[Concepts] Paper 1

CHAPTER 1: INFORMATION REPRESENTATION

1.1 DATA REPRESENTATION

1.1.1 Fundamental Characteristics of Number Systems

Every number system has two fundamental characteristics:

1. **Base (Radix)**: The number of different digits that a system can use to represent numbers
2. **Place Value**: The specific value of a digit based on its position within a number

1.1.2 Denary (Decimal) System - Base 10

- Uses digits 0-9
- Each position represents powers of 10 (10?, 10^1, 10^2, etc.)
- Example: 3,567 = (3 × 10^3) + (5 × 10^2) + (6 × 10^1) + (7 × 10?)

1.1.3 Binary System - Base 2

Key Points:

- Uses only two digits: 0 and 1
- Each bit (binary digit) represents a power of 2
- All data and characters in computers are represented in binary

Binary Place Values:

<TEXT>

```
128 | 64 | 32 | 16 | 8 | 4 | 2 | 1
 2? 2? 2? 2? 2³ 2² 2¹ 2?
```

Example - Converting Denary to Binary:

- Denary 65 in binary: 01000001
- Calculation: 64 + 1 = 65

Example - Converting Binary to Denary:

- Binary 01000001 = 64 + 1 = 65

1.1.4 Binary Prefixes vs Decimal Prefixes

It is crucial to understand the difference between binary prefixes (based on powers of 2) and decimal prefixes (based on powers of 10):

Denary Prefix	Factor	Value	Binary Prefix	Factor	Value
kilo- (k)	$\times 10^3$	1,000	kibi- (Ki)	$\times 2^1$?	1,024
mega- (M)	×10?	1,000,000	mebi- (Mi)	$\times 2^2$?	1,048,576
giga- (G)	×10?	1,000,000,000	gibi- (Gi)	$\times 2^3$?	1,073,741,824
tera- (T)	$\times 10^{12}$	1,000,000,000,000	tebi- (Ti)	×2??	1,099,511,627,776

Important: Always use the correct prefix:

- Computer storage uses binary prefixes (KiB, MiB, GiB, TiB)
- Data transfer rates often use decimal prefixes (kbps, Mbps, Gbps)

1.1.5 Binary Coded Decimal (BCD)

Definition: Binary representation where each individual denary digit is represented by a sequence of 4 bits (nibble).

Characteristics:

- Each nibble can represent denary digits 0-9
- Uses only specific 4-bit patterns (0000 to 1001)
- The patterns 1010 to 1111 are not used in BCD

Example - Converting 429 to BCD:

<TEXT>

```
4 = 0100
2 = 0010
9 = 1001
Therefore, 429 in BCD = 0100 0010 1001
```

Practical Applications:

- Electronic devices displaying numbers (calculators)
- Accurately measuring decimal fractions
- Electronically coding denary numbers

1.1.6 Two's Complement Representation

Two's complement is used to represent negative numbers in binary.

Converting Negative Denary to Binary (Example: -42):

Step 1: Find binary equivalent (ignoring sign)

```
<TEXT>
42 = 00101010 (8-bit representation)
```

Step 2: Convert to one's complement (flip all bits)

```
<TEXT>
00101010 ? 11010101
```

Step 3: Add 1 to get two's complement

```
<TEXT>
11010101 + 1 = 11010110
```

Converting Binary Two's Complement to Denary (Example: 11010110):

Step 1: Flip all bits

```
<TEXT>
11010110 ? 00101001
```

Step 2: Add 1

```
<TEXT>
00101001 + 1 = 00101010
```

Step 3: Convert to denary and apply negative sign

```
<TEXT>
00101010 = 42
Therefore: -42
```

Range in 8-bit Two's Complement:

- Maximum positive: +127 (01111111)
- Maximum negative: -128 (10000000)

Overflow:

- Occurs when the result of an arithmetic operation is too large/small to fit in the allocated bits
- Example: Adding 127 + 1 in 8-bit gives -128 (overflow)

1.1.7 Hexadecimal System - Base 16

Characteristics:

- Uses digits 0-9 and letters A-F
- A=10, B=11, C=12, D=13, E=14, F=15

Converting Denary to Hexadecimal: Example: 165 to Hex

<TEXT>

```
165 ÷ 16 = 10 remainder 5
10 = A
Therefore: 165 = A5 (hex)
```

Converting Hexadecimal to Denary: Example: A5 to Denary

<TEXT>

```
A5 = (10 × 16) + (5 × 1) = 160 + 5 = 165
```

Practical Applications:

- Defining colours in HTML (#FF0000 = red)
- Defining MAC addresses
- Assembly languages and machine code
- Debugging via memory dumps

1.1.8 Character Sets and Encoding

Definition: A character set is a collection of characters that can be represented using binary codes. It typically includes upper and lower case letters, number digits, punctuation marks, and other characters.

Character Encoding Standards:

Standard	Description	Bits per Character	Characters

ASCII	American Standard Code for Information Interchange	7 bits	128
Extended ASCII	Extension of ASCII	8 bits	256
Unicode	Superset of ASCII and extended ASCII	16 or 32 bits	65,536+

ASCII:

- Only supports English alphabet
- 7 bits = 128 possible characters
- Includes control characters (0-31), printable characters (32-126)

Extended ASCII:

- 8 bits = 256 possible characters
- Includes most European languages' alphabets
- Still limited for global languages

Unicode:

- Modern international standard
- Supports all global languages
- UTF-8 uses 1-4 bytes per character
- Backward compatible with ASCII

1.2 MULTIMEDIA - GRAPHICS AND SOUND

1.2.1 Bitmap Images

Definition: Bitmap images are created by assigning a solid colour to each pixel using bit patterns. The image is represented as a grid of pixels, where each pixel's colour is encoded using binary values.

Key Terms:

- **Pixel:** The smallest picture element whose colour can be accurately represented by binary code
- **File Header:** Contains metadata including image size, number of colours, etc.

Image Resolution:

- Definition: The number of pixels that make up an image
- Example: 4096 × 3192 pixels
- Effect: Higher resolution results in sharper, more detailed images

Screen Resolution:

- Definition: The number of pixels that can be viewed horizontally and vertically on a device's screen
- Example: 1680 × 1080 pixels

Colour Depth:

- Definition: The number of bits used to represent the colour of a single pixel
- Formula: If n bits are used, there are 2? colours per pixel
- Example: 16-colour bitmap = 4 bits per pixel (2? = 16)
- Effect: Increasing colour depth improves colour quality but increases file size

File Size Calculation:

<TEXT>

```
File Size = Number of Pixels × Colour Depth
```

Example Calculation:

<TEXT>

```
Image: 1024 × 768 pixels, 24-bit colour
Number of Pixels = 1024 × 768 = 786,432
Colour Depth = 24 bits
File Size = 786,432 × 24 = 18,874,368 bits
 = 18,874,368 ÷ 8 = 2,359,296 bytes
 ? 2.36 MB
```

Applications:

- Scanned images
- Digital photographs
- Computer screen displays
- Small file sizes and easy manipulation when needed

1.2.2 Vector Graphics

Definition: Made up of drawing objects (mathematically defined constructs like rectangles, lines, circles, curves).

Components:

- **Drawing List:** A set of commands defining the vector
- **Properties:** Basic geometric data determining shape and appearance
- **Encoding:** Data is encoded using mathematical formulas

Advantages over Bitmap:

- Objects can be resized without losing quality

- Scalability is the key benefit
- Smaller file sizes for simple images
- Can be enlarged infinitely without pixelation

Disadvantages:

- Cannot represent complex images like photographs
- More complex to create

Applications:

- Company logos
- Architectural drawings
- Icons and symbols
- Fonts (TrueType, PostScript)

1.2.3 Sound Representation

Analogue vs Digital:

Analogue	Digital
Continuous electrical signals	Discrete electrical signals
Infinite detail	Finite representation
Cannot be stored directly	Can be stored in binary

Sound as Analogue Data:

- Sound consists of vibrations through a medium
- Inherently analogue due to infinite detail variation

Conversion Process (Analogue to Digital):

1. **Sampling:** The sound wave's amplitude is measured at set time intervals
2. **Quantization:** Each sample is assigned a binary value
3. **Encoding:** Binary values are stored

Key Terms:

- **Sampling Rate:** Number of samples taken per unit of time (measured in Hz)
  - Effect: Increasing sampling rate improves accuracy but increases file size
  - CD quality: 44,100 Hz
- **Sampling Resolution:** Number of bits used to encode each sample

- Effect: Increasing resolution improves accuracy but increases file size
- CD quality: 16 bits

- **Bit Rate:** Number of bits used to store 1 second of sound
  - Formula: Bit Rate = Sampling Rate × Sampling Resolution
  - Example: 44,100 × 16 = 705,600 bps (approximately 706 Kbps)

1.3 COMPRESSION

1.3.1 Need for Compression

Definition: Compression is the process of reducing file size without significant loss in quality.

Benefits:

- Reduced storage requirements
- Faster data transfer (uses less bandwidth)
- Reduced time needed to search for data

1.3.2 Lossless Compression

Definition: A type of compression that allows original data to be perfectly reconstructed from the compressed file.

Key Feature:

- Uses some form of replacement (substitution)
- No data is permanently deleted

Examples:

- PNG images (for graphics with sharp edges)
- ZIP files
- Text file compression
- Database records
- Run-Length Encoding (RLE)

Run-Length Encoding (RLE):

Definition: A form of lossless compression used for compressing text files and bitmap images.

Mechanism:

- Reduces file size by encoding sequences of adjacent, identical elements
- Encodes as two values: run count and run value

Example: Original: AAAAAAABBBBBCCCCCC Compressed: 7A5B6C

Example - Bitmap: Original row: White White White White White Black Black Compressed: 5W2B

Applications:

- Simple graphics with large areas of same colour
- Database records with repeated values

1.3.3 Lossy Compression

Definition: A type of compression that irreversibly eliminates unnecessary data.

Characteristics:

- File accuracy/quality is lower than lossless
- File size is significantly reduced (often to about 10% of lossless size)
- Some original data is permanently lost

Examples:

- MP3 (sound files)
- JPEG (images)
- MP4 (video files)

Mechanism in Sound Files (MP3):

- **Perceptual Coding:** Removes parts of the sound that are less audible or discernible to human hearing
- Removes frequencies outside human hearing range
- Removes subtle volume differences

Mechanism in Images (JPEG):

- Removes high-frequency details
- Uses mathematical approximations
- Reduces colour precision in less important areas

When to Use Lossy vs Lossless:

Lossless	Lossy
Text documents	Photography
Database files	Video streaming
Program files	Music (streaming)
Spreadsheets	Web graphics (where size matters)

CHAPTER 2: COMMUNICATION AND NETWORKING

2.1 NETWORKS INCLUDING THE INTERNET2.1.1 Introduction to Networks

Definition: Networking devices are interconnected devices that enable fast data transmission within a network.

Benefits of Networking:

1. **File Sharing:** Easily share data between different interconnected devices
2. **Resource Sharing:** Use network-connected output devices like printers or share software
3. **Higher Storage:** Files can be stored in network-connected storage mediums
4. **Communication:** Email and messaging between users
5. **Centralized Management:** Easier to backup and secure data

2.1.2 Types of Networks

LAN (Local Area Network):

Characteristic	Description
Geographic Area	Small area, often within the same building
Ownership	Private ownership
Transmission Medium	Twisted pair cables, coaxial cables, or Wi-Fi
Data Transfer Rate	Higher (100 Mbps to 10 Gbps)

Characteristic	Description
Congestion	Less congestion
Setup Cost	Lower initial cost

WAN (Wide Area Network):

Characteristic	Description
Geographic Area	Large area - city, country, or globally
Ownership	Private or public ownership
Transmission Medium	PSTN, satellite links, leased lines
Data Transfer Rate	Lower
Congestion	Higher congestion
Setup Cost	Higher initial cost

2.1.3 Network Models

Client-Server Model:

- A dedicated server provides applications (administration of users, security, and resources)
- Clients access resources from the server
- Server manages all central functions

Server Types:

- Print Server: Manages print jobs from client computers
- File Sharing Server: Clients access software and user data files stored on the server
- Proxy Server: Acts as intermediary for internet requests
- Email Server: For sending, receiving, and storing emails
- Database Server: Manages DBMS
- Domain Controller Server: Manages user accounts (IDs & passwords)

Client-Server Login Process:

1. Client sends login request to server

2. Server processes request
3. Server grants access if user ID & password are recognized

Thin Clients vs Thick Clients:

Thin Clients	Thick Clients
Runs solely on server resources	Processes most application locally
No local storage	Has local storage and processing power
Smaller purchase cost	More expensive
Requires constant server connection	Can work offline
Improved security	More vulnerable to unauthorized software

Peer-to-Peer (P2P) Network:

- Decentralized network
- Each computer operates independently as a 'peer'
- Acts as both client and server
- No central server required
- Common in file sharing applications (BitTorrent)

Comparison:

Feature	Client-Server	Peer-to-Peer
Centralized backup	Yes	No
Initial setup cost	Higher	Lower
Network traffic	Higher (goes through server)	Lower (direct)
Security	Better (centralized control)	Weaker
Reliability	Server failure affects all	Peer failure affects one

2.1.4 Network Topologies

Bus Topology:

- Single line (bus) connects all devices
- Terminators at each end
- All computers can read data being sent between any two computers
- **Disadvantage:** Unsuitable for heavy traffic due to frequent collisions
- Uses CSMA/CD for collision detection

Star Topology:

- Central server/switch with all computers connected via dedicated connections
- Server can send packets to different devices simultaneously
- No collisions possible
- Most common in modern networks

Mesh Topology:

- Every device (node) is directly interconnected with all other devices
- Commonly used for wireless networks
- High redundancy (if one path fails, another can be used)

Hybrid Topology:

- Combination of two or more topologies
- Example: Connection between two or more LANs of different topologies

2.1.5 Transmission Media

Wired Networks:

Copper Cable:

Advantages	Disadvantages
Less expensive	Doesn't perform well with small charges
Easier to install	Affected by electromagnetic interference
Flexible	Signal degradation over distance
Easy to make terminations	

Fibre-Optic Cable:

Advantages	Disadvantages
Greater bandwidth	Needs expensive optical transmitters/receivers
Improved security	More difficult to install
Lightweight	
Less signal boosting required	
Used in long-distance communications	
Immune to electromagnetic interference	

Wireless Networks:

Type	Advantages	Disadvantages
Radio Waves	Can travel over large distances; inexpensive	Low frequency = less data; affected by interference
Microwaves	Larger bandwidth	Physical obstacles interfere; expensive towers
Satellites	Cost-effective for long-distance	Expensive setup; susceptible to interference

2.1.6 Ethernet

Definition: The most common wired medium for data transmission in LANs or WANs.

CSMA/CD (Carrier Sense Multiple Access with Collision Detection):

Process:

1. Device checks if channel is busy before transmitting
2. If busy, device waits a random time before retrying
3. During transmission, device listens for other transmissions
4. If collision occurs, transmission is aborted
5. Both devices wait random times before retrying

2.1.7 Bit Streaming

Definition: Sequence of digital signals (bits) transferred over a communication path at high speeds.

Types:

Real-time Streaming:

- Live events captured and transmitted directly
- Cannot be paused, fast-forwarded, etc.
- Example: Live TV, video conferencing

On-demand Streaming:

- Pre-existing files are converted and streamed as requested
- Can be paused, fast-forwarded, etc.
- Example: YouTube, Netflix

Importance of High Broadband Speed:

- User has to download and display bits simultaneously
- Higher quality media requires faster speeds
- Real-time streaming needs higher speeds for simultaneous data requests

2.1.8 Cloud Computing

Definition: On-demand provision of computing services over the internet.

Services:

- **Infrastructure:** Storage capacity and higher processing power
- **Platform:** Software, testing & debugging resources

Public Cloud vs Private Cloud:

Feature	Public Cloud	Private Cloud
Access	Third-party providers, shared among multiple users	Single organization, exclusive access
Management	Managed by service providers	Can be managed internally or outsourced

Benefits:

- Less technical knowledge required
- Easy to implement
- Flexibility to scale with growth

Drawbacks:

- Cannot access resources if bandwidth issues
- Poor data privacy (potential data leakage in multi-tenant architecture)

2.1.9 Internet and World Wide Web

Internet:

- Massive, open network of networks
- Uses TCP/IP protocol
- IP addresses identify devices

World Wide Web (WWW):

- Collection of web pages stored on websites
- Uses protocols (HTTP/HTTPS) to transmit data

2.1.10 Network Hardware

Device	Function
Router	Connects two networks; translates IP addresses; acts as gateway and firewall
Switch	Connects devices in LAN; broadcasts to all devices simultaneously
Server	Provides specific functions for computers in the network
NIC (Network Interface Card)	Provides unique MAC address for wired connection
WNIC (Wireless NIC)	Provides unique address for WiFi connection
WAP (Wireless Access Point)	Allows devices to connect via WiFi
Bridge	Connects two LANs using same protocol

Device	Function
Repeater	Regenerates signal to prevent attenuation
Modem	Converts digital to analogue signals for telephone lines

2.1.11 IP Addressing

IPv4:

- 32-bit address
- 4 blocks separated by dots (e.g., 192.168.1.1)
- Each block: 0-255

IPv6:

- 128-bit address
- 8 blocks separated by colons
- Each block: 4 hex values (0000-FFFF)
- Can be shortened by removing consecutive zero blocks

IP Address Structure:

- **Network Identifier (NetID):** Identifies the network
- **Host Identifier (HostID):** Identifies the device within the network

Subnetting:

- Practice of dividing a network into two or more sub-networks
- IP address broken into: NetID + SubnetID + HostID

Public vs Private IP:

- Public IP: Provided by ISP; unique; accessible from internet
- Private IP: Issued by LAN's router; only accessible within LAN
- NAT (Network Address Translation) required for private IPs to access internet

Static vs Dynamic:

- Static: Never changes; used for servers, VPNs
- Dynamic: Changes regularly; more secure; used for general users

2.1.12 DNS (Domain Name Service)

Definition: A naming system that maps domain names to IP addresses.

Function:

- Hierarchy of DNS servers
- Database of URLs and corresponding IP addresses
- Translates human-readable domain names to IP addresses

CHAPTER 3: HARDWARE AND COMPUTER COMPONENTS

3.1 COMPUTERS AND THEIR COMPONENTS

3.1.1 General-Purpose Computer System

A general-purpose computer system comprises:

1. **Processor (CPU):** Executes instructions
2. **Memory:** Stores data and instructions
3. **Input/Output (I/O):** Interfaces with external world

3.1.2 Essential Features of a Computer

Input:

- Takes in data from the outside world
- Devices: keyboard, mouse, scanner, microphone

Output:

- Displays data for human understanding
- Devices: monitor, printer, speaker

Primary Storage (Memory):

- Main memory storing critical program instructions and data
- RAM (Random Access Memory)
- Volatile (loses data when power off)

Secondary Storage:

- Non-volatile storage for data
- Examples: hard disk, SSD, USB flash drive

Removable Secondary Storage:

- File backup and archives
- Portable transfer of files

3.1.3 Embedded Systems

Definition: Miniature computer systems (microprocessors) that are often part of a larger system.

Characteristics:

- Perform few specific functions
- Not general-purpose computers
- No moving parts (more reliable)

Advantages:

- Reliable (no moving parts)
- Require less power
- Cheap to mass-produce

Disadvantages:

- Difficult to program (no interface)
- Expensive expert help needed for repair

3.1.4 Hardware Device Operations

Laser Printer:

1. Laser beam and rotating mirrors draw image on photosensitive drum
2. Image converted to electric charge
3. Charged toner attracts to image
4. Electrostatic-charged paper rolls against drum
5. Charge pulls toner onto paper
6. Heat in fuser fuses toner to paper

3D Printer:

1. Starts with saved digital file (blueprint)
2. Object built by adding layers of material (polymer resin)
3. Object cured (hardened by UV light)

Microphone:

1. Sound waves enter and cause diaphragm vibrations
2. Vibrations cause coil to move past magnetic core
3. Electrical current generated
4. Current digitized

Speaker:

1. Takes electrical signals
2. Voice coil generates electromagnetic field
3. Change in audio signal changes current direction
4. Electromagnet attracted/repelled to permanent magnet
5. Diaphragm vibrates, creating sound waves

Magnetic Hard Disk:

1. Platters covered with magnetizable material
2. Mounted on central spindle, rotated at high speed
3. Surface divided into concentric tracks & sectors
4. Data encoded as magnetic patterns
5. Read/write heads access data

Solid State (Flash) Memory:

- Uses NAND-based flash memory
- Grid of columns & rows with 2 transistors at each intersection
- Floating gate stores electrons (represents 0 or 1)
- Control gate controls charge flow

Optical Disc Reader/Writer:

1. Disc surface has reflective metal layer
2. Spun at high speed
3. Laser beam reads/writes
4. Tracks have amorphous and crystalline states
5. Different states encode bit patterns

Touchscreen:

Type	Description
Resistive	Two charged plates; pressure causes contact
Capacitive	Materials that store electric charge; touch transfers charge

3.1.5 Memory Types

RAM vs ROM:

Feature	RAM	ROM
Volatility	Volatile (loses data when power off)	Non-volatile
Read/Write	Can be read and written	Can only be read
Use	Stores currently executing program	Stores OS kernel, boot-up instructions

Static RAM vs Dynamic RAM:

Feature	SRAM	DRAM
Refresh	Doesn't need to refresh	Must be refreshed
Speed	Faster access time	Slower access time
Power	Uses less power	Needs higher power
Cost	More expensive (complex circuitry)	Less expensive
Structure	Each bit stored in flip-flop	Each bit stored as charge in capacitor
Density	Lower data density	Higher data density
Use	Cache memory	Main memory

PROM vs EPROM vs EEPROM:

Type	Description	Erasure Method
PROM	Programmable once after creation	Cannot be erased
EPROM	Can be reprogrammed	UV light exposure
EEPROM	Can be reprogrammed	Electrical signal

3.1.6 Monitoring and Control Systems

Monitoring System:

- Monitors state external to computer
- No changes made to environment
- No feedback required

Control System:

- Regulates behaviour of other devices/systems
- **Event-driven:** Responds to specific events
- **Time-driven:** Takes action at specific times

Components:

- **Sensor:** Measures analogue property, transmits to processing unit
- **Actuator:** Switches on/off heavy appliances (heater, fan)
- **ADC:** Converts analogue signals to digital
- **Transmission cable:** Transfers signals

Feedback Systems:

- Output affects input of sensors
- Ensures system operates within criteria
- Enables automatic adjustment of conditions

3.2 LOGIC GATES AND LOGIC CIRCUITS

3.2.1 Basic Logic Gates

AND Gate:

- Output is HIGH only if ALL inputs are HIGH
- Symbol: A · B or A AND B

A	B	Output
0	0	0
0	1	0
1	0	0
1	1	1

OR Gate:

- Output is HIGH if ANY input is HIGH
- Symbol: A + B or A OR B

A	B	Output
0	0	0
0	1	1
1	0	1
1	1	1

NOT Gate (Inverter):

- Output is opposite of input
- Symbol: A or NOT A

A	Output
0	1
1	0

3.2.2 NAND and NOR Gates

NAND Gate:

- Opposite of AND (NOT-AND)
- Symbol: A · B

A	B	Output
0	0	1
0	1	1
1	0	1
1	1	0

NOR Gate:

- Opposite of OR (NOT-OR)
- Symbol: A + B

A	B	Output
0	0	1
0	1	0
1	0	0
1	1	0

3.2.3 XOR Gate

XOR (Exclusive OR):

- Output is HIGH if inputs are DIFFERENT
- Symbol: A ⊕ B

A	B	Output
0	0	0

A	B	Output
0	1	1
1	0	1
1	1	0

3.2.4 Constructing Circuits from Truth Tables

Process:

1. Identify required output for each input combination
2. Identify the logic expression
3. Design the circuit using appropriate gates
4. Test with truth table

CHAPTER 4: PROCESSOR FUNDAMENTALS

4.1 CENTRAL PROCESSING UNIT (CPU) ARCHITECTURE

4.1.1 Von Neumann Model

Key Concept:

- Data and programs are indistinguishable
- Can use the same memory for both
- Uses a single processor
- Follows fetch-decode-execute cycle

Components:

- Processor (CPU)
- Memory (for data and instructions)
- Input/Output devices

4.1.2 Registers

Definition: The smallest unit of storage in a microprocessor; allows fast data transfer.

General Purpose Registers:

- Used to temporarily store data values
- Can be used by assembly language instructions
- Example: Accumulator (ACC)

Special Purpose Registers:

Register	Function

PC (Program Counter)	Holds address of next instruction
MDR (Memory Data Register)	Holds data fetched from memory
MAR (Memory Address Register)	Holds address of memory cell to access
ACC (Accumulator)	Holds values processed by ALU
IX (Index Register)	Stores number to modify address
CIR (Current Instruction Register)	Holds current instruction for decoding
Status Register	Holds results of comparisons, arithmetic flags

4.1.3 CPU Components

ALU (Arithmetic and Logic Unit):

- Part of processor that processes instructions
- Performs arithmetic operations (add, subtract, multiply, divide)
- Performs logical operations (AND, OR, NOT)

Control Unit (CU):

- Fetches instructions from memory
- Decodes instructions
- Synchronizes operations
- Sends signals to memory, ALU, and I/O devices

System Clock:

- Timing device connected to processor
- Synchronizes all components
- Generates clock pulses

IAS (Immediate Access Store):

- Memory unit the processor can directly access

4.1.4 Buses

Definition: Set of parallel wires allowing data transfer between components.

Data Bus:

- Bidirectional
- Carries data between processor, memory, and I/O devices

Address Bus:

- Unidirectional
- Carries memory address from processor to MAR

Control Bus:

- Bidirectional
- Transmits control signals from CU
- Ensures components don't conflict

4.1.5 Performance Factors

Clock Speed:

- Number of pulses the clock sends in a given time
- Usually measured in GHz
- Higher clock speed = more cycles per second = faster execution
- Limitation: Heat generation at high speeds

Bus Width:

- Number of bits that can be transferred simultaneously
- Wider bus = more bits transferred at once = faster processing

Cache Memory:

- Stores commonly used instructions
- Larger cache = more instructions stored = less waiting = better performance

Number of Cores:

- Multi-core processors have multiple processing units
- Each core can process different instructions simultaneously
- Improves performance through parallel processing

4.1.6 Ports

Port Type	Description
USB	Connects input and output devices
HDMI	High-definition video and audio output
VGA	Video output only (older displays)

4.1.7 Fetch-Execute Cycle

Fetch Stage:

1. PC holds address of next instruction
2. Address copied to MAR
3. PC incremented
4. Instruction loaded to MDR from address in MAR
5. Instruction from MDR loaded to CIR

Decode Stage:

- Opcode and operand parts identified

Execute Stage:

- Control unit executes instruction
- Return to start

Register Transfer Notation (RTN):

<TEXT>

```
MAR ← [PC]
PC ← [PC] + 1
MDR ← [[MAR]]
CIR ← [MDR]
Decode
Execute
Return to start
```

4.1.8 Interrupts

Definition: A signal from a program seeking the processor's attention.

ISR (Interrupt Service Routine):

- Handles the interrupt
- Different ISRs for different sources

Interrupt Handling Process:

1. Processor checks interrupt register at end of F-E cycle
2. If interrupt flag is set, source detected
3. If low priority, interrupt disabled
4. If high priority:
    - Save register contents to stack
    - Load PC with ISR address
    - Execute ISR
    - Restore registers from stack
    - Continue interrupted program

4.2 ASSEMBLY LANGUAGE

4.2.1 Introduction

Assembly Language:

- Low-level programming language
- Instructions consist of opcode and operand

Machine Code:

- Binary code the processor executes directly
- One-to-one relationship with assembly language

Assembler:

- Software that translates assembly language to machine code
- Replaces mnemonics and labels with binary values

4.2.2 Assembler Types

One-Pass Assembler:

- Converts source code in one sweep
- Cannot handle forward referencing

Two-Pass Assembler:

Pass 1:

- Creates symbol table
- Enters symbolic addresses and labels

Pass 2:

- Translates to machine code using table
- Reports errors

4.2.3 Addressing Modes

Mode	Description
Immediate	Data is the actual value (e.g., LDM #n)
Direct	Load contents at given address (e.g., LDD address)
Indirect	Address to use is at given address (e.g., LDI address)
Indexed	Address = given address + contents of IX (e.g., LDX address)
Relative	Next instruction is offset from current instruction

4.2.4 Instruction Types

Data Movement:

- LDM#: Load immediate
- LDD: Load direct
- LDI: Load indirect
- LDX: Load indexed
- STO: Store

Arithmetic:

- ADD: Add to accumulator

- SUB: Subtract from accumulator
- INC: Increment
- DEC: Decrement

Comparing:

- CMP: Compare with contents at address
- CMP#: Compare with immediate value

Conditional Jumps:

- JPE: Jump if equal (compare TRUE)
- JPN: Jump if not equal (compare FALSE)

Unconditional Jumps:

- JMP: Jump to address

I/O:

- IN: Input character
- OUT: Output character

End:

- END: Return control to OS

4.3 BIT MANIPULATION

4.3.1 Binary Shifts

Left Shift (LSL #n):

- Bits shifted left
- Multiplies by 2^n
- Zeros fill vacated positions

Right Shift (LSR #n):

- Bits shifted right
- Divides by 2^n

- Zeros fill vacated positions

Arithmetic Shift:

- Used for signed integers
- Sign bit (MSB) maintained

Cyclic Shift:

- Bit removed from one end added to other end

4.3.2 Bit Masking

Purpose: Each bit can represent an individual flag. By manipulating bits, flags can be operated upon.

Operations:

Masking to 1:

- Use OR operation with 1

Masking to 0:

- Use AND operation with 0

Testing Bits:

- Use AND to mask bits
- Compare result with pattern

Practical Applications:

- Setting individual bit positions
- Testing specific bits
- Checking patterns

CHAPTER 5: SYSTEM SOFTWARE

5.1 OPERATING SYSTEMS

5.1.1 Need for Operating System

Definition: A set of programs designed to run in the background on a computer system.

Functions:

- Controls operation of computer system
- Provides user interface
- Controls how computer responds to requests
- Controls how hardware communicates
- Provides environment for application software

Why OS is Essential:

- Hardware is unusable without OS
- Acts as interface between user and hardware
- Manages communication between components

5.1.2 Key Management Tasks

Memory Management:

- Memory protection (ensures programs don't use same memory)
- Paging (uses virtual memory)
- Memory allocation

File Management:

- Provides file naming conventions
- Maintains directory structure
- Allocates space to files

- File access control

Security Management:

- Provides usernames & passwords
- Ensures data privacy
- Prevents unauthorized access
- Carries out automatic backup

Hardware Management:

- Installation of driver software
- Controls access to peripherals
- Handles interrupts from devices

Process Management:

- Enables multiprogramming and multitasking
- Resolves conflicts when processes need same resource
- Methods: Round-robin, priority scheduling

5.1.3 Utility Software

Disk Formatter:

- Prepares hard disk for data storage
- Deletes existing data
- Creates sectors and tracks

Virus Checker:

- Checks for viruses
- Removes viruses found
- Monitors incoming/outgoing files

Defragmentation Software:

- Reorganizes files to contiguous sectors
- Reduces read/write head movements
- Improves performance

Disk Repair Software:

- Visualizes disk space usage
- Reports errors (bad sectors)
- Attempts to fix issues

File Compression:

- Reduces file size
- Removes redundant data
- Improves storage efficiency

Backup Software:

- Makes copies of files
- Stores on different medium
- Provides synchronization between devices

5.1.4 Program Libraries

Definition: Pre-written code that can be linked to software under development.

Benefits:

- Saves time (less code to write)
- Smaller testing time (pre-tested)
- Complex algorithms available without understanding implementation

DLL (Dynamic Link Library):

- Shared library file containing code and data
- Loaded to memory only when required
- Available to several applications simultaneously
- Reduces .EXE file size

5.2 LANGUAGE TRANSLATORS

5.2.1 Assembler

Purpose: Translates assembly language to machine code (binary).

Characteristics:

- One-to-one relationship with machine code
- Simple translation process

5.2.2 Compiler vs Interpreter

Feature	Compiler	Interpreter
Translation	Translates entire program before execution	Translates line-by-line
Output	Creates .exe file	No .exe created
Execution	Faster (already compiled)	Slower (translates each time)
Error Reporting	All errors at end	Stops at first error
Development	Used when development complete	Used during development
Debugging	Difficult (all errors at end)	Easier (stops at error)

5.2.3 Two-Step Translation (Java)

Process:

1. Java compiler translates source code to bytecode
2. Java Virtual Machine (JVM) interprets bytecode to machine code

Benefits:

- Platform independence (write once, run anywhere)

5.2.4 IDE Features

Coding Features:

- Context-sensitive prompts
- Variable highlighting (undeclared/unassigned)
- Autocomplete

Error Detection:

- Dynamic syntax checking

- Type checking
- Parameter checking

Presentation:

- Prettyprint (automatic indentation, colour-coding)
- Expand and collapse code blocks

Debugging:

- Single stepping (execute line-by-line)
- Breakpoints (pause at specific line)
- Variables/expressions report window

CHAPTER 6: SECURITY, PRIVACY AND DATA INTEGRITY

6.1 DATA SECURITY

6.1.1 Definitions

Data Security:

- Ensuring data is protected against loss and unauthorized access
- Protection of data on computer system

Data Integrity:

- Ensuring data is valid and does not corrupt after transmission
- Data is accurate and reliable

Data Privacy:

- Ability to determine what data is shared with third party

6.1.2 Threats to Computer and Data Security

Malware:

- Software designed to damage computer or network
- **Virus:** Replicates by inserting copy into other software; can crash computer, delete/corrupt data
- **Spyware:** Gathers information about users' activity; monitors online/offline behaviour

Hacking:

- Illegal access to computer system
- Obtain confidential data
- Can cause identity theft
- Can delete/corrupt data

Phishing:

- Emails attempting to obtain confidential data
- Impersonates legitimate organizations
- Causes identity theft

Pharming:

- Redirects users to fake websites
- Appears legitimate
- Gains confidential data

6.1.3 Security Measures

User Accounts and Passwords:

- Deny unauthorized access
- User-assigned privilege levels
- File access permissions

Firewalls:

- Filters information between computer and internet
- Detects illegal connection attempts
- Blocks unauthorized access

Authentication:

- Determines if someone is who they claim
- Methods: passwords, digital signatures, biometric scans

Anti-virus Software:

- Detects and removes viruses
- Checks files for malicious patterns
- Runs in background

Anti-spyware Software:

- Detects and removes spyware

Encryption:

- Converts data to code
- Doesn't stop access but makes data meaningless
- Requires decryption to read

Data Backup:

- Exact copy of original data
- Stored at different location
- Disk-mirroring: writes to multiple disks simultaneously

6.1.4 Data Security vs System Security

Data Security	System Security
Protection of data on system	Protection of computer system
Prevents corruption, unauthorized use	Prevents viruses, hacking
Example: encryption	Example: firewall, passwords

6.2 DATA INTEGRITY

6.2.1 Data Validation

Definition: Checks if data entered is valid (sensible).

Methods:

Method	Description
Range Check	Data must be between set values
Format Check	Data must follow correct pattern
Length Check	Data must have exact number of characters
Presence Check	Checks if data has been entered
Existence Check	Data entered must exist in database

Method	Description
Limit Check	Value within acceptable min/max
Check Digit	Arithmetic result of other digits; verifies accuracy

6.2.2 Data Verification

Definition: Checks data entered is accurate (correct).

Data Entry Verification:

Method	Description
Visual Check	Person manually compares original with entered data
Double Entry	Enter data twice; compares results

Data Transfer Verification:

Parity Check:

- Number of 1s in byte must be odd or even
- If parity doesn't match, error detected
- Limitation: Cannot detect 2-bit transposition

Checksum Check:

- Data sent as block of bytes
- All bytes added together
- Checksum calculated before and after transmission
- If different, error occurred; block must be resent

CHAPTER 7: ETHICS AND OWNERSHIP

7.1 ETHICS AND OWNERSHIP

7.1.1 Ethics

Definition: A system of moral principles that guide behaviour based on philosophical views.

Computer Ethics:

- Regulates how computing professionals should make decisions
- Professional and social conduct

Professional Bodies:

- BCS (British Computer Society)
- IEEE (Institute of Electrical and Electronic Engineers)
- Provide codes of conduct for members

7.1.2 Ownership

Data Ownership:

- Legal rights and complete control over data
- Can determine who accesses, modifies, shares

Copyright:

- Gives creators rights to control use and distribution
- Protects intellectual property

Need for Legislation:

- Competitors can steal programming ideas
- Software can be easily copied
- Bootlegging (illegal selling)

7.1.3 Software Licensing

Type	Description
Free Software Foundation	Users have freedom to run, copy, distribute, study, change, improve; copyleft (modified versions must also be free)
Open Source Initiative	Source code available; users can review and redistribute
Shareware	Free trial period; payment expected after evaluation
Commercial	Requires payment; includes all features

7.1.4 Artificial Intelligence (AI)

Definition: The ability of computer to perform tasks conventionally associated with human intelligence.

AI Capabilities:

- Learn from past mistakes
- Adapt to prevent problems recurring
- Predict what might happen and raise alerts
- Learn to work more efficiently

AI Applications:

- Autonomous vehicles
- Machine learning through data sets
- Natural language processing
- Robotics

AI Impacts:

Area	Impact
Social	Replacement of manual labour; unemployment; increased leisure time

Area	Impact
Economic	Lower manufacturing costs; increased innovation
Environmental	Detrimental impact; robot manufacture; waste disposal

CHAPTER 8: DATABASE CONCEPTS AND MANAGEMENT

8.1 DATABASE CONCEPTS

8.1.1 Limitations of File-Based Systems

Disadvantages:

- No enforcing control on organization/structure
- Data repeated in different files
- Manual changes required in all files
- Sorting must be done manually or by program
- Data may be in different formats
- Cannot be multi-user (chaotic)
- Security not sophisticated

8.1.2 Database Approach

Database:

- Collection of non-redundant interrelated data
- Organized for efficient access

DBMS (Database Management System):

- Software programs that allow databases to be defined, constructed, and manipulated

8.1.3 Relational Database Terminology

Term	Definition

Entity	Object/event that can be distinctly identified
Table	Contains related entities in rows and columns
Tuple	Row/record in relational database
Attribute	Field/column in relational database
Primary Key	Attribute that uniquely identifies each tuple
Candidate Key	Attribute that can potentially be primary key
Foreign Key	Attribute that relates two different tables
Secondary Key	Candidate key not chosen as primary
Referential Integrity	Prevents inconsistent data in relationships

8.1.4 Entity-Relationship Diagrams

Relationships:

- **One-to-Many:** One record relates to many records
- **One-to-One:** One record relates to one record
- **Many-to-Many:** Many records relate to many records (requires intermediate table)

8.2 DATABASE MANAGEMENT SYSTEMS (DBMS)

8.2.1 Features of DBMS

Data Management:

- Data stored in relational tables
- Secondary storage

Data Dictionary:

- List of all files
- Number of records
- Names and types of fields

Data Modelling:

- Analysis of data objects

- Identifying relationships

Logical Schema:

- Overall view of database
- Entities, attributes, relationships

Data Integrity:

- Block copied when changed
- Saved back when done

Data Security:

- Password allocation
- Automatic backups
- Access rights control

Data Change Handling:

- Exclusive mode (impractical for multiple users)
- Lock all records in table
- Lock current record only
- Warn of simultaneous change

Deadlock:

- Two locks at same time
- One user must abort

8.2.2 DBMS Tools

Developer Interface:

- Creates and manipulates database in SQL

Query Processor:

- Handles high-level queries
- Parses, validates, optimizes
- Creates query plan

8.3 NORMALIZATION

8.3.1 First Normal Form (1NF)

Requirements:

- No repeating attributes
- Intersection of each tuple and attribute contains only one value

8.3.2 Second Normal Form (2NF)

Requirements:

- In 1NF
- Every non-primary key attribute fully dependent on primary key
- Remove incomplete dependencies

8.3.3 Third Normal Form (3NF)

Requirements:

- In 1NF and 2NF
- All non-key elements fully dependent on primary key
- No inter-dependencies between attributes

8.3.4 Many-to-Many Relationships

- Cannot be directly normalized to 3NF
- Must use intermediate table (two-step process)

8.4 SQL (STRUCTURED QUERY LANGUAGE)

8.4.1 Data Definition Language (DDL)

CREATE DATABASE:

<SQL>

```
CREATE DATABASE database-name
```

CREATE TABLE:

```
<SQL>

CREATE TABLE table-name (
 field1 data-type,
 field2 data-type,
 ...
)
```

ALTER TABLE:

```
<SQL>

ALTER TABLE table-name ADD field-name data-type
```

PRIMARY KEY:

```
<SQL>

PRIMARY KEY (field)
```

FOREIGN KEY:

```
<SQL>

FOREIGN KEY (field) REFERENCES table(field)
```

8.4.2 Data Manipulation Language (DML)

SELECT (Query):

```
<SQL>

SELECT field-name
FROM table-name
WHERE condition
```

Operators: =, >, <, >=, <=, <>, IS NULL

ORDER BY: Sort ascending

```
<SQL>

ORDER BY field-name
```

GROUP BY: Group identical data

```
<SQL>

GROUP BY field-name
```

INNER JOIN: Combine fields from different tables

```
<SQL>

INNER JOIN table ON condition
```

INSERT:

```
<SQL>

INSERT INTO table (field1, field2)
VALUES (value1, value2)
```

DELETE:

```
<SQL>

DELETE FROM table WHERE condition
```

UPDATE:

```
<SQL>

UPDATE table
SET field = value
WHERE condition
```

Data Types:

- CHARACTER(n)
- VARCHAR(n)
- BOOLEAN
- INTEGER
- REAL
- DATE
- TIME