Subroutine call

Computer System Architecture

By

M. Morris Mano

Prof.S.Meenatchi, SITE, VIT

Subroutine call and return

- **Subroutine:** a self-contained sequence of inst that perform a given computational task.
- Called many times.
- Branch is executed to begin.
- After execution, a branch is made back to main program.
- Inst that transfer control to a subroutine is known as **call subroutine, jump to subroutine, branch to subroutine** or **branch and save return address (BSA).**
- Call subroutine inst consist of an operation code together with an address that specifies the beginning of subroutine.

Execution of call subroutine instruction

Inst is executed by performing two operation

1. Store the address of the next inst available in the PC in a temporary location so the subroutine knows where to return.
2. Control is transferred to the beginning of the subroutine. Last inst of every subroutine commonly called **return form subroutine**, transfer the return address from the temporary location into the PC.

Program to Demonstrate the use of Subroutines

```
Location
                ORG     100     / Main Program
100             LDA     X       / Load X
101             BSA     SH4     / Call SH4 with X
102             STA     X       / Store result X
103             LDA     Y       / Load Y
104             BSA     SH4     / Call SH4 with Y
105             STA     Y       / Store result Y
106             HLT
107     X,      HEX     1234    / Result = 2340
108     Y,      HEX     4321    / Result = 3210
                                / Subr
109     SH4,    HEX     0       / Save Return Address
10A             CIL
10B             CIL
10C             CIL
10D             CIL
10E             AND     MSK     / Mask lower 4 bit
10F             BUN     SH4 I   / Indirect Return to main
110     MSK,    HEX     FFF0    / Mask pattern
110             END
```

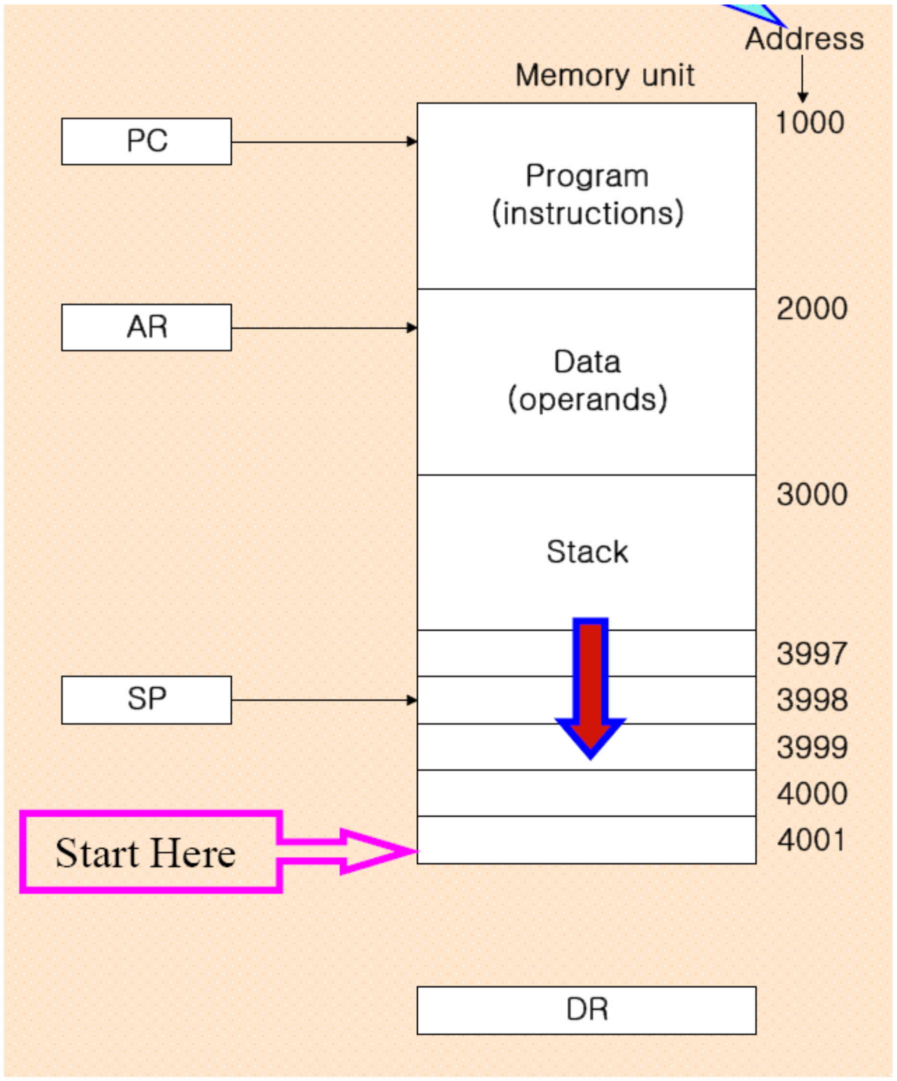
Subroutine CALL hear

X = 102
Y = 105

Different location of return address

- Some computers store return address
  - In the first memory location of the subroutine.
  - In a fixed location in memory.
  - In a processor register.
  - In a memory stack
- Memory stack- efficient
- Advantage:
  - When a subroutine is called, the sequential return addresses can be pushed into the stack.
  - Return from subroutine pop contents of TOS and transferred to PC.

Memory Stack



Implementation of Subroutine call and return

- CALL to subroutine

$SP \leftarrow SP - 1$	: Decrement SP
$M[SP] \leftarrow PC$	: Push content of PC into stack
$PC \leftarrow EA$	: Transfer control to subroutine.

- RETURN from last subroutine

$PC \leftarrow M[SP]$	: POP stack and transfer to PC.
$SP \leftarrow SP + 1$	: Increment SP

- Recursive subroutine
  – It is a subroutine that calls itself.