SISTEMI EMBEDDED CON TIMING PRECISO CON LABVIEW 2010 E RIO (RECONFIGURABLE I/O)

Le caratteristiche di timing incorporate in LabVIEW 2010 aiutano tecnici, ricercatori e progettisti embedded a sfruttare l'hardware RIO di NI con CPU real-time e FPGA (Field-programmable gate array) ad alte prestazioni per creare applicazioni robuste e che rispondono in modo reattivo

A cura di Alessandro Plantamura

Quando si progettano sistemi embedded, il timing è critico – sia che si cerchi di assicurare che un loop di controllo sia eseguito in modo affidabile, che si risponda rapidamente a un segnale di I/O o che l'elaborazione sia abbastanza veloce da soddisfare i requisiti applicativi.

La costruzione di un sistema che gestisca queste esigenze di timing parte dall'hardware; esso deve fornire il giusto equilibrio fra dimensioni, consumo di potenza, throughput elaborativo e latenza. In più, il software gioca un ruolo chiave nella regolazione dell'esecuzione, nella risposta agli interrupt e nell'equilibrare il tempo fra i task di calcolo.

Integrare hardware e software embedded tenendo in considerazione il timing può essere un processo laborioso, ma il software LabVIEW 2010 e l'hardware I/O riconfigurabile (RIO) di NI riducono lo sforzo necessario per costruire sistemi deterministici affidabili che possono gestire anche le applicazioni più stringenti.

L'HARDWARE RIO DI NI OFFRE CPU REAL-TIME E CHIP FPGA (FIELD-PROGRAMMABLE GATE ARRAY) AD ALTE PRESTAZIONI

I target hardware embedded di National Instruments si basano sull'architettura RIO, che include una CPU che normalmente esegue un SO real-time e uno o più chip FPGA.

Con questa combinazione, potete realizzare task che richiedono un timing con risoluzione di nanosecondi o parallelismo massiccio in hardware FPGA, mentre la CPU esegue in modo efficiente operazioni a elevato contenuto di calcolo come matematica a virgola mobile.

La comunicazione fra il chip gli FPGA e la CPU avviene tipicamente su un bus PCI e il driver NI-RIO fornisce un'interfaccia di trasferimento dati potente e diretta da LabVIEW o codice C. Il risultato finale è una piattaforma flessibile che è ideale per un'ampia varietà di prototipi e progetti embedded. E, poiché l'architettura NI-RIO è standard fra molti target hardware differenti, inclusi NI Single-Board RIO e NI CompactRIO, potete massimizzare il riutilizzo del codice quando installate le applicazioni su dispositivi embedded robusti od ottimizzati nei costi.

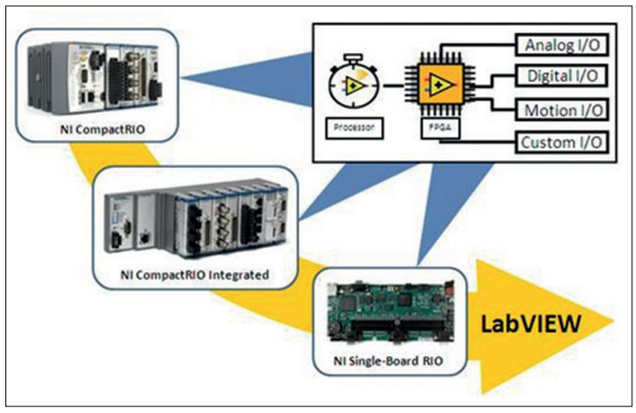

Figura 1. Il dispositivi embedded NI Reconfigurable I/O (RIO) sono costruiti attorno a un'architettura comune caratterizzata da CPU real-time, chip FPGA e I/O modulare

Per esempio, Ventura Aerospace ha usato NI Single-Board RIO, insieme a LabVIEW, per creare un sistema antincendio in grado di prevenire incendi catastrofici sugli aeromobili di trasporto merci FedEx.

Algoritmi di controllo deterministico, networking e data logging sono stati gestiti sulla CPU real-time, mentre altri task critici sono stati implementati in hardware su chip FPGA.

LABVIEW 2010 E IL TIMING INCORPORATO SEMPLIFICANO LA PROGRAMMAZIONE DI SISTEMI DETERMINISTICI AFFIDABILI

Non solo è possibile utilizzare la programmazione grafica di LabVIEW per CPU (comprese quelle che eseguono un SO real-time) e chip FPGA (inclusi quelli in target hardware RIO), ma lo stesso linguaggio LabVIEW è stato progettato fin dall'inizio tenendo presenti timing e sincronizzazione. Qui sotto vi è qualche esempio di come LabVIEW, insieme ai moduli LabVIEW Real-Time e LabVIEW FPGA, renda semplice il timing per applicazioni embedded.

CARATTERISTICHE DI TIMING INTEGRATE NEL MODULO LABVIEW REAL-TIME

Programmazione di applicazioni deterministiche, ad anello chiuso con LabVIEW Real-Time e il Timed Loop

Il modulo Real-Time di LabVIEW 2010 estende la programmazione grafica di LabVIEW a target hardware real-time NI (inclusi i dispositivi RIO) e a selezionato hardware PC di terze parti. In base al target hardware, LabVIEW Real-Time utilizza automaticamente componenti del SO real-time VxWorks standard o ETS per assicurare prestazioni hard real-time deterministiche.

Quando si programma un'applicazione di controllo ad anello chiuso con LabVIEW Real-Time, il Timed Loop offre un modo intuitivo per eseguire periodicamente un sottoinsieme di codice, sincronizzare la velocità del loop con l'hardware, prioritizzare più loop o anche assegnare un loop a un certo core della CPU – tutto con pochi click del mouse. Inoltre, potete scegliere fra clock in kilohertz, megahertz o custom come base di timing del vostro loop.

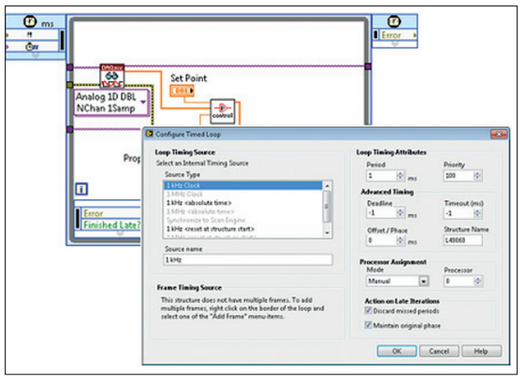

Figura 2. Il Timed Loop di LabVIEW vi permette di configurare intuitivamente il timing di loop, la sincronizzazione con hardware di I/O, la priorità di codice real-time, l'affinità di CPU e altro

SINCRONIZZAZIONE DI LOOP TEMPORIZZATI SU PIÙ TARGET

Potete sfruttare la sincronizzazione IEEE 1588 su dispositivi hardware programmati con LabVIEW Real-Time per fare in modo che una data operazione avvenga pressoché simultaneamente su più sistemi embedded. In base al dispositivo che utilizzate, sono supportate sincronizzazioni IEEE 1588 software sotto il millisecondo, se via software, e/o hardware sotto il microsecondo, se effettuate via hardware.

Per sfruttare la sincronizzazione tra target hardware basata sul software in LabVIEW Real-Time, usate il Timed Loop insieme all'impostazione di sorgente di tipo Absolute Timing.

Dopo la configurazione del Timed Loop, semplicemente una data di inizio e selezionate run per iniziare l'esecuzione del loop su più sistemi contemporaneamente.

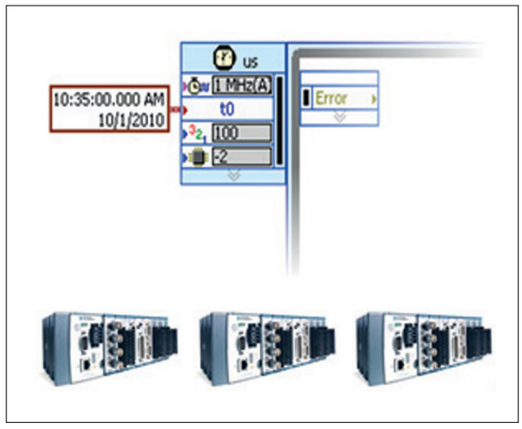

Figura 3. Potete usare la sicronizzazione di tempo IEEE 1588 con il Timed Loop di LabVIEW per iniziare simultaneamente un'operazione su più dispositivi embedded distribuiti (qui un NI CompactRIO)

Uso del toolkit NI Real-Time Execution Trace per un'analisi dettagliata del timing

Usato con il LabVIEW Real-Time Module, il toolkit Real-Time Execution Trace può indicare informazioni dettagliate sul tempo di esecuzione di thread e subVI, priorità ed eredità fra elementi del codice. Potete usare il toolkit non solo per ottimizzare le vostre applicazioni real-time, ma anche per facilitare l'identificazione di problemi potenziali come l'inversione di priorità o condizioni di conflitto.

Esecuzione deterministica di .m file su hardware real-time

Se avete codice su file con estensione .m, potete usare il modulo NI LabVIEW MathScript RT per eseguire deterministicamente tale codice su un sistema embedded che programate con LabVIEW Real-Time.

Essenzialmente, ciò rende lineare e rapido il percorso, un tempo complesso, di esecuzione di file di tipo .m su target hardware.

Combinate semplicemente il vostro .m file con il codice grafico di LabVIEW usando il MathScript Node, quindi trascinate e depositate la vostra applicazione sul target real-time all'interno del progetto LabVIEW.

CARATTERISTICHE DI TIMING INCORPORATE NEL MODULO LABVIEW FPGA

Ottimizzazione di applicazioni FPGA con il single-cycle Timed Loop

Il single-cycle Timed Loop a ciclo singolo nel modulo FPGA di LabVIEW 2010 rimuove i registri dall'interno del loop affinché possiate eseguire in modo ottimale un blocco di codice compatibile.

Ciò rende possibili frequenze di loop fino a 40 MHz per operazioni critiche che devono essere eseguite in modo affidabile a livello hardware. Inoltre, il codice all'interno di un single-cycle Timed Loop richiede meno spazio sull'FPGA, lasciandovi più spazio per altri task.

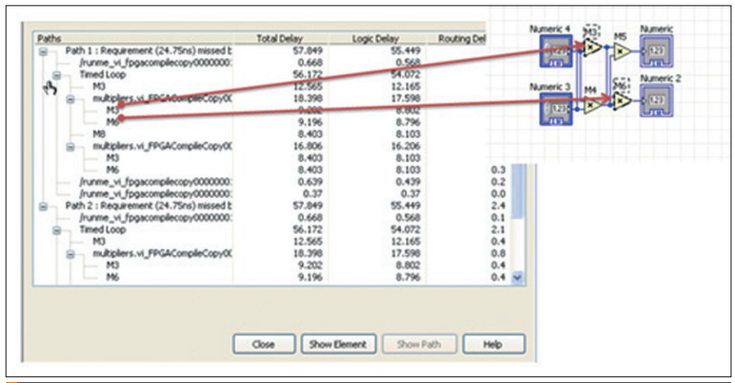

Figura 5. L'evidenziazione dei percorsi critici nel modulo LabVIEW FPGA vi permette di identificare rapidamente i colli di bottiglia nella vostra applicazione FPGA

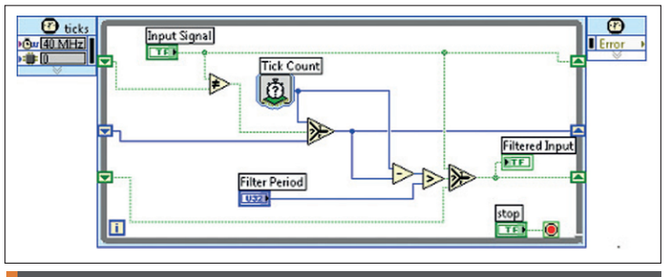

Figura 4. Il single-cycle Timed Loop di LabVIEW FPGA permttette di eseguire una sezione di codice in modo ottimale a frequenze di loop fino a 40 megahertz

Uso del Loop Timer per regolare l'esecuzione di codice FPGA

Eseguire un'operazione periodicamente su un chip FPGA è facile come usare il Loop Timer all'interno di singoli loop sul vostro diagramma a blocchi LabVIEW FPGA.

Usando il Loop Timer, avete la possibilità di controllare il timing di un loop come multiplo del clock hardware con periodi del loop nell'ordine dei nanosecondi, microsecondi o millisecondi. Con una frequenza di clock di default di 40 MHz, il Loop Timer può essere usato per ottenere frequenze di loop in multipli di 25 ns.

SINCRONIZZAZIONE DEL TIMING DI APPLICAZIONI REAL-TIME E FPGA CON INTERRUPT

Con target hardware FPGA selezionati, potete generare interrupt da applicazioni FPGA per notificare eventi a programmi host real-time, come la lettura di dati, il verificarsi di un errore o il completamento di un a task. Sfruttando la funzione Interrupt in LabVIEW FPGA e il metodo Wait on Interrupt in LabVIEW Real-Time, potete sincronizzare il vostro intero sistema embedded. Sono disponibili fino a 32 interrupt.

Identificazione dei colli di bottiglia con evidenziazione dei percorsi critici

L'identificazione e la correzione delle violazioni di timing durante lo sviluppo di applicazioni FPGA può essere tedioso. Per facilitare questo processo, usate il modulo LabVIEW FPGA per evidenziare rapidamente ogni operazione nel percorso critico e saltare alla posizione appropriata nel codice sorgente, dove potete rimediare al problema programmando in modo più efficiente o usando stadi pipeline addizionali (Fig. 5).

CONSIDERATE LABVIEW E NI RIO PER IL VOSTRO PROSSIMO PROTOTIPO O PROGETTO EMBEDDED

L'hardware NI RIO e LabVIEW sono stati usati con successo per fornire un timing stretto e un funzionamento affidabile in migliaia di prototipi e sistemi installati embedded. Quando pianificate il vostro prossimo progetto embedded, considerate i risparmi di tempo e di costi che un hardware pronto all'uso, potente e flessibile può offrire – oltre ai vantaggi di usare la programmazione grafica di LabVIEW, le sue caratteristiche di timing incorporate e centinaia di librerie IP per abbreviare il vostro ciclo di sviluppo.

Note sull'autore

Laureato in ingegneria biomedica al Politecnico di Milano, Alessandro Plantamura lavora in qualità di Technical Marketing Engineer presso National Instruments Italy