Laboratorio di Architettura degli Elaboratori

Elaborato SIS

A. A. 2014/15

Luca Pizzini VR370826

Sommario

Architettura Generale del Circuito

Il dispositivo implementato monitora i battiti cardiaci di un atleta sotto sforzo aerobico. Si basa su un circuito sequenziale che riceve in ingresso il valore di battiti per minuto (BPM) da un rilevatore toracico e può essere impostato in tre diverse soglie: *sotto-soglia* (SO), *in-soglia* (IS) e *oltre-soglia* (OS).
Il circuito indica in uscita in quale delle soglie si trova attualmente il battito e da quanti secondi.
Il circuito è composto da un'unità di controllo e da una di elaborazione e presenta i seguenti ingressi e uscite:

- INIT[1]: quando vale 1 indica che il circuito ha iniziato a rilevare il battito cardiaco. Il controllore deve passare nello stato conteggio dei secondi trascorsi nella soglia attuale. Finché vale 0 non deve esse fatto alcun conteggio né indicato alcun valore in uscita.
- RESET[1]: posto a 1 indica che il controllore deve essere resettato, ovvero pone a 0 il contatore dei secondi.
- BPM[8]: valore dei battiti per minuto ricevuto dal rilevatore.
- SOGLIA[2]: indica la soglia in cui si trova il battito al momento attuale (00 spento, 01 sotto-soglia, 10 in-soglia, 11 oltre-soglia).
- NUMB[8]: indica i secondi trascorsi nella soglia attuale.
- Controllore e datapath sono collegati da due segnali:
  - START[1]: posto a 1 fa iniziare al datapath la lettura dei BPM, il set dello stato soglia e l'inizio del conteggio dei secondi.
  - SA[2]: imposta il controllore allo stato soglia attuale.

Il rilevatore toracico manda un valore BPM al secondo. Ad ogni valore BPM in ingresso, il circuito controlla l'attuale soglia, imposta lo stato corrispondente e inizia a contare (o incrementa il contatore). Quando il valore BPM non fa parte della soglia attuale, il circuito cambia stato e inizia da zero il conteggio. Ad ogni inserimento di BPM, se INIT è attivo e non vi è RESET, il circuito riporta i valori aggiornati alle uscite.
I valori delle soglie sono:

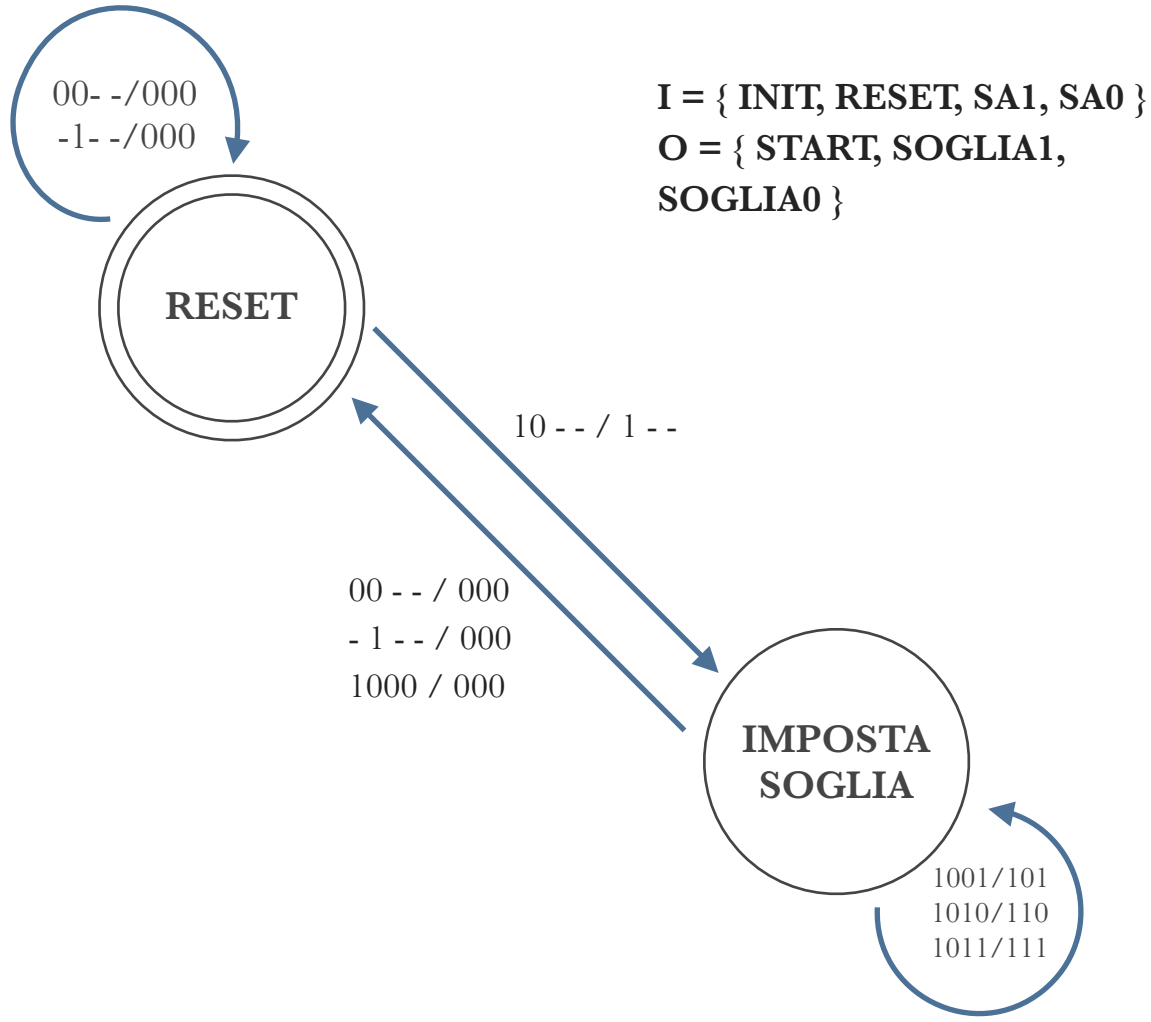
- $0 \leq BPM < 160$: sotto-soglia
- $160 \leq BPM \leq 180$: in-soglia
- $180 < BPM \leq 256$: oltre-soglia

Controllore

Il controllore è una macchina a stati finiti (FSM) che presenta quattro ingressi (INIT, RESET, SA1, SA0) e tre uscite (START, SOGLIA1, SOGLIA0).

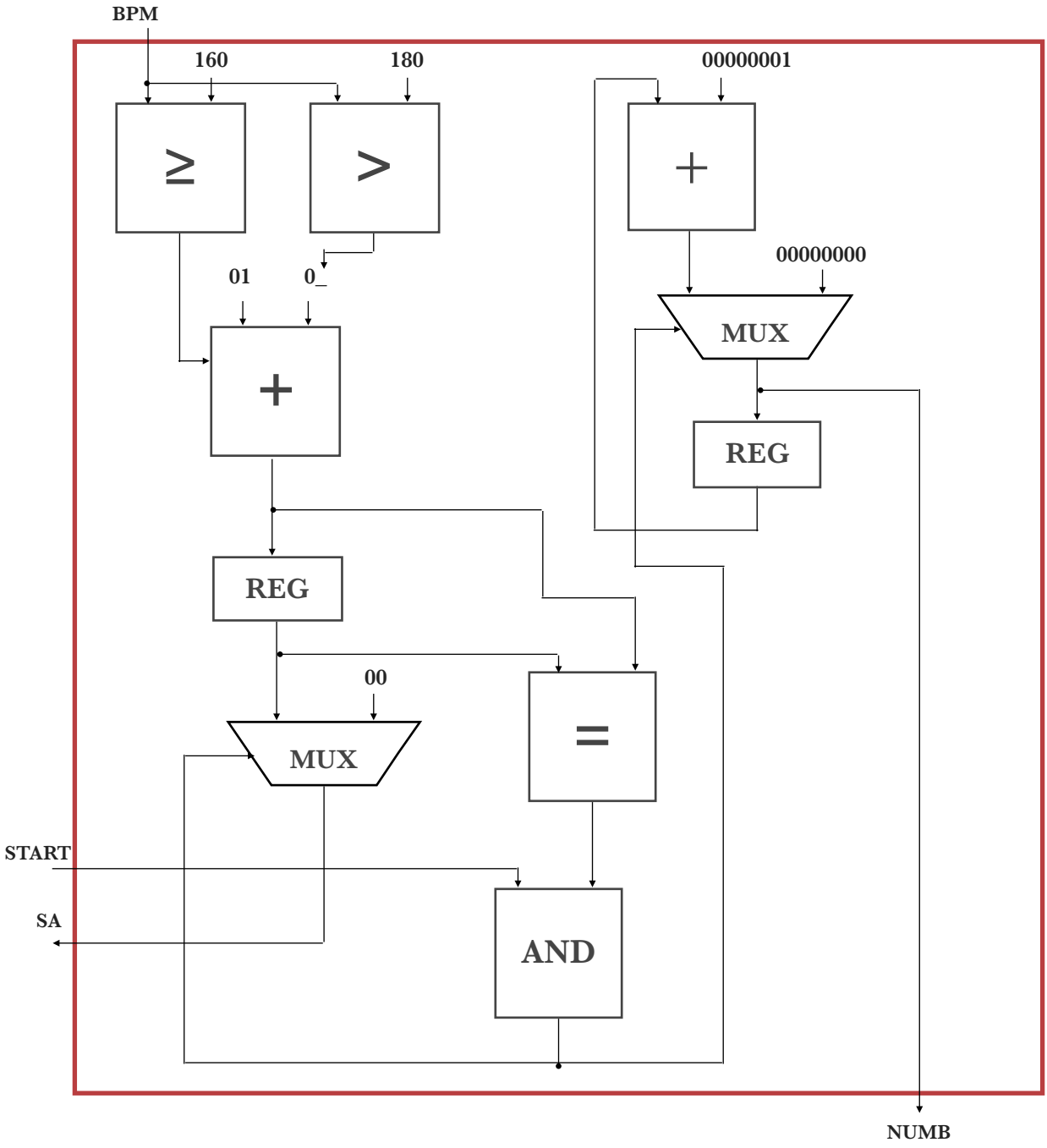
Ho individuato due stati:

- **RESET**: rappresenta lo stato di reset della macchina. La FSM resta in questo stato finchè i segnali di START e RESET non sono attivi oppure è attivo il segnale RESET. Le uscite sono poste a 0 ad indicare che non devono essere fatti conteggi o rilevazioni.
- **IMPOSTA_SOGLIA**: la FSM passa in questo stato quando rileva un segnale START attivo e RESET spento. L'uscita START viene posta a 1 a indicare che viene iniziata la rilevazione della soglia e il conteggio, una volta ricevuto il valore di SA dal datapath le uscite SOGLIA sono specificate in base al valore ottenuto.



Datapath

L'unità di elaborazione riceve in input il valore dei battiti per minuto rilevato dal cardiografo aggiornato ogni secondo e in base al valore START inizia il conteggio dei secondi nella soglia attuale e imposta il controllore nella soglia.



Il circuito riceve in ingresso il valore BMP a 8 bit e inizia il calcolo della soglia attuale: il valore minimo della soglia attiva (01) viene sommato ai due output provenienti da due comparatori in un sommatore a 2 bit. Se BPM è maggiore o uguale a 160 incremento di una unità e imposto la soglia a 10 (in-soglia), se BMP è maggiore a 180 incremento ulteriormente impostando la soglia a 11 (sopra-soglia).
L'uscita del maggiore-uguale è collegata al sommatore tramite l'ingresso CARRY-IN, l'uscita del maggiore è invece posta a 1° bit nel secondo addendo. Il risultato del sommatore passa in un registro a 2 bit dove viene salvato e poi passato ad un multiplexer a 2 ingressi a 2 bit: se attivo l'uscita SA a bit riceve il valore-soglia corrente, altrimenti passo il valore 00 a indicare che il circuito è spento. Il segnale di controllo del multiplexer è ottenuto mettendo in And il segnale START e il risultato di un comparatore che confronta i valori di soglia del ciclo attuale e di quello precedente restituendo 1 in caso di uguaglianza tra i due valori. Quindi il multiplexer riceve in ingresso un segnale attivo se e solo se START è posto a 1 e non è cambiata la soglia tra la rilevazione attuale e la precedente. Lo stesso segnale comanda il multiplexer che gestisce il contatore dei secondi.
La seconda parte dell'elaboratore è composta dal contatore che ha il compito di indicare sull'uscita NUMB a 8 bit il numero di secondi passati nella soglia attuale. Si tratta di un sommatore a 8 bit che riceve in input il valore 1 da sommare al precedente valore contato. Il risultato della somma è posto in ingresso a un multiplexer che lo passa se riceve un segnale attivo, viceversa azzera il contatore. Il registro a 8 bit mantiene il conteggio dei secondi nei cicli di clock successivi.

Statistiche del circuito

Segue una descrizione delle statistiche del circuito prima e dopo l'ottimizzazione per area.

FSM

Dopo aver provato a ridurre il numero di stati con la chiamata *state_minimize stamina* ho codificato gli stati con la funzione *state_assign jedi*:

```
sis> read_blif controllo.blif
sis> state_minimize stamina
Running stamina, written by June Rho, University of Colorado at Boulder
Number of states in original machine : 2
Number of states in minimized machine : 2
sis> state_assign jedi
Running jedi, written by Bill Lin,  UC Berkeley
sis> print_stats
CONTROLLO       pi= 4   po= 3   nodes=  4       latches= 1
lits(sop)=  20  #states(STG)=   2
sis>
```

Dopo l'ottimizzazione con *script.rugged*:

```
CONTROLLO       pi= 4   po= 3   nodes=  5       latches= 1
lits(sop)=  11  #states(STG)=   2
sis>
```

Ho quindi ottenuto una diminuzione del numero di letterali.

Datapath

Il circuito non ottimizzato presenta le seguenti statistiche:

```
sis> read_blif elaborazione.blif
Warning: network `ELABORAZIONE', node "COUT0" does not fanout
Warning: network `ELABORAZIONE', node "COUT1" does not fanout
sis> print_stats
ELABORAZIONE    pi= 9   po=10   nodes= 98       latches=10
lits(sop)= 351
sis>
```

I Warning non sono preoccupanti: non influiranno sull'esecuzione del circuito.

L'obbiettivo è ottenere un minor numero di letterali in quanto è richiesta un'ottimizzazione per area, per fare ciò uso il file *script.rugged* e la funzione fx, ottenendo:

```
ELABORAZIONE    pi= 9   po=10   nodes= 22       latches=10
lits(sop)=  88
sis>
```

FSMD

Essendo l'unione dei due circuiti presenta le seguenti statistiche:

```
MONITORAGGIO_BATTITI    pi=10   po=10   nodes= 27       latches=11
lits(sop)=  99
sis>
```

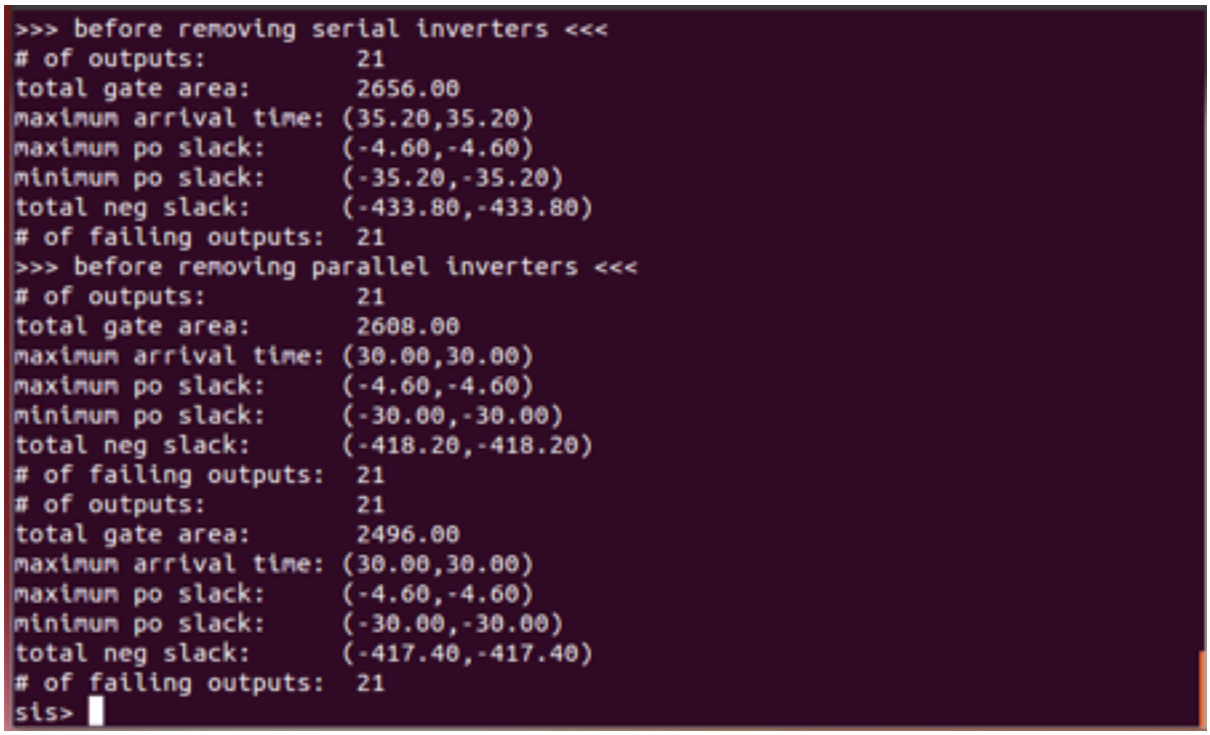
Che dopo l'ottimizzazione diventano:

```
sis> fx
sis> print_stats
MONITORAGGIO_BATTITI    pi=10   po=10   nodes= 21       latches=11
lits(sop)=  96
sis>
```

La riduzione di letterali e nodi risulta limitata in quando FSM e Datapath erano già stati ottimizzati.

Mapping tecnologico

Dopo l'ottimizzazione eseguo la mappatura del circuito mediante la libreria *synch.genlib* ottenendo le seguenti statistiche:



Scelte progettuali

Nell'implementazione del progetto ho effettuato delle scelte su punti non chiariti del progetto o che, secondo me, lasciavano spazio ad interpretazioni:

- Il valore di ingresso 1000 della FSM per come è strutturata l'unità di elaborazione non può presentarsi (se START è attivo non posso ottenere SA=00), l'ho quindi considerato come un'input indicante inattività che dallo stato di conteggio mi porta in stato di reset.
- La scelta di mettere a 1° bit del secondo addendo del sommatore a 2 bit il valore di uscita del maggiore mi permette di risparmiare un sommatore e quindi è stata fatta per semplificare il circuito.
- È stato necessario porre un registro prima dell'uscita SA in quanto altrimenti si sarebbe creato un ciclo.