Lecture 6
Boolean algebra and circuits

COS10003 Computer Logic and Essentials (Hawthorn)

Semester 1 2021

Boolean algebra Logic gates Simplification again Logic circuits Applications Next

Today

1. Boolean algebra
2. Logic gates
3. Simplification again
4. Logic circuits
5. Applications

How logic applies to circuits

The components of logic circuits

Why circuit design is important

Overview

- Boolean algebra is a mathematical structure that incorporates the properties of logic and sets, and deals with variables that could be 0 or 1.
- It was developed in the 19th century.
- In the 1930s it was recognised as being applicable to switching circuits.

Principles of Boolean algebra

- The element 0 is called the zero element, element 1 is called the unit element and a' is the complement of a. These can be any elements and as such are not restricted to the numbers 0 and 1.
- The results of the operations + and * are called sum and product, respectively.
- Do not mistake these for addition or multiplication as in conventional mathematics. They can be any operations that we choose providing that the axioms are satisfied.

It is quite common to drop the * symbol, i.e., instead of a * b, we will frequently write ab.

Sum and product operations

The results of the binary operations can be summarized in tables. For example, if $B = \{0, 1\}$ and $1' = 0$ and $0' = 1$, the Boolean algebra is:

x	y	x * y
1	1	1
1	0	0
0	1	0
0	0	0

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

x	x′
1	0
0	1

Working with expressions

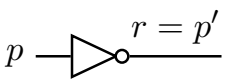
Given the values $a = 1$, $b = 0$ and $c = 1$, what is the value of:

- $a + b$? **1 + 0 = 1**
- $(a + b) * c$?
- $ab + c'$?
- $ab + b'c$?
- $a + c$?

Logic gates and circuits

- In digital circuits, Boolean operations are implemented with gates.
- A gate is an electronic device that produces a result based on two or more input values.
- An integrated circuits contains a collection of gates suited to a particular purpose.

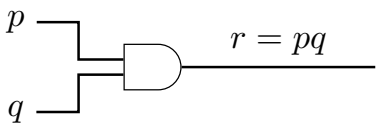
Gates: NOT



input	output
p	r
1	0
0	1

This is the same as complement in Boolean algebra.

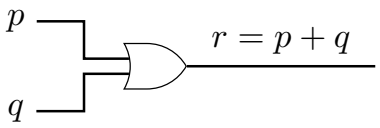
Gates: AND



input		output
p	q	r
1	1	1
1	0	0
0	1	0
0	0	0

This is the same as product in Boolean algebra.

Gates: OR



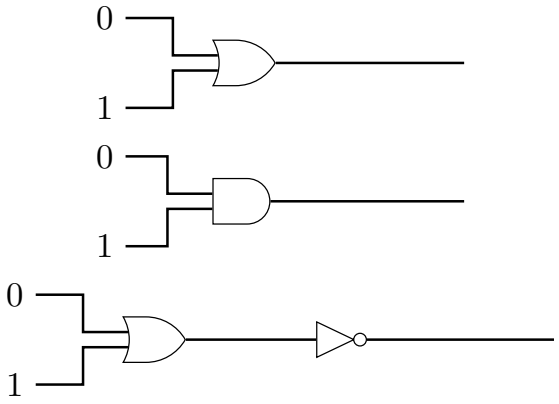
input		output
p	q	r
1	1	1
1	0	1
0	1	1
0	0	0

Side note: notation change

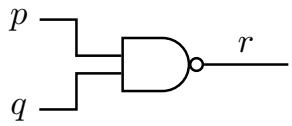
With Boolean algebra, we generally use lowercase letters and ‘ for NOT.
With circuits, we generally use uppercase letters and an overscore for NOT.

Gateways

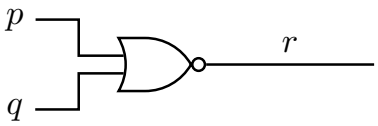
What is the output of the following?



NAND and NOR Gates

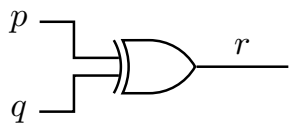


AND followed by NOT

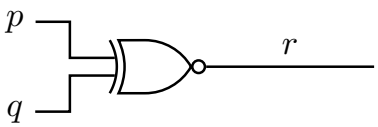


OR followed by NOT

XOR and XNOR Gates



Exclusive OR: either p or q must be 1, not both



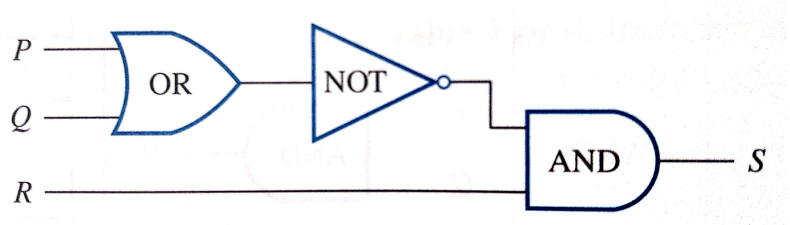
XOR followed by NOT, i.e., both p and q are the same

Logic circuits

- Circuits can be made from these gates.
- Circuits generally have one output, one or more inputs, and no cycles.
- The main gates used are AND, OR, NOT but you might see other gates.

Combining gates into circuits

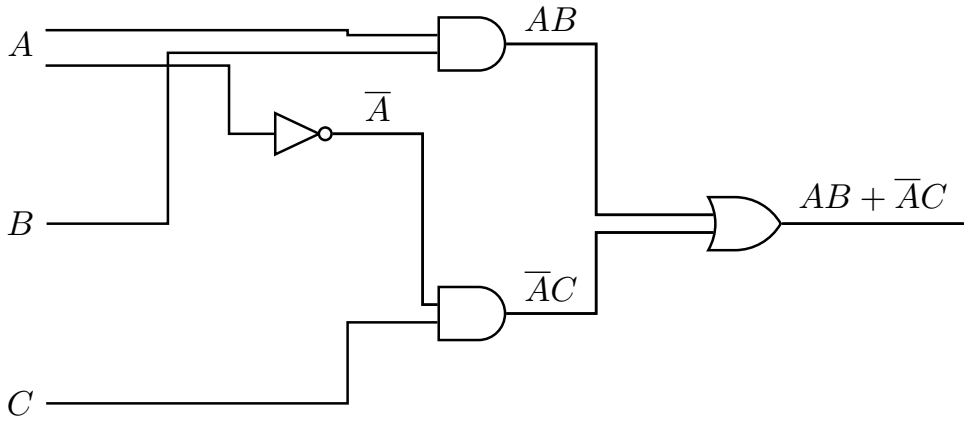
Determine the output of the circuit below given the following input signals: $P = 1, Q = 0, R = 1$.



(This exercise question, solution, and image are taken from (Epp 2011, p.68))

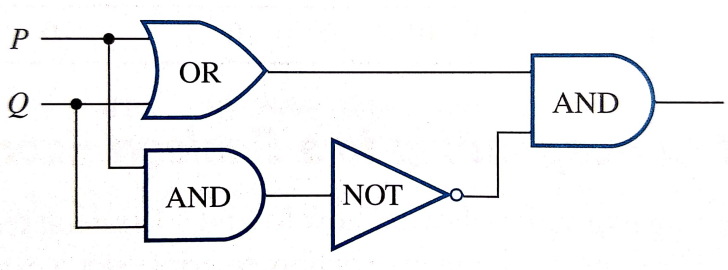
Tracing circuits

To determine the output of any logic circuit, we can determine the output by tracing the feeds. The outcome is a Boolean expression that can be evaluated.



Combining gates into circuits

Find the Boolean expression that corresponds to the circuit below.



(This exercise question, solution, and image are taken from (Epp 2011, p.69))

Boolean Expression of a Circuit

Design a circuit for the Boolean expression $\overline{P}Q + \overline{Q}$.

(This exercise question, solution, and image are taken from (Epp 2011, p.70-71))

Sequences of bits

Let's put a sequence of bits through circuits, e.g., $A = 1001$, $B = 1000$ and $C = 0010$:

- $A * B$? **1000**
- $\overline{A}C$?
- $AB + \overline{A}C$?

Lecture 6
Boolean algebra and circuits

COS10003 Computer Logic and Essentials (Hawthorn)

Semester 1 2021

Axioms of Boolean algebra

In addition to the operations (+, *, ’) and the values (0, 1), we need axioms or rules that our algebra needs to satisfy.
These include concepts that should be familiar to you by now.

Axioms of Boolean algebra

- Associativity: $a * (b * c) = (a * b) * c$, $a + (b + c) = (a + b) + c$
- Commutativity: $a * b = b * a$, $a + b = b + a$
- Identity: $a + 0 = a$, $a * 1 = a$
- Distributivity: $a * (b + c) = (a * b) + (a * c)$, $a + (b * c) = (a + b) * (a + c)$
- Complements: $a + a' = 1$, $a * a' = 0$
- Absorption: $a * (a + b) = a$, $a + (a * b) = a$

Theorems of Boolean algebra

- Idempotent: $a + a = a$, $a * a = a$
- Boundedness: $a + 1 = 1$, $a * 0 = 0$
- Involution: $(a')' = a$, $0' = 1$, $1' = 0$
- De Morgan: $(a + b)' = a' * b'$, $(a * b)' = a' + b'$

Absorption

Given the variables p and q, we should be able to show that $p + pq$ is equal to p.

p	q	$p + pq$
1	1	1 + 1*1 = 1
1	0	1 + 1*0 = 1
0	1	0 + 0*1 = 0
0	0	0 + 0*0 = 0

Simplifying Boolean expressions

Boolean expressions are composed of variables using the Boolean operators +, * and ‘, e.g.,

$$E = (x + y'z)' + (xyz' + x'y + xx'y)'$$

A fundamental product is a single variable or a product of two or more unique variables.

Sum of products

A Boolean expression E is in the sum of products form if E is either a fundamental product or the sum of two or more fundamental products, none of which is included in the other.

$$E_1 = xz' + y'z + xyz'$$
$$E_2 = xz' + x'yz' + xy'z$$

This is not in sum of products form because the xz' occurs both in the first and last terms.

Simplifying

Any expression can be transformed into the sum of products form by applying the following algorithm:

- Apply De Morgan's law and involution to move the complement operation into any parentheses until it applies only to variables.
- Use the distributive law to transform E into a sum of products.
- Use the commutative, idempotent and complement laws to transform each term of E into 0 or a fundamental product.
- Use the absorption law to remove redundancy.

An example

$$((ab)'c)' \quad ((a'+c)(b'+c'))'$$
$$((ab)''+c')(\ (a'+c)' \ +\ (b'+c')'\ )$$
$$(\ (ab)''\ +c')(\ (a')'\ *c'+\ ((b')'\ *\ (c')'\ )$$
$$(ab+c')(ac'+bc)$$
$$(ab+c')ac'\ +\ (ab+c')bc$$
$$a\,b\,a\,c'+\ c'\,a\,c'\ +a\,bb\,c+\ c'\,b\,c$$
$$abc'\ +\ ac'\ +abc+0$$
$$ac'+abc$$

De Morgan's x 2
De Morgan's x 2
Involution x 4
Distributive
Distributive
Idempotent x 4
Absorption

Hints

Like the simplification of logical statements one or more law may need to be invoked more than once to obtain the desired simplification.

Applying simplification to circuits

For circuit design, obviously want the smallest circuit possible.
This means less physical space, less power, and lower cost to manufacture.

Size and depth

The size of a circuit is the number of gates in the circuit.
The depth of a circuit is the number of gates
on the longest path from input to output in the circuit.
For models of computation, seeking the minimal circuit that performs a function.

Find the expression

(Note the inputs B and C should be swapped.)

What is the expression of this circuit?

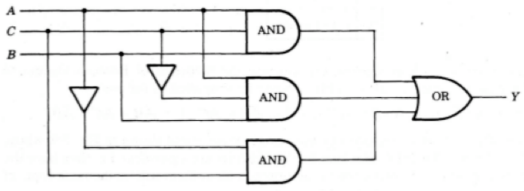


Figure 5.2. An example of an AND-OR circuit composed of two not gates, three AND gates and a single OR gate.

What is the size?

What is the depth?

Simplify the expression

$$
\begin{aligned}
Y &= ABC + A\overline{B}C + \overline{A}B \\
&= AC(B + \overline{B}) + \overline{A}B \\
&= AC * 1 + \overline{A}B \\
&= AC + \overline{A}B
\end{aligned}
$$

Draw new circuit

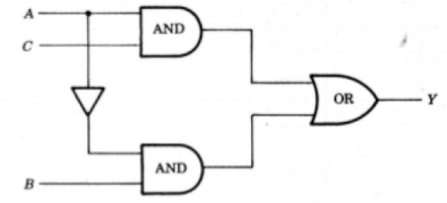


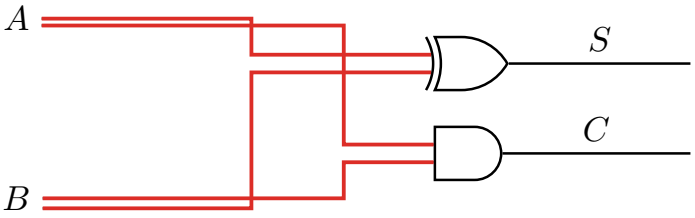
Figure 5.4. A simplified AND-OR circuit obtained by simplifying the Boolean output expression of Figure 5.3. This circuit has the same effect as the more complicated circuit.

What is the size?

What is the depth?

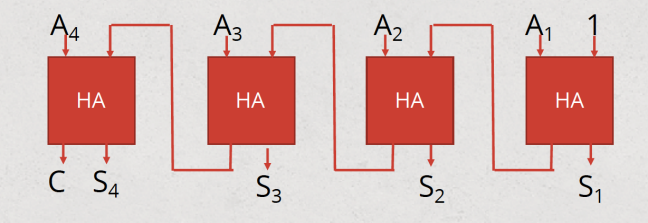
Adding two bits?

This is known as a half adder, and can be composed of an XOR and an AND gate.



Incrementing an integer?

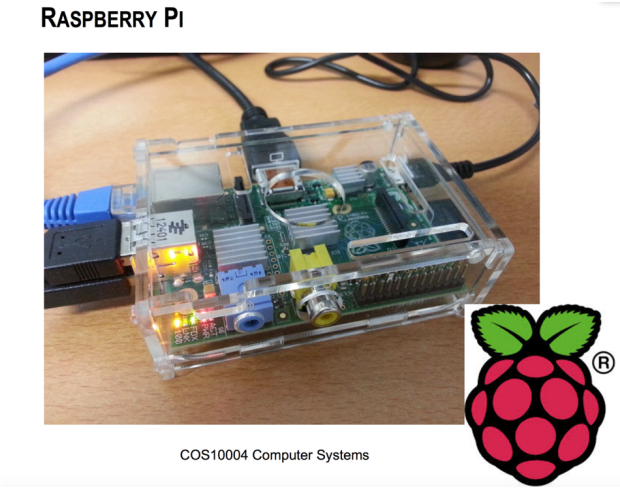
This can be formed of several half adders, where the carry feeds into the next half adder. For example with four bits:



$$A_4A_3A_2A_1 + 1 = CS_4S_3S_2S_1$$

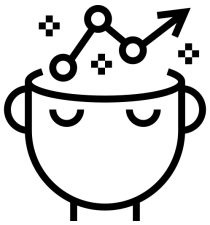
Would you like to know more?

COS10004 Computer Systems picks up where this leaves off.

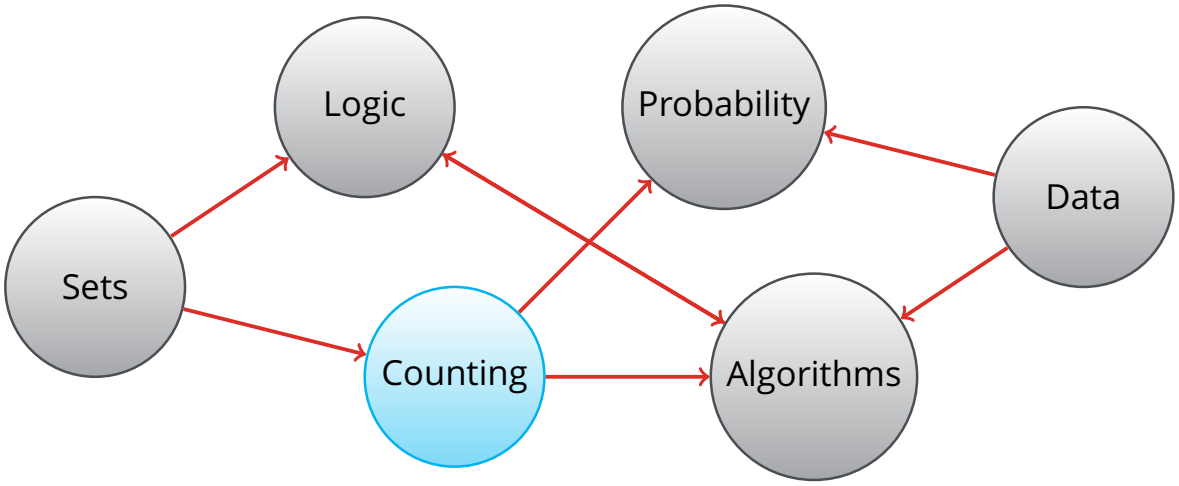


Reflecting

- What are the main components of logic circuits?
- Which principles of logic are also useful for circuits?
- Why are we interested in creating small circuits?



Where to next?



In which we learn how to count.

Lecture 6
Boolean algebra and circuits

COS10003 Computer Logic and Essentials (Hawthorn)

Semester 1 2021

Boolean algebra Logic gates Simplification again Logic circuits Applications Next

Questions I still have

Topics I need to review