Representing Circuits

Module 2

UGTA - Corey Lancelotta

Key Takeaways

LOGIC GATES & THEIR TRUTH TABLES

Identifying Truth Table Values of Logic Gates

CIRCUIT <--> BOOLEAN EQUATION

Identifying Intermediate Values & Focusing on Operators

TRUTH TABLE <--> BOOLEAN EQUATION

Sum of Minterms & Creating Columns

BOOLEAN ALGEBRA PROPERTIES

Showing Equation Equivalence

BOOLEAN EQUATIONS TERMINOLOGY

Proper Terminology

CANONICAL SUM OF MINTERMS

Identitiy, Complementation, and Distribution

GATES & TRUTH TABLES

OR		
a	b	$a + b$
0	0	0
0	1	1
1	0	1
1	1	1

AND		
a	b	$a * b$
0	0	0
0	1	0
1	0	0
1	1	1

XOR		
a	b	$(a \oplus b)$
0	0	0
0	1	1
1	0	1
1	1	0

XNOR		
a	b	$(a \oplus b)'$
0	0	1
0	1	0
1	0	0
1	1	1

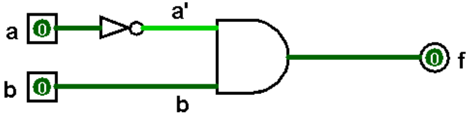
WHAT TO KNOW?

1. Understand how the logic gates manipulate inputs:
   - AND
   - OR
   - NOT
   - NOR
   - NAND
   - XOR
   - XNOR
2. Be able to create truth table representations of gates

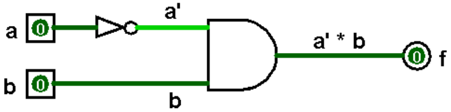
CIRCUIT ---> EQUATION

Identify Intermediate Values

1. Label the relevant intermediate input wires



2. Label the relevant intermediate output wires



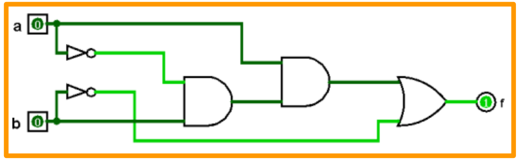
3. If you've reached an output pin, you're done!

f = a' * b

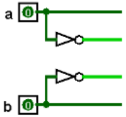
EQUATION ---> CIRCUIT

Prepare your Inputs and Focus on the Operators

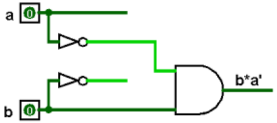
a * (b * a’) + b’



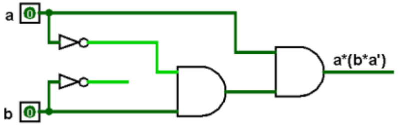
1. Create inputs for each variable in the equation and add branches for negated literals



2. Start with “inner” operators and add the gate with it’s appropriate input



3. Add branching as needed as you create more gates



a	b	c	F	minterm
0	0	0	0	---
0	0	1	1	a’ b’ c
0	1	0	1	a’ b c’
0	1	1	0	---
1	0	0	0	---
1	0	1	1	a b’ c
1	1	0	0	---
1	1	1	1	a b c

TABLE --> EQUATION

1. **Each output column needs its own equation**
2. **Only output values of 1 get a minterm**
3. **When you encounter a 1 in the output column, convert that row’s inputs into a minterm**
4. **Combine all minterms into one equation by “ORing“ them all together**

Useful Mental Note: “The equation F is equal to 1 when the inputs are **(a’ AND b’ AND c) OR (a’ AND b AND c’) OR** etc…”

$$F = a'b'c + a'bc' + ab'c + abc$$

F = ab(c + b’)

a	b	c	b'	a * b	c + b’	F ab(c + b’)
0	0	0	1	0	1	0
0	0	1	1	0	1	0
0	1	0	0	0	0	0
0	1	1	0	0	1	0
1	0	0	1	0	1	0
1	0	1	1	0	1	0
1	1	0	0	1	0	0
1	1	1	0	1	1	1
Step 1			2	3		4

EQUATION --> TABLE

1. Identify how many variable columns you need and fill them in by counting from 0 in binary
2. Identify the negation columns and fill those in
3. Identify the operators that need their own column and fill in that column according to that operator
4. Create your output column and apply the last operator

Useful Note: You may be able to simplify the Boolean equation prior to filling in the table. This can save you a lot of time!

Boolean Algebra Properties

Property	Examples	
Distributive	$a + (b * c) = (a + b) * (a + c)$ $a * (b + c) = (a * b) + (a * c)$	
Idempotence	$a * a = a$	$a + a = a$
Complementation	$a * a' = 0$	$a + a' = 1$
Identity	$a * 1 = a$	$a + 0 = a$
Annihilator	$a + 1 = 1$	$a * 0 = 0$
Commutative	$a * b = b * a$	$a + b = b + a$
Associativity	$a * (b * c) = (a * b) * c$	$a + (b + c) = (a + b) + c$
Double Negation	$(a')' = a$	
Absorption	$a * (a + b) = a$	$a + (a * b) = a$
De Morgan	$a' * b' = (a + b)'$	$a' + b' = (a * b)'$

WHAT TO KNOW?

1. **Be able to recognize a property when it is shown in an equation**
2. **Be able to use these properties to show that two equations are equivalent**

Useful Note: Pay special note to the subtle difference between the Identity and Annihilator properties

Useful Note: Practice makes perfect here. If you haven't taken Discrete Math yet, getting a good grasp of these properties now will be of great benefit!

TERMINOLOGY

1. **Variable** – A "letter" in an equation $F = a' + bc' + c + ab$ (3 Variables)

2. **Literal** – The negated and non-negated form of a variable $F = a' + bc' + c + ab$ (6 Literals)

3. **Product Term** – A combination of literals with an AND operator $F = a' + bc' + c + ab$ (4 Product Terms)

4. **Sum of Products** – A combination of product terms with OR operators $F = a' + bc' + c'(c + ab)$ (NOT Sum of Products Form)

5. **Minterm** – A product term that contains all of the variables $F = a' + abc' + c + ab$ (Only one minterm)

6. **Canonical Sum of Minterms** – When an equation is in sum of products form **and** each term is a minterm $F = a'bc + abc' + bc + abc$ (NOT Canonical Sum of Minterms)

Useful Note: Count **EACH** instance of a literal when finding the number of literals. ($F = a + a$ has 2 literals)

Useful Note: A sole literal **IS** considered to be a product term. ($F = a + ab$ has 2 product terms)

CANONICAL SUM OF MINTERMS

General Process

1. **Convert the equation to Sum of Products Form**

   - Applying the Distributive and De Morgan's property usually accomplishes this

2. **Use the following 3 properties as needed to convert each Product Term into a Minterm**

- Identity	"Multiplying" each Product Term by 1
- Complementation	Replacing the 1 with an appropriate complementation (e.g., (b + b') = 1)
- Distributive	Distribute the original Product Term into the complementation to create new Product Terms

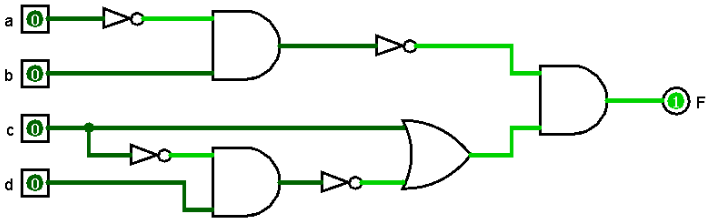
3. **Remove any duplicate minterms at the end.**

Useful Note: If there are **minterms already present in an equation**, simply ignore them until the last step.

Useful Note: An **alternative approach** for converting to Canonical Sum of Minterms is provided in the Additional Content section of Module 3

Practice Problems Conversions

1. Convert the following circuit to a Boolean equation



2. Convert the following Boolean equation to a circuit

G = a + b’ + a’bc

Practice Problems Conversions

3. Convert the following Truth Table to a Boolean equation (J = ?)

a	b	c	J
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

4. Convert the following Boolean Equation to a Truth Table

$$K = a(c + a') + abc$$

Practice Problems Boolean Algebra

5. Are the following Boolean equations equivalent?

   L = ab + b'c' + a(a + b')
   M = a + ab'c' + a'b'c' + ab

6. Is the following Boolean equation in Canonical Sum of Minterms form?

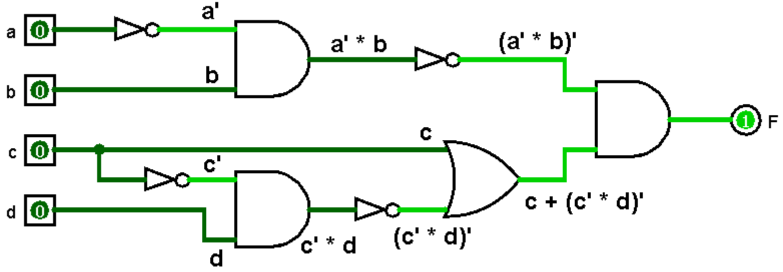
   N = ab'c + a'b(c + a')

7. How many variables and literals are present in the following equation?

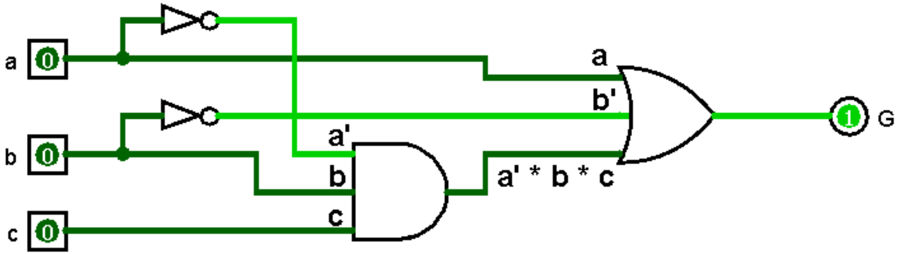
   P = a' + b'c + b(c + a')

Practice Problems Answers

1. F = (a’b)’ * (c + (c’d)’)



2. G = a + b’ + a’bc



Practice Problems Answers

3. J = a’b’c’ + a’b’c + a’bc + abc

a	b	c	J	
0	0	0	1	a’b’c’
0	0	1	1	a’b’c
0	1	0	0	
0	1	1	1	a’bc
1	0	0	0	
1	0	1	0	
1	1	0	0	
1	1	1	1	abc

4. K = a(c + a’) + abc

a	b	c	a'	abc	c + a’	a(c + a’)	K
0	0	0	1	0	1	0	0
0	0	1	1	0	1	0	0
0	1	0	1	0	1	0	0
0	1	1	1	0	1	0	0
1	0	0	0	0	0	0	0
1	0	1	0	0	1	1	1
1	1	0	0	0	0	0	0
1	1	1	0	1	1	1	1

Practice Problems Answers

5. L = ab + b’c’ + a(a + b’)	Absorption [a(a + b’) = a]
ab + b’c’ + a	Identity & Complementation [b’c’ = b’c’(a + a’)]
ab + b’c’(a + a’) + a	Distributive [b’c’(a + a’) = ab’c’ + a’b’c’]
ab + ab’c’ + a’b’c’ + a	Commutative [ab + a = a + ab]
L = a + ab’c’ + a’b’c’ + ab	
M = a + ab’c’ + a’b’c’ + ab	Yes, Equivalent!

6. N = ab’c + a’b(c + a’) is not in Canonical Sum of Minterms for because some terms are not Minterms and it is not in the Sum of Products form.

7. P = a’ + b’c + b(c + a’) has 3 variables and 6 literals. [Remember, count EACH instance of a literal]

Catalina, Jimena. (2020). Salerio Powerpoint template. Retrieved from https://www.slidescarnival.com/salerio-free-presentation-template/1875