

Interfacce di microprocessore

Lezione 10 di Sistemi dedicati

Docente: Giuseppe Scollo

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

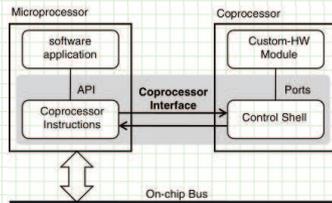
Indice

1. Interfacce di microprocessore
2. argomenti della lezione
3. interfacce mappate in memoria
4. interfacce di coprocessore
5. interfacce di istruzioni custom
6. flusso di progetto di ASIP
7. esempio: interfaccia di istruzione custom Nios-II
8. banchi di registri per istruzioni custom Nios-II
9. riferimenti

di che si tratta:

- > interfacce mappate in memoria
 - > registro mappato in memoria
 - > mailbox con handshake
 - > code FIFO
 - > protocolli di handshake
 - > memoria condivisa
- > interfacce di coprocessore
- > flusso di progettazione di ASIP
- > interfacce di istruzioni custom
 - esempio: l'interfaccia di istruzioni custom Nios-II

seminario di Grazia Pagano (PDF)



Schaumont, Figure 11.7 - Coprocessor interface

interfacce di coprocessore

quando occorre un throughput di dati elevato tra il software e l'hardware custom, un'interfaccia dedicata supera le prestazioni di quelle mappate in memoria un'interfaccia di coprocessore non usa il bus su chip bensì una porta dedicata sul processore, pilotata da istruzioni di coprocessore

sia l'insieme di istruzioni di coprocessore che la specifica interfaccia di coprocessore dipendono dal tipo di processore; non tutti i processori hanno un'interfaccia di coprocessore

la decisione di usare una specifica interfaccia di coprocessore lega il modulo hardware custom a un particolare processore, dunque limita la riusabilità del modulo ai sistemi che usano lo stesso processore

vantaggi principali di un'interfaccia di coprocessore rispetto al bus su chip:

- throughput più alto: perché non vincolato dalla larghezza del bus su chip, né dal meccanismo di trasferimento load/store
- latenza fissa: un bus di coprocessore è una connessione dedicata punto-punto con temporizzazione stabile e predicibile

interfacce di istruzioni custom

l'integrazione di hardware e software può essere notevolmente accelerata come segue:

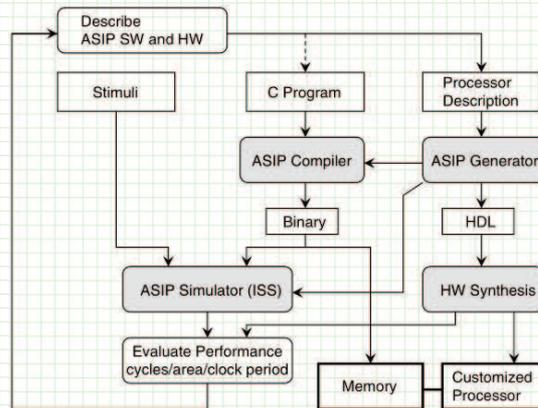
1. riservare una parte dei codici operativi di un microprocessore per nuove istruzioni
2. integrare i moduli hardware custom direttamente nella microarchitettura del microprocessore
3. controllare i moduli hardware custom usando nuove istruzioni derivate dai codici operativi riservati

il risultato di un tale progetto è un *Application-Specific Instruction-set Processor (ASIP)* mentre l'insieme di istruzioni di coprocessore è parte del microprocessore, il progetto di quello di un ASIP è definito dall'applicazione

il progetto di ASIP automatizza gli aspetti più difficili del codesign HW/SW:

- il meccanismo di prelievo e smistamento delle istruzioni nel microprocessore assicura che hardware custom e software restino sincronizzati
- il progetto di ASIP procede in modo incrementale, il che evita di dover fare modifiche drastiche all'architettura di sistema nell'esplorazione di opzioni diverse

flusso di progetto di ASIP

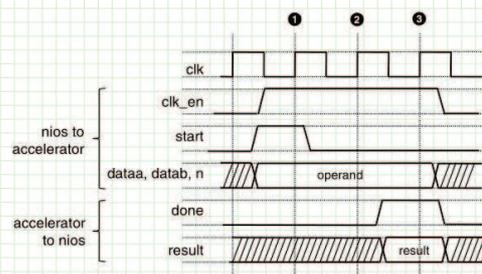


Schaumont, Figure 11.12 - ASIP design flow

il progetto di un ASIP sequenziale generalmente non produce prestazioni migliori del progetto di SOC basato su moduli hardware custom, tuttavia ha minor propensione all'errore
 progressi significativi sono stati fatti nell'automazione del flusso di progetto di ASIP:
 per tutti i blocchi ombreggiati in figura sono reperibili strumenti commerciali

esempio: interfaccia di istruzione custom Nios-II

il processore softcore Nios-II ha un'interfaccia di coprocessore su cui si possono definire istruzioni custom e collegare moduli hardware



Schaumont, Figure 11.15 - Nios-II custom-instruction interface timing

l'interfaccia supporta l'esecuzione a durata variabile di istruzioni custom mediante doppio handshake
 l'input clk_en si usa per disabilitare l'hardware custom quando l'istruzione è inattiva

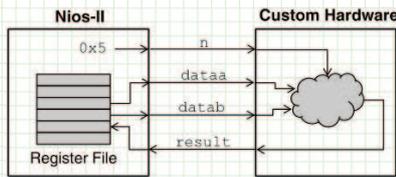
l'istruzione software custom ordina la sua esecuzione, e.g.:

custom 0x5, r2, r3, r5

assegna il valore 0x5 a n e associa le porte dataa, datab, result ai registri r2, r3, r5, nell'ordine;

uso di n: moltiplicazione di diverse istruzioni custom nel modulo hardware

è supportato anche l'uso di un banco di registri locale nel modulo hardware custom



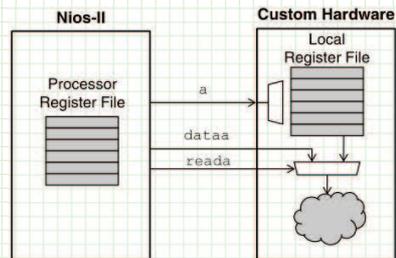
Schaumont, Figure 11.16a - Nios-II custom-instruction integration with processor register file

un'istruzione custom può avere operandi nei due banchi di registri: registri con prefisso r sono collocati nel processore, mentre registri con prefisso c sono collocati nel modulo hardware

è permesso l'uso di entrambi in una stessa istruzione, e.g.

custom 0x5, c2, c3, r5

la figura 11.16b illustra solo il caso del primo operando: il segnale di controllo reada seleziona il banco di registri, del processore o locale



Schaumont, Figure 11.16b - Nios-II custom-instruction integration with local register file

➤ nel primo caso, l'operando è fornito attraverso la porta dataa, associata a un registro del processore

➤ nel secondo caso, l'input a seleziona il registro locale da usare quale operando

riferimenti

letture raccomandate:

Schaumont, Ch. 11, Sect. 11.1.1-11.1.5, 11.2.0, 11.3.0-11.3.1, 11.3.3

per ulteriore consultazione:

Schaumont, Ch. 11, Sect. 11.1.6, 11.2.1-11.2.2, 11.3.2, 11.3.4