Elaborato SIS

Architettura degli elaboratori

Nicola Serlonghi – Alessandro Bloise
VR409046 – VR408120

A.A. 2016/2017
Università degli Studi di Verona

Indice

Architettura generale del circuito

Il dispositivo da noi implementato permette il monitoraggio di un impianto chimico industriale basato su un circuito sequenziale che riceve come input il pH di una soluzione contenuta in un serbatoio, e fornisce in uscita lo stato della soluzione (compatibilmente con delle soglie pre-impostate) in termini acido (A), basico (B) e neutro (N).Il sistema deve sempre portare la soluzione allo stato neutro e se essa lo è già deve semplicemente rimanere in quello stato. Se invece abbiamo una sostanza che è acida o basica da più di 5 cicli di clock provvederemo al sesto ciclo ad aprire la relativa valvola, BS se ci troviamo di fronte ad una soluzione acida o AS se siamo allo stato basico, per portare la soluzione allo stato neutro.
Il circuito è composto da un controllore e da un datapath che conteggia i vari cicli di clock e cambia gli stati della macchina con i seguenti ingressi e uscite.

INPUTS:

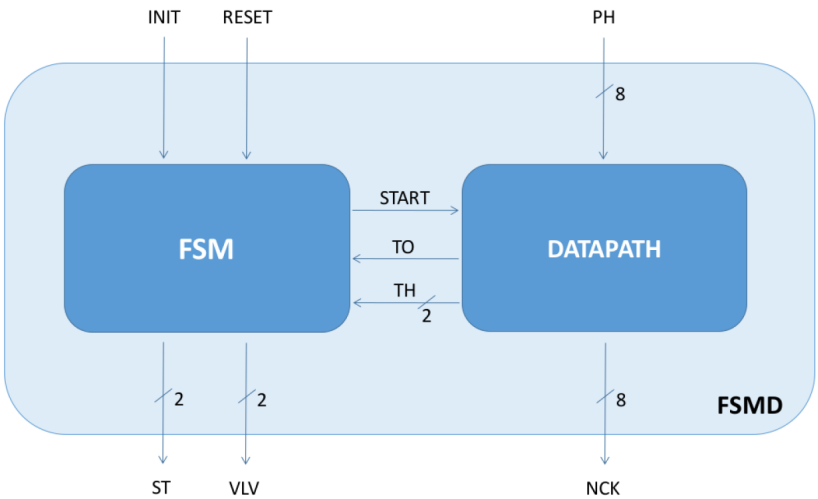
- INIT [1]: quando vale 1 il sistema è acceso; quando vale 0 il sistema è spento e deve restituire 0.
- RESET [1]: quando posto a 1 il controllore deve essere resettato, ovvero tutte le uscite devono essere poste a 0 e il sistema riparte.
- PH [8]: valore del pH misurato dal rilevatore. Il range di misura è compreso tra 0 e 14 con risoluzione di 0,1.

OUTPUTS:

- ST [2]: indica in quale stato si trova la soluzione al momento corrente (01-A, 10-N, 11-B).
- NCK [8]: indica il numero di cicli di clock trascorsi nello stato corrente
- VLV [2]: indica quale valvola aprire per riportare la soluzione allo stato neutro nel caso in cui la soluzione si trovi da più di 5 cicli di clock in stato A o B (01-BS, 10-AS)

Il controllore è collegato al datapath con tre segnali che hanno il seguente significato:

- START [1]: segnale di start che comanda il datapath in funzione di INIT e RESET.
- TO [1]: segnale di timeout che indica quando il sistema si trova in uno stesso stato da più di 5 cicli di clock.
- TH [2]: stato in cui si trova il sistema (noi abbiamo considerato 00-A, 10-N, 11-B)

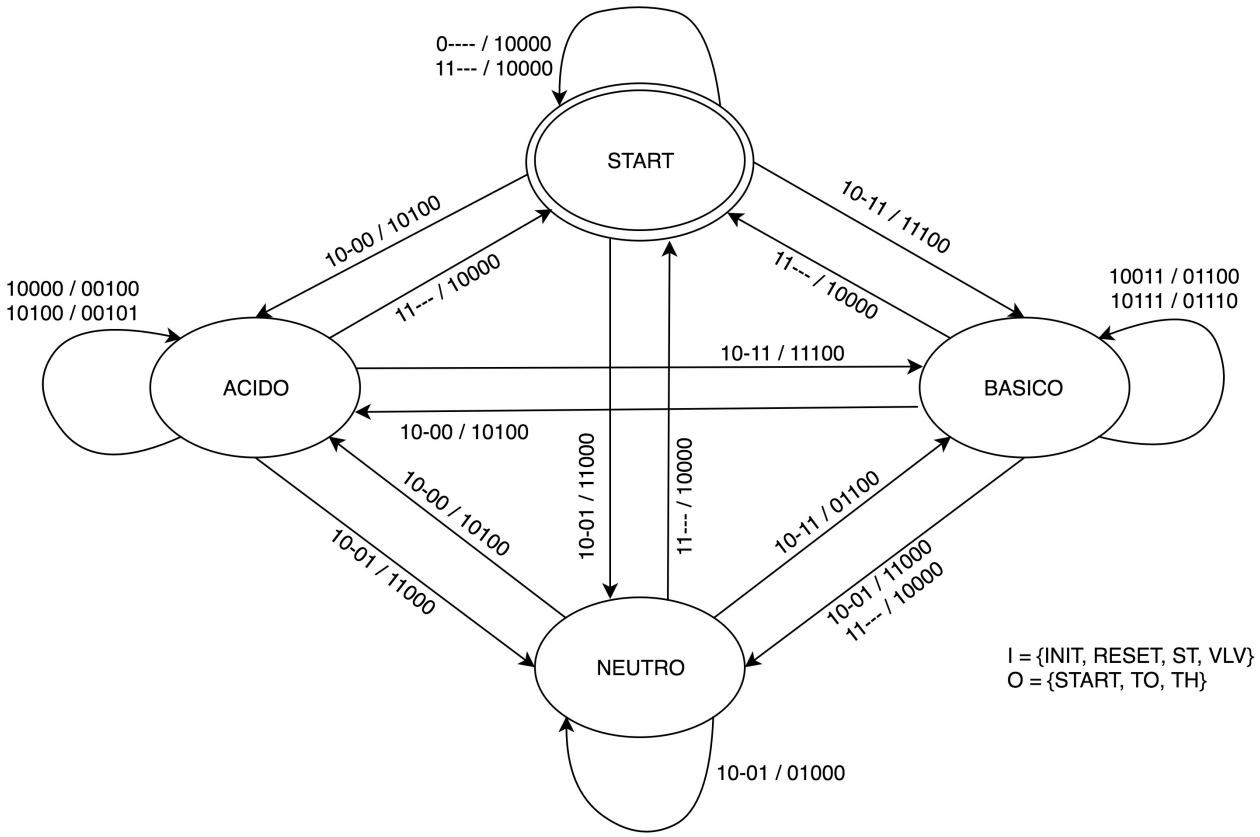


Controllore

La macchina a stati finiti (FSM=Finite State Macchine) è rappresentata da quattro ingressi (INIT, RESET, ST, VLV) e tre uscite (START, TO, TH).
Gli stati individuati sono quattro:

- START: la macchina è spenta perciò l'INIT è posto a 0, rappresenta lo stato di reset iniziale.
- NEUTRO: è lo stato neutro; questo è lo stato ottimale che la macchina deve cercare di raggiungere, il pH in questo stato deve essere maggiore di 6 e minore di 8.
- ACIDO: è lo stato acido; il pH è minore di 6.
- BASICO: è lo stato basico; il pH è maggiore di 8.

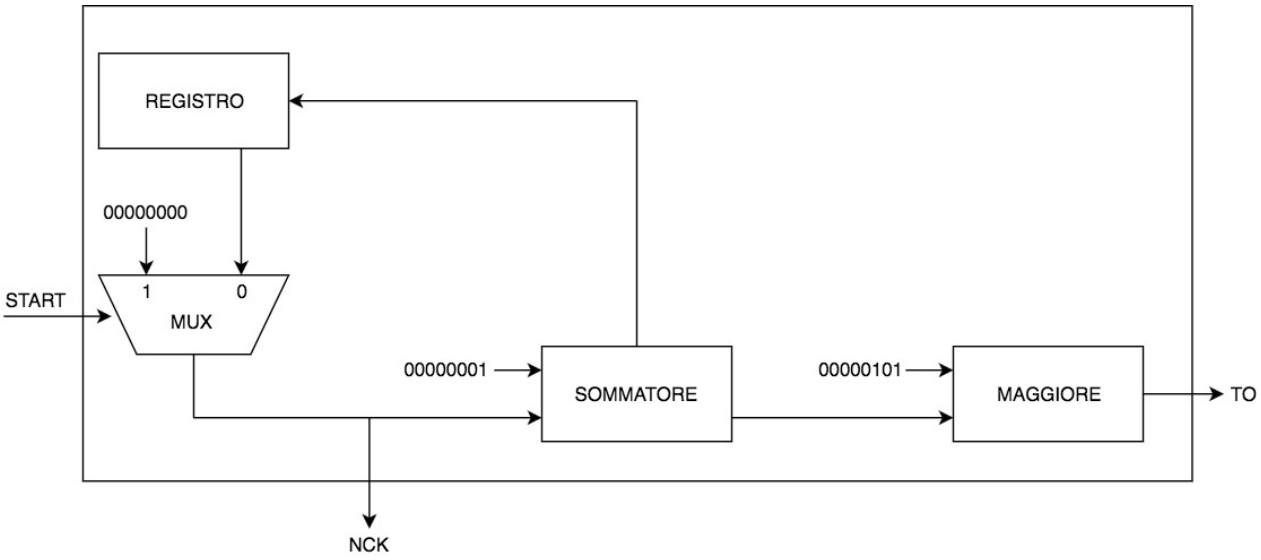
Schema del controllore

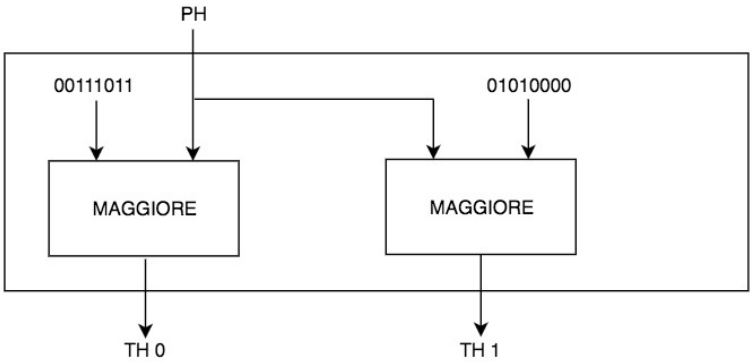


ST che rappresenta lo stato attuale in cui si trova il controllore, viene fornito in ingresso dal datapath e non è uguale all'uscita, TO, della macchina a stati finiti:

	TH	ST
ACIDO	00	01
NEUTRO	10	10
BASICO	11	11

Datapath





Il datapath da noi progettato è suddiviso in due parti: la prima parte è composta da un registro, un mulitplexer, un sommatore ed un maggiore. Essa serve per conteggiare i cicli di clock e funziona nel seguente modo: quando il pH ricevuto in ingresso ci fa cambiare stato, azzero il contatore e lo incremento di uno ad ogni ciclo di clock. Quando esso arriva a 6, quindi sono passati 6 cicli di clock senza che il pH in ingresso mi cambiasse stato, apro la relativa valvola, sopra indicate, per far si che la soluzione torni allo stato di neutro, dopo di che azzero il contatore.
La seconda parte del datapath ha il compito di controllare se il pH fornito come input è acido, basico o neutro. Per fare ciò abbiamo deciso di usare due componenti, il primo controlla se il pH è maggiore di 59 mentre il secondo se è maggiore di 80. Abbiamo usato 59 al posto di 6 e 80 al posto di 8 per rispettare le specifiche date ed avere una risoluzione di 0,1 e tenere come valori neutri anche 60 (6) e 80 (8).

File .blif

- complessivo.blif: file di unione della FSM e del datapath.
- datapath.blif: creazione del datapath con l'utilizzo di altri file blif che costituiscono i componenti.
- fsm.blif: creazione della FSM (controllore).
- sommatore_8bit.blif: componente che serve ad incrementare di uno il registro ad ogni ciclo di clock.
- maggiore_8bit.blif: componente che controlla se il primo valore inserito è maggiore del secondo.
- multiplexer_8bit.blif: componente che serve per azzerare il registro quando START viene portato ad 1.
- registro_8bit.blif: componente per la registrazione dei cicli clock.

Statistiche del circuito

Vengono ora descritte le statistiche del circuito prima e dopo l'ottimizzazione per area. Per prima cosa abbiamo cercato di minimizzare gli stati della FSM tramite i comandi: "state_minimize stamina", "state_assign jedi", "source -x script.rugged" ottenendo le seguenti statistiche riportate in seguito mostrando prima la situazione iniziale:

```
sis> read_blif fsm.blif
sis> print_stats
CONTROLLORE      pi= 5   po= 5   nodes=  5       latches= 0
lits(sop)=    0  #states(STG)=   4
sis>
```

```
UC Berkeley, SIS 1.3.6 (compiled 2016-10-05 23:28:29)
sis> read_blif fsm.blif
sis> state_minimize stamina
Running stamina, written by June Rho, University of Colorado at Boulder
Number of states in original machine : 4
Number of states in minimized machine : 4
sis>
```

Siamo riusciti ad ottenere una diminuzione del numero di letterali proprio come noi volevamo. Il datapath da noi scritto presenta le seguenti statistiche e per ridurre il numero di letterali abbiamo utilizzato nuovamente il comando "source -x script.rugged":

```
sis> read_blif datapath.blif
Warning: network `DATAPATH', node "[283]" does not fanout
sis> print_stats
DATAPATH         pi= 9   po=11   nodes= 61       latches= 8
lits(sop)= 372
sis>
```

```
sis> read_blif datapath.blif
Warning: network `DATAPATH', node "[110]" does not fanout
sis> source -x script.rugged
sweep; eliminate -1
simplify -m nocomp
eliminate -1

sweep; eliminate 5
simplify -m nocomp
resub -a

fx
resub -a; sweep

eliminate -1; sweep
full_simplify -m nocomp
sis> print_stats
DATAPATH        pi= 9   po=11   nodes= 24       latches= 8
lits(sop)=  86
sis>
```

```
sis> state_minimize stamina
Running stamina, written by June Rho, University of Colorado at Boulder
Number of states in original machine : 4
Number of states in minimized machine : 4
sis> state_assign jedi
Running jedi, written by Bill Lin,  UC Berkeley
sis> source -x script.rugged
sweep; eliminate -1
simplify -m nocomp
eliminate -1

sweep; eliminate 5
simplify -m nocomp
resub -a

fx
resub -a; sweep

eliminate -1; sweep
full_simplify -m nocomp
sis> print_stats
CONTROLLORE     pi= 5   po= 5   nodes=  7       latches= 2
lits(sop)=  31  #states(STG)=   4
sis>
```

In fine una volta ottimizzato il circuito, dobbiamo mapparlo con una libreria, in modo da avere delle statistiche più verosimili per quanto riguarda area e ritardo.
Nel nostro caso la libreria assegnata è la “synch.genlib”, e l’abbiamo applicata sulla FSMD tramite il comando “map -s”. Una volta mappato il circuito presenta le seguenti statistiche:

```
sis> read_blif fsmd.blif
Warning: network `FSM_ridotta.blif', node "to" does not fanout
Warning: network `DATAPATH', node "[241]" does not fanout
Warning: network `FSMD', node "[241]" does not fanout
sis> read_library synch.genlib
sis> map -s
warning: unknown latch type at node '{[230]}' (RISING_EDGE assumed)
warning: unknown latch type at node '{[1144]}' (RISING_EDGE assumed)
WARNING: uses as primary input drive the value (0.20,0.20)
WARNING: uses as primary input arrival the value (0.00,0.00)
WARNING: uses as primary input max load limit the value (999.00)
WARNING: uses as primary output required the value (0.00,0.00)
WARNING: uses as primary output load the value 1.00
>>> before removing serial inverters <<<
# of outputs:          22
total gate area:       3152.00
maximum arrival time: (33.00,33.00)
maximum po slack:     (-10.20,-10.20)
minimum po slack:     (-33.00,-33.00)
total neg slack:      (-439.60,-439.60)
# of failing outputs:  22
>>> before removing parallel inverters <<<
# of outputs:          22
total gate area:       3152.00
maximum arrival time: (33.00,33.00)
maximum po slack:     (-10.20,-10.20)
minimum po slack:     (-33.00,-33.00)
total neg slack:      (-439.60,-439.60)
# of failing outputs:  22
# of outputs:          22
total gate area:       2992.00
maximum arrival time: (32.80,32.80)
maximum po slack:     (-10.20,-10.20)
minimum po slack:     (-32.80,-32.80)
total neg slack:      (-437.80,-437.80)
# of failing outputs:  22
sis> print_stats
FSMD            pi=10   po=12   nodes= 88       latches=10
lits(sop)= 180
sis>
```

Il total gate area è 3152.00 mentre il cammino critico è 32.80.