

**Compito del 9 luglio 2008**

Un processore con architettura a singolo bus interno (bus dati a 16 bit, bus indirizzi a 24 bit) annovera tra le sue istruzioni la chiamata a subroutine condizionata “call on zero”

CALLZ op ,

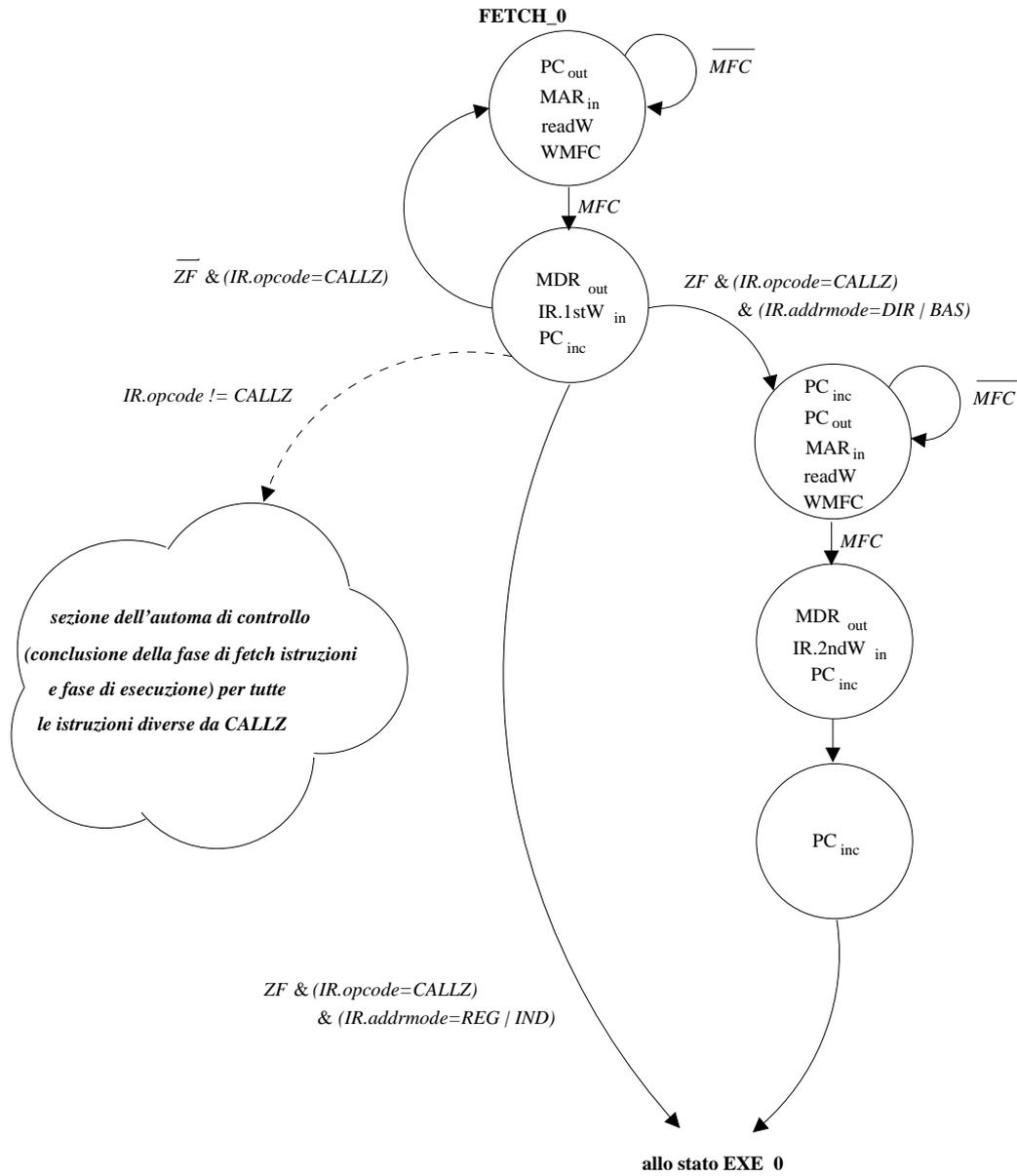
che esegue il salto sulla base del valore corrente del flag ZF. L’operando *op* può essere specificato attraverso uno dei seguenti modi di indirizzamento: (1) registro, (2) diretto, (3) indiretto di registro, (4) base e offset. Il registro coinvolto negli indirizzamenti è in ogni caso R7.

0. Fornire un esempio di chiamata con ciascuno dei modi di indirizzamento previsti, specificando l’esatto significato dell’istruzione nei vari casi.
1. Dare una codifica plausibile per l’istruzione, riportandone la lunghezza per ciascuna delle chiamate del punto precedente.
2. Disegnare l’hardware di parte operativa strettamente necessario al fetch ed all’esecuzione dell’istruzione.
3. Progettare l’automa di controllo per l’istruzione (inclusa la fase di fetch), rappresentandolo attraverso il diagramma degli stati.
4. Calcolare il numero di *cicli di bus* richiesti da ciascun modo di indirizzamento, distinguendo tra fase di fetch e fase di esecuzione. Fare lo stesso per i *cicli di macchina*, e stilare una graduatoria delle chiamate di cui al punto 0 basata sul tempo complessivo richiesto da fetch ed esecuzione.
5. Usare l’istruzione in un programma assembly, strutturato in chiamante e subroutine, che dimezzi tutti gli elementi pari di un vettore di 64 interi.

### # SOLUZIONE #

Riporto per ora il solo automa di controllo, che costituiva la parte più complessa del compito. L’opcode dell’istruzione nel campo “opcode” di IR è di 6 bit, mentre i due bit restanti del primo byte dell’istruzione sono occupati dal campo “addrmode”, che specifica – tra i 4 possibili: “REG”, “DIR”, “IND”, “BAS” – il modo di indirizzamento utilizzato. La parte operativa ha i registri PC e MAR incrementabili con una linea di controllo (cioè senza bisogno di utilizzare la ALU: sono registri contatori), e lo stack pointer SP decrementabile (idem). Per una corretta comprensione dell’automa, considerare che la lunghezza di codifica dell’istruzione CALLZ è di 1 byte per i modi di indirizzamento registro e indiretto (di registro), mentre è di 4 byte per i modi di indirizzamento diretto e base & offset. Infatti, in questi ultimi due casi bisogna specificare nell’istruzione un indirizzo (3 byte); nel caso di indirizzamento diretto, tale indirizzo punta direttamente alla routine a cui saltare, mentre nel caso di indirizzamento base & offset, esso contribuisce, con R7, alla determinazione dell’indirizzo della cella di memoria che contiene l’indirizzo di salto. Notare la necessità di disporre, per i trasferimenti da e per la memoria, di due diverse modalità: per byte singoli (*readB*, *writeB*), e per word di 2 byte (*readW*, *writeW*). Tale necessità è legata a dover trasferire sul bus dati (2 byte) indirizzi di 3 byte – tali trasferimenti sono realizzati con due cicli di bus consecutivi. (*Carlo Colombo*)

# I. Fetch istruzione



## II. Esecuzione

