# COMPUTER SCIENCE

## Computer Organization and Architecture

### Introduction of COA

Lecture_01



Vijay Agarwal sir


01 Introduction of COA

02 Instruction Cycle

COA :

↳ I^st time

12^th Pass PLZ Don't Mind.

+ve Mind set

Basic

PYQ's : 12L+ PYQ's

# GATE COA Syllabus

(9 to 11 Marks)
(10 marks)

- Introduction of COA
- Instruction format & Addressing mode
- ALU, Datapath & Control Unit
- Floating point Representation
- Cache Memory
- Pipelining
- Secondary Memory & I/O Interface

# Introduction of COA

- Introduction
- Components of Computer
- Types of Registers
- Instruction cycle
- Memory Concept
- Byte & word addressable
- System Bus

# Introduction format & Addressing mode

- Instruction concept
- Machine Instruction
- Instruction format
- Expand opcode technique
- Addressing modes concept
- Types of addressing modes
- ISA

# ALU, Datapath & Control unit

- Data path
- Micro Instruction
- Micro Program
- Control Unit Design

# Cache Memory

] 50 + Marks.

- Memory Concept
- Types of Memory Organization
- Cache Memory
  - Cache Organization
  - Mapping Technique
  - Replacement Algorithm
  - Updating Technique & Multilevel Cache

# Pipelining

40 + marks

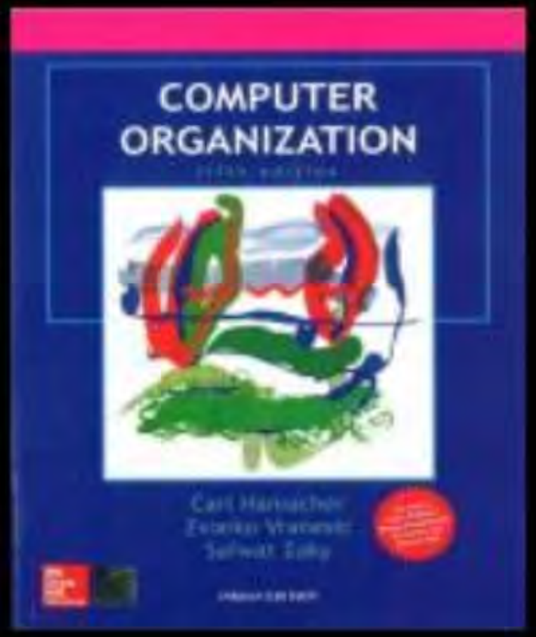
- Pipeline concept
- Pipeline Types
- Performance Evaluation
- Dependencies in pipeline
  - Structural Dependency
  - Data Dependency
  - Control Dependency
- Pipeline Hazards

# Secondary Memory & I/O Interface

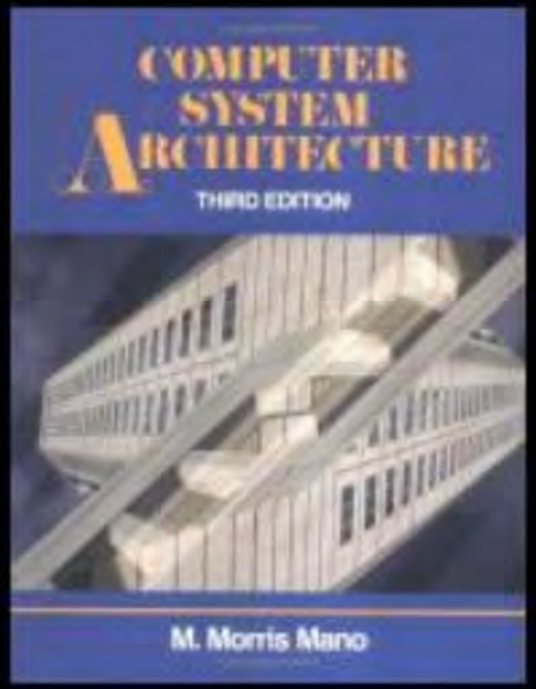
- [x] Disk Concept
- [x] Disk Structure
- [x] Disk Access time
- [x] I/O Interface & its type

# Books

- Carl Hamacher
- William Stallings
- Morris Mano
- JP Hayes







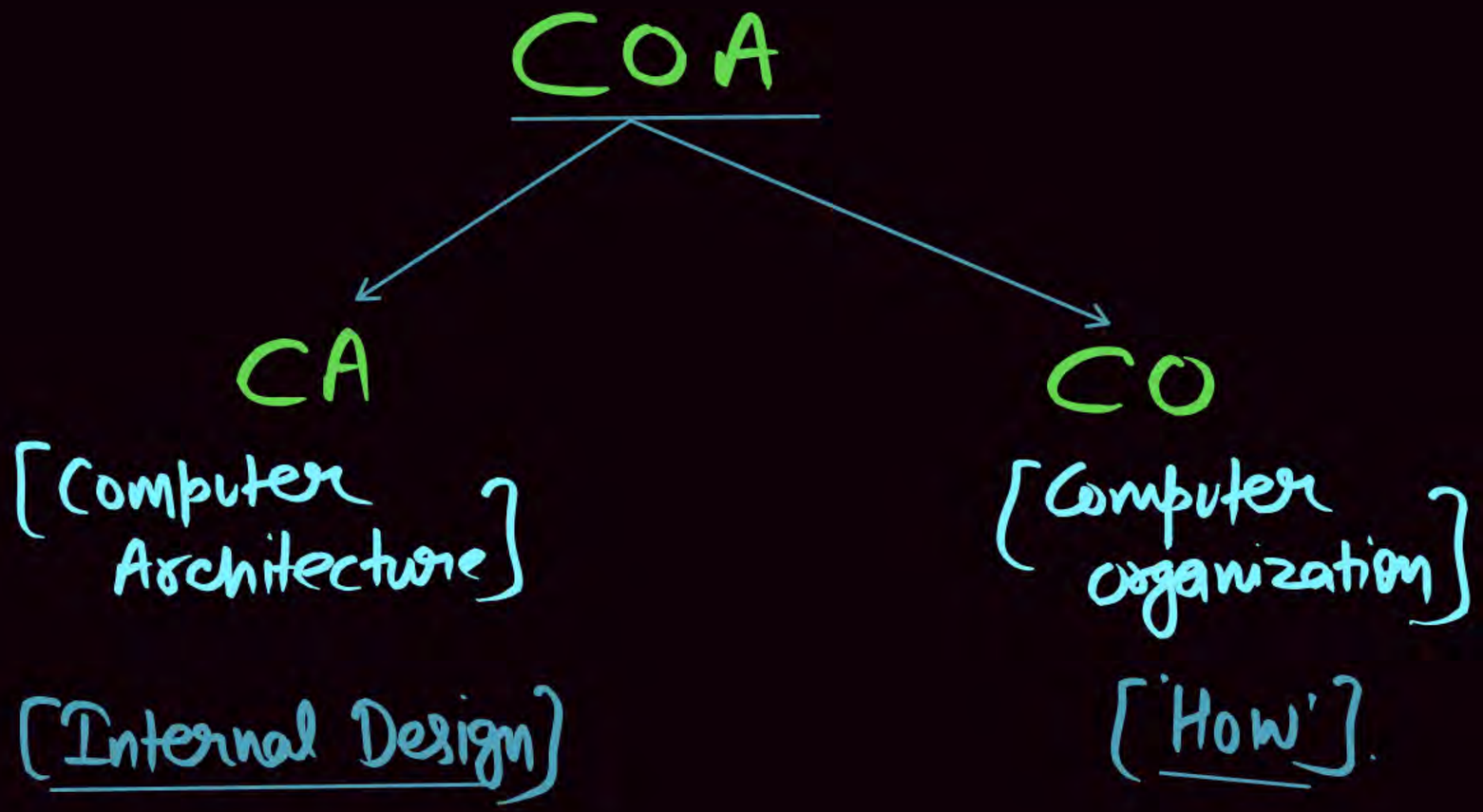

# Computer Generation :

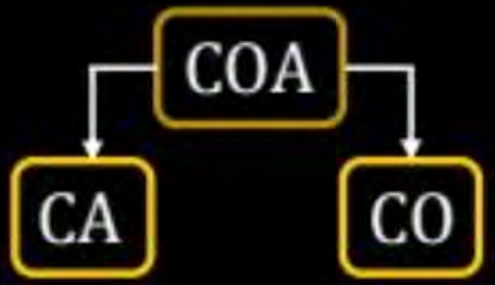
| | I^st | II^nd | III^rd | IV & V |
|---|---|---|---|---|
| | 1942 - 1955 | 1955 - 1964 | 1965 - 1974 | 1974 - Present |
| Components | Vaccumme Tube | Transistor | I.C [Integrated Chip] | VLSI & ULSI |
| Language | Machine Language | Assembly Language | H.L.L, OOP's RDBMS.. | OOP's, RDBMS, ML, AI.. |

Note

The Digit Computer was ENIAC in 1943.

Electronic Numerical and Integrator Computer.


COA
CA
[Computer Architecture]
[Internal Design]
CO
[Computer organization]
['How']




[Computer Architecture]
It deals with

- Instruction
- Addressing Mode
- ALU
- Pipeline
  (Internal Design)

[Computer Organization]
It deals with

How various memory & I/O (Input Output) interact with System (Memory Organization I/O organization)

# COA

## CA:

Computer Architecture is a Attributes which is visible to Programmer.

- Instruction
- Number of bit Required to Represent the Data
- Addressing Mode & etc.

## CO

↳ Deals with How the Feature will Implemented.

Note: Intel X86 has Same Architecture but Different organization

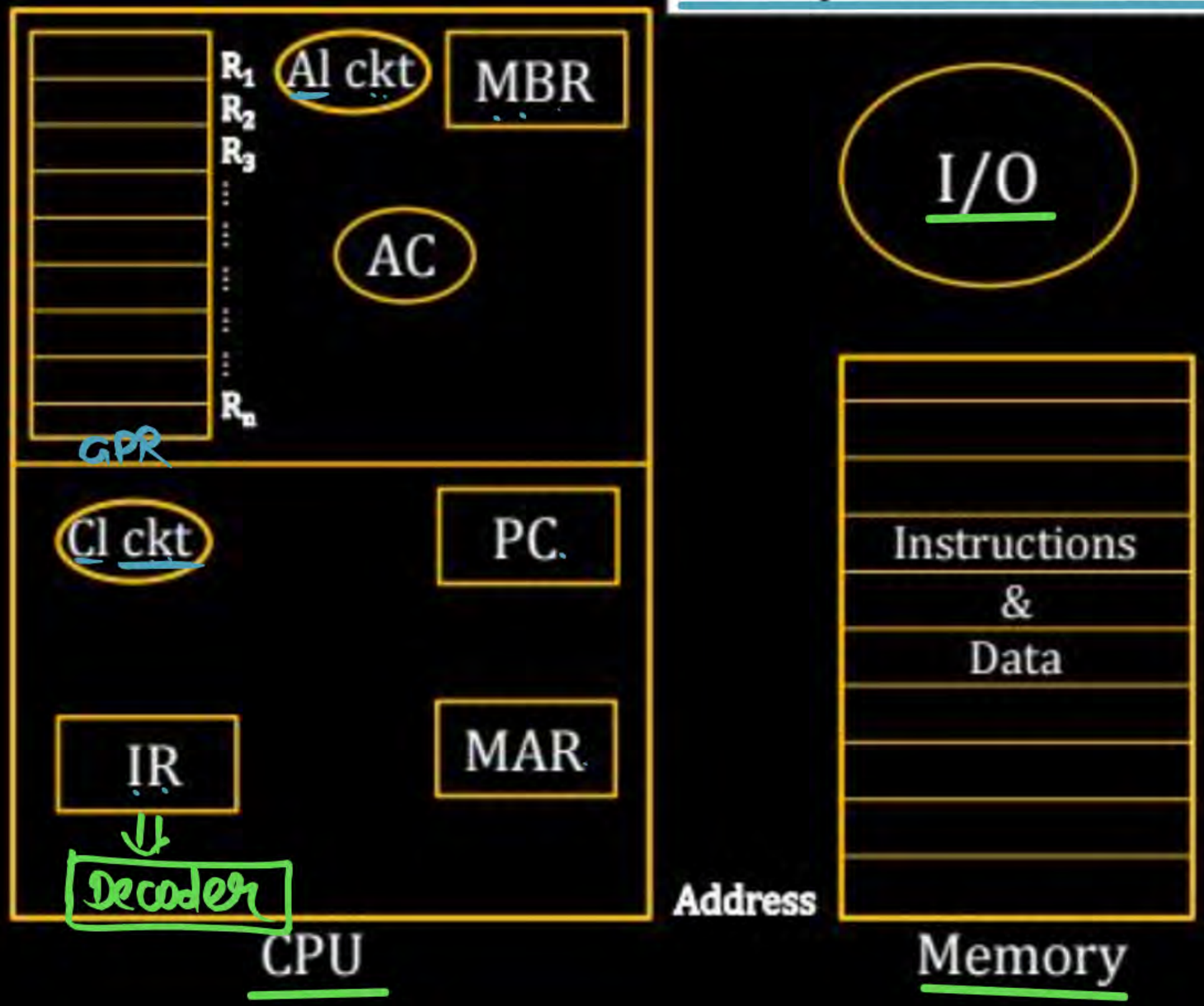
# Intel X86

80186
80286
80386
80486
80586

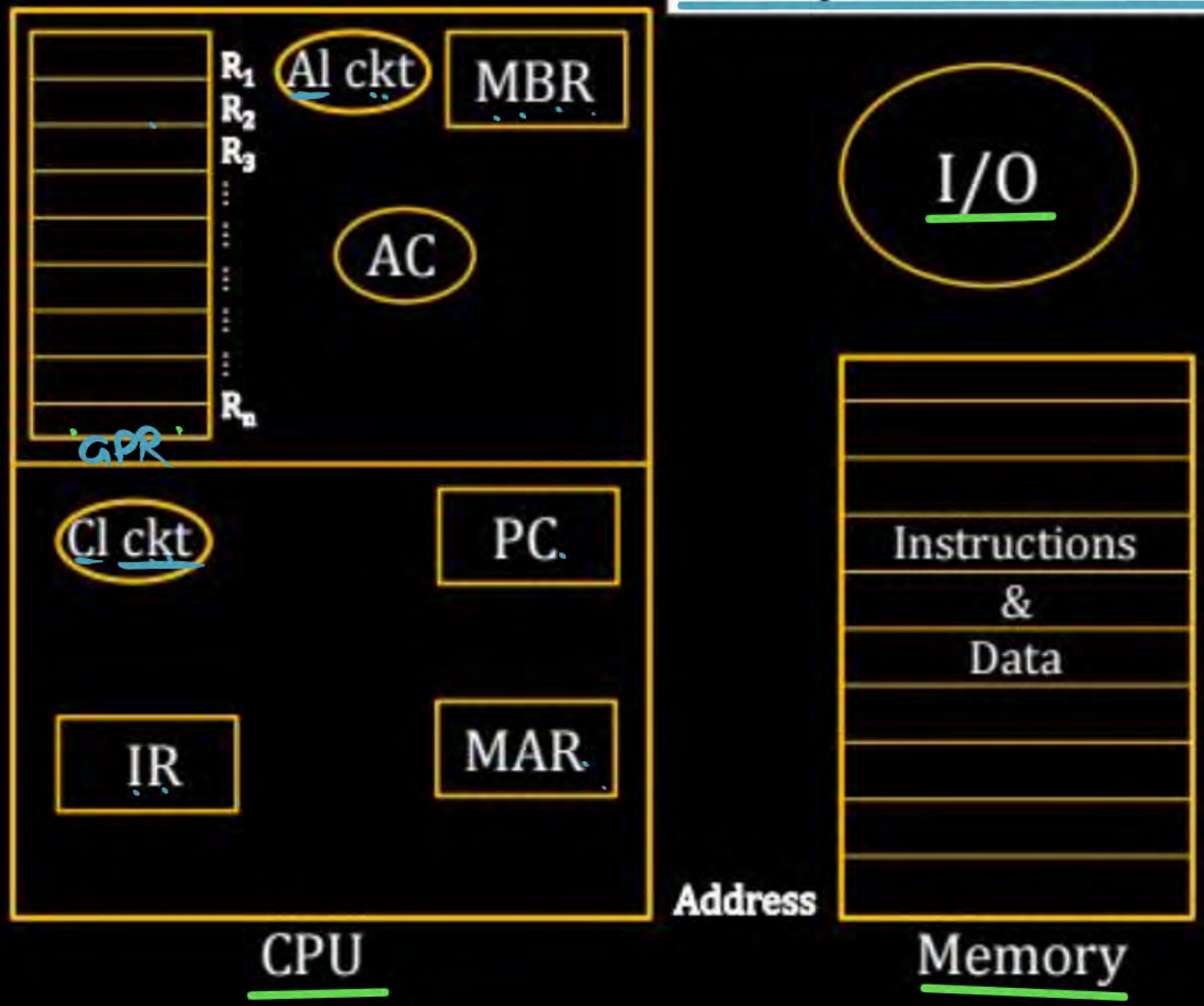
Have the Same Architecture
But Different organization (How features are Implement)

# Component of computer.

# Component of Computer



# Component of Computer



① Program Counter : [PC] : PC Contain the Starting address of the Next Instruction to be executed (Fetched)

② AR | MAR [Memory Address Register] : MAR Contain the Memory Address used for either Read (or) write operation.

'DATA'

③ DR | MDR | MBR [Memory Buffer Register] : Hold the Instruction or DATA.

④ IR [Instruction Register] : It Contain the Instruction which is Currently executed by the CPU.

WHY Instruction in IR ? Because Instruction Format is Predefined in IR.

⑤ AC [Accumulator] : Contain the temporary Result of ALU operation (OR) First operand (DATA) of ALU operation.

⑥ GPR [General Purpose Register]

⑦ SP [Stack Pointer]

⑧ PSW [Program Status Word].

WHY MAR ? : Connected to Address Line.

WHY MBR ? : Connected to Data line.

SYSTEM BUS

① Address Line.

② Data Line.

③ Control Line.

0 PC[Program Counter]: PC contain the Starting Address of the Next Instruction to be fetch (Execute).

MAR[Memory Address Register]: That contain the address of Memory location used for either Read or Write Operation.

MBR[Memory Buffer Register]:
Or
MDR[Memory Data Register]

That contain Instruction & Data

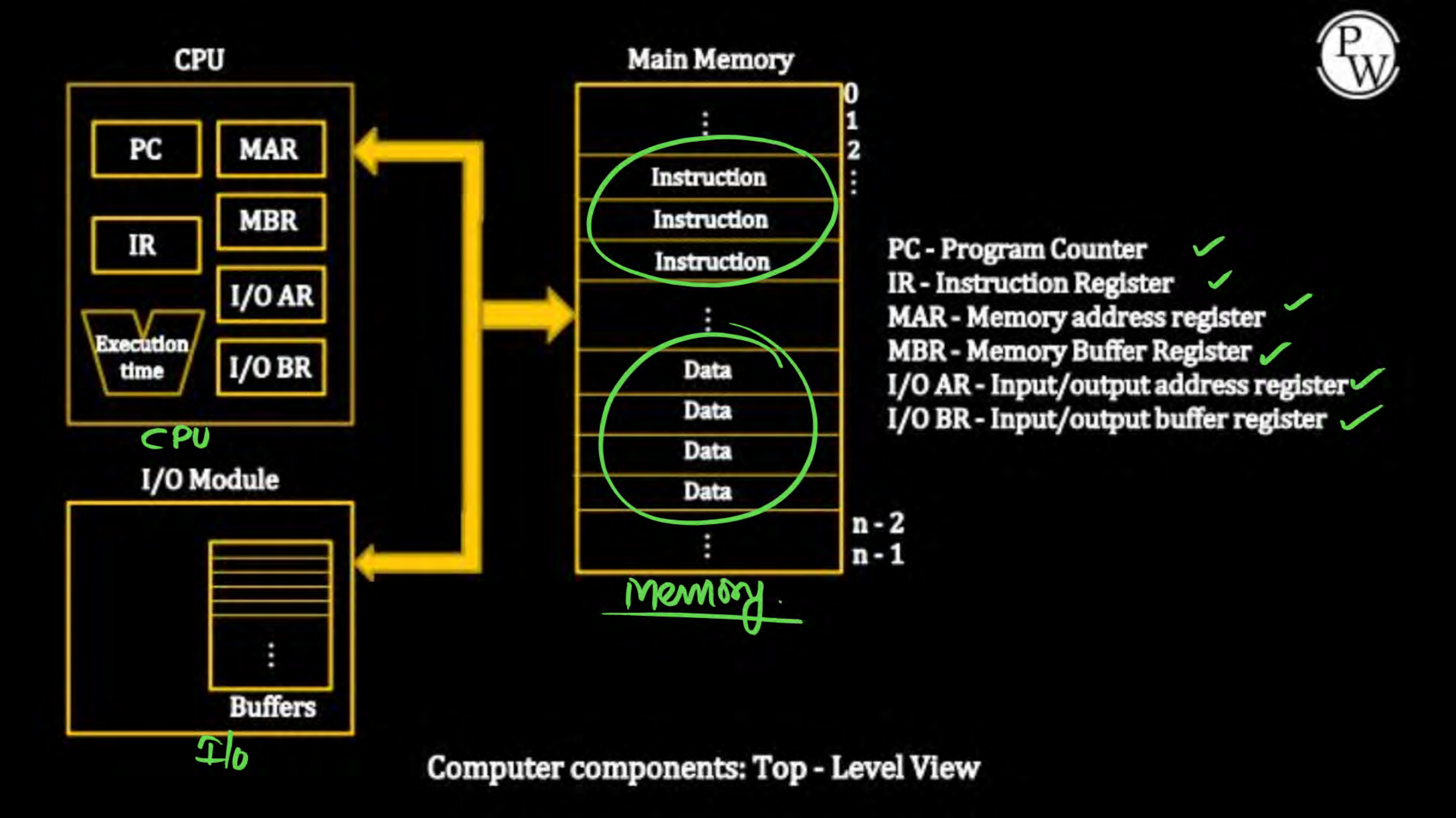
IR (Instruction Register): That contain the Instruction currently being Executed by the CPU.

AC[Accumulator]: That contain temporary Result (or) first Operand (DATA) of ALU operation
Example: ADD[4000] AC ← AC + M(4000)

GPR [General Purpose Register]: That is used for process the data.

PSW(Program Status Register)/ Flag Register: It store the status of the ALU Result

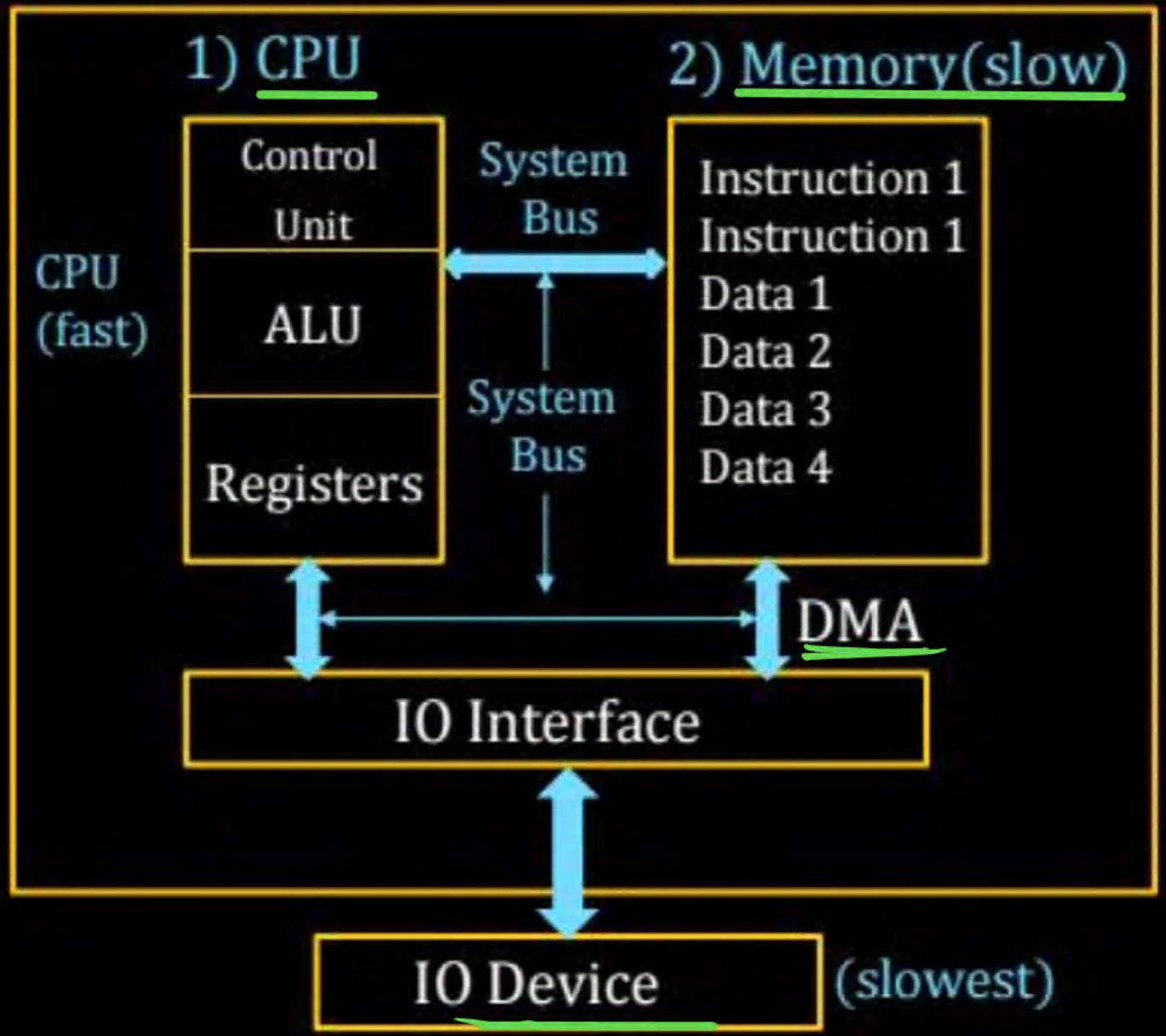
Stack Pointer (SP Register): Contain the TOS (Top of the Stack) address.




Computer components: Top - Level View

# Component of Computer

(1) CPU
(2) Memory
(3) Input/output



# Instruction Cycle

The Process Required for each Instruction Execution.

(OR)

Instruction cycle describe the execution sequence of the Instruction.

# Instruction Cycle

Instruction Cycle Contain 2 Sub cycle.

1. Fetch Cycle.
2. Execute Cycle
   - Decode
   - Execute.

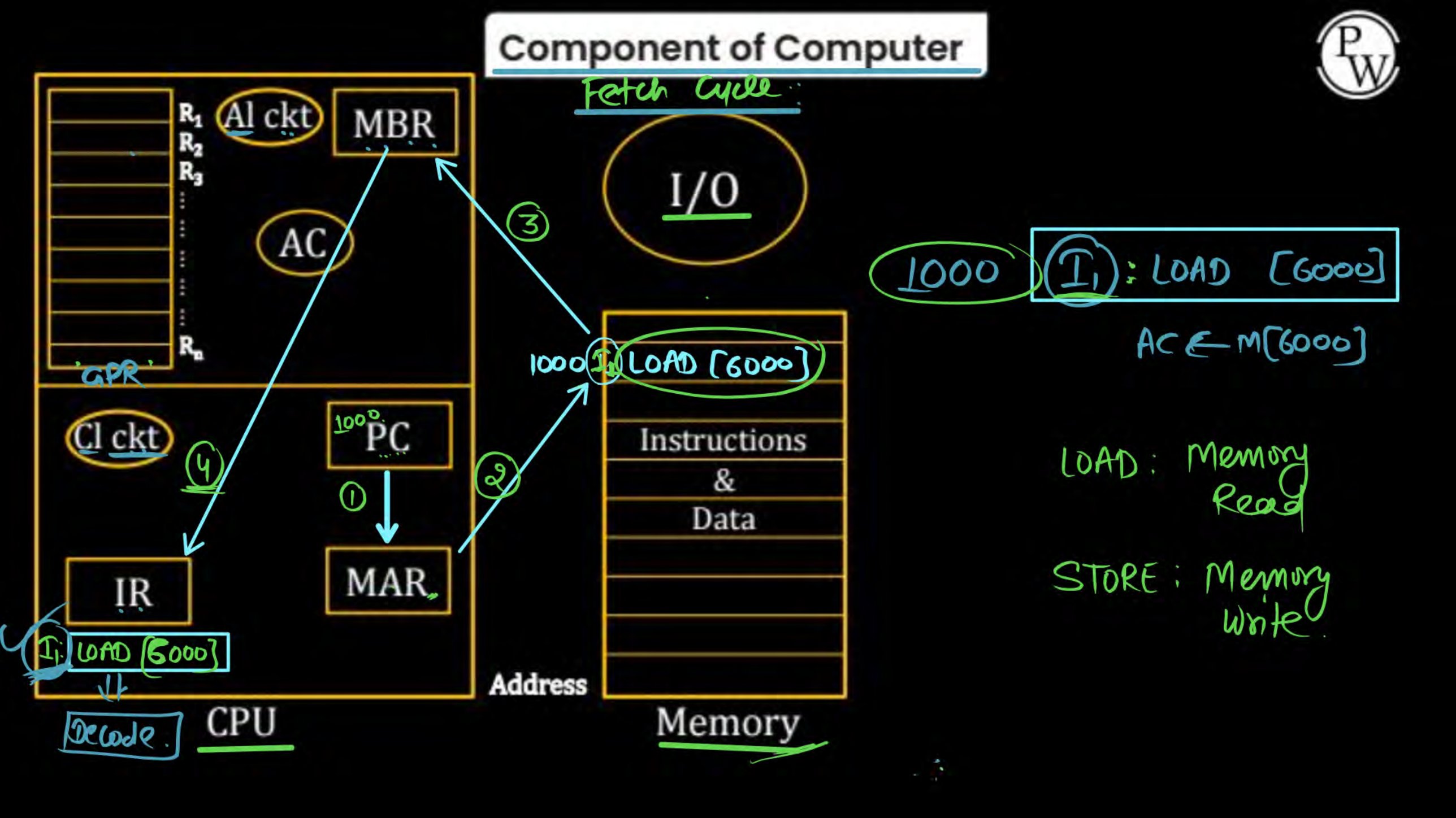
# Instruction Cycle

MEM to CPU (IR)

① Fetch cycle : To Fetch [bring] the Instruction From Memory to CPU. [MEM to CPU [IR]]

& At the end of Fetch cycle PC [Program Counter] is incremented, Now PC will denote Next Instruction Starting Address.

# Component of Computer

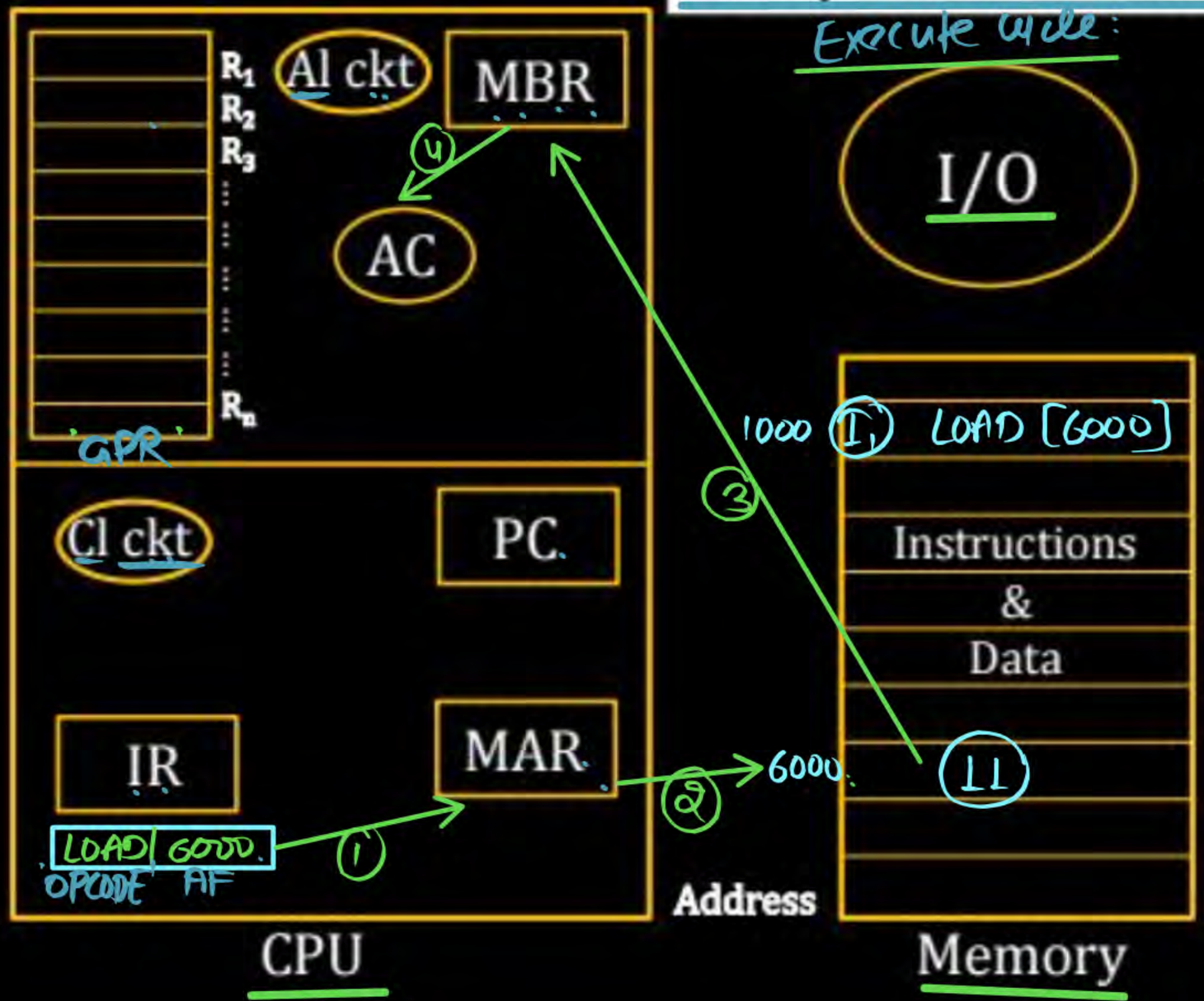


# Instruction Cycle

② Execute Cycle : To Process (to execute) the Fetched Instruction.

- Decode [Analysis of the Instruction]
  WHAT OPCODE, How many operand, Where operand are available.
- Execute. (Processing & Result Storage)

# Component of Computer



# Instruction Cycle

1. Fetch cycle (mem to CPU (IR))
2. Execute cycle.

# Instruction Cycle

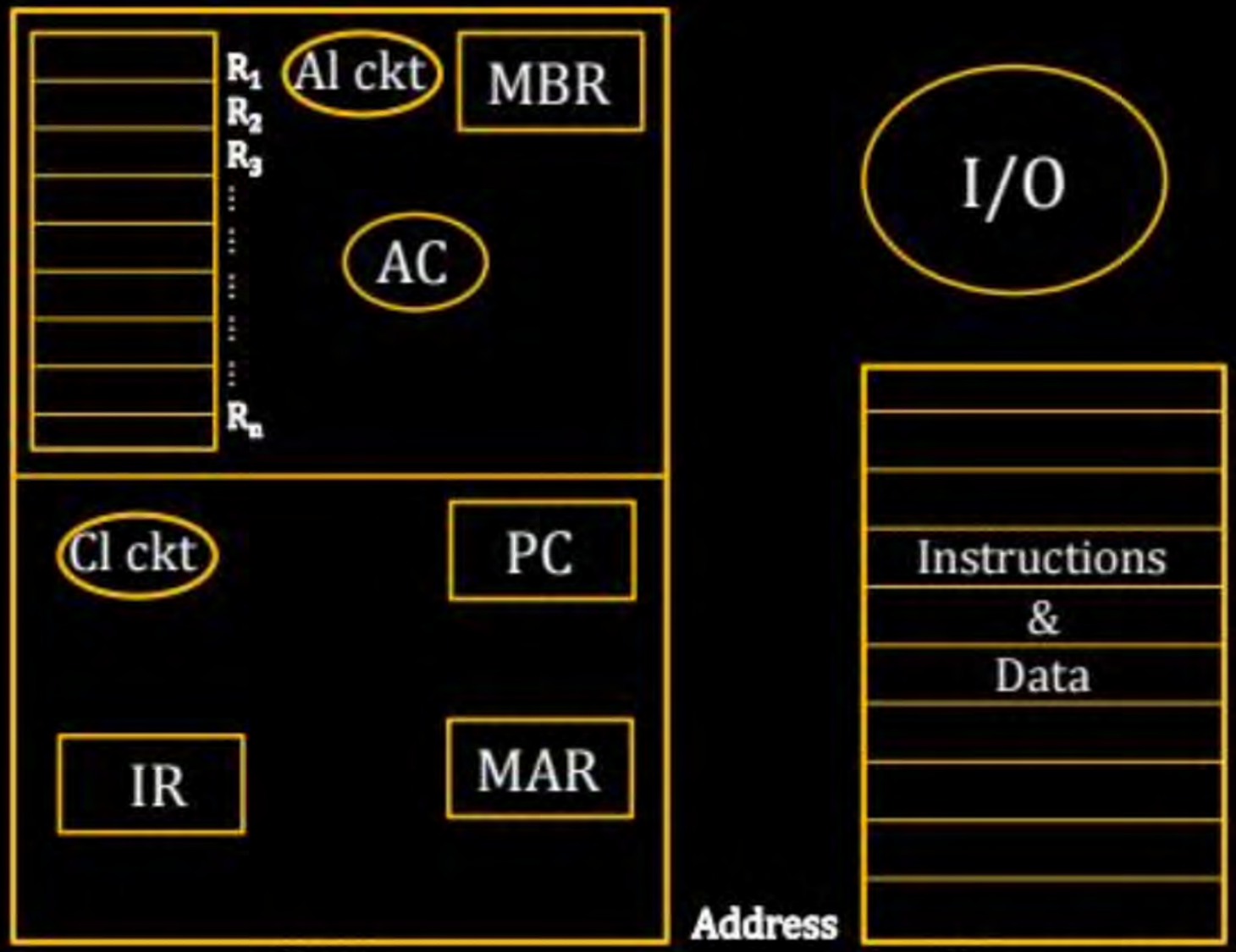
The process required to execute the Instruction.

(or)

Instruction cycle describe the execution sequence of the instruction.

Instruction Cycle contain 2 sub cycle.

1) Fetch cycle
2) Execute cycle
   - Decode
   - Execute


R1
R2
R3
Rn
Al ckt
MBR
AC
Cl ckt
PC
IR
MAR
CPU


I/O


Instructions
&
Data
Memory

Address

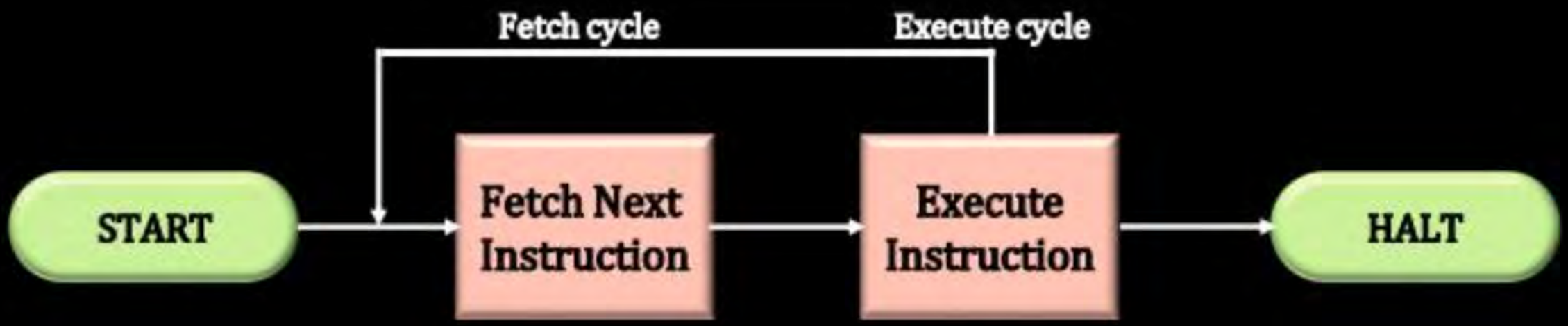
# Steps in Instruction Cycle

- **Fetch cycle**
  1) IAC (Instruction Address Calculation)
  2) IF (Instruction Fetch)

  MEM to CPU (IR).
- **Execute cycle**
  - **Decode**
    3) Decoding (Analysis of Instruction
    (What opcode, how many operand,
    where operands are available) )
    4) OAC (Operand Address Calculation)
    5) OF(Operand Fetch)
  - **Execute**
    6) DP (Data Processing)
    7) Result Storage

# Fetch Cycle

- At the beginning of each instruction cycle the processor fetches an instruction form memory
- The program counter (PC) holds the address of the instruction to be fetched next
- The processor increments the PC after each instruction fetch so that it will fetch the next instruction in sequence
- The fetched instruction is loaded into the instruction register (IR)
- The processor interprets the instruction and performs the required action.




BASIC INSTRUCTION CYCLE

# THANK YOU!