

Corso di laurea in Ingegneria Elettronica

Tesi di laurea

## **LISARM: embedded ARM platform design and optimization**

Relatori:

Prof. Guido Masera

Ing. Maurizio Martina

Ing. Fabrizio Vacca

Candidato:

Carlo Ceriani

Aprile 2007

La diffusione dei dispositivi elettronici, in molti aspetti della vita comune, ha cambiato profondamente gli assetti della produzione industriale e le tecnologie che stanno alla base delle applicazioni richieste dal mercato. Sebbene le tecnologie *System-on-Chip* abbiano dato notevoli possibilità di integrazione di componenti elettronici anche molto eterogenei su un singolo *die*, i tempi di sviluppo delle tecnologie *hardwired* e gli stringenti vincoli imposti dalle logiche di ritorno economico hanno condotto alla ricerca di nuovi approcci. Una delle tecniche più utilizzate attualmente è quella dell'*hardware-software partitioning*, che consiste nel suddividere la complessità dell'applicazione da sviluppare su due livelli differenti: la progettazione di un sistema integrato programmabile potente e flessibile e la creazione di algoritmi complessi in grado di soddisfare le esigenze dell'applicazione. Altre specifiche, non meno importanti, riguardano il basso consumo ed il ridotto ingombro per i dispositivi portatili, che devono garantire grande autonomia delle batterie a fronte di sempre maggiori prestazioni richieste dall'utente. In questo ambito tecnologico le piattaforme programmabili basate su microprocessore sono tra le più diffuse e l'ARM7TDMI rappresenta uno dei processori di maggior successo, grazie alle sue notevoli prestazioni ed alle sue caratteristiche di basso consumo. L'uso di processori *embedded* non è l'unica soluzione tecnologica attualmente sfruttata; sebbene le architetture disponibili sul mercato forniscano molte delle caratteristiche richieste dai produttori del

settore, talvolta esse non sono sufficientemente “su misura” per talune applicazioni critiche o hanno una struttura troppo complessa che va a discapito di occupazione di area e consumo di potenza. L’alternativa è rappresentata dagli *Application Specific Instruction-set Processor* (ASIP), ovvero da processori progettati per specifiche applicazioni che forniscono un set di istruzioni costruito in base alle esigenze degli algoritmi software che devono essere eseguiti. Per la progettazione delle architetture programmabili si utilizzano ambienti di sviluppo software che consentono di descrivere il set di istruzioni in maniera flessibile e riutilizzabile, mediante la stesura di codice in un *Architecture Description Language* (ADL) quale *LISA 2.0*. *LISATek Toolsuite* e *Language for Instruction-Set Architecture* consentono di descrivere il funzionamento di un processore in tutti i suoi aspetti, sia funzionali che temporali, integrando tecnologie attuali quali *pipeline* e *cache*, permettendo di ottenere non solo una descrizione dell’hardware mediante i linguaggi VHDL e Verilog, ma anche un potente simulatore ed i tool necessari per lo sviluppo del software dedicato.

Scopo del presente lavoro di tesi è stato quello di esplorare le possibilità offerte dal linguaggio LISA e dagli strumenti della toolsuite LISATek per la progettazione di una piattaforma programmabile basata sul processore ARM7, la cui ampia documentazione, frutto delle numerose applicazioni basate sul processore medesimo, ha consentito di analizzarne a fondo tutte le caratteristiche da riprodurre nel modello.

Per la costruzione del modello si è essenzialmente fatto uso di file scritti in codice LISA, descrivendo anzitutto le risorse di cui il processore è dotato, registri e memorie in primo luogo. In una prima fase si sono utilizzati i meccanismi offerti dal linguaggio per la definizione completa della sintassi assembly e della complessa codifica del set di istruzioni dell’ARM7. Mediante l’uso del linguaggio C standard, all’interno di specifiche sezioni dei file, il linguaggio LISA ha consentito di definire il comportamento dell’architettura per l’esecuzione delle istruzioni ARM, permettendo di ottenere un primo modello *instruction-accurate* del processore di riferimento. In tale livello di astrazione non si tiene in considerazione la variabile tempo nell’esecuzione delle istruzioni: ciò consente di concentrare l’attenzione sugli aspetti funzionali dell’architettura nel trattamento dei dati designati come operandi ed altre operazioni che controllano il comportamento dell’architettura; LISATek, mediante questa prima astrazione, consente di ottenere la struttura del decodificatore delle istruzioni e del *datapath*. In una successiva fase si sono sfruttate le potenzialità offerte dal linguaggio per descrivere il funzionamento della pipeline del processore, assegnando ai tre stadi (*fetch*, *decode* ed *execute*) le operazioni LISA già create nella prima fase ed inserendo nel codice una serie di istruzioni per la gestione degli eventi e dello *scheduling* di tali operazioni. Questa seconda fase ha condotto alla definizione

del modello *cycle-accurate* del processore, ovvero di un modello che rispetta tutte le tempistiche dell'esecuzione in pipeline dell'ARM7; mediante questa descrizione LISATek genera automaticamente il controllore della pipeline con tutti i segnali necessari al funzionamento dei vari stadi. Tramite successivi raffinamenti del modello, si è definita un'architettura tale da garantire il mantenimento di una gerarchia simile a quella dell'ARM7, in maniera tale che le parti importanti del datapath, come il moltiplicatore 32x8, la ALU ed il *barrel shifter*, potessero risultare come componenti a sé nella descrizione hardware generata e si prestassero a successive ottimizzazioni architetturali.

Simulatore, *assembler*, *linker* e *disassembler* si sono rivelati strumenti di fondamentale importanza nello sviluppo del modello, avendo consentito di verificare passo per passo il corretto funzionamento dell'architettura descritta. Fondamentale ruolo ha rivestito l'*HDL Generator*, che ha permesso di affinare lo stile di programmazione per ottenere una descrizione del modello molto vicina al livello hardware.

Il modello ottenuto (*LISARM*) accetta tutte le istruzioni codificate per il processore ARM7TDMI e le esegue fornendo gli stessi risultati e con le stesse tempistiche di un ARM7, in ragione di cicli macchina, e manifestandone lo stesso comportamento esterno. A causa di alcuni limiti intrinseci della tool suite LISATek, alcune delle funzionalità dell'assembler originale ARM e dell'interfaccia con la memoria dell'ARM7TDMI non sono state adeguatamente riprodotte. Al fine di poter sfruttare gli assai diffusi strumenti per lo sviluppo di software disponibili per ARM, si è provveduto alla creazione di una coppia di tool per l'adattamento della toolchain, ovvero un *pre-assembler* ed un *post-disassembler* scritti in linguaggio C; tali strumenti sono in grado di adattare la sintassi assembly del processore ARM7 a quella del modello ottenuto. Per rendere completamente compatibile l'interfaccia di memoria del modello con quella dell'ARM, infine, è stato prodotto un *wrapper* in VHDL, capace di garantire l'indirizzamento al singolo byte tipico della famiglia di processori ARM, altrimenti non consentita dal modello ottenuto con LISATek.

Nonostante i limiti succitati, la tool suite LISATek ha rivelato tutte le sue potenzialità per la creazione di ASIP, consentendo di ottenere un modello perfettamente funzionante attraverso un omogeneo insieme di file di descrizione, da cui è possibile ottenere simulatore, descrizione HDL dell'hardware e toolchain completa, ma soprattutto eliminando la necessità di modificare manualmente tutti questi oggetti ad ogni parziale revisione del modello.

Sviluppi futuri del lavoro svolto potranno riguardare evoluzioni del modello per adattarlo ad altre applicazioni specifiche, mediante opportune modifiche all'attuale set di istruzioni, eseguite rivedendo l'unica fonte rappresentata dal codice LISA.