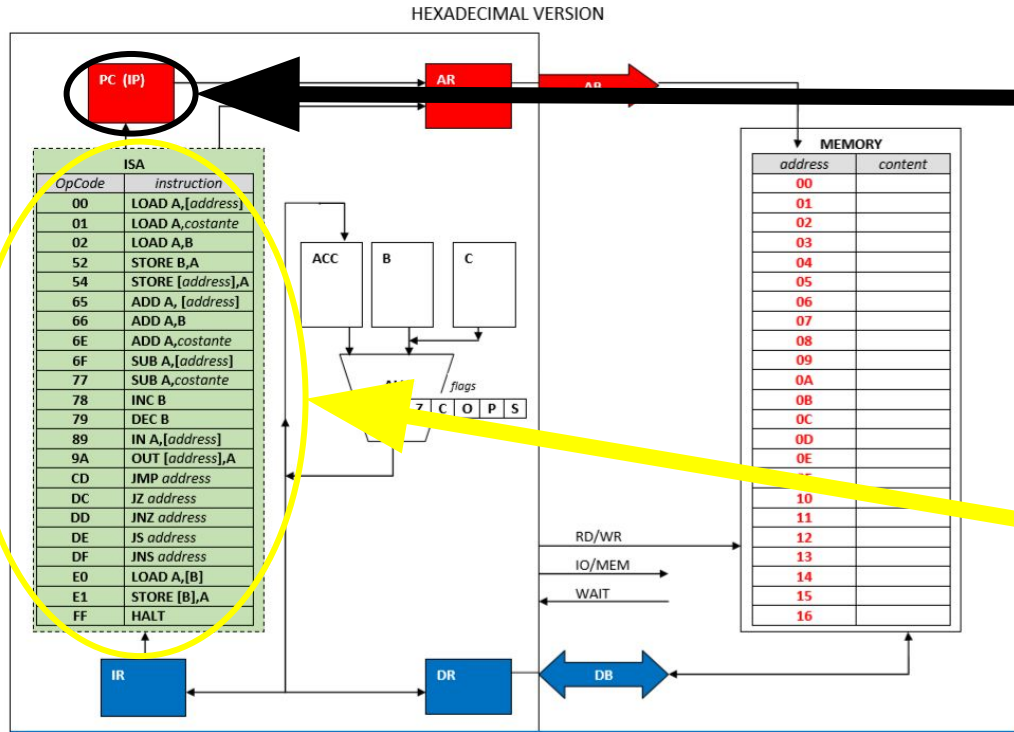


**ciclo**  
**Fetch-Decode**  
**Execute**  
**(FDE)**

**Ernst Tommaso**  
**3°DIT Calvino a.s. 19-20**

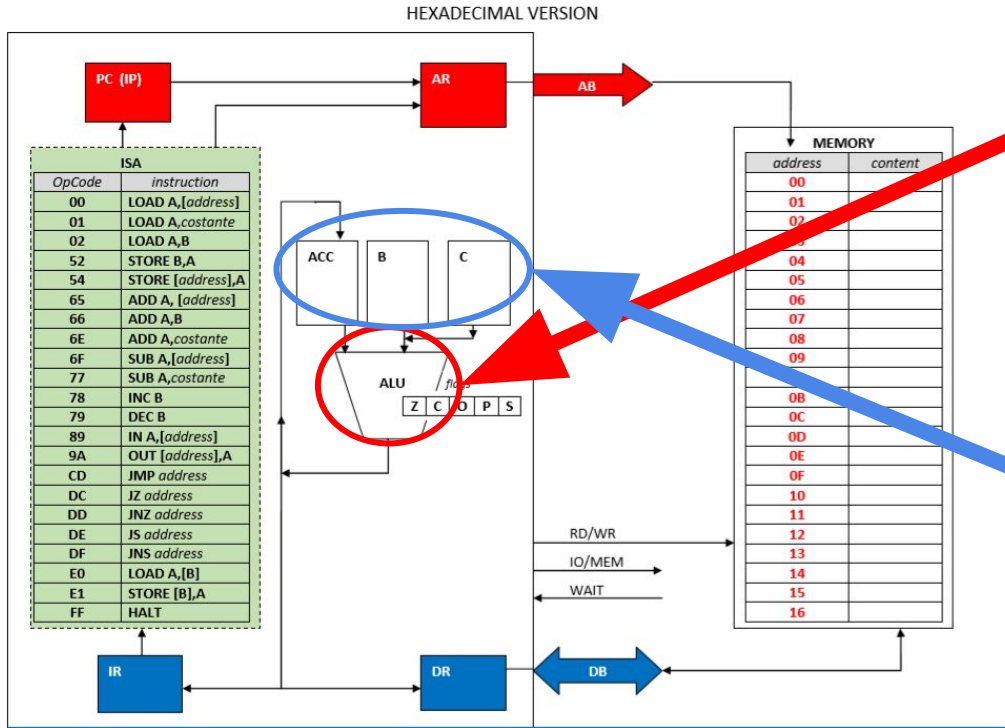
# PC (Program Counter)-ISA (Instruction Set Architecture)



**PC:** registro puntatore che contiene l'indirizzo dell'istruzione da prelevare; ad ogni ciclo FDE il suo valore aumenta di 1

**ISA:** è l'insieme delle istruzioni eseguibili dal  $\mu P$ ; quando arriva un op-code la CU (Control Unit) decodifica tale informazione e attiva i circuiti interni per l'esecuzione dell'istruzione

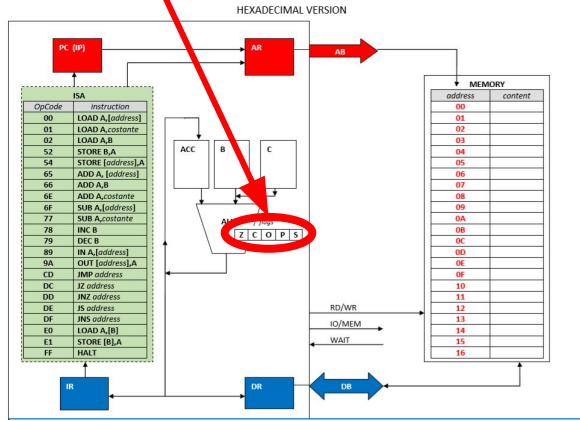
# ALU (Arithmetic-Logic Unit) & REGISTRI



**ALU:** ha il compito di svolgere le operazioni aritmetico-logiche all'interno del microprocessore

**REGISTRI:** sono la memoria interna al microprocessore; il più importante è l'accumulatore, che di norma contiene il risultato delle operazioni della ALU

# i Flag

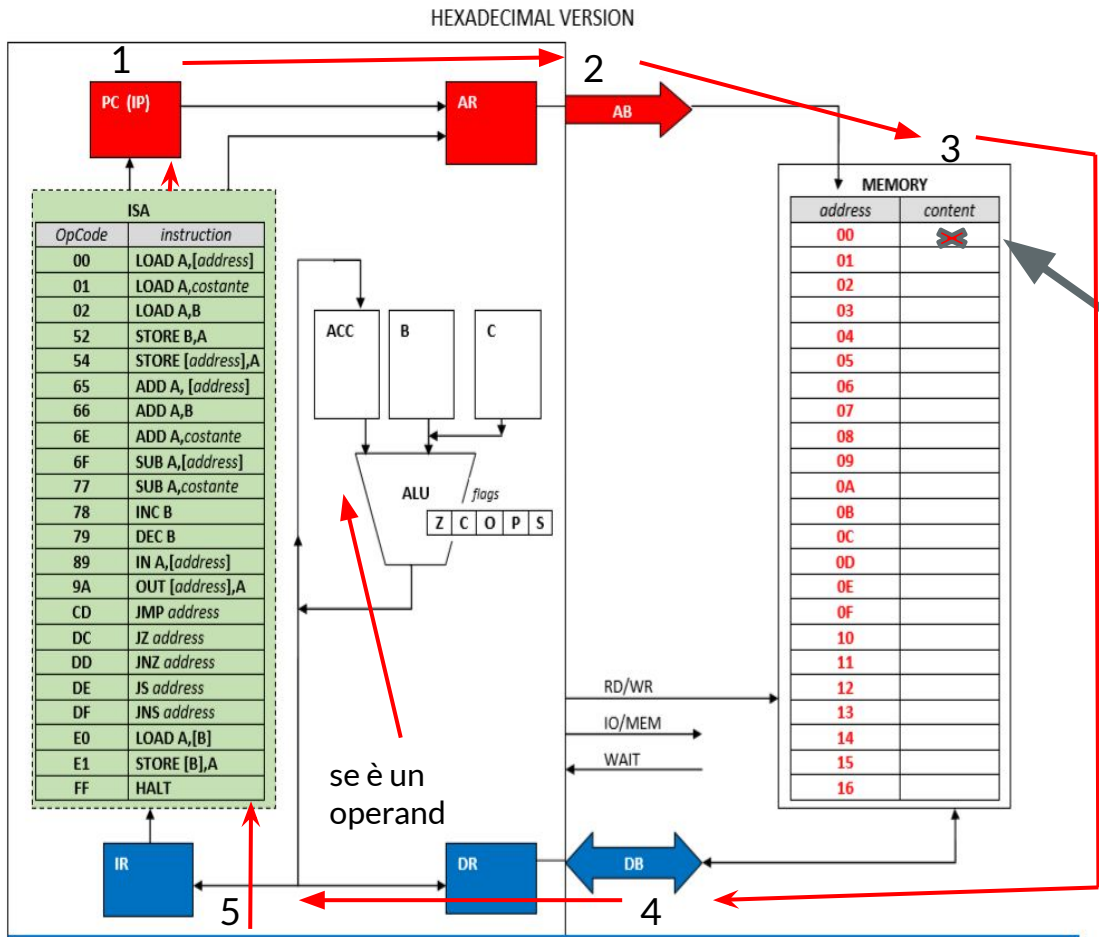


Z = zero flag  
C = carry flag  
O = overflow flag  
P = parity flag  
S = sign flag

in questo schema vengono riportati 5 flags; sono bit “indicatori” che permettono di eseguire correttamente certe istruzioni, per esempio i salti condizionati (JZ, JNZ, JS, JNS vd. pag 6); sono coinvolti dalle operazioni della ALU che li imposta a seconda del risultato delle operazioni stesse

**FLAG ‘Z’** questo è il flag che assume il valore 1 quando il risultato dell’operazione della ALU è uguale a 0, e viceversa diventa 0 quando il risultato è diverso da 0; istruzioni interessate: JZ, JNZ.

**FLAG ‘S’** questo flag assume il valore 1 quando il risultato dell’operazione A è un valore negativo, e viceversa, diventa 0 quando il risultato è maggiore di 0; istruzioni interessate: JS, JNS.



Questa immagine mostra il ciclo FDE all'interno del microprocessore.

**1**: Si parte dal program counter o PC; **2**: l'indirizzo qui contenuto viene posto sull'AB **3**: transita sino alla memoria-istruzioni dalla quale viene prelevato l'op-code o l'operand ivi contenuti. **4**: Tale contenuto viene trasferito sul DB, entra nel  $\mu P$  e **5**: depositato nell'IR (instruction register) se si tratta di un op-code; se si tratta di un operand andrà in un registro interno; nel frattempo il PC aumenta di 1.

# Le istruzioni

- **LOAD A,[ADDRESS]** carica nel registro A il contenuto della cella di memoria di indirizzo **ADDRESS**; simbolico:
- **LOAD A,COSTANTE** carica nel registro A il valore **COSTANTE**; simbolico:
- **LOAD A,B** carica il contenuto di B in A; simbolico:  $A \leftarrow B$
- **STORE B,A** copia il contenuto di A nel registro B; simbolico:
- **STORE [ADDRESS],A** carica il dato presente in A in una cella di memoria di indirizzo **ADDRESS**; simbolico:
- **ADD A,[ADDRESS]** somma il contenuto della cella di memoria di indirizzo **ADDRESS** al contenuto di A; imposta i flag; simbolico:
- **ADD A,B** somma il contenuto del registro B al registro A e pone il risultato in A; imposta i flag
- **ADD A,COSTANTE** somma ad A il valore **COSTANTE** e pone il risultato in A; imposta i flag
- **SUB A,[ADDRESS]** sottrae al registro A il contenuto della cella di memoria di indirizzo **ADDRESS**; pone il risultato in A; imposta i flag
- **SUB A,COSTANTE** sottrae al registro A il valore **COSTANTE**; pone il risultato in A; imposta i flag
- **INC B** incrementa di 1 il registro B ( $B \leftarrow B+1$ ); imposta i flag
- **DEC B** decrementa di 1 il registro B ( $B \leftarrow B-1$ ); imposta i flag
- **IN A,[ADDRESS]** preleva il dato dall'input device di indirizzo **ADDRESS** e lo deposita nel registro A
- **OUT [ADDRESS],A** manda all'output device di indirizzo **ADDRESS** il contenuto del registro A
- **JMP ADDRESS** salta all'indirizzo specificato; simbolico  $PC \leftarrow ADDRESS$
- **JZ ADDRESS** salta all'indirizzo specificato se il risultato è pari a 0; simbolico: IF flagZ==1  $PC \leftarrow ADDRESS$
- **JNZ ADDRESS** salta all'indirizzo specificato se il risultato è diverso da 0; simbolico: IF flagZ==0  $PC \leftarrow ADDRESS$
- **JS ADDRESS** salta all'indirizzo specificato se il risultato è minore di 0; simbolico: IF flagS==1  $PC \leftarrow ADDRESS$
- **JNS ADDRESS** salta all'indirizzo specificato se il risultato è maggiore di 0; simbolico: IF flagS==0  $PC \leftarrow ADDRESS$
- **LOAD A,[B]** carica in A il contenuto della cella di memoria il cui indirizzo è contenuto in B; simbolico:
- **STORE [B],A** carica il contenuto di A all'indirizzo contenuto in B (viene inserito in memoria); simbolico:
- **HALT** ferma il programma e interrompe i cicli FDE