# Relazione dell’elaborato di Architettura degli Elaboratori

A.A. 2013/2014

Studenti:
Cristin Cebotari VR371769
Kevin Daniel Trudu VR369184

# Sommario

# Schema generale del circuito

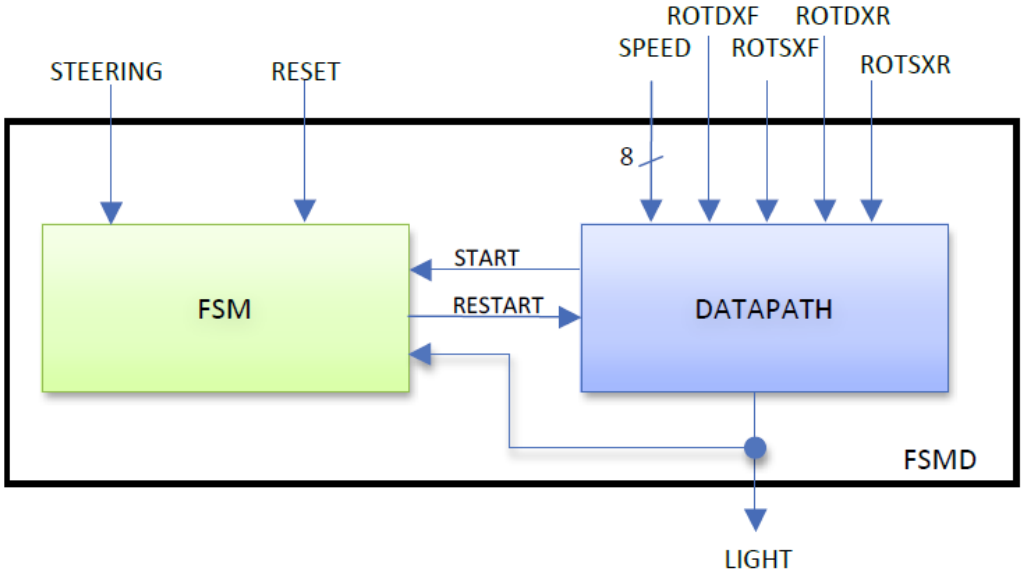
Il dispositivo da noi implementato permette di rilevare la foratura dei pneumatici di un'automobile. Il dispositivo rileva le rotazioni dei quattro pneumatici e manda agli ingressi del dispositivo un valore positivo (1) ad ogni giro completo del pneumatico, per ogni pneumatico. Nel caso di una determinata differenza del numero di giri di questi dovuto a foratura, in una distanza di 100 metri (50 giri della ruota anteriore destra), il dispositivo deve accendere una spia di segnalazione foratura sul cruscotto. Il circuito è composto da un unità di controllo e una per l'elaborazione.

Il circuito presenta 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 dei pneumatici. Altrimenti lascia il dispositivo in uno stato di PARK.
- ROTDXF[1bit], ROTSXF[1bit], ROTDXR[1bit], ROTSXR[1bit]: se uguale a 1, ogni ingresso indica una completa rotazione dei 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 controlla il numero di giri di ogni pneumatico e confronta sia i due anteriori che i due posteriori. Il confronto viene 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 vengono resettati. L a distanza (100 metri) viene rilevata prendendo come campione la ruota anteriore destra, per la quale ad ogni rotazione corrisponde una distanza di 2 metri quindi il campionamento avviene ogni 50 giri della ruota anteriore destra.

# Controllore

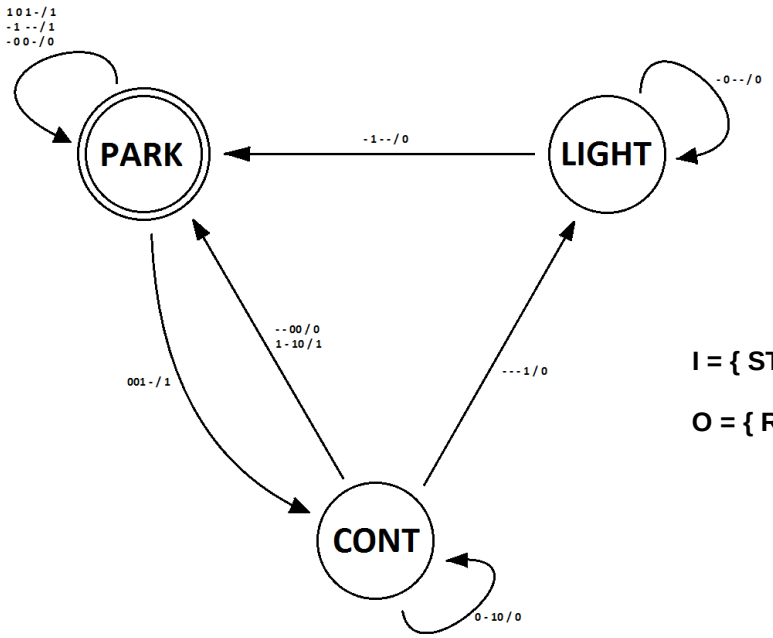
Il controllore è una macchina a stati finiti (FSM) di Mealy che presenta quattro ingressi (STEERING, RESET, START, LIGHT) e un'uscita (RESTART).

Sono stati individuati tre stati che verranno di seguito brevemente descritti:

- **PARK**: è lo stato di reset e la FSM resta in questo stato finché l'autovettura non supera la velocità di 10 km/h. Rappresenta quindi lo stato in cui la velocità dell'autovettura è al di sotto dei 10 km/h oppure quello in cui la macchina è in sosta.

- **CONT**: la FSM arriva in questo stato quando il sensore rileva una velocità superiore di 10 km/h, viene attivato il datapath che inizia a rilevare i giri delle ruote e a calcolare la differenza dei giri di ogni coppia di pneumatici.

- **LIGHT**: è lo stato in cui si ritrova la FSM dopo aver ricevuto in ingresso il segnale dal datapath che corrisponde all'accensione della spia. La FSM ignora ogni ingresso e si pone in attesa di essere resettata.



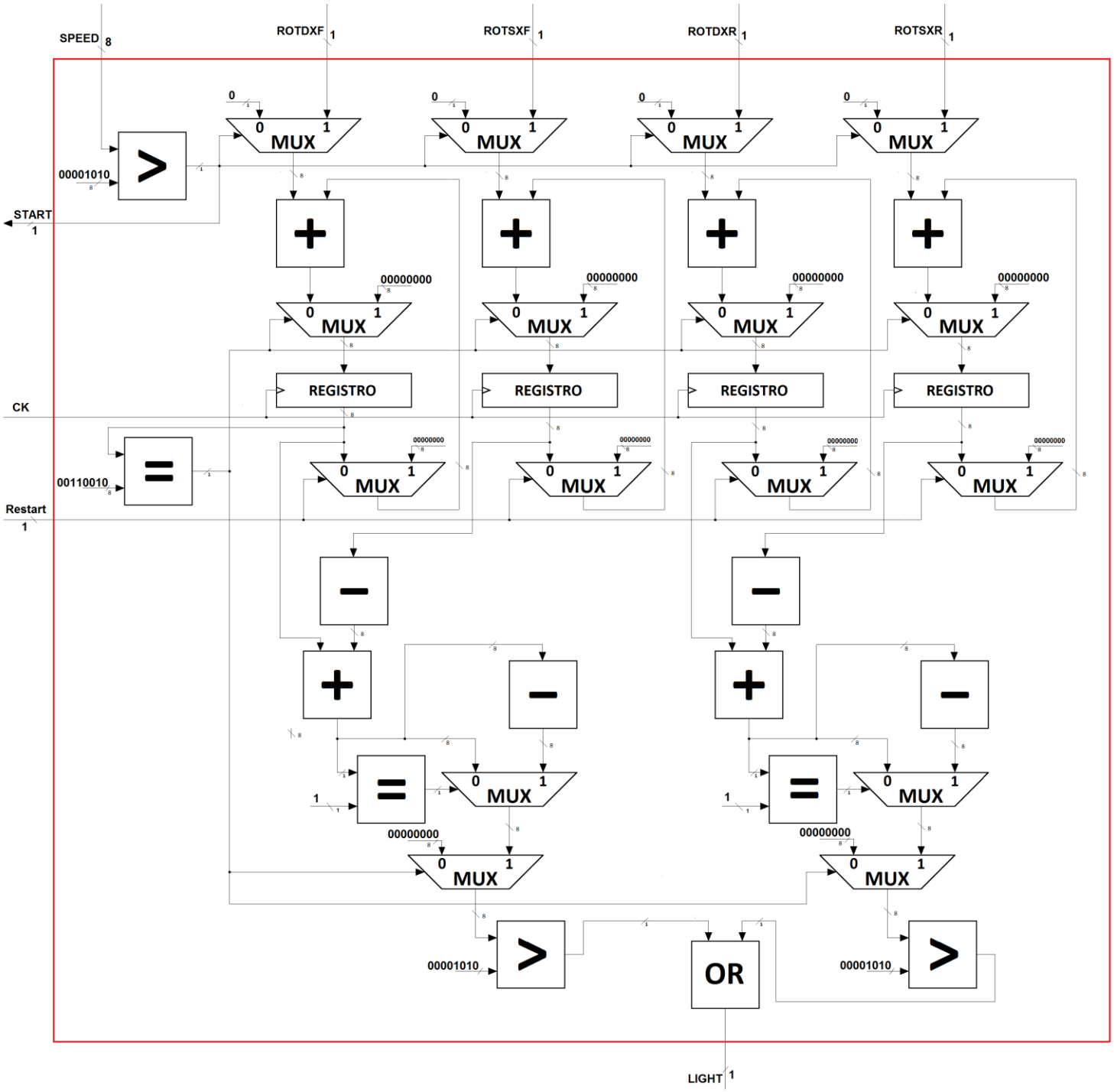

**I = { STEERING, RESET, START, LIGHT }**

**O = { RESTART }**

# Datapath

Il datapath è quella parte del circuito che si occupa principalmente di eseguire calcoli.
Nel nostro caso riceve in input la velocità (SPEED), i giri di ogni ruota(ROTDXF[1bit], ROTSXF[1bit], ROTDXR[1bit], ROTSXR[1bit ]: ), il segnale di inizio da parte del controllore (RESTART) e presenta una sola uscita (LIGHT) collegata al controllore.

Vediamo ora lo schema del circuito:



Per iniziare il conteggio del giro delle ruote, l'automobile deve presentare una velocità strettamente maggiore ai 10 km/h quindi la velocità (SPEED[8bit]) viene confrontata con la costante 10 su otto bit. Se la velocità è effettivamente maggiore di 10, il maggiore produce un segnale (START) pari ad 1 che entra come bit di selezione nei primi quattro multiplexer, i quali ricevono come primo input la costante zero e come secondo input il giro delle ruote, quindi se dal maggiore esce uno zero il multiplexer farà uscire uno zero perché siamo sotto i 10km/h e il conteggio delle ruote non deve avere luogo, altrimenti fa uscire i giri che hanno fatto le ruote.
Collegati ai multiplexer troviamo un componente che esegue l'addizione e riceve come primo addendo l'uscita dei multiplexer di cui abbiamo già dato la spiegazione e come secondo addendo l'uscita di un altro multiplexer collegato ad un registro che verrà specificato tra poco, questo serve a fare il conteggio del giro delle ruote. Le uscite dei sommatori vanno collegate all'entrata dei multiplexer che azzerano i registri come primo input, e ricevono come secondo input la costante zero, ad otto bit, come bit di selezione abbiamo scelto di far entrare l'output di un' uguale che verifica se il contenuto del registro corrispondente alla ruota anteriore destra presenta il valore 50 in binario, nel qual caso, tutti i registri vanno azzerati poiché significa che l'automobile ha percorso 100 metri, quindi esce il valore 1 che entra nei multiplexer come bit di selezione e fanno uscire il valore zero che entra nei registri.
Le uscite dei registri sono collegate a dei multiplexer che ricevono come bit di selezione l'input chiamato RESTART che arriva dalla FSM; quando quel bit è pari ad 1 significa che dobbiamo azzerare i registri perché abbiamo superato i 10 km/h, in questo modo abbiamo trovato un metodo semplice per poter azzerare i nostri registri poiché all'accensione del nostro circuito, i registri possono presentare valori casuali e se iniziamo a fare la somma partendo dal quel valore, il risultato è sbagliato, quindi se RESTART vale 1 i multiplexer fanno passare la costante 0 come secondo addendo della somma, se vale 0 invece fanno uscire il valori contenuti nei registri e vanno a sommarsi al giro che la ruota compie in quel ciclo di clock.
Abbiamo bisogno di eseguire la sottrazione tra i registri corrispondenti alle ruote anteriori e anche tra quelli corrispondenti alle ruote posteriori per verificare se vi è una differenza maggiore del 20%. Abbiamo bisogno di un sottrattore, ma dalla teoria sappiamo che per sottrarre 2 numeri in binario occorre sommare il primo addendo al secondo in complemento a 2, quindi i valori dei registri 2 e 4 entrano nel componente che esegue il complemento a 2 che sullo schema presenta il segno "meno" ed è composto da un negatore (NOT) collegato ad un sommatore che somma all'uscita del NOT la costante 1, e otteniamo in questo modo il secondo addendo in complemento a 2 che entra nei componenti somma prendendo come primo addendo il valore dei registri rispettivamente 1 e 3 e ne eseguono la somma.
Il risultato della somma è in complemento a 2, questo significa che se il primo addendo è maggiore del secondo (caso in cui la ruota anteriore destra è forata) allora prendiamo come "buono" il risultato della somma, ma se il secondo addendo è maggiore del primo (caso in cui la ruota anteriore sinistra è forata), allora otteniamo un numero sempre in complemento a 2 ma negativo, cioè con il bit più significativo pari ad 1, che il nostro circuito non interpreta come numero negativo, quindi nel secondo caso dobbiamo eseguire di nuovo il complemento a 2 sul risultato della somma per "trasformarlo" nel risultato della "sottrazione" senza il segno ovvero in modulo. A questo punto, il bit più significativo della somma entra in un componente che verifica se è uguale alla costante uno, in tal caso significa che la "sottrazione" ha prodotto un numero negativo e perciò per poter essere interpretato, questo numero ha bisogno di essere trasformato faccendone il complemento a 2 in modo da ottenere il modulo della sottrazione così facendo abbiamo risolto i due casi in cui la prima ruota fa più giri della seconda e viceversa. Il bit risultante dall' uguaglianza

precedente serve come bit di selezione per il multiplexer che riceve come primo input la "sottrazione" e come secondo input il modulo della “sottrazione”.
Se il bit di selezione vale 0 significa che la “sottrazione” ha prodotto un numero positivo, quindi che la prima ruota ha effettuato più giri della seconda, nel caso in cui vale 1 significa che la "sottrazione" ha prodotto un numero negativo e quindi scegliamo quel risultato dopo aver eseguito il complemento a 2.
All'uscita dei multiplexer precedenti viene collegato un altro multiplexer che serve a verificare se la ruota anteriore destra ha effettuato 50 giri ricevendo come bit di selezione l'uscita del componente di uguaglianza precedentemente citato. Se il bit di selezione è pari a 0 esce il valore 0 dal multiplexer che va confrontato con la costante 10 che rappresenta il 20% di 50, quindi se il valore è maggiore di 10 allora esce il valore 1 che entra come input nell' ultimo componente(OR) il quale riceve i 2 bit dai maggiori e ne fa quindi l'or. Basta che uno dei due confronti fra le ruote anteriori e quelle posteriori risulti superiore a 10 per far in modo che si accenda la spia. Infine dall' or ricaviamo l'uscita che va collegata alla FSM come input (LIGHT).

# 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 il comando “state_minimize stamina”, ma il programma non è riuscito a ridurre il numero di stati.

Abbiamo quindi codificato gli stati con “state_assign jedi” ottenendo le seguenti statistiche:

```
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 : 3
Number of states in minimized machine : 3
sis> state_assign jedi
Running jedi, written by Bill Lin,  UC Berkeley
sis> print_stats
FSM             pi= 4   po= 1   nodes=  3       latches= 2
lits(sop)=  24  #states(STG)=   3
sis>
```

Che dopo l’ottimizzazione diventano:

```
FSM             pi= 4   po= 1   nodes=  4       latches= 2
lits(sop)=  21  #states(STG)=   3
sis>
```

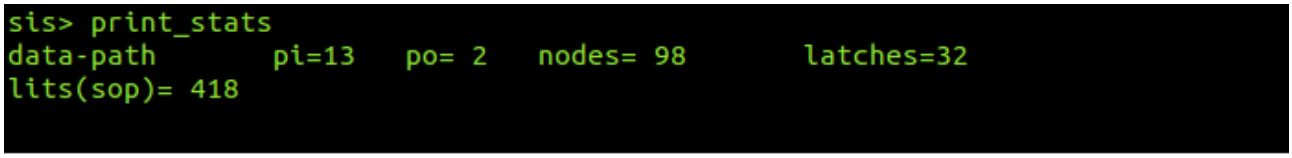
Siamo riusciti ad ottenere una diminuzione del numero di letterali proprio come volevamo.

Il **datapath** da noi scritto presenta le seguenti statistiche:


I warning non sono preoccupanti, in quanto non interferiscono con il risultato del nostro circuito. Viene richiesta l'ottimizzazione per area, quindi cercheremo di ridurre al minimo il numero di letterali.

Per fare ciò abbiamo creato un file che una volta chiamato con il comando "source" fa eseguire una serie di comandi tra cui anche lo "source *script.rugged*", che tra quelli predefiniti di SIS è quello che in generale offre il miglior risultato.

Dopo ulteriori ottimizzazioni le statistiche finali del circuito sono le seguenti:



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

```
sis> print_stats
FSMD            pi=14   po= 1   nodes=102       latches=34
lits(sop)= 439
sis>
```

Passiamo quindi all'ottimizzazione.

```
FSMD            pi=14   po= 1   nodes=102       latches=34
lits(sop)= 439
sis>
```

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

# Mapping

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*". Una volta mappato il circuito presenta le seguenti statistiche:

```
>>> before removing serial inverters <<<
# of outputs:          35
total gate area:       10000.00
maximum arrival time: (57.80,57.80)
maximum po slack:     (-34.00,-34.00)
minimum po slack:     (-57.80,-57.80)
total neg slack:      (-1733.40,-1733.40)
# of failing outputs:  35
>>> before removing parallel inverters <<<
# of outputs:          35
total gate area:       9952.00
maximum arrival time: (58.00,58.00)
maximum po slack:     (-34.20,-34.20)
minimum po slack:     (-58.00,-58.00)
total neg slack:      (-1740.40,-1740.40)
# of failing outputs:  35
# of outputs:          35
total gate area:       9424.00
maximum arrival time: (56.60,56.60)
maximum po slack:     (-34.00,-34.00)
minimum po slack:     (-56.60,-56.60)
total neg slack:      (-1703.40,-1703.40)
# of failing outputs:  35
sis>
```

Il total gate area è 10000.00 mentre il cammino critico è pari a 57.80 .

# Scelte progettuali

Implementando il nostro progetto abbiamo riscontrato alcuni punti che risultavano essere non del tutto chiari, non essendo presenti alcune indicazioni nelle specifiche ci siamo permessi di affermare alcune ipotesi per quanto riguarda il funzionamento del nostro circuito che ora elenchiamo:

1) Sappiamo che se una ruota fa più del 20% dei giri dell'altra significa che è forata e quindi la spia deve accendersi, ma il testo non specificava cosa fare nello sfortunatissimo caso in cui 2 ruote sono forate e fanno lo stesso numero di giri o comunque un numero di giri che non differisce per più di 20% del totale giri della ruota di riferimento, quindi abbiamo escluso questo caso dal nostro circuito, cioè la luce non si accenderebbe in quel caso.

2) All'accensione il nostro circuito presenta valori casuali nei registri, ammettendo il caso in cui ci sia proprio il valore 50 nel primo registro, il circuito andrebbe a eseguire tutti i calcoli e se dal calcolo risulta una differenza di più del 20% allora la spia si accenderebbe, ma questo è improbabile.

3) Per quanto riguarda il reset dei registri abbiamo per semplicità di implementazione deciso che ogni qualvolta che l'auto supera i 10 km/h i registri vengono resettati e posti a 0, questo significa che se l'utente non fa altro che accelerare e frenare facendo così oscillare la velocità sopra e sotto i 10km/h e uno dei pneumatici è forato, allora la spia non si accenderebbe mai poiché non essendo la velocità dell'auto superiore a 10 km/h per almeno 100 metri (50 giri di ruota) i registri non presenterebbero mai il valore 50 e il datapath non darebbe il segnale corrispondente all'accensione della spia.