

Oggi, la prototipazione degli FPGA è decisiva nella certificazione della qualità di un complesso ASIC

I vantaggi dei prototipi basati su FPGA nella verifica dei sistemi ASIC

Brian Caslis
Direttore Marketing
Synplicity

Non è semplice garantire il corretto funzionamento degli ASIC. Sempre più spesso, la pressione imposta da costi e time-to-market può rivelarsi il fattore determinante nel sentenziare la differenza fra un prodotto di successo e un fallimento. Inoltre, i tradizionali metodi di verifica degli ASIC non sono soltanto da considerare un antipatico ritardante collo di bottiglia nel ciclo di sviluppo dei prodotti, giacché sono anche piuttosto imprecisi e non riescono mai a testare realmente tutte le funzionalità dei sistemi, così, nei casi peggiori, possono causare errori e difetti che si ripercuotono fino a decretare il fallimento di qualche costoso ASIC.

Un passo avanti nel miglioramento dell'efficienza dei tool di verifica è l'inserimento di qualche stimolo elettrico o logico in tempo reale, come ad esempio un segnale audio o video in transito sulla rete, un'opzione che ha la peculiarità di rilevare alcuni difetti funzionali in anticipo sulla restante fase di verifica. Tuttavia, le tradizionali tecniche di emulazione e simulazione non sono abbastanza veloci per elaborare i segnali in tempo reale.

Inoltre, molti sistemi sono composti da più parti funzionali sia hardware, sia software, che lavorano contemporaneamente e possono quindi essere simulate correttamente solo in parallelo.

Di conseguenza, a causa dei sempre più complessi requisiti di velocità nel funzionamento dei moderni processori e delle limitate prestazioni degli attuali acceleratori di emulazione e simulazione, solo un'efficace prototipazione può garantire le prestazioni minime sufficienti a effettuare realmente la completa verifica hardware e software dei sistemi. Infine, bisogna anche tener conto dei costi che comporta una verifica e che molte aziende cercano di contenerli scomponendo la verifica in più gruppi di lavoro o terziandone alcune parti.

Confrontare le prestazioni

La verifica dipende sia dalle dimensioni dei progetti, sia dalle dimensioni dei vettori di verifica. Al crescere delle dimensioni dei progetti, ingrandiscono anche i vettori necessari alla loro verifica. Una regola empirica è che per ogni aumento di 10 volte nelle dimensioni di un progetto, parimenti occorrono vettori 10 volte più grandi, il che significa che la complessità della verifica aumenta di 100 volte. Invero, ciò ha spinto molti progettisti a cercare soluzioni alternative in grado di accelerare questa ingombrante, ma necessaria fase. Il nodo critico della verifica sugli ASIC è la simulazione software basata sugli eventi. Questa tecnologia è usata nella maggior parte dei simulatori software più diffusi come

Cadence NC-Verilog e NC-Sim, Synopsys VCS e MTI ModelSim. In pratica, gli eventi sono ricavati in base al disegno del circuito e così anche gli stimoli e quanto serve in ogni momento della veri-

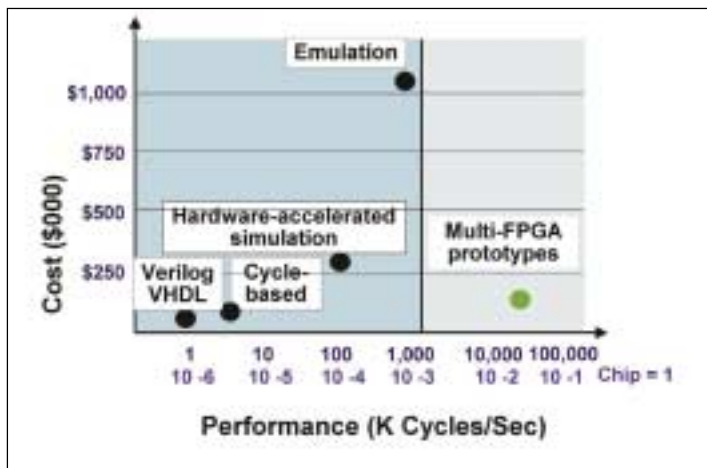


Fig. 1 - Schema che confronta il costo e le prestazioni delle differenti tecnologie di verifica

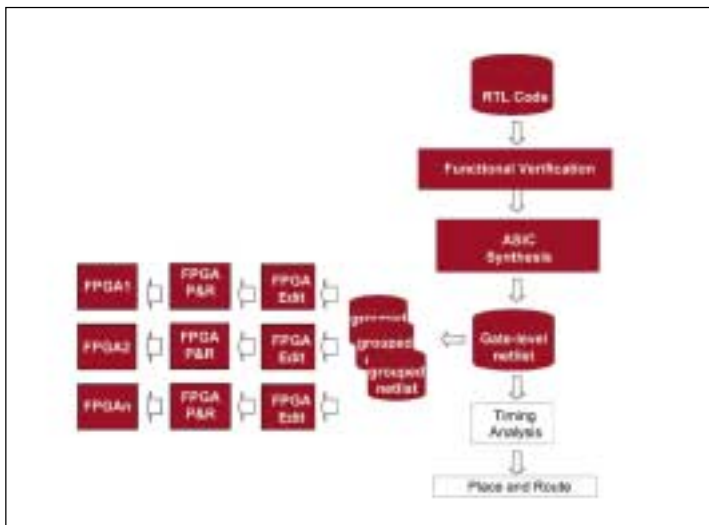
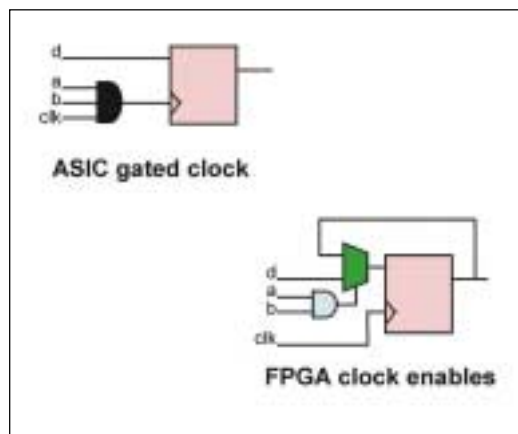


Fig. 2 - L'approccio manuale all'implementazione dei prototipi

Fig. 3 - Un esempio di struttura con clock disabilitato convertita in una struttura abilitata al clock



fica. Naturalmente i diversi simulatori possono differire per il tipo di compilatore o interprete utilizzato. Le prestazioni tipiche medie sono di circa 1000 cicli istruzioni al secondo (cps), ma variano in funzione delle dimensioni del progetto e possono, in alcuni casi, rallentare anche fino a 100 cicli al secondo.

Un'altra metodologia di simulazione recentemente diffusasi è la simulazione basata sui cicli. In pratica, essa si propone di rendere più veloce la simulazione basata sugli eventi, compilando gli eventi in ordine di esecuzione. In tal modo si viene a eliminare la necessità di determinare quest'ordine durante la simulazione e si eliminano inoltre automaticamente quegli eventi per i quali è sufficiente simulare le condizioni di temporizzazione. Questo vantaggio, tuttavia, è anche il motivo per cui non tutti i sistemi possono essere simulati con questa tecnologia, ma solo quelli che sono sincroni per natura e sono assolutamente privi di qualsiasi forma di retroazione combinatoria di tipo asincrono. Se viene soddisfatto questo requisito, allora la simulazione basata sui cicli può essere mediamente da 5 a 10 volte più veloce di quella basata sugli eventi e effettuarsi con velocità fino a 5000 cicli istruzioni al secondo, anche se la velocità di esecuzione media si aggira sui 1000 cps.

Un ulteriore passo in avanti nelle prestazioni della fase di verifica è l'uso congiunto delle risorse software e hardwa-

re. Gli acceleratori hardware, ad esempio, sono delle risorse gestite dalla workstation di comando e il più delle volte lavorano in parallelo alla simulazione basata sugli eventi. Nel nostro caso, l'acceleratore hardware può acquisire e svolgere la maggior parte del lavoro di elaborazione al posto del simulatore software e ciò si traduce in un significativo aumento delle prestazioni globali. Inoltre, l'acceleratore hardware può anche occuparsi dei ritardi di simulazione non compatibili con la simulazione basata sui cicli e permettere così di ampliare la varietà dei sistemi verificabili con quest'efficace metodologia. Le prestazioni di un acceleratore hardware possono arrivare fino a 100000 cicli al secondo e variano in funzione della complessità del sistema e di come viene partizionato il carico di lavoro fra esso e il simulatore software.

Un emulatore hardware, invece, è un simulatore basato sui cicli che viene implementato in forma hardware per incrementarne le prestazioni. Il suo compito è quello di far girare i progetti interamente in hardware, anche se molti tool offrono comunque la possibilità d'interfacciamento con il simulatore software. Generalmente, richiedono uno sforzo maggiore nella fase progettuale per rendere i sistemi più compatibili con gli operatori hardware, ma lo sforzo è certamente compensato dal

considerabile miglioramento delle prestazioni soprattutto in termini di velocità, anche rispetto agli acceleratori. Infatti, le prestazioni degli emulatori hardware possono arrivare fino a 1 milione di cicli al secondo o, detto in altri termini, alla frequenza di 1 MHz, con prestazioni medie attorno ai 500 kHz.

I prototipi riconfigurabili sono sistemi hardware che implementano i sistemi tipicamente sviluppati con i circuiti ASIC sulle schede basate su FPGA. Un'importante peculiarità di questi sistemi è la possibilità di scomporre il progetto originale in più elementi hardware del prototipo, ovvero dell'FPGA. In funzione dell'approccio utilizzato, quest'operazione può richiedere un po' più tempo rispetto alla compilazione per l'emulatore hardware. Tuttavia, una volta che il sistema è trasferito nel prototipo, allora le prestazioni possono raggiungere anche i 10 MHz e dipendono dal tipo di FPGA riconfigurabile prescelto e dalla bontà della partizione delle funzioni vitali del sistema nei diversi elementi logici. I prototipi hardware basati su FPGA sono simili ai prototipi riconfigurabili, ma possono essere implementati sia sulle schede

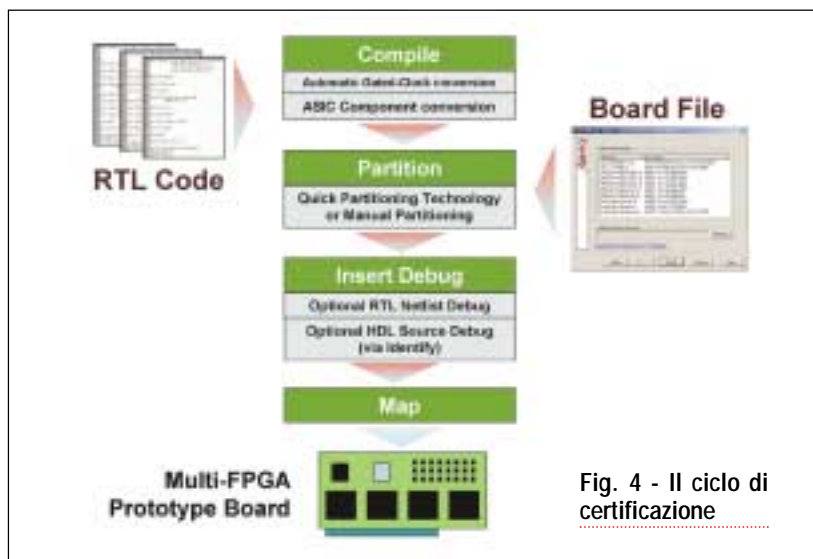


Fig. 4 - Il ciclo di certificazione

standard, sia sulle schede specializzate per operare in particolari ambienti. In entrambi i casi, si tratta sempre di partizionare un sistema ASIC negli elementi logici di un FPGA.

D'altra parte, molti prototipi possono essere connessi agli altri elementi hardware del sistema finale (come processori, memorie, sottosistemi, moduli a radiofrequenza e dispositivi analogici) in modo da poter funzionare singolarmente anche in tempo reale o in una modalità molto vicina al tempo reale. A differenza degli altri approcci di verifica, questi prototipi hanno il vantaggio di poter essere usati anche nei test e collaudi software o hardware successivi. La prestazioni di un prototipo custom possono variare fra 10 e 80 MHz, secondo l'hardware di supporto e la bontà della partizione. Queste prestazioni sono da ritenere abbastanza veloci per verificare la correttezza di funzionamento di un sistema ASIC alle sue più alte frequenze di lavoro, perché ciò che importa è che il prototipo sia veloce almeno tanto quanto gli I/O d'interfaccia del sistema. Perciò, se un prototipo è più veloce dei bus

Fig. 5 - Una lista di venditori che offrono tool specifici per la prototipazione dei progetti

PCI, o delle reti Ethernet, o delle stringhe video, allora può essere senz'altro utilizzato per le verifiche in tempo reale. Infine, i prototipi sono vantaggiosi anche per altre applicazioni, come ad esempio con i sistemi wireless, dove possono servire per testare le funzionalità dei prodotti nel loro reale ambiente di destinazione. Non c'è alcun altro approccio alla verifica in grado di offrire questo stesso vantaggio. In più, il basso costo dei prototipi hardware permette di replicarli su schede multiple per essere, ad esempio, distribuiti a gruppi di lavoro differenti, ciascuno dei quali può occuparsi di una particolare fase di verifica.

Consolidare una tecnologia

Completare correttamente la verifica di un ASIC oggi non è facile. Ci sono molti

Prototyping Tool Providers	
AMO DCC http://www.venus-x.com	Hardi Electronics AB http://www.hardi.se
Aptix http://www.aptix.com	Nallatech http://www.nallatech.com
The Dini Group http://www.dinigroup.com	ProDesign Electronics & CAD-Layout GmbH http://www.prodesign-usa.com
EVE USA, Inc. http://www.eve-team.com	Synplicity http://www.synplicity.com
Gidel http://www.gidel.com	

aspetti da tenere in considerazione e molte variabili che influiscono sul funzionamento dei sistemi e quindi sul successo delle verifiche. Ci sono anche delle funzioni che si possono verificare più efficacemente con alcune tecnologie, piuttosto che con altre. La verifica dei moduli, ad esempio, è più conveniente nella fase di simulazione, mentre i test regressivi sono più adatti all'emulazione, che è più veloce a elaborare i vettori.

D'altra parte, i test in-system non sono adatti né alla simulazione, né all'emulazione perché richiedono collaudi in tempo reale o in una modalità abbastanza vicina al tempo reale. Infine, ci sono anche delle funzioni, come il debug hardware e software, per le quali è indispensabile che la velocità della verifica hardware sia piuttosto elevata e almeno maggiore di 8 MHz. Questi requisiti si possono soddisfare solo realizzando un prototipo del sistema in fase di sviluppo e ciò è confermato dalla moltitudine dei progettisti che, solo grazie ai prototipi, hanno potuto scoprire problemi e difetti nei progetti dei quali non avrebbero mai sospettato l'esistenza.

Per molte aziende in passato la prototipazione era una tecnologia di verifica da usare solo in particolari circostanze. L'approccio, in questo caso, consisteva nell'usare i tool di progetto e i tool di sintesi ASIC esistenti per scomporre manualmente lo sviluppo in più parti e adattare sempre manualmente le varie parti in modo da garantire a ciascuna la minima autonomia di funzionamento. Il problema è che in questo modo, il lavoro per arrivare "a mano" a un prototipo è notevole e richiede numerose iterazioni non solo nello sviluppo, ma anche nel collaudo del prototipo, senza garantire alcuna certezza nell'affidabilità del risultato finale, soprattutto se si tratta di schede specializzate. Questo perché ci sono numerose variabili da affrontare nel passaggio da un circuito ASIC a un dispositivo FPGA, le più critiche delle quali sono la temporizzazione sui gate e il multiplessaggio.

gio degli I/O. Il primo problema è dovuto al fatto che molti circuiti ASIC hanno le temporizzazioni sui gate che non sono adatte a essere trasformate con le strutture disponibili nella dotazione base degli FPGA, ma hanno bisogno di strutture dedicate.

Il secondo problema deriva dal fatto che, una volta ricavato l'FPGA, bisogna trovarne l'appropriata collocazione nella forma hardware. Anche se sembra ci siano abbastanza gate per contenere tutto l'ASIC, può succedere che dopo il partizionamento alcune funzioni escano dalle risorse I/O presenti nell'FPGA. Un esempio tipico ne sono i moltiplicatori ASIC, il cui frazionamento può moltiplicare esponenzialmente il numero dei gate, al punto di renderne impossibile il trasporto su FPGA. L'unica soluzione in questi casi è moltiplicare l'uso degli I/O nel tempo, il che non è tecnicamente difficile, ma non immune da controindicazioni, perché si finisce sicuramente per rallentare le prestazioni del prototipo. Dunque,

al progettista non resta che cercare la condizione ottima.

Le soluzioni possibili

I recenti passi avanti nelle tecnologie hardware e software hanno contribuito a migliorare sensibilmente i processi di prototipazione e verifica. I nuovi tool automatizzano le fasi più tediose e difficili relative alla trasformazione degli ASIC in FPGA come il partizionamento delle funzioni operative, il multiplessaggio degli I/O, la conversione delle temporizzazioni sui gate e la gestione dei pin. Uno di questi tool è il software di prototipazione Certify ASIC RTL di Synplicity, capace di leggere il codice di un ASIC ed eseguire autonomamente gran parte del lavoro di conversione fino a ricavare il prototipo FPGA. Le prestazioni del tool Certify comprendono infatti il partizionamento del codice RTL, il multiplessaggio automatico dei pin (Certify Pin Multiplication, CPM), la conversione delle temporizzazioni sui gate, l'ottimizzazione delle tempo-

rizzazioni a livello di sistema e la sintesi del codice RTL per l'FPGA. Quest'ultima funzione, la sintesi del codice, è piuttosto avanzata perché sfrutta tutte le innovazioni introdotte da Synplicity sia dal punto di vista software, sia in termini di compatibilità hardware con le schede di supporto.

Da quanto detto, segue che la prototipazione di un ASIC non può e non deve essere considerata solo una delle tante parti dell'intero processo di verifica, ma va considerata come una fase del ciclo di sviluppo che, se trascurata, può causare errori e perdite di tempo le cui conseguenze possono protrarsi per mesi. E questo vale a maggior ragione proprio nel caso più frequente in cui l'ASIC in questione sia composto da blocchi IP di diversi fornitori e solo un'accurata prototipazione può garantire il corretto funzionamento del sistema finale. 