Digital Electronics Notes

Mr. Rahimul Islam Mazumdar

Contents

1 UNIT I: Fundamentals of Digital Electronics **4**
- 1.1 Introduction . . . 4
  - 1.1.1 Continuous Signal . . . 4
  - 1.1.2 Digital Signal . . . 4
- 1.2 Boolean Algebra . . . 4
  - 1.2.1 Logic Levels . . . 4
- 1.3 Basic Boolean Operations . . . 4
  - 1.3.1 Truth Tables . . . 5
- 1.4 Boolean Laws . . . 5
  - 1.4.1 Commutative Law . . . 5
  - 1.4.2 Associative Law . . . 5
  - 1.4.3 Distributive Law . . . 6
  - 1.4.4 AND Laws . . . 6
  - 1.4.5 OR Laws . . . 6
  - 1.4.6 Complement Laws . . . 6
  - 1.4.7 Absorption Law . . . 6
  - 1.4.8 De Morgan's Theorems . . . 6
- 1.5 Operator Precedence . . . 6

2 Logic Gates **6**
- 2.1 Basic Gates . . . 7
- 2.2 Universal Gates . . . 7
- 2.3 Special Gates . . . 7

3 Representation of Boolean Expressions **7**
- 3.1 Sum of Products (SOP) . . . 7
- 3.2 Product of Sums (POS) . . . 7

4 Simplification of Boolean Expressions **8**

5 Number Systems **8**
5.1 Types of Number Systems . . . 8
5.1.1 Decimal Number System . . . 8
5.1.2 Binary Number System . . . 8
5.1.3 Octal Number System . . . 8
5.1.4 Hexadecimal Number System . . . 9
5.2 Number System Conversions . . . 9
5.2.1 Decimal to Binary . . . 9
5.2.2 Binary to Decimal . . . 9
5.2.3 Binary to Octal . . . 9
5.2.4 Binary to Hexadecimal . . . 9
5.3 Binary Arithmetic . . . 9
5.3.1 Binary Addition . . . 9
5.3.2 Binary Subtraction . . . 9
5.4 Complements . . . 10
5.4.1 1's Complement . . . 10
5.4.2 2's Complement . . . 10

6 Code Systems **10**
6.1 BCD Code . . . 10
6.2 Gray Code . . . 10
6.3 Excess-3 Code . . . 10

7 Combinational Logic Circuits **10**
7.1 Characteristics . . . 10
7.2 Half Adder . . . 10
7.3 Full Adder . . . 11
7.4 Multiplexer (MUX) . . . 11
7.5 Demultiplexer (DEMUX) . . . 11
7.6 Encoder . . . 11
7.7 Decoder . . . 11
7.8 Comparator . . . 11

8 Sequential Logic Circuits **12**
8.1 Characteristics of Sequential Circuits . . . 12
8.2 Block Diagram of Sequential Circuit . . . 12

9 Memory Elements **12**
9.1 Latch . . . 12
9.1.1 SR Latch . . . 12

10 Flip-Flops **13**
10.1 SR Flip-Flop 13
10.2 JK Flip-Flop 13
10.3 D Flip-Flop 13
10.4 T Flip-Flop 13

11 Registers **13**
11.1 Types of Registers 13

12 Counters **14**
12.1 Types of Counters 14

13 Finite State Machines (FSM) **14**

14 Moore and Mealy Models **14**
14.1 Moore Machine 14
14.1.1 Characteristics 14
14.1.2 Block Representation 15
14.2 Mealy Machine 15
14.2.1 Characteristics 15
14.3 Difference Between Moore and Mealy Machine 15

1 UNIT I: Fundamentals of Digital Electronics

1.1 Introduction

Digital electronics deals with digital systems that process information in the form of binary numbers (0 and 1). Analog electronics, on the other hand, processes information using continuous signals.

1.1.1 Continuous Signal

A continuous signal is a function $f(t)$ whose value is defined for all time t. It varies continuously with respect to time.

1.1.2 Digital Signal

A digital signal is a quantized discrete-time signal that takes only discrete values.

1.2 Boolean Algebra

Boolean algebra deals with logical operations on binary variables. The variables assume only two values: 0 and 1.

Boolean algebra was introduced by George Boole in 1847 and forms the foundation of digital electronic systems.

1.2.1 Logic Levels

Positive Logic

- Logic 0 = False, 0V, OFF
- Logic 1 = True, +5V, ON

Negative Logic

- Logic 0 = True, +5V
- Logic 1 = False, 0V

1.3 Basic Boolean Operations

- Complement (NOT)
- OR operation
- AND operation

$$Y = \overline{A}$$
$$Y = A + B$$
$$Y = A \cdot B$$

1.3.1 Truth Tables

NOT Operation

A	$Y = \overline{A}$
0	1
1	0

OR Operation

A	B	$Y = A + B$
0	0	0
0	1	1
1	0	1
1	1	1

AND Operation

A	B	$Y = A \cdot B$
0	0	0
0	1	0
1	0	0
1	1	1

1.4 Boolean Laws

1.4.1 Commutative Law

$$A + B = B + A$$
$$AB = BA$$

1.4.2 Associative Law

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

1.4.3 Distributive Law

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

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

1.4.4 AND Laws

$$A \cdot 0 = 0, \quad A \cdot 1 = A, \quad A \cdot A = A, \quad A\overline{A} = 0$$

1.4.5 OR Laws

$$A + 0 = A, \quad A + 1 = 1, \quad A + A = A, \quad A + \overline{A} = 1$$

1.4.6 Complement Laws

$$\overline{0} = 1, \quad \overline{1} = 0, \quad \overline{\overline{A}} = A$$

1.4.7 Absorption Law

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

$$A + AB = A$$

1.4.8 De Morgan's Theorems

$$\overline{A + B} = \overline{A} \cdot \overline{B}$$

$$\overline{AB} = \overline{A} + \overline{B}$$

1.5 Operator Precedence

Operator	Priority
Parenthesis	1
NOT	2
AND	3
OR	4

2 Logic Gates

Logic gates are the basic building blocks of digital circuits.

2.1 Basic Gates

- AND Gate
- OR Gate
- NOT Gate

2.2 Universal Gates

- NAND Gate
- NOR Gate

2.3 Special Gates

- EX-OR Gate
- EX-NOR Gate

3 Representation of Boolean Expressions

Boolean expressions can be represented in two forms:

- Sum of Products (SOP)
- Product of Sums (POS)

3.1 Sum of Products (SOP)

In SOP form, product terms are ORed together.

Example:

$$Y = \overline{AB}C + \overline{A}BC + AB\overline{C}$$

3.2 Product of Sums (POS)

In POS form, sum terms are ANDed together.

Example:

$$Y = (A + B + C)(A + \overline{B} + C)$$

4 Simplification of Boolean Expressions

Boolean expressions can be simplified using Boolean algebra laws to reduce hardware complexity.

Example:

$$Y = BC + BC' + BA$$

$$Y = B(C + C') + BA = B + BA = B$$

5 Number Systems

A number system is a method of representing numbers using a set of symbols and rules.

5.1 Types of Number Systems

- Decimal Number System (Base 10)
- Binary Number System (Base 2)
- Octal Number System (Base 8)
- Hexadecimal Number System (Base 16)

5.1.1 Decimal Number System

The decimal system uses ten symbols:

$$0, 1, 2, 3, 4, 5, 6, 7, 8, 9$$

Example:

$$(345)_{10} = 3 \times 10^2 + 4 \times 10^1 + 5 \times 10^0$$

5.1.2 Binary Number System

Binary system uses only two digits:

$$0 \text{ and } 1$$

Example:

$$(1011)_2 = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$$

5.1.3 Octal Number System

Uses digits from 0 to 7.

$$(725)_8 = 7 \times 8^2 + 2 \times 8^1 + 5 \times 8^0$$

5.1.4 Hexadecimal Number System

Uses digits 0–9 and letters A–F.

$$A = 10,\ B = 11,\ C = 12,\ D = 13,\ E = 14,\ F = 15$$

5.2 Number System Conversions

5.2.1 Decimal to Binary

Divide the decimal number repeatedly by 2 and record remainders.

5.2.2 Binary to Decimal

Multiply each bit with corresponding power of 2 and sum the results.

5.2.3 Binary to Octal

Group binary digits in sets of three from right.

5.2.4 Binary to Hexadecimal

Group binary digits in sets of four from right.

5.3 Binary Arithmetic

5.3.1 Binary Addition

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

5.3.2 Binary Subtraction

Performed using borrow method or 2's complement.

5.4 Complements

5.4.1 1's Complement

Obtained by changing 0s to 1s and 1s to 0s.

5.4.2 2's Complement

Obtained by adding 1 to the 1's complement.

6 Code Systems

6.1 BCD Code

Binary Coded Decimal represents each decimal digit using 4-bit binary equivalent.

Example:

$$(59)_{10} = 0101\ 1001_{BCD}$$

6.2 Gray Code

In Gray code only one bit changes between consecutive numbers.

6.3 Excess-3 Code

Obtained by adding 3 to the decimal number and converting to binary.

7 Combinational Logic Circuits

A combinational logic circuit is one whose output depends only on the present input values.

7.1 Characteristics

- No memory element
- Output depends only on current inputs
- No feedback path

7.2 Half Adder

A half adder performs addition of two single-bit binary numbers.

$$\text{Sum} = A \oplus B$$

$$\text{Carry} = AB$$

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

7.3 Full Adder

A full adder adds three bits: A, B, and carry input C_{in}.

$$\text{Sum} = A \oplus B \oplus C_{in}$$

$$C_{out} = AB + BC_{in} + AC_{in}$$

7.4 Multiplexer (MUX)

A multiplexer selects one input out of many inputs based on selection lines.

Example: 4-to-1 MUX

$$Y = \overline{S_1}\,\overline{S_0} I_0 + \overline{S_1} S_0 I_1 + S_1 \overline{S_0} I_2 + S_1 S_0 I_3$$

7.5 Demultiplexer (DEMUX)

A demultiplexer distributes one input signal to multiple outputs based on select lines.

7.6 Encoder

An encoder converts 2^n input lines into n output lines.

Example: 8-to-3 Encoder.

7.7 Decoder

A decoder converts binary information from n input lines to a maximum of 2^n output lines.

Example: 3-to-8 Decoder.

7.8 Comparator

A magnitude comparator compares two binary numbers and determines whether one is greater, equal, or smaller.

8 Sequential Logic Circuits

Sequential logic circuits are digital circuits whose output depends on both present inputs and past history (previous outputs). These circuits contain memory elements.

8.1 Characteristics of Sequential Circuits

- Output depends on present input and previous state.
- Contains memory elements.
- Feedback path exists.
- Requires clock signal for synchronization.

8.2 Block Diagram of Sequential Circuit

A sequential circuit consists of:

- Combinational logic
- Memory elements (Flip-flops)
- Clock signal

9 Memory Elements

Memory elements are used to store binary information.

9.1 Latch

A latch is a level-triggered device whose output changes whenever inputs change.

9.1.1 SR Latch

The SR latch has two inputs:

- S (Set)
- R (Reset)

S	R	Q_{next}	Operation
0	0	No Change	Hold
0	1	0	Reset
1	0	1	Set
1	1	Invalid	Forbidden

10 Flip-Flops

Flip-flops are edge-triggered memory elements used in synchronous sequential circuits.

10.1 SR Flip-Flop

Similar to SR latch but controlled by clock signal.

10.2 JK Flip-Flop

The JK flip-flop eliminates invalid condition of SR flip-flop.

J	K	Q_{next}	Operation
0	0	Q	No Change
0	1	0	Reset
1	0	1	Set
1	1	$\overline{Q}$	Toggle

10.3 D Flip-Flop

The D flip-flop stores one bit of data.

$$Q_{next} = D$$

10.4 T Flip-Flop

The T flip-flop toggles its state when input T is high.

$$Q_{next} = T \oplus Q$$

11 Registers

A register is a group of flip-flops used to store binary data.

11.1 Types of Registers

- Serial-In Serial-Out (SISO)
- Serial-In Parallel-Out (SIPO)
- Parallel-In Serial-Out (PISO)
- Parallel-In Parallel-Out (PIPO)

12 Counters

A counter is a sequential circuit that counts clock pulses.

12.1 Types of Counters

- Asynchronous (Ripple) Counter
- Synchronous Counter
- Up Counter
- Down Counter
- Up-Down Counter

13 Finite State Machines (FSM)

Finite State Machines are sequential circuits whose outputs depend on states and inputs.

FSM consists of:

- Finite number of states
- State transitions
- Output function

14 Moore and Mealy Models

Finite State Machines are classified into two types:

14.1 Moore Machine

In Moore machine, output depends only on present state.

$$Output = f(State)$$

14.1.1 Characteristics

- Output changes only at clock edge.
- More stable output.
- Requires more states compared to Mealy machine.

14.1.2 Block Representation

- Input → Combinational Logic
- State Register
- Output depends only on state

14.2 Mealy Machine

In Mealy machine, output depends on present state and present input.

$$Output = f(State, Input)$$

14.2.1 Characteristics

- Output changes immediately with input.
- Requires fewer states.
- Faster response compared to Moore machine.

14.3 Difference Between Moore and Mealy Machine

Feature	Moore Machine	Mealy Machine
Output depends on	State only	State and Input
Output change	At clock edge	Immediate
Number of states	More	Less
Speed	Slower	Faster
Stability	More stable	Less stable