ELABORATO SIS
ARCHITETTURA DEGLI ELABORATORI

A.A. 2021/2022

A cura di

Loffredo Francesca VR472152

Filippi Matteo VR472558

Kovacs Alexandru Ioan VR471323

INDICE

SPECIFICHE

Il progetto richiede la progettazione di un circuito sequenziale, il cui scopo è portare una soluzione iniziale a PH noto, ad un PH di neutralità. Come input riceve il PH iniziale, il quale può essere acido oppure basico, per PH acido si intende un valore maggiore-uguale di 0 e strettamente inferiore a 7, mentre per basico si intende una soluzione strettamente maggiore a 8 e inferiore o uguale a 14. In base alla soluzione iniziale il circuito apre due valvole, la valvola acida se il PH iniziale è basico e quella basica se il PH è acido, fino a raggiungere la soglia di neutralità, quindi una soluzione con PH compreso tra 7 e 8.

Input del circuito:

- RESET1: riporta il circuito allo stato di RESET, quindi allo stato iniziale.
- START: segnale necessario per iniziare a bilanciare la soluzione.
- PH INIZIALE: composto da 8 bit, suddivisi in parte intera e parte decimale:
    - INT3 – INT2 – INT1 – INT0: 4 bit della parte intera;
    - DEC3 – DEC2 – DEC1 – DEC0: 4 bit della parte decimale;

Output del circuito:

- FINE_OP: indica il termine del processo.
- ERRORE: indica un errore del PH in ingresso, quindi un valore maggiore di 14.
- VALVOLA_ACIDA: segnala di aprire la valvola della soluzione acida, quindi il PH iniziale è basico.
- VALVOLA_BASICA: segnala di aprire la valvola della soluzione acida, quindi il PH iniziale è acido.
- PH FINALE: composto da 8 bit, suddivisi in parte intera e parte decimale:
    - PH_FIN7 – PH_FIN6 – PH_FIN5 -PH_FIN4: parte intera;
    - PH_FIN3 – PH_FIN2 – PH_FIN1 – PH_FIN0: parte decimale;
- CLOCK7 … CLOCK0: composto da 8 bit, indica il numero di cicli di clock impiegati per raggiungere la soglia di neutralità.

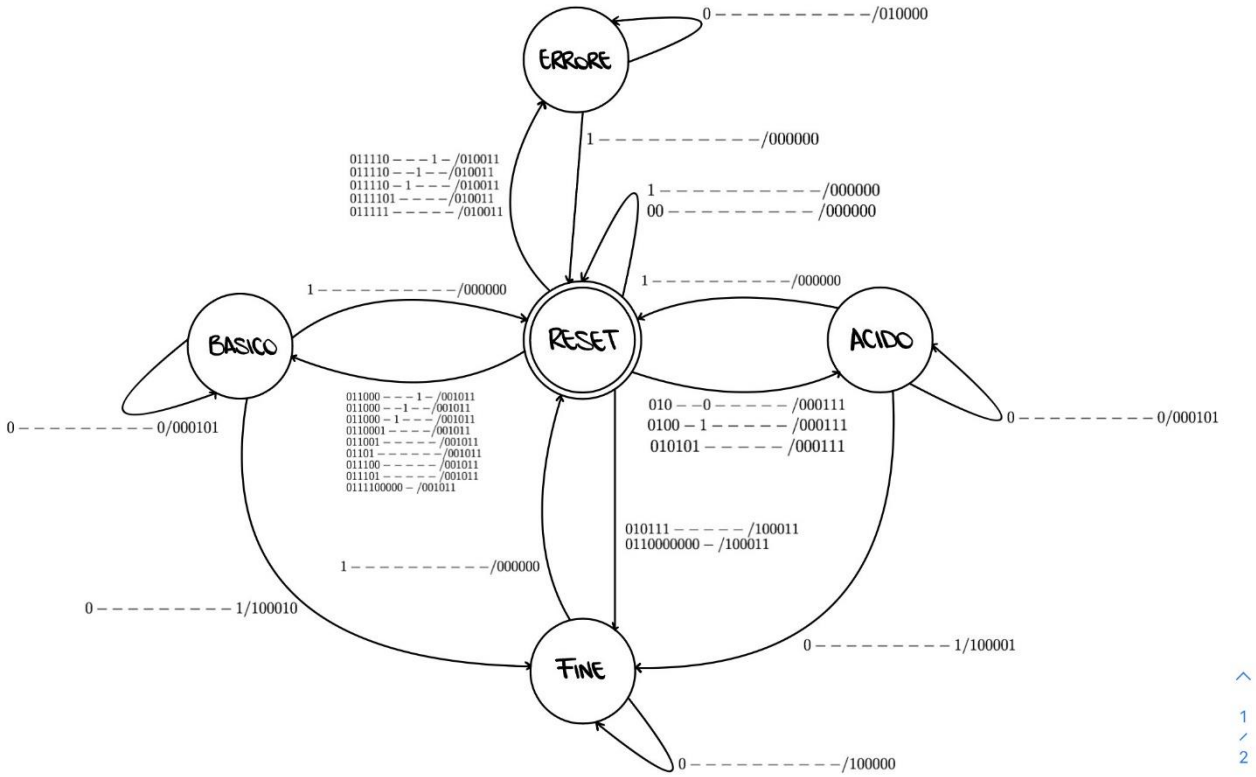
ARCHITETTURA GENERALE DEL CIRCUITO:

Il progetto è stato creato seguendo il modello FSMD, con un circuito controllore: FSM e un circuito elaboratore: DATAPATH.

I primi 4 output provengono dalla macchina a stati finiti, mentre gli ultimi due giungono dal circuito di elaborazione dei dati.

DIAGRAMMA DEGLI STATI DEL CONTROLLORE:

Riportiamo di seguito lo State Transition Graph:



Per realizzare la FSM di questo circuito abbiamo utilizzato cinque stati:

1. **RESET:** è lo stato iniziale, quando l'input va a 0 inizia il circuito.
2. **ERRORE:** il PH iniziale non è valido, ovvero ha un valore maggiore di 14.
3. **BASICO:** indica l'apertura della valvola di soluzione acida, in quanto il PH inziale è basico, ha quindi un valore strettamente maggiore di 8 e minore-uguale di 14. Questa valvola ha il compito di sommare 0.5 al PH inserito in input.
4. **ACIDO:** indica l'apertura della valvola di soluzione basica; quindi, il PH iniziale è acido con un valore maggiore-uguale a 0 e strettamente inferiore a 7. Questa valvola ha il compito di sottrarre 0.25 al PH corrente.
   La somma e la sottrazione sono compito del DATAPATH.
5. **FINE:** indica il termine dell'operazione.

In ogni transizione sono indicati 11 ingressi e 6 uscite:

INGRESSI:

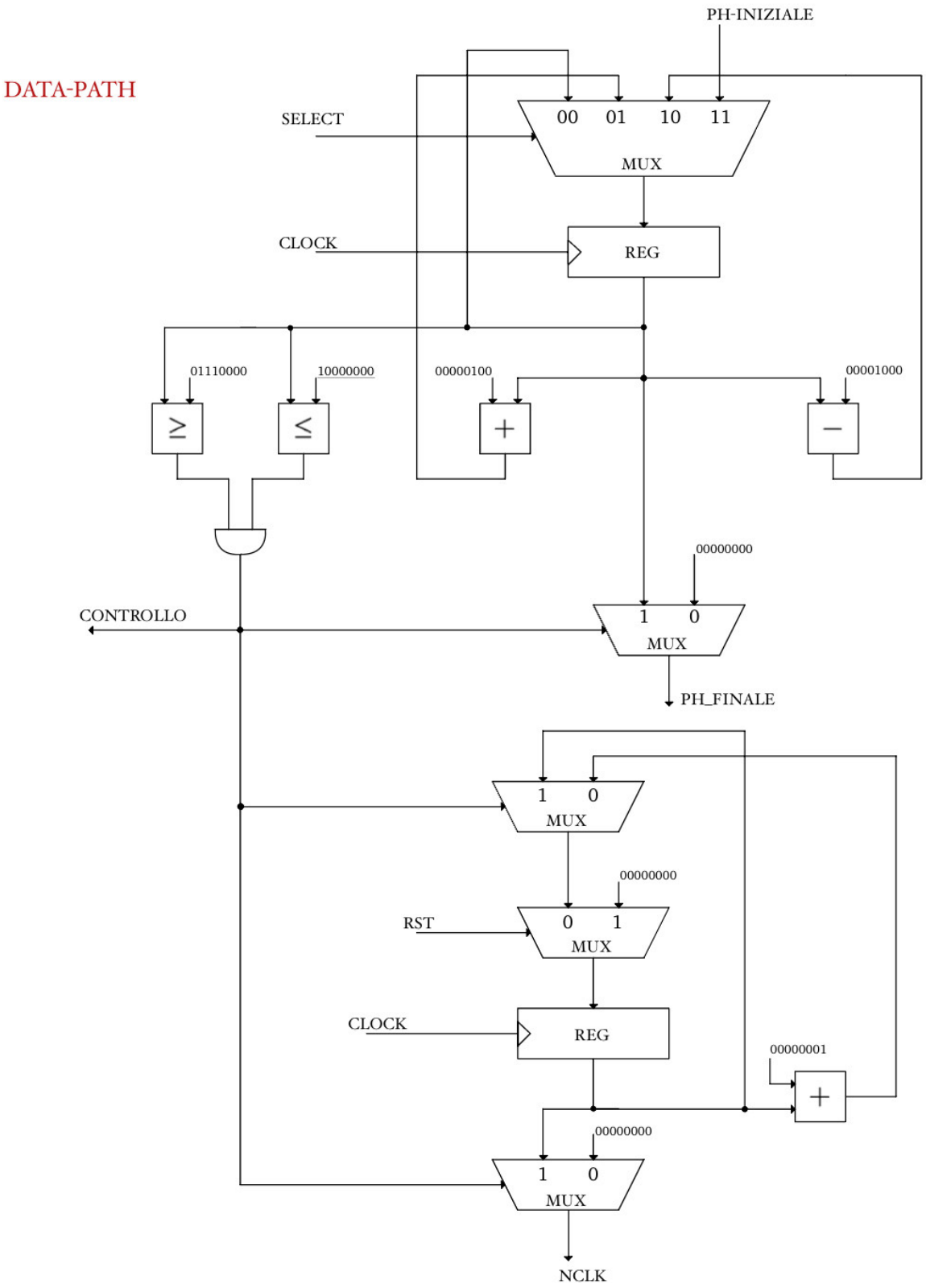
1. **RESET:** quando è a 0 il circuito inizia l'operazione, mentre quando è posto a 1, da qualsiasi stato, torna nello stato iniziale.
2. **START:** quando è posto a 1 il circuito opera.

3. I successivi 8 bit rappresentano il PH:
    a. quando il PH iniziale è tra 0 e 7 (non compreso) si passa allo stato ACIDO;
    b. se il valore è tra 8 (non compreso) e 14 si passa allo stato BASICO;
    c. se il PH è superiore a 14 si va allo stato ERRORE;
    d. se il PH è compreso tra 7 e 8 si procede allo stato FINE;
4. OP_END: input proveniente dal datapath, quando è posto a 1 indica che il PH ha raggiunto la soglia di neutralità, quindi il circuito ha raggiunto il suo scopo e termina l'operazione.

USCITE:

1. FINE_OPERAZIONE: quando è posto a 1 indica la fine dell'operazione, quindi passa allo stato FINE.
2. ERRORE_SENSORE: quando è posto a 1 significa che è stato inserito un valore del PH errato, nonché strettamente superiore a 14, quindi si passa allo stato ERRORE.
3. VALVOLA_ACIDA: se posto a 1 determina che il PH iniziale è basico, quindi viene erogata la soluzione acida, ci si trova nello stato BASICO.
4. VALVOLA_BASICA: se posto a 1 determina che il PH iniziale è acido, quindi viene somministrata la soluzione basica, ci si trova nello stato BASICO.
5. Gli ultimi due bit sono collegati al Datapath, rappresentano il selettore del mux, il quale stabilisce se il PH iniziale è acido, basico o neutro.

ARCHITETTURA DEL DATAPATH:



Le componenti che abbiamo utilizzano sono:

- MULTIPLEXER A 4 INGRESSI DI 8 BIT: come input riceve il PH iniziale, inizialmente il selettore (2 bit) viene posto a 1 1, in quanto indica l'inserimento del PH iniziale, successivamente il selettore cambia in base allo stato della soluzione inserita.
    - Se il selettore è 0 0 la soluzione è neutra e si passa al confronto con i valori maggiori-uguale di 7 e minori-uguale al valore 8.
    - Se il selettore è 0 1 il PH è acido e l'uscita è collegata al sommatore.
    - Se il selettore è 1 0 il PH è basico e si passa al sottrattore.
- REGISTRO A 8 BIT: consente di memorizzare il valore uscito dal mux.
- SOMMATORE A 8 BIT: se la soluzione iniziale è acida viene sommato 0.25 (00000100) al valore inserito, finché il PH non raggiunge la soglia di neutralità. Il multiplexer iniziale aggiorna ogni volta il valore, memorizzato successivamente nel registro.
- SOTRATTORE A 8 BIT: se la soluzione iniziale è basica viene sottratto 0.5 (00001000) al PH inserito, finché non raggiunge la soglia di neutralità. Il multiplexer iniziale aggiorna ogni volta il valore, memorizzato poi nel registro.
- MAGGIORE-UGUALE DI 8 BIT: il valore inserito viene confrontato con i valori maggiori-uguale a 7, se l'output è uguale a 0 indica che il valore è minore di 7.
- MINORE_UGUALE DI 8 BIT: il valore inserito viene confrontato con i valori minori-uguale a 8, se l'output è uguale a 0 significa che il valore è maggiore di 8.
- AND: come input riceve l'output del maggiore-uguale e quello del minore uguale; l'uscita è uguale a 1 quando entrambi gli ingressi sono posti a 1, quindi quando il valore è compreso tra 7 e 8. L'uscita è collegata alla FSM indicando la fine dell'operazione, quindi il PH ha raggiunto la soglia di neutralità.
- MULTIPLEXER A 2 INGRESSI A 8 BIT: come input riceve l'output del registro e il valore 0 (00000000), mentre come selettore prende l'uscita dell'and. Quando l'uscita dell'and è uguale a 1 il mux rilascia il valore memorizzato nel registro, nonché il PH finale. Quando l'uscita dell'and è 0 il multiplexer come output ha il valore 0 (00000000).

CICLI DI CLOCK:

Lo stato di reset collegato al multiplexer viene posto inizialmente a 0 salvando così nel registro il valore 0. Successivamente il reset passa a 1 e rimane costante, salvando nel registro l'uscita di un altro multiplexer. Il quale come input ha il valore nel registro e l'uscita del sommatore, il cui ad ogni ciclo somma 1 (00000001) al valore salvato nel registro. Il primo multiplexer come selettore ha l'uscita dell'and, quando è a 0 salva l'uscita del sommatore, quindi ad ogni ciclo il valore incrementa di 1, mentre quando è posto a 1 la somma si ferma. Il registro è collegato ad un ultimo multiplexer, il quale come selettore ha anch'esso l'uscita dell'and, finché è posta a 0 il mux come output ha 0 (00000000), mentre quando è a 1 rilascia il valore memorizzato nel registro, nonché il numero dei cicli di clock.

Il circuito Datapath si occupa di eseguire i calcoli secondo gli input ricevuti, che sono:

- FSM1 e FSM0: sono i selettori collegati al primo multiplexer, indicando lo stato del PH.
- I7 … I0: PH iniziale, suddiviso in 4 bit che rappresentano la parte intera e 4 che raffigurano la parte decimale.
- RESET: collegato al multiplexer che fa iniziare il conto dei cicli di clock trascorsi fino alla conclusione dell'operazione.

Gli output invece sono:

- F7 … F0: rappresentano gli 8 bit del PH finale, anch'essi suddivisi in parte intera e parte decimale.
- NCLK7 … NCLK0: sono gli 8 bit che conteggiano quanti cicli di clock sono trascorsi dall'inizio dell'operazione fino alla sua conclusione.
- OP_END: è il risultato del circuito controllore che valuta se il PH ha raggiunto la soglia di neutralità; coincide con FINE_OPERAZIONE.

SCELTE PROGETTUALI:

- Quando simuliamo il circuito, SIS elenca dei messaggi di “Warning” dovuti ai riporti del sommatore che genera delle uscite che non vengono utilizzate. In realtà non sono degli errori, poiché possiamo perdere questi dati, che corrispondono al bit di overflow nelle operazioni di somma.
- Al posto del sottrattore abbiamo utilizzato un sommatore a 8 bit, il quale riceve il valore inserito e somma 0.5 negato (11111000), ovvero sottrae il valore 0.5.
- Al primo ciclo di clock il registro presenta valori casuali, che successivamente vengono modificati con i valori inseriti da input. Questo è causato dal fatto che il registro utilizzi un giro di clock in più per memorizzare il valore al suo interno.

STATISTICHE

FSM:

Abbiamo codificato gli stati della FSM con “state_assign jedi ottenendo le seguenti statistiche:

```
UC Berkeley, SIS 1.3.6 (compiled 2017-10-27 16:08:57)
sis> rl FSM.blif
sis> print_stats
FSM             pi=11   po= 6   nodes=  9       latches= 3
lits(sop)= 534  #states(STG)=   5
sis>
```

Che dopo l’ottimizzazione eseguendo il comando “source script.rugged” diventano:

```
sis> rl FSM.blif
sis> print_stats
FSM             pi=11   po= 6   nodes= 13       latches= 3
lits(sop)=  65  #states(STG)=   5
sis>
```

Le riduzioni del numero di letterali sono significative, al costo di un lieve aumento del numero di nodi del network ottimizzato.

DATAPATH:

Di seguito le statistiche pre-ottimizzazione:

```
sis> rl FSM.blif
sis> print_stats
FSM             pi=11   po= 6   nodes=  9       latches= 3
lits(sop)= 534  #states(STG)=   5
sis> rl DATAPATH.blif
Warning: network `DATAPATH', node "overflow0" does not fanout
Warning: network `DATAPATH', node "overflow1" does not fanout
Warning: network `DATAPATH', node "overflow2" does not fanout
sis> print_stats
DATAPATH        pi=11   po=17   nodes=144       latches=16
lits(sop)= 747
sis>
```

I warning sono relativi agli overflow prodotti dalla somma e dalla sottrazione per raggiungere il PH finale, irrilevanti ai fini della visualizzazione delle statistiche.

```
sis> print_stats
DATAPATH        pi=11   po=17   nodes= 47       latches=16
lits(sop)= 218
sis>
```

Anche qui notiamo una notevole riduzione del numero di letterali. A differenza della FSM il numero dei nodi nel circuito ottimizzato è circa il 33% inferiori al circuito originale.

FSMD:

La FSMD, essendo formata dall'unione dei due circuiti precedentemente descritti, presenta statistiche che sono la somma di quelle dei suoi componenti:

```
sis> print_stats
FSMD            pi=10   po=20   nodes=153       latches=19
lits(sop)=1281
sis>
```

Passando all'ottimizzazione:

```
sis> rl FSMD.blif
sis> print_stats
FSMD            pi=10   po=20   nodes= 62       latches=19
lits(sop)= 283
sis> source script.rugged
sis> print_stats
FSMD            pi=10   po=20   nodes= 62       latches=19
lits(sop)= 283
sis>
```

La riduzione di letterali e di nodi è stata minima, in quanto i circuiti del Datapath e della FSM erano già minimizzati.

MAPPING

Una volta ottimizzato il circuito, dobbiamo mapparlo ottimizzando per area. Le statistiche sono le seguenti:

```
sis> map -m 0
sis> print_map_stats
Total Area              = 6024.00
Gate Count              = 174
Buffer Count            = 17
Inverter Count          = 33
Most Negative Slack     = -24.60
Sum of Negative Slacks  = -575.60
Number of Critical PO   = 39
sis>
```

I warning sono stati rimossi dall'output in quanto irrilevanti.