# ELE328: DIGITAL SYSTEMS AND MICROPROCESSORS

- Introduction
  - Scope and Objectives
  - Course Management
- Introduction to Logic Circuits
  - Variables and Functions

**Scope and Objectives**

- Design of Basic Digital Logic Circuits, and Implementation in Appropriate Technology
- Digital Circuits include Combinational and Sequential Circuits from simple circuits to microprocessor
- Use CAD Tools for design entry (Schematics, VHDL), verification using simulation and implementation in appropriate technology

## Course Management

`course information:`

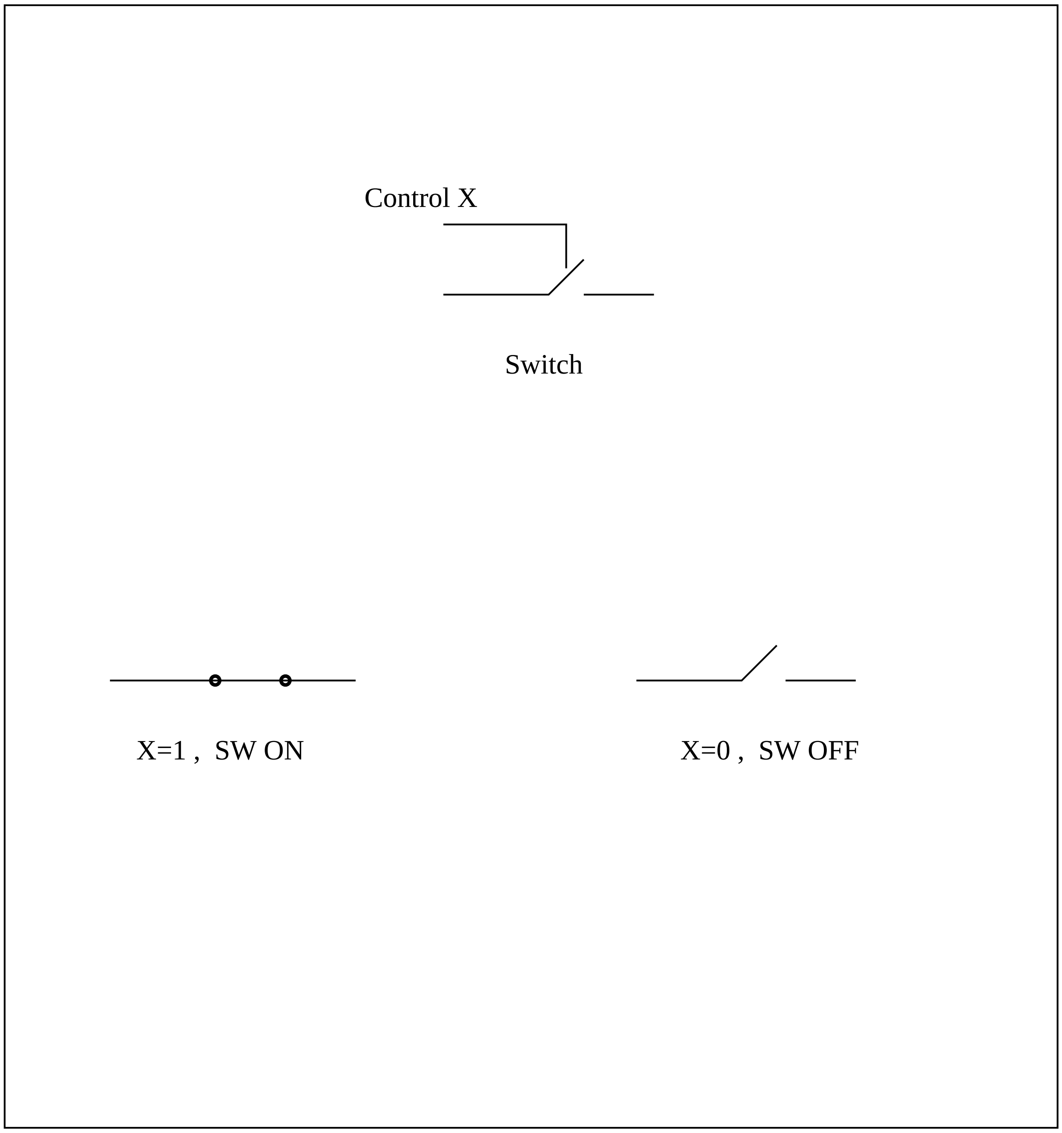
`http://www.ee.ryerson.ca/~courses/ele328`

- Instructor: Nagi Mekhiel Ext 7251
  Email: nmekhiel@ee.yerson.ca

- Text Book:
  - Fundamentals of Digital Logic with VHDL Design, Brown and Varensic, 1st edition, McGraw-Hill
  - Introduction To Digital Logic Design, Hayes, Addison Wesley
  - Any VHDL Reference
  - ELE328 Lab Manual

## Logic Circuits

- **Logic Variables and Functions**
  - Binary variable X has two values: TRUE OR FALSE
    X= 1 (TRUE) , X = 0 (FALSE)
    Variable representation: SWITCH
    SWITCH ON variable = TRUE(=1)
    SWITCH OFF variable = FALSE(0)
  - Logic Function :
    Example F(X) = X
    If input variable X=0,
    Output Function F(X)=X=0
    Implementation: Use SWITCHES




F(X)=X Function Implementation

## Different Logic Functions

AND Logic Function :

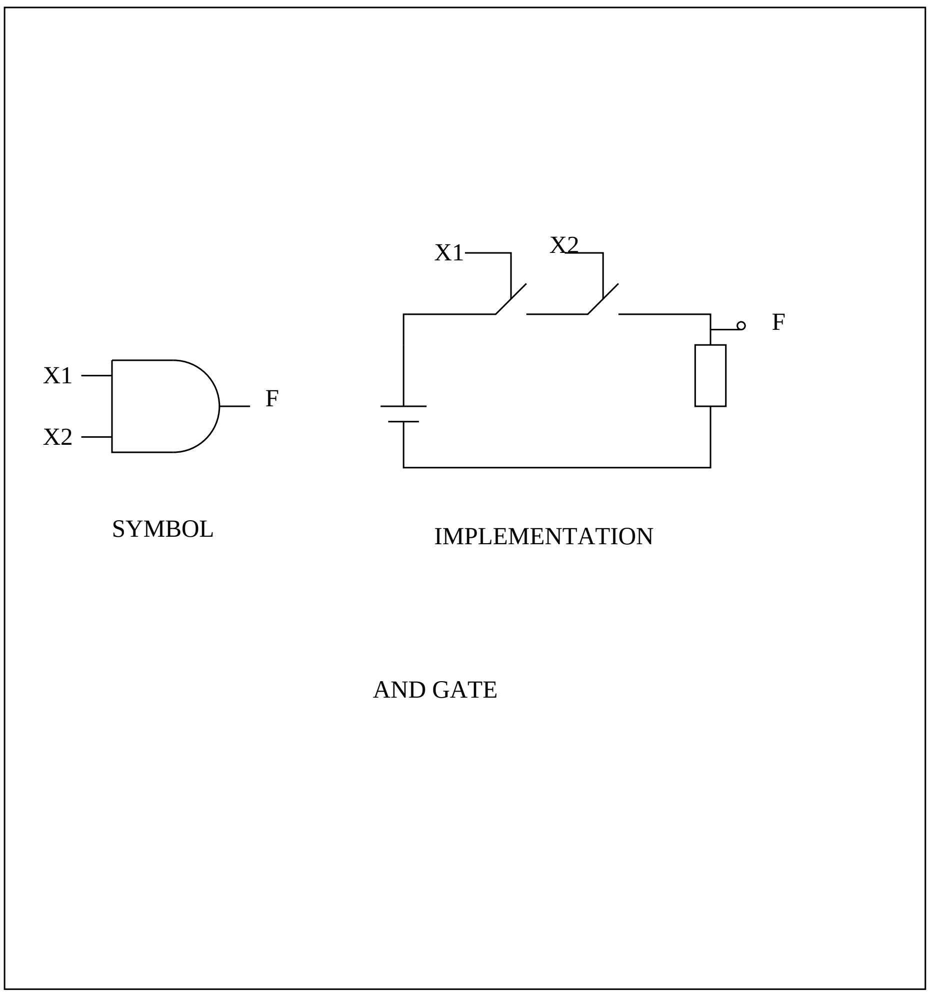
F(X1,X2) = X1.X2

Behavior : OUTPUT is TRUE IF X1 AND X2 are TRUE

Example: Find X1.X2 if X1=1, X2=1

X1 AND X2 =X1.X2=1.1=1

Symbol and Implementation



**Different Logic Functions**

OR Logic Function :

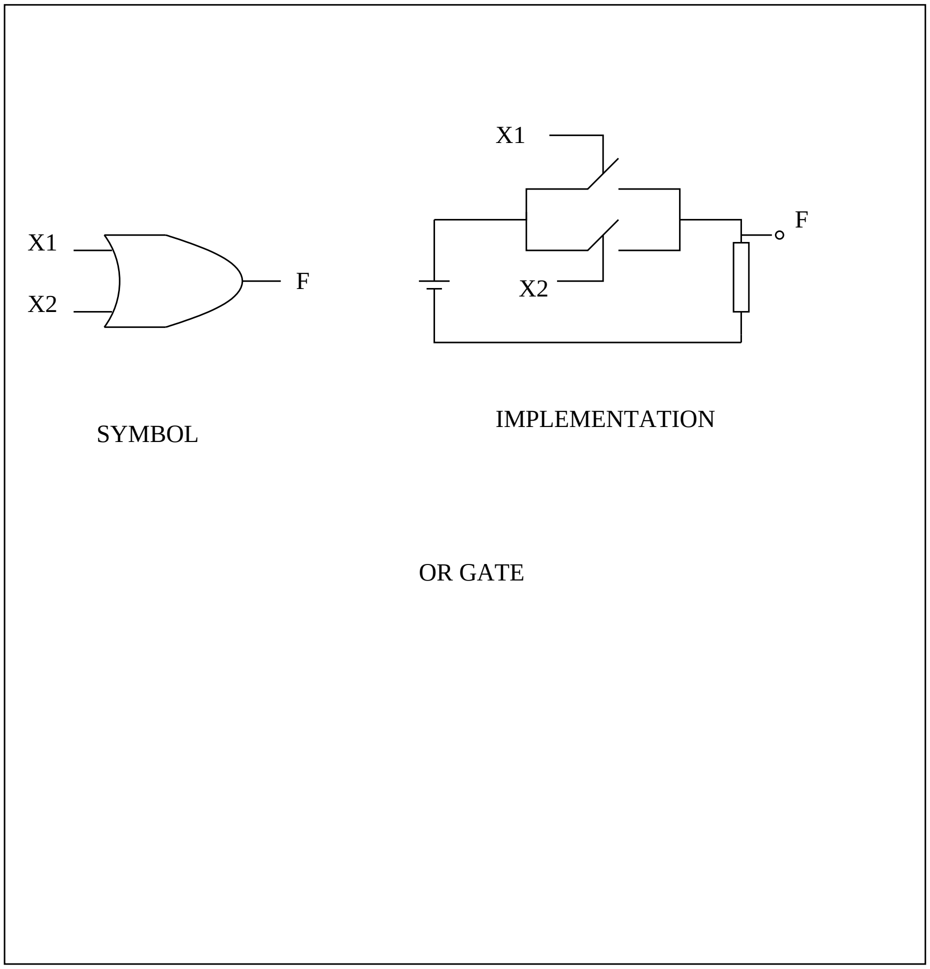
F(X1,X2) = X1+X2

Behavior : OUTPUT is TRUE IF X1 OR X2 IS TRUE

Example: Find X1+X2 if X1=1, X2=1

X1 OR X2 =X1+X2=1+1=1

Symbol and Implementation



**Different Logic Functions**

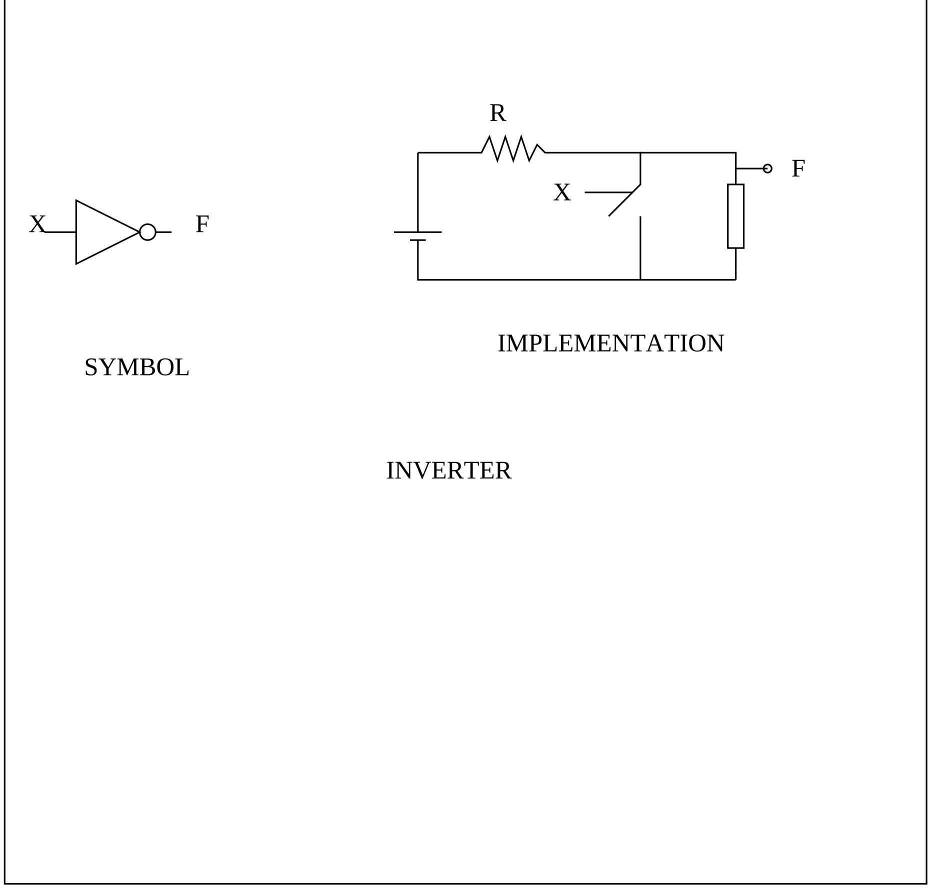
INV Logic Function : F(X1) =!X1

Behavior : OUTPUT is TRUE IF X1 IS FALSE

Example: Find !X1 if X1=1

!X1 = 0

Symbol and Implementation



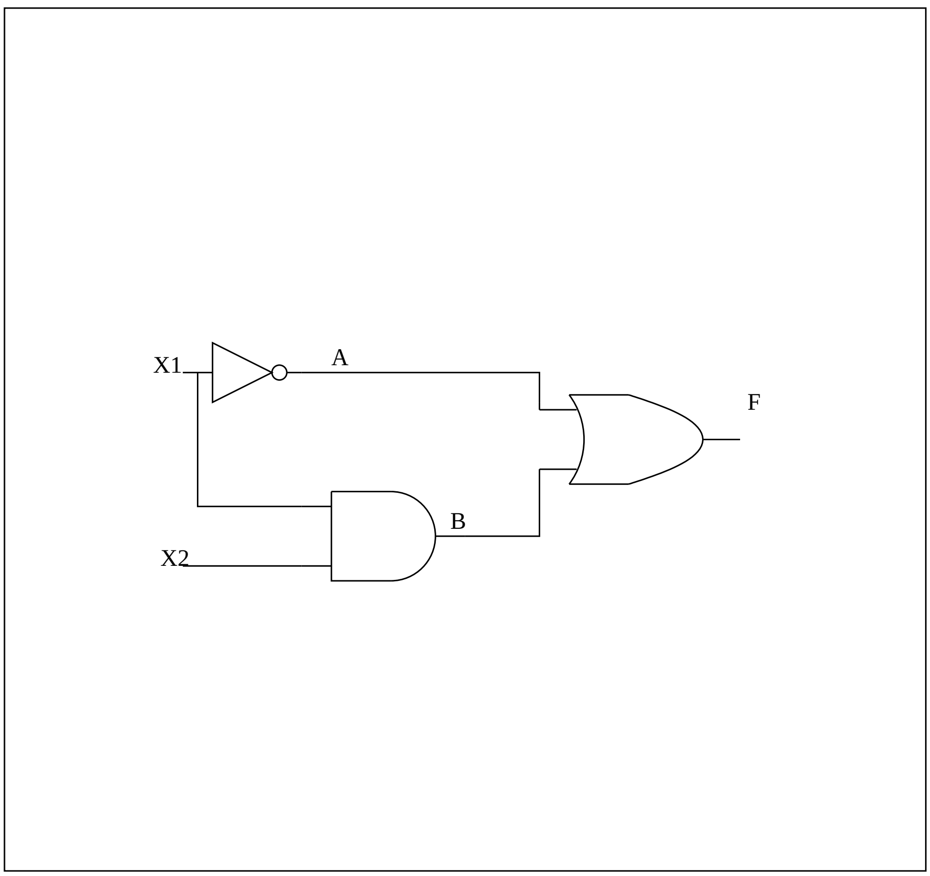
**The Truth Table**

- Representation of complicated Logic Functions
- Gives all information to design logic function
- Table of outputs for all possible input condid-tions
- For three inputs (X1,X2,X3), Conditions = $2^3 = 8$

| X1 | X2 | X3 | Out |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |

## Logic Gates

- **Analysis of Logic Gates:**
  - Find Logic Function from Schematic
  - Find Truth Table from Logic Function
  - Find Timing Diagram to Verify Function
- Example: Given the Schematic below



- Logic Function of the Schematic
  A=!X1, B=X1.X2
  F(X1,X2)=A+B = !X1 + X1.X2

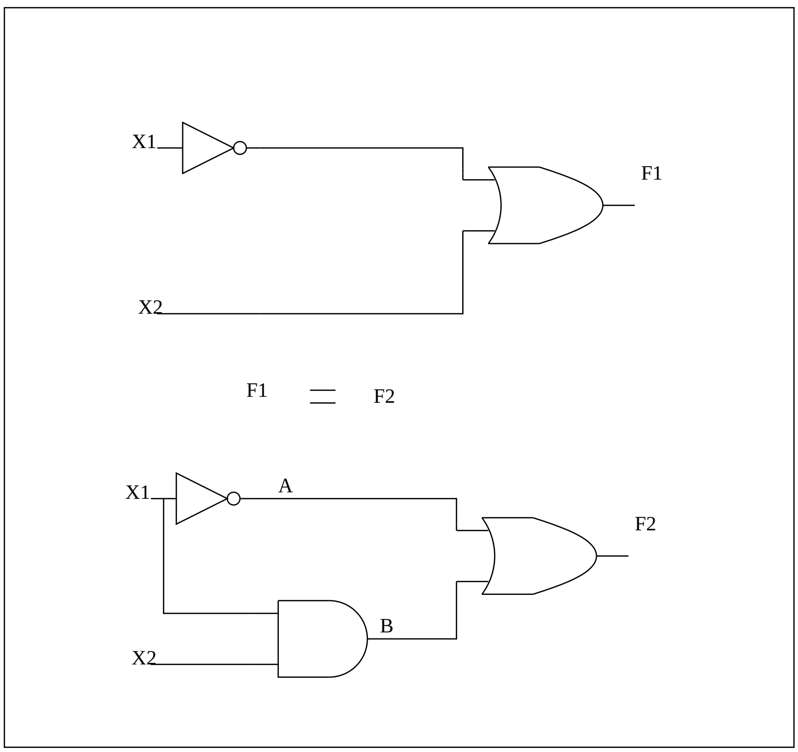
- Truth Table of the Logic Circuit above

| **X1** | **X2** | **F** |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

- Timing Diagram:

## Functionally Equivalent Circuits

- Logic Functions could be implemented using different Circuits
- Example: Consider the Circuit Below:



## Functionally Equivalent Circuits

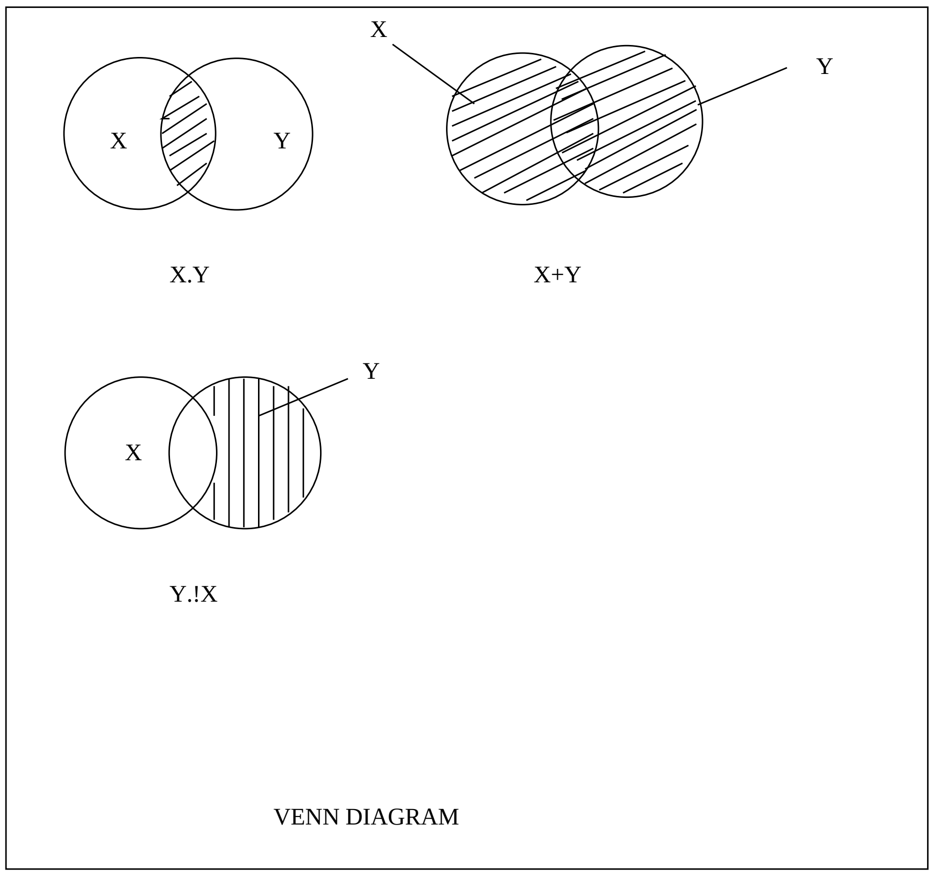
- F1= !X1+X2
- Truth Table of the Logic Circuit above

| **X1** | **X2** | **F1=F2** |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

- **Boolean Algebraic Rules:**
- **Used for Optimization of Logic Functions**
  - X.0=0 , X.1=X
  - X+1=1 , X+0=X
  - X.X=X , X+X=X
  - X.!X=0 , X+!X=1
  - !!X=X
  - X.Y=Y.X , X+Y=Y+X (Could optimize wire Routing)
  - X.(Y.Z)= (X.Y).Z , X+(Y+Z)=(X+Y)+Z
  - X.(Y+Z)=X.Y + X.Z
  - Prove that: (X+Y).(X+Z)=X+YZ
    LHS= X.X+X.Z+Y.X+Y.Z
    =X(1+Y+Z)+YZ=X+YZ

- Prove that X+X.Y=X
  LHS=X.(1+Y)=X.1
- **DeMorgan's Theorem**
  - !(X+Y) = !X.!Y
  - !(X.Y)= !X + !Y
  - Useful to implement Functions using different gates
  - Example: Find !(X1.X2 + (Y1+!X2))
    =!(X1.X2). !(Y1+!X2)
    =(!X1+!X2).(!Y1.X2)
    =!X1.!Y1.X2 + !X2.!Y1.X2=!X1.!Y1.X2

## Venn Diagram




Graphical representation of logic operations

## Synthesis of Logic Functions (Implementation)

- **Method1: Sum of Products**
  - Construct a Truth Table
  - Find each condition where the output=1, call this a minterm
  - Use AND function to implement each minterm so that the output=1, this is a product term
  - Take the SUM of all product terms using OR function

- Use the sum of product to implement the following:

| **cond** | **X1** | **X2** | **F** | **product term** |
|---|---|---|---|---|
| 0 | 0 | 0 | 1 | m0=!X1.!X2 |
| 1 | 0 | 1 | 1 | m1=!X1.X2 |
| 2 | 1 | 0 | 0 | |
| 3 | 1 | 1 | 1 | m3=X1.X2 |

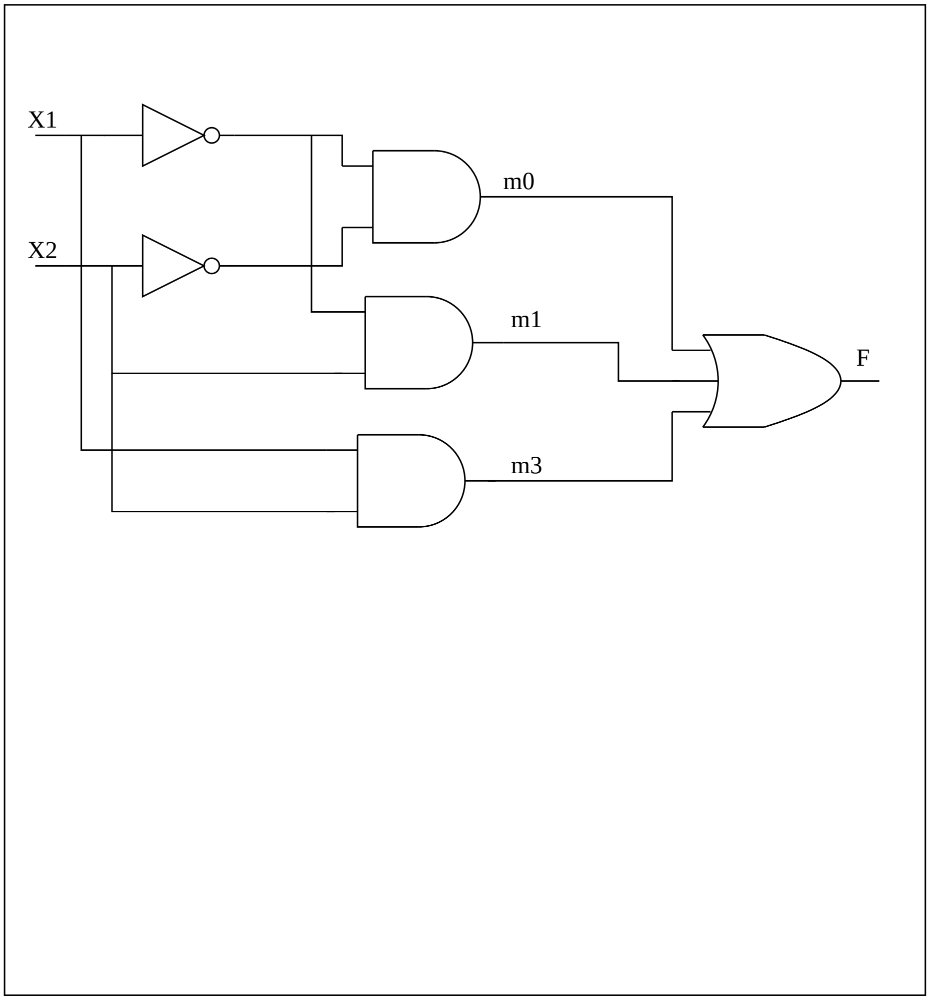
Logic Function= Sum of Products

F=m0+m1+m3

F=!X1.!X2+!X1.X2+X1.X2

## Circuit Implementation: Using AND - OR

F=!X1.!X2+!X1.X2+X1.X2



- **Method2: Product Of Sums**
  - Construct a Truth Table
  - Find each condition where the output=0, call this a Maxterm
  - Use OR function to implement each Max-term
    so that the output=0, this is a sum term
  - Take the PRODUCT of all SUM terms using AND function

- Use the product of sums to implement the following:

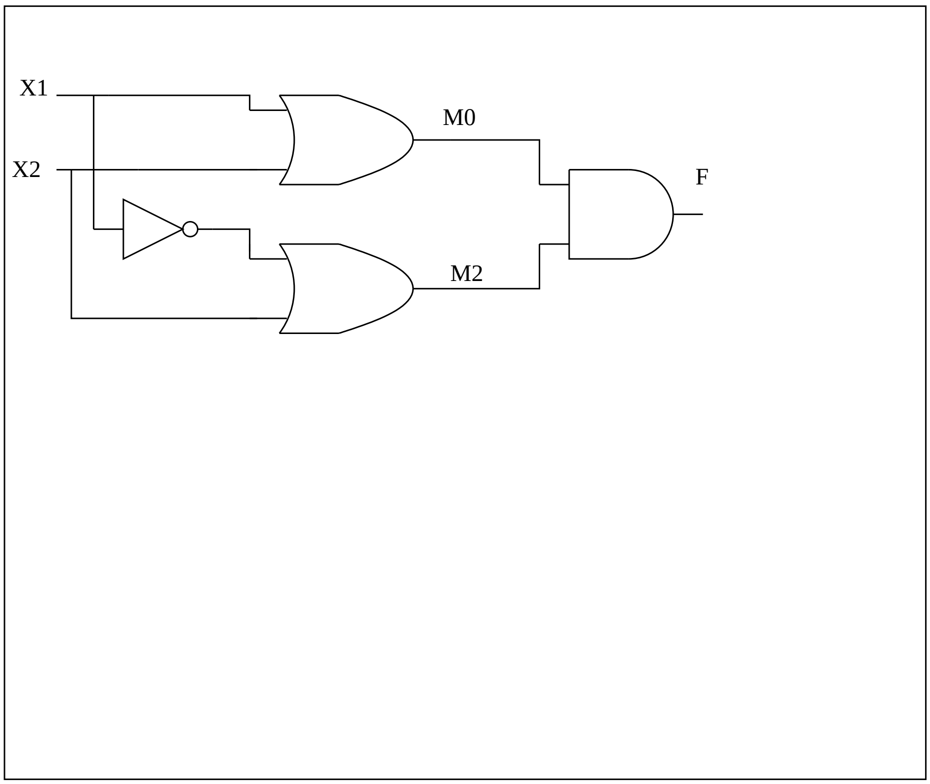
| **cond** | **X1** | **X2** | **F** | **MAX term** |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | M0=X1+X2 |
| 1 | 0 | 1 | 1 | |
| 2 | 1 | 0 | 0 | M2=!X1+X2 |
| 3 | 1 | 1 | 1 | |

- Logic Function= Product of SUMS

- F=M0.M2=(X1+X2).(!X1+X2)

F=M0.M2=(X1+X2).(!X1+X2)

Circuit Implementation using OR - AND



- Example: Design a Majority function for three inputs such that output=1 if two or more inputs=1

| **X1** | **X2** | **X3** | **Out** |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |

Using Sum Of Products and Product of Sums

- Using sum of products F=m3+m5+m6+m7
  F=!X1.X2.X3+X1.!X2.X3
  +X1.X2.!X3+X1.X2.X3

- Using product of sums F= M0.M1.M2.M4
  F=(X1+X2+X3).(X1+X2+!X3)
  .(X1+!X2+X3).(!X1+X2+X3)

**Introduction to CAD Tools**

Computer Aided Design Software to automatically Design, Verify and Implement complex systems

Design Steps:

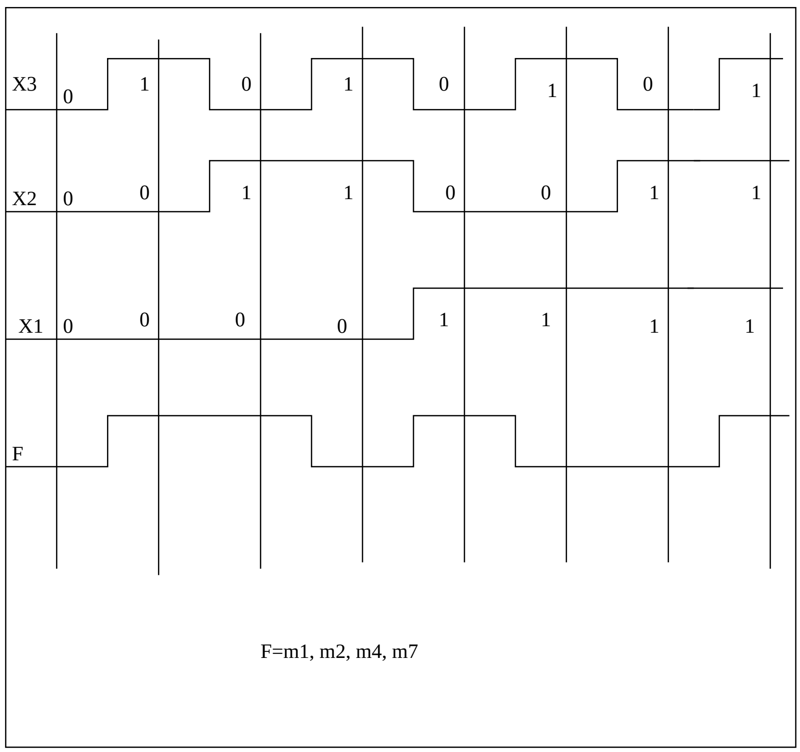
- Design Entry:
  - Using **Schematic Diagram** , enter symbol of each gate and connect gates to get the design (takes time, could have mistakes, easy to understand and follow design )
  - Using **Truth Table from waveforms**, only for combinational circuits, might not cover all conditions)
  - Using **VHDL** Hardware Description Language, easy to change, faster design, better documentation and is portable (does not depend on type of gates)

## Introduction to CAD Tools

Computer Aided Design Software to automatically Design, Verify and Implement complex systems

Design Steps:

- Functional Simulation: use different input conditions in form of waveforms to verify the design (test vectors)




- Synthesis: Generate a logic circuit from design according to target technology

**Introduction to VHDL**

Consists of:

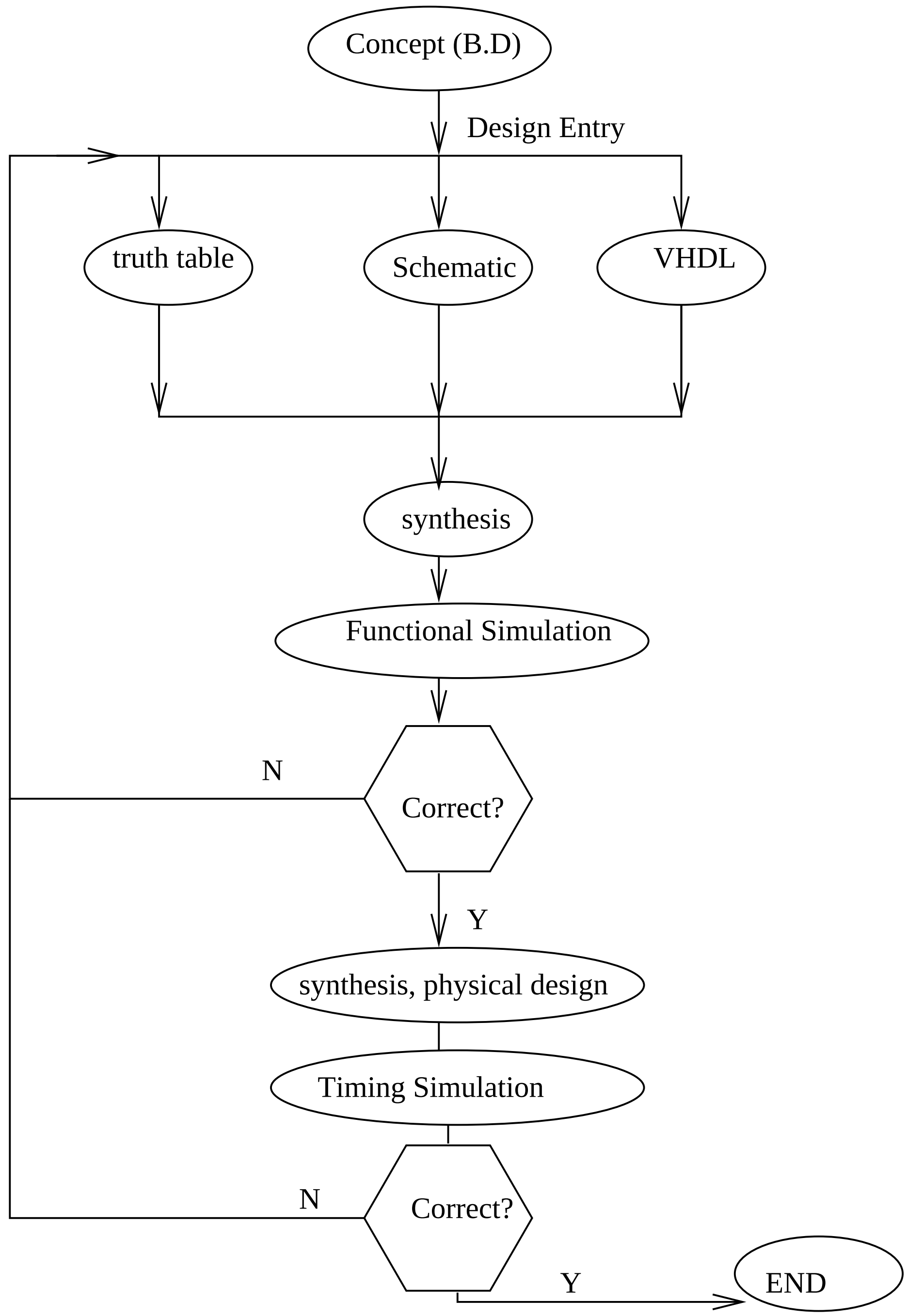
- Declaration: to define inputs and outputs

```
ENTITY Funct IS
      PORT(X1,X2,X3 : IN  STD_LOGIC;
           F        : OUT STD_LOGIC);
END Funct;
```

- Architecture to describe the circuit behavior

```
ARCHITECTURE LogicFunction OF Funct IS
BEGIN
   F<=(X1ANDX2)OR(NOTX2ANDX3);
END LogicFunction;
```

# CAD System


Concept (B.D)
Design Entry
truth table
Schematic
VHDL
synthesis
Functional Simulation
N
Correct?
Y
synthesis, physical design
Timing Simulation
N
Correct?
Y
END

## Solutions to Selected Problems

1-Prove that:

!X1.X3+X1.X2.!X3+!X1.X2+X1.!X2=

!X2.X3+X1.!X3+X2.!X3+!X1.X2.X3

Solution: Expand L.H.S and R.H.S. to standard mint

using the boolean algebraic rule: A = A.(X1+!X1)

R.H.S= !X1.X3(X2+!X2) + X1.X2.(X3+!X3)

+X1.!X2(X3+!X3)

=m1,m2,m3,m4,m5,m6=L.H.S

2- Show that sum of m1,m2,m3,m4,m5,m6,m7

=X1+X2+X3

m1,m3= !X1.!X2.X3+!X1.X2.X3=!X1.X3

m5,m7= X1.!X2.X3+X1.X2.X3=X1.X3

Combine m1,m3,m5,m7=!X1.X3+X1.X3=X3

m4,m5,m6,m7 gives X1

m2,m3,m6,m7 gives X2

## Solutions to Selected Problems

3-Design Simplest circuit for f(X1,X2,X3)
  = SUM of 1,3,4,6,7

Solution:

1,3= !X1.!X2.X3+!X1.X2.X3=!X1.X3

4,6=X1.!X2.!X3+X1.X2.!X3=X1.!X3

3,7=!X1.X2.X3+X1.X2.X3=X2.X3

F=!X1.X3+X1.!X3+X2.X3

4-Design Simplest circuit for f(X1,X2,X3)
 = Product of 0,1,5,7

Solution:

use the following rule  (A+X1).(A+!X1)=A

0,1 = (X1+X2+X3).(X1+X2+!X3)= (X1+X2)

5,7=(!X1+X2+!X3).(!X1+!X2+!X3)=(!X1+!X3)

F=(X1+X2).(!X1+!X3)