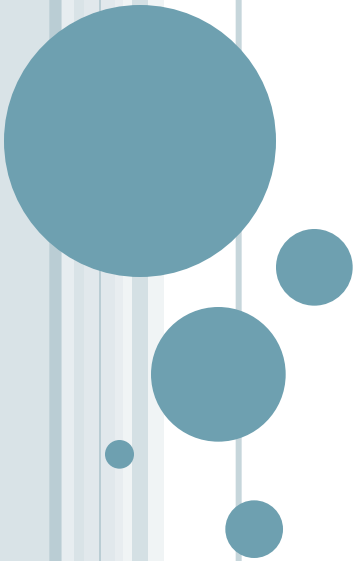
# Subroutine call & Return



# Subroutine

- Subroutine is a self-contained sequence of instructions that performs a given computational task.

- Call Subroutine.

- When called,
  - Push [PC] to TOS
  - [PC] ← 1st address of subroutine
  - Execute SUB
  - Finally execute RET instruction
    - Pop TOS to PC
    - Continue with the instruction which is next to SUB call.

# Locations to store the return address

- First memory location of the subroutine
- Fixed location in memory
- Processor registers
- Memory stack – best option
  - Adv: In the case of sequential calls to subroutines. So, the top of the stack always has the return address of the subroutine which to be returned first.

# Micro-operations

**Call:**

SP ← SP – 1 // decrement stack pointer

M[SP] ← PC // push content of PC onto the stack

PC ← effective address /* transfer control to the subroutine */

**Return:**

PC ← M[SP] // pop stack and transfer to PC

SP ← SP + 1 // increment stack pointer

- **Recursive SUB:-** Subroutine that calls itself
- **Subroutine nesting:-** One subroutine that calls another.
- If only one register or memory location is used to hold the return address, when subroutine is called recursively, it destroys the previous return address.
- So, stack is the good solution for this problem.

# References

Text Book

- William Stallings "Computer Organization and architecture" Prentice Hall, 7th edition, 2006