

National Instruments e l'embedded

Con il modulo LabVIEW FPGA e i suoi dispositivi hardware RIO, l'azienda specializzata in strumenti di testing e misura ha iniziato a creare una piattaforma di sviluppo adatta anche all'embedded e alla programmazione FPGA

di
Gianluca Pizzocolo,
IPSES S.r.l.



Figura 1
Compact RIO

Da anni **National Instruments** è uno dei *leader* mondiali nella fornitura di piattaforme *software* e moduli *hardware* per il *testing* e la misurazione industriale. Le ragioni di questo successo sono sicuramente la semplicità di programmazione e di utilizzo e la forte integrazione tra il loro *software* e il loro *hardware*. Ciò consente lo sviluppo di applicazioni anche complesse in tempi brevi, certo non possibili se si ricorresse a uno sviluppo *embedded* puro.

Tra le piattaforme sviluppate, **LabVIEW** ha sempre avuto un livello di astrazione elevato e certamente anche per questo motivo è sempre stato apprezzato dagli integratori e dagli sviluppatori che avevano più dimestichezza con il *software* che con l'*hardware*.

Ad esempio, la possibilità di visualizzare graficamente un DAC come un quadratino su uno schermo in cui poter scrivere dei numeri e un ADC come un altro quadratino da cui leggere dei numeri, rende sicuramente più immediata e semplice la programmazione.

Tale livello di astrazione è sempre stato ragionevole se utilizzato per sviluppare programmi per PC interfacciati a *hardware* National: infatti il loro *hardware* è già ottimizzato e sviluppato per essere gestito nel migliore modo possibile con semplici VI preconfezionati, ma sino a pochi anni fa era ovviamente impensabile applicare tale logica a un sistema *embedded* che non fosse basato su un PC.

Da ormai un paio d'anni National ha pensato di superare questo

limite e di rendere LabVIEW adatto anche per sistemi *embedded* con FPGA: con il modulo **LabVIEW FPGA** e i suoi dispositivi *hardware RIO (Reconfigurable I/O)* (Figura 1) ha, infatti, iniziato a creare una piattaforma di sviluppo adatta anche all'*embedded* e alla programmazione FPGA.

Grazie all'integrazione della tecnologia **Xilinx**, LabVIEW facilita e riduce il tempo di sviluppo per numerose applicazioni *stand-alone*, aprendo così le porte a una tecnologia che per molti risultava essere ancora inaccessibile per gli elevati costi di apprendimento e di sviluppo.

Se si considera infatti l'utilizzo classico di FPGA, quasi sempre i progettisti *embedded* si occupano direttamente della gestione a

basso livello di tutti i singoli moduli, in modo da avere deterministicamente sotto controllo ogni componente e algoritmo del sistema.

Questo approccio, nonostante l'elevato grado di complessità ormai raggiunto dai sistemi *embedded*, permette senza dubbio di ottimizzare le prestazioni e di assicurare l'assoluto controllo dell'intero sistema, ma, al contempo, fa sì che lo sviluppo richieda tempi lunghi e notevole esperienza specifica.

Mentre è ormai assodato che il linguaggio più adatto per la programmazione dei microcontrollori è il **C**, efficacemente e facilmen-

APPROFONDIMENTI

VHDL e LabVIEW FPGA: caratteristiche generali a confronto

	VHDL	LABVIEW FPGA
Astrazione	Bassa	Elevata
Ottimizzazione	Media/Alta	Media/Bassa
Prestazioni ottenibili	Elevate	Medie
Tempo di sviluppo e debug	Elevato	Medio

Tabella 1

Caratteristiche Hardware con FPGA di National Instruments

	COMPACRIO	FLEXRIO	DAQ SERIE R
Famiglia FPGA	Spartan-3 o Virtex-5	Virtex-5	Virtex-II o Virtex-5
Prezzo	\$900÷2.900	\$3.000÷7.000	\$1.400÷3.800
Accesso diretto ai pin dell'FPGA	No	Sì	No
Real Time OS	Sì	No	No

Tabella 2

te utilizzato anche nei microcontrollori di fascia più alta mantenendo sempre un ottimo compromesso tra astrazione e "fisicità" del dispositivo che si sta programmando, non si può trovare un dualismo simile tra un linguaggio di programmazione e le FPGA: i linguaggi più utilizzati e diffusi in questo caso sono oggi il VHDL e il Verilog, mentre il system-C non è ancora un linguaggio ottimale per essere usato con questa tecnologia.

Tuttavia, la programmazione in VHDL e Verilog non è semplice e spesso nasconde insidie e problematiche apparentemente insuperabili per l'utente inesperto che prova a utilizzarli, anche se parte da un solido background di programmazione di microcontrollori: la causa è principalmente dovuta al fatto che per programmare in FPGA bisogna ragionare in termini spaziali e non più temporali,

oltre a dover considerare i problemi di skew e corse critiche che sono praticamente inesistenti nei microcontrollori.

Oggi, soprattutto grazie al notevole incremento delle dimensioni e delle prestazioni delle FPGA e al loro sempre minor costo, diventa interessante valutare la possibilità di incrementarne il livello di astrazione a lieve scapito dell'ottimizzazione, ma a notevole vantaggio della facilità e velocità d'uso e di sviluppo, questo non solo per utenti "inesperti", ma anche per quelli che hanno dimestichezza con i linguaggi di programmazione hardware.

La piattaforma LabVIEW FPGA, finalmente debuggata e ottimizzata con l'uscita lo scorso giugno della nuova versione 8.6, risponde ora a questa esigenza, rendendo l'uso delle FPGA più immediato e rapido. In questi mesi abbiamo

avuto modo di realizzare diverse applicazioni con questa piattaforma e le sue prestazioni si sono dimostrate efficaci e, soprattutto, ha permesso una notevole riduzione dei tempi di sviluppo.

Ovviamente, non è detto che LabVIEW FPGA sia sempre la soluzione migliore per qualsiasi progetto con FPGA: vi sono, infatti, alcune forti limitazioni tra cui, la più importante, è l'essere applicabile solamente ai moduli hardware forniti da National, quindi LabVIEW non è certamente la soluzione nel caso in cui si progetti da zero una scheda con un'FPGA e si voglia programmare solo quella.

Inoltre, si deve tenere sempre in considerazione che questo linguaggio non consente di sfruttare al massimo le prestazioni e le potenzialità offerte dalle FPGA. In Tabella 1 sono state messe a confronto le caratteristiche generali del VHDL e di LabVIEW FPGA: si sono considerati il livello di astrazione, l'ottimizzazione, le prestazioni ottenibili e i tempi di sviluppo necessari.

Anche se, come evidenziato, LabVIEW FPGA ha certamente alcuni forti svantaggi rispetto al VHDL puro, se si considera che National ha già in catalogo numerosi moduli di I/O (a parte i comuni moduli di acquisizione e generazione sia digitali, sia analogici, sono infatti disponibili moduli con interfacce RS232, RS485, CAN, LIN, Profibus, GSM, Wireless LAN, ZigBee ecc.), che ha reso disponi-

bili le specifiche di interfaccia dei moduli con l'FPGA, permettendo quindi anche a terzi parti di sviluppare moduli propri, aumentando così i moduli a catalogo, e che, molto spesso, non è necessario ottimizzare nel migliore modo possibile l'FPGA, dato che comunque da un'FPGA con prestazioni molto elevate si riescono a ottenere facilmente delle funzionalità soddisfacenti con uno sforzo minimo, la soluzione offerta da National può risultare molto interessante e pratica sia in termini di costi, sia di semplicità di sviluppo. Per chi volesse provare le potenzialità di questo nuovo ambiente di sviluppo, è disponibile il pacchetto "Embedded Software Evaluation Kit", costituito da una Single-Board RIO con un modulo I/O e dal software di valutazione (LabVIEW, LabVIEW FPGA e LabVIEW real-time) della durata di 90 o 180 giorni.

In questo modo qualsiasi sviluppatore può facilmente capire le potenzialità offerte dall'accoppiata FPGA-LabVIEW, senza incorrere nelle tipiche problematiche in cui tutti i neofiti del VHDL si imbattono (gestione del clock all'interno dell'FPGA, corse critiche, gestione corretta delle macchine a stati ecc.): tutta la parte più ostica della programmazione FPGA viene, infatti, mascherata e gestita dal compilatore LabVIEW che, nascostamente, chiama il compilatore Xilinx.

Un altro vantaggio non trascurabile offerto da LabVIEW FPGA è la possibilità di provare i vari moduli sviluppati per FPGA su un PC,

come si prova qualsiasi altro normale VI di LabVIEW (e chi programma in VHDL sa bene che le simulazioni con **Modelsim** o le prove con **Chipscope** richiedono tempi e risorse tutt'altro che trascurabili): tale procedura è ancora un po' complicata, ma è probabile che sarà migliorata con le successive *release*.

Va doverosamente ribadito che, in ogni caso, occorre sempre obbligatoriamente usare l'*hardware* RIO di National: è infatti questo l'unico modo per programmare FPGA con LabVIEW ed eventuale *hardware* aggiuntivo sviluppato *ad-hoc* potrà sì essere connesso, ma non potrà mai sostituire il "core" costituito dal RIO.

Probabilmente è questo il limite più grande e difficilmente superabile sia sotto l'aspetto tecnico, sia sotto l'aspetto commerciale che rende dunque questo linguaggio un sistema di sviluppo proprietario abbastanza chiuso.

Un'altra forte limitazione è data dal fatto che le schede e i sistemi RIO che utilizzano FPGA di fascia media (le **Spartan-3** della Xilinx) sono sempre integrati in un sistema dotato di un processore *real-time*: ciò, ovviamente, fa aumentare sia il costo dell'*hardware*, sia dell'ambiente di sviluppo che, oltre a LabVIEW FPGA, richiederà anche **LabVIEW real-time**.

Per trovare sistemi solo con FPGA occorre passare agli **NI-FlexRIO**, che non hanno il processore *real-time*, ma che montano un'FPGA di

fascia alta (la **Virtex-5** della Xilinx) che però porta ad avere un costo superiore rispetto agli NI-RIO con FPGA e sistema *real-time*.

Nel caso in cui si vogliono sviluppare sistemi più semplici che non richiedano né un *real-time* né una Virtex-5, National diventa meno competitiva a meno che non si voglia (e si possa) integrare tutto in una scheda di acquisizione DAQ della serie R dotata di **Virtex-II**: tali schede possono avere sia interfaccia PCI, sia interfaccia PXI e, oltre a permettere la generazione e l'acquisizione di segnali come per le altre schede DAQ, integrano una FPGA programmabile dall'utente su cui si possono implementare semplici algoritmi di elaborazione.

In *Tabella 2* si sono evidenziate le diverse caratteristiche dell'*hardware* NI con FPGA, inclusa la fascia di prezzo che, come si vede, per uno stesso sistema può variare anche parecchio a seconda del tipo di FPGA montata e della massima frequenza operativa ottenibile.

Ultima nota positiva su LabVIEW FPGA è la possibilità di poter inserire anche del proprio codice VHDL all'interno di un VI: è quindi possibile implementare eventuali algoritmi critici definiti dall'utente in maniera molto diretta senza la necessità di passare dal compilatore di National, ma definendo già del codice VHDL da inviare al compilatore Xilinx insieme al codice LabVIEW convertito.

In conclusione, LabVIEW-FPGA è la certamente una buona soluzione

sia per coloro che, non avendo la pluriennale esperienza necessaria con il VHDL, vogliono comunque programmare hardware NI che integri FPGA, sia per gli utenti esperti che vogliono ridurre i tempi (e dunque i costi) di programmazione, avvicinandosi comunque alle prestazioni che avrebbero con un sistema *embedded* progettato partendo da una normale FPGA Xilinx.

Mentre, nel caso in cui si vogliono sviluppare applicazioni semplici basati su FPGA oppure applicazioni molto performanti e con velocità elevate, la piattaforma di National non è la risposta più adatta poiché nel primo caso sarebbe difficile giustificarne il costo, mentre nel secondo non sarebbe possibile sfruttare pienamente e con un controllo sufficiente ogni blocco logico dell'FPGA.