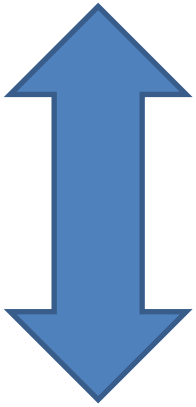
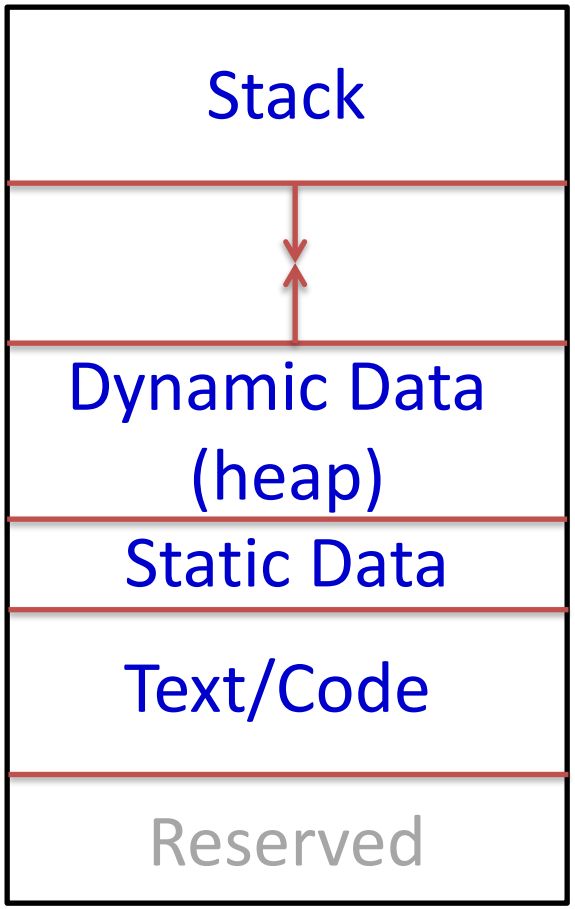
68000 Storage Layout (Software Convention)

High Memory



Low Memory



Stack and dynamic area grow towards one another to maximize storage use before collision

Implementing Functions

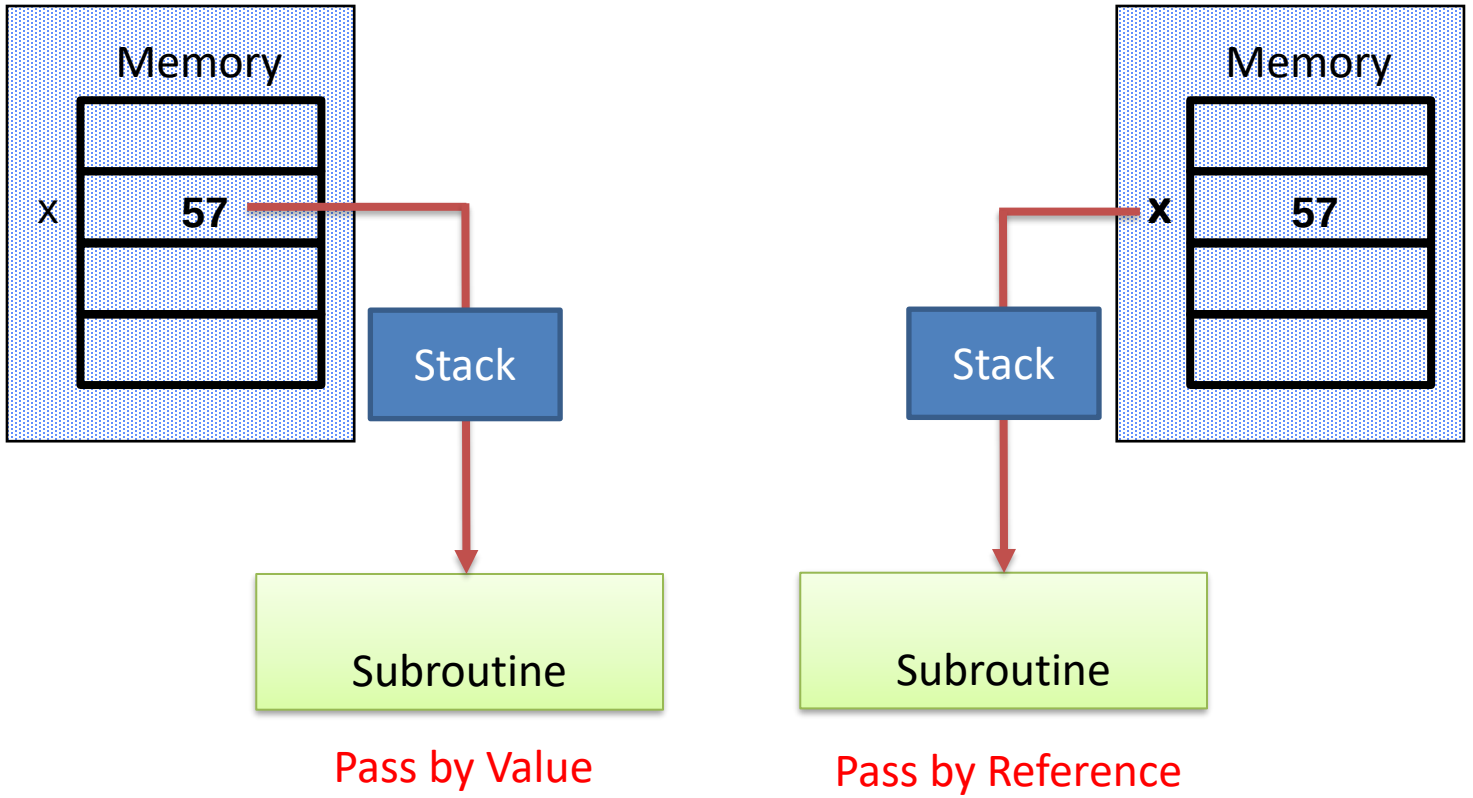
- Functions are required for structured programming
  - Aka: procedures, methods, subroutines…

- Implementing functions in assembly requires several things to be done
  - Parameters must be passed in and return values passed out
    - Registers
    - Stack
  - Memory must be set aside for local variables
    - Heap
    - Stack
  - Execution must continue after the call
    - Stack

Passing Parameters on the Stack

- Both the “caller” and the “callee” must know the number, order and type of parameters being passed
  - In C we use a function prototype
    - void Foo (int a, char b, int *c);
  - In assembler, you must take care of this yourself
  - C passes parameters right to left
- After returning from a subroutine
  - calling code must remove any parameters from the stack
- Parameters may be passed by value or passed by reference
  - Pass-by-value:
    - copy of parameter is passed
  - Pass-by-reference:
    - address of parameter is passed

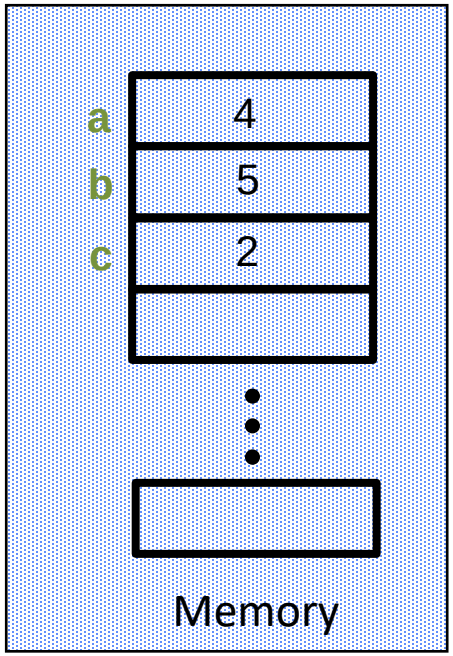
Pass by Value/Reference



C Function – Pass by Value

- Consider the following code

```
void foo() {
    short int a=4, b=5, c=2;
    register int product;

    product = mul3(a,b,c);
}


int mul3(short int x, short int y,
         short int z) {

    return (x*y*z);
}
```

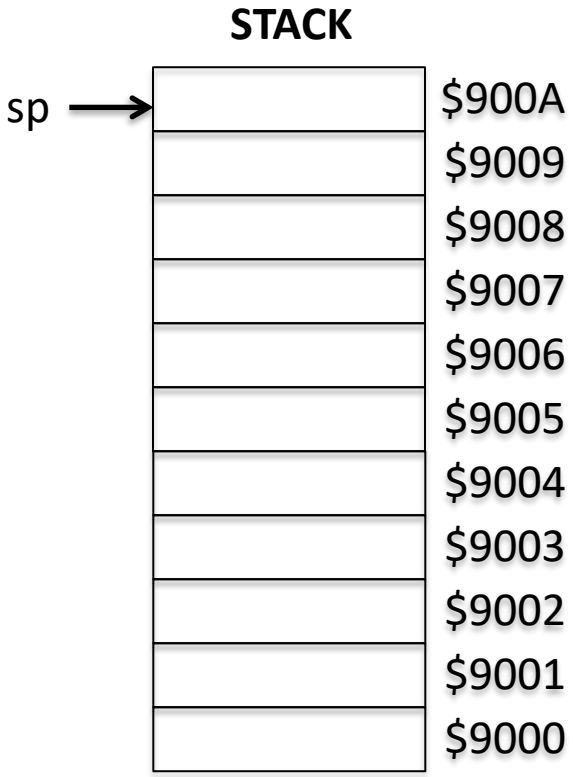


Assembler Code – Pass by Value

```
* foo() calling code
        move.w      c,-(sp)       ;push 1st parameter
        move.w      b,-(sp)       ;push 2nd parameter
        move.w      a,-(sp)       ;push 3rd parameter
        jsr         mul3          ;call subroutine
        lea         6(sp),sp      ;remove parameters


* mul3() multiplies 3 short ints and returns result in d0
mul3    move.w      4(sp),d0      ;d0 = a
        muls        6(sp),d0      ;d0 = a * b
        muls        8(sp),d0      ;d0 = a * b * c
        rts                       ;return

        org         $A000         ;variables
a       dc.w        4
b       dc.w        5
c       dc.w        2
```

Trace

```
******** foo() **********
8000            move.w   c,-(sp)
8006            move.w   b,-(sp)
800C            move.w   a,-(sp)
8012            jsr      mul3
8018            lea      6(sp),sp
801E

******** mul3() **********
8022    mul3    move.w   4(sp),d0
8026            muls     6(sp),d0
802A            muls     8(sp),d0
802E            rts

                org             $A000
A000    a       dc.w            4
A002    b       dc.w            5
A004    c       dc.w            2
```

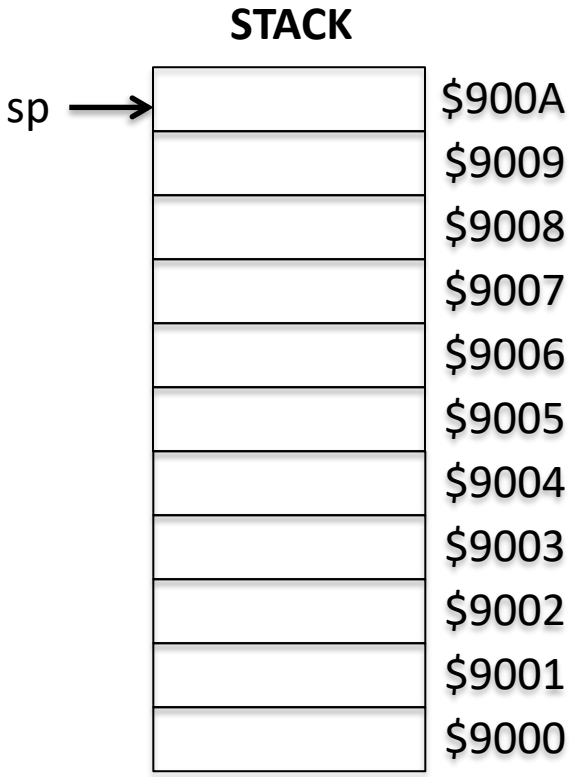


A7/SP	00	00	90	0A

d0	XX	XX	XX	XX

Trace

```
******** foo() **********
8000            move.w    c,-(sp)
8006            move.w    b,-(sp)
800C            move.w    a,-(sp)
8012            jsr       mul3
8018            lea       6(sp),sp
801E

******** mul3() **********
8022    mul3    move.w    4(sp),d0
8026            muls      6(sp),d0
802A            muls      8(sp),d0
802E            rts

                org             $A000
A000    a       dc.w            4
A002    b       dc.w            5
A004    c       dc.w            2
```

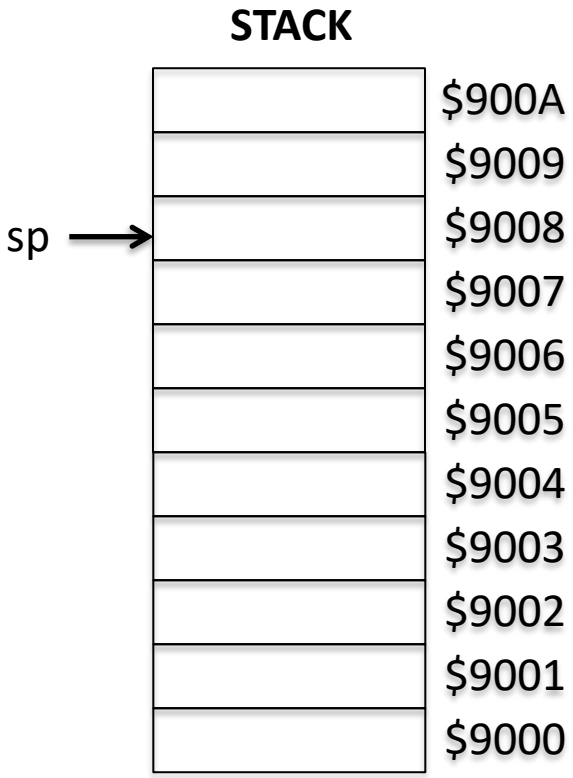


A7/SP	00	00	90	0A

d0	XX	XX	XX	XX

Trace

```
******** foo() **********
8000            move.w   c,-(sp)
8006            move.w   b,-(sp)
800C            move.w   a,-(sp)
8012            jsr      mul3
8018            lea      6(sp),sp
801E

******** mul3() **********
8022    mul3    move.w   4(sp),d0
8026            muls     6(sp),d0
802A            muls     8(sp),d0
802E            rts

                org             $A000
A000    a       dc.w            4
A002    b       dc.w            5
A004    c       dc.w            2
```

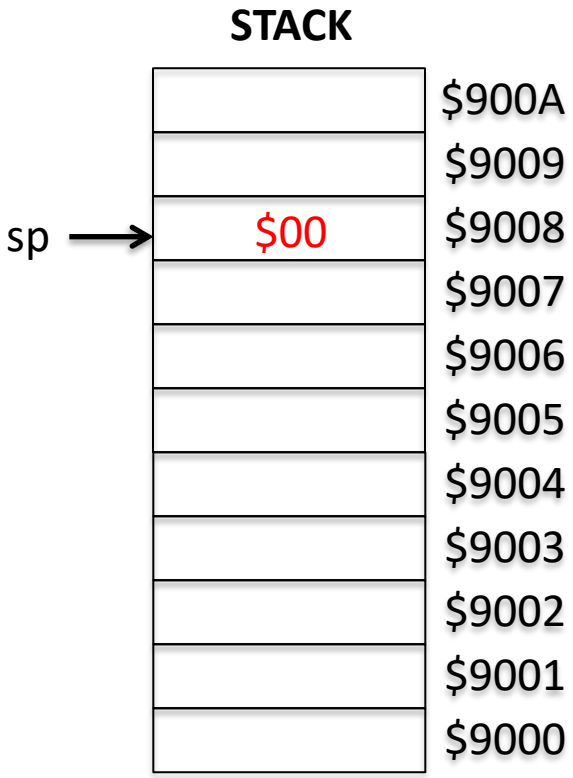


A7/SP	00	00	90	08

d0	XX	XX	XX	XX

Trace

```
******** foo() **********
8000            move.w   c,-(sp)
8006            move.w   b,-(sp)
800C            move.w   a,-(sp)
8012            jsr      mul3
8018            lea      6(sp),sp
801E

******** mul3() **********
8022    mul3    move.w   4(sp),d0
8026            muls     6(sp),d0
802A            muls     8(sp),d0
802E            rts

                org             $A000
A000    a       dc.w            4
A002    b       dc.w            5
A004    c       dc.w            2
```

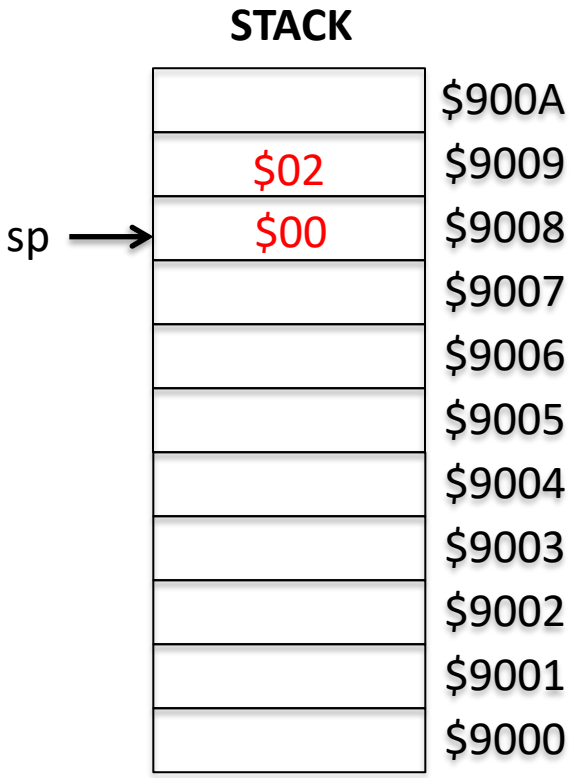


A7/SP	00	00	90	08

d0	XX	XX	XX	XX

Trace

```
******** foo() **********
8000            move.w   c,-(sp)
8006            move.w   b,-(sp)
800C            move.w   a,-(sp)
8012            jsr      mul3
8018            lea      6(sp),sp
801E

******** mul3() **********
8022    mul3    move.w   4(sp),d0
8026            muls     6(sp),d0
802A            muls     8(sp),d0
802E            rts

                org             $A000
A000    a       dc.w            4
A002    b       dc.w            5
A004    c       dc.w            2
```

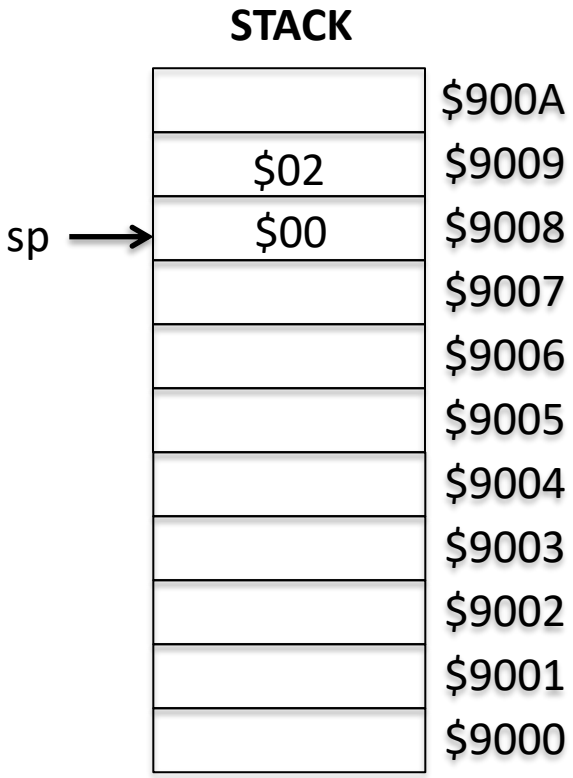


A7/SP | 00 | 00 | 90 | 08

d0 | XX | XX | XX | XX

Trace

```
******** foo() **********
8000            move.w   c,-(sp)
8006            move.w   b,-(sp)
800C            move.w   a,-(sp)
8012            jsr      mul3
8018            lea      6(sp),sp
801E

******** mul3() **********
8022    mul3    move.w   4(sp),d0
8026            muls     6(sp),d0
802A            muls     8(sp),d0
802E            rts

                org             $A000
A000    a       dc.w            4
A002    b       dc.w            5
A004    c       dc.w            2
```

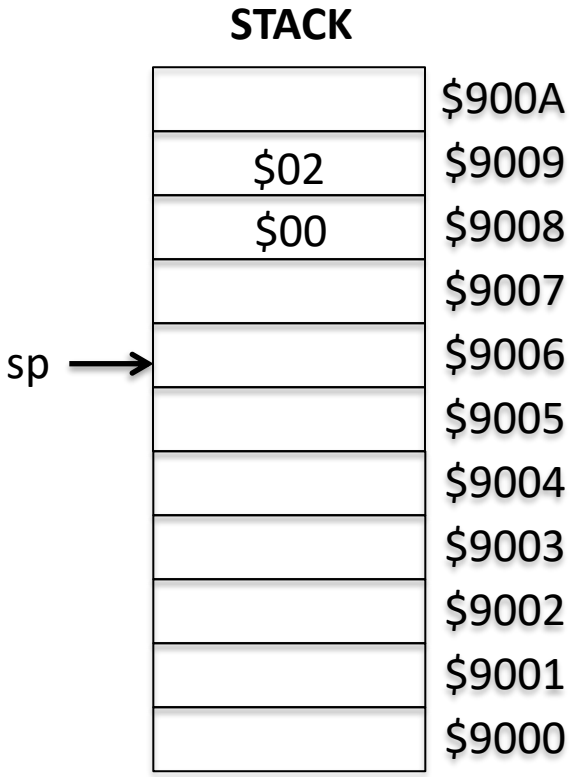


A7/SP | 00 | 00 | 90 | 08

d0 | XX | XX | XX | XX

Trace

```
******** foo() **********
8000            move.w   c,-(sp)
8006            move.w   b,-(sp)
800C            move.w   a,-(sp)
8012            jsr      mul3
8018            lea      6(sp),sp
801E

******** mul3() **********
8022    mul3    move.w   4(sp),d0
8026            muls     6(sp),d0
802A            muls     8(sp),d0
802E            rts

                org             $A000
A000    a       dc.w            4
A002    b       dc.w            5
A004    c       dc.w            2
```

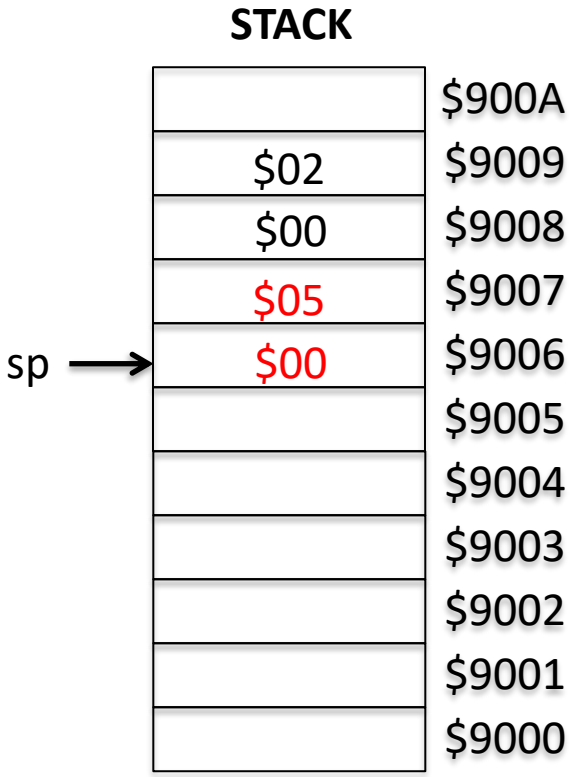


A7/SP | 00 | 00 | 90 | 06

d0 | XX | XX | XX | XX

Trace

```
******** foo() **********
8000            move.w   c,-(sp)
8006            move.w   b,-(sp)
800C            move.w   a,-(sp)
8012            jsr      mul3
8018            lea      6(sp),sp
801E

******** mul3() **********
8022    mul3    move.w   4(sp),d0
8026            muls     6(sp),d0
802A            muls     8(sp),d0
802E            rts

                org             $A000
A000    a       dc.w            4
A002    b       dc.w            5
A004    c       dc.w            2
```

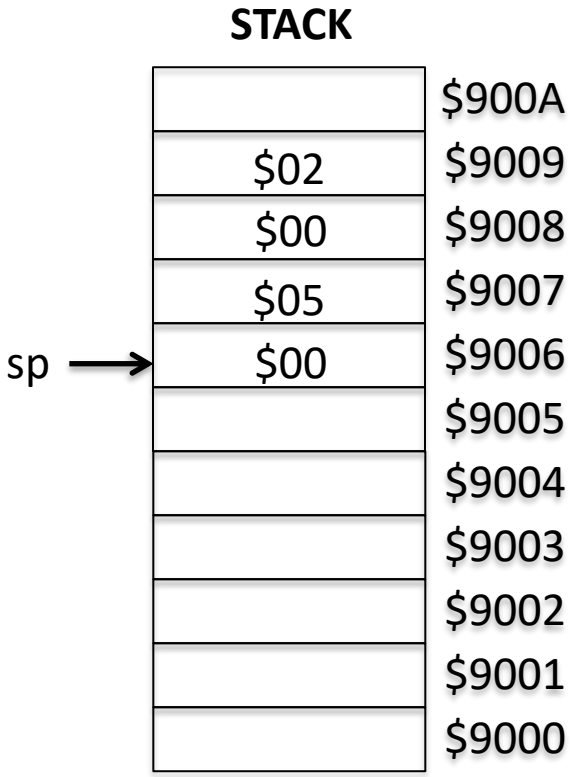


A7/SP | 00 | 00 | 90 | 06

d0 | XX | XX | XX | XX

Trace

```
******** foo() **********
8000            move.w    c,-(sp)
8006            move.w    b,-(sp)
800C            move.w    a,-(sp)
8012            jsr       mul3
8018            lea       6(sp),sp
801E

******** mul3() **********
8022    mul3    move.w    4(sp),d0
8026            muls      6(sp),d0
802A            muls      8(sp),d0
802E            rts

                org               $A000
A000    a       dc.w              4
A002    b       dc.w              5
A004    c       dc.w              2
```

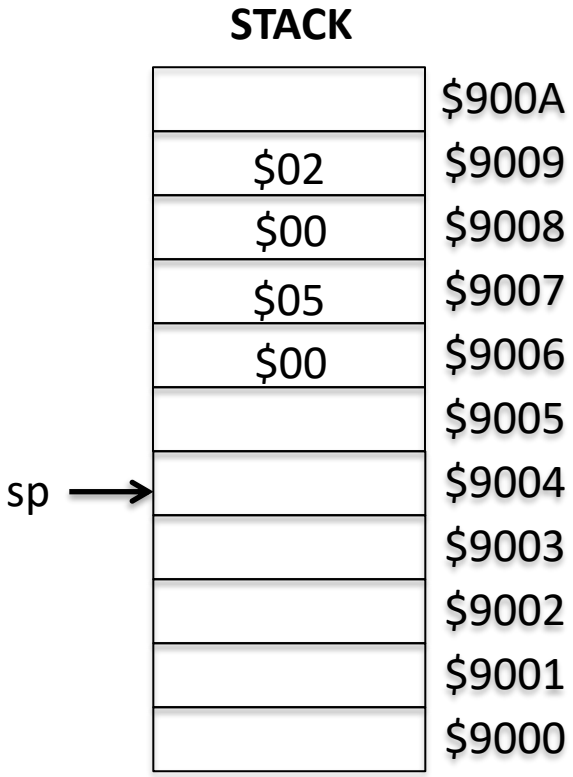


A7/SP	00	00	90	06

d0	XX	XX	XX	XX

Trace

```
******** foo() **********
8000            move.w    c,-(sp)
8006            move.w    b,-(sp)
800C            move.w    a,-(sp)
8012            jsr       mul3
8018            lea       6(sp),sp
801E

******** mul3() **********
8022    mul3    move.w    4(sp),d0
8026            muls      6(sp),d0
802A            muls      8(sp),d0
802E            rts

                org             $A000
A000    a       dc.w            4
A002    b       dc.w            5
A004    c       dc.w            2
```

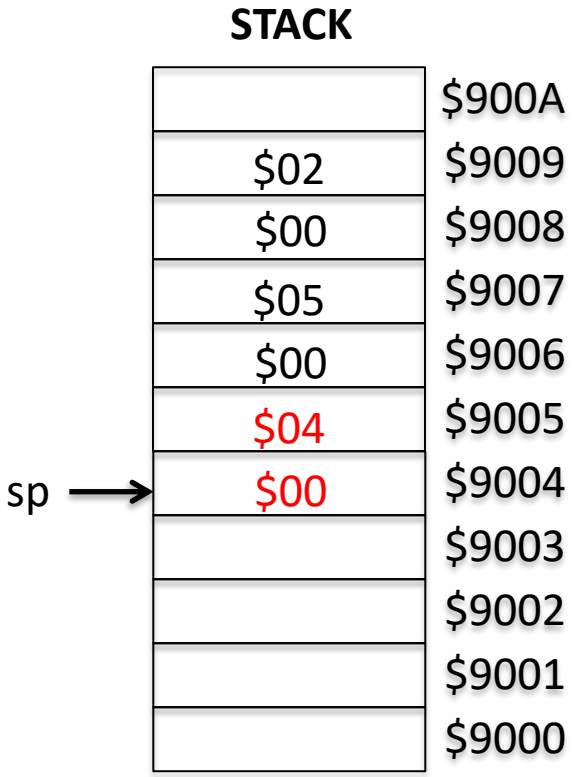


A7/SP | 00 | 00 | 90 | 04

d0 | XX | XX | XX | XX

Trace

```
******** foo() **********
8000            move.w    c,-(sp)
8006            move.w    b,-(sp)
800C            move.w    a,-(sp)
8012            jsr       mul3
8018            lea       6(sp),sp
801E

******** mul3() **********
8022    mul3    move.w    4(sp),d0
8026            muls      6(sp),d0
802A            muls      8(sp),d0
802E            rts

                org             $A000
A000    a       dc.w            4
A002    b       dc.w            5
A004    c       dc.w            2
```

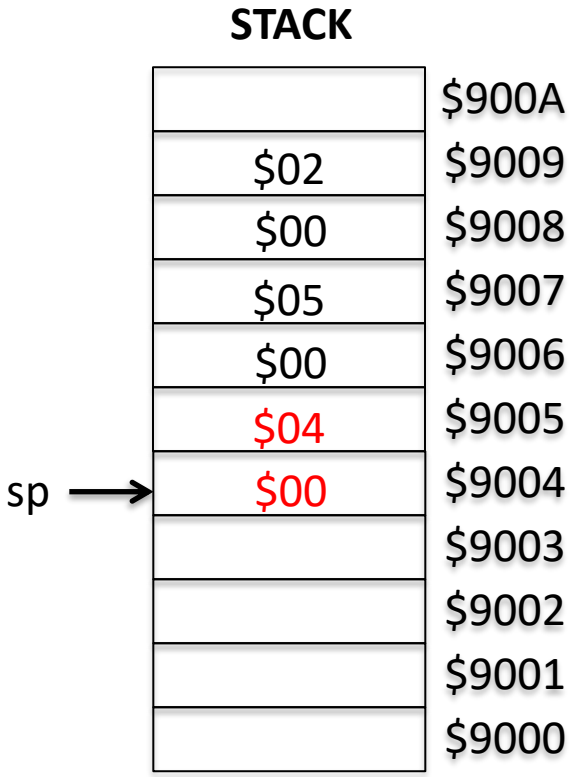


A7/SP | 00 | 00 | 90 | 04

d0 | XX | XX | XX | XX

Trace

```
******** foo() **********
8000            move.w    c,-(sp)
8006            move.w    b,-(sp)
800C            move.w    a,-(sp)
8012            jsr       mul3
8018            lea       6(sp),sp
801E

******** mul3() **********
8022    mul3    move.w    4(sp),d0
8026            muls      6(sp),d0
802A            muls      8(sp),d0
802E            rts

                org             $A000
A000    a       dc.w            4
A002    b       dc.w            5
A004    c       dc.w            2
```

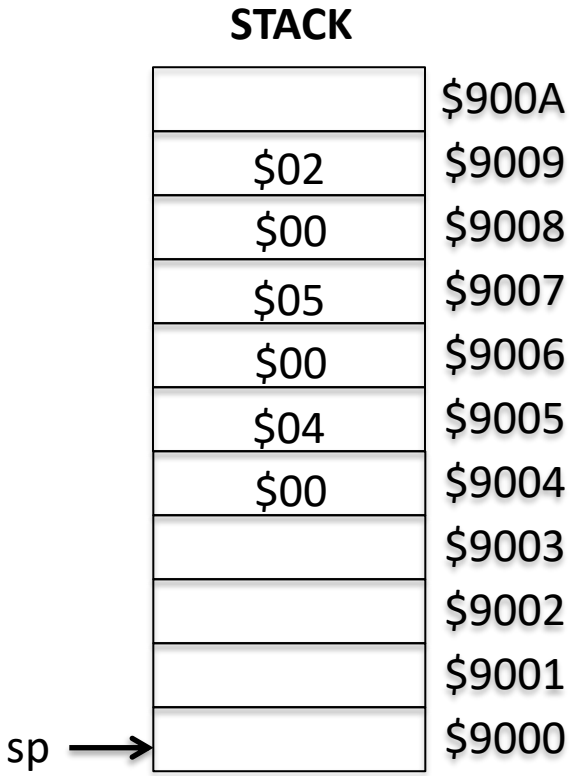


A7/SP | 00 | 00 | 90 | 04

d0 | XX | XX | XX | XX

Trace

```
******** foo() **********
8000            move.w    c,-(sp)
8006            move.w    b,-(sp)
800C            move.w    a,-(sp)
8012            jsr       mul3
8018            lea       6(sp),sp
801E

******** mul3() **********
8022    mul3    move.w    4(sp),d0
8026            muls      6(sp),d0
802A            muls      8(sp),d0
802E            rts

                org             $A000
A000    a       dc.w            4
A002    b       dc.w            5
A004    c       dc.w            2
```

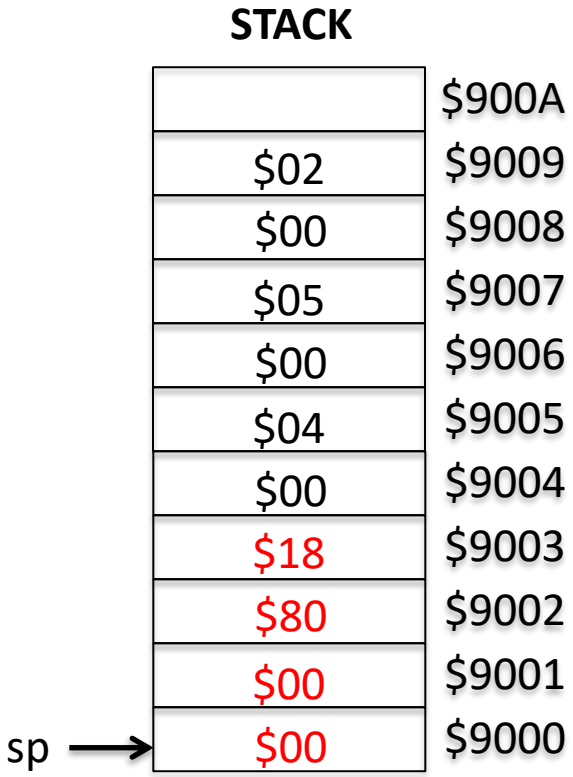


A7/SP | 00 | 00 | 90 | 00

d0 | XX | XX | XX | XX

Trace

```
******** foo() **********
8000            move.w    c,-(sp)
8006            move.w    b,-(sp)
800C            move.w    a,-(sp)
8012            jsr       mul3
8018            lea       6(sp),sp
801E

******** mul3() **********
8022    mul3    move.w    4(sp),d0
8026            muls      6(sp),d0
802A            muls      8(sp),d0
802E            rts

                org             $A000
A000    a       dc.w            4
A002    b       dc.w            5
A004    c       dc.w            2
```

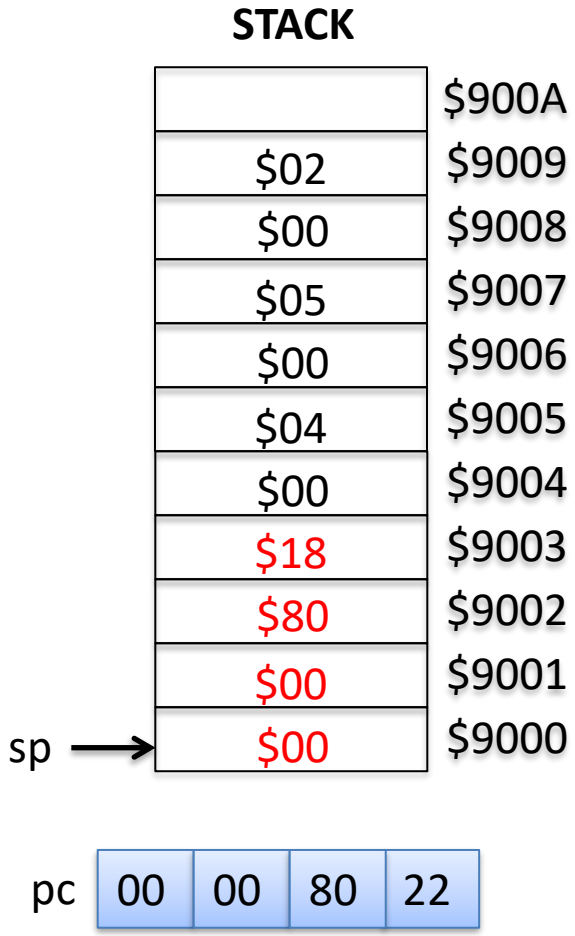


A7/SP | 00 | 00 | 90 | 00

d0 | XX | XX | XX | XX

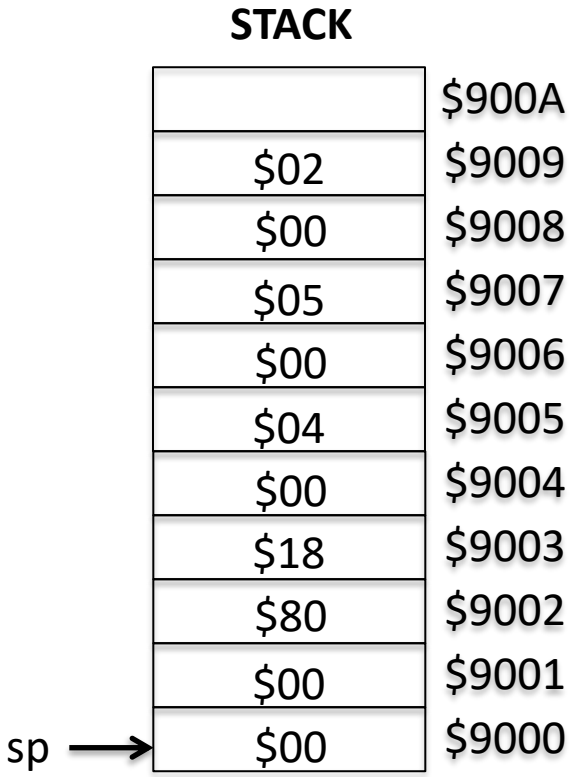
Trace

```
******** foo() **********
8000            move.w    c,-(sp)
8006            move.w    b,-(sp)
800C            move.w    a,-(sp)
8012            jsr       mul3
8018            lea       6(sp),sp
801E

******** mul3() **********
8022    mul3    move.w    4(sp),d0
8026            muls      6(sp),d0
802A            muls      8(sp),d0
802E            rts

                org             $A000
A000    a       dc.w            4
A002    b       dc.w            5
A004    c       dc.w            2
```



Trace

```
******** foo() **********
8000            move.w    c,-(sp)
8006            move.w    b,-(sp)
800C            move.w    a,-(sp)
8012            jsr       mul3
8018            lea       6(sp),sp
801E

******** mul3() **********
8022    mul3    move.w    4(sp),d0
8026            muls      6(sp),d0
802A            muls      8(sp),d0
802E            rts

                org             $A000
A000    a       dc.w            4
A002    b       dc.w            5
A004    c       dc.w            2
```

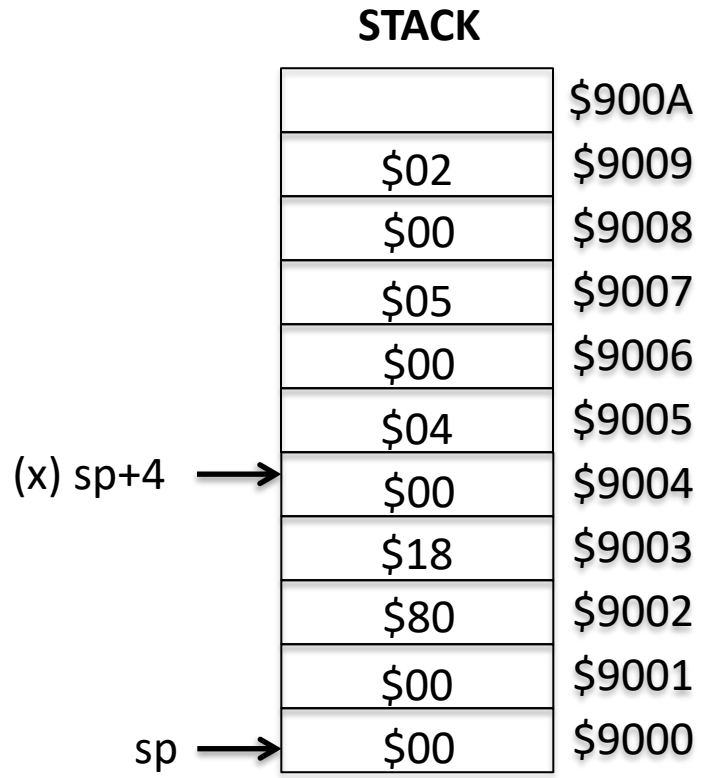


A7/SP | 00 | 00 | 90 | 00

d0 | XX | XX | XX | XX

Trace

```
******** foo() **********
8000            move.w    c,-(sp)
8006            move.w    b,-(sp)
800C            move.w    a,-(sp)
8012            jsr       mul3
8018            lea       6(sp),sp
801E

******** mul3() **********
8022    mul3    move.w    4(sp),d0
8026            muls      6(sp),d0
802A            muls      8(sp),d0
802E            rts

                org             $A000
A000    a       dc.w            4
A002    b       dc.w            5
A004    c       dc.w            2
```

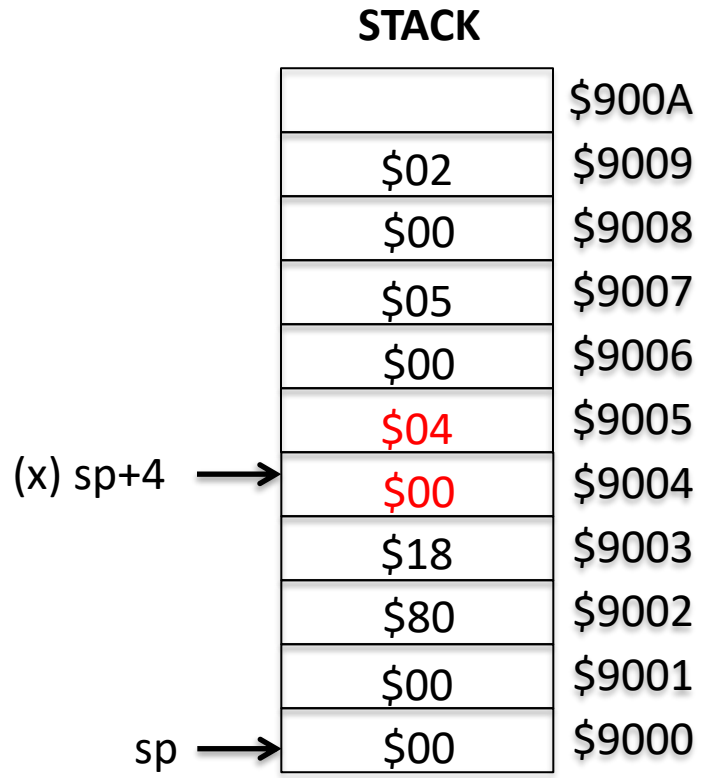


A7/SP | 00 | 00 | 90 | 00

d0 | XX | XX | XX | XX

Trace

```
******** foo() **********
8000            move.w    c,-(sp)
8006            move.w    b,-(sp)
800C            move.w    a,-(sp)
8012            jsr       mul3
8018            lea       6(sp),sp
801E

******** mul3() **********
8022    mul3    move.w    4(sp),d0
8026            muls      6(sp),d0
802A            muls      8(sp),d0
802E            rts

                org             $A000
A000    a       dc.w            4
A002    b       dc.w            5
A004    c       dc.w            2
```

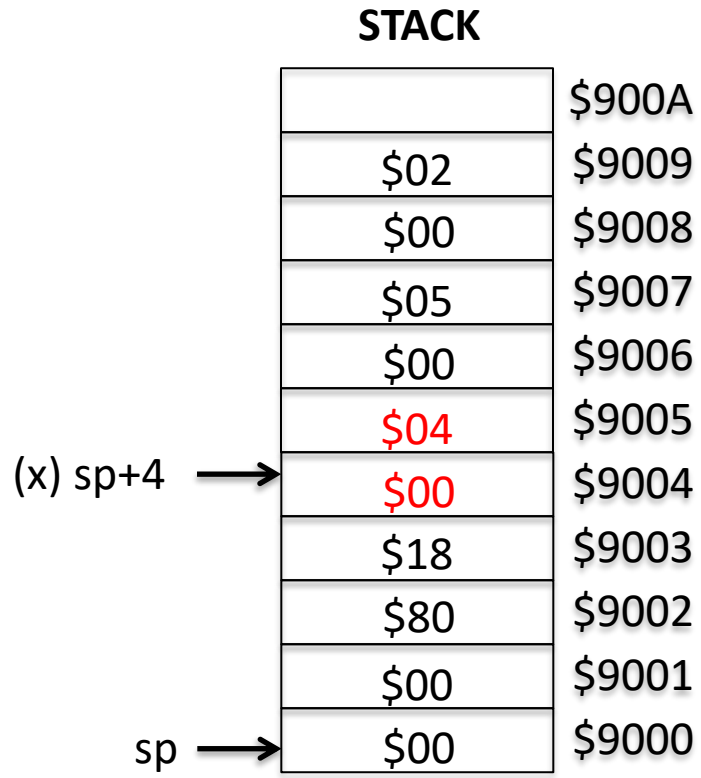


A7/SP	00	00	90	00

d0	XX	XX	00	04

Trace

```
******** foo() **********
8000            move.w    c,-(sp)
8006            move.w    b,-(sp)
800C            move.w    a,-(sp)
8012            jsr       mul3
8018            lea       6(sp),sp
801E

******** mul3() **********
8022    mul3    move.w    4(sp),d0
8026            muls      6(sp),d0
802A            muls      8(sp),d0
802E            rts

                org             $A000
A000    a       dc.w            4
A002    b       dc.w            5
A004    c       dc.w            2
```

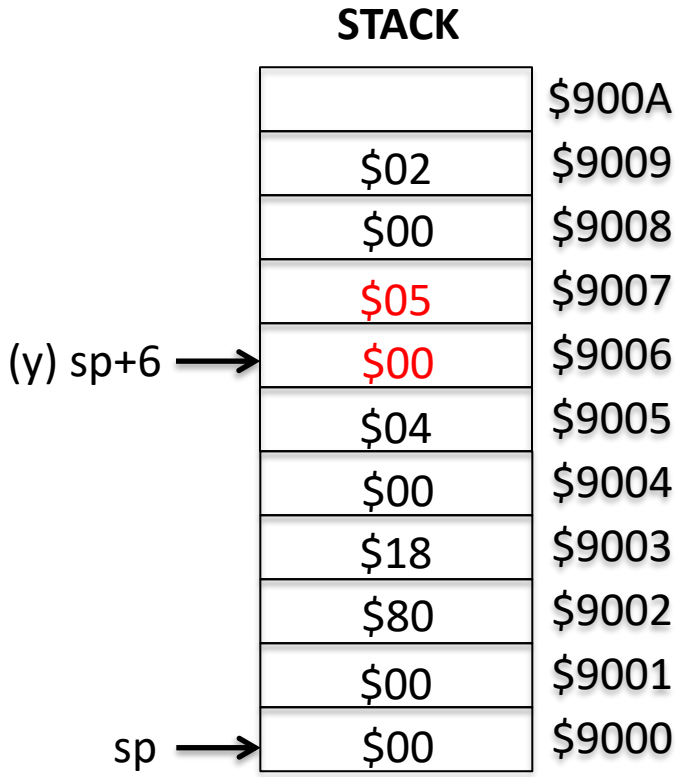


A7/SP	00	00	90	00

d0	XX	XX	00	04

Trace

```
******** foo() **********
8000            move.w    c,-(sp)
8006            move.w    b,-(sp)
800C            move.w    a,-(sp)
8012            jsr       mul3
8018            lea       6(sp),sp
801E

******** mul3() **********
8022    mul3    move.w    4(sp),d0
8026            muls      6(sp),d0
802A            muls      8(sp),d0
802E            rts

                org             $A000
A000    a       dc.w            4
A002    b       dc.w            5
A004    c       dc.w            2
```

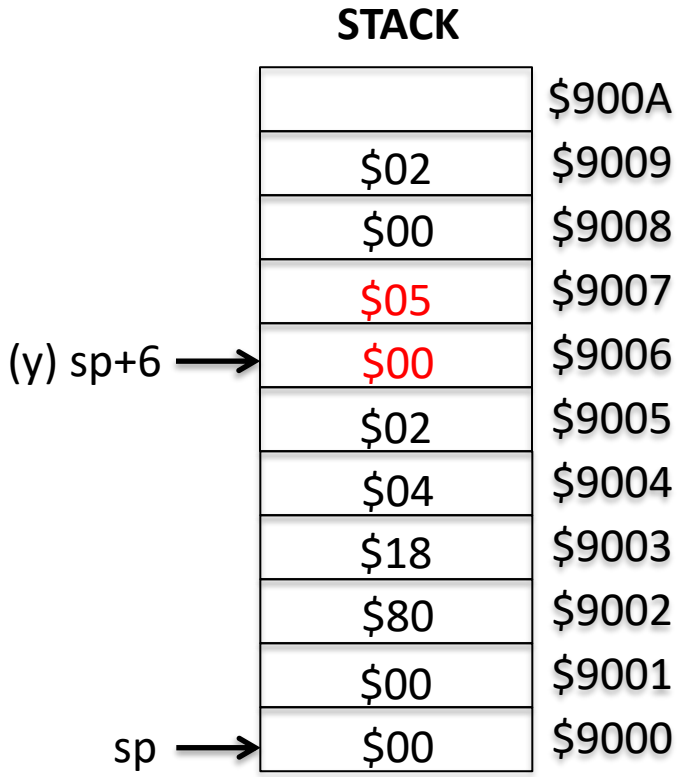


A7/SP	00	00	90	00

d0	XX	XX	00	04

Trace

```
******** foo() **********
8000            move.w    c,-(sp)
8006            move.w    b,-(sp)
800C            move.w    a,-(sp)
8012            jsr       mul3
8018            lea       6(sp),sp
801E

******** mul3() **********
8022    mul3    move.w    4(sp),d0
8026            muls      6(sp),d0
802A            muls      8(sp),d0
802E            rts

                org             $A000
A000    a       dc.w            4
A002    b       dc.w            5
A004    c       dc.w            2
```

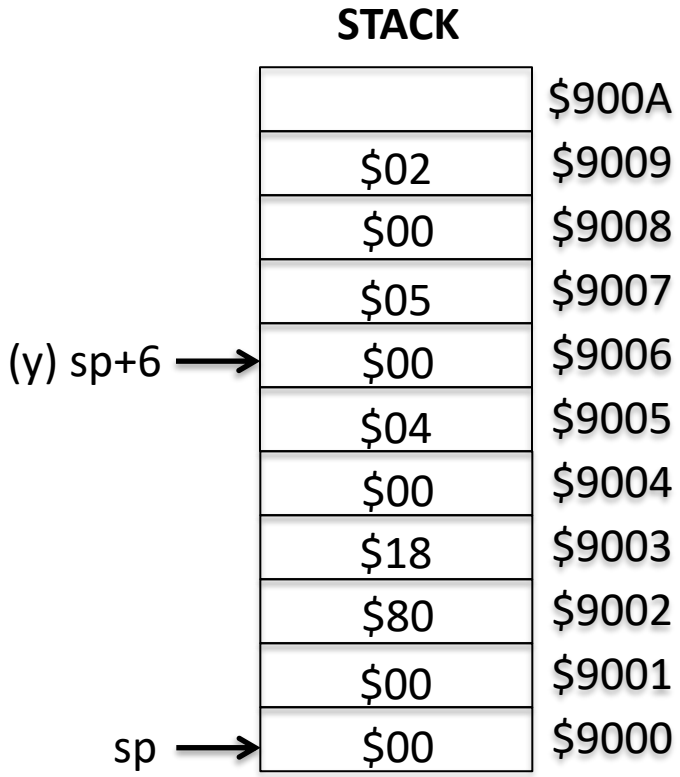


A7/SP | 00 | 00 | 90 | 00

d0 | 00 | 00 | 00 | 14

Trace

```
******** foo() **********
8000            move.w   c,-(sp)
8006            move.w   b,-(sp)
800C            move.w   a,-(sp)
8012            jsr      mul3
8018            lea      6(sp),sp
801E

******** mul3() **********
8022    mul3    move.w   4(sp),d0
8026            muls     6(sp),d0
802A            muls     8(sp),d0
802E            rts


                org             $A000
A000    a       dc.w            4
A002    b       dc.w            5
A004    c       dc.w            2
```

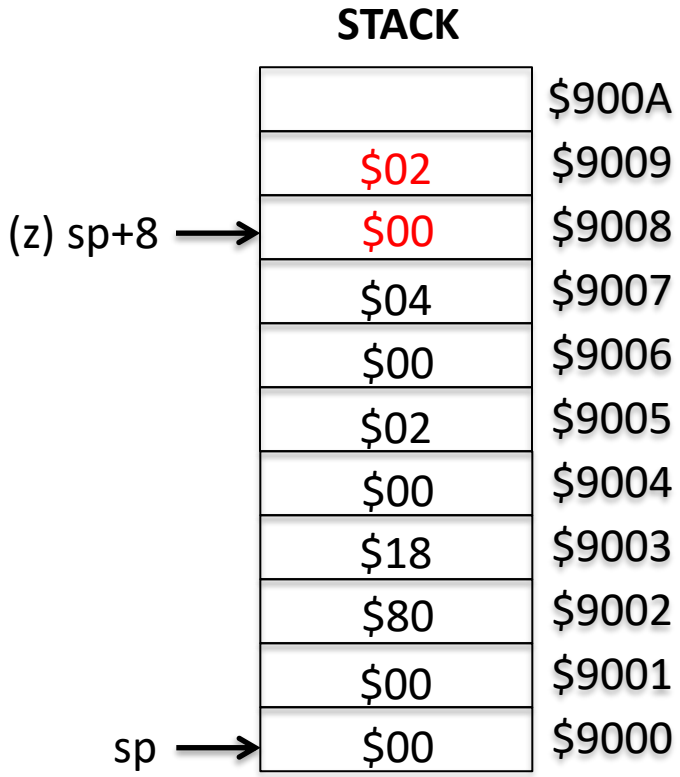


A7/SP | 00 | 00 | 90 | 00

d0 | 00 | 00 | 00 | 14

Trace

```
******** foo() **********
8000            move.w    c,-(sp)
8006            move.w    b,-(sp)
800C            move.w    a,-(sp)
8012            jsr       mul3
8018            lea       6(sp),sp
801E

******** mul3() **********
8022    mul3    move.w    4(sp),d0
8026            muls      6(sp),d0
802A            muls      8(sp),d0
802E            rts

                org               $A000
A000    a       dc.w              4
A002    b       dc.w              5
A004    c       dc.w              2
```

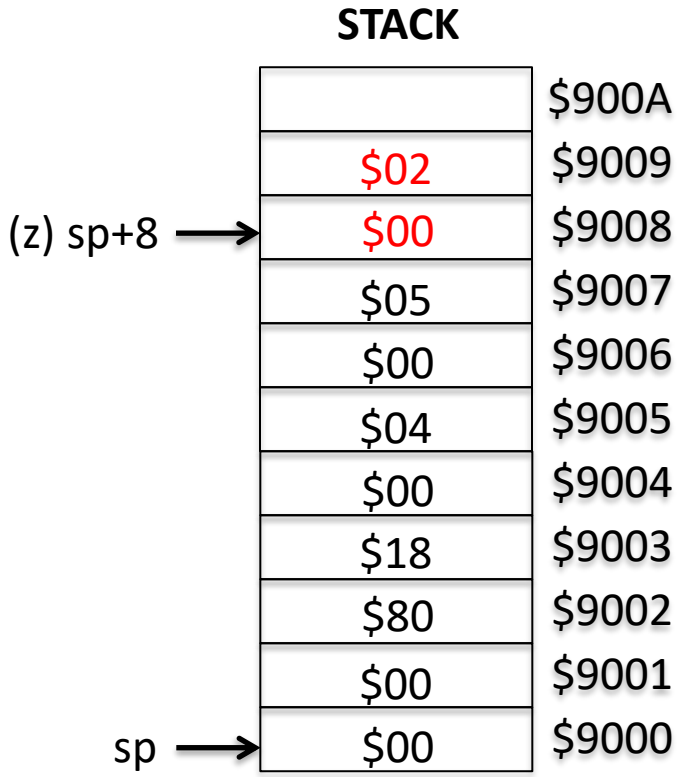


A7/SP	00	00	90	00

d0	00	00	00	14

Trace

```
******** foo() **********
8000            move.w    c,-(sp)
8006            move.w    b,-(sp)
800C            move.w    a,-(sp)
8012            jsr       mul3
8018            lea       6(sp),sp
801E

******** mul3() **********
8022    mul3    move.w    4(sp),d0
8026            muls      6(sp),d0
802A            muls      8(sp),d0
802E            rts

                org             $A000
A000    a       dc.w            4
A002    b       dc.w            5
A004    c       dc.w            2
```

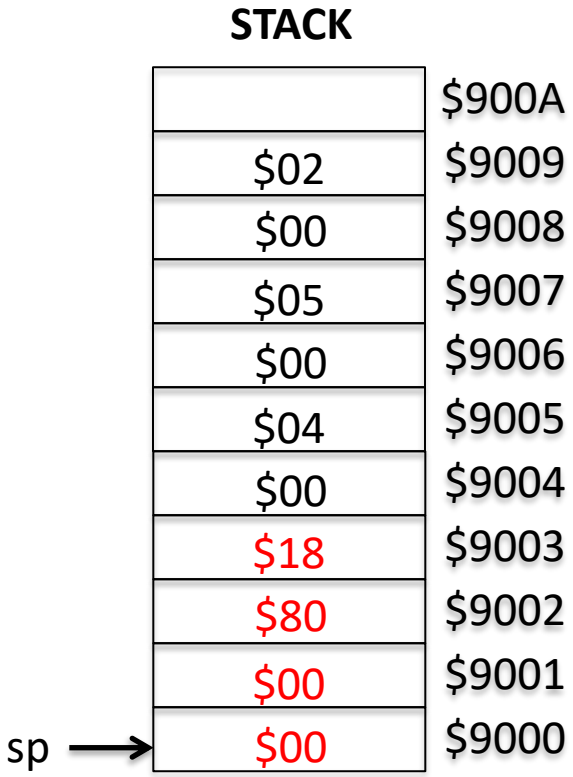


A7/SP | 00 | 00 | 90 | 00

d0 | 00 | 00 | 00 | 28

Trace

```
******** foo() **********
8000            move.w   c,-(sp)
8006            move.w   b,-(sp)
800C            move.w   a,-(sp)
8012            jsr      mul3
8018            lea      6(sp),sp
801E

******** mul3() **********
8022    mul3    move.w   4(sp),d0
8026            muls     6(sp),d0
802A            muls     8(sp),d0
802E            rts

                org             $A000
A000    a       dc.w            4
A002    b       dc.w            5
A004    c       dc.w            2
```

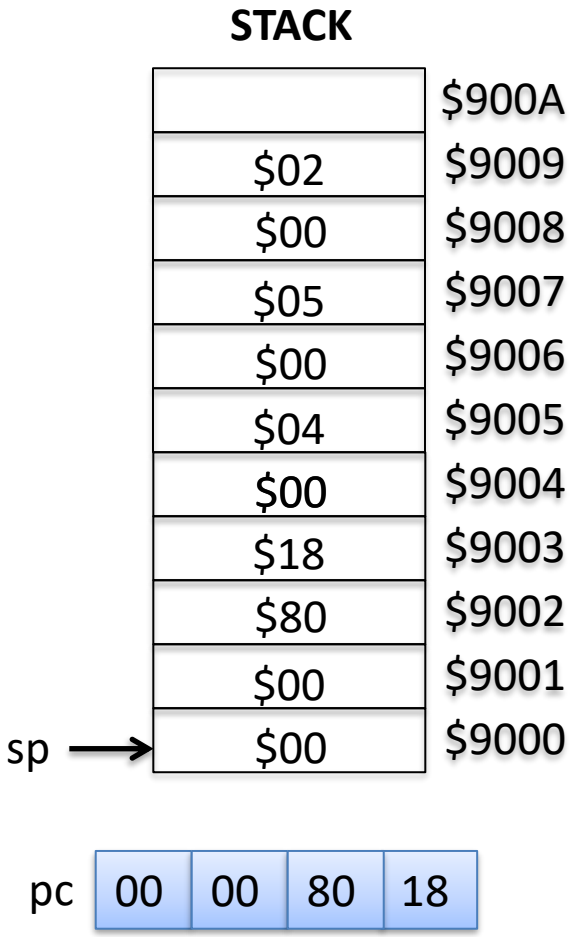


A7/SP | 00 | 00 | 90 | 00

d0 | 00 | 00 | 00 | 28

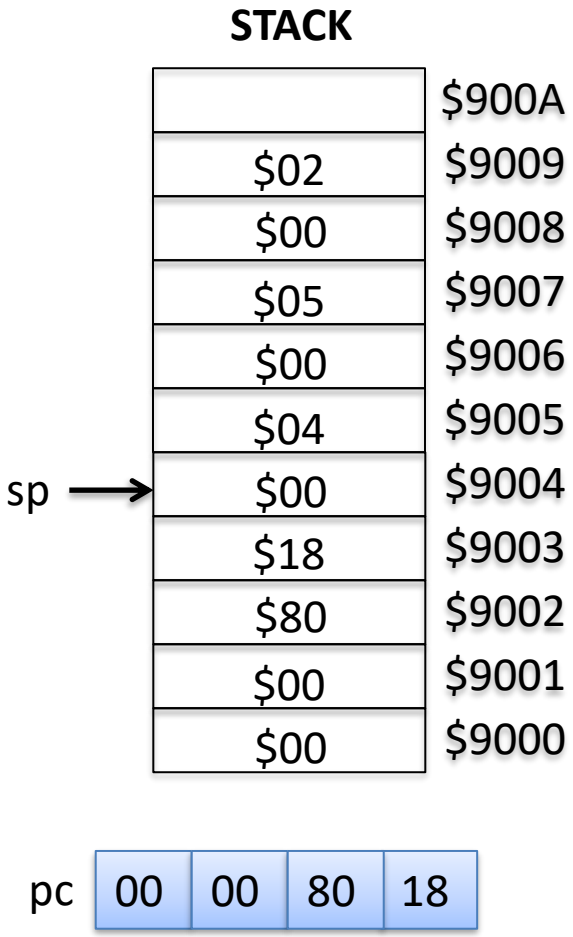
Trace

```
******** foo() **********
8000            move.w   c,-(sp)
8006            move.w   b,-(sp)
800C            move.w   a,-(sp)
8012            jsr      mul3
8018            lea      6(sp),sp
801E

******** mul3() **********
8022    mul3    move.w   4(sp),d0
8026            muls     6(sp),d0
802A            muls     8(sp),d0
802E            rts

                org             $A000
A000    a       dc.w            4
A002    b       dc.w            5
A004    c       dc.w            2
```



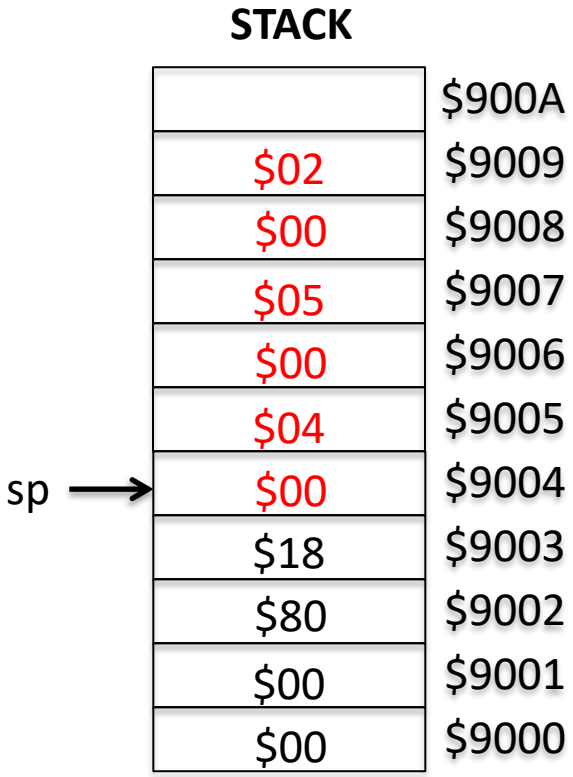
Trace

```
******** foo() **********
8000            move.w    c,-(sp)
8006            move.w    b,-(sp)
800C            move.w    a,-(sp)
8012            jsr       mul3
8018            lea       6(sp),sp
801E

******** mul3() **********
8022    mul3    move.w    4(sp),d0
8026            muls      6(sp),d0
802A            muls      8(sp),d0
802E            rts

                org             $A000
A000    a       dc.w            4
A002    b       dc.w            5
A004    c       dc.w            2
```



Trace

```
******** foo() **********
8000            move.w    c,-(sp)
8006            move.w    b,-(sp)
800C            move.w    a,-(sp)
8012            jsr       mul3
8018            lea       6(sp),sp
801E

******** mul3() **********
8022    mul3    move.w    4(sp),d0
8026            muls      6(sp),d0
802A            muls      8(sp),d0
802E            rts

                org             $A000
A000    a       dc.w            4
A002    b       dc.w            5
A004    c       dc.w            2
```

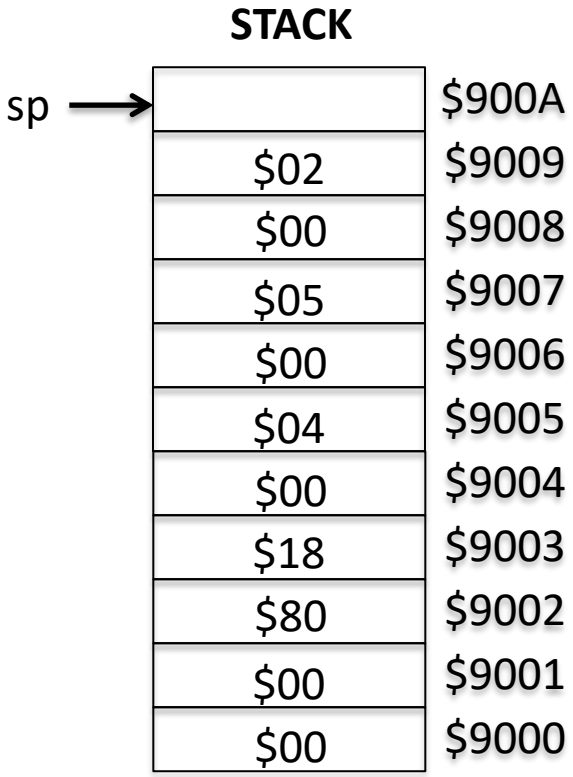


A7/SP | 00 | 00 | 90 | 04

d0 | 00 | 00 | 00 | 28

Trace

```
******** foo() **********
8000            move.w    c,-(sp)
8006            move.w    b,-(sp)
800C            move.w    a,-(sp)
8012            jsr       mul3
8018            lea       6(sp),sp
801E

******** mul3() **********
8022    mul3    move.w    4(sp),d0
8026            muls      6(sp),d0
802A            muls      8(sp),d0
802E            rts

                org             $A000
A000    a       dc.w            4
A002    b       dc.w            5
A004    c       dc.w            2
```



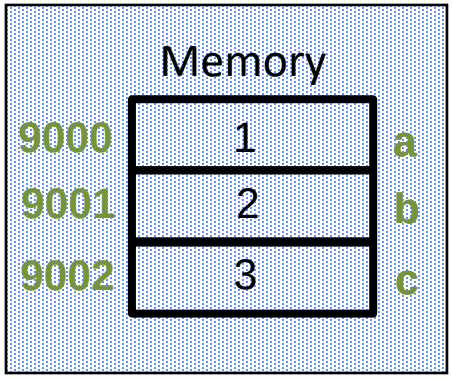
A7/SP | 00 | 00 | 90 | 0A

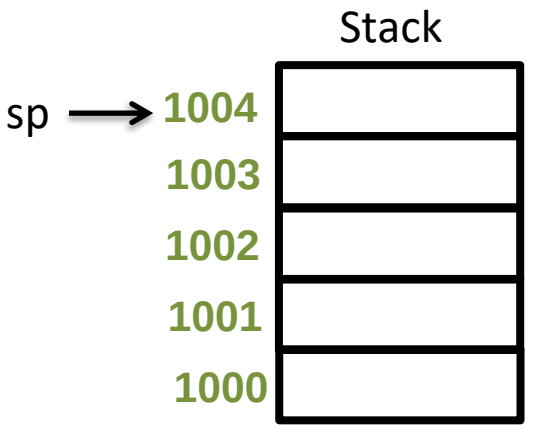
d0 | 00 | 00 | 00 | 28

PEA Instruction

- Consider the following code

```
      org    $9000
a     dc.b   1
b     dc.b   2
c     dc.b   3

        .
        .
        .

      pea    a
```

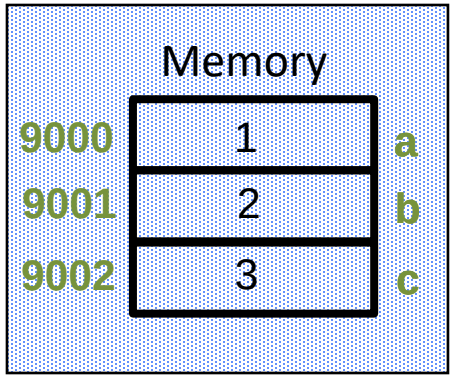


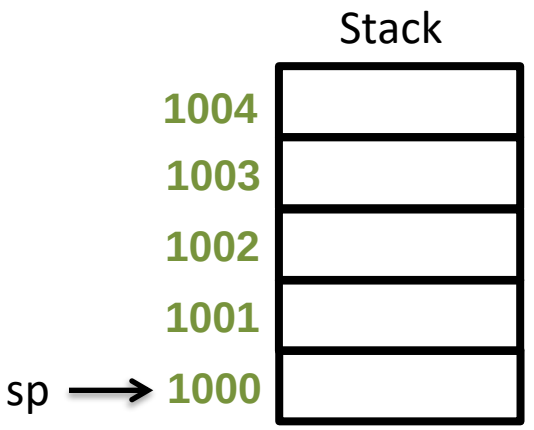


PEA Instruction

- Consider the following code

```
      org    $9000
a     dc.b   1
b     dc.b   2
c     dc.b   3

        .
        .
        .

      pea    a
```

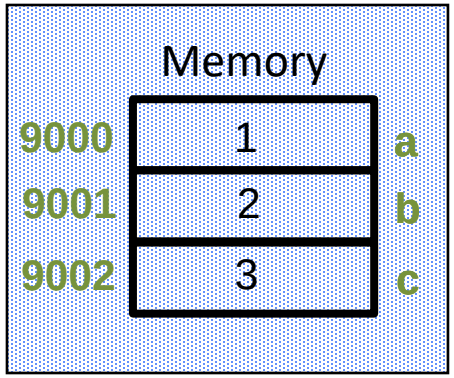


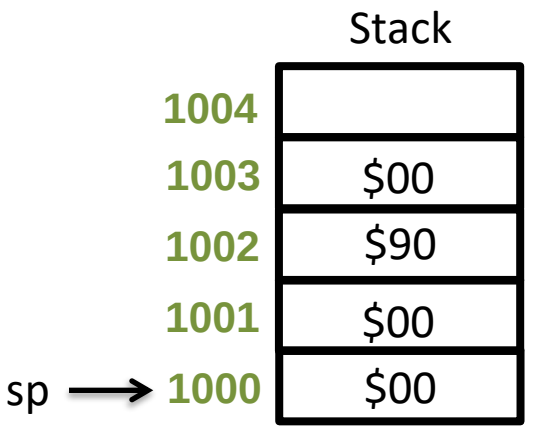


PEA Instruction

- Consider the following code

```
      org    $9000
a     dc.b   1
b     dc.b   2
c     dc.b   3

        .
        .
        .

      pea    a
```

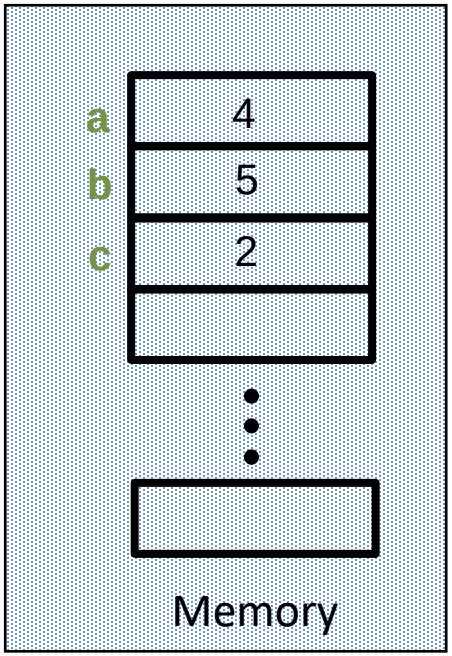




C Function – Pass by Reference

- Consider the following code

```
void foo() {
    short int a=4, b=5, c=2;
    double3(&a,&b,&c);
}


int double3(short int *x, short int *y,
            short int *z) {

    *x = *x * 2;
    *y = *y * 2;
    *z = *z * 2;
}
```

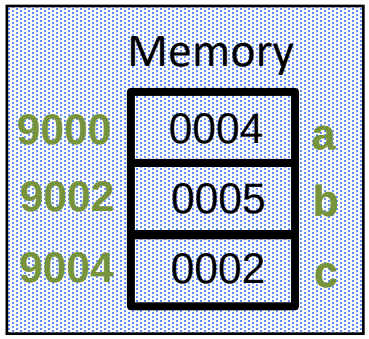


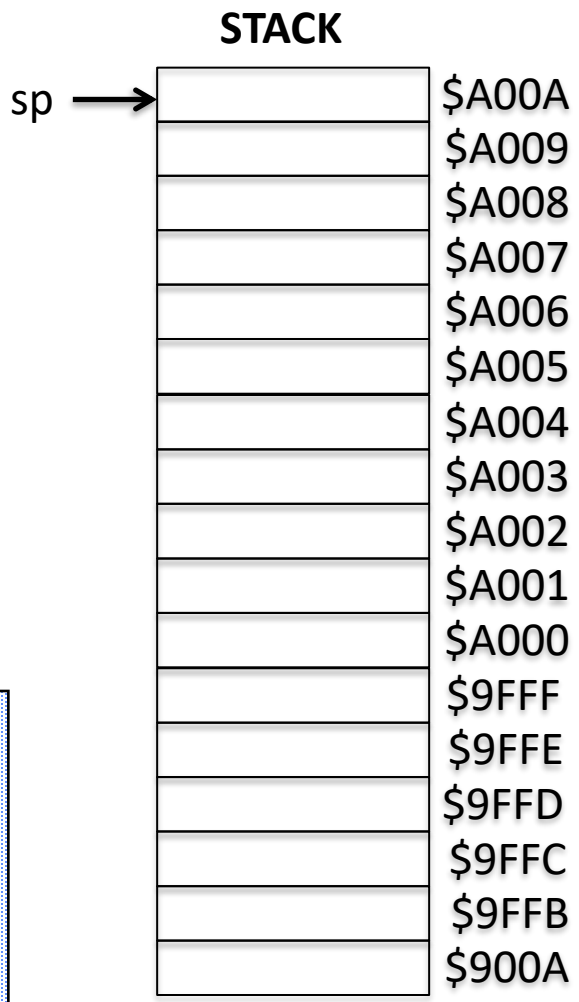
Assembler Code – Pass by Reference

```
        pea         c               ;push 3rd parameter
        pea         b               ;push 2nd parameter
        pea         a               ;push 1st parameter
        jsr         double3         ;call subroutine
        lea         12(sp),sp       ;remove parameters

***** double3() uses 3 pointers to double 3 ints *****
double3 movea.l     4(sp),a0        ;a0 = &x
        asl         (a0)            ;*x = *x * 2;
        movea.l     8(sp),a0        ;a0 = &y
        asl         (a0)            ;*y = *y * 2;
        movea.l     12(sp),a0       ;a0 = &z
        asl         (a0)            ;*z = *z * 2;
        rts                         ;return
        org         $9000           ;function parameters
a       dc.w        4
b       dc.w        5
c       dc.w        2
```

Trace

```
8000     pea          c
8006     pea          b
800C     pea          a
8012     jsr          double3
8018     lea          12(sp),sp
801E
*** SUBROUTINE ***
double3  movea.l      4(sp),a0
         asl          (a0)
         movea.l      8(sp),a0
         asl          (a0)
         movea.l      12(sp),a0
         asl          (a0)
         rts

a        dc.w         4
b        dc.w         5
c        dc.w         2
```



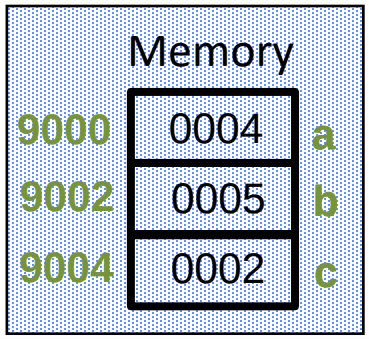


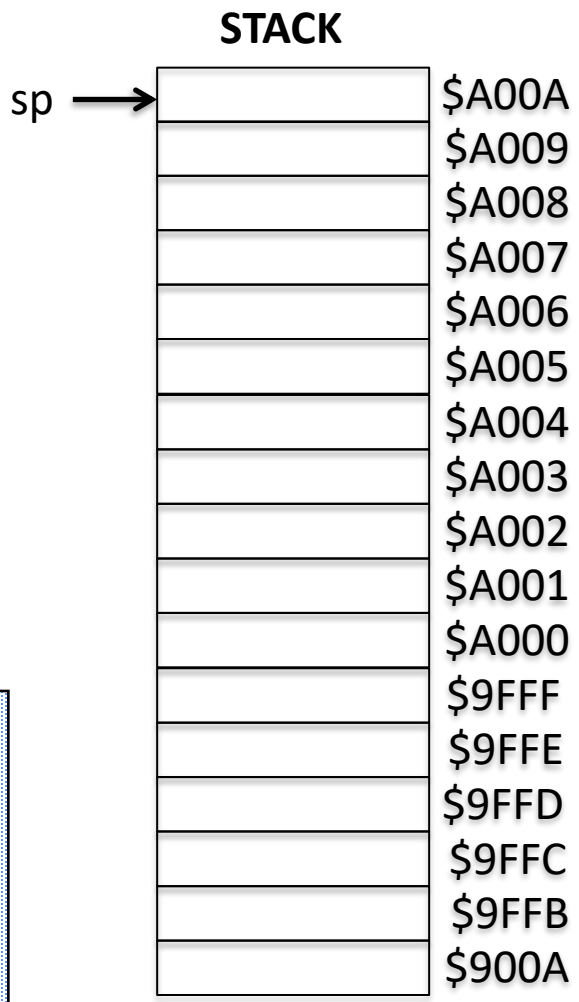
A7/SP	00	00	A0	0A

A0	XX	XX	XX	XX

Trace

```
8000     pea          c
8006     pea          b
800C     pea          a
8012     jsr          double3
8018     lea          12(sp),sp
801E
*** SUBROUTINE ***
double3  movea.l      4(sp),a0
         asl          (a0)
         movea.l      8(sp),a0
         asl          (a0)
         movea.l      12(sp),a0
         asl          (a0)
         rts

a        dc.w         4
b        dc.w         5
c        dc.w         2
```



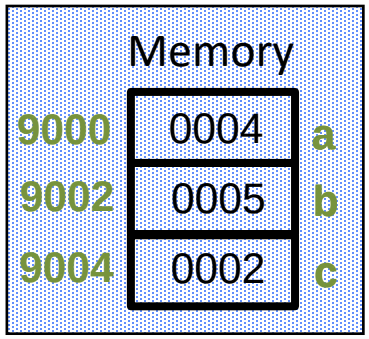


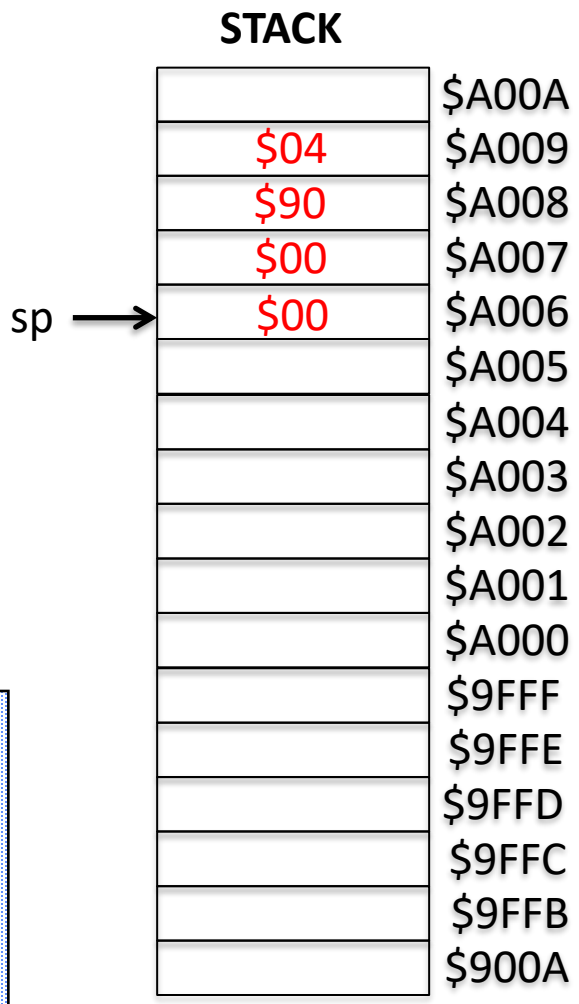
A7/SP	00	00	A0	0A

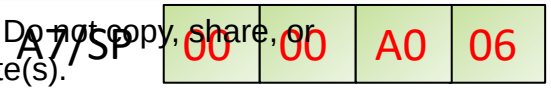
A0	XX	XX	XX	XX

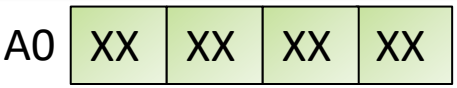
Trace

```
8000     pea          c
8006     pea          b
800C     pea          a
8012     jsr          double3
8018     lea          12(sp),sp
801E
*** SUBROUTINE ***
double3  movea.l      4(sp),a0
         asl          (a0)
         movea.l      8(sp),a0
         asl          (a0)
         movea.l      12(sp),a0
         asl          (a0)
         rts

a        dc.w         4
b        dc.w         5
c        dc.w         2
```



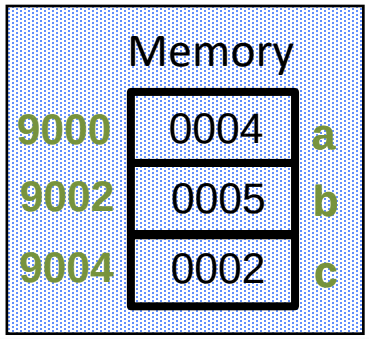


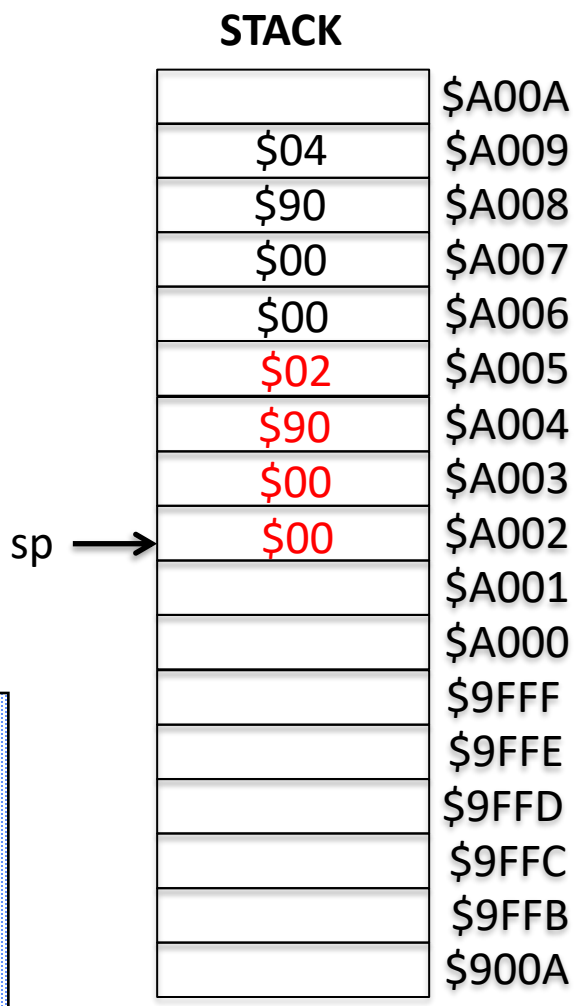


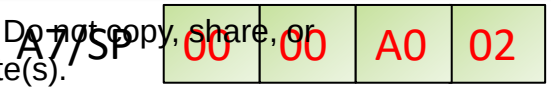


Trace

```
8000     pea          c
8006     pea          b
800C     pea          a
8012     jsr          double3
8018     lea          12(sp),sp
801E
*** SUBROUTINE ***
double3  movea.l      4(sp),a0
         asl          (a0)
         movea.l      8(sp),a0
         asl          (a0)
         movea.l      12(sp),a0
         asl          (a0)
         rts

a        dc.w         4
b        dc.w         5
c        dc.w         2
```

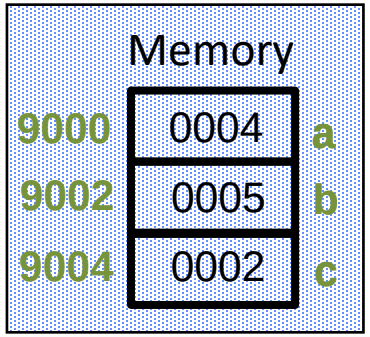


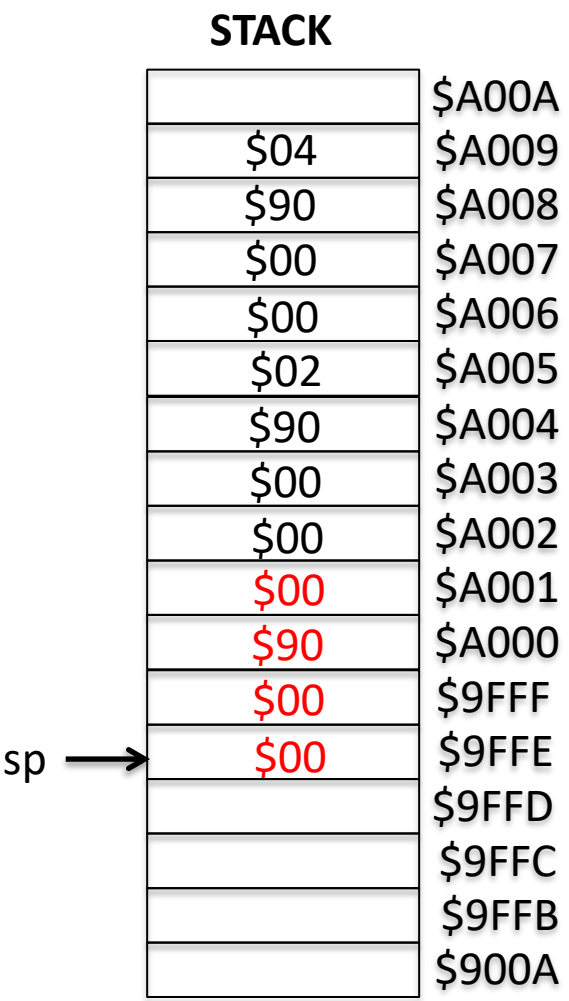




Trace

```
8000     pea          c
8006     pea          b
800C     pea          a
8012     jsr          double3
8018     lea          12(sp),sp
801E
*** SUBROUTINE ***
double3  movea.l      4(sp),a0
         asl          (a0)
         movea.l      8(sp),a0
         asl          (a0)
         movea.l      12(sp),a0
         asl          (a0)
         rts

a        dc.w         4
b        dc.w         5
c        dc.w         2
```



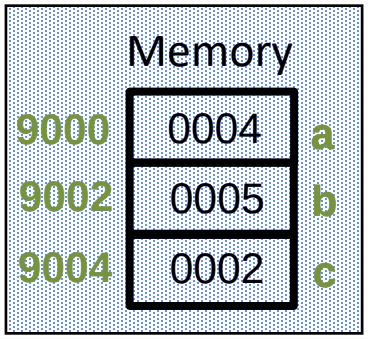


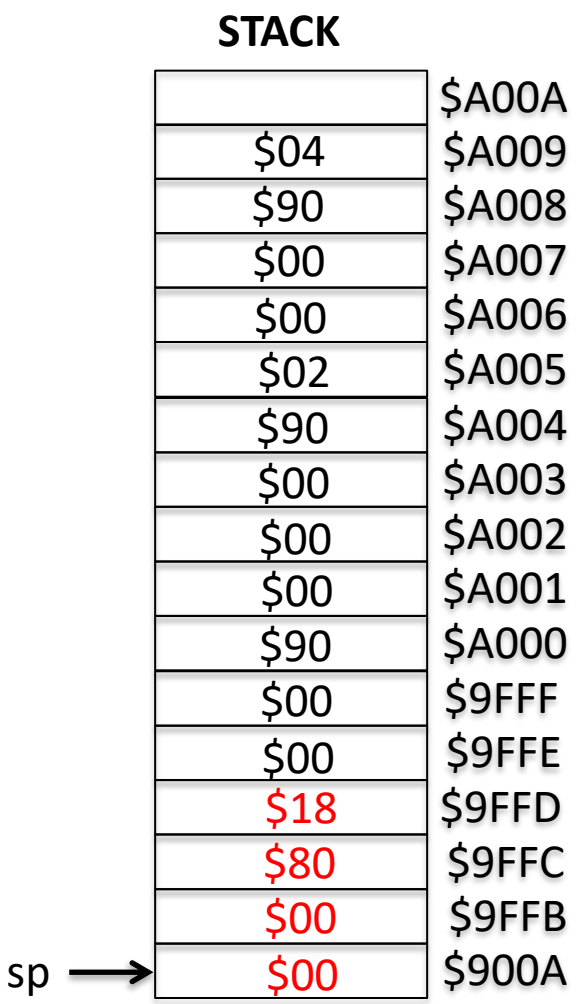
A7/SP | 00 | 00 | 9F | FE

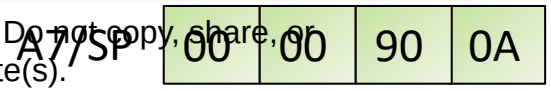
A0 | XX | XX | XX | XX

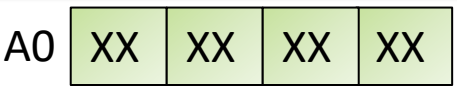
Trace

```
8000      pea          c
8006      pea          b
800C      pea          a
8012      jsr          double3
8018      lea          12(sp),sp
801E
*** SUBROUTINE ***
double3 movea.l        4(sp),a0
        asl            (a0)
        movea.l        8(sp),a0
        asl            (a0)
        movea.l        12(sp),a0
        asl            (a0)
        rts

a       dc.w           4
b       dc.w           5
c       dc.w           2
```



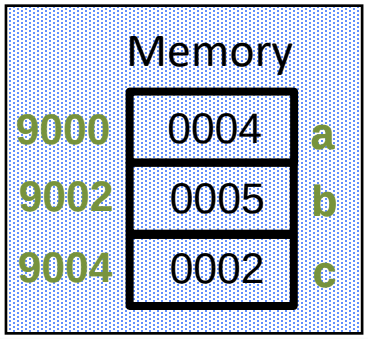


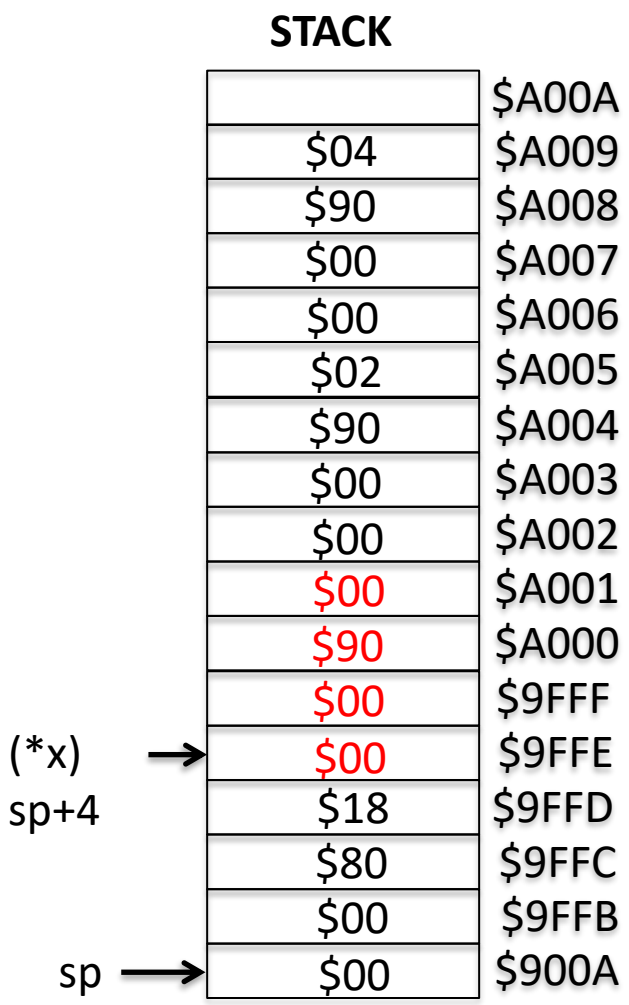




Trace

```
8000      pea          c
8006      pea          b
800C      pea          a
8012      jsr          double3
8018      lea          12(sp),sp
801E
*** SUBROUTINE ***
double3 movea.l        4(sp),a0
        asl            (a0)
        movea.l        8(sp),a0
        asl            (a0)
        movea.l        12(sp),a0
        asl            (a0)
        rts

a       dc.w           4
b       dc.w           5
c       dc.w           2
```





A7/SP | 00 | 00 | 90 | 0A

A0 | XX | XX | XX | XX

Trace

```
8000     pea          c
8006     pea          b
800C     pea          a
8012     jsr          double3
8018     lea          12(sp),sp
801E
*** SUBROUTINE ***
double3  movea.l      4(sp),a0
         asl          (a0)
         movea.l      8(sp),a0
         asl          (a0)
         movea.l      12(sp),a0
         asl          (a0)
         rts

a        dc.w         4
b        dc.w         5
c        dc.w         2
```

a0 →

Memory		
9000	0004	a
9002	0005	b
9004	0002	c

STACK

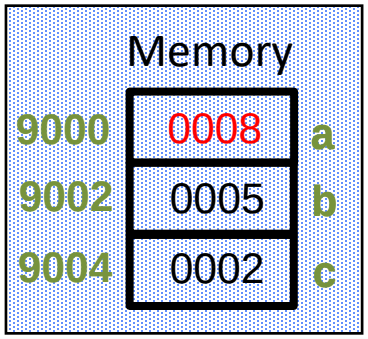
	$A00A
$04	$A009
$90	$A008
$00	$A007
$00	$A006
$02	$A005
$90	$A004
$00	$A003
$00	$A002
$00	$A001
$90	$A000
$00	$9FFF
$00	$9FFE
$18	$9FFD
$80	$9FFC
$00	$9FFB
sp → $00	$900A

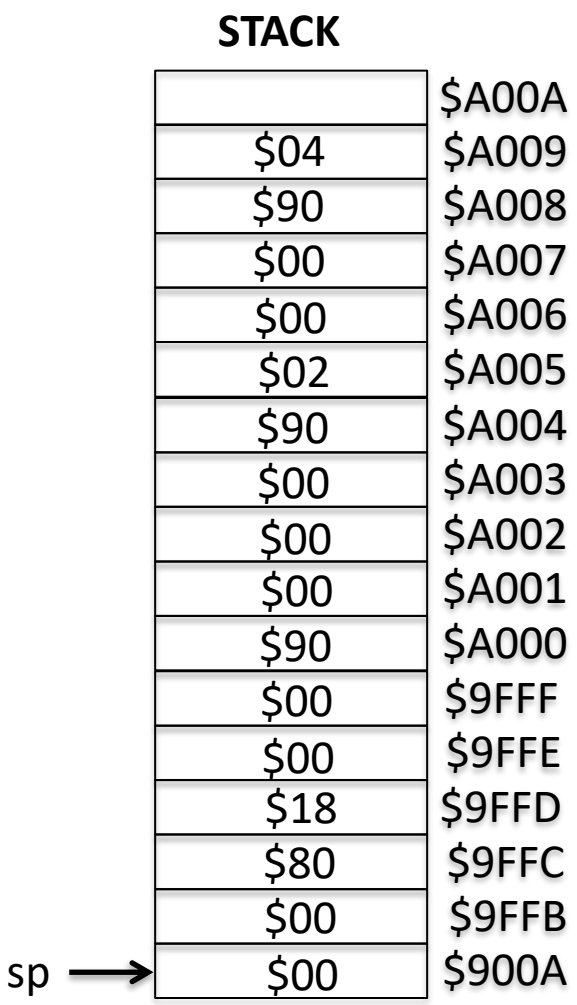
A7/SP	00	00	90	0A

A0	00	00	90	00

Trace

```
8000     pea          c
8006     pea          b
800C     pea          a
8012     jsr          double3
8018     lea          12(sp),sp
801E
*** SUBROUTINE ***
double3  movea.l      4(sp),a0
         asl          (a0)
         movea.l      8(sp),a0
         asl          (a0)
         movea.l      12(sp),a0
         asl          (a0)
         rts

a        dc.w         4
b        dc.w         5
c        dc.w         2
```



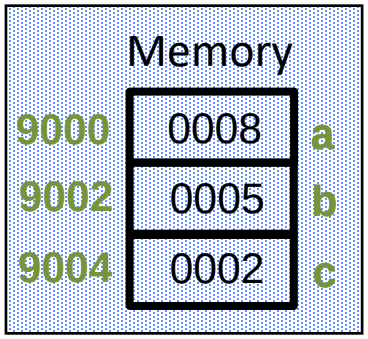


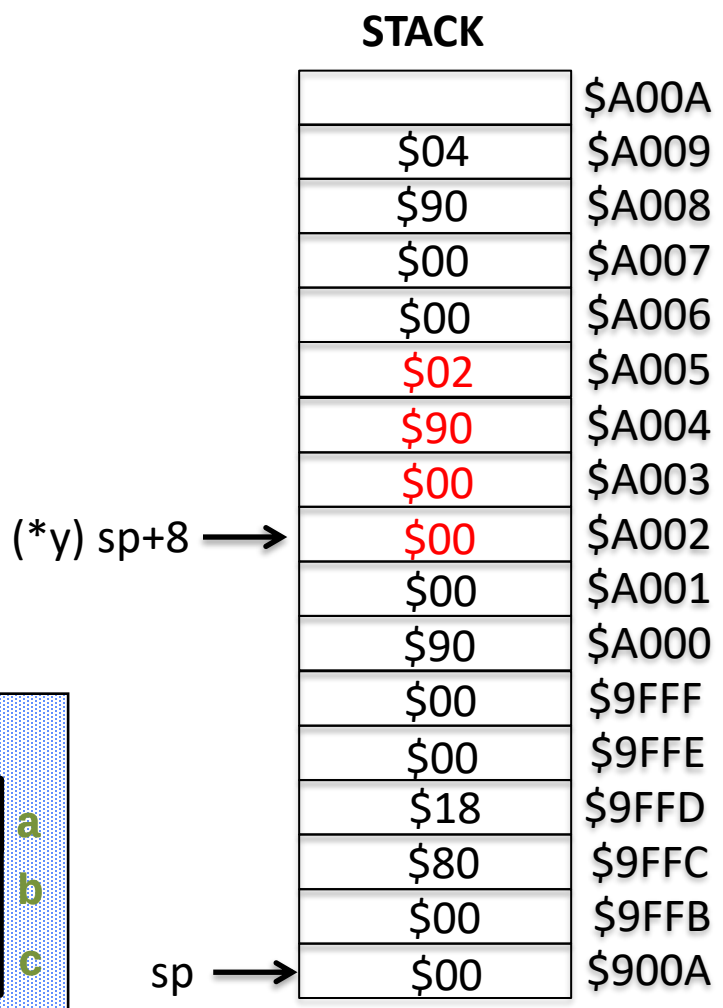
A7/SP	00	00	90	0A

A0	00	00	90	00

Trace

```
8000      pea          c
8006      pea          b
800C      pea          a
8012      jsr          double3
8018      lea          12(sp),sp
801E
*** SUBROUTINE ***
double3   movea.l      4(sp),a0
          asl          (a0)
          movea.l      8(sp),a0
          asl          (a0)
          movea.l      12(sp),a0
          asl          (a0)
          rts

a         dc.w         4
b         dc.w         5
c         dc.w         2
```



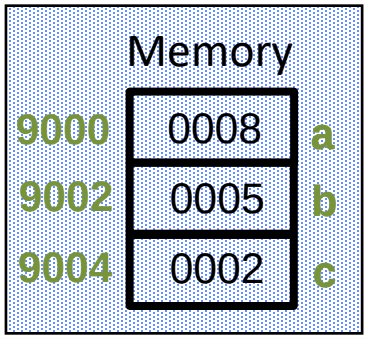


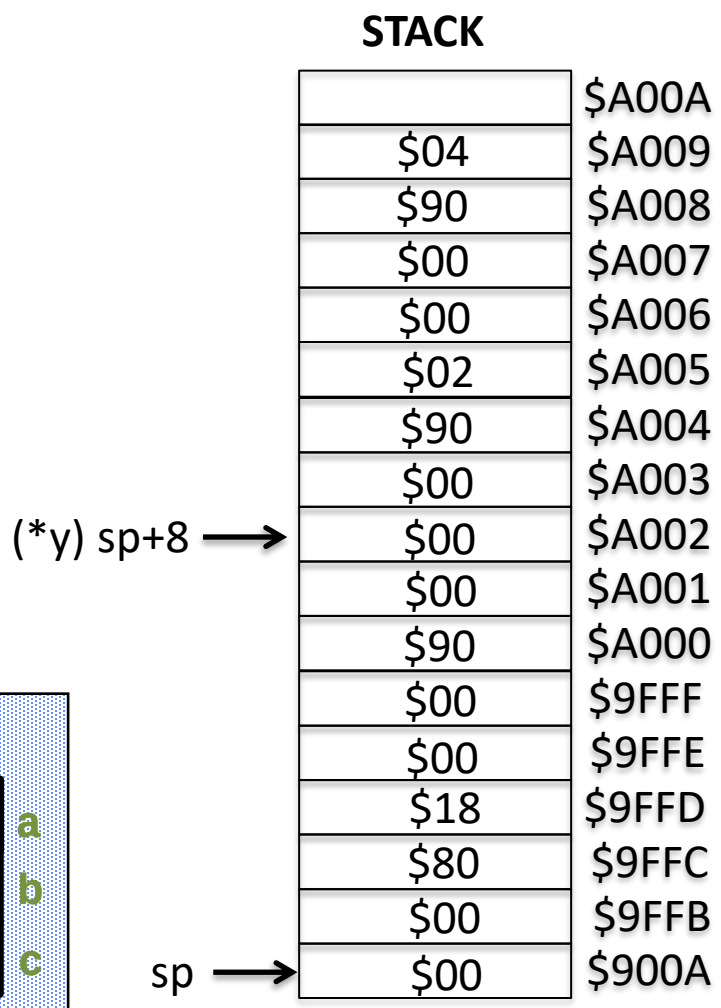
A7/SP | 00 | 00 | 90 | 0A

A0 | 00 | 00 | 90 | 00

Trace

```
8000      pea          c
8006      pea          b
800C      pea          a
8012      jsr          double3
8018      lea          12(sp),sp
801E
*** SUBROUTINE ***
double3   movea.l      4(sp),a0
          asl          (a0)
          movea.l      8(sp),a0
          asl          (a0)
          movea.l      12(sp),a0
          asl          (a0)
          rts

a         dc.w         4
b         dc.w         5
c         dc.w         2
```



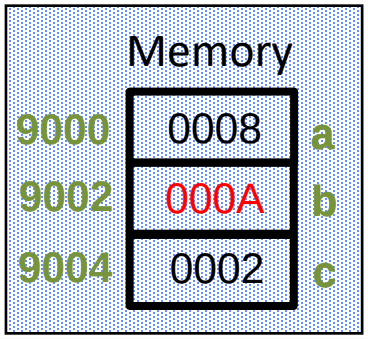


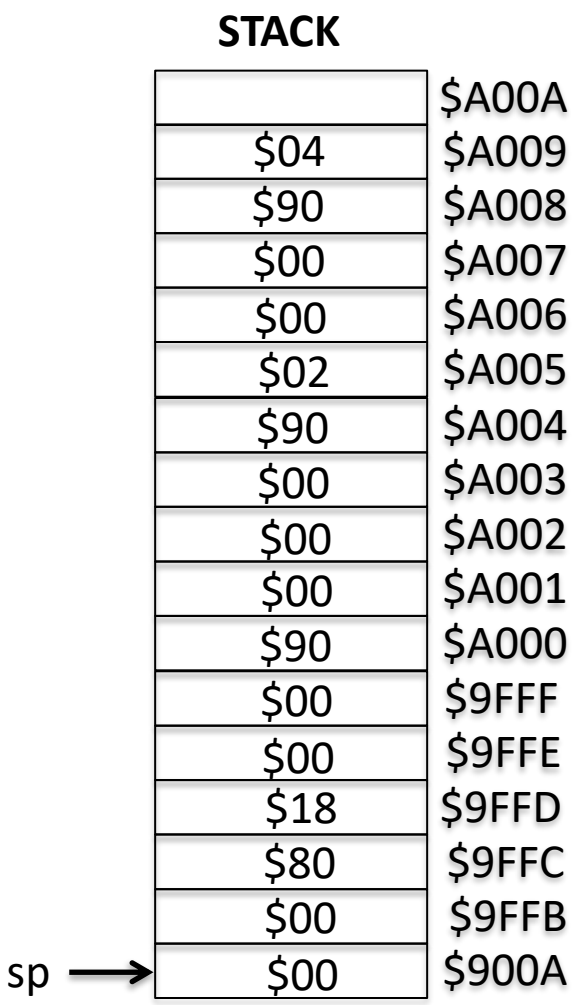
A7/SP | 00 | 00 | 90 | 0A

A0 | 00 | 00 | 90 | 02

Trace

```
8000      pea          c
8006      pea          b
800C      pea          a
8012      jsr          double3
8018      lea          12(sp),sp
801E
*** SUBROUTINE ***
double3   movea.l      4(sp),a0
          asl          (a0)
          movea.l      8(sp),a0
          asl          (a0)
          movea.l      12(sp),a0
          asl          (a0)
          rts

a         dc.w         4
b         dc.w         5
c         dc.w         2
```



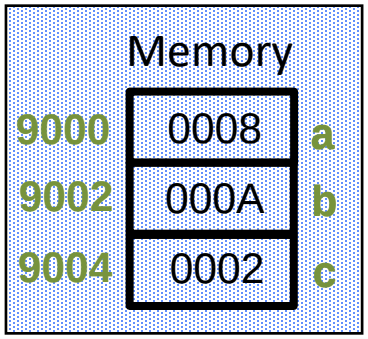


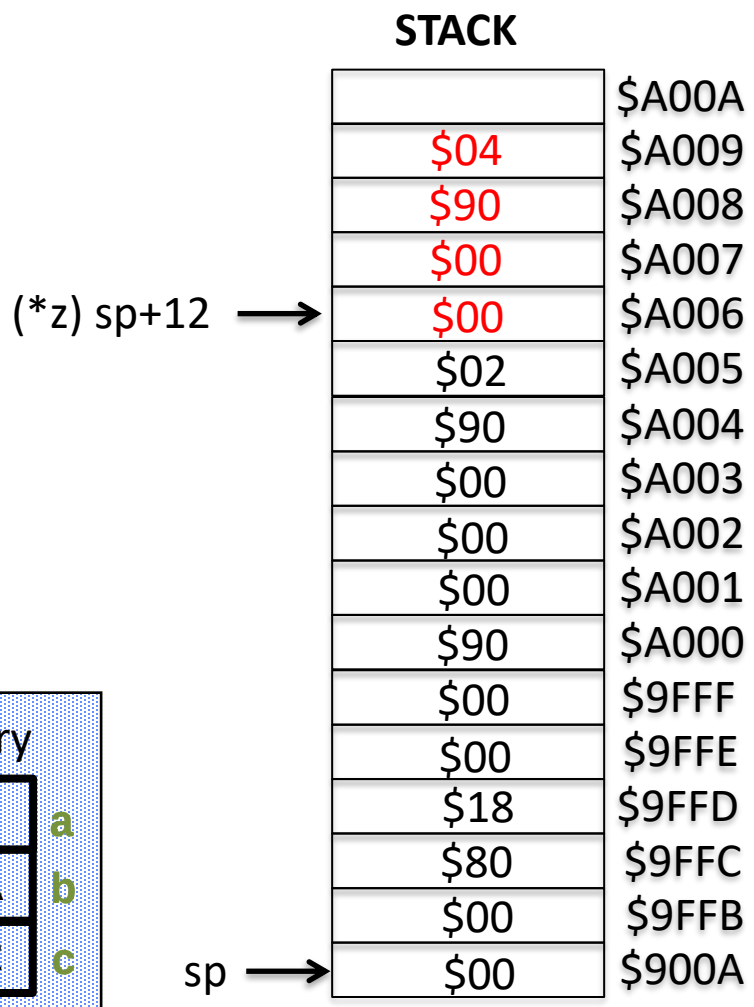
A7/SP	00	00	90	0A

A0	00	00	90	02

Trace

```
8000     pea          c
8006     pea          b
800C     pea          a
8012     jsr          double3
8018     lea          12(sp),sp
801E
*** SUBROUTINE ***
double3  movea.l      4(sp),a0
         asl          (a0)
         movea.l      8(sp),a0
         asl          (a0)
         movea.l      12(sp),a0
         asl          (a0)
         rts

a        dc.w         4
b        dc.w         5
c        dc.w         2
```



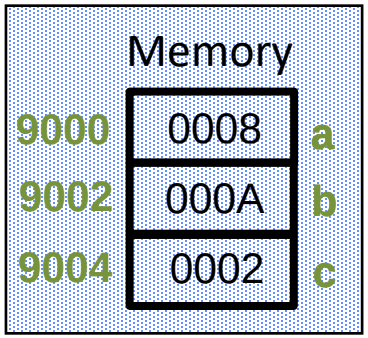


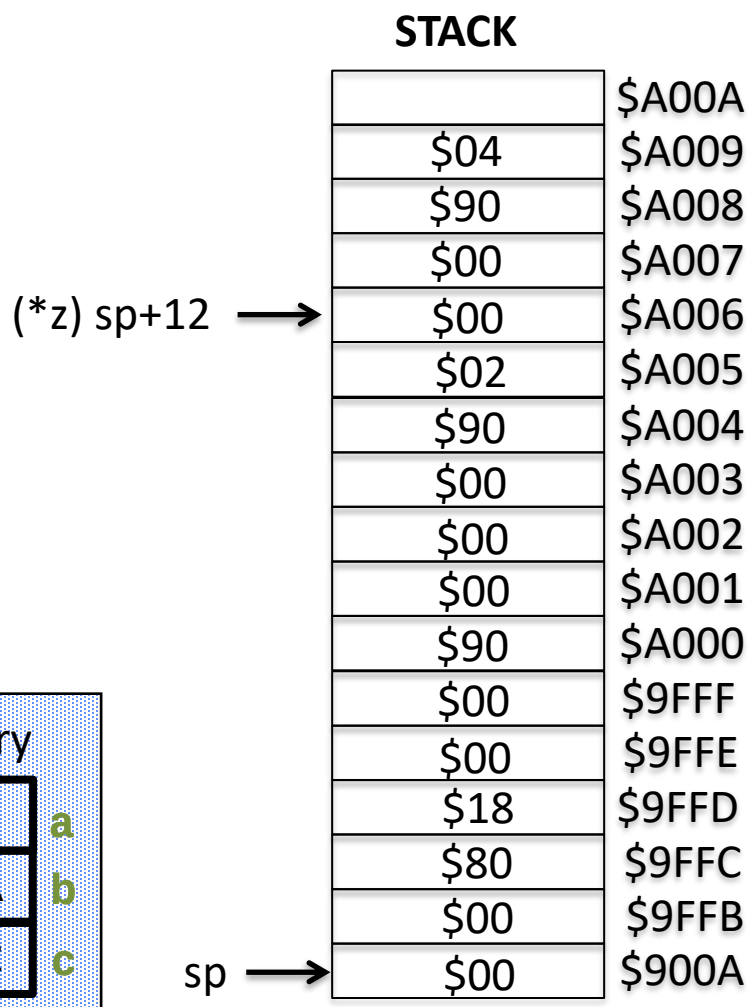
A7/SP	00	00	90	0A

A0	00	00	90	02

Trace

```
8000     pea          c
8006     pea          b
800C     pea          a
8012     jsr          double3
8018     lea          12(sp),sp
801E
*** SUBROUTINE ***
double3  movea.l      4(sp),a0
         asl          (a0)
         movea.l      8(sp),a0
         asl          (a0)
         movea.l      12(sp),a0
         asl          (a0)
         rts

a        dc.w         4
b        dc.w         5
c        dc.w         2
```



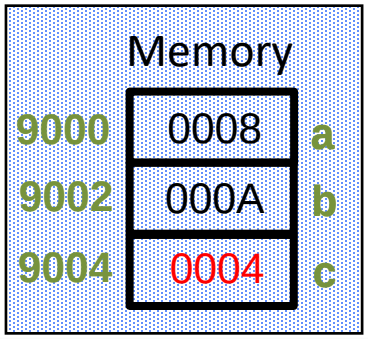


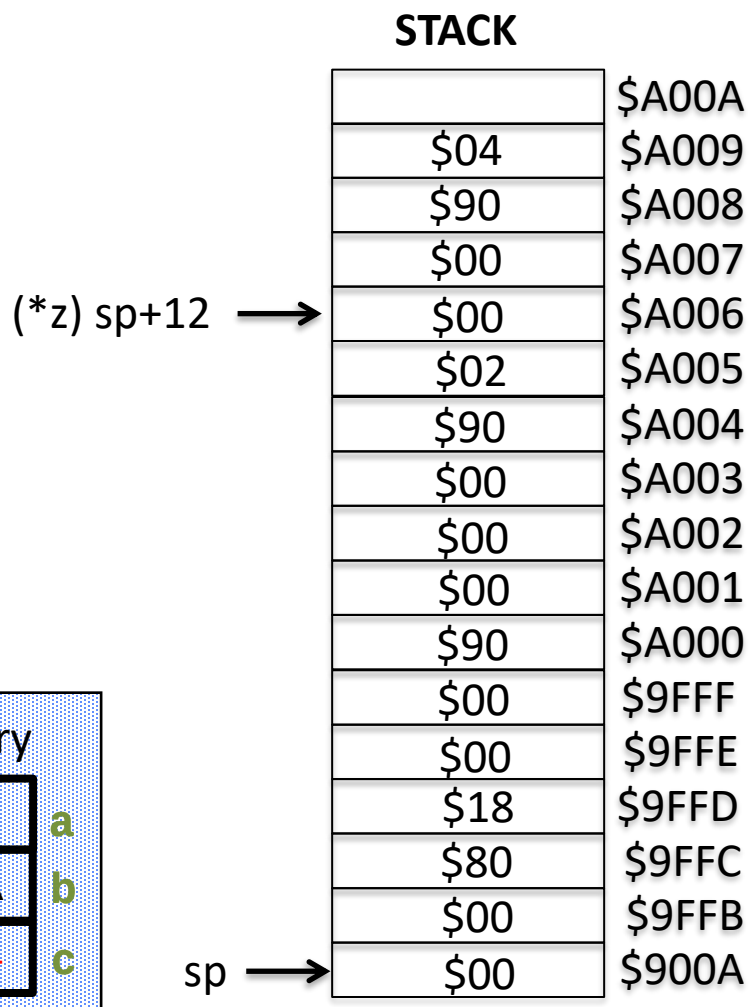
A7/SP | 00 | 00 | 90 | 0A

A0 | 00 | 00 | 90 | 04

Trace

```
8000     pea          c
8006     pea          b
800C     pea          a
8012     jsr          double3
8018     lea          12(sp),sp
801E
*** SUBROUTINE ***
double3  movea.l      4(sp),a0
         asl          (a0)
         movea.l      8(sp),a0
         asl          (a0)
         movea.l      12(sp),a0
         asl          (a0)
         rts

a        dc.w         4
b        dc.w         5
c        dc.w         2
```



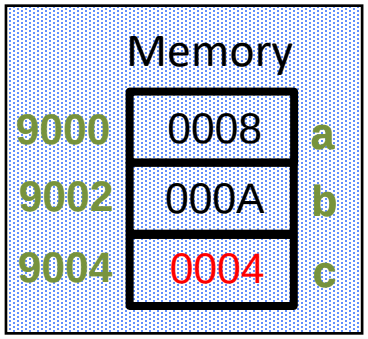


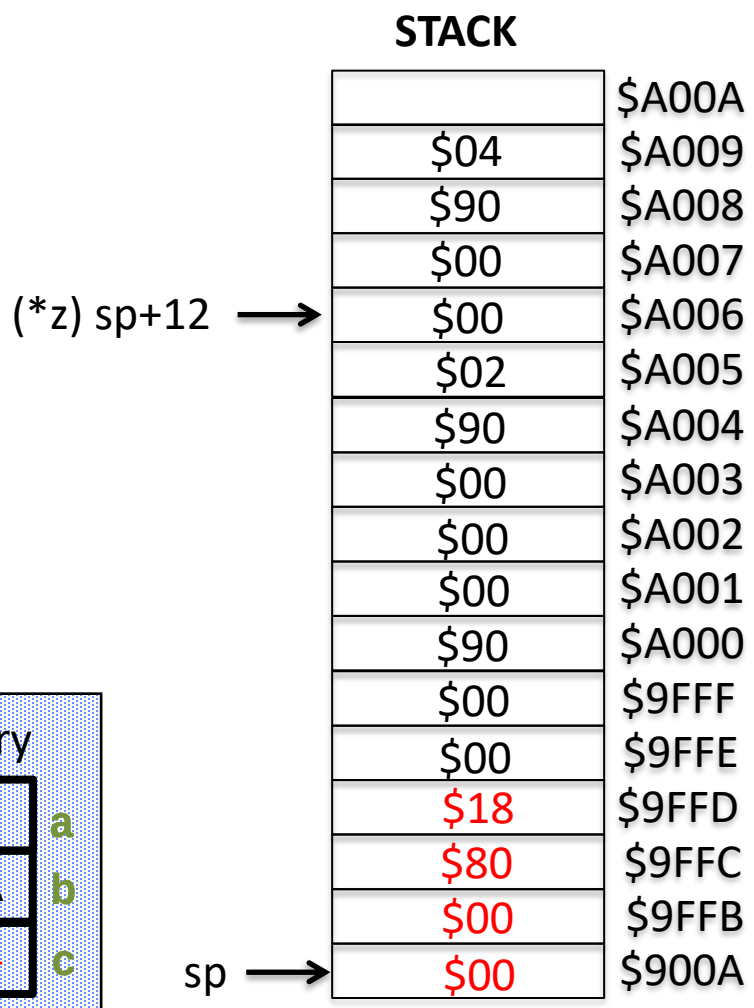
A7/SP	00	00	90	0A

A0	00	00	90	04

Trace

```
8000     pea          c
8006     pea          b
800C     pea          a
8012     jsr          double3
8018     lea          12(sp),sp
801E
*** SUBROUTINE ***
double3  movea.l      4(sp),a0
         asl          (a0)
         movea.l      8(sp),a0
         asl          (a0)
         movea.l      12(sp),a0
         asl          (a0)
         rts

a        dc.w         4
b        dc.w         5
c        dc.w         2
```



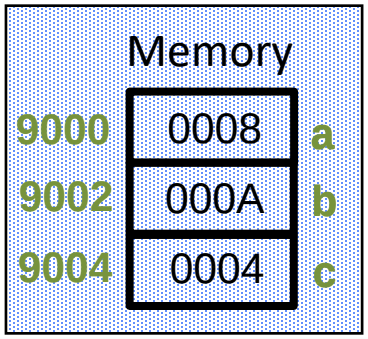


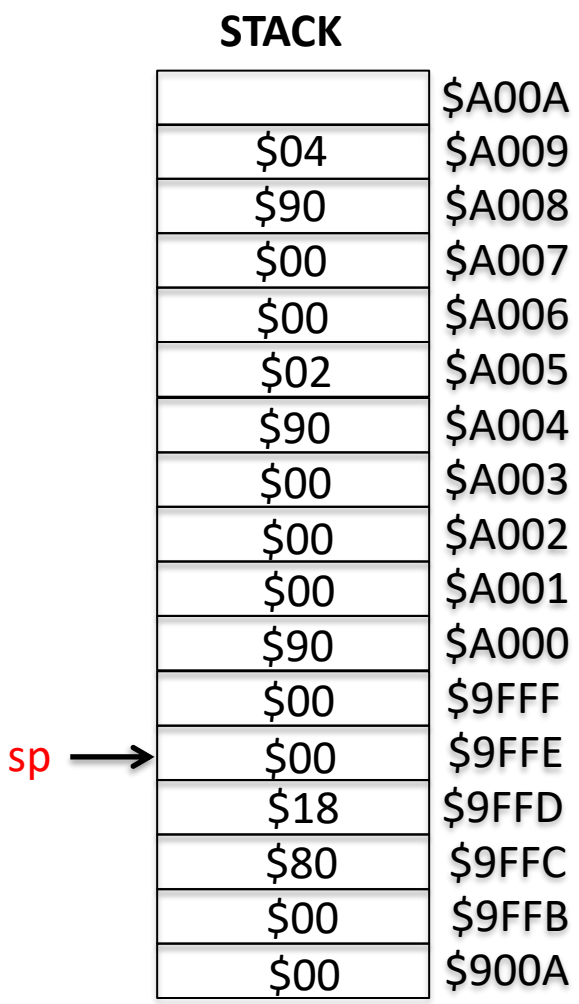
A7/SP | 00 | 00 | 90 | 0A

A0 | 00 | 00 | 90 | 04

Trace

```
8000     pea          c
8006     pea          b
800C     pea          a
8012     jsr          double3
8018     lea          12(sp),sp
801E
*** SUBROUTINE ***
double3  movea.l      4(sp),a0
         asl          (a0)
         movea.l      8(sp),a0
         asl          (a0)
         movea.l      12(sp),a0
         asl          (a0)
         rts

a        dc.w         4
b        dc.w         5
c        dc.w         2
```





A7/SP | 00 | 00 | 9F | FE

A0 | 00 | 00 | 90 | 04

Trace

```
8000      pea          c
8006      pea          b
800C      pea          a
8012      jsr          double3
8018      lea          12(sp),sp
801E
*** SUBROUTINE ***
double3   movea.l      4(sp),a0
          asl          (a0)
          movea.l      8(sp),a0
          asl          (a0)
          movea.l      12(sp),a0
          asl          (a0)
          rts

a         dc.w         4
b         dc.w         5
c         dc.w         2
```

Memory

9000	0008	a
9002	000A	b
9004	0004	c

STACK

sp →		$A00A
	$04	$A009
	$90	$A008
	$00	$A007
	$00	$A006
	$02	$A005
	$90	$A004
	$00	$A003
	$00	$A002
	$00	$A001
	$90	$A000
	$00	$9FFF
	$00	$9FFE
	$18	$9FFD
	$80	$9FFC
	$00	$9FFB
	$00	$900A

A7/SP				
	00	00	A0	0A

A0				
	00	00	90	04

Transparency

- A “transparent” subroutine does not change any registers
  - Also known as “preserving registers across a call”
  - Achieving transparency:
    - Register values are “saved” when first entering the subroutine
    - Restored prior to leaving the subroutine

- Where do we store the registers?
  - On the stack

MOVEM Instruction

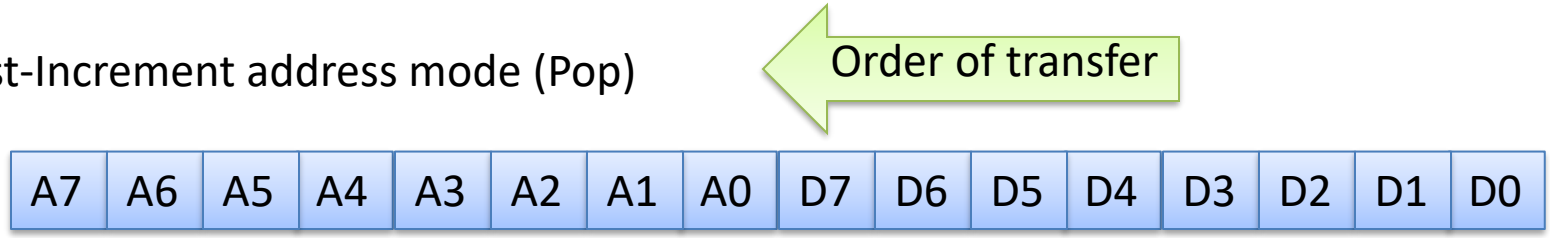
MOVEM Move Multiple Registers

Syntax:
```
MOVEM reg.list,<ea>
MOVEM <ea>,reg.list
```

Operation: registers -> destination
source -> registers

- Pre-decrement address mode (Push)

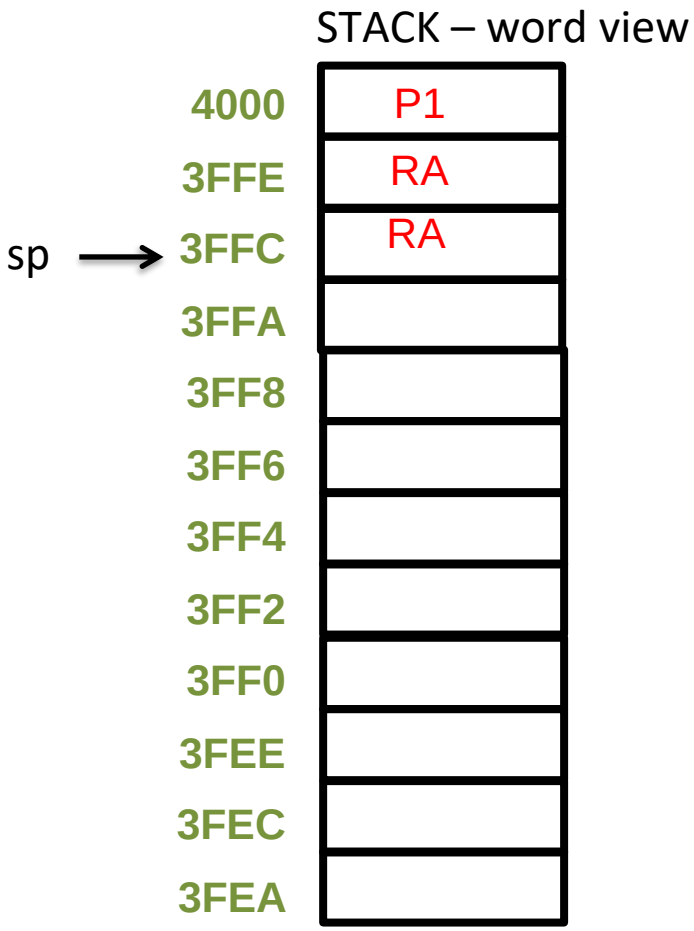
- Post-Increment address mode (Pop)



Example

- Consider the following code

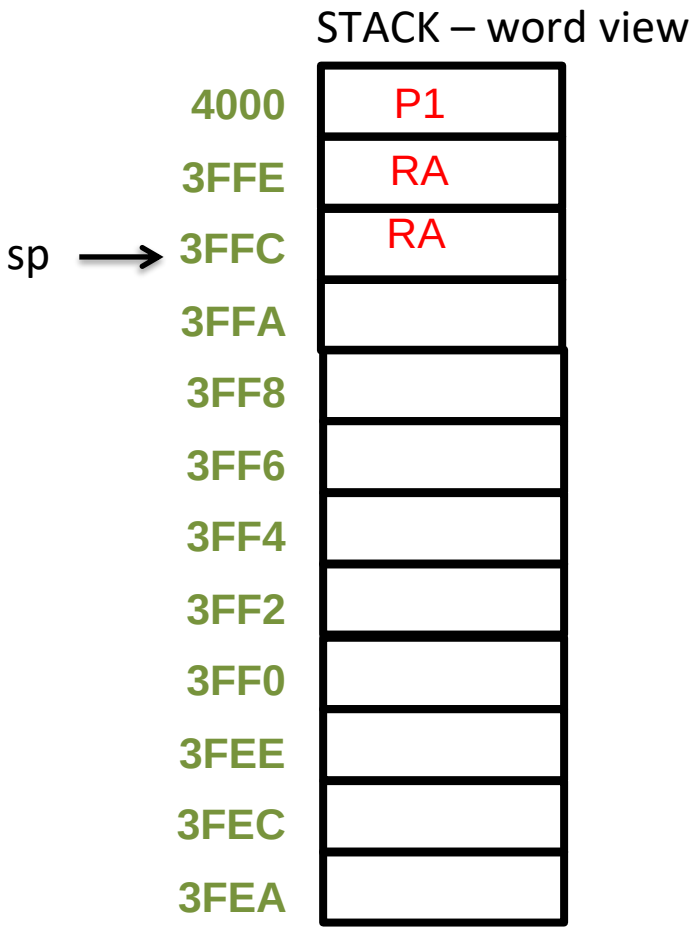
```
SUB   MOVEM.L   D1-D3/A0,-(SP)
          .
          .
          .
      MOVEM.L   (SP)+,D1-D3/A0
      RTS
```



Example

- Consider the following code

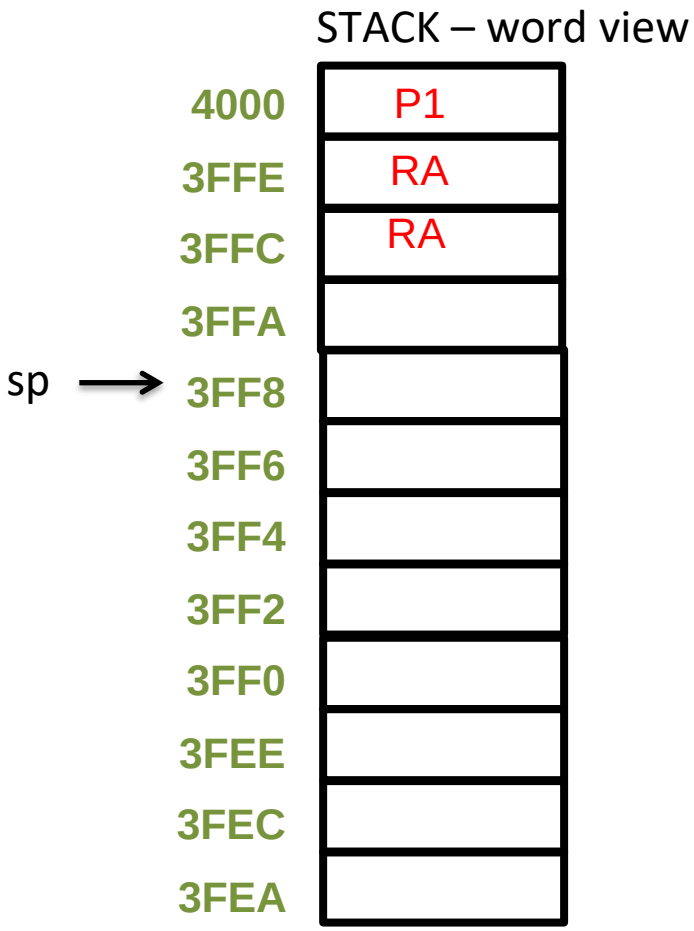
```
SUB   MOVEM.L   D1-D3/A0,-(SP)
          .
          .
          .
      MOVEM.L   (SP)+,D1-D3/A0
      RTS
```



Example

- Consider the following code

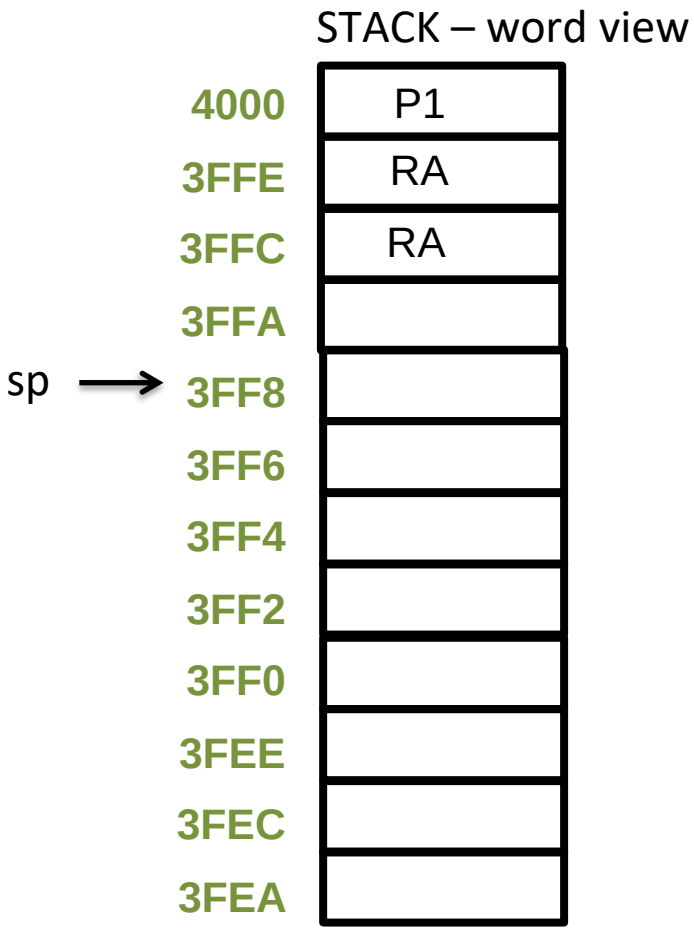
```
SUB   MOVEM.L   D1-D3/A0,-(SP)
          .
          .
          .
      MOVEM.L   (SP)+,D1-D3/A0
      RTS
```



Example

- Consider the following code

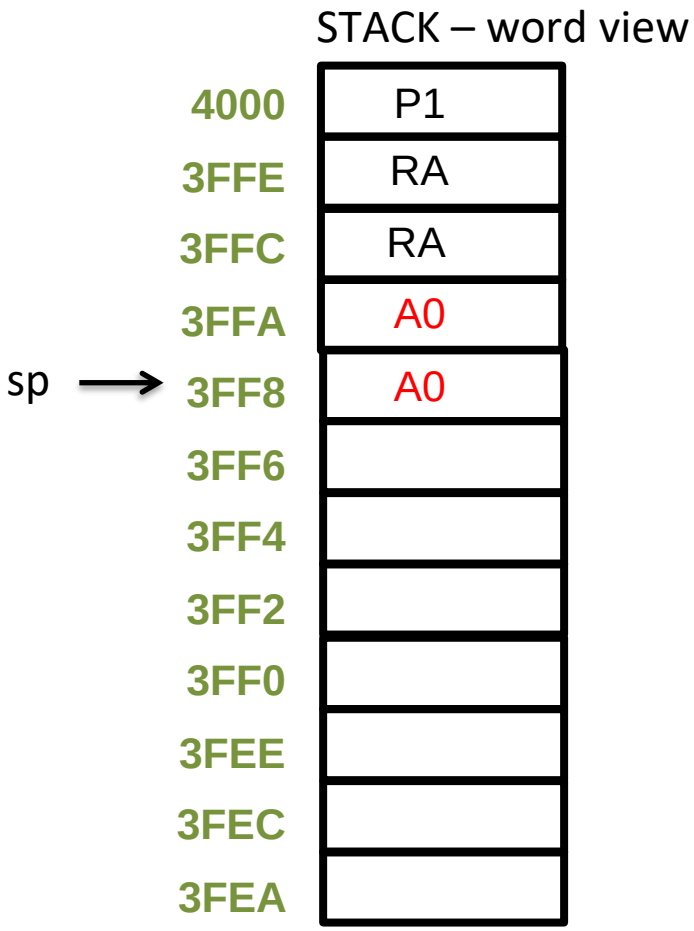
```
SUB   MOVEM.L   D1-D3/A0,-(SP)
          .
          .
          .
      MOVEM.L   (SP)+,D1-D3/A0
      RTS
```



Example

- Consider the following code

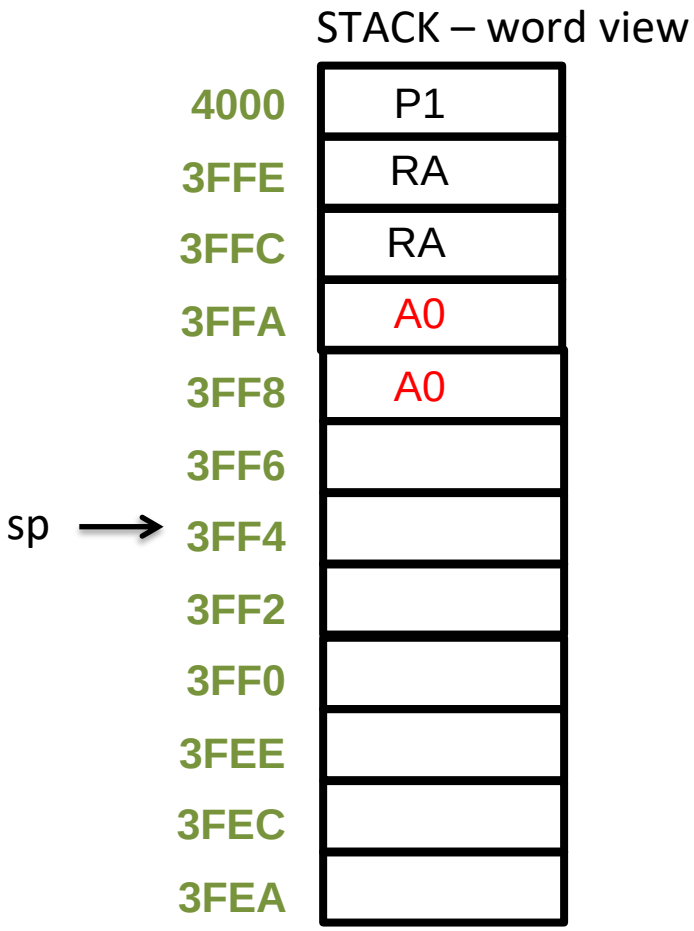
```
SUB   MOVEM.L   D1-D3/A0,-(SP)
          .
          .
          .
      MOVEM.L   (SP)+,D1-D3/A0
      RTS
```



Example

- Consider the following code

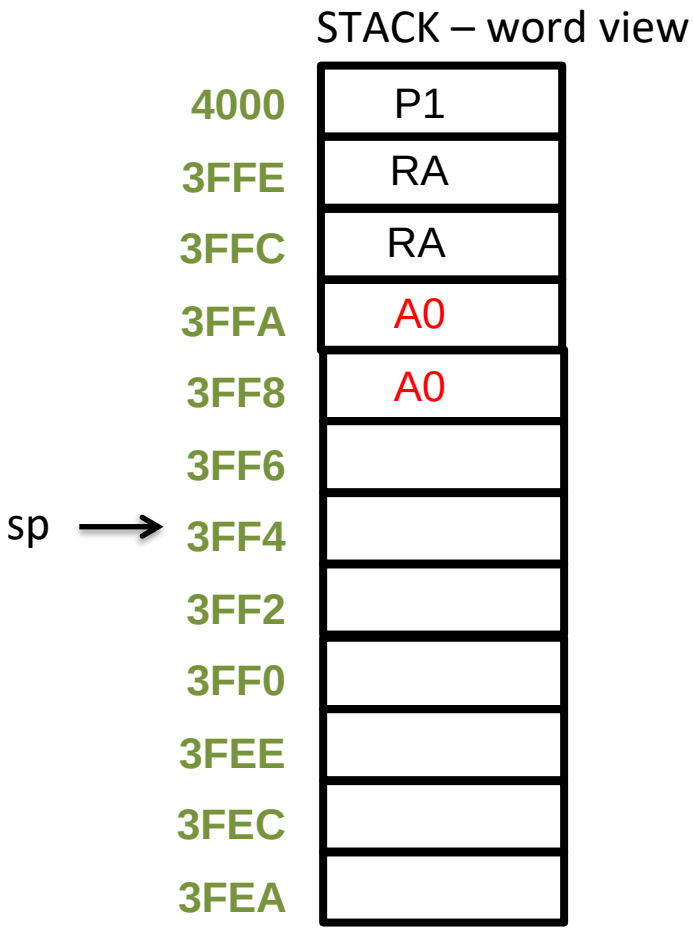
```
SUB   MOVEM.L   D1-D3/A0,-(SP)
          .
          .
          .
      MOVEM.L   (SP)+,D1-D3/A0
      RTS
```



Example

- Consider the following code

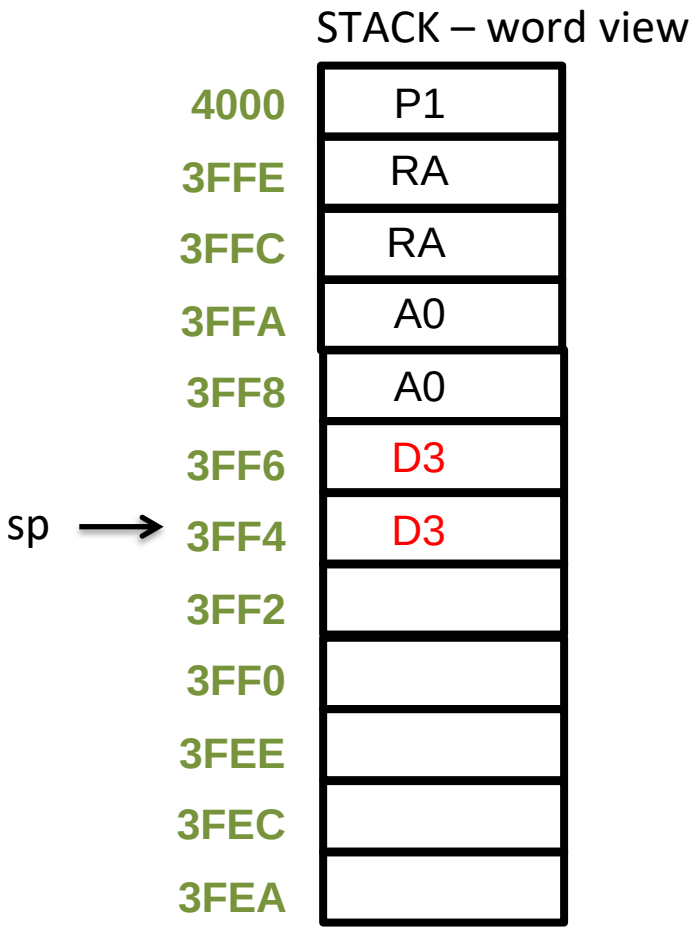
```
SUB    MOVEM.L   D1-D3/A0,-(SP)
         .
         .
         .
       MOVEM.L   (SP)+,D1-D3/A0
       RTS
```



Example

- Consider the following code

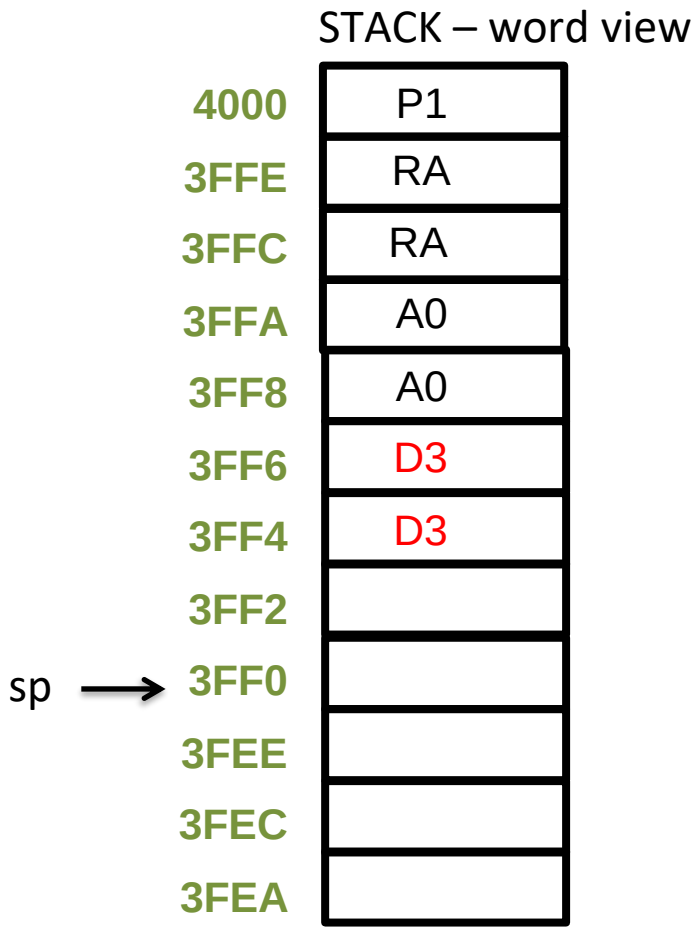
```
SUB   MOVEM.L   D1-D3/A0,-(SP)
          .
          .
          .
      MOVEM.L   (SP)+,D1-D3/A0
      RTS
```



Example

- Consider the following code

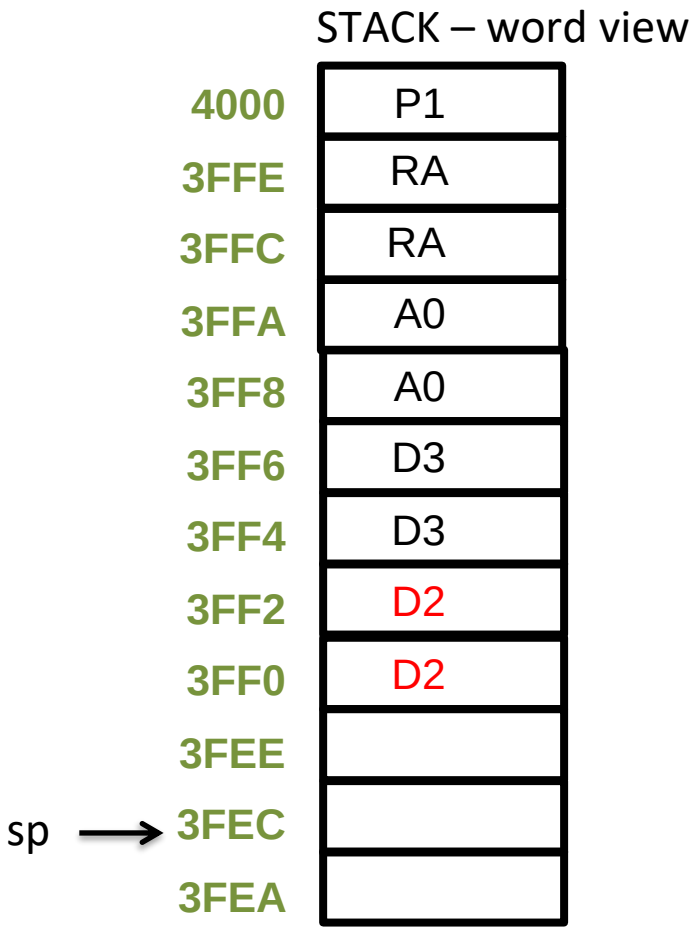
```
SUB    MOVEM.L   D1-D3/A0,-(SP)
         .
         .
         .
       MOVEM.L   (SP)+,D1-D3/A0
       RTS
```



Example

- Consider the following code

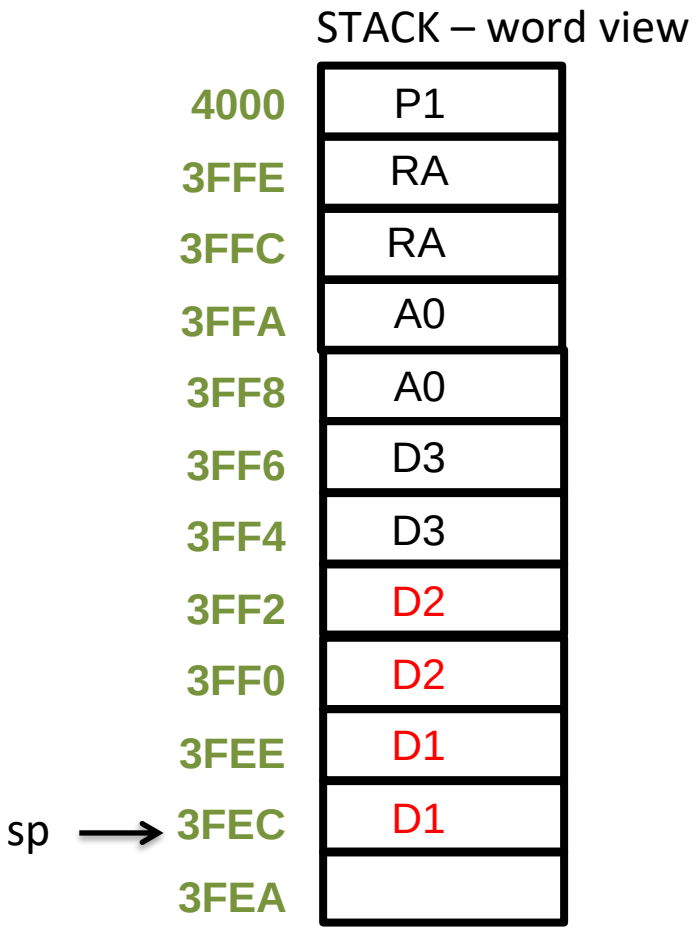
```
SUB    MOVEM.L   D1-D3/A0,-(SP)
         .
         .
         .
       MOVEM.L   (SP)+,D1-D3/A0
       RTS
```



Example

- Consider the following code

```
SUB    MOVEM.L   D1-D3/A0,-(SP)
         .
         .
         .
       MOVEM.L   (SP)+,D1-D3/A0
       RTS
```

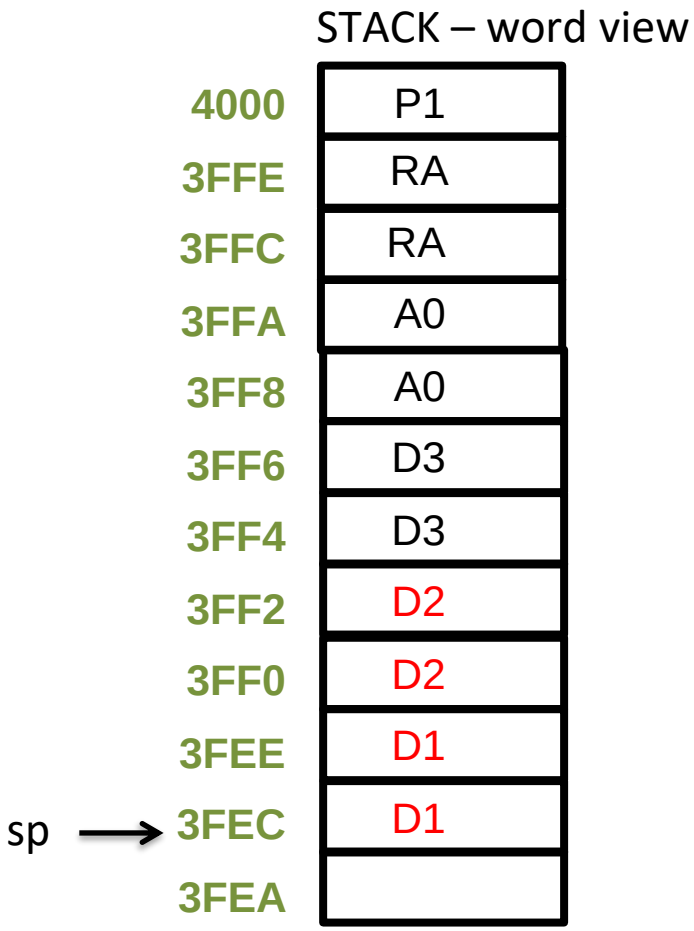


Example

- Consider the following code

```
SUB   MOVEM.L   D1-D3/A0,-(SP)
        .
        .
        .
      MOVEM.L   (SP)+,D1-D3/A0
      RTS
```

D1, D2, D3, and A0 can be safely modified, as their original contents will be restored just prior to leaving the subroutine

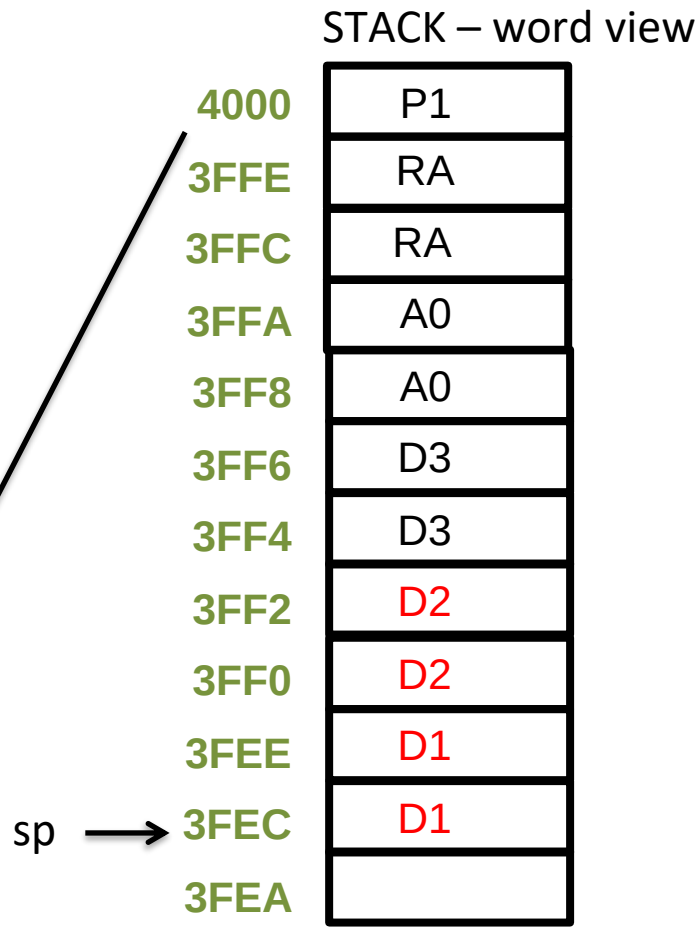


Example

- Consider the following code

```
SUB    MOVEM.L   D1-D3/A0,-(SP)
         .
         .
         .
       MOVEM.L   (SP)+,D1-D3/A0
       RTS
```

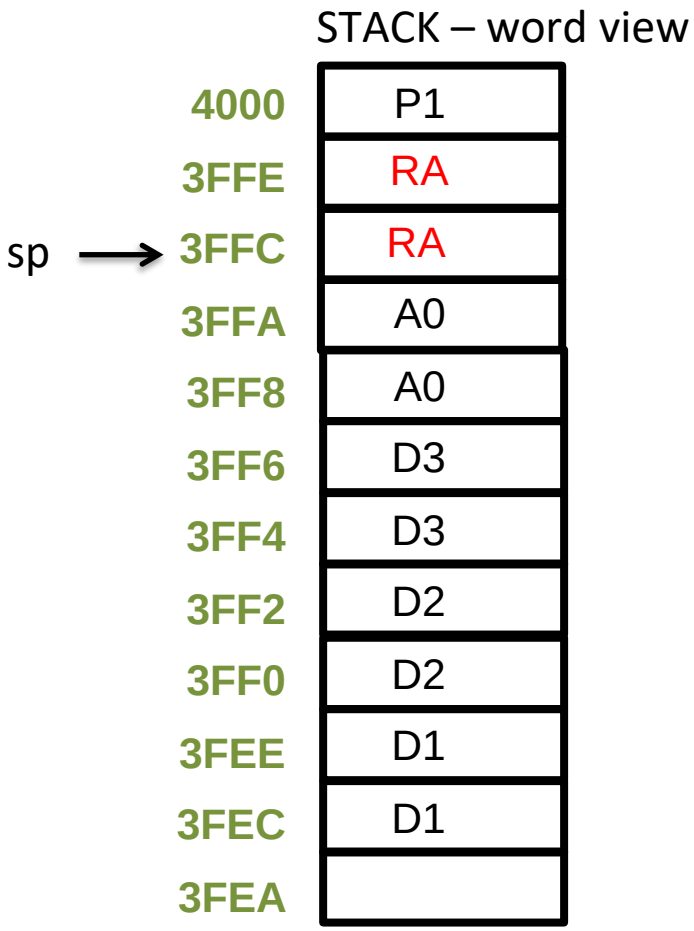
We saved 4 registers so P1 lives at SP + (4 x 4) + 4



Example

- Consider the following code

```
SUB   MOVEM.L   D1-D3/A0,-(SP)
          .
          .
          .
      MOVEM.L   (SP)+,D1-D3/A0
      RTS
```



Summary

- Generality
  - Can be called with any number of arguments
    - Passing arguments on the stack does this
    - Pass-by-value
    - Pass-by-reference
- Transparency
  - Leave the registers as you found them, except if a register is being used to return a value to the caller
    - MOVEM accomplishes this
- Recursive
  - A subroutine should be able to call itself if necessary
    - This is done using stack frames, something that we will discuss next time