

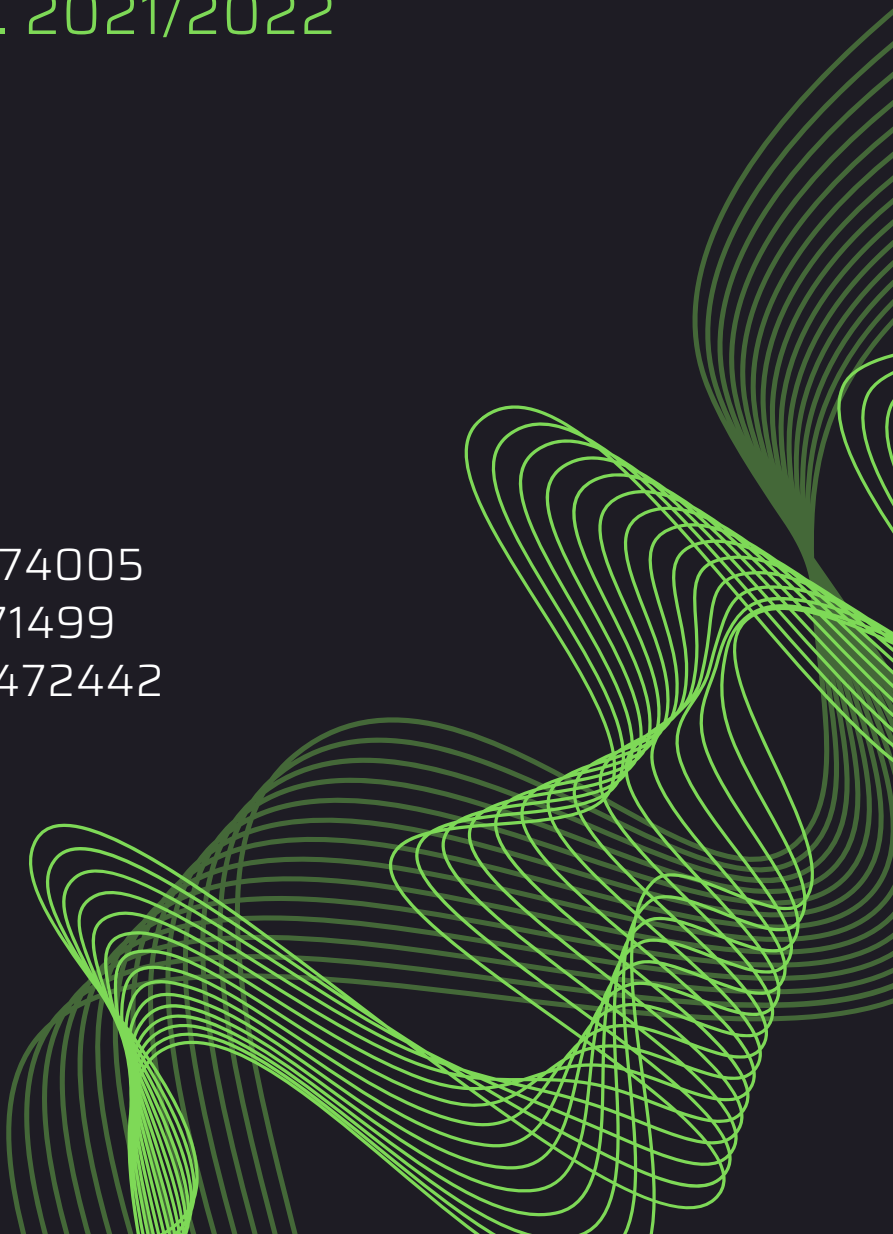


UNIVERSITÀ DEGLI
STUDI DI VERONA

ARCHITETTURA DEGLI ELABORATORI

Elaborato SIS - A.A. 2021/2022

BOTTEGA MARICA - VR474005
VINCENZI BRUNO - VR471499
ZAMUNER SERENA - VR472442



Indice generale

Testo dell'elaborato.....	2
Architettura generale del circuito.....	3
Diagramma degli stati del controllore.....	6
Datapath.....	7
Statistiche del circuito.....	10
Mapping.....	13
Scelte progettuali.....	14

Testo dell'elaborato

Si progetti il circuito sequenziale che controlla un macchinario chimico il cui scopo è portare una soluzione iniziale a pH noto, ad un pH di neutralità. Il valore del pH viene espresso in valori compresi tra 0 e 14. Il circuito controlla due valvole di erogazione: una di soluzione acida e una di soluzione basica. Se la soluzione iniziale è acida, il circuito dovrà procedere all'erogazione della soluzione basica fintanto che la soluzione finale non raggiunga la soglia di neutralità (pH compreso tra 7 e 8). Analogamente, se la soluzione iniziale è basica, il circuito procederà all'erogazione di soluzione acida fino al raggiungimento della soglia di neutralità. Per pH acido si intende un valore strettamente inferiore a 7, mentre per basico si intende una soluzione con pH strettamente maggiore a 8. Il pH viene codificato in fixed-point, con 4 bit riservati per la parte intera e gli altri per la parte decimale. Le due valvole hanno flussi differenti di erogazione. La valvola relativa alla soluzione basica eroga una quantità di soluzione che permette di alzare il pH della iniziale di 0.25 ogni ciclo di clock. La valvola relativa alla soluzione acida eroga una quantità di soluzione che permette di abbassare il pH della soluzione iniziale di 0.5 ogni ciclo di clock.

Il circuito ha 3 ingressi nel seguente ordine:

- RST (1 bit)
- START(1 bit)
- pH (8 bit, 4 parte intera e 4 per la parte decimale)

Gli output sono i seguenti e devono seguire il seguente ordine:

- FINE_OPERAZIONE (1 bit)
- ERRORE_SENSORE (1 bit)
- VALVOLA_ACIDO (1 bit)
- VALVOLA_BASICO (1 bit)
- PH_FINALE (8 bit)
- NCLK (8 bit)

Input e output devono essere definiti nell'ordine sopra specificato (da sinistra verso destra). Le porte con più bit devono essere descritte utilizzando la codifica con il bit più significativo a sinistra.

Architettura generale del circuito

Come volutasi dalle richieste del testo dell'elaborato, la nostra macchina analizza una soluzione liquida e in base alla sua acidità (avente un pH maggiore di zero e strettamente minore di sette) o la sua basicità (avente un pH strettamente maggiore di otto e minore di quattordici) apre una valvola che eroga acidità (aggiunge alla soluzione +0.25 di acidità) o la valvola che eroga basicità (aggiunge alla soluzione +0.50 di basicità). Il nostro dispositivo è composto da un'unità di controllo (FSM) e un'unità di elaborazione (Datapath), l'interazione tra queste due componenti vanno a formare un modello FSMD.

La FSMD riceve in ingresso dall'utente i seguenti input che convergono sia nella FSM che nel Datapath:

- pH[8]: input a otto(8) bit, di cui quattro(4) per la parte intera e quattro(4) per la parte decimale, indicano il pH, che può essere acido, basico, neutro oppure un pH non valido.
- START[1]: input da un(1) bit che se impostato a uno(1) avvia la macchina.
- RST[1]: input da un(1) bit che resetta le operazioni della macchina se impostato a uno(1).

Mentre in uscita dalla FSMD sono presenti i seguenti output:

Output provenienti dall'unità di controllo (FSM):

- FINE_OPERAZIONE[1]: un(1) bit che se assume valore uno(1), indica che la macchina ha terminato il processo di neutralizzazione del pH inserito in input.
- ERRORE_SENSORE[1]: un(1) bit che se assume valore uno(1) indica che c'è stato un errore, più specificamente l'errore fa riferimento ad un pH non valido, cioè maggiore di quattordici(14).
- VALVOLA_ACIDO[1]: un(1) bit che indica l'apertura di una valvola che eroga soluzione acida che diminuisce il pH di -0.50 ad ogni giro di clock se il bit assume valore uno(1).
- VALVOLA_BASICCO[1]: un(1) bit che indica l'apertura di una valvola che eroga soluzione basica che aumenta il pH di +0.25 ad ogni giro di clock se il bit assume valore uno(1).

Output provenienti dall'unità di calcolo (Datapath):

- PH_FINALE[8]: otto(8) bit che indicano il valore del pH neutralizzato alla fine dell'elaborazione del dispositivo.
- NCLOCK[8]: otto(8) bit che indicano il numero di giri di clock effettuati dalla macchina per rendere neutra la soluzione inserita.

L'unità di controllo e l'unità di elaborazione comunicano tra loro attraverso i seguenti segnali:

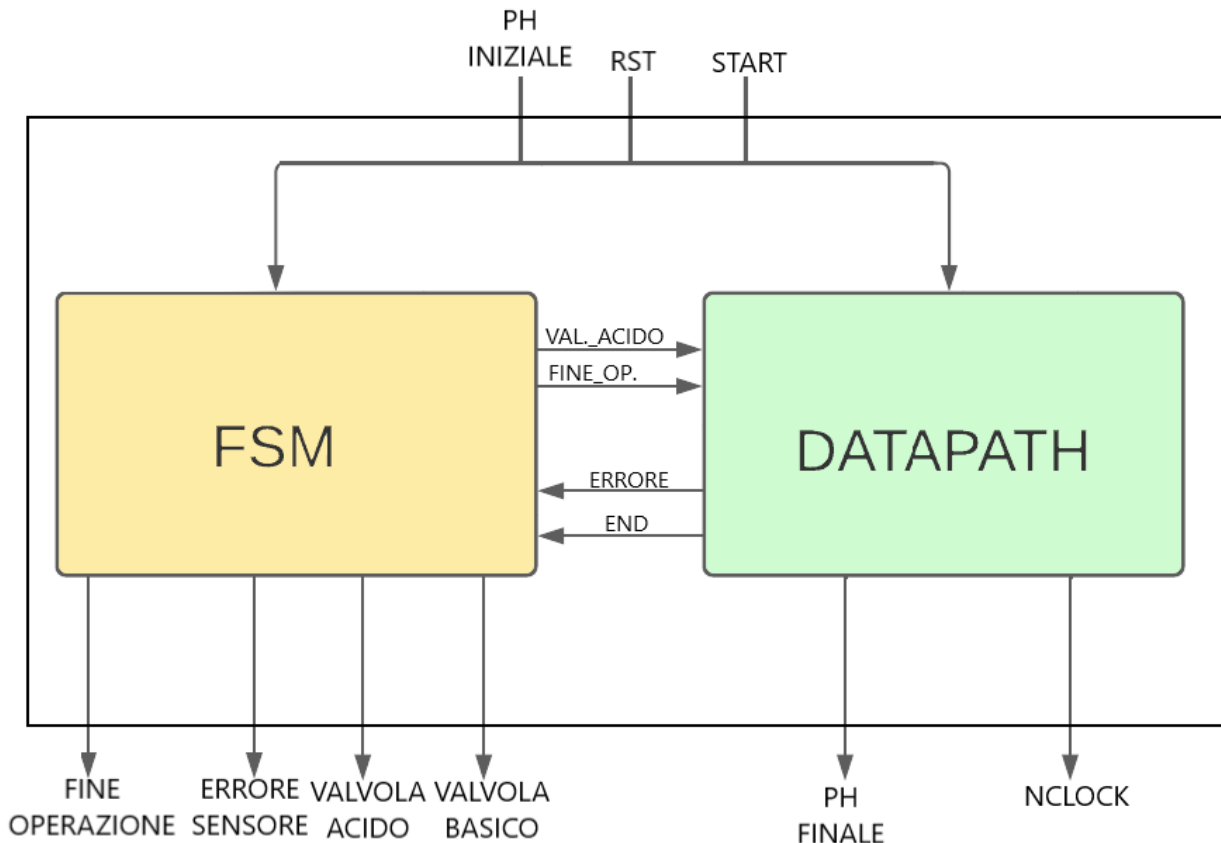
Segnali di controllo (da FSM a Datapath):

- VALVOLA_ACIDO[1]: un(1) bit che se ha valore uno(1), indica che la valvola che eroga soluzione acida si è aperta, altrimenti, se assume valore zero(0) indica l'apertura della valvola che eroga soluzione basica oppure che entrambe le valvole sono chiuse.
- FINE_OPERAZIONE[1]: un(1) bit che se assume valore uno(1) indica che la macchina ha terminato il processo di neutralizzazione del pH inserito in input.

Segnali di stato (da Datapath a FSM):

- ERRORE[1]: un(1) bit che se assume valore uno(1) indica che c'è stato un errore, più specificamente l'errore fa riferimento ad un pH non valido, cioè maggiore di quattordici(14).
- END[1]: un(1) bit che se assume valore uno(1) indica che la macchina ha terminato il processo di neutralizzazione del pH inserito in input oppure il pH inserito era già neutro.

In seguito riportiamo il disegno della FSMD:



Controllore

Il controllore è una macchina a stati finiti, (FSM) di Mealy con dodici(12) bit d'ingresso (RESET[1], START[1], pH[8], END[1], ERRORE[1]) di cui i primi dieci inseriti dall'utente e gli ultimi due, l'input END[1] e ERRORE[1], provenienti dal datapath, inoltre ha quattro(4) bit d'uscita (FINE_OPERAZIONE[1], ERRORE_SENSORE[1], VALVOLA_ACIDO[1], VALVOLA_BASIC0[1]).

Sono stati presi in considerazione quattro stati:

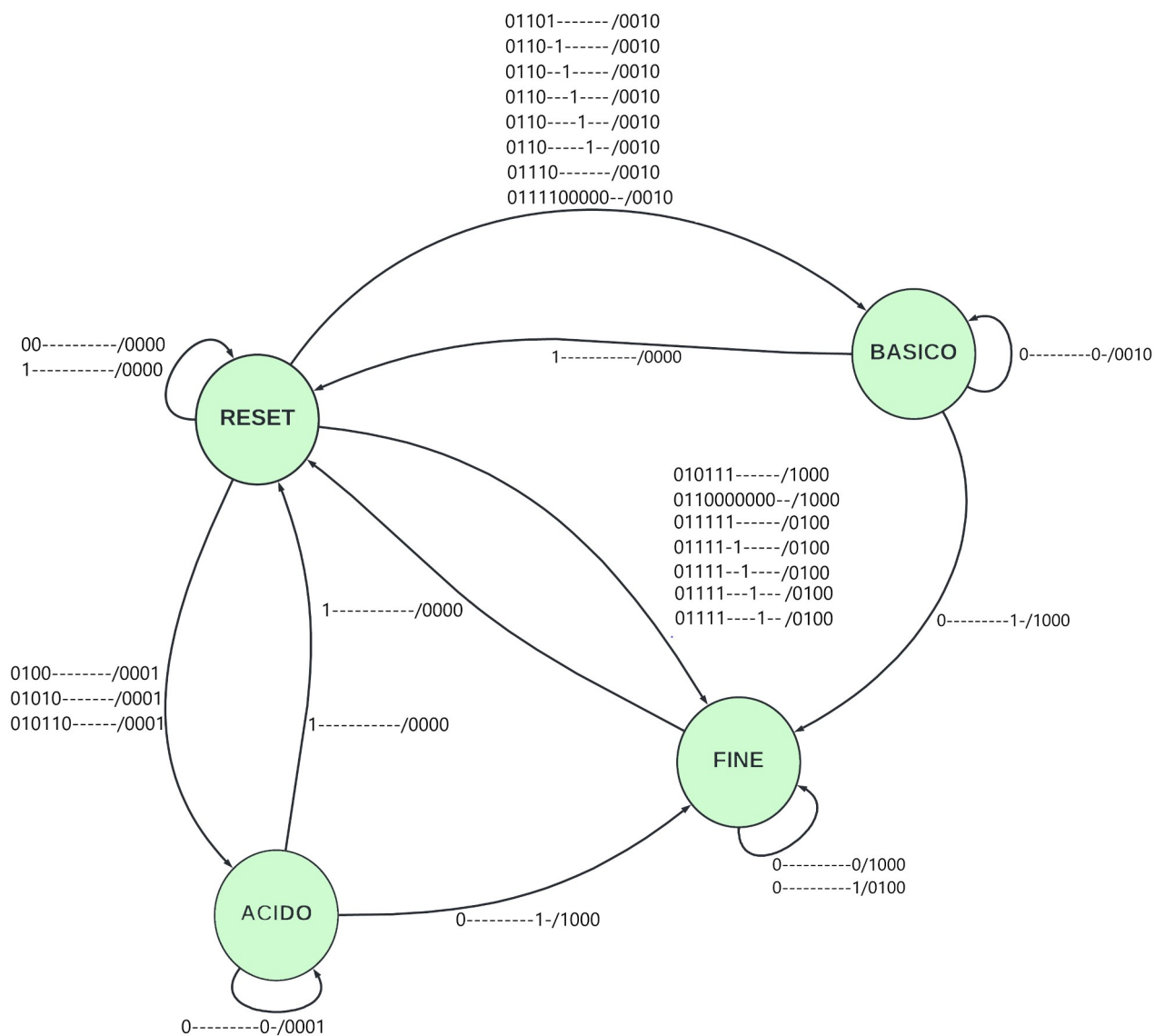
- **RESET:** Il RESET è lo stato iniziale della macchina, quando l'operazione non è ancora iniziata. Finché l'input di START[1] non ha valore uno(1) la FSM rimane in questo stato, mentre se l'input di RESET[1] è uno(1) in qualsiasi altro stato, compreso lo stato di RESET stesso, la FSM torna nello stato di RESET.
- **ACIDO:** La macchina assume lo stato di ACIDO se pH[8] è un valore maggiore uguale di zero(0) e strettamente minore di sette(7). Rimane in questo stato finché il segnale di END[1] proveniente dal datapath è zero(0), mentre passa allo stato di FINE se il segnale è uno(1).
- **BASICO:** La macchina assume lo stato di BASICO se pH[8] è un valore strettamente maggiore di otto(8) e minore uguale di quattordici(14). In modo equivalente allo stato di ACIDO, rimane in questo stato finché il segnale di END[1] proveniente dal datapath è zero(0), mentre passa allo stato di FINE se il segnale è uno(1).
- **FINE:** Lo stato di FINE è lo stato finale della macchina. Quest'ultima assume questo stato in tre occasioni:
 1. Se dallo stato di RESET viene inserito un valore di pH[8] non valido, cioè strettamente maggiore di quattordici, l'output di ERRORE[1] assumerà valore uno(1), mentre gli altri bit di output assumeranno valore zero(0);
 2. Se nello stato di RESET viene inserito un valore di pH[8] già neutro (maggiore uguale a sette(7) e minore uguale a otto(8)), il bit di output FINE_OPERAZIONE[1] assume il valore uno(1) concludendo l'operazione.
 3. Se nello stato di ACIDO o BASICO il segnale di END[1] proveniente dal datapath ha valore uno(1), l'output di FINE_OPERAZIONE[1] assume il valore uno(1).

In seguito viene riportata la STG (State Transition Graph)

Diagramma degli stati del controllore

Input: {RESET[1], START[1], pH[8], END[1]}

Output: {FINE_OPERAZIONE[1], ERRORE_SENSORE[1],
VALVOLA_ACIDO[1], VALVOLA_BASICICO[1]}



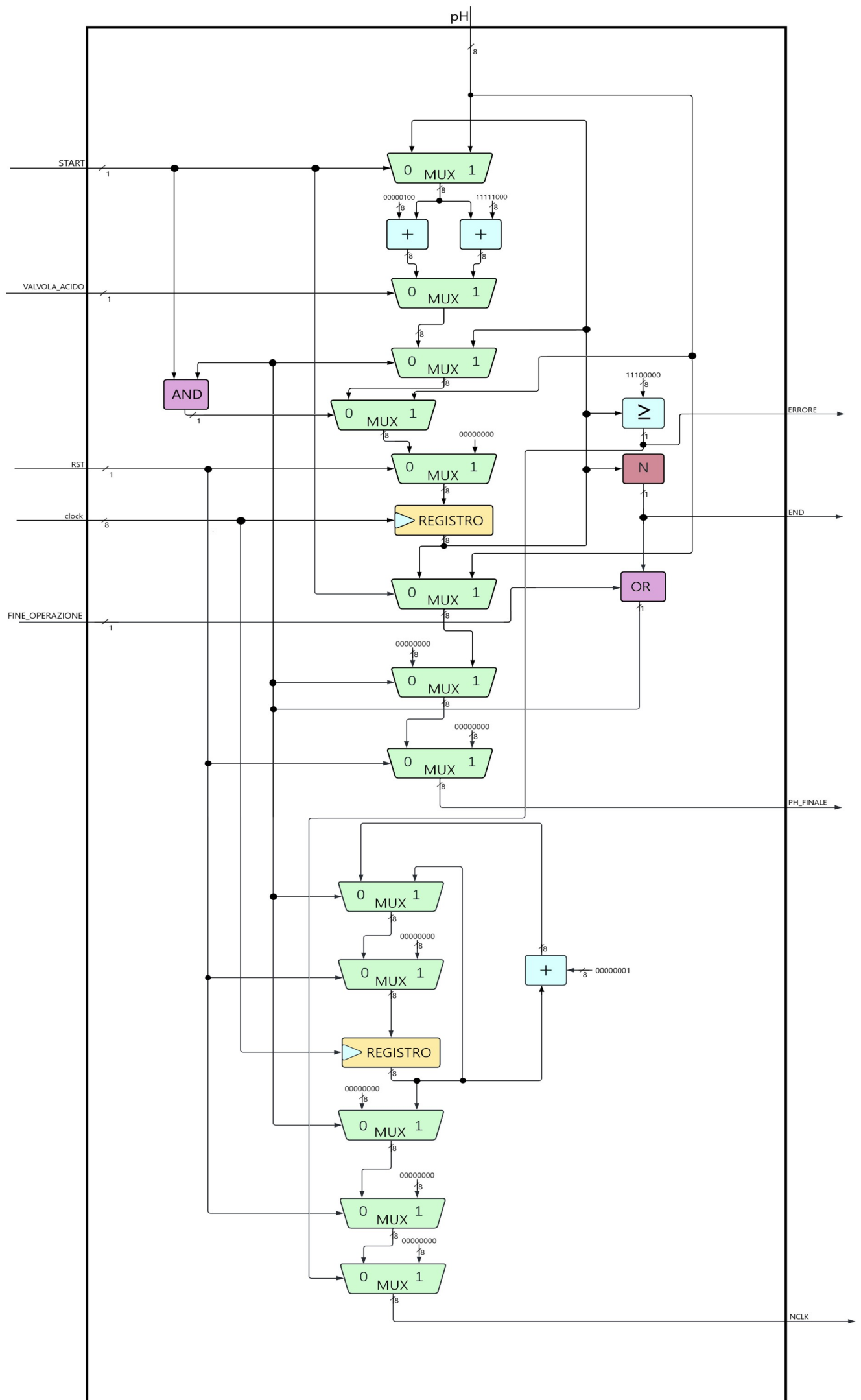
Datapath

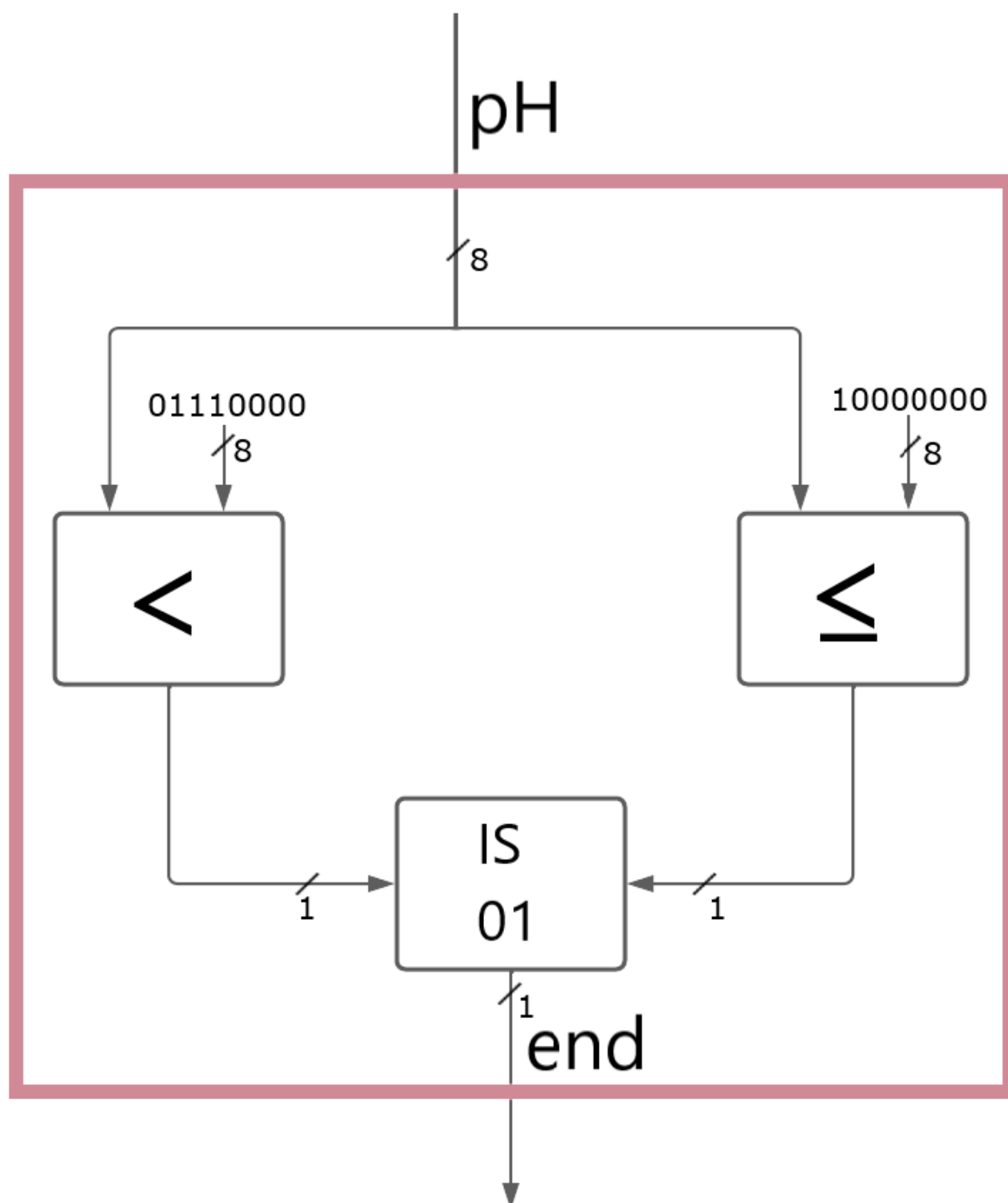
Il datapath, un'insieme di unità di calcolo, che abbiamo strutturato, riceve in input dall'utente il pH iniziale (pH[8]), il segnale di inizio (START[1]) e di riavvio della macchina (RESET[1]), mentre riceve dalla FSM il segnale di apertura o chiusura della valvola di acido (VALVOLA_ACIDO[1]) e il segnale di termine dell'operazione (FINE_OPERAZIONE[1]). In output, il datapath, passa alla FSM il segnale di errore (ERRORE[1]) se il pH è maggiore di quattordici e il segnale di termine neutralizzazione del pH (END[1]), invece dalla FSMD esce il pH neutralizzato (PH_FINALE[8]) e il numero di giri impiegati dal datapath per rendere il pH neutro (NCLOCK[1]).

Il datapath si compone delle seguenti componenti:

- **MULTIPLEXER:** tredici(13) selettori a otto(8) bit con 2 entrate che servono a decidere che tipo di dati devono memorizzare i registri o cosa la FSMD deve dare in output.
- **SOMMATORE:** tre(3) unità aritmetiche. Il primo serve ad aggiungere +0.25 al pH acido, il secondo somma -0.50 al pH basico, mentre il terzo serve per contare il ciclo di clock.
- **REGISTRO:** due(2) registri a otto(8) bit, il primo memorizza il pH mentre il secondo il numero di clock effettuati dal datapath per neutralizzare il pH.
- **MAGGIORE:** Un'(1) unità di controllo che verifica se il pH è maggiore di quattordici.
- **NEUTRO:** un(1) costrutto formato da tre(3) unità di controllo:
 1. **MINORE:** un(1) minore a otto(8) bit che serve a controllare se il pH è minore di sette.
 2. **MINORE UGUALE:** un(1) minore uguale a otto(8) bit per verificare che il pH sia minore o uguale a otto.
 3. **IS 01:** un'(1) costrutto di uguaglianza che vale uno(1) quando MINORE ha valore zero(0) e MINORE UGUALE ha valore uno(1).
- **OR, AND:** due(2) unità logiche da usare come input di scelta nei mux, un OR a uno(1) bit che vale uno(1) quando la fine presa dalla FSM o END[1] calcolato dal datapath valgono uno(1), e un AND a uno(1) bit che vale uno(1) se sia la macchina ha iniziato (preso dalla FSM) e ha finito (pH neutro, calcolato dal datapath)

in seguito viene riportata l'architettura del Datapath.





Statistiche del circuito

Come prima operazione abbiamo provato a minimizzare gli stati della FSM con il comando "state_minimize stamina" senza ottenere alcun cambiamento:

```
sis> rl FSM.blif
sis> print_stats
FSM          pi=12  po= 4  nodes= 6      latches= 2
lits(sop)= 386  #states(STG)= 4
```

In seguito con "script.rugged" le statistiche che si presentano sono le seguenti:

```
sis> source script.rugged
sis> print_stats
FSM          pi=12  po= 4  nodes= 13      latches= 2
lits(sop)= 60  #states(STG)= 4
sis> □
```

Mentre le statistiche del Datapath non ottimizzato sono:

```
sis> rl DATAPATH_NON_OTTIMIZZATO.blif
Warning: network `less_7', node "a4" does not fanout
Warning: network `less_7', node "a5" does not fanout
Warning: network `less_7', node "a6" does not fanout
Warning: network `less_7', node "a7" does not fanout
Warning: network `piu025', node "carry8" does not fanout
Warning: network `piuuno8', node "carry" does not fanout
Warning: network `meno050', node "carry8" does not fanout
Warning: network `DATAPATH', node "aclk3" does not fanout
Warning: network `DATAPATH', node "dclk3" does not fanout
Warning: network `DATAPATH', node "eclk3" does not fanout
Warning: network `DATAPATH', node "carry8" does not fanout
Warning: network `DATAPATH', node "carry" does not fanout
Warning: network `DATAPATH', node "[97]" does not fanout
sis> print_stats
DATAPATH     pi=12  po=18  nodes=164      latches=16
lits(sop)=1274
```

con “script.rugged” eseguito cinque volte le statistiche del Datapath migliorano passando da 1274 letterali a 242 letterali, da 164 nodi a 52:

```
sis> source script.rugged
sis> print_stats
DATAPATH      pi=12   po=18   nodes= 58      latches=15
lits(sop)= 262
sis> source script.rugged
sis> print_stats
DATAPATH      pi=12   po=18   nodes= 59      latches=15
lits(sop)= 258
sis> source script.rugged
sis> print_stats
DATAPATH      pi=12   po=18   nodes= 53      latches=15
lits(sop)= 242
sis> source script.rugged
sis> print_stats
DATAPATH      pi=12   po=18   nodes= 52      latches=15
lits(sop)= 242
sis> source script.rugged
sis> print_stats
DATAPATH      pi=12   po=18   nodes= 52      latches=15
lits(sop)= 242
sis> 
```

In seguito abbiamo creato la FSMD prendendo la FSM e il Datapath già ottimizzati per poi ottimizzare ulteriormente la FSMD ottenuta, abbiamo però ottenuto statistiche peggiori in confronto ad ottimizzare la FSMD con componenti non ancora ottimizzate.

La FSMD presenta le seguenti statistiche prima di qualsiasi ottimizzazione:

```
sis> print_stats
FSMD          pi=10   po=20   nodes=170      latches=18
lits(sop)=1660
```

Dopo aver lanciato il comando “script_rugged” per cinque volte le statistiche della FSM D il numero di letterali è passato da 1660 a 291, mentre il numero dei nodi da 170 a 65:

```
sis> source script.rugged
sis> print_stats
FSMD          pi=10   po=20   nodes= 72      latches=17
lits(sop)= 329
sis> source script.rugged
sis> print_stats
FSMD          pi=10   po=20   nodes= 68      latches=17
lits(sop)= 316
sis> source script.rugged
sis> print_stats
FSMD          pi=10   po=20   nodes= 65      latches=17
lits(sop)= 295
sis> source script.rugged
sis> print_stats
FSMD          pi=10   po=20   nodes= 65      latches=17
lits(sop)= 291
sis> source script.rugged
sis> print_stats
FSMD          pi=10   po=20   nodes= 65      latches=17
lits(sop)= 291
sis> 
```

Mapping

Dopo aver ottimizzato il circuito al meglio, abbiamo eseguito il mapping con la libreria "synch.genlib", eseguendo la mappatura per area come richiesto dal testo dell'elaborato:

```
sis> rl FSMD.blif
sis> read_library synch.genlib
sis> map -m 0
warning: unknown latch type at node '{[4]}' (RISING_EDGE assumed)
warning: unknown latch type at node '{[5]}' (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
sis> map -s
>>> before removing serial inverters <<<
# of outputs:      37
total gate area:    6008.00
maximum arrival time: (36.80,36.80)
maximum po slack:   (-9.40,-9.40)
minimum po slack:   (-36.80,-36.80)
total neg slack:    (-871.00,-871.00)
# of failing outputs: 37
>>> before removing parallel inverters <<<
# of outputs:      37
total gate area:    5992.00
maximum arrival time: (36.80,36.80)
maximum po slack:   (-9.40,-9.40)
minimum po slack:   (-36.80,-36.80)
total neg slack:    (-871.00,-871.00)
# of failing outputs: 37
# of outputs:      37
total gate area:    5928.00
maximum arrival time: (36.80,36.80)
maximum po slack:   (-9.40,-9.40)
minimum po slack:   (-36.80,-36.80)
total neg slack:    (-871.00,-871.00)
# of failing outputs: 37
sis> □
```

Il total gate area è pari a 6008.00 mentre il cammino critico (ritardo) è 36.80.

Scelte progettuali

Abbiamo deciso di estendere il più possibile il Datapath in modo che SIS ottimizzasse al meglio, attraverso gli opportuni comandi, la macchina.

La FSM e il Datapath si scambiano la stessa informazione:

- La FSM passa al Datapath il segnale di controllo FINE_OPERAZIONE[1], che serve per far smettere da subito al Datapath di modificare il pH e aumentare i cicli di clock,.
- Il Datapath passa alla FSM il segnale di stato END[1], che vale uno(1) nel caso in cui il pH diventa neutro, serve alla FSM nel caso in cui ci si trovi nello stato ACIDO o BASICO per passare allo stato FINE.

Nel Datapath abbiamo creato una componente NEUTRO a se stante, in modo di rendere più chiara la lettura del circuito.