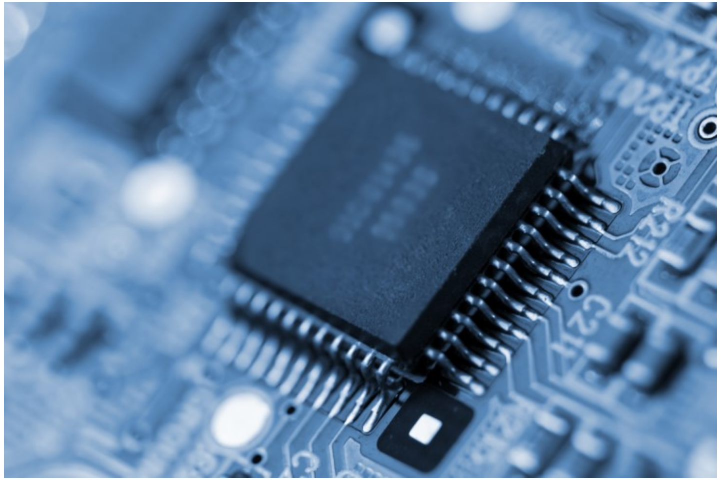
UNIVERSITY OF LEEDS

Computer Processors

Logic minimisation

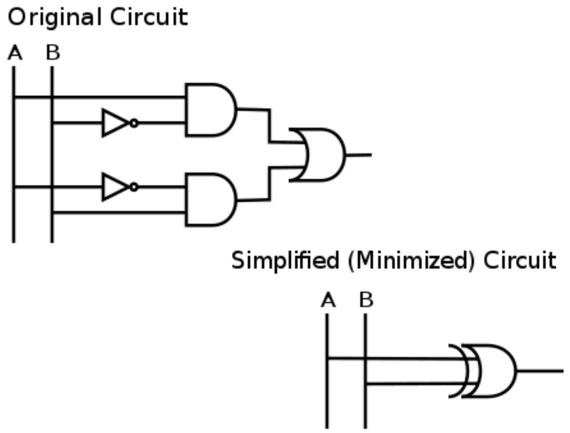


Lecture 6

What is minimisation?

UNIVERSITY OF LEEDS

- Minimisation is the process of optimising some criteria
- The criteria might be
    - Number of transistors
    - Number of different types of gates
    - Depth of a logic circuit
    - Fan-in/Fan-out



Why?

UNIVERSITY OF LEEDS

- There is a limit to transistor density on silicon wafer
- There is a cost implication for each transistor and gate
- There is propagation time through logic gates
    - The greater the depth the longer it takes
- Conceptual simplicity for bug finding
- Simple designs are less likely to go wrong

Algebraic manipulation

UNIVERSITY OF LEEDS

- From Fundamental Mathematical Concepts (COMP1421) you are aware of a set of logical equivalences
- These logical equivalences can be applied to reduce the number of variables and terms in a given expression
- There are no fixed rules to the application of logical equivalences, it comes down to practice

Algebraic manipulation example

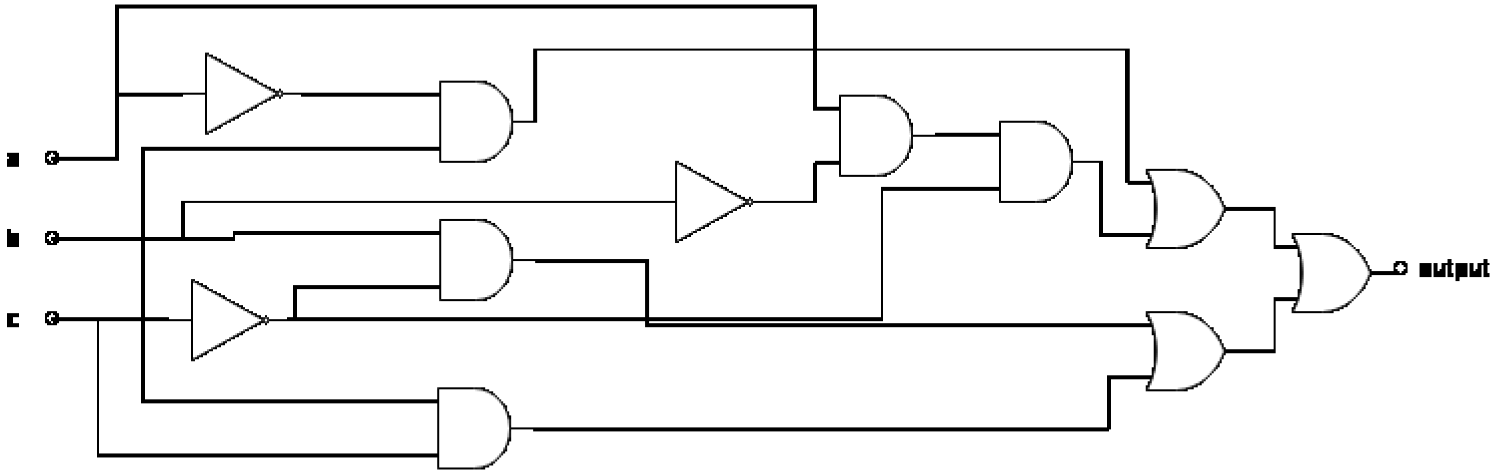
UNIVERSITY OF LEEDS

$$(\neg a \wedge b) \vee (b \wedge \neg c) \vee (b \wedge c) \vee (a \wedge \neg b \wedge \neg c)$$

a	b	c	output
0	0	0	0
0	1	0	1
1	0	0	1
1	1	0	1
0	0	1	0
0	1	1	1
1	0	1	0
1	1	1	1

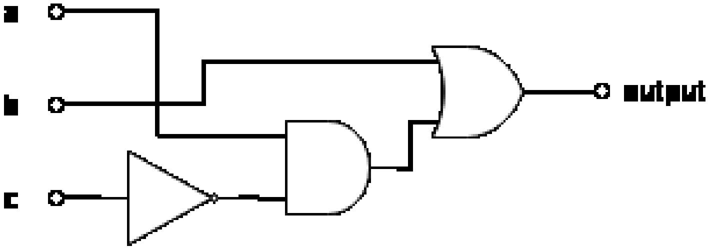
Algebraic manipulation example

$$(\neg a \wedge b) \vee (b \wedge \neg c) \vee (b \wedge c) \vee (a \wedge \neg b \wedge \neg c)$$



Algebraic manipulation example

$$b \vee (\neg c \wedge a)$$



- From 11 gates to 3
- From a spaghetti of wires to a simple design

Karnaugh maps

UNIVERSITY OF LEEDS

- Karnaugh maps are an alternative representation of a truth table with some interesting properties
- Developed in 1953 by Maurice Karnaugh
- Ideal for identifying redundant variables in logic expressions.
- Ideal for up to 6 variables

a	b	output
0	0	0
0	1	1
1	0	1
1	1	1

Truth table

b \ a	0	1
0	0	1
1	1	1

Karnaugh Map

Karnaugh maps

- Karnaugh maps have 0 and 1 entries
- Each entry corresponds to a truth table entry

a	b	output
0	0	0
0	1	1
1	0	1
1	1	1

Truth table

b \ a	0	1
0	0	1
1	1	1

Karnaugh Map

Note: Each of the row and column headings are only one bit different to their adjacent heading

How to use Karnaugh maps

UNIVERSITY OF LEEDS

- Karnaugh maps make it easy to group terms together
- Terms are grouped together on the condition that two entries are both 1's and are side by side (vertically or horizontally)

a	b	output
0	0	0
0	1	1
1	0	1
1	1	1

Truth table

b \ a	0	1
0	0	1
1	1	1

Karnaugh Map

[redacted]

How to use Karnaugh maps

UNIVERSITY OF LEEDS

Let’s examine the entries in the box

The terms are

$$\neg a \land b$$

and

$$a \land b$$

These terms can be combined into

$$b$$

b \ a	0	1
0	0	1
1	1	1

Karnaugh Map

a	b	output
0	0	0
0	1	1
1	0	1
1	1	1

Truth table

[redacted]

How to use Karnaugh maps

Let's examine the entries in the box

The terms are

$$a \wedge \neg b$$

and

$$a \wedge b$$

These terms can be combined into

$$a$$

a	b	output
0	0	0
0	1	1
1	0	1
1	1	1

Truth table

b \ a	0	1
0	0	1
1	$\boxed{1}$	$\boxed{1}$

Karnaugh Map

[redacted]

How to use Karnaugh maps

The two groupings result in the expressions

a b

As with the normal process of producing a sum of products we **Or t**he expressions together.

Resulting in:

$$a \vee b$$

a	b	output
0	0	0
0	1	1
1	0	1
1	1	1

Truth table

b \ a	0	1
0	0	1
1	1	1

Karnaugh Map

[redacted]

Karnaugh maps (3 variables)

- With 3 variables things get slightly awkward
- The karnaugh map will contain 8 entries

		bc			
		00	01	11	10
a	0	0	0	1	0
	1	0	1	1	1

a	b	c	output
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

[redacted]

Karnaugh maps (3 variables)

UNIVERSITY OF LEEDS

Let's consider the entries in the box, they correspond to:

$$a \land \neg b \land c$$

and

$$a \land b \land c$$

This can be reduced to

$$a \land c$$

		bc			
		00	01	11	10
a	0	0	0	1	0
	1	0	1	1	1

<u>Notice</u> the order of the column in the karnaugh map - NOT in binary order.

Ordered in such a way that adjacent entries differ by exactly 1 bit.

Karnaugh maps (3 variables)

		bc			
		00	01	11	10
a	0	0	0	1	0
	1	0	1	1	1

Let's consider the entries in the box, they correspond to:

$$a \wedge b \wedge c$$

and

$$a \wedge b \wedge \neg c$$

This can be reduced to

$$a \wedge b$$

Karnaugh maps (3 variables)

Let’s consider the entries in the box, they correspond to:

$$a \wedge b \wedge c$$

and

$$\neg a \wedge b \wedge c$$

This can be reduced to

$$b \wedge c$$

		bc			
		00	01	11	10
a	0	0	0	1	0
	1	0	1	1	1

[redacted]

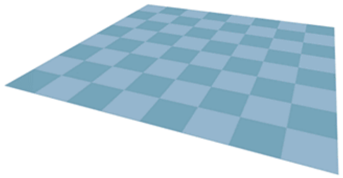
$$(a \wedge b) \vee (a \wedge c) \vee (b \wedge c)$$

Karnaugh maps

UNIVERSITY OF LEEDS

- Karnaugh map groups in general can wrap round the end of a map

		cd			
		00	01	11	10
ab	00	1	0	0	1
	01	0	0	0	0
	11	0	0	0	0
	10	1	0	0	1



- Groups should be of size 2^n for some n
    - For a group of size n, n variables are excluded from the term

By Kieff - Own work, Public Domain, https://commons.wikimedia.org/w/index.php?curid=19541412

Tabular Method

UNIVERSITY OF LEEDS

- Karnaugh maps are a great tool for minimising small logic expressions
- Difficult to implement on a computer due to the reliance on human pattern matching
- Quine-McCluskey algorithm is a modification of Karnaugh maps which is implementable on a computer

Note: The problem of minimising boolean functions is NP-Hard, under some reasonable assumptions it is believed that there is no algorithm that runs in polynomial time that can compute it.

Summary

UNIVERSITY OF LEEDS

- Introduced the concept of logic minimisation
- Demonstrated logic equivalences as a tool for minimisation
- Introduced Karnaugh maps
- Demonstrated the use of Karnaugh maps
- Explained that logic minimisation is a hard problem

Interested in Karnaugh maps? Digital Logic Design Section 2.4.2