

Capitolo 6

Circuiti sequenziali

6.1 Introduzione

I circuiti considerati fino a questo punto sono i circuiti combinatori, nei quali in ogni istante la configurazione di una generica variabile di uscita y_i dipende unicamente dal valore assunto dalle variabili d'ingresso x_1, x_2, \dots, x_n , secondo la funzione Booleana $y_i = f_i(x_1, x_2, \dots, x_n)$, $1 \leq i \leq m$. Il modello generale di un circuito combinatorio è illustrato in figura 6.1; si noti che in esso non compare la variabile temporale, a sottolineare il



Figura 6.1: Schema generale di un circuito di commutazione con n ingressi e m uscite

fatto che i valori presi in considerazione per le n variabili di ingresso si riferiscono allo stesso istante e il comportamento della rete, se escludiamo i fenomeni transitori, è univocamente dedotto dalla tavola di verità che stabilisce il legame tra ciascuna y_i e le variabili d'ingresso x_1, x_2, \dots, x_n .

Nel caso in cui in un sistema il valore delle uscite dipenda anche dalla storia passata della circuiteria che lo costituisce, ovvero dallo *stato* della rete, si parla di sistema o circuito *sequenziale*. Il sistema telefonico è un tipico esempio di sistema sequenziale: se infatti si è in procinto di comporre l'ultima cifra di un numero telefonico, il comportamento del sistema dipenderà anche dalle cifre precedentemente selezionate; questa cifra è l'ingresso attuale del sistema e l'uscita sarà il segnale che effettuerà il collegamento. Ovviamente l'ingresso attuale non è il solo fattore che determina il collegamento, poiché anche le cifre composte precedentemente sono ugualmente importanti. Anche un calcolatore elettronico è un esempio di circuito sequenziale, anzi è l'esempio per eccellenza; di solito in esso vengono usati in maniera sequenziale diversi sottoinsiemi che possono essere di volta in volta sequenziali o combinatori.

Da un punto di vista formale un circuito sequenziale è un *automa a stati finiti* \mathcal{M} , cioè un sistema dinamico *di-*

screto (nella scansione del tempo e nella descrizione del suo stato) e *stazionario* (il sistema si comporta alla stessa maniera indipendentemente dall'istante di tempo in cui agisce). Esso è caratterizzato da:

- un insieme finito $\mathcal{Q} = \{q_1, q_2, \dots, q_S\}$ di stati interni;
- un insieme finito $\mathcal{A} = \{a_1, a_2, \dots, a_K\}$ di valori che possono essere assunti dalle variabili d'ingresso $\mathbf{x} = \{x_1, x_2, \dots, x_n\}$;
- un insieme finito $\mathcal{B} = \{b_1, b_2, \dots, b_D\}$ di valori che possono essere assunti dalle variabili di uscita $\mathbf{y} = \{y_1, y_2, \dots, y_m\}$;
- un insieme di regole, detto *mapa di transizione* τ , che specifica lo stato q^* raggiunto dalla macchina a partire dallo stato q per effetto dell'ingresso \mathbf{x} ;
- un insieme di regole, detto *mapa delle uscite* \mathcal{U} , che specifica il valore y^* assunto dalle variabili di uscita $\{y_1, y_2, \dots, y_m\}$ per effetto dell'ingresso \mathbf{x} applicato allo stato q .

La macchina sequenziale è pertanto definita dai cinque insiemi citati:

$$\mathcal{M} = (\mathcal{Q}, \mathcal{A}, \mathcal{B}, \tau, \mathcal{U})$$

Lo studio sistematico della teoria degli automi verrà fatto nell'ambito del corso di *Complessità e Crittografia* della laurea magistrale, ma si è ritenuto in ogni caso utile introdurre il formalismo per inquadrare meglio il problema. Tanto per fare un breve esempio si consideri la seguente tabella

stato	ingresso	
	0	1
q_1	$q_1/0$	$q_2/1$
q_2	$q_2/1$	$q_3/1$
q_3	$q_1/0$	$q_3/1$

che esprime il funzionamento di un automa con un ingresso e un'uscita binari, e con tre stati interni. La tabella si legge in questo modo: se l'automa si trova nello stato q_1 e si applica 0 in ingresso, allora l'automa rimane nello stato q_1 e manda 0 in uscita; se viceversa si applica 1 in ingresso, allora l'automa passa nello stato q_2 e manda 1 in uscita. La stessa tecnica di lettura viene usata anche per gli altri stati. Poiché per ciascuno degli stati q_1, q_2, q_3 dell'automa la tabella specifica il comportamento dello stesso a seconda che l'ingresso sia 0 o 1, essa incorpora la mapa di transizione τ e la mapa delle uscite \mathcal{U} ; costituisce dunque una forma di rappresentazione di un automa. Il funzionamento di una rete sequenziale può essere schematizzato come in figura 6.2, cioè come composizione di una rete combinatoria e di una memoria. Chiudiamo questa introduzione ricordando che nel nostro caso gli alfabeti d'ingresso \mathcal{I} e d'uscita \mathcal{W} saranno sempre binari.

6.2 Moduli sequenziali asincroni

Nei sistemi sequenziali introdotti al paragrafo precedente, l'informazione sulla storia passata del circuito, ovvero il valore acquisito dalla variabile di stato \mathcal{Q} , deve essere memorizzato su un qualche supporto. Si possono avere diversi tipi di dispositivi di memorizzazione, ma uno dei più usati è il cosiddetto *flip-flop*. In figura 6.3 è illustrato il *flip-flop* già introdotto nella figura 2.26, nel quale l'uscita di un transistor in interdizione viene connessa con l'ingresso di un transistor in conduzione. Il circuito che si ottiene è bistabile, nel senso che esso può stare indifferentemente e stabilmente in uno o nell'altro dei due stati rappresentati in figura 6.3a e 6.3b. Il funzionamento è basato sul fatto che, quando il transistor di sinistra è interdetto (non passa corrente nel circuito di collettore),

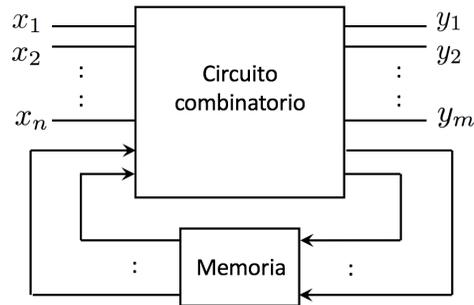
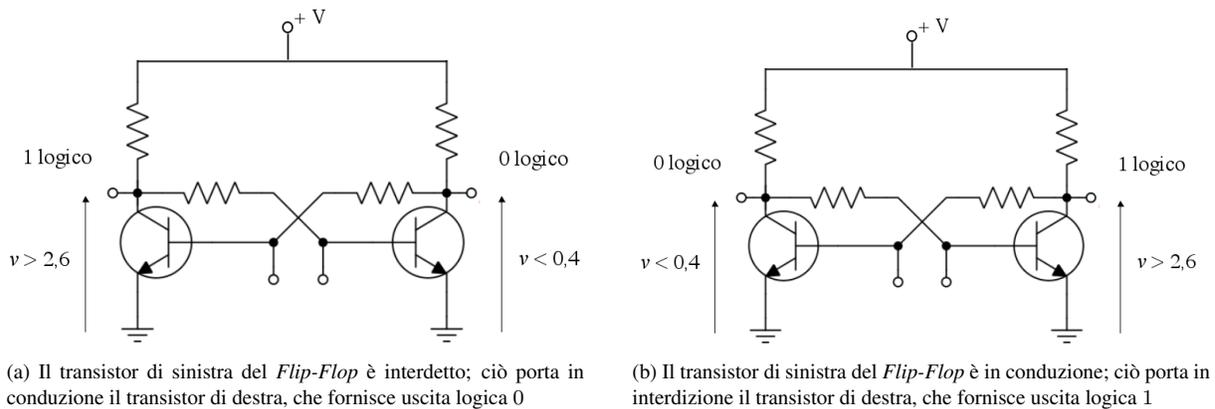


Figura 6.2: Schema logico di una rete sequenziale

Figura 6.3: Circuito bistabile del *Flip-Flop*

allora la sua tensione di collettore è alta ($v > 2,6$ V); ciò polarizza la base del transistor di destra, che entra in piena conduzione, facendo collassare a un valore basso ($v < 0,4$ V) la sua tensione di uscita (figura 6.3a), che corrisponde a uno 0 logico. I ruoli dei due transistor si scambiano quanto la tensione di collettore del transistor di destra viene portata (con un impulso) a un valore ($v > 2,6$ V) (figura 6.3b). Dal funzionamento del circuito risulta evidente che esso costituisce un elemento di memoria di 1 bit; infatti si può decidere che l'uscita del collettore di uno dei due transistor rappresenti la variabile logica il cui valore vogliamo memorizzare; scegliendo p.es. il transistor di destra, se vogliamo memorizzare 1 dobbiamo mandare lo stesso transistor in interdizione; per lo 0 dobbiamo mandarlo in conduzione (in una logica positiva).

6.2.1 Il *Flip-Flop Set-Reset* - FFSR

Anche se perfettamente funzionante, questa realizzazione non viene usata nella pratica per memorizzare *bit*, poiché si preferisce sempre ricorrere alle porte logiche, che costituiscono le unità elementari di qualunque circuito logico. Il vantaggio di tale approccio consiste nel fatto che, stando all'interno di una certa famiglia logica, tutti i segnali di comando e i livelli di tensione sono uniformati per l'intero circuito; possiamo così aggiungere singole unità funzionali senza preoccuparci di uniformare i livelli di tensione, poiché sono già standardizzati all'interno della famiglia. Mostreremo allora le due realizzazioni principali del cosiddetto *Flip-Flop Set-Reset* (FFSR), basate rispettivamente sulle porte NOR e sulle porte NAND.

Latch di NOR

La figura 6.4 illustra un *flip-flop* realizzato con due porte NOR; in gergo viene anche chiamato *Latch di NOR*. La prima cosa che balza all'occhio è il fatto che entrambe le uscite X e Y sono riportate all'ingresso; questo

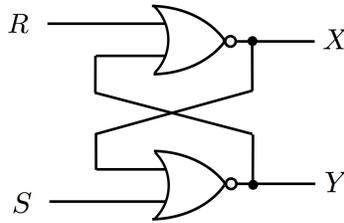


Figura 6.4: Flip-flop realizzato con due porte NOR

lascia presagire che il valore assunto dalle variabili di uscita dipenda anche dalle uscite stesse. Se impostiamo le equazioni del sistema otteniamo

$$\begin{aligned}
 X &= \overline{R + Y} & Y &= \overline{S + X} & \text{e sostituendo} \\
 X &= \overline{R + \overline{R + X}} \\
 &= \overline{R \cdot \overline{S + X}} & \text{De Morgan} \\
 &= \overline{R} \cdot (S + X)
 \end{aligned} \tag{6.1}$$

che conferma la nostra previsione. Nonostante l'equazione (6.1) sia impeccabile, non ci rende conto chiaramente del comportamento di questa semplice rete. La cosa migliore da fare è allora quella di fissare i valori di S , R , X e Y in tutti i modi possibili e vedere quali quaterne sono compatibili con i vincoli imposti dalle equazioni (6.1). Nella figura 6.5a vengono riportate tutte le possibili combinazioni per R , S , X e Y ; quelle in rosso non soddisfano l'equazione (6.1) perché $X \neq \overline{R}(S + X)$ oppure $Y \neq \overline{S} + \overline{X}$; queste configurazioni non sono stabili. Nella successiva figura 6.5b si riportano invece i soli stati stabili, per semplicità di lettura. Si osservi che, a parte il caso in cui $R = S = 1$, che escludiamo per i motivi che vedremo nel seguito, in tutte le altre combinazioni lecite si ha sempre $X = \overline{Y}$, cioè X e Y sono l'uno complementare dell'altro.

Partiamo ora dalla condizione $R = S = 0$; dalla tabella 6.5b osserviamo che ci sono due stati stabili possibili, uno con $X = 0, Y = 1$ e l'altro con $X = 1, Y = 0$; supponiamo di essere nel secondo, cioè $X = 1, Y = 0$, così come evidenziato in figura 6.6a. Supponiamo ora di portare l'ingresso R da 0 a 1 nell'istante t_1 ; quando ciò avviene, l'uscita della porta 1 commuta a $X = 0$ con un certo ritardo τ , legato ai tempi di commutazione dei transistor della porta. Il nuovo segnale $X = 0$ alimenta l'ingresso della porta 2, facendo commutare Y a 1 con un ritardo pari a 2τ . Se ora riportiamo R a 0 (si veda figura 6.6b), X e Y rimangono nella stessa configurazione acquisita, cioè $X = 0$ e $Y = 1$, poiché essa è stabile rispetto a $R = S = 0$, a norma della tabella 6.5b. Riportando ora R nuovamente a 1, non cambia comunque nulla, perché con $X = 0$ e $Y = 1$, R e S possono stare stabilmente in ciascuno dei due stati $R = S = 0$ oppure $R = 1, S = 0$. Quello che è successo è che inviando un impulso sull'ingresso R , che viene chiamato impulso di *Reset*, l'ingresso X va (o permane) a 0.

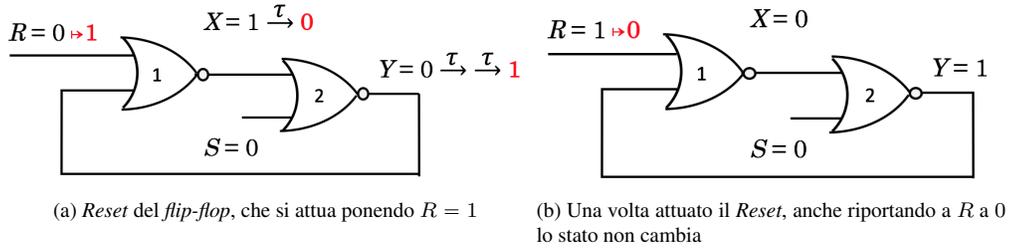
R	S	X	Y	$\overline{R}(S + X)$	$\overline{S + X}$
0	0	0	0	0	1
0	0	0	1	0	1
0	0	1	0	1	0
0	0	1	1	1	0
0	1	0	0	1	0
0	1	0	1	1	0
0	1	1	0	1	0
0	1	1	1	1	0
1	0	0	0	0	1
1	0	0	1	0	1
1	0	1	0	0	0
1	0	1	1	0	0
1	1	0	0	0	0
1	1	0	1	0	0
1	1	1	0	0	0
1	1	1	1	0	0

(a) Tavola di verità dell'equazione 6.1 per tutti gli stati possibili

R	S	X	Y	$\overline{R}(S + X)$	$\overline{S + X}$
0	0	0	1	0	1
0	0	1	0	1	0
0	1	1	0	1	0
1	0	0	1	0	1
1	1	0	0	0	0

(b) Tavola di verità dell'equazione 6.1 riferita ai soli stati stabili, cioè quelli per i quali $X = \overline{R}(S + X)$ e $Y = \overline{S + X}$

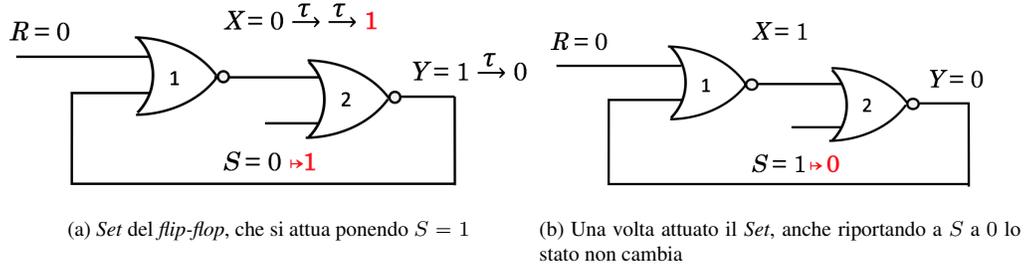
Figura 6.5: Tavola di verità dell'equazione 6.1



(a) Reset del flip-flop, che si attua ponendo $R = 1$

(b) Una volta attuato il Reset, anche riportando a R a 0 lo stato non cambia

Figura 6.6: Reset del flip-flop



(a) Set del flip-flop, che si attua ponendo $S = 1$

(b) Una volta attuato il Set, anche riportando a S a 0 lo stato non cambia

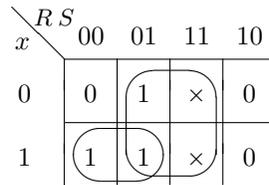
Figura 6.7: Set del flip-flop

Rifacciamo ora il ragionamento partendo dall'altra condizione, quella in cui con $R = S = 0$ si ha $X = 0, Y = 1$ (figura 6.7a). Se portiamo S da 0 a 1 nell'istante t_1 (sempre con $X = 0$), l'uscita della porta 2 passa a 0 con un ritardo τ ; istantaneamente tale $Y = 0$ si propaga all'ingresso della porta 1, e determina la commutazione di X a 1 con un ritardo pari a 2τ ; come prima, tale nuova condizione persiste anche se S viene riportato a 0 (si veda figura 6.7b), X e Y rimangono nella stessa configurazione acquisita, cioè $X = 1$ e $Y = 0$, poiché essa è stabile rispetto a $R = S = 0$, a norma della tabella 6.5b. È chiaro che ora, se anche S venisse riportato a 1, non cambierebbe

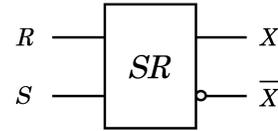
più nulla, perchè con $X = 1$ e $Y = 0$, S può stare stabilmente in ciascuno dei due stati. Quello che è successo è che inviando un impulso sull'ingresso S , che viene chiamato di impulso di *Set*, l'ingresso X va (o permane) a 1. Il

R	S	x	X
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	—
1	1	1	—

(a) Tavola di verità con lo stato futuro



(b) Mappa di Karnaugh della funzione X di figura 6.8a



(c) Simbolo circuitale di un *flip-flop* SR

Figura 6.8: Tavola di verità e mappa di Karnaugh per lo stato futuro

passaggio di stato della X viene evidenziato in figura 6.8a, nella quale si distingue tra stato corrente, indicato con la lettera x , e stato futuro, che viene invece indicato con X . Ricapitolando, il funzionamento di questo dispositivo può essere descritto nel modo seguente: quando $R = S = 0$, un impulso su S porta l'uscita X a 1; un impulso su R riporta l'uscita X a 0; ciò viene evidenziato in figura 6.8a dalle cifre in blu (*Set*) e in rosso (*Reset*). Nella figura 6.8b viene evidenziata la mappa di Karnaugh della funzione X , che consente di esprimere lo stato futuro in funzione di R , S e dello stato presente x

$$X = S + x\bar{R} \quad \text{col vincolo che} \quad RS = 0 \quad (6.2)$$

Un *flip-flop* che operi nel modo descritto prende il nome di *Flip-Flop Set-Reset*, e viene indicato con il simbolo SR. La ragione per cui i due ingressi non possono essere contemporaneamente 1 è duplice; in primo luogo perché essi porterebbero ambedue le uscite a 0, violando la condizione base di funzionamento di un *flip-flop*, secondo la quale le due uscite devono essere sempre complementari; in secondo luogo, se ambedue gli ingressi tornassero a 0 al medesimo istante, lo stato in cui il *flip-flop* si porterebbe non sarebbe prevedibile e al limite potrebbe realizzarsi una condizione di oscillazione. Con il vincolo $RS = 0$ il *flip-flop* SR realizza invece un dispositivo di memorizzazione affidabile, in cui lo stato delle uscite indica quale dei due ingressi si è trovato per ultimo al livello 1.

Poiché un tale circuito reagisce immediatamente alle variazioni dell'ingresso (a parte i tempi di commutazione), portandosi nello stato futuro previsto dalla funzione di transizione di stato, il suo funzionamento è di tipo *asincrono*, cioè non legato ad alcuna forma di sincronismo. E' inoltre evidente che l'impulso d'ingresso che fa commutare il dispositivo deve avere una durata minima se si vuole che la commutazione avvenga con sicurezza; si supponga infatti che S ritorni a zero prima che sia passato il tempo τ ; in tal caso la variazione della variabile S non si propaga in uscita e la X rimarrebbe a 0. Il verificarsi di tale condizione renderebbe evidentemente incerto il funzionamento del circuito.

Latch di NAND

Un discorso del tutto analogo potrebbe essere fatto ricorrendo a due porte NAND al posto delle porte NOR, così come rappresentato in figura 6.9

Le equazioni del sistema sono

$$\begin{aligned} X &= \overline{RY} & Y &= \overline{SX} & \text{e sostituendo} \\ X &= \overline{R \cdot \overline{SX}} \\ &= \overline{R} + \overline{\overline{SX}} & \text{De Morgan} \\ &= \overline{R} + SX \end{aligned} \quad (6.3)$$

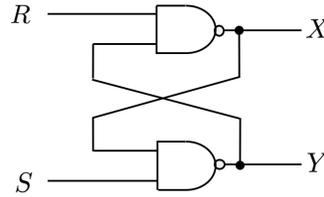


Figura 6.9: Flip-flop realizzato con due porte NAND

e appaiono come le duali delle [6.1](#). Se costruiamo le tavole di verità analoghe a quelle di figura [6.5](#) otteniamo le tavole di figura [6.10](#) dalle quali si deduce che ora la condizione da evitare è $R = S = 0$. Con questa ipotesi si ha sempre $Y = \bar{X}$, come si richiede a un *Flip Flop* SR. Rifacendo lo stesso ragionamento fatto sulle porte NOR si

R	S	X	Y	$\bar{R} + SX$	$\bar{S}\bar{X}$
0	0	0	0	1	1
0	0	0	1	1	1
0	0	1	0	1	1
0	0	1	1	1	1
0	1	0	0	1	1
0	1	0	1	1	1
0	1	1	0	1	0
0	1	1	1	1	0
1	0	0	0	0	1
1	0	0	1	0	1
1	0	1	0	0	1
1	0	1	1	0	1
1	1	0	0	0	1
1	1	0	1	0	1
1	1	1	0	1	0
1	1	1	1	1	0

(a) Tavola di verità dell'equazione [6.3](#) per tutti gli stati possibili

R	S	X	Y	$\bar{R} + SX$	$\bar{S}\bar{X}$
0	0	1	1	1	1
0	1	1	0	1	0
1	0	0	1	0	1
1	1	0	1	0	1
1	1	1	0	1	0

(b) Tavola di verità dell'equazione [6.3](#) riferita ai soli stati stabili, cioè quelli per i quali $X = \bar{R} + SX$ e $Y = \bar{S}\bar{X}$

Figura 6.10: Tavola di verità dell'equazione [6.3](#)

osserva che ora lo stato di partenza è a livello alto ($R = 1, S = 1$), mentre i segnali di attivazione sono a livello basso ($R = 0$ oppure $S = 0$). Ciò è coerente con la dualità tra le relazioni [6.1](#) e [6.3](#) e implica che dare un segnale di *Set* significa porre $S = 0$, ottenendo che X vada o permanga a 0; viceversa dare un segnale di *Reset* significa porre $R = 0$, ottenendo che X vada o permanga a 1. Questo comportamento è coerente con la tavola [6.10b](#).

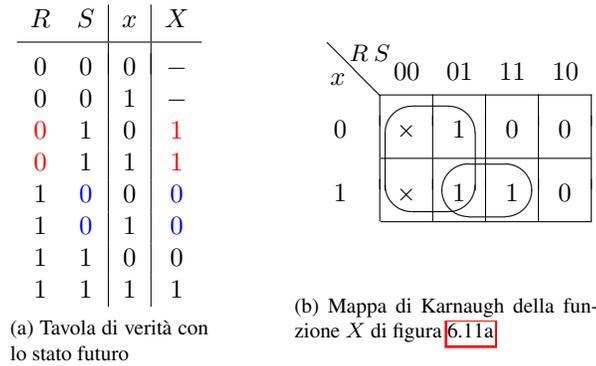


Figura 6.11: Tavola di verità e mappa di Karnaugh per lo stato futuro

Il passaggio di stato della X viene evidenziato in figura 6.11a, analoga alla tavola 6.8a. Ricapitolando, il funzionamento di questo dispositivo può essere descritto nel modo seguente: quando $R = S = 1$, un impulso (negativo) su S porta l'uscita X a 0; un impulso (negativo) su R riporta l'uscita X a 1; ciò viene evidenziato in figura 6.11a dalle cifre in blu (*Set*) e in rosso (*Reset*). Nella figura 6.11b viene evidenziata la mappa di Karnaugh della funzione X , che consente di esprimere lo stato futuro in funzione di R, S e dello stato presente x

$$X = \bar{R} + xS \quad \text{col vincolo che} \quad R + S = 1 \quad (6.4)$$

6.3 Moduli sequenziali sincroni

6.3.1 Il *Flip-Flop* SR sincrono

Il progetto di un circuito sequenziale può essere notevolmente semplificato se le commutazioni possono avvenire solo in corrispondenza in precisi istanti di tempo equintervallati. Questo tipo di funzionamento può essere assicurato se tutti i cambiamenti di stato vengono sincronizzati da opportuni impulsi di orologio (*clock*). È bene chiarire che per *impulso* s'intende un segnale che normalmente si mantiene a un livello, usualmente 0, e va all'altro livello solamente per intervalli di tempo estremamente brevi. Viene invece chiamato *a livelli* un segnale che può rimanere sia a 0 che a 1 per periodi di tempo indefiniti e comunque molto lunghi se paragonati alla durata di un impulso. La dizione "estremamente breve" va evidentemente rapportata alla velocità del circuito, ma normalmente indica una durata dello stesso ordine di grandezza del tempo di ritardo del *flip-flop*. Dunque un circuito sequenziale

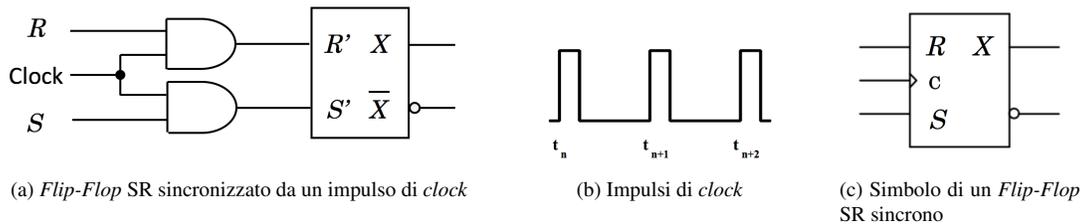


Figura 6.12: *Flip-Flop* SR sincrono e relativo impulso di sincronizzazione

sincronizzato da un impulso di *clock*, detto circuito *sincrono* o circuito *clock-mode*, può cambiare stato solo in corrispondenza di tale impulso; inoltre potrà cambiare stato non più di una volta per ciascun impulso di *clock*. L'unico vincolo necessario per il corretto funzionamento del circuito è che i segnali S e R non cambino durante l'intera durata dell'impulso di *clock*.

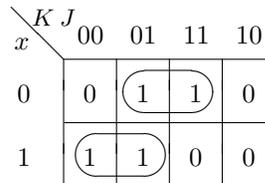
Per realizzare FFSR sincrono bisogna creare un consenso con delle porte AND, così come illustrato in figura 6.12a, il funzionamento del circuito è evidente: quando il segnale di *clock* di figura 6.12b ha valore basso, si ha $C = 0$ e dunque $S' = 0$ e $R' = 0$ indipendentemente dai valori di S e R . Quando invece l'impulso di *clock* scatta, si ha $C = 1$, le porte AND diventano trasparenti rispetto a S e R e il *flip-flop* funziona al solito modo. La tavola di verità e le equazioni che reggono il funzionamento del *flip-flop* sincrono sono le stesse del *flip-flop* SR asincrono; in questo caso S , R e x indicano i valori presenti durante l'impulso di clock, mentre X si riferisce al valore assunto dall'uscita immediatamente dopo tale impulso.

6.3.2 Il Flip-Flop JK - FFJK

Malgrado il *flip-flop* SR sincrono sia assolutamente adatto a qualsiasi realizzazione circuitale, esso è raramente impiegato in pratica, poiché si preferisce far uso del *flip-flop* JK (FFJK) che ora descriviamo. In quest'ultimo il vincolo che ambedue gli ingressi non possano essere 1 allo stesso istante viene a cadere. La tavola di verità del *flip-flop* JK e la corrispondente mappa di Karnaugh sono riportate in figura 6.14. Si noti che il *flip-flop* JK è identico a quello RS con J corrispondente a S e K a R eccetto quando ambedue gli ingressi valgono 1, caso in cui il *flip-flop* comunque cambia stato. Dalla mappa di Karnaugh si evidenzia la struttura della funzione X

K	J	x	X
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

(a) Tavola di verità con lo stato futuro di un *flip-flop* JK

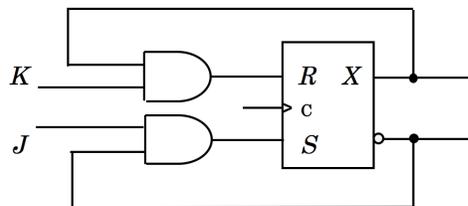


(b) Mappa di Karnaugh della funzione X di figura 6.13a

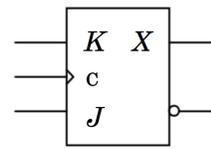
Figura 6.13: Tavola di verità di un *Flip-Flop* JK e semplificazione sulla mappa di Karnaugh

$$X = \bar{x}J + x\bar{K} \tag{6.5}$$

La realizzazione pratica di un FFJK si attua con l'aggiunta di due porte AND a un FFSR sincrono, così come evidenziato in figura 6.14a, quando $x = 1$ si abilita la sola porta AND del *reset* K , quando $x = 0$ si abilita la sola porta AND del *set* J ; se entrambi J e K sono a 1, $x = 1$ forza un *reset*, mentre $x = 0$ forza un *set*.



(a) *Flip-Flop* JK realizzato a partire da un RS sincronizzato



(b) Simbolo circuitale di un *flip-flop* JK

Figura 6.14: *Flip-Flop* JK

6.3.3 Flip-Flop di tipo T e D

Un ulteriore tipo di *flip-flop* è il *flip-flop* T (FFT) - dove la T sta per *Toggle* - in cui il *clock* è il solo segnale d'ingresso, che in questo caso viene chiamato segnale di *trigger*. Quando l'ingresso T è a 1, a ogni istante di *clock* il *flip-flop* cambia stato; se invece l'ingresso T è a 0 esso rimane nello stesso stato; l'equazione di funzionamento si ricava dalla 6.5 ponendo $J = T$ e $K = T$:

$$X = \bar{x}T + x\bar{T} = \begin{cases} \bar{x} & \text{se } T = 1 \\ x & \text{se } T = 0 \end{cases}$$

Come evidenziato dalla figura 6.15a, un FFT si ottiene direttamente da un FFJK semplicemente connettendo assieme gli ingressi J e K ; in questo modo, ogniqualvolta si ha $T = 1$, entrambi J e K sono a 1 e si ha la commutazione dello stato finale per quanto detto precedentemente.

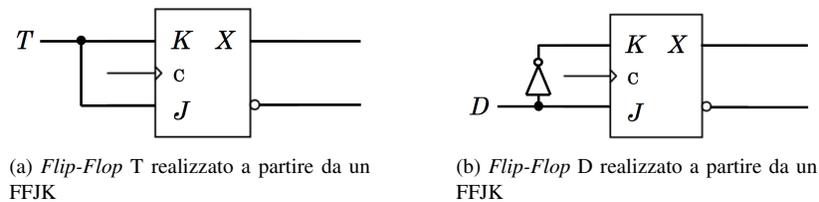


Figura 6.15: *Flip-Flop* del tipo T e D

Un impiego tipico del FFT è quello di dimezzare la frequenza di *clock*, poiché una volta cambiato lo stato di uscita esso non cambia più fino al prossimo impulso di *clock*.

L'ultimo *flip-flop* che analizziamo è quello di tipo D (FFD) - dove la D sta per *Delay*. In esso l'uscita dopo un impulso di *clock* è uguale al valore presente all'ingresso D all'istante di *clock*. L'equazione di funzionamento si ricava dalla 6.5 ponendo $J = D$ e $K = \bar{D}$

$$X = \bar{x}D + x\bar{D} = D \quad (6.6)$$

Dunque lo stato futuro X è lo stato che si avrà nel prossimo impulso di *clock* a seguito dell'ingresso D . In tal modo il *flip-flop* D realizza il trasferimento in uscita del segnale d'ingresso D con un ritardo pari al periodo di *clock* T .

6.4 Registri e contatori

I *flip-flop* costituiscono la circuiteria di base per la memorizzazione di singoli bit in formato elettronico. A partire da essi si possono costruire unità per la memorizzazione di blocchi di m bit denominati *registri*; la figura 6.16 ci mostra un esempio in tal senso, nel quale si costruisce un registro di m celle di memoria a partire da m *flip-flop* di tipo D.

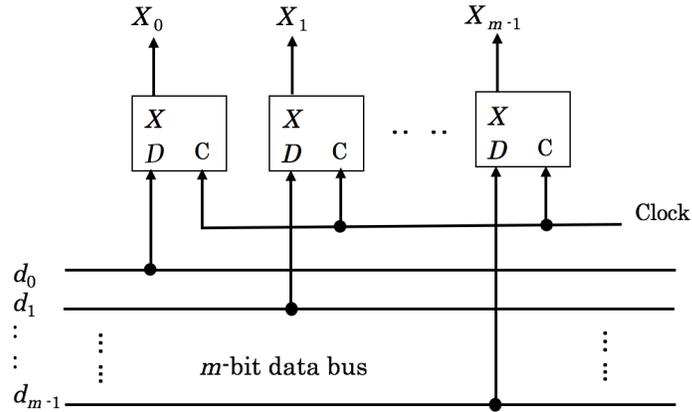


Figura 6.16: Registro di memoria da m -bit

Gli stessi m *flip-flop* di tipo D possono essere organizzati per realizzare i *registri a scorrimento*, che sono di fondamentale importanza per i flussi informativi interni ai calcolatori. Il circuito è rappresentato in figura 6.17 e il suo funzionamento è intuitivo: a ogni istante di tempo il contenuto del registro j -esimo si sposta nel registro $(j-1)$ -esimo e si rende disponibile per l'uscita.

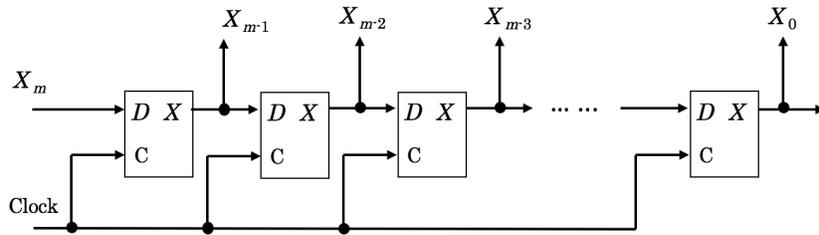


Figura 6.17: Registro a scorrimento

Con m *flip-flop* di tipo T è possibile realizzare anche un *contatore*, che scandisce, una dopo l'altra, tutte le 2^m configurazioni da 00...0 a 111...1. Per capirne il funzionamento facciamo riferimento al contatore a 2 bit di figura 6.18, partendo dalla configurazione 00. Il *flip-flop* di sinistra ha sempre $J = K = 1$, e in corrispondenza del primo

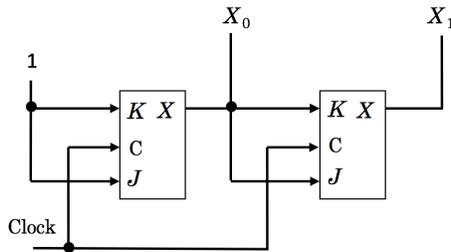


Figura 6.18: Contatore a 2 bit che passa attraverso gli stati 00, 10, 01, 11 per le variabili X_0X_1

istante di *clock* cambia lo stato da $X_0 = 0$ a $X_0 = 1$; al successivo si ha $X_0 = 0$ e così via, per ogni istante in cui compare il segnale di *clock*. Prima che avvenga la commutazione si ha $X_0 = 0$, e dunque $J=K=0$ per il *flip-flop*

di destra; dalla tavola di verità [6.13a](#) di un *flip-flop* JK si vede che, quando $J=K=0$ lo stato finale della variabile X viene conservato, e dunque nel momento in cui il *clock* si manifesta, mentre X_0 cambia X_1 resta nello stato iniziale, cioè $X_1 = 0$; il nuovo stato successivo a 00 è dunque 10. In queste condizioni all'avvento del secondo *clock* X_0 cambia nuovamente e si riporta a 0, mentre sull'ingresso del *flip-flop* di destra si ha ora $J=K=1$, che fa commutare lo stato finale X_1 da 0 a 1; il terzo stato è dunque 01. Nell'ultimo passo c'è il nuovo cambio di X_0 , che torna a 1, con $J=K=0$ nel *flip-flop* di destra, che non porta variazioni all'uscita X_1 , che rimane quindi a 1; il quarto stato è dunque 11. A seguito di un nuovo segnale di *clock* X_0 torna a 0 e X_1 cambia stato, riportandosi a 0. E il ciclo ricomincia. Il contatore può essere facilmente esteso a 3 bit introducendo una terza variabile X_2

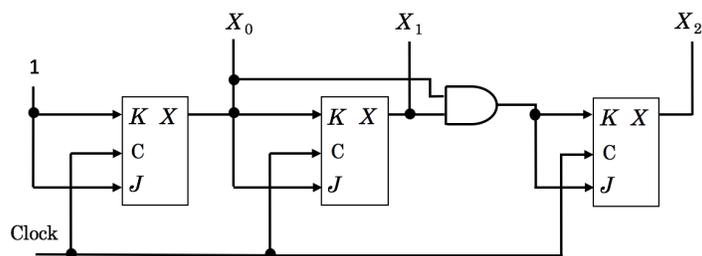


Figura 6.19: Contatore a 3 bit che passa attraverso gli stati 000, 100, 010, 110, 001, 101, 011, 111 per le variabili $X_0X_1X_2$

anche senza fare un progetto sistematico basato sulle tavole di verità e sulle mappe di Karnaugh; a tal scopo basta considerare che per far commutare X_2 da 0 a 1 è necessario che sia $X_0 = X_1 = 1$. Quando ciò avviene X_2 rimane a 1, finché non si ripresenta nuovamente la coppia $X_0 = X_1 = 1$; a quel punto X_2 torna a 0. Tutto ciò viene realizzato con una porta AND, che consente di costruire il contatore a 3 bit di figura [6.19](#).

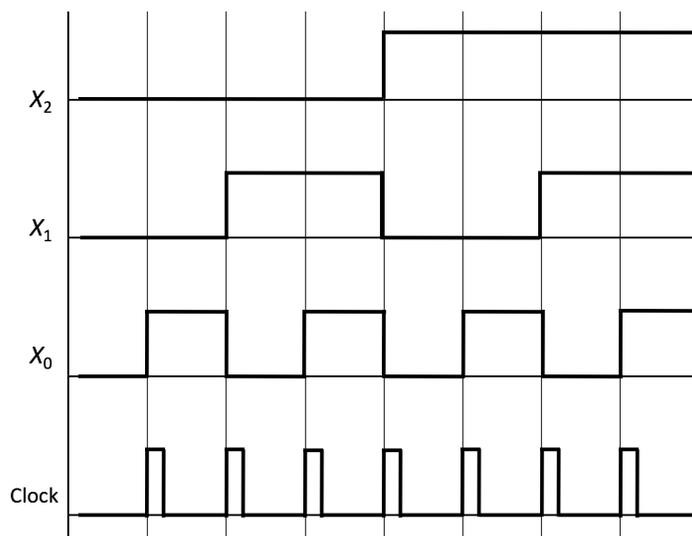


Figura 6.20: Andamento dei segnali associati alle variabili $X_0X_1X_2$ del circuito di figura [6.19](#) le cui variazioni sono controllate dall'impulso di *Clock*