CAPITOLO 1:
SISTEMI EMBEDDED

Una definizione generale di sistema embedded può essere quella che lo definisce come un "sistema di controllo basato su microprocessori che elabora un insieme fissato di istruzioni programmate con lo scopo di controllare le operazioni di apparecchiature elettromeccaniche e automatizzate". Scendendo più nel dettaglio tale controllo consiste nel fornire un interfaccia utente, gestire i dati di input e/o output e il comportamento dell'apparecchiatura cui tali sistemi sono associati [1-3].

Tra le mansioni in cui sono maggiormente usati ci sono compiti di controllo di processi time-critical, perciò la loro presenza si nota soprattutto nelle industrie che si occupano di controllo di processo, ma si possono trovare sistemi embedded anche nell'hardware per le comunicazioni, nei terminali hardware di banche e finanziarie, in attrezzature mediche, nei sistemi di gestione dei trasporti e negli elettrodomestici.

La differenza più significativa tra i sistemi industriali o commerciali e i sistemi embedded (oltre a differenze nel design, in alcuni fattori di forma e nei costi) è che questi ultimi sono installati all'interno di un sistema automatizzato più grande e non possono essere visibili all'esterno.

Oltre alle differenze con i sistemi industriali bisogna mettere in evidenza quelle che li caratterizzano rispetto ai sistemi desktop. Per cominciare si usa parlare di microcontrollori nel dominio embedded e di microprocessori nell'area desktop; tale diversità non è solo terminologica perché, mentre i microcontrollori prevedono l'uso di RAM e ROM per la gestione della memoria e sono associati a numerose periferiche, i microprocessori prevedono l'uso di unità di gestione della memoria e di molte cache.

Dal momento che alcune piattaforme embedded sono nate come specializzazione di architetture destinate all'area desktop le differenze tra queste due aree tecnologiche non stanno nell'organizzazione dei registri, nell'insieme delle istruzioni di base o nel concetto di pipeline,

sono altri i fattori discriminanti: consumo di potenza, costi e integrazione di periferiche. A livello più hardware le caratteristiche peculiari dei sistemi embedded riguardano il tempo di risposta agli interrupt, la qualità di RAM e ROM su un singolo chip e il numero di porte parallele. Più in generale, mentre l'area desktop si preoccupa di valutare la potenza nell'esecuzione dei processi, il mondo embedded deve preoccuparsi di fare il lavoro per una particolare applicazione al costo più basso possibile.

Altri fattori discriminanti sono le prestazioni, la prestazioni di un sistema: devices a 8-16 bit fanno parte della classe dei microcontrollori. Ultimamente si stanno affermando architetture basate sull'uso di processori embedded a 32 bit; il loro successo crescente deve imputarsi soprattutto alla richiesta sempre maggiore di qualità e compattezza nelle dimensioni dei dispositivi hardware che caratterizza i settori dei videogame, dei computer palmari, della telefonia cellulare e della fotografia digitale: esigenze che sono la forza che sta dietro al sempre crescente interesse per il mondo dei processori embedded. Sono proprio le nuove applicazioni sviluppate in questi campi a richiedere lo sviluppo di devices nuove e/o più specializzate. Questa tendenza è talmente radicata che alcuni di questi processori stanno cominciando a incorporare capacità legate alle tradizionali CPU, introducendo parallelamente problematiche legate ai costi di questa scelta, ai vincoli (sempre maggiori) imposti dalle applicazioni e ai consumi. Da un punto di vista economico i costi di sviluppo e il time-to-market impongono che queste nuove devices abbiano un alto coefficiente di riuso da parte di altre applicazioni; così facendo si riducono, anche, i problemi di apprendimento legati allo sviluppo di un nuovo prodotto: sono queste le motivazioni principali che spiegano la tendenza di molte società a disegnare e realizzare processori embedded per un mercato molto specifico, da allargare successivamente.

Guardando al mercato che caratterizza queste tecnologie nella sua globalità si può osservare che, mentre nel passato si poteva partizionare in quattro settori di base nettamente separati, ora con l'emergere di applicazioni multimediali e per sistemi mobili (come telefoni cellulari e computer palmari) non c'è più una separazione netta, perché sono richieste nuove tipologie di processori embedded per la cui realizzazione bisogna attingere alle tecnologie messe a punto nell'ambito dei sistemi desktop.

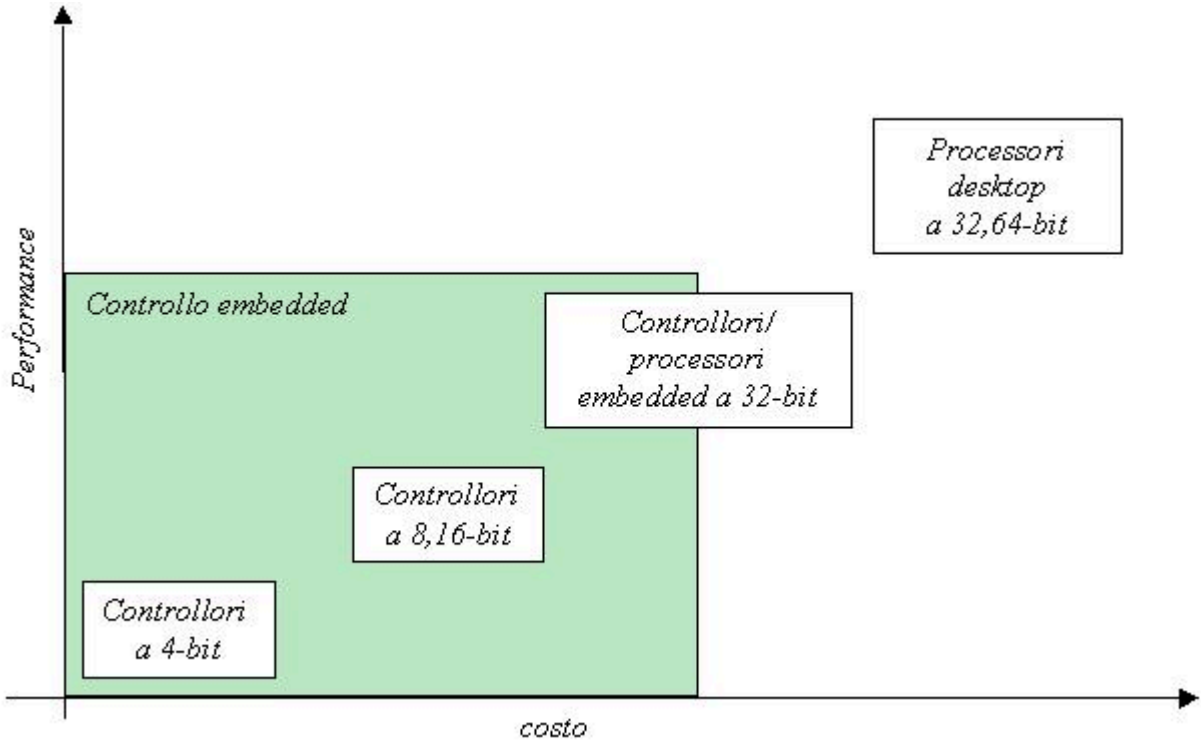

Figura 1: classi di processori e controllori embedded oggi.

Se da un lato si hanno esigenze specifiche legate al campo in cui tali sistemi saranno introdotti, requisiti che riguardano un aumento del livello di integrazione dei componenti e/o della loro prestazioni sono comuni a qualsiasi dominio applicativo.

Tra i parametri più usati quando si devono confrontare processori embedded ci sono:

- consumo di potenza;
- prestazioni;
- densità del codice;
- aspetti multimediali;
- periferiche e integrazione.

Passiamo a considerarli in dettaglio:

- consumo di potenza

  I processori embedded hanno tre modi di funzionamento:

  - in funzionamento: il segnale di clock è propagato all'intero processore e tutte le unità funzionali possono eseguire delle istruzioni;
  - in stand-by: il processore non sta eseguendo un'istruzione, ma sono disponibili tutte le informazioni memorizzate; se arriva un interrupt esterno il processore torna in funzionamento senza perdere informazioni;
  - in clock-off: il sistema non è in funzionamento e deve essere riavviato per tornare in questo stato (il riavvio è un'operazione particolarmente lunga).

  Gli sforzi di riduzione dei consumi si indirizzano ai primi due modi di funzionamento. Tale scopo viene raggiunto arrestando l'attività dei transistors quando i blocchi che gestiscono non viene usato; questo spiega perché ogni registro, flip-flop o latch, è collegato direttamente al clock-tree del processore e, di conseguenza, perché l'implementazione del clock è un fattore cruciale.

  La maggiore integrazione delle periferiche sta facendo crescere l'interesse per le problematiche che riguardano i consumi anche a livello questi dispositivi, portando alla valutazione di un parametro come il consumo di potenza non più a livello di processore (e quindi di "core" della macchina), ma al livello dell'intero sistema.

- prestazioni

  Un parametro come il consumo di potenza non deve essere considerato l'unico punto di riferimento nella valutazione di un processore embedded, deve essere messo in relazione con le sue prestazioni. L'unità di misura finora usata per il confronto delle prestazioni tra processori embedded è il rapporto MIPs/watt, tuttavia questa risulta uno strumento di valutazione insufficiente: anche le istruzioni NOP, durante le quali il processore non lavora attivamente, possono teoricamente entrare in una valutazione che usa il rapporto MIPS/watt come parametro. Per ovviare a questi problemi i programmi di valutazione sono stati sviluppati in modo da non considerare soltanto il numero di istruzioni eseguite.

Un altro fattore limitante delle prestazioni nel mondo embedded è il throughput dei dati o larghezza di banda (bandwidth). In questi sistemi sta diventando sempre più pesante processare i dati (per via di operazioni come la codifica di immagini) al punto da rendere necessario un interesse maggiore nel design delle cache e nell'architettura del bus dei dati. Queste problematiche sono già state incontrate nel processo di sviluppo dei sistemi desktop, ma in questo caso bisogna tener conto di aspetti cruciali e fondamentali come mantenere bassi i costi e potenziare le capacità real-time.

- densità del codice

  Le architetture CISC hanno una buona densità del codice per via delle loro istruzioni più complesse. La filosofia adottata nel design delle architetture RISC, invece, prevede come requisito di base delle istruzioni di lunghezza prefissata in modo da semplificare e accelerare la fase di decodifica delle istruzioni stesse. Questo significa che ad un'istruzione CISC può corrispondere una o più istruzione RISC.
  L'introduzione di queste architetture è, inoltre, legata al concetto di pipeline: le architetture che hanno scelto di adottarle forzano il processore ad avere un processo di codifica delle istruzioni più unificato; ad esempio in questa direzione deve collocarsi la scelta di adottare istruzioni di 32-bit, anche se ebbe effetti negativi sulla densità del codice tanto da indirizzare molti verso lo studio e l'introduzione di nuove strategie.
  Una soluzione in questo senso fu quella di usare istruzioni di 16-bit; in questo modo si è circoscritto il problema legato al codice dei programmi (ad eccezione delle operazioni di gestione degli indirizzi e dei valori diretti), ma è aumentata la larghezza di banda per le istruzioni caricate in memoria.
  Un approccio differente prevede, invece, l'uso di istruzioni a lunghezza variabile di 16, 32, 48-bit; in questo modo i valori diretti non sono implementati facendo riferimento alle tabelle, ma sono codificati direttamente in istruzioni di 48-bit. Così migliora il processo di decodifica nelle architetture CISC, ma resta sempre più complicato rispetto agli approcci a lunghezza fissa perché lo stato della macchina deve decidere rapidamente quale istruzione decodificare e quali dati trasferire al prossimo stadio.
  Un fattore che influenza significativamente la densità del codice è la qualità dei compilatori C: nel mondo embedded ANSI C è lo standard de facto. Tuttavia più aumentano le prestazioni dei processori embedded, più i linguaggi object-oriented giocheranno un ruolo importante.
  Future riduzioni nella densità del codice porteranno sicuri guadagni in prestazioni.

- aspetti multimediali

  Nel mondo embedded si sta assistendo ad una fusione tra le CPU convenzionali e le capacità dei DSP come è avvenuta nel campo dei sistemi desktop; in relazione a questi temi, le discussioni sono sempre più dominate dai cosiddetti "media MIPS". Per implementare

queste capacità si sta allargando l'insieme delle istruzioni dei microcontrollori convenzionali con altre che accelerino l'esecuzione del codice multimediale: da qui l'interesse all'insieme delle istruzioni che caratterizza un processore DSP. Queste istruzioni multimediali aggiuntive servono a supportare operazioni di accumulazione multipla per velocizzare il filtraggio, un modo di indirizzamento potenziato, l'accelerazione grafica e istruzioni che si occupano della trasformata discreta di Fourier per la compressione di immagini in formato MPEG o JPEG. Da un punto di vista più tecnico applicazioni embedded che devono gestire, ad esempio un telefono cellulare, necessitano un buon numero di funzionalità per maneggiare lo stack del protocollo l'interfaccia col mondo esterno, ma in questo modo aumenta il bisogno di equalizzazione, codifica/decodifica della voce e compressione.
Tradizionalmente questi aspetti erano gestiti con un microcontrollore a basse prestazioni e un processore DSP dedicato, come mostrato in figura 2.
Gestire tutto con un solo microprocessore semplifica drasticamente la struttura di tutto il sistema in quanto si ha soltanto un flusso di istruzioni, come si può notare in figura 3.

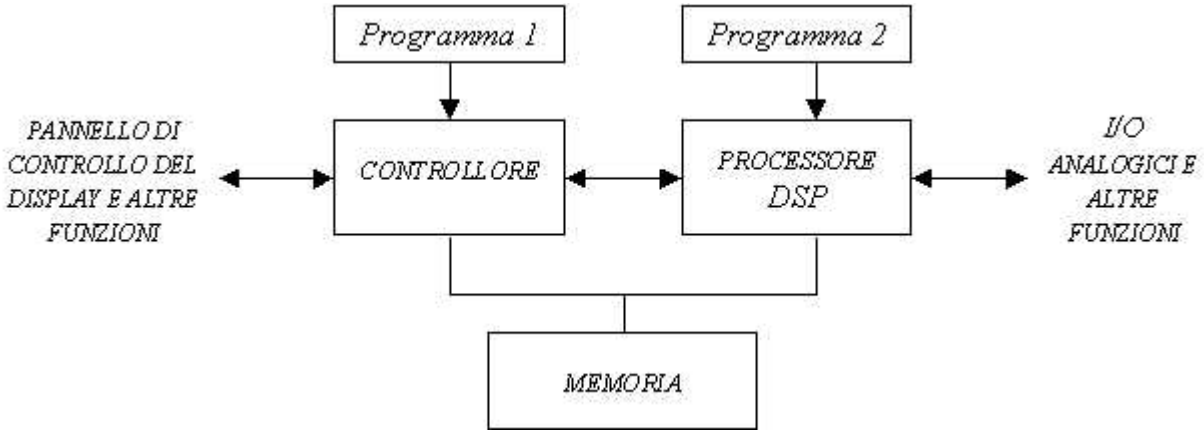

Figura 2: architettura tradizionale di un sistema che richiede controllo e funzionalità DSP, si possono notare i due flussi che seguono le istruzioni di un programma (Programmi 1 e Programma 2).

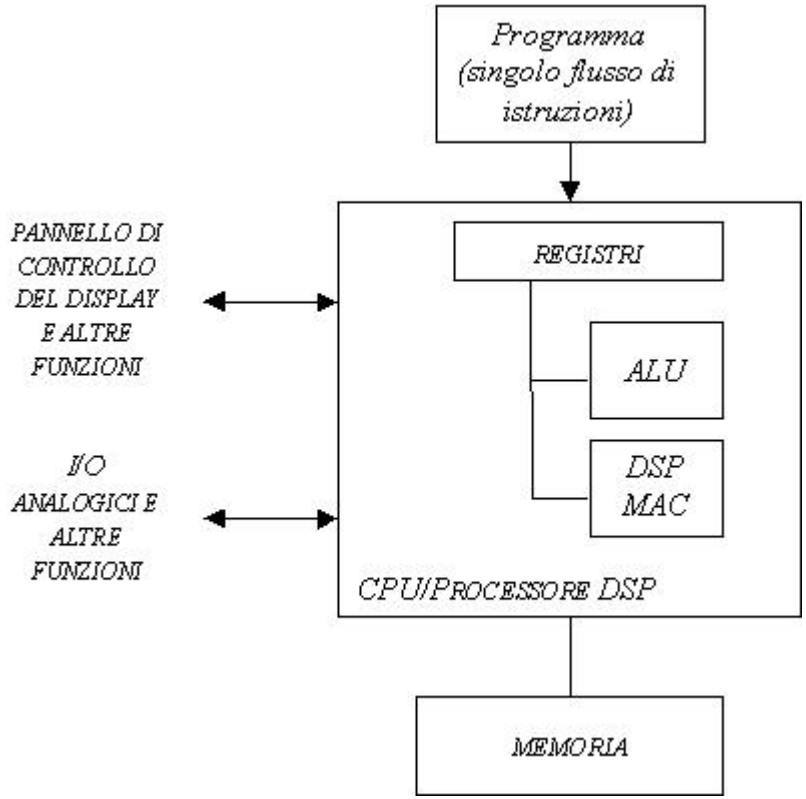

Figura 3: architettura dei sistemi che adottano l'approccio ad un solo processore embedded che contiene anche funzionalità DSP.

- periferiche e integrazione

  Nel campo dei sistemi embedded le scelte fatte nella realizzazione di un processore devono tener conto di quali periferiche integrare nel sistema. A questo punto si hanno diverse possibilità di scelta: ad esempio si potrebbe integrare tutto sulla stessa matrice, ma questa non è sempre la soluzione più economica perché aumenta la complessità del chip così realizzato e diminuisce, dall'altro lato, la sua resa, senza contare la necessità di una fase di testing più ampia. Ci sono due strategie di integrazione delle periferiche che non si differenziano su un piano tecnico, ma vengono adottate a seconda dell'ambito in cui è inserito il sistema:

- si può fornire un "core" di base e integrare logica addizionale per una device "personalizzata";
- si può offrire un processore standard insieme con un altro chip che soddisfi le esigenze specifiche dell'applicazione; si può aumentare la flessibilità fornendo una famiglia di tali chip o cambiando il processore principale. L'uso di architetture che adottano chip aggiuntivi favorisce future standardizzazioni nel mercato embedded.

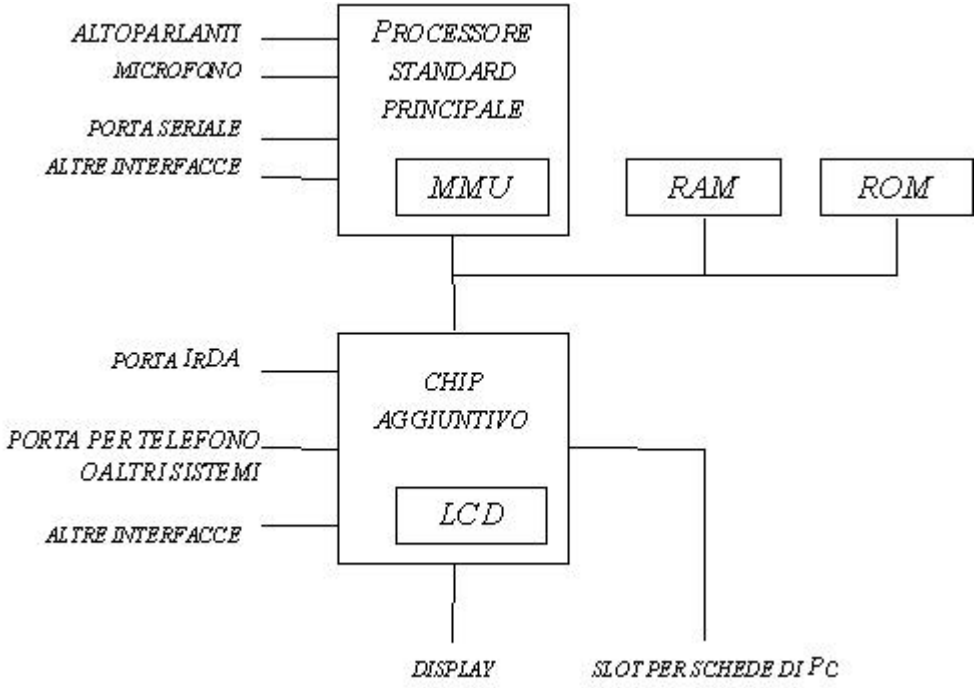

Figura 4: esempio di devices con un approccio a due chip.

La tendenza nel mercato embedded è lo sviluppo di nuove classi di processori e controllori a 32-bit: si parte da controllori low-end a 10 MHz per arrivare a processori embedded che fanno parte dell'area a 300 MIPS.

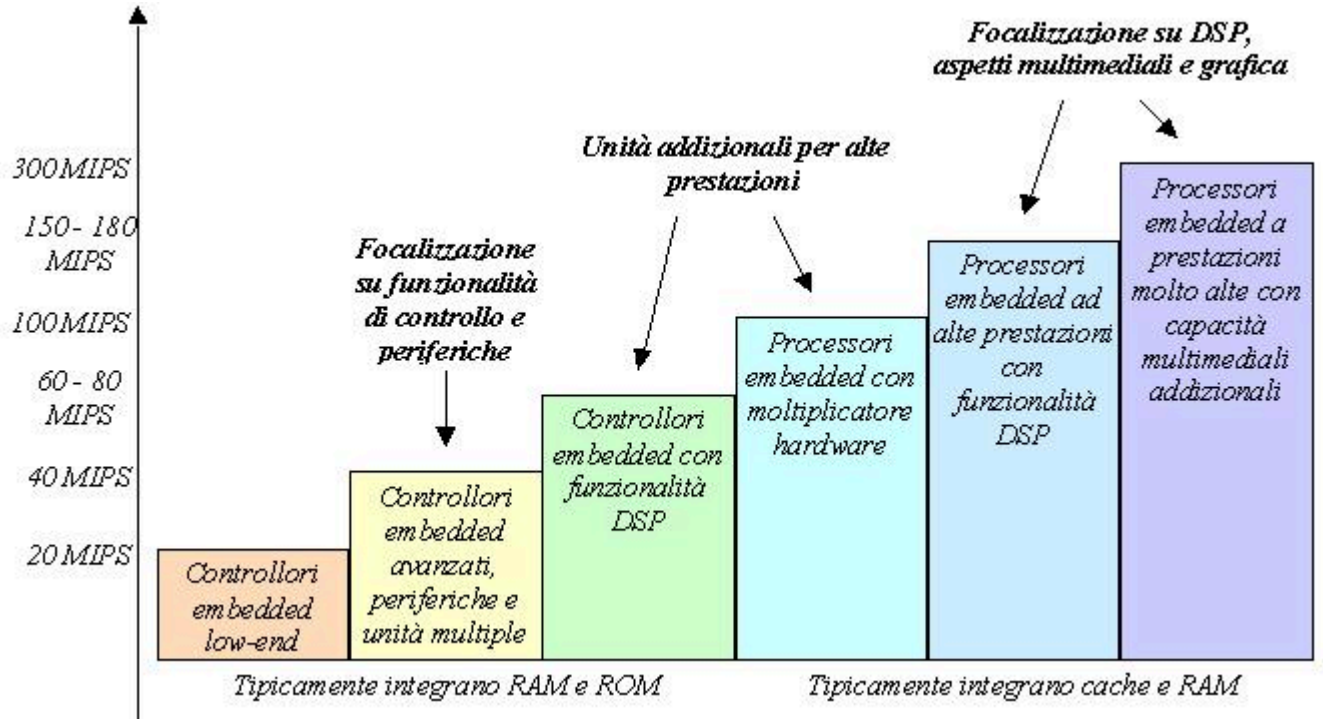

Figura 5: visione del mercato dei processore e controllori embedded a 32-bit.

Alla domanda se ci sarà mai un architettura dominante come nel dominio dei sistemi desktop, oggi bisogna rispondere negativamente perché il mondo embedded è guidato da una grande varietà di applicazioni; tuttavia la tendenza ad usare sistemi operativi e piattaforme standard c è ed è dovuta alla necessità di ridurre i costi di sviluppo, aumentare la riusabilità e migliorare il tempo di ciclo di design. Una possibilità realistica è che ogni segmento di mercato abbia un architettura dominante.