UNIVERSITÀ di VERONA

Dipartimento di INFORMATICA

Laboratorio di Architettura degli Elaboratori

Elaborato SIS e Verilog

A.A. 2023/2024

Musteata Vadim

VR422249

Specifiche progettuali

Il progetto realizzato prevede la gestione di partite di morra cinese.

Ogni partita di più manche deve seguire le seguenti regole:

- Si devono giocare un minimo di quattro manche MinM=(00100) e un massimo di diciannove, deciso dai giocatori avendo in input Inizio=(1), in cui le mosse valgono come stringa di bit da sommare a MinM per ottenere il numero di manche per partita.
- Vince il primo giocatore a riuscire a vincere due manche in più del proprio avversario, giocando un minimo di 4 manche. In casso viene raggiunto il numero massimo di manche, vince il giocatore che ha un vantaggio maggiore. Con nessun vantaggio si finisce in parità.
- Ad ogni manche, il giocatore vincente della manche precedente non può ripetere l'ultima mossa utilizzata solo per la manca successiva.
  Nel caso lo facesse, la manca viene invalidata .
  In caso di pareggio la manche viene conteggiata. Alla manche successiva, entrambi i giocatori possono usare tutte le mosse.

l circuito comprende 5 bit di input che di dividono in 3 ingressi:

Primo [2 bit] – mossa del primo giocatore
00 → nessuna mossa, considerata come mossa non valida
01 → sasso
10 → carta
11 → forbice

Secondo [2 bit] – mossa del secondo giocatore

Le mosse hanno gli stessi codici del primo giocatore.

Inizia [1 bit] – bit di reset
1 → riporta il sistema alla configurazione iniziale e usa le mosse dei due giocatori come numero da sommare a MinM per settare il numero di manche massime della partita.
0 → la partita procede regolarmente.

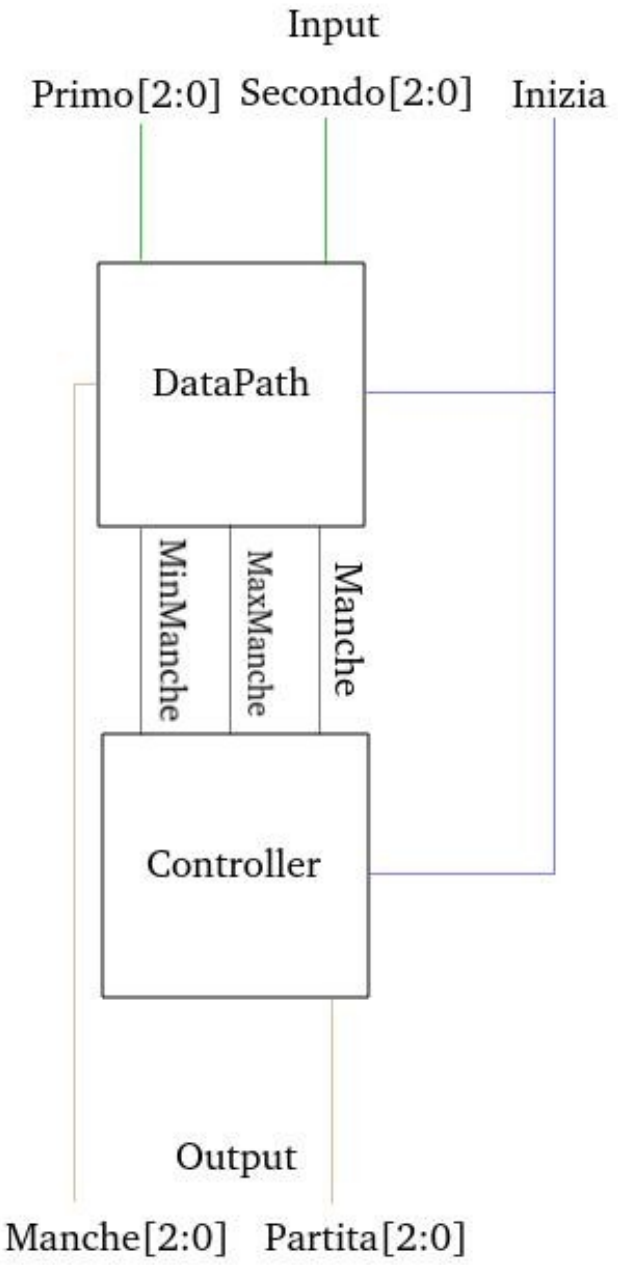
Il circuito comprende inoltre 4 bit di output che di dividono in 2 uscite:

Manche [2 bit] – manche (fornisce il risultato dell'ultima manche giocata)
00 → manche non valida
01 → manche vinta dal giocatore 1
10 → manche vinta dal giocatore 2
11 → manche pareggiata

Partita [2 bit] – partita (fornisce il risultato della partita)
00 → la partita non è terminata
01 → la partita è terminata, ed ha vinto il giocatore 1
10 → la partita è terminata, ed ha vinto il giocatore 2
11 → la partita è terminata in pareggio

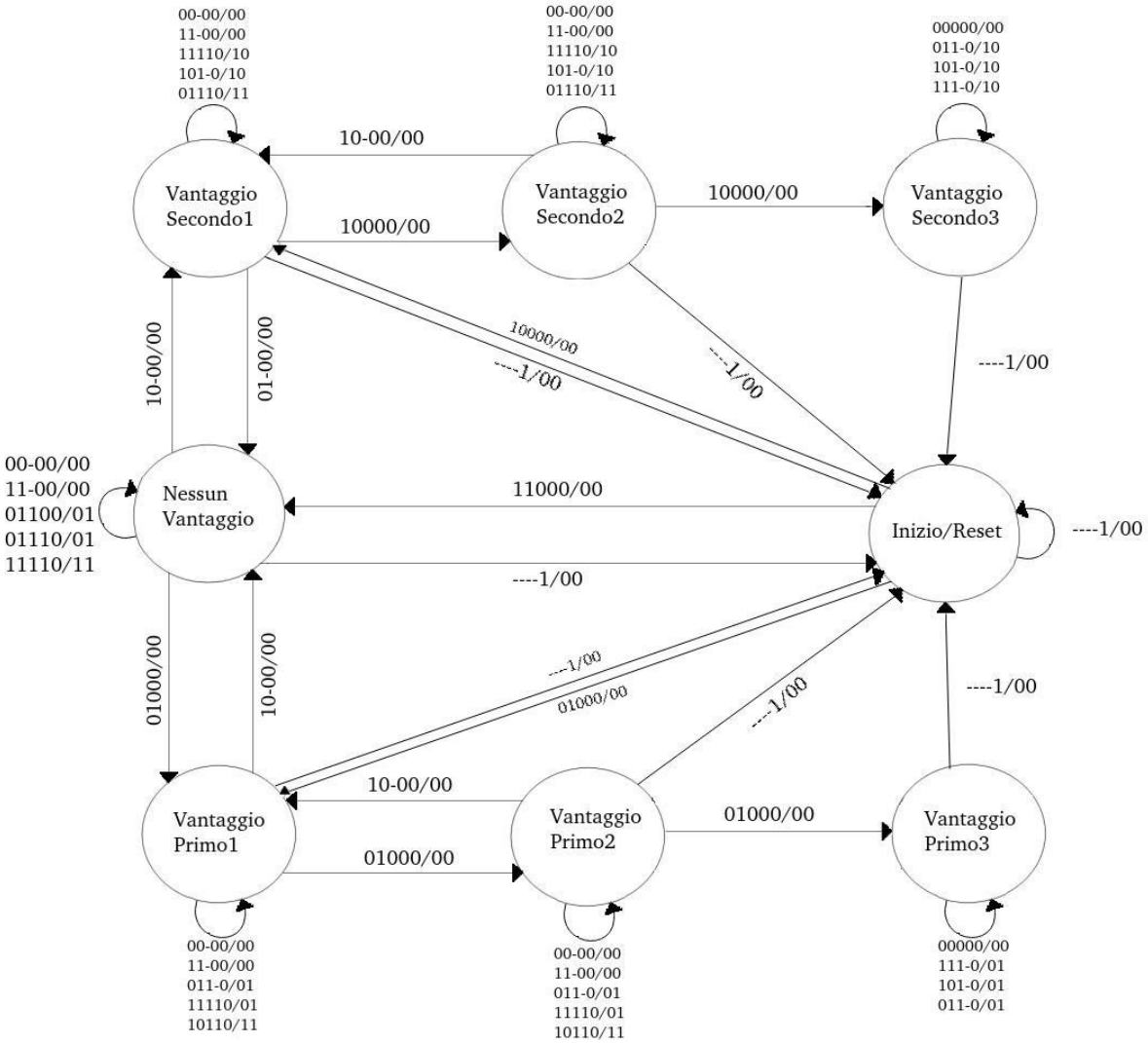
Architettura generale del circuito (schema FSMD)

Le variabili passate dal DataPath al Controllore sono i seguenti:



1. **Manche:** E il risultato dell'ultima manca giocata dai giocatori quale fornita al controller li permette di avanzare da uno stato all'altro e capire chi sta vincendo.
2. **MaxManche:** E un controllo quale verifica se si e raggiunto il numero massimo di manche.
3. **MinManche:** E un controllo per verificare se e stato superato il minimo numero di manche richieste.

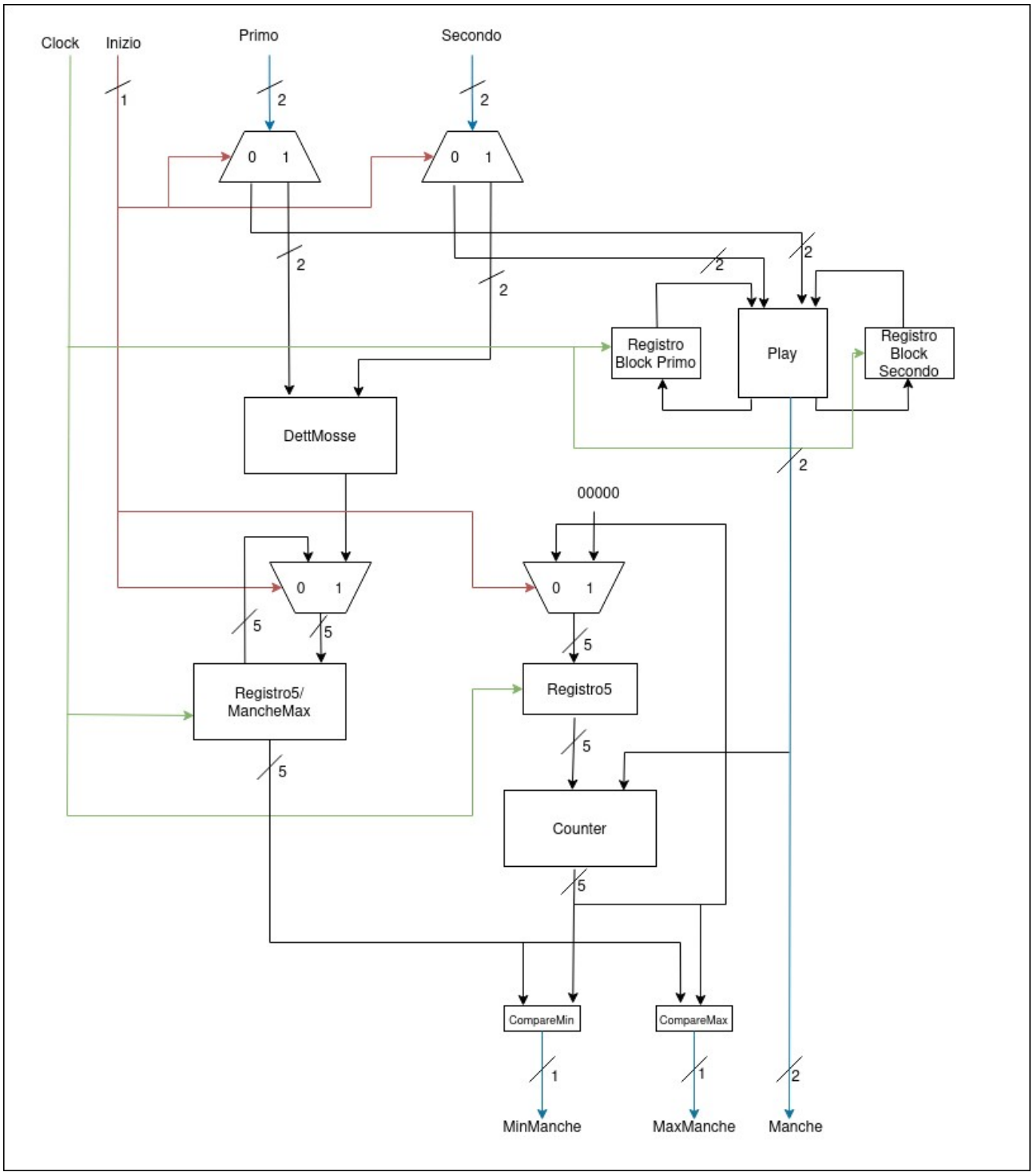
Diagramma del Controllore



La FSM e formata da 8 stati:

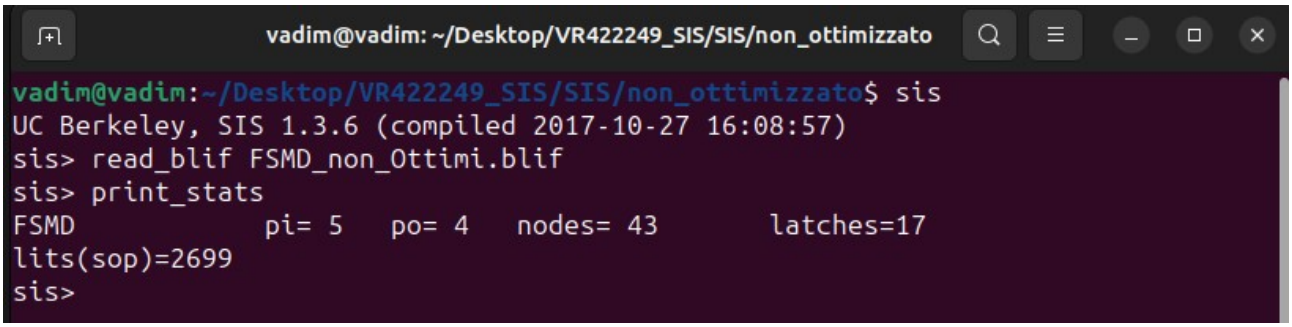
1. Inizia/Reset: Il controllore aspetta un input del bit Inizio=1 prima di iniziare a fare qualsiasi operazione.
2. Nessun Vantaggio: Stato in quale entrambi ha un numero uguale di vantaggio
3. Vantaggio Primo1 / Secondo1: Stati in quale uno dei giocatori ha un punto di vantaggio in più dell’altro.
4. Vantaggio Primo2 / Secondo2: Stati in quale uno dei giocatori ha due punti di vantaggio in più dell’altro.
5. Vantaggio Primo3 / Secondo3: Si può verificare solo all’inizio della partita dato che c'è un minimo di 4 manche da giocare.

Architettura del Datapath

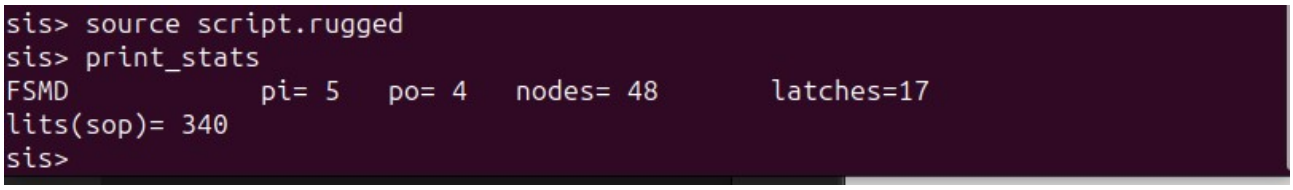


Ottimizzazion

Prima dell'Ottimizzazione il circuito ha 43 nodi e 2699 letterali



Dopo l'Ottimizzazione lanciando tre volte il full_simplify e due volte il source script.rugged si ottiene un circuito di 48 nodi e 340 letterali.



Mapping Tecnologico

```
sis> read_library synch.genlib
sis> map -m 0 -s
warning: unknown latch type at node '{[19]}' (RISING_EDGE assumed)
warning: unknown latch type at node '{[20]}' (RISING_EDGE assumed)
warning: unknown latch type at node '{[21]}' (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
>>> before removing serial inverters <<<
# of outputs:          21
total gate area:       6704.00
maximum arrival time: (75.20,75.20)
maximum po slack:     (-8.20,-8.20)
minimum po slack:     (-75.20,-75.20)
total neg slack:      (-795.00,-795.00)
# of failing outputs:  21
>>> before removing parallel inverters <<<
# of outputs:          21
total gate area:       6464.00
maximum arrival time: (70.80,70.80)
maximum po slack:     (-8.20,-8.20)
minimum po slack:     (-70.80,-70.80)
total neg slack:      (-769.20,-769.20)
# of failing outputs:  21
# of outputs:          21
total gate area:       6224.00
maximum arrival time: (68.80,68.80)
maximum po slack:     (-8.20,-8.20)
minimum po slack:     (-68.80,-68.80)
total neg slack:      (-749.80,-749.80)
# of failing outputs:  21
sis>
```