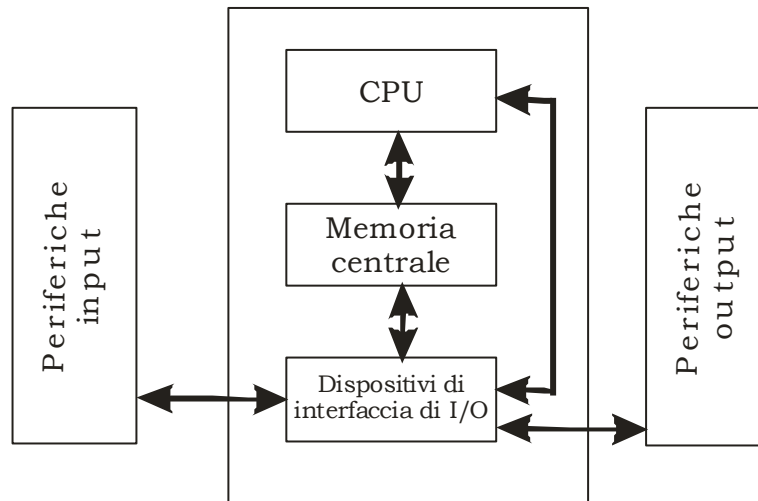


Il sistema a microprocessore



Il disegno precedente rappresenta uno schema logico molto semplificato di un sistema a **microprocessore**. E' definito come schema logico in quanto ognuna delle "scatolette" non si riferisce necessariamente ad un solo componente elettronico presente all'interno del sistema, ma in generale ciascun blocco fa riferimento ad un insieme di dispositivi che svolgono congiuntamente una determinata funzione all'interno del sistema stesso. Ognuna di queste "scatole" rappresenta un'unità funzionale.

Possiamo individuare all'interno del sistema a μP tre unità funzionali:

- la **CPU** (Central Processing Unit)
- la **memoria centrale**
- i **dispositivi di interfaccia di input/output**.

Le unità funzionali scambiano informazioni tra di loro mediante linee di comunicazione, che vengono raggruppate tra di loro secondo la natura delle informazioni trasportate in insiemi di linee detti **bus**. Su ogni linea appartenente ai bus transitano segnali elettrici digitali, i cui valori di tensione variano tra circa 0 V e 0.8 V nel caso in cui il valore binario codificato sulla linea sia lo 0, e tra circa 2.5 V e 5 V nel caso in cui il valore binario sia 1; questi valori sono definiti dallo standard **TTL** (Transistor Transistor Logic).

Esaminiamo di seguito in modo più specifico le unità funzionali ed i collegamenti tra di esse.

La CPU

La **Central Processing Unit** (CPU) è l'unica unità funzionale che coincide dal punto di vista fisico con un solo componente: il microprocessore (abbreviato **μP**). E' un dispositivo **sincrono e programmabile**.

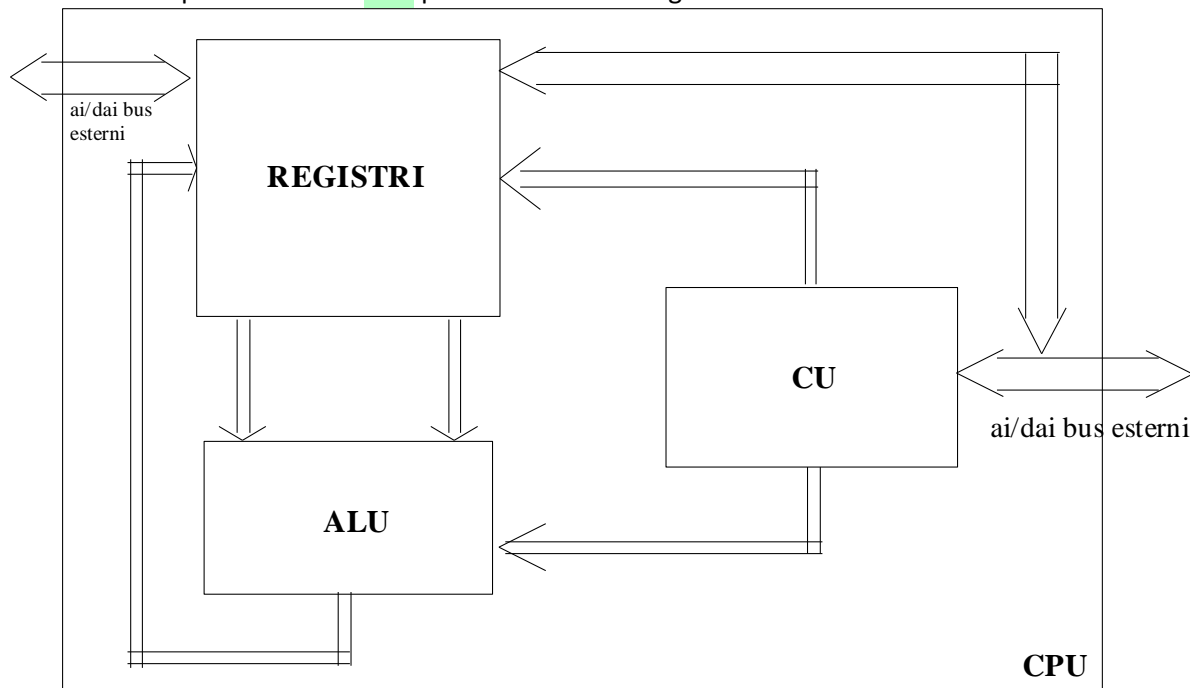
Si dice che la CPU è un dispositivo **sincrono** in quanto la successione delle azioni che compie viene scandita da un'onda rettangolare generata da un apposito dispositivo detto **generatore di clock**. Questo segnale periodico presenta alcune caratteristiche quali il **duty cycle** (la percentuale di tempo in un periodo in cui il segnale rimane a livello logico alto), il **periodo** (tempo che impiega il segnale partendo da uno stato per tornare nel medesimo stato; si misura in secondi o suoi sottomultipli) e la **frequenza** (numero di periodi nell'unità di tempo; si misura in Hertz). La **frequenza del clock** è uno dei **parametri che determinano la velocità del μP** ; ad esempio, nell'INTEL 8086 il clock era a 10 MHz, mentre negli attuali microprocessori la frequenza è dell'ordine dei GHz.

La CPU è inoltre un dispositivo **programmabile** in quanto le sue azioni sono determinate da un programma, cioè da una **sequenza di istruzioni elementari che complessivamente svolgono una certa funzione**. Le istruzioni che la CPU è in grado di eseguire sono le istruzioni in **"linguaggio macchina"**; un programma scritto in un qualsiasi linguaggio di programmazione deve quindi essere tradotto in un programma "equivalente" (= che a fronte degli stessi input restituisce gli stessi output) le cui istruzioni sono in linguaggio macchina.

La CPU costituisce il "cuore", o meglio il "cervello", del sistema; infatti, le azioni che svolge sono in generale le seguenti:

- **Esegue in sequenza le istruzioni del programma in corso d'esecuzione;**
- **Esegue i calcoli;**
- **Coordina e sincronizza tutte le attività dei dispositivi che costituiscono il sistema a μP e che sono ad essa collegati tramite i bus.**

Uno schema molto semplificato di una CPU potrebbe essere il seguente:



Secondo lo schema precedente, all'interno del microprocessore possiamo schematizzare alcuni blocchi funzionali:

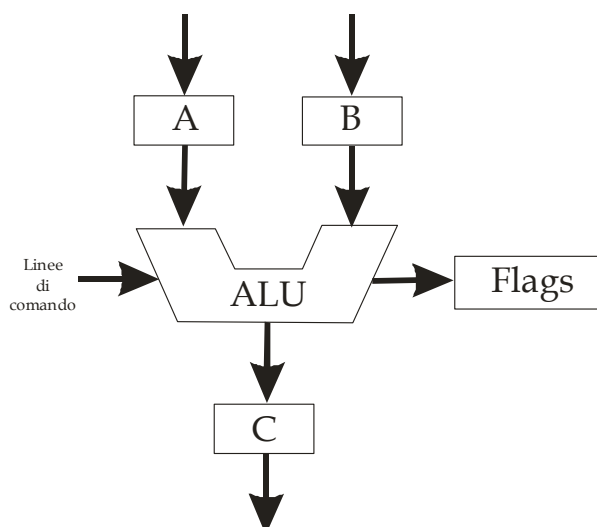
- la **ALU** (Arithmetic Logic Unit);
- la **CU** (Control Unit);
- i **registri**.

La Arithmetic Logic Unit.

L'Unità Aritmetico Logica è l'unità funzionale della CPU preposta all'**elaborazione dei dati**. E' costituita da un **circuito combinatorio programmabile** ed è in grado di eseguire un numero limitato di operazioni logiche e matematiche. Basterebbero quattro semplici operazioni (**AND**, **NOT**, **somma** e **identità**) per permettere alla CPU di poter eseguire tutte le operazioni aritmetiche e logiche, anche le più complesse; infatti la combinazione mediante opportuni algoritmi di queste operazioni semplici permette di eseguire anche le operazioni più complesse. Ad esempio, in una CPU che presenti le quattro operazioni citate in precedenza, per poter eseguire l'operazione booleana OR su due operandi A e B occorre eseguire più operazioni AND e NOT in sequenza in modo tale da ottenere il risultato di una OR applicando la legge di De Morgan :

$$A \oplus B = \overline{\overline{A} \otimes \overline{B}}$$

Ad un livello di dettaglio superiore, la ALU può essere rappresentata nel seguente modo:



I due oggetti denominati A e B sono due registri temporanei che contengono i dati in input alla ALU. Le operazioni che la ALU può eseguire possono essere a due operandi (ad es. l'addizione) o ad un operando (ad es. l'operazione logica NOT); nel caso in cui l'operando sia uno solo, questo viene caricato nel registro temporaneo A. I dati in input vengono presentati alla ALU mediante dei bus interni alla CPU e il risultato memorizzato temporaneamente nel registro C; il risultato viene quindi salvato nella destinazione tramite un altro bus interno.

Come si osserva dallo schema logico, il risultato non è l'unico output prodotto dall'ALU a seguito dell'esecuzione di una operazione; infatti l'esecuzione di un'operazione modifica anche il contenuto del registro dei flags, o registro di stato che esamineremo più in dettaglio nel paragrafo dedicato ai registri.

La Control Unit

La Control Unit è la parte più complessa della CPU. Se la CPU è il "cervello" del sistema a microprocessore, la Control Unit può essere considerata il "cervello" del microprocessore: il suo compito è quello di mandare in esecuzione l'istruzione corrente e attivare di conseguenza gli opportuni segnali di controllo e di sincronismo verso i dispositivi interessati dall'istruzione.

L'istruzione corrente (cioè in corso di esecuzione) è contenuta nel registro **IR (Instruction Register)**. Sul contenuto di questo registro la Control Unit esegue un'operazione di decodifica; tale operazione consiste nell'esame del codice operativo dell'istruzione. Conseguentemente al test dei bit che compongono il codice operativo dell'istruzione contenuta in IR, la CU provvede all'attivazione dei segnali che rendono operativi i dispositivi coinvolti nell'esecuzione dell'istruzione stessa. La CU si preoccupa inoltre di sincronizzare le operazioni relative al reperimento degli operandi dell'istruzione, siano essi contenuti in un registro interno alla CPU o in una locazione di memoria, e di coordinare le azioni che permettono l'eventuale memorizzazione del risultato dell'istruzione.

I registri

I registri interni alla CPU sono aree di memoria temporanee, di dimensioni molto limitate, ma con tempi di accesso molto ridotti. Le funzioni svolte dai registri sono quelle di memorizzare sia i dati che costituiscono gli operandi e i risultati delle istruzioni, sia i riferimenti alle locazioni di memoria centrale in cui possono trovarsi gli operandi e i risultati.

Alcuni registri sono accessibili al programmatore in modo **diretto** o **indiretto**; altri, come il già citato IR (*Instruction Register*), non sono in alcun modo visibili al programmatore e vengono gestiti in modo automatico dal microprocessore.

Possiamo suddividere i registri nei seguenti gruppi:

- **Registri di dati:** sono usati per memorizzare numeri interi. Nelle CPU più semplici o più vecchie, uno speciale registro per i dati è l'accumulatore, usato per calcoli aritmetici.
- **Registri di indirizzo:** contengono gli indirizzi e sono usati per accedere alla memoria.
- **Registro di istruzione:** contiene l'istruzione corrente.
- **Registri speciali:** contengono dati interni della CPU, come l'Instruction Pointer, lo Stack Pointer e il Registro di Stato.
- **Registri indice:** contengono l'indirizzo degli operandi.
- **Registro di stato:** i bit di questo registro, detti anche flag, sono impostati dalla ALU a fronte dell'esecuzione di un'operazione logica o aritmetica e segnalano alcune "caratteristiche" del risultato. Abbiamo ad esempio:
 - **Flag di segno:** indica se il risultato dell'operazione è positivo (0) o negativo (1);
 - **Flag di zero:** indica se il risultato dell'operazione è uguale (1) o diverso (0) da zero;
 - **Flag di overflow:** indica se l'operazione ha generato un errore di overflow;
 - **Flag di carry:** consente di rilevare il "riporto" di un'operazione.I flag rivestono notevole importanza in quanto vengono testati dalla CU per verificare la possibilità di poter effettuare all'interno dei programmi istruzioni di "salto condizionato".
- **Registro IP (Instruction Pointer):** il registro IP è accessibile in modo **indiretto** dal programmatore e contiene il riferimento della successiva istruzione che verrà eseguita dalla CPU. Il registro IP viene modificato dalla CPU in due modi:
 - quando non viene eseguita un'istruzione di salto il valore di IP viene calcolato aggiungendo al valore corrente del registro la dimensione dell'istruzione in corso di esecuzione; in questo modo viene ovviamente calcolato l'indirizzo della successiva istruzione che verrà eseguita:

043B _H	MOV	AL,6	IP: 043D _H
043D _H	ADD	AL,BL	IP: 043F _H
043F _H	MOV	CL,AL	

- quando viene eseguita un'istruzione di salto, il valore di IP viene modificato dall'esecuzione dell'istruzione stessa; la label argomento dell'istruzione di salto rappresenta infatti l'offset dell'istruzione che deve essere eseguita. Pertanto nel registro IP verrà caricato:

- nel caso di salto incondizionato, il valore numerico (cioè l'offset) rappresentato dalla label:

0286H	MOV	CX,18	IP: 0288H
0288H	JMP	LABEL1	IP: 028DH

```

028BH LAB:
028BH      MOV    CL,AL
028DH LABEL1:
028DH      SUB    DL,DH

```

- nel caso di salto condizionato:

- il valore rappresentato dalla label se la condizione è verificata:

```

0AB2H      MOV    AL,0      IP: 0AB4H
0AB4H      CMP    AL,0      IP: 0AB6H
0AB6H      JE     SALTO1    IP: 0ABDH
0AB8H      MOV    CL,AL
0ABAH      JMP    ESCI
0ABDH SALTO1:
0ABDH      ADD    CH,DL

```

- il valore corrente di IP più la dimensione dell'istruzione di salto:

```

0AB2H      MOV    AL,1      IP: 0AB4H
0AB4H      CMP    AL,0      IP: 0AB6H
0AB6H      JE     SALTO1    IP: 0AB8H
0AB8H      MOV    CL,AL
0ABAH      JMP    ESCI
0ABDH SALTO1:
0ABDH      ADD    CH,DL

```

La memoria centrale.

La memoria centrale è il luogo dove il sistema a microprocessore mantiene il programma in corso di esecuzione e i dati sui quali il programma opera. La memoria si divide in:

- **ROM (Read Only Memory):** è una memoria permanente il cui contenuto viene mantenuto anche quando il sistema non è alimentato. Il suo contenuto non può essere modificato (se non in casi eccezionali) e pertanto l'unico accesso consentito è la lettura. La ROM viene utilizzata come supporto per la memorizzazione di routines di sistema che svolgono specifiche funzionalità; in particolare, nella ROM sono memorizzate le routines del BIOS (Basic Input Output Service), cioè quelle procedure che consentono al sistema operativo di gestire le operazioni fondamentali di input - output.
- **RAM (Random Access Memory):** è la porzione di memoria centrale che contiene il programma in corso di esecuzione e i dati su cui opera il programma. E' una memoria volatile perché, a differenza della ROM, il suo contenuto viene perso quando il sistema non è più alimentato. La definizione più corretta per la RAM sarebbe quella di "memoria a lettura e scrittura"; in realtà la definizione di memoria ad accesso casuale nacque quando la tecnologia permise di passare dalle prime memorie a lettura e scrittura ad accesso sequenziale a memorie in cui l'accesso a qualsiasi locazione di memoria avviene con il medesimo tempo.

L'unità di informazione elementare contenuta all'interno della memoria centrale è il byte.

Il problema dell'indirizzamento

Il microprocessore deve essere in grado di identificare ognuna delle locazioni di memoria per leggere dalle locazioni stesse dati e/o istruzioni e per memorizzare all'interno di esse dei dati: tali operazioni di lettura e di scrittura devono essere eseguite su entità che devono essere individuate senza alcuna ambiguità.

I byte che appartengono alla memoria centrale sono identificati dalla CPU in base al loro indirizzo. L'indirizzo è un numero associato ad ognuno dei byte ed è tale che ad un indirizzo corrisponde uno ed un solo byte e, viceversa, ad un byte corrisponde uno ed un solo indirizzo; in altre parole esiste una corrispondenza biunivoca tra un byte e l'indirizzo che lo identifica.

Gli indirizzi dei byte sui quali la CPU vuole eseguire un'operazione di lettura o scrittura sono comunicati alla memoria tramite le linee di Address Bus; ognuna di queste linee può trasportare verso la memoria centrale il valore logico 0 o 1.

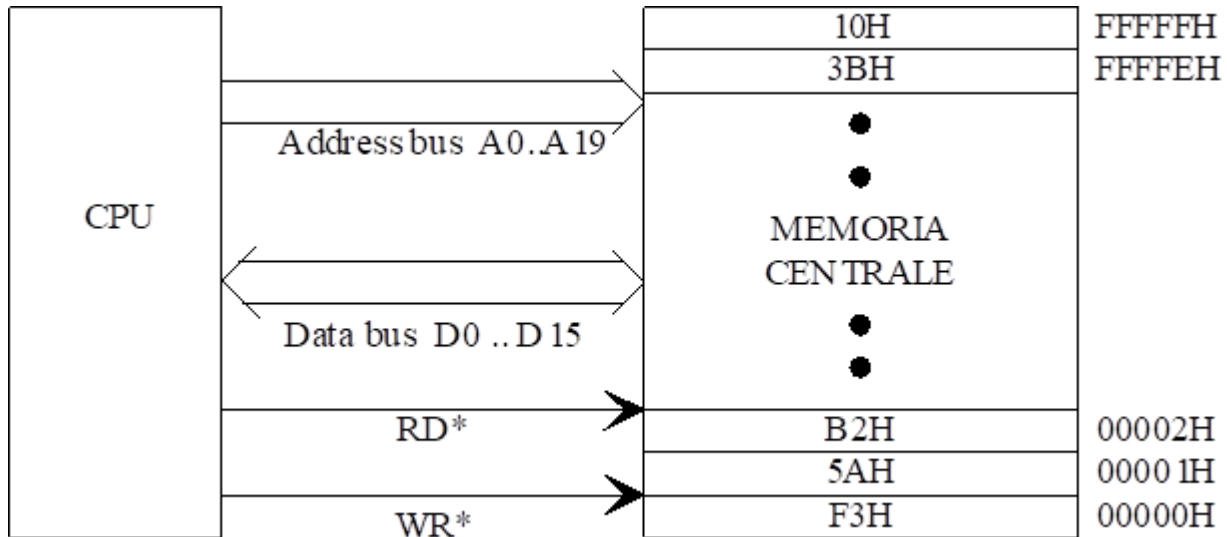
Ad esempio il microprocessore 8086 ha 20 linee di Address Bus: pertanto la sua capacità di indirizzamento è pari a $2^{20} = 1.048.576$ byte, mentre gli attuali microprocessori hanno 40 linee di Address Bus con una capacità di indirizzamento di 1 TB ($2^{40} = 1,099,511,627,776$ byte).

Pertanto, gli indirizzi che la CPU 8086 è in grado di comunicare alla memoria centrale coprono il seguente intervallo:

A ₁₉	A ₁₈	A ₁₇	A ₁₆	A ₁₅	A ₁₄	A ₁₃	A ₁₂	A ₁₁	A ₁₀	A ₉	A ₈	A ₇	A ₆	A ₅	A ₄	A ₃	A ₂	A ₁	A ₀
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
.
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

In sostanza l'intervallo degli indirizzi va da 0 a 1.048.575 espresso in base 10 oppure da 00000_H a FFFFF_H espresso in esadecimale. Quando la CPU vuole leggere o scrivere un dato in una determinata locazione di memoria, deve comunicare alla memoria tramite l'Address Bus l'indirizzo della locazione interessata e tramite le linee WR* e RD* il tipo di operazione che vuole effettuare, se lettura o scrittura.

Lo schema sottostante delinea in modo molto sintetico il collegamento tra la CPU 8086 e la memoria centrale.



Attraverso l'Address Bus la CPU comunica alla memoria l'indirizzo della locazione sulla quale vuole leggere o scrivere. Le linee WR* e RD* del Control Bus permettono alla memoria centrale di capire quale operazione la CPU vuole eseguire sulla locazione di memoria (*lettura o scrittura*) e il Data Bus è il mezzo attraverso il quale vengono trasferiti i dati o le istruzioni. Le possibili impostazioni delle linee RD* e WR* sono le seguenti:

RD*	WR*	
0	0	non possibile (la CPU non può richiedere contemporaneamente una lettura e una scrittura)
0	1	Lettura
1	0	Scrittura
1	1	nessuna operazione

I Bus

I bus sono insiemi di linee che collegano tra loro i componenti del sistema a μP .

Da un punto di vista fisico sono costituiti da piste di rame che connettono i pin dei dispositivi e sulle quali vengono trasportati i segnali elettrici che rappresentano i dati che vengono scambiati all'interno del sistema. I bus presenti all'interno di un sistema sono:

- **Address Bus:** L'Address Bus è **unidirezionale**, in quanto il segnale ha sempre la direzione CPU-memoria o CPU-dispositivi. Mediante questo bus la CPU comunica alla memoria o ai dispositivi l'indirizzo della locazione sulla quale vuole compiere un'operazione di lettura o di scrittura. La "larghezza" dell'Address Bus, ossia il numero di linee che lo costituiscono, stabilisce la capacità massima di memoria indirizzabile all'interno del sistema. La relazione che lega il numero di linee di Address Bus al numero di locazioni di memoria indirizzabili è:

$$m = 2^n$$

dove m è la quantità di locazioni di memoria indirizzabili e n è il numero di linee di Address Bus del Sistema. Essendo dotato di 20 linee di Address bus, un sistema con il μP 8086 ha pertanto una capacità massima di indirizzamento pari a $2^{20} = 1.048.576$ bytes (=1 Megabyte).

- **Data Bus:** Il Data Bus è un insieme di linee **bidirezionali**, in quanto la direzione del segnale può essere sia dalla CPU alla memoria o ai dispositivi (**operazione di scrittura**) che dalla memoria o dai dispositivi verso la CPU (**operazione di lettura**).

lettura). Attraverso il Data Bus vengono trasportati i dati da e verso la CPU e vengono trasferite all'interno della CPU le istruzioni in linguaggio macchina che devono essere eseguite. La "larghezza" del Data Bus, cioè il numero di linee che lo compongono, determina a "quanti" bit è il microprocessore (es. l'80286 è un microprocessore a 16 bit in quanto le linee del suo Data Bus sono 16 e lo stesso vale per gli attuali microprocessori a 64 bit). Ovviamente, più ampio è il Data Bus, più bit in parallelo vengono trasferiti e di conseguenza più veloce è l'esecuzione delle istruzioni.

- **Control Bus**: Nel control bus vengono convenzionalmente raggruppate tutte le linee che servono per permettere un corretto colloquio tra la CPU e la memoria o i dispositivi in modo da garantire la corretta operatività di tutto il sistema.

Su alcune linee la direzione del segnale è verso la CPU, su altre linee è in uscita alla CPU stessa. Tra le linee del Control Bus ricordiamo:

- **CLOCK** (→ CPU): segnale di temporizzazione del sistema;
- **RD*** e **WR*** (CPU →): segnali attivi bassi che indicano, quando impostati a 0, il tipo di operazione (lettura o scrittura) che la CPU vuole eseguire su un dispositivo;
- **RESET** (→ CPU): segnale che permette il riavvio del sistema.