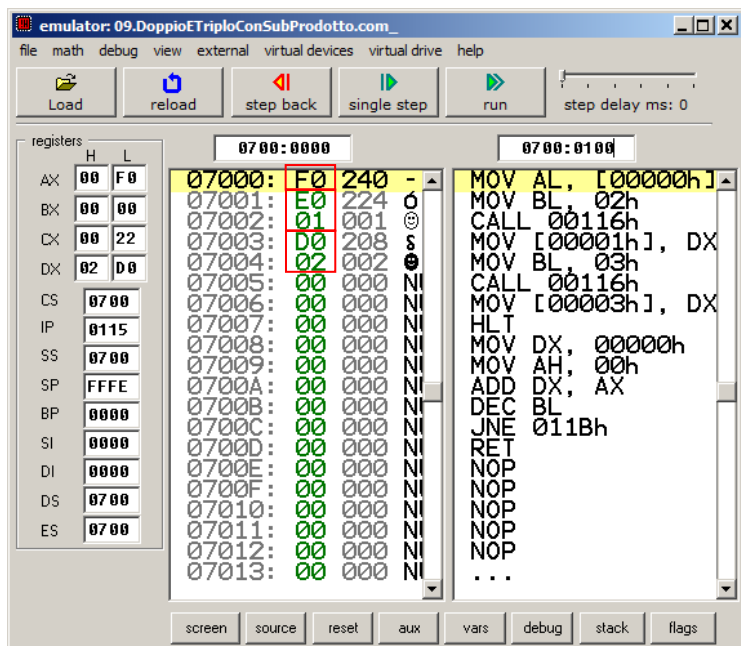


*Esempio Pilota:***DoppioETriploConSubProdotto**

Nella locazione di memoria [0000] è presente un Dato (1 byte). Calcolare il Doppio e il Triplo di tale Dato e, visto che il risultato potrebbe essere a 16 bit, porre il doppio nelle locazioni [0001]/[0002] e il triplo nelle locazioni [0003]/[0004]. Calcolare il Doppio e il Triplo richiamando due volte una stessa Subroutine che calcoli il Prodotto di due dati a 8 bit (1 byte) e restituisca un risultato a 16 bit (2 byte).

*Input*

locazione 0700:0000 (07000) con valore F0

Output

locazioni 0700:0001/0002 (07001/07002) con valore 01E0

locazioni 0700:0003/0004 (07003/07004) con valore 02D0

F0 x 2 = 01E0 (doppio di F0)

F0 x 3 = 02D0 (triplo di F0)

 Codice Assembly 8086

```

org 100h

; *** main program (programma principale) ...

; carica il dato dalla locazione 0000
MOV AL, [0000]

; calcola il doppio chiamando la sub Prodotto
; passando in AL il dato e in BL il valore 2
MOV BL, 02h
CALL Prodotto

; poni il doppio nelle locazioni 0001 e 0002
MOV [0001], DX

; calcola il triplo chiamando la sub Prodotto
; passando in AL il dato e in BL il valore 3
MOV BL, 03h
CALL Prodotto

; poni il triplo nelle locazioni 0003 e 0004
MOV [0003], DX

; ferma l'esecuzione
HLT

; *** subroutines ...

; la subroutine PRODOTTO riceve due dati
; nei registri AL e BL e restituisce, nel
; registro DX, il prodotto di AL x BL

; azzerà DX prima di calcolare il prodotto
PRODOTTO:  MOV DX, 0000h

; azzerà la parte alta di AX (AH) in modo
; da poter sommare AX (16 bit) anziché AL
MOV AH, 00h

; con il seguente ciclo, aggiungi a DX,
; il valore di AX (cioè di AL), per BL volte
; (ALxBL = AL+AL+AL+AL ... per BL volte)

RIPETI:    ADD DX, AX
          DEC BL
          JNZ RIPETI

; ritorna al programma principale
; (recupera dallo STACK l'indirizzo
; dell'istruzione SUCCESSIVA alla CALL)
RET

```