

# Elaborato SIS Lab. Architettura degli Elaboratori

Corso di laurea in Informatica

---

Anno accademico 2015/2016

Sandu Alin Cristinel (VR400082)

Bragoi Vladislav (VR397344)



## INDICE

---

Specifiche del circuito .....	4
Architettura generale del circuito .....	5
Diagramma degli stati del controllore .....	5
Architettura del data-path.....	6
Statistiche del circuito .....	7
Scelte progettuali.....	10

## SPECIFICHE DEL CIRCUITO

Il circuito implementato è un dispositivo per il monitoraggio di un motore a combustione interna. Esso riceve in input il numero di giri/minuto del motore (RPM) e ne fornisce in uscita una modalità di funzionamento tra quelle elencate:

modalità/stato	valore assunto	codifica
Spento:	RPM = 0	[00]
Sotto – Giri (SG):	$0 < \text{RPM} < 2000$	[01]
Regime – Ottimale (OPT):	$2000 \leq \text{RPM} \leq 4000$	[10]
Fuori – Giri (FG):	RPM > 4000	[11]

Inoltre fornisce:

1. Da quanto tempo il sistema si trova nello stato attuale;
2. Un segnale di allarme se il sistema è in stato di “Fuori – Giri” da più di 15 secondi;

Il circuito è composto da un *controllore* e da un *datapath* con le seguenti specifiche:

	Tipo	Nr. Bit	Descrizione
INIT	I	1	Se vale 1 indica che il circuito ha iniziato la rilevazione del numero di giri. Se vale 0 non deve essere fatto alcun conteggio né indicato alcun valore in uscita.
RESET	I	1	Se posto a 1 il controllore deve essere resettato, ovvero il contatore dei secondi deve essere posto a zero.
RPM	I	13	Valore del numero di giri ricevuto dal rilevatore (valore massimo 6500).
MOD	O	2	Indica in quale modalità di funzionamento si trova l'apparecchio al momento corrente.
NUMB	O	8	Indica i secondi trascorsi nell'attuale modalità.
ALM	O	1	Segnala il superamento del tempo limite in FG.

START	C.S.	1	Messo a 1 fa iniziare al datapath la lettura dei RPM, il set dello stato di soglia, e l'inizio del conteggio secondi in soglia.
SA	S.S.	2	Imposta il controllore allo stato di soglia attuale.
TS	S.S.	1	Segnala al controllore il superamento della soglia di tempo.

[ I = Input / O = Output / C.S. = Segnale di Controllo / S.S. = Segnale di Stato ]

Considerato che il rilevatore manda un valore di RPM al secondo, il ciclo di clock è settato a 1 ciclo/secondo. Ad ogni valore di RPM in ingresso, il circuito controlla l'attuale soglia, imposta lo stato corrispondente e inizia a contare (o incrementa il contatore). Nel momento in cui il valore RPM non fa parte dell'attuale soglia, il circuito cambia stato e inizia da zero il conteggio. Ad ogni inserimento di RPM, se INIT è attivo e non vi è reset, il circuito riporta i valori aggiornati delle uscite.

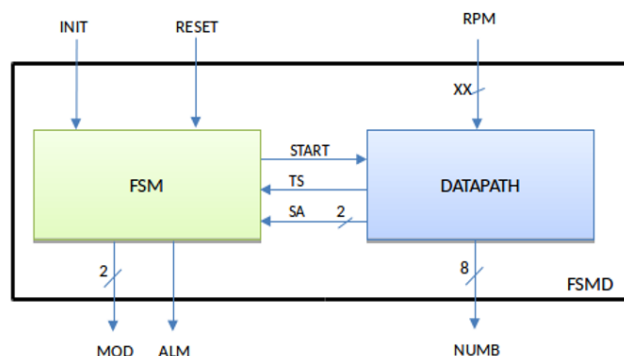


Figura 1: Modello generale

## ARCHITETTURA GENERALE DEL CIRCUITO

Il circuito è composto di FSM (nel file controllore.blif), DATA-PATH (elaboratore.blif) e FSM-D (fsm.d.blif) che ne unisce le funzionalità e permette la corretta esecuzione del sistema, rispettando le specifiche date.

Di seguito è riportata la descrizione dettagliata di ciascun componente utilizzato.

## DIAGRAMMA DEGLI STATI DEL CONTROLLORE

Lo State Transition Graph (realizzato con BVE) presenta la seguente struttura:

- Inputs: INIT, RESET, TS, SA1\*, SA0\*

- Outputs: START, MOD1\*, MOD0\*, ALM.

\*N.B. La progettazione in SIS richiede l'utilizzo di collegamenti a 1 solo bit. Per ovviare a tale problema, i segnali a 2 bit sono stati separati in 2 segnali da 1 bit.

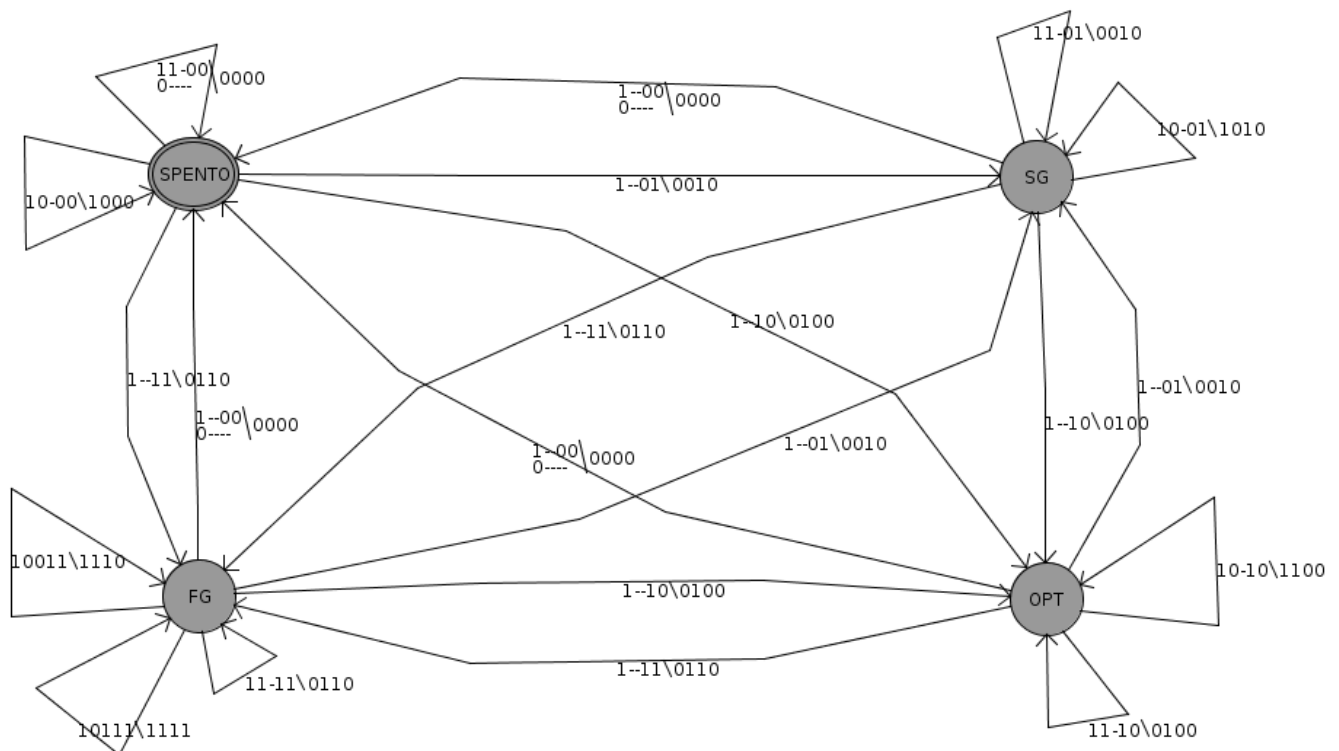


Figura 2: controllore (FSM)

Gli stati utilizzati sono 4:

- I. SPENTO [00]: questo stato ha due funzioni: è stato scelto sia come stato di Reset, sia come stato di appoggio se il rilevatore produce 0 RPM;
- II. SG [01]: è lo stato in cui gli RPM sono inferiori a 2000, ma diversi da 0;
- III. OPT [10]: lo stato OPT è impostato come regime di lavoro ottimale del motore;
- IV. FG [11]: se il motore rimane in questo stato per più di 15 secondi, viene attivato il segnale di allarme, motivo per cui è presente un 3° auto - anello su FG. L'utente può comunque spegnere tale segnale azzerando il conteggio dei secondi trascorsi tramite il segnale RESET.

In ogni stato i secondi vengono incrementati di 1 ad ogni ciclo di clock e azzerati soltanto nel caso in cui l'utente decida di resettare tramite apposito segnale. Ogni transizione da uno stato all'altro comporta comunque l'azzeramento del contatore. Sono stati inoltre previsti collegamenti da ogni stato a tutti gli altri, in quanto, in un secondo gli RPM del motore possono variare notevolmente.

## ARCHITETTURA DEL DATA-PATH

---

L'architettura del data-path è divisa principalmente in due parti: la prima che realizza il contatore a 8 bit\* e la seconda che valuta lo stato/soglia sulla base dei giri effettivi rilevati (RPM).

I componenti utilizzati sono:

- Mux a 8 bit: resetta il valore del contatore (attraverso l'inserimento nel circuito della costante 0 a 8 bit) se il segnale di selezione START vale 0. Se START vale 1, il mux viene attraversato dai bit provenienti dal sommatore;
- Registro a 8 bit: memorizza il valore proveniente dal mux a esso sovrastante;
- Sommatore a 8 bit: incrementa di 1 (costante a 8 bit) il valore del registro, ad ogni ciclo di clock;
- Maggiore a 8 bit: valuta se l'uscita del mux (e di conseguenza il valore aggiornato del registro\*) è maggiore di 15 (costante a 8 bit). In questo caso, il suo output corrisponderà al segnale di uscita TS (Segnale di Stato) che verrà gestito dal controllore;
  
- Minore 13 bit: confronta il valore degli RPM con la costante 2000 a 13 bit. Se la condizione è verificata, la sua uscita vale 1;
- Maggiore a 13 bit: confronta il valore degli RPM con la costante 4000 a 13 bit. Anche in questo caso l'uscita vale 1 se la condizione è verificata;
- Uguale a 13 bit: si rimanda al paragrafo “Scelte progettuali” per la descrizione del funzionamento di tale componente;
- 3 Mux a 2 bit di ingresso: i loro segnali di selezione provengono dai comparatori a 13 bit posti alla loro sinistra. Attraverso dei collegamenti in serie, i tre selezionatori impostano l'uscita SA a valori costanti 01 10 11 o 00.

Come funziona il circuito:

Il circuito del data-path è stato progettato in modo tale che funzioni sempre, indipendentemente dal funzionamento del controllore. Esso infatti rileva costantemente il valore degli RPM e ne fornisce in output lo stato in cui si trova, e sarà compito del controllore scegliere se considerarne i segnali oppure produrre un output costante. Tale soluzione è stata pensata per utilizzare il minor numero di componenti possibili, ed inoltre, un eventuale mux posto all'inizio del circuito (per eventualmente bloccare la rilevazione degli RPM) avrebbe provocato un ritardo di funzionamento o comunque un errore nella realizzazione della FSM.

Il contatore invece, finché START vale 0 continua a caricare il valore “0” nel registro, così che, all'avvio, sarà pronto per iniziare il conteggio effettivo dei secondi e non saranno necessari altri

---

\* [si rimanda al paragrafo “SCELTE PROGETTUALI” per maggiori dettagli]

componenti o controlli per azzerare il suo valore. In questo modo non si producono ritardi nell'esecuzione del circuito e i secondi presentati sono quelli effettivamente trascorsi.

Con questo metodo, pertanto, si lascia al controllore il compito di gestire questa fase di esecuzione impostando a dovere il valore di START.

Il confronto finale, infine, controlla se il circuito ha superato i 15 secondi di funzionamento. In tal modo il data-path produrrà sempre il segnale TS=1 quando il contatore supera il limite impostato, ma anche qui, il valore verrà preso in considerazione dal controllore solo se in stato FG.

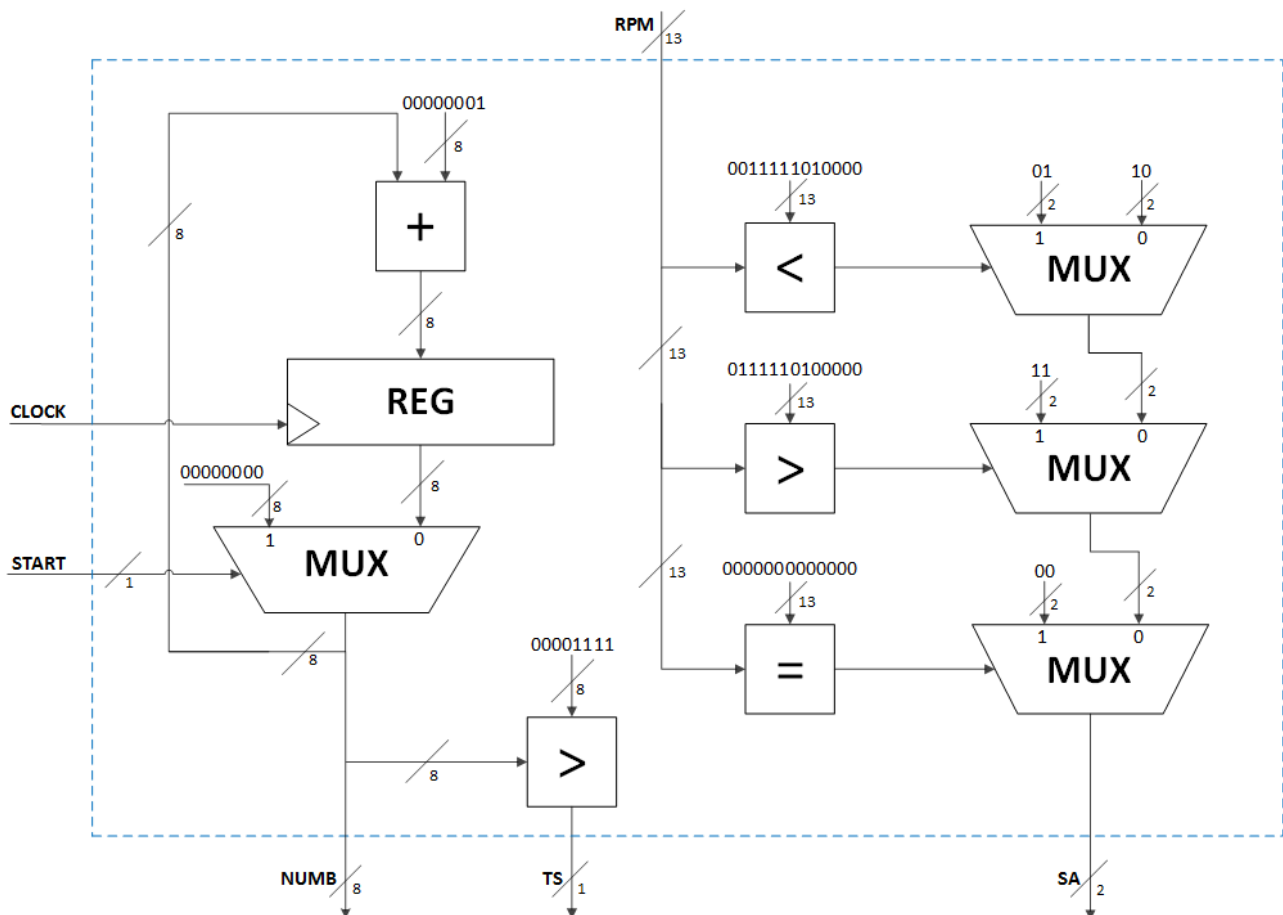


Figura 3: elaboratore (DATA-PATH)

## STATISTICHE DEL CIRCUITO

Di seguito vengono presentate le statistiche dei vari componenti.

**FSM:** Al controllore sono state assegnati le funzioni di uscita ( $\lambda$ ) e stato prossimo ( $\delta$ ) con il comando “stg\_to\_network”, visto che la codifica degli stati era già stata fatta manualmente. Inoltre un’eventuale minimizzazione con stamina non porta a risultati migliori, pertanto la fsm è stata minimizzata solamente con lo “script.rugged”. Le statistiche prima della minimizzazione sono:

```
sis> rl controllore.blif
sis> stg_to_network
sis> print_stats
CONTROLLORE      pi= 5   po= 4   nodes=  6       latches= 2
lits(sop)=  39   #states(STG)=  4
```

Figura 4: statistiche della FSM

e dopo la minimizzazione:

```
sis> source script.rugged
sis> print_stats
CONTROLLORE      pi= 5   po= 4   nodes=  4       latches= 2
lits(sop)=  31   #states(STG)=  4
sis>
```

Figura 5: statistiche della FSM minima

**DATA-PATH:** è stato scelto di non minimizzare il data-path, in quanto minimizzare direttamente la fsmd completa porta a un risultato migliore in termini di area.

Ecco le statistiche a confronto:

```
sis> rl fsmd.blif
sis> print_stats
MOTORE           pi=15   po=11   nodes= 31       latches=10
lits(sop)= 134
sis> source script.rugged
sis> fx
sis> source script.rugged
sis> fx
sis> print_stats
MOTORE           pi=15   po=11   nodes= 33       latches=10
lits(sop)= 130
sis>
```

Figura 6: statistiche della FSMD con data-path minimizzato



```

sis> print_stats
MOTORE      pi=15   po=11   nodes=166   latches=10
lits(sop)= 653
sis> source script.rugged
sis> fx
sis> source script.rugged
sis> fx
sis> print_stats
MOTORE      pi=15   po=11   nodes= 34   latches=10
lits(sop)= 126
sis>

```

Figura 7: statistiche della FSM D con data-path NON minimizzato

Nota: i Warning prodotti da SIS durante il caricamento del file completo, non sono rilevanti ai fini del funzionamento del circuito. Inoltre, sono state provate diverse combinazioni per minimizzare al minimo l'area, ma alla fine la combinazione migliore la si è ottenuta con i comandi illustrati nelle immagini sopra (tra i quali è presente più volte lo “script.rugged” che, tra quelli disponibili in SIS, offre risultati migliori).

Le specifiche prevedevano inoltre di mappare il design sulla libreria tecnologica “synch.genlib”, pertanto la [figura 8](#) ne illustra le statistiche, comprese il numero di gate e il ritardo del circuito completo:

```

sis> map -m 0
sis> map -s
>>> before removing serial inverters <<<
# of outputs:      21
total gate area:    3064.00
maximum arrival time: (27.80,27.80)
maximum po slack:   (-7.60,-7.60)
minimum po slack:   (-27.80,-27.80)
total neg slack:    (-398.80,-398.80)
# of failing outputs: 21
>>> before removing parallel inverters <<<
# of outputs:      21
total gate area:    3048.00
maximum arrival time: (27.80,27.80)
maximum po slack:   (-7.60,-7.60)
minimum po slack:   (-27.80,-27.80)
total neg slack:    (-399.00,-399.00)
# of failing outputs: 21
# of outputs:      21
total gate area:    2808.00
maximum arrival time: (27.60,27.60)
maximum po slack:   (-7.60,-7.60)
minimum po slack:   (-27.60,-27.60)
total neg slack:    (-394.80,-394.80)
# of failing outputs: 21
sis>

```

Figura 8: mapping tecnologico sulla FSM D

## SCELTE PROGETTUALI

---

La progettazione di questo circuito ha richiesto particolare attenzione su alcuni aspetti critici del suo funzionamento e implementazione:

1.
  - a) Su ogni stato sono stati posti 2 auto-anelli equivalenti a RESET 0 e 1 per quei valori - di RPM - che mantengono il circuito nello stesso stato e per i quali l'utente decida di azzerare il conteggio. L'unico stato che presenta 3 auto-anelli, il FG, è in dipendenza del cosiddetto stato di allarme che però non annulla il conteggio dei secondi trascorsi, né porta in una nuova transizione verso nuovi stati. La scelta di non creare un altro stato è dettata semplicemente dal fatto di lasciar continuare il conteggio del tempo trascorso.
  - b) Sono state previste transizioni che rendono la fsm completamente connessa. Si è pensato che una forte accelerazione potrebbe provocare un brusco cambiamento nei valori degli RPM: naturalmente il circuito deve essere in grado quindi, di analizzare e fornire istantaneamente lo stato esatto corrispondente al valore dei giri motore; inoltre, la transizione ad allarme non è ovviamente possibile in un solo ciclo di clock per specifica progettuale (devono infatti essere trascorsi almeno 15 secondi nello stato FG e solo al 16° secondo l'allarme viene attivata).
2.
  - a) Le specifiche del progetto indicano che il segnale NUMB dev'essere posto a 8 bit. La scelta delle altre costanti e valori sulla stessa linea sono quindi condizionate da questa specifica.
  - b) Il valore di NUMB a 8 bit arriverà al massimo a indicare  $(2^8 - 1)$  secondi, cioè al massimo 255 secondi. Tale valore è poco espressivo del numero di minuti passati in un determinato stato. Siccome le specifiche del progetto non richiedevano la trasformazione in sistema sessagesimale si è scelto di demandare questa trasformazione a un ulteriore circuito, oppure si affronterà tale problema in un'eventuale implementazione fisica del circuito.
  - c) Collegare NUMB all'uscita del mux a 8 bit è stata una scelta consapevole. Essa è stata fatta per gestire al meglio il tempo di risposta del circuito rispetto alle simulazioni in SIS. Se per esempio l'avessimo collegato al registro, il circuito si sarebbe trovato in ritardo di un ciclo di clock rispetto alle simulazioni. Lo stesso ragionamento è stato fatto anche per il Maggiore 15, posto sulla stessa linea.
  - d) Si è scelto di rappresentare il valore degli RPM su 13 bit in quanto il valore massimo (6500) in codifica binaria necessita di 13 bit, in base alla formula  $\lceil \log_2 6500 \rceil \approx 13$ . Il circuito in questo modo, non limita le rimanenti combinazioni (per ulteriori 1692 valori di rpm rappresentabili), quindi, lo si può facilmente adattare anche a motori leggermente diversi rispetto a quello considerato in questo progetto.
  - e) Il comparatore Uguale 13 (unito alla costante 0) non era strettamente necessario. Le specifiche a questo proposito si potevano interpretare in diversi modi ed è stato scelto infatti, di sfruttare la transizione a SPENTO non soltanto quando l'input INIT venisse impostato a

0, ma anche quando i valori di RPM fossero anch'essi 000...0, a dimostrazione del fatto che il motore è “spento” anche quando non produce giri motore.

f) Il sistema attraverso il quale viene impostato lo stato attuale è una serialità di 3 multiplexer posti uno sopra l'altro che commutano in accordo con quanto rilevato dai comparatori.