# COA ASSIGNMENT
# UNIT – 3 & UNIT – 4

# UNIT – 3

**1. Explain cache memory mapping techniques (Direct, Associative, Set Associative) with neat diagrams.**

Cache mapping decides how memory blocks are placed in cache. Direct mapping maps one block to one line. Associative mapping allows any block to any line. Set associative combines both. These techniques improve memory access speed.

**2. Explain Virtual Memory and Address Translation with neat diagram.**

Virtual memory allows execution of large programs using disk. Address translation is done using page table which converts virtual address to physical address.

**3. Design an 8K × 8 memory module using 1K × 8 static memory chips with diagram.**

To design 8K×8 memory using 1K×8 chips, 8 chips are required. A decoder is used for chip selection while address and data lines are shared.

**4. Explain different interrupt handling schemes with neat diagram.**

Interrupts can be handled using polling, daisy chaining, priority encoder, and vectored interrupts. These methods resolve simultaneous interrupt requests.

**5. Explain handshake protocol for input data transfer with timing diagram.**

Handshake protocol uses request and acknowledge signals to transfer data safely between devices operating at different speeds.

# UNIT – 4

**1. Multiply (-11) × (+27) using Booth's Algorithm.**

Booth's algorithm handles signed multiplication efficiently. The algorithm examines multiplier bits and performs add/subtract with shifts.

**2. Perform 8 ÷ 3 using Restoring and Non-Restoring Division algorithms.**

Restoring division restores the partial remainder when subtraction gives negative. Non-restoring division avoids restoring by performing opposite operation.

**3. Multiply 13 × 11 using Sequential Binary Multiplier.**

Sequential multiplier multiplies using shift and add operations step by step.

**4. Convert 12.890625 into IEEE 32-bit floating point representation.**

IEEE 754 format represents numbers using sign, exponent and mantissa fields.

**5. Add 12.68 and 23.26 using floating point addition.**

Floating point addition involves exponent alignment, mantissa addition, normalization and rounding.