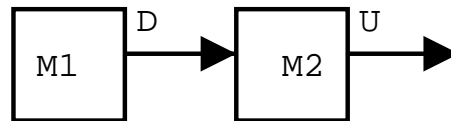


Realizzazione di una macchina calcolatrice

In questo lavoro si è cercato di realizzare una macchina calcolatrice, che avuti in ingresso due numeri (N_1 , N_2) ed una operazione (Op) fornisce in uscita $N_1 Op N_2$. La macchina acquisisce i dati da una tastiera esadecimale. Questa macchina tratta numeri di una cifra codificati con il codice 8421, quindi il numero è codificato su 4 bit.

Schema della macchina (analisi e completamento delle specifiche)

Ad una prima analisi la macchina calcolatrice M è composta da una macchina M_1 che si occupa dell'acquisizione dei dati, e da una macchina M_2 che esegue il calcolo richiesto:



La macchina deve avere:

- due registri dati A , B per la memorizzazione dei numeri su cui operare
- un registro soluzione
- un addizionatore di n bit (ADD)
- un sottrattore di n bit (SUB)
- un moltiplicatore ad n bit (non realizzato)
- un divisore (non realizzato)

la macchina, inoltre, fornisce un segnale binario che segnali un overflow e/o borrow.

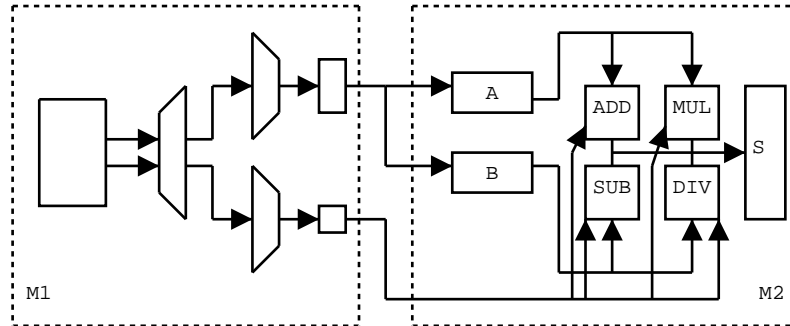
La macchina acquisisce i dati dalla tastiera esadecimale nella macchina M_1 , e genera il risultato nella macchina M_2 . La tastiera esadecimale presenta 16 tasti, 10 numerici (0,...,9) e 6 alfabetici. Si effettuano le seguenti posizioni. Si associano ai tasti numerici i corrispondenti valori numerici, mentre si associano ai tasti alfabetici i seguenti significati:

- A addizione (+)
- B sottrazione (-)
- C moltiplicazione (*)
- D divisione (/)
- E uguale (=)
- F reset

La tastiera esadecimale è fornita di 4 uscite, sulle quali è reso disponibile il codice del tasto premuto (codice 8421). La prima operazione da compiere è, quindi, la decodifica del codice per capire se è stato premuto un tasto numerico o un tasto alfabetico. La prima macchina è formata da una rete combinatoria che effettua la decodifica del codice 8421. Se si riconosce un tasto numerico,

la rete ricodifica il segnale e lo invia ai registri per la memorizzazione, se il tasto non è numerico, allora si è premuto un tasto "operazione" o il tasto "=" o il reset.

Detto ciò si può sviluppare con ulteriore dettaglio lo schema precedente.



Per i registri della macchina si è scelto di utilizzare dei registri a scorrimento (componente commerciale 74165). Questi registri presentano per i dati un ingresso parallelo, e una uscita seriale. E' possibile, quindi, caricarli prelevando il dato decodificato dalla rete e con il bit seriale è possibile alimentare la macchina che si occuperà di svolgere l'operazione chiesta. Il 74165 è dotato di 8 ingressi per il dato parallelo (A-H) e un ingresso seriale. Ha un ingresso per il clock, un ingresso SH/LD' e un ingresso CLK INH. In uscita sono disponibili QH e il suo negato. La macchina calcolatrice sfrutterà le caratteristiche tecniche del 74165 per compiere alcune delle sue funzioni.

Il bit SH/LD' determina il funzionamento del componente, se SH/LD' è alto nel registro è abilitato lo shift, e in uscita ad ogni ciclo di clock verrà fornito uno degli otto valori memorizzati. Se SH/LS' è basso, allora il registro caricherà il valore che si trova al suo ingresso.

Il bit CLK INH è un inibitore del clock, il clock e il clock inhibit sono collegati tramite una porta nor, quindi il clock sarà annullato se entrambi i segnali sono alti.

Rete combinatoria

La tabella di decodifica del codice 8421 fornito dalla tastiera esadecimale è la seguente:

La prima colonna contiene il simbolo presente sulla tastiera, le colonne c_i contengono il codice 8421 associato al tasto, sulle colonne f_i è presente il codice di decodifica (sono stati tralasciati gli zeri), nelle colonne x_i è riportato il modo in cui è stato codificato l'ingresso ottenuto dalla tastiera (codice utilizzato come ingresso all'automa).

	c3	c2	c1	c0	f0	f1	f2	f3	f4	f5	f6	f7	f8	f9	f10	f11	f12	f13	f14	f15
0	0	0	0	0	1															
1	0	0	0	1		1														
2	0	0	1	0			1													
3	0	0	1	1				1												
4	0	1	0	0					1											
5	0	1	0	1						1										
6	0	1	1	0							1									
7	0	1	1	1								1								
8	1	0	0	0									1							
9	1	0	0	1										1						
A	1	0	1	0											1					
B	1	0	1	1												1				
C	1	1	0	0													1			
D	1	1	0	1														1		
E	1	1	1	0															1	
F	1	1	1	1																1

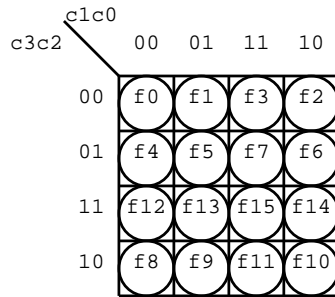
	c3	c2	c1	c0	x2	x1
0	0	0	0	0		1
1	0	0	0	1		1
2	0	0	1	0		1
3	0	0	1	1		1
4	0	1	0	0		1
5	0	1	0	1		1
6	0	1	1	0		1
7	0	1	1	1		1
8	1	0	0	0		1
9	1	0	0	1		1
A	1	0	1	0	1	
B	1	0	1	1	1	
C	1	1	0	0	1	
D	1	0	0	1	1	
E	1	1	1	0	1	1
F	1	1	1	1		

Da queste tabelle si ricavano le seguenti espressioni per x_1 e x_2 :

$$x_1 = f_0 + f_1 + f_2 + f_3 + f_4 + f_5 + f_6 + f_7 + f_8 + f_9 + f_{14}$$

$$x_2 = f_{10} + f_{11} + f_{12} + f_{13} + f_{14}$$

e la mappa di Karnaugh



dalla quale si evincono le equazioni delle funzioni di decodifica:

$$f_0 = c_3'c_2'c_1'c_0'$$

$$f_1 = c_3'c_2'c_1'c_0$$

$$f_2 = c_3'c_2'c_1c_0'$$

$$f_3 = c_3'c_2'c_1c_0$$

$$f_4 = c_3'c_2c_1'c_0'$$

$$f_5 = c_3'c_2c_1'c_0$$

$$f_6 = c_3'c_2c_1c_0'$$

$$f_7 = c_3'c_2c_1c_0$$

$$f_8 = c_3c_2'c_1'c_0'$$

$$f_9 = c_3c_2'c_1'c_0$$

$$f_{10} = c_3c_2'c_1c_0'$$

$$f_{11} = c_3c_2'c_1c_0$$

$$f_{12} = c_3c_2c_1'c_0'$$

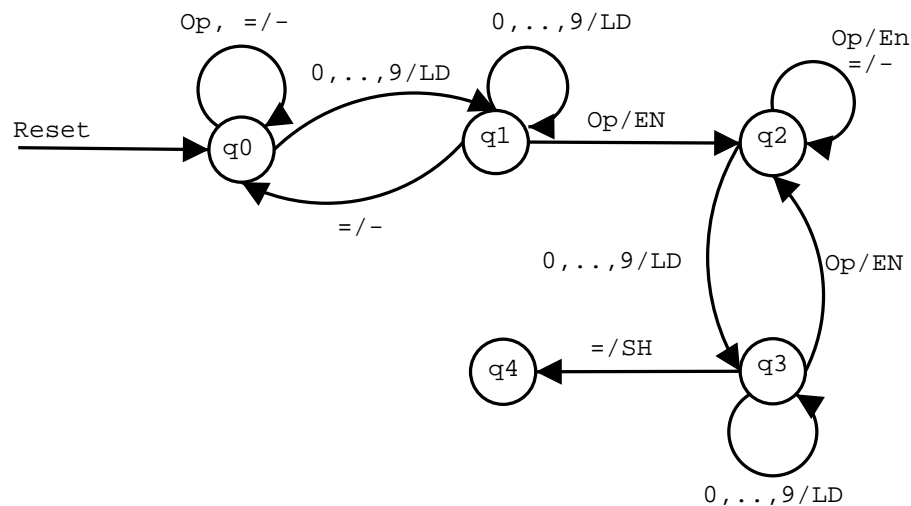
$$f_{13} = c_3c_2c_1'c_0$$

$$f_{14} = c_3c_2c_1c_0'$$

$$f_{15} = c_3c_2c_1c_0$$

Automa a stati finiti e funzionamento generale

La macchina alla partenza e dopo ogni reset si trova in uno stato stabile q_0 (attesa di un evento). Alla pressione di un tasto numerico, la macchina si sposta nello stato stabile q_1 , nel quale si acquisisce il numero digitato e lo si trasferisce nell'opportuno registro. Alla pressione di un tasto di operazione, la macchina si sposta nello stato q_2 e abilita la circuiteria idonea a compiere l'operazione chiesta. Dallo stato q_2 la macchina si sposta nello stato q_3 alla pressione di un nuovo tasto numerico. In q_3 la macchina acquisisce il nuovo dato che trasferirà nel secondo registro. Alla pressione del tasto $=$, la macchina si sposta in q_4 ed esegue l'operazione chiesta. La pressione del tasto F (reset) riporta la macchina nello stato q_0 .



La macchina ha gli ingressi $I = (i_1, i_2, i_3)$ dove:

$i_1 = 0, \dots, 9$

$i_2 = +, -, *, /$ (l'operazione da eseguire)

$i_3 = "="$ (inizio calcolo)

gli stati $Q = (q_0, q_1, q_2, q_3, q_4)$ dove:

q_0 = attesa evento

q_1 = acquisizione del primo numero

q_2 = acquisizione dell'operazione

q_3 = acquisizione del secondo numero

q_4 = esecuzione dell'operazione

le uscite $U = (u_1, u_2, u_3)$ con:

$u_1 = LD$ (carica il numero in un registro)

$u_2 = SH$ (esegui lo shift del registro)

$u_3 = EN$ (abilita la macchina che calcolerà l'operazione)

Tabella di stato

	i1	i2	i3
q0	q1/u1	q0/ -	q0/ -
q1	q1/u1	q2/u3	q0/ -
q2	q3/u1	q2/u3	q2/ -
q3	q3/u1	q2/u3	q4/u2
q4	q4/u2	q4/u2	q4/u2

Minimizzazione degli stati

La tabella non è completamente specificata, non è, allora, possibile determinare una macchina equivalente ma si deve procedere alla ricerca di una macchina compatibile con quella data, ma dotata di minori stati. Due stati per essere compatibili devono generare la stessa uscita per ogni sequenza di ingressi di lunghezza uno, e gli stati seguenti devono essere compatibili.

Esaminando le uscite si riscontra incompatibilità tra lo stato q4 (uscita u2) e tutti gli altri stati, q4 è quindi incompatibile con q0, q1, q2, q3.

La famiglia di insiemi compatibili è a questo passo (q4), (q0, q1, q2, q3).

Esamino le incompatibilità dovute a differenze negli stati seguenti.

(0, 1, 2, 3) con ingresso i1 ->(1, 1, 3, 3)

(0, 1, 2, 3) con ingresso i2 ->(1, 2, 2, 2)

(0, 1, 2, 3) con ingresso i3 ->(0, 0, 2, 4)

lo stato 4 è incompatibile con tutti gli altri, l'incompatibilità è generata dallo stato 3 che risulta essere incompatibile con gli altri stati.

Insiemi di compatibilità: (q4) (q3) (q0, q1, q2)

(0, 1, 2) con ingresso i1 ->(1, 1, 3)

3 è incompatibile con gli altri stati, l'incompatibilità è generata dallo stato 2 che, quindi, è a sua volta incompatibile:

Insiemi di compatibilità: (q4) (q3) (q2) (q0, q1)

(0, 1) con ingresso i1 ->(1, 1)

(0, 1) con ingresso i2 ->(1, 2)

2 è incompatibile con 1, l'incompatibilità è generata dallo stato 1 che risulta essere incompatibile con q0.

La famiglia di insiemi compatibili massimi è: (q0) (q1) (q2) (q3) (q4), e la macchina è minima.

Codifica degli stati e progetto combinatorio

La macchina dispone di 5 stati diversi, per poter codificare gli stati ho bisogno di $\log_2 5 = 3$ variabili binarie, gli ingressi sono 3, e per la loro codifica ho bisogno di 2 variabili binarie.

Codifica degli stati interni: dette y3 y2 y1 le variabili binarie effettuo la seguente codifica:

	y3	y2	y1
q0	0	0	0
q1	0	0	1
q2	0	1	0
q3	0	1	1
q4	1	0	0

Codifica degli ingressi: dette x1, x2 le variabili binarie ho la seguente codifica:

	x2	x1
0,..,9	0	1
+, -, *, /	1	0
=	1	1

L' uscita è codificata con le variabili binarie z1 e z2 nel seguente modo:

	z2	z1
LD	0	0
SH	1	0
EN	1	1

Per la realizzazione degli stati ho scelto flip flop di tipo JK, per la codifica dei 5 stati occorrono 3 flip flop. Sostituendo nella tabella degli stati le codifiche effettuate, si ottiene la seguente tabella

	x2x1		
y3y2y1	01	10	11
000	001/00	000/-	000/-
001	001/00	010/11	000/-
010	011/00	010/11	010/-
011	011/00	010/11	100/10
100	100/10	100/10	100/10

che riordinata e con l'inserimento della colonna 00 diventa:

	x2x1			
y3y2y1	00	01	11	10
000	—	001/00	000/-	000/-
001	—	001/00	000/-	010/11
011	—	011/00	100/10	010/11
010	—	011/00	010/-	010/11
100	—	100/10	100/10	100/10

Avendo scelto flip flop JK per la realizzazione degli elementi di memoria, la tabella dei segnali di posizionamento da utilizzare è la seguente:

	J	K
0->0	0	-
0->1	1	-
1->0	-	1
1->1	-	0

per il flip flop y1 si ha la seguente tabella di posizionamento

y3y2y1	00	01	11	10
000	-	1 -	0 -	0 -
001	-	- 0	- 1	- 1
011	-	- 0	0 1	- 1
010	-	1 -	0 -	0 -
100	-	0 -	0 -	0 -

per il flip flop y2

y ₃ y ₂ y ₁	00	01	11	10
000	-	0 -	0 -	0 -
001	-	0 -	0 -	1 -
011	-	- 0	- 1	- 0
010	-	- 0	- 0	- 0
100	-	0 -	0 -	0 -

per il flip flop y₃

y ₃ y ₂ y ₁	00	01	11	10
000	-	0 -	0 -	0 -
001	-	0 -	0 -	0 -
011	-	0 -	1 -	0 -
010	-	0 -	0 -	0 -
100	-	- 0	- 0	- 0

mappa per le uscite z₁ e z₂

		x ₂ x ₁				x ₂ x ₁			
		00	01	11	10	00	01	11	10
y ₂ y ₁	00	-		-	-	-			1
	01	-		-	1	-	-	-	-
	11	-		-	1	-	-	-	-
	10	-			1	-	-	-	-
		y ₃ =0				y ₃ =1			

		x ₂ x ₁				x ₂ x ₁			
		00	01	11	10	00	01	11	10
y ₂ y ₁	00	-		-	-	-	1	1	1
	01	-		-	1	-	-	-	-
	11	-		-	1	-	-	-	-
	10	-		1	1	-	-	-	-
		y ₃ =0				y ₃ =1			

mappa di posizionamento per J₁

		x ₂ x ₁				x ₂ x ₁			
		00	01	11	10	00	01	11	10
y ₂ y ₁	00	-	1			-			
	01	-	-	-	-	-			
	11	-	1			-			
	10	-	-	-	-	-			
		y ₃ =0				y ₃ =1			

mappa di posizionamento per K₁

		x2x1							
		00	01	11	10	00	01	11	10
y2y1	00	-	-	-	-	-	-	-	-
	01	-	-	1	1	-	-	-	-
	11	-	-	-	-	-	-	-	-
	10	-	-	1	1	-	-	-	-
		y3=0				y3=1			

mappa di posizionamento per J2

		x2x1							
		00	01	11	10	00	01	11	10
y2y1	00	-	-	-	-	-	-	-	-
	01	-	-	-	1	-	-	-	-
	11	-	-	-	-	-	-	-	-
	10	-	-	-	-	-	-	-	-
		y3=0				y3=1			

mappa di posizionamento per K2

		x2x1							
		00	01	11	10	00	01	11	10
y2y1	00	-	-	-	-	-	-	-	-
	01	-	-	-	-	-	-	-	-
	11	-	-	1	-	-	-	-	-
	10	-	-	-	-	-	-	-	-
		y3=0				y3=1			

mappa di posizionamento per J3

		x2x1							
		00	01	11	10	00	01	11	10
y2y1	00	-	-	-	-	-	-	-	-
	01	-	-	-	-	-	-	-	-
	11	-	-	1	-	-	-	-	-
	10	-	-	-	-	-	-	-	-
		y3=0				y3=1			

mappa di posizionamento per K3

		x2x1							
		00	01	11	10	00	01	11	10
y2y1	00	-	-	-	-	-			
	01	-	-	-	-	-			
	11	-	-	-	-	-			
	10	-	-	-	-	-			
		y3=0				y3=1			

Dalle mappe si ricavano le seguenti espressioni per i segnali di posizionamento e per le uscite:

$$J1=y3'x2'$$

$$K1=y3'x2$$

$$J2=y1y3'x1'$$

$$K2=x1x2y1y3'$$

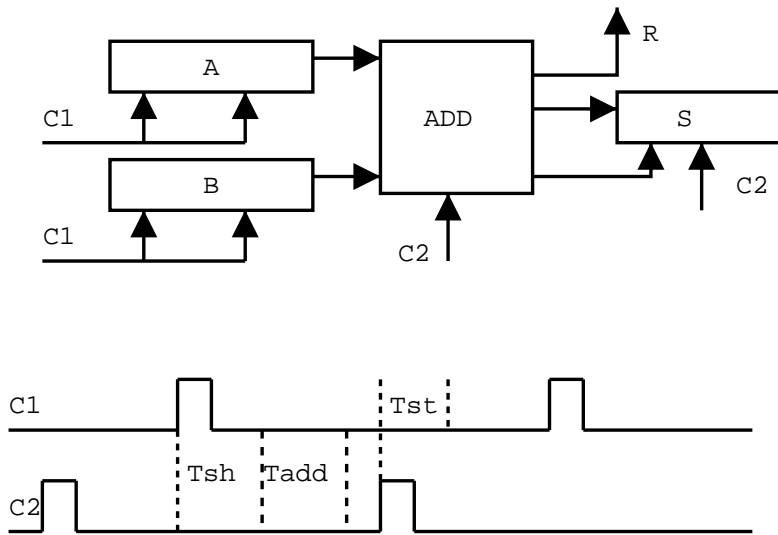
$$J3=y1y2y3'x1x2$$

$$K3=y3'$$

$$z1=y3'x1+y3x1'y1'$$

$$z2=y3'x2+y3$$

La macchina ADD

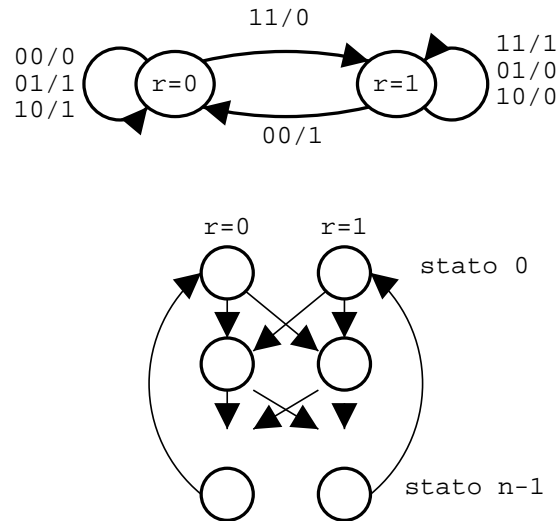


La macchina Add è alimentata dai bit meno significativi dei registri a scorrimento, e a sua volta, durante il calcolo della somma, alimenta con il bit calcolato il registro-soluzione. La macchina Add è una macchina sequenziale a sincronizzazione esterna.

L'addizionatore è composto da:

- un adder sequenziale (Full adder) di un bit
- un contatore modulo-n, impiegato per bloccare il ciclo di addizioni

L'addizionatore FA, è una macchina con due soli stati interni che coincidono con il riporto, due ingressi a, b (i bit da sommare), due uscite S, R (il bit somma e il bit riporto). Lo stato della macchina (riporto $r=0$, riporto $r=1$) è memorizzato tramite un flip flop D. Il Full adder opera secondo il seguente schema:



La macchina si trova in uno dei due stati $r=0$ ($r=1$). Dati i bit da sommare, calcola la somma e si sposta in uno dei due stati in funzione del riporto uscente. Il FA implementa le seguenti funzioni:

$$S_i = a_i \oplus b_i \oplus r_i$$

$$R_i = a_i b_i + a_i r_i + b_i r_i$$

Lo stato iniziale dell'addizionatore è lo "stato 0" (che coincide, inoltre, con lo stato finale della macchina che raggiunge questo stato dopo aver sommato gli ultimi due bit), la macchina è inoltre dotata di un segnale di reset che la riporta nello stato iniziale ($r=0$). Lo stato interno rappresenta anche lo stato del riporto (overflow), se la macchina ritorna allo "stato 0" in $r=0$, non si è verificato overflow, se ritorna allo "stato 0" in $r=1$, allora la somma ha generato un overflow. Ad ogni somma occorre riportare la macchina nello $r=0$.

Il contatore è utilizzato per contare gli otto cicli necessari all'addizionatore per sommare gli otto bit dei registri. All'ottavo clock il contatore emette un segnale di fine somma

Tempificazione

Supponiamo che l'addizionatore sia soggetto ad un clock C2. I registri-addendi devono variare tra due segnali C2 consecutivi (cioè devono fornire i bit da addizionare, solo dopo che i precedenti due sono stati sommati), quindi si suppone soggetto ad un clock C1. Il registro-soluzione, deve caricare la somma generata da Add, quindi può operare in sincronismo con Add, sul clock C2.

I registri a scorrimento forniscono i bit da sommare in sincronismo con il clock C1, i bit saranno allora disponibili in ingresso all'addizionatore dopo un tempo T_{sh} , a questo punto l'addizionatore può calcolare la somma dei due bit, che richiederà un tempo T_{add} , dopo di che la somma e il riporto saranno disponibili in uscita. Si attende un ulteriore tempo di sicurezza, quindi viene emesso il segnale di clock C2. In sincronismo con C2, il registro-soluzione acquisisce il bit somma che è stato calcolato, e il riporto va a cambiare lo stato della macchina. Per completare l'operazione occorre attendere un tempo T_{st} , affinché il registro-somma completi l'acquisizione. Due segnali C1 devono essere distanziati di un tempo maggiore di $(T_{sh} + T_{add} + T_{st} + \delta)$

La macchina SUB

Per il sottrattore vale l'analisi compiuta per l'addizionatore, infatti il sottrattore è realizzabile mediante un addizionatore (FA) previa complementazione di alcuni bit. In

Il sottrattore è definito nel seguente modo:

$$\text{sub_mod_2}(X, Y, S, r, R) \iff \text{add_mod_2}(X, Y', S, r', R')$$

quindi è possibile costruire un sottrattore utilizzando un addizionatore e complementando il (secondo addendo) sottraendo e il riporto entrante.

Semplificando l'equazione per il calcolo del riporto si ha per il FA che realizza il sottrattore ha le seguenti equazioni:

$$S_i = a_i \oplus b_i \oplus r_i$$

$$R_i = a_i' b_i + a_i' r_i + b_i r_i$$

Tempificazione

Per il sottrattore vale l'analisi effettuata per l'addizionatore. Occorre attendere un tempo T_{sh} affinché i bit da sottrarre, resi disponibili dai registri, siano acquisiti dal sottrattore, un tempo T_{add} con un ulteriore tempo di sicurezza, necessario alla macchina per svolgere la differenza, quindi in sincronismo con il clock $C2$, la differenza può essere acquisita dal registro-soluzione e occorre un ulteriore tempo T_{st} , affinché la differenza sia memorizzata.

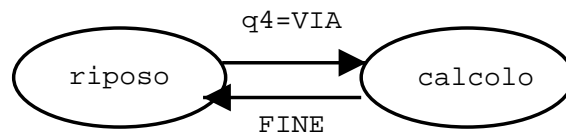
Sistema di controllo

1 - Caricamento dei dati nei registri, reset dei flip flop di riporto e dei contatori, abilitazione della macchina che calcolerà il risultato (stati da q0 a q3)

2 - calcolo dell'operazione (stato q4, si abilitano i registri a funzionare come shift register)

3 - inoltro del risultato

la rete opera secondo il seguente schema:



La macchina acquisirà i dati negli stati da q0 a q3, in queste fasi la macchina pilota, in base al proprio stato, il funzionamento dei registri dati. Il registro 74165 ha a disposizione un ingresso SH/LD' che se alto farà funzionare il componente come registro di scorrimento, se basso abilita il componente a caricare il valore che è posto al suo ingresso. Inoltre dispone di un ingresso CLK INH che se posto al valore alto, inibisce il clock. Gli ingressi SL/LD' e CLK INH sono stati sfruttati per indirizzare ad uno dei due registri il valore da memorizzare. I due registri-dati (R1, R2) possono essere pilotati utilizzando gli stati della macchina.

stato della macchina	SL/LD'_1	INH_1	SH/LD'_2	INH_2
q0	-	1	-	1
q1	0	0	1	1
q2	1	1	1	1
q3	1	1	0	0
q4	1	0	1	0

Il registro R1 deve essere abilitato a caricare il dato quando la macchina si trova nello stato q1, in questo stato il bit di inibizione deve essere posto a zero per permettere al clock di far funzionare il registro. Nello stato q2 e q3 il registro ha caricato il primo numero, e per evitare che venga shiftato, il registro viene inibito ponendo ad 1 il relativo segnale.

Nello stato q2, i registri non devono caricare nessun dato, il valore che è presente è relativo al codice dell'operazione da compiere, i registri sono allora inibiti al clock ed inoltre sono impostati a funzionare come shift register.

Analogamente il registro R2, sarà inibito negli stati q0, q1, q2 della macchina, deve caricare il dato nello stato q3, quindi si pone a zero il valore del segnale di inibizione, permettendo al clock di funzionare e facendo, in questo modo caricare il secondo numero.

Alla pressione del tasto "=", entrambi i registri devono funzionare come shift register e fornire alla macchina selezionata i bit su cui operare.

Quindi:

$$SH/LD'_1 = q2 + q3 + q4 = y3'y2y1' + y3'y2y1 + y3y2'y1 = y1(y3'y2 + y3y2') + y3'y2y1 = y1(y2 \oplus y3) + y3'y2y1$$

$$SH/LD'_2 = q1 + q2 + q4 = y3'(y1 \oplus y2) + y3y1'y2'$$

$$INH_1 = q0 + q2 + q3 = y3'y1' + y3'y1y2$$

$$INH_2 = q0 + q1 + q2 = y3'y1' + y3'y2'y1$$

Alla pressione del tasto “=” si fa partire il calcolo, viene emesso un segnale di VIA la rete fa partire la macchina interessata. Quando questa ha finito l’elaborazione invia un segnale di FINE che riporta la rete allo stato di riposo.

La rete che avvia/interrompe il calcolo è realizzata con un flip flop JK posto in set dal segnale via (f14=1), e in reset dall’ottavo impulso di conteggio del contatore. Una volta attivo il flip flop emette un segnale “work”, che messo in AND con il clock CLK genera il treno di otto clock necessari al contatore. Il segnale “work” viene interrotto nel momento in cui, pervenendo dal contatore il segnale di fine conteggio, il flip flop JK viene resettato.

Tempificazione

Si osserva che, non appena i due registri 74165 hanno caricato il valore da memorizzare, rendono immediatamente disponibili in uscita il bit meno significativo (bit H), quindi durante la fase di acquisizione dei dati (prima che venga permuta il tasto “=” l’addizionatore (sottrattore) ha già calcolato la somma (differenza) dei primi due bit. La somma (differenza) è quindi immediatamente disponibile per essere trasferita nel registro soluzione.

Si suppone che la rete sia sincrona con un unico segnale di clock. Ciò è possibile per le caratteristiche tecniche dei componenti utilizzati.

Il 74165 e il 74164 sono composti da flip flop edge triggered attivi sul fronte di salita, inoltre utilizzando il fatto di avere già due bit calcolati, è possibile sostituire ai due segnali di clock C2 e C1 il fronte di salita del segnale di clock CLK. Sul fronte di salita del clock, i registri a scorrimento forniranno i successivi bit su cui operare, mentre contemporaneamente, il registro-soluzione salverà il bit soluzione del calcolo precedente.

Sistema di controllo delle macchine calcolatrici

Quando la macchina si trova nello stato q2, bisogna abilitare una della 4 macchine calcolatrici. Alle macchine è stato assegnato la seguente codifica:

sommatore (add): 00
 sottrattore (sub): 01
 moltiplicatore (mul): 10
 divisore (div): 11

Nello stato q2 la macchina emette il segnale EN, che permetterà la memorizzazione della macchina selezionata per l’operazione. Dalla rete combinatoria sono disponibili i segnali relativi alla pressione di uno dei tasti ai quali è stato assegnato una operazione aritmetica. Riporto la parte di tabella interessata, con la codifica utilizzata per le macchine

	f10	f11	f12	f13	m1	m2
A (+)	1				0	0
B (-)		1			0	1
C (*)			1		1	0
D (/)				1	1	1

dalla tabella si ricava che: $m1 = f12+f13$ mentre $m2 = f11+f13$. Occorre memorizzare 4 stati diversi, utilizzo 2 flip flop di tipo D. In presenza del segnale EN, i flip flop salveranno il codice della macchina abilitata a compiere l’operazione.

Alla pressione del tasto “=” solo una delle quattro macchine risulterà abilitata, e verso quella macchina sarà inviato il clock.

Test

La macchina è stata testata eseguendo la somma $5+6$ e la differenza $7-3$.

In binario si ha:

$$5 = 00000101$$

$$6 = 00000110$$

$$6+5 = 11 = 00001011$$

$$7 = 00000111$$

$$3 = 00000011$$

$$7-3 = 4 = 00000100$$

Dai grafici dei tempi allegati si può osservare il funzionamento della macchina, durante l'acquisizione dei dati e durante il calcolo della somma e della differenza. Somma e differenza sono calcolati a partire dal bit meno significativo, quindi il risultato va letto al contrario.