

1 Introduzione ai Sistemi a Microprocessore

Il **microprocessore** è uno dei dispositivi più utilizzati in elettronica digitale. Comprende le parti fondamentali di un **calcolatore numerico**, integrate su di un singolo circuito integrato. La sua realizzazione è stata resa possibile dallo sviluppo delle tecnologie di integrazione degli anni '70 (LSI, VLSI) che, a costi modesti, hanno permesso la realizzazione di circuiti integrati molto piccoli (qualche mm di lato) ma, al contempo, adeguatamente complessi. Attualmente, a seconda del tipo, un microprocessore può essere costituito da un numero di transistori compreso tra il centinaio di migliaia e alcuni milioni, integrati su di un unico chip ("scheggia") di silicio, o altro materiale semiconduttore.

Microprocessore (' μ P') e **microcalcolatore** (' μ C') non sono la stessa cosa. Semplicemente, il microprocessore è l'elemento centrale di un microcalcolatore. Quest'ultimo infatti identifica un sistema completo che, come vedremo nel seguito, comprende oltre al microprocessore altri dispositivi: la **memoria**, i circuiti di **ingresso/uscita** dei dati, e altro. Un Personal Computer, ad esempio, è un microcalcolatore.

Quando su di in un singolo dispositivo troviamo integrati tutti i dispositivi di un microcalcolatore, si parla di microcalcolatore **single-chip**. Un particolare tipo di microcalcolatore su singolo chip è il **microcontrollore**. Un microcontrollore, come dice il nome, è progettato per applicazioni di controllo. Oltre agli elementi base di tutti i microcalcolatori, vi troviamo integrati anche dispositivi di ingresso/uscita specializzati, appositamente progettati per il controllo di impianti industriali e/o sistemi complessi. A livello della vita di tutti i giorni, ad esempio, troviamo i microcontrollori impiegati nei televisori, nei videoregistratori, nei lettori di compact disk...

Un microcalcolatore inserito all'interno di un sistema più complesso è chiamato **embedded**. In una applicazione embedded l'utente vede e usa l'intero sistema (per esempio la propria autovettura) senza percepire direttamente la presenza di uno o più microcalcolatori al suo interno.

Un altro tipo di microcalcolatore specializzato è il **processore digitale di segnali** (DSP, Digital Signal Processor), appositamente progettato e ottimizzato per il trattamento, con tecniche digitali, di segnali. Si pensi al trattamento di segnali audio nei sistemi multimediali, come ad esempio nelle schede audio dei personal computer, nei lettori MP3, o negli effetti speciali utilizzati nel settore della produzione musicale o nella cinematografia.

Un generico calcolatore digitale adempie alle seguenti funzioni:

- acquisisce dati dal "mondo esterno";
- li memorizza;
- li elabora, producendo dei risultati;
- interviene nei riguardi del "mondo esterno" in base ai risultati ottenuti.

Si pensi ad esempio ad un microcalcolatore impiegato come sistema di controllo di un impianto industriale. In questo caso il "mondo esterno" è l'impianto; i dati sono acquisiti dal calcolatore mediante opportuni sensori e/o trasduttori collocati nell'impianto stesso (sensori di velocità, di pressione, di temperatura; contapezzi...); dopo la memorizzazione e l'elaborazione dei dati, in base ai risultati ottenuti il calcolatore interverrà sull'impianto tramite specifici attuatori (motori, elettromagneti, pistoni idraulici, elettrovalvole, elettrofreni...).

Anche un sistema progettato come macchina a stati finiti (MSF) può adempiere a queste funzioni, ma un sistema di controllo realizzato a microprocessore presenta molti vantaggi, consentendo potenzialità e flessibilità di gran lunga maggiori.

La differenza più importante tra la soluzione con MSF e quella con μC consiste nel modo in cui il progettista definisce la sequenza delle operazioni che il sistema deve compiere:

- Nella soluzione con **MSF** la sequenza deve essere rigorosamente **predefinita** dal progettista; la MSF è specificatamente “**dedicata**” alla particolare applicazione. Il suo **algoritmo** prevederà una ben precisa evoluzione degli stati, non modificabile a posteriori se non riprogettando da capo la macchina, poichè la sua funzionalità dipende dal particolare collegamento dei dispositivi logici che lo compongono (si parla spesso di **sistema cablato**, che deriva dall'inglese “cabled”, ossia “collegato mediante fili”).
- Nella soluzione con **microcalcolatore**, invece, le sequenze di operazioni da compiere sono predisposte dal progettista attraverso l'impostazione di un **programma**, scritto in un opportuno linguaggio di programmazione. In questi sistemi si distingue tra **hardware** (letteralmente “ferramenta”, ossia il sistema fisico composto di circuiti elettronici) e **software** (il programma che ne specializza il funzionamento). L'architettura del sistema è concepita in modo da rendere possibile modifiche alla sequenza delle operazioni da svolgere senza modificare i circuiti che le svolgono, ma soltanto il “programma” (si parla di **sistema programmato**).

Un vantaggio offerto dai microcalcolatori deriva proprio dalla loro architettura di tipo generale e **modulare**, che può essere adattata alle particolari applicazioni per mezzo di una opportuna scelta dei dispositivi che lo compongono e alla scrittura degli appositi programmi.

Sovente lo stesso hardware può essere utilizzato in applicazioni diverse: può essere sufficiente, infatti, **cambiare solo il programma** che ne specializza il funzionamento.

La modularità e la standardizzazione delle architetture permettono di mantenere i costi di progetto relativamente bassi, specie per i sistemi complessi, in quanto il progettista riutilizza interamente, in tutti i progetti di cui è incaricato, le stesse competenze.

1.1 Elementi fondamentali di un generico calcolatore digitale

La caratteristica peculiare di un generico calcolatore digitale è la capacità di eseguire in sequenza rigorosa, e in tempi relativamente brevi, **liste di istruzioni**. Una lista di istruzioni è chiamata **programma** e viene eseguita grazie alla particolare struttura interna del calcolatore, che nasce specificatamente per **eseguire istruzioni**.

Una **istruzione** è un **ordine impartito alla macchina**, ed identifica una particolare e circoscritta sequenza di operazioni elementari che la macchina eseguirà. Un dato calcolatore è progettato per eseguire un certo **insieme finito di istruzioni**, e solo quelle.

Ad esempio, alcune istruzioni ordinano di copiare un dato da un registro ad un altro, altre di eseguire una somma, oppure di incrementare un numero, ecc.

L'architettura più usata nella costruzione degli attuali calcolatori è quella di **Von Neumann** ⁽¹⁾ che, in sintesi, si basa sui seguenti concetti fondamentali:

- I dati e i programmi condividono lo stesso sistema di memoria;
- È presente una sola unità di elaborazione;
- È eseguita una sola istruzione per volta.

L'architettura di Von Neumann “funziona” dal 1945: questa impostazione, pensata soprattutto in funzione della massima semplicità circuitale, e forse proprio per questo, a livello commerciale non è stata ancora soppiantata da altre, anche se i progressi della scienza dei calcolatori hanno consentito di arrivare ai cosiddetti “supercomputer” e alle macchine “parallele”.

¹ Johann Ludwig Von Neumann (1903-1957), matematico e scienziato di origine tedesca, docente di fisica matematica all'università di Princeton (USA), ove realizzò nel 1945 l'EDVAC (Electronic Discrete Variable Automatic Computer).

Da un punto di vista logico, un calcolatore numerico di questo tipo può essere scomposto in tre blocchi (Fig. 1.1):

- **L'Unità di Memoria (MEM, Memory);**
- **L'Unità di Ingresso/Uscita (I/O, Input/Output).**
- **L'Unità Centrale di Elaborazione (CPU, Central Processing Unit);**

In estrema sintesi, i tre moduli svolgono le seguenti funzioni:

- La **Memoria** contiene sia i dati, su cui il calcolatore lavora, sia i programmi che agiscono sui dati stessi e sul mondo esterno;
- L'unità di **Ingresso/Uscita** consente alla macchina di "collegarsi" al mondo esterno;
- La **CPU** ha il compito di reperire nella memoria ed eseguire, una per una, le istruzioni che compongono il programma; la CPU, inoltre, legge e scrive dati nella memoria e "dialoga", attraverso l'unità di Ingresso/Uscita, con il mondo esterno.

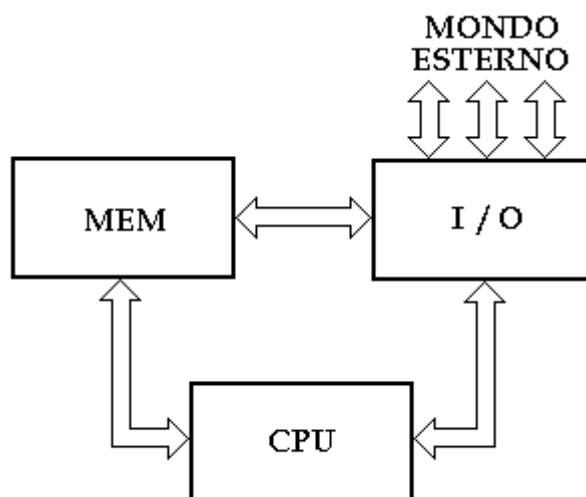


Fig.1.1: Schema logico di un calcolatore digitale

La **CPU** gioca un **ruolo fondamentale**: è, insieme, attore principale e regista del sistema. L'autore del "copione" (il programma) è il progettista. L'intero sistema è una sorta di teatro in cui i singoli attori agiscono rispettando rigorosamente l'ordine imposto dalla **sequenza delle istruzioni** del programma, senza svolgere nulla di autonomo che non derivi dalla esecuzione di una istruzione.

Fin dal primo momento (cioè da quando il calcolatore viene acceso), la CPU preleva in memoria una istruzione e la esegue, per poi tornare in memoria a prelevare la successiva istruzione ed eseguire anche quella, e così via... in **ciclo infinito**.

1.1.1 Considerazioni sulla esecuzione delle istruzioni

Oggi possiamo osservare microcalcolatori potentissimi in grado di eseguire funzioni impensabili solo fino a qualche anno fa (si pensi ai video-giochi delle ultime generazioni). Da questa semplice osservazione potremmo pensare che le attuali CPU siano in grado di eseguire istruzioni potentissime, ma in generale non è così.

Nonostante le apparenze, le singole istruzioni direttamente eseguibili da una CPU permettono di eseguire compiti piuttosto elementari. Ciò che rende il microcalcolatore in grado di eseguire così bene compiti come quelli sopra accennati è la sua velocità: le funzioni complesse saranno ottenute eseguendo in modo velocissimo programmi composti, in realtà, di migliaia di istruzioni elementari.

In generale, l'esecuzione di una istruzione comporta semplicemente il trasferimento dei dati dalla memoria (o dai dispositivi di ingresso) nella CPU, la loro elaborazione (somma, sottrazione, confronto...) e il trasferimento del risultato nella memoria (o nei dispositivi di uscita).

Nel seguito si potrà notare che la maggior parte del tempo (il cosiddetto "tempo macchina") è speso dalla CPU nel leggere istruzioni e nel trasferire dati da una parte all'altra; nonostante l'elevata velocità complessiva di un sistema a microcalcolatore, il tempo effettivamente impiegato per lo svolgimento dei singoli calcoli è solo una piccola parte del tempo totale!

Questa osservazione sembrerebbe sminuire le possibilità dei microcalcolatori, ma occorre subito notare che, per i sistemi ad elevata complessità, la flessibilità e le potenzialità di un *sistema programmabile* restano ineguagliabili.

Infatti, se progettassimo una (o più) MSF per ottenere lo stesso sistema, questo risulterebbe molto più veloce (a pari tecnologia e a pari frequenza di clock), ma sarebbe anche un ingestibile e costosissimo "mostro" di complessità. Per cui, nella stragrande maggioranza dei casi, si opta per una architettura centrata su di un microcalcolatore.

In quei casi in cui il "fattore tempo" dovesse risultare prioritario, in sede di definizione della architettura complessiva del sistema i progettisti ne individueranno gli "elementi critici" e

provvederanno a separare il trattamento di quei segnali che necessitano di velocità dal resto della elaborazione. Solo questi elementi critici saranno quindi progettati “localmente” come veloci MSF dedicate, ma il sistema nel suo complesso continuerà ad avere al centro il microcalcolatore.

1.2 L'architettura a BUS comune

Un microcalcolatore basato sui concetti di Von Neumann è normalmente organizzato intorno ad un unico **BUS** comune (Fig. 1.2). Un **BUS** è un insieme di fili collegato **in parallelo** tra le varie unità componenti il sistema. Il suo nome deriva dall'omonimo mezzo di trasporto: è in effetti tramite questo insieme di fili che l'informazione viene **trasportata** da una unità all'altra. La scelta di questo tipo di collegamento è dettata principalmente da considerazioni di tipo **economico** e di praticità. Nella figura si riconoscono i tre blocchi fondamentali discussi in precedenza: la comunicazione tra di questi avviene utilizzando il fascio di fili in **comune** del BUS.

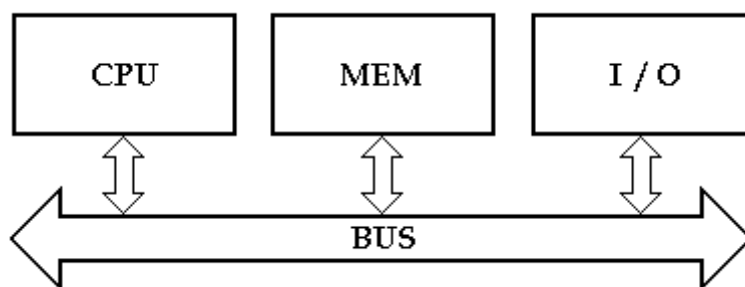


Fig.1.2: Struttura a BUS di un microcalcolatore

Una sola unità alla volta può “impadronirsi” di questi fili per inviare informazioni sul BUS, mentre (potenzialmente) tutti i blocchi connessi possono riceverli in contemporanea. L'unità che in un dato momento trasmette è chiamata **talker**; chi riceve **listener**. L'unità che gestisce chi può o non può appropriarsi dei fili del bus in un dato momento è denominata **master** (o **arbiter**), mentre le altre, che obbediscono, sono chiamate **slave**. Anche se esistono sistemi **multi-master** (come i personal computer multiprocessore, solo per citare un esempio), nel nostro corso esamineremo soltanto sistemi più semplici, aventi la **CPU** come **unico master**.

Una osservazione sull'efficienza del collegamento a BUS. Ogni sotto-unità del sistema si collega alle altre mediante lo stesso insieme di fili. Il BUS assume quindi la caratteristica di **risorsa condivisa**. È evidente che la condivisione dello stesso fascio di fili tra tutti i dispositivi presenti è **una limitazione funzionale**: se due sotto-unità in un certo istante dialogano tra di loro, di fatto occupano il BUS, impedendone l'uso alle altre sotto-unità presenti nel sistema.

Se potessimo collegare tra loro le singole unità mediante **collegamenti riservati**, otterremmo potenzialmente un sistema molto più veloce (sarebbe un sistema “a stella” e non più “a BUS”). Tuttavia, per un microcalcolatore basato sui concetti di Van Neumann questa limitazione non è grave, dal momento che sarà eseguita **una sola operazione alla volta**.

Inoltre, un collegamento di tipo BUS è di gran lunga **più economico** che un collegamento “stellato”, che richiederebbe molti più fili per le interconnessioni “dedicate”.

Infine, un sistema a BUS è più **osservabile** di un sistema con collegamenti a stella: collegando un opportuno strumento diagnostico sui fili del BUS potremo osservare **tutto** quanto viene scambiato tra i vari moduli, permettendo una ricerca guasti e/o malfunzionamenti ragionevolmente semplice.

Per comodità di trattazione, scendiamo ora in maggior dettaglio nella suddivisione dei blocchi funzionali che compongono il sistema (Fig.1.3).

Lasciamo per ora in evidenza la **CPU** come un blocco unico (**master** del sistema), e suddividiamo invece la memoria in due sotto-unità: la **RAM** (memoria di lettura e scrittura) e la **ROM** (memoria di sola lettura), che discuteremo nel capitolo successivo.

Suddividiamo anche il sistema di **I/O** in vari dispositivi, che immagineremo ciascuno specializzato per ogni particolare collegamento con il “mondo esterno” di cui il nostro (generico) sistema avrà bisogno.

Osservando la figura, si nota anche la suddivisione del fascio di fili del BUS in tre sotto-insiemi di collegamenti:

- Il **Bus dei DATI** (Data Bus): consente il trasporto dell'informazione vera e propria (dati e istruzioni) da e verso i vari dispositivi; dal momento che i dati possono essere trasferiti in tutte le direzioni, lo si definisce come collegamento **bidirezionale**;
- Il **Bus degli INDIRIZZI** (Address Bus): come vedremo, consente alla CPU di selezionare il dispositivo "corrente" con il quale scambiare i dati di cui sopra. Poichè la CPU è l'unico master del nostro sistema, gli indirizzi saranno **generati soltanto dalla CPU** e ricevuti da tutti gli altri elementi (vedi, nella figura, la direzione delle frecce);
- Il **Bus dei CONTROLLI**: collegamenti di varia natura, anche funzionalmente molto diversi tra di loro, ma necessari per il controllo dell'intero sistema e per la gestione (arbitraggio) dello scambio di informazioni sugli altri due sotto-bus.

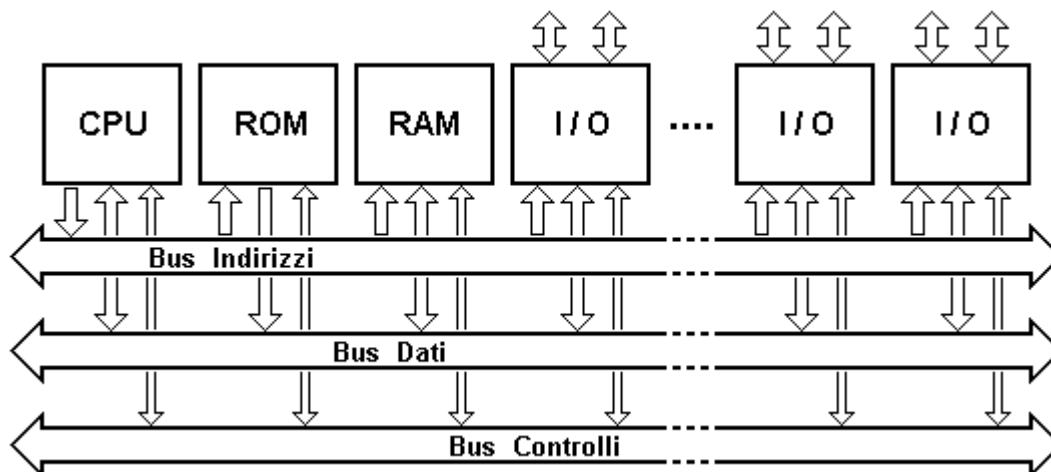


Fig.1.3: Architettura a BUS

La CPU, master del sistema, gestisce lo scambio di informazioni pilotando opportuni segnali sul Bus dei Controlli ma, soprattutto, definendo il dispositivo con il quale dialogare in un dato momento, mediante il suo **Indirizzo**. Tutti i dispositivi presenti nel sistema possiedono **un particolare e univoco indirizzo**, rappresentato da un numero di identificazione.

Supponiamo che la CPU necessiti di **trasmettere un dato** al dispositivo di uscita di indirizzo <n>. La CPU invia l'indirizzo <n> all'intero sistema, tramite il Bus degli Indirizzi. Tutti i dispositivi sono all'ascolto, ma soltanto uno riconoscerà il proprio indirizzo, e solo questo dispositivo attiverà i suoi circuiti in modo da mettersi in grado di **ricevere** il dato. A questo punto, la CPU trasmetterà il dato sul Bus dei Dati, che sarà acquisito soltanto dal dispositivo attivo. Tutto questo avviene, naturalmente, con il supporto dei segnali di controllo e temporizzazione del Bus dei Controlli.

All'opposto, supponiamo ora che la CPU voglia **ricevere un dato** da un dispositivo di ingresso di indirizzo <m>. Anche in questo caso la CPU invia l'indirizzo <m> all'intero sistema, tramite il Bus degli Indirizzi. Soltanto un dispositivo riconoscerà l'indirizzo <m>, e solo questo dispositivo attiverà i suoi circuiti in modo da mettersi in grado di **inviare** il dato. Ottenuto questo, la CPU si mette in attesa del dato, che il dispositivo attivo invierà tramite il Bus dei Dati. Anche in questo caso, il supporto dei segnali del Bus dei Controlli è fondamentale affinché le operazioni si svolgano nella giusta sequenza.

Le **memorie**, come vedremo nel prossimo capitolo, sono internamente suddivise in **locazioni**. Il criterio dell'indirizzo si applica ad ogni singola locazione, per cui se, ad esempio, nel nostro sistema abbiamo 65.536 (2^{16}) locazioni di memoria, **ciascuna sarà identificata da un indirizzo** (da 0 a 65.535). La CPU accederà, sia in scrittura che in lettura, ad una sola locazione di memoria alla volta, specificando sul Bus degli Indirizzi il particolare indirizzo della locazione, esattamente come se ciascuna locazione di memoria fosse un dispositivo diverso: quanto esemplificato qui sopra per i dispositivi di ingresso e uscita si può ripetere per le singole locazioni di memoria.