COMPUTER SCIENCE

Computer Organization and Architecture

Introduction of COA

Lecture_01

Vijay Agarwal sir

01 Introduction of COA

02 Instruction Cycle

GATE COA Syllabus

(9 - 11 marks)

10 Marks

- Introduction of COA
- Instruction format & Addressing mode
- ALU, Datapath & Control Unit
- Floating point Representation
- Cache Memory [2 - 4 M]
- Pipelining [2 - 3 marks]
- 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

V. Imp.

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

50+ Marks PYQ's.

Pipelining

→ v. Imp.

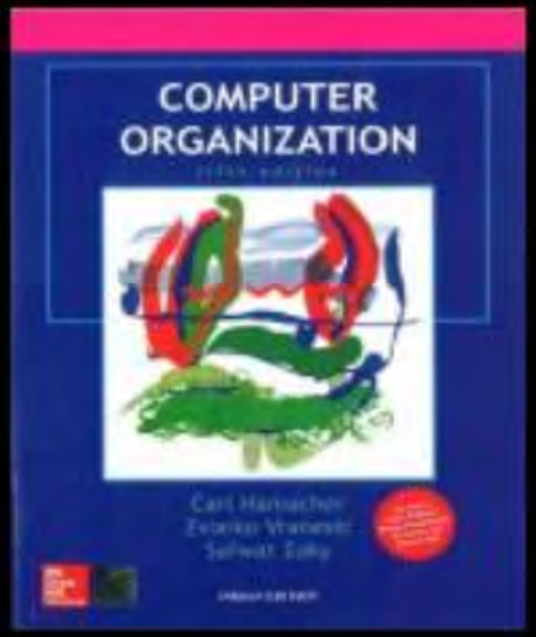
- Pipeline concept
- Pipeline Types
- Performance Evaluation
- Dependencies in pipeline
  - Structural Dependency
  - Data Dependency
  - Control Dependency
- Pipeline Hazards

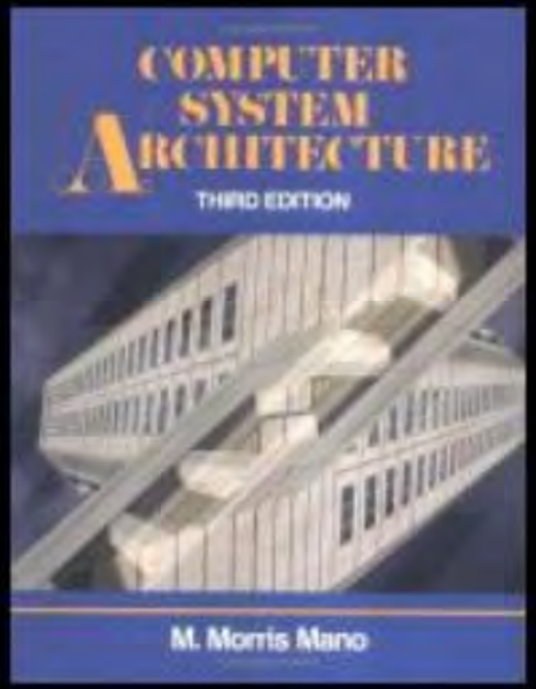
Secondary Memory & I/O Interface

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

Books

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





111+ PYQ's] Solved in the class.

for General Pull PSU Question]

Only 12th Pass

Plz Don't Mind

@ cc with Enjoying [Chamka] Concept

(b) Cc [Crystal clear]

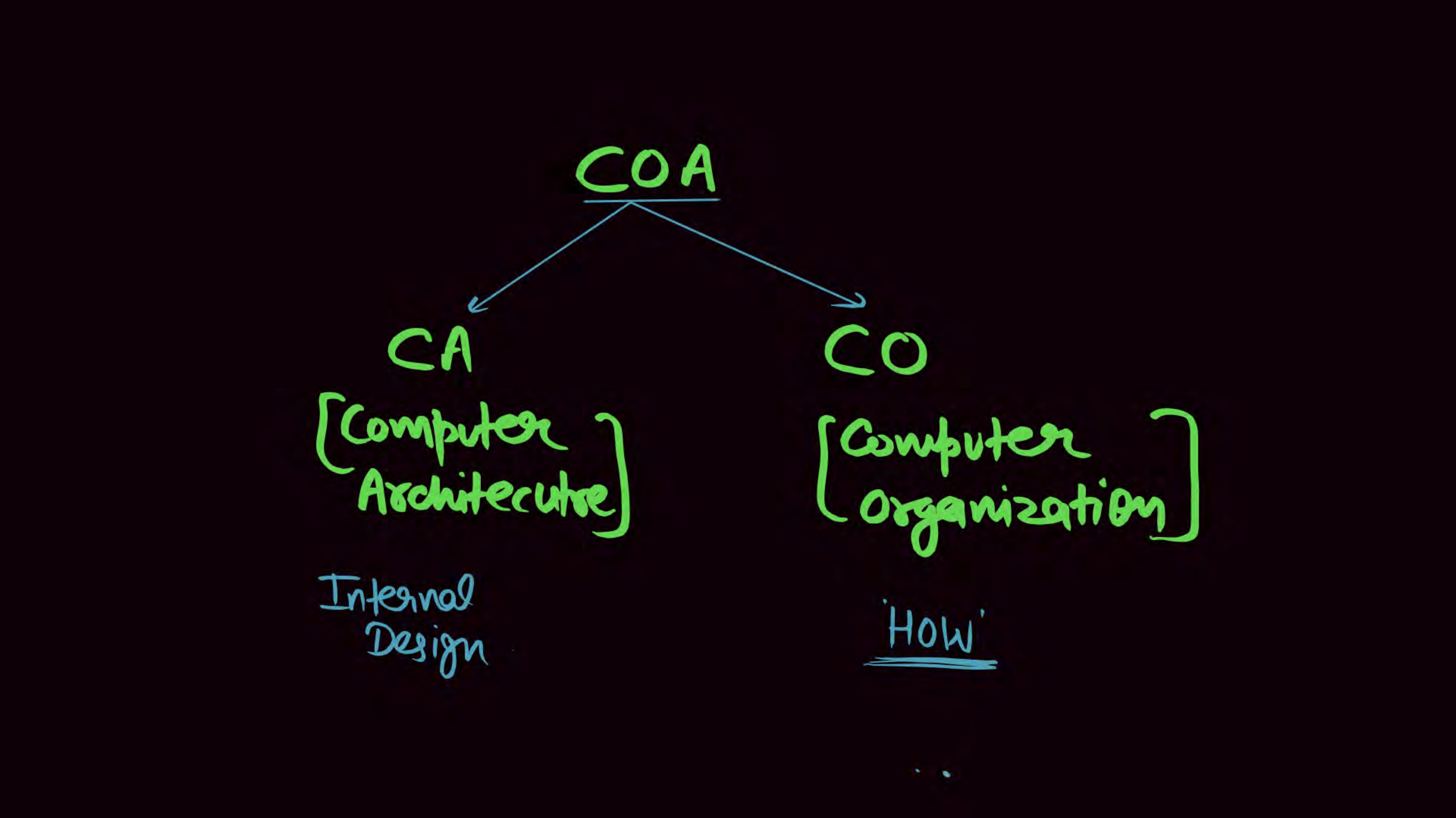
(c) C [clear]

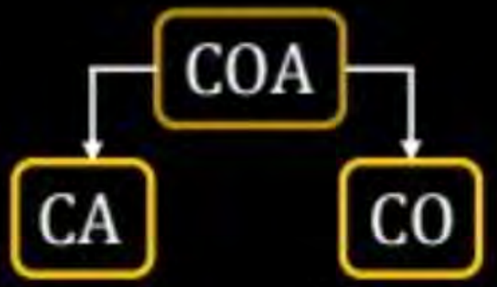
(d) Doubt

Computer Generation

	I st	II nd	III rd	IV & V th Gen.
	1942 - 1955	1955 - 1964	1965 - 1974	1974 - Present
Component	Vaccume tube	Transistor	IC [Integrated chip]	VLSI & ULSI (ultra) [very large Scale Integration]
Language	machine m/c Language	Assembly Lang	H.L.L & OOP's	OOP's, RDBMS & AI & ML

Note: The First Digital Computer was ENIAC in 1943.
Electronic Numerical AND Integrator Computer.


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

Computer Architecture

is a those Attribute which is visible to the Programmer.

- eg → Instruction Format
- eg → Number of bit Required to Represent the Data
- eg → Addressing Mode

Computer organization

Deals 'How' the Features are implemented.

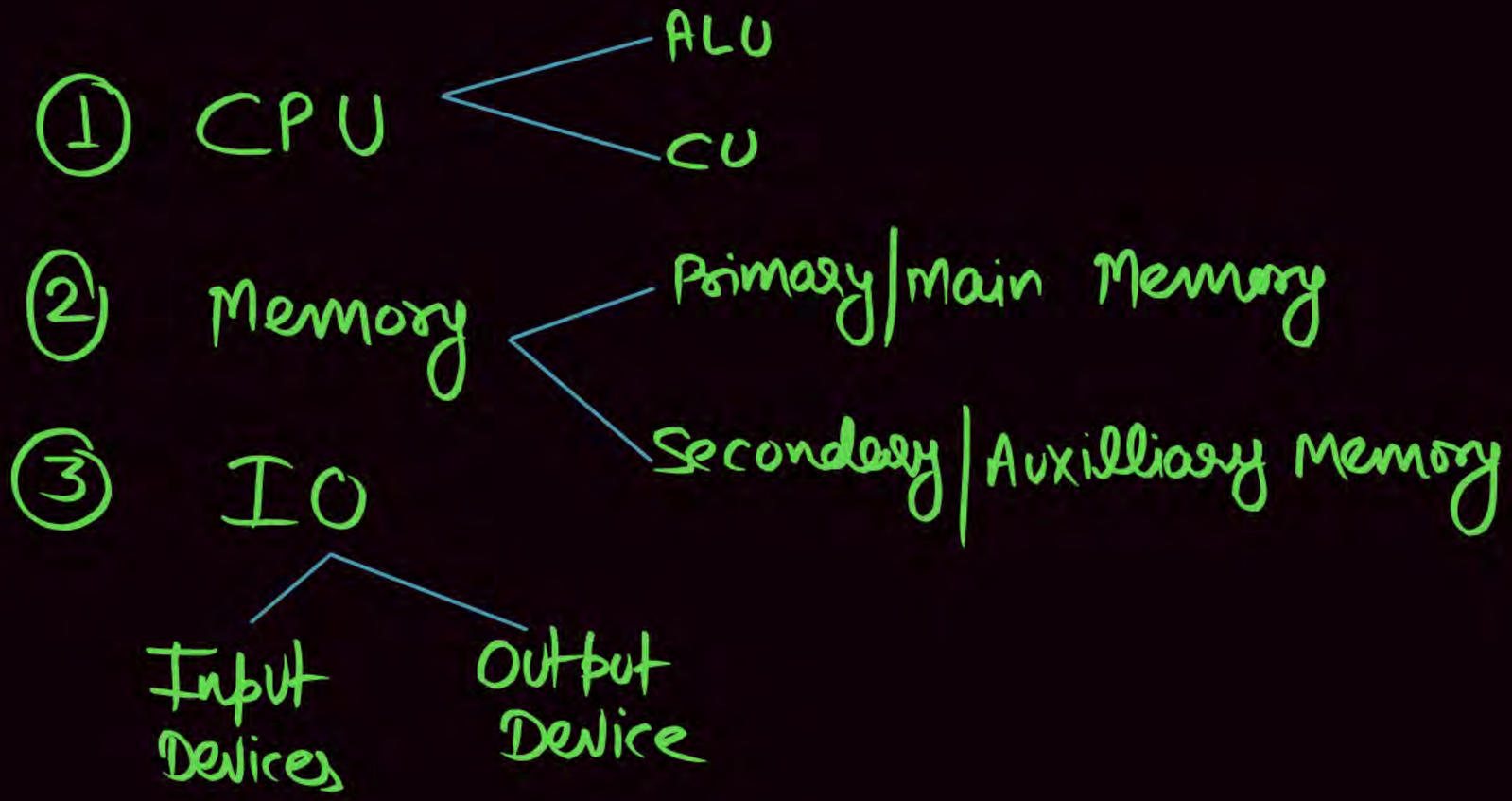
Note: Intel X86 share the Same Architecute but organization is Different

Intel X86

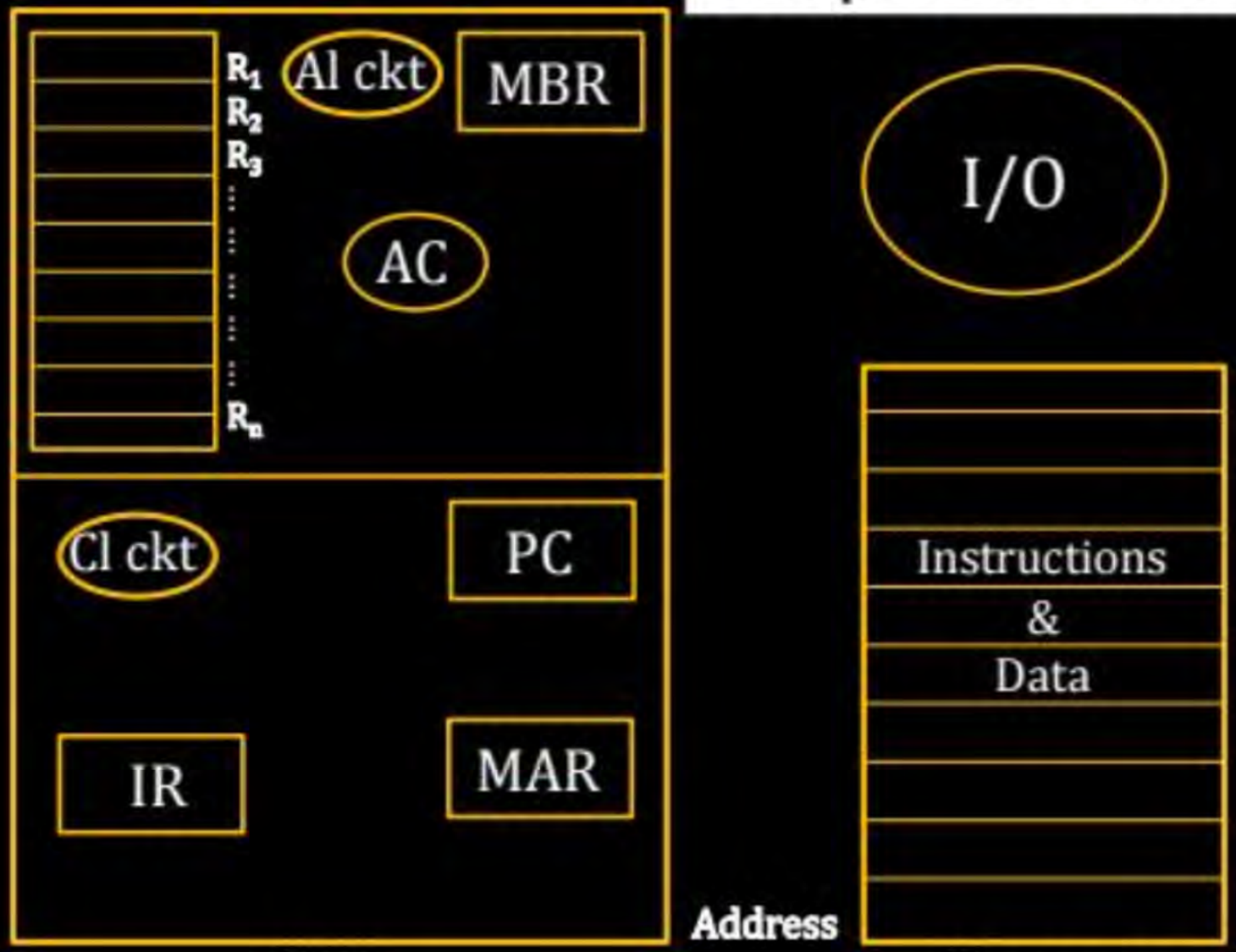
- 80186
- 80286
- 80386
- 80486
- 80586

Same Architecture
But organization is Different. (How features are Implement that is Different)

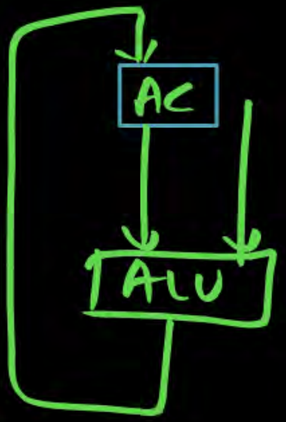
Component of the computer.



Component of Computer



I/O



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

②
AR/MAR [Memory Address Register] : Hold the Memory Address used for Read/write operation.

③ DATA
DR/MDR/MBR [Memory Buffer Register] : Hold the Instruction (OR) Data.

∴

④ IR (Instruction Register) : that Contain the Instruction which is currently executed by the CPU.

WHY IR ? Because Instruction Format is Predefined in IR.

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

⑥ GPR [General Purpose Register]

⑦ SP [Stack Pointer]

⑧ PSW [Program Status Word]

WHY MAR ? → Connected to 'AL' of the BUS

WHY MBR ? → Connected to 'DL' of the BUS.

Will Discuss

System Address

- Address Line (AL)
- Data Line (DL)
- Control Line (CL)

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

ADD R_1

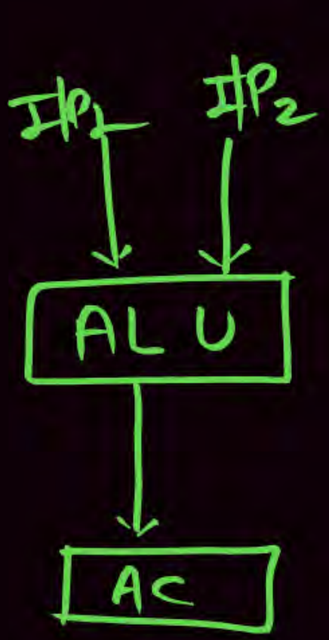
$AC \leftarrow AC + R_1$

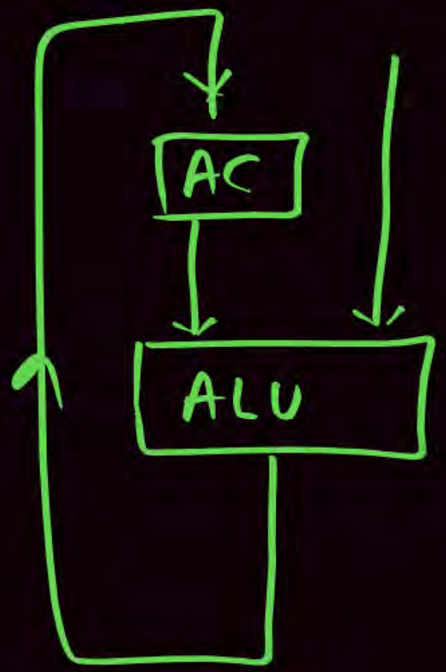
Example: ADD[4000] AC ← AC + M[4000]

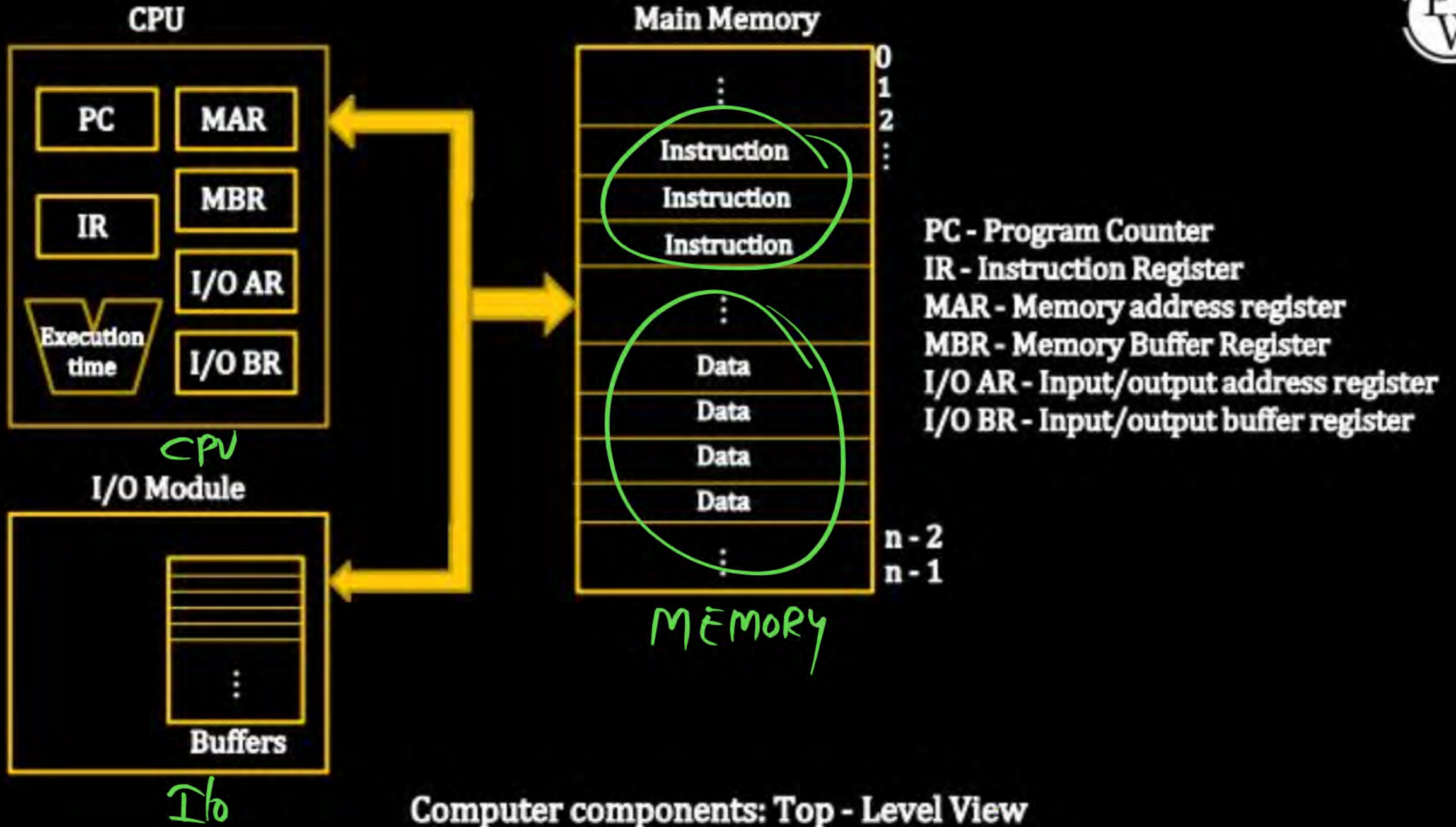
GPR [General Purpose Register]: That is used for process the data.

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.


IP1
IP2
ALU
AC


AC
ALU



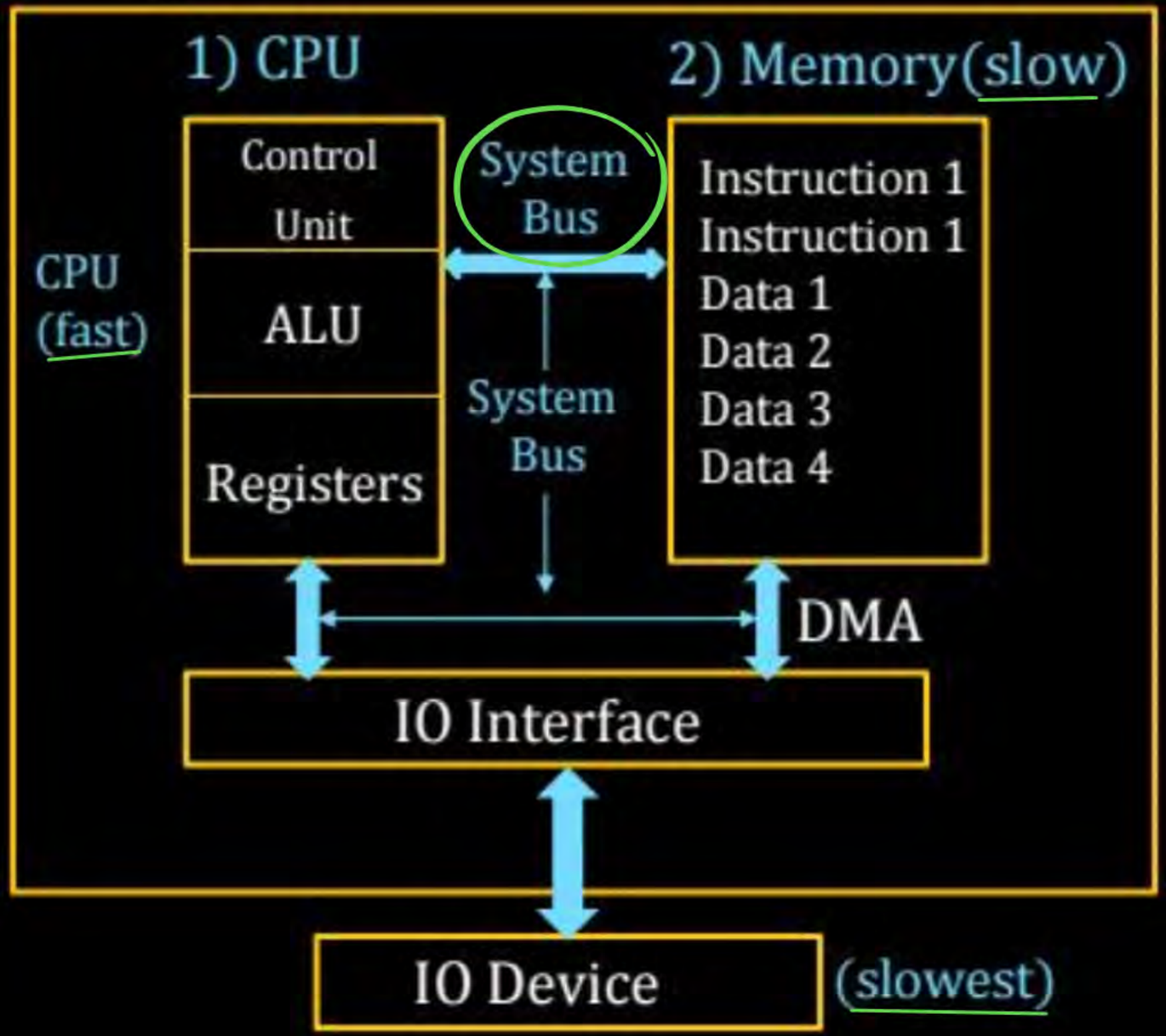
Computer components: Top - Level View

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 Contain 2 Sub cycle.

1. Fetch cycle
2. Execute cycle
   - Decode
   - Execute.

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

Instruction Cycle

① Fetch Cycle : To Fetch [bring] the Instruction from main Memory to the CPU.

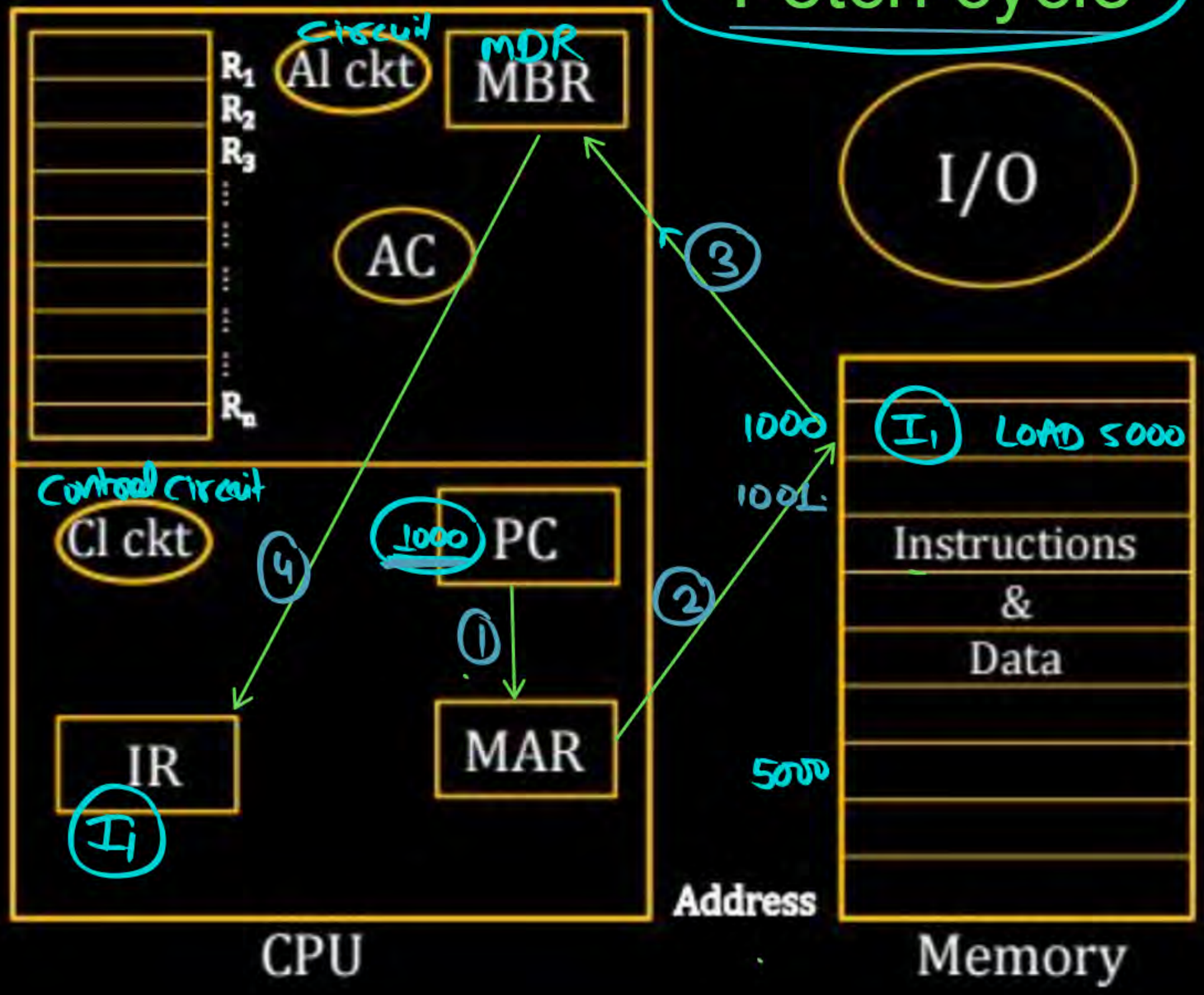
⇓

Mem to CPU (IR)

& At the end of Fetch cycle PC [Program Counter] is incremented.

Fetch cycle

MEM to CPU(IR)



eg

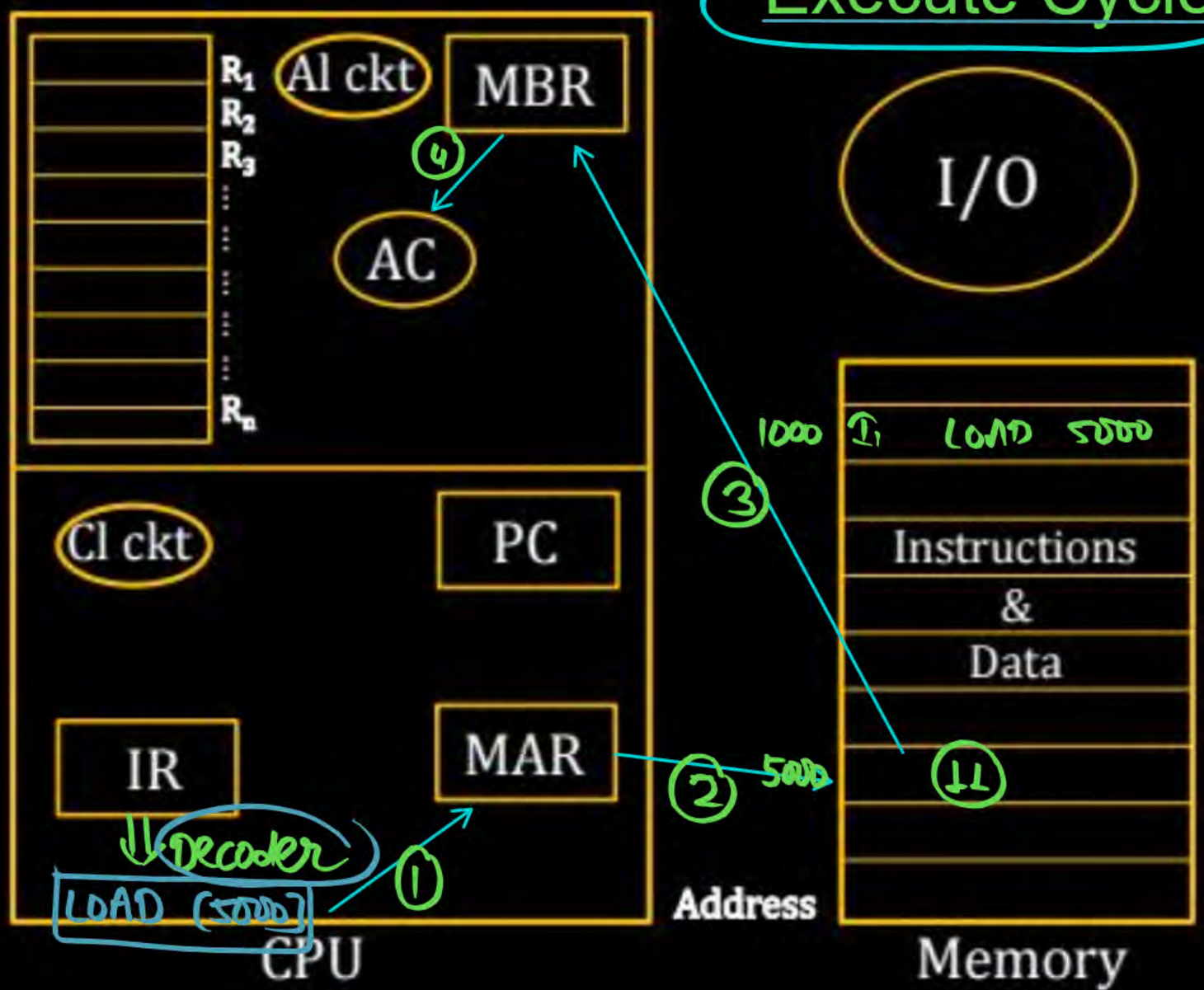
1000 (I1) : LOAD [5000]

AC ← M[5000]

LOAD : Memory Read

STORE : memory Write

Execute Cycle

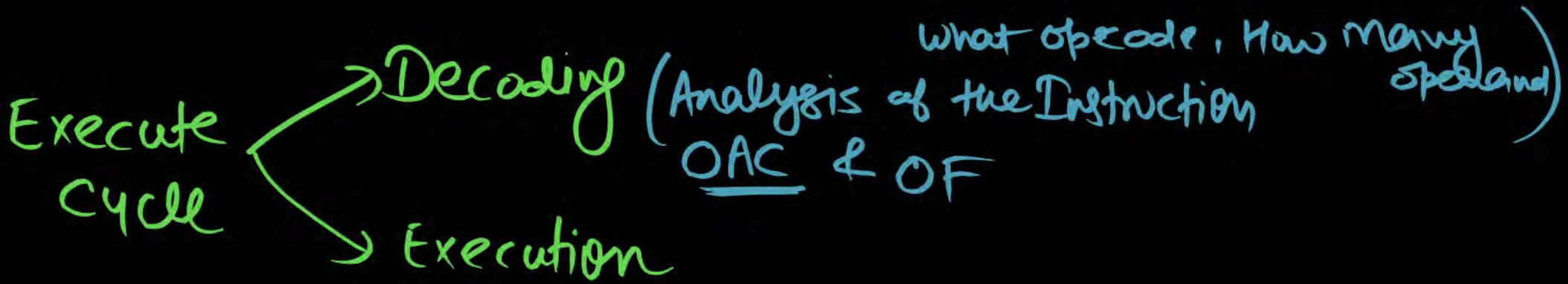


1000 (I1) LOAD [5000]

AC ← M[5000]

Instruction Cycle

② Execute cycle : The objective of the execute cycle is to Execute (to Process) the Fetched Instruction.



Steps in Instruction Cycle

1. Fetch cycle
   1) IAC (Instruction Address Calculation)
   2) IF (Instruction Fetch)
2. Execute cycle
   - Decode
     3) Decoding (Analysis of Instruction
     (What opcode, how many operand,
     where operands are available))
     4) OAC (Operand Address Calculation) (DATA)
     5) OF(Operand Fetch)
   - Execute
     6) DP (Data Processing)
     7) Result Storage

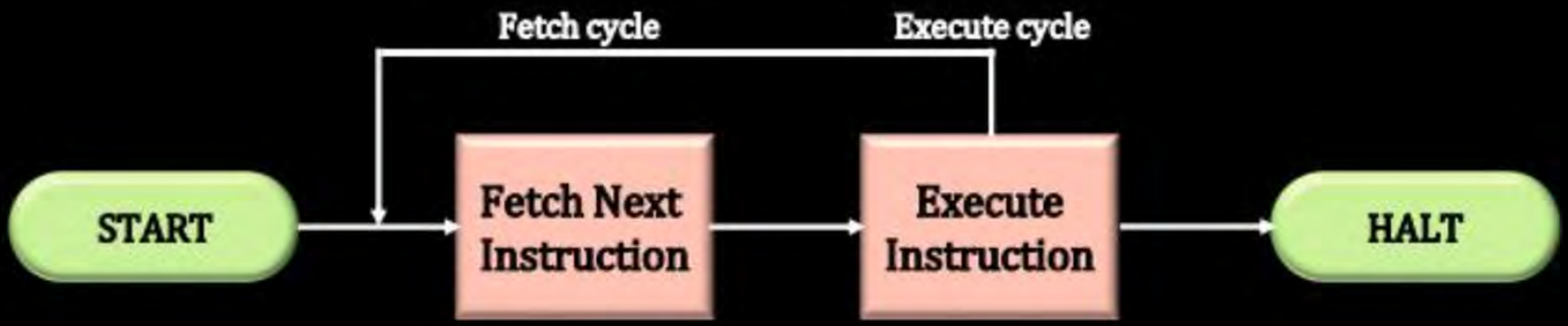
Type of operation
↑
OPCODE ⇒ operational code.
↑
LOAD [5000]
Operand Address

AC ← M[5000]

11 → OPERAND (DATA)

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.



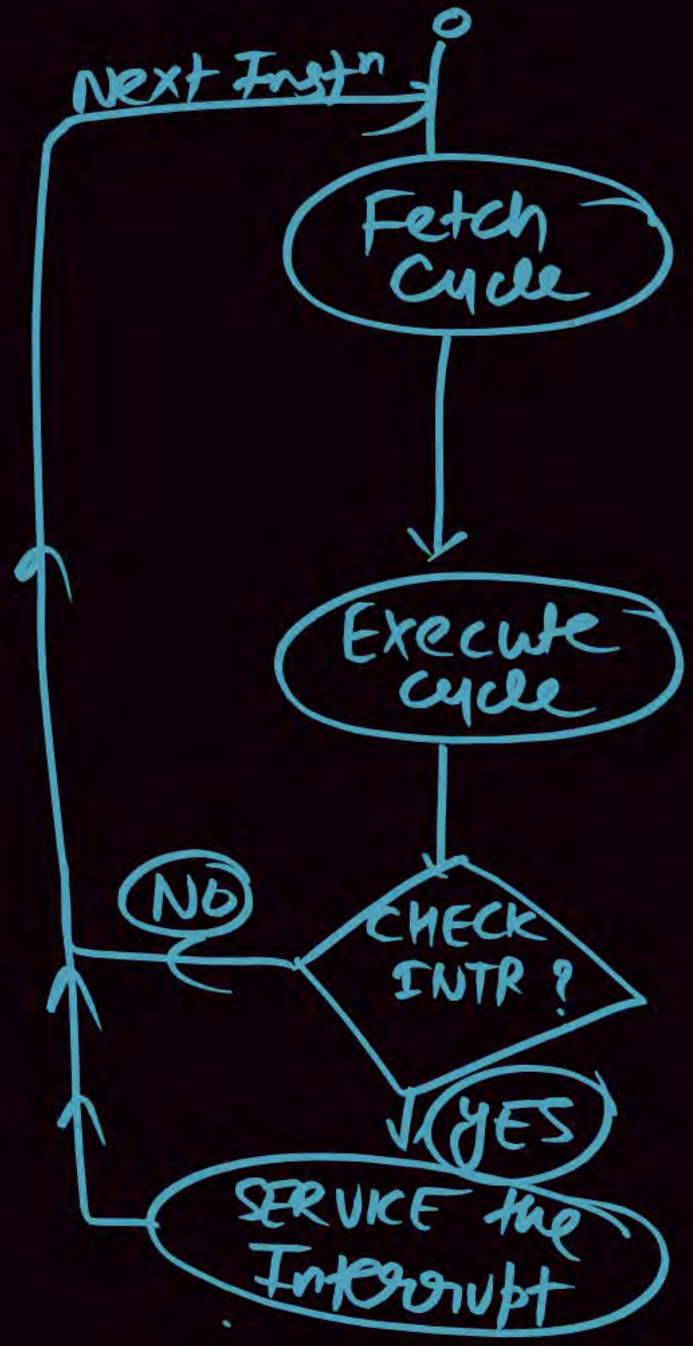
BASIC INSTRUCTION CYCLE

Instruction cycle

1. Fetch cycle
2. Execute cycle
3. Interrupt cycle

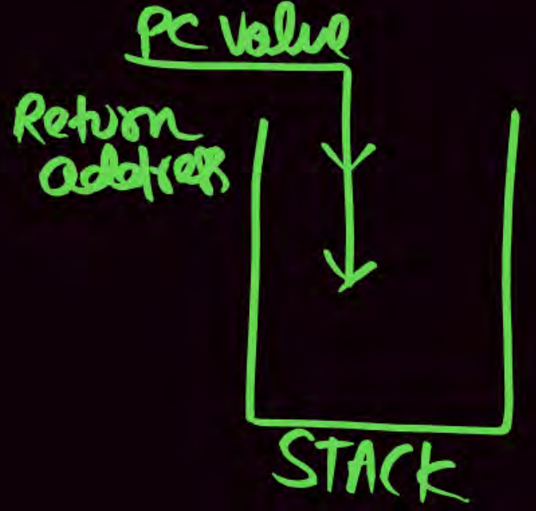
↓

Unusual Event



Interrupt cycle : When CPU encounter the Interrupt then After finishing the current Instruction Execution, Interrupt will be serviced.

When CPU encounter the Interrupt then it push the PC Value in to the Stack as a Return address & Control transferred to ISR.

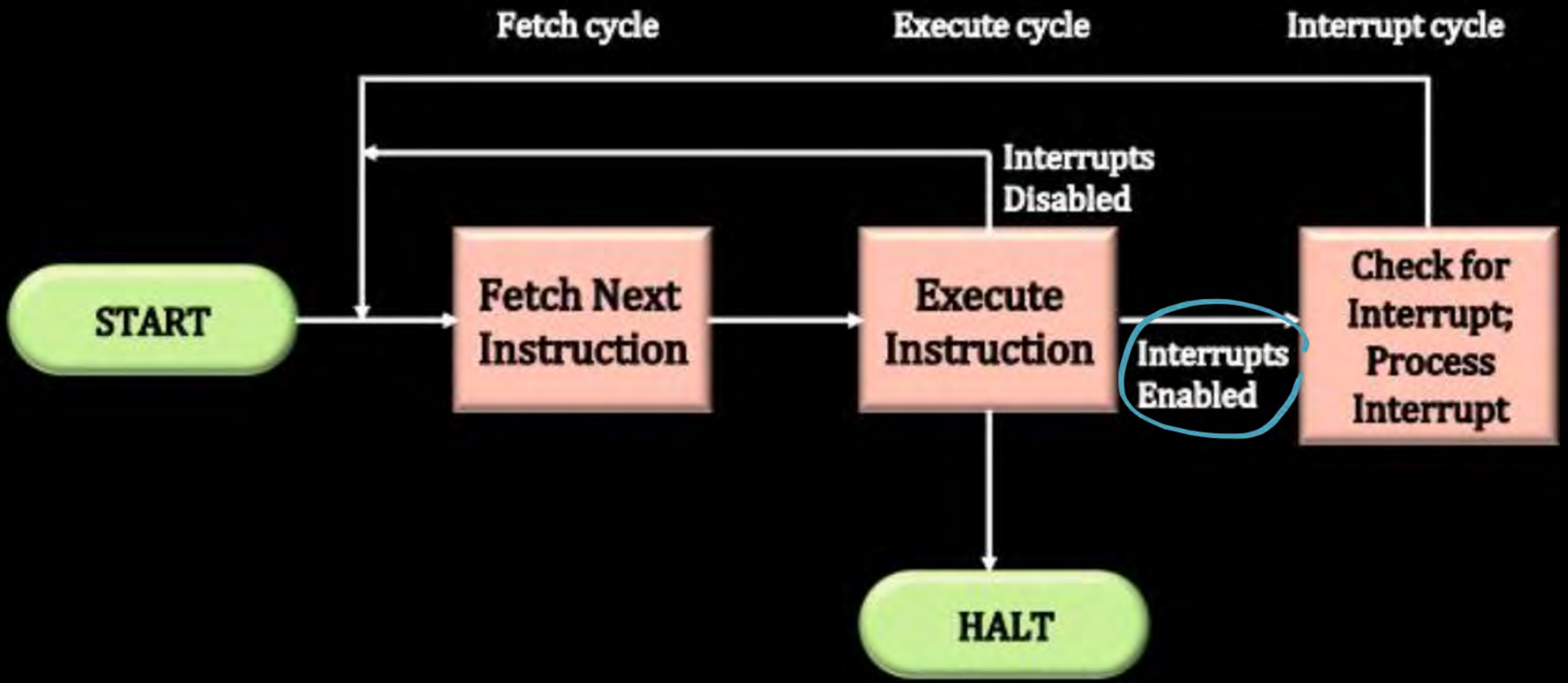


& Control transfer to ISR.

WHY PC Value Stored in STACK.

STACK Work On
LIFO

If we use Queue.



Instruction cycle with Interrupts

THANK YOU!