struttura finale della cella per la rete iterativa desiderata è allora quella illustrata in Fig. 7.

È possibile seguire un diverso approccio partendo dal noto algoritmo seriale per la negazione (complementazione a 2) di un numero binario, che consiste nello scandire i bit a partire dal meno significativo lasciandoli invariati fino a che non viene incontrato un 1, e complementandoli sempre dal successivo bit in poi. Ad esempio:

$$\begin{array}{r}
 -1228 = 1\ 1\ 0\ 1\ 1\ 0\ 0\ 1\ 1\ 0\ 1\ 0\ 0 \\
 \text{complementare?} \quad \text{si} \quad \text{no} \quad \text{no} \quad \text{no} \\
 +1228 = 0\ 0\ 1\ 0\ 0\ 1\ 1\ 0\ 0\ 1\ 1\ 0\ 0
 \end{array}$$

La singola cella, in tal caso, dovrà ricevere (su un ingresso che si può continuare a indicare con *CI*) e ritrasmettere (su un'uscita che si può continuare a indicare con *CO*) alla successiva cella più significativa l'informazione relativa alla complementazione. Se, come nella soluzione precedente, si assume che x_{n-1} venga applicato come ingresso *CI* alla cella meno significativa, allora $CI = 1$ sta ad indicare che l'operando della cella non va complementato, mentre $CI = 0$ sta ad indicare che l'operando va complementato.

Il comportamento generale della cella può allora essere definito dall'algoritmo illustrato in Fig. 8 e dalla conseguente tavola di verità che appare in Tav. III, dove S rappresenta il segno di X , ossia x_{n-1} :

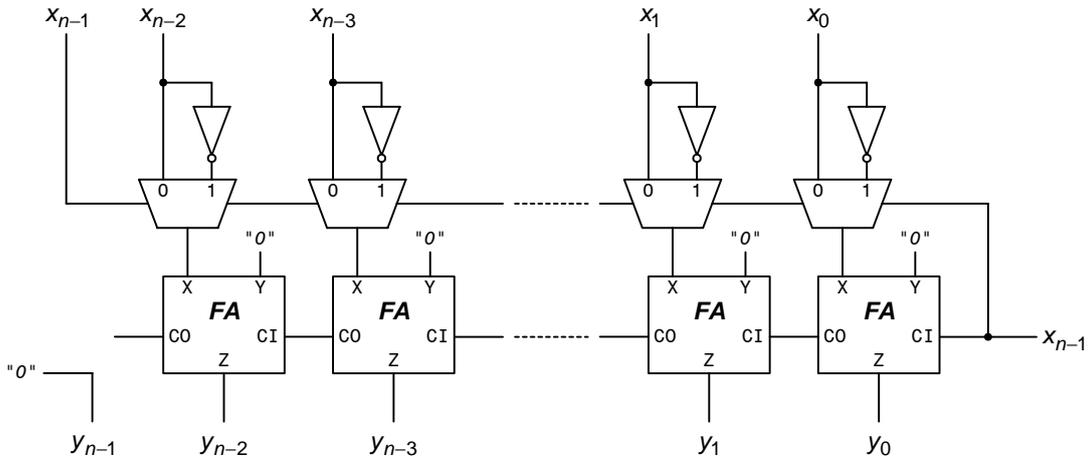


Fig. 5 Struttura della rete con celle Full Adder.

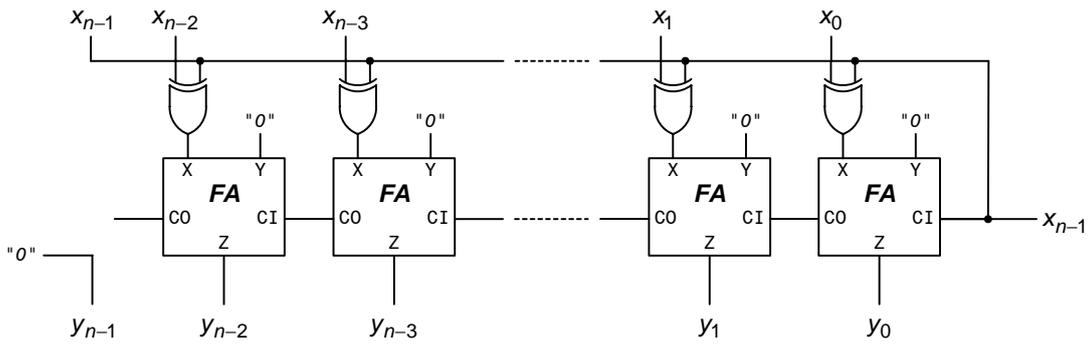


Fig. 6 Rete con Full Adder dopo eliminazione dei multiplexer.

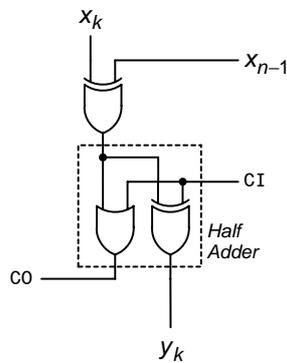


Fig. 7 Cella con uso di Half Adder.

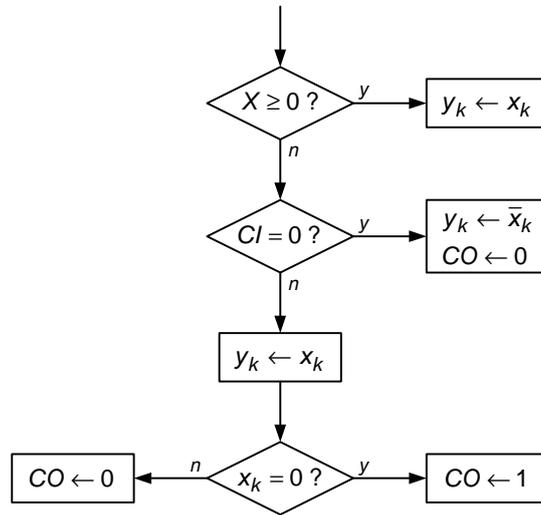


Fig. 8 Algoritmo per la generica cella.

Tav. III Tavola di verità della cella.

S	CI	x_k	y_k	CO
0	0	0	0	-
0	0	1	1	-
0	1	0	0	-
0	1	1	1	-
1	0	0	1	0
1	0	1	0	0
1	1	0	1	1
1	1	1	1	0

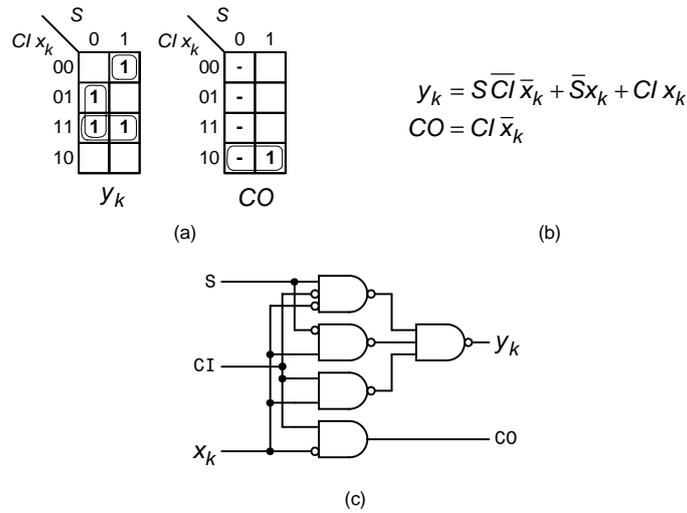


Fig. 9 Realizzazione della cella: (a) mappe di Karnaugh, (b) equazioni, (c) circuito.

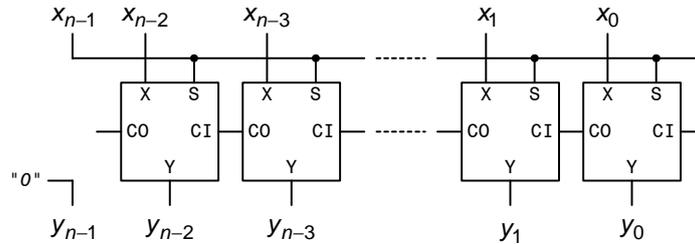


Fig. 10 Struttura finale della rete iterativa per il calcolo del valore assoluto.

- se $S = 0$, ossia se $X \geq 0$, allora CI viene ignorato, CO è un *don't care*, e l'operando x_k viene lasciato invariato e avviato all'uscita y_k ;
- se $S = 1$, ossia se $X < 0$, allora:
 - se $CI = 1$, tutti i bit incontrati finora sono stati degli zeri, dunque l'operando x_k viene lasciato invariato e avviato all'uscita; per quanto riguarda la generazione di CO , si hanno due ulteriori casi:
 - * se $x_k = 0$, deve essere $CO = 1$ perché il successivo bit x_{k+1} di X deve restare invariato;
 - * se $x_k = 1$, deve essere $CO = 0$ perché tutti i successivi bit x_{k+1}, x_{k+2}, \dots di X devono essere complementati;
 - se $CI = 0$, l'operando x_k viene complementato e avviato all'uscita, mentre deve essere $CO = 0$ per continuare a forzare la complementazione dei successivi bit di X .

La cella viene allora realizzata come appare in Fig. 9, mentre la rete finale appare in Fig. 10.

(D4) Progettare un circuito divisore di frequenza per 3 che produca in uscita un'onda quadra quando al suo ingresso viene applicata un'onda quadra.

Una forma d'onda digitale si dice *quadra* quando il suo duty cycle è del 50%, ossia quando il tempo di ON è uguale al tempo di OFF. La relazione temporale tra il segnale in ingresso (*IN*) e quello richiesto in uscita (*OUT*) è illustrata in Fig. 11. Se si usa l'ingresso *IN* come clock per un circuito sincrono, si osserva in primo luogo come l'uscita *OUT* non possa essere generata *direttamente* mediante una macchina di Moore, poiché essa deve poter commutare su entrambi i fronti del clock *IN* (in una macchina di Moore le uscite commutano *solo* in corrispondenza del fronte attivo del clock). Il comportamento desiderato si può ottenere solo facendo dipendere l'uscita *anche* dal clock *IN*.

Dal momento che la frequenza del segnale in uscita deve essere 1/3 di quella in ingresso, il punto di partenza non può che essere un contatore sincrono modulo 3, realizzato ad esempio con flip-flop JK (Fig. 12). È previsto anche un segnale di *RESET* per evitare che il circuito si disponga accidentalmente nello stato non usato $Q_1Q_0 = 11$ all'accensione. Il segnale di uscita *OUT* deve essere basso per tre semiperiodi consecutivi di *IN* e alto per i successivi tre semiperiodi; a parte questa prescrizione, le transizioni possono avvenire in corrispondenza a qualunque stato del contatore; se si impone che la relazione temporale tra *OUT* e le uscite del contatore sia ad esempio quella illustrata in Fig. 13, si può osservare che l'uscita *OUT* può essere ricavata in maniera univoca imponendo che essa valga 1 quando la terna (IN, Q_0, Q_1) vale $(1, 0, 0)$, $(0, 0, 0)$ oppure $(1, 1, 1)$, e che essa valga 0 quando $(IN, Q_0, Q_1) = (0, 1, 0)$, $(1, 0, 1)$ oppure $(0, 0, 1)$. Tale trasformazione può essere realizzata con una rete combinatoria, osservando che (IN, Q_0, Q_1) non può mai assumere i valori $(0, 1, 1)$ o $(1, 1, 1)$ (Fig. 14). Il circuito finale che ne deriva, combinando il contatore modulo 3 e il circuito generatore dell'uscita, è illustrato in Fig. 15.

È facile rendersi conto che il circuito precedente è affetto da alee di diverso tipo, sia logiche che funzionali. Un metodo alternativo per produrre l'uscita desiderata e nel contempo limitare gli effetti delle alee è quello di ritardare una delle uscite di *mezzo* periodo di clock, e poi combinare il segnale così ottenuto con le uscite del contatore in modo da ottenere il risultato richiesto. Ad esempio, si può ritardare l'uscita Q_0 di mezzo clock mediante un flip-flop D al cui ingresso *D* viene applicato Q_0 e il cui ingresso di clock è pilotato da \overline{IN} ; l'uscita DQ_0 che ne risulta (Fig. 16) può essere allora combinata con le uscite del contatore per ottenere la forma d'onda desiderata. In particolare, si osserva dalla figura che *OUT* deve valere 0 quando e soltanto quando $Q_0 = DQ_0 = 0$. Il circuito che ne deriva è allora quello illustrato in Fig. 17.

Non volendo ricorrere a circuiti sincroni corredati di accorgimenti *ad hoc*, e con l'obiettivo di ottenere un'uscita comunque immune da alee, l'unica soluzione è quella di ricorrere ad un circuito asincrono classico. Si consideri il diagramma di stato di Fig. 18: la macchina asincrona di Moore da esso descritta cambia di stato ad ogni transizione (sia $0 \rightarrow 1$ che $1 \rightarrow 0$) dell'ingresso *IN*; poiché la sequenza di uscita si deve ripetere ogni 6 semiperiodi di *IN*, essa è costituita da 6 stati, tutti stabili, e, per imporre che ad ogni transizione si abbia il cambiamento di una sola variabile di stato, si assegna agli stati un codice di Gray costruito in maniera tale che l'uscita *OUT* possa essere prelevata direttamente da una delle variabili di stato (Q_1 in Fig. 18).

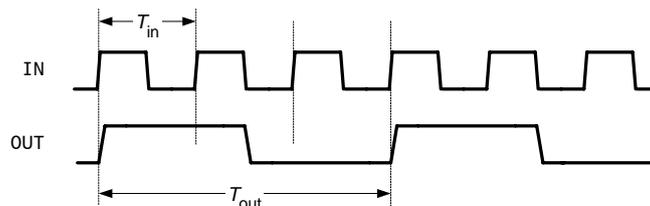


Fig. 11 Relazione temporale tra il segnale di ingresso (*IN*) e il segnale di uscita (*OUT*).

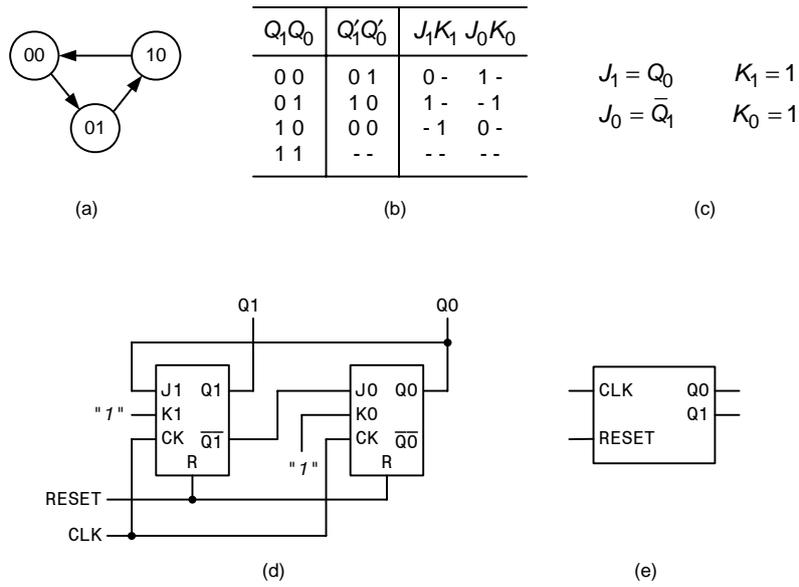


Fig. 12 Contatore sincrono modulo 3 con flip-flop JK: (a) diagramma di stato, (b) tavola di transizione e di eccitazione; (c) equazioni di eccitazione, (d) circuito, (e) rappresentazione come modulo.

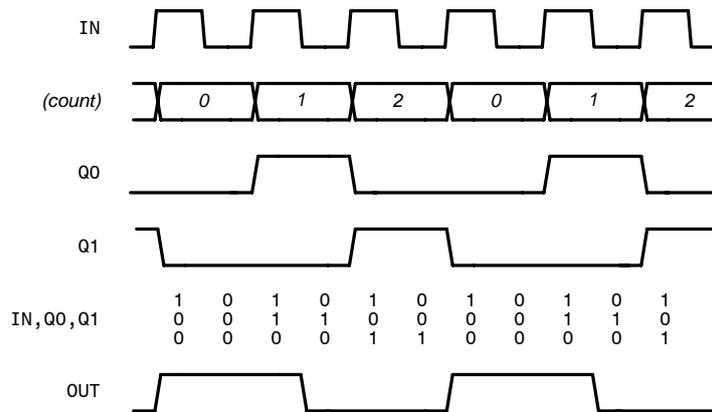


Fig. 13 Relazione temporale tra l'uscita *OUT* e lo stato del contatore modulo 3.

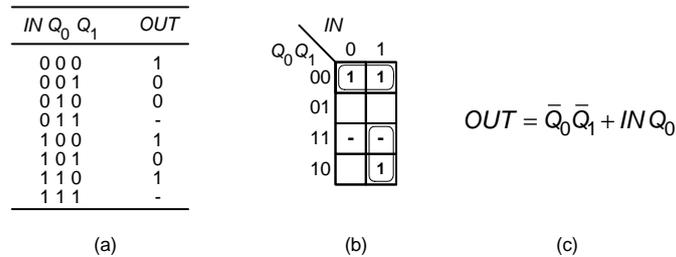


Fig. 14 Rete combinatoria per la generazione di *OUT*: (a) tavola di verità, (b) mappa di Karnaugh, (c) equazione.

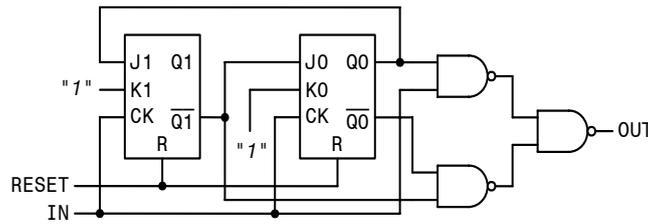


Fig. 15 Circuito finale per la generazione di *OUT* mediante contatore modulo 3.

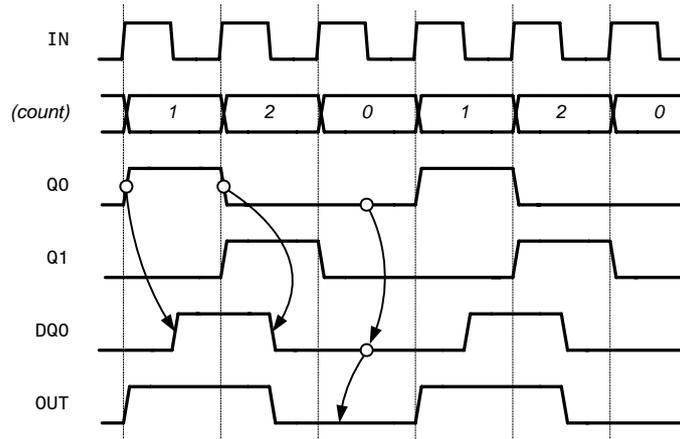


Fig. 16 L'uscita *OUT* è ricavata combinando le uscite di un contatore modulo 3 con un'uscita ritardata di mezzo clock.

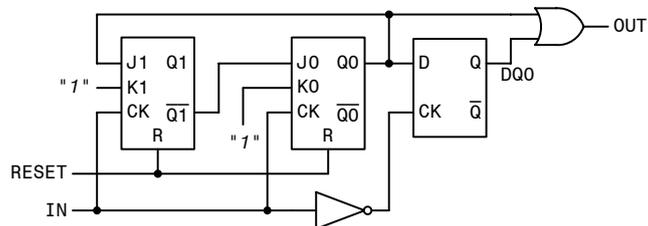
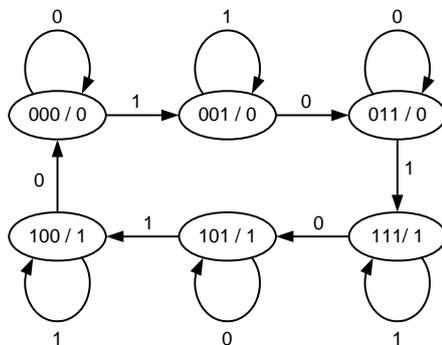


Fig. 17 Circuito per la generazione di *OUT* mediante uscita ritardata.



$Q_1Q_2Q_3$	$Q_1'Q_2'Q_3'$		<i>OUT</i>
	<i>IN</i> = 0	<i>IN</i> = 1	
000	000	001	0
001	011	001	0
010	---	---	-
011	011	111	0
100	000	100	1
101	101	100	1
110	---	---	-
111	101	111	1

(a)

(b)

Fig. 18 Macchina asincrona per la generazione di *OUT*: (a) diagramma di stato, (b) tavola di transizione.

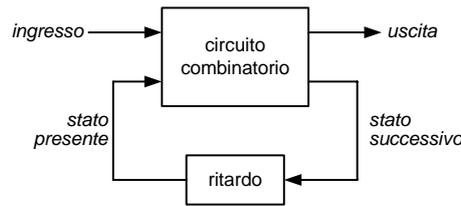


Fig. 19 Modello di macchina asincrona con ritardo come elemento di memoria.

Tav. IV Tavola di eccitazione per i flip-flop SR che mantengono le variabili di stato.

$Q_1 Q_2 Q_3 / N$	$Q'_1 Q'_2 Q'_3$	$S_1 R_1$	$S_2 R_2$	$S_3 R_3$
000 0	000	0-	0-	0-
000 1	001	0-	0-	10
001 0	011	0-	10	-0
001 1	001	0-	0-	-0
010 0	---	--	--	--
010 1	---	--	--	--
011 0	011	0-	-0	-0
011 1	111	10	-0	-0
100 0	000	01	0-	0-
100 1	100	-0	0-	0-
101 0	101	-0	0-	-0
101 1	100	-0	0-	01
110 0	---	--	--	--
110 1	---	--	--	--
111 0	101	-0	01	-0
111 1	111	-0	-0	-0

Una prima tecnica di realizzazione per tale macchina può essere quella che fa ricorso al modello classico in cui gli elementi di memoria sono costituiti da ritardi (Fig. 19). Se si assume che i ritardi siano intrinsecamente contenuti negli elementi (reali, non ideali) che costituiscono il circuito combinatorio per la generazione dello stato successivo, allora è sufficiente realizzare direttamente la tavola di transizione di Fig. 18b ponendo $Q'_k = Q_k$ per ogni k . L'implementazione corrispondente è illustrata in Fig. 20. Si noti come siano stati introdotti dei termini ridondanti ($\overline{Q}_1 Q_2$ nella copertura per Q'_2 e $Q_1 Q_3$ nella copertura per Q'_3) per eliminare le alee statiche; inoltre, nella realizzazione finale il segnale di *RESET* è stato applicato a tutte le porte di secondo livello, per garantire il forzamento dello stato $Q_1 Q_2 Q_3 = 000$.

In alternativa, gli elementi di memoria possono essere realizzati con flip-flop SR, secondo il modello di Fig. 21. In Tav. IV è rappresentata la tavola di eccitazione per i flip-flop SR che costituiscono il registro di stato, mentre in Fig. 22 è illustrata la realizzazione finale.

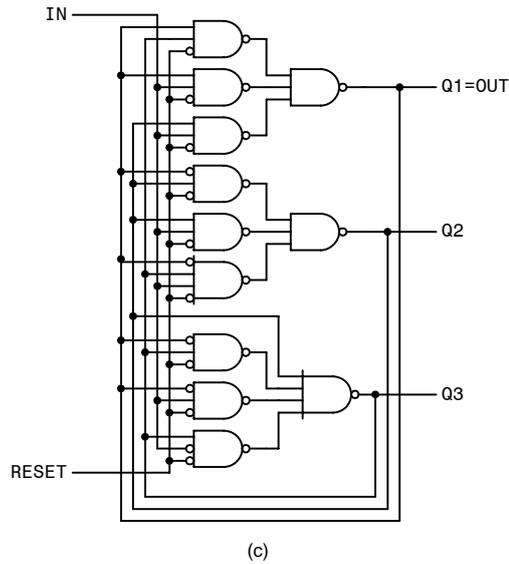
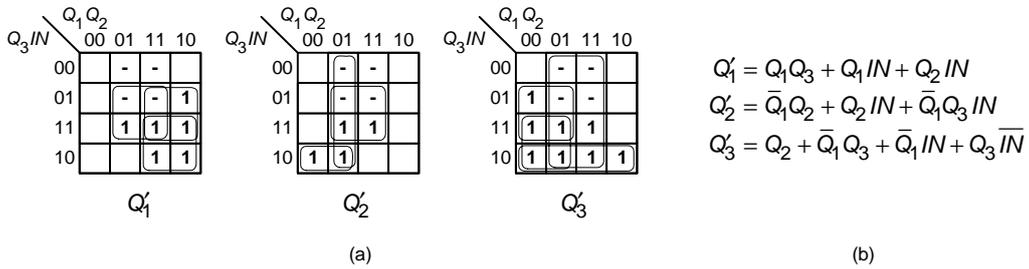


Fig. 20 Realizzazione diretta della macchina asincrona: (a) mappe di Karnaugh, (b) equazioni, (c) circuito finale.

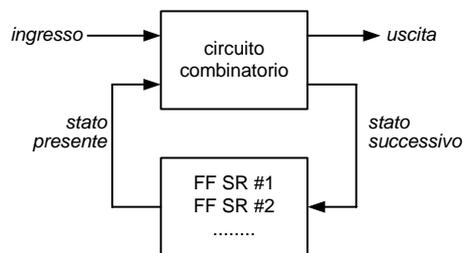


Fig. 21 Modello di macchina asincrona in cui gli elementi di memoria sono realizzati mediante flip-flop SR.

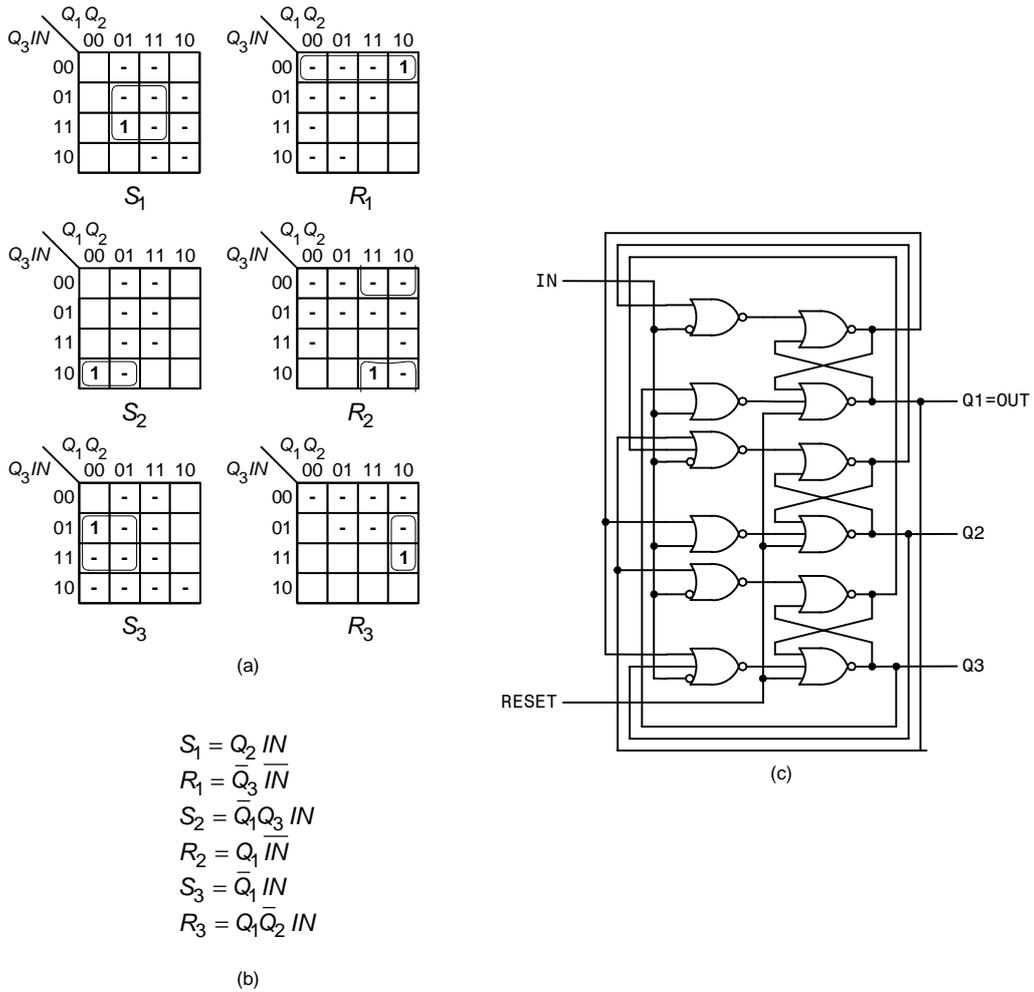


Fig. 22 Circuito asincrono con flip-flop SR come elementi di memoria: (a) mappe di Karnaugh, (b) equazioni di eccitazione, (c) circuito finale.