

# Relazione progetto SIS

Bottacini Luca

Lecini Rustem

Roin Giovanni

5/02/2022

## Indice

<b>Circuito FSM + D</b>	<b>3</b>
Traccia . . . . .	3
Interfaccia del circuito . . . . .	3
Architettura generale . . . . .	4
Segnali interni . . . . .	4
Macchina a stati finiti (FSM) . . . . .	5
Transizioni . . . . .	5
Grafo delle transizioni (STG) . . . . .	5
Unità di elaborazione (Data path) . . . . .	7
Contatore dei cicli . . . . .	7
Corpo principale . . . . .	8
Simulazioni di esempio . . . . .	10
Esempio 1 . . . . .	10
Esempio 2 . . . . .	10

## Circuito FSM + D

Abbiamo sviluppato un circuito che controlla un meccanismo chimico, il cui scopo è portare una soluzione con un pH iniziale noto ad un valore di neutralità.

### Traccia

Il valore del pH viene espresso in valori compresi tra 0,00 e 14,0: nell'intervallo  $[0,00, 7,00)$  si trovano i valori acidi, mentre in quello  $(8,00, 14,0]$  si trovano i valori basici, infine i valori inclusi in  $[7,00, 8,00]$  sono considerati neutrali. Tutti gli altri valori non sono accettabili e comportano un errore.

Il sistema è quindi dotato di due valvole: la prima può *decrementare* il valore del pH di 0.50 in un singolo ciclo di clock, mentre la seconda lo può *incrementare* di 0.25 nello stesso periodo di tempo.

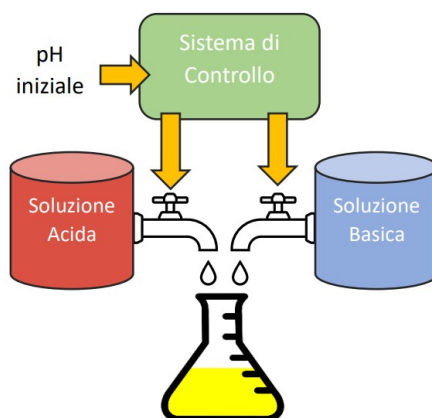


Figura 1: Illustrazione del circuito

### Interfaccia del circuito

Il circuito accetta i seguenti segnali di ingresso:

Nome	Descrizione
RST	Ordina al circuito di tornare allo stato iniziale.
START	Ordina al circuito di leggere il segnale PH_INIZIALE[8].
PH_INIZIALE[8]	Rappresenta il valore iniziale del pH.

L'ingresso PH\_INIZIALE[8] è un byte codificato in **virgola fissa** con 4 bit dedicati alla parte intera.

Il circuito produce i seguenti segnali di uscita:

Nome	Descrizione
FINE_OPERAZIONE	Indica che il sistema ha completato i calcoli.
ERRORE_SENSORE	Indica che il sistema ha ricevuto un pH invalido.
VALVOLA_ACIDO	Richiede il decremento del pH.
VALVOLA_BASICCO	Richiede l'incremento del pH.
PH_FINALE[8]	Rappresenta il valore finale del pH.
NCLK[8]	Rappresenta il numero di cicli impiegati.

L'uscita PH\_FINALE[8] è un byte codificato esattamente come l'ingresso PH\_INIZIALE[8], mentre il byte NCLK[8] viene codificato in **modulo**.

## Architettura generale

Il sistema implementa il modello **FSMD**, cioè collega una *macchina a stati finiti* (detta **FSM**) con un'*unità di elaborazione* (chiamata **Data path**).

Il compito della macchina a stati è quello di contestualizzare i calcoli eseguiti dall'unità di elaborazione, viceversa quest'ultima ha il ruolo di aiutare la macchina a determinare in che stato transitare.

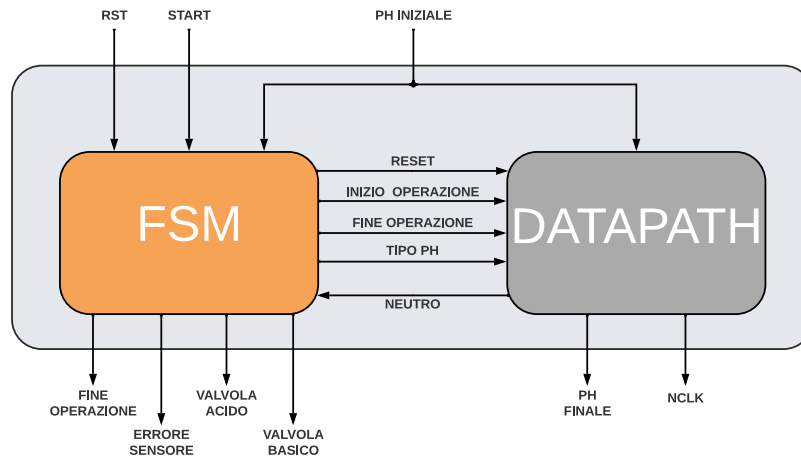


Figura 2: Diagramma del circuito

## Segnali interni

Il collegamento tra i due sottosistemi avviene grazie allo scambio di segnali di stato e controllo; i primi vengono emessi dalla macchina a stati verso l'elaboratore, i secondi seguono il percorso inverso.

I segnali di stato utilizzati sono i seguenti:

Nome	Descrizione
RESET	Ordina all'elaboratore di inizializzare i valori.
INIZIO_OPER.	Comunica all'elaboratore che è iniziata un'operazione.
FINE_OPER.	Comunica all'elaboratore che è finita l'operazione.
TIPO_PH	Permette all'elaboratore di modificare il pH correttamente.

I segnali di controllo utilizzati sono i seguenti:

Nome	Descrizione
NEUTRO	Comunica alla macchina che il pH ha raggiunto la neutralità.

## Macchina a stati finiti (FSM)

Abbiamo individuato cinque stati per questa macchina, cioè:

1. **Reset**: stato iniziale nel quale il circuito attende il pH in ingresso;
2. **Errore**: il valore del pH appena inserito non è valido;
3. **Acido**: il valore del pH attuale è inferiore a 7,00;
4. **Basico**: il valore del pH attuale è superiore a 8,00;
5. **Neutro**: il valore del pH ha raggiunto un valore incluso in  $[7,00, 8,00]$ .

## Transizioni

Lo stato iniziale della macchina è quello di *Reset*, da questo può spostarsi solamente quando riceve il segnale  $START = 1$ , in quel caso:

- quando il pH è superiore a 14,0 transita nello stato di *Errore*;
- quando il pH è minore stretto di 7,00 transita nello stato di *Acido*;
- quando il pH è maggiore stretto di 8,00 transita nello stato di *Basico*;
- quando il pH è già compreso nell'intervallo  $[7,00, 8,00]$  transita nello stato di *Neutro*.

Da ognuno degli stati può tornare a quello iniziale solo quando riceve il segnale  $RST = 1$ , altrimenti si sposta da *Acido* e *Basico* verso *Neutro* quando il segnale di controllo  $NEUTRO = 1$ .

Il segnale  $RST$  ha la precedenza su  $START$ , in altre parole: quando entrambi equivalgono ad 1, il secondo viene semplicemente ignorato.

## Grafo delle transizioni (STG)

Implementando il comportamento sopra descritto, abbiamo costruito il grafo delle transizioni utilizzando i seguenti segnali:

Segnali	D'ingresso	D'uscita
Esterni	RST	FINE_OPER.
	START	ERRORE_SENSORE
	PH_INIZIALE[8]	VALVOLA_ACIDO VALVOLA_BASICCO
Interni	NEUTRO	RESET
		INIZIO_OPER.
		TIPO_PH

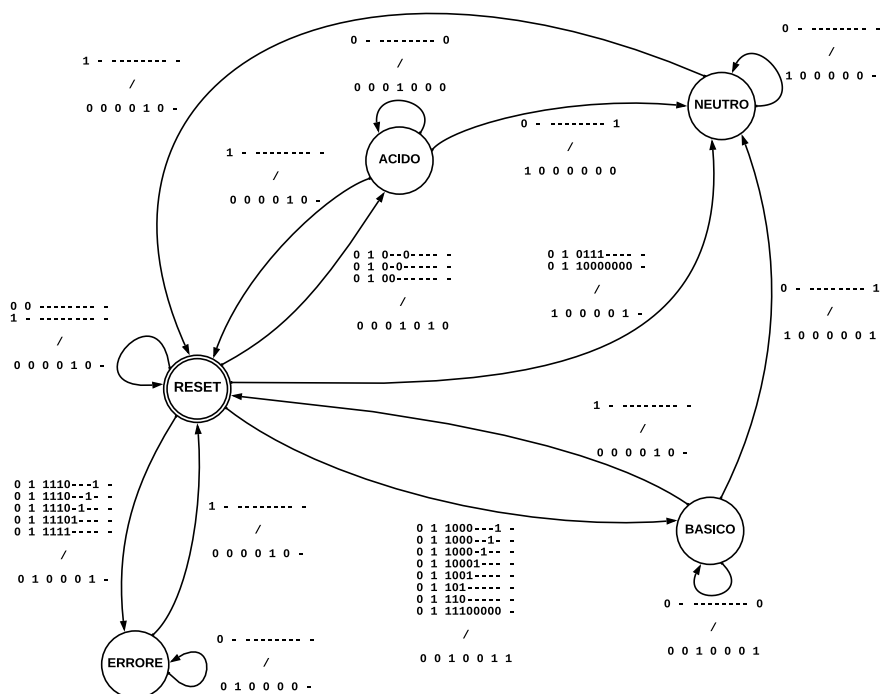


Figura 3: Macchina a stati

**Transizione di esempio** La transizione dallo stato *Reset* verso *Acido* avviene quando riceve:

- i segnali  $RST = 0$  e  $START = 1$ ;
- i segnale  $PH\_INIZIALE[8]$  interno a  $[0, 7)$ .

Il segnale *NEUTRO* viene ignorato perché l'unità di elaborazione non ha ancora memorizzato il pH: viene memorizzato solamente dopo la prima transizione verso uno stato diverso da *Reset*.

Nel codice sorgente tale transizione viene descritta come:

```
010--0----- Reset Acido 0001010
010-0----- Reset Acido 0001010
0100----- Reset Acido 0001010
```

## Unità di elaborazione (Data path)

Abbiamo suddiviso l'unità di elaborazione in più sottoproblemi risolti da delle parti specifiche:

1. *Contatore dei cicli*: memorizza ed incrementa il numero di cicli impiegati;
2. *Corpo principale*: si occupa della modifica del pH.

### Contatore dei cicli

Il contatore è composto da: un registro, tre multiplexer ed un sommatore ad 8 bit.

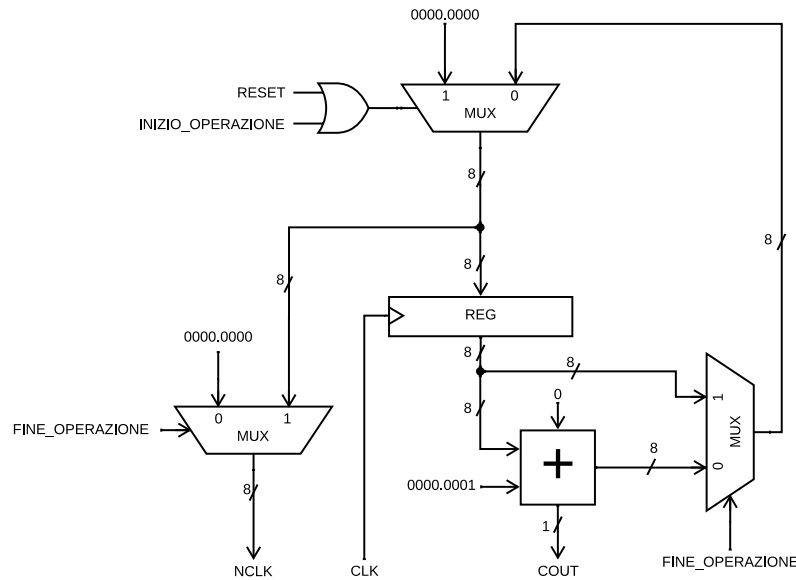


Figura 4: Contatore dei cicli

È il componente dedicato esclusivamente al calcolo dell'uscita `NCLK[8]`.

Ad ogni ciclo incrementa il valore memorizzato di un'unità. Quando riceve i segnali `RESET = 1` o `INIZIO_OPERAZIONE = 1` azzerà il conteggio, mentre quando il segnale `FINE_OPERAZIONE = 1` non incrementa il valore e lo mostra in uscita.

## Corpo principale

Il corpo principale è composto da: quattro multiplexer e due registri ad 8 bit.

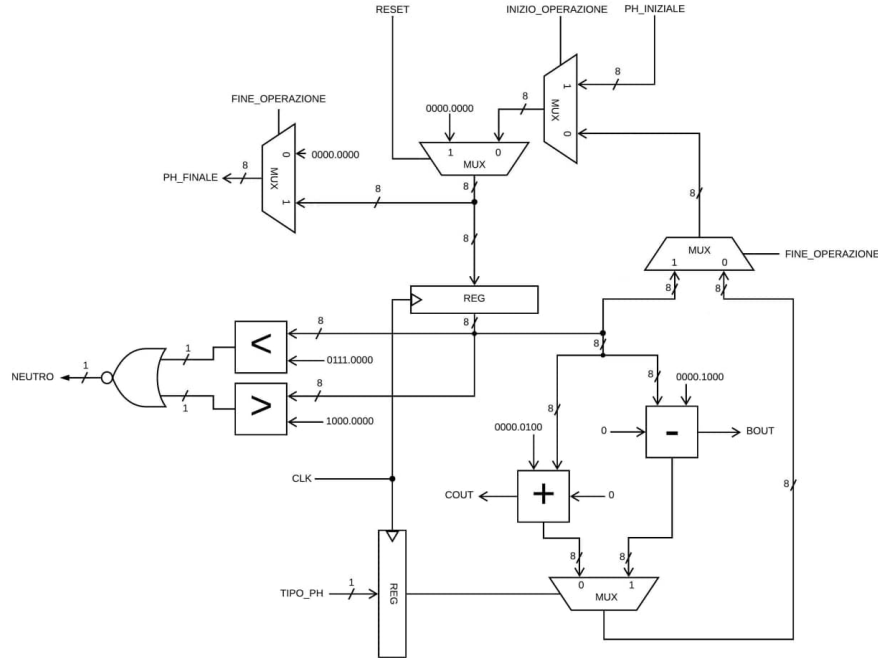


Figura 5: Unità principale

Quando il segnale  $RESET = 1$  il circuito inizializza il registro, invece quando equivale a 0 e  $INIZIO\_OPERAZIONE = 1$  l'elaboratore legge  $PH\_INIZIALE[8]$ .

Il segnale  $RESET$  ha la precedenza su  $INIZIO\_OPERAZIONE$ , in altre parole: quando entrambi equivalgono ad 1, il secondo viene semplicemente ignorato.

Il byte che giunge dai multiplexer viene memorizzato nel registro. Nel ciclo di clock successivo viene fornito il valore al *Verificatore di neutralità* ed al *Modificatore del pH*, infine:

- se il valore del segnale  $FINE\_OPERAZIONE$  equivale a 0 memorizza il valore modificato;
- altrimenti lo lascia intatto e mostra l'uscita  $PH\_FINALE[8]$ .



**Modifica del pH** Il modificatore è composto da: un sommatore, un sottrattore ed un multiplexer ad 8 bit.

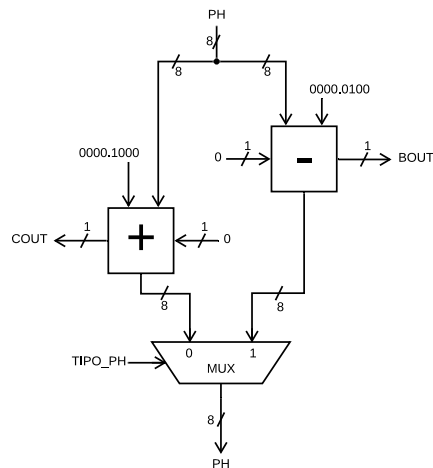


Figura 6: Modificatore del pH

Modifica il valore dell'ingresso PH[8] in funzione del segnale TIPO\_PH, cioè:

- quando TIPO\_PH equivale a 0 incrementa il pH di 0,25;
- quando TIPO\_PH equivale ad 1 decrementa il pH di 0,50.

**Verifica della neutralità** Il componente è composto da: un maggiore ed un minore ad 8 bit ed una porta NOR.

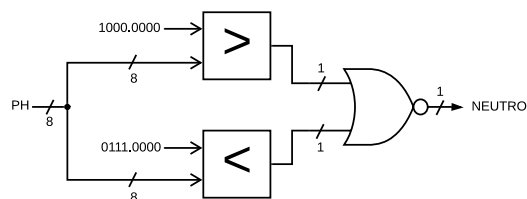


Figura 7: Verificatore di neutralità

Verifica il valore dell'ingresso PH\_INIZIALE[8], cioè:

- se questo è incluso in [7,00, 8,00] allora restituisce 1, cioè *vero*;
- altrimenti restituisce 0 cioè *falso*.

## Simulazioni di esempio

Ecco due esempi di esecuzione del circuito.

### Esempio 1

Inserendo un pH pari a 9,25 e specificando il segnale `START = 1` otteniamo come risultato semplicemente `VALVOLA_ACIDO = 1`.

```
sis> sim 0 1 1 0 0 1 0 1 0 0
```

Network simulation:

```
Outputs: 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

Non fornendo altri spunti al sistema per tre volte otteniamo sempre `VALVOLA_ACIDO = 1`, infatti:

$$9,25 - 0,50 = 8,75$$

$$8,75 - 0,50 = 8,25$$

$$8,25 - 0,50 = 7,75$$

```
sis> sim 0 0 0 0 0 0 0 0 0 0
```

Network simulation:

```
Outputs: 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

Infine dopo aver raggiunto un pH neutro, otteniamo `FINE_OPERAZIONE = 1`, un pH finale pari a 7,75 ed esattamente tre cicli impiegati per completare il calcolo.

```
sis> sim 0 0 0 0 0 0 0 0 0 0
```

Network simulation:

```
Outputs: 1 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 1 1
```

### Esempio 2

Inserendo un pH pari a 15,9375 e specificando il segnale `START = 1` otteniamo come risultato giustamente `ERRORE_SENSORE = 1`.

```
sis> sim 0 1 1 1 1 1 1 1 1 1
```

Network simulation:

```
Outputs: 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

Non fornendo altri spunti al sistema otteniamo sempre `ERRORE_SENSORE = 1`.

```
sis> sim 0 0 0 0 0 0 0 0 0 0
```

Network simulation:

```
Outputs: 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```