COMP110: Principles of Computing

# 4: Logic and memory

# Worksheet 4

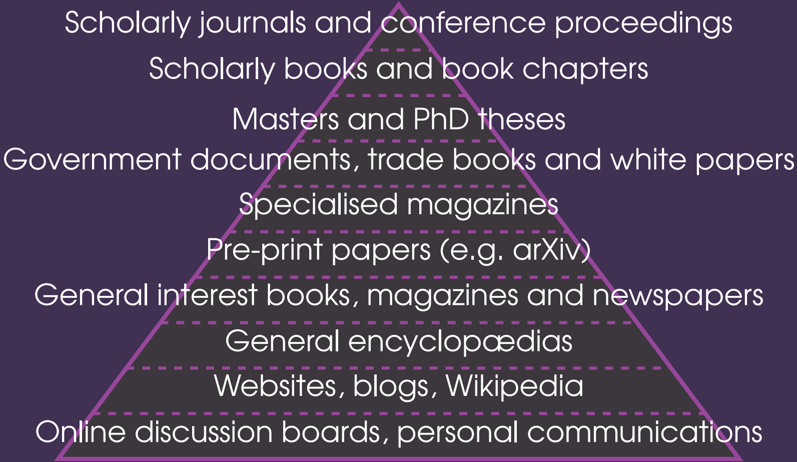
Due **next Friday!**

# Scholarly literature

# Scholarly work

- What is a “scholarly” work?
- How do we know if something is scholarly?

# Pyramid of sources



Scholarly journals and conference proceedings

Scholarly books and book chapters

Masters and PhD theses

Government documents, trade books and white papers

Specialised magazines

Pre-print papers (e.g. arXiv)

General interest books, magazines and newspapers

General encyclopædias

Websites, blogs, Wikipedia

Online discussion boards, personal communications

# Appropriateness of sources

It is important to question the **appropriateness** of sources you use in academic work

- **Validity**: Are claims based upon a correct interpretation of the evidence?
- **Rigor**: Was the method of collecting evidence appropriate to ensure comprehensive coverage while also avoiding bias?

# Appropriateness of sources

- **Reliability**: has the claim been replicated, or at least reviewed, by other academics?
- **Authoritativeness**: do we know who the author is? Does the author have enough experience in the field to present a fair and balanced argument?
- **Venue**: Is the publisher reputable and free of undue editorial influences?

# Appropriateness of sources

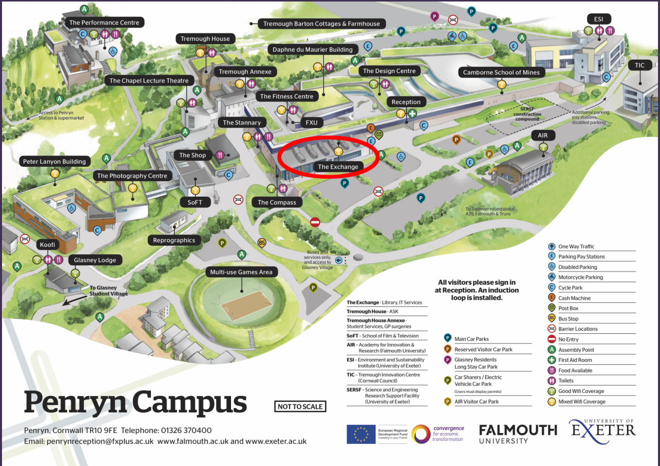
There are of course exceptions where sources are presented as **artefacts** and/or **archives**:

- Citing a newspaper as evidence for a claim based on the reception of a new technology
- Citing a manufacturer's technical manual when describing a technical feature of a platform
- Citing a Reddit post by a well-known industry figure as evidence for expert opinion

The **way** in which sources are **used** is therefore important

# Library resources

# The library


The Performance Centre
Tremough Barton Cottages & Farmhouse
ESI
Tremough House
Daphne du Maurier Building
Tremough Annexe
The Design Centre
Camborne School of Mines
TIC
The Chapel Lecture Theatre
The Fitness Centre
Reception
Access to Penryn Station & supermarket
SERSF construction compound
Additional parking, pay stations, disabled parking
The Stannary
FXU
AIR
The Shop
Peter Lanyon Building
The Exchange
The Photography Centre
SoFT
The Compass
To Treliever roundabout A39, Falmouth & Truro
Reprographics
Koofi
Glasney Lodge
Buses and services only, and access to Glasney Village
Multi-use Games Area
To Glasney Student Village
NOT TO SCALE

**All visitors please sign in at Reception. An induction loop is installed.**

**The Exchange** - Library, IT Services

**Tremough House** - ASK

**Tremough House Annexe** - Student Services, GP surgeries

**SoFT** – School of Film & Television

**AIR** – Academy for Innovation & Research (Falmouth University)

**ESI** – Environment and Sustainability Institute (University of Exeter)

**TIC** - Tremough Innovation Centre (Cornwall Council)

**SERSF** - Science and Engineering Research Support Facility (University of Exeter)

- P Main Car Parks
- P Reserved Visitor Car Park
- P Glasney Residents Long Stay Car Park
- P Car Sharers / Electric Vehicle Car Park (Users must display permits)
- P AIR Visitor Car Park

- ↑ One Way Traffic
- £ Parking Pay Stations
- ♿ Disabled Parking
- Motorcycle Parking
- C Cycle Park
- £ Cash Machine
- ✉ Post Box
- BS Bus Stop
- Barrier Locations
- ⛔ No Entry
- A Assembly Point
- ✚ First Aid Room
- Food Available
- Toilets
- Good Wifi Coverage
- Mixed Wifi Coverage

## Penryn Campus

Penryn, Cornwall TR10 9FE Telephone: 01326 370400
Email: penrynreception@fxplus.ac.uk www.falmouth.ac.uk and www.exeter.ac.uk

European Regional Development Fund Investing in your Future
convergence for economic transformation
FALMOUTH UNIVERSITY
UNIVERSITY OF EXETER

# Library catalogue

http://library.fxplus.ac.uk/

# Web proxy

- Some online resources are only available through the campus network
- If not physically on campus, you can access them via **VPN**
- `https://webvpn.falmouth.ac.uk/`
- Some resources can also be accessed by **web proxy** through the library website
- `https://library.fxplus.ac.uk/subject-guides/games/specialist-databases`

# ACM Digital Library

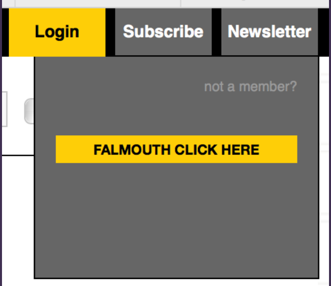
```
http://dl.acm.org.ezproxy.falmouth.ac.uk/
```

# IEEE Xplore

http://ieeexplore.ieee.org.ezproxy.falmouth.ac.uk/

# GDC Vault

http://www.gdcvault.com.ezproxy.falmouth.ac.uk/



There are only a limited number of login slots available — **remember to log out when you have finished!**

# How to find papers to read?

- Specialist databases: ACM Digital Library, IEEE Xplore
- Google Scholar
  - Keyword searches
  - Other work by the same author
  - Work which has cited papers you have read
- Wikipedia
  - Not a reliable source itself!
  - However most articles have good bibiliographies
- Bibliographies of papers you have read

# Finding papers without paying

- Many papers are **paywalled**
- Little known fact: **none** of the money from paywalls goes to the authors of the paper!
- The university **subscribes** to ACM and IEEE to give free access to staff and students
- Many journals offer free **open access**
- Many authors put papers on their **personal websites**
- Many universities (all UK universities) have **open access repositories**
    - Falmouth: `http://repository.falmouth.ac.uk`
- Sites like **sci-hub** have sprung up, providing illegal downloads of papers

# Referencing

# IEEE referencing style

https://ieeeauthorcenter.ieee.org/wp-content/uploads/IEEE-Reference-Guide.pdf

# BibTeX entry types

https://en.wikibooks.org/wiki/LaTeX/Bibliography_Management#BibTeX

# Writing BibTeX entries

- Some websites provide pre-written BibTeX entries for papers
- Beware of copying and pasting these as they are often incomplete, incorrectly formatted or just wrong!
- You must **always** check your bibliography in the compiled PDF and fix any errors
- You **will** lose marks on your written assignments otherwise!

# Logic gates

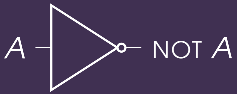
# Boolean logic

- Works with two values: TRUE and FALSE
- Foundation of the **digital computer**: represented in circuits as **on** and **off**
- Representing as 1 and 0 leads to **binary notation**
- One boolean value = one **bit** of information
- Programmers use boolean logic for conditions in `if` and `while` statements

# Not

NOT *A* is TRUE
if and only if
*A* is FALSE

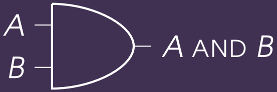
| *A* | NOT *A* |
|---|---|
| FALSE | TRUE |
| TRUE | FALSE |



# And

*A* AND *B* is TRUE
if and only if
**both** *A* **and** *B* are TRUE

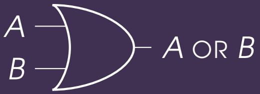
| *A* | *B* | *A* AND *B* |
|---|---|---|
| FALSE | FALSE | FALSE |
| FALSE | TRUE | FALSE |
| TRUE | FALSE | FALSE |
| TRUE | TRUE | TRUE |



# Or

*A* OR *B* is TRUE
if and only if
**either** *A* **or** *B*, **or both,** are TRUE

| *A* | *B* | *A* AND *B* |
|---|---|---|
| FALSE | FALSE | FALSE |
| FALSE | TRUE | TRUE |
| TRUE | FALSE | TRUE |
| TRUE | TRUE | TRUE |



# Socrative `FALCOMPED`

What is the value of

$$A \text{ AND } (B \text{ OR } C)$$

when

$$A = \text{TRUE}$$
$$B = \text{FALSE}$$
$$C = \text{TRUE}$$

?

# Socrative `FALCOMPED`

What is the value of

$$(\textsc{not}\ A)\ \textsc{and}\ (B\ \textsc{or}\ C)$$

when

$$\begin{aligned} A &= \textsc{True} \\ B &= \textsc{False} \\ C &= \textsc{True} \end{aligned}$$

?

# Socrative `FALCOMPED`

For what values of $A$, $B$, $C$, $D$ is

$$A \text{ AND NOT } B \text{ AND NOT } (C \text{ OR } D) = \text{TRUE}$$

?

# Socrative `FALCOMPED`

What is the value of

$$A \text{ OR NOT } A$$

?

# Socrative `FALCOMPED`

What is the value of

*A* AND NOT *A*

?

# Socrative `FALCOMPED`

What is the value of

$$A \text{ OR } A$$

?

# Socrative `FALCOMPED`

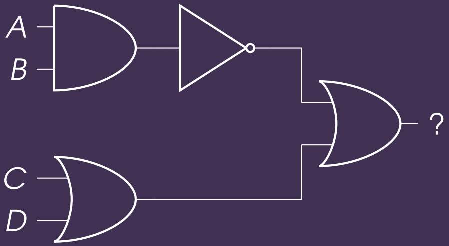
What is the value of

$$A \text{ AND } A$$

?

# Socrative `FALCOMPED`

What expression is equivalent to this circuit?



# Writing logical operations

| Operation | Python | C family | Mathematics |
|---|---|---|---|
| NOT $A$ | `not a` | `!a` | $\neg A$ or $\overline{A}$ |
| $A$ AND $B$ | `a and b` | `a && b` | $A \wedge B$ |
| $A$ OR $B$ | `a or b` | `a \|\| b` | $A \vee B$ |

Other operators can be expressed by combining these

# De Morgan's Laws

$$\text{NOT } (A \text{ OR } B) = (\text{NOT } A) \text{ AND } (\text{NOT } B)$$

$$\text{NOT } (A \text{ AND } B) = (\text{NOT } A) \text{ OR } (\text{NOT } B)$$

Proof: Worksheet 4, questions 3a and 3b

# Truth tables

# Enumeration

- Since booleans have only two possible values, we can often **enumerate** all possible values of a set of boolean variables
- For *n* variables there are $2^n$ possible combinations
- Essentially, all the *n*-bit binary numbers
- A **truth table** enumerates all the possible values of a boolean expression
- Can be used to prove that two expressions are equivalent

# Truth table example

(*A* or not *B*) and *C*

| *A* | *B* | *C* | not *B* | *A* or not *B* | (*A* or not *B*) and *C* |
|---|---|---|---|---|---|
| False | False | False | True | True | False |
| False | False | True | True | True | True |
| False | True | False | False | False | False |
| False | True | True | False | False | False |
| True | False | False | True | True | False |
| True | False | True | True | True | True |
| True | True | False | False | True | False |
| True | True | True | False | True | True |

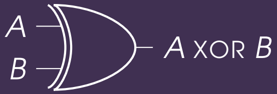
FALMOUTH UNIVERSITY

# Other logic gates

# Exclusive Or

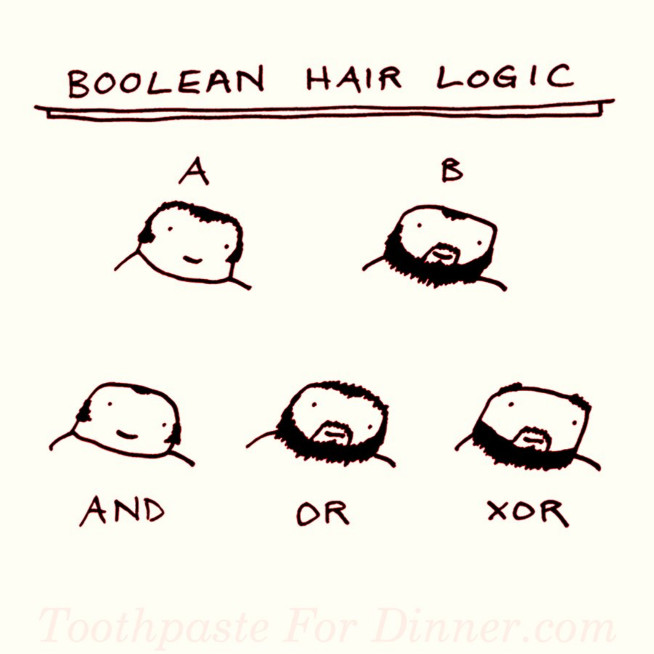
*A* XOR *B* is TRUE
if and only if
**either** *A* **or** *B*, **but not both,** are TRUE

| *A* | *B* | *A* AND *B* |
|---|---|---|
| FALSE | FALSE | FALSE |
| FALSE | TRUE | TRUE |
| TRUE | FALSE | TRUE |
| TRUE | TRUE | FALSE |



# Socrative `FALCOMPED`

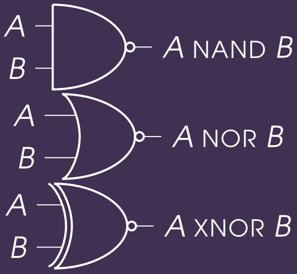
How can *A* XOR *B* be written using the operations AND , OR , NOT ?


BOOLEAN HAIR LOGIC
A
B
AND
OR
XOR
Toothpaste For Dinner.com

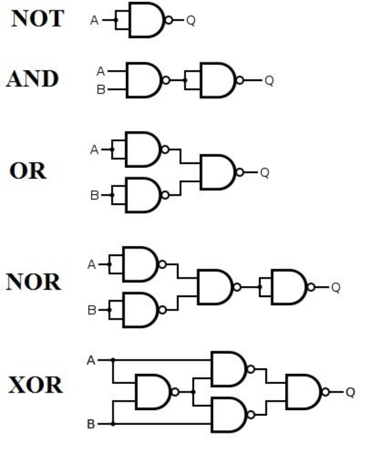
# Negative gates

NAND , NOR , XNOR
are the **negations** of
AND , OR , XOR

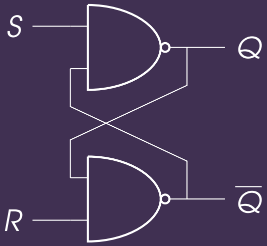
$$A \text{ NAND } B = \text{NOT } (A \text{ AND } B)$$
$$A \text{ NOR } B = \text{NOT } (A \text{ OR } B)$$
$$A \text{ XNOR } B = \text{NOT } (A \text{ XOR } B)$$



# Any logic gate can be constructed from NAND gates



# What does this circuit do?



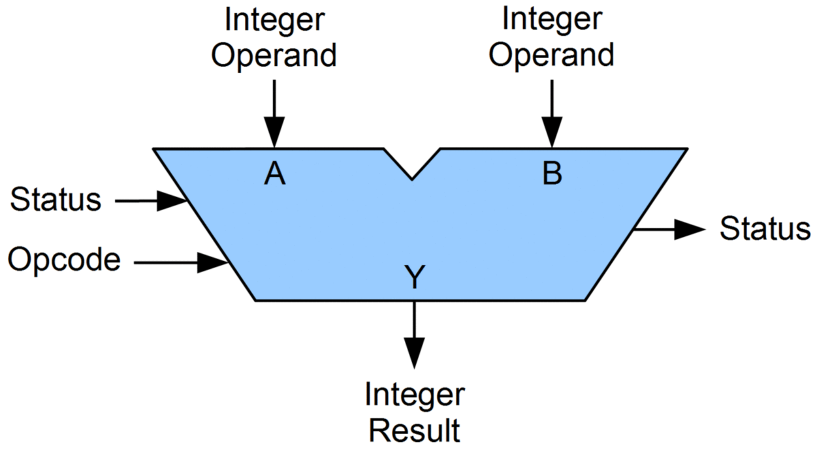
- This is called a **NAND latch**
- It “remembers” a single boolean value
- Put a few billion of these together (along with some control circuitry) and you’ve got **memory**!

# NAND gates

- All arithmetic and logic operations, as well as memory, can be built from NAND gates
- So an entire computer can be built just from NAND gates!
- Play the game: `http://nandgame.com`
- NAND gate circuits are **Turing complete**
- The same is true of NOR gates

# Arithmetic Logic Unit

# Arithmetic Logic Unit



# Arithmetic Logic Unit

- Important part of the CPU
- Inputs:
  - **Operand** words *A*, *B*
  - **Opcode**
  - **Status** bits
- Outputs:
  - **Result** word *Y*
  - **Status** bits
- Opcode specifies how *Y* is calculated based on *A* and *B*

# ALU operations

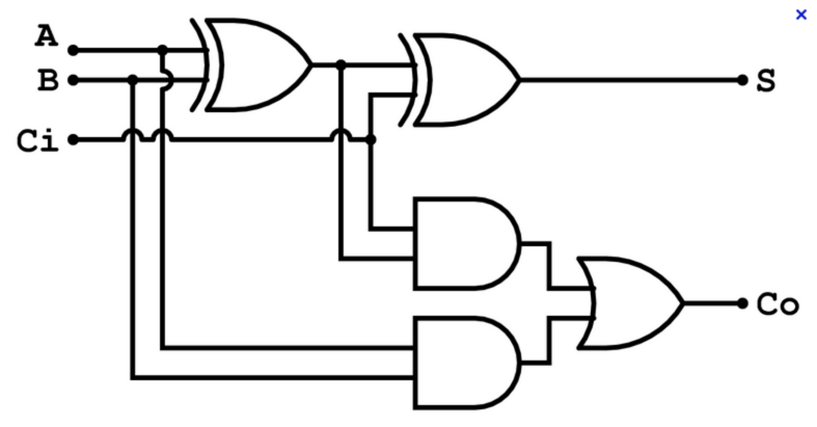
Typically include:

- Add with carry
- Subtract with borrow
- Negate (2's complement)
- Increment, decrement
- Bitwise AND, OR, NOT, . . .
- Bit shifts

# Adding 3 bits

| $A$ | $B$ | $C$ | $A+B+C$ |
|---|---|---|---|
| 0 | 0 | 0 | 00 |
| 0 | 0 | 1 | 01 |
| 0 | 1 | 0 | 01 |
| 0 | 1 | 1 | 10 |
| 1 | 0 | 0 | 01 |
| 1 | 0 | 1 | 10 |
| 1 | 1 | 0 | 10 |
| 1 | 1 | 1 | 11 |

# 1-bit adder

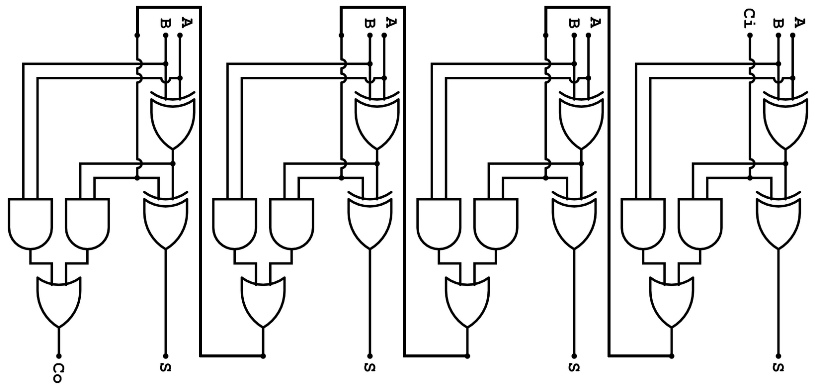


# How does the 1-bit adder work?

Exercise:

- Write down the boolean expressions for *S* and *Co*
- Draw a truth table for these
- Compare the truth table to the addition table on a previous slide

# *n*-bit adder


A
B
A
B
A
B
Ci
B
A
Co
S
S
S
S