Sistemi operativi e strumenti di design per microcontrollori multi e single-core

L'articolo discute brevemente le problematiche legate allo sviluppo di applicativi embedded per dispositivi multicore. Partendo da architetture commerciali basate su FPGA, vengono descritte tecnologie di scheduling innovative, algoritmi di riduzione dello stack e strumenti di design basati su Eclipse utilizzabili per analizzare le caratteristiche temporali di applicativi multithread

Paolo Gai
Amministratore delegato
Evidence

I recenti sviluppi del mercato embedded mostrano la tendenza allo sviluppo e alla realizzazione di sistemi sempre più complessi e allo stesso tempo semplici da usare.
Spesso tali sistemi implementano diverse funzionalità all'interno dello stesso chip, al fine di ridurre il numero globale di centraline in un sistema complesso.
Tale trend è testimoniato ad esempio dagli standard industriali proposti in ambito automotive dal progetto AUTOSAR, che si pone come obiettivo l'integrazione di componenti SW eterogenei all'interno dello stesso sistema.
Da questo scenario complesso sono nati i prodotti di Evidence, e in particolare ERIKA Enterprise, un kernel di dimensione minima per microcontrollori 8, 16, 32 bit, e RT-Druid, uno strumento integrato in Eclipse che permette di guidare lo sviluppo di applicazioni dedicate agevolando la configurazione dell'applicazione e permettendo di guidare lo sviluppatore nella scelta dei migliori parametri (priorità, dimensione dello stack, e così via) da utilizzare per garantire le prestazioni del sistema.

I sistemi multicore e il problema del partizionamento

Una delle ragioni trainanti dello sviluppo di ERIKA Enterprise è stato il supporto per architetture multicore su singolo chip. Le ragioni del passaggio verso sistemi con più processori a bordo sono varie, primo fra tutti il raggiungimento dei limiti fisici delle tecnologie al silicio unito alla necessità di avere bassi consumi: l'utilizzo di soluzioni multicore, di fatto, permette di mantenere il trend di incremento di performance dei dispositivi elettronici senza necessariamente dover aumentare la frequenza di clock (e quindi il consumo) del processore o dover implementare complicate ottimizzazioni HW.
Se da un lato esistono tecnologie e soluzioni industriali che offrono la realizzazione di sistemi con più di un core sullo stesso chip, dall'altro lato non esiste un accordo comune su quale sia la metodologia di sviluppo software più adeguata per questo tipo di sistemi. Nel caso di sistemi multicore, il progettista deve risolvere una serie di problematiche quali:
- la condivisione di codice e dati tra i vari processori;
- la scrittura ottimizzata di codice che permetta di poter spo-

stare codice da una CPU a un'altra senza particolare sforzo;
- il partizionamento delle varie attività da eseguire sui vari processori, fondamentale in sistemi eterogenei che non permettono la migrazione di task tra i vari processori.

Per quanto riguarda il problema del partizionamento, è importante avere una metrica che permetta di guidare il progettista nella scelta di quali funzionalità debbano essere assegnate ai vari processori, in modo da garantire una bassa occupazione di stack (nel caso di piccoli microcontrollori), tempi di risposta certi, e sfruttamento adeguato della potenza di calcolo a disposizione.

Inoltre, bisogna tenere conto di due ulteriori aspetti, che sono la necessità di recuperare il codice esistente scritto per sistemi single core, e la modifica indolore del partizionamento del codice anche in una fase avanzata del design.

L'adozione dei sistemi multicore richiederà strumenti che da un lato permettano di strutturare e implementare future applicazioni, dall'altro permettano il partizionamento del sistema sulle varie CPU, fornendo delle metriche per poter giudicare la bontà delle scelte effettuate e fornendo utili consigli per il miglioramento delle performance applicative.

Evidence Srl

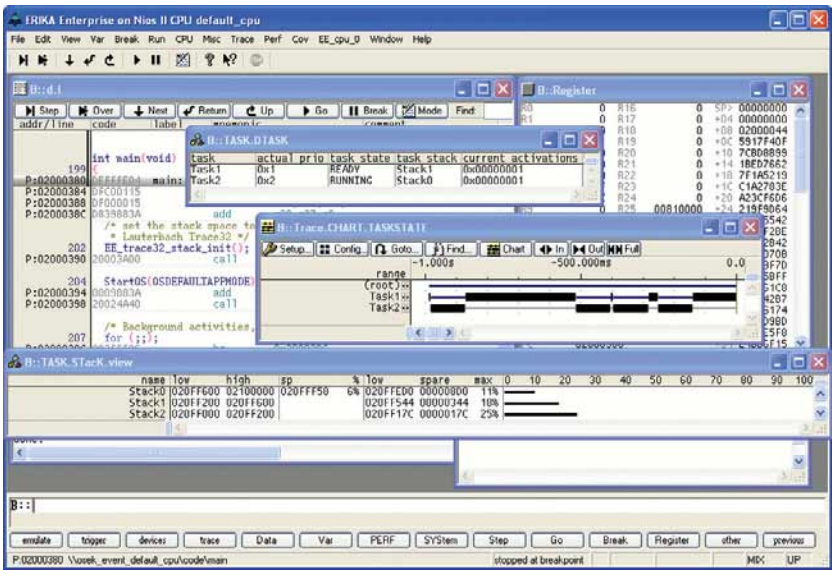
Evidence Srl è una azienda spin-off del ReTiS Lab della Scuola Superiore Sant'Anna di Pisa.
La mission di Evidence è fornire soluzioni innovative per il design e lo sviluppo di sistemi real-time dedicati, con un focus specifico per le architetture multi-core.
Evidence ha sviluppato ERIKA Enterprise, un sistema operativo minimale per microcontrollori 8, 16, 32 bit che supporta in modo specifico architetture multi-core, e RT-Druid, uno strumento di configurazione e design architetturale di applicazioni dedicate basato su Eclipse. Evidence fornisce inoltre servizi qualificati per l'utilizzo di Linux embedded all'interno di prodotti dedicati.

Il sistema operativo ERIKA Enterprise

Il sistema operativo ERIKA Enterprise è stato sviluppato da Evidence con lo scopo di realizzare un sistema operativo minimale per microcontrollori capace di offrire un supporto alla multiprogrammazione garantendo una minima occupazione di memoria.

ERIKA Enterprise implementa una API simile a quella dello standard OSEK/VDX utilizzato in ambito automotive, permettendo il riuso di tecniche di programmazione consolidate nel mondo industriale, come lo scheduling prioritario di tipo sia preemptive che non preemptive, e il supporto per primitive di sincronizzazione e di comunicazione.

ERIKA Enterprise può essere considerato uno dei primi prodotti che agevolano la scrittura di codice per applicazioni multicore. ERIKA Enterprise è al momento disponibile per le architetture multicore Janus e Altera Nios II (vedi riquadro), nonché per altri sistemi single core basati su ARM7, PPC, dsPIC, AVR, e altri. Tra le soluzioni adottate per gestire il supporto multicore, possiamo citare:

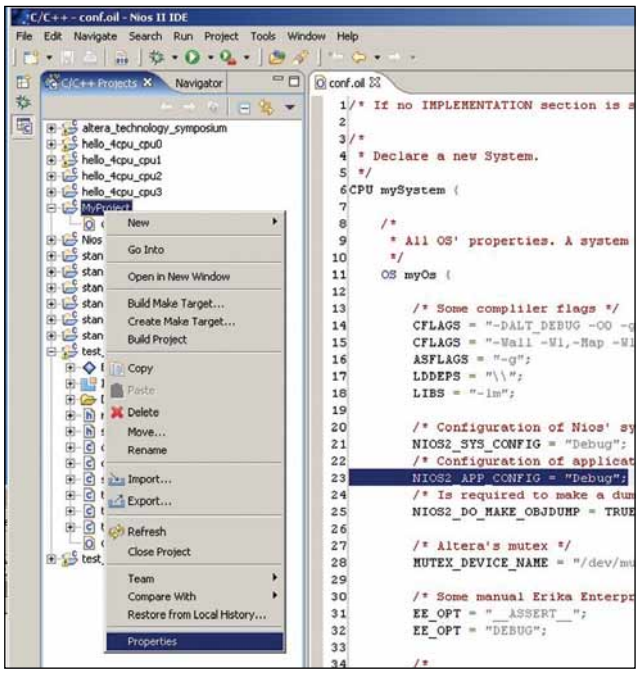


Questa immagine mostra l'integrazione di ERIKA Enterprise con il debugger commerciale Lauterbach Trace32. Tramite il supporto ORTI, il debugger riesce a visualizzare informazioni utili sullo stato del sistema, quali ad esempio i Task esistenti nell'applicativo (finestra "TASK.DTASK"), il loro stato (finestra "TRACE.CHART.TASKSTATE"), e la occupazione di memoria per ciascun stack (finestra "TASK.STACK.VIEW")

Le architetture Multicore supportate da ERIKA Enterprise

La prima architettura storicamente supportata da ERIKA Enterprise è stata il sistema Janus, un progetto innovativo italiano sviluppato all'inizio degli anni 2000 da PARADES, ST Microelectronics e Magneti Marelli. L'idea di base è che uno dei colli di bottiglia è rappresentato dai tempi di accesso alla memoria flash on-chip. Il sistema Janus prevedeva un partizionamento in due parti delle memorie, con l'inserimento di una crossbar switch e di una seconda CPU, permettendo in questo modo un raddoppio delle prestazioni a fronte di un incremento di area inferiore al 10%.

La seconda architettura multicore supportata da ERIKA Enterprise è stato il soft-core Nios II di Altera. La piattaforma Nios II di Altera permette la possibilità di ospitare più processori all'interno di una singola FPGA, collegando assieme processori e memorie con una crossbar switch all'interno di uno strumento chiamato SOPCBuilder.
In entrambe i casi si tratta di sistemi multicore asimmetrici con condivisione di memoria, in cui le CPU sono simili ma non identiche. Le CPU possono scambiarsi notifiche asincrone tramite interrupt interprocessore e operazioni atomiche su bus. I vari task del sistema operativo vengono allocati staticamente su ciascuna CPU.



Questa immagine mostra RT-Druid integrato all'interno della piattaforma Eclipse fornita con Altera Nios II IDE. Di fatto l'utente, utilizzando la stessa interfaccia grafica, e lo stesso modo di operare, ha la possibilità di utilizzare le funzionalità aggiuntive come la generazione di codice per ERIKA Enterprise e l'analisi di schedulabilità

- un algoritmo di scheduling dedicato ai sistemi multiprocessore asimmetrici, capace di garantire condivisione dello stack, assenza di deadlock, gestione delle risorse condivise tramite uso di protocolli non bloccanti (IPCP) e spin-lock;
- una API del sistema operativo che rimane identica nel passaggio da core singolo a multicore;
- una gestione dei protocolli di condivisione dei dati tramite l'utilizzo trasparente di spinlock;
- una gestione automatica della condivisione di memoria senza necessità di dover cambiare il codice esistente, anche in presenza di CPU con diverse mappe di memoria e/o con necessità di disabilitazione della cache;
- supporto di un linguaggio di configurazione estensione dello standard OSEK OIL che permette la specifica del partizionamento dei vari task ai vari processori disponibili, con generazione automatica dei makefile, del codice di configurazione del kernel, e dei file eseguibili;
- generazione di script di debug per debugger multicore basati su Lauterbach Trace32. Utilizzando il supporto multiprocessore fornito da Lauterbach, è possibile effettuare il debug contemporaneo dei vari core permettendo un completo controllo del sistema;
- generazione di informazioni di kernel awareness secondo lo standard OSEK ORTI, permettendo a debugger come Lauterbach Trace32 di ottenere informazioni sullo stato delle strutture dati del kernel, come i task attualmente in esecuzione, lo stato delle risorse, e la quantità di stack massima utilizzata.

I principali vantaggi e caratteristiche nell'utilizzo di ERIKA Enterprise sono:
- disponibilità di un sistema operativo predicibile con caratteristiche real-time per applicazioni di controllo su piccoli microcontrollori anche multicore;
- disponibilità di algoritmi di scheduling tradizionali (come scheduling a priorità fissa, preemptive e non preemptive);
- disponibilità di algoritmi di scheduling innovativi come

Le tecniche di riduzione di memoria su single core

I sistemi a microcontrollore sono utilizzati in applicazioni di controllo e attuazione. Tali applicazioni vengono spesso implementate "ad eventi", con un unico grande ciclo infinito che gestisce il comportamento di una serie di eventi asincroni. Questa tecnica di gestione delle applicazioni ha alcuni vantaggi, quali il ridotto utilizzo di memoria e la semplicità di programmazione, ma anche alcuni svantaggi, in quanto per applicazioni complesse la programmazione risulta essere macchinosa, e, mancando un sistema operativo, i singoli eventi sono di fatto attività non interrompibili che in generale aumentano il tempo di risposta delle attività ad alta frequenza.

I sistemi operativi preemptive, dall'altro lato, permettono la gestione di attività che possono essere interrotte in presenza di attività urgenti (preemption). Se da un lato i sistemi operativi permettono di migliorare i tempi di risposta degli applicativi scritti "ad eventi", dall'altro hanno come effetto collaterale un maggior utilizzo di memoria RAM, in quanto tipicamente ogni task ha necessità di uno stack privato. Nei sistemi a microcontrollore questo può essere un problema, in quanto la memoria RAM è spesso una delle risorse più scarse.

Per ridurre questo effetto, ERIKA Enterprise implementa delle tecniche che permettono di ridurre lo spazio di stack utilizzato dai vari task, mantenendo allo stesso tempo prestazioni simili ai kernel preemptive. Le tecniche utilizzate da ERIKA Enterprise per permettere una riduzione di stack sono:

- Utilizzo di primitive non bloccanti e Immediate Priorità Ceiling

La API esportata da ERIKA Enterprise sfrutta il protocollo Immediate Priority Ceiling (IPCP). Il comportamento di base di IPCP è quello di modificare le priorità dei task che entrano in sezione critica, garantendo che l'accesso ai semafori sia non bloccante. La tecnica garantisce l'assenza del fenomeno della Priority Inversion, e garantisce l'assenza di Deadlock. La condivisione dello stack è una conseguenza delle proprietà di IPCP, in quanto i vari task non si possono mai bloccare nell'accesso alle risorse condivise.

- Task "one-shot"

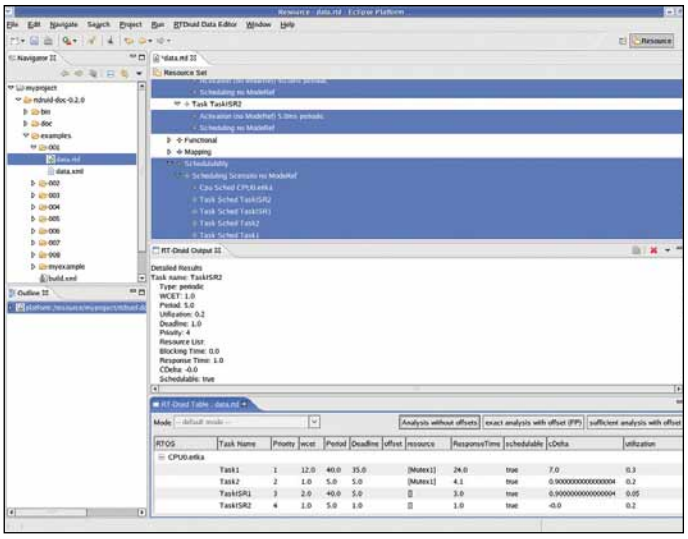
Per ottenere condivisione di stack tra vari task non basta l'utilizzo del protocollo IPCP. I vari task nel sistema devono essere scritti seguendo una tecnica chiamata "one-shot": l'idea di base è che un task, al termine della sua attività, non si blocca in attesa di un evento di risveglio, ma termina la sua esecuzione liberando lo stack da lui occupato.

- Preemption Threshold

ERIKA Enterprise utilizza una tecnica chiamata preemption threshold, che permette di limitare la preemption tra i vari task esistenti del sistema. Di fatto i task in esecuzione innalzano la propria priorità, limitando in questo modo la preemption di altri task a priorità intermedia. L'idea intuitiva è la seguente: se da un lato la preemption tra i task migliora le prestazioni temporali del sistema, dall'altro comporta una maggiore occupazione di memoria rispetto a un sistema non-preemptive. Tramite i preemption thresholds si limita la preemption tra i task, limitando così l'occupazione di memoria, mantenendo comunque la predicibilità e i tempi di risposta richiesti dall'applicativo.

Il punto critico nel design del sistema diventa pertanto legato alla specifica delle priorità dei task. Evidence affronta queste problematiche tramite lo strumento di sviluppo RT-Druid. L'idea che sta dietro allo strumento RT-Druid è quella di fornire feedback al designer utilizzando tecniche di analisi di schedulabilità. Tramite l'analisi di schedulabilità, partendo dalla stima dei tempi di esecuzione, RT-Druid è in grado di fornire informazioni sui tempi di risposta dei vari task, il massimo utilizzo dello stack, ed è in grado di fornire un feedback al progettista per capire quanto tempo di esecuzione è disponibile a ciascuna frequenza di lavoro, eventualmente specificando, nel caso di tempi di risposta non soddisfacenti, di quanto ciascun task a una certa frequenza debba essere ridotto per riportare il sistema entro le specifiche.

immediate priority ceiling e preemption threshold per la riduzione dello stack;
- possibilità di strutturare una applicazione come un insieme di attività concorrenti anche su piccoli microcontrollori senza dover richiedere una elevata quantità di stack;
- configurazione dell'impronta di sistema operativo con eliminazione delle parti di codice non utilizzate;
- supporto per le API standard proposte dal consorzio OSEK/VDX in ambito automotive. Sinteticamente, tra i dati relativi alle performance di ERIKA Enterprise per l'architettura Altera Nios II possiamo citare:
- le dimensioni tipiche per i sistemi minimali: 1.924 byte per sistemi su singola CPU e 4.100 byte per ciascuna CPU nei sistemi multicore;
- i tempi di esecuzione delle principali primitive (considerando un sistema Nios II/s biprocessore a memoria condivisa a 50



Questa immagine mostra RT-Druid (in ambiente Linux) e il corrispondente utilizzo di algoritmi di analisi di schedulabilità per applicazioni embedded. Nella finestra "data.rtd" (in alto) è possibile visualizzare una rappresentazione ad albero del sistema. Nella finestra "RT-Druid output" (al centro) viene visualizzato un report dell'attività di analisi, mentre nella finestra "RT-Druid Table" (in basso) è presente una rappresentazione tabellare del sistema che indica, oltre alle priorità e i tempi di esecuzione dei vari task, anche di quanto tempo ciascun task ha ancora a disposizione prima di rendere il sistema non schedulabile (colonna "CDelta")

MHz): da 2 a 11 microsecondi. I recenti trend industriali stanno portando a un aumento della complessità degli applicativi che vengono eseguiti su microcontrollori anche di fascia bassa.
La complessità e l'integrazione di diverse funzionalità all'interno dello stesso sistema portano come naturale conseguenza la necessità di separare le varie parti del sistema, e tale separazione è agevolata dall'uso di Sistemi Operativi, che gestiscono in modo naturale la concorrenza tra diverse attività a frequenza diversa.
L'utilizzo di sistemi operativi per piccoli dispositivi comporta un overhead trascurabile in termini di tempo e in termini di spazio, grazie alla possibilità di utilizzare primitive non bloccanti, preemption threshold e condivisione di stack che riducono la quantità di memoria necessaria per implementare una applicazione multithread su di un sistema di piccole dimensioni.
L'utilizzo di sistemi operativi diventa sempre più importante quando si considerano le nuove architetture multicore, in quanto il sistema operativo permette in modo naturale di gestire la concorrenza esistente all'interno dell'architettura, risolvendo attraverso opportuni strumenti di configurazione le varie problematiche di gestione e stima delle performance applicative.
ERIKA Enterprise e RT-Druid, prodotti da Evidence, forniscono un ambiente integrato che si muove nell'ottica di semplificare l'utilizzo di sistemi operativi per piattaforme single e multi core di piccole dimensioni, fornendo idonei strumenti per lo sviluppo e la stima delle performance delle moderne applicazioni per microcontrollore.

Evidence
readerservice.it n. 29