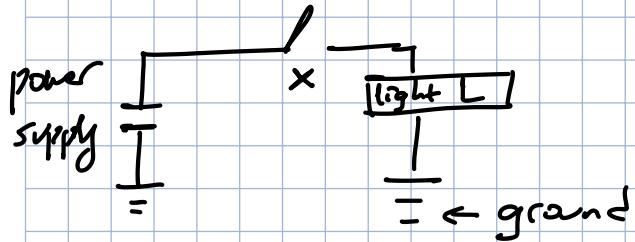
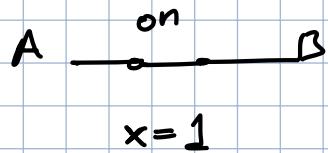
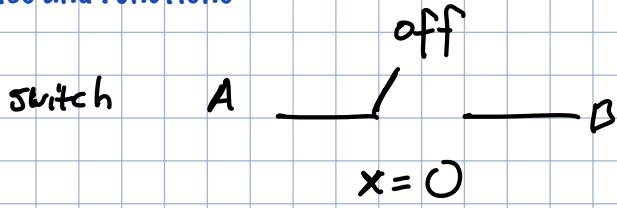


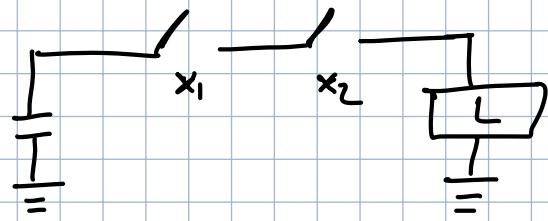
Chapter 2: Introduction to Logic Circuits

Variables and Functions



$$L = \begin{cases} 1 & \text{if } x = \text{on} \\ 0 & \text{if } x = \text{off} \end{cases}$$

$$L = x$$



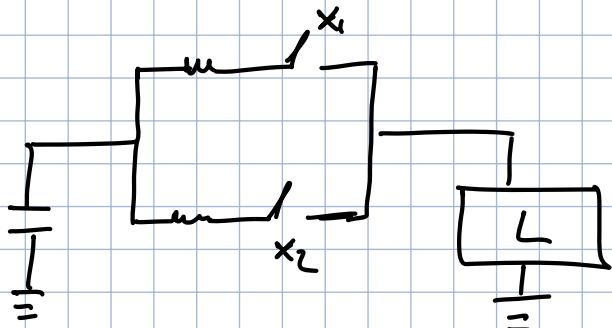
input		output
x ₁	x ₂	L
0	0	0
0	1	0
1	0	0
1	1	1

$$L = x_1 \text{ AND } x_2$$

$$= x_1 \cdot x_2$$

$$= x_1 x_2 \quad \text{not multiplication symbol.}$$

truth table

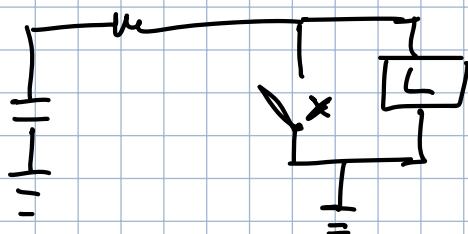


x ₁	x ₂	L
0	0	0
0	1	1
1	0	1
1	1	1

$$L = x_1 \text{ OR } x_2$$

$$\text{not addition symbol}$$

$$= x_1 + x_2$$



x	L
0	1
1	0

$$L = \text{NOT } x$$

$$= \bar{x} \quad (\text{logical inversion})$$

$$= x' \quad (\text{complement})$$

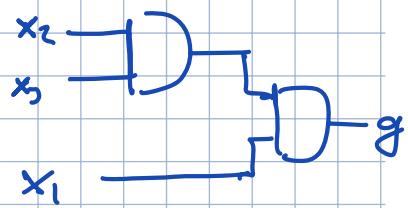
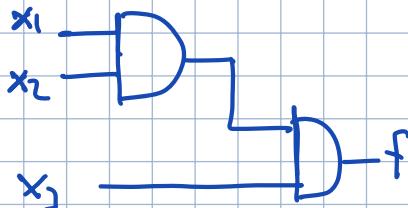
Truth Tables

$$f = (x_1 \cdot x_2) \cdot x_3$$

$$g = x_1 \cdot (x_2 \cdot x_3)$$

x_1	x_2	x_3	$x_1 \cdot x_2$	f	$x_2 \cdot x_3$	g
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	0	0	0	0
0	1	1	0	0	1	0
1	0	0	0	0	0	0
1	0	1	0	0	0	0
1	1	0	1	0	0	0
1	1	1	1	1	1	1

$$f = g$$

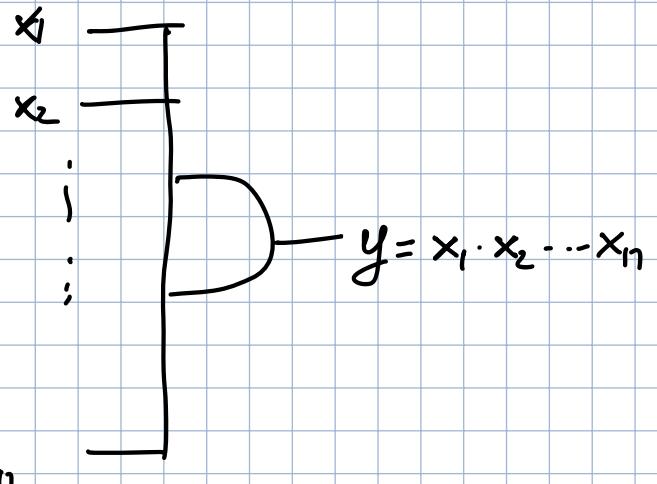
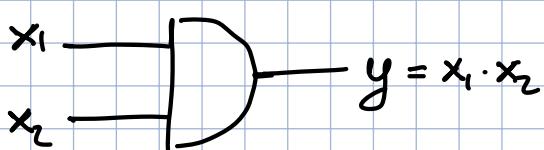


general form:

x_1	x_2	... x_n	$y = f(x_1, \dots, x_n)$
2^n rows {			

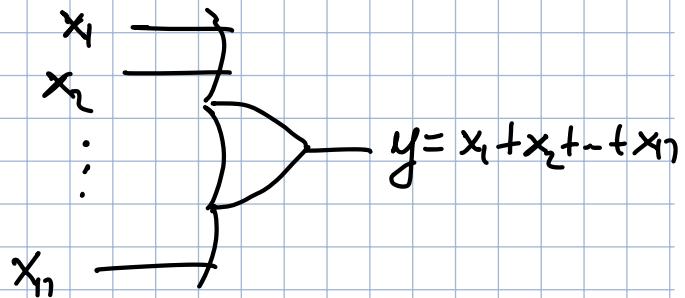
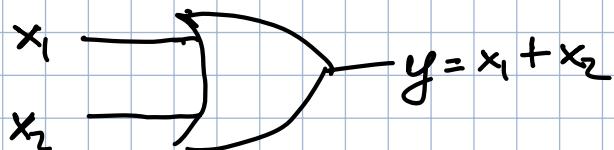
Logic Gates and Networks

AND



$$y = 1 \text{ iff } x_1 = x_2 = \dots = x_n = 1$$

OR

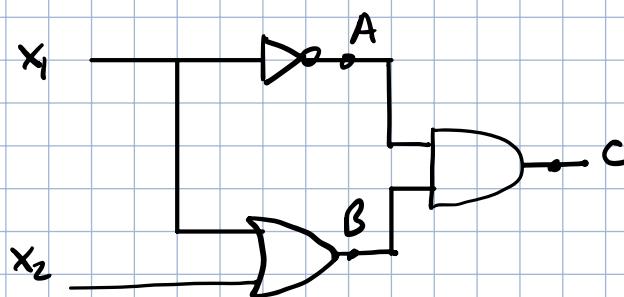


$$y=0 \text{ iff } x_1=x_2=\dots=x_n=0$$

NOT



Example: A logic network



$$A = \bar{x}_1$$

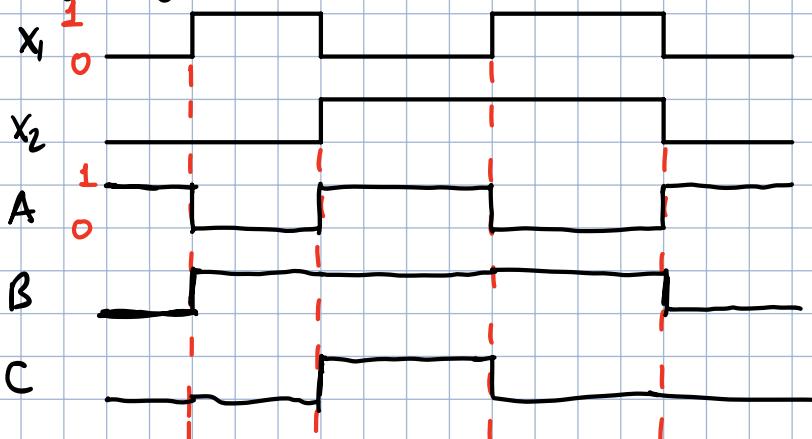
$$B = x_1 + x_2$$

$$C = A \cdot B = \bar{x}_1 \cdot (x_1 + x_2)$$

without parenthesis
Not > AND > OR

Timing diagram :

→ time



x1	x2	A	B	C
0	0	1	0	0
0	1	1	1	1
1	0	0	1	0
1	1	0	1	0

Boolean Algebra

1849 George Boole

1937 Claude Shannon

Axioms

1) $0 \cdot 0 = 0 \quad 1+1=1$

2) $1 \cdot 1 = 1 \quad 0+0=0$

3) $0 \cdot 1 = 1 \cdot 0 = 0 \quad 1+0=0+1=1$

4) if $x=0$ then $\bar{x}=1$ if $x=1$ then $\bar{x}=0$

duality: replace $+$ with \cdot , \cdot with $+$, $0 \leftrightarrow 1$, then all statements remain valid.

* Precedence of operations : in absence of parenthesis NOT > AND > OR

Single variable theorems

5) $x \cdot 0 = 0 \quad x+1=1 \quad \text{null elements}$

6) $x \cdot 1 = x \quad x+0=x \quad \text{identities}$

7) $x \cdot x = x \quad x+x=x \quad \text{idempotency}$

8) $x \cdot \bar{x} = 0 \quad x+\bar{x}=1 \quad \text{complements}$

9) $\bar{\bar{x}}=x \quad \text{involution}$

Proof of 6 by truth table

x	$x \cdot 1$	$x+0$
0	$0 \cdot 1=0$	$0+0=0$
1	$1 \cdot 1=1$	$1+0=1$

two columns are equal
proof done!

More theorems

$$10) x \cdot y = y \cdot x$$

$$x + y = y + x$$

commutation

$$11) x \cdot (y \cdot z) = (x \cdot y) \cdot z$$

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

association

$$12) x \cdot (y + z) = xy + xz$$

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

distribution

$$13) x + x \cdot y = x$$

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

absorption

$$14) x \cdot y + x \cdot \bar{y} = x$$

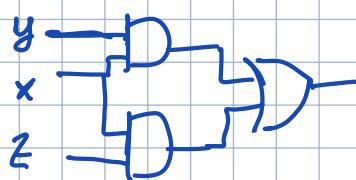
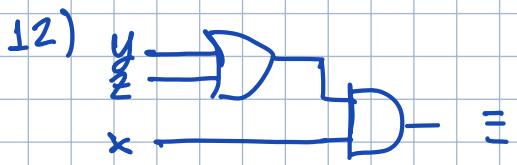
$$(x + y) \cdot (x + \bar{y}) = x$$

combining

$$15) \overline{x \cdot y} = \bar{x} + \bar{y}$$

$$\overline{x + y} = \bar{x} \cdot \bar{y}$$

De Morgan's theorem



15)

x	y	xy	\overline{xy}	$\bar{x} \cdot \bar{y}$	$\bar{x} + \bar{y}$
0	0	0	1	1	1
0	1	0	1	0	1
1	0	0	1	0	1
1	1	1	0	0	0

$$13) x + x \cdot y = x \cdot 1 + x \cdot y \stackrel{(6)}{=} x(1+y) \stackrel{(5)}{=} x \cdot 1 \stackrel{(6)}{=} x$$

$$14) xy + x\bar{y} \stackrel{(42)}{=} x(y + \bar{y}) \stackrel{(8)}{=} x \cdot 1 \stackrel{(6)}{=} x$$

$$15) x + \bar{x}y \stackrel{(42)}{=} (x + \bar{x}) \cdot (x + y) = 1 \cdot (x + y) = x + y$$

$$16) x + \bar{x}y = x + y \quad x \cdot (\bar{x} + y) = x \cdot y$$

$$17) x \cdot y + y \cancel{x} + \bar{x} \cdot z = x \cdot y + \bar{x} \cdot z$$

consensus

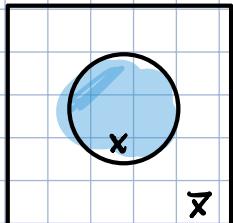
$$(x + y) \cdot (y + z) \cdot (\bar{x} + z) = (x + y) \cdot (\bar{x} + z)$$

Huntington's basic postulates : 5, 8, 10, 12 (other identities can be derived from them)

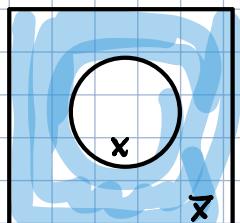
Example: Show that

$$\begin{aligned}
 Q &= XY + YZ + \bar{Y} = X + \bar{Y} + Z \\
 &= \underset{\bar{x}}{Y} \underset{y}{(X+Z)} + \underset{x}{\bar{Y}} \stackrel{(16)}{=} x + y = \bar{Y} + X + Z = X + \bar{Y} + Z
 \end{aligned}$$

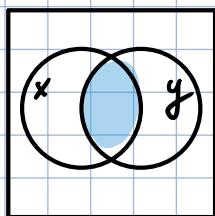
The Venn Diagram



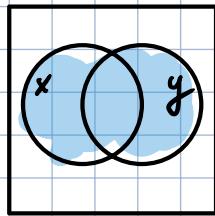
$$f = x$$



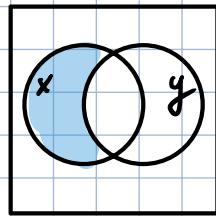
$$f = \bar{x}$$



$$f = x \cdot y$$

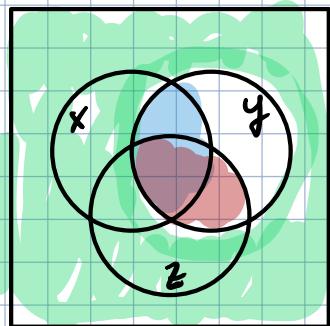


$$f = x + y$$

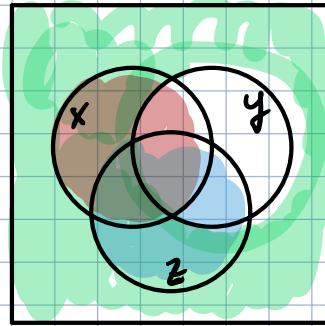


$$f = x\bar{y}$$

Check equivalence of $XY + YZ + \bar{Y} = \bar{Y} + X + Z$ using Venn diagram

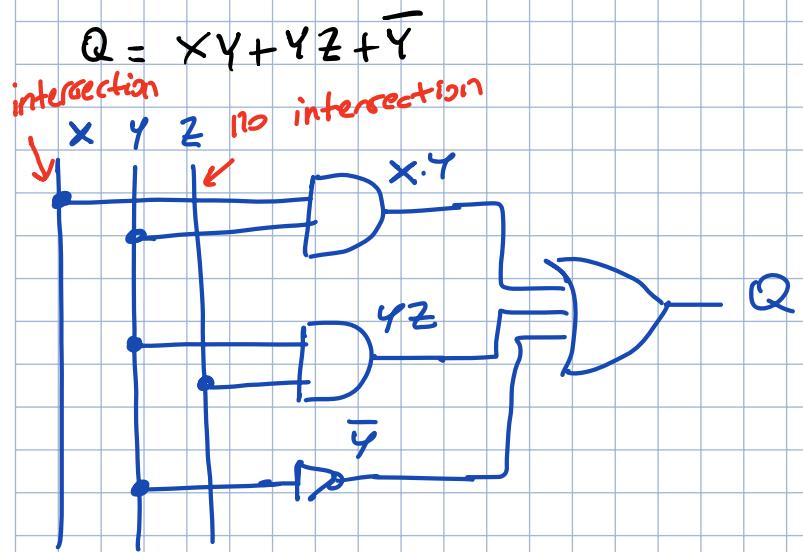


$$f = XY + YZ + \bar{Y}$$

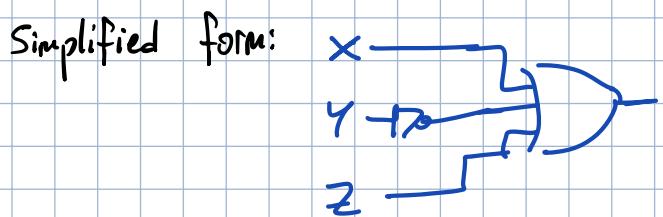


$$g = \bar{Y} + X + Z$$

Synthesis Using AND, OR, NOT gates



We know that $XY + YZ + \bar{Y} = \bar{Y} + X + Z$



saved 2, 2-input AND gates

Alternative:

Sum of products (SOP) and products of sums (POS) forms

minterm: For a function of n variables, i.e., $f(x_1, x_2, \dots, x_n)$, a product term in which each one of the variables appears once.

ex: $n=3$ for any $f(x_1, x_2, x_3)$

AND

$x_1 x_2 x_3, x_1 \bar{x}_2 \bar{x}_3, \bar{x}_1 x_2 \bar{x}_3$ are minterms
 $\# \text{of minterms} = 2^3 = 8$

maxterm: For a function of n variables, i.e., $f(x_1, x_2, \dots, x_n)$, a sum term in which each one of the variables appears once.

OR

ex: $n=3$ $x_1 + x_2 + x_3, x_1 + \bar{x}_2 + \bar{x}_3, \bar{x}_1 + x_2 + x_3$ are some maxterms
 $\# \text{of maxterms} = 2^3 = 8$

minterms, maxterms, and the truth table

Example : $n=3$

row	x_1 x_2 x_3	minterm	maxterm	m_0	M_0
0	0 0 0	$m_0 = \bar{x}_1 \bar{x}_2 \bar{x}_3$	$M_0 = x_1 + x_2 + x_3$	0	1
1	0 0 1	$m_1 = \bar{x}_1 \bar{x}_2 x_3$	$M_1 = x_1 + x_2 + \bar{x}_3$	0	1
2	0 1 0	$m_2 = \bar{x}_1 x_2 \bar{x}_3$	$M_2 = x_1 + \bar{x}_2 + x_3$	0	1
3	0 1 1	$m_3 = \bar{x}_1 x_2 x_3$	$M_3 = x_1 + \bar{x}_2 + \bar{x}_3$	0	1
4	1 0 0	$m_4 = x_1 \bar{x}_2 \bar{x}_3$	$M_4 = \bar{x}_1 + x_2 + x_3$	0	1
5	1 0 1	$m_5 = x_1 \bar{x}_2 x_3$	$M_5 = \bar{x}_1 + x_2 + \bar{x}_3$	0	1
6	1 1 0	$m_6 = x_1 x_2 \bar{x}_3$ (circled)	$M_6 = \bar{x}_1 + \bar{x}_2 + x_3$ (circled)	1	0
7	1 1 1	$m_7 = x_1 x_2 x_3$	$M_7 = \bar{x}_1 + \bar{x}_2 + \bar{x}_3$	0	1

Observation: $m_i = 1$ only on row i , 0 otherwise
 $M_i = 0$ " " " " , 1 "

$$\overline{M_6} = m_6$$

$$\overline{m_6} = M_6$$

$$\bar{M}_i = M_i^c, M_i = m_i$$

SOP Form

Two level. First AND, then OR.

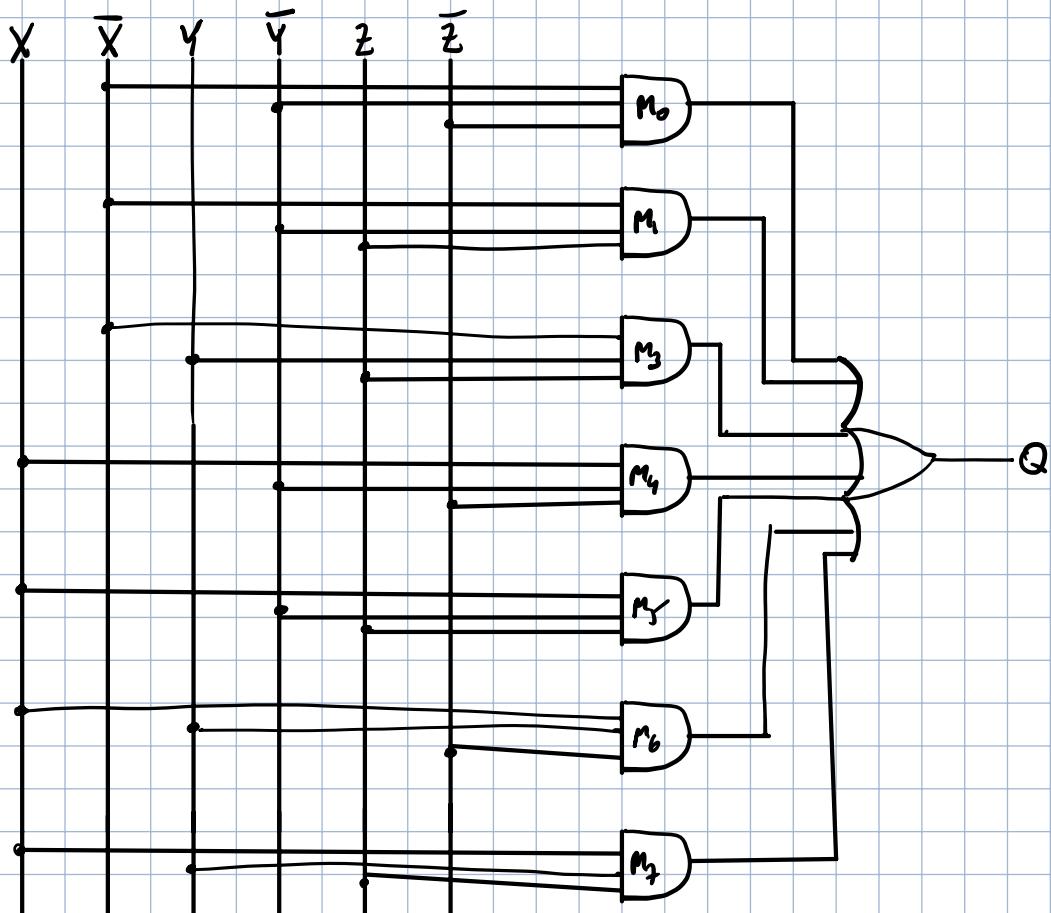
Canonical SOP: OR (sum) of minterms

$$Q = XY + YZ + \bar{Y} = X + \bar{Y} + Z$$

X	Y	Z	Q	m_0	m_1	m_2	m_3	m_4	m_5	m_6	m_7
0	0	0	1	1	0	0	0	0	0	0	0
0	0	1	1	0	1	0	0	0	0	0	0
0	1	0	0	0	0	1	0	0	0	0	0
0	1	1	1	0	0	0	1	0	0	0	0
1	0	0	1	0	0	0	0	1	0	0	0
1	0	1	1	0	0	0	0	0	1	0	0
1	1	0	1	0	0	0	0	0	0	1	0
1	1	1	1	0	0	0	0	0	0	0	1

$$Q = m_0 + m_1 + m_2 + m_3 + m_4 + m_6 + m_7 = \sum m(0, 1, 3, 4, 5, 6, 7)$$

Implementation:



POS Form

Two level. First OR, then AND.

Canonical POS: AND (product) of maxterms

X	Y	Z	P	\bar{P}
0	0	0	1	0
0	0	1	1	0
→ 0	1	0	0	1
0	1	1	1	0
1	0	0	1	0
→ 1	0	1	0	1
1	1	0	1	0
1	1	1	1	0

$$\bar{P} = M_2 + M_5$$

$$P = \overline{\bar{P}} = \overline{M_2 + M_5} = \overline{M_2} \cdot \overline{M_5} = M_2 \cdot M_5$$

010

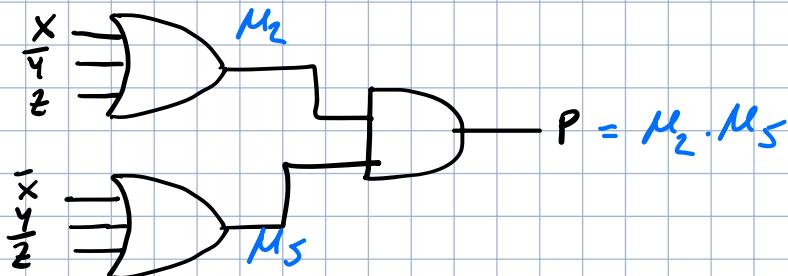
$x + \bar{y} + z$

101

$\bar{x} + y + \bar{z}$

Also/andie: product of maxterms for which $P=0$

Implementation



NAND and NOR gates



NAND

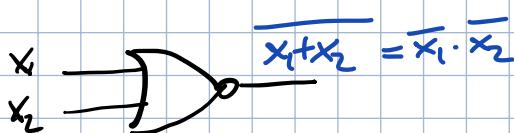


$$= \overline{\overline{X}_1 + \overline{X}_2} = \overline{X}_1 \cdot \overline{X}_2$$

X_1	X_2	$\overline{X_1 \cdot X_2}$
0	0	1
0	1	1
1	0	1
1	1	0

$$\begin{aligned} & \text{Implementation: } X_1, \dots, X_n \text{ enter a single NAND gate.} \\ & \quad \overline{X_1 \cdot X_2 \cdots X_n} = \overline{\overline{X}_1 + \overline{X}_2 + \cdots + \overline{X}_n} \\ & \quad = 0 \text{ iff } X_1 = X_2 = \cdots = X_n = 1 \\ & \quad = 1 \text{ otherwise} \end{aligned}$$

NOR



$$= \overline{X}_1 \cdot \overline{X}_2 = \overline{X_1 + X_2}$$

X_1	X_2	$\overline{X_1 + X_2}$
0	0	1
0	1	0
1	0	0
1	1	0

$$\begin{aligned} & \text{Implementation: } X_1, \dots, X_n \text{ enter a single NOR gate.} \\ & \quad \overline{X_1 + X_2 + \cdots + X_n} = \overline{X}_1 \cdot \overline{X}_2 \cdots \overline{X}_n \\ & \quad = 1 \text{ iff } X_1 = X_2 = \cdots = X_n = 0 \\ & \quad = 0 \text{ otherwise} \end{aligned}$$

* Easy implementation in CMOS

* NAND & NOR are functionally complete

NOT



all logic fn. can be implemented only with NAND gates
(or NOR)

AND $y = \overline{\overline{x} \cdot \overline{y}} = x \cdot y$

OR $x + y = \overline{\overline{x} \cdot \overline{y}} = x + y$

$$\begin{aligned} & \text{Implementation: } x, y \text{ enter a single NOR gate.} \\ & \quad \overline{x + y} = \overline{x} \cdot \overline{y} \\ & \quad = x + y \end{aligned}$$

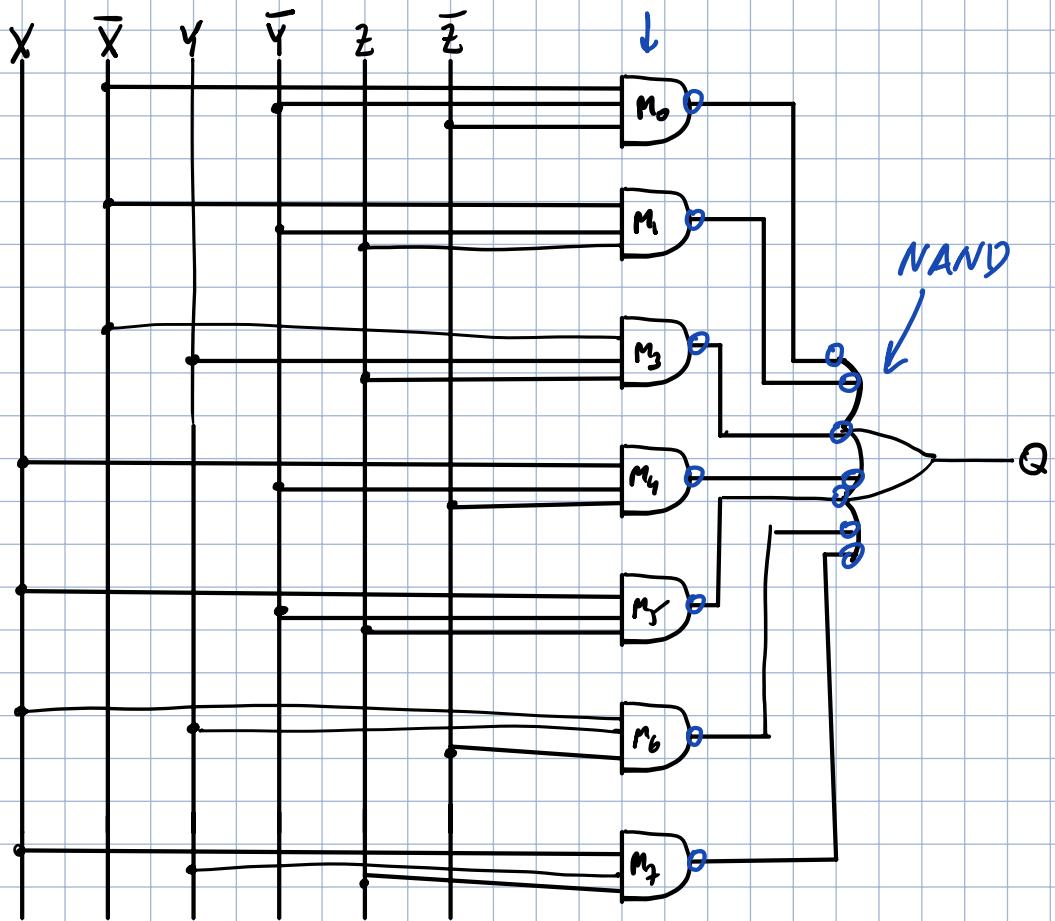
wire
1

SOP Design Using NAND

(sum of minterms)

$$\overline{\text{bubble}} = \overline{F} \quad \text{inverter}$$

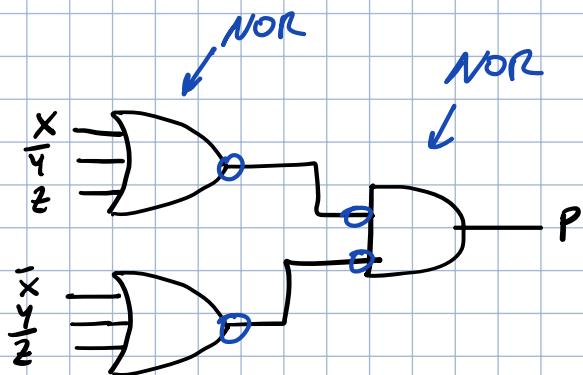
$$\overline{\text{bubble}} = \overline{F} + \overline{G}$$



NAND-NAND cheaper than AND-OR (next chapter)

two layer implementation (cage holder for SOP)

POS Design Using NOR

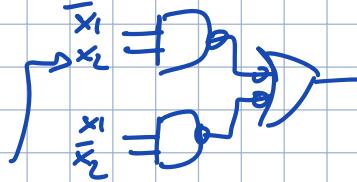
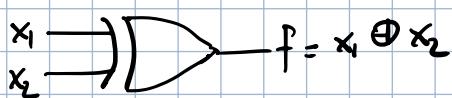


NOR-NOR implementation

Other Gates

XOR : Exclusive OR

x_1	x_2	$x_1 \oplus x_2$
0	0	0
0	1	1
1	0	1
1	1	0



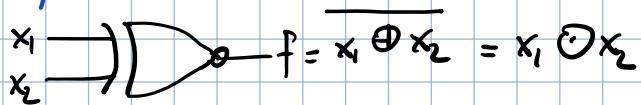
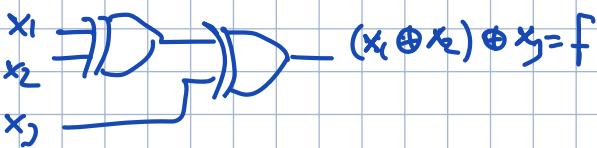
$$SOP \text{ of } XOR = \bar{x}_1 x_2 + x_1 \bar{x}_2 = m_1 + m_2$$

$$POS \text{ of } XOR = (x_1 + x_2) \cdot (\bar{x}_1 + \bar{x}_2) = M_0 \cdot M_3$$

XNOR: Exclusive NOR

x_1	x_2	$x_1 \oplus x_2$
0	0	1
0	1	0
1	0	0
1	1	1

Generalization of XOR: odd function



x_1	x_2	x_3	f
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

$$SOP \text{ of } XNOR: \bar{x}_1 \bar{x}_2 + x_1 x_2$$

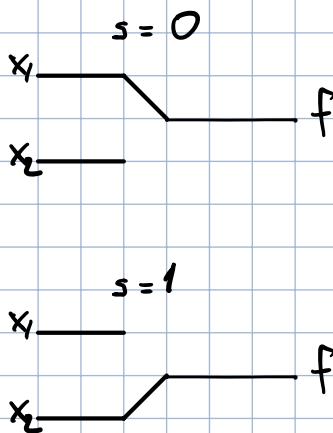
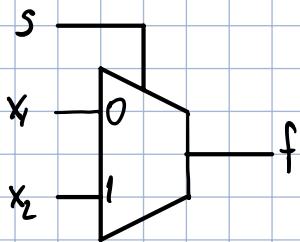
check if inputs are equal.

$$f = \overline{x_1 \oplus x_2} = \overline{m_1 + m_2} = \bar{m}_1 \cdot \bar{m}_2 = M_1 \cdot M_2$$

$$POS \text{ " " } : (x_1 + \bar{x}_2) \cdot (\bar{x}_1 + x_2)$$

Design Example

Multiplexer



truth table

S	X_1	X_2	f
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

$$\text{Canonical SOP} = m_2 + m_3 + m_5 + m_7 = \sum m(2, 3, 5, 7)$$

$$= \bar{S}X_1\bar{X}_2 + \bar{S}X_1X_2 + S\bar{X}_1X_2 + SX_1X_2$$

$$\text{Simplified SOP}$$

$$= \bar{S}X_1(\bar{X}_2 + X_2) + SX_2(\bar{X}_1 + X_1)$$

$$= \bar{S}X_1 + SX_2$$

$$\text{Canonical POS} = M_0 \cdot M_1 \cdot M_4 \cdot M_6$$

$$= (S + X_1 + X_2) \cdot (S + X_1 + \bar{X}_2) \cdot (\bar{S} + X_1 + X_2) \cdot (\bar{S} + \bar{X}_1 + X_2)$$

$$(a+b) \cdot (a+\bar{b}) = a \text{ dual } a \cdot b + a \cdot \bar{b} = a$$

Simplified POS

$$= (S + X_1) \cdot (\bar{S} + X_2)$$

VHDL (Very High Speed Integrated Circuit Hardware Description Language)

```
ENTITY example IS
    PORT(x1,x2,s: IN STD_LOGIC;
          f      :OUT STD_LOGIC);
END example;
```

↓ type

} declare inputs & outputs

```
ARCHITECTURE example_arch OF example IS
BEGIN
    f <= (s OR x1) AND ((NOT s) OR x2);
END example_arch;
```

assignment operator

} describe the function of the circuit

← = <=

$$(S + X_1) \cdot (\bar{S} + X_2)$$