

Relazione progetto SIS

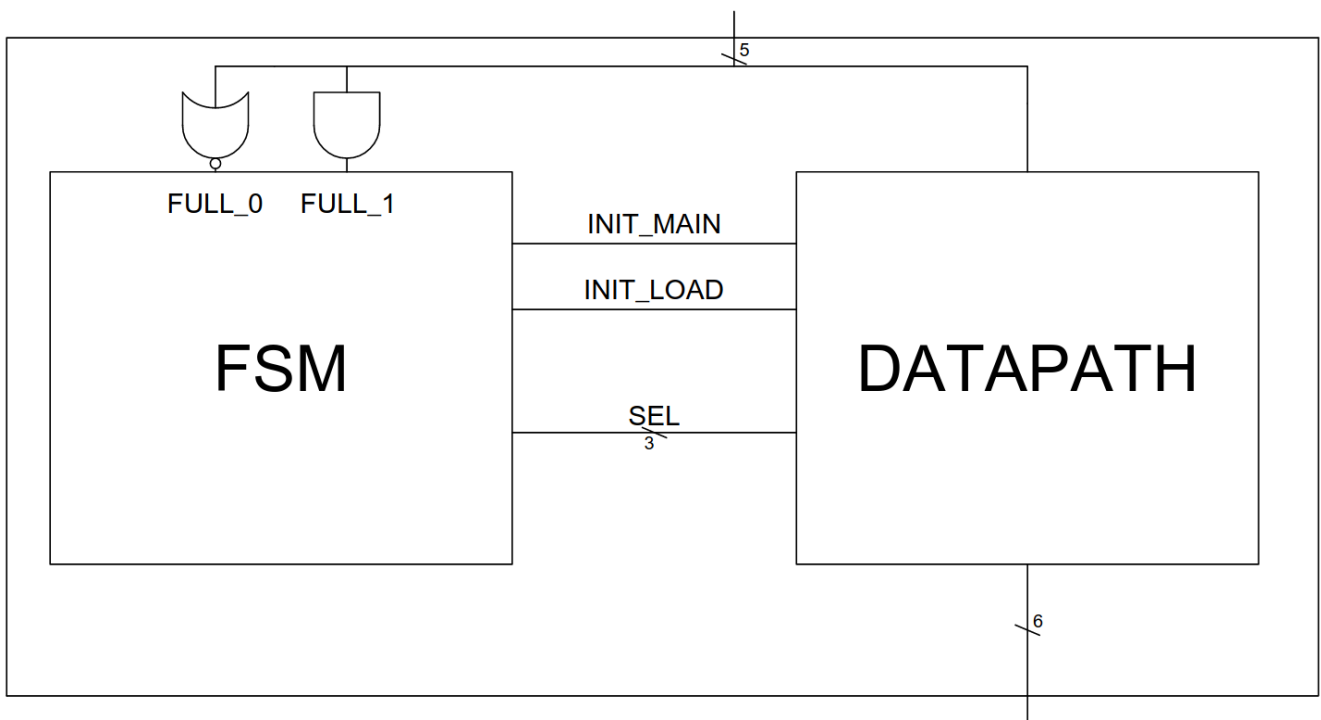
Membri del gruppo

- Alex Zanetti **VR473680**
- Rakib Haque **VR486827**

Introduzione

Progettazione di un sistema per la gestione di un parcheggio con ingresso e uscita automatizzati, suddiviso in tre settori: A(32) ,B(32) ,C(24)

Il sistema viene attivato manualmente la mattina da un operatore e poi funziona autonomamente. L'operatore inserisce manualmente il numero di posti auto disponibili per ogni settore. Gli utenti che desiderano parcheggiare devono selezionare il settore. Il dispositivo risponde aprendo le sbarre e aggiornando lo stato dei posti auto disponibili nei vari settori. Se sorgono errori, la sbarra non si aprirà. Se durante il funzionamento autonomo il sistema riceve **000000**, il sistema si spegnerà.



Funzionamento macchina a stati

Il sistema FSM è controllato da 2 segnali di input e 5 segnali di output.

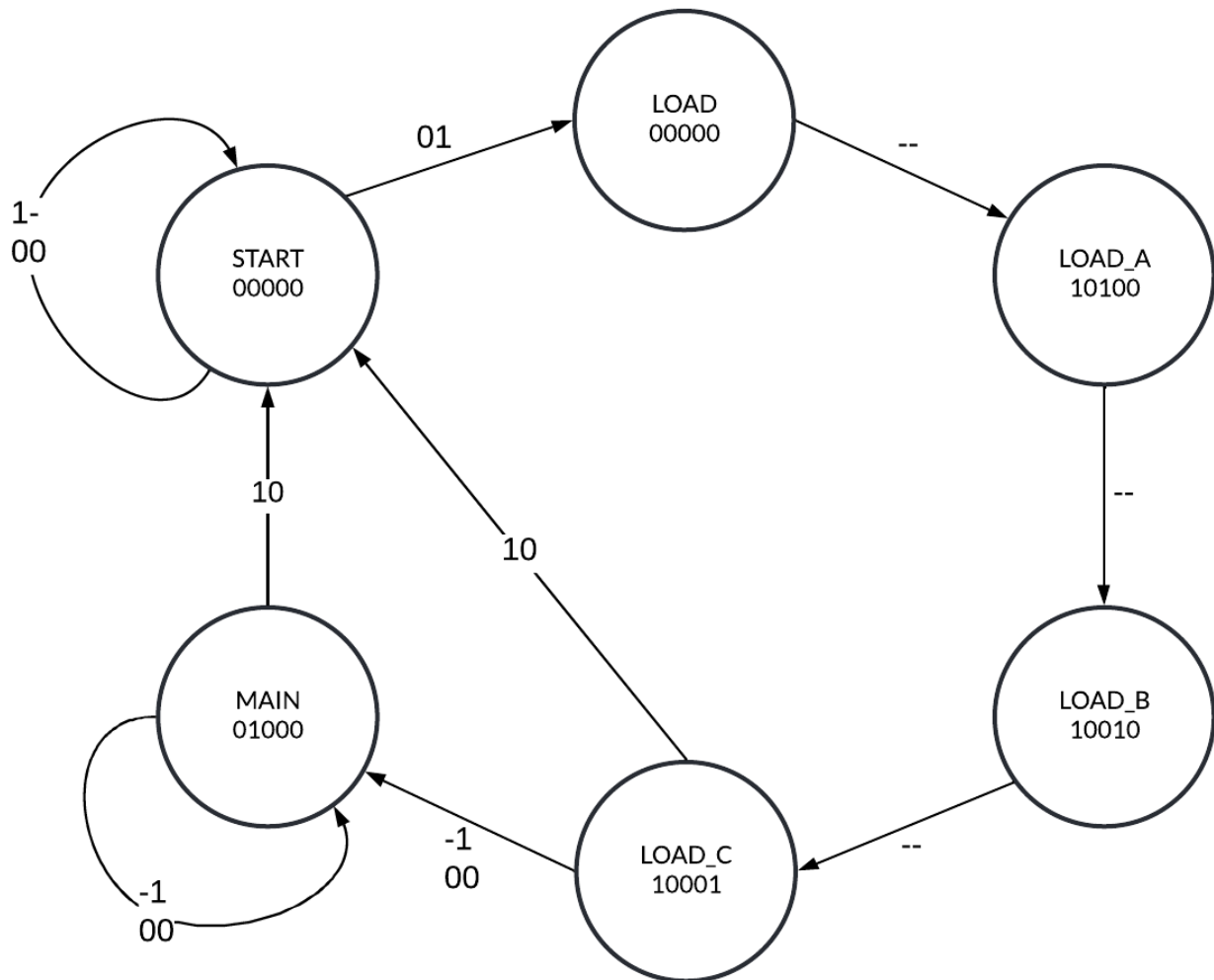
2 segnali di input

- FULL_1 = AND dei 5 input ovvero tutti i bit a 1
- FULL_0 = NOR dei 5 input ovvero tutti i bit a 0

5 segnali di output verso il datapath

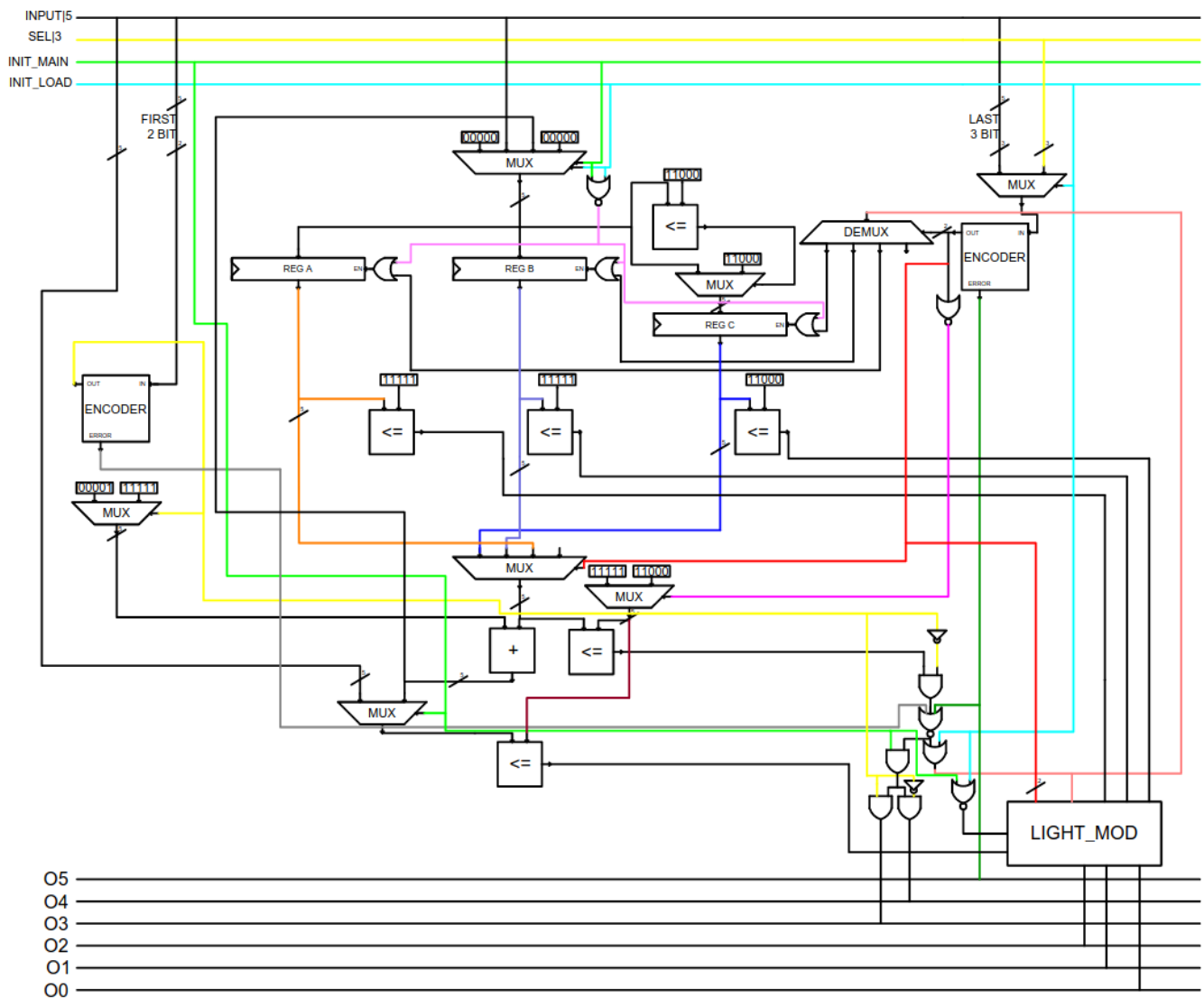
- INIT_MAIN = Attiva il funzionamento normale del parcheggio.
- INIT_LOAD = Attiva la fase di assegnazione dei posti di parcheggio.

- SEL_0, SEL_1, SEL_2 = Selezionano il parcheggio dove assegnare l'input (simulano gli ultimi 3 bit della modalità di parcheggio).

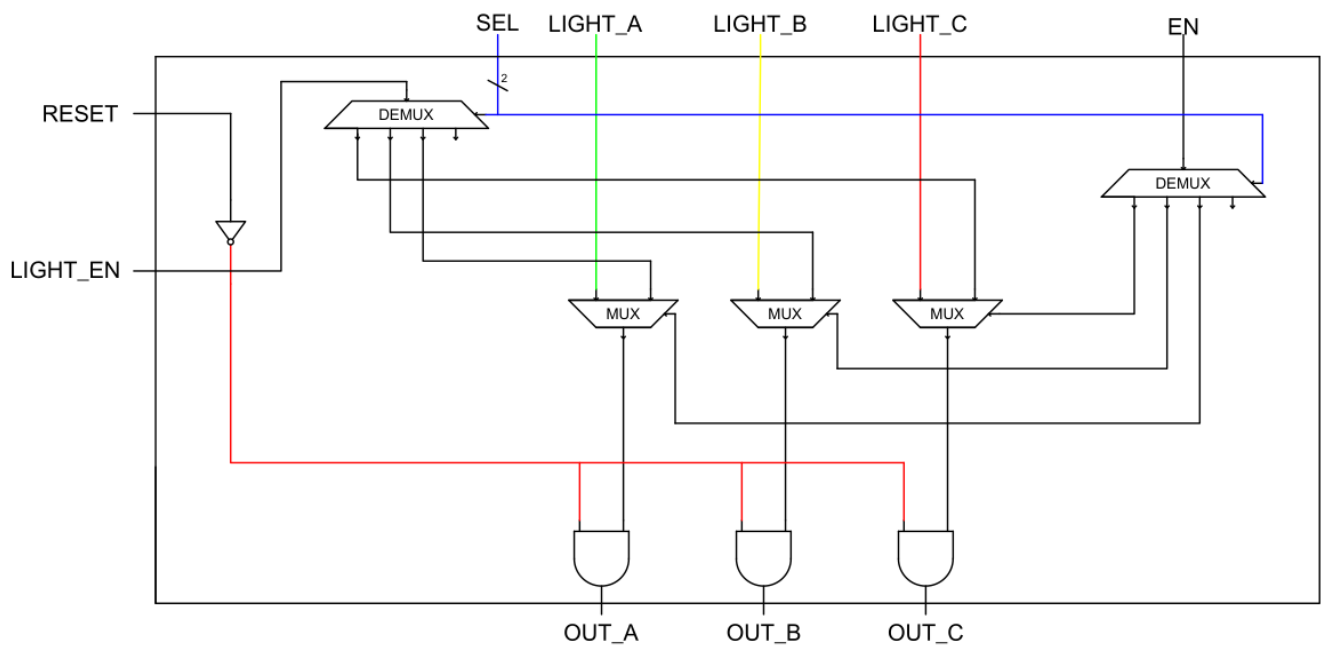


1. Fintanto che l'utente non immetta un FULL_1 il sistema rimarrà su START.
2. Se l'utente immette un FULL_1 il sistema passa da START a LOAD.
3. Per i prossimi 3 input l'utente passerà da LOAD a (LOAD_A,LOAD_B,LOAD,C)
 1. In ogni LOAD verrà selezionato il registro per impostare i posti.
4. Se i bit non sono FULL_0 passa da LOA_C a MAIN
5. Se i bit non sono FULL_0 rimani su MAIN
6. Se i bit sono FULL_0 sia in MAIN che in LOAD_C ritorna a START

Funzionamento datapath



LIGHT_MOD



Encoder ed errori

- Gli encoder servono per decodificare gli input, come la selezione di entrata/uscita o la selezione del settore.

- Gli output degli encoder vengono utilizzati per verificare la presenza di eventuali errori di input (Esempio : errore di settore) e per la selezione dei registri e del valore massimo.

Calcolo

Il calcolo viene effettuato come segue:

1. Viene selezionato un registro tramite un mux.
2. Si verifica se il valore del registro ha raggiunto il suo massimo `ERROR_MAX`.
3. Viene effettuato l'incremento o il decremento di 1 a seconda dell'output del decoder IO. Questo avviene tramite un mux che sceglie tra 11111 e 00001.
4. Viene ricontrollato se il nuovo valore è il massimo e, se lo è, viene accesa la lampadina del settore.
5. Il nuovo valore viene inviato al mux degli input dei registri.
 1. Se non ci sono errori di decoder e di `ERROR_MAX`, il registro selezionato viene abilitato per la scrittura del nuovo numero di posti.

Selezione input

- Tutti i registri hanno lo stesso input che viene scelto da un mux con 4 opzioni:
 1. 00 (reset) - Tutti gli input sono impostati a 0.
 2. 11 (identico a 00)
 3. 10 (LOAD) - L'input proviene dal sistema stesso.
 4. 01 (MAIN) - L'input e l'output del calcolo di +/- 1.
- La scelta dell'input per i registri è controllata da `INIT_MAIN` e `INIT_LOAD`

Selezione registri

- La selezione dei registri viene effettuata tramite un demultiplexer controllato dall'encoder.
- Tutti i registri ricevono gli stessi input, ma la registrazione avviene tramite un segnale di abilitazione generato da un demultiplexer che utilizza come input un `NOR` di tutti i segnali di errore.

Luce del settore

- Ogni registro ha un comparatore che verifica se il settore è pieno e, se lo è, accende la lampadina corrispondente (l'accensione della lampadina ha priorità nella modalità MAIN sull'output del calcolo).

Selettore del massimo

- Si tratta di un semplice mux che, utilizzando come input il decoder di settore e una semplice `NOR` sui suoi 2 output, sceglie il massimo tra `11000` e `11111`

Scelte progettuali

- L'uso di un decoder con un output di errore è stato scelto per la verifica degli errori e per la riduzione dell'uso di mux/demux nella selezione.

- Si è scelto di utilizzare le porte logiche **NOR** e **AND** per filtrare le entrate nel FSM al fine di ridurre il numero di bit in ingresso.
- Per risolvere il problema che il registro C ha posti in meno rispetto agli altri, è stato aggiunto un selettore che permette di scegliere il massimo numero di posti. Tale selettore si basa sull'output del decoder settore e sull'ingresso del registro C. In questo modo, se viene immesso un numero maggiore del necessario, il sistema sceglierà automaticamente il minimo del registro.
- Si è deciso di riutilizzare il decoder settori per l'assegnazione dei registri, al fine di minimizzare l'utilizzo di componenti.

Ottimizzazione

```
sis> ps
FSM          pi= 2   po= 5   nodes= 8       latches= 3
lits(sop)= 53  lits(fac)= 42  #states(STG)= 6
sis> state_minimize stamina
Running stamina, written by June Rho, University of Colorado at Boulder
Number of states in original machine : 6
Number of states in minimized machine : 5
sis> state_assign jedi
Running jedi, written by Bill Lin, UC Berkeley
sis> ps
FSM          pi= 2   po= 5   nodes= 8       latches= 3
lits(sop)= 49  lits(fac)= 38  #states(STG)= 5
```

```
sis> ps
FSMD         pi= 5   po= 6   nodes=212     latches=18
lits(sop)= 863  lits(fac)= 786
sis> source script.rugged
sis> ps
FSMD         pi= 5   po= 6   nodes= 75     latches=18
lits(sop)= 326  lits(fac)= 302
```

Mapping

```
sis> ps
FSMD         pi= 5   po= 6   nodes= 75     latches=18
lits(sop)= 326  lits(fac)= 302
sis> read_library synch.genlib
sis> map -m 0
warning: unknown latch type at node '{[72]}' (RISING_EDGE assumed)
warning: unknown latch type at node '{[73]}' (RISING_EDGE assumed)
warning: unknown latch type at node '{[74]}' (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:          24
total gate area:       6656.00
maximum arrival time: (49.60,49.60)
maximum po slack:      (-7.40,-7.40)
minimum po slack:      (-49.60,-49.60)
total neg slack:       (-969.20,-969.20)
# of failing outputs:  24
>>> before removing parallel inverters <<<
# of outputs:          24
total gate area:       6608.00
maximum arrival time: (47.40,47.40)
maximum po slack:      (-7.40,-7.40)
minimum po slack:      (-47.40,-47.40)
total neg slack:       (-925.20,-925.20)
# of failing outputs:  24
# of outputs:          24
total gate area:       6320.00
maximum arrival time: (47.00,47.00)
maximum po slack:      (-7.40,-7.40)
minimum po slack:      (-47.00,-47.00)
total neg slack:       (-917.60,-917.60)
# of failing outputs:  24
sis> ps
FSMD          pi= 5   po= 6   nodes=188       latches=18
lits(sop)= 416  lits(fac)= 415
```