

## SEMPLIFICARE IL PROGETTO DI SISTEMI DSP BASATI SU FPGA

Andrew Dauman  
Vice president, corporate  
application engineering

Dirk Seynhaeve  
Direttore, corporate  
application engineering

Synplicity

*Una metodologia automatizzata che sfrutta le più recenti innovazioni a livello di modellazione e di sintesi permette di realizzare sistemi DSP basati su FPGA dalle prestazioni particolarmente spinte*

**A**l giorno d'oggi l'utilizzo di FPGA in soluzioni DSP è sempre più diffuso: per rendersi conto di ciò è sufficiente osservare il crescente numero di prodotti che integrano la tecnologia DSP.

Quest'ultima è oramai "onnipresente" non solamente in dispositivi destinati ai mercati consumer, automobilistico e della telefonia, ma anche in apparati via via più sofisticati. Applicazioni quali stazioni base wireless, elaborazione dei segnali radar, riconoscimento delle impronte digitali e SDR (Software Defined Radio) richiedono capacità di elaborazione molto spinte. Esse hanno contribuito a evidenziare i limiti, a livello di prestazioni, dei processori standard e sollecitare lo sviluppo di soluzioni hardware in grado di garantire un

aumento delle prestazioni stesse. Agli inizi degli anni '90, i progettisti hanno pensato di risolvere il problema legato alla richiesta di una maggiore potenza di elaborazione mediante l'utilizzo di più processori. Tale approccio tende a complicare la progettazione a livello di sistema, poiché comporta il coordinamento delle funzionalità di più processori, senza dimenticare gli svantaggi legati ai costi elevati e allo spreco di risorse.

Quando i primi FPGA con funzioni DSP hanno fatto la loro comparsa sul mercato, i progettisti di sistemi DSP hanno iniziato a utilizzare tali dispositivi per rafforzare le potenzialità dei loro processori. In questo caso l'FPGA rappresenta un complemento per il processore, poiché è in grado di accelerare quelle parti dell'algoritmo DSP critiche dal punto di vista delle prestazioni.

L'avvento di FPGA specializzati, come ad esempio i componenti Virtex 4 di Xilinx dotati di 512 circuiti di moltiplicazione-accumula-

zione parallela e Stratix II di Altera, ha messo a disposizione un potenziale enorme per migliorare le prestazioni attraverso la parallelizzazione. Gli FPGA ottimizzati per applicazioni DSP garantiscono prestazioni superiori di un fattore fino a 100 rispetto ad altri tipi di implementazione (Fig. 1). Per questo motivo è sempre più comune trovare un DSP standard in compagnia di un FPGA che esegue le funzioni ad alte prestazioni e questa modalità di impiego degli FPGA in questo modo è destinata ad aumentare progressivamente.

### Soddisfare le esigenze di progetto

Sebbene siano disponibili queste potenzialità a livello hardware, sorge il problema di implementare in maniera efficace questi sistemi DSP basati su FPGA. Progetti di tale ampiezza e complessità mettono a dura prova le tradizionali metodologie di design di sistemi DSP. Ciò è in larga misura imputabile al fatto che il tradizionale flusso di progettazione degli FPGA non sfrutta, nello spazio DSP, due elementi critici che contribuiscono a conferire efficienza ed efficacia

MACOPS: Multiply Accumulate Operations/s	
Rule of thumb:	
Clock frequency * Multipliers	
μP	2.4 GHz * 1/24 = .1GMACOPS
DSP	1 GHz * 1 = 1 GMACOPS
FPGA	200MHz * 500 = 100 GMACOPS
ASIC	200MHz * N = N/5 GMACOPS

Fig. 1 – Confronto tra le velocità di differenti processori

al flusso di design: la tecnologia di sintesi e i blocchi IP (Intellectual Property) portabili.

Tutti coloro che si sono dedicati alla progettazione di ASIC utilizzando la tecnologia di sintesi sono consapevoli dei vantaggi di quest'ultima.

Nel caso dei sistemi DSP basati su FPGA, questa tecnologia è essenziale, in quanto consente di effettuare l'operazione di design entry a un elevato livello di astrazione e valutare in maniera automatica tutti i compromessi possibili in termini di prestazioni e occupazione di area.

La possibilità di eseguire rapidamente il design entry a un elevato livello di astrazione e le doti di automazione non solo consentono di creare un singolo progetto, ma forniscono anche un certo numero di possibili risultati tra i quali effettuare la scelta più idonea. Nel caso di un'applicazione in cui le prestazioni sono prioritarie rispetto all'occupazione di area, potrebbe essere richiesta una realizzazione che preveda l'uso di centinaia di moltiplicatori: una tale soluzione, sebbene garantisca elevate velocità, richiede l'occupazione di un'area non indifferente. Allo stesso modo, laddove l'area rappresenta un elemento critico, una realizzazione che preveda la condivisione di un numero ridotto di moltiplicatori caratterizzati di minori prestazioni, porta a un risultato sicuramente inferiore. Per poter valutare le diverse opzioni è necessario disporre di tool potenti, indispensabili per lo sviluppo ottimizzato di sistemi DSP basati su FPGA,

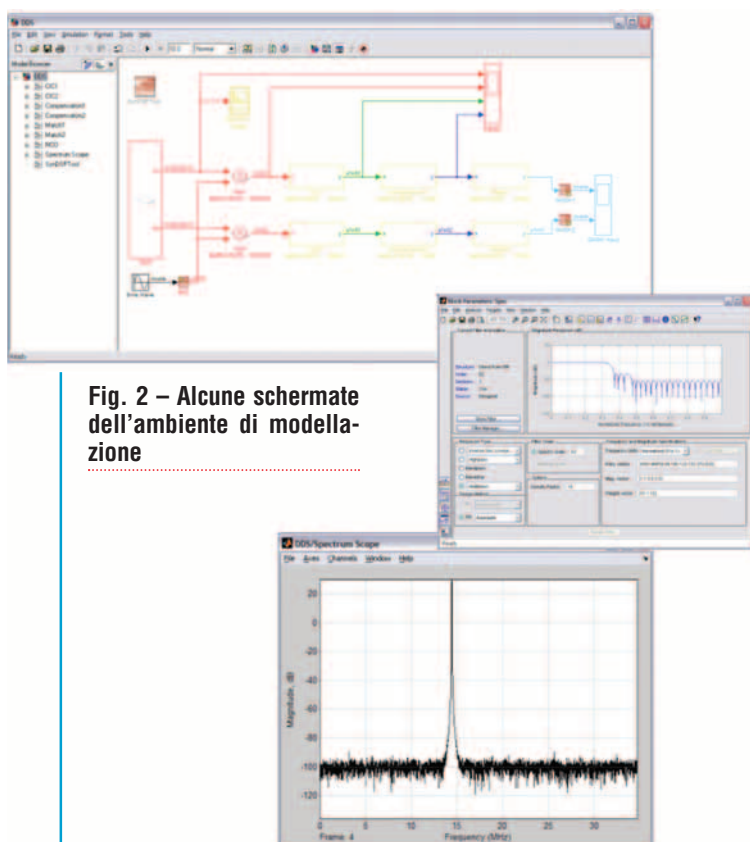
L'altro elemento chiave per lo sviluppo di sistemi DSP efficienti è la disponibilità di blocchi base adeguati, o IP.

Gli IP adatti allo scopo devono possedere due attributi fondamentali: estendibilità e portabilità. La prima permette al progettista di realizzare IP custom senza penalizzazione alcuna in termini di efficienza. I nuovi blocchi funzionali sono carat-

terizzati da una maggiore efficienza perché nel successivo processo di sintesi le parti non necessarie o inutilizzate verranno continuamente ottimizzate. Anche la portabilità è un elemento essenziale per garantire l'efficienza. I progettisti di sistemi DSP devono poter sviluppare i loro algoritmi una volta sola e quindi farli girare in qualsiasi dispositivo del produttore di FPGA senza apportare alcuna modifica. La portabilità, oltre a garantire un'elevata efficienza, dà la possibilità di scegliere l'implementazione più adatta con estrema facilità. Anche la verifica di un sistema DSP non è esente da problemi. In questa fase, l'analisi e il debug di un segnale risultano più complicati rispetto all'esame di grafici nel dominio del tempo e della frequenza e di diagrammi di dispersione. Poiché i segnali digitali sono caratterizzati da tempo di campionamento e da un'ampiezza di tipo discreto, i tool di verifica devono essere capaci di definire e manipolare in maniera efficiente la variabile tempo di applicazioni DSP dove sono presenti più velocità. Oltre a ciò, devono effettuare in tempi rapidi il passaggio da una simulazione in virgola mobile estremamente accurata a una simulazione in virgola fissa con lunghezza di parola finita. Infine è necessaria la presenza di un linguaggio per la modellazione degli algoritmi DSP che comprenda il supporto nativo di concetti quali tempo, risorse in virgola fissa e parallelismo.

## Una sinergia costruttiva

Le recenti evoluzioni nelle tecniche di design mettono a disposizione interessanti soluzioni che consentono ai progettisti di sistemi DSP di affrontare efficacemente le loro specifiche problematiche. Nel campo della modellazione, ad esempio, l'ambiente di progettazione di sistema basato su modelli matematici Simulink di The Mathworks rende disponibili



**Fig. 2 – Alcune schermate dell'ambiente di modellazione**

nuove funzionalità in termini di modellazione e simulazione (Fig. 2). Questo ambiente è in grado di trattare in maniera nativa problematiche tipiche di sistemi DSP quali definizione e gestione del tempo discreto in presenza di più velocità e simulazione in virgola mobile single source.

Per l'implementazione mediante FPGA, la sintesi DSP rappresenta il punto di svolta che lega la verifica DSP con una realizzazione DSP ottimizzata. Grazie a funzionalità del tipo di quelle integrate nel tool Synplify DSP di Synplicity, i progettisti possono disporre di risorse automatizzate e indipendenti dal dispositivo che consentono loro di esaminare in maniera accurata ed efficiente le diverse opzioni di implementazione ed portare a termine la mappatura nel dispositivo target.

L'impiego congiunto della sintesi DSP e Simulink consente di riunificare in un unico ambiente le competenze di coloro che si occupano dell'architettura di sistema e dei progettisti hardware. I primi generano un modello indipendente da un particolare fornitore per Simulink – a livello puramente algoritmico e quindi focalizzando la loro attenzione sulle funzionalità ad alto livello del progetto. Nel momento in cui il modello viene trasferito al progettista hardware, la specifica non ha implicazioni di tipo architetturale. Poiché l'infrastruttura per la verifica dei sistemi DSP dell'ambiente di modellazione consente di integrare senza problemi l'engine di sintesi, il progettista hardware può esaminare i diversi compromessi architetturali

senza modificare la fonte di verifica. Poiché quest'ultima viene preservata, coloro che si occupano dell'architettura di sistema non devono preoccuparsi dei problemi di implementazione hardware, mentre a loro volta i progettisti hardware non hanno a che fare con le specifiche dell'algoritmo. Nello stesso tempo vengono garantite l'integrità e l'ottimizzazione del progetto, mentre la produttività di entrambi i team migliora. Un elemento di fondamentale importanza di questa metodologia è la disponibilità di una libreria generica di funzioni DSP. Gli IP proposti dai vari costruttori tendono a complicare il progetto dell'algoritmo con dettagli non necessari relativi all'implementazione. L'impiego di una libreria di funzioni DSP generica svincolata da parametri architetturali permette di elaborare i segnali di ingresso e di produrre un'uscita sulla base di specifiche ad alto livello. Sfruttando una libreria ad alto livello, anche la latenza associata alle funzioni DSP può essere rinviata alla fase di ottimizzazione dell'architettura. Attraverso la sintesi DSP è possibile ottenere le funzionalità hardware e la relativa implementazione.

Ambienti quali Simulink, librerie portabili e sintesi DSP sono elementi chiave per il miglioramento del progetto di sistemi DSP, anche se non va dimenticata l'importanza di riunificare queste funzionalità all'interno di una metodologia globale che unisca i domini della descrizione RTL e della realizzazione del progetto. Un flusso di

progetto ottimizzato di sistemi DSP accresce le possibilità esistenti ottenute per mezzo di una generica libreria e delle funzionalità combinate di Simulink e della sintesi DSP (Fig. 3). Nella fase di specifica del progetto, l'architetto di sistema opera esclusivamente a livello di astrazione algoritmica. Mediante l'impiego di un insieme di blocchi funzionali, il progettista è in grado di acquisire l'algoritmo sfruttando concetti DSP noti. Nelle fasi successive, la verifica dell'algoritmo risulta notevolmente semplificata grazie alle funzionalità proprie dell'ambiente di verifica DSP di Simulink. Funzioni quali visualizzazione e debugging, oltre alla disponibilità di acceleratori integrati, contribuiscono a facilitare e accelerare la simulazione di progetti a tempo discreto.

Il nucleo centrale di questa metodologia di progetto, il fattore determinante che permette di raggiungere gli obiettivi prefissati in termini di ingombri e prestazioni è la sintesi DSP. Quest'ultima permette di creare un'architettura che minimizza le risorse necessarie per ottenere il livello di prestazioni richiesto. Mediante l'adozione di idonee tecniche di ottimizzazione a livello di sistema come ad esempio il retiming e l'aggiunta di un periodo di latenza, la sintesi DSP permette di raggiungere gli obiettivi prefissati in termini di prestazioni del sistema complessivo. L'architettura risultante viene resa disponibile sotto forma di codice RTL sintetizzabile svincolata da un particolare produttore. In virtù di quest'ultima caratte-

ristica, è possibile sfruttare appieno le potenzialità dei tool di sintesi RTL – come ad esempio la soluzione per la sintesi di FPGA Synplify Pro di Synplicity – per ulteriori ottimizzazioni del progetto.

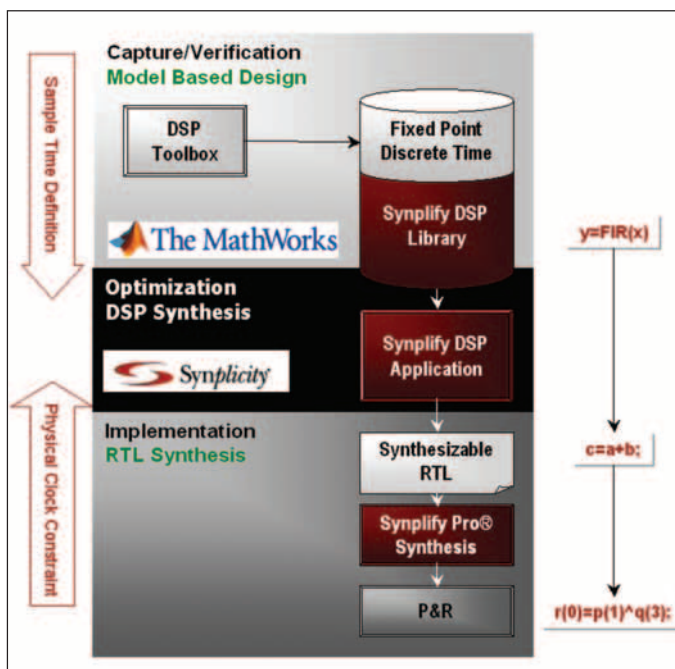
### I risultati ottenuti

Rispetto a un flusso di tipo tradizionale, la metodologia di progettazione di sistemi DSP appena descritta evidenzia significativi vantaggi. Per poter realizzare il nuovo flusso ed effettuare un confronto, sono stati utilizzati Simulink 5.1 di The Mathworks, Synplify DSP 1.0 e Synplify Pro 7.6, entrambi di Synplicity, rispettivamente per la simulazione, la sintesi DSP e l'ottimizzazione. Le applicazioni DSP prese in considerazione per il confronto comprendono un equalizzatore adattativo, un rilevatore di movimento e un filtro FIR passa basso del 50° ordine.

Il tempo necessario per l'operazione di design entry è paragonabile per un progettista che abbia familiarità sia con gli algoritmi DSP sia con le architetture IP. All'aumentare delle dimensioni del progetto è probabile che il flusso di sintesi DSP risulti migliore rispetto a quello tradizionale a causa dell'indipendenza dalla latenza dell'algoritmo e del fatto che la sincronizzazione di percorsi multipli è istantanea.

Il confronto dei risultati di progetto ottenuti con la sintesi DSP e il flusso tradizionale permette di evidenziare le migliori conseguibili con la prima metodologia anche in presenza di differenti scenari di ottimizzazione. Quando l'ottimizzazione ad alto livello non viene eseguita durante la sintesi DSP, ogni ottimizzazione

**Fig. 3 – Flusso di progettazione della sintesi DSP**



risultante è in larga misura attribuibile solamente alla sintesi RTL. Anche senza l'ottimizzazione della sintesi DSP, il numero di unità logiche impiegate diminuisce in maniera coerente in tutti i circuiti di test, a fronte di un miglioramento delle prestazioni.

È buona norma prendere in considerazione diversi scenari di ottimizzazione. Quando è permessa la condivisione delle risorse, è normale attendersi un significativo incremento in termini di utilizzazione delle risorse, a fronte di una penalizzazione per quel che riguarda le prestazioni. I circuiti di test hanno confermato questa affermazione, evidenziando una drastica diminuzione delle risorse utilizzate a costo di una riduzione delle prestazioni abbastanza significativa. È consigliabile adottare questa tecnica di ottimizzazione quando le risorse sono limitate ed è tollerabile un certo degrado a livello di prestazioni.

Le tecniche di ottimizzazione che prevedono il retiming rappresentano un'altra opzione per migliorare i risultati della sintesi DSP. Quando è possibile utilizzare il retiming, si osserva un notevole aumento delle prestazioni in entrambi i casi – sintesi DSP e flusso tradizionale – a fronte dell'utilizzo di un maggior numero di risorse. Per ottenere la temporizzazione desiderata, tool come Synplify DSP di Synplicity prevedono la ridistribuzione dei registri e l'introduzione di pipeline a livello architetturale. L'integrazione del retiming ad alto livello con il retiming a livello di gate – eseguibile mediante tool quali Synplify Pro di Synplicity – comporta un'ulteriore ottimizzazione. L'impiego congiunto del retiming ad alto livello e a livello di gate permette di ottenere risultati veramente ottimizzati, che si traducono in un incremento delle prestazioni senza che ciò comporti l'impiego di risorse aggiuntive.

Le odierne applicazioni DSP, che richiedono prestazioni sempre più spinte, stanno favorendo la diffusione di FPGA specifiche per queste applicazioni, il cui utilizzo comporta l'insorgere di nuovi problemi a livello progettuale. Un flusso di sviluppo di sistemi DSP basati su FPGA di tipo automatizzato che sfrutta le metodologie esistenti è in grado di fornire risultati nettamente migliori rispetto ai metodi tradizionali in termini di ottimizzazione. Le innovazioni apportate in termini di simulazione e modellazione specifiche per applicazioni DSP, i tool di ottimizzazione e sintesi automatizzati e le librerie DSP generiche di tipo portabile sono gli elementi fondamentali di questo flusso. Grazie a queste nuove funzionalità, gli sviluppatori di sistemi DSP hanno a disposizione una soluzione efficace per risolvere le problematiche legate alla produttività e alla qualità del design, che permette loro di sfruttare appieno i vantaggi offerti dalle nuove tecnologie FPGA

**Synplicity (Edaway)**  
**readerservice.it n. 2**