

Introduzione all'uso combinato di Gezel con un simulatore VHDL

Esercitazione 01 di Sistemi dedicati

Docente: Giuseppe Scollo

Università di Catania
Dipartimento di Matematica e Informatica
Corso di Laurea Magistrale in Informatica, AA 2016-17

1 di 8

Indice

1. Introduzione all'uso combinato di Gezel con un simulatore VHDL
2. argomenti dell'esercitazione
3. modelli di hardware in Gezel
4. esempio: traiettorie di Collatz
5. traduzione automatica in VHDL e simulazione
6. note operative
7. riferimenti

2 di 8

in questa esercitazione si trattano:

- modelli di hardware in Gezel: caratteristiche, limiti, usi pratici
- installazione del software Gezel
- traduzione di modelli Gezel in modelli VHDL
- installazione del software Quartus
- compilazione, analisi e tuning di modelli VHDL in Quartus
- creazione di forme d'onda di test in Quartus
- simulazione funzionale con il ModelSim di Quartus

modelli di hardware in Gezel

circuiti digitali sincroni a singolo clock, composti da interconnessione di:

- reti combinatorie
- flip-flop

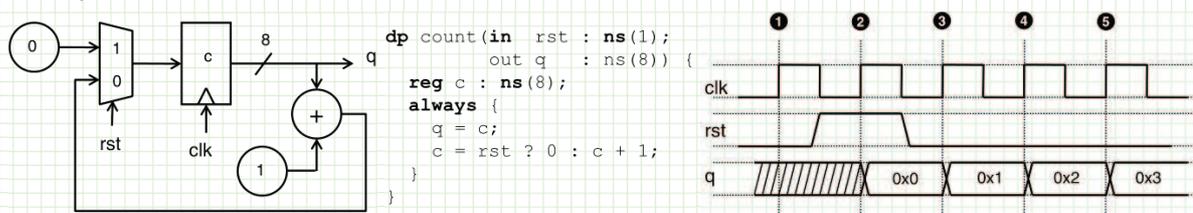
dunque anche componenti di largo uso quali: registri, addizionatori, moltiplicatori ecc.

livello di astrazione: cicli di clock, modelli RTL

non si possono modellare: circuiti asincroni, con latch, con clock multifase ecc.

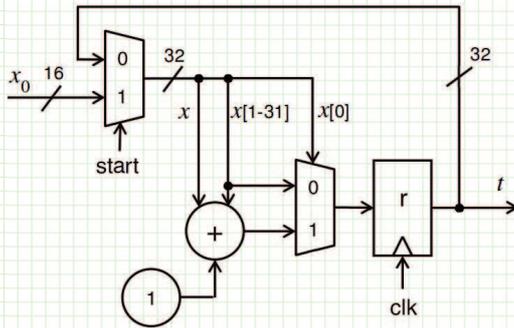
i modelli RTL sono comunque sufficienti in gran parte di casi pratici, per descrivere realizzazioni hardware di algoritmi

esempio:



Schaumont, Fig. 1.1 - Modelli e comportamento di un componente hardware

riconsideriamo l'esempio visto nella prima lezione:



```

dp collatz ( in start : ns(1) ; in x0 : ns(16) ;
            out t ns(32)) {
    reg r : ns(32) ;
    sig x : ns(32) ;
    always {
        t = r ;
        x = start ? x0 : r ;
        r = x[0] ? x + (x >> 1) + 1 : x >> 1 ;
    }
}
    
```

che uso possiamo farne?

e.g., come possiamo visualizzare il suo comportamento?

gli strumenti di sviluppo industriali richiedono descrizioni in linguaggi standard, quali VHDL o Verilog...

traduzione automatica in VHDL e simulazione

il generatore di codice della piattaforma Gezel produce una traduzione in VHDL *sintetizzabile*

esperienza di laboratorio:

1. installare Gezel
2. creare il file sorgente collatz.fdl contenente la descrizione Gezel dell'esempio
3. eseguire da linea di comando: fdlvhd collatz.fdl
4. installare Quartus 13.1 (Web edition) prodotto da Altera e lanciarlo, quindi in tale sistema:
5. creare un nuovo progetto di nome collatz
6. copiare i file .vhd prodotti al passo 3 nella cartella del progetto
7. assegnare i file suddetti al progetto e compilare
8. controllare eventuali messaggi di errore o warning
9. impostare il clock a una frequenza che assicuri un valore positivo per lo slack di caso peggiore
10. creare forme d'onda di test per il circuito, con un input di clock corrispondente alla frequenza del passo precedente e il valore 27 per l'inizio della traiettoria
11. lanciare l'esecuzione della simulazione funzionale
12. ripetere la simulazione per differenti valori di inizio della traiettoria

alcune note per l'esecuzione dell'esperienza su Ubuntu 16.04:

Le note che seguono, in inglese, indicano alcuni accorgimenti pratici per superare piccoli problemi che altrimenti possono presentarsi nell'esecuzione dell'esperienza di laboratorio

- è disponibile il download dell'archivio ZIP di tutte le note

1. note per l'installazione di Gezel
2. note per l'installazione e lancio di Quartus 13.1 (Web Edition) su Ubuntu 16.04
3. note sugli assegnamenti a un progetto Quartus
4. note sulla regolazione del clock con Quartus TimeQuest Analysis
5. note sull'uso di Quartus ModelSim

riferimenti

letture raccomandate:

Schaumont (2012) Cap. 1, Sez. 1.1.1; App. A.1

Quartus II Introduction Using VHDL Designs - For Quartus II 13.1; Altera University Program

Using TimeQuest Timing Analyzer - For Quartus II 13.1, Sez. 1-2, 4; Altera University Program

per ulteriore consultazione:

Schaumont (2012) App. A.2

altri materiali utili per l'esperienza di laboratorio proposta:

Gezel installation manual

Quartus II 13.1 Web Edition download

Altera University Program Installer