

Laboratorio di Architettura degli Elaboratori

Elaborato SIS e Verilog

A.A. 2023/2024

Specifiche:

Si progetti un dispositivo per la gestione di partite della *morra cinese*, conosciuta anche come *sasso-carta-forbici*.

Due giocatori inseriscono una mossa, che può essere *carta*, *sasso*, o *forbici*. Ad ogni manche, il giocatore vincente è decretato dalle seguenti regole:

- Sasso batte forbici;
- Forbici batte carta;
- Carta batte sasso.

Nel caso in cui i due giocatori scelgano la stessa mossa, la manche finisce in pareggio.

Per renderla più avvincente, ogni partita si articola di più manche, con le seguenti regole:

- Si devono giocare un **minimo di quattro manche**;
- Si possono giocare un **massimo di diciannove manche**. Il numero massimo di manche, viene settato al ciclo di clock in cui viene iniziata la partita;
- Vince il primo giocatore a riuscire a **vincere due manche in più del proprio avversario**, a patto di aver giocato **almeno quattro manche**;
- Ad ogni manche, il **giocatore vincente della manche precedente non può ripetere l'ultima mossa** utilizzata. Nel caso lo facesse, la manche non sarebbe valida ed andrebbe ripetuta (quindi, non conteggiata);
- Ad ogni manche, **in caso di pareggio la manche viene conteggiata**. Alla manche successiva, entrambi i giocatori possono usare tutte le mosse.

Il circuito ha **tre ingressi**:

- **PRIMO [2 bit]**: mossa scelta dal primo giocatore. Le mosse hanno i seguenti codici:
 - **00**: nessuna mossa;
 - **01**: Sasso;
 - **10**: Carta;
 - **11**: Forbice;
- **SECONDO [2bit]**: mossa scelta dal secondo giocatore. Le mosse hanno gli stessi codici del primo giocatore.
- **INIZIA [1 bit]**: quando vale 1, riporta il sistema alla configurazione iniziale. Inoltre, la concatenazione degli ingressi PRIMO e SECONDO viene usata per specificare il numero massimo di manche oltre le quattro obbligatorie. Ad esempio, se si inserissero i valori PRIMO = 00 e SECONDO = 00, si indicherebbe di giocare esattamente quattro manche. Se si inserisse il valore PRIMO = 00 e SECONDO = 01, si indicherebbe di giocare al più 5 manche (le 4 obbligatorie, più il valore 1 indicato da 0001). Se si inserissero i valori PRIMO = 10 e SECONDO = 01, si indicherebbe di giocare tredici manche (le 4 obbligatorie, più il valore 9 indicato da 1001). Quando vale 0, la manche prosegue normalmente.

Il circuito ha **due uscite**:

- **MANCHE [2 bit]**: fornisce il risultato dell'ultima manche giocata con la seguente codifica:
 - **00**: manche non valida;
 - **01**: manche vinta dal giocatore 1;
 - **10**: manche vinta dal giocatore 2;
 - **11**: manche pareggiata.
- **PARTITA [2 bit]**: fornisce il risultato della partita con la seguente codifica:
 - **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.

Il circuito deve essere implementato in Verilog (nello stile behavioral) ed in SIS. Gli ingressi ed uscite delle implementazioni Verilog e SIS devono avere lo stesso ordine riportato sopra. Il modulo principale Verilog dovrà chiamarsi **MorraCinese**. Il testbench Verilog deve generare, mediante stampa su file chiamato **testbench.script**, uno script che funga da testbench per il modello SIS.

Il modello Verilog deve generare un file, chiamato **output_verilog.txt** che ad ogni ciclo di clock riporta gli output con il seguente formato:

Outputs: MANCHE[1] MANCHE[0] PARTITA[1] PARTITA[0]

Il file **output_verilog.txt** potrà essere usato per confrontare i risultati ottenuti dal modello SIS. Il progetto in Verilog fornisce una traccia per sviluppare la versione SIS. Dunque, verrà automaticamente scartato qualsiasi progetto in cui le versioni Verilog e SIS non producono la stessa esecuzione.

Materiale da consegnare

1. Sorgenti Verilog dell'intero circuito e del suo testbench:
 - a. Il file principale del modello deve chiamarsi **design.sv**;
 - b. Il file principale del testbench deve chiamarsi **testbench.sv**;
 - c. Il file **output_verilog.txt**, generato come descritto sopra.
2. Sorgenti BLIF dell'intero circuito:
 - a. Il file principale per lanciare la simulazione deve chiamarsi **FSMD.blif** (solo FSMD in maiuscolo);
 - b. La versione caricata deve essere quella già ottimizzata, ma pre-mapping;
 - c. Il file **testbench.script**, generato dal testbench Verilog, come descritto sopra;
 - d. I dati di area e ritardo devono coincidere con quelli riportati nella Relazione;
 - e. Una sotto-cartella denominata **non_ottimizzato/** contenente i sorgenti pre-ottimizzazione.
3. Relazione in formato **pdf** denominata **Relazione.pdf**. La relazione deve essere in formato A4 (no slide) e deve riportare in prima pagina il numero di matricola, nome e cognome di entrambi gli studenti. La relazione deve riportare e spiegare nel dettaglio almeno i seguenti punti:
 - a. L'architettura generale del circuito (schema FSMD);
 - b. Il diagramma degli stati del controllore;
 - c. L'architettura del datapath;
 - d. Le statistiche del circuito prima e dopo l'ottimizzazione **per area**;
 - e. Il numero di gate e ritardo ottenuti mappando il design sulla libreria tecnologica **synch.genlib**;
 - f. La descrizione di eventuali scelte progettuali effettuate.

La struttura della cartella dovrà essere la seguente, pena la non ammissibilità dell'elaborato:

- VRXXXXXX_VRXXXXXX/
 - sis/
 - non_optimizzato/ (cartella contenente i sorgenti pre-ottimizzazione);
 - sorgenti blif del circuito (non ottimizzato);
 - FSMD.blif (ottimizzato);
 - Eventuali altri file blif generati dal processo di ottimizzazione;
 - testbench.script;
 - output_sis.txt contenente l'output filtrato (seguendo le indicazioni viste in lezione 6) ottenuto eseguendo lo script testbench.script.
 - verilog/
 - design.sv;
 - testbench.sv;
 - eventuali altri file verilog necessari alla simulazione;
 - output_verilog.txt.
 - Relazione.pdf

Modalità di consegna:

Tutto il materiale va consegnato elettronicamente tramite procedura guidata sul sito Moodle del corso. Sarà attivata un'apposita sezione denominata "Consegna SIS - <mese> <anno>".

Accedendo alla pagina sarà possibile effettuare l'upload del materiale. La consegna dovrà essere effettuata da solamente uno dei due studenti del gruppo, e la consegna del materiale comporta automaticamente l'iscrizione all'appello orale di entrambi gli studenti.

Il codice e la relazione vanno compressi in un unico file tarball denominato:

VRXXXXXX_VRXXXXXX.tar.gz

Dove VRXXXXXX rappresenta le matricole degli studenti che compongono il gruppo. Ogni gruppo deve essere formato da **esattamente 2 studenti**. Non è possibile consegnare il progetto individualmente, e non è possibile consegnare il progetto in più di due studenti. Nel caso ci fossero difficoltà nel formare la coppia per lo svolgimento del progetto, su Moodle è disponibile un forum apposito.

Il pacchetto tarball deve contenere un'unica cartella denominata VRXXXXXX_VRXXXXXX, organizzata come descritto sopra.

Verranno accettati solo i progetti compressi in formato tarball (.tar.gz oppure .tgz).

Per ottenere il pacchetto come richiesto:

1. Rinominare la cartella contenente tutto il materiale con il nome VRXXXXXX_VRXXXXXX.
2. Salire di un livello rispetto alla cartella e lanciare il comando:

```
$> tar cvfz VRXXXXXX_VRXXXXXX.tar.gz VRXXXXXX_VRXXXXXX/
```

Ad esempio, se il gruppo è formato da due studenti con matricole VR123456 e VR654321, bisogna ottenere il file VR123456_VR654321.tar.gz, contenente la cartella VR123456_VR654321, mediante il comando: `tar czfv VR123456_VR654321.tar.gz VR123456_VR654321/`

Scadenza: **11 Febbraio 2024**, Ore 23:59

Periodo esami orali (indicativo): 15 – 23 Febbraio.

Note importanti:

1. È possibile effettuare più sottomissioni, ma ogni nuova sottomissione cancella quella precedente.
2. Un solo membro del gruppo deve effettuare la sottomissione.
3. Tutti i componenti del gruppo devono essere iscritti alla pagina Moodle del corso.
4. I gruppi possono essere composti da studenti appartenenti ad entrambe le classi, e a tutti gli anni di corso, a patto che tutti i membri del gruppo debbano sostenere l'esame, ossia che nessuno dei membri abbia già verbalizzato sul libretto l'esame di Architettura degli Elaboratori.
5. Non si accettano progetti consegnati via email e/o dopo la scadenza.
6. I progetti che non soddisfano i requisiti sopraelencati non verranno ammessi all'orale e non verranno valutati.
7. **Tutti i progetti verranno testati automaticamente. Solo i progetti che superano i test saranno ammessi alla discussione orale. Per valutare l'ammissione all'orale, verranno valutate:**
 - a. La struttura dell'archivio consegnato, e delle cartelle contenute (check automatico);
 - b. La relazione, e la presenza di tutte le informazioni minime richieste (semi-automatico);
 - c. La correttezza del modello Verilog, e dei file generati (automatico);
 - d. La correttezza del modello Sis (automatico);
 - e. L'equivalenza comportamentale del modello Sis rispetto al Verilog (automatico).
8. I progetti non ammessi potranno essere visionati e discussi al termine delle sessione su richiesta degli studenti.
9. Qualsiasi chiarimento dovrà essere richiesto attraverso il forum "*Chiarimenti sugli elaborati*" disponibile su Moodle, seguendo le regole descritte nella pagina principale del forum.