Microcontrollore

È un dispositivo elettronico integrato in un unico chip a scopi di controllo digitale ben definito. Nel nostro corso si prende in esame il microcontrollore (μC) della ATMEL, famiglia AVR e modello ATxmega256A3BU. Dal data sheet del μC si deducono prestazioni e componenti. In generale un μC è costituito da una CPU (Central Process Unit) che ospita il cuore operativo del dispositivo; esso è supportato da una memoria di programma che ospita il codice operativo ed una memoria che ospita i dati da elaborare. Attraverso un elemento detto BUS che gestisce questo flusso di dati ed un oscillatore che genera segnali di clock in modo da temporizzare le operazioni eseguite il μC dialoga con porte esterne o interne come rispettivamente periferiche e porte di I/O analogiche e digitali. D'ora in poi si farà dunque riferimento al μC della famiglia AVR XMEGA e modello ATxmega256A3BU.

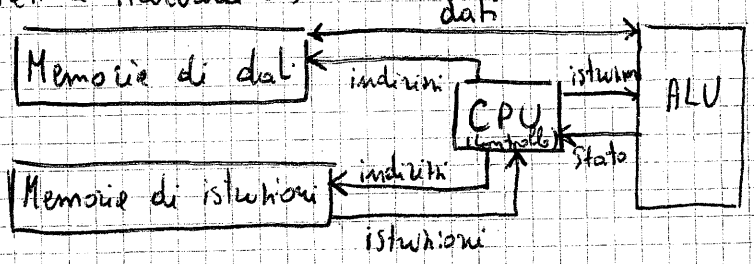
MASTER COPY
Tel. 050 8312126
Cell. 388 9837745

CPU: cuore

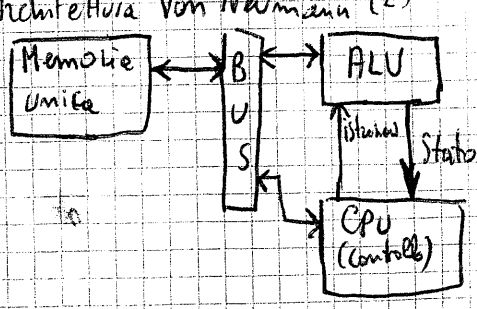
La principale caratteristica di queste CPU a 8/16-bit è di eseguire il codice e le operazioni sui dati, (accedendo alle memorie/periferiche) ed eseguire il programma nella flash memory, detta anche Letture-scrittura, dove si registrano in un vettore di ~~registri~~ registri (n) un ugual numero di bit (n) e dunque di informazioni.

- **32x8-bit** di registri connessi alla ALU (Register file)

Architettura Harvard (1)



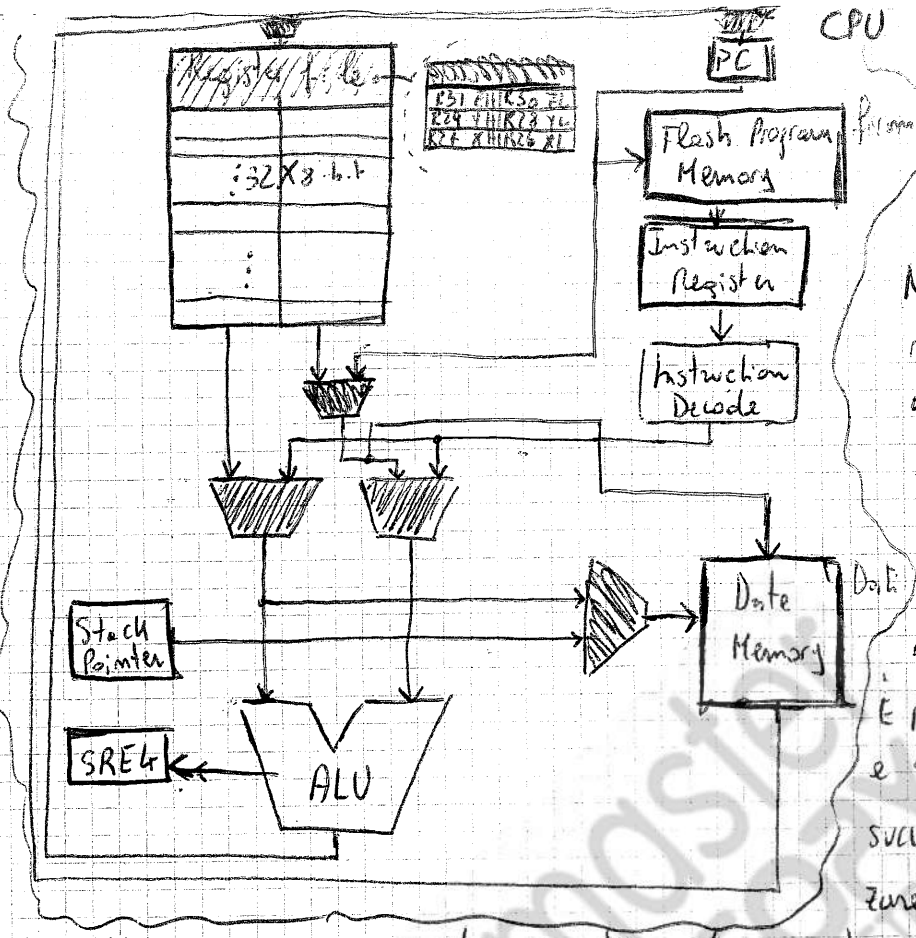
Architettura Von Neumann (2)



Si sono messe a confronto le due architetture hardware per mettere in evidenza svantaggi e vantaggi di una e dell'altra architettura.

Il nostro µC usa la (1) ma vedendolo in generale si nota che per la (2) è usato un unico BUS ed in particolare un'unica memoria riducendo la complessità realizzativa della rete. Il problema è che usando un unico bus e unica memoria c'è il pericolo di sovrapposizione di info e dati e non è possibile leggere e contemporaneamente scrivere le istruzioni e i dati. Questi due problemi vengono superati con la seconda architettura (seconda perché temporalmente ha succeduto la (2)). Quindi, perdendo in semplicità hardware, si guadagna in questi fattori.

La (1) presenta due memorie: una memoria dati (RAM, Random Access Memory) che dialoga con la ALU da cui preleva dati su cui lavorare; una memoria istruzioni (ROM, Read Only Memory) che contiene il codice operativo deseguito dalla CPU (tale codice è anche detto firmware); il firware è il punto di incontro tra hardware e software in quanto predispone le componenti hardware della macchina ed interagire con i software che si eseguono sul µC di fatti. In esso esistono istruzioni di accensione delle componente hardware, sia CPU che altre periferiche. L'aumentata complessità di realizzazione hardware non compete con la velocità incrementata (nella famiglia AVRXMEGA per ogni ciclo di clock si ha l'esecuzione di 1 milione di istruzioni (MIPS, Million Instructions Per Second). La possibilità di implementare up-grading aggiungendo bus o frazionando ulteriormente fetch ed esecuzione consente la versatilità del µC in genere per nell'elaborazione di segnali digitali (Digital Signal Processing, DSP). È inoltre possibile pensare di eseguire in parallelo più operazioni (come esecuzione e letture di istruzioni successive) aggiungendo registri intermedi o ALU aggiuntive per il calcolo di indirizzi. Quindi lo schema della CPU del µC ATxmega256A3BU è:



SREG: Status Register
SP: Stack Pointer
PC: Program Counter

Notare la presenza di 2 memorie separate e quindi architettura di Harvard e la presenza dei register file (32 x 8-bit nel ATxmega256A3BU) e di altri registri intermedi nell'esecuzione di un'operazione. È possibile eseguire un'operazione e nello stesso tempo prelevare la successiva dalla FLASH, per ottimizzare la velocità di esecuzione.

La ALU esegue operazioni tra registri o tra registro - costante. Nel register file oltre ad avere 32 x 8-bit registri, si può usare la coppia X, Y, Z come un unica cella di registro da 16-bit ciascuna come puntatore di indirizzo nella memoria dati.

Perché 2 registri da 8 bit per i puntatori?

La memoria dei dati ha dimensione 24KB, cioè ogni cella ha un indirizzo da 0 a (24K - 1) = 24576 che in binario si esprime con un numero di bit pari a (15) ($2^{15} - 1 = 32767$). In questo modo un registro da 15 bit consente di puntare a qualunque locazione (indirizzo) di memoria dati.

MASTER COPY
Tel. 050 8312126
Cell. 388 9837745

La memoria di dati è divisa come:

I primi 4KB indirizzi → registri di I/O [0, 4095] → memoria di I/O

→ [0x00, 0x3F] → accesso diretto

[0x0040, 0x0FFF] → memoria di I/O estesa

Il resto è SRAM, memoria con cui poter lavorare.

PC: In stato di accensione, il Program Counter contiene l'indirizzo della cella di memoria (22-bit) nella FLASH che deve essere prelevata per prima (in avvio '0'). Viene automaticamente incrementato di una dimensione pari alla cella eseguita.

SP: Lo Stack Pointer contiene l'indirizzo della SRAM in cui sono salvati gli indirizzi
* (da 16 bit) dello ~~codice da eseguire~~ stato attuale di PC a seguito di una subroutine che richiede una prelievo di dati non lineare; è quindi richiesta la necessità di salvare l'attuale indirizzo di FLASH per poi riprenderlo nello SP una volta terminata la subroutine. Dimensione Stack ≡ Dimensione SRAM per lo stack.

SREG: Contiene in forma abbreviata varie informazioni sulle operazioni logiche e aritmetiche più recenti prodotte dalla ALU. Tramite FLAGS è possibile dunque creare operazioni condizionali con l'uso di questi rendendo nel contempo il codice più compatto. Sfrutta l'I/O memory space.

* Ad avvio lo SP indica il più alto indirizzo di SRAM e la uscita dello stack (PUSH instruction) implica un decremento dello SP. All'inverso un prelievo dallo Stack (POP instruction) incrementa lo SP.

Flash Program Memory (256 KB) ~~32 KB~~

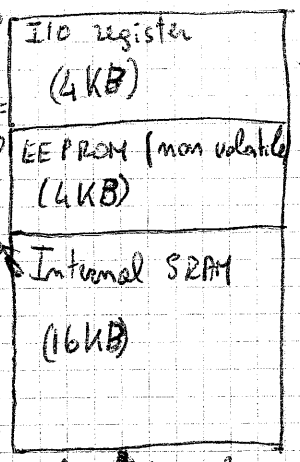
Nella FLASH è contenuto il firmware del µC con istruzioni da 16 o 32 bit; le locazioni sono tuttavia ognuna da 16 bit. Tale memoria si divide primariamente in due locazioni: Application Section (~~240 KB~~ 256KB+8KB) e Boot Loader Section (~~8 KB~~ 8KB) che insieme compongono una memoria ~~flash~~ flash. Le due sezioni si distinguono per diversi codici di protezione che impediscono di sovrascrivere il firmware del µC attraverso la SPM instruction. Tuttavia l'esistenza di tale istruzione ci fa capire che è possibile in alcuni casi sovrascrivere il firmware tenendo conto che un'operazione del genere può essere fatale per il µC stesso.

Vengono usati ~~240 KB~~ 256 KB per il firmware, 8 KB per salvare dati e 8 KB per avviare l'hardware vero e proprio (to boot = avviare).

Alcuni bit vengono usati per altri dati di calibrazione, dati circa caratteristiche dell'oscillatore o componenti analogiche e un'identificazione digitale del prodotto.

Data Memory (AVR XMEGA 256A3BU)

La memoria dati, vista come un blocco continuo, è in realtà divisa per sezioni di diverse memorie per un totale di 64KB la memoria è suddivisa in



→ Indirizzi e dati da periferiche e altri moduli tra cui la CPU; tali locazioni dialogano con i register file. *

→ Memoria programmabile e non-volatile.

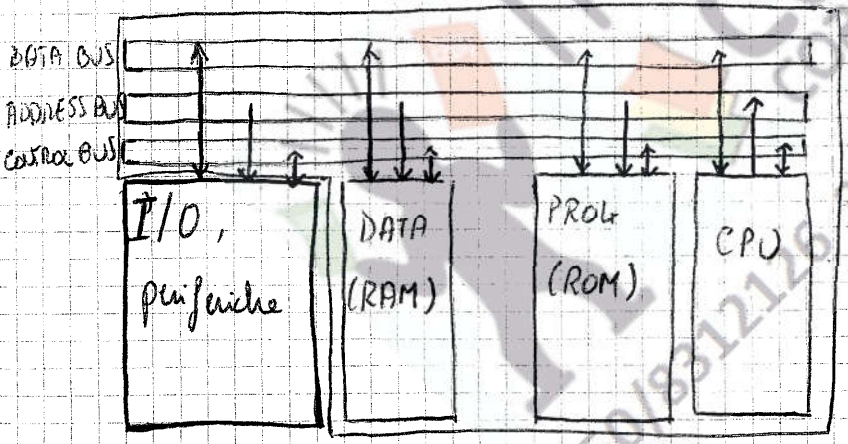
→ Spazio di lavoro, memoria per la restata dello STACK.

* è possibile ampliare la memoria (EMI) fino a 64MB per µC di famiglia AVR che non siano per XMEGA 256A3BU

* alcuni indirizzi hanno forma alfanumerica per essere più facilmente memorizzabili. Si consulti il datasheet laddove interessi questo dettaglio

= I primi 16 indirizzi (0x0000 to 0x000E) vengono usati per inizializzare variabili globali e registrare: valori di flags del SREG

Lo schema più generico di un µC, sottolineando i bus e nel contempo architettura Hardware è:



Notare che è solo la CPU a decidere a quali indirizzi scegliere l'op. di read/write sottolineando il suo ruolo operativo.

MASTER COPY
Tel. 050 8312126
Cell. 388 9837745

Un'ulteriore distinzione si compie tra µP e µC. Il µP è la parte di schema all'interno del blocco nero; le parti in verde (periferiche) rendono unico un µC che è il complesso di due blocchi. Una famiglia di µC (come la AVR XMEGA) è composta da identici µP e varie periferiche e I/O periferiche

Pin di I/O

In un µC è possibile controllare le periferiche dai pin, cavetti piedini metallici su cui si registrano grandezze digitali elaborate (Output) o da elaborare (Input); è dunque un elemento bidirezionale.