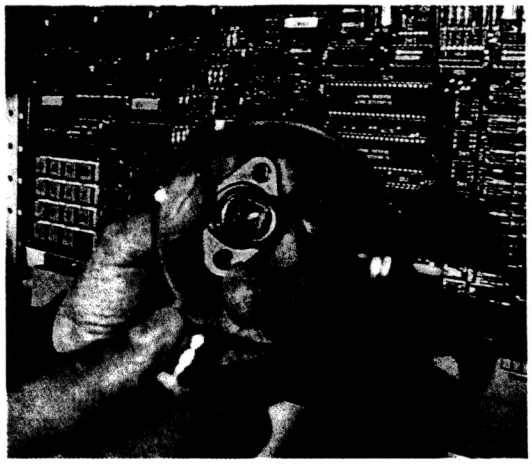
Progetto di Sistemi Digitali Dedicati
Embedded Systems

di Giovanni De Micheli, Maria Giovanna Sami
e Donatella Sciuto

Mentre oggigiorno il mercato dei sistemi digitali dedicati si sta espandendo molto velocemente, il supporto CAD per il loro progetto è ancora limitato. Ciononostante, gli strumenti CAD rappresenteranno la chiave di volta per il progetto efficiente e a basso costo di sistemi dedicati. Per questo motivo, vi è un altissimo interesse allo sviluppo di idee di ricerca in questo settore. Un corso in programma per giugno rappresenterà una delle occasioni migliori per apprendere i fondamenti e le novità in questo campo.



Il progetto concorrente di hardware e software (hardware/software co-design) è uno dei grandi temi di attualità in elettronica ed informatica [5]. Gran parte dei sistemi di elaborazione che richiedono tale metodologia di progetto rientrano nella classe dei cosiddetti sistemi digitali dedicati (embedded system), quali per esempio i sistemi per controllo di processi industriali. In molti casi, tali sistemi sono caratterizzati da forti vincoli di tempo reale e richiedono quindi un progetto che garantisca le massime prestazioni nell'esecuzione dei programmi applicativi e comunque una risposta entro termini temporali stretti agli stimoli forniti dai sistemi interagenti. A tale scopo, spesso si utilizzano supporti architetturali specifici per velocizzare le applicazioni; ciò comporta un progetto dell'hardware che tenga in conto anche i requisiti software. Vale la pena sottolineare come, in questo settore, si renda indispensabile progettare simultaneamente le componenti hardware e software; il software deve essere messo a punto e ingegnerizzato accuratamente in relazione all'applicazione e alle interazioni con le componenti hardware. Poiché l'utente finale del sistema non modifica (di solito) il software del sistema stesso, la qualità globale del sistema (prestazioni, consumo di potenza, affidabilità) diventa il parametro critico per il progettista, ben più importante di parametri software tradizionali quali velocità di compilazione e facilità di programmazione. Il mercato industriale delle applicazione digitali dedicate è in fortissima espansione. È sufficiente pensare a settori quali l'elettronica di bordo per automobili,

Prof. Giovanni De Micheli, Stanford University; prof.ssa Maria Giovanna Sami e prof.ssa Donatella Sciuto, Politecnico di Milano

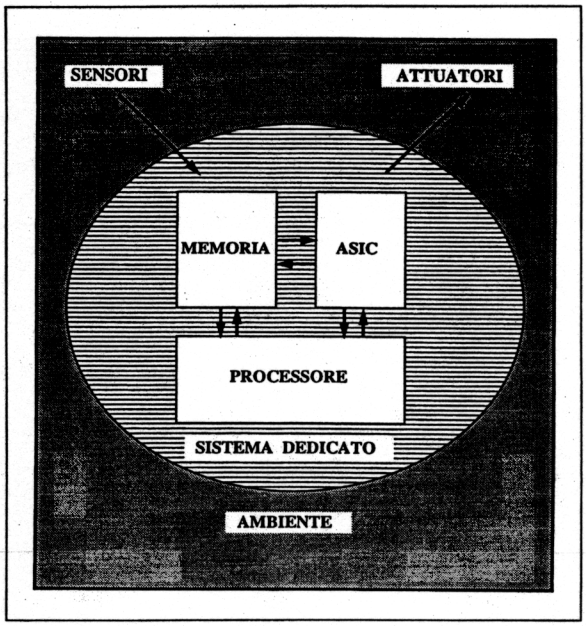


Figura 1 - Componenti essenziali di un sistema digitale dedicato

aeroplani ed imbarcazioni, il controllo dei robot per l'industria manifatturiera, i sistemi di telecomunicazioni e quelli per la gestione e difesa del territorio. Anche il settore dell'elettronica civile ha beneficiato dello sviluppo dei sistemi dedicati: infatti in molte applicazioni (basti pensare a dispositivi portatili, quali i telefoni cellulari) intervengono in modo massiccio sottosistemi dedicati con prestazioni e requisiti più spinti di quelli ottenibili da un convenzionale microprocessore. In questi casi diventa importante anche ridurre il consumo di potenza e permettere una parziale riprogrammabilità del sistema, che verrà compiuta dal produttore per differenziare le caratteristiche visibili al consumatore, mantenendo fissa la struttura hardware e quindi ammortizzando il suo costo su un vasto volume di produzione.

Nel seguito di questo articolo si definiranno concisamente i sistemi cui si applica la metodologia di progetto concorrente hardware e software; si citeranno quindi i principali stili di progetto per tali sistemi, per passare poi a discutere le tecniche ed il supporto CAD necessario per giungere al loro progetto. Sulla tematica del progetto concorrente hardware e software, si terrà nel giugno prossimo a Tremezzo (Como) un convegno internazionale, sotto l'egida del programma scientifico/culturale dell'Alleanza Atlantica. In questo corso, esperti internazionali presenteranno lezioni sulle tematiche più rilevanti del progetto concorrente di hardware e software, con particolare enfasi sui sistemi dedicati e sulla elaborazione digitale dei segnali (DSP).

Sistemi digitali dedicati

I sistemi digitali dedicati presi in esame in questo articolo sono caratterizzati dal-l'avere una componente hardware ed una software. Nel caso più semplice, le applicazioni richiedono solo l'uso di un microcontrollore o di un microprocessore che esegua un programma dedicato memorizzato in ROM; l'interfaccia con il mondo esterno è gestita banalmente attraverso porte di ingresso/uscita collegate ad opportuni trasduttori. Esempi più complessi possono richiedere l'uso di diverse unità di elaborazione collegate in rete, dotate del relativo software dedicato, e di attuatori e sensori più sofisticati; la gestione di eventi esterni di particolare criticità può portare allo sviluppo di dispositivi specifici per il loro trattamento, che interagiscano poi con il resto del sistema programmabile.

Si deve in tal caso ricorrere al progetto di circuiti integrati specifici relativi all'applicazione (ASIC, Application Specific Integrated Circuits). Il progetto di interfacce per unità periferiche (ad esempio sensori ed attuatori) secondo un protocollo desiderato è una parte importante del progetto. Di solito, il progettista usa alcune periferiche standard e pertanto deve aderire a modelli standard di interfacciamento. I controllori (drivers) delle periferiche possono essere realizzati in hardware o da sottoprogrammi software. La scelta dipende dal numero disponibile di porte I/O dell'unità principale hardware e dai vincoli temporali sulla comunicazione con le periferiche.

La Figura 1 mostra schematicamente le componenti principali di un sistema dedicato. A grandi linee, i sistemi digitali dedicati svolgono una funzione di controllo ed una di elaborazione dell'informazione digitale. Per esempio, un sistema di accensione elettronica per autovetture prevede sensori per calibrare la miscelazione di aria e benzina, elaborazione di segnali provenienti dal motore e dal sistema di scarico, ed il sistema per controllare il momento esatto dell'iniezione del carburante e dell'accensione delle candele [9]. Vi sono dunque sia un aspetto controllistico sia uno elaborativo, che contribuiscono a requisiti differenti per la parte hardware e software.

I sistemi dedicati sono spesso classificati come sistemi reattivi (reactive system), in quanto sono progettati per reagire a stimoli d'ambiente eseguendo specifiche funzioni [3, 4]. Spesso, almeno parte di queste funzioni devono essere eseguite entro ben precise finestre temporali. In questo caso, i sistemi dedicati rientrano nella famiglia dei sistemi in tempo reale (real-time system). A loro volta, i vincoli temporali sono classificati come tempo reale stretto (hard) o lasco (soft). Nel primo caso, il mancato soddisfacimento di un vincolo può avere effetti catastrofici, mentre nel secondo caso gli effetti sono soltanto indesiderati. Per esempio, un robot ambulante (per consegna di cartelle cliniche in ospedale) ha vincoli di tempo reale stretto che lo obbligano ad arrestarsi od a virare di fronte ad un ostacolo. Una stampante laser ha invece vincoli laschi sul tempo di stampa di ciascuna pagina.

I vincoli di tempo reale, assieme al costo del sistema finale, sono i parametri di guida nel progetto concorrente di hardware e software. Se da un lato una progettazione di stile essenzialmente software, basata integralmente sull'uso di un microprocessore e di periferiche standard e mirata solo alla ottimizzazione del software, consente il costo minimo, dall'altro lo spostamento del massimo numero di funzionalità in dispositivi hardware dedicati ottimizza il parallelismo di esecuzione e quindi minimizza il tempo di risposta ma richiede indubbiamente costi molto più elevati. Si può peraltro osservare che il bilancio dei costi deve essere valutato in modo opportuno rispetto al volume di produzione ed alle caratteristiche dell'applicazione.

Nasce quindi la necessità di spostare funzioni nella componente hardware piuttosto che in quella software così da soddisfare ai vincoli di tempo reale, pur mantenendosi entro i limiti di costo prefissato. Una volta garantiti i requisiti

imposti dal tempo reale, si possono inoltre prendere in considerazione ulteriori parametri quali consumo di potenza, affidabilità ecc.

Stili di progetto per sistemi dedicati

Esistono varie strutture di sistemi dedicati. Nel seguito, consideriamo le tipologie più importanti e descrittive delle problematiche di progetto dei sistemi dedicati. Come già accennato in precedenza, possono esistere soluzione puramente software, realizzazioni miste, ed implementazione puramente hardware.
Una realizzazione software, per esempio di un problema di controllo, prevede l'uso di un microcalcolatore (o microprocessore) con un programma software dedicato. Per ottimizzare le prestazioni di tale sistema, tradizionalmente si ricorre alla stesura dei programmi direttamente in linguaggio assemblatore, vicino all'architettura della macchina e che quindi consente di sfruttarne al meglio le caratteristiche.
Coprocessori possono essere aggiunti ad una piastra con microprocessore. Tali coprocessori possono essere standard o dedicati. Nel primo caso, un esempio è fornito da un coprocessore matematico; nel secondo un esempio può essere un chip specializzato ad una funzione, quale un dispostivo che effettui una particolare elaborazione su un segnale di ingresso mono o bidimensionale.
Nel settore delle telecomunicazioni, ha preso piede l'uso degli ASIP (Application-specific instruction-set processor), che sono microprocessori dedicati ed ottenuti modificando una architettura madre con l'aggiunta di registri, bus e collegamenti atti a supportare specifiche istruzioni.
Tali ASIP sono soluzioni di compromesso tra l'uso di un processore tradizionale e quello del progetto ex novo di un ASIC. Gli ASIP hanno prestazioni superiori (per applicazioni dedicate) a processori standard; richiedono però dei compilatori retargatable per lo sviluppo del software [11].
Soluzioni puramente hardware possono essere ottenute col progetto di uno o più ASIC che implementino l'intera soluzione del problema applicativo. Anche se questa scelta può portare a massime prestazioni, il suo costo di progettazione e manifattura è spesso notevole ed accettabile solo per alti volumi di vendita del prodotto. I diversi stili di realizzazione sono confrontati in Tabella. È da notare che la realizzazione delle unità fisiche dipende dalla tecnologia di fabbricazione. Le diverse unità possono corrispondere ad altrettanti distinti circuiti integrati su piastra, o ad un modulo multi-chip (MCM) in un unico involucro. O ancora, le diverse unità possono essere integrate su di un unico chip; nel settore dei microcontrollori si sta affermando oggi una filosofia di progetto che vede disponibile il microcontrollore-base come una macro-cella nucleo (core), che può essere combinata sulla stessa piastrina di silicio con altre celle logiche dedicate a specifiche funzioni e realizzate in tecnologia a celle-standard o pre-diffusa.
Un esempio di una cella nucleo, integrata assieme ad altre funzioni, è mostrato in Figura 2.
È da notare che la maggior parte di sistemi di controllo dedicati usano semplici processori a 4 od 8 bit per facilitarne la programmazione in linguaggio assemblatore. Purtroppo è ancora limitato l'uso di processori a 32 bit, con maggiore capacità di calcolo ed ambienti di programmazione più sofisticati, sebbene per molte applicazioni i relativi prezzi risultino concorrenziali ed il consumo di potenza possa essere ancora accettabile [14].

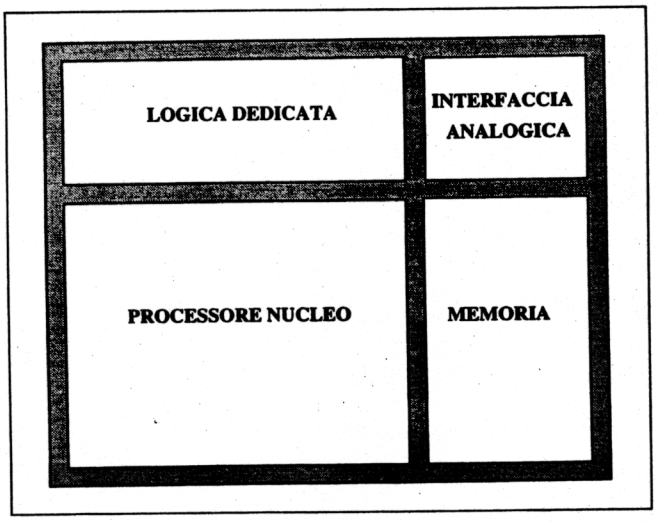
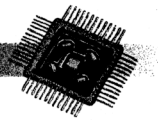


Figura 2 - Esempio di chip con processore nucleo

Supporto CAD per il progetto di sistemi dedicati

Nonostante il proliferare dei sistemi dedicati, la loro progettazione è tuttora spesso curata da esperti del settore con un uso limitato di strumenti CAD. La ricerca relativa alla definizione e al progetto di sistemi misti hardware e software è iniziata da pochi anni, stimolata in particolare dai successi ottenuti nella sintesi e verifica di sistemi hardware [6]; l'attività, anche se intensa, non ha ancora portato alla creazione di una metodologia e di un insieme completo e maturo di strumenti.
Prima che le metodologie di progetto concorrente hardware e software possano entrare a far parte del flusso di progetto industriale dovranno essere sviluppate alcune linee di ricerca fondamentali, che rendano disponibile al progettista un framework completo di specifica, sviluppo, valutazione, simulazione, prototipazione e testing di sistemi. Lo sviluppo di un tale framework può essere strutturato nel seguente insieme di fasi.
- *Sviluppo di un formalismo di specifica di sistema,* che permetta di descrivere ad alto livello i requisiti sia funzionali sia temporali. In questa fase, aspetti legati a tecnologie non elettroniche o informatiche ma caratterizzanti il prodotto finale dovrebbero idealmente essere prese in considerazione, quanto meno in termini di vincoli (si pensi a problemi quali il packaging, la dissipazione di potenza ecc.). Il formalismo deve essere di uso molto facile e molto vicino all'esperienza del progettista, in modo da non richiedere un tempo eccessivo di apprendimento e da ridurre la possibilità di errore; presumibilmente, un formalismo che consenta un ingresso grafico e una facile interazione risulterebbe sotto questo profilo quello più adeguato. Tale formalismo deve al tempo stesso essere dotato di una semantica il più possibile completa e non ambigua in modo da consentire la validazione delle specifiche (che può essere pensata per animazione per le parti grafiche e/o per verifica formale).
- *Definizione di una metodologia di valutazione di costi e prestazioni,* che si



	Coprocessore standard	Coprocessore nucleo	ASIP	ASIC
Prestazioni	Medie	Medie	Alte	Massime
Consumo di potenza	Alto	Medio	Medio-basso	Minimo
Flessibilità	Media	Alta	Alta	Bassa
Tempo di progetto	Basso (sw)	Medio (hw+sw)	Massimo (hw+sw)	Alto(hw)

Tabella di paragone tra vari stili di progetto

presti il più possibile ad essere automatizzata. Alle prestazioni più ovvie, relative alla velocità e all'area di silicio occorre unire quelle (molto meno formalizzate) che riguardano affidabilità e riprogrammabilità. D'altra parte, anche la definizione attuale dei costi è estremamente parziale (relativa da un lato ancora all'area, dall'altro al costo valutabile come anni uomo) ed esclude aspetti quali manutenibilità, riusabilità di parti del progetto ecc.

- *Definizione di metodologie di partizionamento assistito da calcolatore.* Le soluzioni in uso attualmente prevedono che il partizionamento sia deciso dal progettista. Tale soluzione, peraltro lenta, di fatto scoraggia il progettista dall'esplorare a fondo lo spazio di progetto; la disponibilità di un "advisor" che sostenga tale esplorazione e compia le relative valutazioni è indispensabile per rendere la progettazione concorrente efficace. Programmi per il partizionamento automatico di sistemi digitali in componenti hardware e software sono stati studiati e realizzati da vari ricercatori [7, 8, 13]. Per esempio, Ernst et al. [7] ha studiato la sintesi di coprocessori dedicati per la velocizzazione di funzioni critiche. Nel suo approccio, il sistema è rappresentato da un programma in un linguaggio di programmazione (simile al C); da questo modello software vengono estratti componenti hardware che realizzano le componenti critiche. Gupta et al. [8] ha invece usato un modello hardware del sistema da realizzare (nel linguaggio HardwareC), ed ha affrontato il problema di trasferire funzioni non critiche da hardware a software, sotto vincoli di prestazioni, per ridurre il costo della realizzazione ed aumentarne la riprogrammabilità.

- *Estensione delle tecniche di sintesi concorrente hardware/software.* Per tale fase esistono oggi alcuni prototipi sperimentali [1, 3, 4, 7, 8]. Questa fase è strettamente legata alla rappresentazione concorrente di componenti hardware e software, usando formati intermedi standard (da definire) possibilmente costituiti dai linguaggi in uso (VHDL o Verilog HDL per la descrizione hardware, e linguaggi di programmazione per la parte software) così da collegarsi poi agli altri strumenti di progettazione e validazione. La sintesi di sistemi digitali dedicati rappresenta la naturale evoluzione delle tecniche di sintesi architetturale per circuiti hardware [6]. Una ipotesi di lavoro è che l'intero sistema digitale sia rappresentato usando un linguaggio specifico e che sia poi partizionato, manualmente o automaticamente, in componenti hardware e software. Le componenti hardware possono essere realizzate da ASIC usando strumenti di sintesi hardware, mentre le parti software possono essere estratte e compilate. L'interfacciamento e la sincronizzazione delle parti hardware e software rappresentano una parte importante della sintesi. Un'ipotesi alternativa [4,12] è l'uso di un modello che permetta la specifica di un sistema in componenti separate hardware e software ma interagenti, ad esempio estendendo il concetto di macchina a stati finiti [4].

- *Validazione del progetto concorrente,* mediante tecniche di co-simulazione nella quale devono intervenire le specifiche di tipo temporale, la descrizione dell'ambiente esterno di funzionamento, le specifiche hardware e software interagenti. Pertanto, l'uso di particolari linguaggi e formati intermedi è particolarmente importante per quanto riguarda la validazione. La simulazione di circuiti e/o sistemi digitali (dedicati o meno) è il mezzo tradizionale per validarne il corretto funzionamento prima della fabbricazione. Un simulatore è un programma che prende come ingresso un modello del sistema e dei vettori di stimolo e ne calcola la risposta. L'analisi della risposta può mettere in luce potenziali errori di progettazione, anche se in genere non può garantire la copertura di tutti i possibili malfunzionamenti. La velocità di un simulatore è molto importante, perché determina il numero di vettori di uscita analizzabili in un tempo costante, e quindi il grado di confidenza nel progetto. Oggi sono disponibili simulatori veloci a livello RTL (Register-Transfer Level) per modelli hardware descritti nei linguaggi Verilog HDL e VHDL. Modelli di processori standard in Verilog HDL e VHDL sono pure disponibili commercialmente. Pertanto è possibile in linea di principio modellare un sistema misto hardware e software in un linguaggio RTL e validarlo con esistenti simulatori. In pratica, nascono due difficoltà. In primo luogo, la simulazione deve avvenire a livello di ciclo macchina, per esercitare la componente hardware dedicata. Anche se si può ipotizzare un corretto funzionamento del processore, è necessario simularlo passo a passo con l'hardware dedicato, perché potenziali problemi possono nascere dall'interfacciamento. La simulazione di sequenze di istruzioni software di interesse pratico può causare lunghe simulazioni a livello RTL. Pertanto, per migliorare la velocità e l'efficienza della simulazione, conviene rappresentare i processori a livello di istruzione. La seconda difficoltà nasce dal fatto che parecchi sistemi dedicati sono eterogenei; ad esempio, un telefono cellulare ha componenti hardware, firmware e software, più parti analogiche a frequenze differenti. Architetture di questo tipo possono essere trattate da ambienti di simulazione quali il programma Ptolemy [10], relativo specificamente al settore dell'elaborazione dei segnali, che gestisce l'eterogeneità delle specifiche di progetto. Il progettista che usa Ptolemy può modellare aspetti diversi del progetto in corrispondenti stili di rappresentazione. Il simulatore permette l'esecuzione simultanea di tali modelli ed il loro interfacciamento, risolvendo così il problema dell'eterogeneità.

- *Creazione di metodologie integrate di testing (co-testing) per il prototipo.* Questo comporta non solo la verifica di correttezza individuale degli ASIC e del software e della loro interazione, ma possibilmente la garanzia che la presenza di guasti nei dispositivi hardware si traduca comunque in un comportamento determinabile del sistema (in pratica si tratta di estendere i concetti del design for testability in un ambito misto hardware/software).

Ad oggi, non esistono sistemi commerciali che coprano tutto il progetto concorrente hardware/software come rappresentato dalle fasi precedenti. Nell'ambito della ricerca italiana, uno sforzo particolare in questa direzione è stato svolto a Milano dal Cefriel (Centro per la Formazione e la Ricerca in Ingegneria Elettronica), nell'area di ricerca che si occupa delle metodologie di progetto automatico di sistemi hardware. È stato definito un prototipo di framework per la realizzazione di sistemi dedicati alle telecomunicazioni (e quindi di sistemi sostanzialmente reattivi), basandosi su una collaborazione al progetto di Italtel [1, 2]. L'applicazione scelta ha portato ad individuare un'architettura base del sistema che risulta costituito da un processore nucleo, che può variare da caso a caso, e da uno o più ASIC tra loro interagenti. Le comunicazioni con il processore avvengono attraverso un bus ed un protocollo preciso.

La specifica del sistema viene rappresentata utilizzando uno strumento commerciale che permette di descrivere graficamente macchine a stati finite interagenti e, in un linguaggio simile a VHDL, le elaborazioni da svolgere nei singoli stati della macchina. Tale descrizione viene compilata in un formato interno, sul quale è possibile valutare diversi partizionamenti hardware e software, identificando costi e prestazioni di ogni alternativa. L'alternativa scelta viene poi tradotta in VHDL per quanto riguarda la parte hardware mentre la parte software viene compilata in un codice assemblatore virtuale, che utilizza un insieme di istruzioni molto ridotto e generale, che viene successivamente tradotto in modo semplice nel nucleo scelto per l'implementazione. È stato definito un protocollo di interfaccia tra le due parti in modo da poter realizzare una simulazione del sistema, basandosi su un simulatore VHDL e su una implementazione VHDL delle istruzioni del linguaggio assemblatore. La sintesi della parte hardware da VHDL viene realizzata mediante strumenti di sintesi logica commerciali. Più in generale, esistono alcuni prototipi universitari dedicati ad applicazioni specifiche (microcontrollori o sistemi DSP) ed alcune metodologie interne sviluppate dai laboratori di ricerca di alcune grandi aziende di elettronica per l'automobile e di telecomunicazioni, che fanno uso di strumenti CAD commerciali che realizzano in modo automatico alcune delle singole fasi, in particolare quella di cattura delle specifiche funzionali del sistema e di co-simulazione. Sono comunque in corso di sviluppo, da parte di molte società di CAD, nuovi prototipi di sistemi di progettazione concorrente, originati soprattutto da sistemi di sintesi ad alto livello per circuiti ASIC (Cadence, Mentor, Speed SA, Synopsys, Synthesia, Viewlogic); si prevede che strumenti CAD per la progettazione di sistemi dedicati saranno disponibili in breve termine, e daranno un grosso impulso alle vendite di quelle ditte di progettazione che ne adotteranno presto l'uso. ▲

Nato Advanced Study Institute: Hardware/Software co-design

Il *Nato Advanced Study Institute on Hardware/Software co-design* è un corso di due settimane che copre problemi e soluzioni relative al progetto concorrente di hardware e software.
Il corso avrà luogo al Grand Hotel Tremezzo (Como), dal 19 al 30 giugno 1995.
Il corso coprirà i seguenti argomenti:

- metodologie di progetto concorrente;
- tecniche di specifica e modellistica;
- sintesi e verifica di hardware;
- progetto di processori;
- progetto di sistemi dedicati e di controllo;
- applicazioni per le telecomunicazioni;
- compilazione retargatable;
- co-simulazione;
- esempi di progetto.

Le lezioni saranno supportate da dimostrazioni di sistemi di progetto per le applicazioni corrispondenti.

Direttori:
Giovanni De Micheli, Stanford University;
Maria Giovanna Sami, Politecnico di Milano.

Per iscrizioni (entro il 15 aprile 1995)
World-Wide Web: URL http:
//akebono.stanford.edu/users/nanni/nato/index.html
Segreteria: Sig. Laura Caldirola, Politecnico di Milano,
Dip. Elettronica ed Informazione,
Piazza Leonardo da Vinci 32, 20133 Milano.
Tel: (+39) 2 2399 3623 - fax: (+39) 2 2399 3411.
email: caldirol@elet.polimi.it2.

Bibliografia

[1] S. Antoniazzi, A. Balboni, W. Fornaciari, D. Sciuto, *HW/SW Codesign for Embedded Telecom Systems,* Proceedings of Iccd, 1994.

[2] S. Antoniazzi, A. Balboni, W. Fornaciari, D. Sciuto, *The Role of VHDL within the TOSCA Codesign Framework,* Proceedings of Euro-VHDL, 1994.

[3] P. Chou, E. Walkup and G. Borriello, *Scheduling Strategies in the Co-Synthesis of Reactive Real-Time Systems,* IEEE Micro, pag. 37-47, agosto 1994.

[4] M. Chiodo, P. Giusto, A. Jurecska, H. Hsieh, L. Lavagno and A. Sangiovanni, *A Formal Methodology for Hardware/Software Co-design of Embedded Systems,* IEEE Micro, pag. 26-36, agosto 1994.

[5] G. De Micheli, *Computer-Aided Hardware/Software Co-design,* IEEE Micro, pag. 11-16, August 1994.

[6] G. De Micheli, *Synthesis and Optimization of Digital Circuits,* Mc Graw-Hill, 1994.

[7] R. Ernst, J. Henkel and T. Benner, *Hardware-Software Co-synthesis for micro-controllers,* IEEE Design and Test, Dicembre 1993, pag. 64-75.

[8] R. Gupta and G. De Micheli, *System Co-synthesis for Digital Systems IEEE Design and Test,* vol. 10, n. 3, Settembre 1993, pag. 29-41.

[9] X. Hu, J. D'Ambrosio and D-L Tang, *Analysis in Hardware/Software Co-design: An Automotive Case Study,* IEEE Micro, pag. 17-25, Agosto 1994.

[10] A. Kalavade and E. Lee, *A Hardware-Software Co-design methodology for DSP Applications,* IEEE Design and Test, vol. 10, n. 3, Settembre 1993, pag. 16-28.

[11] C. Liem, T. May and P. Paulin, *Instruction-Set Matching and Selection for DSP and ASIP Code Generation,* Proceedings of the European Design and Test Conference, 1994, pag. 31-37.

[12] M. Theissinger, P. Stravers and H. Veit, *CASTLE: An Interactive Environment for HW-SW Codesign,* Proceedings of the Int. Workshop on Hardware/Software Co-design, pag. 203-209, 1994.

[13] D. Thomas, J. Adams and H. Schmitt, *A Model and Methodology for Hardware-Software Co-design,* IEEE Design and Test, vol. 10, n. 3, Settembre 1993, pag. 6-15.

[14] W. Wolf and E. Frey, *Tutorial on Embedded System Design,* Proceedings of Iccd, 1992, pag. 18-21.