

Progetto di un coprocessore multicore mappato in memoria

Lezione 12 di Sistemi dedicati

Docente: Giuseppe Scollo

Università di Catania
Dipartimento di Matematica e Informatica
Corso di Laurea Magistrale in Informatica, AA 2017-18

Indice

1. Progetto di un coprocessore multicore mappato in memoria
2. argomenti della lezione
3. spazio di progetto
4. evoluzione del codesign di calcolo del delay
5. struttura di un coprocessore multicore
6. vincoli all'interfaccia hardware
7. funzioni sulle traiettorie di Collatz
8. idee per il codesign di estensioni funzionali
9. riferimenti

di che si tratta:

- evoluzione del codesign per calcoli sulle traiettorie di Collatz
- evoluzione del calcolo del delay:
 - miglioramento delle prestazioni
 - estensione del dominio
- rapida panoramica di funzioni sulle traiettorie di Collatz
- alcune idee per estensioni funzionali del codesign

spazio di progetto

possibili evoluzioni del codesign realizzato nell'esercitazione precedente si possono prefigurare lungo due direzioni ortogonali di sviluppo:

- miglioramento delle prestazioni
- estensioni funzionali

un esempio di combinazione ortogonale delle due direzioni è dato dai seguenti obiettivi per un primo progetto, affrontato nella prossima esercitazione:

- un sicuro miglioramento delle prestazioni può ottenersi replicando in parallelo l'unità hardware dedicata al calcolo del delay
- d'altra parte, un'estensione funzionale minima di sicuro interesse è l'ampliamento della larghezza dell'input, per l'applicabilità del calcolo del delay a un più ampio dominio di traiettorie

estensioni funzionali più significative sono individuabili dalla considerazione di funzioni, definite sulle traiettorie di Collatz, diverse dal delay ma per il calcolo delle quali è comunque necessaria la generazione delle traiettorie

qui la sfida sta nel progetto di unità hardware di calcolo multifunzione e di un'interfaccia HW/SW più complessa, atta alla selezione delle funzioni di interesse e al relativo trasferimento di dati

una prima alternativa di progetto da considerare per la replica delle unità hardware di calcolo del delay è:

- repliche del componente di calcolo quali istanze distinte sul bus Avalon, o
- costruzione di un componente più complesso, che incorpori più copie del singolo componente di calcolo

la seconda opzione è preferibile in vista di possibili ulteriori estensioni che richiedessero accesso delle diverse istanze a dati condivisi, e.g. definiti come parametri di configurazione
infatti, in tal caso si evitano così transazioni sul bus a tal fine, inoltre si realizza una gerarchia del controllo che assegna al coprocessore funzioni di controllo locale

altre decisioni di progetto riguardano il numero delle istanze parallele di calcolo, dette core nel seguito, e la dimensione dei dati di I/O del coprocessore

- tenendo conto che il processore Nios II può trasferire sul bus in una singola transazione non più di 32 bit, e che può essere utile disporre di un registro di stato nel coprocessore con un bit per ciascun core, conviene limitare a 32 il numero dei core
- dai dati disponibili nel sito web sul problema 3x+1 curato da Eric Roosendaal, e.g. nella tabella dei Class Records, emerge una dimensione minima di 64 bit per dati di input interessanti, mentre 16 bit bastano per l'output del delay

struttura di un coprocessore multicore

l'estensione a 64 bit dell'input del singolo core è ottenuta facilmente con ovvia modifica del sorgente Gezel dall'esercitazione precedente e con la stessa correzione all'output VHDL del traduttore fdlvhd

la descrizione strutturale del coprocessore multicore in VHDL è facilitata dal costrutto iterativo **for** <identifier> **in** <range> **generate**, il cui uso è esemplificato in

Zwolinski, 4.5.2, che nel nostro caso permette la generazione delle $2^n = 32$ istanze del core, con $n = 5$

occorre poi dotare il coprocessore multicore di circuiti per il corretto smistamento dei dati di I/O fra l'interfaccia e un core selezionato dal processore:

- un moltiplicatore a n bit di selezione e porte dati da 16 bit per l'output del delay dal core
- un demoltiplicatore a n bit di selezione e porte dati da 64 bit per l'input del dato d'inizio al core, nonché uno analogo ma con porte dati da 1 bit per l'input del segnale di start

la descrizione in VHDL della moltiplicazione è semplice se si collocano gli output dei core in un vettore da $2^n \times 16$ bit, basta infatti usare un operatore di selezione sul vettore

la descrizione in VHDL della demoltiplicazione, più complessa, è fattibile usando un operatore di scorrimento logico, come esemplificato per un decodificatore generico in Zwolinski, 4.2.3

lo scambio di segnali alle porte di I/O del coprocessore multicore va adattato ai segnali disponibili all'interfaccia Avalon-MM, tenendo conto di alcuni vincoli su questi, quali:

- i segnali di trasferimento dati, writedata e readdata, devono avere la stessa larghezza
- il segnale address individua un registro di I/O nello spazio di memoria assegnato al coprocessore, a partire da 0 e con indirizzamento di parola per default

poiché il processore Nios II può trasferire in una singola transazione non più di 32 bit, si conviene che questa sia la larghezza di parola del coprocessore all'interfaccia Avalon, ovvero la larghezza dei segnali writedata e readdata

da ciò consegue che l'output del delay va esteso con zeri, mentre l'input del dato d'inizio traiettoria va acquisito in due cicli di bus

lo spazio degli indirizzi di registro del coprocessore è dunque l'intervallo $[0, 3 \times 2^n]$, tenendo conto di un indirizzo per il registro di stato, dunque address è largo $n+2$ bit

un'opportuna gestione degli indirizzi di registro permette una rapida identificazione del core (o del registro di stato) dal valore dell'input address, per esempio:

address[n,1] se address[n+1] = 0

address[n-1,0] se address[n+1,n] = 10

registro di stato se address[n+1,n] = 11

funzioni sulle traiettorie di Collatz

il già indicato sito web sul problema $3x+1$ è una fonte preziosa di ispirazione per il codesign di estensioni funzionali del sistema considerato fin qui; ecco alcune funzioni sulle traiettorie di Collatz (definite sui valori d'inizio, che determinano le traiettorie):

- $\text{Glide}(x_0)$: minimo k per il quale $x_k < x_0$
- $\text{Mx}(x_0)$: valore di picco nella traiettoria da x_0
- $\text{Completeness}(x_0)$: rapporto fra il numero di valori dispari e il numero di valori pari nella traiettoria da x_0 a $x_{\text{delay}(x_0)-1}$

alcune traiettorie sono più interessanti di altre perché stabiliscono primati (come in gare agonistiche); sono definite in tal senso le seguenti proprietà: x_0 è un

- Class Record se è il minimo x che abbia $\text{delay}(x) = \text{delay}(x_0)$
- Delay Record se ogni $x < x_0$ ha $\text{delay}(x)$ minore di quello di x_0
- Path Record se ogni $x < x_0$ ha picco $\text{Mx}(x)$ minore di quello di x_0
- Completeness Record se ogni $x < x_0$ ha $\text{Completeness}(x)$ minore di quella di x_0

seguono alcune considerazioni utili al progetto di estensioni funzionali del codesign finalizzate alla ricerca di tali rarità

estensioni funzionali del core di calcolo del delay, per fornire anche il calcolo delle tre funzioni introdotte nella pagina precedente, sono ragionevolmente semplici

per il calcolo della completeness, il circuito dovrebbe solo fornire il numero di valori dispari nella traiettoria (1 finale escluso); il numero di valori pari si ha per differenza dal delay ed è meglio calcolare in software il loro rapporto (un numero in virgola mobile)

la selezione delle funzioni desiderate va considerata in due contesti diversi:

- quale parametro di configurazione del componente hardware (prima della sua sintesi): ciò è utile per risparmiare area quando non si intende utilizzare tutte le funzioni disponibili
- quale input del componente hardware per le operazioni di lettura, per la selezione di risultati desiderati fra quelli calcolati

è utile alla ricerca dei Record introdotti prima lo studio di loro proprietà, ottenute da risultati teorici, che escludono a priori intere classi di valori di x_0 dallo spazio di ricerca

per esempio, 5 è l'unico Class Record nella classe di congruenza $5 \pmod{8}$; questa situazione, spiegata dalla cosiddetta *coalescenza* di traiettorie, si generalizza alle classi di congruenza $(2^{k-2} + (k \bmod 2)(2^{k-1} - 1)) \pmod{2^k}$ per $k > 2$

tecniche di ottimizzazione di algoritmi di ricerca di Record sono reperibili in bibliografia

riferimenti

Zwolinski, Ch. 4, Sect. 4.2.3, 4.5.2

E. Roosendaal, *On the $3x+1$ problem*, www.ericr.nl/wondrous

E. Roosendaal, *Technical details*, www.ericr.nl/wondrous/techpage.html

J. Lagarias, *The $3x+1$ problem and its generalizations*, *Amer. Math. Monthly* 92 (1985) 3-23

G.T. Leavens and M. Vermeulen, *$3x+1$ search programs*, *Computers Math. Applic.* 24:11 (1992) 79-99

G. Scollo, *Looking for Class Records in the $3x+1$ Problem by means of the COMETA Grid Infrastructure*, in: R. Barbera (Ed.) *Proc. Symp. "GRID Open Days at the University of Palermo"*, Palermo, 6-7 Dec. 2007; Consorzio COMETA (2008) pp. 255-263