Activity 1B Classical Computation (Siddhant) 60 min

Theory 1 / Introduction to the bit and information (15 min)

Goal: Students learn how to represent information. How this information evolves and represents anything that is to be conveyed in a very simple manner at the fundamental level.

Teacher's Guide for information to be delivered:

- a "bit" is the smallest unit of storage
- A bit stores just a 0 or 1
- "In the computer it's all 0's and 1's" ... bits
- Anything with two separate states can store 1 bit, examples:
    - In a chip: electric charge = 0/1
    - In a hard drive: spots of North/South magnetism = 0/1

A bit is too small to be much use, hence combine bits together.

Number of bits	Different Patterns
1	0 1
2	00 01 10 11
3	000 001 010 011 100 101 110 111

Bits together: Powerful to represent chunk of information!

n bits yields 2^n patterns (2 to the nth power). Use these patterns to mean different numbers and symbols!

Students' Handout:

-

Conclusion from the activity:

Information storage = Done! Is that all we need? NO!

Discussion 2 (10 min)

Fun activity [Goal]: Come up with a funny bit! Be creative!

Teacher's Guide:

Ask the students to think of the funniest two level system they can come up with. Represent a number with it!? Who gets the most laughter gets a goodie.

Student:

Addresses the stated problem in their own creative way. For instance, puts forward the idea that one can use the bottle/cup placed upright or upside down and make a bit out of it.

Teacher's Guide:

How to calculate and compute something from these bits? How to make bits useful!

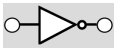
=> Play around with these collection of bits. For instance, add, subtract and do all sorts of complicated "boolean" (Yes/No or True/False) operations!

Build the logic and by constructing operations.

Theory 2: (35 min)

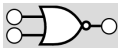
Goal: What operations can we do with bits? Can we mimic all possible operations? What is the minimal set of 'gates' for this?

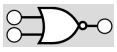
Notation to stick to for entire program:

 NOT gate

 OR gate

 AND gate

 NOR gate

 NAND gate

Can also make a nice big poster to be referred to both in 1B and 1C modules.

Teacher's Guide:

Introduce decimal to binary! How to write numbers? **(10 mins)**

- Step 1: Divide the given decimal number by 2 and note down the remainder.
- Step 2: Now, divide the obtained quotient by 2, and note the remainder again.
- Step 3: Repeat the above steps until you get 0 as the quotient.
- Step 4: Now, write the remainders in such a way that the last remainder is written first, followed by the rest in the reverse order.

Show example in the omnicalculator:

https://www.omnicalculator.com/conversion/binary

Students:

Convert different decimals to binary in the calculator and verify the steps.

Teacher's Guide:

Every time you hit a power of 2, you increase one bit. Table up to decimal 10:

Decimal Numbers	Binary Numbers
0	0

Decimal Numbers	Binary Numbers
1	1
2	10
3	11
4	100
5	101
6	110
7	111
8	1000
9	1001
10	1010

Introduce bitwise addition! **(10 mins)**

- Binary addition is much like your normal everyday addition (decimal addition), except that it carries on a value of 2 instead of a value of 10.
- For example: in decimal addition, if you add 8 + 2 you get ten, which you write as 10; in the sum this gives a digit 0 and a carry of 1. Something similar happens in binary addition when you add 1 and 1; the result is two (as always), but since two is written as 10 in binary, we get, after summing 1 + 1 in binary, a digit 0 and a carry of 1.
- Therefore in binary:
    - 0 + 0 = 0
    - 0 + 1 = 1
    - 1 + 0 = 1
    - 1 + 1 = 10 (which is 0 carry 1)
- Can show on the calculator @ https://www.calculator.net/binary-calculator.html

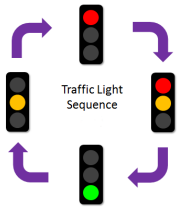
Students:

Play around with the binary adder for different binary numbers and verify the logic.

Teacher's Guide:

How to move to real world applications with bit computation?

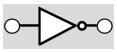
Example: They are used in the functioning of street lights and traffic lights! Based on which side is ongoing traffic, which road lanes should stop!



Let us build the gates piece by piece. Starting with the most basic ones!

Introduction to the logic gates **(10 mins)**

We construct truth tables, which tell us what are the inputs for the gate and what are the outputs! We cover three basic and intuitive gates and later move on to the complex gates!

- NOT gate: 

Input	Output
0	1
1	0

One input and one output!

- OR gate: 

Input		Output
A	B	C
0	0	0
0	1	1
1	0	1
1	1	1

Two inputs and one output!

[Inputs can be more]

- AND gate: 

Input		Output
A	B	C
0	0	0

0	1	0
1	0	0
1	1	1

Two inputs and one output!

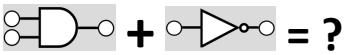
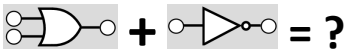
[Inputs can be more]

Discussion 2: Putting it altogether (5 min)

Goal: To use a given set of gates and make complex gates out of those gates.

Teacher's Guide:

Activity: Students can make truth tables for combined gates **(5 mins)**

-  **= ?**
-  **= ?**

  They just need to invert the final output.

Students:

Students make Truth table for NAND and NOR gates using the information above!

Fill the table below each for NAND and NOR gates.

Input		Output
A	B	C
0	0	?
0	1	?
1	0	?
1	1	?

Additional information:

- https://www.cuemath.com/numbers/binary-addition/ for binary addition
- https://www.electronics-tutorials.ws/logic/universal-gates.html for logic gates!