Number Systems

Positional Weight

$$\begin{array}{cccccccccc} r^{n-1} & r^{n-2} & & r^{1} & r^{0} & & r^{-1} & r^{-2} & & r^{-m} \\ \uparrow & \uparrow & & \uparrow & \uparrow & & \uparrow & \uparrow & & \uparrow \end{array}$$

$$(\underbrace{d_{n-1}\, d_{n-2} \cdots d_1\, d_0}_{\text{Integer Part}} \underset{\text{decimal Point}}{\uparrow\!\!.}\; \underbrace{d_{-1}\, d_{-2} \cdots d_{-m}}_{\text{fractional Part}})_r$$

~~fractional~~ Integer Part

decimal Point

$r \rightarrow$ Radix or Base of Number system

$n \rightarrow$ no. of digits in integer part

$m \rightarrow$ no. of digits in fractional part

- Value of Number = Sum of (Each digit multiplied by the corresponding power of the radix i.e. positional weight)

$$(N)_r = (d_{n-1} \times r^{n-1}) + (d_{n-2} \times r^{n-2}) + \cdots + (d_1 \times r^1) + (d_0 \times r^0) + (d_{-1} \times r^{-1}) + \cdots + (d_{-m} \times r^{-m})$$

where,

- range of integer $d \rightarrow \quad 0 \le d_i \le (r-1)$

(i) Decimal Number System

radix $= 10$

range of valid integers $= 0 \rightarrow 9$

Total values (10)

Ex. $(145.86)_{10}$

(ii) Binary no. System

radix $= 2$

range of valid integers $= 0 \rightarrow 1$

Total values (2) Ex. $(1011.101)_2$

Scanned with CamScanner

(iii) Octal Number System

radix = 8

range of valid integers = 0 → 7

Total values = 8

Ex. $(367.721)_8$

(iv) Hexadecimal Number System

radix = 16

range of valid integers = 0 → 15
(0 to 9, A, B, C, D, E, F)

Total Values (16)

Ex. $(621A.82B)_{16}$

Number-Base Conversions

(i) Decimal to Binary/Octal/Hexadecimal {r}

for Integer Part → Successive Division Method by r ↑(remainders)

for fractional Part → Successive Multiplication by r Method ↓(Integer Part)

Ex. (i) $(57.825)_{10} \longrightarrow (111001.1101)_2 \longrightarrow (\quad)_8 \longrightarrow (\quad)_{16}$

(ii) $(125.201)_{10} \longrightarrow (175.14672)_8 \longrightarrow (7D.3374B)_{16}$

(ii) Binary/Octal/Hexadecimal (r) to Decimal

→ Multiply all coefficients/digits with their respective/positional weights and add them all.

Ex (i) $(1011.101)_2 \rightarrow (\quad)_{10}$

(ii) $(367.721)_8 \rightarrow (\quad)_{10}$

(iii) $(621A.825)_{16} \rightarrow (\quad)_{10}$

(iii) Binary to Octal / Hexadecimal

Octal: Make groups of 3-bits and write equivalent Octal code

Hexadecimal: Make groups of 4-bits and write equivalent Hex-code.

(iv) Octal / Hexadecimal to Binary

Consider every digit and replace it with its 3-bit/4-bit binary equivalent.

- Ex (i) $(746.52)_8 \rightarrow (\quad)_2$

(ii) $(17E.BF0D)_{16} \rightarrow (\quad)_2$

(iii) $(1110100.0100111)_2 \rightarrow (\quad)_8$

(iv) $(1011001110.011111)_2 \rightarrow (\quad)_{16}$

Binary Arithmatic

Addition	Sum	Subtraction	
0+0	= 0	0−0	= 0
1+0	= 1	1−0	= 1
0+1	= 1	1−1	= 0
1+1	= 0 and Carry 1	0−1	= 1 with Borrow=1
1+1+1	= 1 and Carry 1		

Scanned with CamScanner

Ex. (i) $(1101.101)_2 + (111.011)_2 =$

(ii) $(1010.01)_2 - (111.111)_2 = (0010.011)_2$

complements of Numbers

(i) r's complement

(ii) (r-1)'s complement

formulae and short-cut method

Ex. (i) 1101100

(ii) 1101100.1011

] find 1's complement

(iii) 0.1011

(iv) 1101100.1011

] find 2's complement

(v) $(37.562)_{10}$

(vi) $(23567)_{10}$

] find 9's complement

(vii) $(1056.074)_{10}$

(viii) $(0.8642)_{10}$

] find 10's complement

(ix) $(407.270)_8$

(x) $(0156.0037)_8$

] find 7's complement

(xi) $(346)_8$ find 8's complement

(xii) $(83D.9F)_{16}$ find 15's complement

(xiii) $(A8C)_{16}$ find 16's complement

*find 10's complement of $(935)_{11}$

Scanned with CamScanner

COMPLEMENTS

(i) r's Complement

for a positive number N in base r with an integer part of n digits, r's complement can be defined as

$$\left.\begin{array}{lll} & r^n - N & \text{for } N \neq 0 \\ \text{and} & 0 & \text{for } N = 0 \end{array}\right\} \begin{array}{l} r > 1 \\ r \neq 1 \end{array}$$

Ex. 10's complement
2's complement
16's complement
8's complement

for example:

(i) 10's complement of $(52520)_{10} \rightarrow 47480$

(ii) 10's complement of $(0.3267)_{10} \rightarrow 0.6733$

(iii) 10's complement of $(25.639)_{10} \rightarrow 74.361$

(iv) 2's complement of $(101100)_2$

$= (2^6)_{10} - (101100)_2$

$= (1000000 - 101100)_2 = 010100$

(v) 2's complement of $(0.0110)_2 = (1 - 0.0110) = 0.1010$

(ii) $(r-1)$'s Complement (Diminished-r Complement)

for a positive number N in base r with an integer part of n digits and a fraction part of m digits, $(r-1)$'s complement can be defined as

$$\boxed{r^n - r^{-m} - N}$$

such as, 9's complement
7's complement
1's complement
15's complement

Examples:

(i) 9's complement of $(52520)_{10} \rightarrow 47479$

(ii) 9's complement of $(0.3267)_{10} \rightarrow (0.6732)$

(iii) 9's complement of $(25.639)_{10} \rightarrow 74.360$

(iv) 1's complement of $(101100)_2$

$$= (2^6 - 1)_{10} - (101100)_2$$

$$= (111111 - 101100)_2 = 010011$$

(v) 1's complement of $(0.0110)_2$ is

$$(1 - 2^{-4})_{10} - (0.0110)_2 = 0.1001$$

Scanned with CamScanner

r's Complement

→ positive no. N in Base r with integer part of n-digits

$$= r^n - N \quad \text{for } N \neq 0$$

$$= 0 \quad \text{for } N = 0$$

Short-Cut Method

10's Complement →
- Leave all Least significant zeros unchanged.
- Subtract first non-zero least significant digit from 10
- Subtract all other higher significant digits from 9.

(r−1)'s Complement

→ Positive number N in base r with an integer part of n digits and a fraction part of m digits, the (r−1)'s complement of N is given by

$$r^n - r^{-m} - N.$$

Short-Cut Method

9's Complement → . Subtract all digits from 9.

~~Left here~~

$$\boxed{* \text{ r's Complement} = (r-1)\text{'s Complement} + r^{-m} \text{ to the least significant digit}}$$

* r's Complement = (r−1)'s Complement + 1 (to the LS Digit of (r−1)'s complement)

Scanned with CamScanner

SHORT-CUT METHOD

(i) r's Complement

r's complement of a number in base-r can be formed by leaving all least-significant zeros unchanged, subtracting the first non-zero least significant digit from r and then subtracting all other higher significant digits from (r-1).

10's Complement

- Leave all least significant zeros as it is
- Subtract first non-zero least significant digit from 10.
- Subtract rest of most significant digits from 9.

2's Complement

- Leave all least significant zeros as it is
- Leave first 1 unchanged
- Replace all other bits by its complement i.e. replace 1 by 0 and 0 by 1.

Scanned with CamScanner

(ii) $(r-1)$'s Complement

$(r-1)$'s complement of ~~a~~ a number in base-r can be formed by subtracting all digits from $(r-1)$.

9's Complement

- Subtract every digit from 9

1's Complement

- Subtract every bit from 1
- In other words, Replace all 1's by 0's and all 0's by 1's.

NOTE: r's complement of a number can be obtained from the $(r-1)$'s complement by adding r^{-m} to the least significant digit.

E.g. 2's complement of a binary number can be obtained by taking its 1's complement and then adding a 1 to the least significant bit.

Scanned with CamScanner

NUMBER REPRESENTATION IN BINARY

1.) Sign-Magnitude Representation
2.) 1's Complement "
3.) 2's complement "

1.) Sign-Magnitude Representation

In 8-bit representation, the MSB i.e. the 8^{th} bit represent the sign and remaining 7-bits represent the magnitude.

Range of Nos. represented in Sign-Magnitude representation $\Big\}$ $-(2^{n-1}-1)$ to $+(2^{n-1}-1)$
i.e. -127 to $+127$

2.) 1's Complement Representation

Steps :→ (i) Write the positive number in binary (8-bit) form.
(ii) Find the 1's complement of the binary number.

Range of Nos. represented in 1's complement representation $\Big\}$ $-(2^{n-1}-1)$ to $+(2^{n-1}-1)$
i.e. -127 to $+127$

3.) 2's Complement Representation

Steps :→ (i) Write the positive number in 8-bit binary form.
(ii) Find the 2's complement of the binary number.

Range of Nos. represented in 2's complement representation $\Big\}$ $-(2^{n-1})$ to $+(2^{n-1}-1)$
i.e. -128 to $+127$

Scanned with CamScanner

2	41	Rem.
2	20	1
2	10	0
2	5	0
2	2	1
	1	0

$(41)_{10} = (101001)_2$

Sign-Magnitude Representation

MSB → Sign Bit

$+ \rightarrow 0$

$- \rightarrow 1$

2	25	
2	12	1
2	6	0
2	3	0
	1	1

$(25)_{10} = (11001)_2$

$+25 = 0011001$

↑
~~Sign bit~~

for 8-bit representation

8^{th} bits → Sign

7-bits → Magnitude

- $(25)_{10} = (11001)_2$

① $= (0011001)_2 \longrightarrow$ Magnitude representation by 7 bits

Signed-Representation or Sign-Magnitude Representation

$+25 = \underbrace{0}_{\text{Sign bit}}\underbrace{0011001}_{\text{Magnitude}}$

- $-25 = \underbrace{1}_{\text{Sign bit}}\underbrace{0011001}_{\text{Magnitude}}$

② 1's complement representation

$(25)_{10} = (\underbrace{0011001}_{\text{Magnitude only}})_2$

$(+25)_{10} = (00011001)_2$

1's complement representation of +ve number is same as the sign-magnitude representati

$(-25)_{10} = (\underbrace{0011001}_{\text{Magnitude of } (25)_{10}}) \xrightarrow{\text{using 8-bits}} (00011001)$

↓ Take 1's Comp.

(11100110)

Scanned with CamScanner

3. 2's Complement Representation

$(+25)_{10}$ → will be same as the 8-bit sign-magnitude representation

$(-25)_{10} \rightarrow (\underline{00011001})_2$ → 8-bit sign-magnitude representation of $(+25)$

$\downarrow$ 2's complement

(11100111)

Scanned with CamScanner

Ex. $(+25)_{10}$ in Sign-Magnitude Repr.ⁿ → (00011001)
in 1's Complement → (00011001)
in 2's " → (00011001)
Same

$(-25)_{10}$ in Sign-Magnitude Repr.ⁿ → (10011001)
in 1's Complement → (11100110)
in 2's complement → (11100111)

SUBTRACTION

(i) Using r's Complement
(ii) Using (r-1)'s Complement

i) Subtraction using r's Complement

Say, we have two positive numbers M and N in base r and we have to find (M−N).

- **Steps**: (1.) Add the minuend M to the r's complement of the subtrahend N.

(2.) Check the result obtained in step 1 for an end-carry (carry from MSB)

(a) If an end-carry occurs, Discard it

(b) If an end-carry does not occur, take r's complement of the number obtained in step 1 and put a negative sign in front.

Scanned with CamScanner

Some Important Questions

(1.) $-46-25$ using 2's complement

$-46-25=-71$

$(46)_{10} = (00101110)_2$

$(25)_{10} = (00011001)_2$

2's Comp. representation of $(-46)_{10} \rightarrow 11010010$

" " " $(-25)_{10} \rightarrow 11100111$

Addition +

1) 10111001

10111001

Discard the carry

i.e. the answer is negative and is in 2's complement form

$\therefore$ Result = − 2's complement of (1011100l)

$= -(01000111) = -71.$

Scanned with CamScanner

(7) BCD Addition

Rule: Add corresponding BCD numbers according to their weights i.e. positional weight 0 with
" " 0
Positional weight 8 with
" " 8.
and so on.

If the BCD sum is less than or equal to 9 then it remain the valid BCD value.

But

If the BCD sum exceeds the value 9, then to make it a valid BCD code, we have to add $(6)_{10}$ or $(0110)_{BCD}$ in the corresponding numbers.

If after addition of $(0110)_{BCD}$ a carry is generated in 4^{th}, 8^{th}, 12^{th} or so on, bits, then it is transferred to the 5^{th}, 9^{th}, 13^{th} or so on respectively.

Ex. $(386)_{10} + (756)_{10}$

$(386)_{10}$	0011	1000	0110	BCD
$+ (756)_{10}$	+ 0111	0101	0110	BCD
	1010	1101	1100	Invalid BCD code
	+ 0110	0110	0110	Add $6_{10} = 0110_{BCD}$
0001	0001	0100	0010	$= (1142)_{10}$

new Digit

BCD Subtraction

Here we use the 1's complement method to perform the subtraction.
i.e. 1's complement of the subtrahend is added with the minuend. (1st Adder)

If ~~the~~ an end-Around carry (EAC) occurs, then the above result is passed to 2nd Adder and there the addition will follow the following algorithm.

If an end-around carry (EAC) does not occur, then the 1's complement of the ~~ab~~ above result will be passed to 2nd Adder which will follow the following algorithm further;

Decade Result	End-Around Carry (Sign of Total Result)	
	(+)EAC = 1	(−)EAC = 0
	Transfer true Result of adder 1	Transfer 1's complement of Adder 1
Cn=1	0000 added in Adder2	1010 added in adder2
Cn=0	1010 added in Adder 2	0000 added in adder 2 (Discard Carry in adder 2)

↓ Result will be +ve

↓ Result will be −ve

Scanned with CamScanner

Eg Add

$$\begin{array}{r} 96 \\ +29 \\ \hline \end{array}$$

$$\begin{array}{rrr} & 1001 & 0110 \\ & 0010 & 1001 \\ \hline & 1011 & 1111 \\ + & 0110 & 0110 \\ \hline 0001 & 0010 & 0101 \\ 1 & 2 & 5 \end{array}$$

Eg Perform the following decimal additions for use with the 8421 BCD code

(i) $386_{10} + 756_{10} = (1142)_{10}$

(ii) $123_{10} + 987_{10} = (1110)_{10}$

Direct Hex. Addition

$$\begin{array}{r} 8B \\ +9F \\ \hline 12B_H \end{array} \qquad \begin{array}{r} 36 \\ A7 \\ \hline DD_H \end{array}$$

BCD subtraction

Eg: $(68-31)_{10}$

$$\begin{array}{llll} 68 & 0110\ 1000 & & 0110\ 1000 \\ 31 & 0011\ 0001 & + & 1100\ 1110 \\ \hline & & 1 & 0011\ 0110 \\ & & & +1 \\ \hline & & & \underbrace{0011}_{3}\ \underbrace{0111}_{7} \end{array}$$

EAC and C2

C_1

Eg $(86 - 97)_{10}$ $97 = 1001\ 0111$
$= 0110\ 1000$

$$
\begin{array}{r}
1000\ 0110 \\
+\ 0110\ 1000 \\
\hline
1110\ 1110
\end{array}
$$

0 = -ve

Take 1's comp.

$\underbrace{0001}_{1}\ \underbrace{0001}_{1} = (11)_{10}$

Eg (i) $546 - 429 = 117$

(ii) $429 - 546 = -117$

→ (+) EAC = 1 , transfer true results of

→ (−) EAC = 0 , transfer 1's comp. of result

<u>Weighted Codes</u>

BCD Subtraction

1 One's Complement of the BCD subtrahend is entered into adder 1 to be added to the minuend.

2. If sign of the total result becomes positive i.e. the difference output is positive or EAC = 1 then the algorithm is

Decade Result	Sign of Total Result	
	(+) EAC = 1	(–) EAC = 0
	Transfer true results of adder 1	Transfer 1s comp. of result of adder 1
$C_n = 1$	0000 added in adder 2	1010 added in adder 2
$C_n = 0$	1010 added in adder 2	0000 added in adder 2 (Discard carry in adder 2)

Algorithm of BCD Subtraction

3 If the sign of total result is negative (– EAC = 0), then the comp. of true result of adder 1 is transferred to adder 2.

Q.1. Perform $(68 - 31)_{10}$ in BCD.

$(68)_{10} = (0110\ 1000)_{BCD}$

$(31)_{10} = (0011\ 0001)_{BCD}$

1's comp. of subtrahend 00110001

= 11001110

```
   0110  1000
+  1100  1110
-------------
(1) 0010 (1)0110
      +1      +1      Adder 1
-------------
   0011  0111         Transfer true result of Adder 1
-------------
   ↓↓↓↓  ↓↓↓↓
   0011  0111
+  0000  0000         Cn=1, 0000 is added to
-------------         each group.
   0011  0111
-------------
                      (37)10
```

(1) EAC (+) = 1, So the true result of adder 1 is transferred to adder 2.

(2) In adder 2, 0000 is added to the each 4-bit code (decade) when Cn = 1 i.e. carry generated is 1.

(3) Otherwise if Cn = 0, i.e. no carry then 1010 is added.

Scanned with CamScanner

② Perform $(86-97)_{10}$ using BCD subtraction

$$\begin{array}{rl} 86 & 1000\ 0110 \\ -97 & 1001\ 0111 \\ \hline -11 & \end{array}$$

1's comp. of subtrahend i.e. 97

1001 0111 = 0110 1000

$$\begin{array}{rl} & 1000\ 0110 \\ + & 0110\ 1000 \\ \hline & 1110\ 1110 \end{array}$$

EAC (-) = 0, So transfer the 1's comp. of result to adder 2

$$\begin{array}{rl} & 0001\ 0001 \\ + & 0000\ 0000 \\ \hline & 0001\ 0001 \\ \hline \end{array}$$

$C_n = 0$, so 0000 is added to each decade in adder 2

$(11)_{10}$

→ As no carry is generated, so 0000 is added in each group of 4-bits in adder 2. So, the result is 0001 0001 in BCD, i.e. equivalent to 11 in decimal. A minus sign is placed before 11. The answer will be −11.

③ $546_{10} - 429_{10}$

$$\begin{array}{r} 546 \\ -429 \\ \hline (117)_{10} \end{array}$$

Scanned with CamScanner

ii) 0100 0010 1001   429

1010 1011 1001   −546

1110 1110 0010   −117

Transfer 1's Comp to Adder 2

0001 0001 1101   Cn = 1 Add 1010

\+ 0000 0000 1010   Cn = 0 Add 0000

$\underbrace{0001}_{1}$ $\underbrace{0000}_{1}$ 1 $\underbrace{0111}_{7}$ Discard carry.

So, the difference output is −117.

429 − 546 = −117

(i) 546   0101 0100 0110

−429   0100 0010 1001

Take 1's comp. of $(429)_{10}$ 1011 1101 0110

0101 0100 0110

\+ 1011 1101 0110

---

1 0001 0001 1100

+1

0001 0001 1101   Cn = 1 Add 0000

\+ 0000 0000 1010   Cn = 0 Add 1010

$\underbrace{0001}_{1}$ $\underbrace{0001}_{1}$ 1 $\underbrace{0111}_{7}$ Discard carry

$(117)_{10}$

Scanned with CamScanner

UNIT = 2 BOOLEAN ALGEBRA

Basic properties

① **Associative law** - A binary operator $*$ on a set S is said to be associative whenever:-

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

② **Commutative law** A binary operator $*$ on a set S is said to be commutative whenever

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

③ **Identity element** A set S is said to have an identity element w.r.t. a binary operation $*$ on S if there exists an element $e \in S$ with the property

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

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

$$x * y = e$$

⑥ <u>Distributive law</u> If ~~these~~ $*$ and $\cdot$ are two binary operators on a set S, $*$ is said to be distributive over $\cdot$ whenever

$$x * (y \cdot z) = (x * y) \cdot (x * z)$$

<u>Basic Theorems</u> (in accordance with Boolean values)

① Closure

② Commutative $\quad x \cdot y = y \cdot x$

$$x + y = y + x$$

③ Distributive

$$x \cdot (y + z) = xy + xz$$

$$x + (y \cdot z) = (x + y) \cdot (x + z)$$

④ Complement

$$x + \bar{x} = 1$$

$$x \cdot \bar{x} = 0$$

⑤ Identity $\quad x + 0 = x$ (Additive Identity)

$x \cdot 1 = x$ (Multiplicative Identit

Scanned with CamScanner

UNIT 2 Boolean Algebra & Logic Gates

Boolean Algebra

A boolean algebra is an algebric system consisting of the set {0,1}. Here Boolean '0' and '1' do not represent actual numbers but instead represent the state of a voltage variable called <u>logic level</u>.

<u>Eg:</u> Complement of A is represented by $\bar{A}$.

Boolean Postulates

There are 5 postulates of Boolean Algebra.

1. If $A=1$ then $\bar{A}=0$
2. $0+0=0$
3. $0+1=1$
4. $1+0=1$
5. $1+1=1$

1. If $A=0$ then $\bar{A}=1$
2. $0 \cdot 0 = 0$
3. $0 \cdot 1 = 0$
4. $1 \cdot 0 = 0$
5. $1 \cdot 1 = 1$

Scanned with CamScanner

Postulates and Theorems of Boolean Algebra

	Normal (One) form	Dual Form	
1.	$x + 0 = x$	$x \cdot 1 = x$	
2.	$x + x' = 1$	$x \cdot x' = 0$	
3.	$x + x = x$	$x \cdot x = x$	
4.	$x + 1 = 1$	$x \cdot 0 = 0$	
5.	$x + y = y + x$	$x \cdot y = y \cdot x$	Commutative
6.	$(x+y)+z = x+(y+z)$	$x \cdot (y \cdot z) = (x \cdot y) \cdot z$	Associative
7.	$x \cdot (y+z) = (x \cdot y) + x \cdot z$	$x + (y \cdot z) = (x+y) \cdot (x+z)$	Distributive
8.	$(x')' = x$		
9.	$(x+y)' = x' \cdot y'$	$(x \cdot y)' = x' + y'$	De-Morgan's Theorem
10.	$x + xy = x$	$x(x+y) = x$	Absorption

Scanned with CamScanner

Duality

Postulates of Boolean algebra remains valid if the operators & identity elements are interchanged.

① $x + x = x$

By duality theorem

$x \cdot x = x$

② $x + 1 = 1$

By duality theorem $x \cdot 0 = 0$

Operator Precedence

Parentheses, NOT, AND, OR

Ques1 Simply the following Boolean functions to a minimum number of literals.

① $\underline{x + x'y} = (x+x')(x+y) = 1 \cdot (x+y)$

from $x+yz = (x+y)(x+z)$

$= (x+y)$

② $x(x'+y) = xx' + xy = 0 + xy = xy$

③ $x'y'z' + x'yz + xy'$

$= x'z(y + y') + xy'$

$= x'z + xy'$

④ $xy + x'z + yz$

$= xy + x'z + yz(x + x')$

$= xy + x'z + xyz + x'yz$

$= xy(1 + z) + x'z(1 + y)$

$= xy + x'z$

⑤ $(x + y) \cdot (x' + z) \cdot (y + z)$ by duality

De Morgan's Theorem

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

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

Ques1 (i) $\overline{(\overline{A} + \overline{A+B})(\overline{B} + \overline{B+C})}$

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

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

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

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

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

Scanned with CamScanner

CANONICAL and STANDARD FORMS

A binary variable say x may have its Normal form (x) or its complement form (x' or $\bar{x}$).

for two variables (x and y)

Possible combinations (AND Terms) : $\underbrace{x'y',\ x'y,\ xy' \text{ and } xy}_{\text{Standard Product or Minterm } (m_j)}$

Possible Combinations (OR Terms) : $\underbrace{x+y,\ x+y',\ x'+y \text{ and } x'+y'}_{\text{Standard Sum or Maxterm } (M_j)}$

* For n variables, total no. of minterms or Maxterms will be 2^n

  These 2^n minterms or Maxterms represent values in binary equivalent from 0 to 2^n-1.

* Each minterm (m_j) is obtained from an AND term of the n variables, with each variable being primed if the corresponding bit of binary number is a 0 and unprimed if it is a 1.

* Each Maxterm (M_j) is obtained from an OR term of the n variables, with each variable being unprimed if the corresponding bit of binary number is a Zero(0) and Primed if it is a 1.

* Here, j denotes the decimal equivalent of the binary number formed by combination of n variable.

* NOTE: $$m_j' = M_j$$

Table: for 3 variables x, y and z

			Minterms		Maxterms	
x	y	z	Term	Designation	Term	Designation
0	0	0	$x'y'z'$	m_0	$x+y+z$	M_0
0	0	1	$x'y'z$	m_1	$x+y+z'$	M_1
0	1	0	$x'yz'$	m_2	$x+y'+z$	M_2
0	1	1	$x'yz$	m_3	$x+y'+z'$	M_3
1	0	0	$xy'z'$	m_4	$x'+y+z$	M_4
1	0	1	$xy'z$	m_5	$x'+y+z'$	M_5
1	1	0	xyz'	m_6	$x'+y'+z$	M_6
1	1	1	xyz	m_7	$x'+y'+z'$	M_7

Canonical form: Any Boolean function can be expressed as sum of minterms or as product of Maxterms known as canonical forms.

Sum of minterms → Sum of Product Terms (SOP) Σ

Product of Maxterms → Product of Sum Terms (POS) Π

The sum of products is a boolean expression containing AND terms, called product terms, of one or more literals each. The Sum denotes the ORing of these terms. e.g.

$$F = y' + xy + x'yz'$$

Scanned with CamScanner

The product of Sums is a boolean expression containing OR terms, called sum terms. Each term may have any number of literals. The product denotes the ANDing of these terms.

e.g. $F = x(y'+z)(x'+y+z'+w)$

Example:

x	y	z	function f_1	function f_2
0	0	0	0	0
0	0	1	1	0
0	1	0	0	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

from above

$f_1 = m_1 + m_4 + m_7$ and

$f_2 = m_3 + m_5 + m_6 + m_7$

If we take $f_1' = m_0 + m_2 + m_3 + m_5 + m_6$

$= (x'y'z') + x'yz' + x'yz + xy'z + xyz'$

$\therefore f_1 = (f_1')' = [x'y'z' + x'yz' + x'yz + xy'z + xyz']'$

$= (x+y+z)\cdot(x+y'+z)\cdot(x+y'+z')\cdot(x'+y+z')\cdot(x'+y'+z)$

$= M_0 . M_2 M_3 M_5 M_6$

Scanned with CamScanner

Conversion between Canonical forms

$$F(A,B,C) = \Sigma(1,4,5,6,7) = \Pi(0,2,3)$$

How ?

$\Rightarrow F'(A,B,C) = \Sigma(0,2,3) = m_0 + m_2 + m_3$

Now,

$$F(A,B,C) = [F'(A,B,C)]' = (m_0 + m_2 + m_3)'$$

$$= m_0' \cdot m_2' \cdot m_3'$$

$$= M_0 \cdot M_2 \cdot M_3 \quad \left\{ \text{as } m_j' = M_j \right.$$

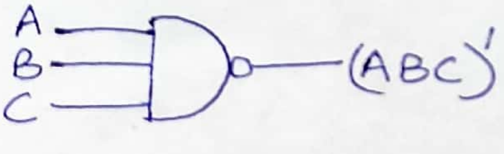
$\Rightarrow F(A,B,C) = \Pi(0,2,3)$

Similarly,

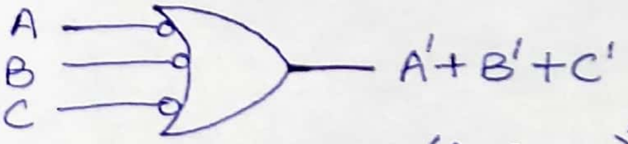
If $F(x,y,z) = \Pi(0,2,4,5)$

then, $F(x,y,z) = \Sigma(1,3,6,7)$.

Invert-AND (NAND)

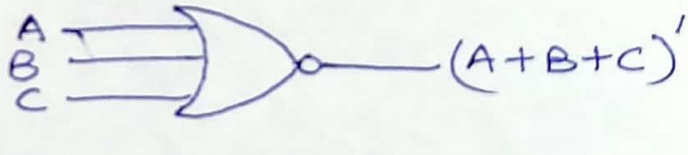


$\equiv$

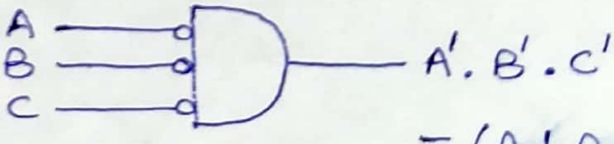


$= (A.B.C)'$

Invert-OR (NOR)



$\equiv$



$= (A+B+C)'$

Scanned with CamScanner

(3.) (a.) $x + xy = x$

$$x + xy = x.1 + xy$$
$$= x(1+y)$$
$$= x.1 = x$$

(b.) $x(x+y) = x$ by Duality

Boolean functions

A boolean function is an expression formed with binary variables, two binary operators OR (+) and AND (.), the unary operator NOT ($\bar{\ }$ or $'$), parentheses and equal sign

bar prime

Literal

A literal is a primed or unprimed variable.

Example:

(1.) The boolean function

$$F_1 = xyz'$$

The function F_1 is equal to 1 if $x=1$ and $y=1$ and $z'=1$ otherwise $F_1=0$.

(2.) $F_2 = x + y'z$

The function $F_2 = 1$ if $x=1$ or if $y=0$ with $z=1$

Scanned with CamScanner

(3.) $F_3 = \underbrace{x'y'z}{} + \underbrace{x'yz}{} + \underbrace{xy'}{}$

Either cases for $F_3 = 1$

(4.) $F_4 = \underbrace{xy'}{} + \underbrace{x'z}{}$

Either cases for $F_4 = 1$

Tabular Representation

For n variables, we can have 2^n different combinations possible. ranging from 0 to $2^n - 1$

Thus.

x	y	z	F_1	F_2	F_3	F_4
0	0	0	0	0	0	0
0	0	1	0	1	1	1
0	1	0	0	0	0	0
0	1	1	0	0	1	1
1	0	0	0	1	1	1
1	0	1	0	1	1	1
1	1	0	1	1	0	0
1	1	1	0	1	0	0

Scanned with CamScanner

2.) OR

$F = x + y$

x	y	F
0	0	0
0	1	1
1	0	1
1	1	1

3.) Inverter (NOT)

$F = x'$ or $\bar{x}$

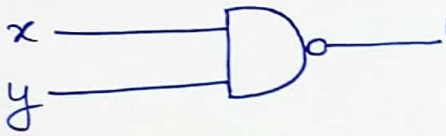
x	F
0	1
1	0

4.) Buffer (Transfer)

$F = x$

x	F
0	0
1	1

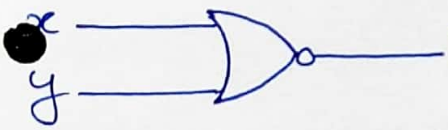
5.) NAND



$F = (x \cdot y)'$

x	y	F
0	0	1
0	1	1
1	0	1
1	1	0

6.) NOR



$F = (x + y)'$

x	y	F
0	0	1
0	1	0
1	0	0
1	1	0

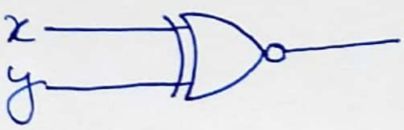
7.) Exclusive-OR (XOR)

$F = xy' + x'y$

$= x \oplus y$

x	y	F
0	0	0
0	1	1
1	0	1
1	1	0

8.) Exclusive-NOR / Equivalence



$F = xy + x'y'$

$= x \odot y$

x	y	F
0	0	1
0	1	0
1	0	0
1	1	1

Scanned with CamScanner

COMPLEMENT OF A FUNCTION

Method 1: Using De-Morgan's Theorem

De-Morgan Theorem : $(x+y)' = x' . y'$

$(x . y)' = x' + y'$

Ex. 1 $F_1 = x'yz' + x'y'z$

Then $F_1' = (x'yz' + x'y'z)'$

$= (x'yz')' . (x'y'z)'$

$= [(x')' + y' + (z')'] . [(x')' + (y')' + z']$

$= (x + y' + z) . (x + y + z')$

Method 2: Using Dual of the function

Step 1: Take Dual of the given function.

Step 2: Complement each literal after taking its Dual (step 1).

* Literal :→ A literal is a primed or unprimed variable.

Ex. 1 $F_1 = x'yz' + x'y'z$

Dual of $F_1 = [x' + y + z'] . [x' + y' + z]$

Complementing each literal $F_1' = [x + y' + z] . [x + y + z']$

Ex. 2 (i) $F = x(y'z' + yz)$

(ii) $F = (x+y)(x'+z)(y+z)$

Scanned with CamScanner

Minimization using Boolean Algebra

1. simplify to minimum no. of Literals

(i) $xy + xy'$

$= x(y+y') = x$

(ii) $(x+y)(x+y')$

$= x + (yy')$

$= x$

(iii) $xyz + x'y + xyz'$

$= xy(z+z') + x'y$

$= xy + x'y$

$= y(x+x')$

$= y$

(iv) $zx + zx'y$

$= zxy + zxy' + zx'y$

$= \underbrace{zxy + zxy'} + \underbrace{zx'y + zxy}$

$= zx(y+y') + zy(x'+x)$

$= zx + zy$

$= z(x+y)$

(v) $(A+B)'(A'+B')'$

$= A'B'.[(A')'.(B')']$

$= A'B'.[A.B]$

$= AA'.BB'$

$= 0$

Scanned with CamScanner

(vi) $y(wz' + wz) + xy$

$= wy(z + z') + xy$

$= y(w + x)$.

(2.) Reduce to reqd. no. of literals

(i) $\underbrace{ABC}_{G_1} + \underbrace{A'B'C + A'BC}_{G_3} + \underline{ABC'} + \underbrace{A'B'C'}_{G_2}$ to 5 literals

$= \underbrace{AB}_{G_1} + \underbrace{A'B'}_{G_2} + \underbrace{A'C}_{G_3}$

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

(ii) $BC + AC' + AB + BCD$ To 4 literals

$= \underline{ABC + A'BC} + \cancel{ABC'} + \underline{AB'C'} + \cancel{ABC} + \underline{ABC'} + \underline{BCD}$

$= \underbrace{ABC + A'BC} + \underbrace{AB'C' + ABC'} + BCD$

$= \underline{BC} + AC' + \underline{BCD}$

$= BC(1 + D) + AC'$

$= BC + AC'$.

(iii) $[(CD)' + A]' + A + CD + AB$ to 3 literals

$= [(CD)']' \cdot A' + A + CD + AB$

$= A'CD + A + CD + AB$

$= CD + A$.

(iv) $(A+C+D)(A+C+D')(A+C'+D)(A+B')$ to 4 literals

$\Rightarrow$ $F = (A+C+D)(A+C+D')(A+C'+D)(A+B')$

we can write

$$F = [F']' = \left[(A+C+D)' + (A+C+D')' + (A+C'+D)' + (A+B')' \right]'$$

$$= \left[A'C'D' + A'C'D + A'CD' + A'B \right]'$$

$$= \left[A'C' + A'D' + A'B \right]'$$

$$= \left[A'(C'+D'+B) \right]'$$

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

$$= A + B'CD.$$

Alternatively

$$(A+C+D)(A+C+D')(A+C'+D)(A+B')$$

$$= \left[(A+C) + DD' \right] \cdot \left[(A+D) + CC' \right] \cdot (A+B')$$

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

$$= A + B'CD.$$

Scanned with CamScanner

(3.) Find complement and reduce to min^m. no. of literals

(i) $F = (BC' + A'D)(AB' + CD')$

$$F' = [(BC' + A'D)(AB' + CD')]'$$

$$= [AB'.BC' + AB'.A'D + CD'.BC' + CD'.A'D]'$$

$$= [0]' = 1.$$

(ii) $F = B'D + A'BC' + ACD + A'BC$

$\cancel{F' = (B+D').(}$

$$F' = [B'D + A'BC' + A'BC + ACD]'$$

$$= [B'D + A'B + ACD]'$$

$$= (B+D').(A+B').(A'+C'+D')$$

$$= (AB + AD' + B'D').(A'+C'+D')$$

$$= A'B'D + ABC' + AC'D' + \underbrace{B'C'D' + \underbrace{ABD' + AD'}_{AD'} + B'D'}_{B'D'}$$

$$= \underbrace{A'B'D' + ABC' + A\,\underbrace{C'D' + AD'} + B'D'}$$

$$= B'D' + AD' + ABC'$$

$$= B'D' + A(D' + BC')$$

Scanned with CamScanner

Simplification of Boolean functions

1) using K-map (in Syllabus only)
2) using quine-Mcklusky Method

Karnaugh Map (K-map)

Two-variable K-map

x	y	minterms
0	0	$m_0 \rightarrow x'y'$
0	1	$m_1 \rightarrow x'y$
1	0	$m_2 \rightarrow xy'$
1	1	$m_3 \rightarrow xy$

$x \backslash y$	0	1 (y)
0	$x'y'$	$x'y$
x { 1	xy'	xy

$x \backslash y$	0	1
0	m_0	m_1
1	m_2	m_3

Replica to right →

3-Variable K-map

x	y	z	minterm
0	0	0	$m_0 \rightarrow x'y'z'$
0	0	1	$m_1 \rightarrow x'y'z$
0	1	0	$m_2 \rightarrow x'yz'$
0	1	1	$m_3 \rightarrow x'yz$
1	0	0	$m_4 \rightarrow xy'z'$
1	0	1	$m_5 \rightarrow xy'z$
1	1	0	$m_6 \rightarrow xyz'$
1	1	1	$m_7 \rightarrow xyz$

in Gray codes

$x \backslash yz$	00	01	11	10
0	$x'y'z'$	$x'y'z$	$x'yz$	$x'yz'$
1	$xy'z'$	$xy'z$	xyz	xyz'

$x \backslash yz$	00	01	11	10
0	m_0	m_1	m_3	m_2
1	m_4	m_5	m_7	m_6

$x \backslash yz$	00	01 (z)	11 (z, y)	10 (y)
0	0	1	3	2
x { 1	4	5	7	6

Replica to Downwards give 4-variable K-map

Scanned with CamScanner

4-Variable K-map

w	x	y	z	minterms
0	0	0	0	$m_0 \rightarrow w'x'y'z'$
0	0	0	1	$m_1 \rightarrow w'x'y'z$
0	0	1	0	$m_2 \rightarrow w'x'yz'$
0	0	1	1	$m_3 \rightarrow w'x'yz$
0	1	0	0	$m_4 \rightarrow w'xy'z'$
0	1	0	1	$m_5 \rightarrow w'xy'z$
0	1	1	0	$m_6 \rightarrow w'xyz'$
0	1	1	1	$m_7 \rightarrow w'xyz$
1	0	0	0	$m_8 \rightarrow wx'y'z'$
1	0	0	1	$m_9 \rightarrow wx'y'z$
1	0	1	0	$m_{10} \rightarrow wx'yz'$
1	0	1	1	$m_{11} \rightarrow wx'yz$
1	1	0	0	$m_{12} \rightarrow wxy'z'$
1	1	0	1	$m_{13} \rightarrow wxy'z$
1	1	1	0	$m_{14} \rightarrow wxyz'$
1	1	1	1	$m_{15} \rightarrow wxyz$

wx \ yz	00	01	11	10
00	m_0	m_1	m_3	m_2
01	m_4	m_5	m_7	m_6
11	m_{12}	m_{13}	m_{15}	m_{14}
10	m_8	m_9	m_{11}	m_{10}

z

	wx \ yz	00	01	11	10	
	00	0	1	3	2	
	01	4	5	7	6	x
w	11	12	13	15	14	x
w	10	8	9	11	10	

y

AB \ CDE	000	001	011	010	110	111	101	100
00	0	1	3	2	6	7	5	4
01	8	9	11	10	14	15	13	12
11	24	25	27	26	30	31	29	28
10	16	17	19	18	22	23	21	20

Figure 3-11 Five-variable map

Scanned with CamScanner

	ABC \ DEF	000	001	011	010	110	111	101	100	
							D			
	000	0	1	3	2	6	7	5	4	
	001	8	9	11	10	14	15	13	12	C
	011	24	25	27	26	30	31	29	28	C
	010	16	17	19	18	22	23	21	20	B
	110	48	49	51	50	54	55	53	52	B
A	111	56	57	59	58	62	63	61	60	C
A	101	40	41	43	42	46	47	45	44	C
	100	32	33	35	34	38	39	37	36	
				E	E	E	E			
			F	F			F	F		

Figure 3-12 Six-variable map

Exercise Questions on topics

(i) Number System and Number-Base Conversions

(ii) Complements

(iii) Subtraction through complement method.

Q.1. Determine the radix-2 representation of the integer whose radix-8 representation is 3456.

Q.2. Give procedure to convert from radix-2 to radix-2^k and vice-versa.

Q.3. Perform the following subtractions:

(i) $(213.AB)_{16} - (116.06)_{16}$ — $(FD.A5)_{16}$

(ii) $(33-57)_{10}$ using 2's complement — $(-24)_{10}$

(iii) $(1ABC)_{16} - (1DEF)_{16}$ using 2's complement

Q.4. find the radix value for

(i) $(25)_{10} + (30)_8 = (54)_r$ — $r = 9$

(ii) $(\sqrt{71})_r = (8)_r$ — $r = 9$

	Answers
Q.5. (i) find the 10's complement of $(935)_{11}$	$(175)_{11}$
(ii) find 9's complement of 00000	99999
(iii) find 2's complement of 00000, 10000	100000 and 10000
(iv) Add $(296)_{12}$ and $(57)_{12}$	$(331)_{12}$

Scanned with CamScanner

TUTORIAL # 3

Q.1 Simplify — **Answers**

(i) $[A\bar{B}(C+BD)+\bar{A}\bar{B}]C$ — $\bar{B}C$

(ii) $(A+B)'(A'+B')'$ — 0

(iii) $BC+AC'+AB+BCD$ — $BC+AC'$

(iv) $(A+C+D)(A+C+D')(A+C'+D)(A+B')$ — $A+B'CD$

Q.2 find Complement and simplify

(i) $B'D+A'BC'+ACD+A'BC$ — $B'D'+A(D'+BC')$

(ii) $AB'+C'D'$ — $(A'+B)(C+D)$

Q.3 Implement using NAND gates only

(i) $F(x,y,z) = \Sigma_m(0,6)$

(ii) EX-OR

(a) using NAND gates only

(b) using NOR gates only

Home Assignment

① Simplify:

(i) $\overline{Y}\overline{Z}+\overline{W}\overline{X}\overline{Z}+\overline{W}XY\overline{Z}+WY\overline{Z}$ — Ans. $\overline{Z}$

(ii) $[(CD)'+A]'+A+CD+AB$ — Ans. $A+CD$

(iii) $zx+zx'y$ — Ans. $z(x+y)$

(iv) $\overline{X}(X+Z)+\overline{A}+AZ$ — Ans. $Z+\overline{A}$

② Implement the following functions using NAND and NOR gates only (individually) after simplification:

(i) $F_1 = (B'+D')(A'+C'+D)(A+B'+C'+D)(A'+B+C'+D')$

(ii) $F_2 = AB'+C'D'+A'CD'+DC'(AB+A'B')+DB(AC'+A'C)$

Scanned with CamScanner

Q1. Draw the NAND equivalents for following basic gates: AND, OR, NOT and XOR.

Q2. Simplify (using Laws of Boolean algebra) to minimum number of literals:

(i) $F = AB' + ABD + ABD' + A'C'D' + A'BC'$

(ii) $F = zx + zx'y$

Q3. Minimize the following using K-map:

(i) $F(A,B,C,D) = \Pi_M(1,4,6,9,10,11) * d(13,14,15)$

(ii) $Y = AB(C'+D) + CD(A'+B') + A'CD$

(iii) $BE + B'DE'$ is a simplified version of the expression:

$A'BE + BCDE + BC'D'E + A'B'DE' + B'C'DE'$

Find Don't-Care conditions, if exists.

(iv) $F(A,B,C,D,E,F) = \Sigma(6,9,13,18,19,25,27,29,41,45,57,61)$

Q4. Express the following as SOP and POS forms:

(i) $F(A,B,C) = (A'+B)(B'+C)$

(ii) $F(x,y,z) = (xy + z)(y + xz)$

Q5. Show that the dual of Exclusive-OR is equal to its compliment.

Scanned with CamScanner

Binary Arithmatic

(i) Addition (iii) Multiplication

(ii) Subtraction (iv) Division

(i) Rules:

		Sum	Carry
1 + 1 =	0	0	1
0 + 0 =	0	0	0
0 + 1 =	1 + 0 = 0		0

Ex.

$$\underbrace{(110110)_2}_{\text{Augend}} + \underbrace{(101111)_2}_{\text{Addend}}$$

```
  11 11
  110110
  101111
 --------
 1100101   Sum
```

(ii) $(011010)_2 - (100110)_2$

$n_1 < n_2$

Perform $n_2 - n_1$ and put -ve sign in front.

```
   2  2
  100110
  011010
 --------
  000100
```

(iii) $(101001.10)_2 \times (101.11)_2$

```
      101001.10   Multiplicand
         101.11   Multiplier
 ------------------
      101001 10
     10100110 x
    10100110 x x
   00000000 x x x
  10100110 x x x x
 ------------------
  1110111 0.1010
```

(iv) Binary Division

Rules: $\frac{1}{1} = 1$

$\frac{0}{1} = 0$

$\frac{0}{0}$ and $\frac{1}{0}$ → Not Allowed

Ex. $(11001.1)_2 \div (101)_2$

```
          101.0001
101 ) 11001.1
      -101
      -----
       00101
        -101
      ----------
         000 1000
              101
         --------
              011
```

∴ Quotient = 101.0001

Remainder = 011

BCD Arithmatic

~~(A) Addition~~

NOTE: Signed BCD Numbers are represented as an additional 4-bits in MS position for the sign.

for +ve number : 0000

for -ve number : 1001

Scanned with CamScanner

$f = \Sigma(0,1,2,8,10,11,14,15)$

(a)

	w	x	y	z
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
8	1	0	0	0
10	1	0	1	0
11	1	0	1	1
14	1	1	1	0
15	1	1	1	1

(b)

	w	x	y	z
0,1 (1)	0	0	0	–
0,2 (2)	0	0	–	0
0,8 (8)	–	0	0	0
2,10 (8)	–	0	1	0
8,10 (2)	1	0	–	0
10,11 (1)	1	0	1	–
10,14 (4)	1	–	1	0
11,15 (4)	1	–	1	1
14,15 (1)	1	1	1	–

(c)

	w	x	y	z
0,2,8,10	–	0	–	0
0,8,2,10	–	0	–	0
10,11,14,15	1	–	1	–
10,14,11,15	1	–	1	–

$$\boxed{F = x'z' + wy}$$

$F = \Sigma(1,4,6,7,8,9,10,11,15)$ 1011

(a)

	w	x	y	z
1 ✓	0	0	0	1
4 ✓	0	1	0	0
8 ✓	1	0	0	0
6 ✓	0	1	1	0
9 ✓	1	0	0	1
10 ✓	1	0	1	0
7 ✓	0	1	1	1
11 ✓	1	0	1	1
15	1	1	1	1

(b)

	w	x	y	z
1,9	–	0	0	1
4,6	0	1	–	0
8,9 ✓	1	0	0	–
8,10 ✓	1	0	–	0
6,7	0	1	1	–
9,11 ✓	1	0	–	1
10,11 ✓	1	0	1	–
11,15	1	–	1	1
7,15	–	1	1	1

(c)

	w	x	y	z
8,9,10,11	1	0	–	–
8,10,9,11	1	0	–	–

	wxyz	Prime Implicants
1,9	– 0 0 1	$x'y'z$
4,6	0 1 – 0	$w'xz'$
6,7	0 1 1 –	$w'xy$
11,15	1 – 1 1	wyz
7,15	– 1 1 1	xyz
8,9,10,11	1 0 – –	wx'

Selection of Prime Implicants

Term		1	4	6	7	8	9	10	11	15
$x'y'z$	1, 9	(X)					X			
$w'xz'$	4, 6		(X)	X						
$w'xy$	6, 7			X	X					
xyz	7, 15				X					X
wyz	11, 15								X	X
wx'	8, 9, 10, 11					(X)	X	(X)	X	
		✓	✓			✓		✓		

(X) single-Cross in a Column → Essential Prime Implicants (Must be taken)

1 ✓
4 ✓
6 ✓
7 X
8 ✓
9 ✓
10 ✓
11 ✓
15 X

To include 7 and 15, we must include xyz in final simplified expression

$$F = x'y'z + w'xz' + wx' + xyz$$

Scanned with CamScanner

ASSIGNMENT No. 1

Q.1. Convert the following:

(i) Decimal 225.225 to Binary, Octal and Hexadecimal

(ii) Octal 623.77 to Decimal, Binary and Hexadecimal

(iii) Hexadecimal 2AC5.D to Decimal, Octal and Binary

(iv) $(9725)_{10}$ into BCD and Excess-3.

(v) $(396)_{10}$ into Binary Code, BCD Code, Excess-3 code, Gray Code.

Q.2. Perform the following Subtractions:

(i) $(753)_{10} - (864)_{10}$

(ii) $(20)_{10} - (1000)_{10}$

(iii) $(10010)_2 - (10011)_2$

(iv) $(11010)_2 - (10000)_2$

(v) $(846)_9 - (782)_9$

(vi) $(439)_{12} - (583)_{12}$

using r's complement method and (r−1)'s Complement method.

Q.3. Find the value of x in the following equations:

(i) $(211)_x = (152)_8$

(ii) $(33)_x = (68)_{12}$

439
252

Q.4. Perform the following:

(i) $(436)_{10} - (685)_{10}$ BCD Subtraction

(ii) $(849)_{10} + (436)_{10}$ BCD Addition

Scanned with CamScanner

ASSIGNMENT # 4

1.) Write the first 20 decimal digits in base 3.

2.) Convert the decimal number 250.5 to base 3, base 4, base 7, base 8 and base 16.

3.) Convert the following:

(i) $(12121)_3 \rightarrow (\quad)_{10}$ (ii) $(8.3)_9 \rightarrow (\quad)_{10}$

(iii) $(50)_7 \rightarrow (\quad)_2$ (iv) $(198)_{12} \rightarrow (\quad)_8$

(v) $(623.77)_8 \rightarrow (\quad)_{16}$ (vi) $(2AC5.D)_H \rightarrow (\quad)_8$

4.) Perform the following subtractions:

(i) 753 − 864 using (a) 10's complement
(b) 9's complement

(ii) 100 − 110000 using (a.) 2's complement
(b) 1's complement

5.) Find the complement of following boolean functions and reduce them to a minimum number of literals:

(i) $(BC' + A'D)(AB' + CD')$

(ii) $B'D + A'BC' + ACD + A'BC$

(iii) $[(AB)'A][(AB)'B]$

6.) Show that the dual of Exclusive-OR is equal to its complement.

7.) Write the truth tables for 2-input, 3-input and 4-input Exclusive-OR and Exclusive-NOR.

8.) Simplify the following functions (using boolean algebra):

(i) $[A+C+D][A+C+D'][A+C'+D][A+B']$ to 4 literals

(ii) $BC + AC' + AB + BCD$ to 4 literals

Scanned with CamScanner

unit I → Number Sys. Time : 1 Hr. 30 mins
Unit III → Minimization Tech.

TEST PAPER

(1.) $(432)_5 = (x)_7$ find x.

(2.) Write Dual for $(A+\overline{B}+C)(\overline{A}+BD)+\overline{C}\,\overline{D}$

(3.) Write names of properties/laws :

(i) $x+xy=x$

(ii) $x+yz=(x+y)(x+z)$

(iii) $x+1=1$ is same as $x.0=0$

(4.) $(546)_{10} - (429)_{10}$ BCD Subtraction

(5.) Write the range

(i) 8-bit BCD

(ii) 8-bit 2's complement form

(iii) 8-bit Sign Magnitude form

(iv) 8-bit Unsigned Magnitude form

(6.) Excess-3 code is given for a no. as : 110010010101
find Reflected code equivalent for the original number.

(7.) Simplify using Tabulation Method.

$F(A,B,C,D) = \Sigma m(1,3,7,11,15) + d(0,2,5)$

(8.) Implement (using NAND gates only)

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

(9.) Prove (using Boolean algebra):

(i) $ab' + b'c' + a'c' = ab' + a'c'$

(ii) $a'b' + ab + a'b = a' + b$

Scanned with CamScanner

(10.) Perform

(i) $(11.1001)_2 - (01.110)_2$ using 2's complement

(ii) $(32)_8 + (67)_8$ Direct without conversion

(11.) Minimize

(i) $F(A,B,C,D) = \Pi_M(1,4,6,9,10,11) * d(13,14,15)$

(ii) $F = y' + x'z'$ with $d = yz + xy$

(12.) Obtain weighted Binary code for Base-12 digits using weights of 5421.

(13.) Prove whether the weighted code 4, 4, 3, −2 is self-complimentary or not.

(14.) Convert 250.5 to

(i) Base 3

(ii) Base 7

(iii) Base 16

(15.) Add and Multiply following numbers without converting to decimal

(i) $(135.4)_6$ and $(43.2)_6$

(ii) $(296)_{12}$ and $(57)_{12}$

Scanned with CamScanner

NAME:
ROLL NO:
SECTION:

JAIPUR ENGINEERING COLLEGE AND RESEARCH CENTRE
Department of Computer Science Engineering

COURSE : B.Tech	**SEMESTER III**	**SECTION : A & B**
SUBJECT : DIGITAL ELECTRONICS		**CODE : 3CS3**
TIME: 1:00 Hr	**Unit Test -1**	**MM: 10**

COURSE OUTCOME
CO1: To acquaint the students with the fundamental principles of two-valued logic and various devices used to implement logical operations on variables.

Instructions: Attempt all sections

Q. 1 For a number with radix 'r' having **p**-digits in whole part and **q**-digits in fractional part, Write the formula for

(i) **r's Complement :** ______________________

(ii) **(r − 1)'s Complement:** ______________________ **(1)**

Q. 2 For the relation $(432)_5 = (x)_7$ the value of x will be ______________ **(1)**

Q. 3 For the expression: $(A + \overline{B} + C)(\overline{A} + BD) + \overline{C}\overline{D}$, write the Dual:

______________________________ **(1)**

Q. 4 Write the definition of Self-Complementing Codes with one example. **(1)**

____

____

Q. 5 Perform the subtraction using 9's complement method: $(\mathbf{1010})_{10} - (\mathbf{1100110})_{10}$ **(3)**

OR

Q. 5 The Excess-3 code of a number x is **(110010010101).** Find the equivalent reflected code for x. **(3)**

Q. 6 Write the name of laws: **(3)**

(i) $x + xy = x$ ______________________

(ii) $x + yz = (x + y)(x + z)$ ______________________

(iii) $\overline{(x + y)} = \overline{x} \,.\overline{y}$ ______________________

(iv) $x + 1 = 1$ *is same as* $x \,.0 = 0$ ______________________

(v) $x + y = y + x$ ______________________

(vi) $x + (y + z) = (x + y) + z$ ______________________

OR

Q. 6 Draw the logic diagram, using appropriate gates, for the expression $F = \overline{A + \overline{B}C\,(\overline{C + D})}$.
Note: Only Normal Inputs are available. **(3)**

Scanned with CamScanner

Exercise

1.) Simplify (to minimum number of literals)

(i) $xy + xy'$ (ii) $xyz + x'y + xyz'$ (iii) $zx + zx'y$

(iv) $(A+B)'(A'+B')'$ (v) $xy + y(wz + wz')$

2.) Obtain the truth table of the function:

$$F = xy + xy' + y'z$$

a.) Implement the func? using AND, OR and NOT gates

b.) Implement it with only OR and NOT gates.

c.) Implement it with only AND and NOT gates.

3.) Express the following SOP and POS forms:

(i) $F(w,x,y,z) = y'z + wxy' + wxz' + w'x'z$

(ii) $F(A,B,C) = (A'+B)(B'+C)$

(iii) $F(x,y,z) = 1$

(iv) $F(x,y,z) = (xy+z)(y+xz)$

4.) Convert the following to other canonical form:

(i) $F(x,y,z) = \Sigma(1,3,7)$

(ii) $F(A,B,C,D) = \Pi(0,1,2,3,4,6,12)$

5.) Show that the dual of the Exclusive-OR is equal to its complement.

Scanned with CamScanner