Relazione progetto SIS

Bottacini Luca Lecini Rustem Roin Giovanni

5/02/2022

Indice

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 `0.25` in un singolo ciclo di clock, mentre la seconda lo può *incrementare* di `0.50` nello stesso periodo di tempo.

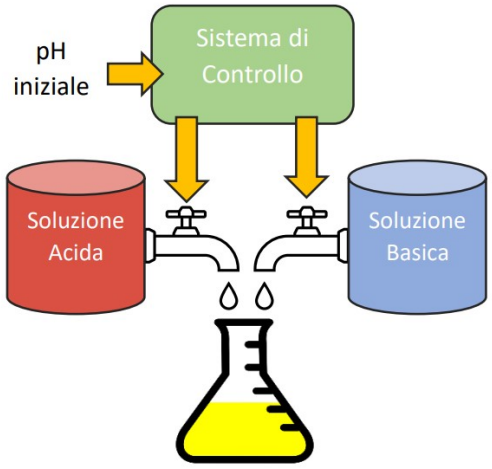

Figura 1: Illustrazione del circuito

Interfaccia del circuito

Il circuito accetta i seguenti segnali di ingresso:

Ingresso	Descrizione
`RST`	Ordina al circuito di tornare allo stato iniziale. Prevale su qualsiasi altro ingresso.
`START`	Ordina al circuito di leggere il valore presente nell'ingresso `PH[8]`.
`PH_INIZIALE[8]`	Rappresentazione del valore iniziale assunto dal pH della soluzione.

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:

Uscita	Descrizione
`FINE_OPER.`	Indica che il sistema ha completato le operazioni. Ovvero il pH è neutro.
`ERRORE_SENSORE`	Indica che il sistema ha ricevuto in ingresso un valore di pH non accettabile.
`VALVOLA_ACIDO`	Richiede l'apertura della valvola che decrementa il valore del pH.
`VALVOLA_BASICO`	Richiede l'apertura della valvola che incrementa il valore del pH.
`PH_FINALE[8]`	Rappresentazione del valore finale assunto dal pH della soluzione.
`NCLK[8]`	Rappresentazione del numero di cicli utilizzati per completare le operazioni.

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.

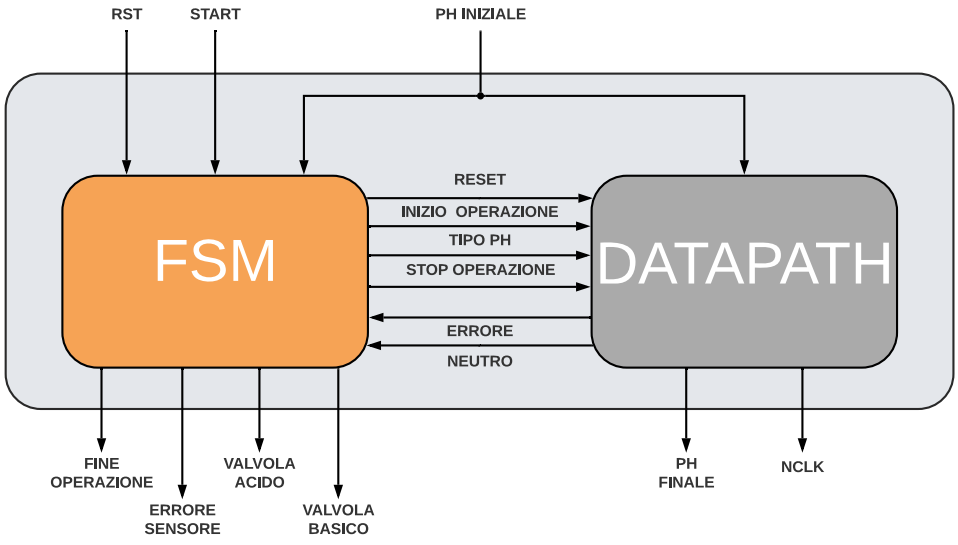

Figura 2: Diagramma del circuito

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:

Segnale	Descrizione
`RESET`	Ordina all'elaboratore di reinizializzare i valori.
`INIZIO_OPER.`	Comunica all'elaboratore che è appena stato inserito un pH.
`TIPO_PH`	Permette all'elaboratore di determinare come modificare il pH.
`STOP_OPER.`	Comunica all'elaboratore di non modificare i valori memorizzati.

I segnali di controllo utilizzati sono i seguenti:

Segnale	Descrizione
`ERRORE`	Comunica alla macchina che il valore del pH non è accettabile.
`NEUTRO`	Comunica alla macchina che il valore del 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 segnale di controllo `ERRORE` vale `1` transita nello stato di *Errore*;
- Quando il bit più significativo del segnale `PH[8]` vale `0` e non sono presenti errori, transita nello stato *Acido*;
- Quando il bit più significatico del segnale `PH[8]` vale `1` e non sono presenti errori, transita nello stato *Basico*.

La macchina si sposta nello stato *Neutro* quando il segnale di controllo `NEUTRO` vale `1`, infine, da ognuno degli stati può tornare a quello iniziale quando riceve il segnale `RST = 1`.

Segnali della macchina I segnali utilizzati dalla macchina a stati sono i seguenti in ordine di presentazione:

Segnali	D'ingresso	D'uscita
Esterni	RST	FINE_OPERAZIONE
	START	ERRORE_SENSORE
	PH_INIZIALE[8]	VALVOLA_ACIDO
		VALVOLA_BASICO
Interni	ERRORE	RESET
	NEUTRO	INIZIO_OPERAZIONE
		TIPO_PH
		STOP_OPERAZIONE

Grafo delle transizioni (STG)

Replicando il comportamento sopra descritto, abbiamo quindi costruito il seguente grafo delle transizioni:

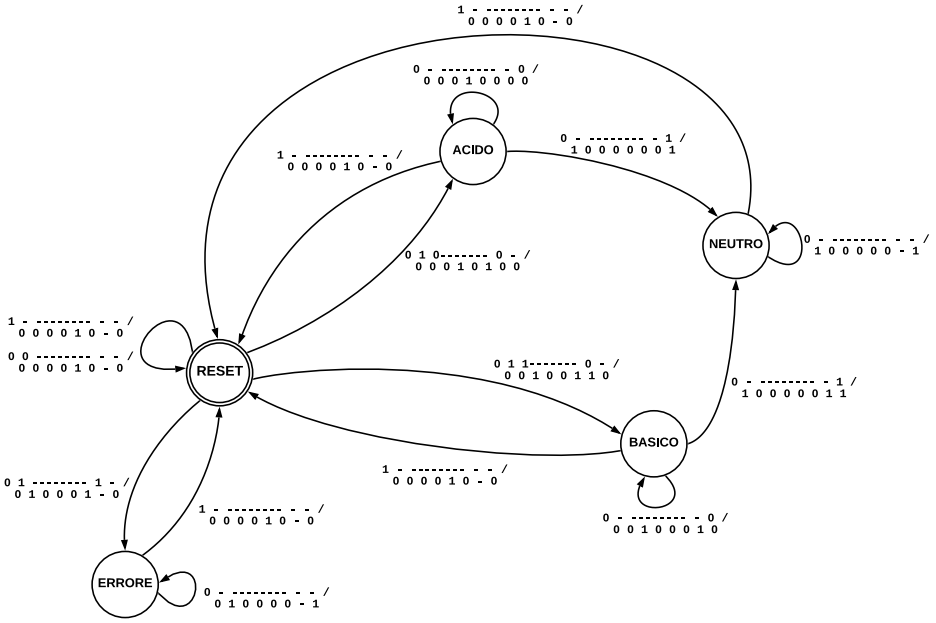

Figura 3: Macchina a stati

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

- il segnale `RST` equivale a `0`;
- il segnale `START` equivale ad `1`;
- il bit più significativo di `PH_INIZIALE[8]` vale `1`;
- il segnale `ERRORE` equivale a `0`.

Viene ignorato il segnale `NEUTRO` perché il circuito deve prima memorizzare il valore e solo in un ciclo successivo è in grado di rilevare la sua eventuale neutralità; infatti la macchina a stati non è in grado di raggiungere lo stato *Neutro* senza prima transitare per *Acido* o *Basico*.

Nel codice sorgente tale transizione viene descritta come:

```
011-------0- Reset Basico 00100110
```

Unità di elaborazione (Data path)

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

1. *Contatore dei cicli*: memorizza ed incrementa il numero di cicli impiegati;
2. *Modificatore del pH*: aggiorna il valore del pH;
3. *Verificatore di neutralità*: determina se il valore del pH è interno a `[7,00, 8,00]`;
4. *Verificatore di errore*: determina se il valore del pH è superiore a `14,0`.

Conteggio dei cicli

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

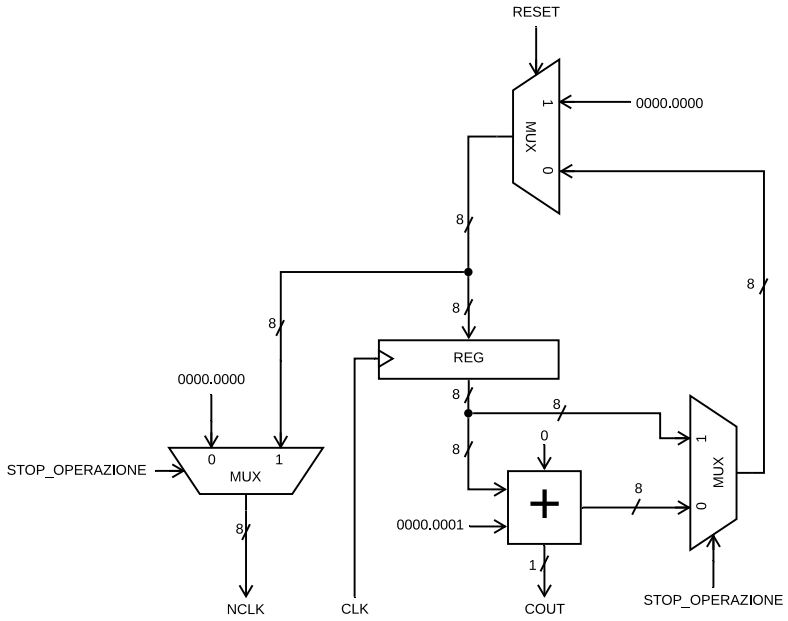

Figura 4: Contatore dei cicli

È un componente dedicato esclusivamente al calcolo dell'uscita `NCLK[8]`, mentre gli altri collaborano tra loro sia per determinare i segnali di controllo, che soprattutto per calcolare l'uscita `PH_FINALE[8]`.

Incrementa di `1` il valore memorizzato nel registro ad ogni ciclo ad eccezione dei casi in cui riceve il segnale `STOP = 1`. Invece, quando l'ingresso `RESET` equivale ad `1`, indipendentemente dal valore dell'altro, azzera il valore memorizzato nel registro.

Modifica del pH

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

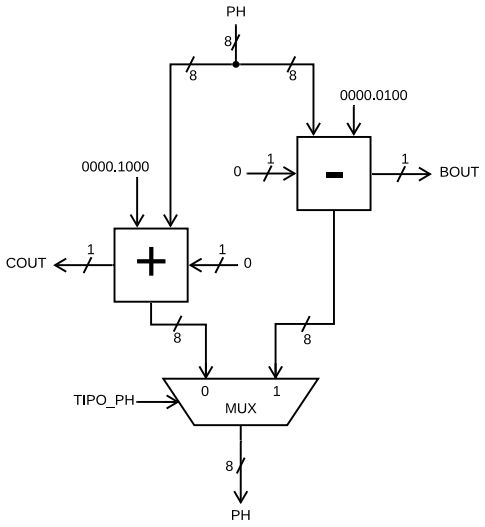

Figura 5: Modificatore del pH

Modifica il valore dell'ingresso `PH[8]` in funzione del segnale `TIPO_PH`, cioé:

- nel caso in cui `TIPO_PH` equivale a `0` incrementa il pH di `0,50`;
- nel caso in cui `TIPO_PH` equivale ad `1` decrementa il pH di `0,25`.

Verifica della neutralità

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

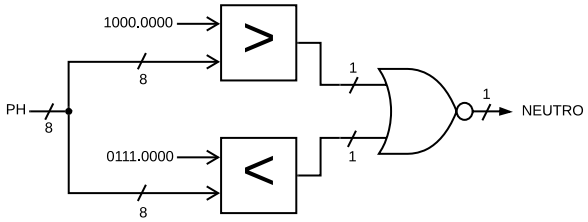

Figura 6: Verificatore di neutralità

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*.

Verifica degli errori

Il componente è composto da un maggiore ad 8 bit.

Verifica il valore dell'ingresso `PH_INIZIALE[8]`, cioè:

- se questo è superiore a `14,0`, allora restituisce `1`, cioè *vero*;
- altrimenti restituisce `0` cioè *falso*.

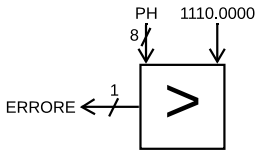

Figura 7: Verificatore di errore

Unità completa

È composta da: due registri, quattro multiplexer ad 8 bit, un modificatore, un verificatore di errore ed uno di neutralità.

Coordina gli altri componenti tramite registri e multiplexer aggiuntivi.

Corpo principale Quando sia il segnale `INIZIO_OPERAZIONE` che `RESET` valgono `0` il circuito continua ad elaborare il valore inserito precedentemente, al contrario:

- se `RESET` equivale ad `1`, indipendentemente dagli altri ingressi, il circuito restituisce un byte azzerato;
- oppure, se `INIZIO_OPERAZIONE` vale `1` restituisce il segnale `PH_INIZIALE[8]`;

Questo valore viene quindi analizzato tramite il *Verificatore di errore* per determinare se la codifica è accettabile, viene restituito se il segnale `STOP_OPERAZIONE` equivale ad `1` e viene finalmente memorizzato nel registro.

Nel ciclo di clock successivo il circuito utilizza il *Modificatore del pH* per aggiornare il valore, e ancora:

- quando il valore del segnale di stato `STOP_OPERAZIONE` equivale a `0` restituisce il valore modificato;
- altrimenti se equivale ad `1` restituisce il valore memorizzato.

Infine usufruisce del *Verificatore di neutralità* per determinare se il valore è neutro ed indirizza il nuovo risultato all'interno dei multiplexer iniziali.

Il segnale di uscita `PH_FINALE[8]` non viene restituito finché il segnale `STOP_OPERAZIONE` non quivale ad `1`: in quel caso restituisce il valore prodotto dai multiplexer iniziali.

Contatore Il circuito utilizza il *Contatore dei cicli* per riuscire a determinare quante operazioni ha impiegato per raggiungere il risultato. Il segnale di uscita `NCLK[8]` non viene restituito finché il segnale `STOP_OPERAZIONE` non diviene uguale ad `1`.

Segnali dell'unità I segnali utilizzati dall'unità a stati sono i seguenti in ordine di presentazione:

Segnali	D'ingresso	D'uscita
Esterni	`PH_INIZIALE[8]`	`PH_FINALE[8]`
		`NCLK[8]`
Interni	`RESET`	`ERRORE`
	`INIZIO_OPERAZIONE`	`NEUTRO`
	`TIPO_PH`	
	`STOP_OPERAZIONE`	

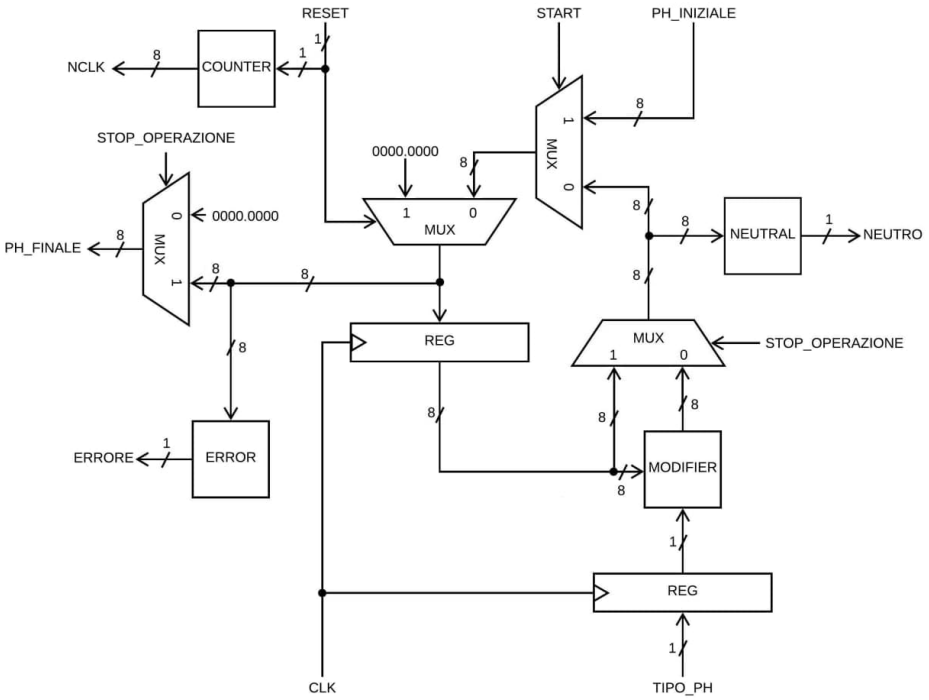

Figura 8: Unità di elaborazione

Il registro ad 1 bit che memorizza il segnale `TIPO_PH` è presente per evitare di creare un ciclo all'interno del circuito e renderlo non deterministico.

Sostituendo il contenuto dei componenti all'interno del corpo principale otteniamo il seguente circuito:

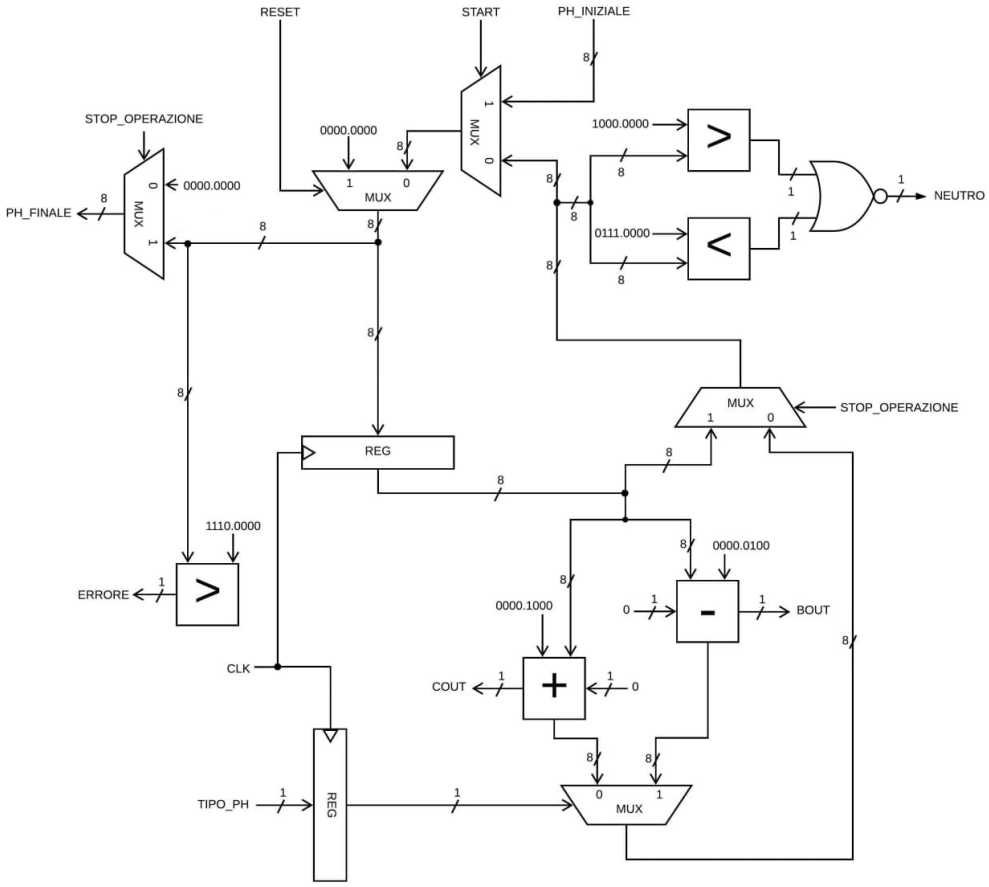

Figura 9: Unità di elaborazione

Il *Contatore dei cicli* invece è un componente autonomo e abbiamo scelto di non includerlo in questa immagine per problemi di spazio, è comunque presente all'interno del circuito.

Simulazioni di esempio

Dopo aver progettato i due sottosistemi abbiamo provato alcuni flussi di esecuzione: il primo vede come ingresso un pH pari a `5,75` che quindi impiega 4 cicli per completare l'operazione con un pH finale di `5,75`; nel secondo invece abbiamo tentato di inserire un pH non valido e dopo aver segnalato l'errore non ha elaborato oltre.

```
# Inserendo RST = 0, START = 1, PH = 5,75.
sis> simulate 0 1 0 1 0 1 1 1 0 0

# Otteniamo VALVOLA_BASICO = 1.
Network simulation:
Outputs: 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
Next state: 00101110000000001010

# Inserendo tutti i valori a 0.
sis> simulate 0 0 0 0 0 0 0 0 0 0

# Prosegue con l'elaborazione.
Network simulation:
Outputs: 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
Next state: 00110010000000010010

# Inserendo tutti i valori a 0.
sis> simulate 0 0 0 0 0 0 0 0 0 0

# Prosegue con l'elaborazione.
Network simulation:
Outputs: 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
Next state: 00110110000000011010

# Inserendo tutti i valori a 0.
sis> simulate 0 0 0 0 0 0 0 0 0 0

# Prosegue con l'elaborazione.
Network simulation:
Outputs: 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
Next state: 00111010000000100010

# Inserendo tutti i valori a 0.
sis> simulate 0 0 0 0 0 0 0 0 0 0

# Otteniamo FINE_OPERAZIONE = 1, PH = 7,25, NCLK = 4.
Network simulation:
Outputs: 1 0 0 0 0 1 1 1 0 1 0 0 0 0 0 0 0 1 0 0
Next state: 00111010000000100001
```