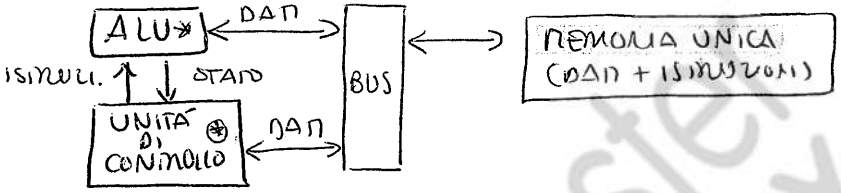
IL MICROCONTROLLORE

- MICROCONTROLLORE → architettura di un computer
  - CPU processore, esegue operazioni (CENTRAL PROCESSING UNIT) — ALU, UNITÀ DI CONTROLLO
  - ROM contiene il programma (READ ONLY MEMORY)
  - RAM contiene i dati (RANDOM ACCESS MEMORY)
  - PERIFERICHE DI I/O

- CORE = Singola CPU

- Ci sono 2 ARCHITETTURE CHE SI SONO AFFERMATE nel tempo:

  - o VON NEUMAN



* ALU svolge le operazioni aritmetiche (vedi dopo)

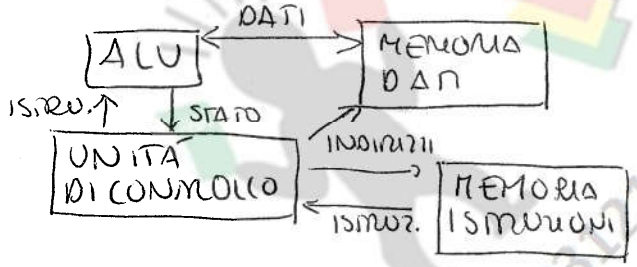
⊛ UC decide cosa deve fare la ALU e capisce dati deve svolgere le operazioni

* il BUS accede alla MEMORIA SFRUTTANDO GLI INDIRIZZI DEI DATI/ISTRUZIONI FORNITEGLI dalla ALU e dalla UNITÀ DI CONTROLLO

* VANTAGGIO = MEMORIA UNICA → UNICO INSIEME DI FILI di collegamento tra la memoria e il resto del circuito

* SVANTAGGIO = non posso prelevare CONTEMPORANEAMENTE ISTRUZIONI E DATI, RISCHIO DI SOVRAPPOSIZIONI TRA DATI E ISTRUZIONI

  - o HARVARD



MASTER COPY
Tel. 050 8312126
Cell. 388 9837745

* Ci sono 2 MEMORIE DIFFERENTI: una per i DATI e una per le ISTRUZIONI

* È l'ARCHITETTURA PIÙ DIFFUSA (anche quella del nostro µ)

* la MEMORIA DATI si può realizzare tramite una RAM ed HA ACCESSO DIRETTO ALLA ALU

* la MEMORIA ISTRUZIONI è una MEMORIA NON VOLATILE (non è possibile modificarla) e può essere realizzata con una ROM. È detto FIRMWARE l'insieme delle istruzioni di programma nella memoria permanente

NB HARDWARE = parte fisica SOFTWARE = istruz. di prog. nella memoria volatile (modificate continuamente → variano)

* VANTAGGIO = Sono PRELEVATE DATI E ISTRUZIONI CONTEMPORANEAMENTE (eseguire e prelevare contemporaneamente)

MASTER COPY
Tel. 050 8312126
Cell. 388 9837745

LA CPU (PROCESSORE)

La CPU ESEGUE UNA SERIE DI ISTRUZIONI:
- LEGGE PROGRAMMA E DATI DALLE MEMORIE
- GESTISCE LA ALU ATTRAVERSO LA MEMORIA INTERNA DI LAVORO (REGISTRI)

COMPONENTI DELLA CPU

- REGISTRI INTERNI (MEMORIA DI LAVORO)
  Sono 32 contenuti da 8 bit che memorizzano un valore logico per ogni b.t. Contengono gli OPERANDI SU CUI OPERA LA ALU, la quale PONE IL RISULTATO SEMPRE IN UN REGISTRO.
  DA QUESTI I DATI POSSONO ESSERE TRASFERITI ALLE ALTRE MEMORIE ESTERNE ALLA CPU

- ALU (ARITMETICAL LOGIC UNIT)
  compie operazioni tramite le vie locali dai registri e dalla unità di controllo

- CONTATORE DI PROGRAMMA (PC)
  Registro particolare da 22 bit. Ad ogni segnale di clock il suo CONTENUTO VIENE USATO PER SCEGLIERE L'ISTRUZIONE DA ESEGUIRE. CONTIENE L'INDIRIZZO DELL'ISTRUZIONE CHE VA USATA. Ad ogni clock il suo contenuto viene aggiornato. Garantisce lo svolgimento ordinato delle istruzioni.
  A volte istruzioni possono MODIFICARNE IL CONTEGGIO (il CPU - dello CONTATORE DI PROGRAMMA perché "conta" le istruzioni da eseguire)

- STATUS REGISTER
  Registro particolare che CONTIENE I FLAG (1 b.t del registro). QUESTI INDICANO E DANNO INFORMAZIONI SULL'EFFETTO AVUTO DALL'ISTRUZIONE E LO MANDANO A QUELLA SUCCESSIVA. Alcune istruzioni hanno un effetto dovuto a seconda dei flag. ATTENZIONE FA PARTE DELLA CPU MA È CONTENUTO NELLA MEMORIA DATI ESTESA

- STACK POINTER
  Contiene (è un registro speciale) l'INDIRIZZO DELLA PRIMA CELLA VUOTA DELLA MEMORIA DATI (vedi dopo) ATTENZIONE È CONTENUTO NELLA MEMORIA DATI ESTESA

CIRCUITI PERIFERICI

Sono circuiti dedicati ad APPLICAZIONI SPECIFICHE. Hanno una struttura HARDWARE DEDICATA ALLA RISOLUZIONE DI PARTICOLARI TIPI DI PROBLEMI e possono essere CONFIGURATI A TALE SCOPO DALLA CPU. Quest'ultima DIALOGA CON LE PERIFERICHE per la risoluzione di problemi dedicati.
La CONFIGURAZIONE AVVIENE CON LA SCRITTURA DA PARTE DELLA CPU AGLI INDIRIZZI DI PERIFERICA (scrive nei registri di periferica attraverso delle istruzioni adeguate)
LE OPERAZIONI FATTE SONO LE PERIFERICHE SONO FATTE TRAMITE LA SCRITTURA E LA LETTURA DEI CONTENUTI DEI REGISTRI DI PERIFERICA

Esempi di periferica:

- PIN DI INGRESSO E USCITA:
  Una PERIFERICA DI I/O SERVE A SCRIVERE/LEGGERE DEI VALORI DIGITALI SUI PIN che sono dei COLLEGAMENTI TRA CIRCUITO INT DEL MICROCONTROLLORE E UN CIRCUITO ESTERNO → USO: DIALOGO CON DISPOSITIVI ESTERNI ALLA CPU

- PERIFERICHE PER LA GESTIONE DI EVENTI NEL TEMPO ex timer (tempo trascorso tra un evento e un altro), contatore (conta quante volte è successo un evento)

- PERIFERICHE GENERAMENTE DI SEGNALI generazione di segnali di cui di forma variabile/varia nell'ampiezza

- CONVERTITORE ANALOGICO-DIGITALE riceve una TENSIONE ANALOGICA (TENSIONE ELETTRICA) e la TRADUCE IN UN SEGNALE DIGITALE (I/O) ex. COMPARATORE ANALOGICO confronto tensione analogica e restituisce un valore digitale

- CONV. DIGITALE-ANALOGICO riceve un SEGNALE DIGITALE e lo MANDA IN UNA TENSIONE ANALOGICA

CIRCUITI AUSILIARI

Circuiti non interni direttamente alla CPU ma HANNO LO SCOPO DI FAR FUNZIONARE NEL GIUSTO MODO IL μ CONTROLLORE

- GENERATORE DI CLOCK

  Generazione di un ONDA QUADRA (TRAMITE UN OSCILLATORE) DI FREQ VARIABILE (È POSSIBILE SCEGLIERNE LA FREQ.). QUESTA ONDA QUADRA DETERMINA LA TEMPISTICA CON LA QUALE VENGONO ESEGUITE LE OPERAZIONI DEL μ

- SUPERVISORE DI ALIMENTAZIONE

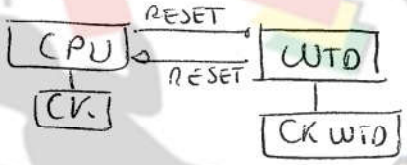
  Supporto per l'alimentazione. CIRCUITO CHE IMPEDISCE IL FUNZIONAMENTO DELLA MACCHINA SOTTO UNA DETERMINATA SOGLIA DI TENSIONE (DELL'ALIMENTAZIONE)

- SUPERVISORE DELLA MODALITÀ OPERATIVA

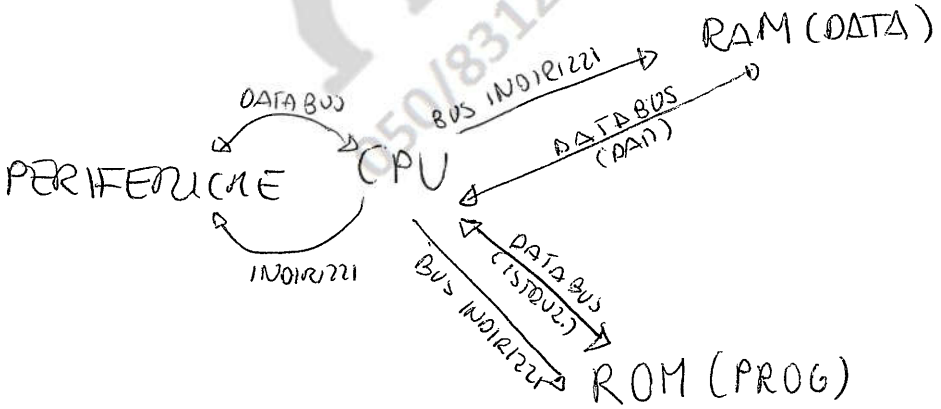
  Gestisce VARI LIVELLI DI STAND BY APPLICABILI AL μ

- WATCH DOG TIMER

  Circuito AUTONOMO DALLA CPU CON UN CLOCK PERSONALE. ~~USE~~ IL WATCH DOG (il suo funzionamento) È LEGATO AL RESET DELLA CPU e la CPU È LEGATA AL RESET DEL WATCHDOG. PUÒ essere ATTIVATO O MENO DAL PROGRAMMATORE. Il CONTEGGIO DEL WATCH DOG, partendo da un CERTO TOTALE, dà IL VIA AL RESET DELLA CPU. La CPU HA, A SUA VOLTA, UNA FUNZIONALITÀ CHE LE PERMETTE DI AZZERARE IL WTD. Questo permette il RESET DELLA CPU NEL CASO CHE QUESTA NON SI FUNZIONI PIÙ CORRETTAMENTE. Se la CPU si blocca e non c'è azzeramento del WTD quello, scaduto il suo conteggio RESETTA LA CPU.



MASTER COPY
Tel. 050 8312126
Cell. 388 9837745




master
copy
MASTER COPY
Tel. 050 8312126
Cell. 388 9837745
COPISTERIA
050/8312126 388/9837745

LINGUAGGIO ASSEMBLY: INTRODUZIONE

Linguaggio ASSEMBLY / MACCHINA = strumento più vicino alla realtà fisico-elettronica della macchina con cui si può programmare. A differenza dei linguaggi di alto livello (C++) ogni macchina o famiglia di macchine ha un DETERMINATO LINGUAGGIO CON ELEMENTI STRETTAMENTE COLLEGATI AL TIPO DI MACCHINA SU CUI SI PROGRAMMA (funziona solo su quella). Nel linguaggio macchina NON C'È COMPILATORE: IL PROGRAMMA DIVENTA AUTOMATICAMENTE L'ESEGUIBILE. Le ISTRUZ. ASS. CORRISPONDERANNO IN MODO UNIVOCO A DEI BIT SPECIFICI. OGNI ISTRUZ. HA UN DET. TEMPO DI ESECUZIONE → cambiando le istruzioni posso sapere quanto ci mette ad eseguirle (posso temporizzare l'algoritmo)

OGNI CHIP O OGNI FAMIGLIA DI CHIP HA IL SUO ASSEMBLY:

la nostra famiglia è ATMEL con l'architettura AVR

STRUTTURA DI UN PROGRAMMA ASSEMBLY

Nel linguaggio macchina l'elemento base che codifica un'istruz. è UN CODICE MNEMONICO (facilmente ricordabile) che dà la cosa fare alla macchina su certi OPERANDI specificati

```
. org 100          ] DIRETTIVA *

MULT:   push   CNT
        clr    OUT 1
        mov    OUT Ø, IN Ø
        ldi    CNT, 8
        lsr    OUT Ø
H1:     brcc   M2
        add    OUT 1, IN1
M2:     ror    OUT 1
        ror    OUT Ø
        :
```

LABEL "etichetta" | CODICE MNEMONICO | OPERANDI | ; commento

→ nelle istruzioni in cui voglio sapere l'indirizzo di quella istruzione uso una label

MODO PER INDICARE L'INDIRIZZO DI UNA ISTRUZIONE

ex MULT: = quando il programma viene tradotto annoterà il valore dell'indirizzo della prima istruzione che lo segue (ovvero l'indirizzo di push)

* DIRETTIVA: non è destinata a diventare codice macchina, è un COMANDO RIVOLTO ALL'ASSEMBLATORE. Questo ha molto effetto sul modo in cui viene tradotto il programma

ex . org 100 → le istruzioni che seguono le scrive a partire dall'indirizzo 100

- .equ pippo=8 → associa a una costante un valore (pippo) (8)
- .def → definizione di una variabile

[17.06]

28 SETT