Example Procedure Call

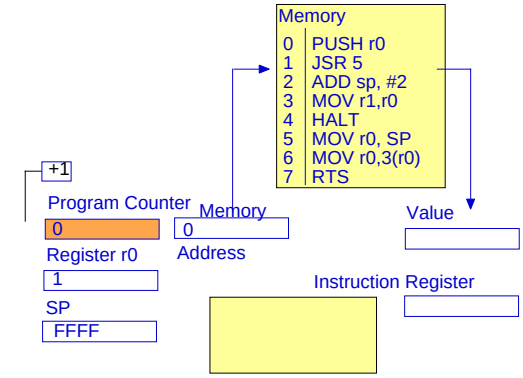
```
int simple(int a_value);
{
    return (a_value);
}
```

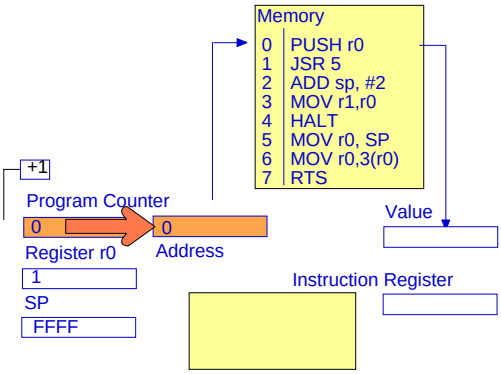
Procedure accepting one variable

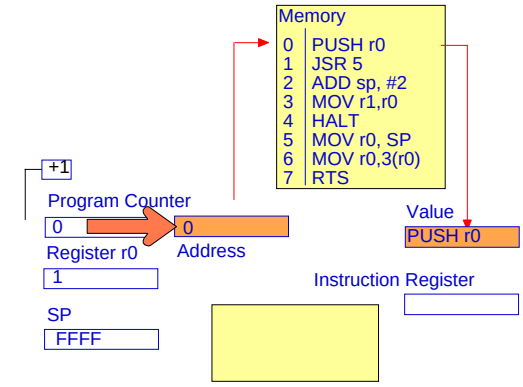
and

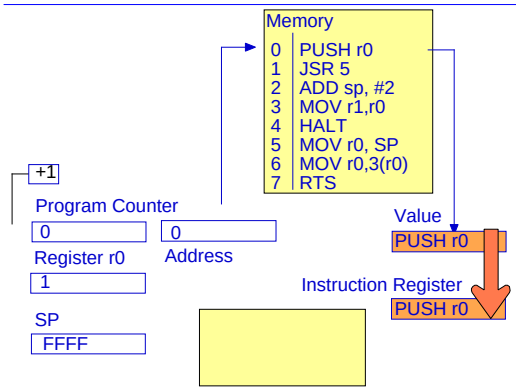
returning one value

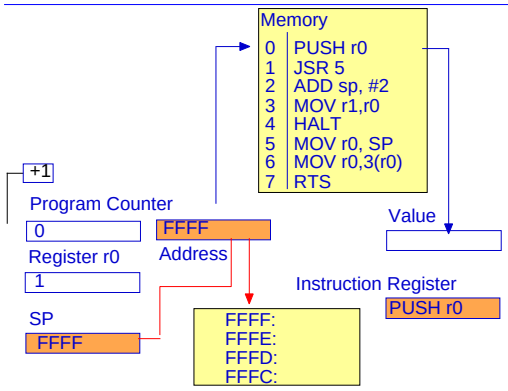
Suppose initial value is in r0
and final value is required in r1.

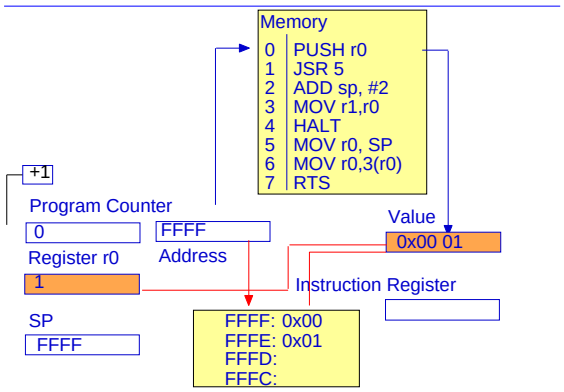


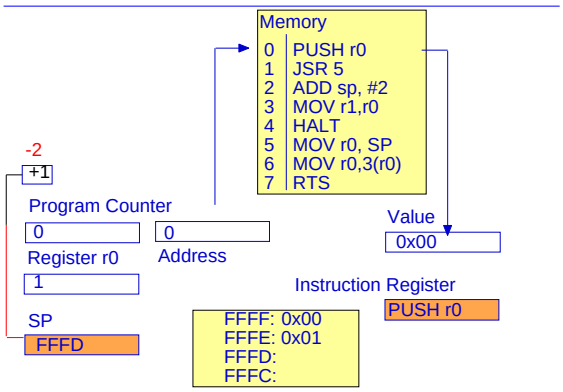


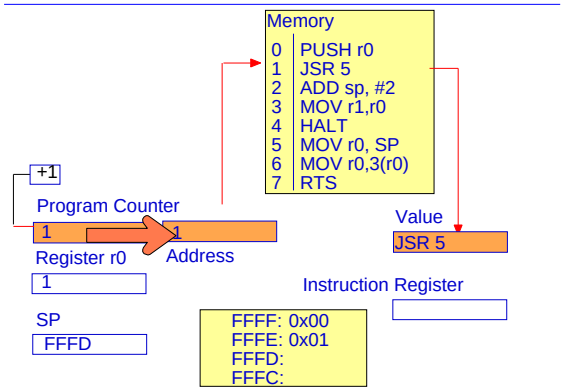


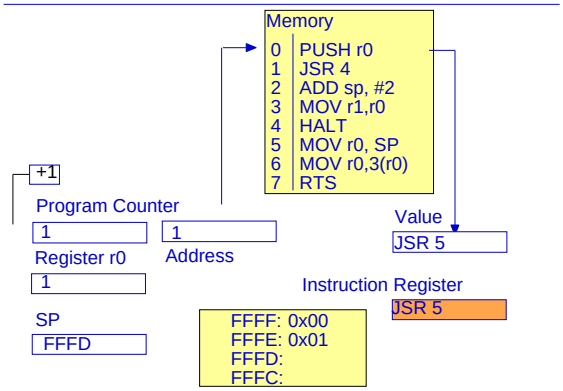


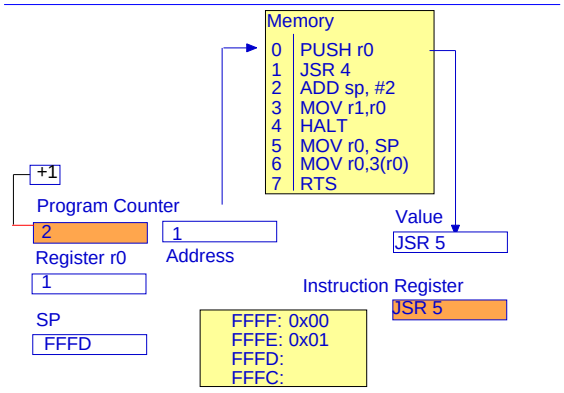


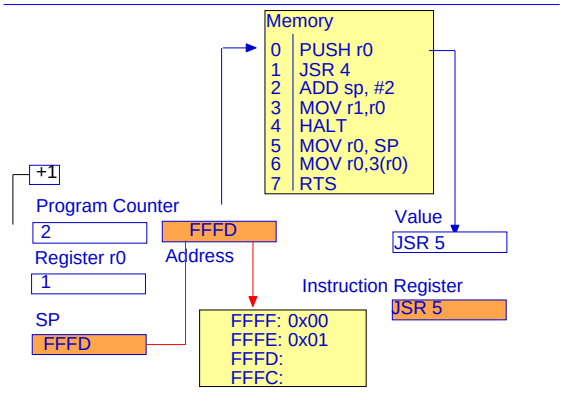

Memory
0 PUSH r0
1 JSR 5
2 ADD sp, #2
3 MOV r1,r0
4 HALT
5 MOV r0, SP
6 MOV r0,3(r0)
7 RTS
+1
Program Counter
0
FFFF
Address
Value
0x00 01
Register r0
1
Instruction Register
SP
FFFF
FFFF: 0x00
FFFE: 0x01
FFFD:
FFFC:

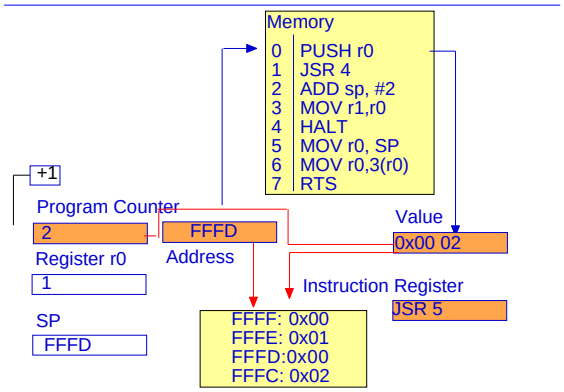

Memory
0 PUSH r0
1 JSR 5
2 ADD sp, #2
3 MOV r1,r0
4 HALT
5 MOV r0, SP
6 MOV r0,3(r0)
7 RTS
-2
+1
Program Counter
0
0
Address
Value
0x00
Register r0
1
Instruction Register
PUSH r0
SP
FFFD
FFFF: 0x00
FFFE: 0x01
FFFD:
FFFC:

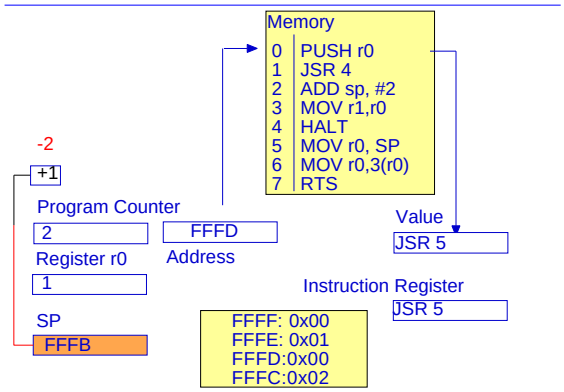

Memory
0 PUSH r0
1 JSR 5
2 ADD sp, #2
3 MOV r1,r0
4 HALT
5 MOV r0, SP
6 MOV r0,3(r0)
7 RTS
+1
Program Counter
1
1
Address
Value
JSR 5
Register r0
1
Instruction Register
SP
FFFD
FFFF: 0x00
FFFE: 0x01
FFFD:
FFFC:

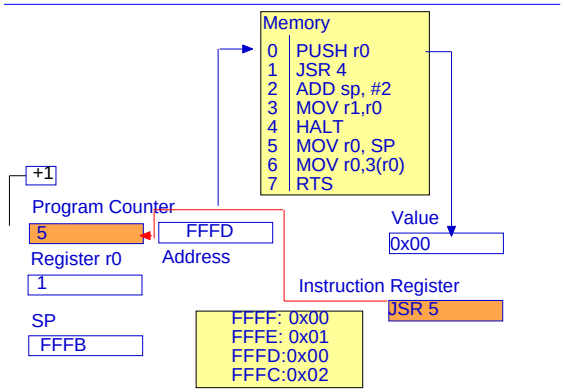

Memory
0 PUSH r0
1 JSR 4
2 ADD sp, #2
3 MOV r1,r0
4 HALT
5 MOV r0, SP
6 MOV r0,3(r0)
7 RTS
+1
Program Counter
1
1
Address
Value
JSR 5
Register r0
1
Instruction Register
JSR 5
SP
FFFD
FFFF: 0x00
FFFE: 0x01
FFFD:
FFFC:

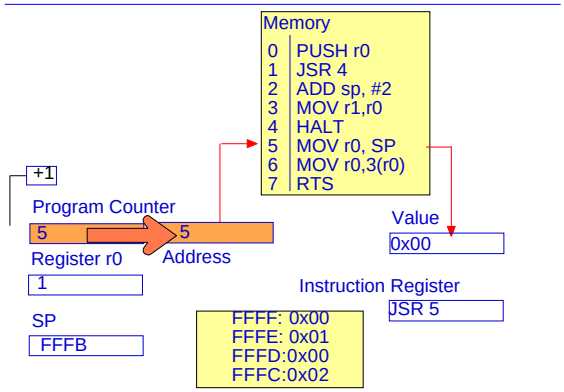

Memory
0 PUSH r0
1 JSR 4
2 ADD sp, #2
3 MOV r1,r0
4 HALT
5 MOV r0, SP
6 MOV r0,3(r0)
7 RTS
+1
Program Counter
2
1
Address
Value
JSR 5
Register r0
1
Instruction Register
JSR 5
SP
FFFD
FFFF: 0x00
FFFE: 0x01
FFFD:
FFFC:

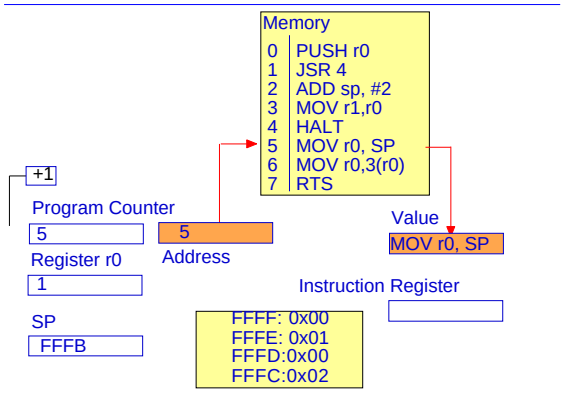

Memory
0 PUSH r0
1 JSR 4
2 ADD sp, #2
3 MOV r1,r0
4 HALT
5 MOV r0, SP
6 MOV r0,3(r0)
7 RTS
+1
Program Counter
2
FFFD
Address
Value
JSR 5
Register r0
1
Instruction Register
JSR 5
SP
FFFD
FFFF: 0x00
FFFE: 0x01
FFFD:
FFFC:

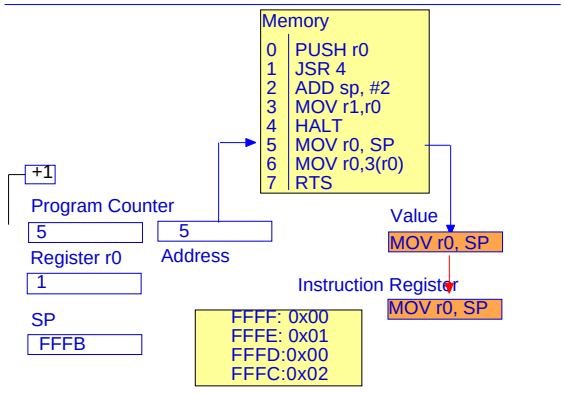

Memory
0 PUSH r0
1 JSR 4
2 ADD sp, #2
3 MOV r1,r0
4 HALT
5 MOV r0, SP
6 MOV r0,3(r0)
7 RTS
+1
Program Counter
2
FFFD
Address
Value
0x00 02
Register r0
1
Instruction Register
JSR 5
SP
FFFD
FFFF: 0x00
FFFE: 0x01
FFFD:0x00
FFFC: 0x02

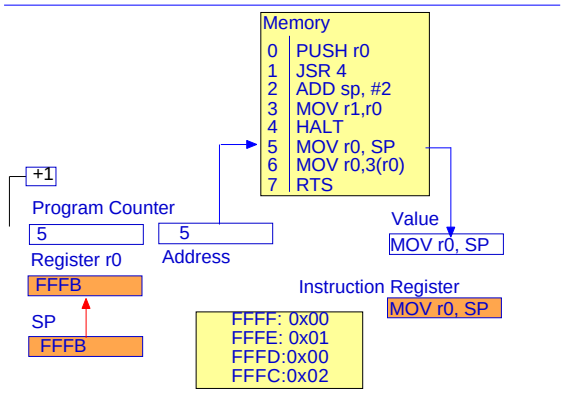

Memory
0 PUSH r0
1 JSR 4
2 ADD sp, #2
3 MOV r1,r0
4 HALT
5 MOV r0, SP
6 MOV r0,3(r0)
7 RTS
-2
+1
Program Counter
2
FFFD
Address
Value
JSR 5
Register r0
1
Instruction Register
JSR 5
SP
FFFB
FFFF: 0x00
FFFE: 0x01
FFFD:0x00
FFFC:0x02

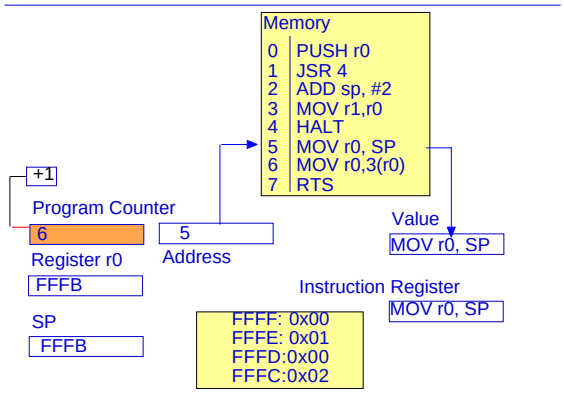

Memory
0 PUSH r0
1 JSR 4
2 ADD sp, #2
3 MOV r1,r0
4 HALT
5 MOV r0, SP
6 MOV r0,3(r0)
7 RTS
+1
Program Counter
5
FFFD
Address
Value
0x00
Register r0
1
Instruction Register
JSR 5
SP
FFFB
FFFF: 0x00
FFFE: 0x01
FFFD:0x00
FFFC:0x02

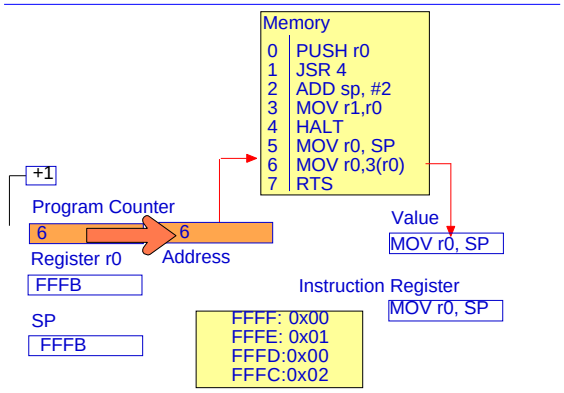

Memory
0 PUSH r0
1 JSR 4
2 ADD sp, #2
3 MOV r1,r0
4 HALT
5 MOV r0, SP
6 MOV r0,3(r0)
7 RTS
+1
Program Counter
5
5
Address
Value
0x00
Register r0
1
Instruction Register
JSR 5
SP
FFFB
FFFF: 0x00
FFFE: 0x01
FFFD:0x00
FFFC:0x02

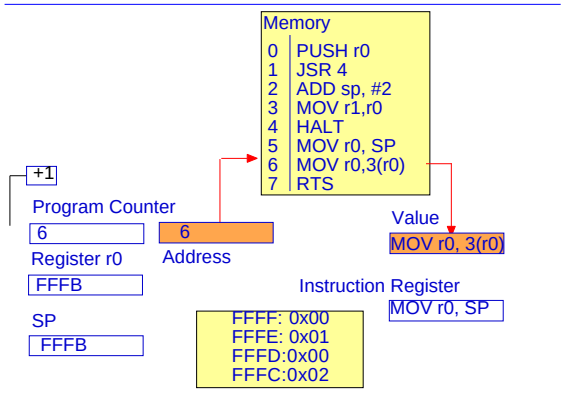

Memory
0 PUSH r0
1 JSR 4
2 ADD sp, #2
3 MOV r1,r0
4 HALT
5 MOV r0, SP
6 MOV r0,3(r0)
7 RTS
+1
Program Counter
5
5
Address
Value
MOV r0, SP
Register r0
1
Instruction Register
SP
FFFB
FFFF: 0x00
FFFE: 0x01
FFFD:0x00
FFFC:0x02

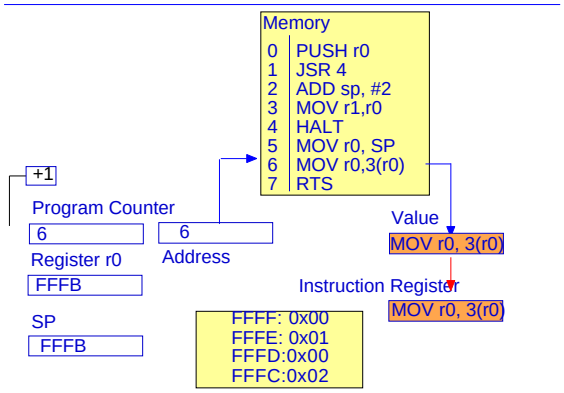

Memory
0 PUSH r0
1 JSR 4
2 ADD sp, #2
3 MOV r1,r0
4 HALT
5 MOV r0, SP
6 MOV r0,3(r0)
7 RTS
+1
Program Counter
5
5
Address
Value
MOV r0, SP
Register r0
1
Instruction Register
MOV r0, SP
SP
FFFB
FFFF: 0x00
FFFE: 0x01
FFFD:0x00
FFFC:0x02

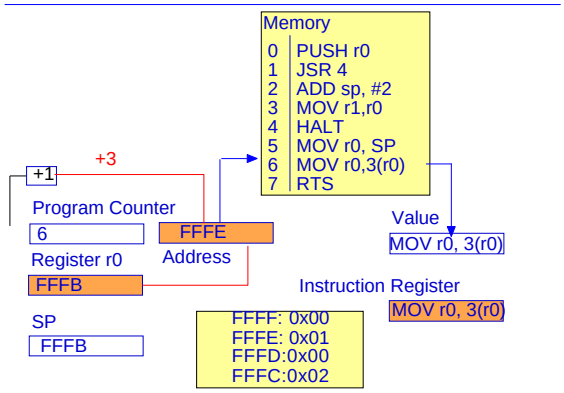

Memory
0 PUSH r0
1 JSR 4
2 ADD sp, #2
3 MOV r1,r0
4 HALT
5 MOV r0, SP
6 MOV r0,3(r0)
7 RTS
+1
Program Counter
5
5
Address
Value
MOV r0, SP
Register r0
FFFB
Instruction Register
MOV r0, SP
SP
FFFB
FFFF: 0x00
FFFE: 0x01
FFFD:0x00
FFFC:0x02

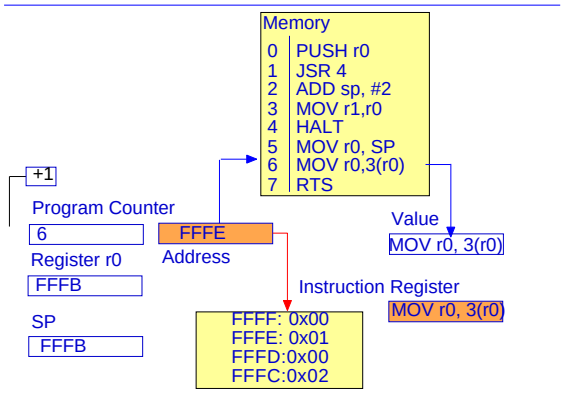

Memory
0 PUSH r0
1 JSR 4
2 ADD sp, #2
3 MOV r1,r0
4 HALT
5 MOV r0, SP
6 MOV r0,3(r0)
7 RTS
+1
Program Counter
6
5
Address
Value
MOV r0, SP
Register r0
FFFB
Instruction Register
MOV r0, SP
SP
FFFB
FFFF: 0x00
FFFE: 0x01
FFFD:0x00
FFFC:0x02

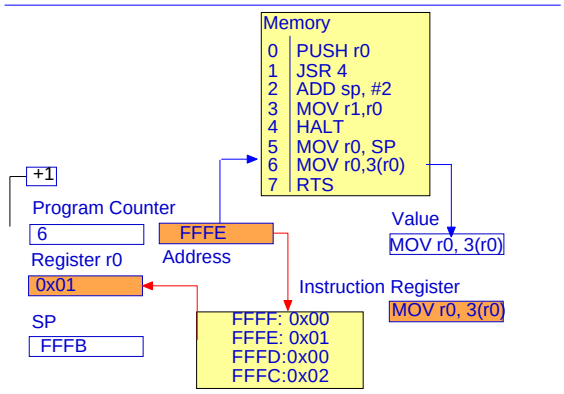

Memory
0 PUSH r0
1 JSR 4
2 ADD sp, #2
3 MOV r1,r0
4 HALT
5 MOV r0, SP
6 MOV r0,3(r0)
7 RTS
+1
Program Counter
6
6
Address
Value
MOV r0, SP
Register r0
FFFB
Instruction Register
MOV r0, SP
SP
FFFB
FFFF: 0x00
FFFE: 0x01
FFFD:0x00
FFFC:0x02

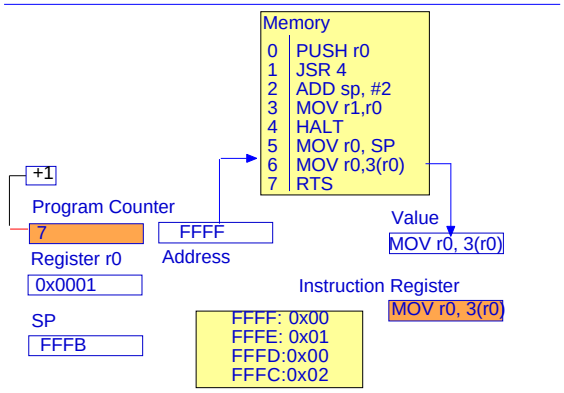

Memory
0 PUSH r0
1 JSR 4
2 ADD sp, #2
3 MOV r1,r0
4 HALT
5 MOV r0, SP
6 MOV r0,3(r0)
7 RTS
+1
Program Counter
6
6
Address
Value
MOV r0, 3(r0)
Register r0
FFFB
Instruction Register
MOV r0, SP
SP
FFFB
FFFF: 0x00
FFFE: 0x01
FFFD:0x00
FFFC:0x02

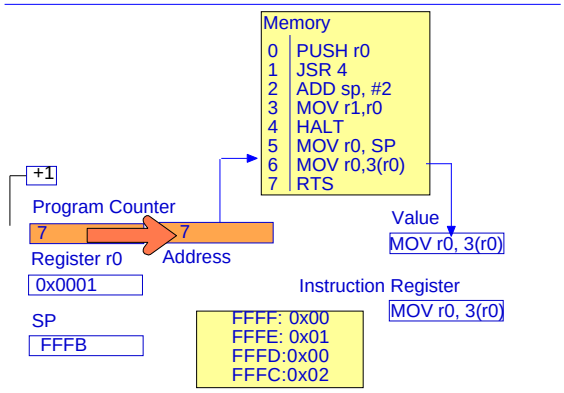

Memory
0 PUSH r0
1 JSR 4
2 ADD sp, #2
3 MOV r1,r0
4 HALT
5 MOV r0, SP
6 MOV r0,3(r0)
7 RTS
+1
Program Counter
6
6
Address
Value
MOV r0, 3(r0)
Register r0
FFFB
Instruction Register
MOV r0, 3(r0)
SP
FFFB
FFFF: 0x00
FFFE: 0x01
FFFD:0x00
FFFC:0x02

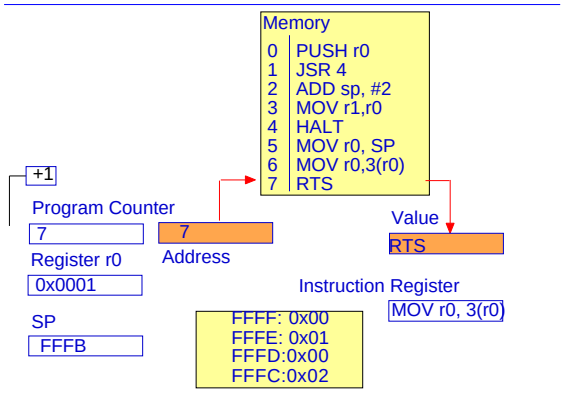

Memory
0 PUSH r0
1 JSR 4
2 ADD sp, #2
3 MOV r1,r0
4 HALT
5 MOV r0, SP
6 MOV r0,3(r0)
7 RTS
+3
+1
Program Counter
6
FFFE
Address
Value
MOV r0, 3(r0)
Register r0
FFFB
Instruction Register
MOV r0, 3(r0)
SP
FFFB
FFFF: 0x00
FFFE: 0x01
FFFD:0x00
FFFC:0x02

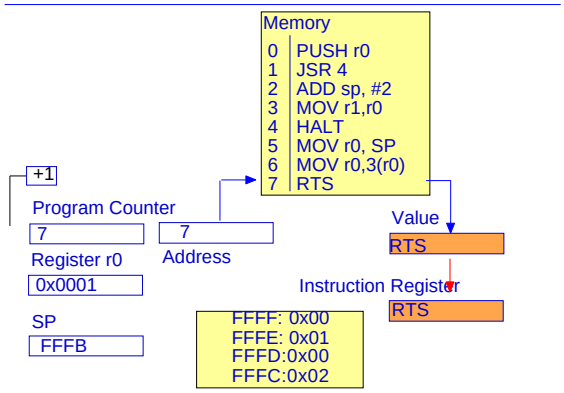

Memory
0 PUSH r0
1 JSR 4
2 ADD sp, #2
3 MOV r1,r0
4 HALT
5 MOV r0, SP
6 MOV r0,3(r0)
7 RTS
+1
Program Counter
6
FFFE
Address
Value
MOV r0, 3(r0)
Register r0
FFFB
Instruction Register
MOV r0, 3(r0)
SP
FFFB
FFFF: 0x00
FFFE: 0x01
FFFD:0x00
FFFC:0x02

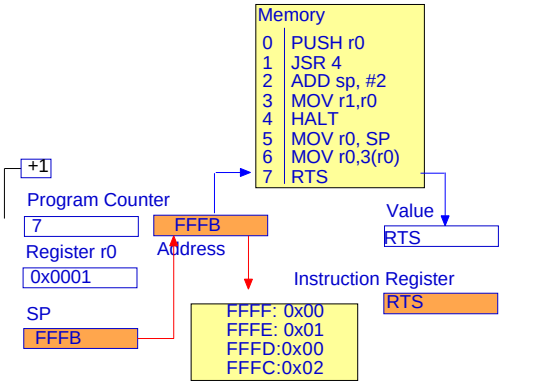

Memory
0 PUSH r0
1 JSR 4
2 ADD sp, #2
3 MOV r1,r0
4 HALT
5 MOV r0, SP
6 MOV r0,3(r0)
7 RTS
+1
Program Counter
6
FFFE
Address
Value
MOV r0, 3(r0)
Register r0
0x01
Instruction Register
MOV r0, 3(r0)
SP
FFFB
FFFF: 0x00
FFFE: 0x01
FFFD:0x00
FFFC:0x02

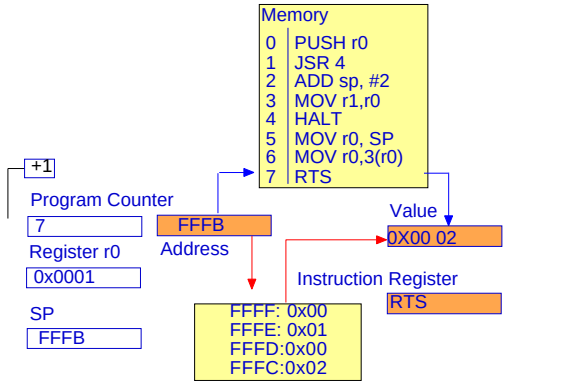

Memory
0 PUSH r0
1 JSR 4
2 ADD sp, #2
3 MOV r1,r0
4 HALT
5 MOV r0, SP
6 MOV r0,3(r0)
7 RTS
+1
Program Counter
7
FFFF
Address
Value
MOV r0, 3(r0)
Register r0
0x0001
Instruction Register
MOV r0, 3(r0)
SP
FFFB
FFFF: 0x00
FFFE: 0x01
FFFD:0x00
FFFC:0x02

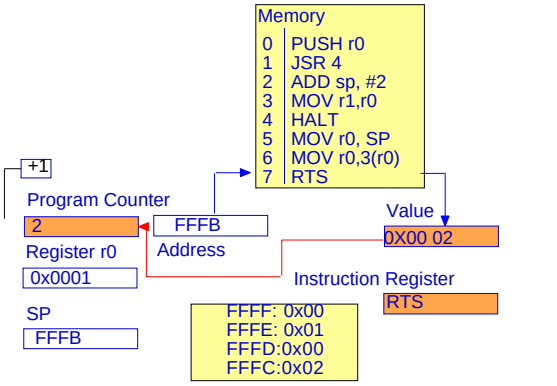

Memory
0 PUSH r0
1 JSR 4
2 ADD sp, #2
3 MOV r1,r0
4 HALT
5 MOV r0, SP
6 MOV r0,3(r0)
7 RTS
+1
Program Counter
7
7
Address
Value
MOV r0, 3(r0)
Register r0
0x0001
Instruction Register
MOV r0, 3(r0)
SP
FFFB
FFFF: 0x00
FFFE: 0x01
FFFD:0x00
FFFC:0x02

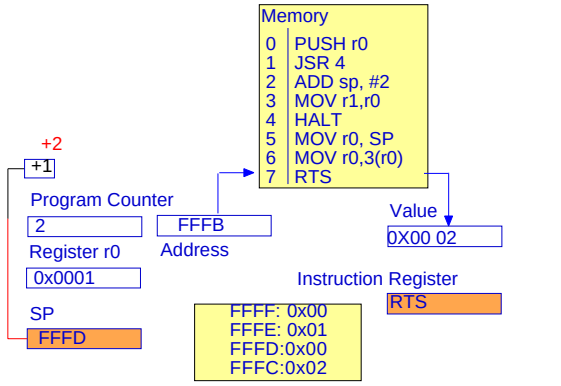

Memory
0 PUSH r0
1 JSR 4
2 ADD sp, #2
3 MOV r1,r0
4 HALT
5 MOV r0, SP
6 MOV r0,3(r0)
7 RTS
+1
Program Counter
7
7
Address
Value
RTS
Register r0
0x0001
Instruction Register
MOV r0, 3(r0)
SP
FFFB
FFFF: 0x00
FFFE: 0x01
FFFD:0x00
FFFC:0x02

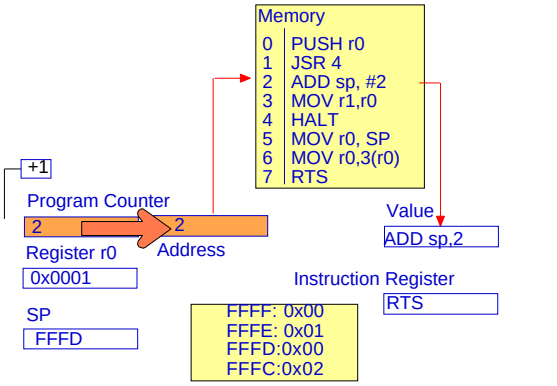

Memory
0 PUSH r0
1 JSR 4
2 ADD sp, #2
3 MOV r1,r0
4 HALT
5 MOV r0, SP
6 MOV r0,3(r0)
7 RTS
+1
Program Counter
7
7
Address
Value
RTS
Register r0
0x0001
Instruction Register
RTS
SP
FFFB
FFFF: 0x00
FFFE: 0x01
FFFD:0x00
FFFC:0x02

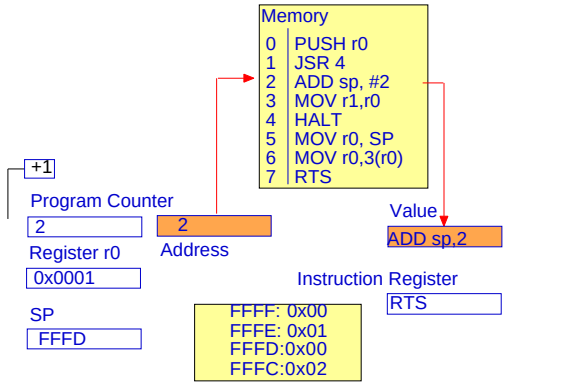

Memory
0 PUSH r0
1 JSR 4
2 ADD sp, #2
3 MOV r1,r0
4 HALT
5 MOV r0, SP
6 MOV r0,3(r0)
7 RTS
+1
Program Counter
7
FFFB
Address
Value
RTS
Register r0
0x0001
Instruction Register
RTS
SP
FFFB
FFFF: 0x00
FFFE: 0x01
FFFD:0x00
FFFC:0x02

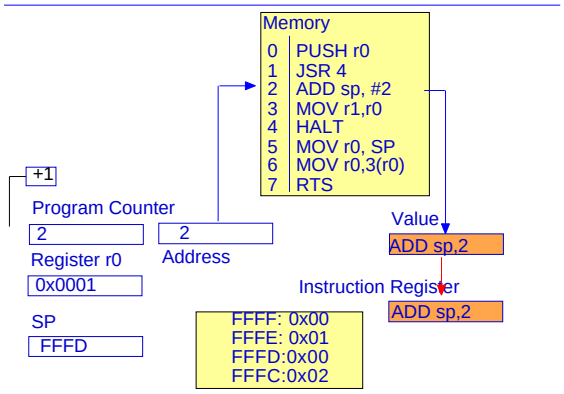

Memory
0 PUSH r0
1 JSR 4
2 ADD sp, #2
3 MOV r1,r0
4 HALT
5 MOV r0, SP
6 MOV r0,3(r0)
7 RTS
+1
Program Counter
7
FFFB
Address
Value
0X00 02
Register r0
0x0001
Instruction Register
RTS
SP
FFFB
FFFF: 0x00
FFFE: 0x01
FFFD:0x00
FFFC:0x02

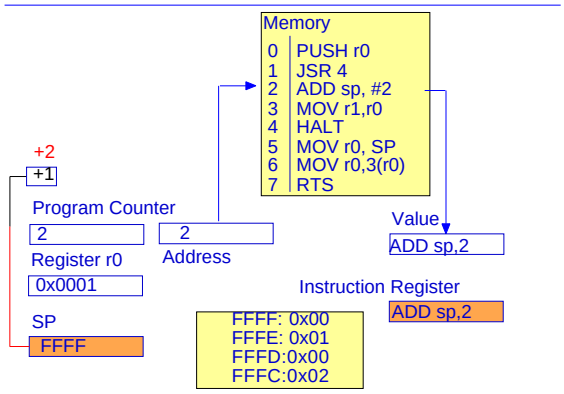

Memory
0 PUSH r0
1 JSR 4
2 ADD sp, #2
3 MOV r1,r0
4 HALT
5 MOV r0, SP
6 MOV r0,3(r0)
7 RTS
+1
Program Counter
2
FFFB
Address
Value
0X00 02
Register r0
0x0001
Instruction Register
RTS
SP
FFFB
FFFF: 0x00
FFFE: 0x01
FFFD:0x00
FFFC:0x02

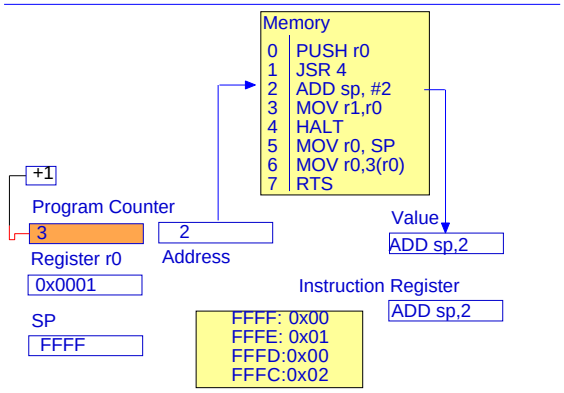

Memory
0 PUSH r0
1 JSR 4
2 ADD sp, #2
3 MOV r1,r0
4 HALT
5 MOV r0, SP
6 MOV r0,3(r0)
7 RTS
+2
+1
Program Counter
2
FFFB
Address
Value
0X00 02
Register r0
0x0001
Instruction Register
RTS
SP
FFFD
FFFF: 0x00
FFFE: 0x01
FFFD:0x00
FFFC:0x02


Memory
0 PUSH r0
1 JSR 4
2 ADD sp, #2
3 MOV r1,r0
4 HALT
5 MOV r0, SP
6 MOV r0,3(r0)
7 RTS
+1
Program Counter
2
2
Address
Value
ADD sp,2
Register r0
0x0001
Instruction Register
RTS
SP
FFFD
FFFF: 0x00
FFFE: 0x01
FFFD:0x00
FFFC:0x02


Memory
0 PUSH r0
1 JSR 4
2 ADD sp, #2
3 MOV r1,r0
4 HALT
5 MOV r0, SP
6 MOV r0,3(r0)
7 RTS
+1
Program Counter
2
2
Address
Value
ADD sp,2
Register r0
0x0001
Instruction Register
RTS
SP
FFFD
FFFF: 0x00
FFFE: 0x01
FFFD:0x00
FFFC:0x02


Memory
0 PUSH r0
1 JSR 4
2 ADD sp, #2
3 MOV r1,r0
4 HALT
5 MOV r0, SP
6 MOV r0,3(r0)
7 RTS
+1
Program Counter
2
Register r0
0x0001
SP
FFFD
2
Address
Value
ADD sp,2
Instruction Register
ADD sp,2
FFFF: 0x00
FFFE: 0x01
FFFD:0x00
FFFC:0x02


Memory
0 PUSH r0
1 JSR 4
2 ADD sp, #2
3 MOV r1,r0
4 HALT
5 MOV r0, SP
6 MOV r0,3(r0)
7 RTS
+2
+1
Program Counter
2
Register r0
0x0001
SP
FFFF
2
Address
Value
ADD sp,2
Instruction Register
ADD sp,2
FFFF: 0x00
FFFE: 0x01
FFFD:0x00
FFFC:0x02


Memory
0 PUSH r0
1 JSR 4
2 ADD sp, #2
3 MOV r1,r0
4 HALT
5 MOV r0, SP
6 MOV r0,3(r0)
7 RTS
+1
Program Counter
3
Register r0
0x0001
SP
FFFF
2
Address
Value
ADD sp,2
Instruction Register
ADD sp,2
FFFF: 0x00
FFFE: 0x01
FFFD:0x00
FFFC:0x02