

VERBALE DELLA RIUNIONE DEL GRUPPO DRES D

Minuta in Pillole – 2 Febbraio 2006

L'incontro si è svolto presso la sala conferenze del DEI del Politecnico di Milano. I presenti alla riunione sono stati 10.

SOMMARIO DELLA RIUNIONE

- Presentazione di Matteo Giani: "DRES D: Current issues about Specification Partitioning and Scheduling, aka BOH"
- Presentazione di Matteo Murgida e Alessandro Panella: "IP-Core Generator"
- Presentazione di Alessio Montone e Antonio Piazzì: "Implementazione di un bridge tra OPB e Wishbone"

RESOCONTO DELLA RIUNIONE

- **Presentazione di Matteo Giani: "DRES D: Current issues about Specification Partitioning and Scheduling, aka BOH"**
Definizione di sistema riconfigurabile: sistema che è in grado di permettere il cambiamento della configurazione hardware di alcune sue componenti, mentre sta eseguendo l'applicazione per la quale è stato progettato. Tale caratteristica permette di adattare il sistema alle esigenze dell'applicazione durante la sua evoluzione. La riconfigurazione può essere totale o parziale. L'approccio seguito dal gruppo DRES D si concentra sulla riconfigurabilità parziale interna e sceglie quindi la specifica del problema più complessa, ma non trascura comunque gli altri tipi di riconfigurabilità. Si è cercato di individuare le variabili e lo spazio delle soluzioni che la riconfigurabilità introduce in fase di progettazione e sintesi dell'architettura. Sono state considerate l'area occupata dall'architettura sintetizzata e le prestazioni temporali. Lo spazio delle soluzioni è compreso tra area massima a schedule ottimo da una parte e area minima a schedule pessimo dall'altra. L'area identificata è allocata staticamente in fase di sintesi e progettazione dell'architettura. Tuttavia la soluzione statica sul dispositivo potrebbe non essere fattibile perché l'area del dispositivo può essere minore dell'area necessaria per soddisfare i requisiti temporali del progettista. Infatti lo schedule ha una lunghezza massima ammissibile in quanto il progettista deve rispettare vincoli temporali per l'applicazione e alla lunghezza dello schedule corrisponde l'area necessaria per la sua implementazione. In questo scenario la riconfigurabilità dinamica può essere utile. Si vuole quindi allocare e deallocare area sul dispositivo mentre l'applicazione sta girando fornendo l'illusione di avere a disposizione un dispositivo più

“grosso” di quello che si ha realmente. Si rende così disponibile all'applicazione più area di quella reale. Questo comporta il fatto di dover tenere conto dei tempi di riconfigurazione nello schedule.

E' necessario quindi individuare i moduli riconfigurabili sul dispositivo. Inoltre è necessario identificare come e quando effettuare il piazzamento dei moduli sul dispositivo. Si parte dalle specifiche, che sono espresse sotto forma di codice. Su questo codice iniziale viene effettuata una traduzione intermedia e da qui viene effettuata un'analisi sul partizionamento. All'interno del grafo si cerca di individuare strutture ricorrenti, chiamate isomorfismi. Due grafi si dicono isomorfi quando è possibile associare i nodi dell'uno ai nodi dell'altro in modo biunivoco. Identificare strutture ricorrenti permette di risparmiare area sul dispositivo e ridurre il tempo di esecuzione. I principali problemi da affrontare sono l'identificazione di sottografi isomorfi all'interno di un grafo generico e la definizione di una buona dimensione dei cluster individuati. Per l'identificazione dei cluster, risulta necessario definire apposite metriche.

Si cerca quindi di suddividere la specifica iniziale in sottografi che abbiano la proprietà di essere isomorfi tra loro. Ogni sottografo rappresenta un'unità riconfigurabile. Si produce un grafo, chiamato task graph, i cui nodi sono raggruppamenti dei nodi della specifica di partenza. Ogni nodo è identificato da un'etichetta che rappresenta il tipo del cluster del grafo di partenza e quindi la funzionalità implementata. Nella seconda fase dell'approccio si decide l'ordine in cui eseguire i task identificati e il loro posizionamento sul dispositivo. I principali problemi da affrontare riguardano la modellazione dell'area del dispositivo e i tempi di riconfigurazione. Risulta inoltre necessario studiare il caso in cui dalle specifiche non risulti possibile individuare “grossi” cluster isomorfi.

- **Presentazione di Matteo Murgida e Alessandro Panella: "IP-Core Generator"**

In questa presentazione è stato descritto lo strumento software chiamato IP-Core Generator, la cui funzione è la generazione automatica dell'interfaccia degli IP-Core, ed è stato illustrato il modo in cui descrivere un core interfacciabile automaticamente tramite l'IP-Core Generator. Questo strumento è utile nella progettazione hardware per risolvere i problemi di interfacciamento degli IP-Core. L'obiettivo è fare in modo che nei progetti hardware si concentri l'attenzione sulla funzionalità da implementare piuttosto che sull'interfacciamento. Inoltre si favorisce il riuso dei core in diverse architetture.

L'IP-Core Generator si basa su due fasi, svolte a loro volta da due componenti differnti del sistema: il reader ed il writer. Il reader si occupa del riconoscimento di costrutti tipici nel codice VHDL e della successiva trasformazione e crea una lista di segnali usata in seguito. Il writer utilizza la lista di segnali generata dal reader per creare lo stub, che si occupa del processo di lettura e scrittura sul bus. In particolare lo stub deve indirizzare correttamente i segnali provenienti dal bus sulle varie porte del core e connettere correttamente i segnali in uscita dal core sulle linee del bus. Esso rappresenta una fase intermedia tra la creazione del core e la creazione dell'IP-Core. Quindi la struttura generale risulta essere la seguente: l'IP-Core istanzia l'interfaccia e lo stub, mentre lo stub a sua volta istanzia il core.

In questa ottica il core è l'unica parte creata dal progettista, mentre tutto il resto dell'IP-Core è generato dall'IP-Core Generator. Il core deve essere scritto in modo da essere interfacciabile automaticamente. Di conseguenza esso non deve implementare la funzione di decodifica degli indirizzi e deve avere porte standard oltre naturalmente

alle porte custom. Le porte standard si riferiscono ai segnali di clock, di reset e di interrupt, mentre le porte custom sono le porte dipendenti solo dalla funzionalità del core e non dall'esterno.

Uno sviluppo futuro di questo lavoro riguarda l'estensione delle funzionalità dell'IP-Core Generator in modo da poter gestire più tipi di interfacce ed fornire una maggiore flessibilità di descrizione dei core da parte del designer.

- **Presentazione di Alessio Montone e Antonio Piazzì: "Implementazione di un bridge tra OPB e Wishbone"**

In questa presentazione è stata descritta l'implementazione di un bridge tra il bus OPB e il bus Wishbone. L'OPB è un bus sincrono e arbitrato sviluppato da IBM. Sia il bus indirizzi sia il bus dati sono a 32 bit. Il Wishbone può essere implementato come bus o come connessione punto a punto ed è sincrono. La presenza dell'arbitraggio dipende dall'implementazione. Il bus dati è a 32 bit, mentre il bus indirizzi non ha limitazioni.

La progettazione del bridge è stata scomposta considerando separatamente la comunicazione da OPB a Wishbone e da Wishbone a OPB. Sono stati quindi sviluppati due core in parallelo. E' risultato necessario avere un'interfaccia master sul bus destinazione e un'interfaccia slave sul bus sorgente. Nell'interfaccia slave è implementata la funzione di decodifica.

Durante la fase di implementazione sono stati incontrati problemi a livello di sintesi. I problemi riguardavano principalmente la condivisione di segnali da parte di processi diversi; problema noto come "double driven". Una volta risolti questi problemi si è così ottenuto un codice sintatticamente corretto e sintetizzabile. Sono state effettuate diverse simulazioni tramite ModelSim creando opportuni file di testbench.

Sono stati infine presentati i dati relativi alla sintesi dell'IP-Core. Il bridge tra OPB e Wishbone occupa circa 1/5 dello spazio occupato dal bridge tra OPB e PLB.

Lo sviluppo futuro di questo lavoro riguarda l'importazione del bridge in EDK.

PROPOSTE DI PROGETTO

In questa sezione verranno inserite le vari proposte di progetto che si formalizzeranno all'interno del gruppo di ricerca legato a DRESO.

- Con riferimento alla presentazione di Matteo Giani, si propongono lavori relativi a scheduling e partizionamento di grafi non isomorfi, per maggiori informazioni si contatti direttamente Marco.
- E' in arrivo la board di Atmel con una FPSLIC. Chiunque fosse interessato ad un progetto su questa nuova architettura è pregato di contattare direttamente Marco.

URGENZE

In questa sezione verranno elencate le urgenze legate ai vari lavori che gravitano intorno a DRESO, di seguito alcuni esempi.

- Si stanno cercando dei volontari per la definizione del logo di DRES D, chiunque fosse interessato è pregato di contattare Chiara Sandionigi: rsandio@tin.it
- Si stanno cercando dei volontari per la definizione dei template per le presentazioni ed i documenti prodotti all'interno di DRES D, chiunque fosse interessato è pregato di contattare Chiara Sandionigi: rsandio@tin.it
- **Wanted:** Si cercano persone che affianchino Chiara e Marco nella realizzazione di questo documento. Chiunque fosse interessato, è pregato di mandare una mail a: marco.santambrogio@polimi.it

NEWS

- Tre volontari si sono proposti per lo sviluppo del sito DRES D. Si accettano anche altre collaborazioni.
- Nasce DRES D News, periodico informativo del team DRES D con cadenza bimensile, la prima uscita di DRES D News è prevista per i primi di Maggio. Si cercano volontari per lo sviluppo di questo periodico. Un grazie particolare alla prof.ssa Bolchini che ci ha segnalato uno strumento per l'editing sia dei verbali che di DRES D News.

PROSSIMI INCONTRI

- **Giovedì 16 marzo 2006**

LINK CONSIGLIATI

Chiunque fosse interessato a segnare dei link è pregato di mandare una mail a Chiara Sandionigi: rsandio@tin.it

- FPSLIC: Al seguente link potrete recuperare il materiale riguardante la nuova board che sarà a breve disponibile in Micro: <http://www.atmel.com/products/FPSLIC/>
- Reconfigurable Architectures Workshop, RAW: una delle conferenze più interessanti per il campo della riconfigurabilità dinamica. Dal sito indicato potete accedere anche alle passate edizioni della conferenza e recuperare un pò di articoli che possono essere interessanti: <http://www.ece.lsu.edu/vaidy/raw06/>
- VLSI-SoC: Visto che presenteremo un paio di lavori che si stanno finendo di realizzare in Micro e su cui anche alcuni di voi stanno lavorando, ecco a voi il sito di una seconda conferenza: <http://tima.imag.fr/conferences/VLSI-SoC06/>

Documento interno al gruppo DRES D, Dipartimento di Elettronica e Informazione, Politecnico di Milano.

Ci scusiamo per eventuali imprecisioni legate al fatto che questo documento ha lo scopo di essere un breve verbale informativo relativo a una riunione del gruppo DRES D. Per questo motivo non saranno forniti riferimenti bibliografici.