

# ARCHITETTURA DEL CALCOLATORE

## AVVERTENZE

Questi appunti non vogliono essere degni sostituti di un buon libro di testo e hanno il solo scopo di aiutare gli studenti dei corsi di informatica che affrontano lo studio dell'architettura di un computer. Gli argomenti affrontati in aula saranno dunque qui trattati in modo stringato, dando per scontata l'attiva partecipazione degli studenti alle lezioni.

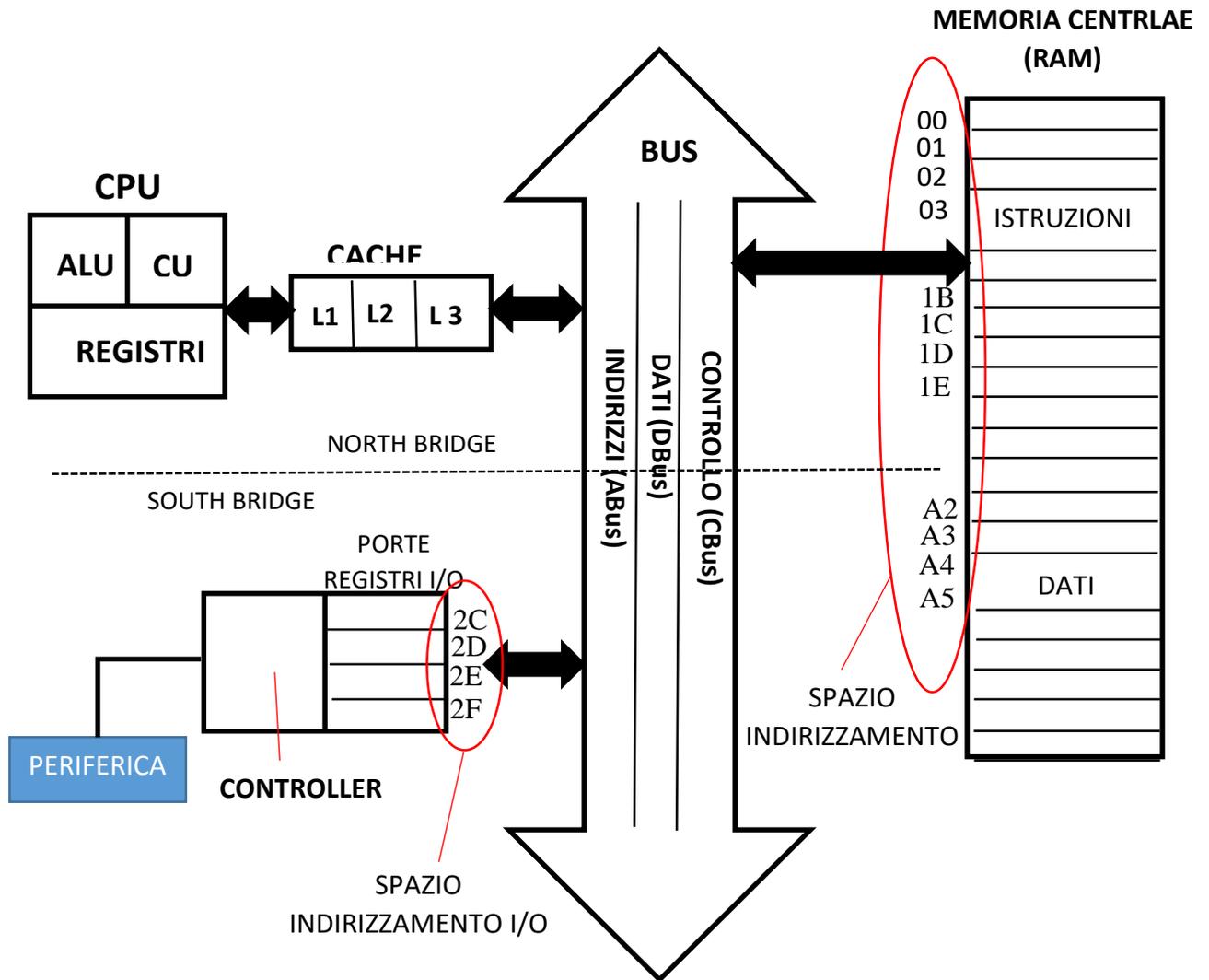
Si consiglia quindi, durante lo studio, di integrare queste dispense con i propri appunti.

Per ogni dubbio, chiarimento, o segnalazione contattatemi via mail scrivendo a

[francesco.capezi@gmail.com](mailto:francesco.capezi@gmail.com)

Francesco Capezio

# ARCHITETTURA DI UN CALCOLATORE



Lo schema in Figura1 illustra l'architettura interna di base di un calcolatore.

**BUS:** è il canale di comunicazione tra tutti i componenti installati sulla scheda madre.

**Northbridge:** è la parte del bus che collega CPU e RAM. E' la parte più veloce

**Southbridge:** la parte del bus che collega CPU e Periferiche, tramite i registri di Input/Output (detti "porte")

**BUS degli indirizzi (ABUS)** questo bus serve per identificare l'indirizzo sul quale si vuole operare. Questo indirizzo può essere una cella della RAM o una porta dello spazio di indirizzamento I/O.

Nella esempio semplificato che mostra la RAM come un armadio pieno di cassette sul registro ABUS viene scritto il numero del cassetto (il suo "indirizzo", scritto sull'esterno)

**BUS dei dati (DBUS)** questo bus serve per trasferire il contenuto della cella della RAM (o del registro-porta I/O) identificato con l'indirizzo scritto su ABUS. Il contenuto potrebbe essere una istruzione o un dato. Nell'esempio dell'armadio con i cassette con il DBUS si preleva il contenuto del cassetto.

**BUS di controllo (CBUS).** E' composto da una serie di **segnali** utili per controllare l'operato del sistema. I più noti sono

- **I/O – MEM:** il segnale a 0 indica un'operazione nello spazio di indirizzamento della RAM; a 1 indica un'operazione di I/O
- **R/W** il segnale a 0 indica un'operazione di scrittura; a 1 di lettura
- **WAIT** il segnale è a 0 durante l'esecuzione (devo aspettare) va a 1 quando l'operazione è terminata
- **INTR** è il segnale di ricezione di un interrupt da parte di una periferica

**INTA** è il segnale che l'interruzione da parte della periferica è terminata

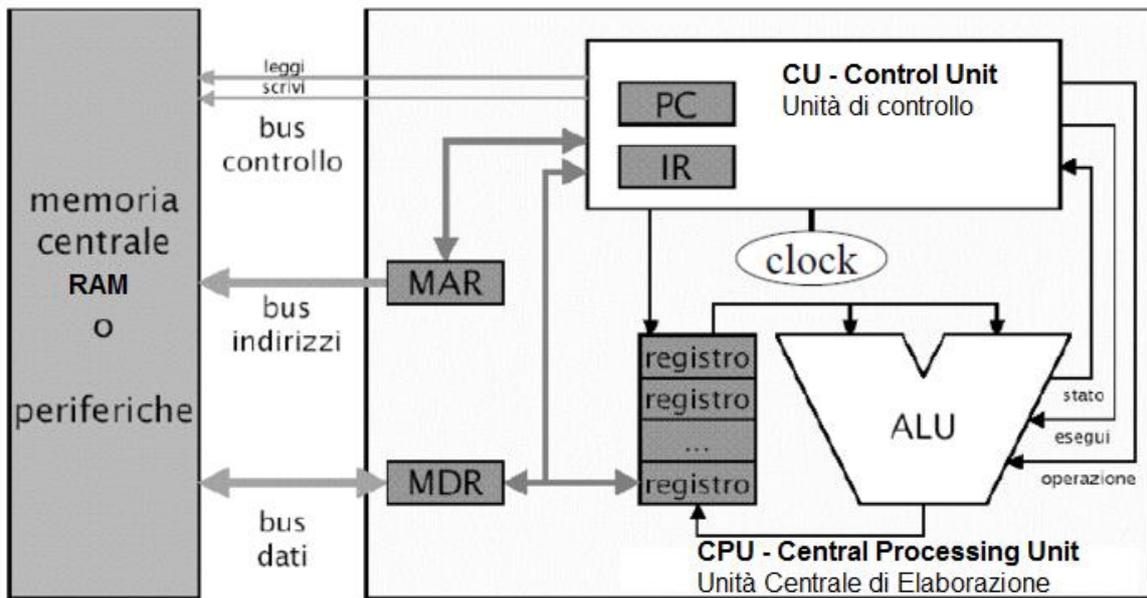
## **INPUT/OUTPUT**

Le periferiche sono gestite tramite un **controller**. La CPU accede alle periferiche utilizzando dei registri particolari dette **porte**. Questi registri operano come la RAM: hanno degli indirizzi (spazio di indirizzamento I/O) e la CPU può leggere dati (periferiche di Input) e scrivere (periferiche di output).

Quando una periferiche *di input* vuole inviare dei dati (il mouse quando si muove, la stampante etc.) manda un **interrupt**, ovvero utilizza un segnale del CBUS. La CPU interrompe l'esecuzione di un programma per iniziare una **Interrupt Service Routine (ISR)** allo scopo di leggere o scrivere sulle porte della periferica in questione.

Per velocizzare questo processo (ed evitare l'interruzione del programma da parte della CPU) è stato introdotto per molte periferiche di **Direct Memory Access (DMA)** ovvero la possibilità per il controller di leggere direttamente in memoria RAM, oppure di trasferire blocchi di dati da una periferica ad un'altra, senza interrompere l'operato della CPU con gli interrupt.

## CPU E REGISTRI



La Figura1 mostra chiaramente la composizione della **CPU** e il ruolo dei vari componenti:

**ALU (Aritmetich Logic Unit)** è la parte della CPU che esegue le operazioni matematiche e logiche, utilizzando un insieme di registri di supporto.

**CU (Control Unit)** E' la parte delle CPU che gestisce le operazioni, ovvero l'utilizzo dei registri per l'esecuzione delle istruzioni contenute nella RAM (ciclo di Fetch/Decode/Execute)

**REGISTRI:** ognuno ha una sua specifica funzione:

**PC (Program Counter)** Contiene l'indirizzo (della RAM) della prossima istruzione da eseguire

**IR (Instruction Register)** Una volta prelevata (finita la fase di Fetch) conterrà il **codice macchina dell'istruzione**

**MAR (Memory Address Register)** contiene l'indirizzo sul quale voglio operare. L'operazione può essere la lettura dell'istruzione, la lettura/scrittura di un dato sulla ram o l'utilizzo dello spazio di indirizzamento degli Input/Output. L'indirizzo viene utilizzato tramite il **ABus (Address Bus)**

**MDR (Memory Data Register)** contiene il **dato** sul quale si vuole operare. Il dato può essere stato letto dalla RAM o da una periferica (attraverso la relativa **porta**) oppure lo si vuole scrivere. Il dato verrà trasferito attraverso il **DBus (Data Bus)**

**PSW (Process Status Word)** E' un registro che contiene alcuni **flag** ovvero bit posti a zero o uno a seconda del loro significato. I flag possono segnalare

- se un'istruzione o un'operazione è giusta a termine
- errore di **overflow** dei calcoli (cifre troppo grandi)
- errore di **divisione per zero**
- ricezione di un interrupt dalla CPU

## Ciclo di Fetch/Decode/Execute

È il ciclo che serve alla CPU per **eseguire le istruzioni di un programma caricato nella RAM.**

Tutto il software che serve a far funzionare un calcolatore è salvato in una memoria di massa (esempio HDD o SSD). Il primo software ad essere caricato sulla RAM per essere eseguito è il **Sistema Operativo (SO)**, identificato dal **BIOS** durante la fase di **bootstrap**.

Dunque il S.O. è il primo programma che viene eseguito dalla CPU tramite il ciclo di Fetch/Decode/Execute.

Ogni volta che l'utente manda in esecuzione un programma (ad esempio facendo doppio-click con il mouse sull'icona) le istruzioni del programma in questione vengono caricate sulla RAM, pronte per l'esecuzione.

Le istruzioni di un programma sono scritte in **linguaggio macchina**, ovvero in codice binario (rappresentato in esadecimale) e scritte nelle celle della ram.

### Fase di Fetch

Nel registro PC è contenuto l'indirizzo dell'operazione da eseguire. La CU copia questo indirizzo sul MAR per prelevare tramite il ABUS.

L'istruzione (contenuta nella cella della RAM all'indirizzo prestabilito) viene copiata nel IR tramite il DBUS.

Viene incrementato di uno il PC.

### Fase di Decode

La CU decodifica il linguaggio macchina dell'istruzione contenuta nel IR. Vengono preparati i dati necessari all'operazione attraverso ABUS e DBUS. I dati vengono copiati su registri di supporto.

### Fase di Execute

Viene eseguita l'istruzione. Se è un'istruzione di calcolo interviene la ALU che processa i dati salvati sui suoi registri. L'istruzione potrebbe invece essere un aggiornamento di un indirizzo o il salvataggio di un dato.

Il ciclo riprende dalla fase di Fetch.