

# Elaborato SIS – Laboratorio Architettura degli Elaboratori

---

Corso di laurea in Informatica

Alessio Zampieri (VR381609) e Daniela Frison (VR378908)

19/02/2014

## Indice:

Indice:	2
Specifiche:	3
Architettura Generale del circuito:	4
Diagramma degli stati del controllore:	4
Architettura del data-path:	5
Simulazioni:	7
Statistiche del circuito:	7
Prima della minimizzazione:	7
Dopo la minimizzazione:	7
Minimizzazione degli stati:	7
Minimizzazione per area:	7
Numero di gate e ritardo:	8
Prima della minimizzazione:	8
Dopo la minimizzazione per area:	9
Descrizione scelte progettuali effettuate:	9
Note:	9

## Specifiche:

Si progetti un dispositivo per la rilevazione di foratura pneumatici di un'automobile. Il dispositivo dovrà essere modellato come circuito sequenziale (FSM+Datapath) in SIS. Si consideri che un sistema elettromeccanico rilevi le rotazioni dei quattro pneumatici e manda agli ingressi del dispositivo un valore positivo (1) a ogni giro completo dello pneumatico, per ogni pneumatico. Nel caso di una determinata differenza del numero di giri di questi dovuto a foratura, in una distanza di 100 metri, il dispositivo deve accendere una spia di segnalazione foratura sul cruscotto.

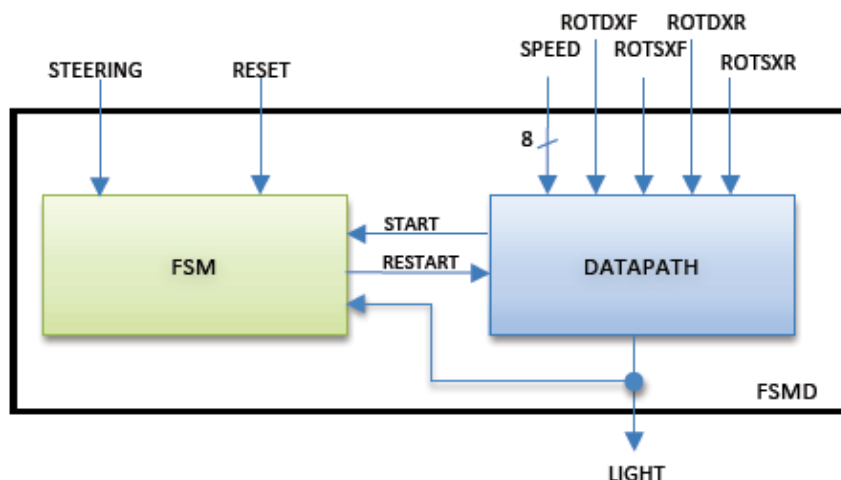
Il circuito dovrà avere i seguenti ingressi e uscite:

- STEERING[1bit]: se uguale a 1, indica che l'automobile è in fase di sterzo e, di conseguenza, il dispositivo non deve azionare alcun controllo, deve resettare qualsiasi conteggio numero giri e confronti pneumatici.
- RESET[1bit]: se uguale a 1, il dispositivo deve resettarsi completamente.
- SPEED[8bit]: indica la velocità istantanea (in Km/h) dell'automobile, da 0 a max 255. Quando è maggiore di 10 Km/h, deve far partire il conteggio delle rotazioni degli pneumatici. Altrimenti lascia il dispositivo in uno stato di PARK.
- ROTDXF[1bit], ROTDXR[1bit], ROTDXR[1bit], ROTDXR[1bit]: se uguale a 1, ogni ingresso indica una completa rotazione degli pneumatici destro anteriore, sinistro anteriore, destro posteriore, sinistro posteriore, rispettivamente. Altrimenti, l'ingresso vale 0.
- LIGHT[1bit]: deve essere impostato a uno per accendere la spia di segnalazione foratura sul cruscotto.

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

- START[1bit]: è messo a uno se la velocità è maggiore di 10 Km/h.
- RESTART[1bit]: è messo a uno se, con una velocità istantanea maggiore a 10 Km/h, l'automobilista aziona lo sterzo. Deve quindi resettare e far ripartire tutti i conteggi giri pneumatici precedenti.
- LIGHT[1bit]: porta in ingresso al controllore l'uscita del dispositivo LIGHT. Quando LIGHT vale 1, il controllore deve portarsi in uno stato (LIGHT) fino al reset completo del dispositivo.

Il dispositivo deve contare il numero di giri di ogni pneumatico e confrontare sia i due anteriori sia i due posteriori. Il confronto è effettuato ogni 100 metri e se la differenza in almeno un confronto supera il 20%, il dispositivo deve alzare il segnale di uscita LIGHT (e portarsi sullo stato di LIGHT). Ogni 100 Metri i contatori devono essere resettati. Si consideri che la distanza (100 metri) sia rilevata prendendo come campione la ruota anteriore destra, per la quale ad ogni rotazione corrisponde una distanza di 2 metri.



## Architettura Generale del circuito:

Il circuito è composto di FSM (controllore) e data-path (elaboratore). I due file che contengono la rappresentazione di queste componenti sono presenti nella cartella allegata con i nomi FSMA.blif e data-path.blif. Il file CONTROLLO\_RUOTE.blif permette di collegare la macchina a stati finiti e l'elaboratore in modo che il circuito sia completo e possa funzionare correttamente.

Di seguito saranno descritte in modo dettagliato tutte le componenti utilizzate.

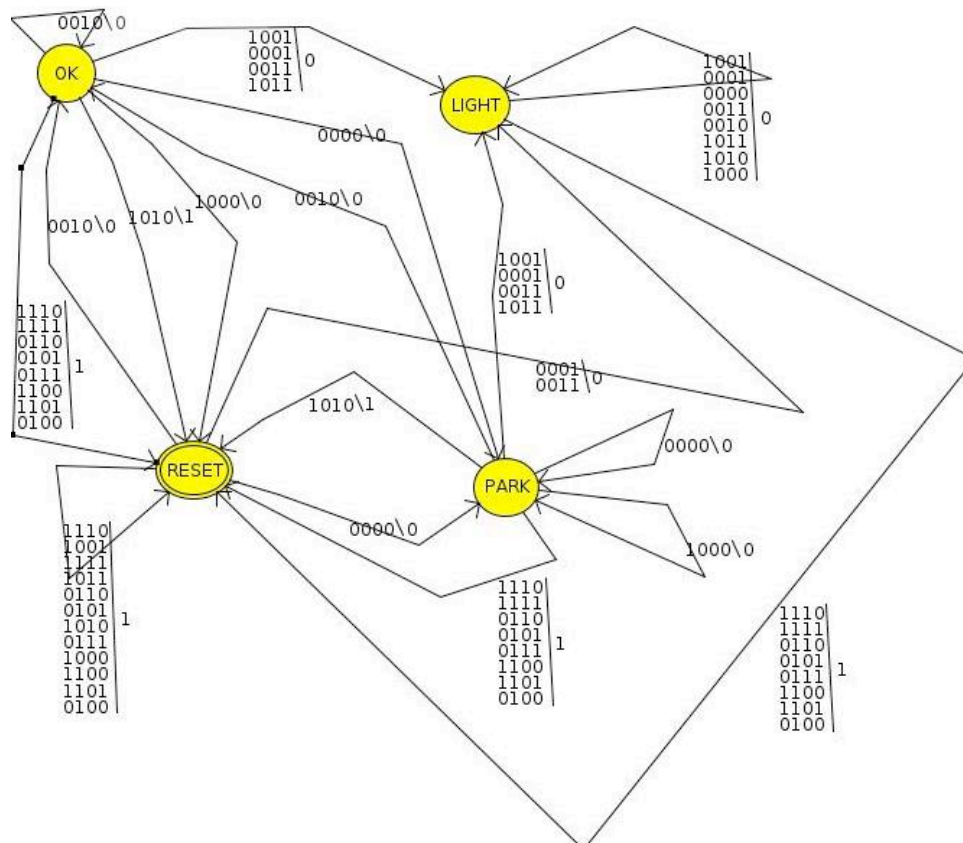
## Diagramma degli stati del controllore:

Riportiamo di seguito lo State Transition Graph (realizzato con BVE):

Per realizzare il controllore di questo circuito, ci siamo serviti di quattro stati:

1. RESET: è lo stato iniziale che comprende lo stato di STERZO (ovvero quando si devono azzerare i conteggi).
2. OK: è lo stato in cui si continua il conteggio; ci si trova in questo stato nel caso in cui la luce sia spenta, la velocità sia maggiore di 10 km/h e non si stia sterzando.
3. LIGHT: è lo stato in cui si accende la luce; da questo stato si esce (e quindi si spegne la luce) solo nel caso in cui reset valga 1.
4. PARK: si raggiunge questo stato quando la velocità è inferiore ai 10 km/h.

In ogni transizione sono indicato quattro ingressi e un'uscita, in ordine: Steering Reset Start Light / Restart.



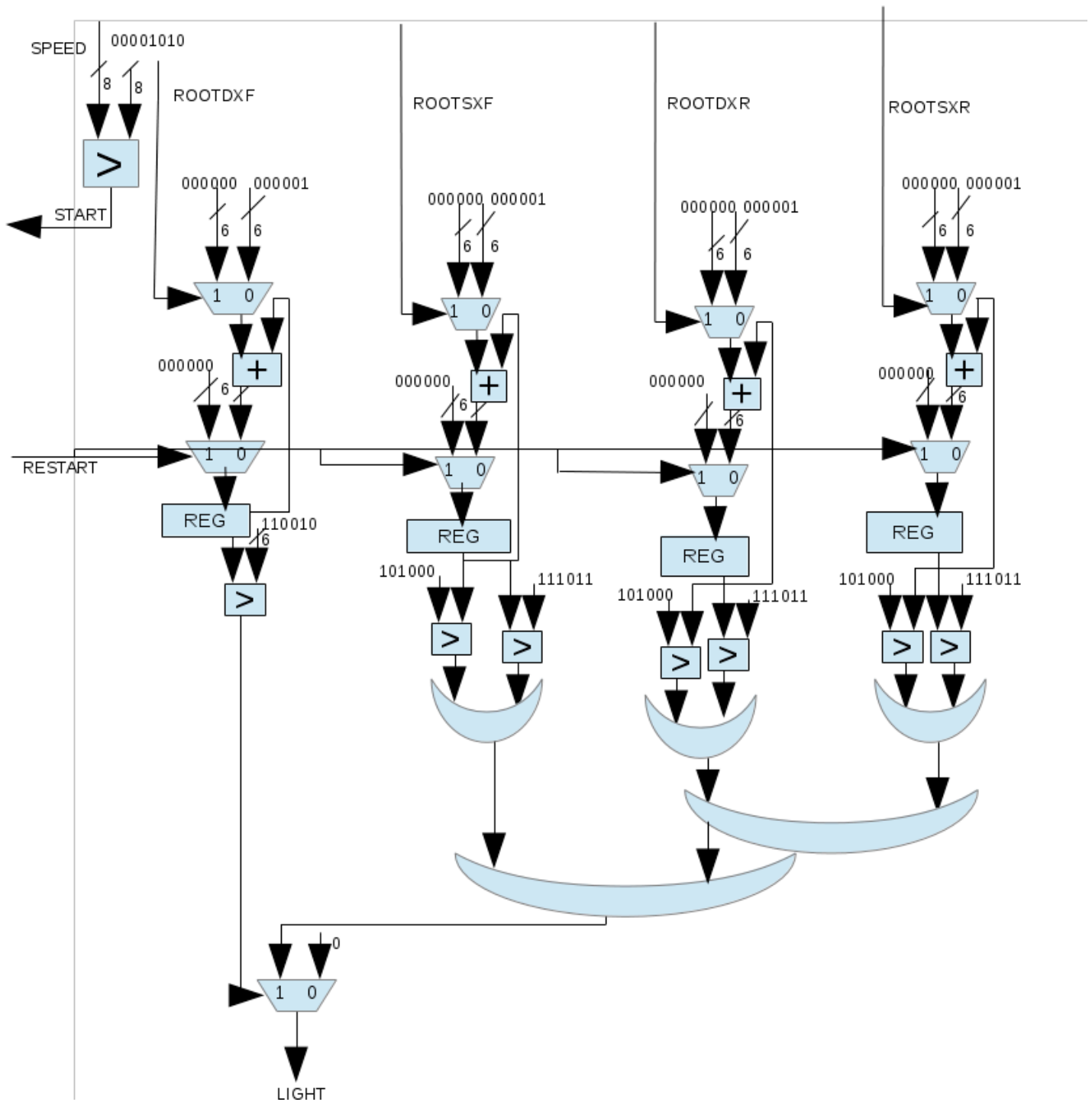
## Architettura del data-path:

Le componenti che abbiamo utilizzato per realizzare il data-path (ovvero l'elaborazione del circuito) sono:

- 1 MAGGIORE\_6bit.blif, che ci permette di confrontare il valore di velocità che riceviamo in ingresso su 8 bit (ingresso SPEED) con la costante 10 codificata su 8 bit. Tale confronto produce il segnale di stato START, che è utilizzato come ingresso dal controllore (FSM).
- 4 MULTIPLEXER\_2ingressi\_6bit.blif, che ci permettono di inserire il valore 0 o 1 codificati su 6 bit in base al valore assunto dagli ingressi ROOTDXF, ROOTSXF, ROOTDXR e ROOTSXR. Questi MULTIPLEXER sono necessari per il sommatore poiché il sommatore riceve due ingressi a n bit per restituire un'uscita a n bit.
- 4 SOMMATORE\_6bit.blif, permettono di eseguire la somma tra il valore di uscita del MULTIPLEXER e il valore precedentemente presente nel registro.
- 4 MULTIPLEXER\_2ingressi\_6bit.blif, utili per resettare i registri nel caso in cui RESTART assuma valore 1.
- 4 REGISTRO\_6bit.blif, consente di memorizzare il valore per poterlo ricordare per il ciclo di clock successivo.
- 7 MAGGIORE\_6bit.blif, usati per permettere il confronto con il numero di giri misurati dalla ruota e le costanti (quarantanove per quanto riguarda la ruota anteriore destra, un valore compreso tra quaranta e sessanta per le altre ruote).
- 5 OR.blif, indispensabili per controllare se almeno una delle ruote ha registrato un valore minore di quaranta o maggiore di sessanta (ovvero il 20% di differenza dal valore misurato dalla ruota anteriore destra che è presa come riferimento).
- 1 MULTIPLEXER\_2ingressi\_1bit.blif, consente di produrre l'uscita LIGHT solo nel caso in cui si sia arrivati a cinquanta con la ruota anteriore destra e almeno una delle altre ruote abbia registrato un valore maggiore di sessanta o minore di quaranta (ovvero almeno una delle ruote sia stata forata). Questo MULTIPLEXER mantiene l'uscita LIGHT a zero finché la ruota anteriore destra non ha misurato i 100 m; solo in quel momento, dispone in uscita il valore risultante dal confronto.

Abbiamo scelto di confrontare il valore misurato con le ruote anteriore sinistra (ROOTSXF), posteriore sinistra (ROOTSXR) e posteriore destra (ROOTDXR) con le costanti 40 e 60, poiché rappresentano il 20% in più e in meno rispetto al valore 50 (100 m).

Di seguito la rappresentazione del data-path:



## Simulazioni:

Gli ingressi sono: steering, reset, speed7, speed6, speed5, speed4, speed3, speed2, speed1, speed0, rootdx, rootsx, rootdxr, rootsxr.

L'uscita è light.

Nella prima simulazione abbiamo considerato la condizione in cui non si ha lo sterzo, si vogliono resettare i registri e la velocità è inferiore a 10 km/h, ma la luce era accesa in precedenza. Infatti, nel ciclo di clock successivo si nota che la luce si spegne ( infatti nella FSM si raggiunge lo stato di Reset).

```
sis> simulate 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
Network simulation:
Outputs: 1
Next state: 010000000000000000000000000000
sis> simulate 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
Network simulation:
Outputs: 0
Next state: 010000000000000000000000000000
```

## Statistiche del circuito:

### Prima della minimizzazione:

```
sis> print_stats
CONTROLLO_RUOTE pi=14    po= 1    nodes=226    latches=26
lits(sop)=1034
```

### Dopo la minimizzazione:

### Minimizzazione degli stati:

il comando usato è: state\_minimize stamina.

```
sis> read_blif FSM4.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
```

### Minimizzazione per area:

Abbiamo lanciato il comando fx, che esegue l'operazione di estrazione, ovvero rappresenta una sotto-espressione comune a più nodi con un nuovo nodo.



```
sis> fx
sis> print_stats
CONTROLLO_RUOTE pi=14    po= 1    nodes=105    latches=26
lits(sop)= 314
sis> source script.rugged
sis> print_stats
CONTROLLO_RUOTE pi=14    po= 1    nodes= 43    latches=26
lits(sop)= 296
sis> █
```

## Numero di gate e ritardo:

### Prima della minimizzazione:

```
sis> read_library synch.genlib
sis> map -s
warning: unknown latch type at node '{{[1]}}' (RISING_EDGE assumed)
warning: unknown latch type at node '{{[2]}}' (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:      27
total gate area:    6288.00
maximum arrival time: (29.80,29.80)
maximum po slack:   (-9.80,-9.80)
minimum po slack:   (-29.80,-29.80)
total neg slack:    (-647.00,-647.00)
# of failing outputs: 27
>>> before removing parallel inverters <<<
# of outputs:      27
total gate area:    5808.00
maximum arrival time: (27.60,27.60)
maximum po slack:   (-9.80,-9.80)
minimum po slack:   (-27.60,-27.60)
total neg slack:    (-589.80,-589.80)
# of failing outputs: 27
# of outputs:      27
total gate area:    5664.00
maximum arrival time: (27.40,27.40)
maximum po slack:   (-9.80,-9.80)
minimum po slack:   (-27.40,-27.40)
total neg slack:    (-584.60,-584.60)
# of failing outputs: 27
```



## Dopo la minimizzazione per area:

```
sis> fx
sis> map -s
>>> before removing serial inverters <<<
# of outputs:          27
total gate area:       6248.00
maximum arrival time:  (29.80,29.80)
maximum po slack:      (-9.80,-9.80)
minimum po slack:      (-29.80,-29.80)
total neg slack:       (-647.00,-647.00)
# of failing outputs:   27
>>> before removing parallel inverters <<<
# of outputs:          27
total gate area:       5768.00
maximum arrival time:  (27.60,27.60)
maximum po slack:      (-9.80,-9.80)
minimum po slack:      (-27.60,-27.60)
total neg slack:       (-589.80,-589.80)
# of failing outputs:   27
# of outputs:          27
total gate area:       5624.00
maximum arrival time:  (27.40,27.40)
maximum po slack:      (-9.80,-9.80)
minimum po slack:      (-27.40,-27.40)
total neg slack:       (-584.60,-584.60)
# of failing outputs:   27
```

## Descrizione scelte progettuali effettuate:

Per quanto riguarda il data-path, com'è descritto sopra, abbiamo fatto direttamente il confronto con le costanti quaranta e sessanta per misurare il 20% di differenza rispetto a cinquanta. In particolare, abbiamo inserito il valore sessanta in modo da considerare anche il caso in cui sia la ruota anteriore destra ad essere bucata.

Per quanto riguarda la FSM, abbiamo accorpato gli stati RESET e STERZO per provare a ridurre al massimo l'area occupata dal circuito. Abbiamo poi considerato che l'uscita RESTART assuma valore uno sia nel caso in cui si abbia lo sterzo (da specifica), sia nel caso in cui l'entrata RESET valga 1 (pensando che il desiderio, in caso di reset, sia quello di resettare i registri).

Per quanto riguarda la minimizzazione, abbiamo minimizzato per area anziché per ritardo, come richiedevano le specifiche.

### Note:

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.