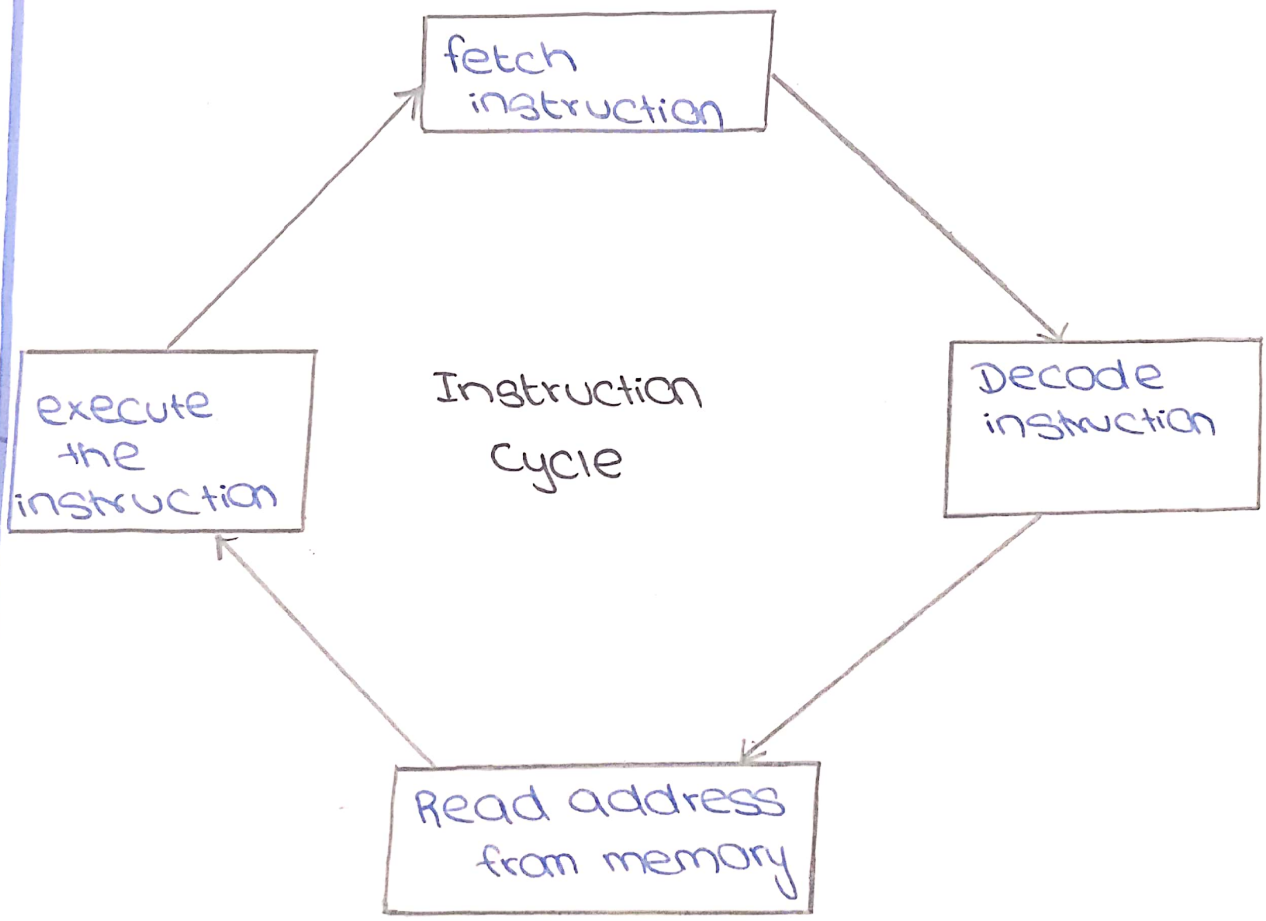
COMPUTER ORGANIZATION AND ARCHITECTURE

FATIMA KHAN CT-23024

ASSIGNMENT ONE

Q1 Draw instruction execution cycle

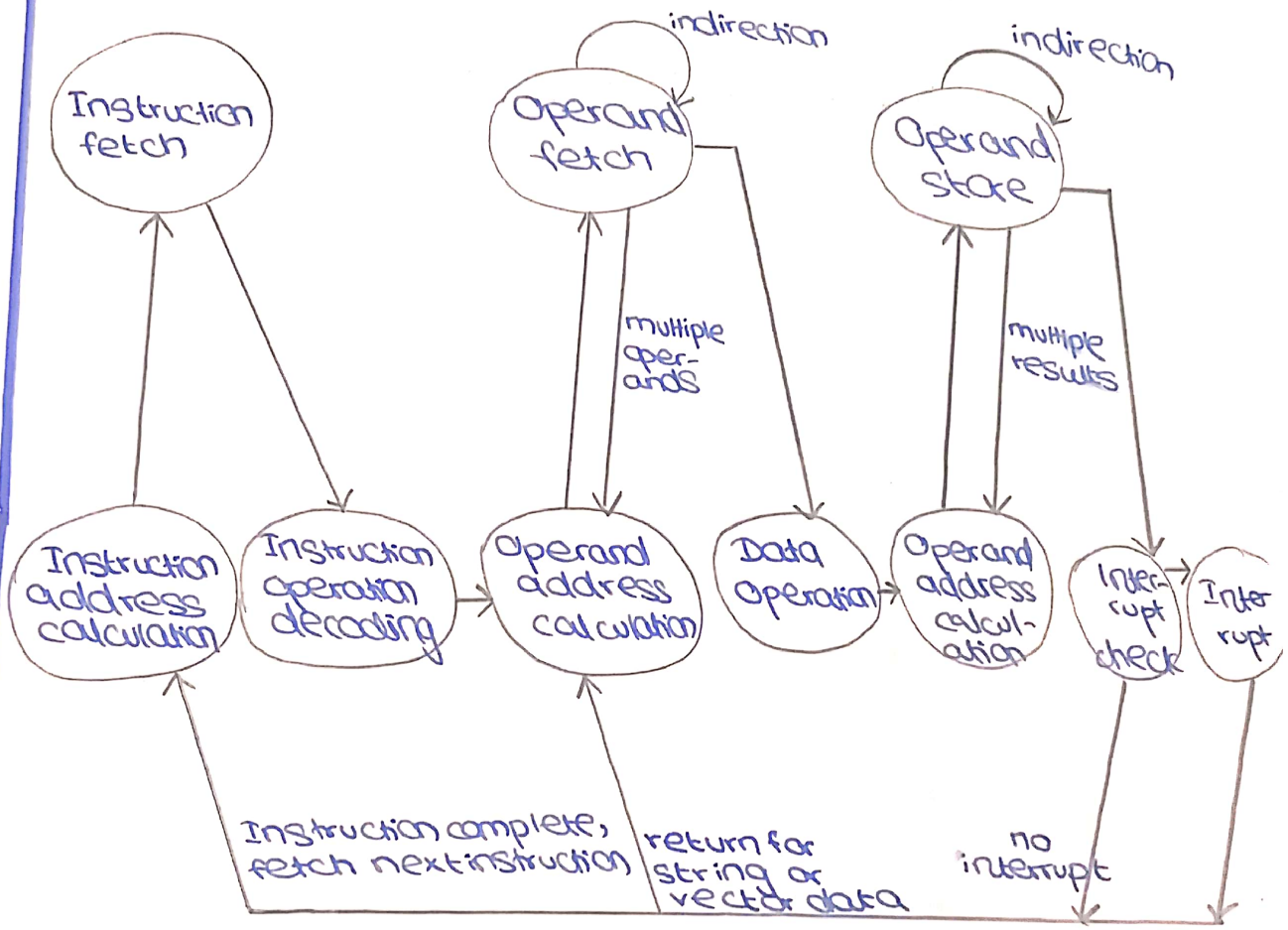
INSTRUCTION EXECUTION CYCLE



Q2 Describe instruction cycle state with the help of diagram

INSTRUCTION CYCLE STATE

DIAGRAM:-



DESCRIPTION:-

For any given instruction cycle, some states may be null and others may be visited more than once. The states can be described as follows:-

Instruction Address Calculation (IAC)

Determines the address of the next instruction to be executed. Usually, this involves adding a fixed number to the address of the previous instruction. For example, if each instruction is 16 bits long and memory is organized into 16-bit word, then add 1 to the previous address. If, instead, memory is organized as individually addressable 8-bit bytes, then add 2 to the previous address.

Instruction Fetch (IF)

Reads instruction from it's memory location into the processor.

Instruction Operation Decoding (IOD)

Analyze instruction to determine type of operation to be performed and operands to be used

Operand Address Calculation (OAC)

If the operation involves reference to an operand in memory of or available via I/O, then determine the address of operand

Operand Fetch (OF)

Fetch the operand from memory or read it in from I/O

Data Operation (DO)

Perform the operation indicated in the instruction

Operand Store (OS)

Write the result into memory or out to I/O

Interrupts

Virtually, all computers provide a mechanism called interrupt, by which other modules (I/O memory) may interrupt the normal processing of the processor. Interrupts are provided primarily as a way to improve processing efficiency.

Q3 Explain X086 architecture in detail

X086 ARCHITECTURE

INTRODUCTION

The X086 architecture is one of the most widely used instruction set architectures (ISA) in the world, forming the foundation of most modern personal computers, servers and embedded systems. Initially, developed by Intel Corporation in the late 1970s, it has undergone multiple iterations and enhancements, making it a dominant force in computing for over four decades.

ORIGINS

The x86 architecture traces it's origins to Intel's 16-bit 8086 microprocessor, released in 1978.

ARCHITECTURAL DESIGN AND FEATURES

It features a complex Instruction set computer (CISC) architecture, supporting multiple operations in a single instruction. It features a 20-bit address bus, allowing access to 1MB of memory and a 16 bit data bus for data trahsfer.

The segmented memory architecture

divides memory into segments addressed via a segment register and an offset, enabling efficient memory access despite the 16-bit data bus. The 8086 consists of two main execution units:

Bus Interface Unit (BIU) – Fetches and decodes instructions, manages memory and handles data transfers.

Execution Unit (EU) – Executes instructions and processes data

It has 14 internal registers (each of 16 bits) and supports pipelining, significantly improving performance.

MEMORY SEGMENTATION

The 20-bit address bus addresses 1MB of memory, segmented into 16 blocks of 64KB each.

The processor operates with only four 64KB segments at a time

Code Segment (CS) – stores program instructions

Data Segment (DS) – stores data variables

Stack Segment (SS) – Manages function calls and interrupts

Extra Segment (ES) – Used for additional memory operations

BUS INTERFACE UNIT (BIU)

The BIU connects the 8086 to external memory and I/O devices and performs

Address Generation

Instruction Fetching = uses a 6 byte prefetch queue (FIFO) for pipelining

Instruction Pointer (IP) = Holds the offset of the next instruction in CS

PREFETCH QUEUE (6 BYTES)

Improves execution speed by fetching while executing

works sequentially but is flushed during a branch instruction

EXECUTION UNIT (EU)

The EU is responsible for

1 Fetching instructions from the BIU queue

2 Decoding and executing arithmetic and logic operations

3 Managing internal data transfers and generating control signals

REGISTERS

General Purpose Registers (AX, BX, CX, DX) — store intermediate values and act as accumulators

Special Purpose Registers (SP, BP, SI, DI) — Handles stack operations and memory indexing

Instruction Register and Decoder - Decodes instructions and coordinates execution

Flag register (16 bit, 9 flags):
- Status Flag (CF, PF, AF, ZF, SF, OF)
- Control Flag (TF, IF, DF)

ALU

Performs 8 bit and 16 bit arithmetic and logic operations

DECODE AND CONTROL UNITS

DECODE UNIT

Translates machine code instructions into micro operations
Handles conditional jumps, loops and data transfers

CONTROL UNIT

Manages instruction execution by coordinating all components
Handles interrupts and system management tasks

8086 BUSES

Address Bus (20 bit) Transmits memory addresses

Data Bus (16 bit) - Transfers data between CPU and memory

Control Bus - sends control signals (read/write, interrupts etc)

DIAGRAM