

## GLI HARDWARE E LA LOGICA DI FUNZIONAMENTO DI UN CALCOLATORE by styx^

- 1.0 PREMESSA
- 2.0 LA MEMORIA
  - 2.1 TIPO DI ACCESSO
    - 2.1.0 ACCESSO CASUALE
    - 2.1.1 ACCESSO SEQUENZIALE
    - 2.1.2 ACCESSO SEMI-CASUALE
  - 2.2 FUNZIONE SVOLTA
    - 2.2.0 REGISTRI
    - 2.2.1 MEMORIA CENTRALE
    - 2.2.2 MEMORIA DI MASSA
- 3.0 ESEMPI DI MEMORIA INTERNA
  - 3.1 REGISTRI
    - 3.1.0 PARALLELL REGISTER
    - 3.1.1 SHIFT REGISTER
  - 3.2 RAM
  - 3.3 ROM
- 4.0 MEMORIA DI MASSA
  - 4.1 REGISTRAZIONE SU NASTRO
  - 4.2 REGISTRAZIONE SU DISCO MAGNETICO
    - 4.2.0 TESTINE FISSE
    - 4.2.1 TESTINE MOBILI
  - 4.3 REGISTRAZIONE SU MEMORIE OTTICHE
- 5.0 BUS
- 6.0 ALU - ARITHMETIC LOGIC UNIT
- 7.0 REGISTRI
  - 7.1 REGISTRI GENERALI
  - 7.2 REGISTRI DI ISTRUZIONI
  - 7.3 REGISTRO CONTATORE DI PROGRAMMA
  - 7.4 REGISTRO STACK POINTER
  - 7.5 REGISTRO DI STATO
- 8.0 LA CPU
  - 8.1 LA LOGICA CABLATA
  - 8.2 LA LOGICA MICROPROGRAMMATA
- 9.0 DISPOSITIVI/INPUT/OUTPUT
  - 9.1 DISPOSITIVI DI INPUT

- 9.1.0 LETTORI DI CARATTERI MAGNETICI
  - 9.1.1 SENSORI
  - 9.1.2 DISPOSITIVI PER L'ACQUISIZIONE DELLA VOCE
  - 9.1.3 MODEM
  - 9.2 DISPOSITIVI DI OUTPUT
    - 9.2.0 VIDEO
    - 9.2.1 STAMPANTI
    - 9.2.2 ATTUATORI
- 10.0 FINALMENTE TUTTO IL CALCOLATORE
- 11.0 FINE?
- 11.1 RINGRAZIAMENTI

## 1.0 PREMESSA

Salve a tutti sono styx^:questa volta vi parlerò da cosa è costituito l'hardware di un calcolatore,come funziona e come interagisce con il software. E' un argomento che non a tutti interessa,ma è molto importante conoscerlo per capire a fondo come è fatta quella scatola metallica che avete vicino ai piedi. Spero che sia di vostro gradimento. Buona lettura! :)

## 2.0 LA MEMORIA

La memoria,come sappiamo tutti,è un componente indispensabile dell'hardware,che permette di aiutare la CPU nei suoi calcoli e permette di:

- immagazzinare dati
- conservare dati
- restituire dati

Le memorie si differenziano tra di loro a seconda di:

- tipo di accesso
- funzione svolta
- il tempo in cui vengono conservate le informazioni

### 2.1 TIPO DI ACCESSO

La memoria in base al tipo di accesso si suddivide ulteriormente in tre tipi di accesso:

- accesso casuale
- accesso sequenziale
- accesso semi-casuale

#### 2.1.0 ACCESSO CASUALE

L'accesso casuale è l'accesso in cui le informazioni vengono,appunto,cercate in modo casuale. Il tempo necessario per trovare le informazioni in questo modo è indipendente dalla posizione occupata dalla stessa informazione all'interno della memoria.

#### 2.1.1 ACCESSO SEQUENZIALE

In questo tipo di accesso,le informazioni vengono cercate nello stesso modo in cui erano state memorizzate. In questo caso il tempo necessario per trovare l'informazione è dipendente dalla

posizione occupata dalla stessa informazione all'interno della memoria.

### 2.1.2 ACCESSO SEMI-CASUALE

Questo tipo di accesso cerca un'area, in modo casuale, in cui "crede" vi sia l'informazione, per poi cercare al suo interno l'informazione in modo sequenziale. In questo tipo di accesso il tempo necessario per trovare l'informazione varia a seconda dalla posizione dell'informazione. Inoltre deve anche fare una ricerca sequenziale, quindi la velocità di accesso è molto più alta rispetto alle altre.

## 2.2 FUNZIONE SVOLTA

In base alla funzione svolta esse si suddividono:

- registri
- memoria centrale
- memoria di massa

### 2.2.0 REGISTRI

I registri vengono utilizzati per l'elaborazione interna delle CPU (infatti se conoscete l'assembler sapete già della loro esistenza). Questa memoria ha un accesso casuale, è di dimensioni molto ridotte e l'accesso è molto veloce, superiore a tutte le altre.

#### 2.2.1 MEMORIA CENTRALE

Viene utilizzata per la memorizzazione di dati, file, programmi e quant'altro. Il tipo di accesso è casuale, ma è di dimensioni molto grandi (ovviamente :)

#### 2.2.2 MEMORIE DI MASSA

Sono memorie ad accesso sequenziale o semi-casuale e servono per conservare in modo permanente le informazioni. I tempi di accesso, proprio per questa loro funzione, è molto lungo.

## 3.0 ESEMPI DI MEMORIE INTERNE

A questo punto vi parlerò in maniera più approfondita di alcune memorie (registri, ram, rom) che, secondo me, sono le più importanti.

### 3.1 REGISTRI

Come abbiamo detto i registri servono per le operazioni all'interno delle CPU. I tempi di accesso sono molto veloci, ma ha scapito della loro grandezza: infatti posso memorizzare solo pochi byte, che vanno da 1 a 8. Le CPU moderne supportano, per esempio, registri a 32 bit (es. `eax`, `ebx`, `ecx`, `edx` che sono registri General Purpose). Per il loro tipo di memorizzazione i registri si suddividono in:

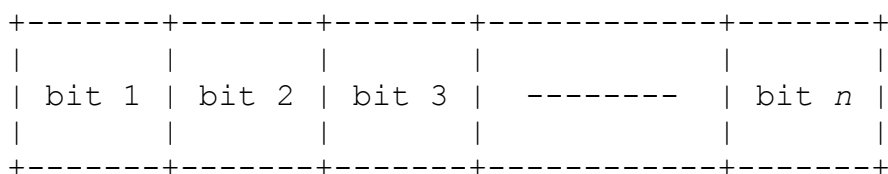
- parallel register (registro in parallelo)
- shift register (registro a scorrimento)

### 3.1.0 PARALLEL REGISTER

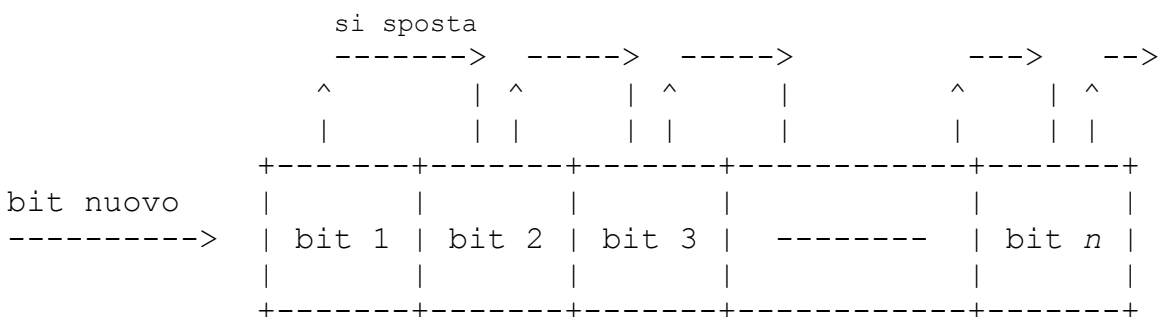
I parallel register memorizzano le informazioni memorizzando contemporaneamente tutti i bit del dato. Questo avviene in due fasi: prima vengono messi a 0 tutti i bit presenti, poi vengono portati ad 1 quelli che devono assumere quel valore.

### 3.1.1 SHIFT REGISTER

I registri di scorrimento, ovviamente, memorizzano i dati in maniera differente: i bit che compongono il dato vengono memorizzati in modo sequenziale. Praticamente ad ogni impulso del clock, l'*n*-esimo bistabile viene spostato nel bistabile *n+1*. Nel primo bistabile viene registrato il nuovo bit del nuovo dato. Vediamo come:

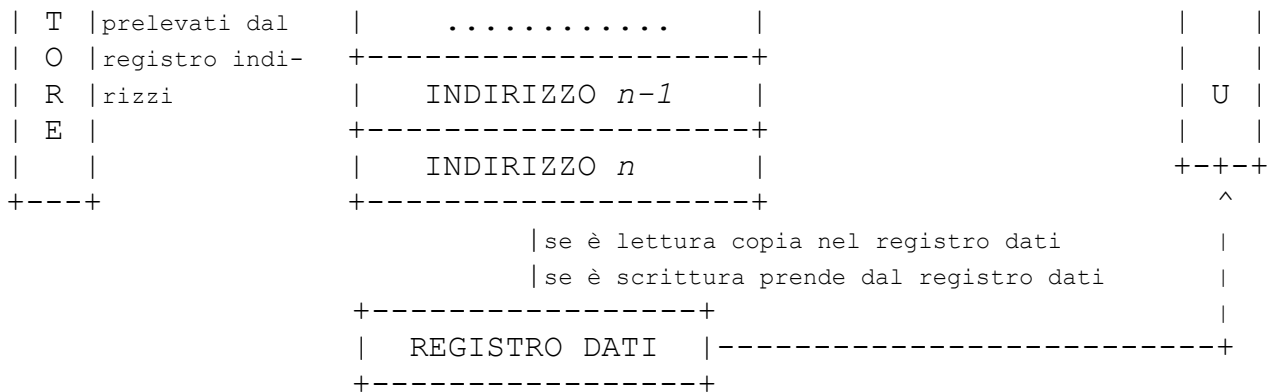


Impulso di clock:



Arriva il "nuovo bit", che prende il posto di "bit 1", che a sua volta prende il posto di "bit 2", che prende il posto di "bit 3", etc.





Facile, no?

Le prime RAM erano dette "RAM statiche" in quanto i dati rimanevano memorizzati in modo inalterato finché non si fosse scritto un altro dato oppure la tensione dell'alimentazione scendeva sotto un livello minimo. Queste prime RAM potevano solamente gestire una bassa memorizzazione, quindi ora non sono più utilizzate. Ora le RAM sono dette "RAM dinamiche", possono gestire un'elevata memorizzazione e che operano grazie ad un condensatore. Poiché questo dopo un periodo di tempo si scarica, bisogna ricaricarlo tramite un circuito, detto "circuito di rinfresco (refresh)", affinché l'alimentazione rimanga abbastanza costante e non scenda sotto la soglia minima.

E' importante dire che la memoria delle RAM (dinamiche o statiche), è memoria volatile, cioè perde le informazioni se per esempio si spegne il pc e non hanno quindi più l'alimentazione necessaria. Per provvedere a ciò sono state create le ROM (Read Only Memory).

### 3.3 ROM (Read Only Memory)

Le ROM (Read Only Memory) sono memorie ad accesso casuale, con bassa capacità di memorizzazioni, ma memorizzano dati e informazioni in modo permanente. Esse non sono modificabili dall'utente o dallo spegnimento del pc e possono essere solamente lette. Esse fanno parte della memoria centrale, fanno partire tutte quelle funzioni indipendenti dall'utente (per esempio funzioni di inizializzazione, come il 'bios') e non devono essere mai modificate durante l'esistenza di un calcolatore. In seguito sono state introdotte le PROM (Programmable Read Only Memory) in modo da poter essere programmate una sola volta, durante la quale sono inserite tutte le informazioni, per non essere modificate mai più.

### 4.0 MEMORIE DI MASSA

Come abbiamo detto, le memorie descritte sopra sono memorie volatili. Come fare dunque per memorizzare dati in modo

permanente? Semplice, si utilizzano le memorie di massa. Queste sono quelle memorie che memorizzano un elevato numero di dati e in modo permanente. Esse non permettono però al processore di elaborare direttamente i loro dati e hanno quindi bisogno di dispositivi di input/output per importare/esportare informazioni per essere elaborate. La maggior parte delle memorie di massa sono di tipo 'magnetizzabile', cioè i dati vengono letti/scritti su materiali magnetici da una testina. In questo modo si è guadagnata molta più capacità di memorizzazione (rispetto alle memorizzazioni per bit) a scapito della velocità, a causa dei movimenti lenti degli strumenti meccanici, quali testine e supporti. Quindi le memorie di massa si suddividono per il loro tipo di registrazione magnetica, che può essere su nastro o su disco.

## 4.1 REGISTRAZIONE SU NASTRO

Questo tipo di registrazioni vengono fatte su un nastro, appunto, di poliestere ricoperto di ossido di ferro, lungo molti metri, attorcigliato su una bobina.

Questo nastro è suddiviso in "piste longitudinali" su cui i bit vengono scritti in colonne trasversali ad esse. I dati sono posti in blocchi, grandi a seconda del tipo di memoria centrale e dal calcolatore, intervallati da zone che non vengono utilizzate, chiamate "gap". Vediamo ora come è fatto questo nastro magnetico:

```

---+---+-----+---+-----+-----
--|+++| * <-- bit-|+++|-*-*-*-*-*|-----
--|+++|-*-----|+++|-*-*-*-*-*|----- <-- piste longitudinali
--|+++|-*-----|+++|-*-*-*-*-*|-----
--|+++|-* <-- bit-|+++|-*-*-*-*-*|-----
--|+++|-*-----|+++|-*-*-*-*-*|-----
--|+++|-*-----|+++|-*-*-*-*-*|-----
---+---+-----+---+-----+-----
    |gap|  BLOCCO  |gap|  BLOCCO  |
    +---+-----+---+-----+-----
    
```

Come vedete i bit sono scritti in colonne (ogni bit è un '\*') all'interno del blocco. Le informazioni contenute nel blocco sono prese tutte insieme e vengono "trasportate" alla memoria centrale che le elabora. E' necessario dunque che nella memoria centrale ci sia un'area che riesca a mantenere tutte le informazioni del blocco che riceve: il buffer.

Prima abbiamo detto che il nastro viene scritto/letto da una testina: in questo modo vuol dire che normalmente il nastro è fermo, poi al momento della lettura/scrittura questo si mette in "moto". Dunque possiamo dividere tutto ciò in 3 fasi:

- accelerazione: per portare il nastro a velocità corretta







Non sono riuscito a fare un cerchio decente, quindi pensate a questo quadrato come un cerchio e le linee all'interno come concentriche. Come al solito i gap sono quelle aree riempite con gli `\*`, i settori sono quelle aree con scritto `settore`, mentre le tracce sono quelle colonne di `|`, che teoricamente sono messe in modo concentrico. I bit vengono scritti in modo sequenziale sulla stessa traccia anziché essere scritti modo trasversale ad esse. Dunque, i dischi si suddividono in 2 gruppi:

- testine fisse
- testine mobili

#### 4.2.0 TESTINE FISSE

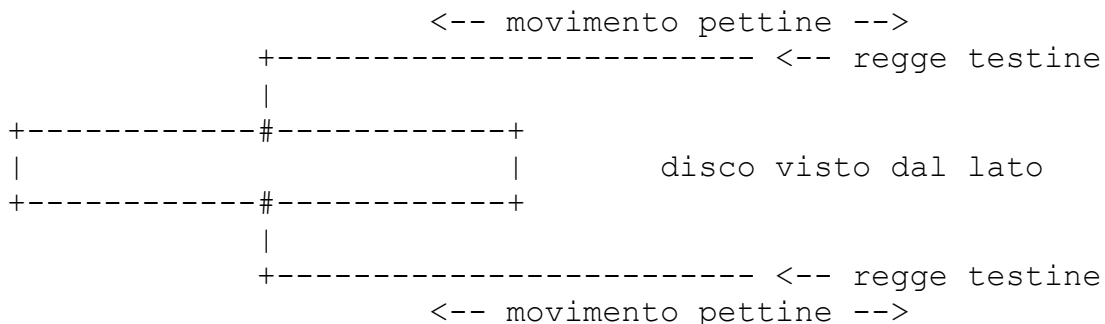
In questo modo le testine sono disposte in maniera rigida sopra la superficie del disco: ad ogni testina corrisponde ad una traccia. Vediamo un po' come dovrebbe essere fatto:



Le testine sono rappresentate dall' `#` e, come vedete, ognuna di queste corrisponde ad una traccia concentrica.

#### 4.2.1 TESTINE MOBILI

In quest'altro modo le testine sono mobili e si muovono radialmente sul disco, facendo in modo che ci sia una sola testina per ogni faccia. Per esempio:



Il pettine si muove avanti e indietro, facendo scrivere/leggere ad una sola testina ('#') più tracce. Questo tipo di testine sono più economiche, ma a scapito di una velocità di lettura/scrittura molto più bassa: va sommato il tempo necessario per cercare una traccia e quello necessario per posizionarvi sopra la testina. Comunque questo tipo di testine vengono utilizzate ampiamente, per esempio gli stessi hard disk, o anche detti 'dischi di Winchester', sono a testina mobile. Negli hard disk un disco magnetico di dimensioni ridotte, ma con grande possibilità di memorizzazione, è inserito con la testina all'interno di un contenitore metallico sigillato e a prova di qualsiasi materiale "disturbante", come per esempio la polvere. La testina non è attaccata al disco, ma posizionata ad una distanza minimale (pochi micron), che permetta al disco di ruotare a velocità altissime, evitando crash. Inoltre, per accelerare i tempi di lettura/scrittura, si "allega" al disco una memoria elettronica, cache memory, nella quale sono memorizzate le informazioni che vengono richieste frequentemente, riducendo sostanzialmente gli accessi al disco. Un altro esempio di unità con testine mobili sono i floppy disc, che funzionano più o meno nello stesso modo.

### 4.3 REGISTRAZIONE SU MEMORIE OTTICHE

Questo tipo di registrazioni si effettuano solitamente su dischi di metallo, sensibili ad un raggio laser. Durante la fase di scrittura, il laser viene modificato a seconda del dato binario da registrare fa dei piccoli fori sulla superficie del disco in corrispondenza dei bit a valore 1, registrandoli su una spirale continua. Nella fase di lettura avviene il contrario: viene utilizzato un raggio di potenza molto inferiore che legge la presenza o meno di fori, mandando l'energia riflessa ad un fotodiode che ne converte la maggiore o minore intensità in codice binario.

Fino a qualche anno fa i dischi ottici avevano come lato negativo il fatto di poter essere scritti solamente una volta, anche se la

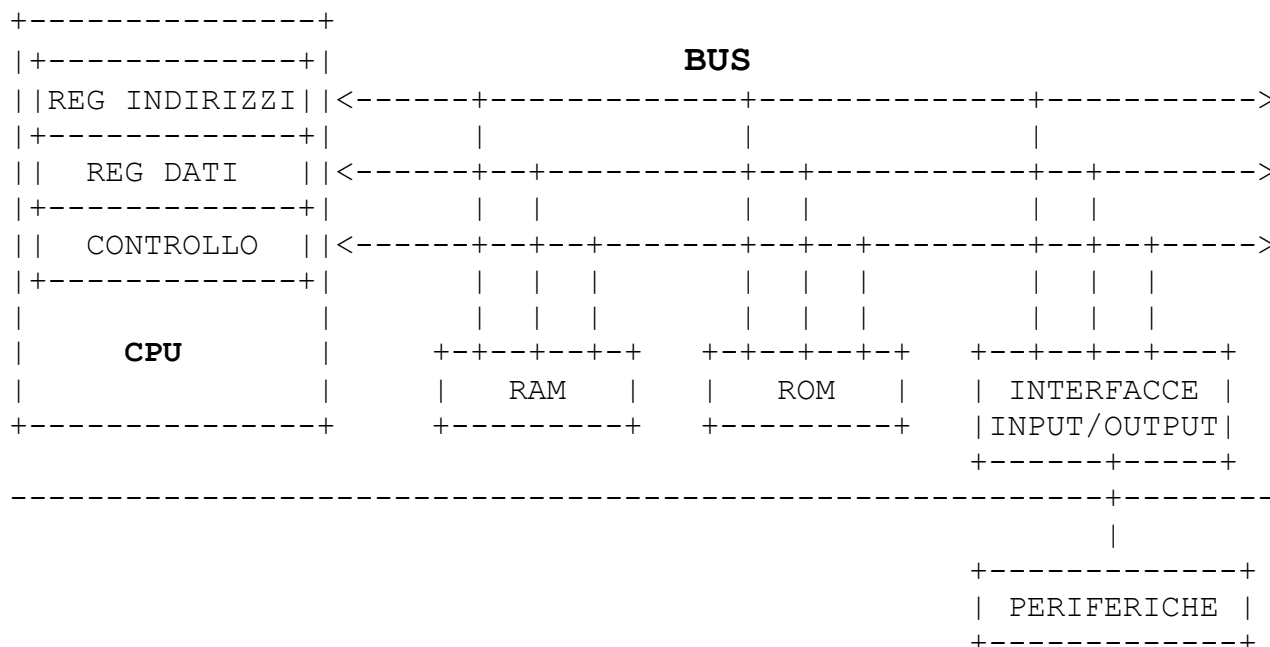
quantità di dati che si potevano scrivere era molto grande (es. Compact Disc Read Only Memory - CDRom). Ora però questo problema non c'è più in quanto, come sanno tutti, sono nati anche dischi ottici che possono essere scritti più di una volta (RW-CDROM) e capaci di immagazzinare molti più dati (DVD-ROM).

## 5.0 BUS

Il fatto di implementare realmente un grandissimo numero di collegamenti tra le varie componenti di un calcolatore creerebbe numerosi problemi, dunque si è ricorso al bus, che collega tutte le unità di un calcolatore a un sistema di trasmissione comune. Esso si suddivide in:

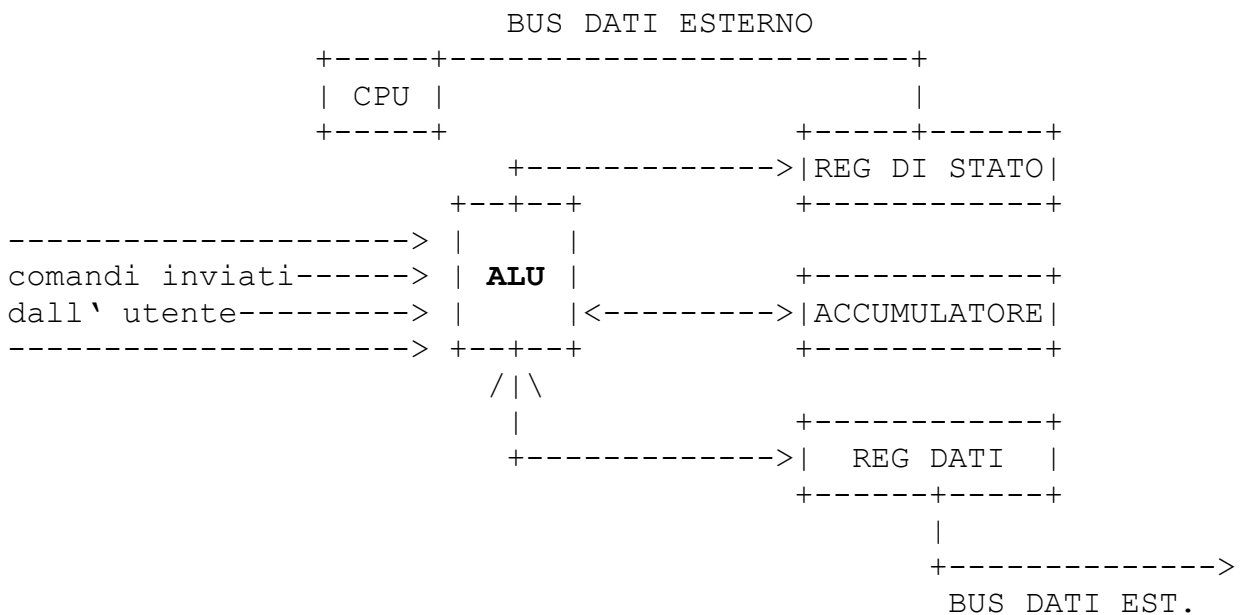
- bus dati: un insieme di linee che trasmettono dati tra le varie unità.
- bus indirizzi: insieme di linee che trasmettono indirizzi, con i quali il processore seleziona l'unità da utilizzare.
- bus di controllo: insieme di linee che trasmettono segnali di controllo

Allora bisogna dunque aggiungere che il bus consente una sola comunicazione per volta; una sola unità può trasmettere, mentre più di una può ricevere; i trasferimenti sono guidati dall'unità di controllo che decide quale unità debba utilizzare il bus. Vediamo un po' come dovrebbe essere:



## 6.0 ALU - ARITHMETIC LOGIC UNIT

L'unità aritmetico logica serve per svolgere le principali funzioni logiche, nonché funzioni aritmetiche, funzioni di confronto e scorrimento a destra/sinistra di bit. Quando un utente intende fare un'operazione (es. una somma), invia dei comandi all' ALU, che specificano che bisogna effettuare una somma, la quale prende gli operandi all'interno o dell'accumulatore oppure all'interno del registro dati. Il risultato viene riposto sempre all'interno dell'accumulatore e nello stesso tempo modifica dei bit (flag di condizione) all'interno del registro di stato, che andranno a informare la CPU dell'operazione appena effettuata. Vediamo, come al solito, un bello schemino:



## 7.0 I REGISTRI

Prima abbiamo detto che la CPU è dotata di una propria memoria, capace di fare calcoli e suddivisa in vari registri: ora andremo ad esaminare proprio questi registri e le loro funzioni logiche.

### 7.1 REGISTRI GENERALI

Questi registri sono importanti in quanto al loro interno vi sono gli operandi e i risultati intermedi di una operazione. La loro ampiezza è di un word. È in particolare importante il registro accumulatore, nel quale vengono depositati risultati di operazioni, che permette di comunicare (lettura/scrittura) con le porte input/output.

### 7.2 REGISTRO DI ISTRUZIONI

Questo registro contiene l'istruzione necessaria per l'esecuzione di un programma, letta dalla memoria, finché quest'ultimo è svolto. Questo è diviso in due parti:

- una parte dove è contenuto il codice operativo, la quale è collegata con il bus interno dati e con il decodificatore, che è collegato a sua volta con l'unità di controllo.
- l'altra parte contiene gli indirizzi degli operandi ed è collegato con il bus interno degli indirizzi.

### 7.3 REGISTRO CONTATORE DI PROGRAMMA

La CPU quando deve eseguire un programma ha bisogno di reperire continuamente istruzioni da eseguire all'interno della memoria. Questo è ciò che fa questo registro ("program counter"). Le istruzioni sono immagazzinate una dopo l'altra, dunque il registro deve incrementarsi da solo, facendo in modo che una volta finita l'istruzione passi a quella seguente. Questo auto-incremento avviene dopo che l'istruzione è stata caricata nel registro istruzioni, in modo che al momento dell'esecuzione contenga già l'indirizzo della prossima istruzione da eseguire.

### 7.4 REGISTRO STACK POINTER

Solitamente quando c'è bisogno di chiamare un sottoprogramma e fare in modo che lo stato del processore e dei registri siano mantenuti uguali, si utilizza il registro di stack pointer. Qui i dati vengono immagazzinati a "pila", cioè una sopra l'altro, e il primo ad essere immagazzinato è il primo a essere preso (FIFO: First In, First Out). Più o meno è così:

```
+-----+
|  DATO 1  | CIMA DELLO STACK
+-----+
|  DATO 2  |
+-----+
|  .....  |
+-----+
|  DATO n  | BASE DELLO STACK
+-----+
```

### 7.5 REGISTRO DI STATO

Ne abbiamo parlato a proposito dell'ALU, dunque sappiamo che esso modifica dei bit (flag) che permettono alla CPU di conoscere le ultime condizioni verificatesi nell'ultima operazione. In particolare i flag possono essere:

- flag di carry: viene posto a 1 in caso di riporto in una operazione aritmetica, altrimenti è settato a 0.
- flag di overflow: viene posto a 1 se il risultato di un'operazione aritmetica oltrepassa la capacità di elaborazione della ALU.
- flag di zero: vale 1 se il risultato di una operazione è 0.
- flag di negatività: riproduce il bit del segno del risultato.

Questi flag sono conosciuti bene da chi conosca un minimo di assembler. Quel "chi" non sono io ;)

## 8.0 LA CPU

L'unità di controllo (CPU) è l'"organo" più importante di un calcolatore, in quanto ne costituisce il cuore: infatti egli si occupa di interpretare e decodificare le istruzioni che riceve, attivando gli altri "organi" del calcolatore, a ritmi dettati dal clock. Si può suddividere in due tipi:

- logica cablata
- logica microprogrammata

### 8.1 LA LOGICA CABLATA

Questo primo tipo è un insieme di circuiti logici sequenziali definiti dal costruttore, formato dall'ALU e dai registri. In questo tipo di CPU il vero problema è che le istruzioni e le operazioni possibili sono solamente quelle "fatte" dal costruttore e non possono essere modificate dall'utente: questo crea un grande problema in quanto se si volesse ingrandire il numero di queste istruzioni bisognerebbe ingrandire la rete logica di controllo.

### 8.2 LA LOGICA MICROPROGRAMMATA

A causa dei problemi creati dal primo tipo di CPU, si è deciso di crearne una nuova, basata sulla microprogrammazione. Con la microprogrammazione la serie di istruzioni emesse ad ogni clock possono essere considerate come comandi elementari di complessità minore rispetto al linguaggio macchina. Quindi la CPU "diventa" dunque un calcolatore a cui viene chiesto ad ogni clock l'esecuzione del microprogramma associato ad una particolare istruzione. All'interno di questa CPU si distinguono:

- memoria di controllo: contiene tutti i microprogrammi (ROM)
- contatore di microprogramma: uguale al program counter
- registro di microistruzione: dove è contenuta la microistruzione corrente





```
| clock |  
+-----+
```

## 9.1 DISPOSITIVI DI INPUT

Questi dispositivi servono per l'acquisizione di dati da parte del calcolatore e si suddividono in:

- tastiere alfanumeriche
- scanner
- lettori di caratteri magnetici od ottici
- sensori
- dispositivi per l'acquisizione della voce
- modem

Parlerò in dettaglio dei dispositivi meno famosi. :D

### 9.1.0 LETTORI DI CARATTERI MAGNETICI

Questo tipo di lettura è basata su caratteri alfanumerici che sono stampati su carta con un particolare inchiostro magnetico tramite un particolare dispositivo. Il lettore che legge questi caratteri è munito da una testina che legge questo inchiostro e associa ogni tratto magnetico della parola con un font interno, trasformandolo quindi nel codice binario corrispondente.

### 9.1.1 SENSORI

Questi dispositivi acquisiscono dall'esterno una particolare grandezza fisica (es. temperatura, velocità, etc.) trasformandola mediante un trasduttore in segnale elettrico e in seguito in binario tramite un convertitore analogico/digitale.

### 9.1.2 DISPOSITIVI PER L'ACQUISIZIONE DELLA VOCE

La difficoltà della creazione di questi dispositivi risiede nel fatto che i termini usabili nel linguaggio umano sono tantissimi ed inoltre da uomo a uomo varia la velocità nel parlare, il tono e il timbro della voce. Ci sono due tipi di tecniche per la sua creazione.

La prima: con questa tecnica si fanno prima una serie di prove per l'acquisizione della voce in modo che il calcolatore possa memorizzare gli aspetti più significativi di essa. In seguito il calcolatore può quindi confrontare le parole memorizzate con le parole immesse dall'utente per riuscire a risalire alla parola vocalizzata.

La seconda: questa seconda tecnica si basa sull'intelligenza artificiale di un calcolatore, poiché gli si "chiede" al calcolatore di riconoscere le parole, le regole

grammaticali, logiche, etc. presenti in un discorso. Inoltre il calcolatore deve quindi ricostruire un discorso a partire da alcune parole contenute in esso.

### 9.1.3 MODEM

Tutti sanno che internet "unisce" la linea telefonica al nostro calcolatore: ma il fatto è che i segnali che costituiscono il primo sono differenti da quelli che costituiscono il secondo. Infatti la linea telefonica utilizza i segnali analogici, mentre il calcolatore segnali digitali. Dunque il modem, diminutivo di modulatore/demodulatore, non fa altro che convertire i segnali digitali in quelli analogici (modulazione) e quelli analogici in digitale (demodulazione). Capit? Bene.

## 9.2 DISPOSITIVI DI OUTPUT

Questo tipo di dispositivi servono per tradurre in una lingua comprensibile per l'uomo i risultati delle operazioni di un calcolatore e mostrarle. Abbiamo diversi dispositivi:

- video
- stampanti
- attuatori

### 9.2.0 VIDEO

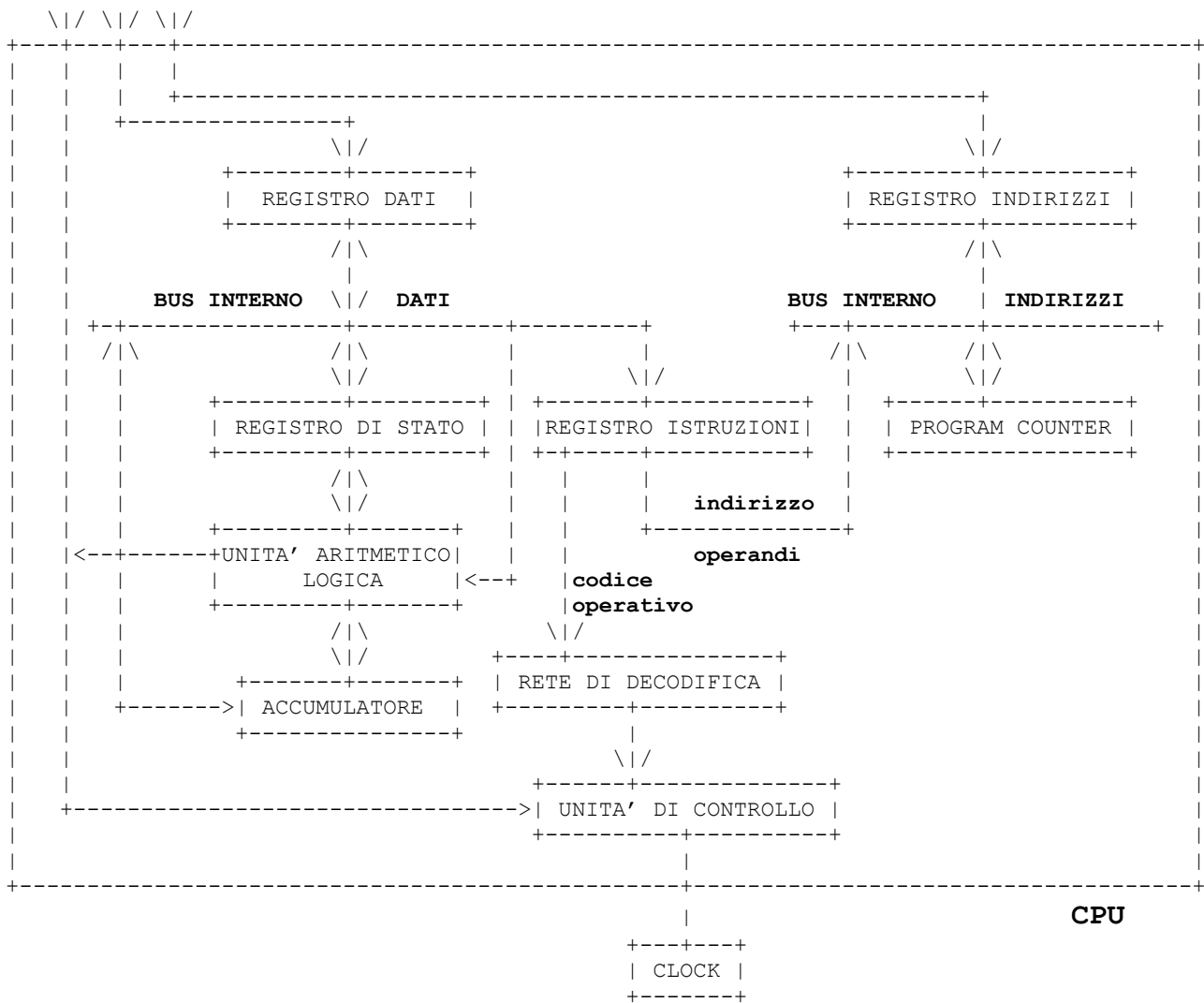
I video sono formati da uno schermo che mostra i codici binari elaborati dal calcolatore in immagini o testi. Ci sono schermi catodici o al plasma. I schermi con tubo catodico hanno appunto questo tipo di tubo in cui i fasci elettronici effettuano scansioni orizzontali dello schermo. La maggiore intensità di questo fascio in un punto porta alla creazione di un pixel: un'immagine è data dalla loro "aggregazione". Ci sono, come ho detto, anche gli schermi al plasma, caratterizzati da un fascio più fitto in cui il dettaglio dell'immagine è maggiore, ma ha scapito della velocità di aggiornamento.

### 9.2.1 STAMPANTI

Le stampanti "stampano", appunto, i risultati dell'elaborazione su carta. Esse si suddividono in base al modo di stampa in 4 tipi:

- a impatto
- termiche
- a getto
- al laser





Capito? Bene se siete arrivati fino qui vuol dire di sì! :)

## 11.0 FINE ?

Bene, se siete arrivati fino a leggere qui vuol dire che l'articolo vi è piaciuto. Questo è solo l'inizio di ciò che dovete conoscere affinché sappiate cosa state facendo e come funziona un pc (calcolatore). Perdonatemi se ho scritto fesserie oppure in un italiano da "ignorante"! :D

Se avete domande mi trovate su [irc.azzurra.org](http://irc.azzurra.org) sui canali:

- #hacklab
- #asm
- #ondaquadra
- #spine
- #C
- #hack4freedom

Oppure potete mandarmi un'e-mail su [the.styx@gmail.com](mailto:the.styx@gmail.com)

## 11.1 RINGRAZIAMENTI

Saluto tutti quelli di hacklab ([www.hacklab.tk](http://www.hacklab.tk)) e tutti quelli che mi aiutano continuamente. Saluto OverIP, mydecay, Traktopel, Tiger87, Izzy, fritzzu, eazy, d4n e tutti quelli che mi sono scordato ;) Ciao a tutti, alla prossima! :\*

styx^

FINITO DI SCRIVERE IL: 12/10/2004

SOURCE:

"I sistemi", Garavaglia, Petracchi ©  
[www.google.it](http://www.google.it)