; Calling assembly from C
; Passing argument convention
; arg1  A4, return result
; arg2  B4
; arg3  A6
; return address  B3

.title  "dotp.sa"
.def    dotp
.sect   "code"
dotp:   .proc A4, B4, A6, B3
      .reg  p_m, m, p_n, n, prod, sum, count
      mv    A4, p_m    ;p_m now has the address of m
      mv    B4, p_n    ;p_n now has the address of n
      mv    A6, count  ;count = the number of iterations
      mvk   0, sum     ;sum=0
      loop: .trip 40    ;minimum 40 iterations through loop
      ldh   *p_m++, m   ;load element of m, postincrement pointer
      ldh   *p_n++, n   ;load element of n, postincrement pointer
      mpy   m, n, prod  ;prod=m*n
      add   prod, sum, sum  ;sum += prod
      [count] sub    count, 1, count  ;decrement counter
      [count] b       loop  ;branch back to loop
      mv    sum, A4    ;store result in return register A4
.endproc A4, B3

Figure 6-7: Linear assembly code for dot product example