

Università degli Studi di Verona

Relazione di Elaborato SIS

Corso di laurea in Informatica – Architettura degli
Elaboratori - Laboratorio

Mezzacasa Cristian VR489474

Zeni Davide VR486320

Anno accademico 2022/2023

Specifiche

Si progetti un dispositivo per la gestione di un parcheggio con ingresso/uscita automatizzati.

Il suddetto parcheggio è suddiviso in 3 settori: settori A e B, costituiti da 31 posti macchina ciascuno, ed il settore C con 24 posti macchina. Al momento dell'ingresso l'utente deve dichiarare il settore in cui vuole parcheggiare, analogamente al momento dell'uscita deve dichiarare da quale settore proviene.

Il parcheggio rimane libero durante la notte, permettendo la libera entrata e/o uscita delle automobili. La mattina il dispositivo viene attivato manualmente da un operatore inserendo la sequenza a 5 bit 11111. Al ciclo successivo il sistema attende l'inserimento del numero di automobili presenti nel settore A (sempre in 5 bit) e ne memorizza il valore. Nei due cicli successivi avviene lo stesso per i settori B e C. Nel caso in cui il valore inserito superi il numero di posti macchina nel relativo settore si considerino tutti i posti occupati.

A partire dal quarto ciclo di clock il sistema inizia il suo funzionamento autonomo. Ad ogni ciclo un utente si avvicina alla posizione di ingresso o uscita e preme un pulsante relativo al settore in cui intende parcheggiare.

Il circuito ha 2 ingressi definiti nel seguente ordine:

- IN/OUT [2 bit]: se l'utente è in ingresso il sistema riceve in input la codifica 01, nel caso sia in uscita riceve la codifica 10. Codifiche 00 e 11 vanno interpretate come anomalie di sistema e quella richiesta va ignorata (ovvero non va aperta alcuna sbarra).
- SECTOR [3 bit]: i settori sono indicati con codifica one-hot, ovvero una stringa di 3 bit in cui uno solo assume valore 1 e tutti gli altri 0. La codifica sarà pertanto 100-A, 010-B, 001-C. Codifiche diverse da queste vanno interpretate come errori di inserimento e la richiesta va ignorata.

Ad ogni richiesta il dispositivo risponde aprendo una sbarra e aggiornando lo stato dei posti liberi nei vari settori. Se un utente chiede di occupare un settore già completo il sistema non deve aprire alcuna sbarra.

Il circuito deve avere 4 output nel seguente ordine:

- SETTORE_NON_VALIDO [1 bit]: se il settore inserito non è valido questo bit deve essere alzato
- SBARRA_IN/SBARRA_OUT [1 bit per sbarra]: questo bit assume valore 0 se la sbarra è chiusa, 1 se viene aperta. La sbarra rimane aperta per un solo ciclo di clock, dopo di che viene richiusa (anche se la richiesta al ciclo successivo è invalida)
- SECTOR_(A/B/C) [1 bit a settore]: questo bit assume valore 1 se tutti i posti macchina nel settore sono occupati, 0 se ci sono ancora posti liberi.

Il dispositivo si spegne quando riceve la sequenza 00000 in input. Nel caso in cui venga inserito un settore non valido la sbarra deve rimanere chiusa. Anche in questo caso, i valori dei settori devono comunque essere riportati rispetto alla loro situazione attuale.

Architettura Generale del Circuito

Il circuito è composto da FSM e data-path, rappresentati ciascuno dagli omonimi file blif. Le due sezioni del circuito vengono messe in comunicazione fra loro dal file FSMD.blif.

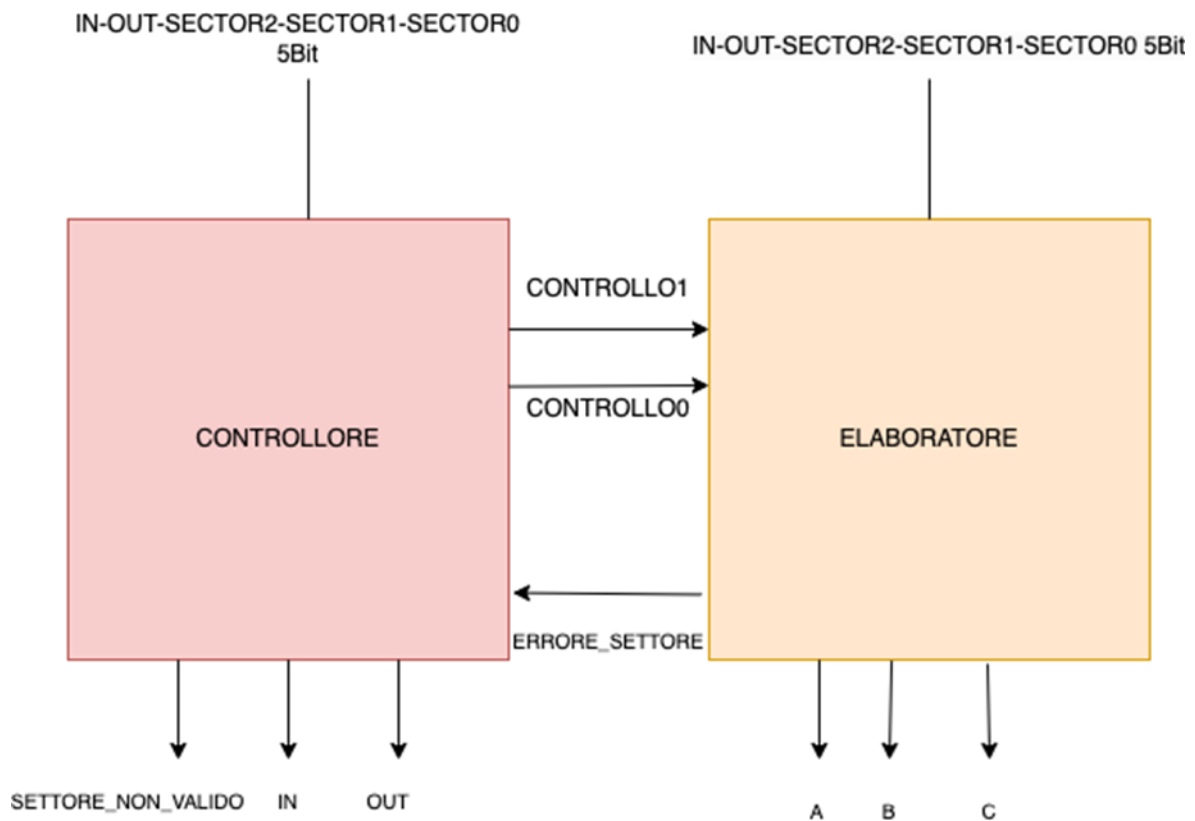
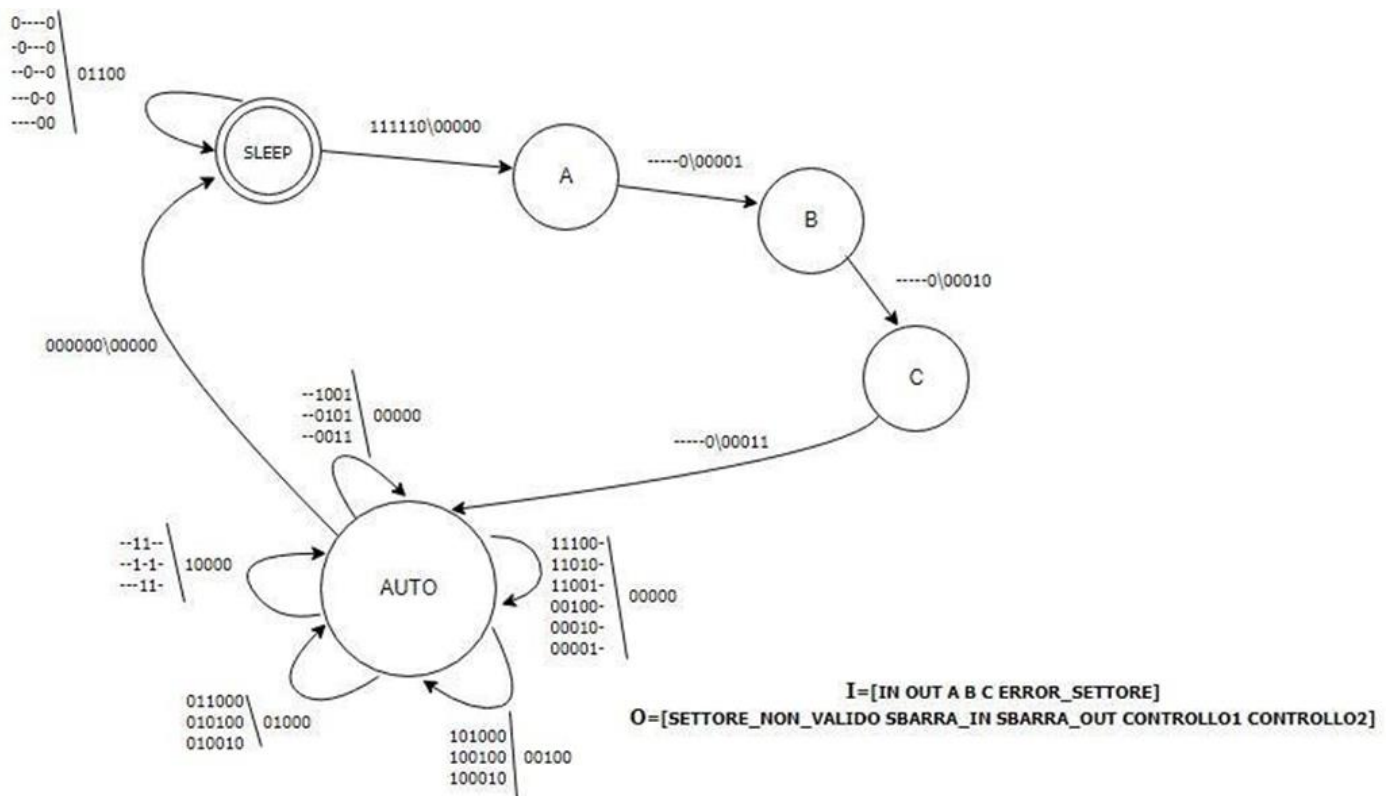


Diagramma degli Stati del

Il controllore è una FSM composta da cinque stati, rispettivamente:

- **SLEEP:** indica lo stato in cui la macchina risulta essere spenta. Nel corso della notte (più correttamente da quando la macchina viene spenta fino alla nuova accensione successiva) la FSM si trova in questo stato.
- **A B C** (tre stati diversi): indicano rispettivamente i settori omonimi del parcheggio. In questi stati vengono indicate le occupazioni dei tre settori nel momento immediatamente successivo all'accensione.
- **AUTO:** indica lo stato nella quale la FSM si imposta per la ricezione delle entrate o uscite delle auto nel parcheggio.

La FSM è definita da sei ingressi e cinque uscite, riportati affianco alla STG di Mealy sottostante:

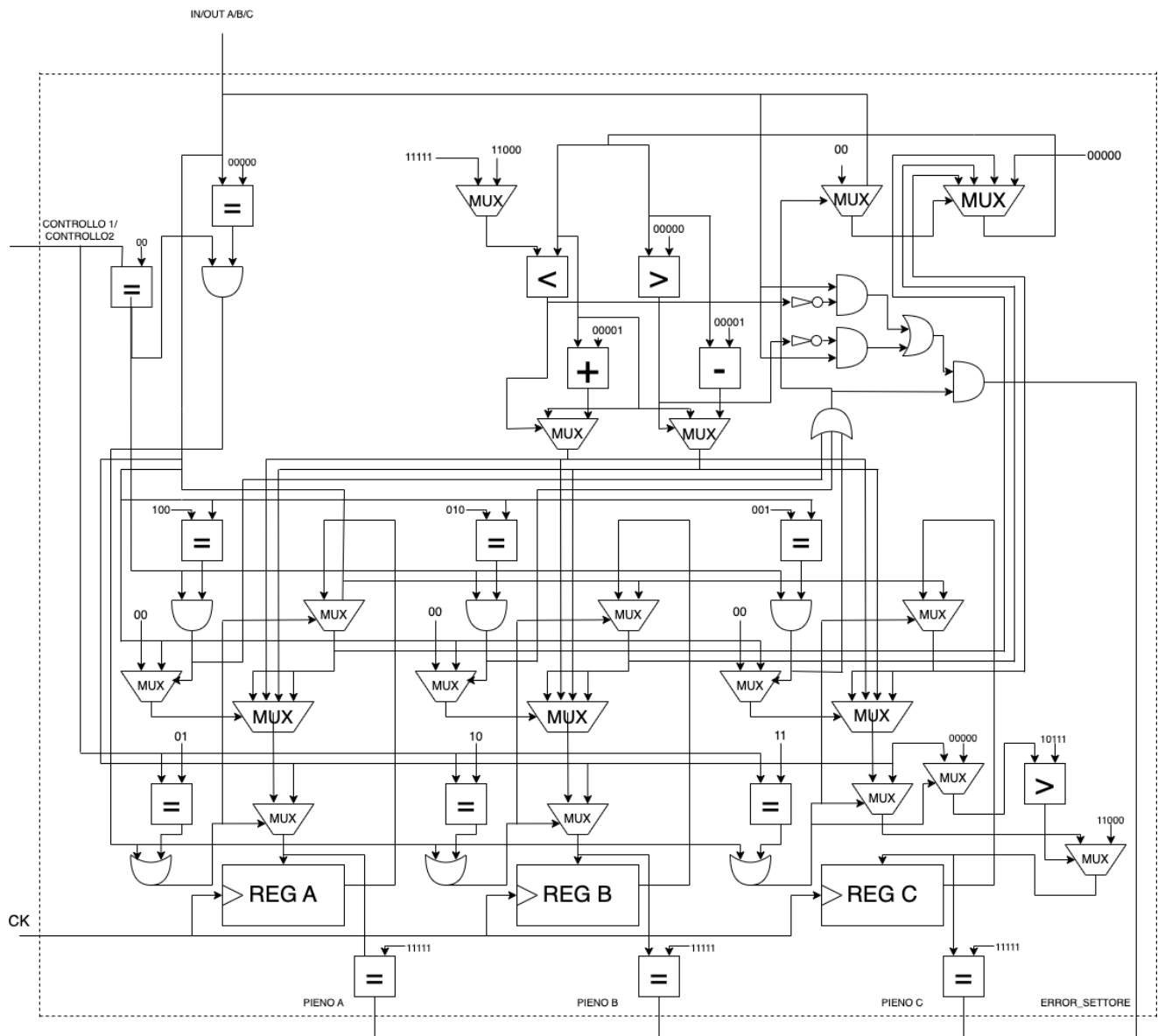


Data-path

I componenti usati per la realizzazione del data-path sono:

- 6 comparatore5bit.blif
- 3 comparatore3bit.blif
- 3 comparatore2bit.blif
- 2 maggiorestretto5bit.blif
- 1 minorestretto5bit.blif
- 1 sommatore5bit.blif
- 1 sottrattore5bit.blif
- 4 multiplexer4_5.blif (quattro ingressi da 5bit)
- 11 multiplexer2_5.blif
- 3 multiplexer2_2.blif
- 3 registro5bit.blif
- 7 and.blif
- 4 or.blif
- 1 or3.blif

A seguito, l'immagine del datapath.



Fase di Inserimento

Nella fase di inserimento degli input si hanno

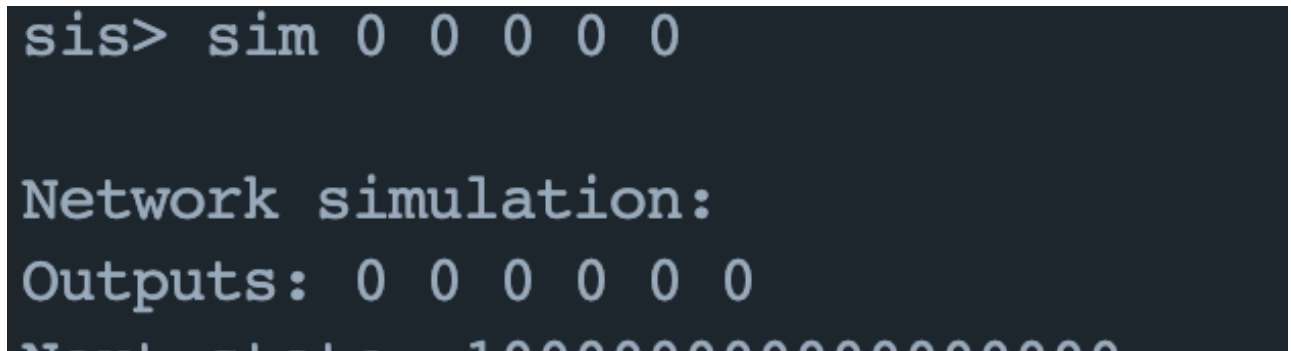
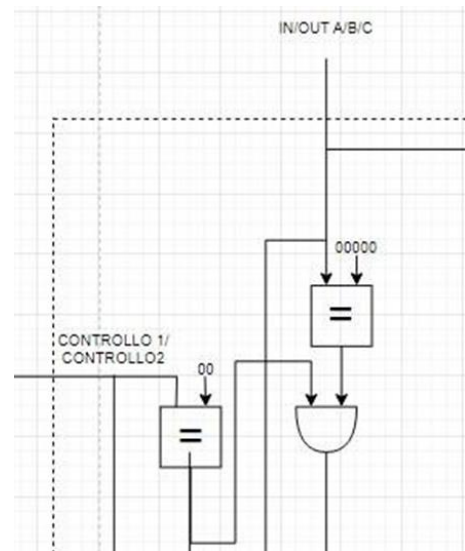
due tipologie di input:

- 2bit di controllo provenienti dalla Fsm
- 5bit di ingresso

Il datapath sa che è nella fase di init grazie ai bit di controllo:

01 -> Stato A 10 -> Stato B 11 -> Stato C

I due comparatori servono per il spegnimento dell'automa una volta ricevuto una sequenza di 5bit a zero la macchina si deve spegnere resettando i registri.



Registro

Il registro deve variare il proprio valore solo in due fasi

- Quando si è nella fase di inizializzazione dei settori Stato A/B/C
- Quando 'automa è in funzione stato AUTO e selezionato il settore con una sequenza di bit corretta
- Reset dei registri

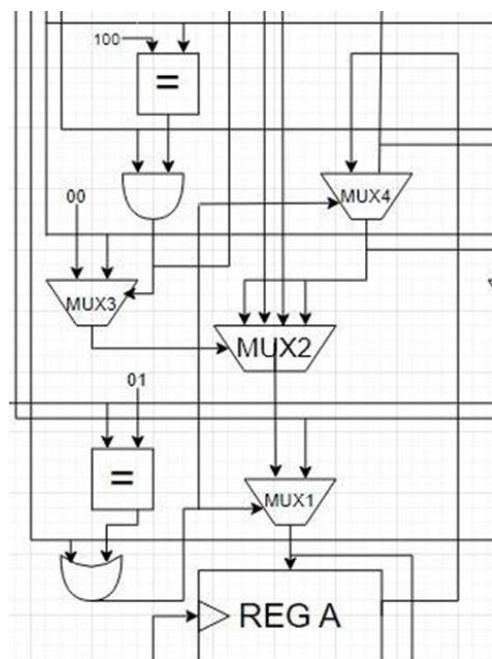
infatti viene utilizzato un comparatore che assume valore positivo se gli input ricevuti dalla FSM è uguale a 01.

Abbiamo utilizzato un Or che serve per il reset dei registri.

All'interno del circuito ci sono 4 multiplexer

Multiplexer 1: ha la funzionalità di selezionare il valore in entrata oppure il valore sommato-sottratto-corrente in base al valore della porta Or

Multiplexer 2 (4 entrate da 5 bit e 2 bit di selezione): questo componente andrà a selezionare il valore corretto da mandare al registro:



- Valore del registro in caso non si dovesse fare nessuna modifica
- Valore risultante dalla operazione di somma
- Valore risultante dalla operazione di sottrazione

Per poter selezionare l'output corretto del multiplexer 2 devo sapere se è stato selezionato il settore A e se sono nella fase AUTO. Quando il multiplexer 3 assume un valore valido seleziona ENTRATA/USCITA

Multiplexer 4 (2 entrate 5 bit): come input riceve il valore del registro oppure il valore inserito questo serve per la fase di inizializzazione:

- Selettore 1 -> Salvo il valore ricevuto in input
- Selettore 0 -> Uso il valore del registro

Selezione del registro

La selezione viene effettuata grazie all'aiuto di una porta Or che assume valore positivo quando almeno un settore è stato selezionato.

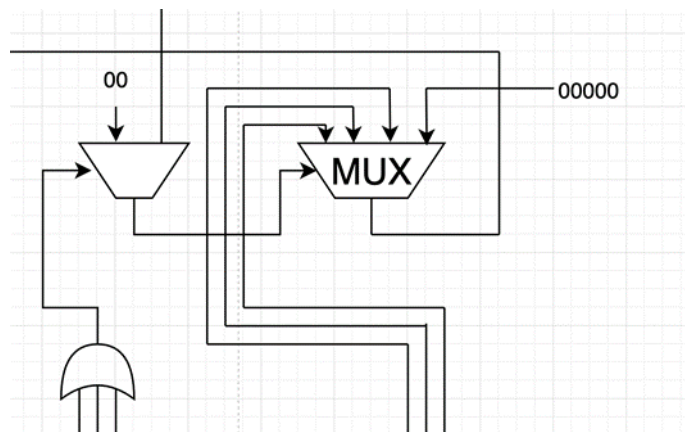
Multiplexer 2 viene utilizzato per selezionare il registro corretto:

11 -> valore di default

00 -> viene selezionato il settore C

01 -> viene selezionato il settore B

10 -> viene selezionato il settore A

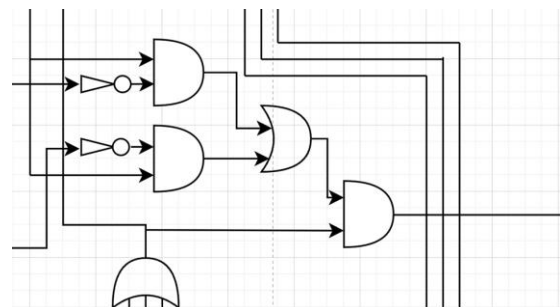


l'input che viene inserito nel multiplexer è I2 I1 dell'input del datapath **10100** Grazie all'Or ho la certezza che non vengano letti dei falsi valori.

Errore settore

L'errore settore è un segnale di stato che viene mandato alla FSM in caso si ha un tentativo di entrare in un settore pieno oppure uscita da un settore vuoto perché altrimenti in questi casi l'FSM restituirebbe output non validi.

Funzionamento: la porta AND in input ricevono due valori in input cioè il settore è pieno oppure e inoltre riceve il terzo bit di inserimento stesso ragionamento per il registro vuoto e quindi per la seconda porta AND.

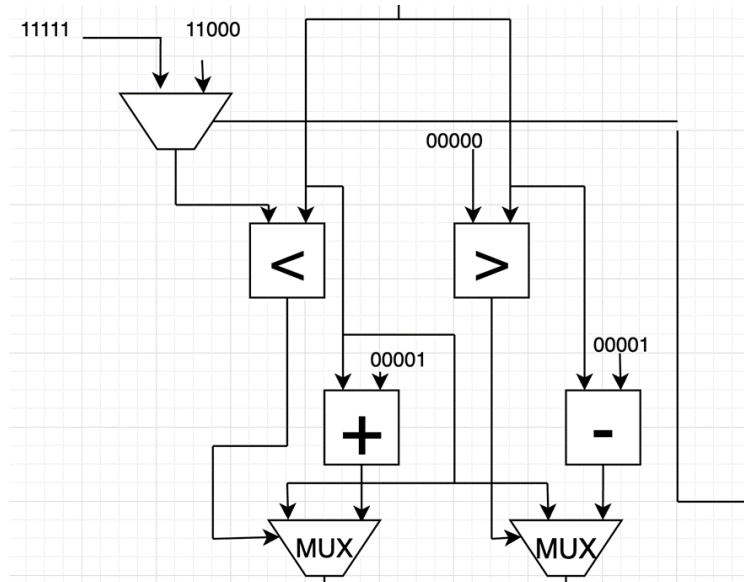


Sommatori e sottrattori

inizialmente si era optato per fare un sommatore e un sottrattore per ogni registro ma questo comportava ad avere un numero alto di aria quindi abbiamo optato per generalizzare il tutto la sottrazione viene effettuata solamente se il valore in ingresso è maggiore di 0 e nel caso in cui il valore sia uguale a 0 il multiplexer andrà a selezionare il valore ricevuto in input e non la sottrazione.

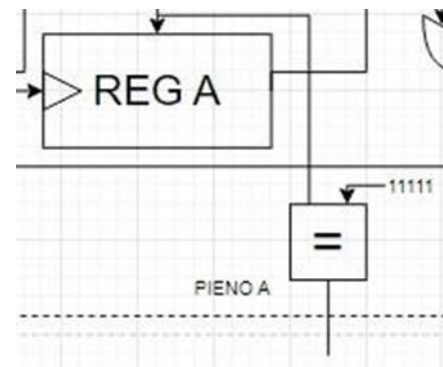
La somma è un po' più problematica perché il settore C ha meno spazi disponibili 11000 (24) quindi abbiamo inserito un multiplexer con due costanti che vengono selezionate in base se il settore C è stato selezionato o meno.

i due valori vanno nei Mux dei 3 registri.



Pieno A/B/C

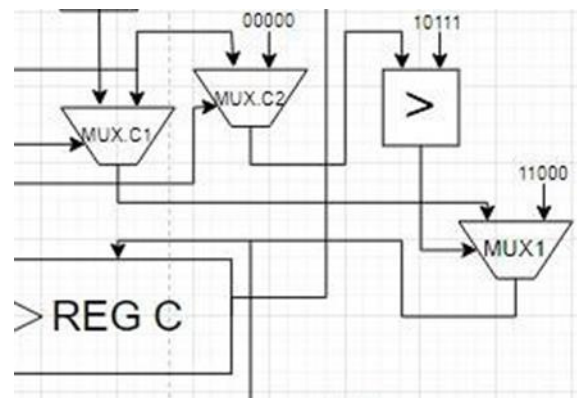
Viene eseguito un controllo per poter dire se i settori sono pieni (e quindi alzare il corrispettivo bit di output): per i settori A e B viene confrontato il valore in input ai loro registri con la loro massima capienza, cioè 31 (11111) posti macchina, per il settore C il confronto viene eseguito con la capienza massima di 24 posti (11000).



Verifica Settore C

Poiché sono presenti valori da input che superano la soglia di posti macchina occupabili nel settore C, nel registro omonimo è necessario eseguire un'ulteriore operazione:

- MUX.C1: ha la medesima funzionalità del MUX1 degli altri registri;
- MUX.C2: riceve l'input che viene passato ad un confronto di maggioranza stretta con 23 (se è maggiore sarà un numero da 24 a 31) in caso di selettore valido (1, fornito da un OR);
- MUX1: contiene l'input e il numero 24; in caso di selettore valido fornito dal confronto questo multiplexer fornirà la capienza massima al registro;



Statistiche

- Prima della minimizzazione

```
fsmd          pi= 5   po= 6   nodes=233   latches=18  
lits(sop)=1110
```

Minimizzazione per area

Per la minimizzazione abbiamo eseguito i seguenti comandi:

- source script.rugged
- fx andando così a scomporre nodi più grandi in nodi più piccoli dove viene trovata una sottoespressione comune

```
bsis> print_stats  
  
fsmd          pi= 5   po= 6   nodes= 77   latches=18  
lits(sop)= 336
```

Dopo la minimizzazione per area si raggiunge un numero di letterali di 336 (-230% rispetto alla versione non ottimizzata), nodi 77 (-202% rispetto alla versione non ottimizzata).

Numero di gate e ritardo

Per il mapping del circuito è stata utilizzata *synch.genlib*

- Prima dell'ottimizzazione

```
>>> before removing serial inverters <<<
# of outputs:          24
total gate area:       11652.00
maximum arrival time:  (45.20,45.20)
maximum po slack:      (-2.80,-2.80)
minimum po slack:      (-45.20,-45.20)
total neg slack:       (-907.20,-907.20)
# of failing outputs:  24
>>> before removing parallel inverters <<<
# of outputs:          24
total gate area:       11588.00
maximum arrival time:  (45.60,45.60)
maximum po slack:      (-2.80,-2.80)
minimum po slack:      (-45.60,-45.60)
total neg slack:       (-908.00,-908.00)
# of failing outputs:  24
# of outputs:          24
total gate area:       9764.00
maximum arrival time:  (44.20,44.20)
maximum po slack:      (-2.80,-2.80)
minimum po slack:      (-44.20,-44.20)
total neg slack:       (-880.00,-880.00)
# of failing outputs:  24
```

- Dopo l'ottimizzazione

Il mapping è stato effettuato, ottimizzato per area map -m 0

```
>>> before removing serial inverters <<<
# of outputs:          24
total gate area:       7424.00
maximum arrival time:  (37.20,37.20)
maximum po slack:      (-6.40,-6.40)
minimum po slack:      (-37.20,-37.20)
total neg slack:       (-637.00,-637.00)
# of failing outputs:  24
>>> before removing parallel inverters <<<
# of outputs:          24
total gate area:       7424.00
maximum arrival time:  (37.20,37.20)
maximum po slack:      (-6.40,-6.40)
minimum po slack:      (-37.20,-37.20)
total neg slack:       (-637.00,-637.00)
# of failing outputs:  24
# of outputs:          24
total gate area:       6992.00
maximum arrival time:  (36.40,36.40)
maximum po slack:      (-6.40,-6.40)
minimum po slack:      (-36.40,-36.40)
total neg slack:       (-629.40,-629.40)
# of failing outputs:  24
```

Differenze dalla versione non ottimizzata:

9764 -> 6992 con un miglioramento del 39%

44 -> 36 con un miglioramento del 22%

Scelte Progettuali

Nel progetto abbiamo pensato di aggiungere una funzionalità al sistema che riguarda la fase in cui risulta essere spento, ovvero di lasciare le sbarre d'entrata e uscita aperte nottetempo per la libera circolazione delle auto. Inserito l'input di spegnimento il codice fornisce un output che presenta i bit SBARRA_IN e SBARRA_OUT alzati.