Sbobine di

ELETTRONICA DIGITALE

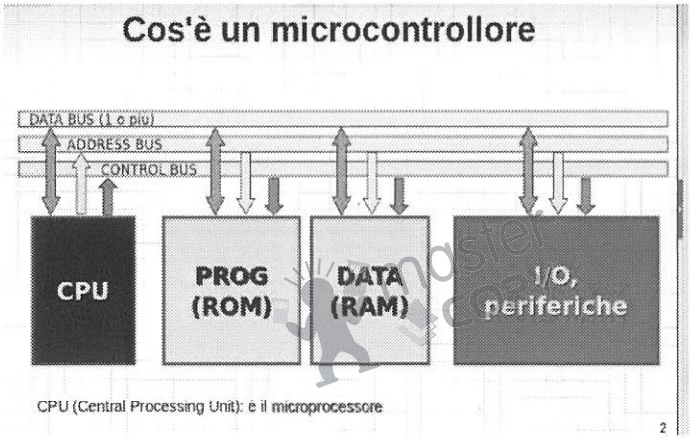
2020/2021

Corso di Ingegneria biomedica, prof. Roncella.

A cura di Federica & Agnese

Lezione 1

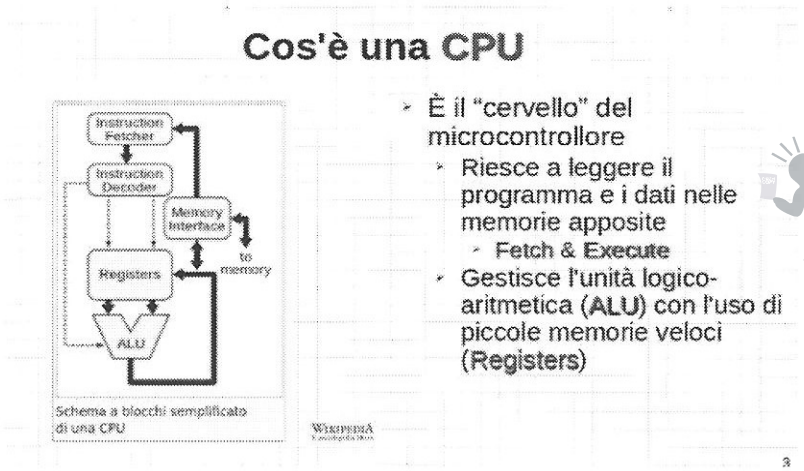
MICROCONTROLLORE



Il microcontrollore è un oggetto composto da vari sotto blocchi, si riconosce il blocco blu scuro caratterizzato dalla CPU (central processing Unit=microprocessore) che è il vero oggetto digitale che svolge il programma, all'interno quindi di un microcontrollore c'è un microprocessore.

La parola stessa controllore indica la funzione che questi circuiti hanno nel nostro mondo, sono circuiti che nascono insieme ad un oggetto fisico un insieme di cose circostanti di cui rappresentano l'intelligenza e le capacità di interazione con il mondo reale. Esempi biomedici: la pompa di infusione di insulina che contengono un microcontrollore, il pacemaker è un microcontrollore con gli elettrodi e un po' di sensing intorno, tutta la platea degli indossabili e degli impiantabili, glucosimetro. Altri esempi: nei forni c'è il microcontrollore, nel frigorifero o nella lavatrice.

Il microcontrollore nasce pensando all'applicazione, per svolgere l'applicazione c'è bisogno dell'intelligenza che esegue il programma, il microprocessore. Il programma è un insieme di istruzioni che vanno scritte su un contenitore, questo contenitore è la program memory (ROM), poi c'è un contenitore dove vengono messi i dati da elaborare (RAM), e poi ha una serie di circuiti specializzati che si chiamano periferiche che servono a sollevare la parte principale del circuito da compiti molto specifici.

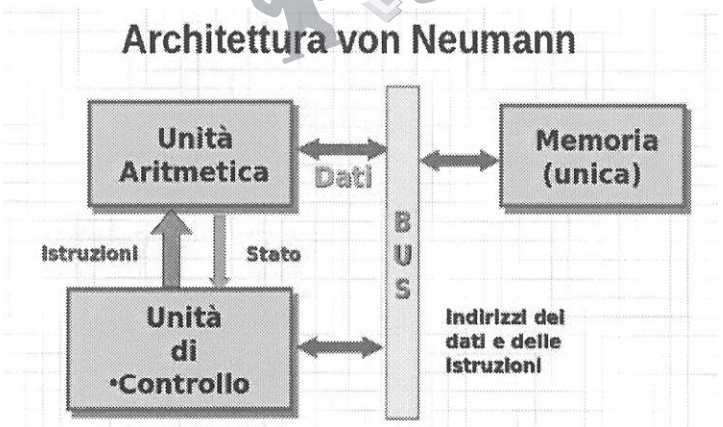


Come fa un sistema elettronico a eseguire un programma?

Immaginiamo di aver progettato un circuito digitale che sa fare solo due cose, sa leggere una memoria, una riga alla volta, interpreta la riga che ha letto, nella riga c'è scritto un programma da eseguire con un codice particolare e sulla base del valore letto esegue il comando previsto facendo ricorso a una parte operativa che è l'unità logica aritmetica (ALU), i dati nel processore sono normalmente contenuti in piccole memorie veloci che si chiamano registri. Quindi

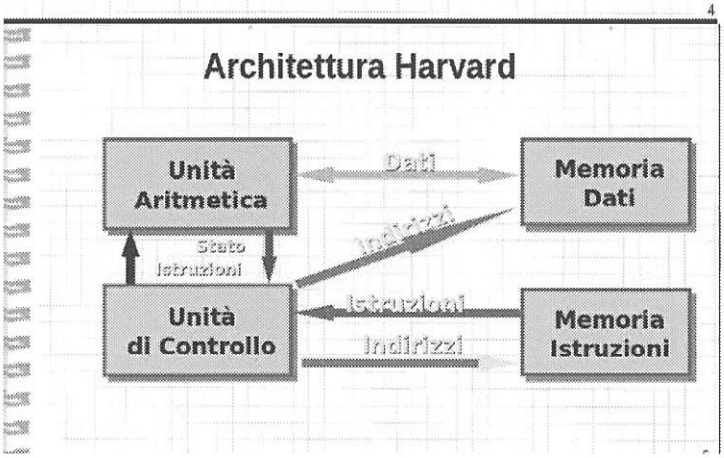
c'è l'instruction Fetcher che è un circuito digitale che prende il comando dalla memoria, il comando (scritto in bit) viene decodificato, il comando per esempio dice che è una somma e dice anche cosa bisogna sommare (es registro 1 e registro 10), allora il decoder attiva registro 1 e attiva registro 10 e dice alla ALU di fare la somma, il risultato viene messo in un terzo registro, vedremo che uno dei due registri di partenza si usa anche per il risultato, viene sovrascritto.

A proposito di CPU introduciamo qualche piccola informazione che ci e permette di orientarci nel mondo dei processori, nello schema visto abbiamo visto che tra le informazioni necessarie al processore c'è il programma e i dati da elaborare, se tali informazioni sono in un unico contenitore, in un'unica memoria, si parla di **architettura di von Neumann.**



Per ridurre la complessità delle connessioni bisognava usare un unico contenitore diviso in due parti, da una parte c'era il programma, dall'altra parte c'erano i dati ma per leggere e scrivere si usavano gli stessi fili.

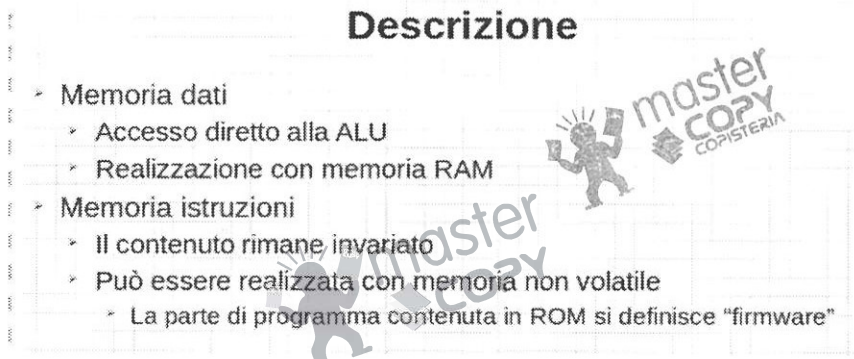
La prima architettura in realtà è **l'architettura di Harvard**



In questo caso la memoria dati e la memoria istruzioni sono in posti diversi con fili diversi, sicuramente una struttura di questo tipo è molto efficace perché contemporaneamente posso leggere un'istruzione e mettere a posto il risultato dell'istruzione precedente, possiamo fare insieme il prelievo delle istruzioni e operazioni sui dati ma in quei tempi questa soluzione era troppo complessa e si è affermata l'architettura di von Neumann usata come evoluzione di quella di Harvard nel senso che usa meno fili per collegare la parte che faceva i conti e prelevava le istruzioni dalla memoria.

Intorno agli anni 80 quando si è iniziato a pensare come elaboratore informazioni in tempo reale è stata ripescata l'architettura di Harvard, dopo che dal 64 ha prevalso Von Neumann.

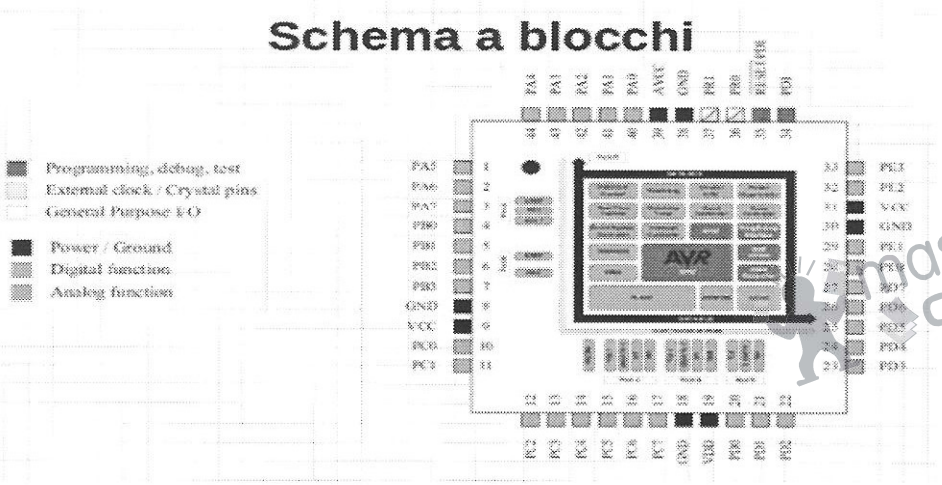
XMEGA è una architettura di Harvard anche perché essendo tutto nello stesso pezzo di silicio non era più necessario risparmiare i fili, l'architettura di Harvard è vantaggiosa perché non deve alternare operazioni sul programma e operazioni sulla memoria.



Nell 'architettura di Harvard la memoria di programma è permanente si chiama firmware, quindi tra l'*hardware* che è l'oggetto fisico, il *software* che è il programma scritto che può essere modificato c'è il *firmware* che è il programma che non viene più cambiato; viene messo nella memoria permanente e diventa il nucleo di quello che sa fare il microcontrollore.

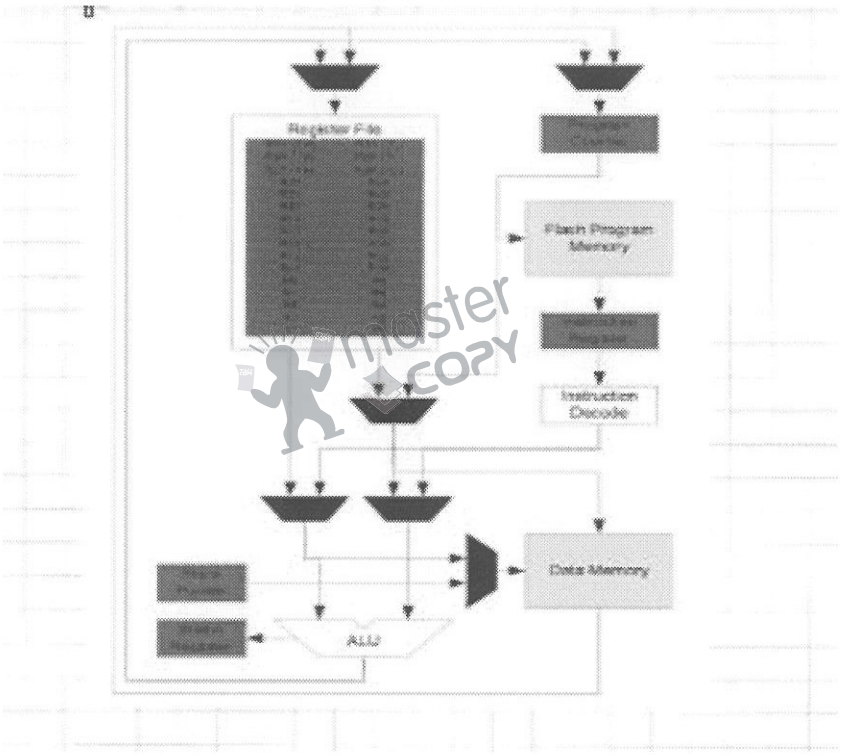
Una macchina di elaborazione generale è parzialmente un dispositivo come un cellulare o un tablet dove invece il programma che gira cambia a seconda dell'app che pigiamo, il sistema operativo prende dalla memoria permanente un pezzo di codice, lo trasferisce nella memoria di programma e lo manda in esecuzione, nelle macchine che eseguono molti compiti diversi la memoria di programma è importante che venga modificata mentre la parte fissa del programma è una memoria non volatile e si chiama firmware. La tendenza recente è che il firmware possa essere aggiornato ma è una procedura rischiosa.

Questo è lo schema a blocchi dell'XMEGA32



I blocchi blu sono i blocchi digitali, il contenitore è l'oggetto fisico (1 cm di lato) i piedini sono di metallo e sono saldati alla scheda, ci sono blocchi anche con funzione analogica e di potenza. Un microcontrollore ha anche dei circuiti specializzati non solo che fanno elaborazioni di tipo logico ma misurano anche tensioni possono fare anche elaborazioni sulle grandezze fisiche di tipi analogico.

CPU dell' XMEGA256A3BU



Il blocco blu sono i **Register File**, sono i registri di lavoro, cioè l'elenco dei registri; piccoli contenitori di memoria veloci dove mettere i numeri su cui lavorare. Sono una lista da R0 fino a R31 sono 32 registri e li hanno divisi in due metà e si trovano a coppie, a metà ho R14 R15. Ci sono alcune istruzioni che lavorano solo con metà, i registri più flessibili sono da R16 in poi (secondo banco), i primi registri da R0 in poi (primo banco) sono più limitati. Che tipo di numero ci dà un registro?

Un registro contiene al massimo 8 bit cioè possono contenere da numeri da 0 a 255, se metto una costante 2000 in un registro non posso farlo perché al massimo ci va 255, se devo mettere un numero con segno si usa la strategia del complemento a due e i numeri vanno da (-128 a +127), la cosa importante riguarda alcune coppie, i 16 bit che formano la coppia R1 e R0 viene usata da alcune operazioni di tipo boole per scrivere il risultato ,in particolare le moltiplicazioni metteranno il risultato direttamente nella coppia r0 r1, le coppie r27 r26 r28 r29 r30 r31 si chiamano *puntatori* (pointer) perché possono essere usate da alcune istruzioni come contenitori di un indirizzo di memoria. Quando assumono questa funzione si chiamano con un altro nome (XL XH e la coppia r26 r27 si chiama x, la seconda coppia si chiama y e la terza coppia si chiama z).

La lettera L sta per low (parte bassa meno significativo) e H (high più significativo)

Il blocco che fa i conti è la **ALU** (aritmetic logic unit) in cui vengono presi questi due operandi e vengono combinati insieme per dare questo risultato. I dati vengono da questi trapezi neri e vanno ad altri due trapezi neri. I trapezi si chiamano *multiplexer* sono smistatori, sono canali che possono scegliere di far proseguire in uscita un'alternativa fra due o possibili alternative, quale alternativa esce viene decisa da dei fili di controllo (in verde) racchiusi in un fascio e questo fascio di fili gestisce tutti questi smistamenti e viene da una parte della cpu che si chiama la parte di controllo del processore.