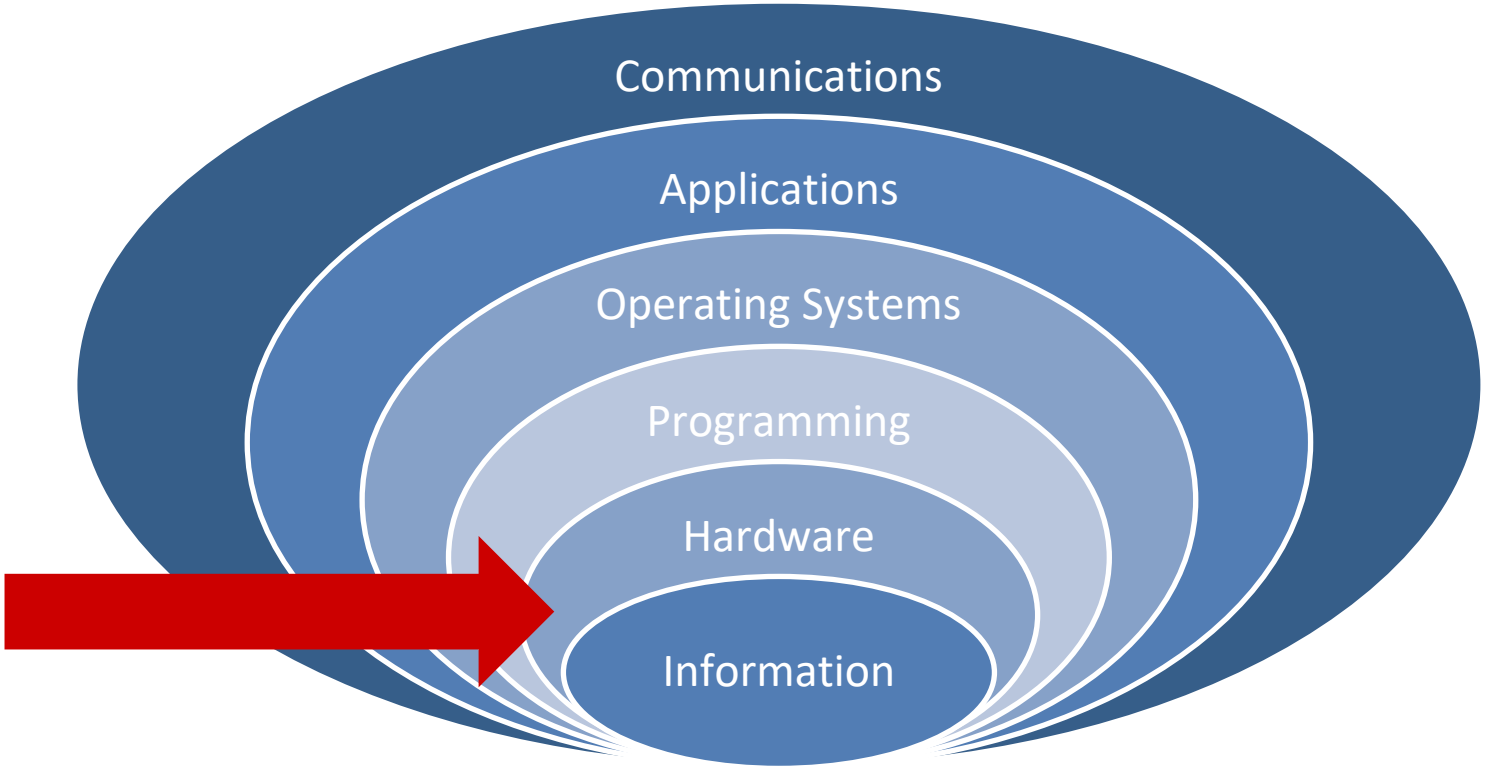
Computer Science Fundamentals

Lecture topic: Gates and Circuits

Lecturer: Olga Yugay

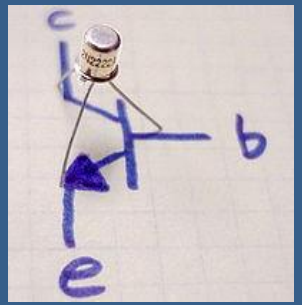
Today's focus

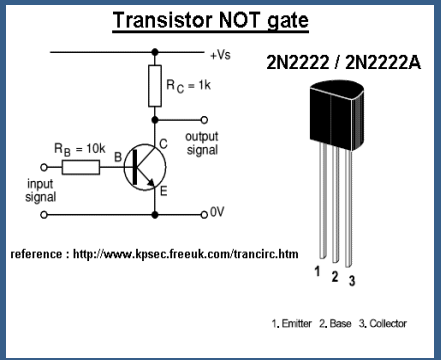


Today's focus

Hardware

Logic gates





Agenda

- Explore how computers use electric signals to represent and manipulate binary values
  - basic gates
  - gates combined into circuits
  - behavior gate using Boolean expressions, truth tables, and logic diagrams
  - compare and contrast a half adder and a full adder
- Python

Why study about circuits? Inside the microchip

Khan Academy and Code.org | Circuits & Logic [1]

[1] Khan Academy and Code.org, 2018, Circuits & Logic. [online] Available at: < https://youtu.be/Sc3lh3D4rCw >

Some background

Background

- Logic gates are the **building blocks**
- Anything from small toys to supercomputers will have digital gates in them (phones, washing machines, lifts and etc.)
- Whenever a decision needs to be taken based on some inputs, a logic gate can be designed and used

[2] Dale, N., & Lewis, J. (2020). Computer Science Illuminated (7th ed.). Jones & Bartlett Learning, Ch 4

Background

- Major computer components rely on logic gates
  - Arithmetic and Logic Unit
  - Primary memory a.k.a RAM (Random Access Memory)
  - Even secondary memory is now switching to using integrated circuits -> SSD (Solid State drives)[3]

[3] Dale, N., & Lewis, J. (2020). Computer Science Illuminated (7th ed.). Jones & Bartlett Learning, Ch 4

Background

- What are Computers?
  - **electronic** devices;
  - the most fundamental **hardware elements** of a computer **control the flow of electricity**.

Electronic signal, 0s and 1s

- Any given electronic signal has a level of voltage.

voltage level 0 to 2 volts	voltage level 2 to 5 volts
“low”	“high”
binary 0	binary 1

- Signals in a computer are constrained to be within one range or the other.[4]

[4] Dale, N., & Lewis, J. (2020). Computer Science Illuminated (7th ed.). Jones & Bartlett Learning, Ch 4

Basic hardware

- **Gate** - A device that performs a basic operation on electrical signals, accepting one or more input signals and producing a single output signal
  - There are **six** most fundamental types
  - Each type of gate performs a **particular logical function.**[5]

[5] Dale, N., & Lewis, J. (2020). Computer Science Illuminated (7th ed.). Jones & Bartlett Learning, Ch 4

Basic hardware

- **Circuit** - A combination of **interacting gates** designed to accomplish a specific logical function
  - circuits can be designed to **perform arithmetic** and to **store values**.
  - the output value of one gate often serves as an input value for one or more other gates.
  - The flow of electricity through a circuit is controlled by the carefully designed logic of the interacting gates. [6]

[6] Dale, N., & Lewis, J. (2020). Computer Science Illuminated (7th ed.). Jones & Bartlett Learning, Ch 4

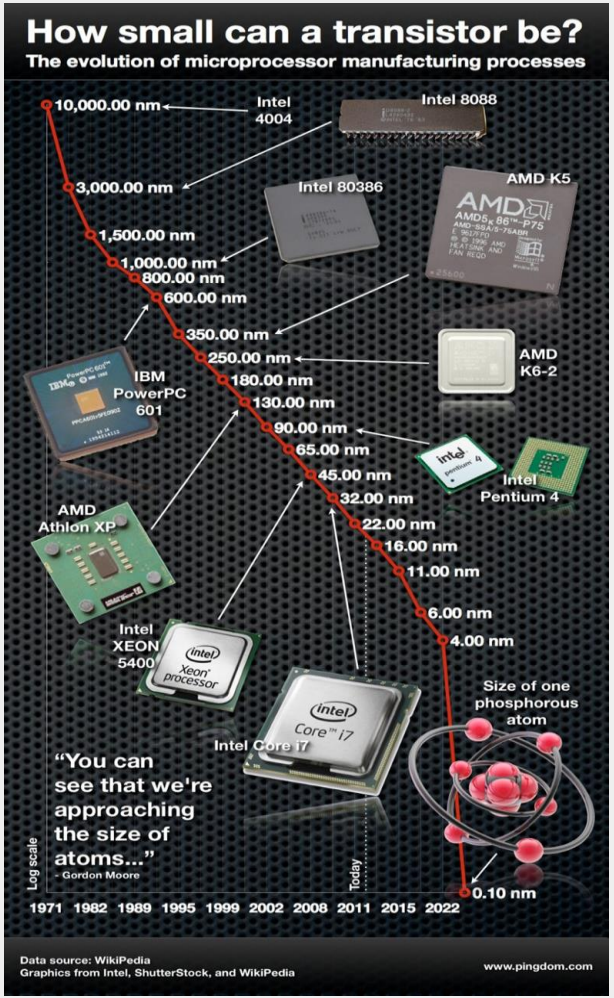
"The number of transistors incorporated in a chip will approximately double every 24 months."
—Gordon Moore, Intel co-founder

Transistor

- Transistor is basic hardware used to build the logic gates

  *__nm, nanometer__ is one millionth of a milimeter

- In **2012** a team of researchers in Australia has managed to create a transistor that is the **size of an atom**. [7]



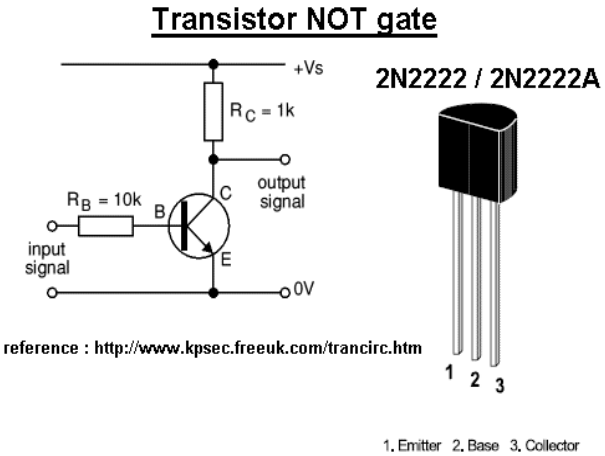
[7] Pingdom, S., 2012. The single-atom transistor is here – the amazing evolution of microprocessors. [Blog] TECH MUSINGS, Available at: <https://www.pingdom.com/blog/the-single-atom-transistor-is-here-the-amazing-evolution-of-microprocessors-infographic/ >

Transistor

- Transistor - A device that acts either as a wire or a resistor, depending on the voltage level of an input signal
  - Semiconductor - material such as silicon that is neither a good conductor nor a good insulator [8]

FIGURE 1.6 A vacuum tube

FIGURE 1.7 A transistor, which replaced the vacuum tube



[8] Dale, N., & Lewis, J. (2020). Computer Science Illuminated (7th ed.). Jones & Bartlett Learning, Ch 4

Transistors Explained - How transistors work

- Transistors Explained - How transistors work [9]

Arduino

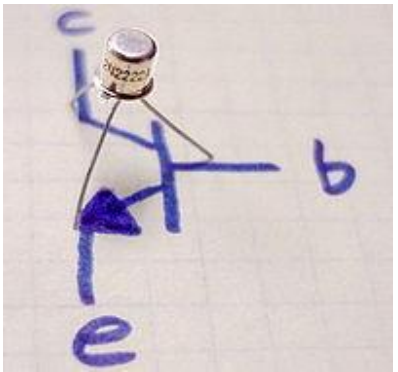
- Top 10 Arduino projects for beginners to advance
- https://www.arduino.cc/en/Guide/Introduction

[9]The Engineering Mindset, 2021, *Transistors Explained - How transistors work*. [online] Available at: <https://www.youtube.com/watch?v=J4oO7PT_nzQ >

Gates and Circuits: Notational methods

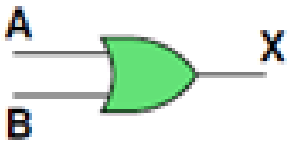
- There are three different, but equally powerful, **notational** methods for describing the behavior of gates and circuits:

1. Boolean expressions
2. Logic diagrams
3. Truth tables [8]



[8] Dale, N., & Lewis, J. (2020). Computer Science Illuminated (7th ed.). Jones & Bartlett Learning, Ch 4

Example: OR gate[9]

1. **X=A+B**
2. 

3. Truth Table

A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

[9] Dale, N., & Lewis, J. (2020). Computer Science Illuminated (7th ed.). Jones & Bartlett Learning, Ch 4

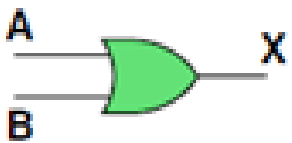
1. Boolean algebra: Boolean expressions[10]

- An English mathematician named George Boole invented a form of algebra in which variables and functions take on only one of two values (0 and 1).
- Specific operations and properties in Boolean algebra allow us to define and manipulate circuit logic using a mathematical notation.
- **OR gate boolean expression**: **X=A+B**

[10] Dale, N., & Lewis, J. (2020). Computer Science Illuminated (7th ed.). Jones & Bartlett Learning, Ch 4

2. Logic diagram [11]

- …is a graphical representation of a circuit. Each type of gate is represented by a specific graphical symbol.
- By connecting those symbols in various ways we can visually represent the logic of an entire circuit.
- **OR gate logic diagram:**



[11] Dale, N., & Lewis, J. (2020). Computer Science Illuminated (7th ed.). Jones & Bartlett Learning, Ch 4

3. Truth table [12]

- defines the function of a gate by listing all possible input combinations that the gate could encounter, and the corresponding output.
- **OR gate truth table:**

Truth Table

A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

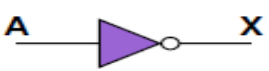
[12] Dale, N., & Lewis, J. (2020). Computer Science Illuminated (7th ed.). Jones & Bartlett Learning, Ch 4

6 most fundamental gate types

- NOT
- AND
- OR
- XOR
- NAND
- NOR

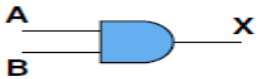
6 most fundamental gate types[13]

NOT gate inverts its single input value.

Boolean Expression	Logic Diagram Symbol	Truth Table
$x = \overline{A}$		

A	X
0	1
1	0

AND gate produces 1 if both input values are 1.

Boolean Expression	Logic Diagram Symbol	Truth Table
$x = AB$		

A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

[13] Dale, N., & Lewis, J. (2020). Computer Science Illuminated (7th ed.). Jones & Bartlett Learning, Ch 4

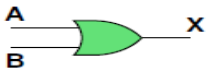
6 most fundamental gate types [14]

OR gate produces 1 if one or the other or both input values are 1.

Boolean Expression

$x = A + B$

Logic Diagram Symbol



Truth Table

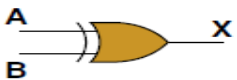
A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

XOR gate produces 1 if one or the other (but not both) input values are 1.

Boolean Expression

$x = A \oplus B$

Logic Diagram Symbol



Truth Table

A	B	X
0	0	0
0	1	1
1	0	1
1	1	0

[14] Dale, N., & Lewis, J. (2020). Computer Science Illuminated (7th ed.). Jones & Bartlett Learning, Ch 4

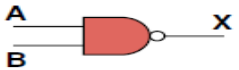
6 most fundamental gate types [15]

NAND gate produces the opposite results of an AND gate.

Boolean Expression

$$x = \overline{AB}$$

Logic Diagram Symbol



Truth Table

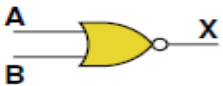
A	B	X
0	0	1
0	1	1
1	0	1
1	1	0

NOR gate produces the opposite results of an OR gate.

Boolean Expression

$$x = \overline{A + B}$$

Logic Diagram Symbol

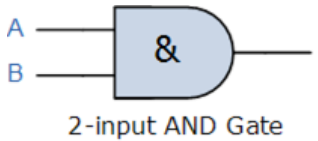


Truth Table

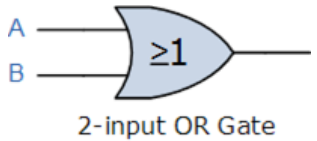
A	B	X
0	0	1
0	1	0
1	0	0
1	1	0

[15] Dale, N., & Lewis, J. (2020). Computer Science Illuminated (7th ed.). Jones & Bartlett Learning, Ch 4

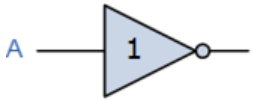
Let’s practice!

Symbol	Truth Table		
 2-input AND Gate	A	B	X
	0	0	
	0	1	
	1	0	
	1	1	
Boolean Expression X = AB	Read as A AND B gives x		

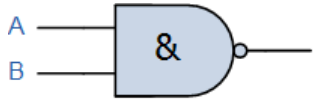
Let’s practice!

Symbol	Truth Table		
 2-input OR Gate	A	B	X
	0	0	
	0	1	
	1	0	
	1	1	
Boolean Expression X = A+B	Read as A OR B gives x		

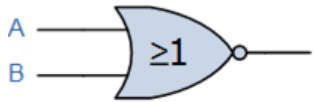
Let’s practice!

Symbol	Truth Table	
	A	x
	0	
	1	
Boolean Expression X = A’	Read as inversion of A gives Not x	

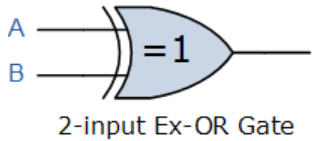
Let’s practice!

Symbol	Truth Table		
 2-input NAND Gate	A	B	x
	0	0	
	0	1	
	1	0	
	1	1	
Boolean Expression Q = (AB)’	Read as A AND B gives NOT-x		

Let’s practice!

Symbol	Truth Table		
 2-input NOR Gate	A	B	x
	0	0	
	0	1	
	1	0	
	1	1	
Boolean Expression x = (A+B)’	Read as A OR B gives NOT-x		

Let’s practice!

Symbol	Truth Table		
 2-input Ex-OR Gate	A	B	x
	0	0	
	0	1	
	1	0	
	1	1	
Boolean Expression Q = A⊕B			

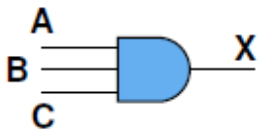
Gates with more inputs[16]

- Same rules apply but now to three inputs

Boolean Expression

ABC

Logic Diagram Symbol



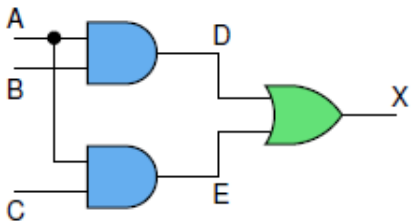
Truth Table

A	B	C	X
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

[16] Dale, N., & Lewis, J. (2020). Computer Science Illuminated (7th ed.). Jones & Bartlett Learning, Ch 4

Circuits[17]

- **Combinational circuit** - A circuit whose output is solely determined by its input values

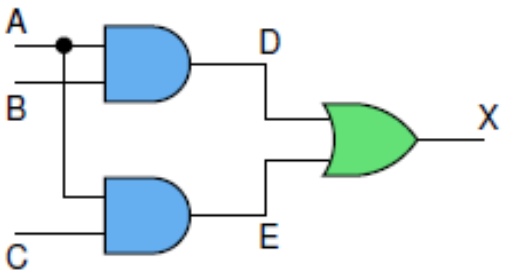


- **Sequential circuit** – A circuit whose output is a function of input values **and** the current state of the circuit (usually depend on clock cycles)
  - Normally used for storage

[17] Dale, N., & Lewis, J. (2020). Computer Science Illuminated (7th ed.). Jones & Bartlett Learning, Ch 4

Combinatorial circuit [18]

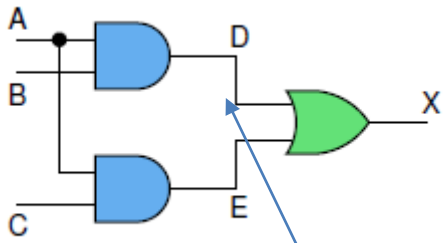
- Gates are combined into circuits by using gate output as an input to another gate



[18] Dale, N., & Lewis, J. (2020). Computer Science Illuminated (7th ed.). Jones & Bartlett Learning, Ch 4

Combinatorial circuit: Example 1

Logic diagram



Truth table

A	B	C	D	E	X
0	0	0			
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			

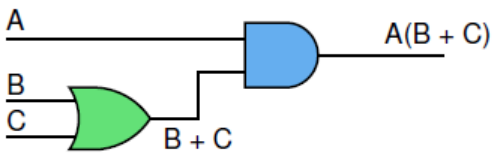
$X = (AB + AC)$

Boolean expression

Combinatorial circuit: Example 2

- Represent

  A(B + C) using logic diagram and truth table



A	B	C	B + C	A(B+C)
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

as good as possible…[19]

- When we build a combinatorial circuit from some kind of specification, we always try to make it ***as good as possible*.**
  - In some applications, **minimize the number of gates** (or the number of transistors).
  - In other, optimize **a *delay*** (the time it takes a signal to traverse the circuit) , or **low *power consumption***.
- In general, a mixture of such criteria must be applied.

[19] Dale, N., & Lewis, J. (2020). Computer Science Illuminated (7th ed.). Jones & Bartlett Learning, Ch 4

Circuit equivalence [20]

- **Circuit equivalence** - The same output for each corresponding input value combination for two circuits

- A(B+C) = AB + AC

[20] Dale, N., & Lewis, J. (2020). Computer Science Illuminated (7th ed.). Jones & Bartlett Learning, Ch 4

Few properties of Boolean algebra [21]

PROPERTY	AND	OR
Commutative	AB = BA	A + B = B + A
Associative	(AB)C = A(BC)	(A + B) + C = A + (B + C)
Distributive	A(B + C) = (AB) + (AC)	A + (BC) = (A + B) (A + C)
Identity	A1 = A	A + 0 = A
Complement	A(A') = 0	A + (A') = 1
De Morgan's law	(AB)' = A' OR B'	(A + B)' = A'B'

[21] Dale, N., & Lewis, J. (2020). Computer Science Illuminated (7th ed.). Jones & Bartlett Learning, Ch 4

What is the most basic operation of a computer?

What is the most basic operation of a computer?

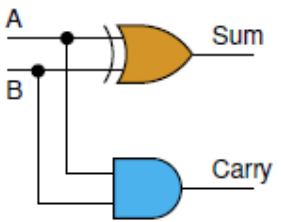
Add two numbers!

- At digital logic level the addition is performed in binary
- Special circuits are used to carry out this operation - **adders**

Half Adders [22]

- **Half adder** – A circuit that computes the sum of two bits and produces the correct carry bit
  - Half adder is fine for adding two single digits, but it **cannot** be used as is to compute the sum of two binary values with multiple digits each

A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

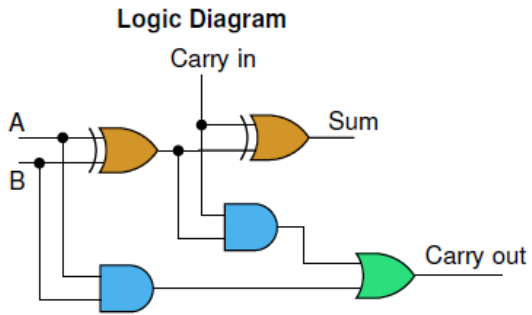


[22] Dale, N., & Lewis, J. (2020). Computer Science Illuminated (7th ed.). Jones & Bartlett Learning, Ch 4

Full Adders [23]

- A circuit called **a full adder** takes the carry-in value into account.
  - use two half adders to make a full adder

Homework: review how full adder works (Dale. Ch. 4)



Truth Table

A	B	Carry-in	Sum	Carry-out
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

[23] Dale, N., & Lewis, J. (2020). Computer Science Illuminated (7th ed.). Jones & Bartlett Learning, Ch 4

CPU chips

- CPU is an advanced circuit with input and output lines

Recommended reading and videos

1. Dale, N., & Lewis, J. (2020). Computer Science Illuminated (7th ed.). Jones & Bartlett Learning, Chapter 4, end of the chapter exercises
2. MIT (2020, June 5). If transistors can't get smaller, then software developers have to get smarter. Tech Xplore - Technology and Engineering news. Retrieved May 31, 2022, from https://techxplore.com/news/2020-06-transistors-smaller-software-smarter.html
3. Youtube.com. 2017. Boolean Logic & Logic Gates: Crash Course Computer Science #3. [online] Available at: <https://www.youtube.com/watch?v=gI-qXk7XojA > [Accessed 29 May 2022].
4. Youtube.com. 2017. How Computers Calculate - the ALU: Crash Course Computer Science #5 [online] Available at: <https://www.youtube.com/watch?v=1I5ZMmrOfnA&list=PL8dPuuaLjXtNlUrzyH5r6jN9ulIgZBpdo&index=7 > [Accessed 29 May 2022].

Python

Solutions to previous seminars

Recall countdown function, that prints the numbers from 10 to 0 and prints “Blastoff!”

```
def countdown(n):
    if n <= 0:
        print('Blastoff!')
    else:
        print(n)
        countdown(n -1)
countdown(10)
```

*You were asked on write a countup function. Solution is on the right

```
def countup(n):
    if n >= 0:
        print('Blastoff!')
    else:
        print(n)
        countup(n+1)

countup(-4)
```

Solutions to previous seminars

```
# Code #1 – simple iteration
string_name = "homework"

# Iterate over the string
for element in string_name:
    print(element, end=' ')
print("\n")

# Code #2 – range () function
string_name = "HOMEWORK"

# Iterate over index
for element in range(0, len(string_name)):
    print(string_name[element])
```

Reminder: MyMark (homework from TW3)

Write a program that will ask to input the module name and number of components on the module with corresponding weighting. For example:

Module name: Computer science fundamentals
Number of components: 2
Component 1 weighting: 40%, Component 2 weighting: 60%
Enter mark 1: 70, Enter mark 2: 60

It should also ask to enter mark for the components and show you the final mark.

Additionally you may include a feature, where user can input the desired overall mark for the module and program outputs the required marks for the components.

Save it it in TW4/ folder

Push your updates to Github repository.

Python for beginners in one hour

Python for Beginners - Learn Python in 1 Hour.[24]

[24] Hamedani, M., 2020. *Python for Beginners - Learn Python in 1 Hour*. [online] Youtube.com. Available at: <https://www.youtube.com/watch?v=kqtD5dpn9C8> [Accessed 5 June 2022].

References

1. Khan Academy and Code.org, 2018, Circuits & Logic. [online] Available at: <https://youtu.be/Sc3lh3D4rCw> [Accessed 30 May 2022].
2. Dale, N., & Lewis, J. (2020). Computer Science Illuminated (7th ed.). Jones & Bartlett Learning, Chapter 4
3. Pingdom, S., 2012. The single-atom transistor is here – the amazing evolution of microprocessors. [Blog] TECH MUSINGS, Available at: <https://www.pingdom.com/blog/the-single-atom-transistor-is-here-the-amazing-evolution-of-microprocessors-infographic/> [Accessed 30 May 2022].
4. Downey, A. (2015). Think python. O'Reilly. Chapter 1, 2
5. Hamedani, M., 2020. Python for Beginners - Learn Python in 1 Hour. [online] Youtube.com. Available at: <https://www.youtube.com/watch?v=kqtD5dpn9C8> [Accessed 29 May 2022].
6. Youtube.com. 2017. Boolean Logic & Logic Gates: Crash Course Computer Science #3. [online] Available at: <https://www.youtube.com/watch?v=gI-qXk7XojA> [Accessed 29 May 2022].
7. Youtube.com. 2017. How Computers Calculate - the ALU: Crash Course Computer Science #5 [online] Available at: <https://www.youtube.com/watch?v=1I5ZMmrOfnA&list=PL8dPuuaLjXtNlUrzyH5r6jN9ulIgZBpdo&index=7> [Accessed 29 May 2022].
8. The Engineering Mindset, 2021, *Transistors Explained - How transistors work*. [online] Available at: https://www.youtube.com/watch?v=J4oO7PT_nzQ