LA COMPRESSIONE MULTIMEDIALE CON PLATFORM FPGA

Robert D. Turney, Paul Schumacher
Xilinx Research Labs, Xilinx Inc.

La prossima generazione di algoritmi di compressione multimediale è emersa sotto forma degli standard JPEG2000 e MPEG-4.
Recenti sviluppi sulla densità delle porte logiche e delle funzioni integrate su silicio hanno schiuso nuove interessanti possibilità di progettazione per coloro che si occupano di elaborazione multimediale. L'articolo analizza i requisiti dei sistemi di compressione multimediale e valuta la complessità di progettazione dell'elaborazione della compressione multimediale. L'articolo approfondisce quindi sul flusso di progettazione necessario a presentare il livello di astrazione richiesto dalle implementazioni di elaborazione multimediale. In ultima analisi una panoramica degli standard di compressione emergenti di JPEG2000 e MPEG-4 riferita alla mappatura in tecnologia FPGA per sistemi multimediali in tempo reale.

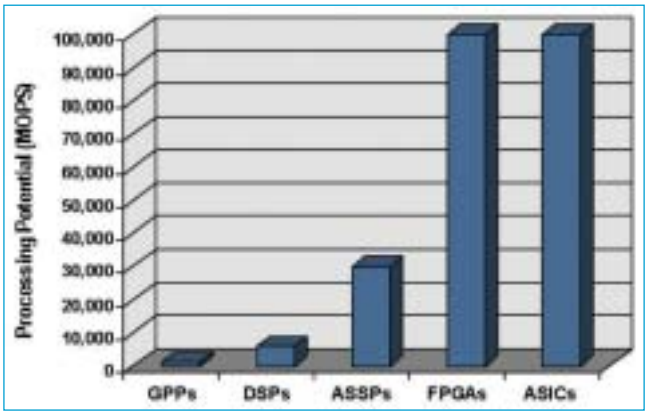

Fig. 1 - MOPS di calcolo per dispositivi al silicio

Requisiti per la compressione multimediale

Le comunicazioni digitali multimediali si stanno rapidamente affermando con un'ampia base di mercati dalle applicazioni wireless, al trattamento delle immagini medicali, al cinema digitale. La tecnologia di compressione multimediale riveste un ruolo chiave e determinante per far sì che la comunicazione raggiunga un utilizzo efficiente delle risorse di memorizzazione e di banda.
Il lavoro iniziale in quest'area, ovvero JPEG, MPEG-1 e MPEG-2, ha avuto luogo dalla fine degli anni '80 alla metà degli anni '90. Questo lavoro di standardizzazione della International Standard Organization (ISO) è continuato con il completamento degli standard JPEG 2000 e MPEG-4.
Oggi l'operato dell'ISO continua sotto forma di esplorazione di tecnologie video tridimensionali e scalabili per nuovi standard di base. Ci si potrebbe chiedere: perché la tecnologia video e audio digitale continua ad evolvere? La risposta giace nell'infrastruttura computazionale in base alla quale sono sviluppati gli algoritmi. Non appena più cicli di iterazione, memoria e banda di memoria sono messi a disposizione attraverso nuove tecnologie su silicio, lo spazio algoritmico di compressione è reso accessibile e delle alternative diventano fattibili.
Il lavoro originario nella compressione digitale dell'immagine è stato cristallizzato negli standard JPEG e MPEG-1. I fondamenti della compressione digitale dell'immagine sono fissati in tre principali algoritmi, quello della trasformata, quantizzazione e codifica di sorgente. L'uso più ampiamente diffuso dello standard JPEG è la forma "baseline" dello standard che utilizza una trasformata discreta coseno (DCT) 8x8 per la trasformata. La fase di quantizzazione consiste nell'azzeramento dei coefficienti DCT che sono giudicati non visivamente significativi. Se la soglia di quantizzazione è troppo severa, appariranno blocchi artefatti nell'immagine risultante decompressa.
Per il JPEG ciò tipicamente accade per un rapporto di compressione superiore a 30 a 1. La tecnica a codifica di sorgente del JPEG baseline è la codifica di Huffman ed è un'operazione priva di perdite.
Per le immagini fisse il JPEG baseline catturato via software in genere fornisce un tempo di risposta adeguato per la compressione e la decompressione. I requisiti per il tempo reale di 30 trame al secondo entrano in gioco quando usiamo il motion JPEG per una sequenza di immagini come nel trattamento delle immagini medicali o nella sorveglianza video. Per mantenere la nostra velocità di trama abbiamo ora bisogno di essere in grado di calcolare la DCT, la quantizzazione e la codifica Huffman in 33 millisecondi.
I lavori sugli standard JPEG sono continuati con l'adozione dello standard

JPEG2000. Lo standard JPEG 2000 fa uso di una tecnologia completamente diversa con trasformata wavelet discreta (DWT), codifica del piano dei bit e codifica aritmetica quali principali calcoli algoritmici. JPEG2000 non possiede la caratteristica struttura a blocchi del JPEG a elevati fattori di compressione ma soffre di un effetto "mosquito" attorno agli angoli.
I rapporti di compressione JPEG 2000 raggiungono tipicamente anche 50 a 1 per una qualità equivalente a 30 a 1 per il JPEG. Esistono diversi interessanti attributi di JPEG2000, alcuni dei quali includono:

- Eccellenti prestazioni a basso bitrate: un graduale degrado verso velocità di trasferimento di bit inferiori.
- Flusso di codice immerso: JPEG2000 fornisce un flusso di codice "immerso", cioè aggiungere più bit al flusso di codice dà una superiore qualità dell'immagine.
- Resilienza all'errore: è stato aggiunto a JPE200 un grosso numero di funzionalità resilienti all'errore.
- Compressione con e senza perdite: JPEG2000 può ottenere entrambi i tipi di conversone.
- Architettura non soggetta a licenze: la Parte I dello standard JPEG2000 è gratuita per l'implementazione da parte di chiunque senza canoni o diritti d'autore.

Per sequenze motion JPEG2000 in tempo reale il calcolo della DWT, del codificatore di bit e del codificatore aritmetico è più complesso in termini di MOPS, dimensioni della memoria e banda di memoria.
La standardizzazione della compressione video è portata avanti nell'International Telecommunication Union (ITU) e nell'ISO MPEG. Questi standard estendono gli algoritmi intratrama per includere la ridondanza temporale di trama mantenendo al contempo la DCT basata su blocchi per la trasformata e la codifica a lunghezza variabile per la codifica di sorgente.
Si fa riferimento a questo insieme di standard da MPEG-1, MPEG-2 (H.262), MPEG-4 versione 2, e MPEG-4 Parte 10 Advanced Video Codec (AVC, H.264), come compressione video basata su DCT con stima e compensazione del moto. Nella compressione video si desidera un bit rate costante e lo si usa come cifra di merito al posto del rapporto di compressione. I codec video sono sviluppati e quindi misurati per mezzo delle curve di distorsione della velocità che tracciano il rapporto segnale rumore di picco (PSNR) verso il bit rate.
Lo scopo dello sviluppo del codec video è di ridurre la velocità di trasferimento dei bit mantenendo allo stesso tempo la qualità.
In fase di sviluppo di un video codec occorre tenere in considerazione la complessità per il calcolo, la memoria e la banda di memoria. I codec MPEG sono progettati con una complessità di un ordine di grandezza superiore nell'encoder rispetto al decoder e di conseguenza sono asimmetrici nei confronti della complessità. Questo progetto intenzionalmente asimmetrico doveva facilitare i decoder a basso costo nei primi progetti di codec.
MPEG-4 si sta espandendo verso spazi di applicazione che includono ambienti di comunicazione a due vie ed è emerso il problema della codifica in tempo reale di MPEG-4.
Per l'implementazione di standard di compressione multimediale in tempo reale si vorrebbero esplorare i requisiti rispetto a MOPS, ingombro della memoria e banda di memoria. Se si è in grado di fare ciò é possibile allora considerare la mappatura nello spazio del dispositivo su silicio per la propria applicazione.
Se si considerano i MOPS di calcolo, la figura 1 fornisce una panoramica delle scelte che abbiamo oggi. Per molte applicazioni DSP, esiste in genere un grande numero di opzioni di implementazione, alcune ad uso universale e altre specifiche per un'applicazione.
Al limite superiore della complessità computazionale vi sono le applicazioni multimediali quali l'elaborazione video e audio. In genere un singolo requisito di elaborazione può essere soddisfatto con uno dei processori ma la maggior parte dei sistemi implicano in alcuni casi una catena di elaborazione video con più canali.
Come si vedrà nella prossima sezione, anche per un profilo relativamente semplice a risoluzione CIF, la codifica MPEG-4 può superare i 7000 MOPS senza prestare particolare attenzione all'algoritmo impiegato per la stima del moto.
Mentre il potenziale di elaborazione può essere un aspetto molto importante per il confronto fra opzioni di piattaforme, esistono molte altre caratteristiche di progetto che forniscono un quadro più completo.
La tabella 1 elenca alcuni di questi altri aspetti da considerare in un processo di progettazione, che includono: sforzo di progettazione, flessibilità dell'architettura di design, flessibilità in funzionamento, velocità di punta ed efficienza energetica. Sebbene i processori general purpose (GPP), i processori digitali di segnale (DSP) e i processori di segnali per applicazioni specifiche (ASSP) offrano con un basso sforzo di progettazione una piattaforma per la cattura dell'algoritmo, la loro flessibilità e i potenziali di elaborazione possono essere insufficienti

	GPP	DSP	ASSP	FPGA	ASIC
Sforzo di progettazione	+++	++	+++	-	--
Flessibilità dell'Architettura	--	--	--	++	+++
Flessibilità in funzionamento	++	+	--	+++	--
Velocità di punta	--	-	+++	++	+++
Efficienza energetica	-	+	+++	+	+++

Tabella 1 - Confronto fra piattaforme multimediali

per molte applicazioni embedded di alta fascia.
Gli ASIC d'altra parte forniscono un potenziale di elaborazione e un'efficienza energetica eccellenti, ma i costi diretti per NRE sono diventati sempre più proibitivi a meno che non siano previsti volumi estremamente alti.
Esiste un potenziale di elaborazione significativamente alto in una piattaforma di co-design hardware/software quale un field programmable gate array (FPGA) in cui può essere sfruttato l'accesso simultaneo e può essere creata una piattaforma computazionale programmabile di alta fascia.
Come si vede dalla figura 1, se il potenziale di elaborazione in un FPGA viene imbrigliato, diventa una soluzione di alta fascia molto appetibile.

Complessità di progettazione per la compressione multimediale

Per comprendere la complessità di progettazione per la compressione rispetto allo standard MPEG-4 occorre tenere conto che diversi punti di operazione sono definiti attraverso un processo di definizione di profili e di livelli. I profili in MPEG definiscono la tecnologia da usare come DCT o la codifica di sorgente a lunghezza variabile. I livelli in MPEG definiscono quanto di un parametro debba essere elaborato per ottenere la conformità quali i requisiti in termini di MacroBlocchi per secondo. L'altro importante concetto sulla compressione da comprendere riguardo l'attività sugli standard è legato alla conformità. Strettamente parlando i decoder vengono specificati mentre gli encoder devono generare un flusso di bit conforme, ma i loro requisiti non sono specificati nello standard. In termini di una quantifica dei MOPS di calcolo ciò significa che vi sono diversi punti operativi che sono definiti in relazione a quale profilo e livello.
L' applicazione meglio corrisponde. Oltre a ciò vi sono diversi compromessi di progetto per gli encoder che possono essere fatti, e che sono specifici per un'applicazione.
Sebbene gli encoder MPEG siano in ordine di grandezza più complessi, essi hanno il vantaggio che necessitano solo di creare un singolo flusso di bit conforme, mentre i decoder MPEG devono essere conformi al profilo e al livello per cui sono stati progettati. La figura 2 mostra i MOPS di calcolo relativi per profili e livelli MPEG-4 che attraversano due ordini di grandezza all'interno dello standard MPEG-4.
I requisiti di memoria di picco e i requisiti di banda di memoria sono anch'essi dipendenti dal profilo e dal livello. Ad esempio nel caso degli encoder MPEG-4, requisiti tipici di memoria sono da 3 a 10 megabyte mentre per i decoder sono da 1 a 3 megabyte.
Requisiti aggiuntivi di banda di memoria dipendono dall'architettura della specifica implementazione per la vostra applicazione.
Motion JPEG2000 presenta inoltre una vasta gamma di requisiti che dipendono dall'applicazione finale. Un'applicazione di sorveglianza video per un flusso video da 640*480 a 30 Frame/s richiede approssimativamente 4200 MOPS; il trattamento delle immagini medicali da 1024*1024 a 60 Frame/s con codifica senza perdite può richiedere 29000 MOPS; e il cinema digitale con una dimensione di trama di 4096*2048 a 24 Frame/s richiede 93000 MOPS, tutti usando JPEG2000.
Per complicare il discorso l'analisi dei requisiti di calcolo, l'ingombro della memoria e la banda di memoria dipendono dall'immagine e dalla sequenza. Nel caso sia di JPEG, sia di MPEG i MOPS di calcolo effettivi per trama sono variabili nel tempo in relazione all'applicazione. Per MPEG intra trama le risorse di calcolo svolgono operazioni completamente diverse rispetto al caso in cui vengano calcolate operazioni per il movimento inter trama.
Ciò dovrebbe garantire un'architettura che implica qualche grado di scalabilità per gestire le variazioni in complessità. Un'importante parametro per i codec video è la capacità di mantenere la vostra velocità di trama.
Di conseguenza non è di consolazione il fatto che é architettura possa eseguire i MOPS medi ma piuttosto che debba supportare i vostri MOPS di picco previsti per lo scenario applicativo nel caso peggiore.

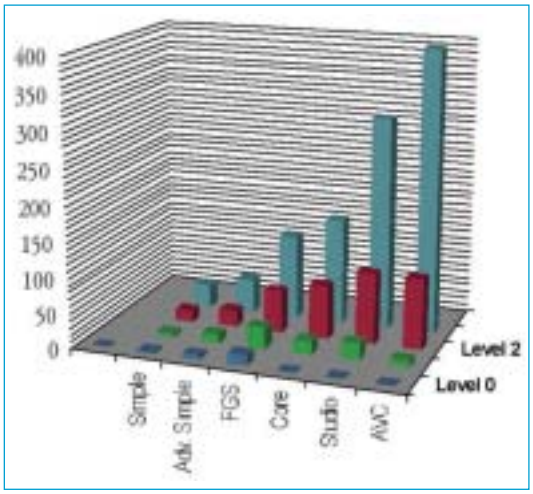

Fig. 2 - MOPS in MPEG-4. I MOPS sono normalizzati a 15000 per gli encoder, a 700 per i decoder

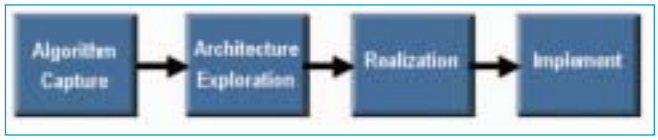

Fig. 3 - Flusso Strumentale Multimediale Proposto

Flusso di progettazione per la compressione multimediale

Per implementare efficacemente i sistemi di compressione multimediale occorre incominciare con il codice degli standard che è scritto in linguaggio C. Anche se non si sta seguendo uno degli standard di compressione, esprimere il progetto iniziale in codice C è vantaggioso dal momento che lo sviluppatore dell'algoritmo esprime la

funzionalità e fa girare i test case. Ciò che non contiene il codice C è un modo ottimale di esprimere l'encoder o il decoder in un modo tale che può essere ottenuta l'operazione in tempo reale. Per comprendere i passi necessari per ottenere un'operazione in tempo reale si genera un flusso che consiste in algoritmo di cattura, esplorazione dell'architettura, realizzazione e implementazione.

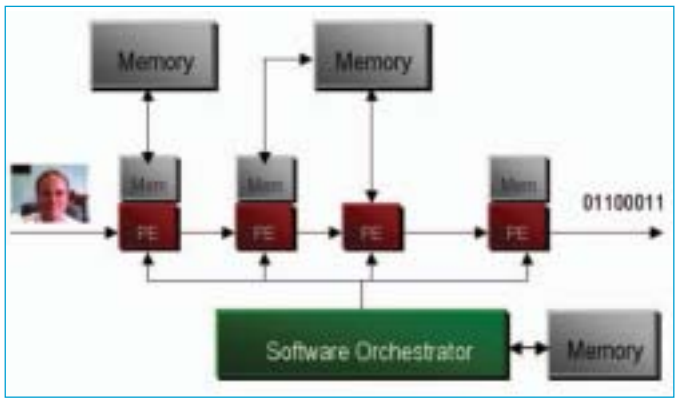

Fig. 4 - Architettura Multimediale Soft Proposta

Le attuali metodologie di codice e compilazione per l'implementazione di compressione multimediale su dispositivi al silicio quali processori general purpose, processori DSP, e media processor presentano il vantaggio che il flusso di progettazione è ben compreso e l'insieme di abilità necessarie inizialmente è costituito da semplici buone pratiche di programmazione. Il punto in cui sorgono i problemi è l'implementazione in tempo reale di operazioni di media streaming usando la tecnologia dei processori al silicio. Sia che si abbiano una o una moltitudine di unità di elaborazione all'interno del processore, il punto è che i flussi di operazioni in tempo reale dovrebbero passare attraverso le unità di elaborazione diverse volte nel corso dell'algoritmo di compressione. Questa creazione di colli di bottiglia dello streaming quindi richiede che il progettista ottimizzi molto attentamente l'uso delle poche unità di elaborazione comprendendo e utilizzando speciali istruzioni su misura per indirizzare il problema dello streaming.

Dall'altro lato dello spettro di soluzioni su silicio si hanno flussi di progettazione FPGA e ASIC che consentono il grado di parallelismo di elaborazione necessario per ottenere l' operazione in tempo reale.

La penalizzazione che si paga per queste prestazioni extra attraverso il parallelismo è un flusso di progettazione per la programmazione meno efficiente. Il flusso di progettazione FPGA e ASIC tradizionale richiede che i designer progettino con un linguaggio di descrizione hardware (HDL) e indirizzino il vostro dispositivo al silicio attraverso la sintesi. Esistono molti gradi di libertà dal momento che un'architettura computer Von Neumann o Harvard non è l'obiettivo. Innanzitutto un'architettura, e in secondo luogo l'utente deve decidere come comunicare fra le strutture nel proprio algoritmo.

L'insieme di abilità richieste per progettare in questa infrastruttura è più vasto rispetto alle abilità di programmazione di base e spesso richiede una conoscenza specifica per l'obiettivo. A complicare la situazione negli ultimi anni è diventato comune per FPGA e ASIC includere processori all'interno del dispositivo al silicio. Ciò consente una fusione di questi due paradigmi di progettazione senza tool di progettazione per facilitare la semplicità all'uso o il partizionamento.

Per flussi di progettazione multimediali per la compressione la fase iniziale di cattura include anche la cattura dell'algoritmo che facilita l'incanalamento del flusso dei dati durante la fase di esplorazione.

Dal momento che lo sscopo è di ottenere l'operazione in tempo reale (30 frame/secondo, 5940 Macroblocchi/secondo) il codice sequenziale C inizialmente catturato può essere scritto con un flusso di dati funzionale in mente. Durante la fase di esplorazione del flusso di progettazione si potrebbe voler essere in grado di fare una stima in anticipo sulle prestazioni dato un parallelismo funzionale del codice catturato.

É possibile ad esempio trovare che il codificatore aritmetico di JPEG2000 è richiesto per eseguire 20000 MOPS e decidere di incorporare qualche parallelismo a grana non fine a questo punto per ottenere i parametri a livello di sistema. Durante la fase di realizzazione del flusso di progettazione si possono prendere decisioni informate circa i dispositivi computazionali al silicio e i dispositivi di memoria.

A questo livello deve essere eseguito un raffinamento delle comunicazioni fra le unità funzionali per ottenere una stima migliore delle prestazioni. A titolo di esempio si può trovare che l'algoritmo di stima del moto che abbiamo scelto non è appropriato per il tipo di memoria selezionato per via della natura casuale dell'indirizzamento.

In ultima analisi durante l'ultima fase del progetto, la fase di implementazione, si vorrebbero confermare le stime sulle risorse sull'uso di silicio e raffinare gli elementi di elaborazione per ottenere la prestazione in tempo reale.

Implementazione della compressione multimediale

L'implementazione degli algoritmi di compressione per i sistemi in tempo reale richiede la comprensione degli algoritmi così come della tecnologia del silicio per l'implementazione. Si vorrebbe essere in grado di modificare il nostro algoritmo di stima del moto ad esempio con compromessi a partire dalla qualità del risultato così come dal costo di implementazione.

Applicazioni in questo campo rientrano in categorie che includono: 1) processori (General Purpose, DSP, Media) dove FPGA e ASIC sono i candidati all'implementazione o 2) per cui

solo FPGA e ASIC sono candidati all'implementazione. Per la categoria 1 le considerazioni sul flusso di progettazione in genere rivestono un ruolo chiave e la facilità all'uso è il percorso a minore resistenza e si tende verso l'uso del processore più efficace per costo e potenza per ottenere l'operazione in tempo reale. In realtà una soluzione FPGA potrebbe fornire un'opzione conveniente per il costo ma per via dell'insieme di abilità richieste per implementare un FPGA e dello stato corrente del flusso di progettazione per gli FPGA, essi non sono scelti a causa di considerazioni di mercato.

Le considerazioni sugli ASIC sono normalmente dominate dai volumi e dai costi NRE. Ciò porta a usare FPGA per implementazioni in applicazioni di alta fascia e in applicazioni di fascia bassa e media se le problematiche nel flusso di progettazione possono essere risolte. Per facilitare un flusso di progettazione FPGA multimediale si consideri un' architettura per mezzo della quale lo streaming media può fluire attraverso una matrice FPGA con elementi di elaborazione (PE) distribuiti attorno al percorso dei dati. La figura 4 mostra un esempio di un canale video composto di elementi di elaborazione funzionale con memoria locale controllata da un'unità di elaborazione generale.

L'unità di elaborazione generale controllerebbe il flusso di dati attraverso il canale e anche modificherebbe l'impostazione dei parametri su una base per trama.

Per esempio se occorre incorporare il controllo del bit rate in tempo reale l'orchestratore software faciliterebbe questo cambiamento gestendo la richiesta dalla rete e rispondendo attraverso la modifica del parametro appropriato nell'elemento di elaborazione che esegue la quantizzazione.

È importante riconoscere che questa architettura non è fissa ma programmabile e chiaramente ciò che è mostrato qui è una possibile sistemazione di elementi di elaborazione della memoria e di controllo software.

Aggiungere alcune strutture e vincoli all'architettura FPGA generale ha il vantaggio di astrarre il progettista a livello di sistema dagli elementi specifici degli FPGA fino a una fase inoltrata nel processo di progettazione.

Conclusioni

In questo lavoro si è discusso la compressione multimediale con il fine di esporre alcune delle problematiche di rilievo di fronte all'implementazione di sistemi di compressione multimediale in tempo reale. Sono stati considerati i requisiti di compressione multimediale. La complessità di progettazione ed è stato proposto un flusso di progettazione. Infine si è considerata la crescente necessità di maggiore astrazione nello spazio su silicio per rendere possibile l'utilizzo efficiente di una tecnologia al silicio da diversi milioni di porte. ✍

Xilinx

readerservice.it n.01