Computer Organization and Architecture (COA) - Brief Notes

UNIT 1: Structure of Computers

1. **Computer Functional Units:** Main components—Input, Output, Memory, ALU, and Control Unit—work together for processing data.
2. **Von-Neumann Architecture:** Data and instructions share the same memory; follows stored program concept.
3. **Bus Structures:** Data Bus (data transfer), Address Bus (location), Control Bus (control signals).
4. **Basic Operational Concepts:** Instruction cycle—Fetch, Decode, Execute, Store.
5. **Data Representation:** Fixed-point (integers), Floating-point (real numbers), Error codes (parity, CRC).
6. **Register Transfer & Micro Operations:** Register data movement and operations like arithmetic, logic, and shifting.

UNIT 2: Microprogrammed Control

1. **Control Memory:** Stores microinstructions to generate control signals.
2. **Control Unit:** Hardwired (fast, fixed) or Microprogrammed (flexible, slower).
3. **Arithmetic & Logic Operations:** Binary addition, subtraction, multiplication, division, floating-point.
4. **Pipelining:** Overlapping execution stages (Fetch, Decode, Execute) for speed.
5. **Pipeline Types:** Arithmetic, Instruction, and RISC pipelines improve CPU performance.
6. **Vector & Array Processing:** Perform operations on multiple data simultaneously for high-speed computing.

UNIT 3: Introduction to Microprocessor Architecture

1. **Microprocessor Basics:** CPU on a chip performing arithmetic, logic, and control tasks.
2. **Instruction Set:** List of all operations a processor can perform.
3. **Architecture Design Principles:** Focus on efficient hardware design for instruction execution.
4. **Intel 8086 Example:** 16-bit processor; EU and BIU units, registers, segmentation, interrupts, addressing modes.

UNIT 4: Assembly Language Programming

1. **Simple Programs:** Use mnemonics like MOV, ADD, SUB for direct hardware control.
2. **Instructions:** Data Transfer (MOV), Arithmetic (ADD), Logical (AND), Branch (JMP), Call (CALL).
3. **Assembler Directives:** ORG, DB, DW, END guide assembler behavior.
4. **Procedures & Macros:** Reusable blocks and code expansions for simpler programming.

UNIT 5: Memory and Digital Interfacing

1. **Addressing & Decoding:** Ensures correct memory/device access using address lines and decoders.
2. **Interfacing RAM, ROM, EPROM:** Memory connection to CPU via address/data buses.
3. **Programmable Peripheral Interface:** 8255 chip connects CPU with external devices.
4. **Cache Memory:** Small, fast memory that stores frequently used data for quick access.
5. **Virtual Memory:** Uses secondary storage as main memory; uses paging, segmentation, and TLB.

Course Summary:

Understanding computer structure, control, microprocessor architecture, assembly programming, and memory interfacing forms the core of COA. It helps explain how computers work internally and efficiently execute tasks.