ARCHITUTTETTURA DEI SISTEMI DIGITALI

08/02/2022

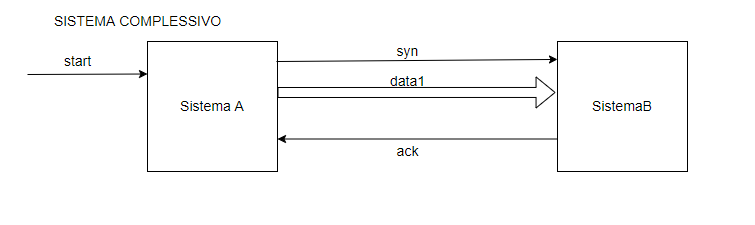
Christian Marescalcol Mat M63001367

**TRACCIA**

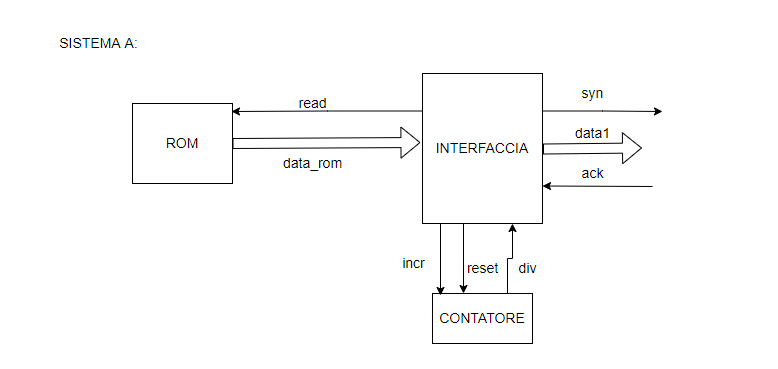
Un nodo A invia mediante un protocollo di handshaking completo 4 coppie di stringhe da 8 bit ad un nodo B. Per ogni coppia Aj Mj inviata, il nodo B effettua il calcolo Aj mod Mj. L’acquisizione di una coppia avviene al termine del calcolo sulla coppia precedente. La rete di controllo va realizzata in logica cablata, ma va in ogni caso descritta in modo schematico la differenza di progettazione con la logica microprogrammata.

**SCHEMA**

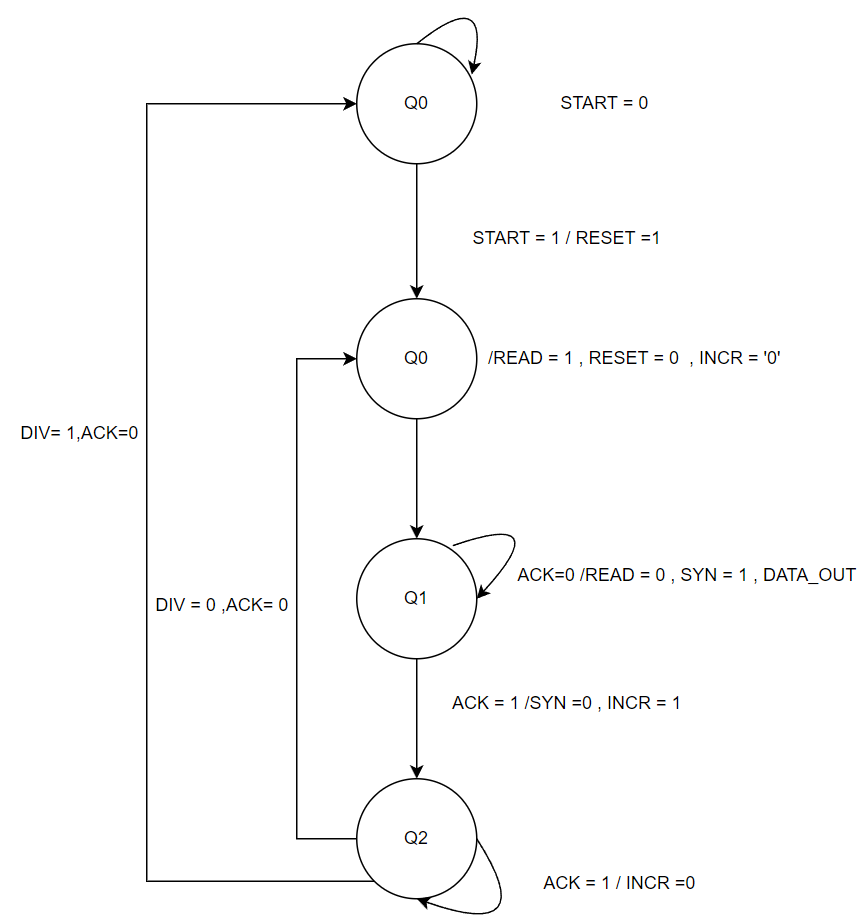
* SISTEMA COMPLESSIVO:



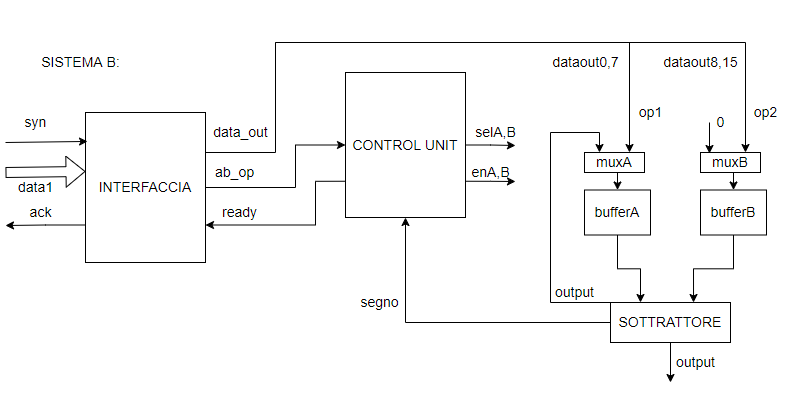
* SISTEMA A:



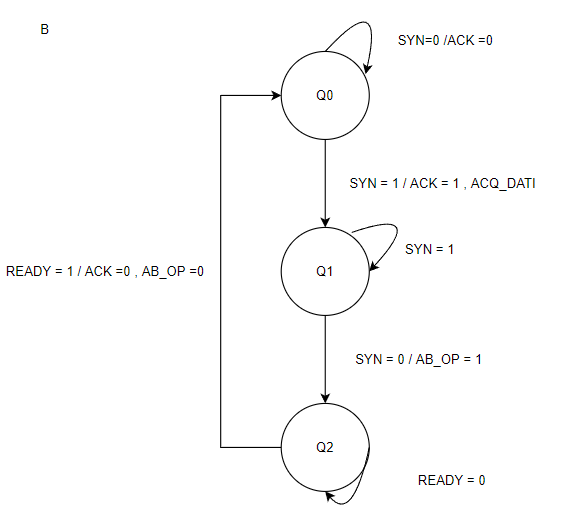
* INTERFACCIA A:



* SISTEMA B:



* INTERFACCIA B:



Per realizzare la macchina che effettua l’operazione di modulo tra A e B si utilizza una decomposizione funzionale del sistema in parte operativa e parte di controllo. Per la parte operativa si utilizzano: un sottrattore, 2 buffer e 2 mux opportunamente interconnessi. A partire dall’algoritmo per la realizzazione del modulo tra A e B, si realizza poi la parte di controllo.

load A,B

subtract A,B

if(segno = ‘0’) then

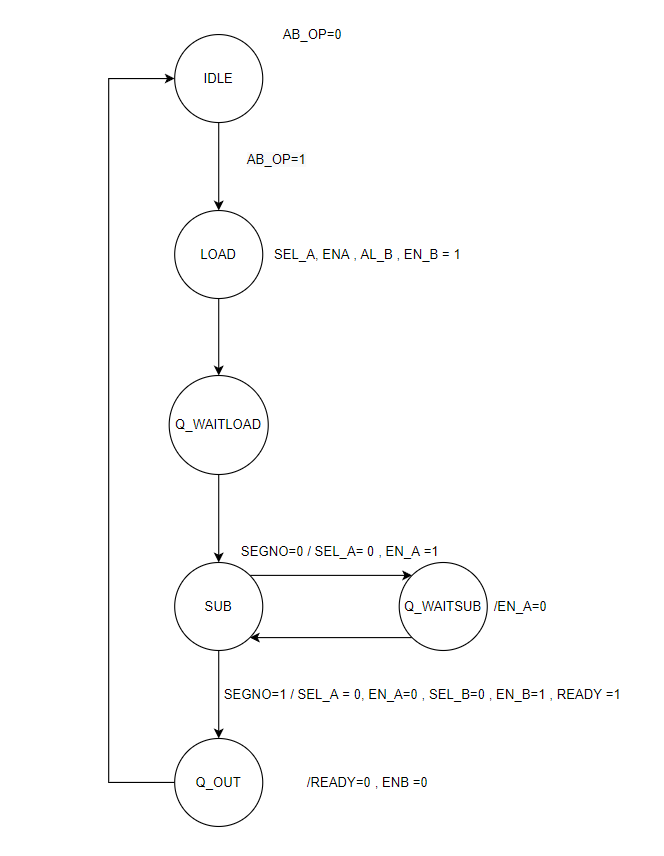
go to subtract A,B

output A

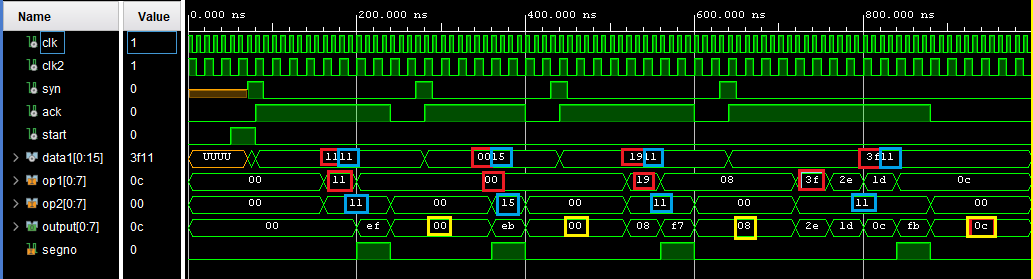
**APPROCCIO LOGICA CABLATA**

Si può procedere utilizzando un approccio a logica cablata, come è stato implementato in questo esercizio, realizzando una FSM che descrive l’algoritmo. Si introducono dei nuovi stati di:

* idle, per attendere l’abilitazione da parte dell’interfaccia all’esecuzione dell’operazione sui dati A e B;
* wait\_load, per attendere il caricamento degli operandi A, B ed effettuare la prima sottrazione;
* wait\_sub, per attendere che il segno della sottrazione si propaghi, evitando ulteriori sottrazioni (in questo modo controllo il numero di sottrazioni da effettuare);



**SIMULAZIONE**

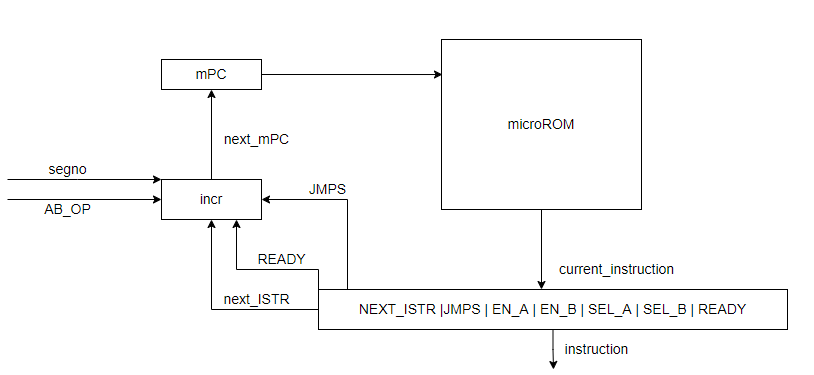


I valori compresi nei riquadri rossi si riferiscono agli Ai, i valori nei riquadri blu si riferiscono ai Bi, mentre i valori presenti nei riquadri gialli rappresentano Ai mod Bi.

**APPROCCIO LOGICA MICROPROGRAMMATA**

L’approccio microprogrammato prevede l’utilizzo dei seguenti componenti:

* microROM;
* registro mPC;
* logica di controllo per calcolare il next\_mPC;



Il formato della microistruzione è dato da:

NEXT\_ISTR |JMPS | EN\_A | EN\_B | SEL\_A | SEL\_B | READY

Dove NEXT\_ISTR indica l’indirizzo della prossima istruzione; JMPS indica di effettuare un salto condizionato in base al valore del segno; en\_A, en\_B, sel\_A, sel\_B servono per abilitare i buffer e selezionare correttamente gli operandi; READY per comunicare all’interfaccia che è terminata l’esecuzione.

Una possibile implementazione delle microistruzioni e della microROM è la seguente:

idle

NEXT\_ISTR = 0, JMPS = 0, ENA = 0, ENB = 0, SEL\_A = 0, SEL\_B = 0, READY = 0  istr 0

load

NEXT\_ISTR = 2, JMPS = 0, ENA = 1, ENB = 1, SEL\_A = 1, SEL\_B = 1, READY = 0 istr 1

wait\_load

NEXT\_ISTR = 3, JMPS = 0, ENA = 1, ENB = 1, SEL\_A = 1, SEL\_B = 1, READY = 0 istr 2

sub

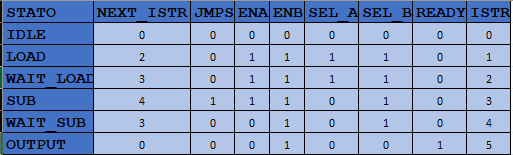
NEXT\_ISTR = 4, JMPS = 1, ENA = 1, ENB = 1, SEL\_A = 0, SEL\_B = 1, READY = 0 istr 3

wait\_sub

NEXT\_ISTR = 3, JMPS = 0, ENA = 0, ENB = 1, SEL\_A = 0, SEL\_B = 1, READY = 0 istr 4

output

NEXT\_ISTR = 0, JMPS = 0, ENA = 0, ENB = 1, SEL\_A = 0, SEL\_B = 0, READY = 1 istr 5



Una possibile logica di controllo che calcola il prossimo valore di mPC è la seguente:

if JMPS = 1 and SEGNO = 1  
 next\_mPC = NEXT\_ISTR + 1

elsif AB\_OP = 1 and NEXT\_ISTR = 0 and READY = 0  
 next\_mPC = 1

else next\_mPC = NEXT\_ISTR