



# SNS COLLEGE OF ENGINEERING

Kurumbapalayam (PO), Coimbatore - 641 107

Accredited by NAAC-UGC with 'A' Grade

Approved by AICTE, Recognized by UGC & Affiliated to Anna University, Chennai

## DEPARTMENT OF INFORMATION TECHNOLOGY

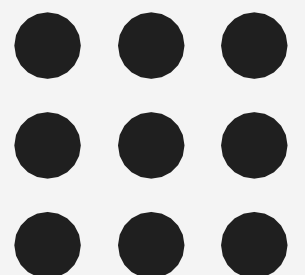
**COURSE NAME: 19IT301 COMPUTER ORGANIZATION**

**AND ARCHITECTURE**

**II YEAR/ III SEM**

**Unit 1 : BASIC STRUCTURE OF COMPUTERS Topic 5:**

**Instruction and Instruction Sequencing**





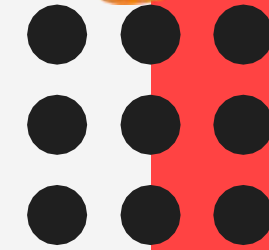
# “Must-Perform” Operations

- Data transfers between the memory and the processor registers
- Arithmetic and logic operations on data
- Program sequencing and control
- I/O transfers





# Register Transfer Notation



- Identify a location by a symbolic name standing for its hardware binary address (LOC, R0, DATAIN, ...)
- Contents of a location are denoted by placing square brackets around the name of the location
$$R1 \leftarrow [LOC]$$
$$R3 \leftarrow [R1] + [R2]$$
- This type of notation is Register Transfer Notation (RTN)



# Assembly Language Notation



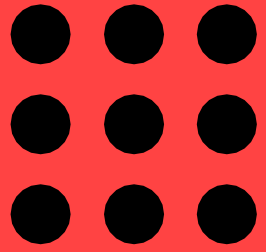
- Assembly language(symbolic machine code) takes complete control over the system and its resources.
- Represent machine instructions and programs.
- Move LOC, R1
- Add R1, R2, R3



# Instruction Formats



- Three-Address Instructions
  - Format: **Operation Source1, Source2, Destination**
  - ADD R2, R3, R1  $R1 \leftarrow [R2] + [R3]$
- Two-Address Instructions
  - Format: **Operation Source, Destination**
  - ADD R2, R1  $R1 \leftarrow [R1] + [R2]$
- One-Address Instructions
  - ADD M  $AC \leftarrow [AC] + [M]$
  - Load A
  - Store A



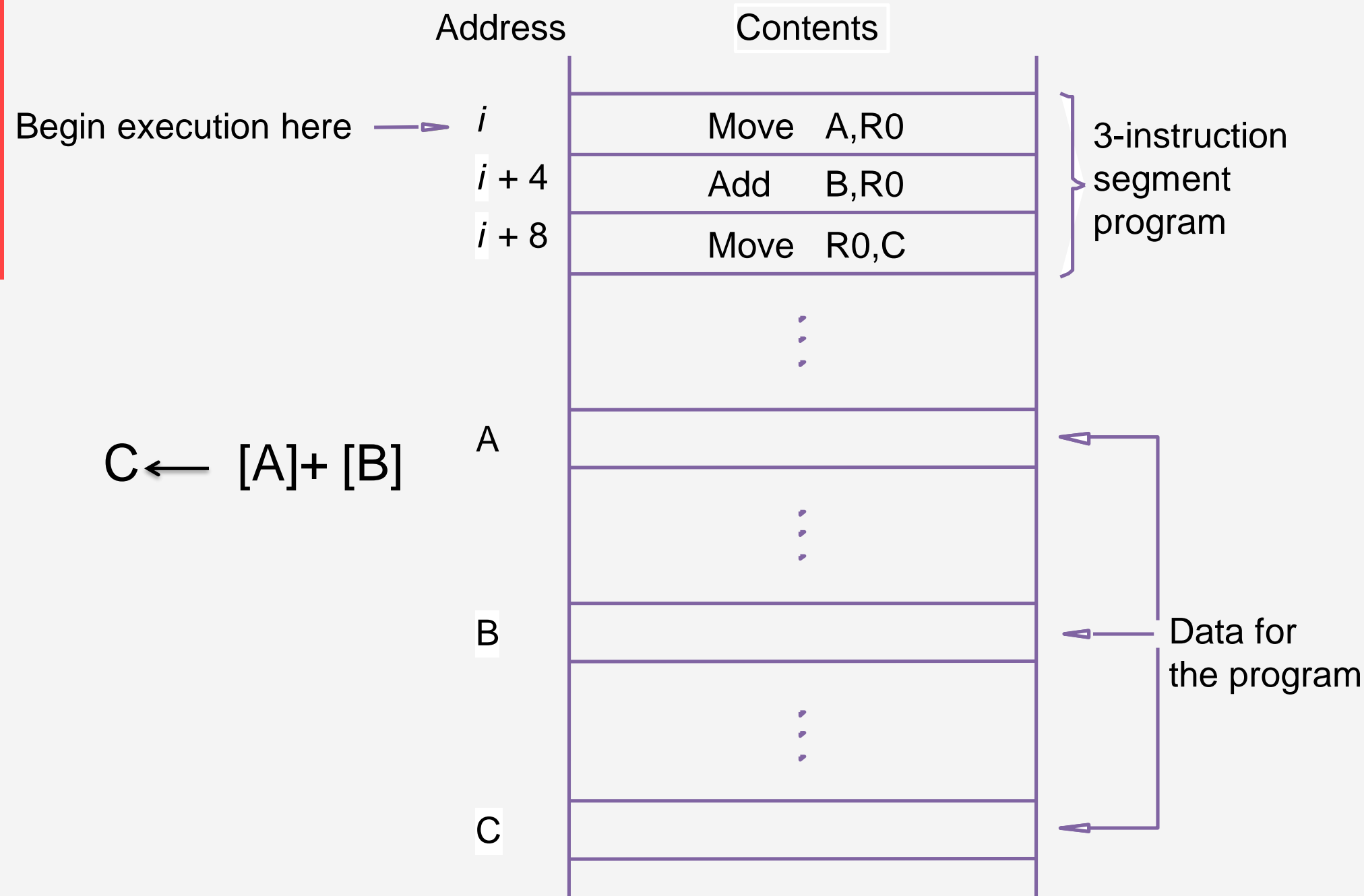
# Instruction Formats



Example: Evaluate  $C = A+B$  using processor registers,

Move	A,	$R_i$
Move	B,	$R_j$
Add	$R_i,$	$R_j$
Move	$R_j,$	C

# Instruction Execution and Straight-Line Sequencing



## Assumptions:

- One memory operand per instruction
- 32-bit word length
- Memory is byte addressable
- Full memory address can be directly specified in a single-word instruction

## Two-phase procedure

- Instruction fetch
- Instruction execute

Processor control circuits use PC to fetch and execute instructions, one at a time, in the order of increasing addresses **called straight-line sequencing**





# Branching



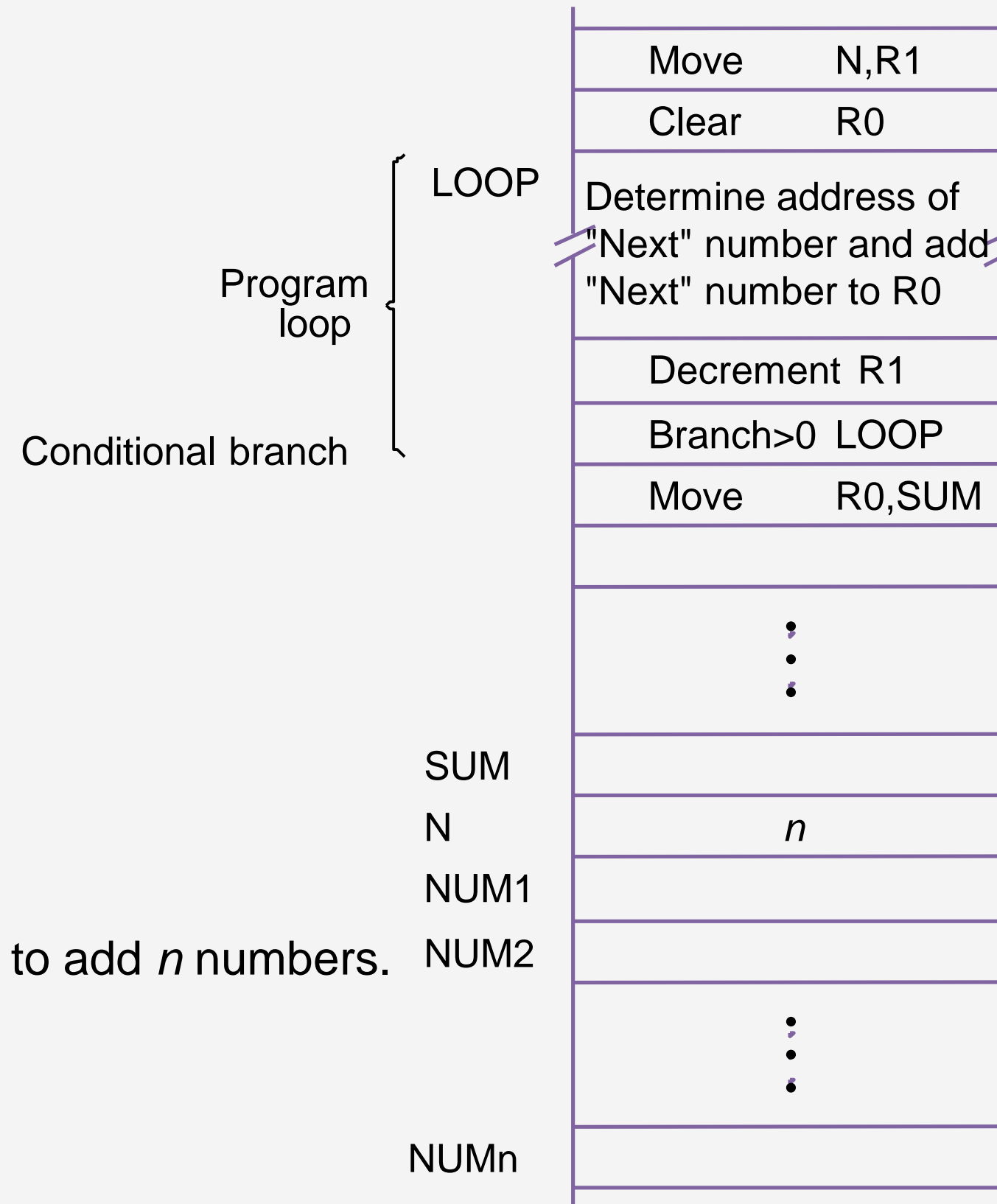
$i$   
 $i + 4$   
 $i + 8$   
  
  
  
 $i + 4n - 4$   
 $i + 4n$   
  
  
  
SUM  
NUM1  
NUM2  
  
  
  
NUM  $n$

Move	NUM1,R0
Add	NUM2,R0
Add	NUM3,R0
	•
	•
	•
Add	NUMn, R0
Move	R0, SUM
	•
	•
	•
	•
	•
	•

A straight-line program for adding  $n$  numbers.  
 SNSCE / IT/ III Sem/V VaishnaveeAP-IT



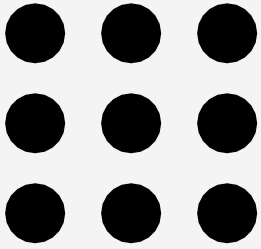
# Branching



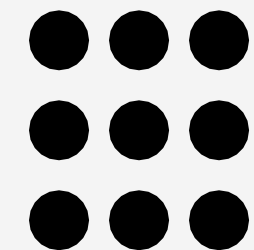
Using a loop to add *n* numbers.



# Condition Codes



- The processor keeps track of information about the results of various operations.
- This is accomplished by recording the required information in individual bits, called **Condition Code Flags**.
- Flags are grouped together in a special processor-register called the **condition code register (or status register)**.
  - 4 commonly used flags
    - ✓ N (negative)
    - ✓ Z (zero)
    - ✓ V (overflow)
    - ✓ C (carry)

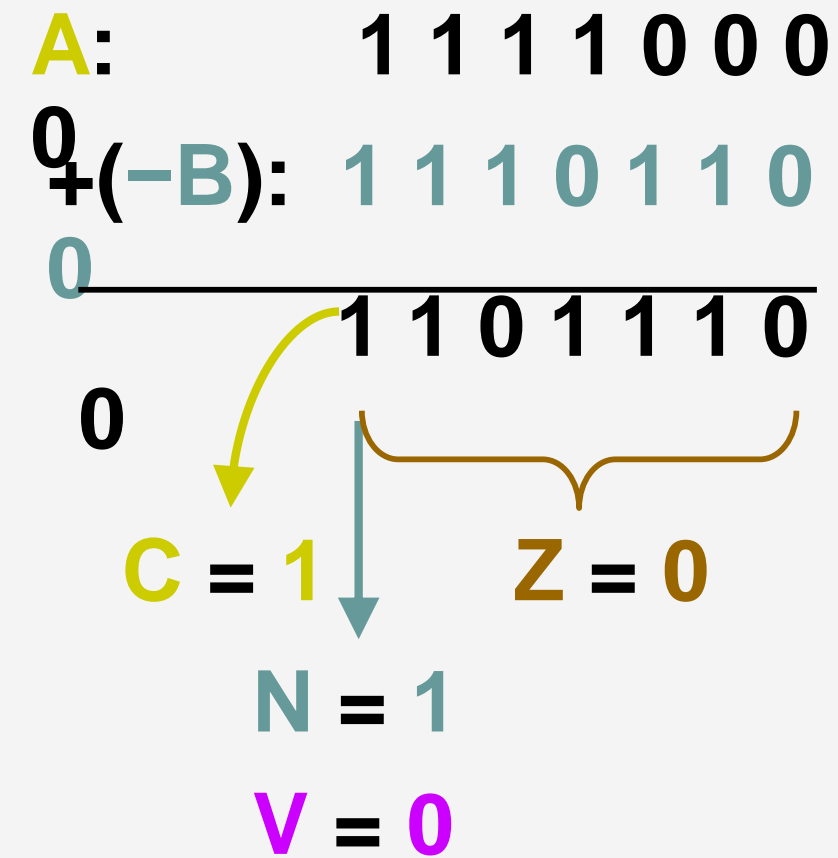


# Conditional Branch Instructions

- Example:

- A: 1 1 1 1 0 0 0 0

- B: 0 0 0 1 0 1 0 0





Thank You