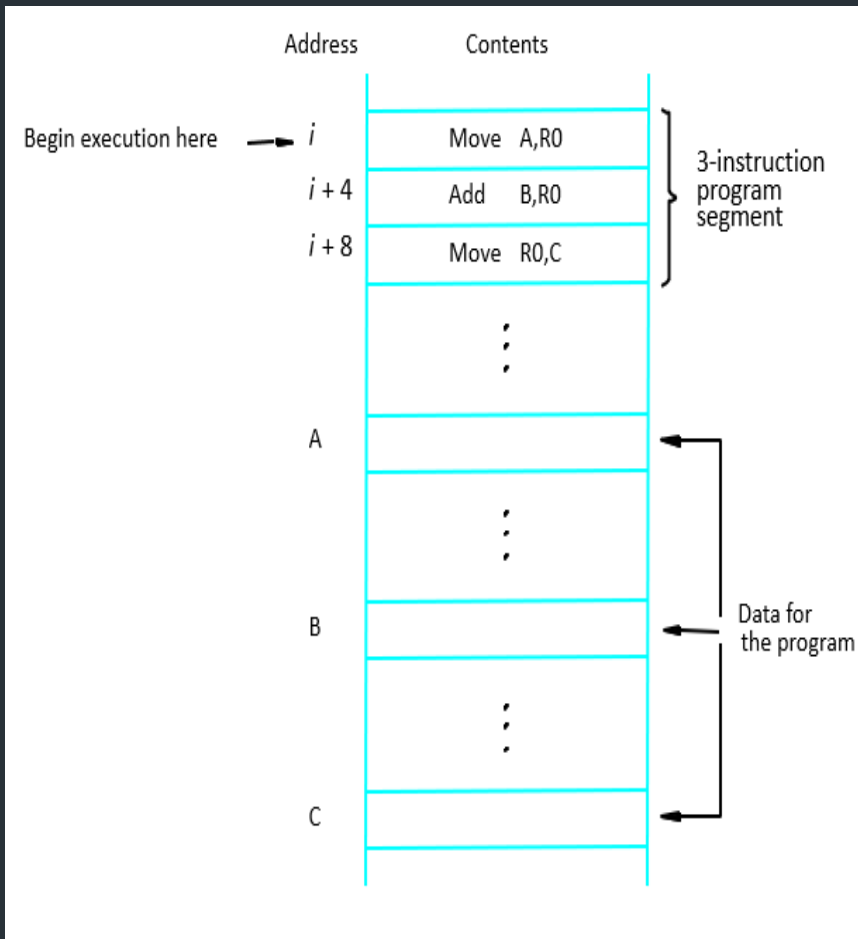


Instruction Execution and Sequencing

Module 3 Part2A



A program for $c \leftarrow [A] + [B]$

Instruction execution...

- Program Counter: It holds the address of next instruction
 - 'i' is placed into the PC
 - It is a Two phase procedure :
 - Instruction Fetch
 - Instruction Execution
 - During instruction execution, PC is incremented by 4 ,i+4, i+8,i+12,..

Assembly Language

- machine instructions are represented by pattern of 0's and 1's. such patterns are awkward to deal with when discussing or preparing programs
- Therefore we use symbolic names to represent the patterns
- We use normal words like Move, Add, Increment, and Branch for the instruction operations to represent the corresponding binary code patterns
- When writing programs for specific computer, such words are normally replaced by mnemonics, such as MOV, ADD, INC, and BR.

Assembly Language

- similarly we use R3 to refer to register 3, and LOC to refer to memory location
- A complete set of such symbolic names and rules for their use constitute a programming language, called as Assembly language.
- Programs written in assembly language can be automatically translated into a sequence of machine instructions by a program called an Assembler.

Assembly Language

- A user program is usually entered into the computer through a keyboard and stored either in the memory or on a magnetic disk.
- At this point, the user program is simply a set of lines of alphanumerical characters.
- When the assembler code is executed, it reads the user program, analyzes it, and then generates the desired machine language program.
- The user program is called source program and the assembled machine language program is called Object program.



SUBROUTINES and CALL

Subroutines

- Particular **subtasks** performed (many times) using different data values. The sub task is called subroutine.
- Program branches to a subroutine: *calling a subroutine* (**CALL instruction**)
- By executing 'RET' (return instruction); it returns to the main program.
- CALL- RET mechanism is 'SUBROUTINE LINKAGE METHOD'
- Link register- saves the return address at specified location

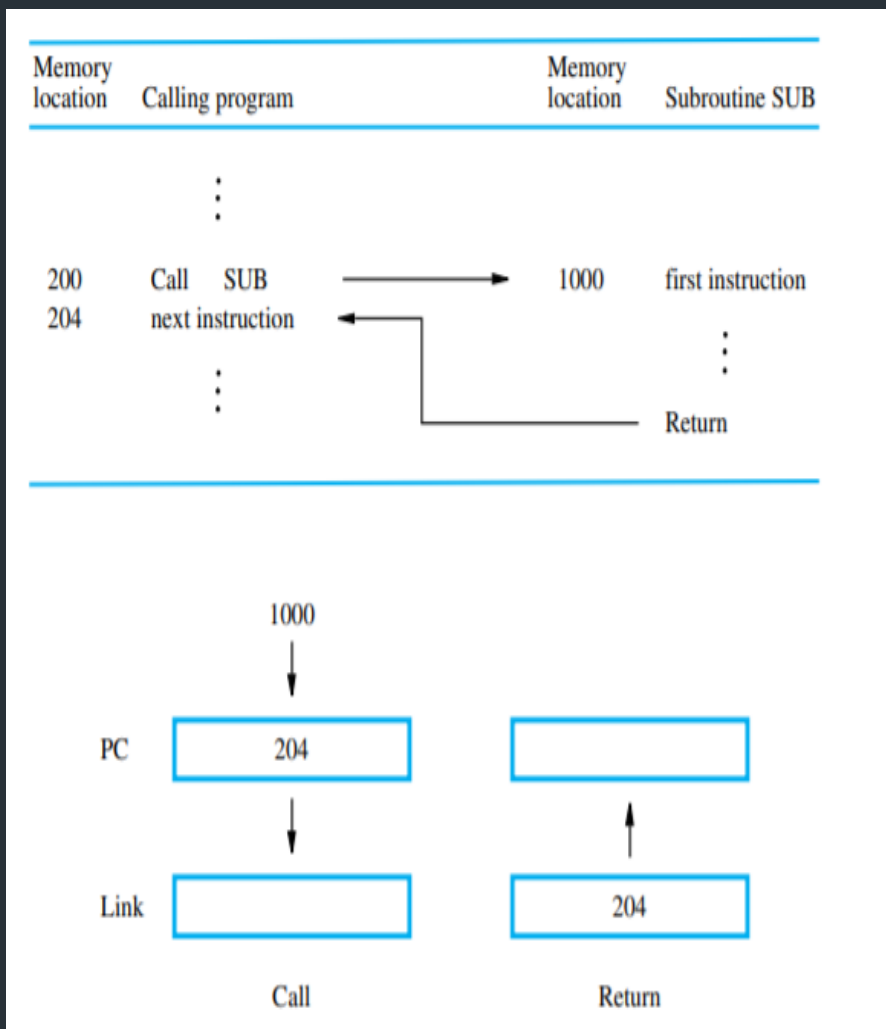
Subroutines

The Call instruction is just a special branch instruction that performs the following operations:

- Store the contents of the PC in the link register
- Branch to the target address specified by the instruction

The Return instruction is a special branch instruction that performs the operation:

- Branch to the address contained in the link register



Calling program

Move	N,R1	R1 serves as a counter.
Move	#NUM1,R2	R2 points to the list.
Call	LISTADD	Call subroutine.
Move	R0,SUM	Save result.
:		

Subroutine

LISTADD	Clear	R0	Initialize sum to 0.
LOOP	Add	(R2)+,R0	Add entry from list.
	Decrement	R1	
	Branch>0	LOOP	
	Return		Return to calling program.