

Grazie alle istruzioni SIMD, è possibile ottenere sensibili miglioramenti nel rendimento dei motori di decodifica video MPEG-4 e migliore qualità nelle immagini

Lo sviluppo di un decoder MPEG-4 su un motore SIMD programmabile e un microprocessore configurabile

Dennis Moolenaar
Tensilica

La compressione digitale dei segnali video è diventata una funzione indispensabile per un'ampia varietà di prodotti che devono saper gestire svariati protocolli video standard.

I più diffusi attualmente sono MPEG-1, MPEG-2, H.263 e MPEG-4 e i progettisti, se vogliono ottenere architetture flessibili e prestazioni dignitose, devono quasi sempre integrarli insieme in qualsiasi prodotto video.

Tuttavia, il vero problema non è ideare la miglior soluzione e progettare, ma realizzare una soluzione sufficientemente ottima in una ragionevole

quantità di tempo. Un diffuso approccio è quello di utilizzare blocchi hardware già pronti, ma ciò risulta conveniente solo per i sistemi piccoli e semplici perché al crescere della complessità aumenta in proporzione anche il numero dei blocchi hardware necessari a svolgere tutte le diverse funzioni richieste nelle specifiche e, con la stessa proporzione, aumentano il costo e la specificità del sistema, ovvero l'impossibilità di adattarlo a più applicazioni e aggiornarne le funzioni. Un altro approccio consiste nell'utilizzare processori programmabili per migliorare la flessibilità dei siste-

mi e il time-to-market necessario al loro sviluppo. Sfortunatamente, i processori programmabili standard non sono capaci di gestire la decodifica dei segnali video in tempo reale, a meno di non farli funzionare a più GHz di velocità di clock, cosa improponibile sugli apparecchi portatili.

I progettisti sono pertanto costretti a inventare soluzioni specifiche per ridurre il carico di lavoro sui processori standard, in modo da poter elaborare i segnali video anche con le ridotte velocità di clock dei processori consumer.

Un'alternativa è quella di implementare elevate prestazioni video utilizzando un core processore configurabile ed estensibile con istruzioni di tipo SIMD (single-instruction, multiple-data).

I core processori di questo genere sono oggi disponibili presso più produttori e il fatto di saper gestire le istruzioni SIMD si dimostra un grande vantaggio nell'ottimizzazione delle funzioni di elaborazione sui segnali video. Si può analizzare un esempio di

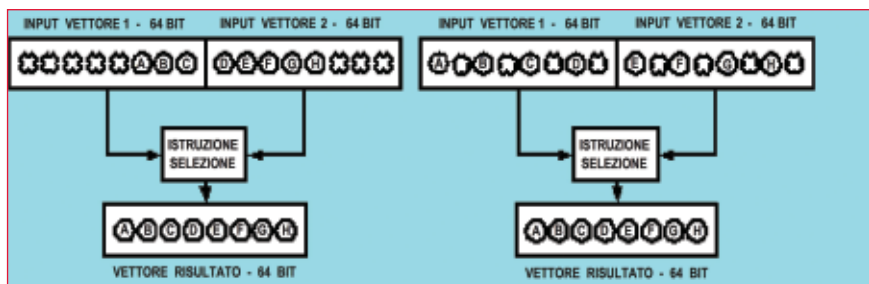


Fig. 1 – L'istruzione di selezione può combinare in diversi modi le informazioni contenute in più vettori. La figura mostra le due tecniche di selezione più comuni

Tabella 1 – I cinque formati QCIF richiesti nella codifica MPEG-4

Video Clip	Numero di frame	Dimensioni stringhe (kbyte)	Velocità bit media (@15 frame al secondo)
Miss America	149	66	52 kbps
Suzie	149	129	104 kbps
Foreman	400	724	217 kbps
Car Phone	381	606	191 kbps
Monsters Inc.	1438	1456	146 kbps @ 18 fps

configurazione di questo genere prendendo come riferimenti la decodifica video MoMuSys MPEG-4, pubblicata sul sito ufficiale www.osi.org, e il core microprocessore Xtensa di Tensilica, utilizzando il linguaggio TIE (Tensilica Instruction Extension) per aggiungere nel core le istruzioni SIMD necessarie alle funzioni di decodifica MPEG-4. La scelta della decodifica MoMuSys è ideale per i prodotti più commerciali perché si tratta di una tecnica pensata per essere facilmente decifrabile e adattabile alle applicazioni, anche se questo beneficio è ottenuto a scapito dell'efficienza. In effetti MoMuSys comprende solo un sottoinsieme delle codifiche video standard MPEG-4 e non è in grado di supportarne tutte le prestazioni; questo svantaggio è però largamente compensato dal beneficio di poter creare un'architettura configurabile ed estensibile.

Il decoder video MPEG-4 MoMuSys

Il comitato per la promozione degli standard MPEG ha sviluppato due implementazioni software al fine di minimizzare le ambiguità e perfezionare l'efficienza dei motori di codifica e decodifica standard. Entrambi que-

garantisce il supporto di tutti i cinque formati QCIF (Quarter Common Intermediate Format) riportati nella tabella 1 ed è quindi sufficiente per un buon livello di prestazioni. Nel mettere a punto il decoder MoMuSys si può riscontrare che le moltiplicazioni fra i numeri interi sono di fatto quelle che consumano il maggior numero di cicli di decodifica. Questa difficoltà può essere risolta grazie alle qualità del processore Xtensa, la cui architettura hardware include un moltiplicatore 16x16 bit che può essere configurato

proprio per ridurre del 25% il tempo medio di decodifica delle moltiplicazioni fra numeri interi. La simulazione del codice dimostra infatti che il MoMuSys usa l'aritmetica a virgola mobile per le trasformate coseno inverse (iDCT) tipiche dello standard MPEG-4, anche se per queste operazioni l'aritmetica in virgola mobile non sarebbe

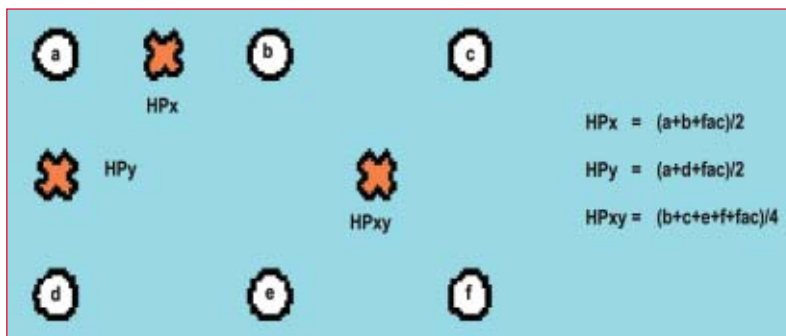


Fig. 2 – La tecnica di compensazione delle immagini di movimento detta "half-pel" (HP)

sti modelli sono scaricabili dal sito OSI. Tensilica ha optato per la versione Mobile Multimedia Systems (MoMuSys) dello standard MPEG-4, i cui algoritmi di codifica e decodifica sono scritti in ANSI-C e possono essere compilati con qualsiasi compilatore GNU. Sebbene non supporti il controllo della velocità e dei frame, esso

necessaria. Un'implementazione integra delle iDCT è molto più veloce e offre una precisione sufficiente a soddisfare i requisiti imposti dalle direttive standard IEEE 1189-1990 DCT. Questa è stata pertanto utilizzata nella configurazione del motore di decodifica MoMuSys di cui si parla in quest'articolo.

Tabella 2 – I vantaggi dell'ottimizzazione delle prestazioni grazie alle tecniche di compensazione sulle immagini

Video Clip	Cicli per compensazione immagini movimento in MPEG-4 originale	Cicli per compensazione immagini movimento in MPEG-4 ottimizzato	Accelerazione nelle istruzioni TIE
Miss America	1,620 G cicli	21,43 M cicli	75,6 x
Suzie	1,660 G cicli	27,18 M cicli	61,1 x
Foreman	4,571 G cicli	90,04 M cicli	50,8 x
Car Phone	4,296 G cicli	79,35 M cicli	54,1 x
Monsters Inc.	15,578 G cicli	194,96 M cicli	79,9 x

Lo sviluppo del motore SIMD MPEG-4

Per creare le estensioni del processore con il linguaggio TIE, il progettista può utilizzare due tecniche di ottimizzazione del codice che consentono entrambe di ridurre il ciclo di esecuzione,

ma in due modi diversi: unendo più operazioni in un'unica istruzione TIE; operando simultaneamente su più dati con operazioni single-instruction multiple-data, SIMD.

Malgrado i due metodi non si escludano a vicenda, il secondo è quello che offre i maggiori vantaggi nella decodifica MPEG-4, perché i segnali video contengono sempre molte parti che si ripetono uguali per più frame e, dunque, consentono frequentemente di eseguire una stessa istruzione su ampie quantità di dati.

Per esempio, consideriamo la compensazione delle immagini di movimento, dove sia necessario caricare otto elementi dai frame video di riferimento e trasferirli sul frame corrente. Le istruzioni SIMD TIE possono fare tutto ciò utilizzando due sole istruzio-

tori d'ingresso di 64 bit ciascuno e viene impiegata per organizzare le informazioni sui pixel mutuamente correlate, migliorando l'elaborazione SIMD. Analizzando il codice sorgente si possono individuare le funzioni che occupano maggior spazio e che impegnano maggiormente il motore di decodifica. Fra esse, la compensazione delle immagini di movimento, le iDCT, la decodifica dei codici a lunghezza variabile, e gli algoritmi di codifica spaziale del colore.

La compensazione ottima dei movimenti

La compensazione delle immagini di movimento richiede un'abbondante ridondanza delle immagini da frame a frame per produrre un elevato rapporto di compressione video. La codifica

mi fra ogni coppia di righe adiacenti, arrotondando e dividendo tramite le istruzioni SIMD di addizione e scorrimento. La compensazione orizzontale è un po' più complessa perché i dati non sono organizzati per colonne. In questo caso occorrono due istruzioni SIMD di selezione per organizzare i dati per colonne e solo dopo il motore procede all'interpolazione.

Per questa fase, il codice originale MoMuSys richiede 272 istruzioni che vengono eseguite in due cicli, mentre il codice ottimizzato con le operazioni SIMD richiede solo 40 istruzioni eseguite in un unico ciclo, con un miglioramento netto nel rendimento di circa 7 volte. Inoltre, anche nel caso in cui le specifiche MPEG-4 non lo richiedano, con quest'approccio si può implementare anche la tecnica di compensazione "quarter-pel" schematizzata nella figura 2 a destra, dove l'interpolazione avviene fra quattro pixel vicini.

Compensazione tra frame

I decoder video possono usare due tecniche per gestire la compensazione sulle immagini di movimento di frame confinanti (outside-the-frame): possono creare un'immagine di pixel interpolati, "pel", circoscritta attorno a ogni frame di riferimento, oppure possono testare la posizione di ogni "pel" per verificare che non esca al di fuori del frame di riferimento, correggendone in tal caso la posizione. Il vantaggio del primo approccio è che nessuna istruzione aggiuntiva viene eseguita durante i cicli critici dell'algoritmo di compensazione delle immagini di movimento. Lo svantaggio è che occorre prevedere l'uso di una maggior quantità di memoria. Il secondo approccio non richiede più memoria, ma necessita di istruzioni aggiuntive, alcune delle quali da eseguire durante i cicli critici dell'algoritmo di compensazione. Il decoder MoMuSys implementa entrambi i metodi. Il primo metodo viene eseguito copiando il frame di riferimento in un grande buf-

Tabella 3 – L'accelerazione ottenibile ottimizzando il calcolo delle trasformate coseno discrete inverse

Video Clip	Cicli per iDCT originali	Cicli per iDCT ottimizzate	Accelerazione nelle istruzioni TIE
Miss America	3220,86	345,24	9,4 x
Suzie	3283	343,77	9,5 x
Foreman	3414,7	342,22	10,0 x
Car Phone	3356,83	342,62	9,8 x
Monsters Inc.	3349,21	343,26	9,8 x

ni e otto trasferimenti dati, il che significa moltiplicare per otto le prestazioni rispetto alle altre tecniche. Si può dimostrare che tutte le altre parti dell'algoritmo di decodifica a eccezione del trattamento del flusso di bit possono essere ottimizzate in modo simile e che i motori SIMD sono i più efficienti nella decodifica dei segnali MPEG-4. Il motore SIMD MPEG-4 che ne risulta può operare su otto elementi dati a 16 bit oppure su quattro a 32 bit ed eseguire simultaneamente le seguenti istruzioni: aritmetico/logiche (addizione, sottrazione, scorrimento, combinazione logica), caricamento e memorizzazione a 128 bit, moltiplicazione e accumulo, comparazione e spostamento condizionale, selezione. L'operazione di selezione mostrata nella figura 1, per esempio, genera un vettore di uscita da due vet-

MPEG-4 genera vettori di stima del movimento sui blocchi di pixel utilizzando un apposito algoritmo di ricerca sui pixel (pixel-search). La decodifica MPEG-4 raffina il vettore di stima risultante, migliorando la qualità delle immagini decodificate, grazie all'algoritmo di interpolazione "half-pel" illustrato nella figura 2 a sinistra. Questo algoritmo migliora le prestazioni perché il valore di ciascun pixel viene calcolato in base a due valori "pel" e a un fattore di ridondanza "fac", ricavati dall'elaborazione delle informazioni su due pixel vicini. La compensazione half-pel verticale è più semplice perché i dati sono già organizzati per righe. Prima, il motore SIMD carica i dati da interpolare in una matrice da 8x8, un'operazione per la quale sono necessari 9 trasferimenti. Dopodiché interpola i dati calcolando i valori otti-

fer, dove viene costruita l'immagine attorno al frame. In linea di principio, questo dovrebbe essere sufficiente a una buona compensazione delle immagini di movimento, ma il MoMuSys implementa anche il secondo metodo. In questo caso, il vantaggio di aver scelto un core processore ampliabile e programmabile consente di decidere la quantità di risorse più adatta a ogni condizione applicativa.

Prestazioni e compensazione

Prima di riassumere i vantaggi in termini di prestazioni che queste tecniche di compensazione offrono, soprattutto sulle immagini di movimento, si deve tener conto di un'ulteriore tecnica di ottimizzazione sulle informazioni video. Nel decoder MoMuSys, infatti, la funzione AddImagel occupa

La trasformata coseno discreta inversa

Ci sono due modi per implementare un algoritmo per il calcolo delle trasformate iDCT a due dimensioni (2D) sulle matrici di dati 8x8: una iDCT 2D matriciale; due iDCT 1D sequenziali prima sulle colonne e poi sulle righe della matrice 8x8. Il decoder MPEG-4 MoMuSys utilizza il secondo approccio che è anche quello più conveniente per le istruzioni SIMD, perché permette di eseguire otto iDCT 1D in parallelo accelerando notevolmente la velocità di calcolo. Innanzitutto si possono eseguire direttamente le otto iDCT 1D sulle colonne e poi, previa organizzazione dei dati relativi ai pixel in riga, le otto iDCT sulle righe, ottenendo in sole due operazioni la iDCT 2D sull'intera matrice

processore può accelerare queste funzioni muovendo i dati attraverso degli appositi registri a scorrimento con le istruzioni TIE, che semplificano la manipolazione e lo spostamento dei singoli bit. Queste però non sono operazioni SIMD. Il processore può anche essere configurato in modo da ottimizzare la decodifica dei codici a lunghezza variabile. Questi algoritmi si usano perché consentono di diminuire il numero medio di bit necessario a codificare le parole di un codice, assegnando le stringhe di bit più corte alle parole che ricorrono più frequentemente nel codice e le stringhe di bit più lunghe alle parole meno usate. Grazie alla codifica a lunghezza variabile il numero di bit totale può essere significativamente ridotto nelle codifiche MPEG-4 e a livello delle matrici iDCT da 8x8.

Tabella 4 – L'ottimizzazione della decodifica MPEG-4 consente di accelerare le prestazioni da 28 fino a 40 volte, rispetto alla decodifica MPEG-4 senza motori SIMD

Video Clip	Cicli decoder MPEG-4 originali	Cicli decoder MPEG-4 ottimizzato	Frequenza di clock ottimizzata (su 15 frame/secondo)	Accelerazione nelle istruzioni TIE
Miss America	3,126 G cicli	76,81 M cicli	7,7 MHz	40,1 x
Suzie	3,389 G cicli	102,19 M cicli	10,3 MHz	33,2 x
Foreman	10,045 G cicli	359,5 M cicli	13,5 MHz	27,9 x
Car Phone	9,222 G cicli	308,7 M cicli	12,2 MHz	29,9 x
Monsters Inc.	29,327 G cicli	822,8 M cicli	8,6 MHz	35,6 x

circa il 3,24% del ciclo principale. Questa funzione serve ad aggiungere il frame contenente tutti i risultati delle iDCT al frame contenente tutti i risultati delle compensazioni. Eseguire la compensazione delle immagini di movimento e poi aggiungervi i risultati delle iDCT permette di fare a meno di un'elaborazione apposita su questi ultimi. Quest'accorgimento non solo riduce il numero dei trasferimenti dati da e verso la memoria, ma anche la quantità di memoria utilizzata. Invece di usare la memoria necessaria per contenere due frame interi, infatti, basta la quantità di memoria su misura per un frame e per sei blocchi iDCT. La tabella 2 mostra l'accelerazione percepibile nell'esecuzione del codice, grazie all'implementazione di queste tecniche.

8x8. La completa trasformazione di una matrice iDCT utilizzando le istruzioni di selezione SIMD richiede 24 cicli, ma comprendendo le operazioni di spostamento i cicli diventano 48. Si possono comunque ridurre ancora utilizzando una memoria di buffer per lo spostamento dei dati da e verso la memoria. La tabella 3 mostra l'accelerazione conseguibile ottimizzando l'esecuzione delle trasformate coseno discrete inverse.

Il trattamento del flusso di bit

I decoder video utilizzano un set di funzioni dedicate al trattamento del flusso di bit nel processamento dei segnali video. Tutti i dati vengono estratti dal flusso delle immagini video utilizzando queste funzioni. Il

Ottimizzazioni extra sui decoder MPEG-4

Una parte dell'algoritmo di compressione MPEG-4 trasforma i valori delle matrici 8x8 nel dominio della frequenza, dove divide i risultati per una cifra di quantizzazione. Il flusso di bit del segnale video trasporta anche questo

valore di quantizzazione e il decoder deve utilizzarlo per effettuare la trasformata inversa, ovvero la "dequantizzazione". L'algoritmo, in pratica, deve decodificare le iDCT per ricostruire gli originali valori delle DCT. Le istruzioni SIMD consentono di quantizzare e dequantizzare simultaneamente più valori di DCT e iDCT, riducendo il numero medio dei cicli necessari a quest'operazione.

La predizione AC/DC viene usata nello standard MPEG solo per gli "intra-frame", che generalmente sono molto meno frequenti degli "inter-frame". D'altra parte, i primi richiedono più cicli di decodifica rispetto ai secondi ed è quindi importante ottimizzarne l'elaborazione, anche perché la velocità di clock necessaria a questa funzione viene assunta come

limite di riferimento inferiore per la frequenza di clock del decoder. Il decoder, in altre parole, dev'essere abbastanza veloce da poter decodificare un "intra-frame" in tempo reale. La predizione AC/DC riduce il numero di bit necessari per la codifica di un "intra-frame" stimando i valori DC e/o AC nelle matrici iDCT. La predizione dei valori AC, in particolare, richiede un elevato numero di cicli a causa delle divisioni e delle operazioni logaritmiche che comporta e questi calcoli potrebbero essere estratti fuori dall'algoritmo principale solo riscrivendo in toto il codice del MoMuSys. L'ottimizzazione comunque consente di ridurre i cicli necessari per la predizione AC/DC da 2080 a 520.

Il decoder MPEG-4 MoMuSys utilizza per i colori la codifica spaziale YCbCr. Questa rappresentazione deve spesso essere trasformata in una differente codifica spaziale, come la RGB, durante la decodifica.

Tuttavia, trasformare i dati dalla codifica YCbCr alla RGB richiede la moltiplicazione fra matrici e inoltre i valori

trasformati di R, G e B devono a loro volta essere spostati di 16 bit per ottenere i reali valori RGB. Pertanto la conversione dei colori è senza dubbio un'altra operazione critica nella decodifica perché comporta una moltiplicazione matriciale per ciascun pixel. Per soddisfare i severi requisiti di conversione spaziale dei colori fra la codifica YCbCr e la RGB, il motore SIMD MPEG-4 contiene un moltiplicatore progettato per eseguire moltiplicazioni fra 16x16 bit.

L'elevata precisione di questo moltiplicatore consente di fare a meno di implementare istruzioni aggiuntive specifiche per le operazioni matriciali e inoltre il motore SIMD permette di eseguire moltiplicazioni matriciali simultaneamente su 8 pixel, migliorando di otto volte il rendimento dell'algoritmo di conversione spaziale dei colori.

La tabella 4 mostra che il fattore di ottimizzazione del decoder MPEG-4 MoMuSys, dovuto alla presenza delle operazioni SIMD, varia fra 28 e 40, secondo l'entità del flusso di dati

video. Di conseguenza, questo decoder MPEG-4 basato su core processore estensibile può funzionare a una velocità di clock più bassa di quella necessaria a qualsiasi altro processore dedicato.

La ragione di una così ampia diversità di prestazioni dipende dalle diverse caratteristiche dei flussi di bit contenenti le informazioni video, ciascuno dei quali può richiedere una diversa quantità di calcoli e soprattutto un differente numero di trasformate iDCT. La configurazione del processore per la decodifica MPEG-4 richiede circa 53K gate, mentre i gate supplementari per il motore SIMD programmabile sono 67K. Ne risulta un decoder flessibile e programmabile capace di elaborare i segnali video in modo molto efficiente e relativamente semplice da sviluppare, con sensibili benefici nel time-to-market dei nuovi prodotti. *LS*