

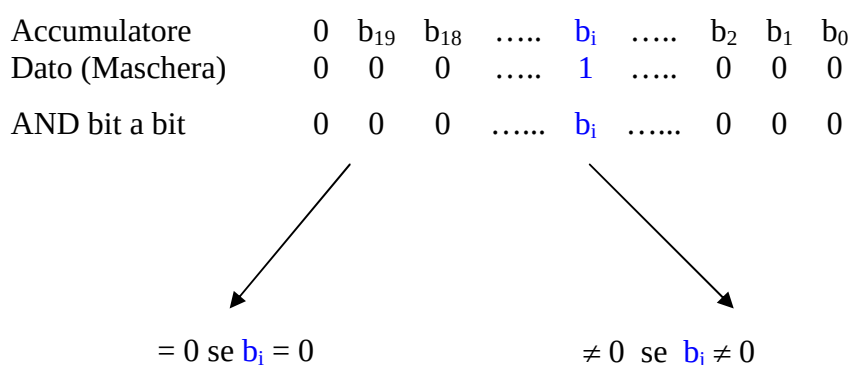
Problema 2.5

Scrivere un programma **SEC** per contare il numero di **1** nella stringa di bit immagazzinata nell'accumulatore, supponendo che il bit più a sinistra sia **0** (possono essere utilizzate delle costanti nella realizzazione del programma).

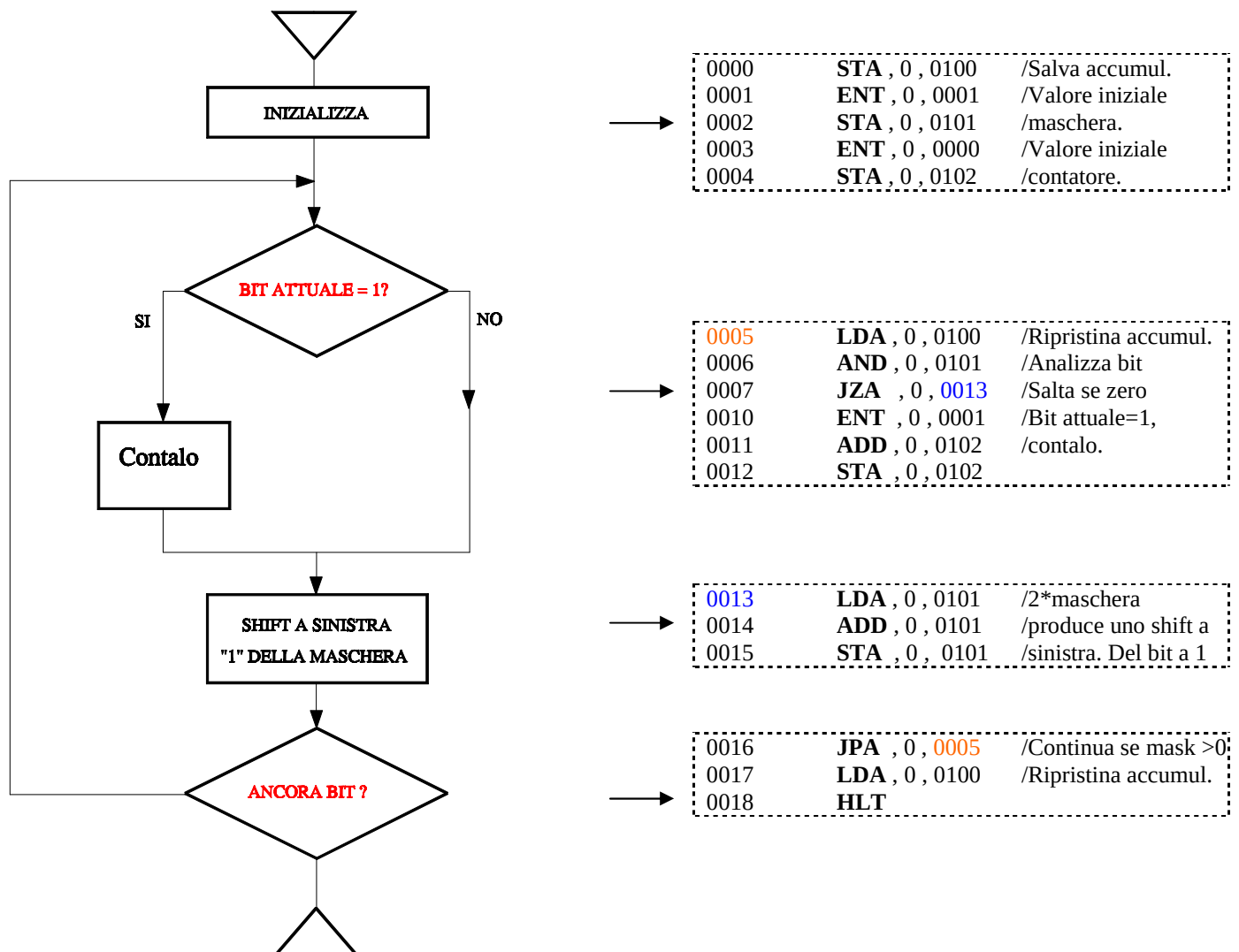
Soluzione

I bit da esaminare sono 20, da bit 0 a bit 19, dato che bit 20 (bit più a sinistra) è 0 per ipotesi.

Il criterio per determinare se un bit è diverso da zero, è quello di usare un dato che contiene "1" solo nella posizione del bit che si vuole valutare e di eseguire la seguente operazione logica:



L'algoritmo deve far scorrere il bit **1** della maschera dalla posizione meno significativa (0000001_8) alla più significativa ($\text{bit}_{19}=1$: 2000000_8). Ciò si ottiene raddoppiando il valore di **DATO** ad ogni passo. Quando **DATO** = 4000000_8 si ha la fine della procedura dato che il bit_{20} non deve essere valutato. Si osserva che il precedente valore della maschera è il primo **non positivo** (il 21-esimo bit è 1).



NB. Si è utilizzata la locazione di memoria di indirizzo **0100₈** per memorizzare temporaneamente la stringa di bit contenuta in accumulatore, dato che, questo registro deve essere usato per tutte le elaborazioni (“*salvato l’accumulatore*”). Durante la procedura, quando serve, e alla fine del programma, l’accumulatore viene ripristinato con la stringa iniziale (“*ripristino accumulatore*”)

Si è fatta inoltre implicitamente l’ipotesi che il calcolo di $2 \cdot \text{mask}$, nell’ultimo passo, sia eseguito come:

$$\begin{array}{r}
 0100000000000000000000 + \\
 0100000000000000000000 \\
 \hline
 1000000000000000000000
 \end{array}$$

In realtà la somma dei due valori positivi, interpretata come valore aritmetico, genera una situazione di overflow. Infatti il massimo valore positivo, in lunghezza 21 bit ed in codice modulo e segno, è $3777777_8 = 1048575_{10}$, mentre la somma della maschera con se stessa ha come modulo $100000000000000000000_2 = 4000000_8 = 1048576_{10}$ non contenibile in questa lunghezza (sarebbero

necessari 22 bit aggiungendo lo zero iniziale per codificare correttamente il risultato positivo). Nella lunghezza di 21 bit si legge, nel codice modulo e segno, il risultato errato **-0** che segnala la condizione di overflow.