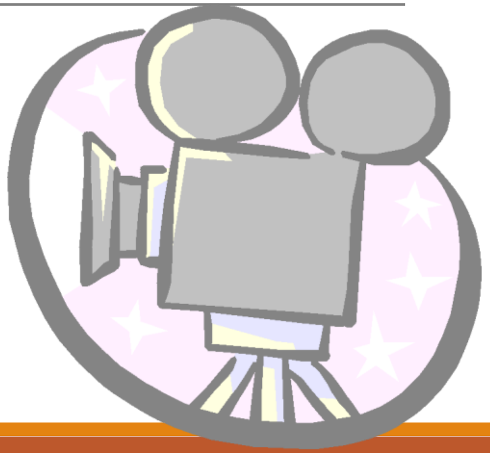
Digital Logic and Circuit
Paper Code: CS-102



Topics of Course

- **Number System**
- **Boolean algebra and Logic Gates**
- **Simplification of Boolean Functions**
- **Combinational Circuits**
- **Sequential Circuits**

Suggested Readings

- M. Morris Mano, Digital Logic and Computer Design, PHI.

What do you mean by word “Digital”?

What is this?

1000001100111010010111001001101010010000001
1000001100111010010111001001101010010000001
1000001100111010010111001001101010010000001
1000001100111010010111001001101010010000001
1000001100111010010111001001101010010000001
1000001100111010010111001001101010010000001
1000001100111010010111001001101010010000001

ASCII Table

Dec	Hex	Char	Action (*if non-printing*)	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
0	0	NUL	(null)	32	20	Space	64	40	@	96	60	`
1	1	SOH	(start of heading)	33	21	!	65	41	A	97	61	a
2	2	STX	(start of text)	34	22	"	66	42	B	98	62	b
3	3	ETX	(end of text)	35	23	#	67	43	C	99	63	c
4	4	EOT	(end of transmission)	36	24	$	68	44	D	100	64	d
5	5	ENQ	(enquiry)	37	25	%	69	45	E	101	65	e
6	6	ACK	(acknowledge)	38	26	&	70	46	F	102	66	f
7	7	BEL	(bell)	39	27	'	71	47	G	103	67	g
8	8	BS	(backspace)	40	28	(	72	48	H	104	68	h
9	9	TAB	(horizontal tab)	41	29	)	73	49	I	105	69	i
10	A	LF	(NL line feed, new line)	42	2A	*	74	4A	J	106	6A	j
11	B	VT	(vertical tab)	43	2B	+	75	4B	K	107	6B	k
12	C	FF	(NP form feed, new page)	44	2C	,	76	4C	L	108	6C	l
13	D	CR	(carriage return)	45	2D	-	77	4D	M	109	6D	m
14	E	SO	(shift out)	46	2E	.	78	4E	N	110	6E	n
15	F	SI	(shift in)	47	2F	/	79	4F	O	111	6F	o
16	10	DLE	(data link escape)	48	30	0	80	50	P	112	70	p
17	11	DC1	(device control 1)	49	31	1	81	51	Q	113	71	q
18	12	DC2	(device control 2)	50	32	2	82	52	R	114	72	r
19	13	DC3	(device control 3)	51	33	3	83	53	S	115	73	s
20	14	DC4	(device control 4)	52	34	4	84	54	T	116	74	t
21	15	NAK	(negative acknowledge)	53	35	5	85	55	U	117	75	u
22	16	SYN	(synchronous idle)	54	36	6	86	56	V	118	76	v
23	17	ETB	(end of trans. block)	55	37	7	87	57	W	119	77	w
24	18	CAN	(cancel)	56	38	8	88	58	X	120	78	x
25	19	EM	(end of medium)	57	39	9	89	59	Y	121	79	y
26	1A	SUB	(substitute)	58	3A	:	90	5A	Z	122	7A	z
27	1B	ESC	(escape)	59	3B	;	91	5B	[	123	7B	{
28	1C	FS	(file separator)	60	3C	<	92	5C	\	124	7C	|
29	1D	GS	(group separator)	61	3D	=	93	5D	]	125	7D	}
30	1E	RS	(record separator)	62	3E	>	94	5E	^	126	7E	~
31	1F	US	(unit separator)	63	3F	?	95	5F	_	127	7F	DEL

American standard code for information exchange

128 characters are represented by different decimal numbers

As $2^7=128$

So 7 bits are required for representing ASCII code

Decimal to Binary & Binary to Decimal Conversion

2	65	1	LSB
2	32	0	
2	16	0	
2	8	0	
2	4	0	
2	2	0	
	1	1	MSB

$(65)_{10}= (1000001)_2$

Binary to decimal

1000001

$1\times2^6+0\times2^5+0\times2^4+0\times2^3+0\times2^2+0\times2^1+1\times2^0$

64+0+0+0+0+0+1

65

ANKITA

ASCII: A: 65 N: 78 K: 75 I:73 T:84 A:65

A	N	K	I	T	A
65	78	75	73	84	65

1000001 1001110 1001011 1001001 1010100 1000001

100000110011101001011100100110101001000001

What is this?

1000000110011101001011100100110101001000001
1000000110011101001011100100110101001000001
1000000110011101001011100100110101001000001
1000000110011101001011100100110101001000001
1000000110011101001011100100110101001000001
1000000110011101001011100100110101001000001
1000000110011101001011100100110101001000001

1000001	1001110	1001011	1001001	1010100	1000001
1000001	1001110	1001011	1001001	1010100	1000001
1000001	1001110	1001011	1001001	1010100	1000001
1000001	1001110	1001011	1001001	1010100	1000001
1000001	1001110	1001011	1001001	1010100	1000001
1000001	1001110	1001011	1001001	1010100	1000001
1000001	1001110	1001011	1001001	1010100	1000001

(a)

65	78	75	73	84	65
65	78	75	73	84	65
65	78	75	73	84	65
65	78	75	73	84	65
65	78	75	73	84	65
65	78	75	73	84	65
65	78	75	73	84	65

(b)

A	N	K	I	T	A
A	N	K	I	T	A
A	N	K	I	T	A
A	N	K	I	T	A
A	N	K	I	T	A
A	N	K	I	T	A
A	N	K	I	T	A

(c)

Number system: Weighted and un-weighted codes

Decimal (base 10) : weights in power of 10.

- Decimal digits : 0, 1,2,3,4,5,6,7,8,9

Binary (base 2) : weights in power of 2.

- Binary digits (bits) : 0, 1

Octal (base 8) : weights in power of 8.

- Octal digits : 0,1,2,3,4,5,6,7

Hexadecimal (base 16) : weights in power of 16

- Hexadecimal digits : 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F

Un-weighted codes

Binary codes for the decimal digits

Decimal digit	(BCD) 8421	Excess-3
0	0000	0011
1	0001	0100
2	0010	0101
3	0011	0110
4	0100	0111
5	0101	1000
6	0110	1001
7	0111	1010
8	1000	1011
9	1001	1100

Binary Logic

Binary logic consists of binary variables and logical operations.

The variable may be designated by letters of alphabet such as A, B

1. AND: This operation is represented by a dot or by the absence of an operator. For example, $x*y = z$ or $xy = z$ is read “x AND y is equal to z.”

2. OR: This operation is represented by a plus sign. For example, $x + y = z$ is read “x OR y is equal to z,”
3. NOT: This operation is represented by a prime (sometimes by a bar). For example, $x' = z$ (or $\bar{x} = z$) is read “not x is equal to z”

Truth Table of Logical operations

Truth Tables of Logical Operations

AND

x	y	$x \cdot y$
0	0	0
0	1	0
1	0	0
1	1	1

OR

x	y	$x + y$
0	0	0
0	1	1
1	0	1
1	1	1

NOT

x	x'
0	1
1	0

Truth Table of Logical operations

Truth Tables of Logical Operations

AND

x	y	$x \cdot y$
0	0	0
0	1	0
1	0	0
1	1	1

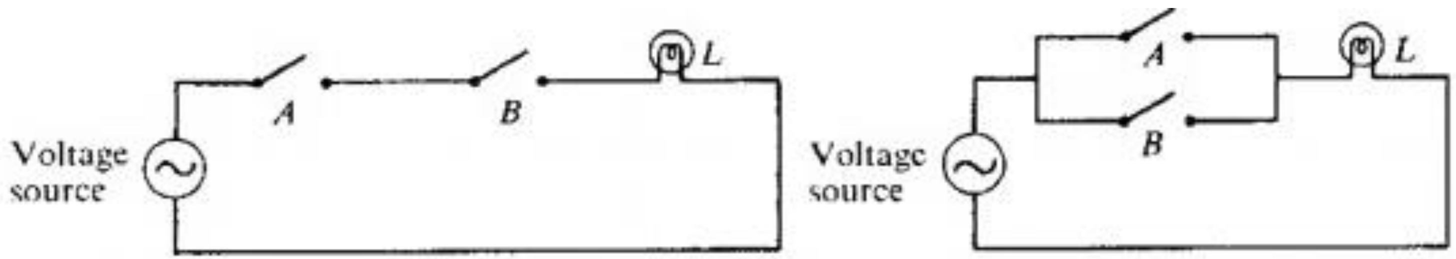
OR

x	y	$x + y$
0	0	0
0	1	1
1	0	1
1	1	1

NOT

x	x'
0	1
1	0

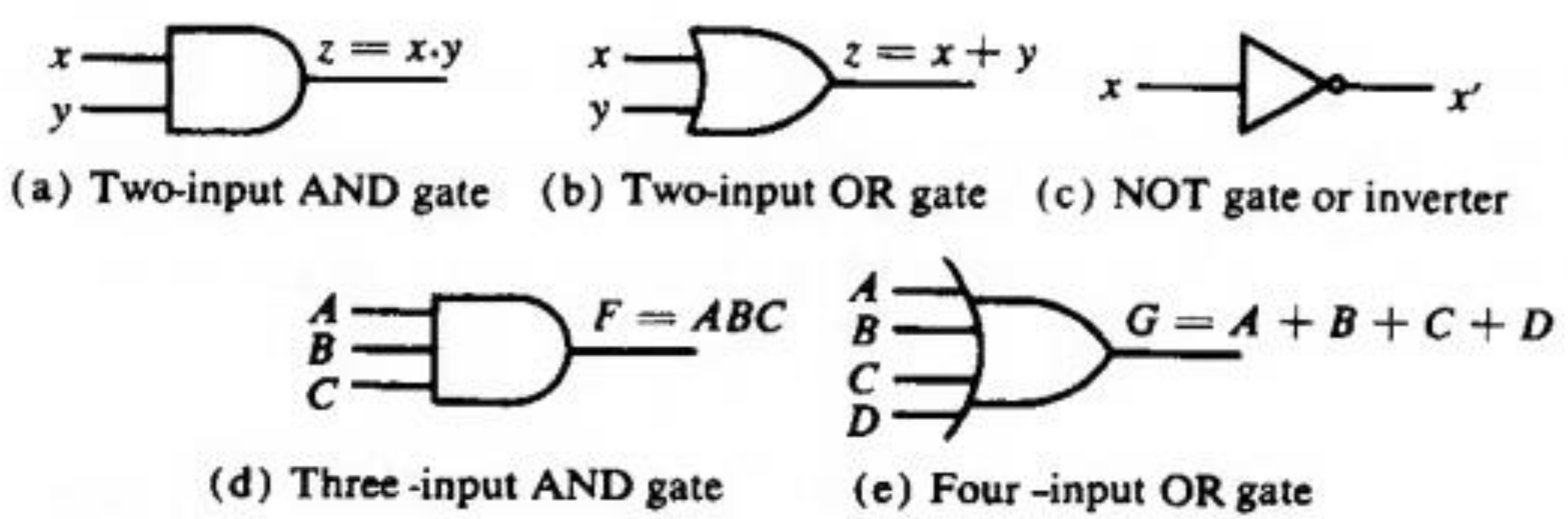
Switching circuits that demonstrate binary logic



(a) Switches in series – logic AND

(b) Switches in parallel – logic OR

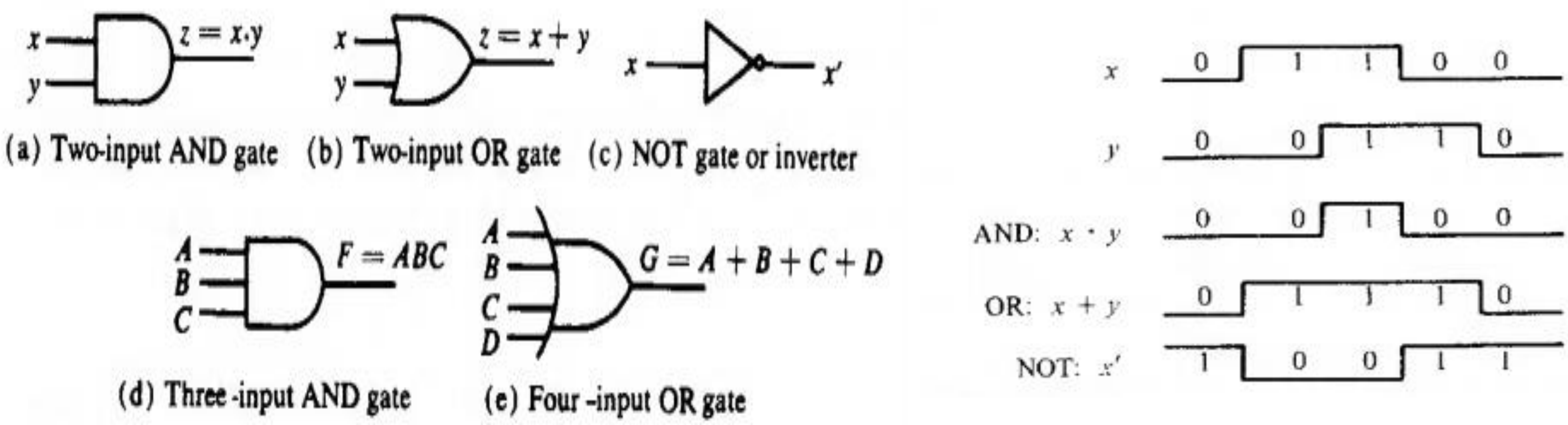
Logic gates



(a) Two-input AND gate (b) Two-input OR gate (c) NOT gate or inverter

(d) Three -input AND gate (e) Four -input OR gate

Input output signals for Logic gates



Boolean Algebra

Boolean algebra may be defined with a set of elements, a set of operations and a number of unproved axioms or postulates.

If S is a set, x and y are certain objects then $x \in S$ denotes that x is an element of S.

The postulates of a mathematical system form the basic assumptions from which it is possible to deduce the rules, theorems, and properties of the system.

The most common postulates used to formulate various algebraic structures are:

1. Closure . A set S is closed with respect to a binary operator if, for every pair of elements of S, the binary operator specifies a rule for obtaining a unique element of S.

For example, the set of natural numbers N = {l,2,3,4,...}is closed with respect to the binary operator plus (T) by the rules of arithmetic addition

The set of natural numbers is not closed with respect to the binary operator minus (-) by the rules of arithmetic subtraction because 2 - 3 =- I and 2, 3 ∈ N, while (-1) ∉ N.

2. Associative law. A binary operator * on a set S is said to be associative whenever

$$(x * y) * z = x * (y * z) \qquad \text{for all } x, y, z \in S.$$

3. Commutative law. A binary operator * on a set S is said to be commutative whenever

$$x * y = y * x \qquad \text{for all } x, y \in S$$

4. Identity element

A set S is said to have an identity element with respect to a binary operation * on S if there exists an element $e \in S$ with the property

$$e*x=x*e=x \qquad \text{for every } x \in S$$

Example.

The element 0 is an identity element with respect to operation + on the set of integers I = {..., -3, -2, -1, 0, 1, 2, 3, ... } since

$$x + 0 = 0 + x = x \qquad \text{for any } x \in I$$

5. Inverse. A set S having the identity element e with respect to a binary operator * is said to have an inverse whenever, for every $x \in S$, there exists an element $y \in S$ such that

$$x * y = e$$

Example: In the set of integers I with $e = 0$, the inverse of an element a is $(-a)$ since $a + (-a) = 0$.

6. Distributive law. If * and • are two binary operators on a set S, * is said to be distributive over whenever x * (y • z) = (x * y) • - (x * z)

AXIOMATIC DEFINITION OF BOOLEAN ALGEBRA

In 1854 George Boole introduced a systematic treatment of logic and developed for this purpose an algebraic system now called Boolean algebra.

In 1938 C. E. Shannon introduced a two-valued Boolean algebra called switching algebra, in which he demonstrated that the properties of bistable electrical switching circuits can be represented by this algebra.

For the formal definition of Boolean algebra, we shall employ the postulates formulated by E. V. Huntington in 1904 .

Boolean algebra is an algebraic structure defined on a set of elements B together with two binary operators + and • provided the following (Huntington) postulates are satisfied:

1. (a) Closure with respect to the operator +.
   (b) Closure with respect to the operator •.

2. (a) An identity element with respect to +, designated by 0: $x + 0 = 0 + x = x$.
   (b) An identity element with respect to •, designated by 1: $x \bullet 1 = 1 \bullet x = x$.

3. (a) Commutative with respect to + : $x + y = y + x$.
   (b) Commutative with respect to : $x \bullet y = y \bullet x$.

4 . (a) • is distributive over +: $x \bullet (y + z) = \{x \bullet y) + \{x \bullet z)$.
   (b) + is distributive over •: $x + (y\ z) = (x + y) \bullet (x + z)$.

5. For every element $x \in B$, there exists an element $x' \in B$ (called the complement of x) such that (a) $x + x' = 1$ and (b) $x.x' = 0$.

6. There exists at least two elements $x, y \in B$ such that $x \neq y$.

Comparing Boolean algebra with arithmetic and ordinary algebra, we note the following differences:

1. Huntington postulates do not include the associative law. However, this law holds for Boolean algebra and can be derived (for both operators) from the other postulates.

2. The distributive law of + over •, i.e., $x + (y \bullet z) = (x + y) \bullet (x + z)$, is valid for Boolean algebra, but not for ordinary algebra.

3. Boolean algebra does not have additive or multiplicative inverses; therefore, there are no subtraction or division operations.

4. Postulate 5 defines an operator called complement that is not available in ordinary algebra.

5. Ordinary algebra deals with the real numbers, which constitute an infinite set of elements. Boolean algebra, B is defined as a set with only two elements, 0 and 1

Basic theorem & properties of Boolean Algebra

Duality: The Huntington postulates have been listed in pairs and designated by part (a) and part (b). One part may be obtained from the other if the binary operators and the identity elements are interchanged.

This important property of Boolean algebra is called the **duality principle**.

It states that every algebraic expression deducible from the postulates of Boolean algebra remains valid if the operators and identity elements are interchanged.

Example: B={0, 1} e={0, 1}

$$x+0=0+x=x$$

$$x.1=1.x=x$$

Basic theorem & properties of Boolean Algebra

Postulates and Theorems of Boolean Algebra

Postulate 2	(a) $x + 0 = x$	(b) $x \cdot 1 = x$
Postulate 5	(a) $x + x' = 1$	(b) $x \cdot x' = 0$
Theorem 1	(a) $x + x = x$	(b) $x \cdot x = x$
Theorem 2	(a) $x + 1 = 1$	(b) $x \cdot 0 = 0$
Theorem 3, involution	$(x')' = x$	
Postulate 3, commutative	(a) $x + y = y + x$	(b) $xy = yx$
Theorem 4, associative	(a) $x + (y + z) = (x + y) + z$	(b) $x(yz) = (xy)z$
Postulate 4, distributive	(a) $x(y + z) = xy + xz$	(b) $x + yz = (x + y)(x + z)$
Theorem 5, DeMorgan	(a) $(x + y)' = x'y'$	(b) $(xy)' = x' + y'$
Theorem 6, absorption	(a) $x + xy = x$	(b) $x(x + y) = x$

THEOREM 1(a): $x + x = x$.

$x + x = (x + x) \cdot 1$	by postulate:	2(b)
$= (x + x)(x + x')$		5(a)
$= x + xx'$		4(b)
$= x + 0$		5(b)
$= x$		2(a)

THEOREM 1(b): $x \cdot x = x$.

$x \cdot x = xx + 0$	by postulate:	2(a)
$= xx + xx'$		5(b)
$= x(x + x')$		4(a)
$= x \cdot 1$		5(a)
$= x$		2(b)

Thank you