

CHAPTER 3: HARDWARE AND VIRTUAL MACHINES

3.1 RISC AND CISC PROCESSORS

3.1.1 RISC (Reduced Instruction Set Computers)

Characteristics:

- Fewer instructions
- Simpler instructions
- Small number of instruction formats
- Single-cycle instructions where possible
- Fixed-length instructions
- Only load/store instructions access memory
- Fewer addressing modes
- Multiple register sets
- Hard-wired control unit
- Pipelining easier

3.1.2 CISC (Complex Instruction Set Computers)

Characteristics:

- More instructions
- Complicated instructions
- Many instruction formats
- Multi-cycle instructions
- Variable-length instructions
- Many types of memory addressing instructions
- More addressing modes
- Fewer registers

- Microprogrammed control unit
- Pipelining difficult

3.1.3 Pipelining

Definition: Instruction-level parallelism where the fetch-decode-execute cycle is separated into several stages.

Five Stages:

1. Instruction Fetch (IF)
2. Instruction Decode (ID)
3. Operand Fetch (OF)
4. Instruction Execution (IE)
5. Result Write Back (WB)

Advantages:

- Multiple instructions processed simultaneously
- Increased throughput
- More efficient use of processor components

Disadvantages:

- Interrupt handling complex
- Pipeline stalls possible

Interrupt Handling in Pipelines:

- When interrupt occurs, must clear pipeline
- Store current state of all instructions in pipeline
- Execute ISR
- Resume interrupted instructions

3.1.4 Parallel Processing Architectures

SISD (Single Instruction Single Data):

- Single processor
- Early computers
- No pipelining

SIMD (Single Instruction Multiple Data):

- Multiple processors
- Array processors
- Same instruction on multiple data points

MISD (Multiple Instruction Single Data):

- Multiple processors
- Same data, different instructions
- Used for sorting

MIMD (Multiple Instruction Multiple Data):

- Modern computers
- Each processor executes different instructions
- Most common parallel architecture

Massively Parallel Computers:

- Vast amounts of processing power
- Bus structure for multiple processors
- Network infrastructure for multiple hosts
- Used for complex mathematical problems

3.2 BOOLEAN ALGEBRA AND LOGIC CIRCUITS

3.2.1 Boolean Algebra Laws

Identity Law:

<TEXT>

$$A + 0 = A$$

$$A \cdot 1 = A$$

Null Law:

<TEXT>

$$A + 1 = 1$$

$$A \cdot 0 = 0$$

Idempotent Law:

<TEXT>

$$A + A = A$$

$$A \cdot A = A$$

Inverse Law:

<TEXT>

$$A + A' = 1$$

$$A \cdot A' = 0$$

Commutative Law:

<TEXT>

$$A + B = B + A$$

$$A \cdot B = B \cdot A$$

Associative Law:

<TEXT>

$$(A + B) + C = A + (B + C)$$

$$(A \cdot B) \cdot C = A \cdot (B \cdot C)$$

Distributive Law:

<TEXT>

$$A + B \cdot C = (A + B) \cdot (A + C)$$

$$A \cdot (B + C) = A \cdot B + A \cdot C$$

De Morgan's Laws:

<TEXT>

$$(A + B)' = A' \cdot B'$$

$$(A \cdot B)' = A' + B'$$

3.2.2 Karnaugh Maps (K-Maps)

Purpose: Simplify Boolean expressions and reduce number of gates needed.

Benefits:

- Minimises Boolean expressions
- Minimises logic gates
- More efficient circuits

Methodology:

1. Fill K-map from truth table
2. Group '1's in powers of 2 (1, 2, 4, 8)
3. Wrap around edges allowed
4. Within each group, only constant values remain
5. Write expression from groups

Example K-map (2 variables):

```
<TEXT>

AB'
0 1
A 0 0 1
1 1 1
```

3.2.3 Half Adders and Full Adders

Half Adder:

- Adds two bits
- Outputs sum and carry

A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0

A	B	Sum	Carry
1	1	0	1

Sum = A XOR B Carry = A AND B

Full Adder:

- Adds three bits (two operands + carry in)
- Outputs sum and carry out

3.2.4 Flip-Flops

SR Flip-Flop:

- Set-Reset flip-flop
- Stores 1 bit of data
- Two inputs: S (set) and R (reset)

JK Flip-Flop:

- Improvement over SR
- All input combinations valid
- Uses clock pulse for synchronization
- Toggle action when J=K=1

Purpose:

- Store bits of data
- Create memory from flip-flops
- Used in registers and counters

3.3 VIRTUAL MACHINES

3.3.1 Concept of Virtual Machines

Definition: Software that provides an exact copy of hardware. The process interacts with the software interface provided by the OS, which provides this copy. The OS kernel handles interaction

with actual host hardware.

Benefits:

- Multiple OS on single system
- Testing without affecting main system
- Legacy software on newer hardware
- Server consolidation

Drawbacks:

- Performance drop from native OS
- Time and effort for implementation

Examples:

- VMware
- VirtualBox
- Hyper-V

Revision #1

Created 2026-03-16 12:16:01 UTC by Samuel Lee

Updated 2026-03-16 12:16:15 UTC by Samuel Lee