CS230: Digital Logic Design and Computer Architecture

The Digital world: Logic gates

https://www.cse.iitb.ac.in/~biswa/courses/CS230/main.html

https://www.cse.iitb.ac.in/~biswa/

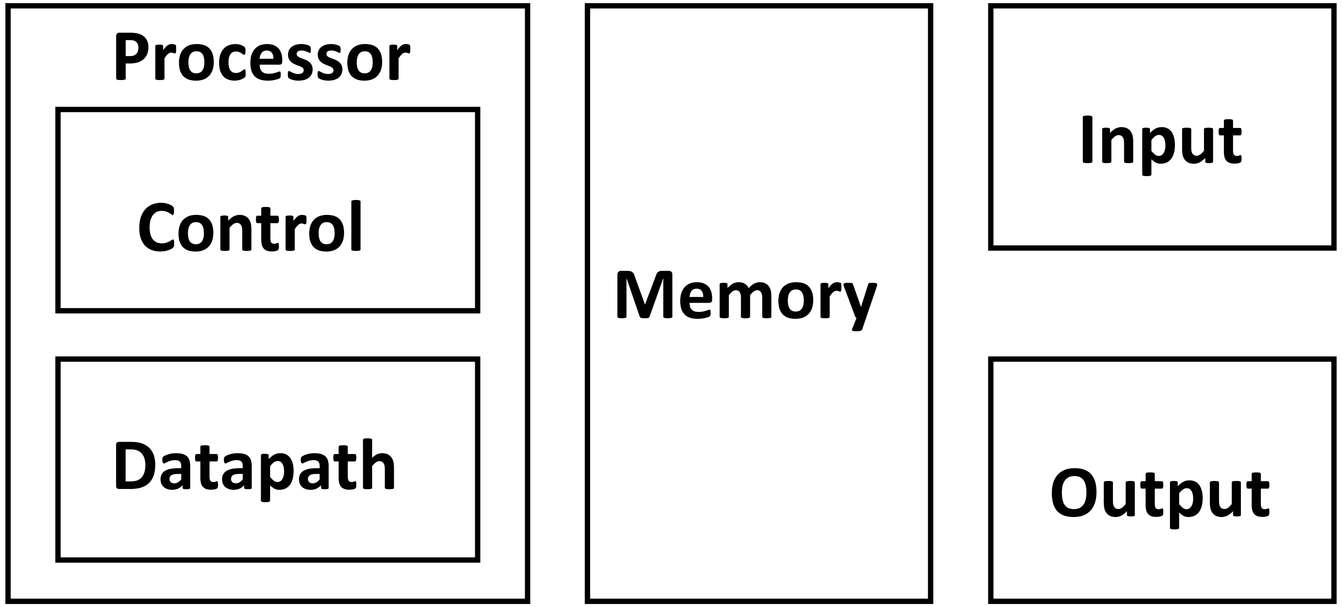
Phones (smart/non-smart) on silence plz, Thanks

Logistics

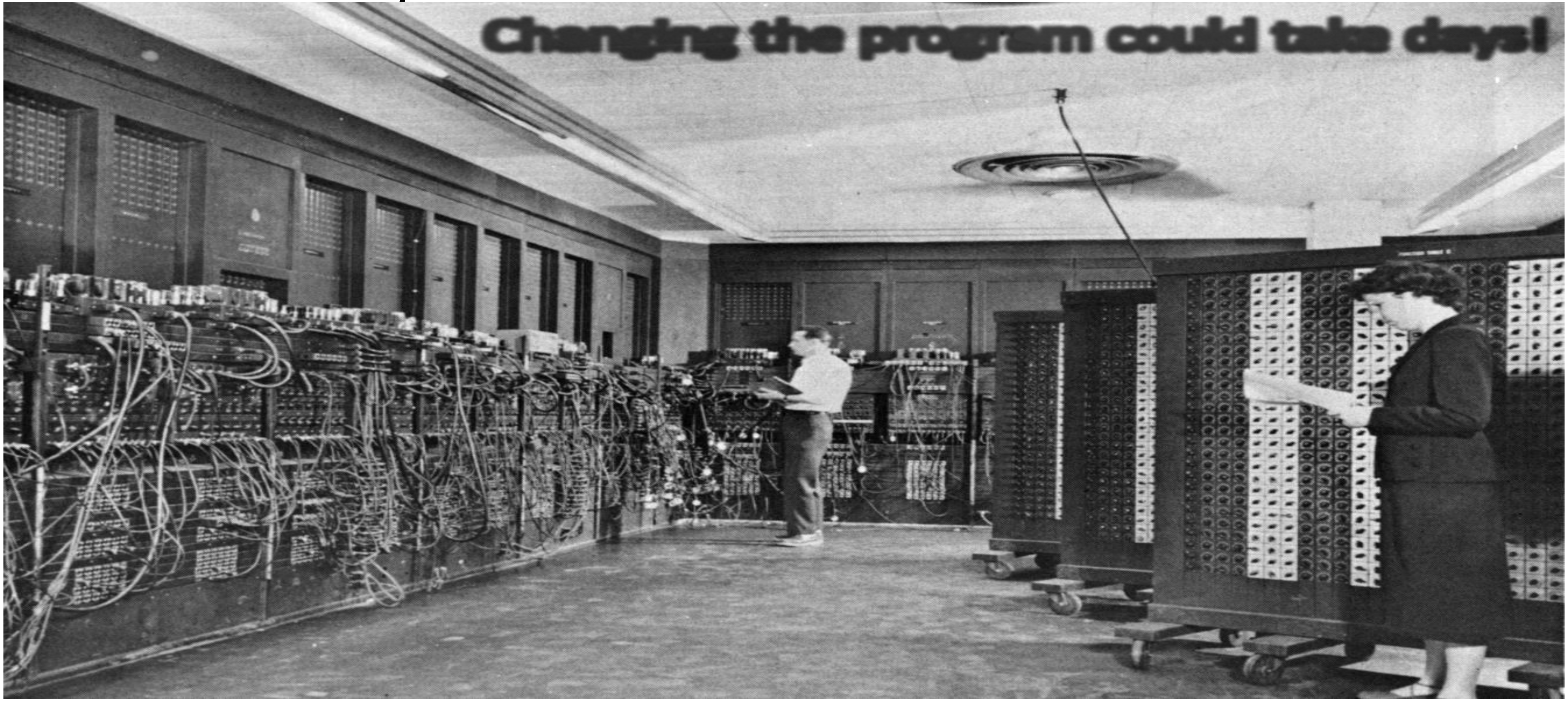
- Join Piazza now
- Lab on Tuesday, 2 PM SL1 to SL3, attendance compulsory
- You can meet me and discuss if anything is not clear
- Problem set 1 in a week or two. Ungraded, for your practice only

Let’s get started

Since 1946 all computers have had 5 components



In those days



Let’s get started: One Step at a time

World of Digital computers

Not Analog

Digital: World of TRUE/FALSE or 1/0

World of binary variables

Logic circuits performing operations on binary variables: Logic gates

Digits vs bits

- Digits = powers of 10

  … 100, 10, 1, $^1/_{10}$, $^1/_{100}$, $^1/_{1000}$ …

  … 10^2, 10^1, 10^0, 10^{-1}, 10^{-2}, 10^{-3} …

  Ex: $(36.25)_{10} = 3*10 + 6*1 + 2*{}^1/_{10} + 5*{}^1/_{100}$

- Bits = powers of 2

  … 8, 4, 2, 1, $^1/_2$, $^1/_4$, $^1/_8$ …

  … 2^3, 2^2, 2^1, 2^0, 2^{-1}, 2^{-2}, 2^{-3} …

  Ex: $(100100.01)_2 = 1*32 + 1*4 + 1*{}^1/_4$

Decimal to binary

- Left of decimal point
  - Repeatedly divide integer part by 2 until you get 0
  - Read remainders bottom to up

$22 = (?)_2$ $(10110)_2$

22
11 R 0
5 R 1
2 R 1
1 R 0
0 R 1

Decimal to binary

- Right of decimal point
  - Repeatedly multiply fractional part by 2 until you get 1
  - Read integer portion top to bottom

$0.8125 = (0.1101)_2$

0.8125
1.6250
1.25
0.5
1.0

Both?

- What if there are both left and right of the decimal point?
  - Do them separately and combine

- **22.8125 = $(10110.1101)_2$**

up down

22
11 R 0
5 R 1
2 R 1
1 R 0
0 R 1

0.8125
1.6250
1.25
0.5
1.0

Binary Number System

1000's column
100's column
10's column
1's column

$$9742_{10} = 9 \times 10^3 + 7 \times 10^2 + 4 \times 10^1 + 2 \times 10^0$$

nine thousands | seven hundreds | four tens | two ones

16's column
8's column
4's column
2's column
1's column

$$10110_2 = 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = 22_{10}$$

msb | lsb

one sixteen | no eight | one four | one two | no one

Convert 84_{10} to binary

Convert 84_{10} to binary

- 1010100_2
- Similarly hexadecimal (base 16)
- $2ED_{16}$
- Convert this to decimal and binary

Binary addition

- Simple
- 1 + 0 = 1
- 0 + 0 = 0
- 1 + 1 = 0 with carry 1

So far unsigned, what about signed

most significant bit denotes sign and remaining N-1 bits denote value (Sign/magnitude numbers)

5_{10}: 0101_2

-5_{10}: 1101_2

Binary addition does not make sense

5 + (-5) = 0 BUT NOT IN SIGNED/MAGNITUDE

BTW, ZERO HAS TWO REPRESENTATIONS IN SIGN MAGNITUDE +0 AND -0 , WHICH IS SO CONFUSING

SOLUTION? 2'S COMPLEMENT

The 2's complement way for negative numbers

Take the complement of a binary number and add 1 to the lsb (least significant bit)

-5_{10}: $?_2$

5_{10}: 0101, complement: 1010, 2's complement: 1011

Range of Numbers

System	Range
Unsigned	$[0, 2^N - 1]$
Sign/Magnitude	$[-2^{N-1} + 1, 2^{N-1} - 1]$
Two's Complement	$[-2^{N-1}, 2^{N-1} - 1]$

Remember sign/magnitude has two zeros ☺

Sign Extension

To represent a signed number in 2's complement form using large number of bits

Repeat the sign bit at the msbs as needed

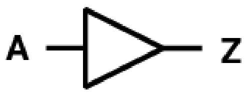
Overflow

1101 + 0101 ?

PAUSE

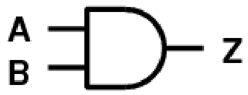
Common Logic Gates

Buffer



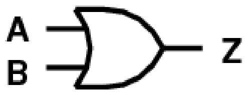
A	Z
0	0
1	1

AND



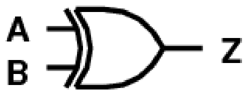
A	B	Z
0	0	0
0	1	0
1	0	0
1	1	1

OR



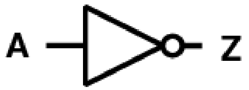
A	B	Z
0	0	0
0	1	1
1	0	1
1	1	1

XOR



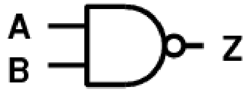
A	B	Z
0	0	0
0	1	1
1	0	1
1	1	0

Inverter



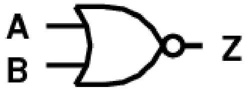
A	Z
0	1
1	0

NAND



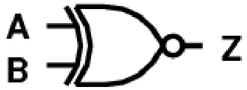
A	B	Z
0	0	1
0	1	1
1	0	1
1	1	0

NOR



A	B	Z
0	0	1
0	1	0
1	0	0
1	1	0

XNOR



A	B	Z
0	0	1
0	1	0
1	0	0
1	1	1

Universal Logic gates? Coffee points++

NAND and NOR

A bit of Boolean algebra

Dual

Operations with 0 and 1: 1. $X + 0 = X$ 2. $X + 1 = 1$	1D. $X \bullet 1 = X$ 2D. $X \bullet 0 = 0$	AND, OR with identities gives you back the original variable or the identity (dot: AND, plus: OR)
Idempotent Law: 3. $X + X = X$	3D. $X \bullet X = X$	AND, OR with self = self
Involution Law: 4. $\overline{(\overline{X})} = X$		double complement = no complement
Laws of Complementarity: 5. $X + \overline{X} = 1$	5D. $X \bullet \overline{X} = 0$	AND, OR with complement gives you an identity
Commutative Law: 6. $X + Y = Y + X$	6D. $X \bullet Y = Y \bullet X$	Just an axiom…

Contd.

Associative Laws:

7. $(X + Y) + Z = X + (Y + Z)$
$= X + Y + Z$

7D. $(X \bullet Y) \bullet Z = X \bullet (Y \bullet Z)$
$= X \bullet Y \bullet Z$

Parenthesis order does not matter

Distributive Laws:

8. $X \bullet (Y + Z) = (X \bullet Y) + (X \bullet Z)$

8D. $X + (Y \bullet Z) = (X + Y) \bullet (X + Z)$

Axiom

Simplification Theorems:

9. $X \bullet Y + X \bullet \overline{Y} = X$

9D. $(X + Y) \bullet (X + \overline{Y}) = X$

10. $X + X \bullet Y = X$, how?

10D. $X \bullet (X + Y) = X$

11. $(X + \overline{Y}) \bullet Y = X \bullet Y$

11D. $(X \bullet \overline{Y}) + Y = X + Y$

Useful for simplifying expressions

Actually worth remembering — they show up a lot in real designs…

DeMorgan's Law (Can you prove it)?

12. $\overline{(\boldsymbol{X} + \boldsymbol{Y} + \boldsymbol{Z} + \cdots)} = \bar{\boldsymbol{X}}.\bar{\boldsymbol{Y}}.\bar{\boldsymbol{Z}}. \ldots$

12D. $\overline{(\boldsymbol{X} . \boldsymbol{Y}. \boldsymbol{Z}. \ldots)} = \bar{\boldsymbol{X}} + \bar{\boldsymbol{Y}} + \bar{\boldsymbol{Z}} + \ \ldots$

- **Think of this as a transformation**
  - Let's say we have:

$$F = A + B + C$$

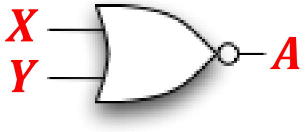
  - Applying DeMorgan's Law (12), gives us

$$\boldsymbol{F} = \overline{\overline{(\boldsymbol{A} + \boldsymbol{B} + \boldsymbol{C})}} = \overline{(\bar{\boldsymbol{A}}.\bar{\boldsymbol{B}}.\bar{\boldsymbol{C}})}$$

At least one of A, B, C is TRUE --> It is **not** the case that A, B, C are **all** false

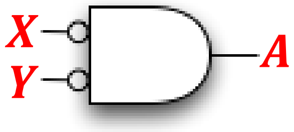
Contd. with a Truth Table

$$A = \overline{(X + Y)} = \overline{X}\,\overline{Y}$$

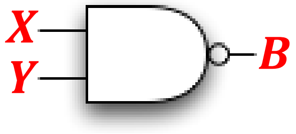


X	Y	$\overline{X+Y}$	$\overline{X}$	$\overline{Y}$	$\overline{X}\,\overline{Y}$
0	0	1	1	1	1
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	0

NOR is equivalent to AND with inputs complemented

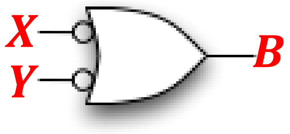


$$B = \overline{(XY)} = \overline{X} + \overline{Y}$$



X	Y	$\overline{XY}$	$\overline{X}$	$\overline{Y}$	$\overline{X} + \overline{Y}$
0	0	1	1	1	1
0	1	1	1	0	1
1	0	1	0	1	1
1	1	0	0	0	0

NAND is equivalent to OR with inputs complemented



Remember: It is not

$$\overline{(\boldsymbol{X}.\boldsymbol{Y})} = \bar{\boldsymbol{X}}.\bar{\boldsymbol{Y}}$$

$$\overline{(\boldsymbol{X}+\boldsymbol{Y})} = \bar{\boldsymbol{X}}+\bar{\boldsymbol{Y}}$$

Definitions of interest

- A ***normal term*** is a product or sum term in which no variable appears more than once.
  - *Examples:* $a, \bar{a}, a+c, \bar{a}cd$ are normal terms; $\bar{a}+a, \bar{a}a$ are not normal terms.

- A ***minterm*** of *n* variables is a normal product term with *n* literals. There are 2^n such product terms.
  - *Examples of 3-variable minterms:* $\bar{a}bc, abc$
  - *Example:* $\bar{a}b$ is not a 3-variable minterm.
- A ***maxterm*** of *n* variables is a normal sum term with *n* literals. There are 2^n such sum terms.
  - *Examples of 3-variable maxterms:* $\bar{a}+b+c, a+b+c$

Definitions of interest

- A ***sum of products (SOP)*** expressions is a set of product (AND) terms connected with logical sum (OR) operators.
  - *Examples:* $a, \bar{a}, ab+c, \bar{a}c+bde, a+b$ are SOP expressions.
- A ***product of sum (POS)*** expressions is a set of sum (OR) terms connected with logical product (OR) operators.
  - *Examples:* $a, \bar{a}, a+b+c, (\bar{a}+c)(b+d)$ are POS expressions.

Definitions of interest

- The ***canonical sum of products (CSOP)*** form of an expression refers to rewriting the expression as a sum of minterms.
  - *Examples for 3-variables:* $\bar{a}bc + abc$ is a CSOP expression; $\bar{a}b + c$ is not.
- The ***canonical product of sums (CPOS)*** form of an expression refers to rewriting the expression as a product of maxterms.
  - *Examples for 3-variables:* $(\bar{a}+b+c)(a+b+c)$ is a CPOS expression; $(\bar{a}+b)c$ is not.
- There is a close correspondence between the truth table and minterms and maxterms.

SOP: Sum of Products

Also known as **disjunctive normal form** or **minterm expansion**

Find all the input combinations (minterms) for which the output of the function is TRUE.

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

0 1 1   1 0 0   1 0 1   1 1 0

- Each row in a truth table has a minterm
- A minterm is a product (AND) of literals
- Each minterm is TRUE for that row (and only that row)

All Boolean equations can be written in SOP form

Contd.

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

$$F = \overset{0\ 1\ 1}{\overline{A}BC} + \overset{1\ 0\ 0}{A\overline{B}\,\overline{C}} + \boxed{\overset{1\ 0\ 1}{A\overline{B}C}} + \overset{1\ 1\ 0}{AB\overline{C}} + \overset{1\ 1\ 1}{ABC}$$

Activates this term

- Only the shaded product term — $A\overline{B}C$ $= 1 \cdot \overline{0} \cdot 1$ — will be 1

Contd.

- Standard "shorthand" notation
  - If we agree on the order of the variables in the rows of truth table…
    - then we can enumerate each row with the decimal number that corresponds to the binary number created by the input pattern

A	B	C	F	
0	0	0	0	
0	0	1	0	
0	1	0	0	
0	1	1	1	
1	0	0	1	**100 = decimal 4 so this is minterm #4, or m4**
1	0	1	1	
1	1	0	1	
1	1	1	1	**111 = decimal 7 so this is minterm #7, or m7**

f =

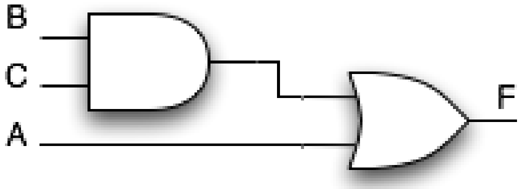
We can write this as a sum of products

Or, we can use a summation notation

Contd.

A	B	C	minterms
0	0	0	$\bar{A}\bar{B}\bar{C}$ = m0
0	0	1	$\bar{A}\bar{B}C$ = m1
0	1	0	$\bar{A}B\bar{C}$ = m2
0	1	1	$\bar{A}BC$ = m3
1	0	0	$A\bar{B}\bar{C}$ = m4
1	0	1	$A\bar{B}C$ = m5
1	1	0	$AB\bar{C}$ = m6
1	1	1	ABC = m7

Shorthand Notation for Minterms of 3 Variables



2-Level AND/OR Realization

F in canonical form:

$F(A,B,C) = \sum m(3,4,5,6,7)$

$= m3 + m4 + m5 + m6 + m7$

F =

canonical form ≠ minimal form

F

We are on the same page?

Row number	x_1	x_2	x_3	Minterm	Maxterm
0	0	0	0	$m_0 = \overline{x}_1\overline{x}_2\overline{x}_3$	$M_0 = x_1 + x_2 + x_3$
1	0	0	1	$m_1 = \overline{x}_1\overline{x}_2 x_3$	$M_1 = x_1 + x_2 + \overline{x}_3$
2	0	1	0	$m_2 = \overline{x}_1 x_2\overline{x}_3$	$M_2 = x_1 + \overline{x}_2 + x_3$
3	0	1	1	$m_3 = \overline{x}_1 x_2 x_3$	$M_3 = x_1 + \overline{x}_2 + \overline{x}_3$
4	1	0	0	$m_4 = x_1\overline{x}_2\overline{x}_3$	$M_4 = \overline{x}_1 + x_2 + x_3$
5	1	0	1	$m_5 = x_1\overline{x}_2 x_3$	$M_5 = \overline{x}_1 + x_2 + \overline{x}_3$
6	1	1	0	$m_6 = x_1 x_2\overline{x}_3$	$M_6 = \overline{x}_1 + \overline{x}_2 + x_3$
7	1	1	1	$m_7 = x_1 x_2 x_3$	$M_7 = \overline{x}_1 + \overline{x}_2 + \overline{x}_3$

POS: Product of Sum

Find all the input combinations (maxterms) for which the output of the function is FALSE.

Product of Sums (POS)

product

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

sums

Each sum term represents one of the "zeros" of the function

This input

$$\begin{array}{ccc} 0\ 0\ 0 & 0\ 0\ 1 & 0\ 1\ 0 \\ F = (A + B + C) & (A + B + \overline{C}) & (A + \overline{B} + C) \end{array}$$

Activates this term

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

For the given input, only the shaded sum term will equal 0

$$A + \overline{B} + C = 0 + \overline{1} + 0$$

Anything ANDed with 0 is 0; Output F will be 0

The function evaluates to FALSE (i.e., output is 0) if **any** of the Sums (maxterms) causes the output to be 0

Contd.

1. **Minterm to Maxterm conversion:**
   rewrite minterm shorthand using maxterm shorthand
   replace minterm indices with the indices not already used
   E.g., $\mathrm{F}(A,B,C) = \sum m(3,4,5,6,7) = \prod M(0,1,2)$

2. **Maxterm to Minterm conversion:**
   rewrite maxterm shorthand using minterm shorthand
   replace maxterm indices with the indices not already used
   E.g., $\mathrm{F}(A,B,C) = \prod M(0,1,2) = \sum m(3,4,5,6,7)$

3. **Expansion of** $\mathbf{F}$ **to expansion of** $\overline{F}$**:**

$$\mathrm{E.g.}, \mathrm{F}(A,B,C) = \sum m(3,4,5,6,7) \longrightarrow \overline{F}(A,B,C) = \sum m(0,1,2)$$
$$= \prod M(0,1,2) \longrightarrow = \prod M(3,4,5,6,7)$$

4. **Minterm expansion of** $\mathbf{F}$ **to Maxterm expansion of** $\overline{F}$**:**
   rewrite in Maxterm form, using the same indices as $\mathbf{F}$

$$\mathrm{E.g.}, \mathrm{F}(A,B,C) = \sum m(3,4,5,6,7) \longrightarrow \overline{F}(A,B,C) = \prod M(3,4,5,6,7)$$
$$= \prod M(0,1,2) \longrightarrow = \sum m(0,1,2)$$

Digital Logic ☹

Boring lectures will go away soon ☺

Once we jump into architecture

Till then "Hang in there"