Date 29/09/22
Page No.

* Digital Systems & Binary Numbers
* Boolean Algebra & Logic Gates
* Gate-Level Minimization
* Combinational Logic
* Sequential Circuits
  - Flip-Flops
* Registers and Counters
  - Counters
  - Shift Registers
* Design of RTL

Ch-01
Digital Systems & Binary Numbers

Signal - Physical quantity which depends on one or more than one independent quantity

$$y = x(t)$$

dependent, independent

Digital Signal
No. of amplitude levels = finite

Digital System
Uses digital signal, manipulates, process it and provides definitive output.

Decimal Number System
(Base or Radix = 10)

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

$$(\underset{2}{4}\,\underset{1}{3}\,\underset{0}{2})_{10} = 4\times10^2 + 3\times10^1 + 2\times10^0$$

Base	Number System
2	Binary
8	Octal
10	Decimal
16	Hexadecimal

Binary Number System (Base = 2)
0 and 1 : binary digits
↳ bits

1B = 8 bits

- 2 bit number system
  Possibility : $2^2 = 4$
  00, 01, 10, 11
  Max value : $2^n - 1$
  3

- 3 bit
  $2^3 = 8$
  $2^n - 1$
  7

$$(\underset{3}{1}\,\underset{2}{1}\,\underset{1}{0}\,\underset{0}{1})_2 = 1\times2^3 + 1\times2^2 + 0\times2^1 + 1\times2^0$$
$$= 8 + 4 + 0 + 1$$
$$= (13)_{10}$$

$$(13)_{10} = (1101)_2$$

2	13	
2	6	1
2	3	0
	1	1

Q $(\underset{4\,3\,2\,1\,0}{11101} \cdot \underset{-1\,-2\,-3}{001})_2$

$= 1\times 2^4 + 1\times 2^3 + 1\times 2^2 + 0\times 2^1 + 1\times 2^0 + 0\times 2^{-1} + 0\times 2^{-2} + 1\times 2^{-3}$

$= 16 + 8 + 4 + 0 + 1 + 0 + 0 + \frac{1}{8}$

$= 29 + 0.125$

$= 29.125$

Q $(17.25)_{10}$

10001.01

2	17	
2	8	1
2	4	0
2	2	0
	1	0

0.25×2
$= 0.50$
0.50×2
$= 1$

6-10-22

Standard formula

$a_m \cdots a_3 a_2 a_1 a_0 \cdot a_{-1} a_{-2} \cdots a_{-m}$

$= a_m r^m + \ldots + a_3 r^2 + a_1 r^1 + a_0 r^0 + a_{-1} r^{-1} + a_{-2} r^{-2} + \ldots + a_{-m} r^{-m}$

r = radix / base

Q $(\underset{4\,3\,2\,1\,0}{11010} \cdot \underset{-1\,-2}{11})_2$

$= 1\times 2^4 + 1\times 2^3 + 0\times 2^2 + 1\times 2^1 + 0\times 2^0 + 1\times 2^{-1} + 1\times$

$= 16 + 8 + 0 + 2 + \frac{1}{2} + \frac{1}{4}$

$= 26 + 0.75$

$= (26.75)_{10}$

Q $(\underset{2\,1\,0}{127} \cdot \underset{-1}{4})_8$

$= 1\times 8^2 + 2\times 8^1 + 7\times 8^0 + 4\times 8^{-1}$

$= 64 + 16 + 7 + \frac{1}{2}$

$= 87 + 0.5$

$= (87.5)_{10}$

Q $(\underset{3\,2\,1\,0}{4021} \cdot \underset{-1}{2})_5$

$= 4\times 5^3 + 0\times 5^2 + 2\times 5^1 + 1\times 5^0 + 2\times 5^{-1}$

$= 500 + 0 + 10 + 1 + \frac{2}{5}$

$= (511.4)_{10}$

Q $(\underset{3\,2\,1\,0}{A21F})_{16}$

$= 10\times 16^3 + 2\times 16^2 + 1\times 16^1 + 15\times 16^0$

$= 40960 + 512 + 16 + 15$

$= (41503)_{10}$

Q $(BA9)_{13}$
2 1 0

$= 11 \times 13^2 + 10 \times 13^1 + 9 \times 13^0$
$= 1859 + 130 + 9$
$= (1998)_{10}$

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

2	41	
2	20	1
2	10	0
2	5	0
2	2	1
	1	0

Q $(41.25)_{10} = (101001.01)_2$

0.25×2
$= 0.5$ 0
0.5×2
$= 1.00$ 1

Q $(87.5)_{10} = (127.4)_8$

8	87	
8	10	7
	1	2

$0.5 \times 8 = 4.0$

Q $(511.4)_{10} = (4021.2)_5$

5	511	
5	102	1
5	20	2
	4	0

$0.4 \times 5 = 2.0$

Q $(41503)_{10}$

16	41503	
16	2593	15
16	162	1
	10	2

10 2 1 15
$= (A21F)_{16}$

3 bit (8)
000 - 0
001 - 1
010 - 2
011 - 3
100 - 4
101 - 5
110 - 6
111 - 7

2 bit
00 - 0
01 - 1
10 - 2
11 - 3

Q $(471)_8 = (100\ 111\ 001)_2$

Q $(011\ 000\ 111)_2 = (307)_8$

Q $(2FA1)_{16} = (0010\ 1111\ 1010\ 0001)_2$
0010 1111 1010 0001

Base 10 → Base r
* whole - repeated division by r
* fraction - repeated multiplication by r

Base r → Base 10
Weighted sum r^n

Date 12-10-22
Page No.

Binary Arithmetic

Binary Addition

A	B	S	C carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Q. $(21)_{10}$
$(14)_{10}$

$$\begin{array}{r} 10101 \\ +\ 01110 \\ \hline 100011 \end{array} = 35$$

Q.

$$\begin{array}{r} 10111 \\ +\ 01111 \\ \hline 100110 \end{array}$$

$= 32+4+2$
$= 38$

$= 16+4+2+1 = 23$
$= 8+4+2+1 = 15$
38

Binary Subtraction

A	B	D	B borrow
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

$$\begin{array}{r} 10111 \\ -\ 01111 \\ \hline 01000 \end{array} = 8$$

Date
Page No.

Q.

$$\begin{array}{r} 1011 \\ \times\ 101 \\ \hline 1011 \\ 0000 \\ 1011 \\ \hline 110111 \end{array} = 55$$

Q. $(526)_8$
$+(477)_8$
$(1225)_8$

Q. $(236)_7$
$+(455)_7$
$(1024)_7$

$(11)_{10} = (14)_2$

Q. $(ABCD)_{16}$
$+(FDA0)_{16}$
$(1A96D)_{16}$

$$\begin{array}{r|l} 16 & 22 \\ 16 & 1\ \ 6 \\ & 0\ \ 1 \end{array}$$

Q. $(AAA)_{12}$
$+(999)_{12}$
$(1887)_{12}$

Date 13-10-22
Page No.

Complement of Numbers

* Simplify the subtraction
* radix complement (r's complement)
* diminished radix complement
  (r-1)'s complement

r = 10 10's complement
9's complement

r = 2 2's complement
1's complement

Diminished radix complement

$(N)_r$ having n digits
$(2014)_{10}$: N = 2014
n = 4
r = 10

(r-1)'s complement of above number is $(r^n - 1) - N$

r = 10
9's complement of $(22098)_{10}$

n = 5 N = 22098 r = 10
$(10^5 - 1) - 22098$
= 99999 - 22098
= 77901

Date
Page No.

* 9's comp of decimal no can be obtained by subtracting each digit from 9

r = 2 1's complement
$(1011)_2$
$(2^4 - 1) - 1011$
15 - N
1111 - 1011
= 0100

* 1's comp can be obtained by subtracting each bit from 1

* Complement or NOT gate of each bit gives 1's complement

Radix complement

$(N)_r$ having n digits, r's complement of above number is
$r^n - N$ for $N \neq 0$

10's complement of $(22098)_{10}$

$10^5 - 22098$
= 77902

2's complement of $(11011)_2$

11011 → 00100 (1's comp)

00101 → 2's comp

$(ABC)_{16}$

15's complement

$$\begin{array}{r} F\,F\,F \\ -\,A\,B\,C \\ \hline 5\,4\,3 \end{array}$$

16's complement = 544

Subtraction using (r-1)'s complement

$A = (23567)_{10}$

$B = (1942)_{10}$

A - B = A + (-B)

→ 9's complement for B = 98057

→ Addition

$$\begin{array}{r} 23567 \\ +\ 98057 \\ \hline 121624 \\ +\ 1 \\ \hline 21625 \end{array}$$

Overflow ← 1

B - A = B + (-A)

$$\begin{array}{r} 01942 \\ -\ 23567 \end{array}$$

9's complement of A

$$\begin{array}{r} 99999 \\ -\ 23567 \\ \hline 76432 \end{array}$$

$$\begin{array}{r} 01942 \\ +\ 76432 \\ \hline 78374 \end{array}$$

→ 9's complement of result

$$\begin{array}{r} 99999 \\ 78374 \\ \hline 21625 \end{array}$$

78374 = - 21625

Steps:

Sub of two unsigned numbers using (r-1)'s complement:

For A-B:

* Both no have same no. of digits or bits
* Calculate (r-1)'s complement of subtrahend B and add result with minuend A
* If result a carry a value then add carry to least significant digit

and result is positive
(s) don't have carry then take
(r-1)'s comp of result and
put -ve sign to left

17-10-22

Q Subtraction using radix complement technique

1. 23 - 13

Using 9's complement

23 + 86 = 109
= 10

13 - 23
13 + 76 = 89
-(9's complement of ans)
-10

Using 1's complement

23 → 10111
13 → -01101

```
  10111
+ 10010
 101001
+      1
 01010
```

13 01101
23 -10111

01010 = -10

```
  01101
+ 01000
  10101
```

10's complement

23, 13
23 - 13 = 23 + (-13)
10's complement of 13 = 9's comp + 1
= 87

```
   23
+  87
  110
discard
```
= 10

13 - 23 = 13 + (-23)
10's comp of 23 = 77

```
   13
+  77
   90
```

-(10's complement) = -10

2's complement

23 - 13 = 23 + (-13)
= 10111
+ 10011
101010 = 01010

13 - 23 = 13 + (-23)

```
  01101
+ 01001
  10110
```

Q. 4637
-2579

9's complement

```
  4637
+ 7420
 12057
    + 1
  2058
```

10's complement

```
   4637
 + 7421
  12058
  = 2058
```

```
101000
- 10101
```

1's complement

```
  + 101000
    101010
  1010010
          + 1
    010011
```

2's complement

```
   101000
  + 10101
remove 101 00 11 = 010011
```

Date 18 - 10 - 22

Page No.

Binary Code

BCD - Binary Coded Decimal

BCD = binary (D ≤ 9)

Decimal	Binary	BCD
0	0000	0000
1	0001	0001
2	0010	0010
3	0011	0011
4	0100	0100
5	0101	0101
6	0110	0110
7	0111	0111
8	1000	1000
9	1001	1001
10	1010	00010000

* x decimal - $4x$ BCD

368 → BCD → 0011 0110 1000

457 → 0100 0101 0111

589 → 0101 1000 1001

448 → 0100 0100 1000

```
 448   →   0100  0100   1000
+589      + 0101  1000   1001
1037        1010  1101   10001
          + 0110  0110   0110
        00010000  10011  0111
           1  0   3      7

+184      0001   1000    0100
 576    + 0101   0111    0110
 760      0111  10000    1010
                + 0110   + 0110
                  0110   10000
            7     6      0
```

Decimal	BCD (8421)	2421	Excess-3 (+3)	84-2-1
0	0000	0000	0011	0000
1	0001	0001	0100	0111
2	0010	0010	0101	0110
3	0011	0011	0110	0101
4	0100	0100	0111	0100
5	0101	0101	1000	1011
6	0110	0110	1001	1010
7	0111	0111	1010	1001
8	1000	1110	1011	1000
9	1001	1111	1100	1111

* 2421, Excess-3 are self complimentary code

Self-complimentary codes
Such codes have a property that 9's comp of decimal no can be obtained directly by 1's compliment of binary number

* 2421, 84-2-1, BCD are weighted code

Gray code
* Used in minimization technique

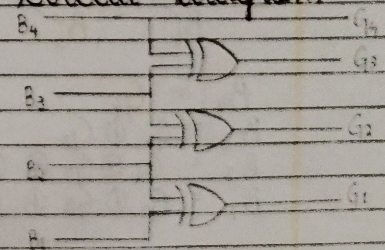
$B \rightarrow B_4 \quad B_3 \quad B_2 \quad B_1$ (⊕ ⊕ ⊕ → XOR operation)

$G \rightarrow G_4 \quad G_3 \quad G_2 \quad G_1$

$B \rightarrow 1\ 0\ 0\ 1$
$G \rightarrow 1\ 1\ 0\ 1$

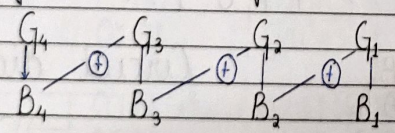
$G_4 = B_4$
$G_3 = B_4 \oplus B_3$
$G_2 = B_3 \oplus B_2$
$G_1 = B_2 \oplus B_1$

Circuit diagram



Decimal	Binary	Gray
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001
15	1111	1000
16	10000	11000

Gray to Binary

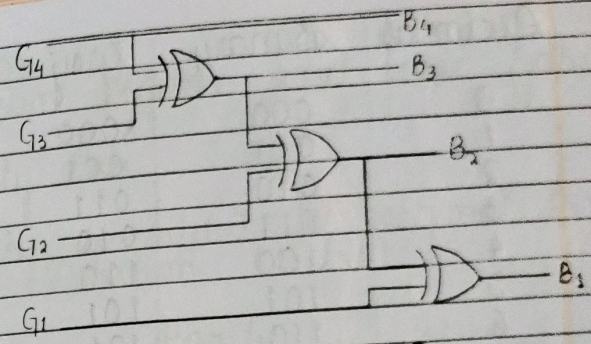


$B_4 = G_4$

$B_3 = G_4 \oplus G_3$

$B_2 = G_4 \oplus G_3 \oplus G_2$

$B_1 = G_4 \oplus G_3 \oplus G_2 \oplus G_1$



Decimal	Gray	Binary
0	0000	0000
1	0001	0001
2	0011	0010
3	0010	0011
4	0110	0100
5	0111	0101
6	0101	0110
7	0100	0111
8	1100	1000
9	1101	1001
10	1111	1010
11	1110	1011
12	1010	1100
13	1011	1101
14	1001	1110
15	1000	1111
16	11000	10000

Date 19-10-22
Page No.

Decimal	Binary	Gray
0	000	000
1	001	001
2	010	011
3	011	010
4	100	110
5	101	111
6	110	101
7	111	100

Decimal	Gray	Binary
0	00	00
1	01	01
2	11	10
3	10	11

Ch-02
Boolean Algebra and Logic Gates

Parity Bits

* Parity is added to original binary
* Error Detection (1-bit)

Parity

Satisfied - 0
Not satisfied - 1

Even
* 1's even P = 0
* 1's odd P = 1

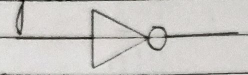
Odd
* 1's odd P = 0
* 1's even P = 1

Even parity table - Odd parity table

B_3	B_2	B_1	P (Even)	P (Odd)
0	0	0	0	1
0	0	1	1	0
0	1	0	1	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	0

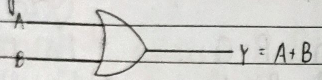
Boolean Algebra & Logic Gates

NOT gate : $A = \bar{A}$



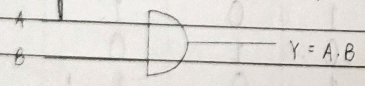
A	$\bar{A}$
0	1
1	0

OR gate : $A + B$ $(A \cup B)$



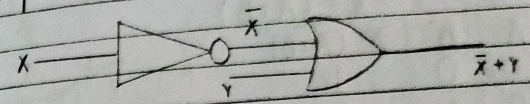
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

AND gate : $A \cdot B$ $(A \cap B)$

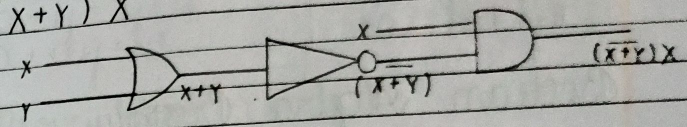


A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

1. $\bar{X} + Y$



2. $(X+Y)\,X$



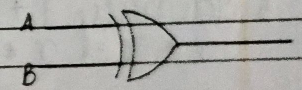
20-10-22

* NAND and NOR are universal gates

XOR gate

$X = A \oplus B = AB + AB$

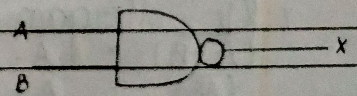
A	B	$X = A \oplus B$
0	0	0
0	1	1
1	0	0
1	1	1



NAND gate

$X = (\overline{A \cdot B})$

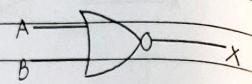
A	B	$X = \overline{A \cdot B}$
0	0	1
0	1	1
1	0	1
1	1	0



NOR Gate

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

A	B	$X = \overline{(A+B)}$
0	0	1
0	1	0
1	0	0
1	1	0



Boolean Algebra Properties

1. Closure $\{+, \cdot\}$
2. Commutative $\{+, \cdot\}$
3. Associative $\{+, \cdot\}$
4. Inverse $\{+, \cdot\}$ $\quad x * y = e$
5. Identity $\{+, \cdot\}$

   $e * x = x * e = x$

   $0 + x = x + 0 = x$

   $1 \cdot x = x \cdot 1 = x$
6. Distributive $\{+, \cdot\}$

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

   $A + (B \cdot C) = (A + B) \cdot (A + C)$
7. De Morgan's Law

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

   $\overline{(AB)} = \bar{A} + \bar{B}$

1. $x + 0 = x$
2. $x + \bar{x} = 1$
3. $x + x = x$
4. $x + 1 = 1$
5. $\overline{(\bar{x})} = x$
6. $x + y = y + x$
7. $x + xy = x$

1. $x \cdot 1 = x$
2. $x \cdot \bar{x} = 0$
3. $x \cdot x = x$
4. $x \cdot 1 = x$
6. $x \cdot y = y \cdot x$
7. $x(x + y) = x$

Prove that $x + x = x$

$(x + x) \cdot 1$

$\Rightarrow (x + x) \cdot (x + \bar{x})$

$\Rightarrow x + x \cdot \bar{x}$

$= x + 0 = x$

Prove that $x + 1 = 1$

$(x + 1) \cdot 1$

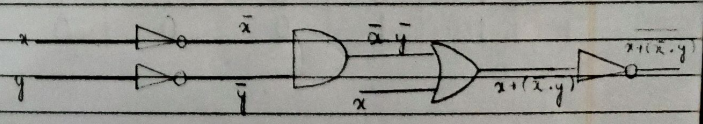
$= (x + 1)(x + \bar{x})$

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

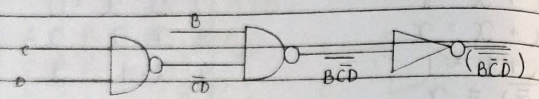
$= x + \bar{x}$

$= 1$

Q. $Y = \overline{x + (\bar{x} \cdot \bar{y})}$



$A = \overline{B\bar{C}\bar{D}}$



Boolean Function

$F = AB + BC$

$\cdot$

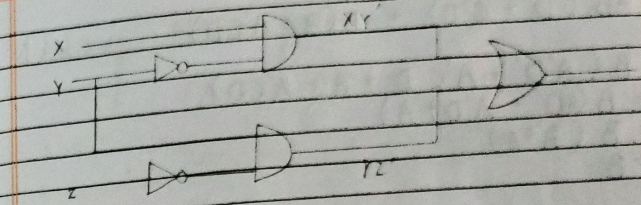
$'$

$\oplus$

F can be expressed using
- truth table
- schematic / logic gate diagram

$F = XY' + YZ'$

X	Y	Z	XY'	YZ'	F
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	0	1	1
0	1	1	0	0	0
1	0	0	1	0	1
1	0	1	1	0	1
1	1	0	0	1	1
1	1	1	0	0	0



Simplification of boolean function

$F = \underset{T_1}{X'Y} + \underset{T_2}{Y'Z} + \underset{T_3}{Z'}$

Terms : 3
Literals : 5

Date 27-10-22
Page No.

Q. $A'B(D'+C'D) + B(A+A'CD)$

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

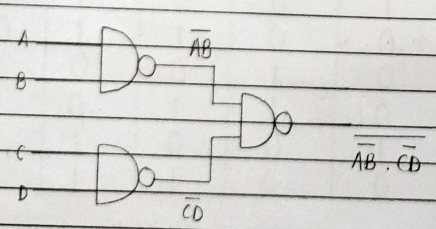
Q. $A'C' + ABC + AC'$

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

NAND : SOP
NOR : POS

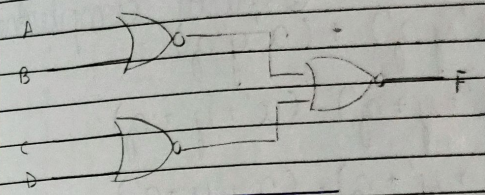
01-11-22

$F = AB + CD$ NAND
$= \overline{\overline{AB + CD}}$
$= (\overline{\overline{AB} \cdot \overline{CD}})$



Using NOR gate

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



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

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

Canonical and standard form

Canonical form
* minterm
* maxterm

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

1. Using de-morgan
$(F)' = (x'yz' + x'y'z)'$
$= (x'yz')' \cdot (x'y'z)'$
$= (x + y' + z) \cdot (x + y + z')$

2. Using direct method - Duality

① Operator change
② Literal complement

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

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

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

Q $F = x(y'z' + yz)$

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

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

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

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

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

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

Minterm: each minterm is always a standard prod. obtained from an AND term of n variables, with each variable is being primed when bit is 0 & unprimed when bit is 1

x	y	z	Term	Minterm	Term	Maxterm
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

$F = xy + y'z$

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

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

$= m_7 + m_6 + m_5 + m_1$

$= \Sigma m(1,5,6,7)$

$F = A + B'C$

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

$= AB'C + ABC' + AB'C + A'B'C + ABC + AB'C'$

$= m_5 + m_6 + m_5 + m_1 + m_7 + m_4$

$= m_1 + m_4 + m_5 + m_6 + m_7$

$= \Sigma m(1,4,5,6,7)$

Canonical form in boolean expression is sum of minterms or product of maxterm.

Maxterm: each maxterm is additive which is obtained from OR term of n variables, with each variable being primed when 1 & 0 when unprimed

2-11-22

$F = xy + y'z$

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

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

$= M_7 + M_6 + M_5 + M_1$

$= \Pi M(1, 5, 6, 7)$

$F = xy + x'z$

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

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

$= m_7 + m_6 + m_3 + m_1$

$= \Sigma m(1, 3, 6, 7)$

$F = xy + x'z$

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

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

$= (x'+y+zz')(x+yy'+z)(xx'+y+z)$

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

$= M_4 \cdot M_5 \cdot M_0 \cdot M_2$

$= \Pi(0, 2, 4, 5)$

Q. $F = x' + y(zx + z')$

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

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

$= m_3 + m_2 + m_1 + m_0 + m_7 + m_6 + m_2$

$F = \Pi(4, 5)$

K - map (Karnaugh)

1. It is a pictorial form of truth table.
2. It is a diagram made of square where each square represents minterm of the function which needs to be minimised.
3. It's a visual diagram of all possible ways a function can be represented.
4. The simplified expression produced by K-map is of two forms either SOP or POS.
5. K-map always gives you minimum number of gates with minimum number of inputs.

The distance from one square to another is always 1 bit change & it is reflected using Gray code.

2-bit K-map

2x2

x \ y	0	1
0	$x'y'$	$x'y$
1	xy'	xy

$F = \Sigma m(3)$
$= m_3$

x \ y	y'	y
x' 0		
x 1		1

$F = xy$

$F = \Sigma m(1,2,3)$
$= m_1 + m_2 + m_3$

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

$= x + y$

x \ y	0	1
0		1
1	1	1

$x'y + xy$
$= y$

$xy' + xy$
$= x$

3-variable K-map

x \ yz	y'z' 00	y'z 01	yz 11	yz' 10
x' 0				
x 1				

$F = \Sigma m(1,3)$
$= x'y'z + x'yz$
$= x'z$

F: x \ yz	00	01	11	10
0		1	1	
1				

$F = x'z$

No min possible

F: x \ yz	00	01	11	10
0			1	
1				1

$F = x'yz + xyz'$

x'	1			1
x				

$x'z$

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

y

3-11-22

$F = \Sigma m(1, 3, 6, 7)$

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

$x'z + xy$

$F(x, y, z) = \Sigma(2, 3, 4, 5)$
$= m_2 + m_3 + m_4 + m_5$

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

$x'y + xy'$

$F(x, y, z) = \Sigma(3, 4, 6, 7)$

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

$xz' + yz$

$F(x, y, z) = \Sigma(0, 2, 4, 5, 6)$

$x \backslash y$	00	01	11	10
0	1			1
1	1	1		1

$z' + xy'$

$F(x, y, z) = \Pi(0, 1, 2)$
$= \Sigma(3, 4, 5, 6, 7)$

$x \backslash y$	00	01	11	10
0			1	
1	1	1	1	1

$x + yz$

$F(x, y, z) = \Pi(4, 5, 6, 7)$
$= \Sigma(0, 1, 2, 3)$

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

x'

$F = A'C + A'B + AB'C + BC$

a) Express as sum of minterms
b) Find minimal SOP expression

$2^0 = 1 \Rightarrow 4$ blocks
$2^1 = 2 \Rightarrow 3$
$2^2 = 4 \Rightarrow 2$
$2^3 = 8 \Rightarrow 1$

⊙ 4-variable K-map

F \ ab \ cd	cd' 00	c'd' 01	c'd 11	cd 10
a'b' 00	0	1	3	2
a'b 01	4	5	7	6
ab 11	12	13	15	14
ab' 10	8	9	11	10

F \ ab \ cd	00	01	11	10
00			1	1
01				
11				
10			1	1

b'c

F \ ab \ cd	00	01	11	10
00	1			1
01				
11				
10	1			1

b'd'

F \ ab \ cd	00	01	11	10
00	1	1		1
01	1	1		1
11	1	1		1
10	1	1		1

c' + d'

F = Σ(0, 1, 2, 3, 5, 6, 11, 13)

F \ ab \ cd	cd' 00	c'd' 01	c'd 11	cd 10
a'b' 00	1	1	1	1
a'b 01		1		1
ab 11		1		
ab' 10			1	

a'b' + bc'd + a'cd' + b'cd

8-11-22

Don't care condition

Simplify the boolean expression
F(w, x, y, z) = Σ(1, 3, 7, 11, 15)
which has don't care condition of
d(w, x, y, z) = Σ(0, 2, 5)

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

Don't care
yz + w'x
or
yz + w'z

yz + w'x'z

$F(A,B,C,D) = \Sigma(0,6,8,13,14)$
$d(A,B,C,D) = \Sigma(2,4,10)$

AB \ CD	00	01	11	10
00	1			X
01	X			1
11		1		1
10	1			X

$CD' + A'C'D' + B'D' + ABC'D$

9-11-22

SOP - $xy + yz + zw$

$xy(z+z')(w+w') + yz(x+x')(w+w') + zw(x+x')(y+y')$

POS using K-map

$F(A,B,C,D) = \Sigma(0,1,2,5,8,9,10)$

AB \ CD	C'D' 00	C'D 01	CD 11	CD' 10
A'B' 00	1 (0)	1 (1)	0 (3)	1 (2)
A'B 01	0 (4)	1 (5)	0 (7)	0 (6)
AB 11	0 (12)	0 (13)	0 (15)	0 (14)
AB' 10	1 (8)	1 (9)	0 (11)	1 (10)

$F = B'D' + A'C'D + B'C'$

$F' = AB + CD + BD'$

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

$F(A,B,C,D) = \Sigma(0,1,2,5,8,9,10)$
$= \Pi(3,4,6,7,11,12,13,14,15)$

AB \ CD	C+D 00	C+D' 01	C'+D' 11	C'+D 10
~~A+B 00~~	0	1	0 (3)	2
~~A+B' 01~~	0 (4)	5	0 (7)	0 (6)
~~A'+B' 11~~	0 (12)	0 (13)	0 (15)	0 (14)
~~A'+B 10~~	8	9	0 (11)	10

Date 15-11-22
Page No.

XOR Gate

$F = A \oplus B = AB' + A'B$

A \ B	0	1
0		1
1	1	

A	B	F
0	0	0
0	1	1
1	0	1
1	1	0

X-NOR Gate

$F = (A \oplus B)' = (A \odot B)$

A \ B	0	1
0	1	
1		1

A	B	F
0	0	1
0	1	0
1	0	0
1	1	1

$F = A'B' + AB$

$(A \oplus B)' = (A'B + AB')'$
$= (A'B)'(AB')'$
$= (A+B')(A'+B)$
$= AB + A'B'$

Identities

1. $x \oplus 0 = x$
2. $x \oplus 1 = x'$
3. $x \oplus x = 0$
4. $x \oplus x' = 1$
5. $(x \oplus y)' = x \oplus y' = x' \oplus y$
6. $A \oplus B = B \oplus A$
7. $(A \oplus B) \oplus C = A \oplus (B \oplus C)$

Date
Page No.

$A \oplus (B \oplus C) = (A \oplus B) \oplus C$
$= A \oplus (B'C + BC')$
$= A'(B'C + BC') + A(B'C + BC')'$
$= A'(B'C + BC') + A(BC + B'C')$
$= A'B'C + A'BC' + ABC + AB'C'$
$= m_1 + m_2 + m_7 + m_4$

A \ BC	B'C' 00	B'C 01	BC 11	BC' 10
A' 0		1 (1)		1 (2)
A 1	1 (4)		1 (7)	

$A \oplus B \oplus C \oplus D$
$= (A \oplus B) \oplus (C \oplus D)$
$= (A'B + AB') \oplus (C'D + CD')$
$= (A'B + AB')'(C'D + CD') + (A'B + AB')(C'D + CD')'$
$= (AB + A'B')(C'D + CD') + (A'B + AB')(CD + C'D')$
$= ABC'D + ABCD' + A'B'C'D + A'B'CD' +$
$A'BCD + A'BC'D' + AB'CD + AB'C'D'$
$= m_{13} + m_{14} + m_1 + m_2 + m_7 + m_4 + m_{11} +$
m_8

AB \ CD	C'D' 00	C'D 01	CD 11	CD' 10
A'B' 00		1 (1)		1 (2)
A'B 01	1 (4)		1 (7)	
AB 11		1 (13)		1 (14)
AB' 10	1 (8)		1 (11)	

Even parity - XOR
Odd parity - XNOR

Even parity

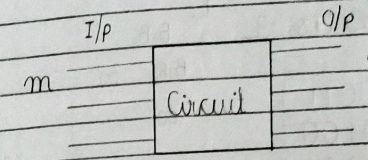
A	B	C	D	F
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

F \ CD, AB	00	01	11	10
00	0	1 1	3	1 2
01	1 4	5	1 7	6
11	12	1 13	15	1 14
10	1 8	9	1 11	10

$F = A \oplus B \oplus C \oplus D$

Date 16 - 11 - 22

Combinational Circuit



$0 < m < \infty$
$0 < n < \infty$

Design Procedure

1. From specification of circuit determine required no. of I/P and O/P and assign symbol to each
2. Derive truth table that defines relationship betⁿ input and output
3. Obtain boolean funⁿ for each O/P in terms of I/P
4. Draw logic diagram and verify the correctness using truth table

Code Conversion
BCD → Excess-3
Excess-3 → BCD
Binary → Gray
Gray → Binary

BCD to Ex-3

BCD	Excess-3
$B_1\ B_2\ B_3\ B_4$	$F_1\ F_2\ F_3\ F_4$
0000	0011
0001	0100
0010	0101
0011	0110
0100	0111
0101	1000
0110	1001
0111	1010
1000	1011
1001	1100

E_1

B_1B_2 \ B_3B_4	00	01	11	10
00				
01		1	1	1
11	X	X	X	X
10	1	1	X	X

$E_1 = B_1 + B_2B_4 + B_2B_3$

E_2

B_1B_2 \ B_3B_4	$B_2'B_4'$ 00	$B_3'B_4$ 01	B_3B_4 11	B_3B_4' 10
$B_1'B_2'$ 00		1	1	1
$B_1'B_2$ 01	1			
B_1B_2 11	X	X	X	X
B_1B_2' 10		1	X	X

$E_2 = B_2B_3'B_4' + B_2'B_4 + B_2'B_3$

E_3

B_1B_2 \ B_3B_4	00	01	11	10
00	1		1	
01	1		1	
11	X	X	X	X
10	1		X	X

$E_3 = B_3'B_4' + B_3B_4$

E_4

B_1B_2 \ B_3B_4	00	01	11	10
00	1			1
01	1			1
11	X	X	X	X
10	1		X	X

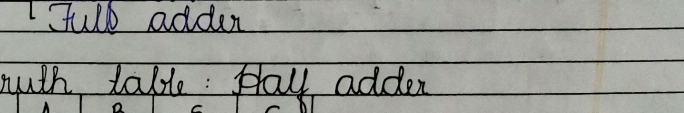
$E_4 = B_4'$

Adder

- Half adder
- Full adder

Truth table : Half adder

A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

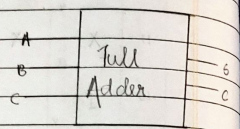


$S = A'B + AB' = A \oplus B$

$C = AB$

Full adder

A	B	C	S	C
0	0	0	0	0
0	0	1	1	0
0	1	0	1	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



S

A \ BC	00 (B'C')	01 (B'C)	11 (BC)	10 (BC')
A' 0	0	1	3	1 2
A 1	1 4	5	1 7	6

$S = A'B'C + A'BC' + AB'C' + ABC$
$= A \oplus B \oplus C$

C

A \ BC	00 (B'C')	01 (B'C)	11 (BC)	10 (BC')
A' 0	0	1	1 3	2
A 1	4	1 5	1 7	1 6

$C = BC + AC + AB$

Excess -3 ⟶ BCD

Excess -3				BCD			
E_1	E_2	E_3	E_4	B_1	B_2	B_3	B_4
0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	1
0	1	0	1	0	0	1	0
0	1	1	0	0	0	1	1
0	1	1	1	0	1	0	0
1	0	0	0	0	1	0	1
1	0	0	1	0	1	1	0
1	0	1	0	0	1	1	1
1	0	1	1	1	0	0	0
1	1	0	0	1	0	0	1

B_1

E_1E_2 \ E_3E_4	00	01	11	10
00	X 0	X 1	3	X 2
01	4	5	7	6
11	1 12	X 13	X 15	X 14
10	8	9	1 11	10

$B_1 = E_1E_2 + E_1E_3E_4$

B_2

E_1E_2 \ E_3E_4	00	01	11	10
00	X	X		X
01			1	
11		X	X	X
10	1	1		1

$B_2 = E_2'E_4' + E_2'E_3' + E_2E_3E_4$

B_3

E_1E_2 \ E_3E_4	00	01	11	10
00	X	X		X
01		1		1
11		X	X	X
10		1		1

$B_3 = E_3'E_4 + E_3E_4'$

B_4

E_1E_2 \ E_3E_4	00	01	11	10
00	X	X		X
01	1			1
11	1	X	X	X
10	1			1

$B_4 = E_4'$

Binary → Gray

Binary				Gray			
B_1	B_2	B_3	B_4	G_1	G_2	G_3	G_4
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1

G_3

B_1B_2 \ B_3B_4	00	01	11	10
00			1	1
01	1	1		
11	1	1		
10			1	1

$G_3 = B_2B_3' + B_2'B_3$

G_2

B_1B_2 \ B_3B_4	00	01	11	10
00				
01	1	1	1	1
11				
10	1	1	1	1

$G_2 = B_1'B_2 + B_1B_2'$

G_1

B_1B_2 \ B_3B_4	00	01	11	10
00				
01				
11	1	1	1	1
10	1	1	1	1

$G_1 = B_1$

G_4

B_1B_2 \ B_3B_4	00	01	11	10
00		1		1
01		1		1
11		1		1
10		1		1

$G_4 = B_3'B_4 + B_3B_4'$

Gray				Binary			
G_1	G_2	G_3	G_4	B_1	B_2	B_3	B_4
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	1
0	1	0	1	0	1	1	0
0	1	1	0	0	1	0	0
0	1	1	1	0	1	0	1
1	0	0	0	1	1	1	1
1	0	0	1	1	1	1	0
1	0	1	0	1	1	0	0
1	0	1	1	1	1	0	1
1	1	0	0	1	0	0	0
1	1	0	1	1	0	0	1
1	1	1	0	1	0	1	1
1	1	1	1	1	0	1	0

B_1

G_1G_2 \ G_3G_4	00	01	11	10
00				
01				
11	1	1	1	1
10	1	1	1	1

$B_1 = G_1$

B_2

G_1G_2 \ G_3G_4	00	01	11	10
00				
01	1	1	1	1
11				
10	1	1	1	1

$B_2 = G_1'G_2 + G_1G_2'$

B_3

G_1G_2 \ G_3G_4	00	01	11	10
00			1	1
01	1	1		
11			1	1
10	1	1		

$B_3 = G_1 \oplus G_2 \oplus G_3$

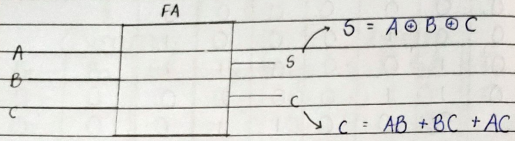
B_4

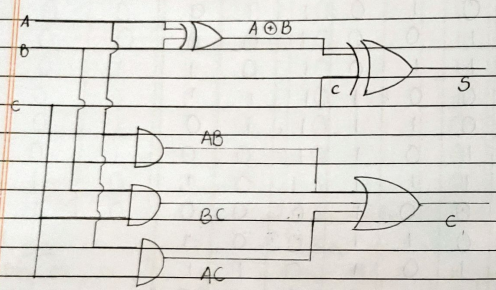
G_1G_2 \ G_3G_4	00	01	11	10
00		1		1
01	1		1	
11		1		1
10	1		1	

$B_4 = G_1 \oplus G_2 \oplus G_3 \oplus G_4$

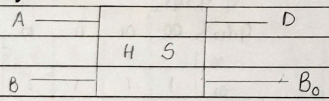
Date 17-11-22
Page No.

Full Adder





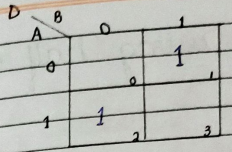
Half Subtractor



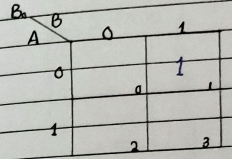
A - B = D

I/P		O/P	
A	B	D	B_0
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

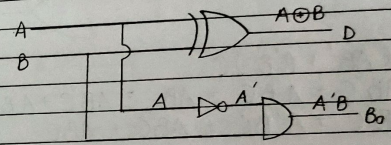
Date
Page No.



$D = A'B + AB'$
$= A \oplus B$



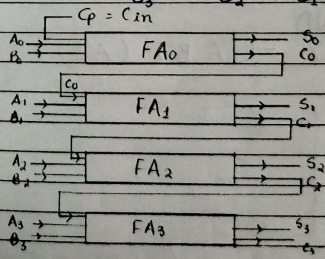
$B_0 = A'B$



n-bit parallel adder

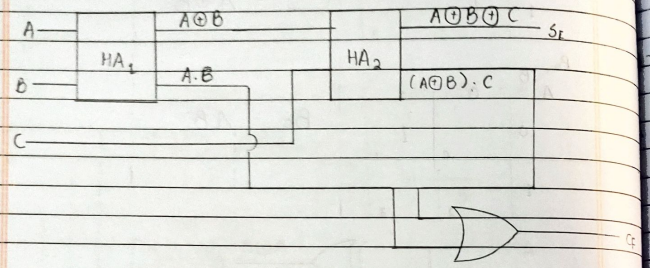
$$\begin{array}{cccccc} & C_3 & C_2 & C_1 & C_0 & C_p \\ & A_3 & A_2 & A_1 & A_0 & \\ + & B_3 & B_2 & B_1 & B_0 & \\ & S_3 & S_2 & S_1 & S_0 & \\ & C_3 & C_2 & C_1 & C_0 & \end{array}$$

$C_p = 0$



Date 22-11-22
Page No.

Design full adder using half adder



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

$C_F = AB'C + A'BC + AB = AB'C + A'BC + AB(C+C')$

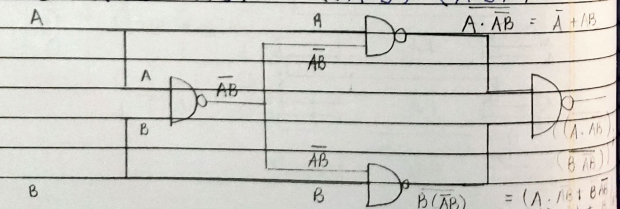
$= AB'C + A'BC + ABC + ABC'$

C_F \ A \ BC	00	01	11	10
0	0	1	1 (3)	2
1	1 (4)	1 (5)	1 (7)	1 (6)

$C_F = AB + BC + AC$

XOR using NAND

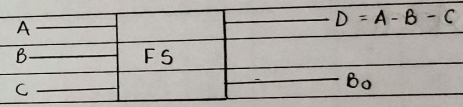
$A \oplus B = ((A \cdot B') + A'B))' = ((A \cdot B')'(A'B)')'$



Date
Page No.

Full Subtractor

A	B	C	D	B_0
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1



D \ A \ BC	00	01	11	10
0	0	1 (1)	3	1 (2)
1	1 (4)	5	1 (7)	6

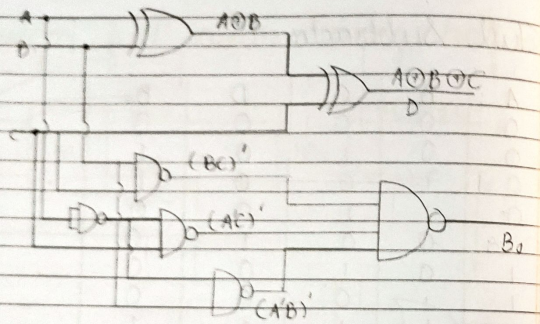
$D = A'B'C + A'BC' + AB'C' + ABC$

$= A \oplus B \oplus C$

B_0 \ A \ BC	00	01	11	10
0	0	1 (1)	1 (3)	1 (2)
1	4	5	1 (7)	6

$B_0 = BC + A'C + A'B$

$B_0 = ((BC + A'C + A'B)')' = ((BC)'(A'C)'(A'B)')'$



23-11-22

BCD Adder

Binary sum

D	K	Z_8	Z_4	Z_2	Z_1
0	0	0	0	0	0
1	0	0	0	0	1
2	0	0	0	1	0
3	0	0	0	1	1
4	0	0	1	0	0
5	0	0	1	0	1
6	0	0	1	1	0
7	0	0	1	1	1
8	0	1	0	0	0
9	0	1	0	0	1
10	0	1	0	1	0
11	0	1	0	1	1
12	0	1	1	0	0
13	0	1	1	0	1
14	0	1	1	1	0
15	0	1	1	1	1
16	1	0	0	0	0
17	1	0	0	0	1
18	1	0	0	1	0
19	1	0	0	1	1

BCD sum

C_N	S_8	S_4	S_2	S_1
0	0	0	0	0
0	0	0	0	1
0	0	0	1	0
0	0	0	1	1
0	0	1	0	0
0	0	1	0	1
0	0	1	1	0
0	0	1	1	1
D	1	0	0	0
0	1	0	0	1
1	0	0	0	0
1	0	0	0	1
1	0	0	1	0
1	0	0	1	1
1	0	1	0	0
1	0	1	0	1
1	0	1	1	0
1	0	1	1	1
1	1	0	0	0
1	1	0	0	1

C_N Z_8Z_4 / Z_8Z_4 — K = 0

	00	01	11	10
00	0	1	3	2
01	4	5	7	6
11	1 (12)	1 (13)	1 (15)	1 (14)
10	8	9	1 (11)	1 (10)

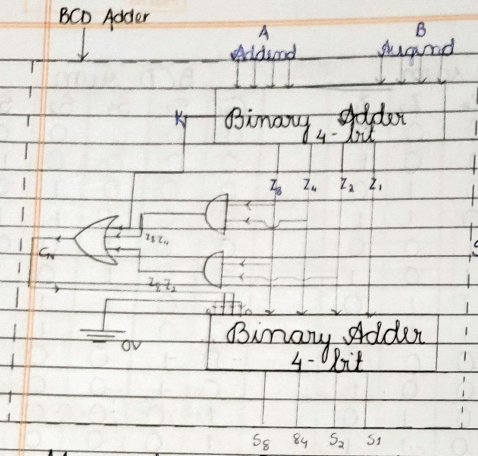
$K Z_8 Z_4 Z_2 Z_1$ MOB Z_8Z_4 / Z_8Z_4 — K = 1

	00	01	11	10
00	1 (8)	1 (9)	1 (11)	1 (10)
01	X (12)	X (13)	X (15)	X (14)
11	X (4)	X (5)	X (7)	X (6)
10	X (0)	X (1)	X (3)	X (2)

$K'(Z_8Z_4 + Z_8Z_2)$ $\qquad$ K

$C_N = K + Z_8Z_4 + Z_8Z_2$

Date 24 - 11 - 22
Page No.

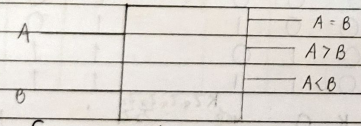


$C_N = 0$ (
$S_8 S_4 S_2 S_1 =$
$Z_8 Z_4 Z_2 Z_1$

$C_N = 1$
$S_8 S_4 S_2 S_1 + 0110$
$= Z_8 Z_4 Z_2 Z_1$

Magnitude Comparator

A combinational circuit which compa
two binary no and provides
result (>, <, =)



1-bit Comparator

A	B	A = B	A > B	A < B
0	0	1	0	0
0	1	0	0	1
1	0	0	1	0
1	1	1	0	0

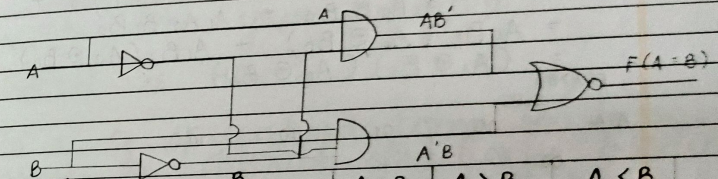
Date
Page No.

$F(A = B) = A'B' + AB$
$= (A \odot B)$

$F(A > B) = AB'$

$F(A < B) = A'B$

$F(A > B) + F(A < B) = A'B + AB' = A \oplus B$
$= [F(A = B)]'$



A_1	A_0	B_1	B_0	A = B	A > B	A < B
0	0	0	0	1	0	0
0	0	0	1	0	0	1
0	0	1	0	0	0	1
0	0	1	1	0	0	1
0	1	0	0	0	1	0
0	1	0	1	1	0	0
0	1	1	0	0	0	1
0	1	1	1	0	0	1
1	0	0	0	0	1	0
1	0	0	1	0	1	0
1	0	1	0	1	0	0
1	0	1	1	0	0	1
1	1	0	0	0	1	0
1	1	0	1	0	1	0
1	1	1	0	0	1	0
1	1	1	1	1	0	0

Date
Page No.

F(A = B)

A_1A_0 \ B_1B_0	00	01	11	10
00	1			
01		1		
11			1	
10				1

$F(A = B) = A_1'A_0'B_1'B_0' + A_1A_0B_1'B_0 + A_1A_0B_1B_0 + A_1A_0'B_1B_0'$

$= A_1'B_1'(A_0 \oplus B_0)' + A_1B_1(A_0 \oplus B_0)'$

$= (A_1 \oplus B_1)'(A_0 \oplus B_0)'$

F(A>B)

A_1A_0 \ B_1B_0	00	01	11	10
00				
01	1			
11	1	1		1
10	1	1		

$F(A > B) = A_1B_1' + A_0B_1'B_0' + A_1A_0B_0'$

F(A<B)

A_1A_0 \ B_1B_0	00	01	11	10
00		1	1	1
01			1	1
11				
10			1	

$F(A < B) = A_1'B_1 + A_1'A_0'B_0 + A_0'B_1B_0$

Date
Page No.

What is the requirement of software simulation?

A software simulation is model of your software that allows you to demonstrate its key functions and operations

Using software simulations, you can show your customers how everything works in your program

What is HDL?

It's a specialized computer language used to describe structure & behavior of electronic circuits

Verilog design suite

Abstraction levels

Behavioral

Dataflow

HDL Operators

Symbol	Operation
+	binary addition
-	binary subtraction
&	bitwise AND
|	bitwise OR
^	bitwise XOR
~	bitwise NOT
==	equality
&&	logical AND
||	logical OR
!	logical NOT
>	greater than
<	less than
{}	concatenation
?:	conditional

$O = S_1 S_0' D + S_1' S_0 D + ACE$

$E = A + BC + B'D$

$F = B'C + BC'D'$

```
module Circuit_Boolean_CA (E, F, A, B, C, D);
output E, F;
input A, B, C, D;
assign E = A || (B && C) || ((!B) && D);
assign F = ((!B) && C) || (B && (!C) && (!D));
end module
```

```
module Circuit_Boolean_CA (O, S1, S0, A, C, D, E);
output O;
input S1, S0, A, C, D, E;
assign O = (S1 && (!S0) && D) || ((!S1) && (S0) && (D))
          || (A && C && E);
end module;
```

Steps to execute any program in Vivado

1. Double click on vivado s/w
2. Create project, Next, write project name, project type: RTL (Register transfer level), project, Next
3. Add sources, target board, simulator language: Verilog
4. Add constraints: nil, Next
5. Default part: select board: get board
6. Add sources: add or create design sources
7. Create file, File name: half_adder, finish
8. define module: i/o port defn, ok, yes, save
9. In sources: design sources half_adder module is created (half_adder.v)
10. Double click on that, Program editor will come with your prog
11. Verilog code is the same, see for errors. Write test bench code: it gives stimulus to module & module gives o/p
12. Add sources: add or create simulation sources
13. Write file name: half_adder_tb, Ok, finish
14. Click on simulation sources: click on half_adder_tb.v
15. Write test bench code

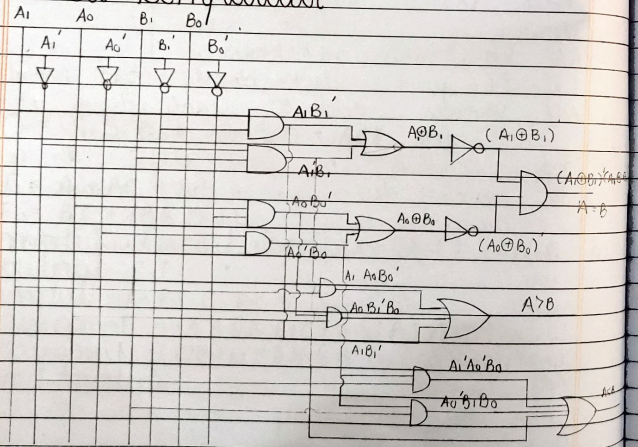
$A_1B_1' + A_1A_0B_1' + A_0B_1'B_0'$
$A_1'B_1 + A_1'A_0'B_0 + A_0'B_1B_0$
$(A_1 \oplus B_1)' (A_0 \oplus B_0)'$
$(A_1B_1' + A_1B_0)' (A_0B_1' + A_0'B_0)'$

Date
Page No.

16. Save
17. Run simulation
18. O/P is displayed in next window
19. To see schematic click RTL analysis
click schematic ok
20. Schematic is displayed in next window
21. Save a image to be pasted
1. Design
2. Test bench
3. Result
4. Schematic

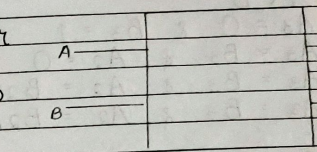
29-11-22

2-bit comparator



Date
Page No.

4-bit comparator



$A = A_3\ A_2\ A_1\ A_0$
$B = B_3\ B_2\ B_1\ B_0$

$A = B \Rightarrow A_3 = B_3\ \&\ A_2 = B_2\ \&\ A_1 = B_1\ \&\ A_0 = B_0$
$F = (A_3 \oplus B_3)'(A_2 \oplus B_2)'(A_1 \oplus B_1)'(A_0 \oplus B_0)'$

$A > B$ $\quad A = A_1A_0$
$\quad B = B_1B_0$

1. $A_1 > B_1 \quad - \quad A_1B_1'$
2. $A_1 = B_1\ \&\ A_0 = 1,\ B_0 = 0 \quad - \quad A_1B_1A_0B_0'$
$\quad A_1'B_1'A_0B_0'$

$A < B$
1. $A_1 < B_1 \quad A_1'B_1$
2. $A_1 = B_1,\ A_0 = 0,\ B_0 = 1 \quad - \quad A_1B_1A_0'B_0$
$\quad A_1'B_1'A_0'B_0$

For 4-bit comparator

This comparator is used to compute 2 binary no of 4 bits and provide 3 condition in output i.e. $A=B$, $A>B$, $A<B$

$A > B$
1. $A_3 = 1,\ B_3 = 0$
2. $A_3 = B_3\ \&\ A_2 = 1,\ B_2 = 0$
3. $A_3 = B_3,\ A_2 = B_2,\ A_1 = 1,\ B_1 = 0$
4. $A_3 = B_3,\ A_2 = B_2,\ A_1 = B_1,\ A_0 = 1,\ B_0 = 0$

$A > B = A_3B_3' + (A_3 \oplus B_3)' A_2B_2' + (A_3 \oplus B_3)'(A_2 \oplus B_2)' A_1B_1' + (A_3 \oplus B_3)'(A_2 \oplus B_2)'(A_1 \oplus B_1)' A_0B_0'$

Date
Page No.

$A < B$

1. $A_3 = 0$ & $B_3 = 1$
2. $A_3 = B_3$ & $A_2 = 0$, $B_2 = 1$
3. $A_3 = B_3$ & $A_2 = B_2$ & $A_1 = 0$, $B_1 = 1$
4. $A_3 = B_3$ & $A_2 = B_2$ & $A_1 = B_1$ & $A_0 = 0$, B_0

$A = B$

$A_3 = B_3$ & $A_2 = B_2$ & $A_1 = B_1$ & $A_0 = B_0$

30-11-22 $F(A = B)$

$F = (A_3 \oplus B_3)'(A_2 \oplus B_2)'(A_1 \oplus B_1)'(A_0 \oplus B_0)'$

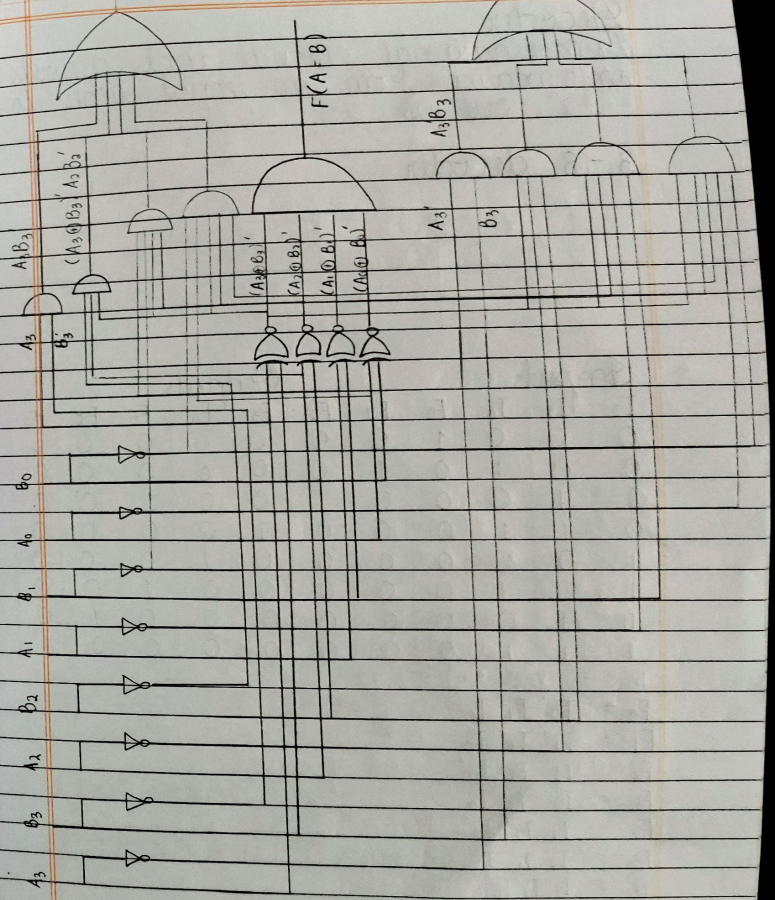
$F(A > B)$

$F = A_3B_3' + (A_3 \oplus B_3)' A_2B_2' + (A_3 \oplus B_3)'(A_2 \oplus B_2)'$
$+ (A_3 \oplus B_3)'(A_2 \oplus B_2)'(A_1 \oplus B_1)' A_0B_0'$

$F(A < B)$

$F = A_3'B_3 + (A_3 \oplus B_3)' A_2'B_2 + (A_3 \oplus B_3)'(A_2 \oplus B_2)' A_1'B_1$
$+ (A_3 \oplus B_3)'(A_2 \oplus B_2)'(A_1 \oplus B_1)' A_0'B_0$

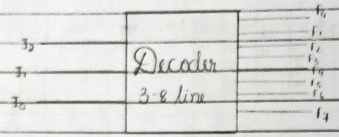
Date
Page No.



Decoder

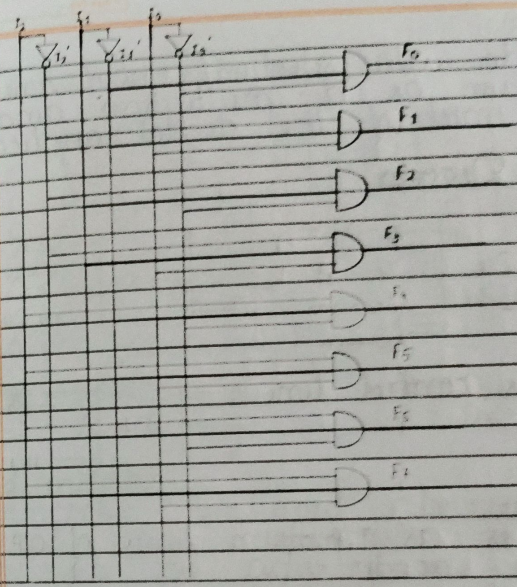
Combinational circuit that converts information from n-input lines to 2^n output

3 - 8 decoder



Inputs			Outputs							
I_2	I_1	I_0	F_0	F_1	F_2	F_3	F_4	F_5	F_6	F_7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

$F_0 = I_2' I_1' I_0'$
$F_1 = I_2' I_1' I_0$
$F_2 = I_2' I_1 I_0'$
$F_3 = I_2' I_1 I_0$
$F_4 = I_2 I_1' I_0'$
$F_5 = I_2 I_1' I_0$
$F_6 = I_2 I_1 I_0'$
$F_7 = I_2 I_1 I_0$



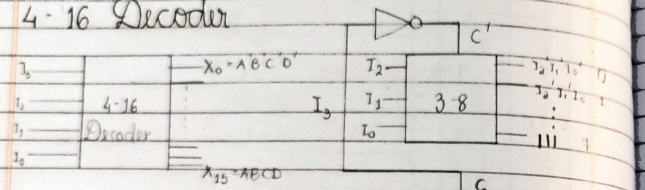
Output of basic decoder represents minterm of any SOP
A decoder with combination of external OR gate can form any combinational circuit

Full adder

$S = \Sigma(1, 2, 4, 7)$
$C = \Sigma(3, 5, 6, 7)$

Decoder with selection/control line can function as a multiplexer and used in most of the digital applications

4-16 Decoder



* MSB is control line

1-12-22

Encoder

2^n lines as I/P → n lines of O/P

m-n encoder

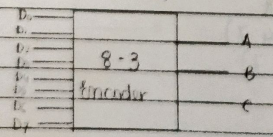
$m \leq 2^n$

4-2 encoder

8-3 encoder

16-4 encoder

In encoder output lines generate the binary code corresponding to input



I/P								O/P		
D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	A	B	C
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1

$A = \Sigma(4,5,6,7)$
$= D_4 + D_5 + D_6 + D_7$

$B = D_2 + D_3 + D_6 + D_7$

$C = D_1 + D_3 + D_5 + D_7$

Limitations

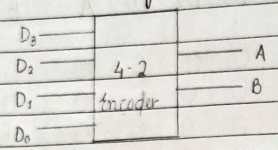
1. Encoder mentioned above can have only input active at a time. If two inputs are active output provides an undefined combination

2. To resolve this encoder must establish a input priority to ensure that one input is encoded

Priority Encoder

Encoder circuit which includes a priority function

4-2 Priority Encoder



I/P				O/P		Validity
D_3	D_2	D_1	D_0	A	B	
0	0	0	0	X	X	0
0	0	0	1	0	0	1
0	0	1	0	0	1	1
0	1	0	0	1	0	1
1	0	0	0	1	1	1

I/P				O/P	
D_3	D_2	D_1	D_0	A	B
0	0	0	0	X	X
0	0	0	1	0	0
0	0	1	X	0	1
0	1	X	X	1	0
1	X	X	X	1	1

D_3	D_2	D_1	D_0	A	B	V
0	0	0	0	X	X	0
0	0	0	1	0	0	1
0	0	1	0	0	1	1
0	0	1	1	0	1	1
0	1	0	0	1	0	1
0	1	0	1	1	0	1
0	1	1	0	1	0	1
0	1	1	1	1	0	1
1	0	0	0	1	1	1
1	0	0	1	1	1	1
1	0	1	0	1	1	1
1	0	1	1	1	1	1
1	1	0	0	1	1	1
1	1	0	1	1	1	1
1	1	1	0	1	1	1
1	1	1	1	1	1	1

A

D_3D_2 \ D_1D_0	00	01	11	10
00	X			
01	1	1	1	1
11	1	1	1	1
10	1	1	1	1

B

D_3D_2 \ D_1D_0	00	01	11	10
00	X		1	1
01				
11	1	1	1	1
10	1	1	1	1

$A = D_2 + D_3$

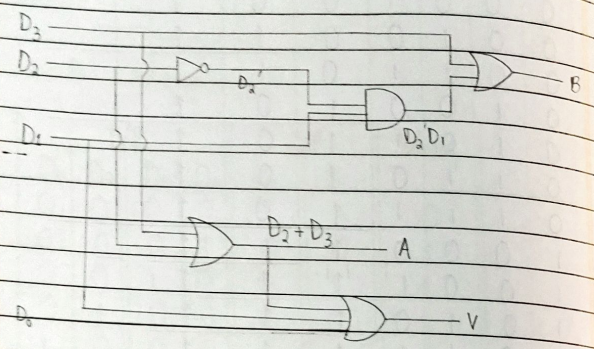
$B = D_3 + D_2 D_1$

V

D_3D_2 \ D_1D_0	00	01	11	10
00	X	1	1	1
01	1	1	1	1
11	1	1	1	1
10	1	1	1	1

$V = D_3 + D_2 + D_1 + D_0$

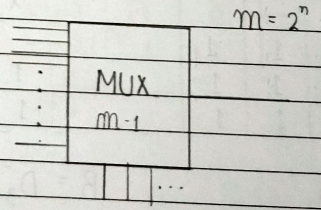
$A = D_2 + D_3$
$B = D_3 + D_2'D_1$
$V = D_3 + D_2 + D_1 + D_0$



6-12-22

Multiplexer

Combinational circuit which selects binary information from one of many input lines and direct to single output line

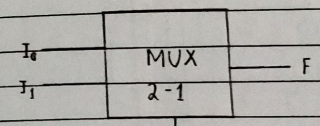


n Select line : 2^n - inputs
↳ depends on input

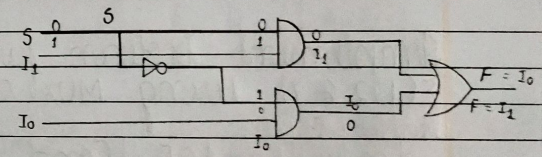
Selection of particular input line is controlled by set of selection line

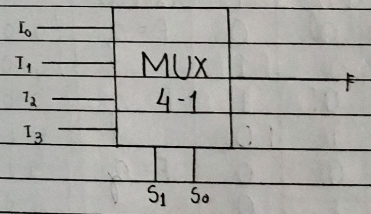
Any $m-1$ (2^n-1) MUX can be designed from $m-2^n$ decoder

Any MUX can be designed using AND gate with 2^n input lines and all output of AND gate are connected to OR gate.



S	F
S'	I_0
S	I



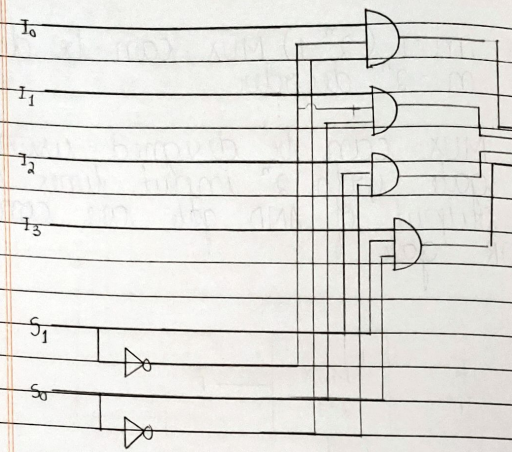


S_1	S_0	F
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

Complete 16×1 MUX with 2 no. 8×1 MUX and one 2×1 MUX

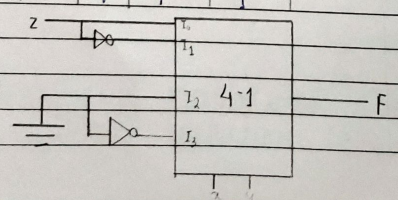
8-1 MUX

Date
Page No.



Implement boolean funⁿ $F(x,y,z) = \Sigma(1,2,6,7)$ using MUX circuit

x	y	z	F	Condⁿ	
0	0	0	0	F = z	x = 0, y = 0
0	0	1	1		
0	1	0	1	F = z'	x = 0, y = 1
0	1	1	0		
1	0	0	0	F = 0	x = 1, y = 0
1	0	1	0		
1	1	0	1	F = 1	x = 1, y = 1
1	1	1	1		



Implement boolean funⁿ using 8×1 MUX $F(A,B,C,D) = \Sigma(0,1,3,4,8,9,15)$

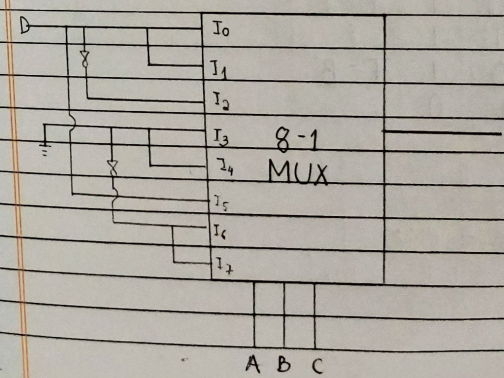
Implement full adder with two 8×1 MUX

" " 4×1 MUX

Date
Page No.

$F(A,B,C,D) = \Sigma(1,3,4,11,12,13,14,15)$

A	B	C	D	F	Condⁿ	
0	0	0	0	0	F = D	A = B = C = 0
0	0	0	1	1		
0	0	1	0	0	F = D	A = B = 0, C = 1
0	0	1	1	1		
0	1	0	0	1	F = D'	A = C = 0, B = 1
0	1	0	1	0		
0	1	1	0	0	F = 0	A = 0, B = C = 1
0	1	1	1	0		
1	0	0	0	0	F = 0	A = 1, B = C = 0
1	0	0	1	0		
1	0	1	0	0	F = D	A = C = 1, B = 0
1	0	1	1	1		
1	1	0	0	1	F = 1	A = B = 1, C = 0
1	1	0	1	1		
1	1	1	0	1	F = 1	A = B = C = 1
1	1	1	1	1		

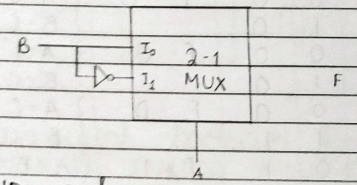


Date 7-12-22
Page No.

Solve XOR gate using 2x1 MUX

$F = A \oplus B = A'B + AB'$

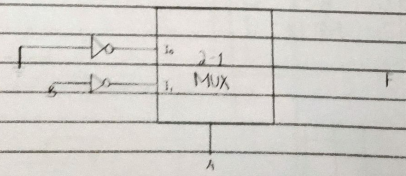
A	B	F	Cond^n
0	0	0	F = B
0	1	1	
1	0	1	F = B'
1	1	0	



NAND gate

$F = (A \cdot B)'$

A	B	F	Cond^n
0	0	1	F = 1
0	1	1	
1	0	1	F = B'
1	1	0	



Date
Page No.

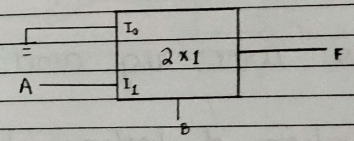
MSB bit as selection bit

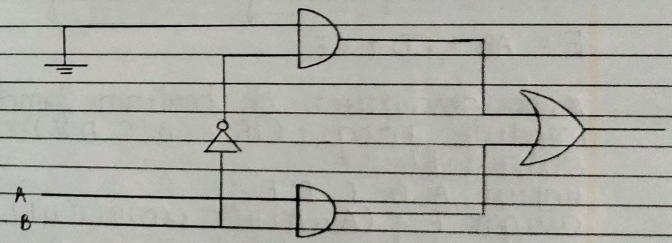
AND gate

$F = A \cdot B$

A	B	F
0	0	0
0	1	0
1	0	0
1	1	1

	0	1
	2	3
F	0	A





Date 8-12-22
Page No.

Verilog HDL
Most common hardware description language used by IC designers

Comment line
- //
- /* */

Keywords
assign, case, wire, reg, and, or, nand, module

HDL description of boolean function

$F = AB + CD + CE$

```
// HDL description of boolean function
module prog_1 (F, A, B, C, D, E);
output F;
input A, B, C, D, E;
assign F = (A && B) || (C && D) || (C && E);
endmodule
```

$F = AB + C(D+E)$

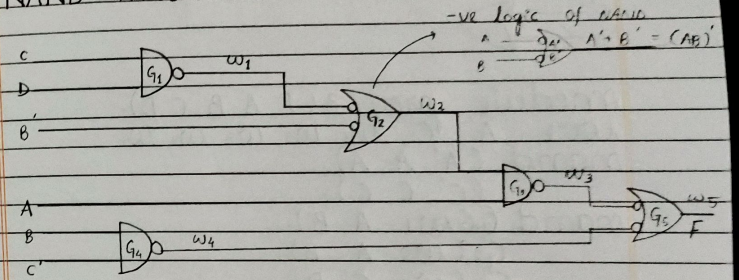
```
module prog_2 (F, A, B, C, D, E);
output F;
input A, B, C, D, E;
assign F = (A && B) || (C && (D || E));
endmodule
```

Date
Page No.

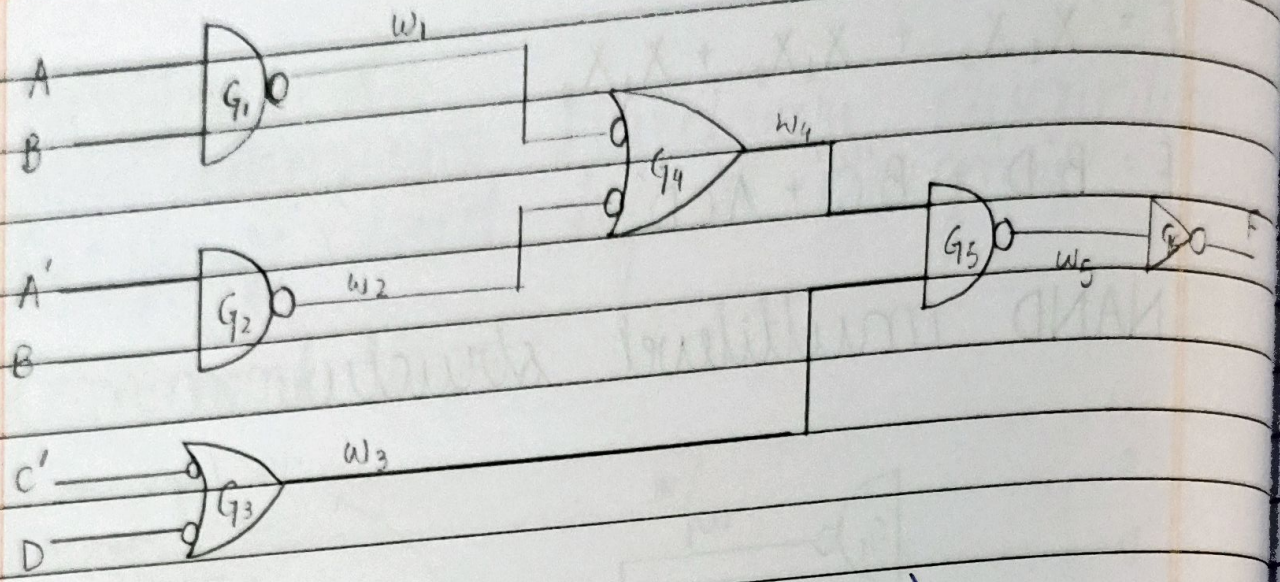
$F = X_1'X_3' + X_1X_3 + X_1X_2$ $\qquad F = YZ + YY'X'$

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

NAND multilevel structure




```
// HDL description of multilevel NAND realization
module prog_2 (F, A, B, C, D);
wire w1, w2, w3, w4, B', C';
nand (B', B, B);
     (C', C, C);
nand G1 (w1, C, D);
     G2 (w2, w1, B');
     G3 (w3, w2, A);
     G4 (w4, B, C');
     G5 (F, w3, w4);
endmodule
```




```
module prog_3(F, A, B, C, D);
wire A', C', W1, W2, W3, W4, W5
nand (A', A, A);
     (C', C, C);
nand G1(W1, A, B);
     G2(W2, A', B);
     G3(W3, C', D);
     G4(W4, W1, W2);
     G5(W5, W4, W3);
not G6(F, W5);
endmodule
```