



Chapter 1

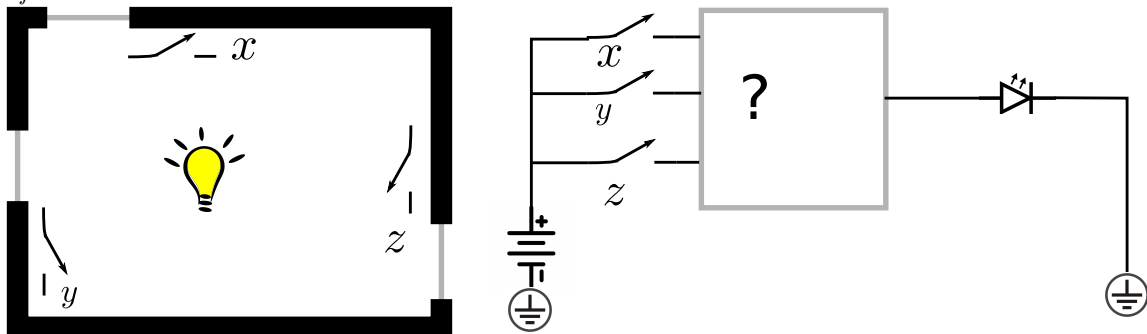
Boolean Algebra

1.1 Learning objectives

1. Representing digital circuits
2. Converting between different notations: Boolean expression, logic networks and switching circuits
3. Converting between different logic network specifications: truth table, minterm, maxterms, product of sums canonical form and sum of product canonical form.
4. Introduce truth tables as Behavioral Verilog
5. This handout has 11 homework problems totaling to 140 marks

1.2 Motivating Problem

Example 1.1. Assume that a large room has three doors and that a switch near each door controls a light in the room. It has to be possible to turn the light on or off by changing the state of any one of the switches.



Name	C/Verilog	Boolean expr.	Truth Table	Switching circuit	(ANSI) symbol	Venn diagram															
AND Gate	L = x1 & x2	$L = x_1 \cdot x_2$	<table><tr><th>x_1</th><th>x_2</th><th>$x_1 \cdot x_2$</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	x_1	x_2	$x_1 \cdot x_2$	0	0	0	0	1	0	1	0	0	1	1	1			
x_1	x_2	$x_1 \cdot x_2$																			
0	0	0																			
0	1	0																			
1	0	0																			
1	1	1																			
OR Gate	L = x1 x2	$L = x_1 + x_2$	<table><tr><th>x_1</th><th>x_2</th><th>$x_1 + x_2$</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	x_1	x_2	$x_1 + x_2$	0	0	0	0	1	1	1	0	1	1	1	1			
x_1	x_2	$x_1 + x_2$																			
0	0	0																			
0	1	1																			
1	0	1																			
1	1	1																			
NOT Gate	L = ~ x1	$L = \bar{x}_1 = x'_1$	<table><tr><th>x_1</th><th>\bar{x}_1</th></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table>	x_1	\bar{x}_1	0	1	1	0												
x_1	\bar{x}_1																				
0	1																				
1	0																				

Boolean algebra and gates

3 basic gates

AND gate

OR gate

NOT gate

Boolean Algebra $\left(\begin{matrix} x \in \{0,1\} \\ y \in \{0,1\} \end{matrix} \right)$

$$x \cdot y$$

$$x + y$$

$$\overline{x}$$

$$0 \equiv \text{FALSE}$$

$$1 \equiv \text{TRUE } x$$

(UMAINE IS A TOP SCHOOL)
(UMAINE SPORTS IS THE BEST)
 y

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

Aside : $\underbrace{\{0,1\}}_{\text{set}}$
 $\in \leftarrow$ in / is an element of
 $x \in \{0,1\}$
 The variable x is an element the set $\{0,1\}$

Truth table

x	y	$x \cdot y$
0	0	0
0	1	0
1	0	0
1	1	1

Truth Table for OR gate

x	y	$x + y$
0	0	0
0	1	1
1	0	1
1	1	1

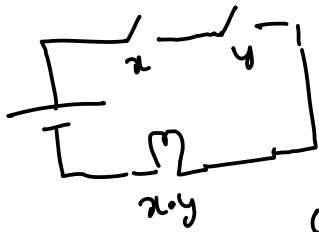
Truth table for NOT gate

x	\overline{x}
0	1
1	0

NOT (UMAINE IS A TOP SCHOOL)

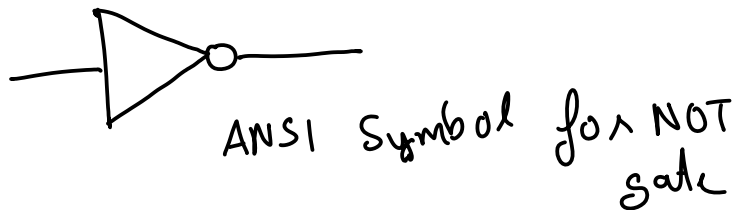
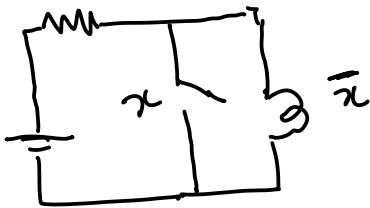
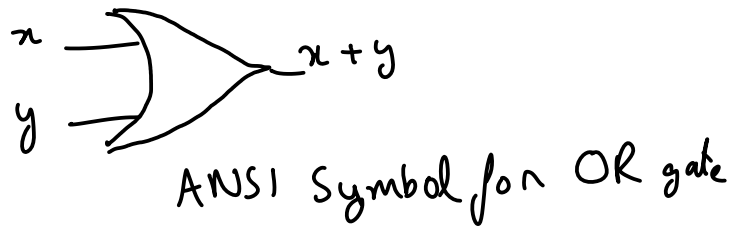
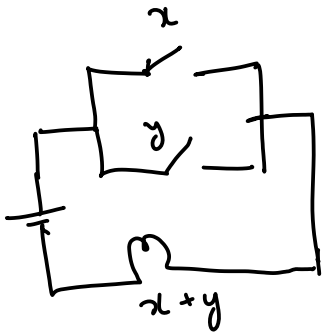
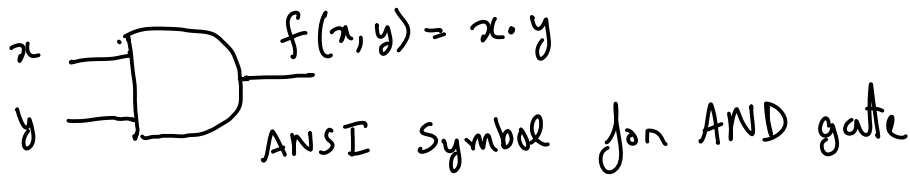
Digital circuits

AND/OR/NOT

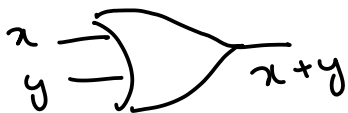


The bulb should turn on only if both switches are on (AND gate). Should we connect the switches in series or parallel?

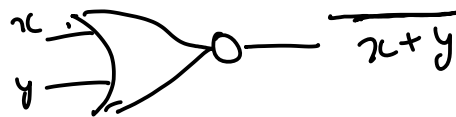
One possible implementation of AND gate



OR

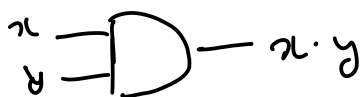


NOR

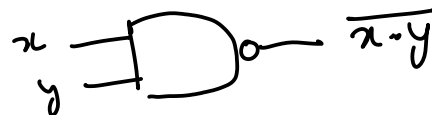


x	y	x+y	$\overline{x+y}$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

AND



NAND



x	y	x.y	$\overline{x.y}$
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

C / System Verilog

AND

OR

NOT

$x \& y$ bitwise

$x | y$

$\sim x$
↑ tilde

$x \&\& y$ logical

$x || y$

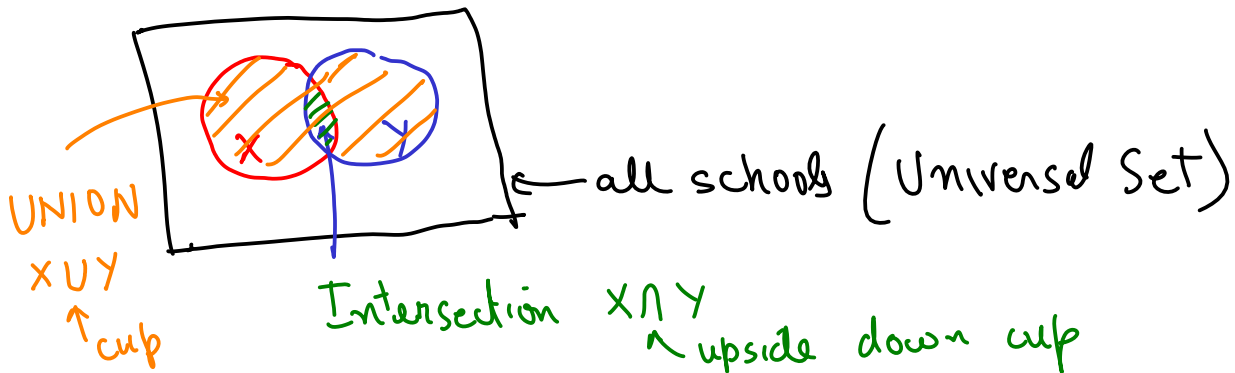
$! x$

Venn Diagram \rightarrow Karnaugh Maps = (Truth table + Venn Diagram)

x is a top school

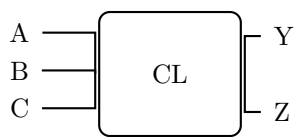
$x \in X$ is a set of all top schools \rightarrow x is a top school

$y \in Y$ is a set of all top sports teams \rightarrow y has a top sports team

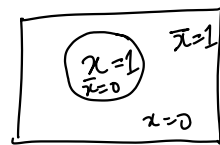


Intersections correspond to AND operations
Unions correspond to OR operations

1.3 Digital circuits or networks

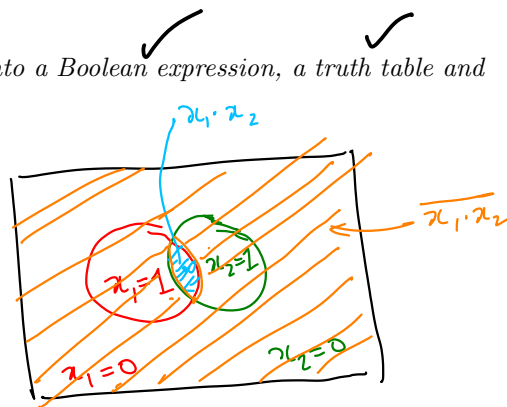
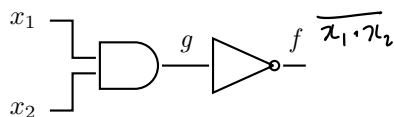


$$Y = F(A, B, C) \quad Z = G(A, B, C)$$

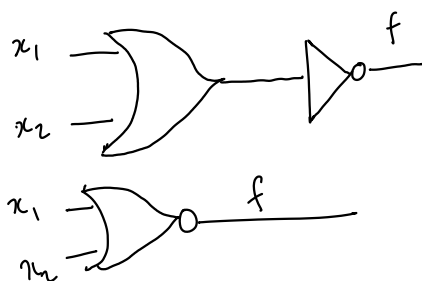


1.4 Two input networks

Example 1.2. Convert the following (ANSI) network into a Boolean expression, a truth table and a Venn diagram.

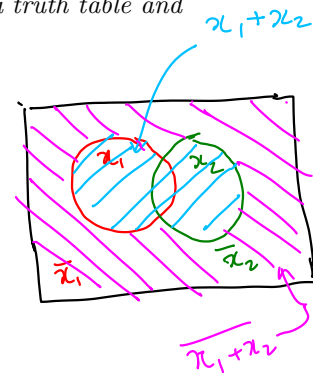


Example 1.3. Convert the following Boolean expression into a (ANSI) network, a truth table and a Venn diagram:

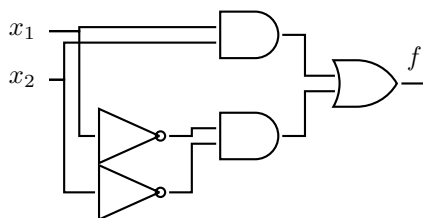


$$f = \overline{x_1 + x_2}$$

x_1	x_2	$x_1 + x_2$	$\overline{x_1 + x_2}$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0



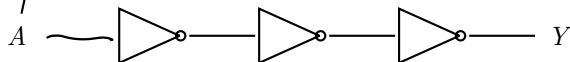
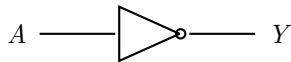
Problem 1.1 (10 marks). Convert the following (ANSI) network into a Boolean expression, a truth table and a Venn diagram.



Example 1.4. Convert the following Boolean expression into a network, a truth table and a Venn diagram:

$$f = x_1\bar{x}_2 + \bar{x}_1x_2$$

Problem 1.2 (5 marks). Can two different circuits have the same truth table? Can two different truth tables have the same circuit? Consider the following two circuits for example



ANSI networks / Boolean expressions

How

Analysis

Truth table / Venn Diagram

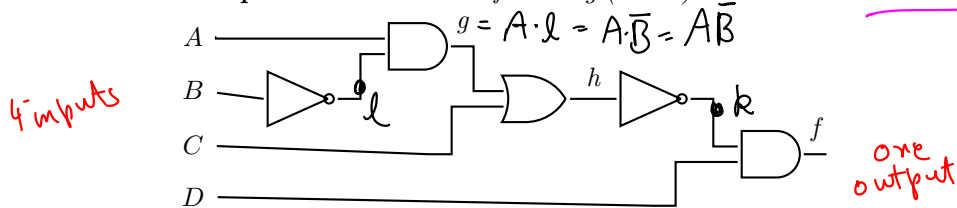
what

Output / Input relationship

Remark 1.1. Truth tables and Venn diagrams define what the combinational circuit should do. Truth tables define output for every input. Boolean expression and networks define how to achieve the desired input output relationship.

1.5 Multi-input networks

Example 1.5. Convert the following (ANSI) network into a Boolean expression and a truth table.



$$l = \bar{B}$$

$$g = A\bar{B}$$

$$h = g + C = A\bar{B} + C$$

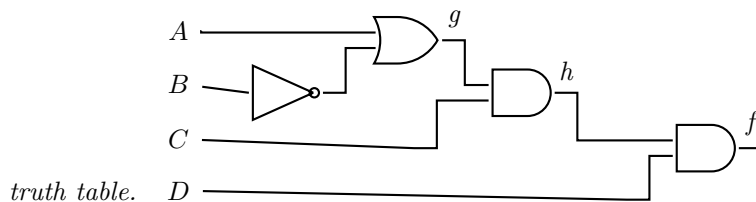
$$k = \bar{h} = \overline{A\bar{B} + C} = \overline{A\bar{B}} \cdot \bar{C} \neq \overline{A\bar{B}} + \bar{C}$$

$$f = k \cdot D = \overline{(A\bar{B} + C)} D$$

Truth table

#	A	B	C	D	\bar{B}	$A\bar{B}$	$A\bar{B} + C$	$\overline{A\bar{B} + C}$
0	0	0	0	0	1	0	0	1
1	0	0	0	1	1	0	0	1
2	0	0	1	0	1	0	1	0
3	0	0	1	1	1	0	1	0
4	0	1	0	0	0	0	0	1
5	0	1	0	1	0	0	0	1
6	0	1	1	0	0	0	1	0
7	0	1	1	1	0	0	1	0
8	1	0	0	0	1	1	1	0
9	1	0	0	1	1	1	1	0
10	1	0	1	0	1	1	1	0
11	1	0	1	1	1	1	1	0
12	1	1	0	0	0	1	1	0
13	1	1	0	1	0	1	1	0
14	1	1	1	0	0	1	1	0
15	1	1	1	1	0	1	1	0

Problem 1.3 (20 marks). Convert the following (ANSI) network into a Boolean expression and a



Truth table

#	A	B	C	D	\bar{B}	$A\bar{B}$	$A\bar{B}+C$	$A\bar{B}C$	$(A\bar{B}+C) \cdot D$
0	0	0	0	0	1	0	0	0	0
1	0	0	0	1	1	0	0	0	0
2	0	0	1	0	1	0	1	0	0
3	0	0	1	1	1	0	1	0	0
4	0	1	0	0	0	0	0	0	0
5	0	1	0	1	0	0	0	0	0
6	0	1	1	0	0	0	1	0	0
7	0	1	1	1	0	0	1	0	0
8	1	0	0	0	1	1	1	0	0
9	1	0	0	1	1	1	1	0	0
10	1	0	1	0	1	1	1	1	0
11	1	0	1	1	1	1	1	1	0
12	1	1	0	0	0	0	0	0	0
13	1	1	0	1	0	0	0	0	0
14	1	1	1	0	0	0	1	0	0
15	1	1	1	1	0	0	1	1	0

1.6 Minterms and Maxterms

1.6.1 Minterms

Minterm is a product involving all inputs (or complements) to a function. Every row of a truth table has a corresponding minterm. Minterm is true if and only if the corresponding row in the table is active.

Minterms defined as follows for each row of a two input truth table:

	A	B	minterm	minterm name	A	B	$\bar{A}B = m_1$	m_3	$m_1 + m_3$
0	0	0	$\bar{A}\bar{B}$	m_0	0	0	0	0	0
1	0	1	$\bar{A}B$	m_1	0	1	1	0	1
2	1	0	$A\bar{B}$	m_2	1	0	0	0	0
3	1	1	AB	m_3	1	1	0	1	1

Consider a two input circuit whose output Y is given by the truth table:

A	B	Y	minterm	minterm name
0	0	0	$\bar{A}\bar{B}$	m_0
0	1	1	$\bar{A}B$	m_1
1	0	0	$A\bar{B}$	m_2
1	1	1	AB	m_3

then $Y = \bar{A}B + AB = m_1 + m_3 = \sum(1, 3)$.

This also gives the sum of products canonical form.

Example 1.6. What is the minterm m_{13} for a 4-input circuit with inputs x, y, z, w (ordered from MSB to LSB) $x y z w$

$$(13)_{10} \rightarrow (1101)_2 \leftarrow \text{LSB}$$

MSB

$$m_{13} = x y \bar{z} w$$

MSB
Most Significant Bit
LSB
Least Significant Bit

Problem 1.4 (5 marks). What is the minterm m_{23} for a 5-input circuit with inputs a, b, c, d, e (ordered from MSB to LSB).

$$m_{23} = \bar{a} b c d e$$

Example 1.7. Convert the following 4-input truth table into sum of minterms and sum of products canonical form.

minterm name	A	B	C	D	f
m_0	0	0	0	0	0
m_1	0	0	0	1	1
m_2	0	0	1	0	0
m_3	0	0	1	1	0
m_4	0	1	0	0	0
m_5	0	1	0	1	1
m_6	0	1	1	0	0
m_7	0	1	1	1	0
m_8	1	0	0	0	0
m_9	1	0	0	1	0
m_{10}	1	0	1	0	0
m_{11}	1	0	1	1	0
m_{12}	1	1	0	0	0
m_{13}	1	1	0	1	1
m_{14}	1	1	1	0	0
m_{15}	1	1	1	1	0

Sum of minterms

$$f = m_1 + m_5 + m_{13} = \sum (1, 5, 13)$$

$$= \sum m(1, 5, 13)$$

Sum of products canonical form

$$f = A\bar{B}\bar{C}D + \bar{A}B\bar{C}D + AB\bar{C}D$$

$5_{10} \rightarrow 0101$
A B C D

Problem 1.5 (10 marks). Convert the following 4-input truth table into sum of minterms and sum of products canonical form.

minterm name	A	B	C	D	f
m_0	0	0	0	0	0
m_1	0	0	0	1	0
m_2	0	0	1	0	0
m_3	0	0	1	1	1
m_4	0	1	0	0	0
m_5	0	1	0	1	0
m_6	0	1	1	0	0
m_7	0	1	1	1	1
m_8	1	0	0	0	0
m_9	1	0	0	1	0
m_{10}	1	0	1	0	0
m_{11}	1	0	1	1	1
m_{12}	1	1	0	0	0
m_{13}	1	1	0	1	1
m_{14}	1	1	1	0	1
m_{15}	1	1	1	1	0

1.6.2 Maxterms (Dual of minterms)

Maxterm is a sum involving all inputs (or complements) to a function. Every row of a truth table has a corresponding maxterm. ~~Minterm~~ ^{maxterm} is false if and only if the corresponding row in the table is active.

Maxterms are defined as follows for each row of a two input truth table:

A	B	maxterm	maxterm name
0	0	$A + B$	M_0
0	1	$A + \bar{B}$	M_1
1	0	$\bar{A} + B$	M_2
1	1	$\bar{A} + \bar{B}$	M_3

		M_0	M_2
A	B	$A + B$	$\bar{A} + B$
0	0	0	1
0	1	1	1
1	0	1	0
1	1	1	1

Consider a two input circuit whose output Y is given by the truth table:

A	B	Y	maxterm	maxterm name
0	0	0	$A + B$	M_0
0	1	1	$A + \bar{B}$	M_1
1	0	0	$\bar{A} + B$	M_2
1	1	1	$\bar{A} + \bar{B}$	M_3

$$M_2 = \bar{A} + B$$

$A \quad B$

$$(2)_{10} \rightarrow (10)_2$$

then $Y = (A + B)(\bar{A} + B) = M_0 M_2$.

Writing a functional specification in terms of minterms is also called product of sums canonical form.

Example 1.8. Write the maxterm M_{11} for 4-input Boolean function with the ordered inputs A, B, C, D . \leftarrow LSB $(11)_{10} = (1011)_2$ $M_{11} = \bar{A} + B + \bar{C} + \bar{D}$

Example 1.9. Convert the following 4-input truth table into product of maxterms and product of sums canonical form.

maxterm name	A	B	C	D	f
M_0	0	0	0	0	0
M_1	0	0	0	1	0
M_2	0	0	1	0	0
M_3	0	0	1	1	1
M_4	0	1	0	0	0
M_5	0	1	0	1	0
M_6	0	1	1	0	0
M_7	0	1	1	1	1
M_8	1	0	0	0	0
M_9	1	0	0	1	0
M_{10}	1	0	1	0	0
M_{11}	1	0	1	1	1
M_{12}	1	1	0	0	0
M_{13}	1	1	0	1	1
M_{14}	1	1	1	0	1
M_{15}	1	1	1	1	0

$$f = M_0 M_1 M_2 M_4 M_5 M_6 M_8 M_9 M_{10} M_{12} M_{15}$$

$$= \prod M(0, 1, 2, 4, 5, 6, 8, 9, 10, 12, 15)$$

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

$$(A + \bar{B} + C + \bar{D})(A + \bar{B} + \bar{C} + D)(\bar{A} + B + C + D)(\bar{A} + B + C + \bar{D})$$

$$(\bar{A} + B + \bar{C} + D)(\bar{A} + \bar{B} + C + D)(\bar{A} + \bar{B} + \bar{C} + \bar{D})$$

Product of sums canonical form

Problem 1.6 (10 marks). Convert the following 4-input truth table into product of maxterms and products of sums canonical form.

masterm name	A	B	C	D	f
M_0	0	0	0	0	0
M_1	0	0	0	1	1
M_2	0	0	1	0	1
M_3	0	0	1	1	1
M_4	0	1	0	0	1
M_5	0	1	0	1	0
M_6	0	1	1	0	1
M_7	0	1	1	1	1
M_8	1	0	0	0	0
M_9	1	0	0	1	1
M_{10}	1	0	1	0	1
M_{11}	1	0	1	1	1
M_{12}	1	1	0	0	0
M_{13}	1	1	0	1	1
M_{14}	1	1	1	0	1
M_{15}	1	1	1	1	0

Example 1.10. Write the 3-input truth table for the function $f = m_2 + m_3 + m_7$.

\overline{x}	y	z		
#	x	y	z	f
0	0	0	0	0
1	0	0	1	0
2	0	1	0	1
3	0	1	1	1
4	1	0	0	0
5	1	0	1	0
6	1	1	0	0
7	1	1	1	1

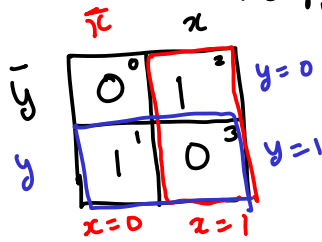
Problem 1.7 (10 marks). Write the 3-input truth table for the function $f = M_4 M_5 M_7$.

Problem 1.8 (10 marks). Write the truth table for the function $f = \bar{A}B\bar{C} + AB\bar{C}$.

1.7 Karnaugh maps (Truth table + Venn Diagram)

Two input K-maps

	x, y	f
m ₀	0 0	0
m ₁	0 1	1
m ₂	1 0	1
m ₃	1 1	0

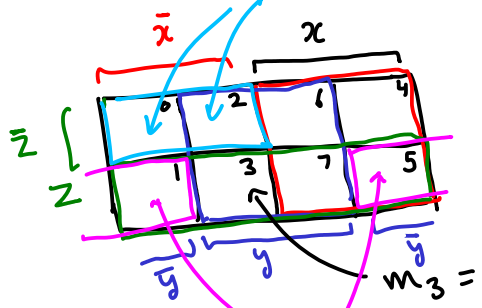


Helpful in simplifying Boolean expressions
Digital circuits

Three input K-maps

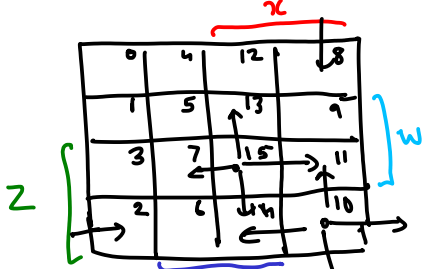
MSB → LSB

x	y	z	f
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

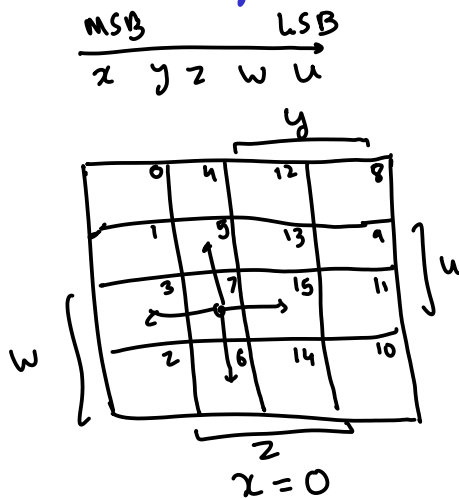


Each rectangular grouping of 2, 4, 8, 16, ... corresponds to a product term

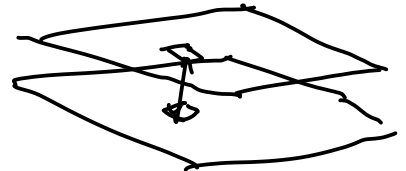
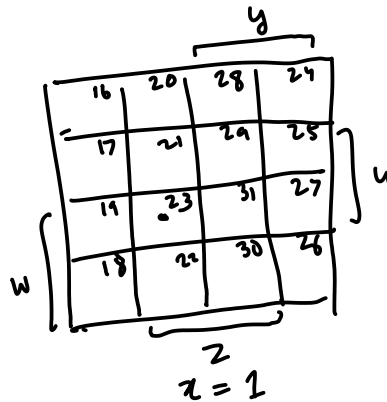
Four input K-maps



Five input K-maps



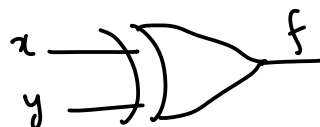
4 input k-map must have 4-neighbors for each cell
K-map wraps around



1.8 More Gates and notations summary

XOR = Exclusive OR gate

x	y	f
0	0	0
0	1	1
1	0	1
1	1	0



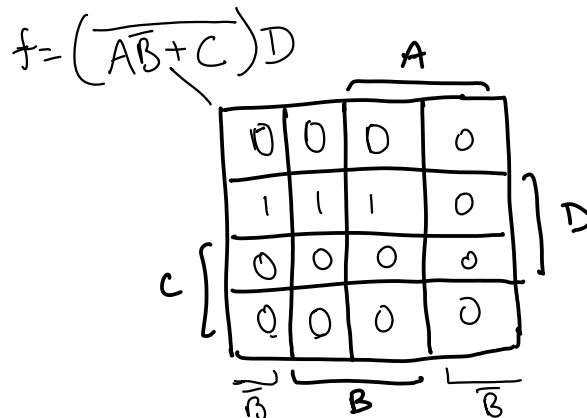
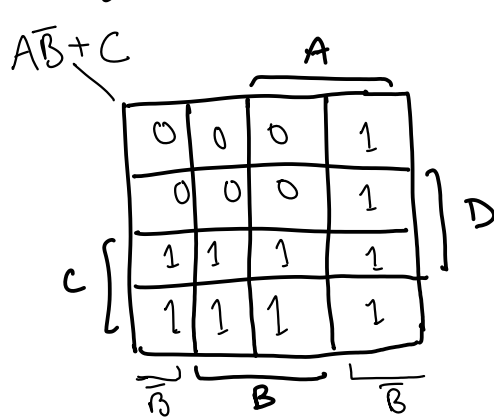
$$f = x \oplus y$$

$$f = x \sim y \quad \text{in C/Verilog}$$

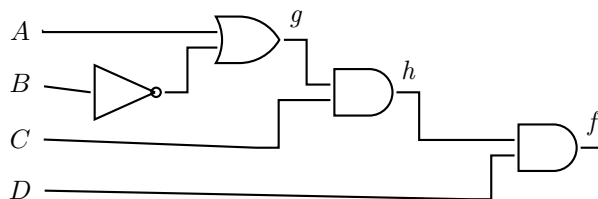
Name	C/Verilog	Boolean expr.	Truth Table	(ANSI) symbol	K-map															
NAND Gate	$Q = \sim (x1 \ \& \ x2)$	$Q = \overline{x_1 \cdot x_2} = \overline{x_1}x_2 + x_1\overline{x_2}$	<table><tr><th>x_1</th><th>x_2</th><th>$\overline{x_1 \cdot x_2}$</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	x_1	x_2	$\overline{x_1 \cdot x_2}$	0	0	1	0	1	1	1	0	1	1	1	0		
x_1	x_2	$\overline{x_1 \cdot x_2}$																		
0	0	1																		
0	1	1																		
1	0	1																		
1	1	0																		
NOR Gate	$Q = \sim (x1 \ \ x2)$	$Q = \overline{x_1 + x_2}$	<table><tr><th>x_1</th><th>x_2</th><th>$\overline{x_1 + x_2}$</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	x_1	x_2	$\overline{x_1 + x_2}$	0	0	1	0	1	0	1	0	0	1	1	0		
x_1	x_2	$\overline{x_1 + x_2}$																		
0	0	1																		
0	1	0																		
1	0	0																		
1	1	0																		
XOR Gate	$Q = x1 \ \wedge \ x2$	$Q = x_1 \oplus x_2$	<table><tr><th>x_1</th><th>x_2</th><th>$x_1 \oplus x_2$</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	x_1	x_2	$x_1 \oplus x_2$	0	0	0	0	1	1	1	0	1	1	1	0		
x_1	x_2	$x_1 \oplus x_2$																		
0	0	0																		
0	1	1																		
1	0	1																		
1	1	0																		
XNOR Gate	$Q = \sim (x1 \ \wedge \ x2)$	$Q = \overline{x_1 \oplus x_2}$	<table><tr><th>x_1</th><th>x_2</th><th>$\overline{x_1 \oplus x_2}$</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	x_1	x_2	$\overline{x_1 \oplus x_2}$	0	0	1	0	1	0	1	0	0	1	1	1		
x_1	x_2	$\overline{x_1 \oplus x_2}$																		
0	0	1																		
0	1	0																		
1	0	0																		
1	1	1																		

Example 1.11. Convert the following Boolean expression into a K-map. $f = (\overline{AB} + C)D$

msb $\xrightarrow{\text{LSB}}$
 $A \ B \ C \ D$ 4-input K-map



Problem 1.9 (10 marks). Convert the following logic circuit into a K-map.



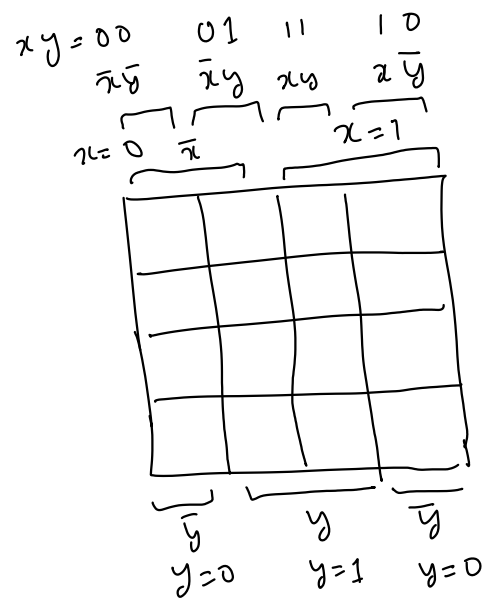
1.9 Boolean Algebra

1.9.1 Axioms of Boolean algebra

1. $0 \cdot 0 = 0$
2. $1 + 1 = 1$

Definitions

\neq Axioms are fundamental truths that you take to be true and do not ask proofs for.



$$3. 1 \cdot 1 = 1$$

$$4. 0 + 0 = 0$$

$$5. 0 \cdot 1 = 1 \cdot 0 = 0$$

$$6. \bar{0} = 1$$

$$7. \bar{1} = 0$$

$$8. x = 0 \text{ if } x \neq 1$$

$$9. x = 1 \text{ if } x \neq 0$$

1.9.2 Single variable theorems (Prove by drawing K-maps)

$$\begin{array}{l}
 1. \ x \cdot 0 = 0 \\
 x \quad \begin{array}{l} \nearrow 1 \\ \searrow 0 \end{array} \quad \begin{array}{l} 1 \cdot 0 = 0 \\ 0 \cdot 0 = 0 \end{array} \quad \begin{array}{l} \text{by axiom (5)} \\ \text{by axiom (1)} \end{array}
 \end{array}$$

$$2. x + 1 = 1$$

$$3. x \cdot 1 = x$$

$$4. x + 0 = x$$

$$5. x \cdot x = x$$

$$6. x + x = x$$

$$7. x \cdot \bar{x} = 0$$

$$x + \overline{x} = 1$$

$$\overline{\overline{x}} = x$$

5. Combining: $x \cdot y + x \cdot \overline{y} = x \cdot (y + \overline{y}) = x$

$$\text{LHS} = x \cdot y + x \cdot \overline{y} = x \cdot (y + \overline{y}) = x \cdot 1 = x$$

6. DeMorgan's theorem: $\overline{x \cdot y} = \overline{x} + \overline{y}$, $\overline{x + y} = \overline{x} \cdot \overline{y}$.

7. Concensus:

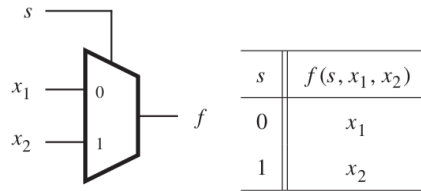
(a) $x + \overline{x} \cdot y = x + y$

(b) $x \cdot (\overline{x} + y) = x \cdot y$

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

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

Example 1.12 (Multiplexer). *Multiplexer is a circuit used to select one of the input lines x_1 and x_2 based only select input s . When $s = 0$, x_1 is selected, x_2 is selected otherwise. Find a boolean expression and a circuit for multiplexer*



Example 1.13. *Simplify $f = \bar{A}\bar{B}\bar{C} + A\bar{B}\bar{C} + A\bar{B}C$ using boolean algebra.*

1.1 - 1.9

Problem 1.10 (30 marks, Exercise 2.14 [1]). *Simplify the following Boolean equations using Boolean theorems.*

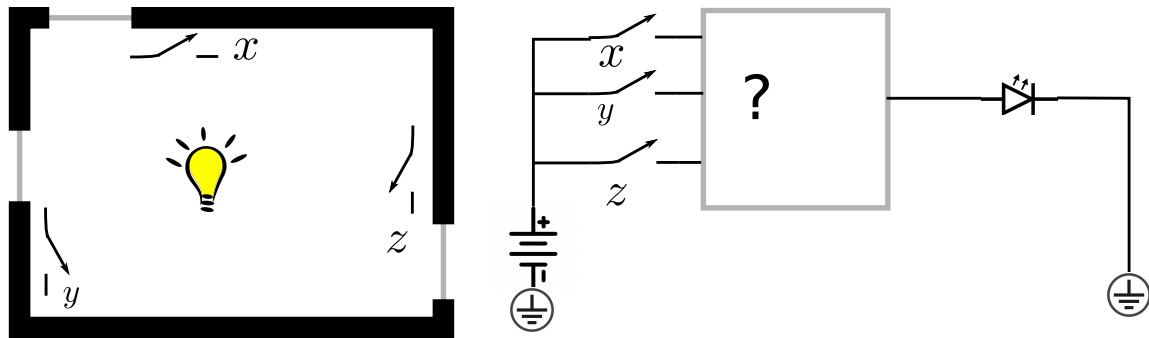
$$Y = \bar{A}BC + \bar{A}B\bar{C} \quad (1.1)$$

$$Y = \overline{ABC} + A\bar{B} \quad (1.2)$$

$$Y = ABC\bar{D} + \overline{ABCD} + \overline{(A + B + C + D)} \quad (1.3)$$

Example 1.14. *Simplify $f = \bar{A}\bar{A}\bar{C} + \bar{A}\bar{B}C$ using K-maps.*

Example 1.15. *Assume that a large room has three doors and that a switch near each door controls a light in the room. It has to be possible to turn the light on or off by changing the state of any one of the switches.*



Problem 1.11 (20 marks, Exercise 2.38 [1]). An M -bit thermometer code for the number k consists of k 1's in the least significant bit positions and $M - k$ 0's in all the more significant bit positions. A binary-to-thermometer code converter has N inputs and $2^N - 1$ outputs. It produces a $2^N - 1$ bit thermometer code for the number specified by the input. For example, if the input is 110, the output should be 0111111. Design a 3:7 binary-to-thermometer code converter. Give a simplified Boolean equation for each output.