4.3 Istruzioni di un linguaggio Assembler

Con linguaggio Assembler si intende un insieme di parole chiave costituite da pochi caratteri, in genere 3, che rappresentano le operazioni elementari svolte dal processore. Con esse è possibile costruire programmi contenenti le istruzioni per il processore.
Le parole chiave derivano dai termini inglesi corrispondenti alle operazioni.
Esempi di istruzioni Assembler sono:

LDA #X	(*Load*) Carica nel registro A il valore X
LDA X	(*Load*) Carica nel registro A il valore contenuto nella cella di memoria di indirizzo X
STA X	(*Store*) Memorizza il contenuto del registro A nella cella di memoria di indirizzo X
ADD	(*Add*) Aggiunge il contenuto del registro A al contenuto di un altro registro e memorizza il risultato nel registro A
SUB	(*Subtract*) Sottrae al contenuto del registro A il contenuto di un altro registro e memorizza il risultato nel registro A
JMP X	(*Jump*) Salta incondizionatamente all'istruzione identificata con l'etichetta X
JIZ X	(*Jump if zero*) Salta all'istruzione identificata con l'etichetta X se il valore del registro A è zero
HLT	End of assembler program

Per esempio, il seguente programma Assembler calcola il prodotto di due numeri che si trovano in memoria agli indirizzi 102 e 103 come somme successive e memorizza il prodotto all'indirizzo 104. Il processore utilizza solo due registri A e B.

```
        LDA #0          azzera il registro A
        STA 104memorizza il valore 0 nell'indirizzo 104
        LDA 102         carica nel registro A il contenuto dell'indirizzo 102
ripeti: JIZ [uscita]    salta all'etichetta uscita se il valore del registro A è zero
        LDA 104         carica nel registro A il contenuto dell'indirizzo 104
        LDB 103         carica nel registro B il contenuto dell'indirizzo 103
        ADD             somma il contenuto dei due registri
        STA 104memorizza il contenuto del registro A nell'indirizzo 104
        LDA 102         carica nel registro A il contenuto dell'indirizzo 102
        LDB #1          assegna il valore 1 al registro B
        SUB             sottrae il valore del registro B dal valore del registro A
        STA 102memorizza il contenuto del registro A nell'indirizzo 102
        JMP [ripeti]    salta all'etichetta ripeti per ripetere il gruppo di istruzioni
uscita: HLT
```

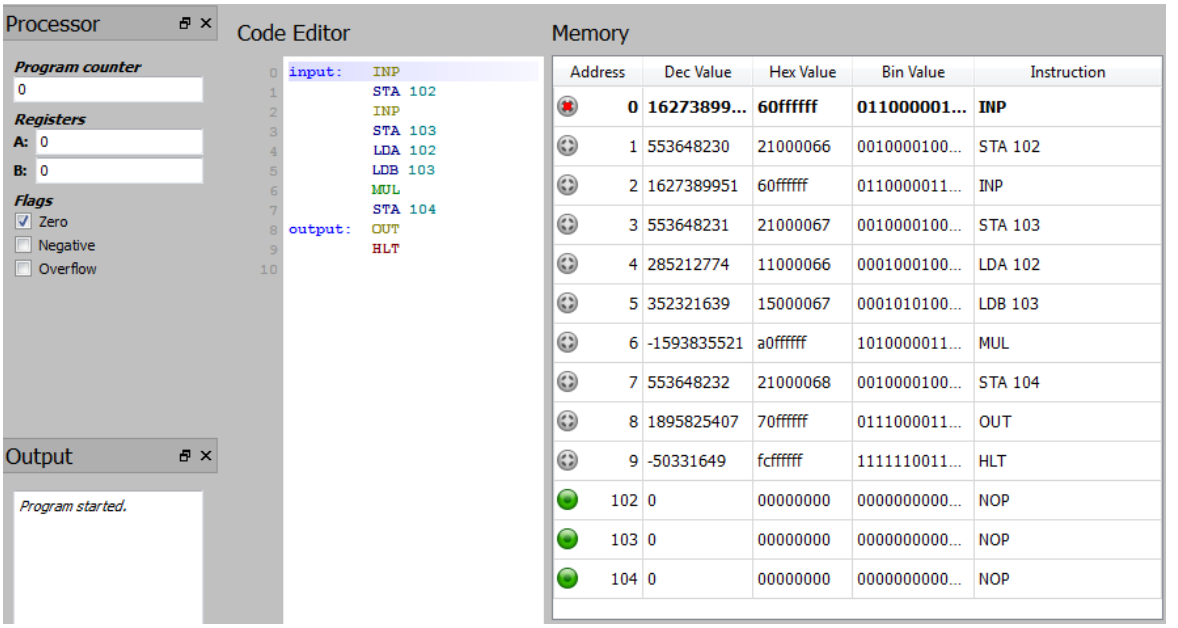
Supponiamo ora che il processore sia in grado di interpretare istruzioni più complesse del linguaggio Assembler quali il calcolo di una moltiplicazione o di una divisione. Aggiungiamo inoltre le istruzioni per acquisire dati dall'unità di input e per visualizzare un risultato sull'unità di output.
Alle istruzioni Assembler aggiungiamo le seguenti:

INP	(*Input*) Richiede un dato dall'unità di input e lo scrive nel registro A
OUT	(*Output*) Invia il contenuto del registro A all'unità di output
MUL	(*Multiply*) Moltiplica il contenuto del registro A per il contenuto di un altro registro e memorizza il risultato nel registro A
DIV	(*Divide*) Divide il contenuto del registro A per il contenuto di un altro registro e memorizza il risultato nel registro A

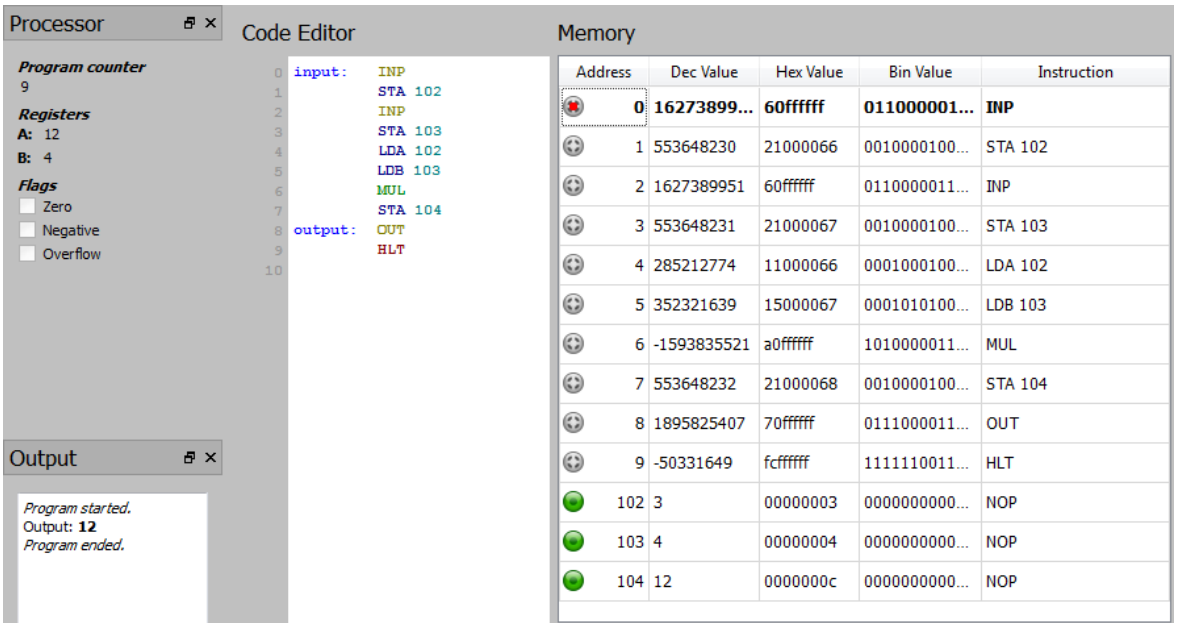
Il programma Assembler precedente, per il calcolo del prodotto di due numeri, può essere ora riscritto in modo più compatto:

```
input:  INP
        STA 102
        INP
        STA 103
        LDA 102
        LDB 103
        MUL
        STA 104
output: OUT
        HLT
```

Situazione iniziale



Situazione finale



Si osservi che le operazioni descritte, codificate con le istruzioni del linguaggio Assembler, rappresentano le attività elementari di un sistema di elaborazione secondo il modello della macchina di Von Neumann: acquisizione di dati dalle unità di input, trasferimento di dati dalla memoria al processore, calcoli sui registri del processore, uscita dei risultati sulle unità di output.