

# OnesCounter

Alessandro Busnelli

August 28, 2025

## 1 Descrizione programma

Il programma presentato ha l'obiettivo di analizzare un numero intero e di determinare il numero di bit a valore 1 all'interno della sua rappresentazione binaria. Oltre al conteggio totale, vengono calcolati separatamente anche il numero di 1 nelle posizioni pari e dispari. Questo esercizio, apparentemente semplice, permette di toccare vari aspetti fondamentali della teoria dei calcolatori e dell'architettura MIPS.

Dal punto di vista teorico, il problema si collega al concetto di **rappresentazione binaria dei dati**, che è alla base del funzionamento di qualsiasi calcolatore digitale. Tutte le operazioni aritmetiche e logiche, infatti, avvengono su stringhe di bit, e la CPU mette a disposizione istruzioni specifiche per manipolare singoli bit o gruppi di bit, come le istruzioni logiche AND, OR, XOR e gli shift (srl, sll).

Il programma sfrutta queste istruzioni per **mascherare** (attraverso un AND con una costante) il bit meno significativo e per **scorrere** progressivamente tutti i bit del numero (tramite shift logici a destra). Questo è un tipico esempio di **bitwise manipulation**, ossia manipolazione a livello di singoli bit, che rappresenta una delle operazioni più veloci ed efficienti che la CPU può eseguire.

Dal punto di vista architetturale, il codice mette in luce l'uso dei **registri del processore MIPS** per scopi diversi:

- registri temporanei (\$t) usati per mascherature, shift e conteggi;
- registri salvati (\$s) per mantenere valori coerenti anche in caso di sotto-programmi;
- registri speciali (\$zero, \$v0) che permettono di rispettare le convenzioni MIPS.

Il flusso del programma riflette il funzionamento interno della CPU: ad ogni ciclo di clock, l'unità di controllo legge un'istruzione dal segmento di testo, decodifica il codice operativo (opcode), seleziona i registri e la ALU (Arithmetic Logic Unit) esegue l'operazione richiesta. Nel nostro caso, la ALU esegue operazioni logiche (AND) e aritmetiche (ADD, ADDI), mentre l'unità di controllo coordina i salti condizionati (beq, bne) che implementano i cicli del programma.

In questo modo, un problema di alto livello come il **conteggio degli 1** in un numero binario viene scomposto in istruzioni elementari MIPS, ognuna delle quali corrisponde ad un'operazione eseguibile dall'hardware. Questo riflette il principio dei **livelli di astrazione** nei calcolatori: dall'algoritmo logico, al linguaggio assembly, fino alle micro-operazioni interne al processore.