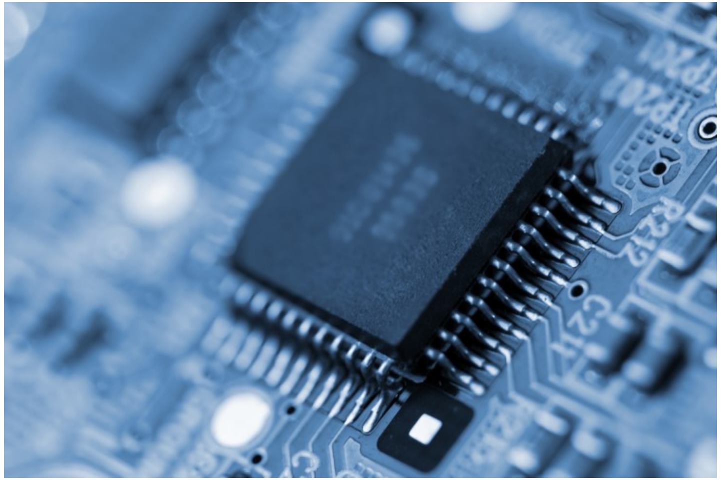
UNIVERSITY OF LEEDS

Computer Processors

Common logic configurations



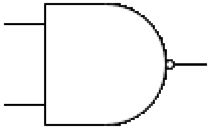
Lecture 5

Building Gates from Nands

UNIVERSITY OF LEEDS

- We have seen that all Boolean expressions can be expressed in terms of **AND**, **OR** and **NOT**
- All logic gates can be expressed in terms of a **NAND**
- **NAND** is sometimes referred to as a universal logic gate

A	B	Output
0	0	1
0	1	1
1	0	1
1	1	0

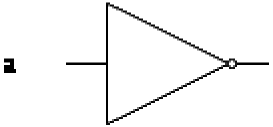


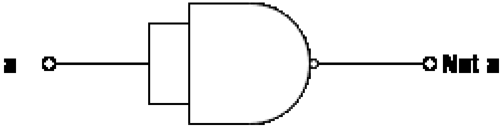
Not

- When **a** is low the output is high
- When **a** is high the output is low

a	**Not** a
0	1
1	0

A	B	Output
0	0	1
0	1	1
1	0	1
1	1	0

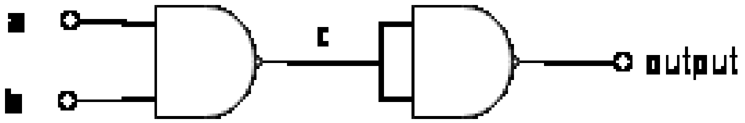




And

- **Nand** is a negated **And**
- Negate a **Nand** (double negation) the result is an **And**

a	b	c	Output
0	0	1	0
0	1	1	0
1	0	1	0
1	1	0	1



Double negation law, remember it from last semester?

Or

UNIVERSITY OF LEEDS

- **Or** made out of **Nand**’s can be devised

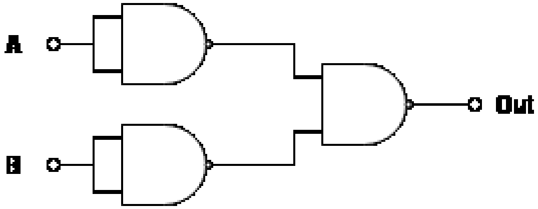
  using the Boolean laws

$$x \vee y$$

$$\neg\neg x \vee \neg\neg y$$

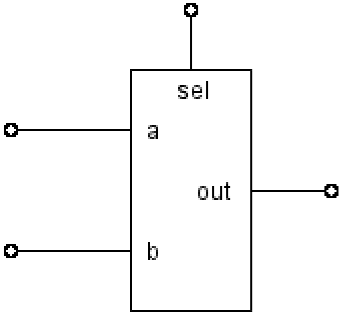
$$\neg(\neg x \wedge \neg y)$$

a	b	Output
0	0	0
0	1	1
1	0	1
1	1	1



Multiplexor

- A multiplexor, sometimes called a MUX, is a 3 input gate
- One input is a selection bit
- One of the other two inputs, **a** and **b**, are outputted depending on the value of selection bit

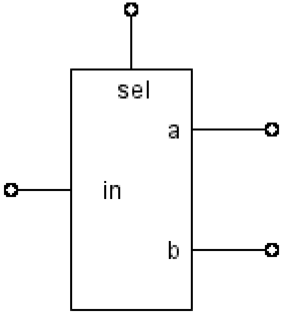


a	b	sel	out
0	0	0	0
0	1	0	0
1	0	0	1
1	1	0	1
0	0	1	0
0	1	1	1
1	0	1	0
1	1	1	1

Demultiplexor

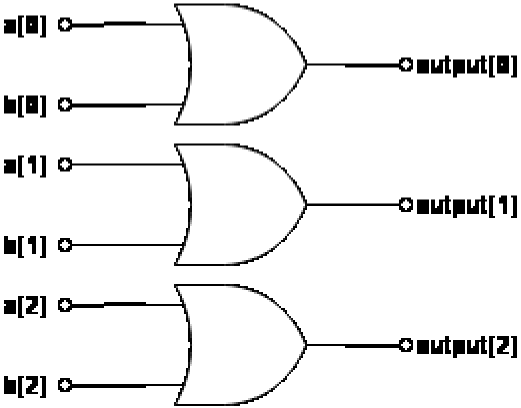
- Demultiplexor performs the opposite function to a multiplexor
- Two inputs **in** and a **selection bit**
- Two outputs **a** and **b**
- Outputs **in** on either **a** or **b** depending on the value of the selection bit

Selection bit	a	b
0	in	0
1	0	in



Multi-Bit gates

- Computer hardware use buses of a specified width
- n-bit variety of most of the logic gates
  - n-bit **Or**
  - n-bit **And**
  - n-bit **Not**
  - n-bit **Mux**



3-bit **Or**

Multi-way gates

- Many logic gates which have 2 inputs generalise to having n bits
  - n-way **Or**
  - n-way **And**
  - n-way **Xor**
  - n-way **Mux**

a	b	c	out
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

3-way **And**

Multi-way gates

- Many logic gates which have 2 inputs

  generalise to having n bits

  - n-way **Or**
  - n-way **And**
  - n-way **Xor**
  - n-way **Mux**

a	b	c	out
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

3-way **Or**

Multi-way gates

- Many logic gates which have 2 inputs

  generalise to having n bits

  - n-way **Or**
  - n-way **And**
  - n-way **Xor**
  - n-way **Mux**

a	b	c	out
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	0

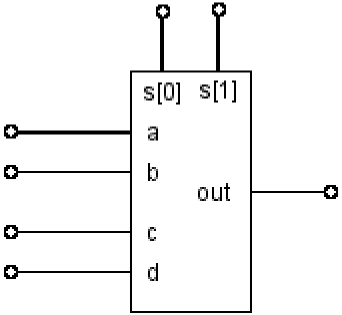
3-way **Xor**

Multi-way/Multi-bit Multiplexor

- What might a 4-way 16-bit multiplexor look like?
- How many selection bits will it have?
- How wide is the input bus it 'switches'?
- How wide is the output bus it 'switches'?

<u>Note</u>:

Multi-way multiplexor can be constructed out of multiplexors



s[0]	s[1]	out
0	0	a
0	1	b
1	0	c
1	1	d

Look up tables

- An alternative to combinatorial logic
- Essentially a large memory chip
- Inputs are used as address lines
- Output is the value stored in a specific memory location
- Alternatively can be implemented using multiplexors, with hardwire values

In 1957 with the IBM 1401 and 1620 computer parts of the processing unit were replaced by lookup tables implemented in ROM to perform calculations.

By ArnoldReinhold - Own work, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=18288954

Summary

- Introduced construction of elementary logic gates from **Nand**’s
- Introduced multiplexors
- Introduced demultiplexors
- Introduced multi-way/multi-bit gates
- Introduced Lookup tables

Interested in logic design? Digital Logic Design