

# CONTROL UNIT

martedì 22 novembre 2022 18:22

Iniziamo a cercare di capire "come" la nostra CU può andare a pilotare, l'unità di processamento appena descritta;

L'UNITÀ DI CONTROLLO IMPLEMENTA ISTRUZIONI MULTICICLO. L'ESECUZIONE DELLA KIA ISTRUZIONE VIENE SUDDIVISA IN VARI CICLI MACCHINA e CIASCUN CICLO MACCHINA VIENE DIVISO IN VARI STATI MACCHINA.



IN CIASCUNO DI QUESTI STATI MACCHINA ANDREMO AD ABILITARE PEZZI DI ESECUZIONE DELL'UNITÀ DI PROCESSAMENTO.

**ESEMPIO:** Leggo il contenuto di R1P e lo SCRIVO sul databus = STATO MACCHINA

Voce 011 24:57

TUTTO QUELLO CHE SUCCEDERE VIENE IDENTIFICATO COL TERMINE:

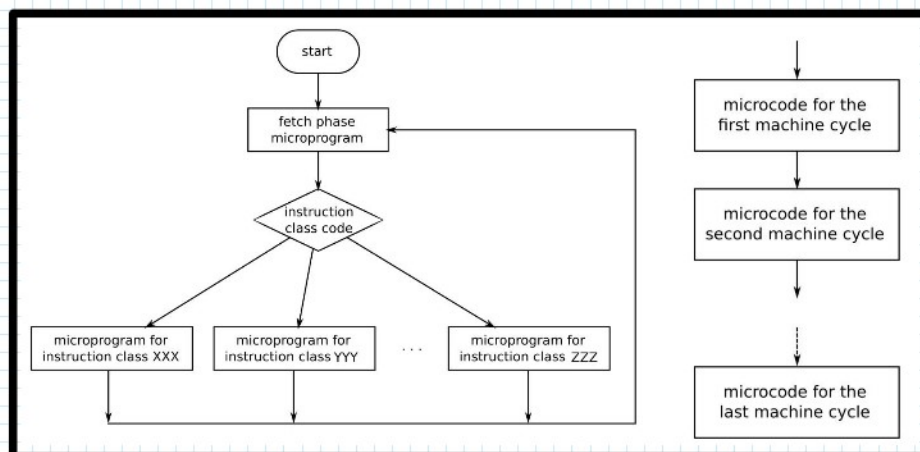
## MICRO OPERAZIONI

Ogni ISTRUZIONE, siccome è spezzettata in vari stati macchina, VERRÀ IMPLEMENTATA da un numero corrispondente di MICROOPERAZIONI.

Una MICRO OPERAZIONE è una sequenza di stati macchina IN CUI LA CU di CIASCUNO STATO macchina abilita alcuni segnali di controllo all'interno dell'unità di PROCESSAMENTO per pilotare l'hardware che abbiamo visto fino a questo momento.



ALLA FINE LA MIA CU È UNA MACCHINA A STATI CHE FA SEMPRE LA STESSA COSA:



QUAL'È L'INPUT DI QUESTA MACCHINA A STATI ?

• PER ESEGUIRE IL MICROPROGRAMMA DI UNA ISTRUZIONE, LA CU USERÀ COME INPUT:

- LA CLASSE E IL TIPO DELL'ISTRUZIONE, CONTENUTA IN IR;
- LE VARIABILI DI CONDIZIONE CHE ARRIVANO DA FUORI LA CPU;
- LE VARIABILI DI CONDIZIONE CHE ARRIVANO DA DENTRO LA CPU, AD ESEMPIO I BIT DI FLAGS.
- LA MODALITÀ DI INDIRIZZAMENTO DEGLI OPERANDI - DERIVATA DA IR.

• Il numero di segnali di output della CU dipende dall'implementazione della PU e dei moduli esterni

COME IMPLEMENTIAMO QUESTA MACCHINA A STATI?

QUAL'ERA LA FUNZIONE COMPATTA DI UNA MACCHINA A STATI? UNA ROM.

UTILizzeremo una ROM PER IMPLEMENTARE LA NOSTRA CU PERCHÈ È COMPATTA!

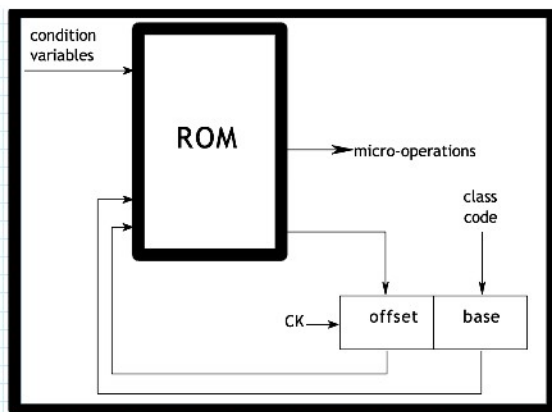
PER TUTTE LE VARIABILI DI CONDIZIONE MI PERMETTE DI DEFINIRE UN INDIRIZZO, E QUELLO INDIRIZZO SELEZIONA LA COLONNA, E QUELLA COLONNA DEFINISCE L'INSIEME DELLE MICROOPERAZIONI!

Siccome abbiamo





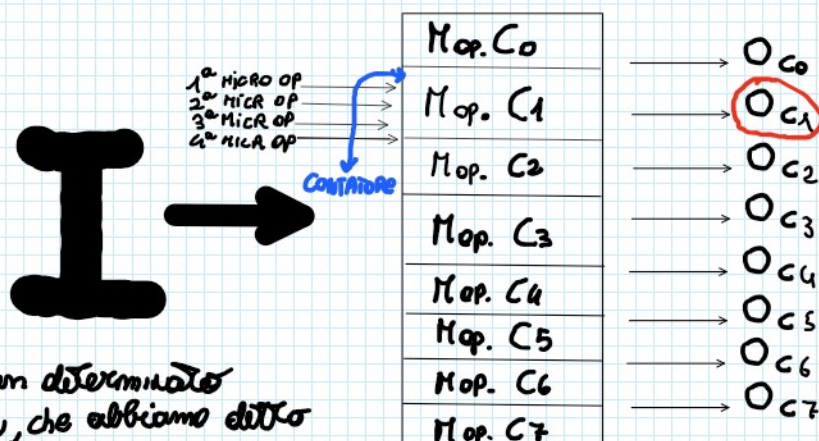
Siccome abbiamo organizzato le nostre istruzioni in classi, possiamo usare una ROM paginata;



Ciascuna pagina conterra al suo interno un microprogramma di una determinata classe di istruzioni;

Dato un segnale di input, che sono le variabili di condizione, per quelle variabili di condizione vanno generate contemporaneamente le micro operazioni per tutte le classi di istruzioni.

A quel punto possiamo selezionare la pagina che ci interessa: utilizzando l'opcode della nostra classe di istruzioni.



dato un determinato input  $i$ , che abbiamo detto essere tutte le variabili di

condizione, generiamo un insieme di output: output per la classe 0, per la classe 1, etc...

Questa ROM lineare viene divisa in pagine, sono dei blocchi omogenei della nostra ROM che all'interno le micro operazioni della classe 0 (Mop. C0), le micro operazioni della classe 1 (Mop. C1) ....

Quindi agli input, che sono le mie variabili di condizione, mi permettono di capire quali scelte io devo fare all'interno;

L'OPCODE MI PERMETTE DI scegliere un solo output tra tutti quelli che vengono generati dalla mia ROM, ad esempio O C1

Mi manca di sapere all'interno della pagina a quale micro operazione sono arrivato, quindi entro la mia prima micro operazione, la 2<sup>a</sup>, ..... etc!

L'ESECUZIONE di CIASCUNA di queste micro operazioni, determinerà un altro output, e quell'output sono i segnali di controllo che vanno verso l'unità di processamento.



DATA LA CLASSE DELL'ISTRUZIONE, COME FACCIO A SAPERE QUALE MICROOPERAZIONE DEVO ESEGUIRE? LE MICROOPERAZIONI SEGUONO LO STESSO SCHEMA DELLE MIE ISTRUZIONI IN MEMORIA, LE **ESEGUO UNA DOPO L'ALTRA**.

Quindi devo mantenere un contatore che tiene traccia delle prossime microoperazioni da eseguire;

IL MODULO M, che memorizza lo stato della mia macchina a stati, È composto da due parti:

- **BASE**: CHE IDENTIFICA LA PAGINA DELLA MIA ROM IN FUNZIONE DELLA CLASSE DELL'ISTRUZIONE

- **OFFSET**: IDENTIFICA ESATTAMENTE QUEL CONTATORE.

DATA CIASCUNA MICROOPERAZIONE IO POSSO DIRE: "vai alla successiva", oppure "fai un salto verso un'altra"

UNO DEGLI OUTPUT DELLA MIA UNITÀ DI CONTROLLO SARÀ UN VALORE CHE AGGIUNGERE' L'OFFSET.

Posso modificare il contenuto di offset per implementare il mio micro programma!

Tutto questo è rappresentato come una macchina a stati.

Che per efficienza implementiamo con una **ROM**!

Le variabili di condizione IN INPUT SONO TANTE!

Se ci ricordiamo nell'implementazione della ROM io davanti ho un **decoder**!

MA IL DECODER RICEVE TANTE VARIABILI DI INPUT.

- ZMP C, ha bisogno del CF come variabile di condizione
- NOV non ha bisogno di variabili di condizione.

NON HA SENSO MANDARE TUTTI GLI INPUT? NON POSSO CERCARE DI RIDURLI?

DI QUANTI INPUT HO davvero bisogno su questa ROM? di quante variabili di condizione ho davvero bisogno? Del numero massimo richiesto dalla micro operazione che me ne chiede di più.

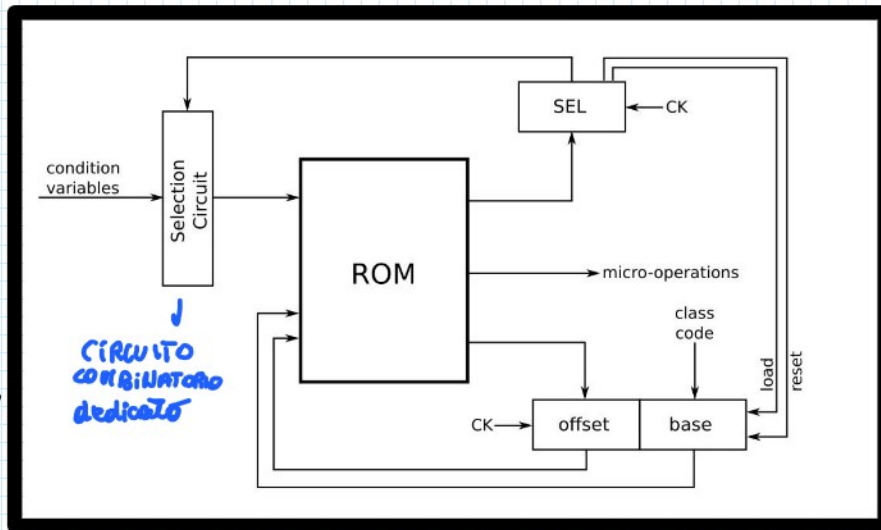
**Ridotto le mie variabili di INPUT, selezionando solo quelle che mi servono!**

Quindi utilizzo un circuito di selezione.



SE MI ARRIVANO  
100 variabili di  
condizione, MA  
AL PIÙ ME NE  
SERVONO 3, del  
circuitto di  
selezione ne  
posso uscire 3.

Riducendo drasticamente  
la complessità  
del decoder  
di ingresso alla  
ROM.



CIRCUITO  
COMBINATORIO  
DEDICATO

Se parliamo di una JUMP C, Jump carry, a me serve propagare il carry!  
Se parliamo di una JUMP Z, a me serve lo ZF non il CF.

IL CIRCUITO di selezione deve selezionare le variabili esatte che servono per quella specifica  
micro operazione!!!

L'output della ROM va in un **Registro tampone**, che dice al mio circuito di selezione  
quali sono gli input necessari.

**Programmatore:** "PER LA PROSSIMA MICROOPERAZIONE TIRAMI SOLO IL CARRY FLAG"

**ROM:** "Uff! ... VA BENE"