

Progetti Verilog – A.A. 2018-2019

Architettura degli Elaboratori – Corso A

Scopo del progetto

Qualunque progetto scegliate fra quelli proposti, il progetto consiste nella progettazione e realizzazione di una unità firmware. Il progetto dell'unità deve avvenire secondo la metodologia sviluppata nel corso di AE. La realizzazione va effettuata implementando l'unità come modulo Verilog rispettando le specifiche e l'interfaccia descritta nel testo. Nella consegna è richiesto anche un modulo test che metta in evidenza le funzionalità del modulo Verilog implementato e che costituisce parte integrante del progetto. Per l'implementazione dell'unità firmware si richiede esplicitamente la progettazione della Parte Operativa e della Parte Controllo come moduli distinti. Ciascuno dei due moduli può essere implementato utilizzando altri moduli, per esempio quelli che mettono a disposizione i componenti standard. Per ciascuno dei moduli implementati è consigliato realizzare un modulo di test che ne possa verificare il corretto funzionamento prima che il modulo venga utilizzato all'interno di altri moduli.

Strumenti

Per la realizzazione del codice si consiglia di utilizzare l'interprete **iverilog** (disponibile all'indirizzo <http://iverilog.icarus.com/>) di cui esistono versioni funzionanti sotto Linux, Mac OS X e Windows. Per la stesura del codice potete utilizzare qualunque editor di vostro gradimento. Il test dei risultati avverrà utilizzando una macchina Linux.

Esistono toolkit per la programmazione di FPGA che mettono a disposizione ambienti integrati per la realizzazione di programmi in Verilog che non richiedono licenza (Quartus Altera/Intel e Vivado WEBpack Xilinx) che comunque richiedono svariati giga di spazio disco e hanno una curva di apprendimento considerevolmente maggiore di quella di **iverilog**.

Modalità di consegna

Potete consegnare il progetto in concomitanza delle date degli appelli scritti di AEa 2018—19. La correzione del progetto può richiedere un po' di tempo, ma di norma gli orali potrebbero aver luogo insieme agli orali di chi ha superato lo scritto. I progetti riportati qui sotto sono validi per tutti gli appelli dell'anno accademico.

Progetti

Processore MINI D-RISC

Si richiede la realizzazione di un processore sequenziale, cioè che esegue una istruzione assembler alla volta, che metta a disposizione le sole istruzioni LOAD, STORE, ADD, SUB, IF<, IF>, IF=, CALL, e GOTO. Nella realizzazione del processore si può omettere la parte relativa al trattamento delle eccezioni (ESITO diverso da zero dalla memoria), mentre per il trattamento delle eccezioni occorre implementare l'intera fase firmware. L'unità ha un'interfaccia di memoria standard come unica interfaccia verso il mondo esterno, in aggiunta agli indicatori a transizione di livello per la gestione delle interruzioni.

```
module minidisc(input IN, output ACKINT,  
               output RDYM, output [31:0]DATAOUT, output [31:0]IND, output [1:0]OP,  
               input ACKM, input [2:0]ESITO, input [31:0]DATAIN);  
endmodule
```

Cache set associativa

Si richiede di realizzare una cache set associativa a due vie, con linee da $\sigma=8$ parole che implementa una politica write through. Alla cache arrivano direttamente richieste con indirizzi fisici. La cache è direttamente collegata ad una memoria interallacciata a 4 moduli tramite un'interfaccia simile all'interfaccia standard. La sola differenza consiste nei registri DATAIN e DATAOUT che sono di costituiti da 4 parole anziché da una parola sola.

```
module cachesetassoc(// interfaccia verso la memoria  
                    output RDYM, output [127:0]DATAOUT,  
                    output [31:0]IND, output [1:0]OP,  
                    input ACKM, input [2:0]ESITO, input [127:0]DATAIN,  
                    // interfaccia dalla MMU  
                    input RDYP, input [0:1]OPP, input [31:0]DATAINP, input [31:0]INDP,  
                    output ACKP, output [2:0]ESITOP, output [31:0]DATAOUTP);  
endmodule
```

Unità IM processore pipeline D-RISC

Si richiede di realizzare una unità che implementa l'unità IM del processore D-RISC come descritta durante le lezioni e nel materiale didattico. In particolare, l'unità dovrebbe generare lo stream di istruzioni da 32 bit verso l'unità IU e prevedere la possibilità di ricevere richieste di aggiornamento della propria copia dell'Instruction Counter dalla stessa unità IU.

```
module IM(// interfaccia dalla unità IU per la ricezione di un nuovo valore IC  
         input RDY1, input [31:0] NEWIC, output ACK1,  
         // interfaccia verso la unità IU per la spedizione della coppia <istruzione  
         corrente, IC>  
         input ACK2, output [31:0] IC, output [31:0] INSTR, output RDY2);  
endmodule
```

Unità stack con operazioni aritmetiche

Si richiede di realizzare una unità che presenta un'interfaccia di memoria "standard" (come quella del sottosistema di memoria verso il processore D-RISC ma priva del campo INDIRIZZO) e che implementa le seguenti operazioni esterne:

- **PUSH(va1)** : inserisce in cima allo stack il valore va1 di tipo intero

- **POP()** : restituisce il valore in cima allo stack, cancellandolo
- **SOMMA (o sottrazione)** : restituisce il risultato della somma (o della sottrazione) fra le due celle in cima allo stack. Se non vi sono almeno due posizioni occupate, restituisce un esito negativo.
- **MEDIA (n)** : restituisce il valore della media (intera) delle n posizioni in cima allo stack, restituendo un errore qualora lo stack non contenesse n valori.

Lo stack ha capacità pari a 1024 parole da 32 bit. Eventuali errori devono essere comunicati mediante il campo ESITO.

```
module stack(// interfaccia dal processore
    input RDY, input [2:0]OP, input [31:0]DATAIN,
    output ACK, output [31:0]DATAOUT, output [2:0]ESITO);
endmodule
```

Istruzioni per l'uso

La realizzazione del progetto richiede i seguenti passi:

1. Scelta di uno dei progetti proposti. La scelta deve essere concordata con il docente prima di procedere alla realizzazione del progettino, o a ricevimento o per email (email con Subject "AEa 2018-19: scelta progetto Verilog". Nel corpo del messaggio vanno indicati Nome Cognome e Matricola).
2. Progettazione della soluzione, con la metodologia e gli strumenti imparati nel corso di AE, quindi senza l'utilizzo di Verilog
3. Realizzazione di un modulo Verilog che implementa l'unità scelta. Il modulo Verilog va costruito secondo un processo di composizione di moduli elementari che implementano i componenti standard visti a lezione e/o eventuali moduli "primitive" (reti combinatorie) progettate ad hoc. Nei moduli che implementano i componenti standard vanno cablati a mano gli eventuali ritardi.
4. Preparazione di una breve relazione (max 10 pagine!) che illustri:
 - a. principali scelte progettuali
 - b. eventuali moduli particolarmente significativi (progettazione, implementazione, caratteristiche)
 - c. caratteristiche del modulo utilizzato per il test
 - d. risultati che comprovano il funzionamento del modulo (per esempio, finestra gtkwave con l'output del programma di test)
 - e. manuale d'uso, ovvero tutta l'informazione necessaria per utilizzare il modulo e riprodurre i risultati presentati nella relazione
5. Iscrizione all'appello secondo i termini e le modalità stabilite per l'appello standard
6. Invio della relazione (in PDF) e di un archivio con tutti i file relativi al progetto per email al docente entro il giorno in cui si svolge lo scritto dell'appello. Il messaggio dovrà avere il Subject impostato come "AEa 2018-19: Consegna progetto Verilog" (per favore utilizzate esattamente questo testo, altrimenti potrei non trovare correttamente i vostri messaggi) e nel testo del messaggio dovranno essere indicati Nome, Cognome e Matricola.