

Capitolo 11

Il controllo di piccoli sistemi

Nel capitolo precedente abbiamo visto come un sistema possa essere decomposto in due reti sequenziali sincrone dette parte operativa e parte controllo. Abbiamo descritto la parte operativa a livello di registri e moduli di interconnessione, e il suo funzionamento comandato dalle μ istruzioni generate dalla parte controllo.

Studieremo in questo capitolo la struttura interna della parte controllo per sistemi di piccole dimensioni, trattando tale parte con le metodologie proprie delle reti sequenziali. In particolare svilupperemo il progetto della parte controllo con la tecnica di sintesi delle reti sequenziali, partendo dal diagramma degli stati per giungere, attraverso la tabella delle transizioni, alle espressioni algebriche delle funzioni di prossimo stato e di uscita.

Se le dimensioni della rete sequenziale di controllo, e in particolare il suo numero di stati e di ingressi, crescono oltre un limite che renda intrattabile la sintesi classica, dovremo ricorrere ad altre tecniche di progetto. Questa situazione, che si riscontra tipicamente nello studio di un calcolatore visto come sistema, sarà affrontata nei prossimi capitoli.

11.1 Parte controllo autonoma

Come noto la parte controllo riceve dall'esterno i segnali I , che codificano le istruzioni, e β , che codificano le condizioni che provengono dalla parte operativa, ed emette i segnali α che compongono le μ istruzioni (fig. 11.1).

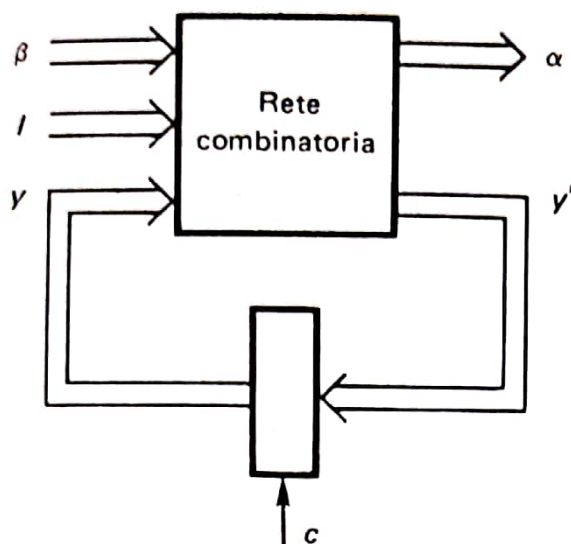


Figura 11.1

La parte controllo di un sistema vista come rete sequenziale.

Consideriamo per cominciare il caso limite in cui non vi siano istruzioni I né condizioni β . La parte controllo si riduce quindi a una rete sequenziale autonoma e il suo funzionamento diviene periodico (§ 6.1); il sistema nel suo complesso ripete allora indefinitamente una stessa sequenza di operazioni, cioè è caratterizzato da una sola μ sequenza ripetuta ciclicamente.

Lo studio dei sistemi suddetti permette di dirigere l'attenzione sull'esecuzione di singole μ sequenze, e come tale è utile per comprendere alcuni concetti di base. Condurremo questo studio attraverso la formulazione e la soluzione completa di un problema, e lo estenderemo in seguito al caso reale in cui siano presenti istruzioni I , che consentano la selezione di diverse μ sequenze, o condizioni β provenienti dalla parte operativa.

Problema 11.1

Progettare un sistema che contiene una memoria M con parole di 16 bit; acquisisce dati di 8 bit (1 byte), in sincronismo con l'impulso c , da un canale esterno DATI; memorizza tali dati a due a due in parole di memoria consecutive, a partire dall'indirizzo specificato, nell'istante iniziale, da un canale esterno IND.

La parte operativa del sistema potrà essere composta come è indicato in figura 11.2. La memoria M ha la struttura usuale (§ 6.3), e vi si scrive con il segnale $S=1$. Un selettore commuta il canale DATI

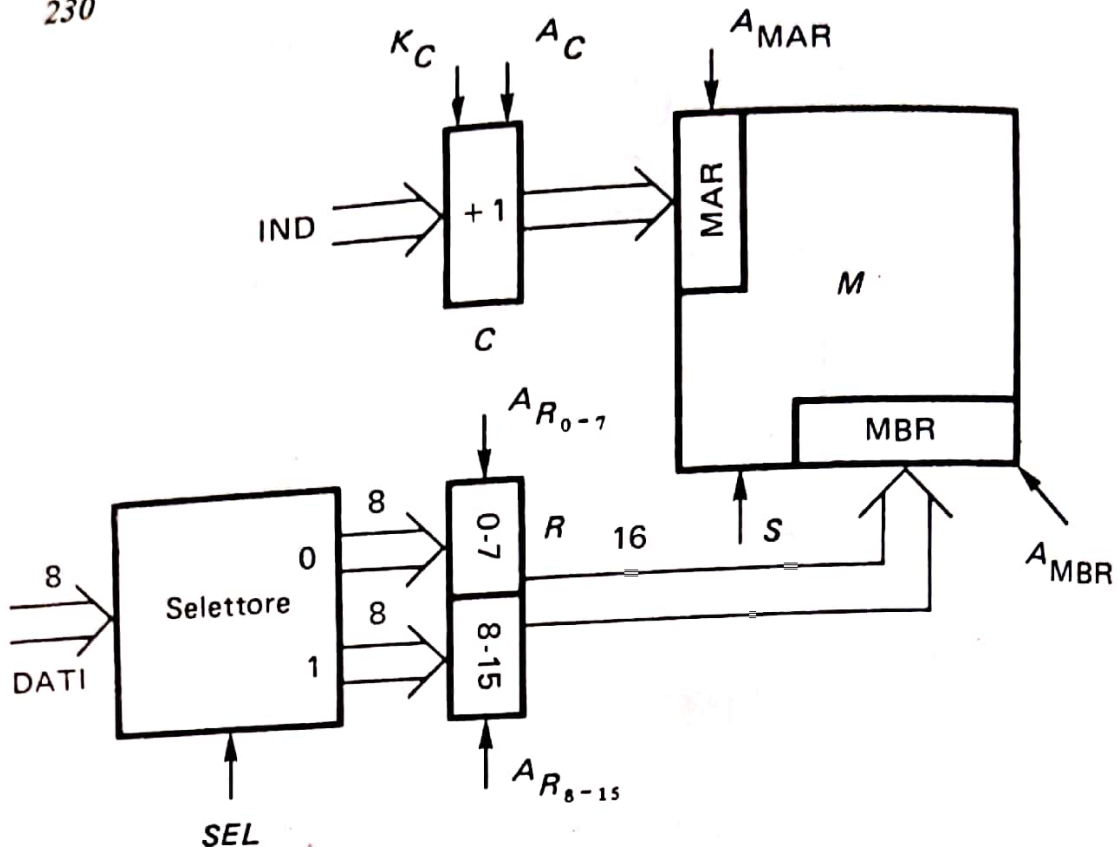


Figura 11.2

Problema 11.1: la parte operativa.

sulla prima o seconda metà di un registro R di 16 bit, necessario poiché non si dispone di accesso separato alle due metà del registro MBR.¹ Il canale IND è collegato a un registro contatore C , nel quale si producono gli indirizzi via via crescenti per i dati.

La μ sequenza che descrive il funzionamento della parte operativa è la seguente:

- 1: $\text{DATI} \rightarrow R_{0-7}$, $\text{IND} \rightarrow C$;
- 2: $\text{DATI} \rightarrow R_{8-15}$;
- 3: $R \rightarrow \text{MBR}$, $C \rightarrow \text{MAR}$, $\text{DATI} \rightarrow R_{0-7}$;
- 4: $\text{MBR} \rightarrow M[\text{MAR}]$, $\text{INCR}(C) \rightarrow C$, $\text{DATI} \rightarrow R_{8-15}$, **goto** 3

Le μ operazioni 1 e 2 caricano inizialmente i registri R e C ; quindi il funzionamento diviene ciclico, e ripete indefinitamente le μ operazioni 3 e 4 nelle quali si scrive un dato in memoria e si ricarica contemporaneamente il registro R .

¹ Il caricamento dei dati nelle due metà di R sarà comandato da due segnali indipendenti per l'abilitazione di R_{0-7} e R_{8-15} : con tale schema il selettore non è strettamente necessario, ma può essere ugualmente impiegato per evitare la duplicazione del segnale all'uscita del canale.

Si noti che il ciclo comprende due μ operazioni, quindi il funzionamento è correttamente sincronizzato con l'arrivo dei byte dal canale DATI, che in due intervalli elementari τ completa l'invio di una parola. Se la parte operativa non consentisse di caricare una parola in memoria in tempo 2τ , si perderebbero dei dati di ingresso.

La sequenzializzazione delle operazioni nella μ sequenza è in gran parte obbligata, ma vi sono alcuni gradi di libertà sulla contemporaneità di assegnamenti indipendenti: per esempio $\text{INCR}(C) \rightarrow C$ potrebbe essere spostato dalla μ operazione 4 alla 3.

Le μ istruzioni del sistema sono le seguenti (i segnali di comando sono rinominati $\alpha_1, \dots, \alpha_8$):

	$A_{R_{0-7}}$	$A_{R_{8-15}}$	A_C	A_{MAR}	A_{MBR}	K_C	S	SEL	
	α_1	α_2	α_3	α_4	α_5	α_6	α_7	α_8	
1:	1	0	1	0	0	0	0	0	
2:	0	1	0	0	0	—	0	1	
3:	1	0	0	1	1	—	0	0	
4:	0	1	1	0	0	1	1	1	goto 3

La parte controllo del sistema emetterà la successione di μ istruzioni 1, 2, 3, 4, ripetendo ciclicamente 3, 4. Possiamo quindi progettare la parte controllo, cominciando a tracciare il diagramma degli stati e l'equivalente tabella di flusso (fig. 11.3). Notiamo che a è lo stato iniziale, e il diagramma è quasi periodico.

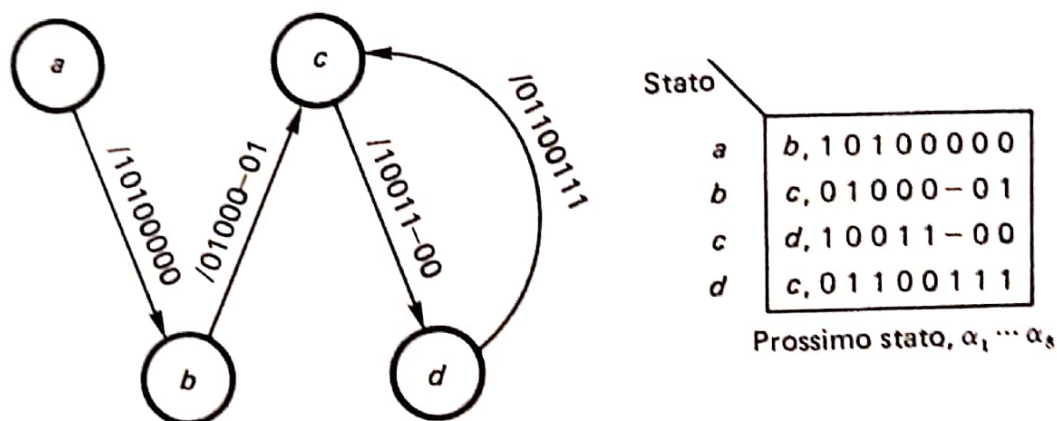


Figura 11.3

Problema 11.1: diagramma degli stati e tabella di flusso della parte controllo.

$y_1 y_2$		
a:	0 0	0 1, 1 0 1 0 0 0 0 0
b:	0 1	1 1, 0 1 0 0 0 - 0 1
c:	1 1	1 0, 1 0 0 1 1 - 0 0
d:	1 0	1 1, 0 1 1 0 0 1 1 1

$y'_1 y'_2, \alpha_1 \dots \alpha_8$

$$y'_1 = y_1 + y_2$$

$$\alpha_1 = \bar{y}_1 \bar{y}_2 + y_1 y_2$$

$$\alpha_2 = \alpha_8 = \bar{y}_1 y_2 + y_1 \bar{y}_2$$

$$\alpha_3 = \bar{y}_2$$

$$\alpha_4 = \alpha_5 = y_1 y_2$$

$$\alpha_6 = \alpha_7 = y_1 \bar{y}_2$$

$$y'_2 = \bar{y}_1 + \bar{y}_2$$

Figura 11.4
Problema 11.1: tabella delle transizioni e funzioni.

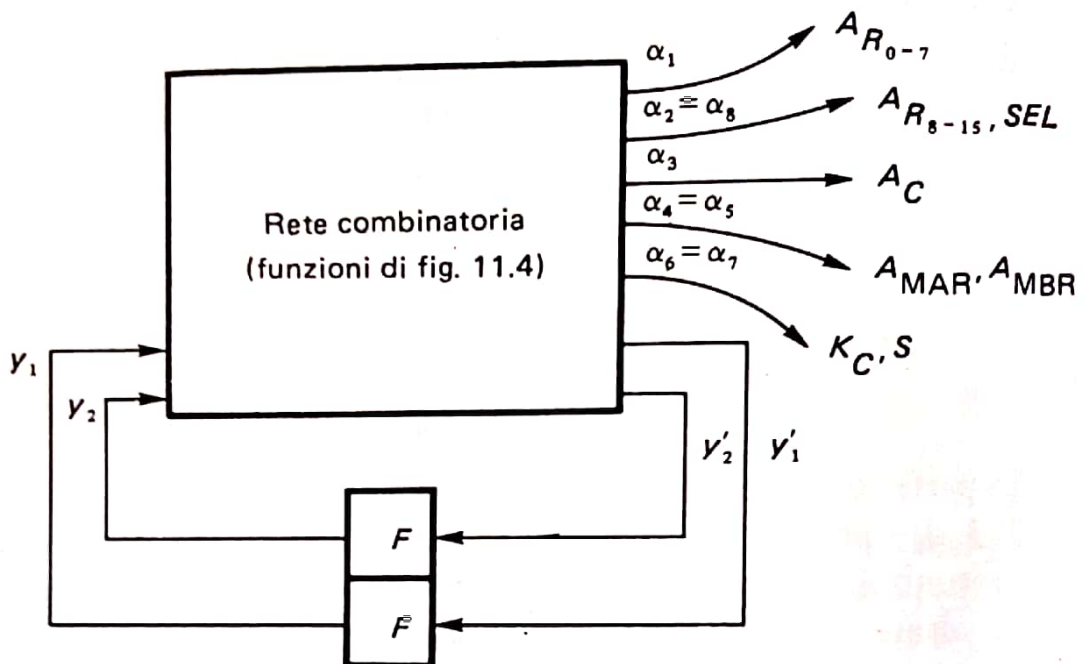


Figura 11.5
Problema 11.1: la parte controllo.

La tabella di flusso è minima; si ottiene da essa la tabella delle transizioni di figura 11.4, e da questa si ottengono le forme algebriche per le funzioni di prossimo stato e di uscita ivi indicate. Si noti che le funzioni di uscita sono di fatto cinque e non otto, perché alcune di esse sono uguali: ciò è ovvio in particolare per le $\alpha_2 \equiv A_{R_{8-15}}$ e $\alpha_8 \equiv SEL$ (vedi fig. 11.2); le non specificazioni per la α_6 sono state scelte in modo da avere $\alpha_6 = \alpha_7$.

La parte controllo è quindi indicata nella figura 11.5. Il progetto del sistema si conclude connettendo le uscite α della parte controllo ai corrispondenti punti da comandare nella parte operativa (fig. 11.2), e introducendo nello schema le connessioni per distribuire l'impulso di sincronismo su tutti i registri.

11.2 Parte controllo e istruzioni

Nel paragrafo precedente abbiamo esaminato un sistema con parte controllo autonoma. Vediamo ora come lo studio debba essere condotto se sono presenti istruzioni applicate agli ingressi I della parte controllo (fig. 11.1), tuttavia ancora priva di ingressi di condizione β .

Cominciamo con l'ampliare il precedente problema 11.1, trasformandolo nel nuovo:

Problema 11.2

Progettare un sistema con istruzioni I di un bit, che esegua le operazioni del problema 11.1 per $I=1$, e arresti le operazioni per $I=0$.

Utilizzando la parte operativa del problema 11.1, possiamo direttamente definire la tabella di flusso della parte controllo, che ha ora due colonne, relative ai due stati di ingresso $I=0$ e $I=1$ (fig. 11.6). Per $I=0$ la rete rimane stabilmente nello stato iniziale a , generando la μ istruzione 00000-0-, che congela il contenuto di tutti i registri (il significato dei segnali α è lo stesso del problema 11.1). Per $I=1$ la rete inizia a operare dallo stato a , lungo la seconda colonna della tabella: tale colonna è identica alla tabella di flusso del problema 11.1.

L'istruzione $I=1$ termina quando si pone nuovamente $I=0$ dall'esterno. Abbiamo qui fatto l'ipotesi che questa interruzione debba essere eseguita nell'intervallo successivo all'arrivo dell'ultimo byte dal canale DATI (cioè quando la parte controllo si trova nello stato c : ciò spiega

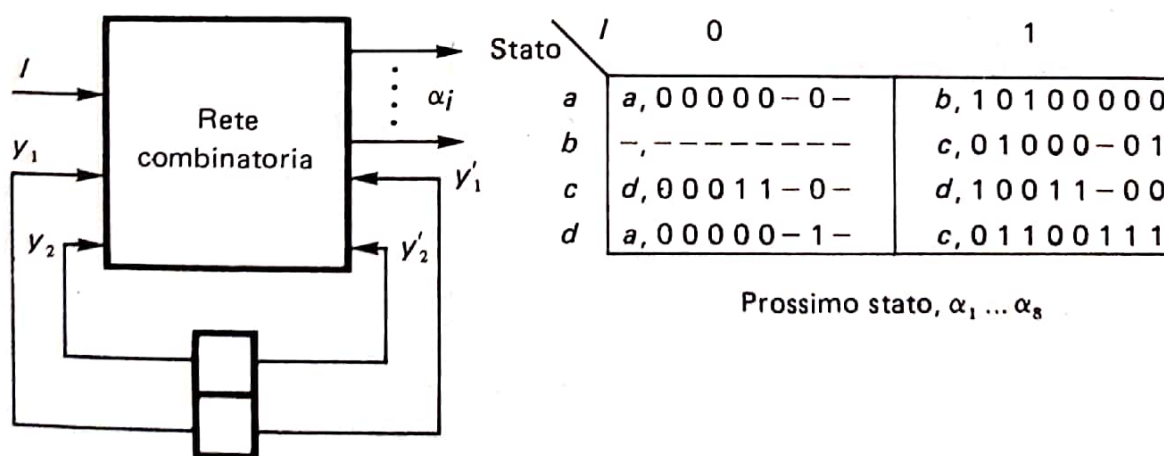


Figura 11.6

Problema 11.2: parte controllo e tabella di flusso.

le μ istruzioni contenute nelle celle di colonna 0 righe c, d , che comandano il trasferimento in memoria dell'ultima parola senza caricare ulteriori byte: il lettore è invitato a esaminare attentamente il contenuto di queste celle. La stessa ipotesi spiega anche le non specificazioni contenute nella cella di colonna 0 riga b , che non può mai essere raggiunta. Studiamo ora un nuovo problema.

Problema 11.3

Progettare un sistema che contiene un registro P a complementazione, un registro Q , e una ALU del tipo di quella di figura 10.5. Il sistema opera su numeri interi in complemento a 2, ed esegue le istruzioni di due bit:

$I_1 I_0$

0 0 : ALT

0 1 : $-P - Q \rightarrow P$

1 1 : $Q - P \rightarrow Q$

La parte operativa del sistema può avere la forma mostrata nella figura 11.7, congruente con le μ sequenze, e corrispondenti μ istruzioni, indicate dopo la figura, che sfruttano le operazioni della ALU.

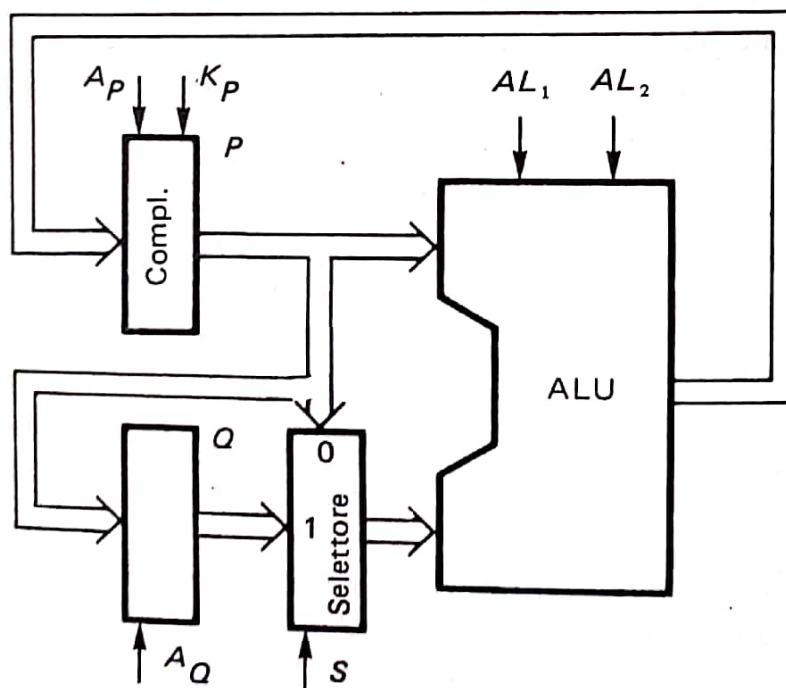


Figura 11.7

Problema 11.3: parte operativa.

istruzione	μ sequenze	$A_P \ A_Q \ K_P \ AL_1 \ AL_2 \ S$					
		α_1	α_2	α_3	α_4	α_5	α_6
0 0	ϕ	0	0	—	—	—	—
0 1	1: $P + Q \rightarrow P$;	1	0	0	1	0	1
	COMP(P) $\rightarrow P$;	1	0	1	—	—	—
	$P + 1 \rightarrow P$, goto 1	1	0	0	0	1	0 goto 1
1 1	1: $P \rightarrow Q, Q \rightarrow P$;	1	1	0	0	0	1
	$P - Q \rightarrow P$;	1	0	0	1	1	1
	$P \rightarrow Q, Q \rightarrow P$, goto 1	1	1	0	0	0	1 goto 1

Sono qui necessari alcuni commenti.

Istruzione 01: il calcolo di $-P - Q$ è trasformato in $-(P + Q)$, ove il cambio di segno è eseguito mediante complementazione e incremento di 1; tale incremento è eseguito con la ALU, facendo transitare il valore di P nell'ingresso inferiore.

Istruzione 11: a causa delle connessioni con la ALU, il calcolo di $Q - P$ richiede un preventivo scambio del contenuto dei registri, ove il trasferimento $Q \rightarrow P$ è realizzato attraverso la ALU. La terza μ operazione contiene nuovamente il trasferimento $Q \rightarrow P$ per ripristinare il valore originale di P .

Termine delle istruzioni: la frase go che chiude le μ sequenze indica che queste devono essere ripetute indefinitamente finché l'istruzione corrispondente è mantenuta agli ingressi della parte controllo.

Progettiamo ora la parte controllo nell'ipotesi che un'istruzione possa essere sostituita con un'altra solo quando è terminata la sua esecuzione. La tabella di flusso è quella di figura 11.8 ove, per ogni istruzione, lo stato iniziale è a e la sequenza si sviluppa lungo una colonna, in stati ad essa dedicati. Le istruzioni si possono cambiare solo quando la parte controllo si trova nello stato a : ciò spiega il significato delle caselle completamente non specificate.

Le classi di compatibilità di stati della tabella di flusso sono mostrate nella figura 11.8; una scelta di classi coprente, chiusa e minima, e la conseguente tabella ridotta, sono mostrate nella figura 11.9. La realizzazione della parte controllo a partire da quest'ultima tabella è lasciata al lettore.

		$I_1 I_0$		0 0		0 1		1 1		Classi di compatibilità (a), (bd), (be), (cd), (ce)
Stato										
a		a, 0 0		b, 1 0 0 1 0 1		d, 1 1 0 0 0 1				
b		-, -		c, 1 0 1 - - -		-, -				
c		-, -		a, 1 0 0 0 1 0		-, -				
d		-, -		-, -		e, 1 0 0 1 1 1				
e		-, -		-, -		a, 1 1 0 0 0 1				

Prossimo stato, $\alpha_1 \dots \alpha_6$

Figura 11.8

Problema 11.3: tabella di flusso per la parte controllo.

		$I_1 I_0$		0 0		0 1		1 1	
Scelta minima		Stato							
ϵ : (a)	ϵ			ϵ , 0 0		γ , 1 0 0 1 0 1		γ , 1 1 0 0 0 1	
γ : (bd)	γ			-, -		δ , 1 0 1 - - -		δ , 1 0 0 1 1 1	
δ : (ce)	δ			-, -		ϵ , 1 0 0 0 1 0		ϵ , 1 1 0 0 0 1	

Prossimo stato, $\alpha_1 \dots \alpha_6$

Figura 11.9

Problema 11.3: tabella di flusso minima.

11.3 Parte controllo con ingressi di condizione

Siamo infine giunti a studiare la struttura più generale per un sistema, nella quale la parte controllo è a sua volta guidata da segnali β inviati dalla parte operativa.

Un primo semplice esempio, che non include istruzioni esterne, si può ancora costruire come estensione del problema 11.1.

Problema 11.4

Progettare un sistema che esegue le operazioni del problema 11.1, aggiungendovi un terzo canale di ingresso NUM che all'inizio comunica il numero (>0) di parole di due byte da memorizzare. Eseguita tale memorizzazione il sistema deve arrestarsi.

Nella parte operativa della figura 11.2 si inserisce un nuovo registro contatore BLOC, caricato inizialmente dal canale NUM e successivamente decrementato per ogni parola memorizzata. Quando il valore di BLOC raggiunge lo zero numerico, cioè l'OR di tutti i bit di BLOC è

zero, l'operazione si arresta (si veda la condizione simile incontrata nella μ sequenza 1 del § 10.2, fig. 10.2).

La nuova struttura del sistema è riportata nella figura 11.10. La μ sequenza di comando è la seguente (si confronti con la μ sequenza per il problema 11.1):

```

1: IND  $\rightarrow$  C, DATI  $\rightarrow$   $R_{0-7}$ , NUM  $\rightarrow$  BLOC;
2: DATI  $\rightarrow$   $R_{8-15}$ , DECR(BLOC)  $\rightarrow$  BLOC;
3: if OR(BLOC) = '1'
    then R  $\rightarrow$  MBR, C  $\rightarrow$  MAR, DATI  $\rightarrow$   $R_{0-7}$ ;
    4: MBR  $\rightarrow$   $M[MAR]$ , INCR(C)  $\rightarrow$  C, DATI  $\rightarrow$   $R_{8-15}$ ,
        DECR(BLOC)  $\rightarrow$  BLOC, goto 3
    else R  $\rightarrow$  MBR, C  $\rightarrow$  MAR;
    5: MBR  $\rightarrow$   $M[MAR]$ 
fi
  
```

La tabella di flusso della parte controllo è riportata nella figura 11.11: i segnali $\alpha_1, \dots, \alpha_8$ hanno lo stesso significato del problema 11.1, ed esistono due nuovi segnali $\alpha_9 = A_{\text{BLOC}}$ e $\alpha_{10} = K_{\text{BLOC}}$ per comandare il caricamento e il decremento del registro BLOC.

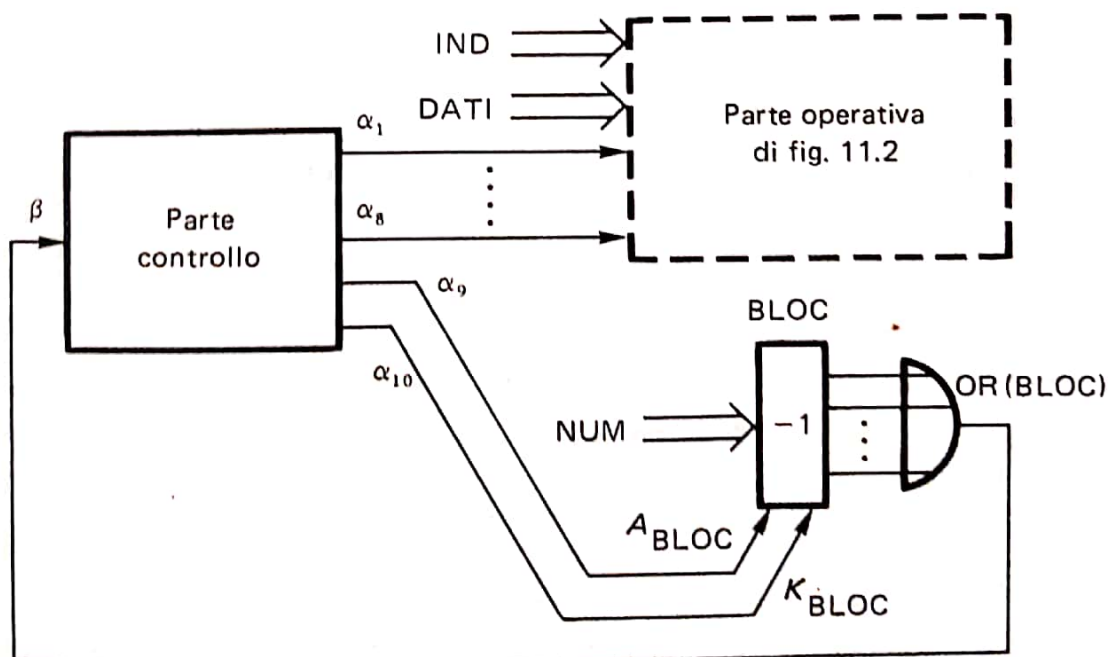


Figura 11.10

Problema 11.4: schema complessivo della rete.

Nello stato iniziale a il segnale $\beta = \text{OR}(\text{BLOC})$ può valere 0 o 1 in conseguenza di elaborazioni precedenti. Caricato $\text{NUM}(>0)$ in BLOC , si ha certamente $\beta=1$ nello stato b , e di qui si procede con la μ sequenza relativa a $\beta=1$ nella seconda colonna, finché si raggiunge la condizione $\beta=0$: tale condizione può essere raggiunta solo quando la rete si trova nello stato c , poiché i comandi $\alpha_9 = \alpha_{10} = 1$ per il decremento di BLOC vengono inviati portando la rete in c (vedi tabella di flusso). Di qui l'elaborazione si conclude con la μ sequenza della prima colonna, che trasferisce in memoria l'ultima parola e termina con lo stato di blocco e .

La tabella di figura 11.11 può essere ridotta a quattro stati, poiché $b \approx e$, e si può completare il progetto della parte controllo con metodo standard.

Vediamo infine un esempio in cui la parte controllo riceve ingressi di istruzione e condizione.

Problema 11.5

Progettare un sistema che contiene due registri a spostamento destro P e Q , opera su interi, ed esegue le istruzioni di un bit:

$I=0$: ALT

$I=1$: if ($P + Q$ non dà supero) then $P + Q \rightarrow P$
 else $P/2 + Q/2 \rightarrow P$

β Stato	0	1
a	$b, 1010000010$	$b, 1010000010$
b	-, - - - - - - - - -	$c, 01000-0111$
c	$d, 00011-0-0-$	$d, 10011-000-$
d	$e, 00000-1-0-$	$c, 0110011111$
e	$e, 00000-0-0-$	-, - - - - - - - - -

Prossimo stato, $\alpha_1 \dots \alpha_{10}$

Significato dei segnali:

α_1	α_2	α_3	α_4	α_5	α_6	α_7	α_8	α_9	α_{10}	β
$A_{R_{0-7}}$	$A_{R_{8-15}}$	A_C	A_{MAR}	A_{MBR}	K_C	S	SEL	A_{BLOC}	K_{BLOC}	$\text{OR}(\text{BLOC})$

Figura 11.11

Problema 11.4: tabella di flusso per la parte controllo.

La parte operativa del sistema dovrà includere una ALU per eseguire le addizioni richieste dall'istruzione $I=1$. La ALU ha in genere n uscite R_0, \dots, R_{n-1} per il risultato, più un'uscita R_n di supero (vedi fig. 3.15: l'operazione eseguita dalla ALU è in supero di capacità se e solo se $R_n=1$): il bit R_n sarà allora impiegato per verificare la condizione contenuta nell'istruzione $I=1$. Per quanto discusso alla fine del paragrafo 10.3, sarà opportuno trasferire il bit R_n in un flip-flop S in un primo intervallo di tempo, e verificare il valore di S nell'intervallo successivo.

La parte operativa del sistema è mostrata nella figura 11.12. Il flip-flop S è dotato di un ingresso di azzeramento Z_S ($A_S=Z_S=1$ azzerà il contenuto di S , vedi esercizio 11.5): per chiarezza di funzionamento ammettiamo di trovare S azzerato all'inizio delle operazioni, e lo azzeriamo nuovamente alla fine.

Le μ sequenze e relative μ istruzioni sono le seguenti:

istruzioni	μ sequenze
0	ϕ
1	1: $(P+Q)_n \rightarrow S$; 2: if $S='0'$ then $P+Q \rightarrow P$, goto 1 else 3: $SD(P) \rightarrow P$, $SD(Q) \rightarrow Q$, $'0' \rightarrow S$; 4: $P+Q \rightarrow P$, goto 1 fi

μ operazioni	μ istruzioni								
	A_P	A_Q	A_S	K_P	K_Q	Z_S	AL_1	AL_2	
	α_1	α_2	α_3	α_4	α_5	α_6	α_7	α_8	
ϕ	0	0	0	—	—	—	—	—	
1:	0	0	1	—	—	0	1	0	
2: if $S='0'$ then	1	0	0	0	—	—	1	0	goto 1
3: else	1	1	1	1	1	1	—	—	
4:	1	0	0	0	—	—	1	0	goto 1 fi

La frase **goto 1** che appare alla fine delle μ sequenze relative a $I=1$ comanda l'iterazione di questa istruzione, finché non sia interrotta dalla $I=0$. Facendo l'ipotesi che la $I=0$ possa intervenire solo quando

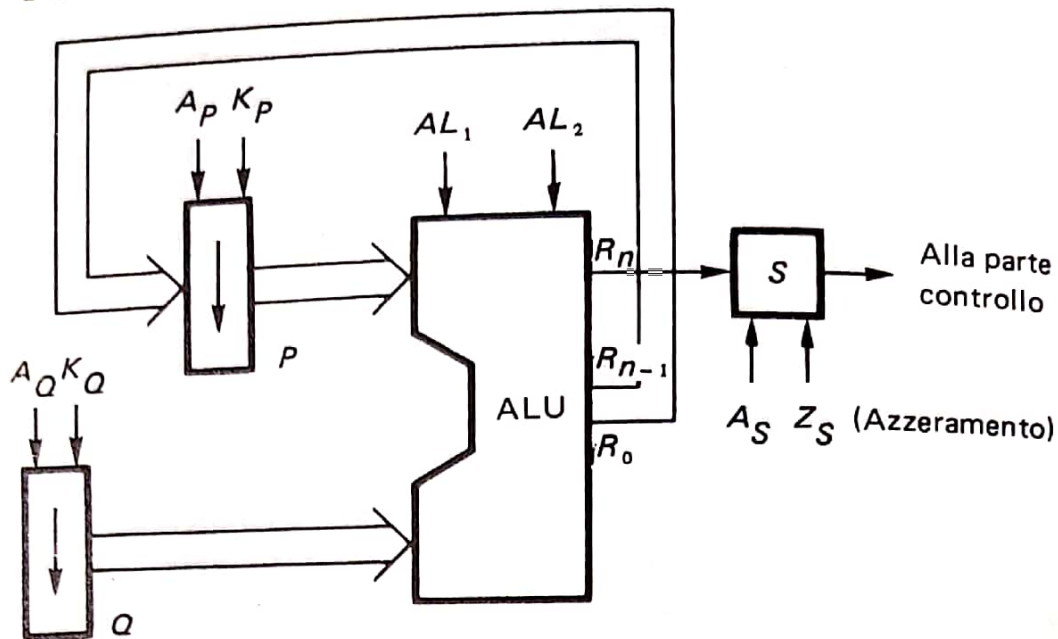


Figura 11.12

Problema 11.5: parte operativa.

è terminata un'esecuzione della $I=1$, la tabella di flusso della parte controllo è quella di figura 11.13, ove, nella mezza tabella relativa a $I=0$, è specificato il contenuto della sola casella iniziale.

Il lettore è invitato a considerare attentamente il contenuto della tabella, e in particolare la colonna 11, ove le caselle nelle righe a e c contengono non specificazioni perché non possono essere mai raggiunte: la prima per l'ipotesi che il flip-flop S sia inizialmente azzerato ($S=0$ nello stato a), e la seconda perché la rete si porta nello stato c azzerando il flip-flop (vedi i comandi in colonna 11 riga b).

Si ha evidentemente $b \approx c$, e la tabella di flusso si riduce a quella di figura 11.14. Questa tabella è interessante perché, pur avendo solo due stati, consente l'esecuzione della μ sequenza di tre passi relativa alla

Stato \ I S				
	0 0	0 1	1 1	1 0
a	$a, 000---$	$-, ---$	$-, ---$	$b, 001--010$
b	$-, ---$	$-, ---$	$c, 111111--$	$a, 1000--10$
c	$-, ---$	$-, ---$	$-, ---$	$a, 1000--10$

Prossimo stato, $\alpha_1 \dots \alpha_8$

Figura 11.13

Problema 11.5: tabella di flusso per la parte controllo.

All'inizio Q contiene il moltiplicando; P contiene il moltiplicatore; ACC contiene '0'. Detta n la lunghezza di parola, l'algoritmo è:

All'inizio Q contiene il moltiplicando; P contiene il moltiplicatore; ACC contiene '0'. Detta n la lunghezza di parola, l'algoritmo è:

$$\text{ripeti } n \text{ volte } \begin{cases} \text{if } P_0 = 1 \text{ then } \text{ACC} + Q \rightarrow \text{ACC}; \\ \text{SD}(\text{ACC}|P) \rightarrow \text{ACC}|P; \end{cases}$$

Alla fine delle operazioni $ACC|P$ contiene il prodotto, formato da $2n$ bit, mentre si perde il moltiplicatore.

Disponendo di un ulteriore registro CON che contiene il valore di n , e nell'ipotesi che le addizioni non diano mai supero, progettare la parte controllo che guida la moltiplicazione.

11.3 Progettare un sistema che contiene due registri ACC e Q ed esegue le istruzioni di due bit:

 $I_1 \quad I_0$

0 0 : ALT

0 1 : if (ACC + Q dà supero) then ALT

else $ACC + Q \rightarrow ACC$

$$1 \quad 1 : \text{ACC} - Q + 1 \rightarrow Q$$

11.4 In una memoria M è memorizzata una lista L . Un elemento E di L occupa due celle consecutive di M , che contengono il campo informazione INF e il campo puntatore PUN di E . L'indirizzo di E è quello della prima cella. Nessun elemento può occupare la cella di indirizzo 0. Elementi consecutivi E_1, E_2 occupano coppie di celle arbitrarie: l'indirizzo di E_2 è contenuto nel campo PUN di E_1 . L'indirizzo dell'elemento iniziale di L è contenuto in un registro IND, e l'elemento finale contiene '0' nel campo PUN.

Progettare un sistema per la ricerca in L di un elemento K proveniente da un canale esterno. Se $K \in L$, l'indirizzo di memoria di K si scrive in un registro RIS; se $K \notin L$, si scrive '0' in RIS.

11.5 Per memorizzare bit di condizione è in genere utile disporre di un flip-flop di tipo *FAC*, dotato di un ulteriore ingresso di azzeramento *Z*. Il funzionamento è:

$A = 0$ resta nello stato presente;

$$A = 1 \begin{cases} Z = 0 & \text{carica dall'esterno;} \\ Z = 1 & \text{carica '0';} \end{cases}$$

Progettare il flip-flop (vedi § 4.3).