MICROPROCESSORI PER DSP. Modelli recenti e tools di sviluppo: alcune applicazioni nel settore telecom.

G. Bertini*, V. Di Salvo**, W. Gaffi***, R. Neri***, A. Rocchi****

Nota Interna B4 – 10, maggio 2003

*ISTI-CNR
** Ricercatore Progetto MOSART
*** Associati alla ricerca
**** Inc. di Coll. professionale

INDICE

1. Sommario.

E' noto che la disponibilità dei processori specializzati per il Digital Signal Processing (DSP) ha consentito l'utilizzo delle tecniche numeriche in sostituzione di quelle analogiche in moltissime applicazioni di trattamento di segnali in real-time, facendo addirittura esplodere settori di mercato impensati (basti pensare alla telefonia mobile). A partire dagli anni '80 anche all'IEI si sono impiegati vari tipi di DSP in stazioni di lavoro orientate alla sintesi di segnali musicali ed in ricerche sul controllo attivo di rumore acustico.

Di pari passo con la miniaturizzazione della componentistica, la tecnologia ha messo a disposizione modelli sempre più complessi e potenti. Allo stato attuale sono numerosi i costruttori che realizzano processori tipo DSP ottimizzati alle varie esigenze applicative. Nella presente nota sono richiamate le principali caratteristiche di base dei processori DSP ed elencati i modelli recenti più rappresentativi con i relativi tools di sviluppo. Alcuni di questi modelli vengono attualmente usati in vari progetti all'interno Lab. Segnali e Immagini dell'ISTI, nell'ambito di ricerche e contratti con ditte esterne.

2. Introduzione ai processori DSP

1.1. Introduzione

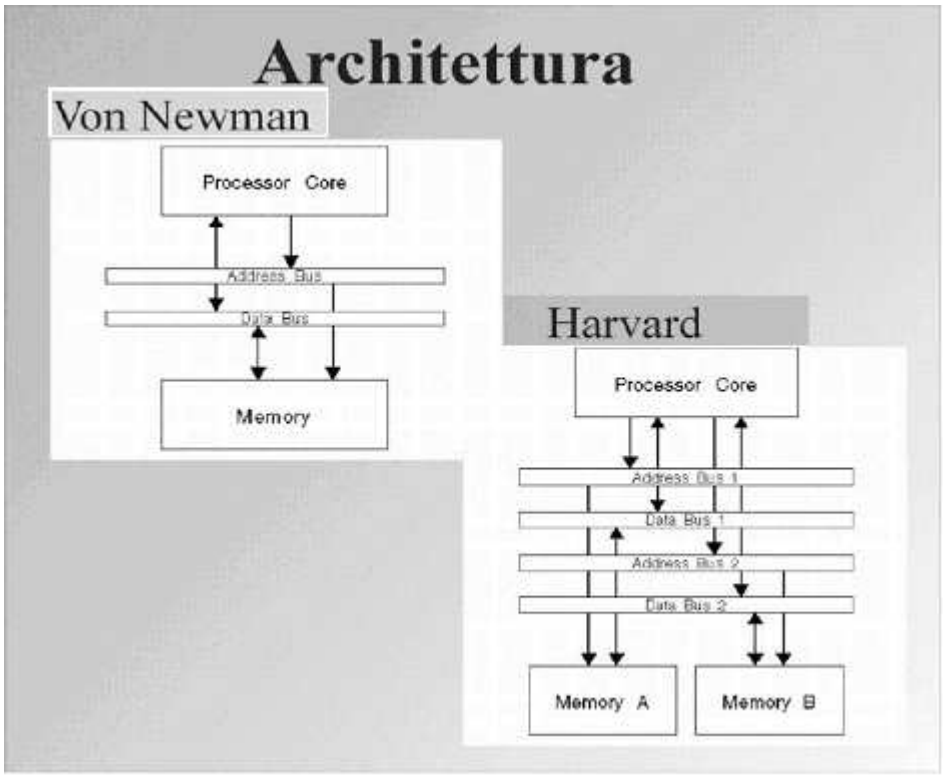
I Digital Signal Processor (DSP) come è noto sono stati introdotti negli anni '80 per risolvere in modo ottimale determinate classi di problemi di elaborazione digitale dell'informazioni (in particolare segnali audio-video digitalizzati) da realizzare con stretti vincoli temporali. Negli ultimi anni si sono imposti come soluzione ideale a problematiche applicative di media ed elevata complessità anche in molti altri settori tecnologici. L'analisi e la sintesi dei segnali vocali e musicali, l'elaborazione delle immagini, l'audio digitale, l'automazione, la strumentazione biomedicale e di laboratorio, e in generale l'elettronica destinata alle applicazioni consumer, sono solo alcune delle applicazioni dei DSP. Un processore DSP ha la funzione di realizzare manipolazioni matematiche di segnali provenienti dal "mondo reale" opportunamente convertiti dall'ambito analogico a quello digitale. Un DSP può essere considerato specifico per una determinata applicazione (task), oppure essere di supporto ad altri elaboratori svolgendo la funzione di "co-processore". In questo caso i DSP vengono utilizzati per elaborazioni ripetitive su segnali in tempo reale (ad es. operazioni di compressione video o acceleratori di grafica) mentre ai processori "general purpose" rimangono affidate tutte le altre operazioni dei computer, la manipolazioni di grandi quantità di dati, il calcolo scientifico, la gestione del sistema operativo, delle unità di input/output, ecc.
I DSP vengono forniti "su misura" per la precisione richiesta dalla specifica applicazione a 8, 16, 24 e 32 bit; su alcuni tipi è possibile addirittura avere il formato variabile compreso anche l'organizzazione della memoria. L'aritmetica può essere pure a virgola fissa (fixed-point) o virgola mobile (floating point). La tecnologia attuale poi consente delle alte performance unitamente a consumi contenuti, favorendo la realizzazione di moltissimi dispositivi che richiedono bassa potenza di alimentazione; esempio di impiego tipico i telefoni cellulari, i palmari ecc.
Uno dei punti salienti dei DSP è quello di eseguire una MAC (moltiplicazione di due operandi ed accumulazione del risultato in memoria) in una sola istruzione (in un solo ciclo macchina): ad esempio, sono in grado di realizzare filtraggi del tipo:

$$Y_{(n)} = \sum_{k=1}^{M} a_k X_{(n-k)}$$

anche con decine e centinaia di termini in real-time ed eseguire accessi multipli a blocchi di memoria all'interno della stessa istruzione. Altra operazione che viene realizzata in modo efficiente è il trasferimento dati e da blocchi di memoria alla cpu e viceversa .
L'insieme di queste caratteristiche fa si che i DSP forniscano processamento real-time in moltissime applicazioni (di regola per real-time si intende il completamento di calcoli del tipo detto sopra o FFT ecc. all'interno di un intervallo di campionamento di un segnale). I DSP attualmente in commercio riescono a compiere serie di operazioni in parallelo (sfruttando delle tecniche di pipeline) o basandosi su istruzioni con struttura molto ampia (VLIW) o avendo delle vere e proprie architetture interne parallele. La differenza fondamentale tra un processore general purpose e un DSP è dovuta alla struttura interna, basata sull'utilizzo di più bus per le connessioni tra le memorie e le altre sottounità, come descritto in dettaglio nel seguito.

1.2. Architettura dei Dsp

I DSP basano il loro funzionamento sull'architettura Harvard, mentre i General Purpose su quella di tipo Von Newman. L'architettura Harvard deriva da proposte studi eseguiti presso l'omonima università americana. Agli albori della storia dei calcolatori venne realizzato in quella sede un calcolatore a programma memorizzato, con memorie separate per le istruzioni ed i dati. Quindi il termine "architettura Harvard" serve per designare processori con spazi di memoria distinti per dati e istruzioni.



Dal punto di vista della memoria, un processore di tipo Von Newman esegue un solo accesso ad ogni ciclo, sfruttando altre soluzioni per ridurre i tempi di esecuzione; ad esempio con l'uso della cosiddetta "cache" (memoria di piccole dimensioni ad accesso rapido, direttamente interfacciata sul

chip del processore, che contiene la porzione di programma in uso in quel momento).

Un DSP compie 2 (o più) accessi alla memoria per ciclo di istruzione; inoltre i dispositivi più costosi presentano anche la cache. Quindi ad alto livello, la distinzione nell'architettura della memoria consta nella doppia via di accesso del processore alla memoria dati e a quella programma.
La maggior parte delle istruzioni vengono eseguite in 1 solo ciclo (per lo meno le istruzioni che sono più usuali, cioè a maggior probabilità di essere richiamate) e questo fa sì che la velocità di calcolo sia massima. Ad es nel caso del filtro prima citato si verifica l'accesso simultaneo alla memoria dati (per ottenere la variabile) e alla memoria programma (per ottenere il coefficiente).
I DSP sono caratterizzati da un set di istruzioni limitato ma adatto alle tipiche operazioni di elaborazione dei segnali. In ogni caso un DSP è in grado di eseguire una o più operazioni in parallelo, adotta soluzioni specifiche per garantire l'efficienza dei loop, possiede una address unit dedicata per indirizzare contemporaneamente diverse sottounità e supporta diversi modi di funzionamento per l'I/O ad alte prestazioni. Tra quest'ultima va citato il modo di interfacciamento "seriale veloce full-duplex" verso i Codec sigma delta, caratteristici nel settore dell'audio digitale.(bibliografia).

1.3. Vantaggi dell'uso di un DSP

Rispetto all'elaborazione in analogico, una serie di fattori che ha permesso lo sviluppo della tecnologia dei DSP:

- STABILITÀ: Il fattore termico rende altamente instabile le parti di acquisizione e di tutte le sezioni di elaborazione analogica. Nell'elaborazione digitale eventuali fattori di instabilità intervengono soltanto nel momento della conversione da analogico a digitale e viceversa.
- PREVEDIBILITÀ (di lunga durata)
- Possibilità di IMPLEMENTARE CARATTERISTICHE ADDIZIONALI
- FLESSIBILITÀ (programmabilità)
- Rapporto COSTO/PRESTAZIONI
- Tecnologia in uso in molti prodotti elettronici di LARGO CONSUMO
- I TASK che vengono implementati dai DSP rappresentavano il "collo di bottiglia" in molte applicazioni. Sono pochi in percentuale, ma rappresentano la parte più onerosa dal punto di vista computazionale.
-

1.4. Esempi di applicazioni DSP

Oggi i DSP sono di larghissimo impiego e vengono utilizzati in un vasto numero di attrezzature elettroniche, anche di uso quotidiano. Nella Tabella seguente (estratta dal sito della Texas Instruments, leader nel settore DSP) sono riportati in dettaglio i tipi di operazioni ricorrenti e i settori di impiego.

General-Purpose DSP
Digital filtering
Convolution
Correlation
Hilbert transforms
Fast Fourier transforms (FFTs)
Adaptive filtering
Windowing
Waveform generation
Discrete cosine transforms
Hartley transforms

Instrumentation
Spectrum analysis
Function generation
Pattern matching
Seismic processing
Transient analysis
Digital filtering
Phase-locked loops

Control
Disk control
Servo control
Robot control
Laser printer control
Engine control
Motor control

Automotive
Engine control
Vibration analysis
Antilock brakes
Antiskid brakes
Adaptive ride control
Global positioning navigation
Voice commands
Digital radio
Cellular telephones
Active suspension
Noise suppression
Electronic power steering
4-wheel steering
Air bag control
System diagnosis
Radar detectors
Intelligent cruise control

Telecommunications
Hand-free speaker phones/ echo cancellations
ADPCM transcoders
Digital PBXs
Line repeaters
Channel multiplexing
1200- to 33,600-bps modems
Adaptive equalizers
DTMF encoding/decoding
Data encryption
Low-speed transcoders/ vocoders
ISDN basic/primary rate interfaces
FAX
Cellular telephones
Cordless telephones
Digital speech interpolation (DSI)
Packet switching and protocol
Videoconferencing/video compression/multimedia
Spread spectrum communications
Answering machines
Cable modems
Network switching
Modems

Consumer
Radar detectors
Power tools
Digital audio/TV
Music synthesizer
Educational toys
Answering machines
Multimedia
Digital cameras
Digital videodisk players
White goods (dishwashers, washing machines, etc.)
Karaoke
Feature phones
Arcade games
Set top boxes

Graphics/Imaging
3-D rotation
Robot vision
Image transmission/ compression
Pattern recognition
Image enhancement
Homomorphic processing
Workstations
Animation/digital map

Voice/Speech
Voice mail
Speech vocoding
Speech recognition
Speaker verification
Speech enhancement
Speech synthesis
Text-to-speech

Industrial
Robotics
Numeric control
Security access
Power line monitors
Active noise cancellation
Electronic meters

Computers
Laser printers/copiers
Scanner/bar-code scanner
Optical character recognition (OCR)
Neural networks
High-speed array processors
Imaging
Videoconferencing
Modems
Networking controller

Military
Secure communications
Radar processing
Sonar processing
Image processing
Navigation
Missile guidance
Radio frequency modems

Come risulta dalla tabella i DSP coprono una vasta fetta del mercato elettronico e si prevede un forte sviluppo anche per il futuro. Gli apparecchi basati sui DSP sono fortemente competitivi in specifici campi (basti pensare alla diminuzione dei prezzi di costo dei telefoni cellulari negli ultimi anni e al forte incremento dello sviluppo tecnologico degli stessi) e per eseguire particolari task (algoritmi di compressione audio, video e del parlato).

1.5. Specifiche di un Task per DSP

I task eseguiti dai DSP sono implementati per risolvere problemi di natura diversa che renderebbero onerosa la computazione per ogni tipo di processore. Un DSP deve essere in grado di:

eseguire PROCESSI NUMERICI RIPETITIVI
fornire una determinata ACCURATEZZA NUMERICA
se necessario poter eseguire anche CALCOLI IN VIRGOLA MOBILE
avere ELEVATA VELOCITÁ DI TRASFERIMENTO DALLA/ALLA MEMORIA
soddisfare richieste di PROCESSAMENTO REAL-TIME

I processori DSP devono soddisfare le specifiche minimizzando:

- COSTI
- DISSIPAZIONE DI POTENZA
- USO DELLA MEMORIA realizzare programmi che utilizzino poca memoria, poiché la memoria è uno dei fattori principali che determina il costo di un dispositivo.
- TEMPO DI SVILUPPO L'ottimizzazione di questo parametro non è semplice: occorre una buona conoscenza delle caratteristiche del processore, che non sono note a priori, ma solo conseguentemente al suo utilizzo. La programmazione del DSP spesso è fatta con un linguaggio a basso livello (assembler o linguaggio macchina), questo implica un aumento nel tempo di sviluppo (è più problematico programmare in linguaggi a basso livello). Tipicamente, meno un DSP costa e più difficile è la sua programmazione, non solo in termini di costo del dispositivo, ma anche dell'ambiente di sviluppo nel suo complesso.

In generale i dispositivi DSP possono essere raggruppati in tre grandi famiglie:

- DSP che, pur essendo pensati per il processamento dei segnali, si possono definire di tipo "*general purpose*", ovvero adatti a qualsiasi tipo di elaborazione essendo dotati di un set di istruzioni più o meno esteso.
- "*Application specific DSP*" ovvero specifici per una determinata applicazione, cioè progettati e realizzati con caratteristiche tali da svolgere in modo efficiente un determinato insieme di task (DASP Texas); quando si tratta di un solo task si chiamano *algorithmic specific* (ad esempio il calcolo della FFT con chip della Zoran*).*
- "*Microprocessori di tipo general purpose utilizzati anche in applicazioni DSP*" (come il Pentium MMX della Intel) pur con un'architettura general purpose, viene consentita l'elaborazione diretta di alcuni dati tramite l'introduzione d'istruzioni in grado di fornire risposte in tempo reale.

1.6. Criteri di scelta di un DSP

Una delle prime domande da porsi nel riguardo della scelta di un DSP è quella della valutazione del task che questo dovrà andare a compiere. Ovvero sarà necessario capire quale tipo di lavoro il dispositivo realizzerà. Sarà di fondamentale importanza la scelta dei seguenti parametri:

- FORMATO ARITMETICO: il DSP dovrà essere in grado di trattare dati dell'ampiezza necessaria (in bit) alla precisione con la quale si vuole realizzare l'applicazione e quindi si dovrà stabilire quale forma assumerà il dato (a quanti bit, che tipo di operazione aritmetica introdurre, quanti bit significativi dopo la virgola, ecc..).
- AMPIEZZA FORMATO DATI
- VELOCITÀ campionare segnali per poterli ricostruire alla frequenza massima ottenuta.
- ORGANIZZAZIONE DELLA MEMORIA
- TOOL DI SVILUPPO DISPONIBILI: uno dei punti più importanti nella scelta di un DSP: a volte vengono preferiti DSP meno potenti ma ai quali la casa costruttrice o affiliata, distribuisce a corredo un tool di sviluppo potente e affidabile che fa diminuire il tempo di realizzazione di un'applicazione o la sua programmazione.
- SUPPORTO AL MULTIPROCESSING: possibilità di usare più processori insieme per aumentare la potenza di calcolo oppure per suddividere i compiti di processamento dei segnali per processore, anziché per procedure.

- POWER CONSUMPTION MANAGEMENT, essenziale per le applicazioni che richiedono basso assorbimento di corrente, come ad esempio nei cellulari
- COSTO.

1.6.1. Formato aritmetico

Ne esistono di due tipi: FLOATING POINT e FIXED POINT

Il formato aritmetico floating point è più facile da programmare, poiché non serve riformulare l'applicazione per adattarla alla rappresentazione numerica interna del processore. I numeri vengono rappresentati con mantissa ed esponente: non si dovranno considerare altri algoritmi per la rappresentazione dei numeri affinché la precisione sia mantenuta. Il dispositivo con floating point è tipicamente più costoso, proprio perché al suo interno possiede dell'hardware dedicato (**floating point unit**). In generale, le applicazioni per le quali si sceglie un DSP di tipo floating point sono particolari ed hanno caratteristiche tali da sopportare un maggior costo a fronte di una relativa facilità di programmazione e maggiore precisione nei risultati.

Con il formato aritmetico fixed point si hanno caratteristiche opposte rispetto al caso precedente: complicato da programmare ma costo di realizzazione contenuto. Un DSP fixed point è complicato da programmare a causa della difficoltà di rappresentazione dei numeri reali, che necessariamente devono essere approssimati al fine di essere rappresentati come frazioni di numeri interi. Inoltre si dovrà decidere la precisione e scalare il dato nella dimensione disponibile per il processore.

1.6.2. Ampiezza Dati

L'ampiezza dei dati influisce pesantemente su:

- DIMENSIONI DEL CHIP – COSTO
- TIPI DI DEVICE CONNESSI (MEMORIE, ETC.) il DSP non potrà essere concepito come strumento a sé stante: l'ampiezza dei dati che fornirà o che riceverà dovranno essere compatibili al DSP, quindi, tutti i device ad esso connessi, dovranno avere lo stesso formato.
- PRECISIONE / VELOCITÁ RICHIESTA DALL'APPLICAZIONE è un parametro difficile da valutare, poiché la velocità di un processore non ha significato univoco. Un processore dovrebbe essere veloce ad eseguire una determinata operazione. La clock rate non fornisce informazioni sulla velocità, può dare solo indicazioni sulla dissipazione.

1.6.3. Velocità di esecuzione delle istruzioni

La velocità di un'applicazione svolta con un DSP si può misurare in effetti con sicurezza una volta che si è implementata l'applicazione stessa. Definizioni di misura della velocità:

- MIPS
- OPERAZIONI (AD ES. MAC)
- LOOP PERFORMANCE
- BENCHMARK (KERNEL FUNCTIONS: FIR, ALGORITMI DI CODIFICA ecc.)

Esistono dei programmi, detti benchmark, che permettono di misurare la velocità di esecuzione di determinate applicazioni significative di un DSP. L'unico problema è relativo al fatto che, molte volte, i benchmark sono predisposti o realizzati da chi progetta il DSP e quindi potrebbero nascondere situazioni favorevoli. Una delle misure tradizionali della velocità di un processore è il MIPS (milioni di istruzioni al secondo); la sua misura è completamente scollegata dal tipo di applicazione in esecuzione. Può essere selezionato un determinato set di istruzioni e valutato il tempo di esecuzione di questo con diversi DSP. Un esempio concreto è quello di determinare, ad esempio, quanto un DSP impiega ad eseguire un'istruzione MAC, anche questo parametro può essere impreciso (coloro che realizzano il DSP possono fare in modo che risponda a questo tipo di operazione in tempi veramente eccellenti che si verificano in situazioni di loop estesi e non come istruzione singola. Una delle misurazioni che può risultare efficace è la performance di loop, ovvero quanto impiega un processore ad eseguire un determinato loop (esecuzione di una parte operativa e controllo su passaggio a ciclo successivo o a istruzione di salto). Questa misurazione da sola potrebbe non essere sufficiente a determinare le prestazioni di un processore, ma dovrebbe entrare a

far parte delle misure che potrebbero comporre un benchmark ideale. La soluzione più veritiera, in definitiva, sembra essere quella dell'utilizzo di benchmark costituiti da quelle che vengono denominate funzioni kernel. Non si può pensare di scrivere un programma una volta per tutte e farlo girare sui DSP da testare. Esistono dei gruppi di lavoro, solitamente affiliati ad Università, che hanno il compito di valutare i tempi di esecuzione di determinate routine e funzioni kernel, opportunamente disegnate per quel processore. La media pesata di questo insieme di test fornisce un voto per le prestazioni di quel determinato DSP. Uno dei benchmark più noti è il BTDImarks. E' stato implementato da un gruppo di ricerca (Berkley Design Technology) dell'Università di Berkley. Per ogni kernel function vengono misurati tre parametri:

- TEMPO DI ESECUZIONE
- OCCUPAZIONE DELLA MEMORIA
- DISSIPAZIONE DI POTENZA.

Solitamente i DSP vengono utilizzati per applicazioni che necessitano di bassa dissipazione di potenza e, quindi, installati su dispositivi alimentati a batteria o in genere portatili. La velocità di esecuzione di una determinata applicazione su un DSP è un parametro importante ma difficile da misurare. Il problema sta nel definire in modo preciso il temine "velocità" nella specifica applicazione presa di riferimento. Altro parametro che influenza la scelta di un DSP è l'organizzazione della memoria in riferimento ad accessi interni ed esterni. La dimensione della cache (memoria interna) determina il costo del processore ma anche la sua velocità.
Nella cache vengono immagazzinati i dati con più alta probabilità di essere utilizzati o comunque quelli vicini (sfruttando il principio di località, il salto che il puntatore deve compiere alla locazione di memoria che sottende il dato deve essere il più breve possibile). Se la mole di dati utilizzata contemporaneamente dall'applicazione è superiore alla dimensione della cache del DSP, il processore non è adatto. Si dovrà fare, nella scelta della cache un trade-off tra costo e dimensione. I DSP fixed point hanno memoria interna ed esterna di dimensioni ridotte, mentre quelli di tipo floating point usano cache piccola e sono in grado pilotare una discreta quantità di memoria esterna. Sul mercato è possibile trovare DSP equivalenti con diversi tagli di memoria interna. Lo stesso dispositivo può essere prodotto con lo stesso processore, con identiche caratteristiche dal punto di vista della ALU, della clock rate, ecc. ma con diverse dimensioni di cache. Un'applicazione, nel giro di qualche anno, potrebbe richiedere requisiti computazionali maggiori, in ogni caso prevedibili; sarebbe auspicabile poter adattare il processore alle nuove esigenze di calcolo quindi, all'atto della progettazione del DSP si dovrà compiere anche uno sforzo di inventiva su quello che potrà essere lo sviluppo futuro dell'applicazione stessa, al fine di poter continuare ad utilizzare lo stesso core del dispositivo. La memoria quindi dovrà essere *scalabile*. Le applicazioni dovranno, altresì, essere compatibili con successive release (versioni) del DSP: solitamente la casa produttrice fornisce le previsioni sullo sviluppo che il DSP subirà (ad esempio in termini di clock rate).

I DSP devono essere corredati da **tools di sviluppo** che ne permettano l'implementazione e la diminuzione del "time to market", ovvero del periodo che intercorre tra il momento in cui inizia la progettazione di un dispositivo (sul quale installare un determinato DSP) e quello in cui avviene l'immissione del prodotto stesso sul mercato. Migliore sarà il tool di sviluppo maggiore sarà il time to market. L'ambiente di sviluppo può essere fornito direttamente dalla casa costruttrice del DSP o da una casa satellite (holding dell'aziemda produttrice). Gli strumenti di sviluppo possono essere suddivisi in queste categorie:

- COMPILER
- ASSEMBLER
- LINKER (ha la funzione di posizionare le routine generate dall'assemblatore in memoria)
- SIMULATOR (l'uso del simulatore è di fondamentale importanza: solitamente è un pacchetto software che da la possibilità a terzi di sviluppare applicazioni per quel determinato processore. Permette di avere a disposizione applicazioni nel momento in cui il DSP viene

commercializzato o implementato su schede).

- DEBUGGER (permette di far interagire programmatore e programma: avanzamento step by step, controllo dei registri e della memoria)
- LIBRERIE (operazioni già previste che possono essere considerate delle chiamate a funzioni)
- IN-CIRCUIT EMULATOR (permette di generare la simulazione direttamente su un apparato elettronico. E' possibile controllare che il DSP esegua veramente ciò che si è simulato: si va a valutare il comportamento dei registri e della memoria e lo si compara con ciò che prima era stato determinato via software)
- REAL TIME O.S.
- SCHEDE STARTER KIT con a bordo una versione tipica del DSP complete di interfacce di input-output adatte a settori applicativi che si prestano ad effettuare delle prove sperimentali.

Insieme al simulatore viene fornita anche la libreria di interfaccia all'host. La scheda di sviluppo "in-circuit emulator" permette di avere un'interazione diretta con il processore: la sua funzione è quella di consentire il monitoraggio del DSP in modo non invasivo, apponendo semplicemente dei morsetti fisici direttamente sul dispositivo. Nella programmazione di un DSP sarebbe meglio disporre di strumenti di sviluppo ad alto livello. Tipicamente i DSP vengono programmati in linguaggio *ASSEMBLER* (basso livello), anche se negli ultimi anni sono stati diffusi compilatori efficienti che sono in grado di raggiungere un medio livello di ottimizzazione delle risorse messe a disposizione del dispositivo. Sono sempre più diffusi compilatori, assemblatori o linker che permettono non solo di trasformare il codice di alto livello in linguaggio macchina, ma anche di immettere le routine o i dati nel punto della memoria che si ritiene più efficace. La programmazione ad alto livello di un DSP è un fattore determinante nella scelta del processore un fattore di successo per la sua immissione nel mercato. I tools di sviluppo, inoltre, devono necessariamente interagire con il linguaggio di programmazione al fine di rendere minore il time to market (ovvero il tempo che intercorre dalla progettazione del dispositivo alla sua immissione sul mercato), ad esempio, il debugger deve essere in grado di fornire il numero della linea di programma nel quale si è verificato un determinato errore, questa operazione deve essere possibile anche quando si passa dalla simulazione virtuale a quella sull'"in-ciruit emulator".

1.6.4. Supporto al multiprocessing

Altro criterio fondamentale, da tenere in considerazione nella scelta di un DSP adatto all'applicazione che si vuole svolgere, è il supporto al multiprocessing e cioè:

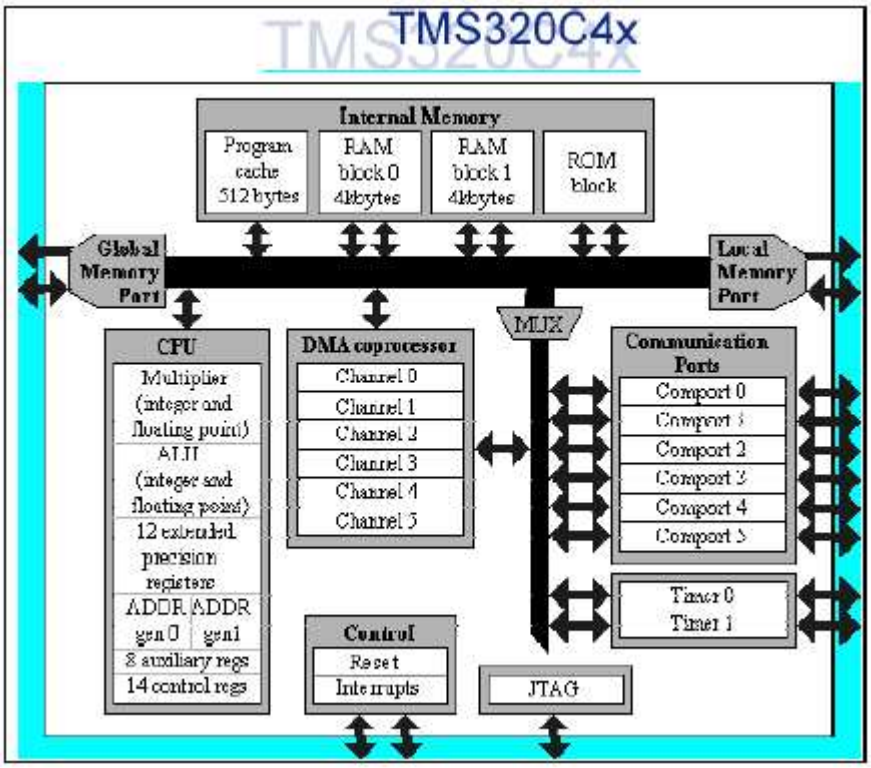
- SOLUZIONE SCALABILE
- RIUSABILITÁ
- PORTABILITÁ

In alcuni casi si verifica che i DSP non presentano da soli la capacità computazionale necessaria alle esigenze del programmatore, ovvero il processore non è in grado di far girare il programma alla velocità adeguata o comunque, con le caratteristiche necessarie all'applicazione. Inoltre questa potrebbe essere di tipo *scalabile* e la si vorrebbe, quindi, dimensionare a passi successivi (ad esempio potrebbero cambiare le dimensioni dei dati da manipolare e ciò potrebbe obbligare di adottare ogni volta un processore diverso). Inoltre la possibilità di supporto al multiprocessing offre anche il vantaggio di poter riutilizzare le considerazioni, le analisi o il codice di altri progetti. Questo fa sì che il time to market diminuisca. Un DSP che supporti il multiprocessing deve presentare facilità di *interconnessione*. Una scheda che supporti più processori dello stesso tipo deve essere di facile progettazione, in modo particolare non deve essere elaborata la circuiteria di interconnessione tra i DPS. E' anche possibile che la comunicazione tra processori necessiti di altri dispositivi, magari pensati per far comunicare fra loro le CPU dei DSP, come nel caso del *chipset*. Il mercato si sta indirizzando verso la specificità dei processori, ovvero vengono realizzati dispositivi dedicati, chiamati LINKING PROCESSOR (calcolo computazionale, comunicazione con la memoria principale, comunicazione tra processori). L'uso di questa metodologia permette di abbattere il costo di realizzazione di un nuovo DSP. Altro parametro pertinente al discorso affrontato è la

performance di interconnessione. Se su una scheda si usassero due processori non progettati per lo scambio reciproco di dati, la loro prestazione di comunicazione potrebbe non essere efficiente al massimo. La velocità, il throughput, la velocità di trasferimento e di risposta di un processore rispetto all'altro, il sovraccarico e la latenza sono gli elementi da considerare nella comunicazione tra processori: dovrà esistere un protocollo di comunicazione che soddisfi determinate caratteristiche.

- FACILITÁ DI INTERCONNESSIONE
- TIME TO DESIGN INTERPROCESSOR COMMUNICATION CIRCUITRY
- COST OF LINKING PROCESSORS
- PERFORMANCE DI INTERCONNESSIONE
- THROUGHPUT
- OVERHEAD
- LATENCY

Un esempio è il TMS320C4x, un processore della famiglia C4 della Texas Instruments, appositamente progettato per essere utilizzato in più di un unità su schede.



Viene fornito con due set separati di indirizzi di dati (2 address bus e 2 data bus): uno considerato "private" (memoria riservata) e uno per la memoria globale. La memoria globale viene utilizzata come area di scambio tra i processori. Il dispositivo, a seconda delle versioni, è già dotato di un chip di comunicazione parallela a 4 o 6 porte a 8 bit. Il dispositivo era stato progettato con la possibilità di essere collegato ad altri uguali in cascata per eseguire calcoli in parallelo (ogni processore veniva considerato come un atomo di computazione o nodo di calcolo, tipico dei *transuputer*).

1.6.5. Power Consumption Management

Un DSP viene spesso utilizzato per la creazione di dispositivi mobili, portatili. Oltre avere elevata capacità computazionale deve offrire bassi consumi di energia. La curva del carico di lavoro, generalmente, per un processore è tale da presentare dei picchi (***idle***) e dei momenti di quasi azzeramento dell'attività computazionale (***sleep***). Un buon processore dovrebbe essere in grado di interpretare il carico di lavoro e quindi determinare automaticamente il consumo energetico, questo significa disporre di un circuito che offra la possibilità di lavorare con diverse prestazioni e a

diverse "*rate*" di sintonia (si utilizzano dei divisori di clock programmabili che permettano di lavorare a diverse clock rate). Il processore centrale deve prevedere, per quanto riguarda il "*power consumption management*", di decidere quando spegnere o riaccendere alcuni dei dispositivi che ha intorno (come avviene per la gestione degli hard disk all'interno di un computer). Ricapitolando i fattori da tenere in considerazione per la gestione della dissipazione della potenza sono:

- RIDOTTO VOLTAGGIO DI FUNZIONAMENTO
- MODI DI FUNZIONAMENTO "SLEEP" O "IDLE"
- DIVISORI DI CLOCK PROGRAMMABILI
- CONTROLLO ALIMENTAZIONE PERIFERICHE

1.6.6. Costo

Altro fattore determinante nella scelta di un processore è il suo costo. In prima istanza deve essere proporzionale al costo del dispositivo sul quale lo si vuole installare. Si deve sempre tener conto di ciò che si vuole andare a realizzare o progettare con il DSP e per quanto tempo si vuole che il dispositivo realizzato rimanga sul mercato (possibilità di upgrade). Nella progettazione di nuovi processori si deve tenere conto della compatibilità con quello che è stato creato in passato. Spesso questo è un fattore che limita le potenzialità del nuovo processore ma serve a garantire la potenzialità e l'usabilità di ciò che è stato già creato. Il ciclo di vita del prodotto deve essere valutato oculatamente. Fattori che possono incidere sul costo del dispositivo sono:

- FLESSIBILITÀ il costo di un processore è inversamente proporzionale alla flessibilità dello stesso; questo potrebbe comportare la vetustà del dispositivo o l'elevato costo da sostenere per lo sviluppo delle applicazioni.
- QUANTITÀ DI MEMORIA ON-CHIP / PRESTAZIONI il costo di un processore è direttamente proporzionale alla quantità di memoria che verrà utilizzata o installata direttamente sul chip.
- PACKAGING stessi prodotti posso presentarsi in modo assai diverso a seconda dell'uso che si farà di essi. Un dispositivo può presentare le stesse caratteristiche di calcolo computazionale ma essere destinato ad ambienti diversi che presentano disomogenee caratteristiche: umidità, temperatura, vibrazioni, ecc. Lo stesso processore può essere anche fornito in dimensioni diverse (solitamente tre tagli) a seconda della sua destinazione: un DSP inserito all'interno di un telefono cellulare dovrà essere piccolo e leggero.
- DESIGN UPDATING le casse costruttrici dei DSP caricano sul costo del dispositivo anche le potenzialità in aggiornamento dell'architettura .
- QUANTITÀ DI PEZZI il prezzo può variare sostanzialmente in rapporto al numero di pezzi prodotti e/o acquistati. Sui siti internet dei produttori di DSP infatti si possono leggere le variazioni di prezzo in base al numero desiderato. Più processori si acquistano minore sarà il prezzo del singolo pezzo. I costruttori di DSP producono linee di architettura anche per svariati anni, in accordo con l'impatto del prodotto sul mercato o di quello che pensano sarà lo scenario futuro, garantendo aggiornamenti (ad esempio dopo quanto tempo verrà rilasciato un processore uguale ma con clock rate duplicata, con più memoria on chip, ecc). Nel momento in cui viene dimessa la produzione del processore il suo prezzo, normalmente, crolla. Questo potrebbe essere un fatto vantaggioso per gli sviluppatori: per le cui applicazioni risulta utile il DSP non più prodotto (di cui oramai esistono molte librerie e applicazioni in circolazione).

1.6.7. Fattori determinanti

Per fattori determinanti si intendono le caratteristiche che contraddistinguono in maniera rilevante i diversi DSP. Sono:

- UNITÀ DI ESECUZIONE IN PARALLELO inteso come numero di unità che eseguono una parte dell'applicazione: pipeline, ovvero quante operazioni vengono svolte in parallelo contemporaneamente
- VLIW Very Long Instruction Word: sono i processori in grado di realizzare ottimizzazione del codice e l'esecuzioni di più operazioni nella stessa istruzione in modo automatico.

- SUPERSCALARITÀ – chip DSP potenti nei quali è realizzata una sorta di pipeline hardware
- SIMD – Single Instruction Multiple Data
- DIMENSIONE DELLA INSTRUCTION-WORD
- ISTRUZIONI RISC VS ISTRUZIONI COMPLESSE
- VELOCITÀ DI TRASFERIMENTO DALLA/ALLA MEMORIA
- PIPELINE
- ACCELLERATORI HARDWARE – hardware dedicato all'espletamento di una operazione che altrimenti terrebbe troppo impegnata la risorsa principale
- VELOCITÀ DI CLOCK

1.7. Evoluzione dei Dsp

L'evoluzione dei DSP ha visto tre passaggi storici: i primi, detti *tradizionali* o di *prima generazione* avevano la potenzialità di eseguire un'operazione di tipo MAC in un colpo di clock (relativi all'architettura Harvard), si sono evoluti nei *DSP convenzionali migliorati nelle prestazioni*.
Il gradino successivo della scala ha visto la progettazione di due diverse architetture:
la *VLIW* (Very Long Instruction Word) e la *superscalare*. Una terza via è stata l'evoluzione dei processori *general pourpose con applicazioni DSP*.
Dopo la prima generazione di DSP, quindi, si sono avute le seguenti evoluzioni, ognuna delle quali ha preso strade diverse:

- DSP "CONVENZIONALI" MIGLIORATI (Lucent DSP16xxx, ADI ADSP-2116x)
- VLIW (VERY LONG INSTRUCTION WORD) (TI TMS320C6xx, Siemens Carmel)
- SUPERSCALARI (ZSP ZSP164xx)

PROCESSORI GENERAL PURPOSE CON ESTENSIONI DSP (PowerPC with AltiVec, TriCore, Pentium MMX)

1.7.1. DSP Convenzionali

Presentano le seguenti caratteristiche comuni:
1 ISTRUZIONE PER CICLO-MACCHINA ("SINGLE ISSUE")

- HARDWARE DEDICATO PER L'INDIRIZZAMENTO, MODI DI INDIRIZZAMENTO SPECIALIZZATI
  Ad esempio, a volte, le locazioni di memoria potrebbero essere indirizzate in modo circolare, cioè con la possibilità di avere un puntatore a memoria che una volta raggiunta la massima locazione, ritorna all'inizio dell'area di memoria disponibile: questo tipo di processo può essere svolto da un dispositivo hardware.
- MEMORIA ON-CHIP AD ACCESSO MULTIPLO
- HARDWARE DEDICATO PER I LOOP, poiché i loop sono una delle parti principali degli algoritmi DSP.
- PERIFERICHE ED INTERFACCE I/O SPECIALIZZATE ON-CHIP spesso i DSP presentano sulle schede sulle quali sono montati delle porte seriali ad alta velocità che permettono la comunicazione con altri dispositivi installati (quali, ad esempio, la Jtag, l'hardware per il debuggin, tools di sviluppo).
- COSTO, DISSIPAZIONE IN POTENZA E MEMORIA RIDOTTI.

1.7.2. DSP convenzionali migliorati

Rispetto ai precedenti, presentano una serie di implementazioni che ne permettono un uso più coretto e specifico. L'architettura non subisce trasformazioni di rilievo. Le caratteristiche aggiuntive sono:

- GRADO DI PARALLELISMO SUPERIORE (2nd multiplier, adder)
- OPERAZIONI SIMD (limitate – ovvero perdendo in precisione di calcolo si stabilisce di usare un grado di parallelismo maggiore nelle operazioni logiche: con una word di 32 bit è possibile eseguire due operazioni a 16 bit con saturazione, overflow e ecc.)

- HARDWARE ALTAMENTE SPECIALIZZATO (application-oriented data path operations)
- CO-PROCESSORI (Viterbi decoding, FIR filtering, etc.)

Alcuni esempi di DSP convenzionali migliorati sono il Texas TMS320C3x, il Lucent DSP16xxx o l'ADI ADSP-2116x.
Il nuovo tipo di approccio comporta il vantaggio che i DSP consentono un miglioramento delle prestazioni poiché non vi è stato sforzo di progettazione, ma solo un adeguamento a quelle che sono le esigenze delle applicazioni. Vengono mantenuti i costi, la dissipazione in potenza e competitiva, la densità del codice. Si ha, però, l'introduzione di nuovi svantaggi, quali:

- aumento della complessità, architetture difficili da programmare
- difficili target per i compilatori
- miglioramento ottenibile è limitato

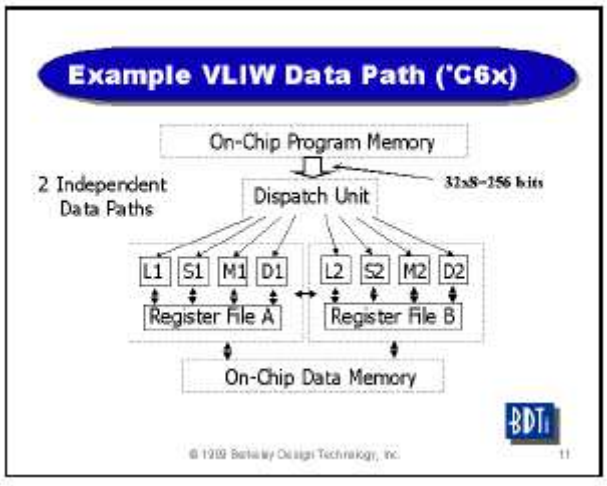
Gli svantaggi elencati hanno come conseguenza diretta quella di generare difficoltà nella realizzazione di applicazioni efficienti. Il programmatore, per utilizzare contemporaneamente queste caratteristiche e per poter sfruttare il maggior grado di parallelismo, deve necessariamente rinunciare a qualcosa: ad esempio, si può trovare nell'esigenza di disporre i dati secondo un certo pathern in memoria per renderli accessibili ai nuovi dispositivi in modo efficace, quindi di dover spendere cicli perché questi siano disposti nel modo prestabilito. Il prezzo da pagare potrebbe essere quello di tempo in più per lo sviluppo dell'applicazione. È abbastanza difficile, di conseguenza, ottenere delle ottimizzazioni automatiche: è complicato per questo tipo di DSP disporre di ambienti di sviluppo in grado di fornire codice ottimizzato. L'ulteriore evoluzione è stata quella di uno sforzo progettuale al fine di far corrispondere le istruzioni della memoria con le unità di esecuzione disponibili sul processore. Nascono quindi le architetture VLIW e superscalari. Quello che si cerca di ottenere è il cosiddetto ILP: Instruction Level Parallelism, al fine di ottenere una parallelizzazione delle istruzioni.

1.7.3. VILW (Very Long Instruction Word)

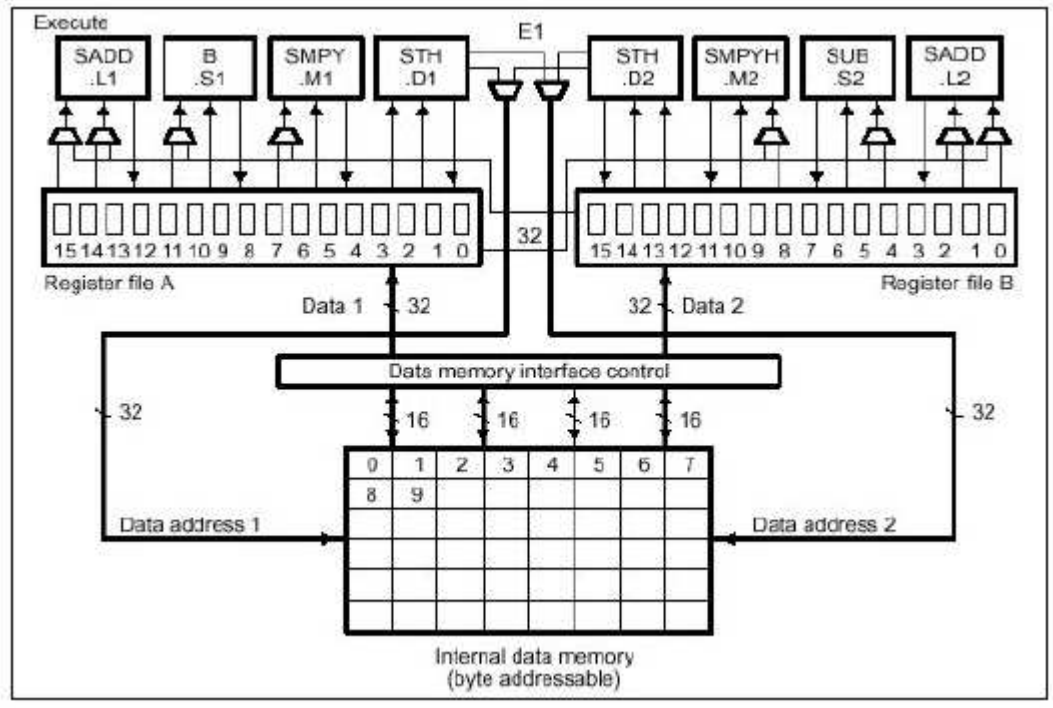
L'approccio vliw ha come caratteristica di prendere dalla memoria diverse istruzioni contemporaneamente e di considerarle come una singola. Vliw è riferito al fatto di voler considerare blocchi da 4 o da 8 istruzioni alla volta come una singola istruzione, questo è reso possibile da una certa disposizione delle stesse in memoria. Il grosso del lavoro di parallelizzazione viene eseguito dal compilatore. Nella filosofia vliw, l'architettura viene sviluppata parallelamente al compilatore e, in pratica, questo si occuperà di svolgere gli affiancamenti delle istruzioni in memoria in modo da non creare conflitti o data hazard (lettura di dati prima che questi siano effettivamente disponibili). Se vengono caricate n istruzione alla volta significa che ci sono n unità di esecuzione e, tra queste ce ne saranno un buon numero eseguibili in parallelo, già predisposte dal compilatore. Questo è possibile grazie alla rappresentazione interna dei dati e dell'uso di un'unità di controllo che decide l'esecuzione delle istruzioni. Il parallelismo non viene risolto via hardware, ma, se si vuole, via software o preventivamente all'esecuzione. Le architetture vliw presentano le seguenti caratteristiche:

- OPERAZIONI INDIPENDENTI MULTIPLE PER CICLO-MACCHINA, RAGGRUPPATE IN UNA SINGOLA "LARGE INSTRUCTION" O "INSTRUCTION PACKET"
- ARCHITETTURE REGOLARI, RISC-LIKE OPERATIONS
- AMPIO SET DI REGISTRI

Alcuni esempi di architetture VLIW, per applicazioni DSP, sono: TI TMS320C6xx, Siemens Carmel, ADI TigerSHARC.



I dati seguono due differenti percorsi hardware, ognuno dotato di un set di registri, di un register file e di quattro unità di esecuzione. Il C6 carica 8 istruzioni alla volta, ognuna di 32 bit, quindi la sua long instruction word è composta da 256 bit. Le istruzioni devono essere disposte in maniera tale che l'unità di esecuzione non debba eseguire operazioni che riguardano la singola istruzione, ma il blocco di istruzioni stesse. Questo è un notevole carico per la progettazione circuitale. Avere unità di esecuzioni semplici porta il DSP ad essere meno costoso di uno superscalare: questo è il motivo per il quale il mercato tende verso la filosofia vliw. L'uso dei registri comporta dissipazione di potenza.



Usare DSP vliw comporta i seguenti vantaggi:

- INCREMENTO DELLE PRESTAZIONI
- ARCHITETTURE PIÙ REGOLARI (potenzialmente più facili da programmare, migliori target per compilatori)
- SCALABILI – ovvero la capacità di aumentare il numero di unità logiche, senza stravolgere le architetture. Nella figura è riportato lo schema dell'ultima evoluzione del C6, si vede come il numero di unità di esecuzione sia sensibilmente aumentato.
- I DSP vliw presentano anche una serie di svantaggi:

- NUOVO TIPO DI COMPLESSITÀ PER IL PROGRAMMA/COMPILATORE la programmazione a "mano" delle applicazioni risulta assai più complicata: o il programmatore (o il tool di generazione del codice) deve tenere traccia dello scheduling delle istruzioni, o la presenza di pipeline molto lunghe e latenze elevate possono rendere le prestazioni di picco non indicative
- AUMENTO NELLA DIMENSIONE DEL CODICE – l'implicazione diretta è la necessità di elevata velocità di trasferimento dalla/alla memoria
- ELEVATA DISSIPAZIONE IN POTENZA.

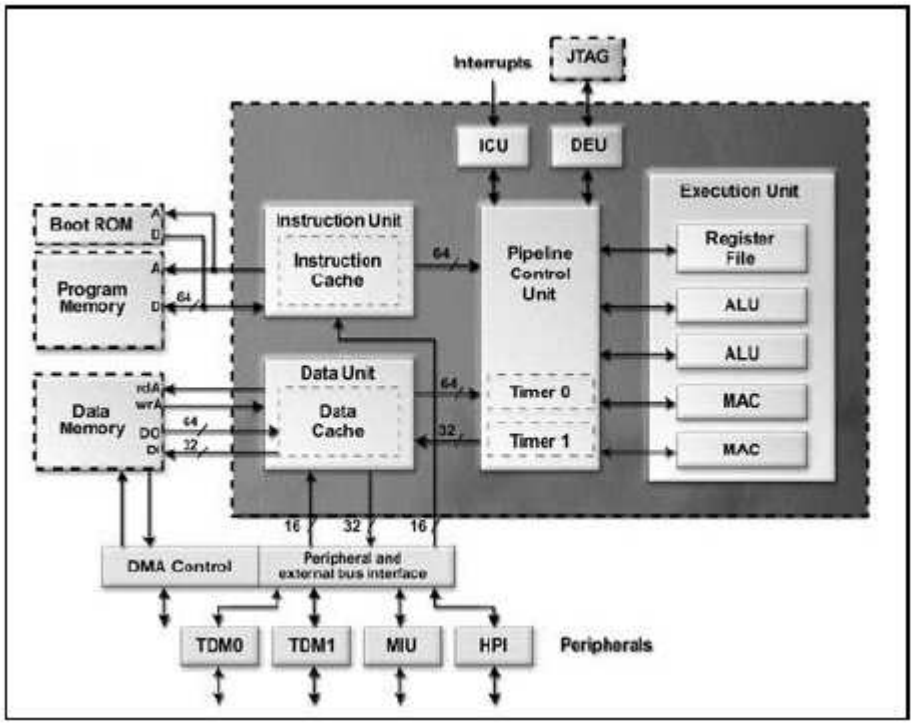
1.7.4. Architetture superscalari

Nell'approccio superscalare l'esigenza dell'ILP è risolto via hardware. In pratica, esiste un meccanismo di individuazione della dipendenza dei dati run time, durante l'esecuzione del codice. Le istruzione vengono disposte in modo tale da essere eseguite in parallelo. La metodologia descritta apparteneva alla corrente di ideatori dei supercalcolatori, dei computer ad alto potere computazionale. Con questa si raggiunge un elevato grado di parallelismo, paragonabile a quello dei vliw.

Le loro caratteristiche possono essere riassunte nei seguenti punti:

- UTILIZZANO LE TECNICHE DELLE CPU HIGH-END
- ISTRUZIONI MULTIPLE ESEGUITE PER CICLO-MACCHINA
- RISC-LIKE INSTRUCTION SET
- ALTO GRADO DI PARALLELISMO

Alcuni esempi di architetture superscalari per applicazioni DSP sono: ZSP ZSP164xx, Siemens TriCore.



Le architetture superscalari presentano i seguenti vantaggi:

- ALTISSIMO INCREMENTO NELLE PRESTAZIONI
- ARCHITETTURE PIÙ REGOLARI (potenzialmente più facili da programmare, migliori target per i compilatori)

- IL PROGRAMMATORE (o il tool di generazione del codice) NON DEVE TENERE CONTO DELLO SCHEDULING DELLE ISTRUZIONI alcune delle difficoltà di scheduling vengono assorbite dall'hardware del processore: le unità di esecuzione, al momento dell'esecuzione delle applicazioni o delle istruzioni del codice, sono in grado di decidere autonomamente se andare avanti (all'istruzione successiva) oppure attendere risultati provenienti da altre unità.
- LA DIMENSIONE DEL CODICE NON AUMENTA IN MANIERA SIGNIFICATIVA
- ALTISSIMO CONSUMO DI ENERGIA – la circuiteria interna del processore aumenta in modo sensibile, quindi ci saranno più parti da alimentare.
- IL COMPORTAMENTO DINAMICO COMPLICA LO SVILUPPO DEL SOFTWARE a meno che non si utilizzi un processore più potente rispetto ai requisiti richiesti, non si sa se alcune istruzioni verranno eseguite dopo altre (data hazard). Il processore decide se far eseguire l'istruzione successiva in base ai conflitti che si verificano internamente o in accordo con le tabelle che vengono implementate di volta in volta (scoreboard) nelle quali si tengono in considerazioni i data hazard. Questo introduce una variabile di indeterminazione nello sviluppo del software. L'ottimizzazione del codice risulta assai complicata. Per superare questo problema è necessario utilizzare, come si affermava precedentemente, un DSP a capacità di calcolo superiore rispetto alle esigenze dettate dall'applicazione: questa filosofia è in antitesi con quello che un programmatore o uno sviluppatore ricerca da applicazioni DSP, ovvero rintracciare un processore ritagliato per il tipo di implementazione che si andrà a sviluppare. L'utilizzo di processori superscalari comporta un aumento dei costi ed una diminuzione delle prestazioni (rispetto al paradigma costo – prestazioni).
- LA VARIABILITÀ DEL TEMPO DI ESECUZIONE PUÒ DIVENTARE UN PROBLEMA
- L'OTTIMIZZAZIONE DEL CODICE È DIFFICILE.

1.7.5. Processori General Purpose con estensioni DSP

I processori general purpose ad alte prestazioni per PC e workstation stanno diventando sempre più adatti per task DSP. Esempi di processori general purpose con estensioni DSP sono:

- MMX PENTIUM
- POWERPC WITH ALTIVEC

In pratica il set delle istruzioni di questi processori è stato allargato al fine di migliorare e di supportare applicazioni multimediali. L'estensione non ha sotteso un cambio dell'architettura, ma ha determinato scelte diverse per l'hardware disponibile: ad esempio nel Pentium MMX alcuni registri utilizzati per calcoli floating point sono stati utilizzati per realizzare operazioni parallele, a minor precisione rispetto alla potenzialità della ALU. La differenza sostanziale tra DSP e processori general purpose consta nel fatto che i secondi hanno clock rate elevatissime (attualmente siamo al di sopra del Gigahertz). I general purpose sono superscalari, le istruzioni vengono caricate scomposte in RISC e run time (in tempo di esecuzione) il processore decide cosa far procedere. I general purpose hanno elevata velocità di trasferimento da e verso la memoria e presentano la branch prediction cache (BPC), che in modo statistico stabilisce dove il codice andrà a finire nello stack della memoria e cioè dove il program counter andrà a posizionarsi dopo avere eseguito un'istruzione di salto (risparmio di tempo): il codice viene, in pratica, ottimizzato. Riassumendo, i general purpose con estensioni DSP presentano le seguenti caratteristiche:

- ALTISSIME CLOCK RATES
- SUPERSCALARI (MULTI-ISSUE)
- MOLTIPLICAZIONI ED ALTRE OPERAZIONI ARITMETICHE IN 1 CICLO-MACCHINA
- ELEVATA VELOCITÀ DI TRASFERIMENTO DALLA/ALLA MEMORIA
- BRANCH PREDICTION
- SPESSO HANNO ESTENSIONI SIMD
- Il loro utilizzo comporta i seguenti vantaggi:
- ALTISSIME PRESTAZIONI DSP

- PROCESSORI GIÀ PRESENTI SUI PC che implica largo target di destinatari per l'applicazione che si realizzerà (ad esempio tool per videoconferenza oppure per la telefonia via internet sono applicazioni DSP, in passato per avere le stesse caratteristiche era necessario installare una nuova scheda all'interno del computer)
- SONO GIÀ DISPONIBILI TOOL PER I PIÙ DIFFUSI quindi è possibile realizzare applicazioni a basso costo.
- IL RAPPORTO COSTO-PRESTAZIONI PUÓ FARE CONCORRENZA AI FLOATING-POINT DSP
- Ma anche i seguenti svantaggi:
- NON È POSSIBILE PREVEDERE IL TEMPO DI ESECUZIONE (può causare problemi nelle applicazioni real-time).
- LO SVILUPPO DI CODICE DSP OTTIMIZZATO È DIFFICILE
- POCHI TOOL PER LO SVILUPPO DSP DISPONIBILI pur esistendo un numero elevato di applicazioni, queste non sono state pensate per i DSP; possono nascere problemi nella gestione della memoria e delle altre periferiche e nell'acquisizione di segnali. I tool messi a disposizione con i DSP solitamente presentano delle "facilities" intorno all'ambiente di sviluppo.
- ELEVATA DISSIPAZIONE IN POTENZA poiché non sono dispositivi pensati per essere alimentati in modo mobile. Inoltre presentano molti registri e utilizzano un'elevata quantità di memoria, motivo, questo, anche delle loro performance.
- IL RAPPORTO COSTO-PRESTAZIONI NON PUÒ CONCORRERE CON QUELLO DI UN DSP FIXED-POINT

Una successiva evoluzione nel campo delle architetture DSP è rappresentata dall'*INTEGRAZIONE*. Gli schemi di progettazione dei core dei DSP vengono rilasciati dalle case costruttrici affinché altri possano impiantare direttamente il processore su sistemi integrati (system on chip). In sostanza, i processori DSP vengono utilizzati come parti di altri processori.
Ad esempio la Intel, casa costruttrice del Pentium, potrebbe acquistare il core di un DSP della Texas Instruments ed impiantarlo direttamente sull'architettura del Pentium. Il DSP non è più a se stante ma entra a far parte dell'architettura del sistema. I produttori di questi chip realizzano blocchi altamente specializzati per determinate applicazioni. Naturalmente, il costruttore dovrà valutare il mercato che un tale dispositivo potrà avere, a causa della sua alta specializzazione. L'integrazione comporta i benefici dei general purpose associati a quelli dei DSP convenzionali. Probabilmente, in futuro, ci si orienterà verso questo tipo di simbiosi. Attualmente, ad esempio, sull'architettura degli hard disk viene implementato lo Z80, che lavora a supporto del processore principale al fine dell'ottimizzazione di alcuni task. Un ulteriore tassello dell'evoluzione dei DSP è rappresentato dallo sviluppo parallelo dell'interazione tra architettura hardware e tool di sviluppo. Si è già visto quanto questi ultimi siano importanti nella scelta di un DSP. Generalmente, è meglio scegliere un DSP supportato da un ambiente potente piuttosto che un altro più potente ma con un ambiente meno sviluppato, poiché ha una potenza non è facilmente sfruttabile. L'interazione tra tool di sviluppo ed hardware non ha solo il significato di elaborare software che ne faciliti l'usabilità, ma anche prevedere nell'architettura delle porte o dei blocchi che siano studiati per interagire dinamicamente con il software. Le porte che operano questa specifica sono dette porte JTAG e, oramai, sono implementate su quasi tutti i DSP presenti sul mercato.
La possibilità di avere un debug in tempo reale, migliori compilatori C, compilatori C ottimizzati per determinate architetture e quindi l'esigenza sempre più ridotta di ricorre all'Assembly se non per pezzi ridotti di codice sono gli ingredienti che costituiscono i fattori di successo per un DSP. Riassumendo, quindi, l'evoluzione dei tool per lo sviluppo di applicazioni DSP comporterà:

- LA DIFFUSIONE DELLE PORTE ON-CHIP JTAG-COMPATIBILI
- MIGLIORAMENTO GENERALE DELLE RISORSE DI DEBUG
- IL SUPPORTO PER IL DEBUG REAL-TIME
- L'UTILIZZO SEMPRE PIÙ MASSICCIO DI COMPILATORI C, ANCHE SE LE PRESTAZIONI DI PICCO RICHIEDONO SEMPRE L'ASSEMBLY

Attualmente il mercato si sta dirigendo verso il potenziamento dei punti sopra elencati. Sono già disponibili gli "instruction set simulators" di tipo *cycle accurate*, ovvero dei simulatori che, prima di avere l'architettura, rendono possibili delle valutazioni su quanti cicli verranno spesi dal codice prima di completare un determinato task. Questo può facilitare lo sviluppo del software. La tendenza è quella di fornire ambienti integrati di sviluppo implementati da compilatore, debugger, assemblatore e jtag (nei general purpose questo accade con il Visual C++). Lo scopo è quello di avere un'interfaccia uniforme, cioè che vada bene per i diversi DSP che evidenzi però la differenza tra le varie unità di esecuzione.

2. Esempi di DSP commerciali Floating-Point

Texas Instrument TMS320C6711 ed Analog Device ADSP 21065L

2.1. Introduzione

Man mano che i processori diventano più potenti, sia in termini di frequenza di clock che di parallelismo a livello di istruzioni, cresce il divario tra la velocità dei processori e quella delle memorie e, con esso, il collo di bottiglia dovuto agli accessi alla memoria. In genere la soluzione a questo problema viene dall'utilizzo di più livelli di cache sempre più grandi. Finché si tratta di processori general-purpose che eseguono applicazioni generiche, è ampiamente dimostrato che questa soluzione funziona in modo soddisfacente, ma quando si tratta di codici specifici, come elaborazione dei segnali, la situazione è diversa. In questi casi, infatti, si ha a che fare con codici che presentano uno scarso, a volte nullo, fattore di riutilizzo dei dati. Questo si traduce in esecuzioni che generano un cache miss dietro l'altro e presentano una quantità di computazione troppo bassa per ammortizzare il costo dei miss stessi. Per limitare gli effetti negativi di questo comportamento si ricorre al data prefetching, il cui scopo è far arrivare alcuni dati in cache prima che il programma ne abbia bisogno, in modo da ridurre quanto più è possibile il numero di cache miss che non possono essere ammortizzati dal rischeduling runtime del processore.
Quindi per fare data prefetching su processori DSP bisogna tenere presente che le condizioni al contorno sono profondamente diverse rispetto al caso dei general-purpose. Infatti mentre lì si tratta di sfruttare meglio la cache, in questo contesto bisogna gestire la memoria dati e il controller DMA interni.
In questo scenario non bisogna cercare nel codice i riferimenti alla memoria che provocheranno *cache miss*, ma, dopo aver valutato quali strutture dati possono essere allocate staticamente sulla memoria interna, si devono opportunamente suddividere le restanti strutture dati, allocate in memoria esterna, in blocchi che verranno portati su buffer temporanei nella memoria interna, in modo da essere disponibili prima che il programma li richieda. Ovviamente quando viene richiesto un dato che ancora non è disponibile in memoria interna, la CPU dovrà aspettare che arrivi. Mancando un meccanismo di caching, bisognerà, quindi, gestire esplicitamente la sincronizzazione tra CPU e DMA in modo da garantire la correttezza del programma e l'attesa minima da parte della CPU. Inoltre bisognerà anche occuparsi della coerenza delle informazioni, quindi i dati modificati dovranno essere ricopiati nella memoria esterna prima di poter riutilizzare il buffer temporaneo in cui erano stati precedentemente portati.

Quindi, nel caso DSP, sostanzialmente si tratta di effettuare degli spostamenti di dati tra i vari livelli della gerarchia di memoria per portarli il più vicino possibile al processore, in previsione del loro

utilizzo. A questo scopo bisogna trasformare il codice sorgente, inserendo delle chiamate a funzioni che programmano direttamente il controller del DMA e realizzano trasferimenti di dati in sovrapposizione al calcolo. L'idea è di creare una sorta di pipeline tra la computazione e il trasferimento dei dati, in modo che il numero di elementi portati in memoria interna sia tale da tenere il più possibile impegnata la CPU durante il successivo trasferimento.
Chiaramente l'algoritmo dovrà avere anche la possibilità di decidere di volta in volta che taglia devono avere i blocchi di dati da trasferire. Questo oltre che dal particolare codice che si deve eseguire, dipende da una serie di altri fattori. Infatti da un lato bisogna mantenere basso il traffico tra i due livelli di memoria, ma dall'altro i buffer temporanei non possono essere molto grandi perché ci possono essere dati acceduti spesso che necessitano di essere allocati sulla memoria interna. Ad esempio, nel caso di una moltiplicazione tra una matrice e un vettore la situazione ideale sarebbe di poter allocare il vettore sulla memoria interna e di portare di volta in volta un certo numero di righe della matrice su un buffer temporaneo. Quindi per poter stabilire in che punto del codice inserire l'istruzione che dà inizio al trasferimento di un blocco, bisogna sviluppare un modello di costo che tenga conto della latenza delle varie istruzioni, dei tempi di accesso ai due tipi di memoria, della velocità del DMA, delle dimensioni della memoria interna e del bilanciamento (trade off) tra dimensione dei buffer temporanei e numero di trasferimenti.

I processori DSP sono molto adatti ad applicazioni embedded, quindi costituiscono il cuore di molti dispositivi in cui è necessario fare elaborazione real time di segnali su apparati qualsiasi, macchine ecc. Ad esempio è possibile trovarli in stampanti, dischi, modem, sistemi di controllo, telefoni cellulari, prodotti consumer in genere. Sono ottimizzati per algoritmi DSP, cioè filtri, finestrature, convoluzioni, trasformate di Fourier, trasformate di Laplace, etc. Questi lavorano tutti su segnali campionati ed hanno le seguenti caratteristiche: spendono la maggior parte del tempo di esecuzione in loop che ad ogni iterazione eseguono solo poche istruzioni, presentano un flusso di controllo indipendente dai dati, fanno molti accessi alla memoria e presentano uno scarso riutilizzo dei dati.

Siccome operazioni composte, tipo MAC (Multiply And Accumulate), sono frequentissime nei codici DSP, in genere i processori DSP le possono eseguire in un unico ciclo di clock, grazie ad un hardware specializzato. Per sfruttare questa capacità, però, occorre garantire un'adeguata banda passante nei confronti della memoria. Per questo motivo i DSP sono basati sull'architettura Harvard, cioè hanno due spazi di memoria separati, a cui si può accedere contemporaneamente tramite due bus distinti, uno per le istruzioni e uno per i dati. Alcuni presentano anche più data path, in modo da poter eseguire simultaneamente operazioni di tipo load/store o verso le unità di I/O.

Non sempre i DSP, viste le caratteristiche dei codici per i quali sono pensati, non hanno cache per i dati, ma prevedono una piccola memoria interna, in cui il programmatore può allocare ciò che ritiene più opportuno, in maniera statica o dinamica. In maniera statica bisogna decidere a priori quali dati vanno nella memoria interna e comunicarlo al linker con opportune direttive; in maniera dinamica, invece, bisogna esplicitamente programmare il controller del DMA, che di solito è interno al processore, per gestire trasferimenti di dati tra memoria esterna ed interna in runtime.
Per la loro natura embedded, questi processori devono rispettare alcuni vincoli su dimensione, peso e dissipazione di potenza, pur mantenendo basso il costo. Questo ha fatto sì che solo il progresso tecnologico degli ultimi anni permettesse la nascita di DSP di fascia alta. Per migliorare le performance di queste architetture, oltre ad aumentare la frequenza del clock, i progettisti hanno seguito principalmente due direzioni che mirano entrambe ad ottenere una maggiore quantità di lavoro per ogni ciclo macchina.

In un caso sono state aggiunte più unità funzionali parallele al data path ed è stato esteso il set di istruzioni per poter codificare più operazioni in una singola istruzione, seguendo la filosofia SIMD, un po' come è avvenuto nel mondo dei general-purpose con le estensioni MMX. L'idea qui è di

suddividere gli insiemi di dati in sottoinsiemi di pochi elementi su cui operare in parallelo con una sola istruzione capace di gestire più unità contemporaneamente. Alcuni DSP della linea SHARC della Analog Device seguono questo tipo di architettura.

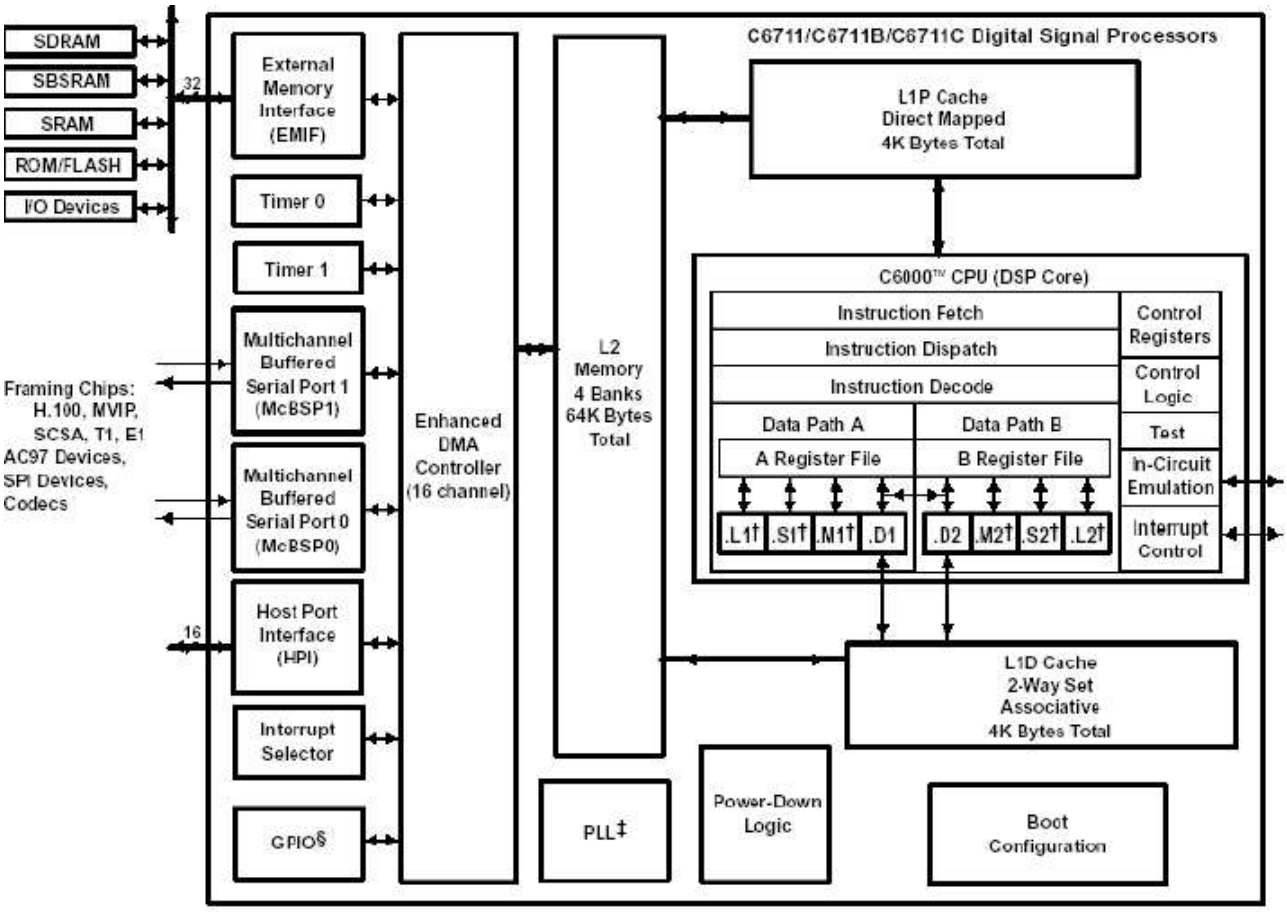
Nell'altro caso, invece, l'architettura è stata modificata in modo da permettere al processore di emettere più di un'istruzione per ogni ciclo di clock. Siccome un'architettura superscalare avrebbe richiesto delle sofisticazioni hardware che non avrebbero permesso di rispettare i vincoli fisici a cui deve essere soggetto un processore DSP, e siccome i codici DSP, a causa della loro staticità, non sfrutterebbero in pieno tali sofisticazioni, questa filosofia ha portato alla realizzazione di architetture VLIW. La sostanziale differenza tra architetture superscalari e VLIW è che, mentre nella prima un hardware dedicato rischedula in modo dinamico le istruzioni in runtime, nella seconda lo scheduling è demandato al compilatore e quindi è statico. Pertanto un'architettura VLIW è più efficiente in termini di consumi ma è più penalizzata rispetto alla latenza degli accessi in memoria. I DSP della linea C6000 della Texas Instrument sono un esempio di architettura VLIW.

Di pari passo con l'evoluzione architetturale c'è stata anche un'evoluzione degli strumenti di sviluppo software. Così, mentre i loro predecessori si programmavano quasi esclusivamente in assembler, i DSP di ultima generazione iniziano a supportare linguaggi ad alto livello come il C e ad essere dotati di una serie di tool che avvicinano l'ambiente di programmazione agli standard a cui è abituato il programmatore medio. Inoltre sono disponibili sul mercato sempre più librerie di funzioni ottimizzate ad hoc, spesso scritte in assembler ma con la possibilità di essere chiamate dall'interno di programmi C, sviluppate sia dai produttori che da terze parti.

Questi progressi aprono nuovi orizzonti all'utilizzo di questa categoria di processori. Infatti, da un lato la maggiore potenza permette di prenderli in considerazione per elaborazioni più pesanti, dall'altro la migliore programmabilità ne rende possibile l'utilizzo al di fuori del contesto monoapplicazione per cui erano stati originariamente creati.

2.2. DSP TMS320C6711 Texas Instrument

Come è noto alcune case produttrici di DSP, tra cui la Texas Instrument, stanno iniziando a immettere sul mercato dei DSP dotati di una piccola cache per i dati. A questo scopo la Quadrics Supercomputers World mette a disposizione il processore DSP TMS320C6711 che è la versione con cache del DSP TMS320C6701 nella linea di produzione C6000 della Texas Instrument. Di seguito viene mostrata la struttura interna del DSP TMS320C6711



Le caratteristiche principali di questo DSP sono veramente notevoli con unita funzionali indipendenti, 32 registri a 32 bit, un range di indirizzamento di 4Gbyte (32 bit), supporto a formato dei numeri sia big endian che little endian e, in ultimo, la possibilità di eseguire 900 milioni di istruzioni in floating point al secondo.

Questo DSP ha anche diverse periferiche integrate:

• Interfaccia verso memoria esterna a 32 bit (external memory interface EMIF):

• Interfaccia diretta verso memorie sincrone (SDRAM e SBSRAM)

• Interfaccia diretta verso memorie asincrone (SDRAM e EPROM)

• Controller DMA (enhanced direct-memory-access EDMA)

• Interfaccia verso un host a 16-bit (host-port interface HPI)

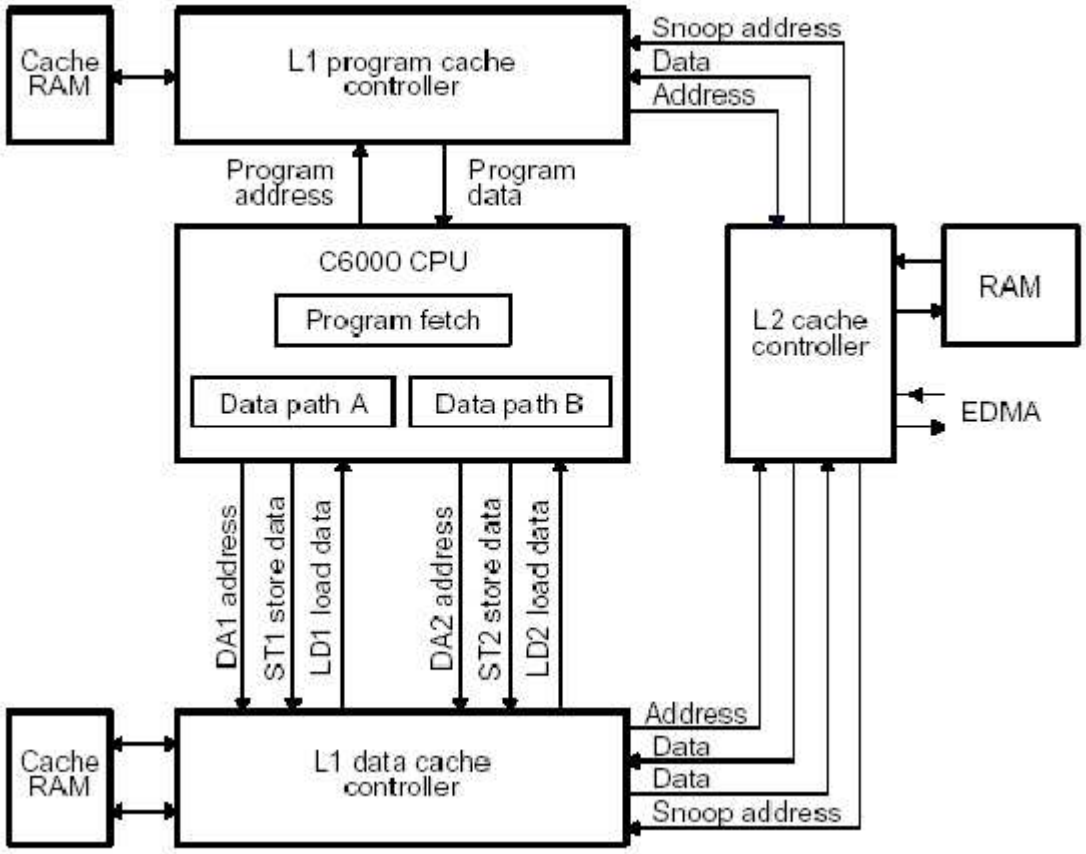
• Due seriali multistandard (multi-channel buffered serial ports McBSPs)

• Due timer a 32-bit

• Blocco di generazione del Clock con un PLL integrato

• Supporto alla IEEE-1149.1 (JTAG) per l'emulazione e il test

Il C6711 usa un'architettura basata su una cache a due livelli. La cache programma di primo livello (L1P) è da 32 Kbit, la cache dati di primo livello (L1D) è da 32 Kbit. La cache di secondo livello (L2) consiste di uno spazio di memoria di 512 Kbit che è diviso tra spazio di programma e di dati. La memoria L2 può essere configurata come una memoria mappata, una cache o una combinazione delle due. Il funzionamento del sistema, mostrato nella figura seguente, è riassumibile nel seguente modo: se l'istruzione da eseguire (o il dato da elaborare) non è nella cache di livello 1 (la più vicina alla CPU) allora viene cercata nella cache di levello 2. Se anche qui non c'è viene eseguito un accesso alla memoria esterna tramite EDMA. Si fa notare che a mano a mano ci si allontana dalla CPU cresce la lentezza della memoria, per cui gli accessi a livelli di memoria inferiori (L1-L2-Memoria esterna) sono sempre effettuati leggendo (o scrivendo) blocchi di dati.



La CPU contiene otto unità funzionali: quattro ALU virgola fissa/mobile, 2 ALU virgola fissa e due moltiplicatori virgola fissa/mobile; esse sono divise in due zone separate, la zona A e la zona B (come riportato nel diagramma dei blocchi funzionali della CPU), in ciascuna delle quali è anche contenuto un register file. Una zona contiene le unità funzionali L1,S1,M1 e D1; l'altra zona contiene le unità L2,M2,S2 e D2. I due register file contengono ciascuno sedici registri da 32 bits ciascuno, per un totale di trentadue registri per uso generale.

Le quattro unità funzionali di ciascuna parte della CPU possono liberamente dividersi i sedici registri appartenenti alla loro zona. Addizionalmente, ciascuna zona assegna un singolo bus dati connesso a tutti i registri dell'altra, tramite il quale i due set di unità funzionali possono accedere ai dati contenuti nei register file della parte opposta. Se l'accesso ai registri da parte delle unità funzionali avviene dalla stessa parte della CPU, il register file può servire tutte le unità in un ciclo di clock; se invece l'accesso ai registri avviene usando l'attraversamento del file register, la CPU supporta una scrittura e una lettura per ogni ciclo di clock.

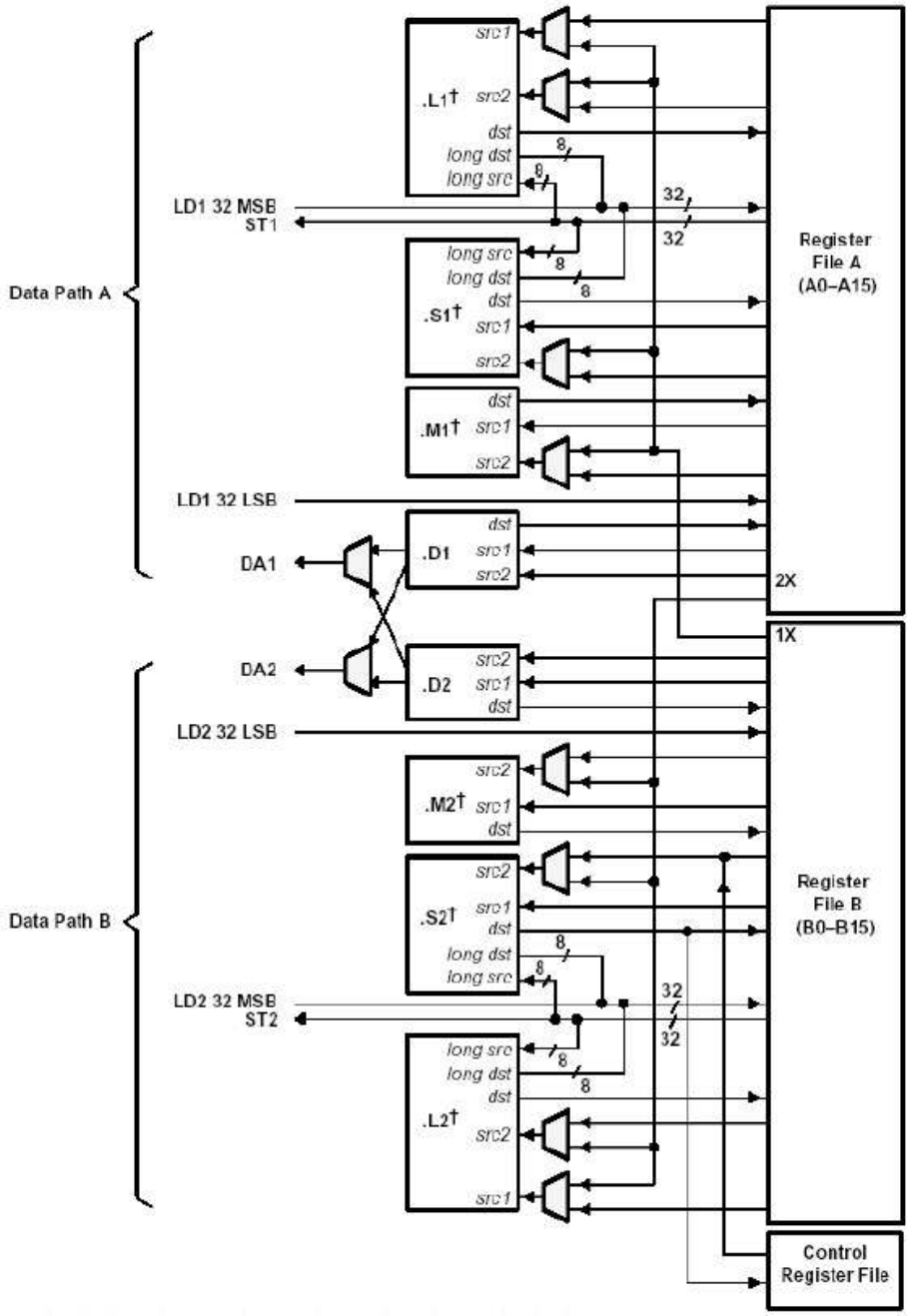
Un'altra caratteristica chiave delle CPU della famiglia C67 è l'architettura denominata "load/store" (che rappresenta una delle peculiarità delle architetture di tipo RISC), secondo la quale tutte le istruzioni lavorano sui registri (non accedono direttamente alla memoria).

Le due unità di indirizzamento dati (D1 eD2) sono responsabili di tutti i trasferimenti di dati tra i register file e la memoria. Gli indirizzi dei dati guidati dalle unità D, permettono agli indirizzi di dato generati da un file register di essere usati per caricare o immagazzinare (load o store) dati, in o da un altro file register. Le CPU C67X supportano una varietà di modi di indirizzamento indiretto, usando modi di indirizzamento circolare o lineare con un offset di 5 o 15 bits. Tutte le istruzioni sono condizionali e quasi tutte possono accedere ad ognuno dei 32 registri, tuttavia alcuni sono scelti per supportare specifici indirizzamenti o per memorizzare le condizioni per istruzioni condizionali (se la condizione non è automaticamente vera).

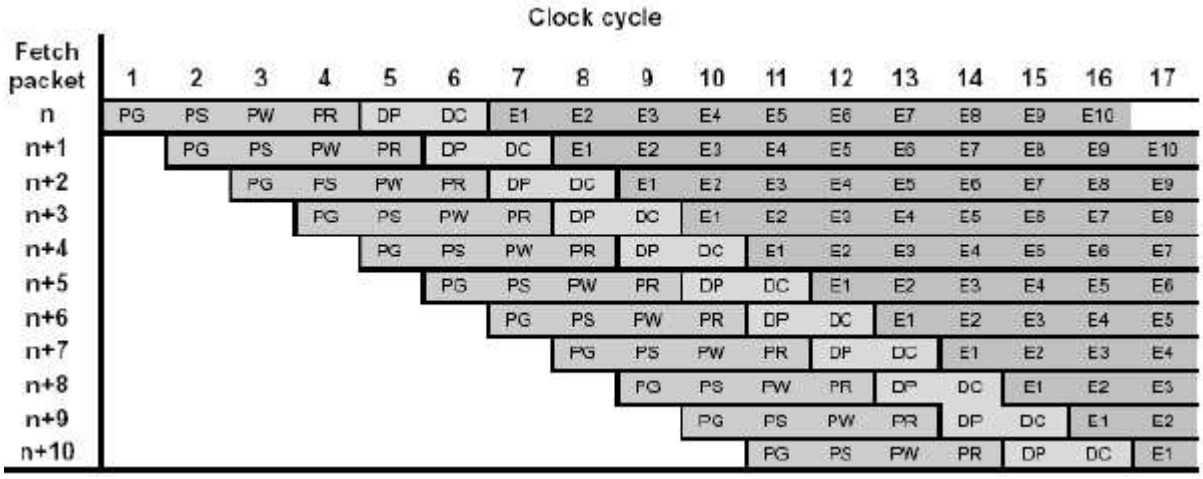
Le due unità funzionali M1 e M2 sono dedicate a moltiplicare. Le unità S1 e S2 , L1 e L2 realizzano un set di generiche funzioni: aritmetiche, logiche, di salto, con risultati disponibili ad ogni ciclo di clock. Tutte le unità M, S e L supportano operazioni floating point.

Il flusso di processo inizia quando un pacchetto di caricamento istruzioni lungo 256 bit viene caricato dalla memoria di programma. Le istruzioni lunghe 32 bit destinate alle unità funzionali individuali sono contrassegnate tramite un bit uguale a "1" nella posizione di bit meno significativo dell'istruzione (LSB). Le istruzioni che sono concatenate insieme per un'esecuzione simultanea (fino a otto in totale) compongono un pacchetto di esecuzione. Uno "0" nella posizione di LSB di un'istruzione rompe la catena, mettendo di fatto le istruzioni che seguono nel pacchetto di esecuzione successivo. Se un pacchetto di esecuzione è troppo lungo e oltrepassa la lunghezza del pacchetto di caricamento (256 bit), l'assembler lo posiziona nel successivo, mentre la parte rimanente del corrente pacchetto di caricamento è riempita con istruzioni NOP.

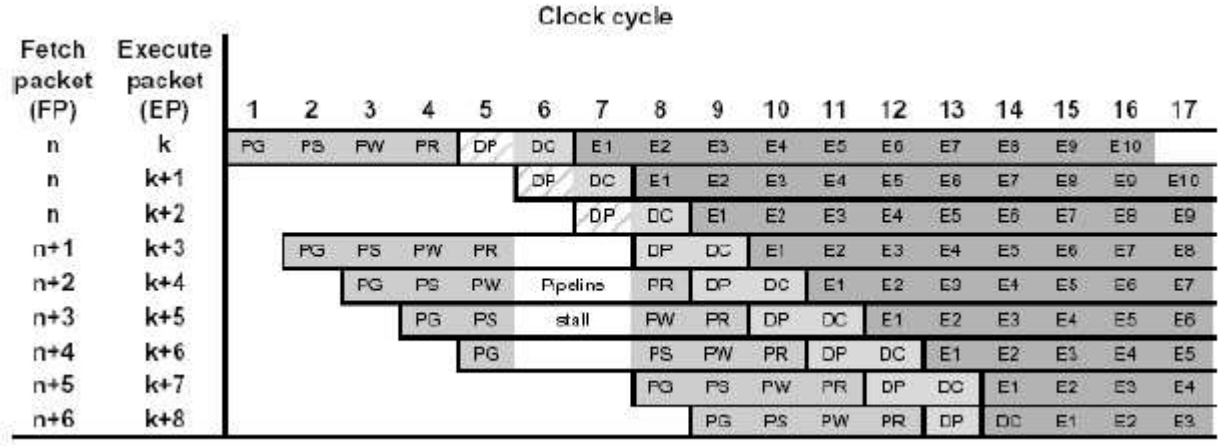
Il numero di pacchetti di esecuzione dentro un pacchetto di caricamento può variare da uno a otto e questi sono inviati alle loro rispettive unità funzionali con un ritmo di uno ogni ciclo di clock, il successivo pacchetto di caricamento di 256 bit non è caricato finché tutti i pacchetti di esecuzione del corrente pacchetto di caricamento non sono stati inviati. Dopo esser state decodificate, le istruzioni vengono simultaneamente spedite a tutte le unità funzionali attive per un ritmo massimo di esecuzione di otto istruzioni ogni ciclo di clock. Sebbene molti risultati siano immagazzinati nei registri a 32 bit, essi possono essere successivamente trasferiti in memoria come byte o come mezza parola. Tutte le istruzioni di caricamento e di immagazzinamento possono essere indirizzate a byte, a mezza parola o a parola intera.



La Pipeline del C6711 ha una struttura molto complessa vista la possibilità di eseguire fisicamente (cioè avere contemporaneamente nella fase di "execute") fino a 8 istruzioni. Inoltre ogni pacchetto da 8 istruzioni (256 bit) viene suddiviso al massimo in 16 fasi (4 di fetch, 2 di decodifica e 10 di esecuzione) per cui è teoricamente possibile elaborare 16 pacchetti da 8 istruzioni contemporaneamente.



Chiaramente come mostrato nella figura seguente possono verificarsi degli stalli nella pipeline, dovuti alla mancanza di dati nella cache o alla dipendenza incrociata dei risultati, come mostrato nella figura seguente.

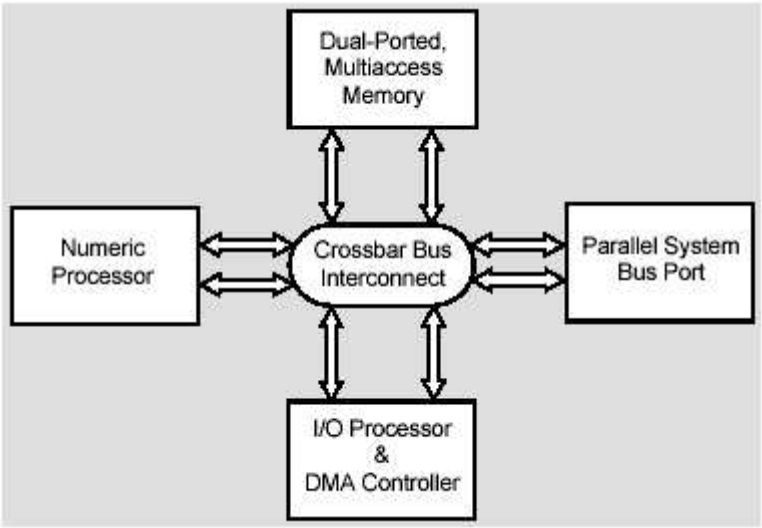


2.3. ADSP-21065L SHARC Analog Device

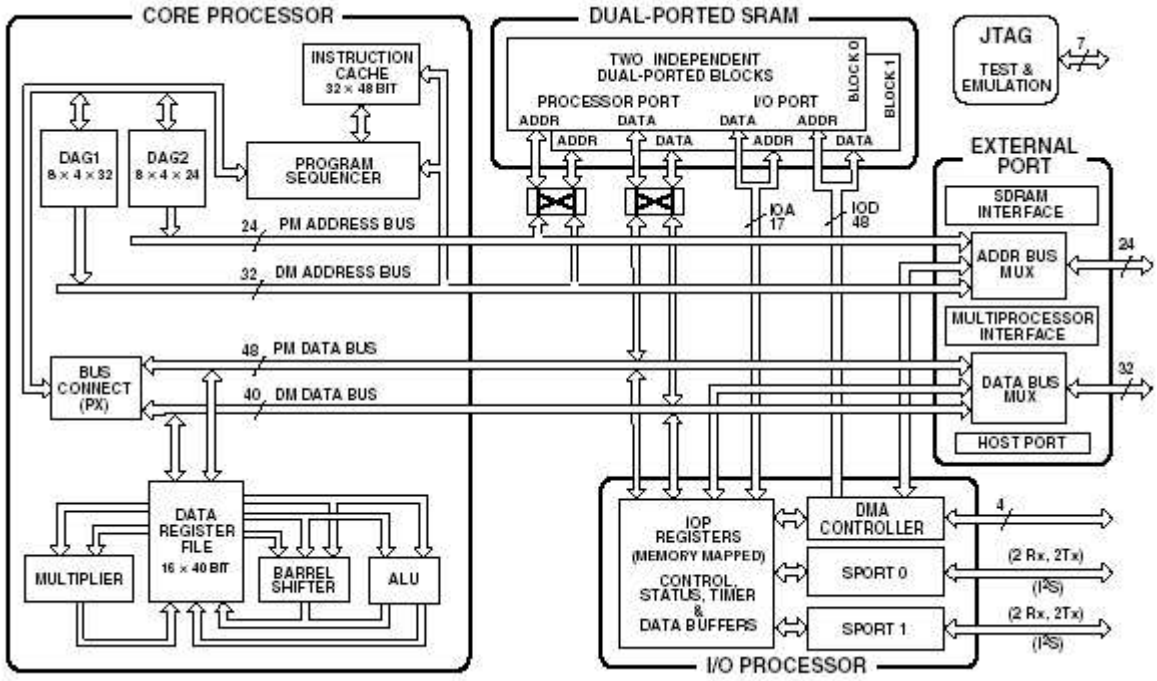
L' ADSP-21065L SHARC (L sta per low power e SHARC = Super Harvard Architecture) è un processore per segnali digitali a 32 bit, ad alte prestazioni, utilizzato per comunicazioni audio digitali, e applicazioni per strumenti industriali. Con 180 MFLOPS, l'ADSP-21065L ha due porte nella SRAM e supporta una periferica di I/O. Con la sua cache, il processore può eseguire ogni istruzione in un singolo ciclo. Il 21065L ha quattro bus indipendenti, di cui due utilizzati per i dati, uno per le istruzioni,e un altro per l'I/O. Il 21065l ha queste caratteristiche:

- 32 bit IEEE floating-point di unità di calcolo, un moltiplicatore, ALU e shifter che supporta 180 MFLOP, o 180 MOPS a 32 bit fixed point.

- registro dati
- generatore di indirizzi (DAG1, DAG2).
- Program sequencer
- 544 Kbits di SRAM
- porte esterne per interfacciare la SDRAM con altre periferiche e memorie.
- Porta per interfacciarsi con altri processori.
- Controllore DMA
- Porta seriale con due ricevitori e due trasmettitori con supporto TDM e I^2S.
- Due timer programmabili e dodici porte I/O programmabili
- Porta d'accesso per JTAG



La figura sopra mostra la Super Harvard Architettture dell' ADSP-21065L, che consiste nell'incrociare il bus del core all'I/O processor, alla memoria e alla porta parallela.



In figura viene mostrata l'architettura interna del ADSP-21065L: il PM (Program memory) bus, è composto dal PMA (Program Memory Address) e PMD (Program Memory Data) bus; il DM (Data Memory) bus è composto dal DMA (Data Memory Address) e DMD (Data Memory Data) bus;e I/O bus è composto dal IOA (I/O Address) e IOD (I/O Data) bus. Il PM bus può accedere alle istruzioni o ai dati. Durante un singolo ciclo, il processore può accedere a due operandi dati, uno sul PM bus e uno sul DM bus, accessi e istruzioni nella cache, e esegue trasferimenti in DMA. La porta esterna fornisce al processore un'interfaccia per la memoria esterna; memoria mappata di I/O, per altri processori; e altri multiprocessori 21065L. La porta esterna compie un pilotaggio tra il bus interno e quello esterno e fornisce un controllo impartito tra la memoria globale e il dispositivo di I/O.

2.ARCHITETTURA ADSP-21065L

L'architettura del 21065l è contraddistinta da:
• DSP core
• Dual-ported memory
• External port interface
• Host processor interface
• I/O Processor
• Serial ports
• DMA controller
• Booting
• Development tools

2.3.1. DSP core

DSP core consiste in:
• tre unità di calcolo
• registri dati
• Program Sequencer e due registri Data Address Generators
• Instruction Cache
• DSP core buses
• Due timer programmabili e dodici general-purpose I/Os
• Quattro hardware esterni dedicati agli interruptus
Questi profili addizionali sostengono e aumentano i componenti del DSP core:
• commutazione di contesto
• serie di istruzioni complete
IL DSP core contiene tre indipendenti unità di calcolo:
• ALU
esegue una serie di operazioni standard aritmetiche e logiche in fixed-point e floating-point
• Moltiplicatore con accumulatore fixed-point.
Esegue moltiplicazioni in floating-point e fixed-point, e somma di moltiplicazioni in fixed-point
• Shifter
Esegue degli shift logici e aritmetici, e operazioni con operandi a 32 bit.

Le unità di calcolo processano i dati in tre formati:
• 32-bit, fixed-point
• 32-bit, floating-point
• 40-bit, floating-point
REGISTER FILE

Viene usato nelle applicazioni dove si deve trasferire dati tra le unità di calcolo e i bus dati e per metterci dei risultati intermedi.Il Register File ha due serie di 16 registri, e sono tutti a 40 bit. Il Register File, insieme con il core abilita il flusso di dati fra le unità di calcolo e la memoria interna.
PROGRAM SEQUENCER E DAG
Il Program Sequencer genera gli indirizzi per l'accesso alla memoria. Insieme il PS e Data Address Generator (DAG) permettono alle operazioni di calcolo di essere eseguite con massima efficienza . Usando una istruction cache il 21065l può simultaneamente andare a prendere (nella cache) e accedere a due operandi dati (nella memoria). Il DAG implementa il buffer circolare in hardware.
Il PS fornisce l'indirizzo del comando per la memoria programma. Il DAG genera l'indirizzo di memoria quando si deve trasferire i dati tra la memoria e registri. I due DAG abilitano il processore per l'uscita simultanea di indirizzi per due operandi in lettura o scrittura. Il DAG1 fornisce gli indirizzi a 32 bit nella memoria dati, mentre il DAG2 fornisce gli indirizzi a 24 bit per la memoria programma e per l'accesso ai dati nella memoria programma. Ogni DAG tiene conto fino a 8 puntatori, 8 modificatori e 8 valori di lunghezza. Si può modificare un puntatore usando un'indirizzamento indiretto con un valore specificato in un registro prima dell'accesso (premodify) o dopo (postmodify). Per fare un'indirizzamento automatico in un buffer circolare si deve associare il valore di lunghezza con ogni puntatore; e si può stabilire il limite in memoria del buffer circolare. Il buffer circolare permette un efficiente implementazione di ritardi di linea e altre strutture dati richieste nel processamanto dei segnali digitali e comunemente usati nei filtri digitali e nelle trasformate di Fourier. Il DAG testa automaticamente che negli indirizzi dei puntatori si riduce l'overhead, incrementa le prestazioni e semplifica le implementazioni.

2.3.2. Instruction Cache

Il PS contiene comandi per la cache di 32 word che abilitano alle operazioni 3 bus per prelevare istruzioni e due valori di dati. La cache è selettiva, ogni istruzione che crea conflitti con l'accesso ai dati nella memoria programma viene nascosta. Questo permette una velocità massima di esecuzione nelle operazioni di core looped, come filtri digitali, somma di moltiplicazioni, e FFT.

2.3.3. DSP Core Buses

Il DSP core ha 4 bus:
• Program Memory Address
Trasferisce gli indirizzi per le istruzioni.
• Data Memory Address
Trasferisce gli indirizzi per i dati.
• Program Memory Data
Trasferisce istruzioni.
Dato che il PM Data bus e a 48 bit, può contenere istruzioni di questa lunghezza. I dati in Fixed point e floating point in singola precisione sono allineati fino al più alto dei 32 bit questo bus.
• Data Memory Data
Trasferisce dati.
Il DM Data bus è a 40 bit di lunghezza e stabilisce un percorso per trasferire il contenuto di ogni registro dentro il processore verso ogni altro registro o altra locazione di memoria dati in un singolo ciclo. I dati in Fixed point e floating point in singola precisione sono allineati fino al più alto dei 32 bit questo bus.

2.3.4. Timer Programmabili E Porte I/O Di Tipo General-Purpose

Il ADSP 21065L ha al suo interno due indipendenti blocchi di timer programmabili. Ogni blocco funziona in uno dei seguenti modi:
- Timer Counter mode
- Pulse count and Capture mode
In Timer Counter mode, il processore può generare una forma d'onda con un'arbitraria ampiezza entro un massimo periodo di 71,5secondi. In Pulse Count e Capture mode, il processore può

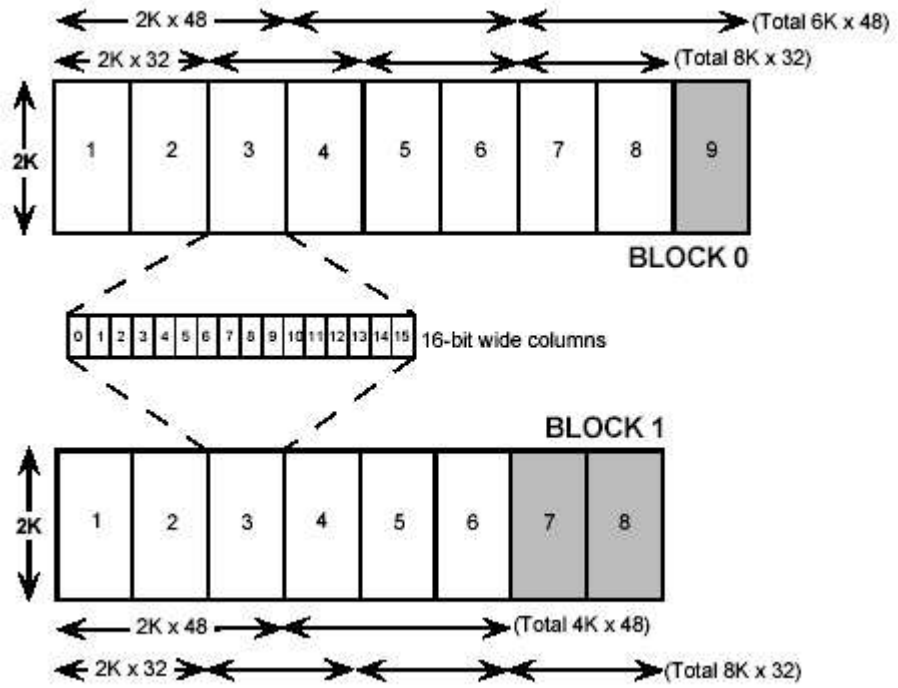
misurare l'ampiezza degli impulsi e il periodo della forma d'onda in ingresso. Il 21065L ha inoltre 12 porte di I/O che possono funzionare sia come ingresso che come uscita. Come uscita queste porte possono trasmettere a unità periferiche; come ingresso forniscono il test per un branching condizionale.

2.3.5. Interrupts

IL 21065L ha 4 meccanismi di interruzione esterna: 3 interruzioni generali IRQ_ e uno speciale per il reset. Il processore ha anche generatori interni di interruzioni per il timer, operazioni di controllo del DMA, overflows del circular buffer, stack overflows e può creare anche interrupt software.

2.3.6. Dual Ported Memory

Il 21065L contiene 544 Kbits sul chip di SRAM, organizzata dentro due banchi: banco 0 ha 288 Kbit, e il banco 1 ha 256 Kbit. Il banco 0 è disposto su 9 colonne di 2Kx16 bit, e il banco 1 è disposto su 8 colonne di 2Kx16 bit come mostrato in figura:



Ogni banco di memoria ha due accessi per ogni singolo ciclo, indipendentemente da chi ci accede che sia il processor core, l' I/O o il controller DMA. Questo permette il trasferimento di due dati da parte del core e una da parte dell'I/O, all'interno di un singolo ciclo. Sul ADSP 21065L, la memoria può essere configurata al massimo in parole di 16K di 32 bit di dati, in word di 34K per 16 bit, o in word di 10K per 48 bit, o combinazioni differenti di word fino ad arrivare ad un massimo di 544Kbits. Tutta la memoria può essere organizzata in 16 bit, 32 bit o 48 bit. Il 21065L supporta 16 bit in floating point, che effettivamente duplica l'ammontare dei dati da inserire sul chip. Converte i 32 bit in floating point in 16 bit floating point all'interno di una singola istruzione. Finchè ogni blocco di memoria conserva combinazioni di codice e dati, l'accesso è più efficiente quando un blocco immagazzina dati, usa il DM bus per trasferirli, e gli altri blocchi che immagazzinano istruzioni e dati, usano il PM bus per trasferirli. Usando il PM e DM bus in questo modo, con uno dedicato a ogni blocco di memoria, assicura con un singolo ciclo l'esecuzione di due dati da trasferire, prendendo l'istruzione disponibile nella cache. L'esecuzione in un singolo ciclo è anche mantenuta quando il dato dell'operando è trasferito fuori dal chip attraverso la porta esterna.

2.3.7. External Port Interface

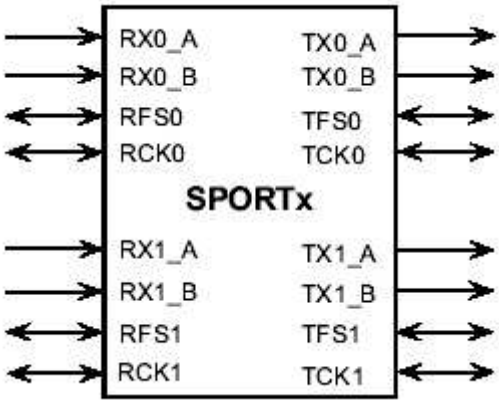
La porta esterna del 21065L fornisce al processore un'interfaccia fra la memoria e unità periferiche. I 64Mx32 bit di parola, fuori dal chip è racchiuso nel 21065L in un unico spazio di indirizzi. I bus interni al chip sono separati per PM indirizzi, PM dati, DM indirizzi, DM dati, indirizzi di I/O, e dati di I/O, l'uso molteplice verso la porta esterna per creare un sistema esterno di bus con un singolo bus a 24 bit per gli indirizzi e uno a 32 bit per i dati. Il 21065L prevede una SDRAM controller che supporta e si interfaccia con i 16Mb e 64Mb SDRAMs. Vengono decodificate le linee di indirizzi generate dai banchi di memoria selezionati per trasmesse gli indirizzi verso i dispositivi di memoria esterni.
Il 21065L ha una memoria programmabile specifica , riconosce il controllo della memoria esterna e permette al processore di interfacciarsi con l'unità periferica sostiene e disabilita l'esigenza di tempo.

2.3.8. I/O Processor

Il 21065L I/O Processor (IOP) include 2 porte seriali ognuna con due trasmettitori e due ricevitori, e un DMA controller.

2.3.9. Serial Port

Il 21065L ha 2 porte seriali sincronizzate che forniscono un'interfaccia per una vasta varietà di segnali e segnali miscelati di dispositivi periferici. La porta seriale può lavorare alla massima velocità di clock del processore fornendo ogni dato ad una velocità di 30 Mbit/s. Ogni serial port ha un primario e un secondario per canali di Tx e Rx come mostrato in figura:



Indipendentemente dalla funzione fornita per la trasmissione e per la ricezione favorisce una flessibilità per le comunicazioni seriali. I dati possono essere trasferiti automaticamente dalla serial port verso la memoria attraverso il DMA. Ogni porta seriale supporta tre modi di operare: standard mode, I^2S mode (interfaccia usata comunemente nell'audio), e TDM (Time Division Multiplex) modalità multicanale.
La serial port lavora con il formato di trasmissione little-endian o big-endian, con la possibilità di selezionare la lunghezza di parola di 3 o 32 bit. Il clock della serial port e i pacchetti dati possono essere sincronizzati e generati internamente o esternamente. La serial port inoltre include una parola chiave che distingue l'aumento delle comunicazioni tra vari processori.

2.3.10.DMA Controller

Il controlloer DMA (direct memory access) è un accesso diretto alla memoria,è un canale di comunicazione tra la memoria e le unità periferiche; permette il trasferimento dei dati senza l'intervento del processore. Il DMA opera indipendentemente e in modo invisibile rispetto al processore in maniera tale che quest'ultimo può eseguire altre operazioni. Può trasferire dati attraverso la porta seriale tra al memoria interna e quella esterna, periferiche esterne o a altri

processori. Se il DMA colloquia tra la memoria esterna e un'unità periferica esterna ha un'altra possibilità di scelta: durante il trasferimento, il DMA controlla automaticamente i pacchetti dati e toglie la parola al bus esterno. Sono disponibili 10 canali di DMA nel 21065L, 8 attraverso la serial port e due con la porta esterna (per altri processori o altra memoria del 21065L o trasferimenti di I/O). In modo asincrono è possibile controllare delle unità periferiche attraverso due porte esterne usando il canale del DMA per la richiesta e l'assegnazione del bus ($DMAR_{1-2}$ e $DMAG_{1-2}$). Inoltre il DMA include la generazione di interrupt subito dopo il completamento del trasferimento in DMA per la connesione automatica del trasferimento.

2.3.11.Development Tools

Il ADSP-21065L è supportato da un completo set di sviluppo software e hardware, che comprende la scheda EZ-KIT Lite e il VisualDSP++. La scheda EZ-KIT Lite può essere usata anche per il 21060/62,

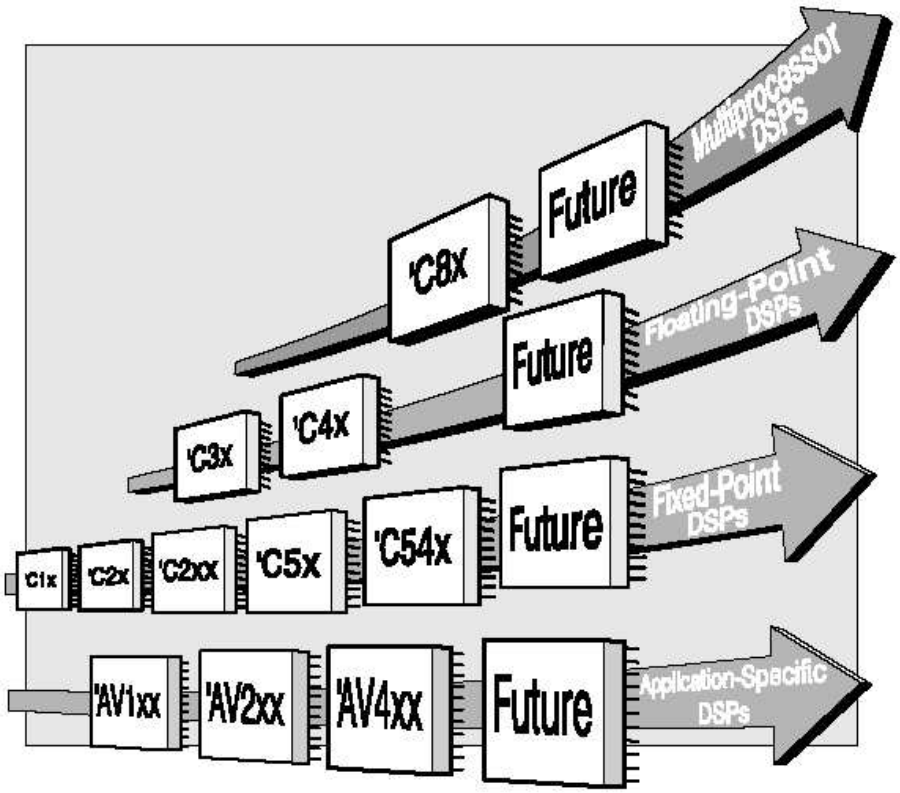
E' possibile anche emulare il 21065L, a eccezione di mostrare e modificare due nuovi registri della SPORTs. L'emulatore non mostra questi due registri, però il suo software può usarle. Tutti gli sviluppatori della SHARC e l'ambiente integrato del VisualDSP supportano il 21065L.
Il VisualDSP permette di sviluppare e rimuovere errori per le applicazioni dai singoli programmi integrati. Gli strumenti di sviluppo della SHARC includono l'utilizzo dell'assembler basato su sintassi algebrica; un clinker, un caricatore, un cycle-accurate, un compilatore C e un C run-time library che include DSP e funzioni matematiche. Entrambi i Debugging dei programmi in C e Assembler con il VisualDSP debugger si può:
- Vedere e incrociare i codici C e assembler
- Inserire dei break points
- Osservare le linee di codice
- Controllare l'esecuzione dei programmi
- Caricare e scaricare memoria
- Creare dei debugger per windows

L'ambiente di sviluppo Integrated Development Environment (IDE) del VisualDSP permette di definire e manipolare progetti multi-utente. La sua casella di dialogo consente di configurare e manipolare tutti gli strumenti di sviluppo dello SHARC. L'emulatore EZ-KIT fornisce una velocità massima per l'emulazione e permette di controllare e modificare memoria, registri e lo stack del processore.

3.3 Panoramica di sviluppo DSP di T.I. e Analog Devices

Diamo un idea di come le case produttrici di DSP presentano le performance previste dei loro modelli, tramite le cosiddette Road maps, citando come esempio dapprima la Texas Instruments (TI) e successivamente l'Analog Devices.

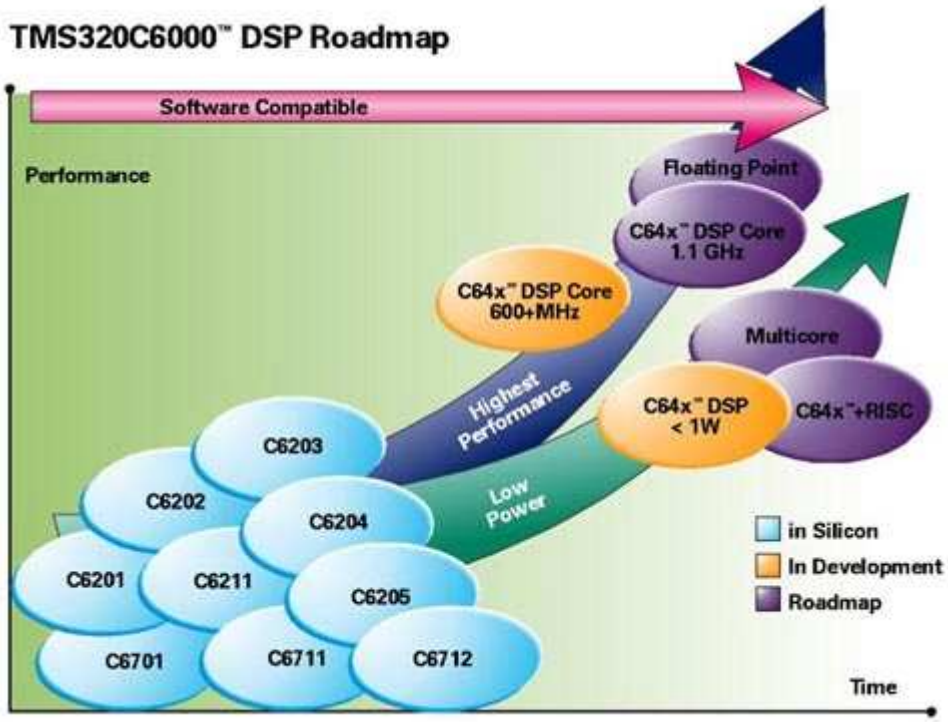
Di seguito viene riportato il road map di tutte le famiglie di DSP della Texas Instruments, in base ad ogni specifica di utilizzo: DSP fixed e floating point, e DSP per applicazioni specifiche.



Un ulteriore dettaglio viene fornito nelle tabelle che seguono per la famiglia C6000™

.

La famiglia C6000 della Texas Instruments usufruisce di un software visuale (Code Composer Studio) per i DSP fixed-point e floating-point, Ad es. il nuovo DSP fixed-point C64x caratterizzato da velocità di clock fino a 1,1 GHz è adatto ad applicazioni wireless broadband networks e digital imaging . Il DSP floating-point C67x fornisce un range di alte prestazioni e permette applicazioni avanzate di home theatre, industrial automation, voice e speech recognition e la televisione digitale.



.

TMS320C6000™ DSP Platform Offers the Broadest Price/Performance Spectrum

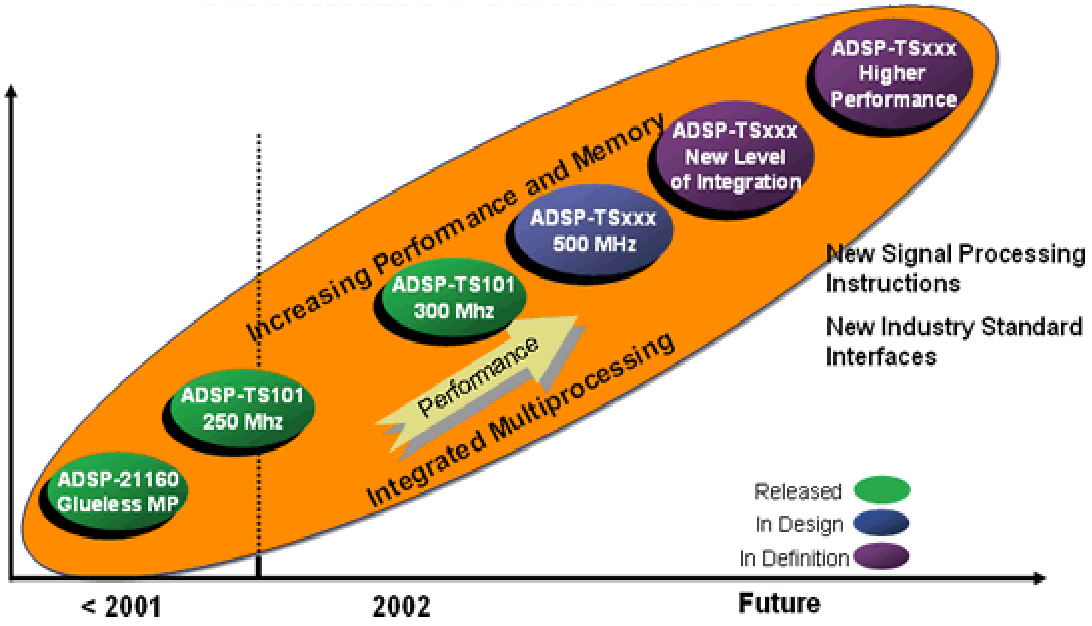
SOFTWARE COMPATIBLE

	C6201	C6202	C6203	C6204	C6205	C6211	C6701	C6711	C6712
Performance	1600 MIPS	2000 MIPS	2400 MIPS	1600 MIPS	1600 MIPS	1200 MIPS	1 GFLOPS	900 MFLOPS	600 MFLOPS
Memory (Bytes)	128K	384K	896K	128K	128K	72K L1/L2	128K	72K L1/L2	72K L1/L2
Key Feature	McBSP	Expansion Bus	Pin Comp with C6202	Expansion Bus	PCI On-Chip	HPI	Flt-Pt Perf.	HPI	Lowest Cost
BGA Package	35 mm/352 27 mm/352	27 mm/352 18 mm/384	27 mm/352 18 mm/384	18 mm/340 16 mm/288	16 mm/288	27 mm/256	35 mm/352	27 mm/256	27 mm/256
Internal Power (Typical)	1.3 W @ 200 MHz	2.1 W @ 250 MHz	1.3 W @ 300 MHz	0.8 W @ 200 MHz	0.8 W @ 200 MHz	0.9 W @ 150 MHz	1.4 W @ 167 MHz	1.1 W @ 150 MHz	0.7 W @ 100 MHz
100 Ku Price (Estimated)	$65.50	$96.95	$151.25	$29.25	$35.25	$19.25	$115.95	$28.50	$9.95
Samples	✓	✓	✓	✓	✓	✓	✓	✓	✓
Production	✓	✓	Nov 00	Nov 00	Nov 00	✓	✓	✓	1Q01

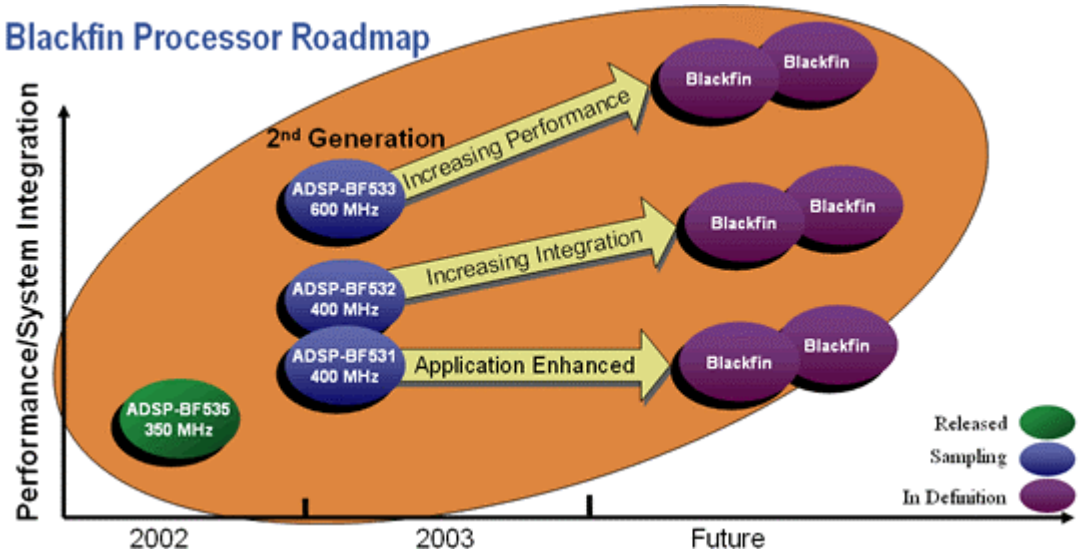
THE WORLD LEADER IN DSP AND ANALOG — TEXAS INSTRUMENTS

Roadmap dei prodotti di Analog Devices.

Assieme alla famiglia ADSP21xxx (citiamo l'ancora valido 21065L) l'Analog Devices propone l'architettura parallela TigerSHARC per applicazioni che riguardano il trattamento di segnali passabanda in banda base, applicazioni audio multicanale ecc.



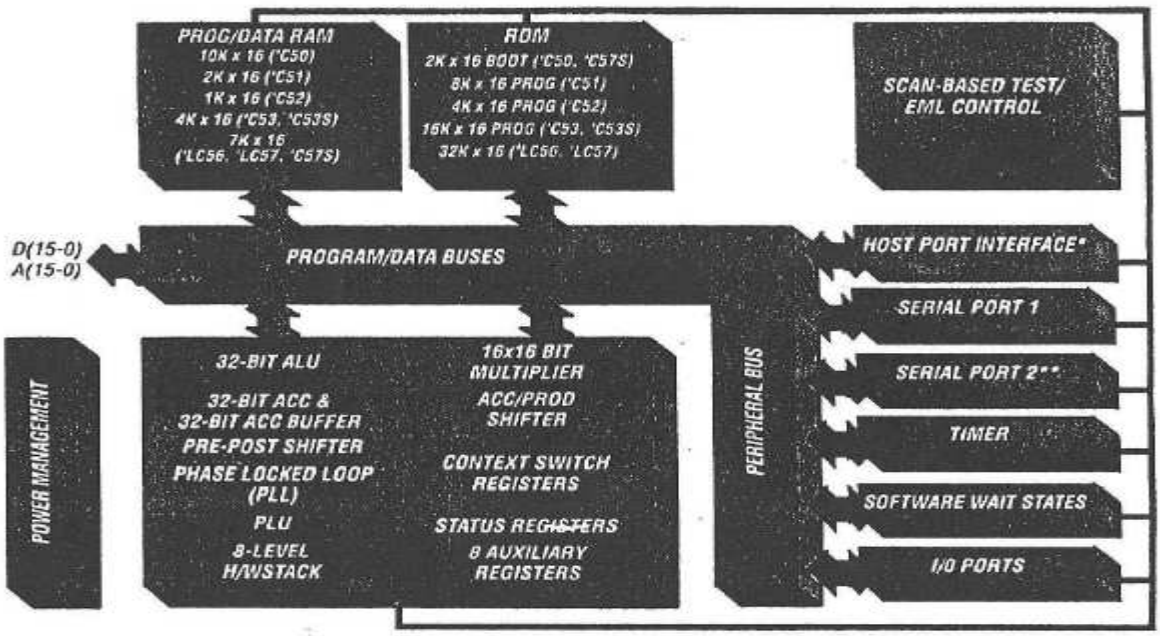
L'ADSP-BF535 è il primo membro della famiglia dei processori Blackfin seguito da una serie di altri modelli quali l'ADSP-BF531, ADSP-BF532 e ADSP-BF533, tutti contraddistinti da bassi consumi e alte performance (con tensioni di esercizio anche di 0,7 volts!).



3. Cenni su due modelli di DSP fixed point

3. 1 TMS320C50 Fixed Point

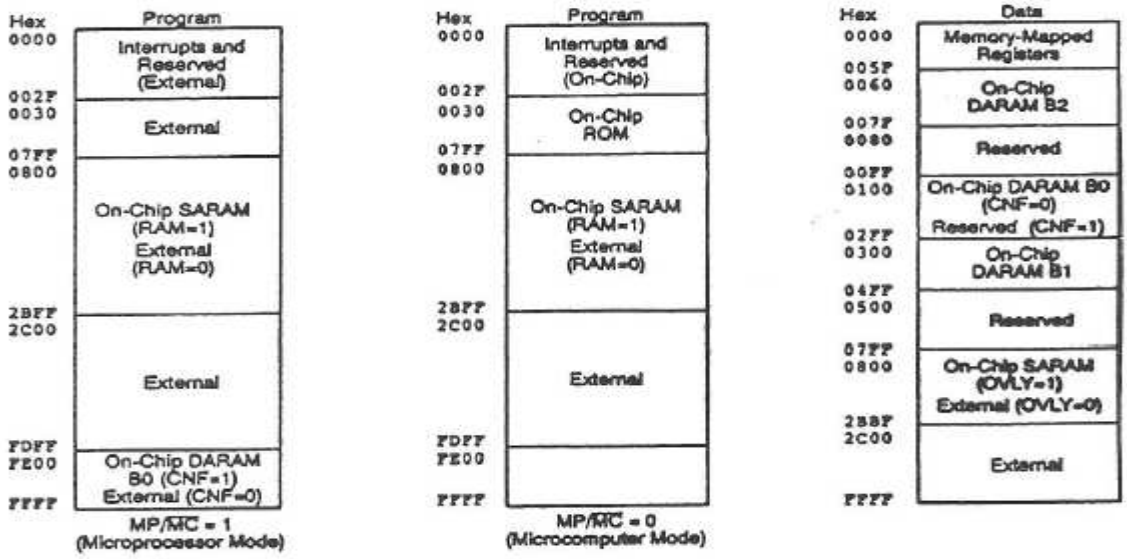
Di seguito viene riportato lo schema a blocchi con le principali caratteristiche del DSP della Texas Instruments TMS320C50, un DSP in virgola fissa tra i più economici presenti sul mercato.



Schema a blocchi TMS320C50

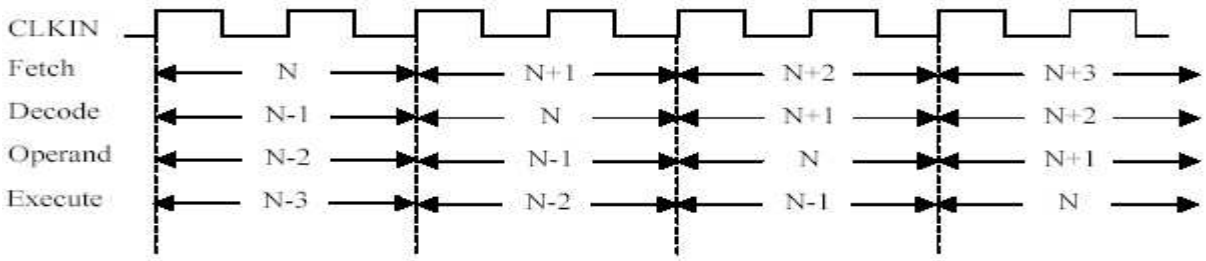
Il dispositivo integra memorie (RAM e ROM) e periferiche (interfacce parallele e seriali sincrone, timer, un'interfaccia JTAG e un'interfaccia strutturata a bus), oltre al moltiplicatore e alla ALU. L'architettura è Harvard, come si nota dalla doppia mappa di memoria (programma e dati) riportata di seguito. Le RAM sono di tipo SARAM (single access RAM) e DARAM (dual access RAM): i 9kbytes di SARAM sono indirizzabili sia come memoria programma che come memoria dati, ma non possono contemporaneamente essere utilizzati per entrambe le funzioni, dato che il dispositivo fisico (i 9kbytes di RAM) è unico. Il dispositivo ammette due modi di funzionamento: il modo microprocessore ipotizza di avere codice in memoria esterna invece che in ROM interna.

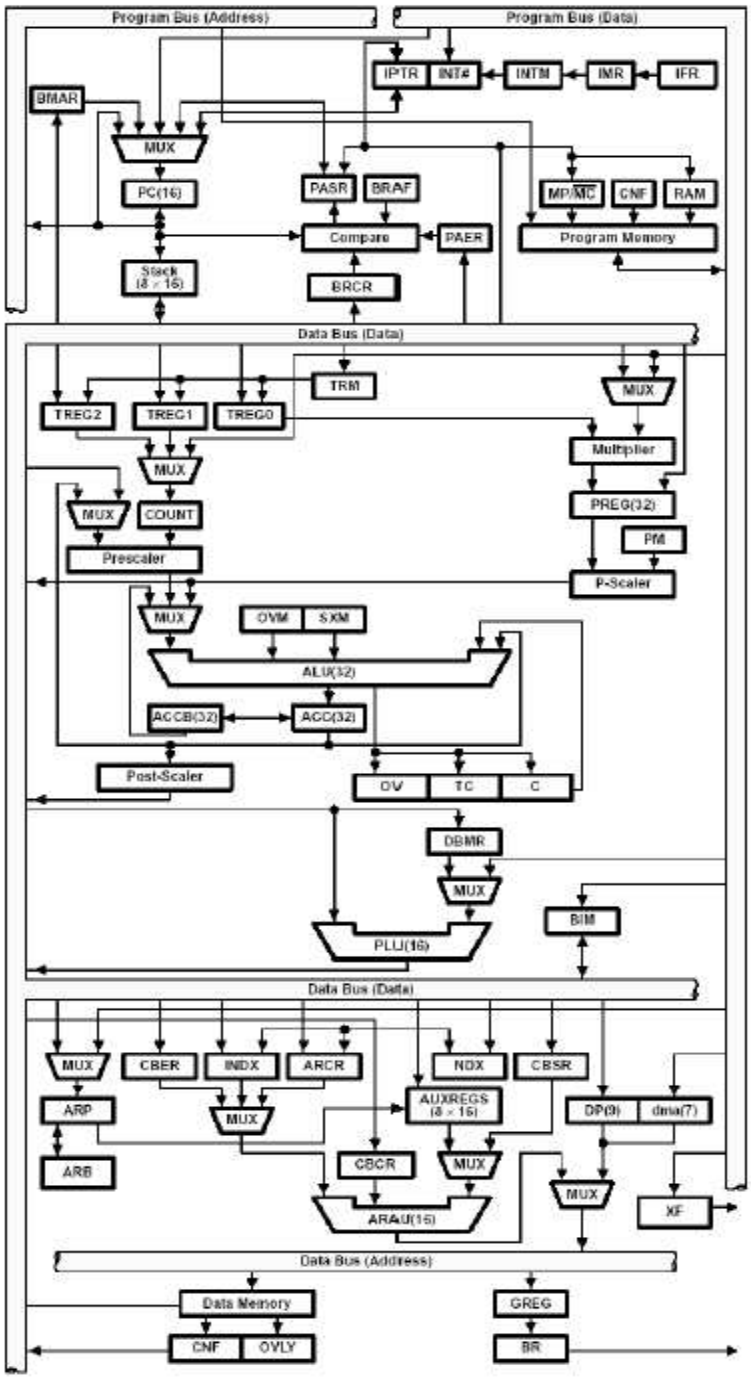
I vettori di interrupt, compreso il vettore di reset, sono organizzati in una tabella ad indirizzo spiazzabile mediante programmazione (reg. IPTR). Gli interrupt sono mascherabili singolarmente e la logica di utilizzo dei relativi flag è quella classica indicata per i microprocessori. Le periferiche vengono gestite grazie a registri dedicati e organizzati in una tabella, proprio come avviene nei μC. Di seguito viene mostrata l'architettura d'insieme: si noti la netta separazione tra la memoria dati e la memoria programma e tra i relativi bus. La presenza di 4 bus (programma-indirizzi, programma-dati, operandi-indirizzi, operandi-dati) consente una struttura pipeline a 4 livelli: fetch (bus programma-indirizzi), decodifica o decode (bus programma dati), selezione degli operandi o operand (bus operandi-indirizzi) e esecuzione o execute (bus operandi-dati).



Mappa di memoria

Il BIM (Bus Interface Module) consente lo scambio di dati tra i bus programma e operandi. Si nota anche come a fronte di un clock a 40MHz, le operazioni vengono eseguite a 20MHz.





Il program counter è a 16 bit, dato che lo spazio di indirizzamento è 2^{16}. Per migliorare le prestazioni, lo stack non è mappato in memoria, ma viene realizzato con un'architettura LIFO ad elevato parallelismo: in questo modo in un unico ciclo vengono memorizzati nello stack i valori di tutti i registri di interesse (PC, accumulatori,..). Mentre per gli altri registri lo stack ha profondità 1 (non si ammettono interrupt annidati), il PC ha una stack con profondità 8 e pertanto consente l'implementazione di routine annidate. Si noti la presenza di strutture per l'esecuzione di istruzioni a blocchi tipo block repeat (BRAF-flag-, PASR-start address-, PAER-end address-,BRCR-contatore-). Sono presenti 4 strutture di calcolo: la ALU con i suoi shifter, il moltiplicatore, un'unità dedicata alle operazioni logiche di tipo read-modify-write (Parallel Logic Unit –PLU-) e un'unità dedicata al calcolo dell'indirizzo dell'operando (Auxiliary Register Arthmetic Unit –ARAU-). Si noti inoltre la presenza di una struttura a bus request (BR) e un registro di appoggio di memoria

globale (GREG) per lo scambio dati tra più processori. Le modalità di indirizzamento comprendono l'indirizzamento immediato, l'indirizzamento inerente verso registri dedicati (Es. TREG per le moltiplicazioni), l'indirizzamento diretto organizzato a pagine di 127 word (trattandosi di un processore a 16 bit ogni indirizzo individua una word) e l'indirizzamento indiretto. L'indirizzamento diretto avviene specificando solo i sette bit meno significativi dell'indirizzo, che vengono caricati nell'Instruction Register (IR), mentre i rimanenti 9 bit sono specificati dal registro DP. Il formato delle istruzioni ad indirizzamento diretto è particolarmente compatto (16bit): gli 8 bit più significativi contengono l'op-code, il bit 7 è a zero per identificare la modalità di indirizzamento diretta e i rimanenti sette bit, unitamente al contenuto del registro DP, specificano l'indirizzo (data memory address). L'indirizzamento indiretto si basa sull'uso di 8 registri ausiliari (Auxiliary register file); di questi 8 registri viene utilizzato quello specificato dal registro a 3 bit ARP (Auxiliary Register Pointer). Come si è detto l'ARAU consente di effettuare operazioni aritmetiche sui registri ausiliari.

3.2 ADSP 2189M Analog Devices

Questo processore può funzionare con clock a 75 Mhz ed è ottimizzato per applicazioni numeriche ad alta velocità e basso consumo (alimentazione a 2,5 volt). Esso mantiene l'architettura base della famiglia 2100, con le tre unità di calcolo (ALU, MAC e Shifter), un generatore di indirizzi per i dati e un program sequencer . Le unità computazionali oltre che dai soliti due BUS differenti per le aree di memoria Programmi e Dati sono interconnesse da un ulteriore BUS dedicato (R-BUS) consentendo in tal modo l'esecuzione delle istruzioni di calcolo contestualmente al prelievo dei nuovi dati; in tal modo è possibile completare nello stesso ciclo di clock fino a quattro istruzioni di una architettura seriale SISD (Single Instruction Single Data). Il 2189M integra 192 Kb di memoria configurata in 32K words (a 24 bit) per i programmi e 48K words (a 16 bit) per i dati. Il chip dispone di due porte seriali ad alta velocità (delle quali una impegnata dal CODEC e l'altra è utilizzata per le comunicazioni con l'esterno), una porta DMA interna a 16 bit, varie linee di interrupt e flag di I/O programmabili. È presente inoltre un timer interno capace di generare interruzioni di programma ad intervalli prefissati.

Per quanto riguarda il kit ADSP 2189M EZ-LITE è fornito dall'Analog Devices come tools di

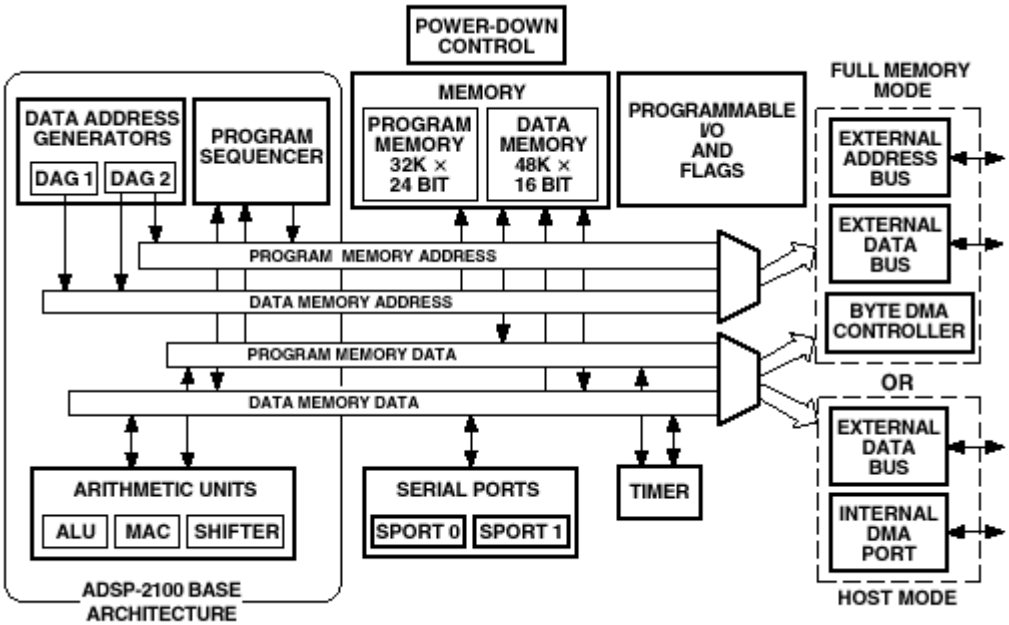


Fig. – Architettura del 2189M

sviluppo specifico per il trattamento del segnale vocale; e in effetti la sezione di conversione AD/DA lavora fino a 4-5 Khz, impiegando un generico codec (AD 73322), a differenza dei kit di altri DSP della stessa casa, che hanno codecs con prestazioni hi-fi nell'intera gamma audio.

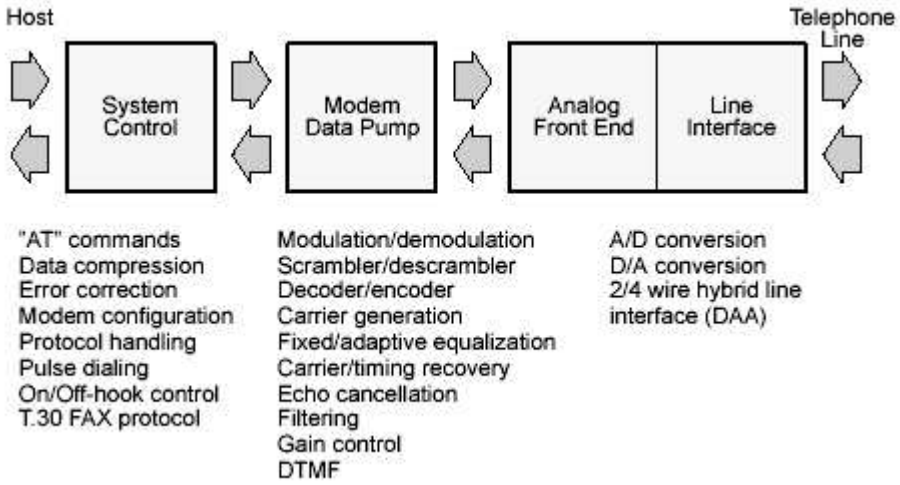
Questo DSP con il sui starter kits è stato impiegato in un progetto coordinato CNR per la realizzazione di un apparecchio a basso consumo per la misura del rumore acustico.

4. Applicazioni tipiche dei DSP.

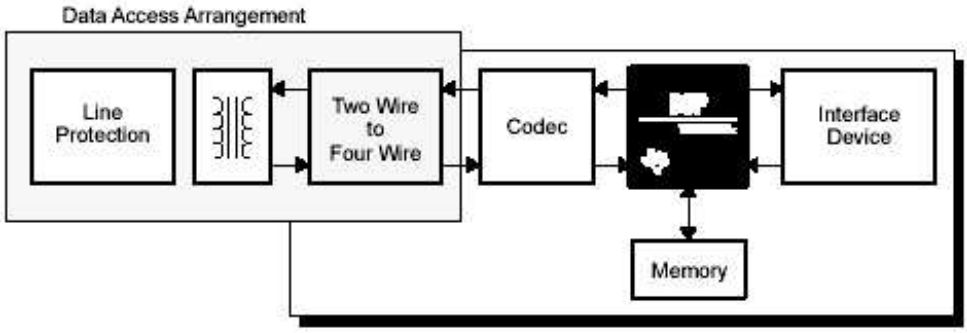
Riportiamo e descriviamo brevemente alcuni schemi a blocchi che mostrano esempi di impiego dei DSP nel ***settore telecomunicazioni*** riportati più ampiamente nella documentazione fornita dalle maggiori case produttrici.
Le telecomunicazioni sono una tra le più fiorenti aree di applicazioni che fanno uso delle più aggiornati dispositivi dell'elettronica in termini di velocità di esecuzione delle operazioni e bassi consumi. Il processamento dei dati in real-time richiede tipicamente l'impiego dei processori DSP.

4.1. Modem

Negli ultimi anni come è noto le trasmissioni modem sono passate dai 1,2 Kb/s, fino agli attuali 56Kb/s. Tutto questo è stato possibile grazie all'aumento delle potenzialità di calcolo dei DSP. Una tipica applicazione dei DSP nei modem è mostrata in figura.



Originariamente, l'architettura dei modem includeva un microcontrollore che gestiva le interfacce e il generico sistema di controllo. I costruttori di DSP quali Analog Device, Texas Instruments ecc. forniscono chipset che comprendono core processor, codec, e memorie, i quali possono implementare soluzioni modem, come in figura, nella quale sono elencati anche i nomi dei componenti del chipset:



TI Digital Signal Processors
- TMS320C54x
- TMS320C5x
- TMS320C32
- TMS320C2xx

Coder/Decoders
- Analog interface circuits
  - –TLC320AC01, –TLC320AC02
  - –TLC320AD55, –TLC320AD56

Data Access Arrangement
- Low distortion, low power, supply of amps
  - –TLE2662, TL08X, –TLE2064,
  - –TLC2272, TLE2144

Interface Device
- Internal UARTS
  - –TL16C550B/C, –TL16C554,
  - –TL16C750, –TL16PN550,
  - –TL16C552A
- Stand Alone: RS232 driver/receivers
  - –SN75C188/9, –MC1488/1489
- PCMCIA: UART + PCMCIA LOGIC
  - –TL16PC564A

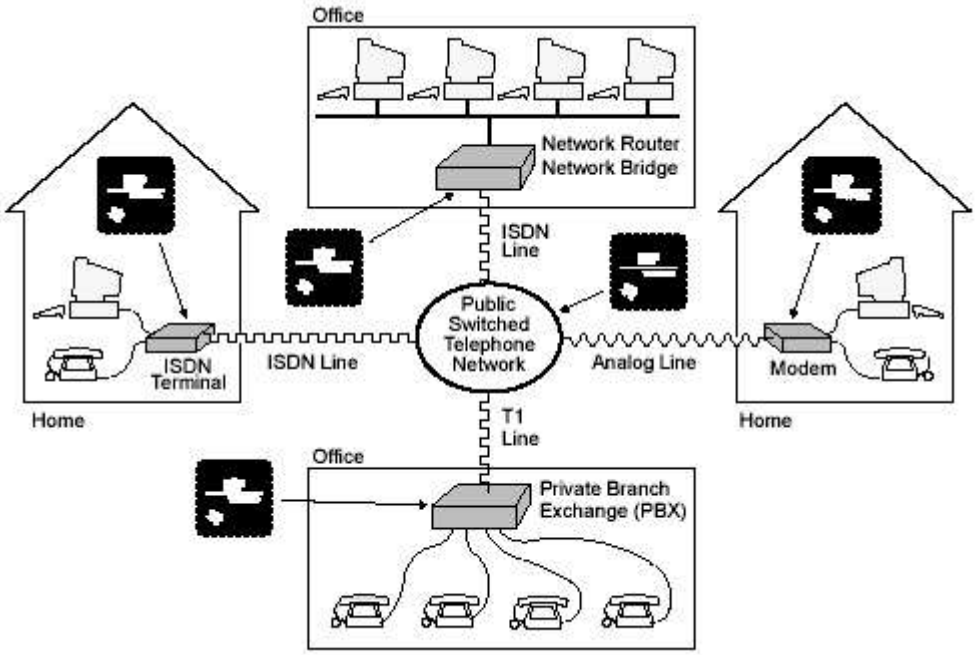
Additional Circuitry
- –Voltage supervisors (TL77XX)
- –Voltage regulators (TL750LXX, 79LXX)
- –Virtual grounds (TLE2425)
- –Voltage converters (LT1054)

4.2. ISDN

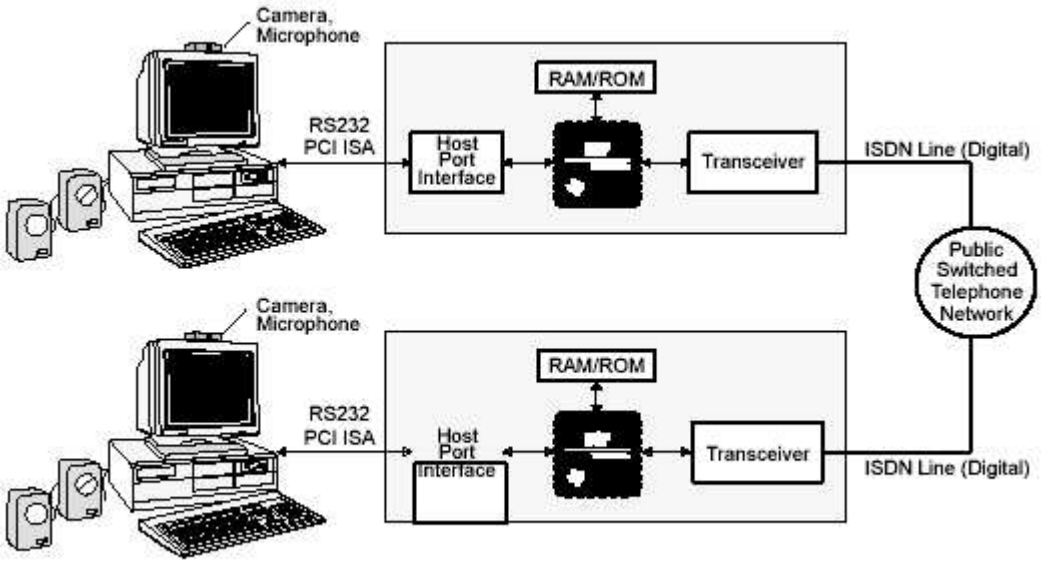
L' Integrated Service Digital Network (ISDN) offre un incremento della velocità di trasmissione dati, con una significativa riduzione dei costi. In base alla velocità raggiunta con l'ISDN è possibile la realizzazione delle seguenti applicazioni:

- Interactive publishing
- Telecommuting
- Inexpensive videoconferencing
- LAN_to-LAN connectivity
- Teleradiology
- Remote health care
- Teleteaching
- Remote broadcasting
- Collaborative CAD/CAM engineering

Con l' ISDN è possibile attraverso la normale linea telefonica (twisted-pair) avere più scambi di informazioni simultaneamente, che siano trasmissioni dati o conversazioni vocali. I DSP forniscono il processamento dei dati e assicurano la connessione tra i vari dispositivi dell'ISDN. I DSP sono incorporati in tutti i dispositivi di interfaccia dell'ISDN e del PSDN (Public Switched Telephone Network), come in figura:

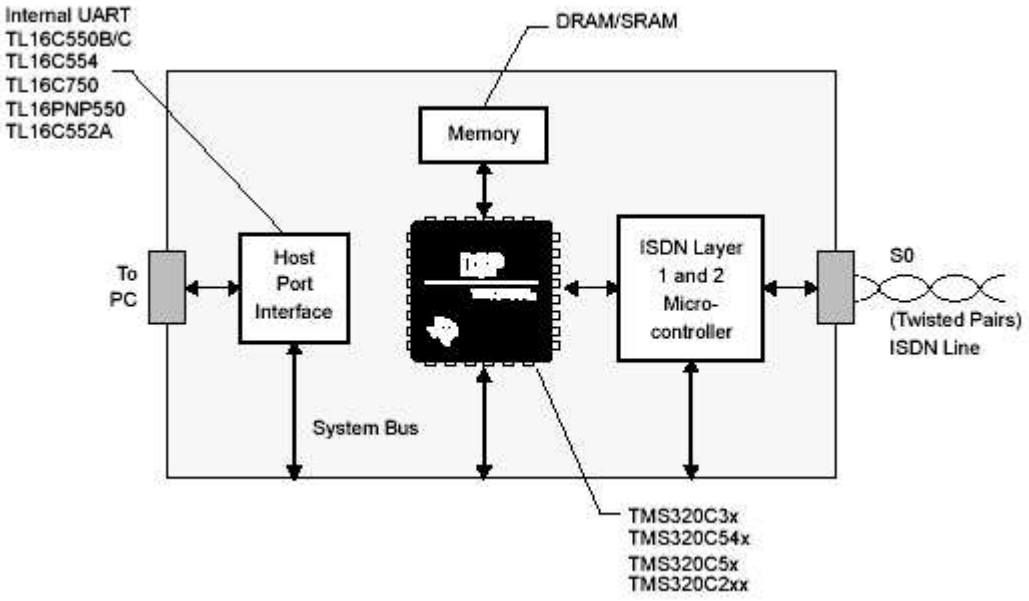


ISDN videoconferencing



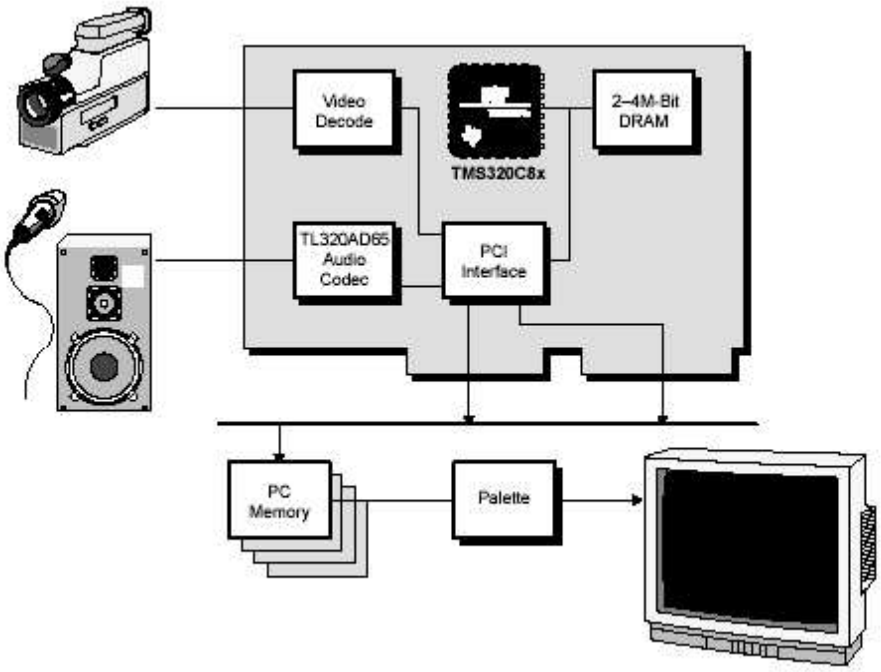
Dual-Mode ISDN Modems

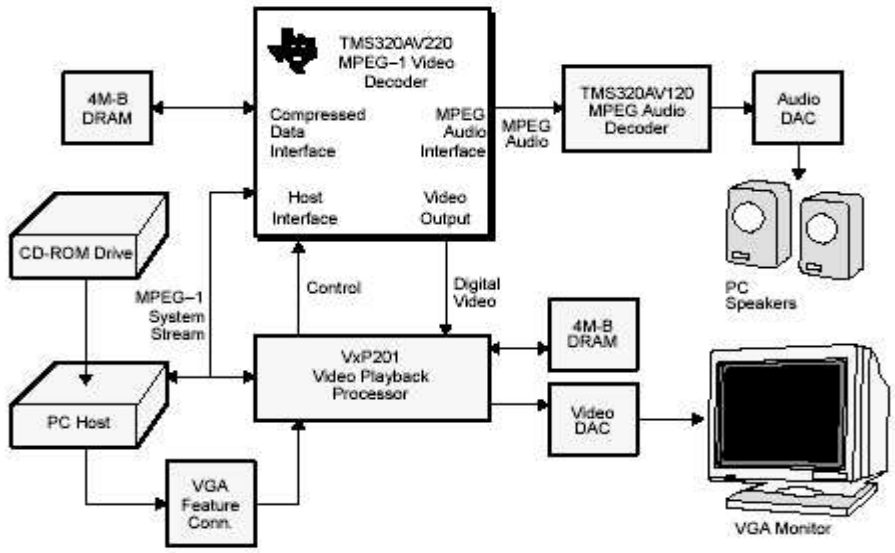
Una comunicazione dual-mode avviene quando uno dei due terminali è ISDN e l'altro è un modem analogico ed è mostrata in figura:



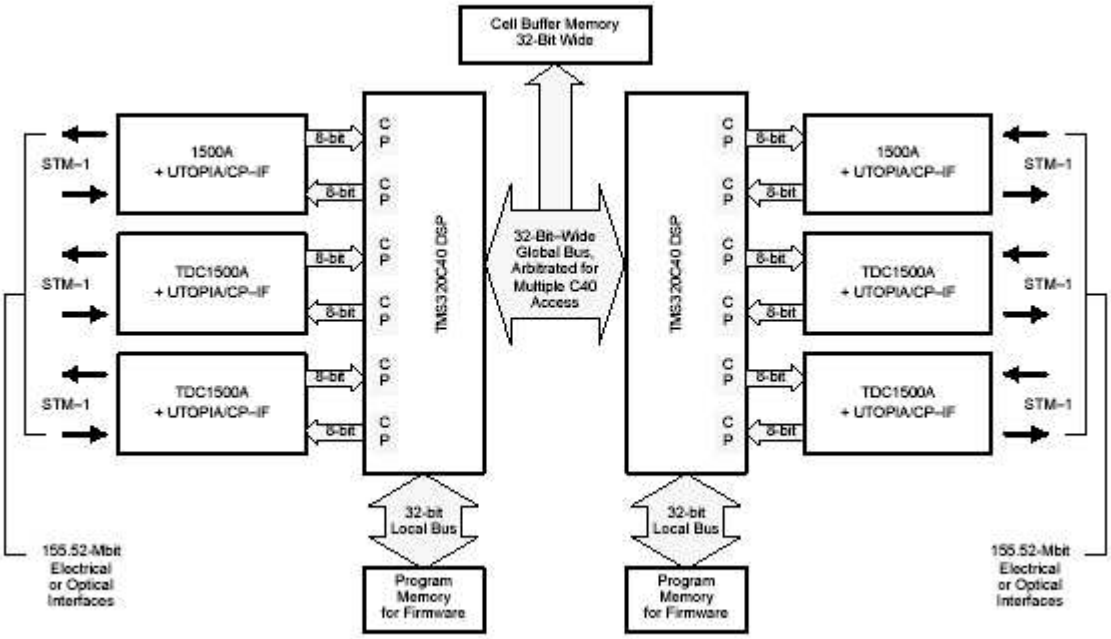
In aggiunta, nell'esecuzione di vari algoritmi come per esempio acoustic echo cancellation and ISDN supervisory code, il modem commuta automaticamente fra analog e digital mode.
Quando la comunicazione avviene con un modem analogico, il dual-mode modem commuta nel modo analogico. In questo modo, il DSP esegue una modulazione e demodulazione, tuttavia se il segnale è inviato attraverso ISDN, si deve trasmettere in formato digitale, da qui la linea ISDN è soltanto digitale.

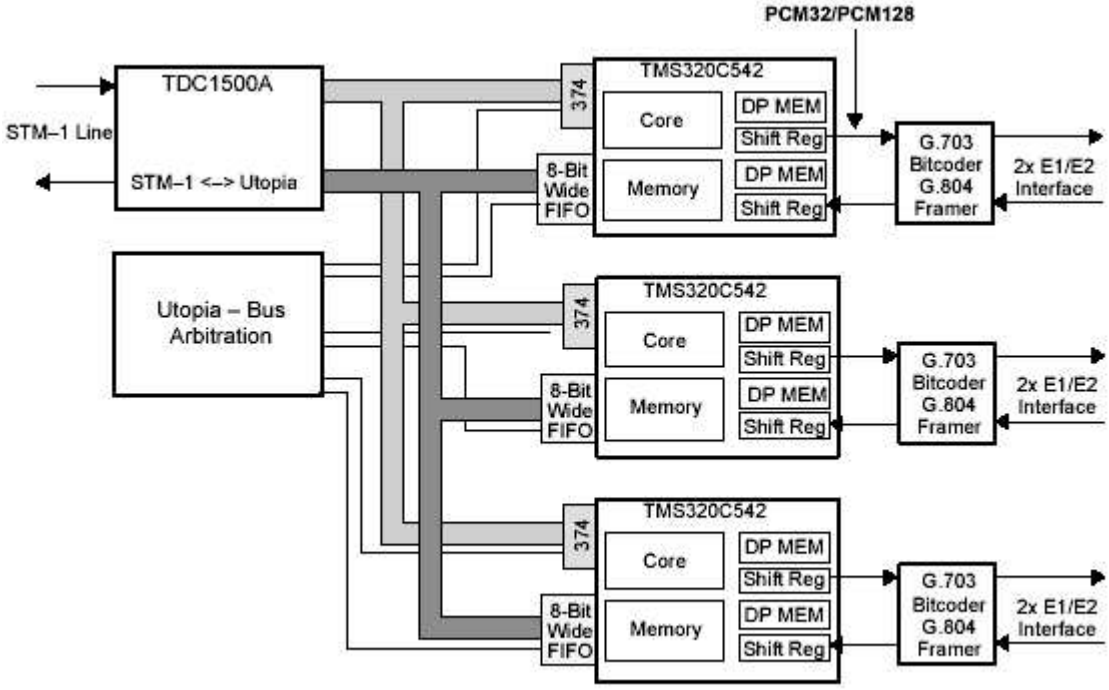
4.3. Applicazioni multimediali





4. 4. Networking Controllers



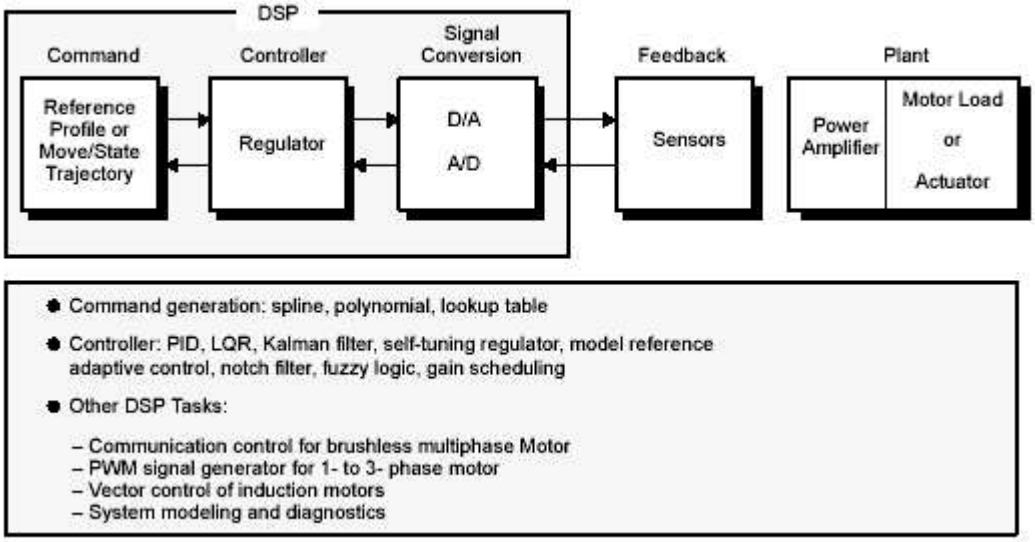


N-ISDN to ATM Switch Based on a TMS320C542

4. 5. Controllo motori

Un altro settore di impiego in cui si sono inseriti proficuamente i DSP è quello del controllo motori. A tale scopo vengono predisposti sui chip delle funzioni tipiche per queste applicazioni quali il controllo di linee PWM ecc. ed un set di algoritmi specifici per il controllo di assi e per l'acquisizione di dati da resolvers. (ad es. vedere modelli TMS320c24xx della TI). Riportiamo sotto una disposizione controllo di un impianto.

Digital motor control



5. Integrazione tra DSP e Microcontrollori

Stanno emergendo parecchie architetture che combinano su un singolo chip nuclei di microcontrollori indipendenti con nuclei di Dsp. Alcuni produttori hanno messo a punto architetture nelle quali alcune funzionalità Dsp sono integrate ai microcontrollori esistenti. Descriviamo un esempio di una soluzione proposta dalla Microchip Technology.

L'architettura **digital signal controller dsPIC** di **Microchip Technology**, mette insieme le funzionalità di controllo dei microcontrollori con le potenzialità di calcolo proprie della elaborazione numerica matematiche Il digital signal controller dsPIC è stato sviluppato con estrema attenzione ai costi e all'ambiente di sviluppo, che si presenta familiare ai tradizionali utilizzatori di microcontrollori. I digital signal controller forniscono una soluzione su chip singolo che non necessita di componenti addizionali, tipicamente richiesti finora. Dotati di funzionalità Dsp all'interno di un'unica architettura possono risolvere bene compiti legati alla gestione dei motori ad induzione nei sistemi di riscaldamento; ventilazione e condizionamento dell'aria; controllo motore ad alta efficienza energetica per elettrodomestici; demotica. Si prevede che il suo uso potrà diffondersi a breve termine anche in applicazioni attualmente dominio dei DSP, quali la gestione della multifunzione telefonica (identificativo del chiamante, cancellazione del disturbo, Dtmf); le segreterie digitali (compressione della voce); i modem a bassa velocità; i cancellatori d'eco; le macchine per la distribuzione automatica (con monitoraggio remoto tramite internet); le applicazioni di sicurezza biometrica (ad es. per il controllo delle impronte digitali); gli alimentatori non interrompibili (Ups) di fascia alta e la gestione degli alimentatori.

6. Conclusioni

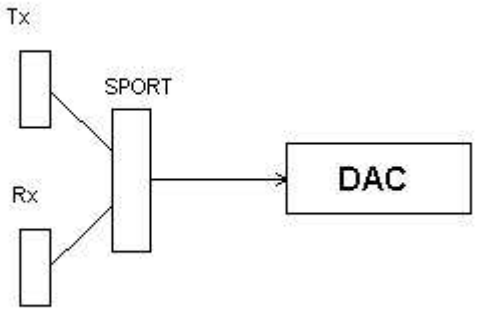
Sicuramente in futuro sono previsti ulteriori e maggiori sviluppi nelle prestazioni dei DSP. Sono infatti già disponibili unita' funzionanti ad 1 GHz. Con tali performance saranno potenziati ad esempio i transceiver nelle stazioni radio base di sistemi di telecomunicazione wireless (per le nuove tecniche di accesso WCDMA) i vari nuclei di calcolo per apparati Ip (GigaRouters, Mpls-Switches, DSLAMs etc), gli apparati biomedicali, sistemi per il radio-rilevamento Sar (e applicazioni tipiche in campo Radar), sistemi di radiolocalizzazione GPS, applicazioni multimedia ecc.
In base a quanto è successo negli ultimi anni e' possibile immaginare che lo sviluppo delle prestazioni dei singoli chip di DSP, oltre alle molte applicazioni di cui abbiamo parlato, consentirà l'implementazione di altri algoritmi complessi che attualmente trovano soluzione solo nell'uso di schede multi-DSP.

APPENDICE A

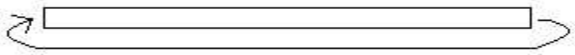
Riportiamo la struttura del sw per la gestione di dati su scheda ADSP21065L EZLITE della Analog Device di un modem a ultrasuoni per un progetto c/t dell'ISTI con una ditta privata, ma valido in generale.

ESEMPIO DI TRASFERIMENTO DEI DATI SU ADSP21065

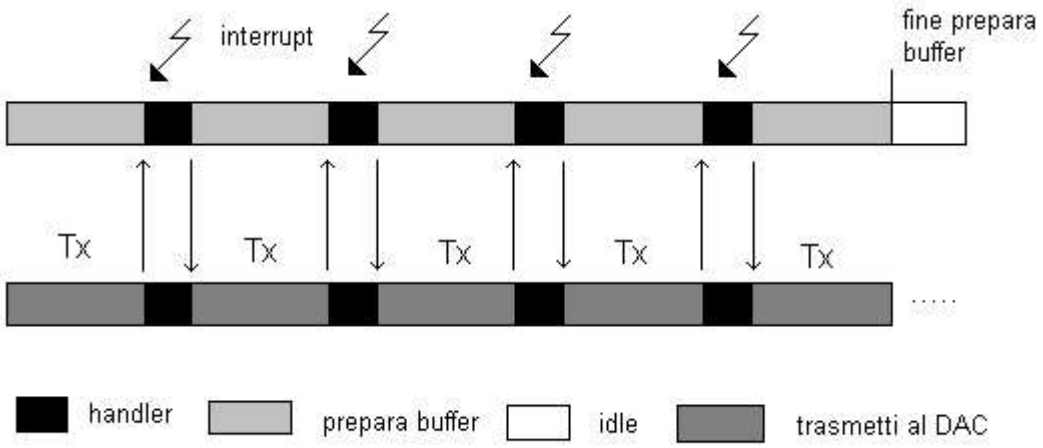
Il segnale una volta elaborato (con opportuni algoritmi di calcolo), deve essere inviato al CODEC per la conversione digitale/ analogica. Il DAC del CODEC preleva i dati tramite la SPORT,con una cadenza di 48Kbit/s, mentre i dati che arrivano dal processore hanno una velocità di 30Mbit/s.



Avendo considerato che il processore è più veloce della cadenza con cui vengono inviati i campioni al DAC tramite la SPORT, abbiamo deciso di utilizzare quella che viene chiamata " tecnica del doppio buffer". Viene dichiarato un normale buffer monodimensionale il quale viene diviso in due sottobuffer indipendenti, dove in uno si prendono i dati da inviare al DAC, mentre nell'altro si inseriscono i nuovi dati. Ogni sottobuffer viene reso "buffer circolare" ovvero si definisce un puntatore che è in grado di circolare al suo interno.



Si deve quindi creare un processo comune nel quale, si deve scandire un sottobuffer e in contemporanea riempire l'altro. Per far questo si deve abilitare il riconoscimento delle interruzioni (interrupt) del DAC, che attraverso le quali può avvertire il processore che ha quasi finito di leggere i dati da un buffer, (il processore nel frattempo deve aver già riempito l'altro buffer, e deve essere in attesa) la struttura circolare utilizzata rende possibile l'allacciamento da un buffer all'altro senza perdere sincronia, così facendo al DAC arriva un flusso continuo di dati a 48 Kbit/s. E' evidente che quando i dati vengono presi dall'altro buffer il primo deve essere nuovamente riempito.



Il nostro corpo del programma avrà la seguente forma :

```
for (; ;)
  {  if ( flag )  prepara buffer ( )
     idle ( )
  }
```

```
                                    handler
                                          {  if  ultimo  Tx
                                             (  flag  =  ! flag )

                                             <   TX   >
                                          }
```

Quando il programma entra nel ciclo infinito del for (; ;), viene riempito il buffer, durante questa operazione può essere interrotto in qualsiasi momento da un'interrupt, passando così all'handler, il quale verifica che tutti i dati siano stati trasferiti al DAC. Una volta riempito il buffer il ciclo for resta in idle (attesa) che si svuoti l'altro buffer ; per poi riempirlo nuovamente.

Bibliografia

E. A. Lee " Programmable DSP Architecture: Part 1 & 2" , IEEE ASSP Magazine, oct. 1988, jan. 1989.

TMS320 Fixed-point DSP Assembly Language Tools. User's Guide Texas Instruments, 1990.

Bertini, G. Fabbri, D. "MultiC25, un sistema multiDSP con schede LeonardC25", IEI-CNR Prog Finaliz. Calcolo Parallelo, Rapp. Interno R/2/108 Aprile 1993.

Bertini, G. Giua, P.E. Marras, G. Rocchi, S. Vignoli, V. "Specifiche di progetto del dosimetro per la misura accurata della dose personale di rumore" Nota interna IEI-CNR, B4 –29, dicembre 2000. Progetto Speciale CNR "Diamond".

Bertini, G. Magrini, M. "Apparato Codificatore/Multiplatore/Modulatore digitale per radiodiffusione FM (progetto DCMM)" Nota interna IEI-CNR, B4 – 36 dicembre 1999.

Analog Devices "VisualDSP User's Guide and Reference" , "C Compiler Guide and Refernce", "ADSP 21065L Microcomputer Data Sheet and ADSP21065L EZLITE" Norwood 1999.

Kumura T. et al. « VLIW DSP for Mobile Applications » Proc. ICASSP 2001,. ICASSP Magazine July 2002, pp. 10-21.

Kolagotla K. et al. « High perfomance Dual-MAC DSP Architecture », Proc. ICASSP 2001,. ICASSP Magazine July 2002, pp. 42-53.

Texas Instruments "TMS320C6000 One-Day Workshop with Starter Kit 'C6711 DSK", "Code Composer Studio – Getting Started Guide" Firenze, Aprile 2003. EZLITE

Vari siti on line delle varie case di DSP.