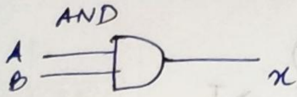
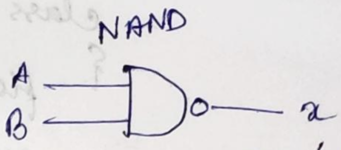
# CO CA

## Logic gates

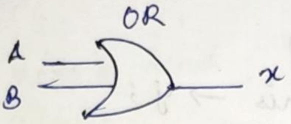


$x = A \cdot B$

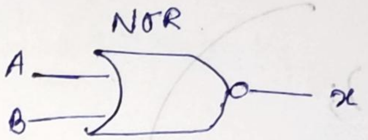
$= AB$

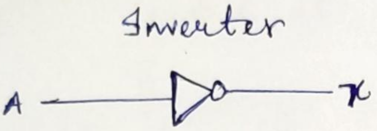


$x = (AB)'$

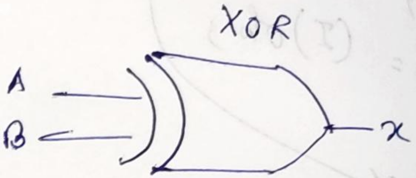


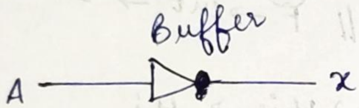
$x = A + B$





$x = A'$





$x = A.$

## K-Map.

| A \ B | 0 | 1 |
|---|---|---|
| 0 | | |
| 1 | | |

| A \ BC | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | | | | |
| 1 | | | | |

| AB \ CD | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 1 | 1 | 0 | 0 |
| 01 | 0 | 0 | 0 | 1 |
| 11 | 0 | 0 | 0 | 0 |
| 10 | 1 | 1 | 0 | 1 |

SOP

$\sum = 0, 1, 6, 8, 9, 10.$

B'C' fixed.

AB'D'

A'BCD'

$F(A, B, C, D) = \Sigma($

don't care - 6, 12.

SOP

| AB \ CD | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 1 | 1 | 0 | 1 |
| 01 | 0 | 1 | 0 | X1 |
| 11 | 1X | 0 | 0 | 0 |
| 10 | 1 | 1 | 0 | 1 |

$B'C' + B'D' + A'C'D + A'CD' + AC'D'$.

Computer System Architecture by Moris Mano.

| | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 1 | 1 | 0 | 1 |
| 01 | 0 | 1 | 0 | X0 |
| 11 | 0X | 0 | 0 | 0 |
| 10 | 1 | 1 | 0 | 1 |

POS

| | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 1 | 1 | 0 | 1 |
| 01 | 0 | 1 | 0 | X0 |
| 11 | 0X | 0 | 0 | 0 |
| 10 | 1 | 1 | 0 | 1 |

$Y = B'C' + B'D' + A'C'D + A'CD' + AC'D'$. SOP

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

Full adder.

| i/p | | | O/P | |
|---|---|---|---|---|
| x | y | z | C | S |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 |

S. 1, 2, 4, 7.

C. 3, 5, 6, 7

sum

| x \ yz | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 | 0 |

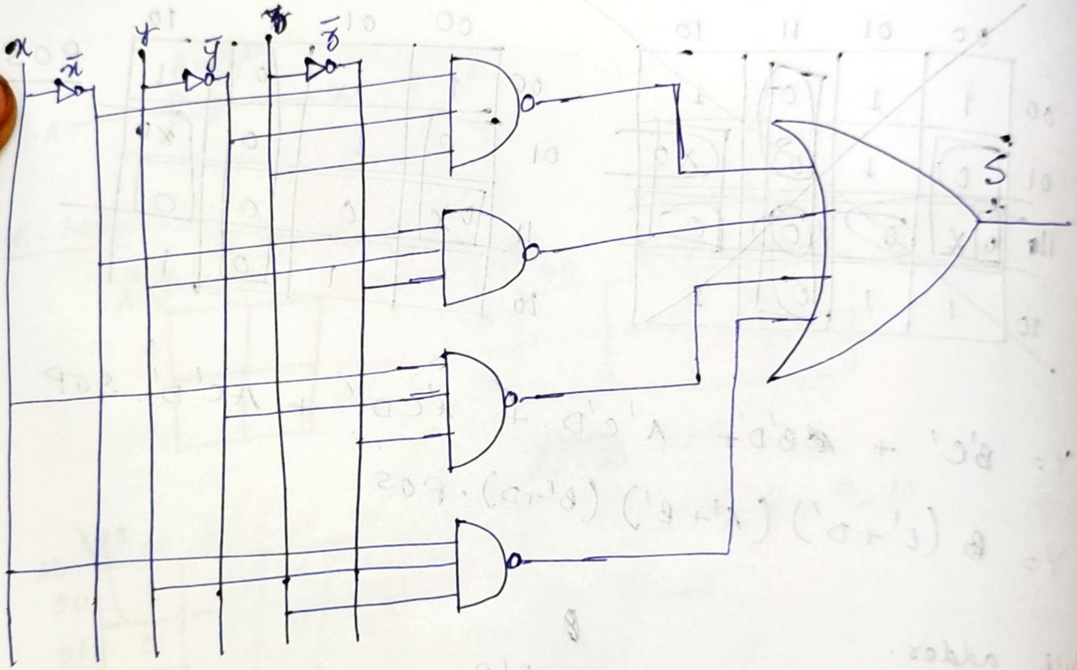
yz

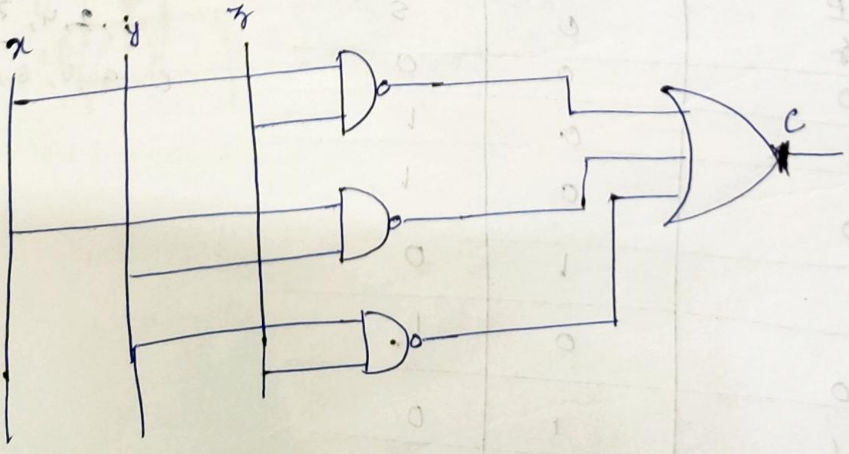
| x \ yz | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 | 1 |

CARRY

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

$C = xz + xy + \cancel{x}yz$





30|10|22 ',' operator.

b = 2, c = 3    precedence → left to right

a = (++b, ++c); returns right most value.

a = 4

```
class A
{
    int x;
  public:
      A operator , (A ob)
      {
           ob t;
           t.x = ob.x;
           return ob;
      }
```

3|11|22

COCA

TTL → Transitor Transistor Logic.

ECL → Emitter coupled logic.

MOS → Metal oxide semiconductor.

CMOS → Complementary MOS.

Decoders. 3x8 decoder

| Enable | $A_2$ | $A_1$ | $A_0$ | O/P |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | |
| 1 | 0 | 0 | 1 | |
| 1 | 0 | 1 | 0 | |
| 1 | 0 | 1 | 1 | |
| 1 | 1 | 0 | 0 | |
| 1 | 1 | 0 | 1 | |
| 1 | 1 | 1 | 0 | |
| 1 | 1 | 1 | 1 | |

D7 D6 ~~D5~~ D4 D3

D2

| Enable | $A_2$ | $A_1$ | $A_0$ |
|---|---|---|---|
| 0 | x | x | x |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 |

8 x 3 encoder Octal to binary.

FA7 → 111110100111

```
2 | 78 - 0
2 | 39 - 1
2 | 19 - 1
2 | 9 - 1
2 | 4 - 0
2 | 2 - 0
    1
```

1001110

~~[illegible]~~

$(736.4)_8 = (\ )_{10}$

7 3 6 . 4

$7\times8^2+3\times8^1+6\times8^0+\frac{\cancel{4}\,1}{\cancel{8}\,2}$

= 478.5.

```
  64
 x 7
 448
  24
 472
   6
 478
```

$(41.6875)_{10} = (?)_2$ 101001.1011

2 | 41 − 1
2 | 20 − 0
2 | 10 − 0
2 | 5 − 1
2 | 2 − 0
1

101001.

0.6875
× 2
1.3750
× 2
0.7500
× 2
1.5000
× 2
1.0000

+13  01101
+6  + 0110
10011

−6 ⇒

cat = cat + dog
cat − + = dog.
catdog

6/11/22.

```
template <class T> //generic
void swap <T> (T &a, T &b)
{
    T temp;  temp = a;  a = b;  b = temp;

}
int main ()
{
    swap <int> (2, 3);
    swap <float> (1.5, 0.75);

}
template <class T, int a> // specific
int add <T> (T x).
{  return x+a;
}
```

```
int main ()
{
    add <int,2> (5);
    add <float, 3> (4.5);
```

9/11/22 **String**

#include <string>

string s = " Hello" ;

cin >> s ; [Space is the end of a string].

getline (cin, s);

int → string s = s + "10" ; [concaetenation].

---

**COA** 10/11/22

13
− 6

1101
− 0110 $\xrightarrow{2's}$

110 $\xrightarrow{1's}$ 001 $\xrightarrow{2's}$ 010.

1101
1010
~~1~~0111

**Including sign bit.**

13
− 6

01101
00.110 $\xrightarrow{1's}$ 11001 $\xrightarrow{2's}$ 11001
1
11010

0 1 1 0 1
1 1 0 1 0
~~1~~0 0 1 1 1

0 0 1 1 1 (7)
0 1 1 1 0 −(14) $\xrightarrow{1's}$ 10001 $\xrightarrow{2's}$ 1 0 0 0 1
1
1 0 0 1 0

0 0 1 1 1
1 0 0 1 0
1) 1 0 0 1 $\xrightarrow{1's}$ 1) 0 1 1 0 ⟶ 1) 0 1 1 1
1
1 0 1 1 1

1001
1001

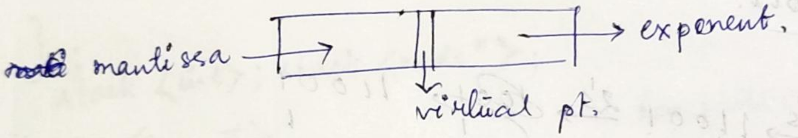
```
      1101
    x1001
    ------
      1101
     0000x
    0000xx
   1101xxx
   -------
   1110101
```

```
111 ) 110110 (0.11
     -111
     ----
      9900
     -111
     ----
       101
```

7) 5000 (

```
   70 | 01000110
 + 80 | 01010000
 -----|---------
```

Use flip-flop when we have an extra bit due to similar sign. It is out side register.



Parity bit

Odd /even parity → Odd no. of 1s, 2 odd ⟶ detects error correctly.
even " " " 2 even

LAB

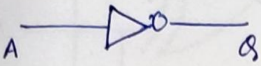
(i) Verification of universal gates

(ii) Realization of half adder

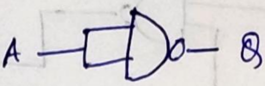
(iii) Realization of full adder using half adder.

12/11/22.

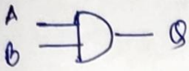
NOT



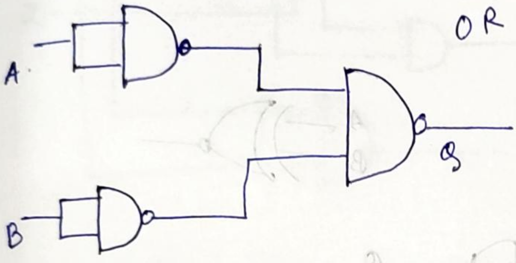
| In | Out |
|---|---|
| 0 | 1 |
| 1 | 0 |



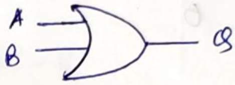
AND



| Input | | Output |
|---|---|---|
| A | B | Q |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1. |

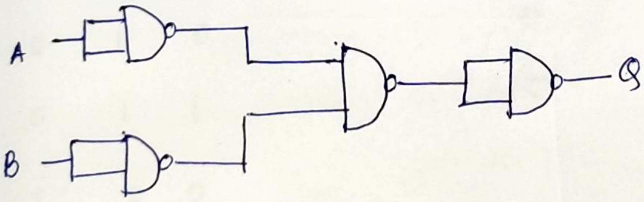


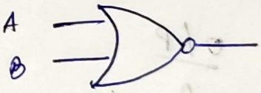
OR



| Input | | Output. |
|---|---|---|
| A | B | Q |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

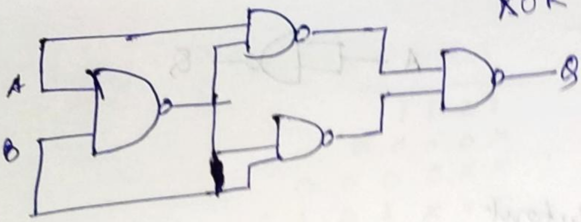
NOR

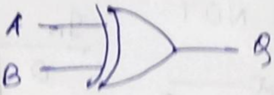




| Input | | Output |
|---|---|---|
| A | B | Q |
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

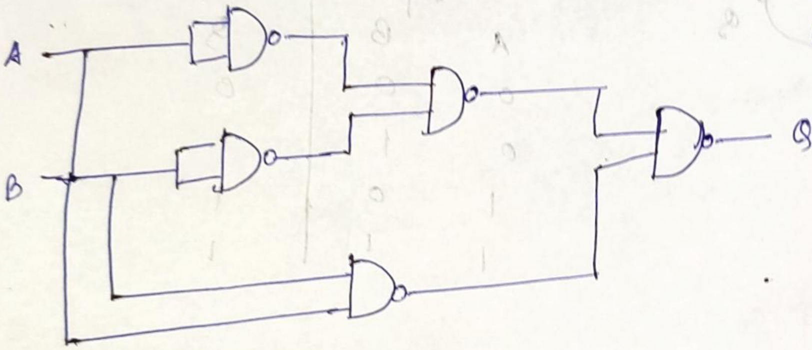
XOR

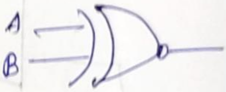




| Input |   | Output |
|---|---|---|
| A | B | Q |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

XNOR

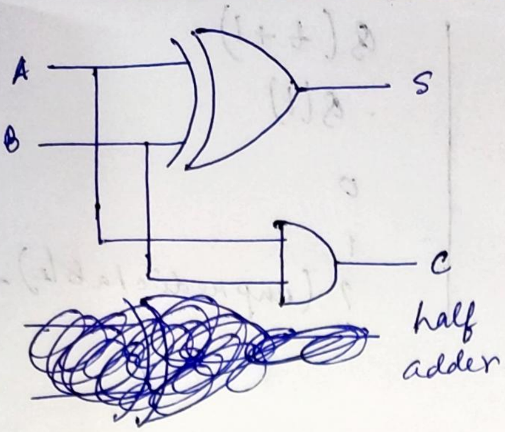




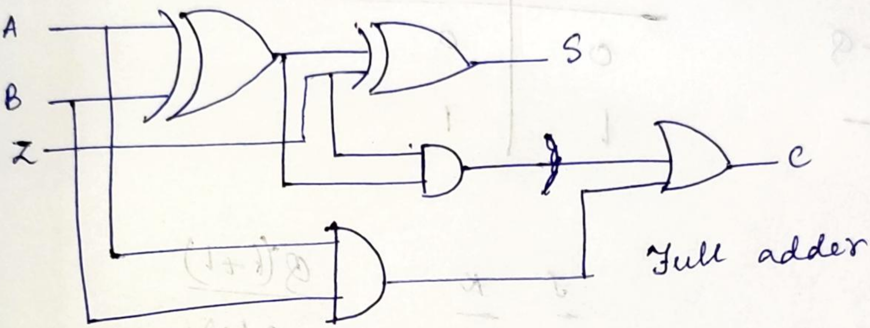
| i/P |   | O/P |
|---|---|---|
| A | B | Q |
| 0 | 0 | |
| 0 | 1 | |
| 1 | 0 | |
| 1 | 1 | |

Using XOR & AND → Half adder.

| i/P |   | O/P |   |
|---|---|---|---|
|   |   | S | C |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |



half adder

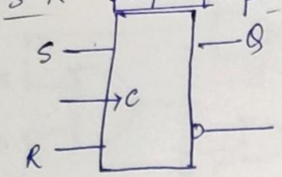


Full adder

Half adder

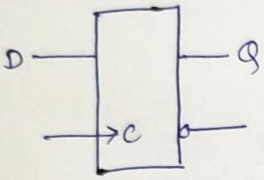
| A | B | Z | S | ~~C~~ | C |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | | 0 |
| 0 | 0 | 1 | 1 | | 0 |
| 0 | 1 | 0 | 1 | | 0 |
| 0 | 1 | 1 | 1 | | 0 |
| 1 | 0 | 0 | 1 | | 1 |
| 1 | 0 | 1 | 0 | | 1 |
| 1 | 1 | 0 | 0 | | 1 |
| 1 | 1 | 1 | 1 | | 1 |

## SR flip flop.



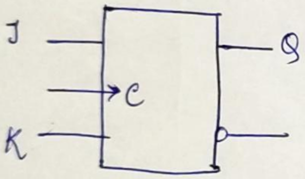
| S | R | Q(t+1) |
|---|---|---|
| 0 | 0 | Q(t) |
| 0 | 1 | 0 |
| 1 | 0 | 1 |
| 1 | 1 | ? (unpredictable). |

## D flip-flop.



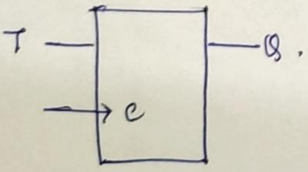
| D | Q(t+1) |
|---|---|
| 0 | 0 |
| 1 | 1 |

## JK - Flip Flop



| J | K | Q(t+1) |
|---|---|---|
| 0 | 0 | Q(t). |
| 0 | 1 | 0 |
| 1 | 0 | 1 |
| 1 | 1 | Q'(t). |

## T flip - flop.



| T | Q(t+1) |
|---|---|
| 0 | Q(t) |
| 1 | Q'(t). |