**Name - Srajan Ahuja**

**100 number - 1001858728**

**Date - 01/31/2021**

**Homework 2**

**Q1.** Describe the Difference engine.

**A1.**

The Difference Engine was created by Charles Babbage in 1822, it is a mechanical device which has the purpose of computing polynomials as well as logarithmic and trigonometric functions as it was the need of the hour in that day and age since people had to rely on time consuming step reckoner for various computations and use cases (eg. changes in a military cannon would lead to a new set of values for all the variables involved). The Difference Engine consists of 25,000 components that weigh 15 tons. Babbage couldn’t complete it’s invention due to lack of funding but it was later built in 1991 by historians.

**Q2.** From the following list, which can be considered as a “general purpose computer”?

a. Step Reckoner

b. Difference Engine

c. Analytical Engine

**A2.**

Only option c - Analytical Engine can be considered as a general purpose computer for the following reasons : -

* It has memory to store data.
* It can schedule multiple operations in a queue.

**Q3.** What were the limitations of Harvard Mark 1? How does current technology deal with those problem?

**A3.**

Harvard Mark 1 has the following limitations :-

1. Time Consuming Design :- HM1 used relays for circuit switching, which weren’t fast enough to compute complex problems (3 additions/subtractions per second, Multiplication - 6 seconds, Division - 15 seconds, Trigonometric problems - more than a minute)
2. Mechanical Wear and Tear :- HM1 had 3500 relays which made it a high maintenance computing device which could average around changing one faulty relay everyday making it unreliable to solve problems on a daily basis.

Modern day computing is electronic computing instead of electro-mechanical like HM1, decreasing the time complexity by faster circuit switching and making it low maintenance as there is no movement based mechanical aspect to it.

**Q4.** Why is Silicon Valley so called?

**A4.**

The Santa Clara Valley had a huge industry for manufacturing transistors and semiconductors, which were mostly created using Silicon. Hence it got the name of Silicon Valley.

**Q5.** Make boolean logic tables for AND, OR, and XOR (A XOR B).

**A5.**

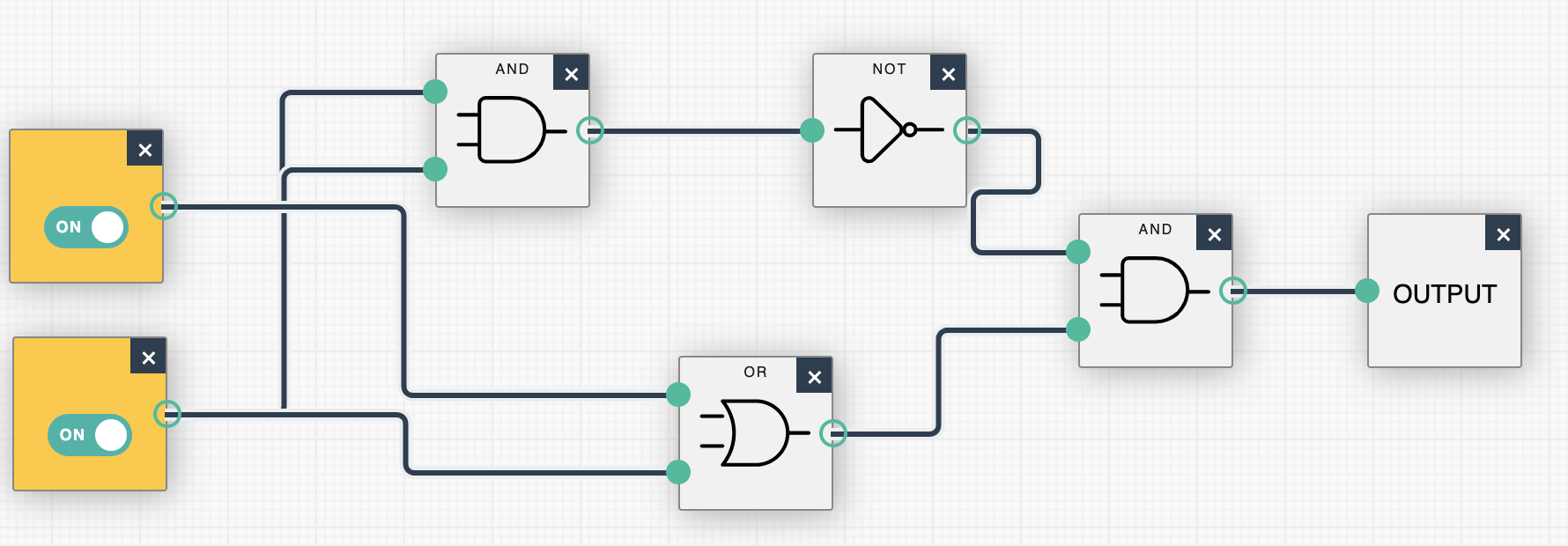
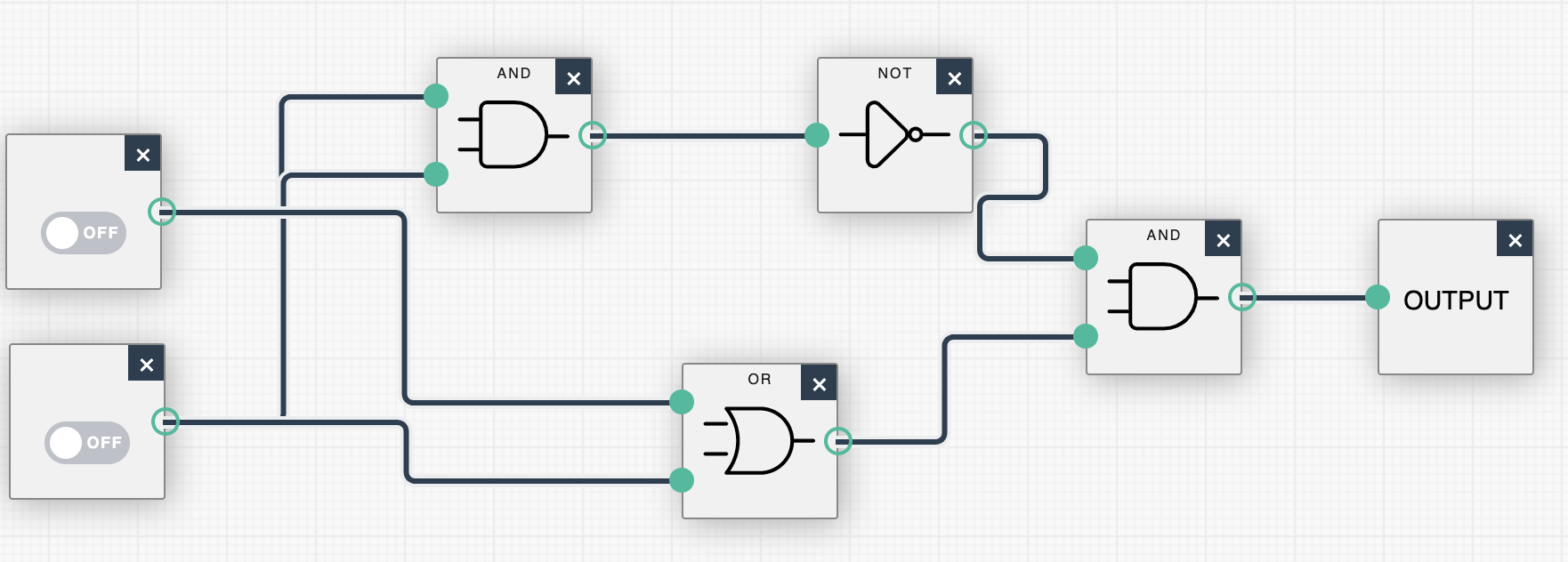
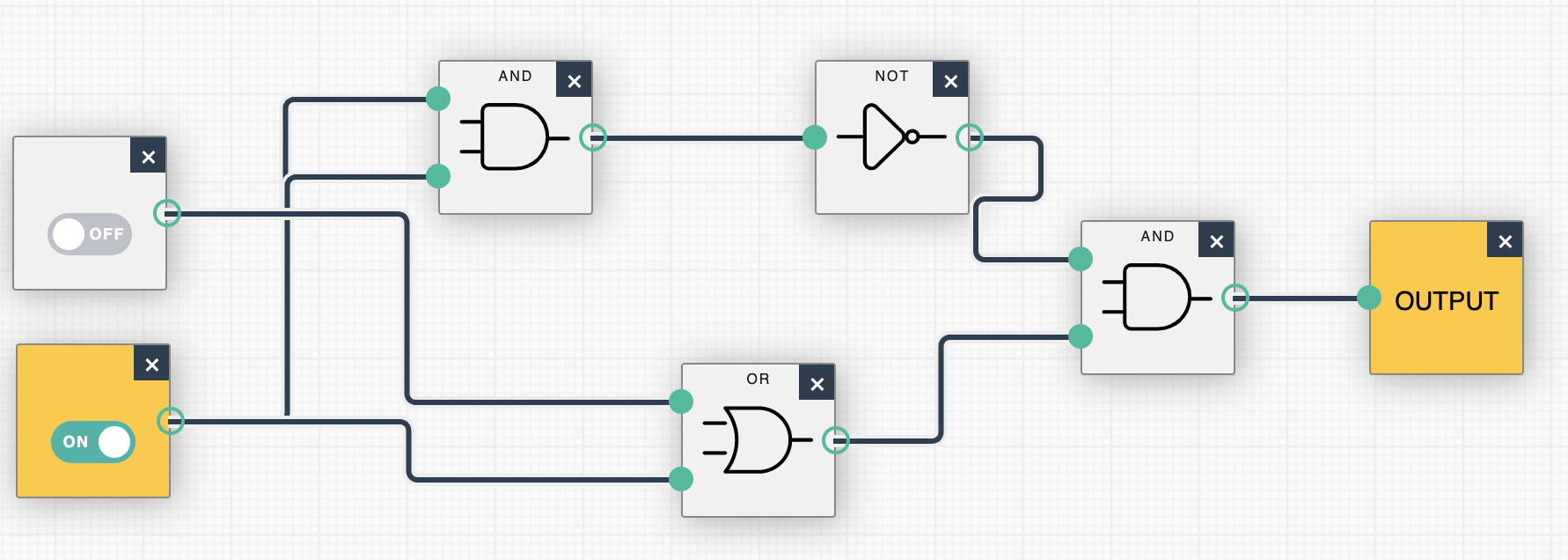
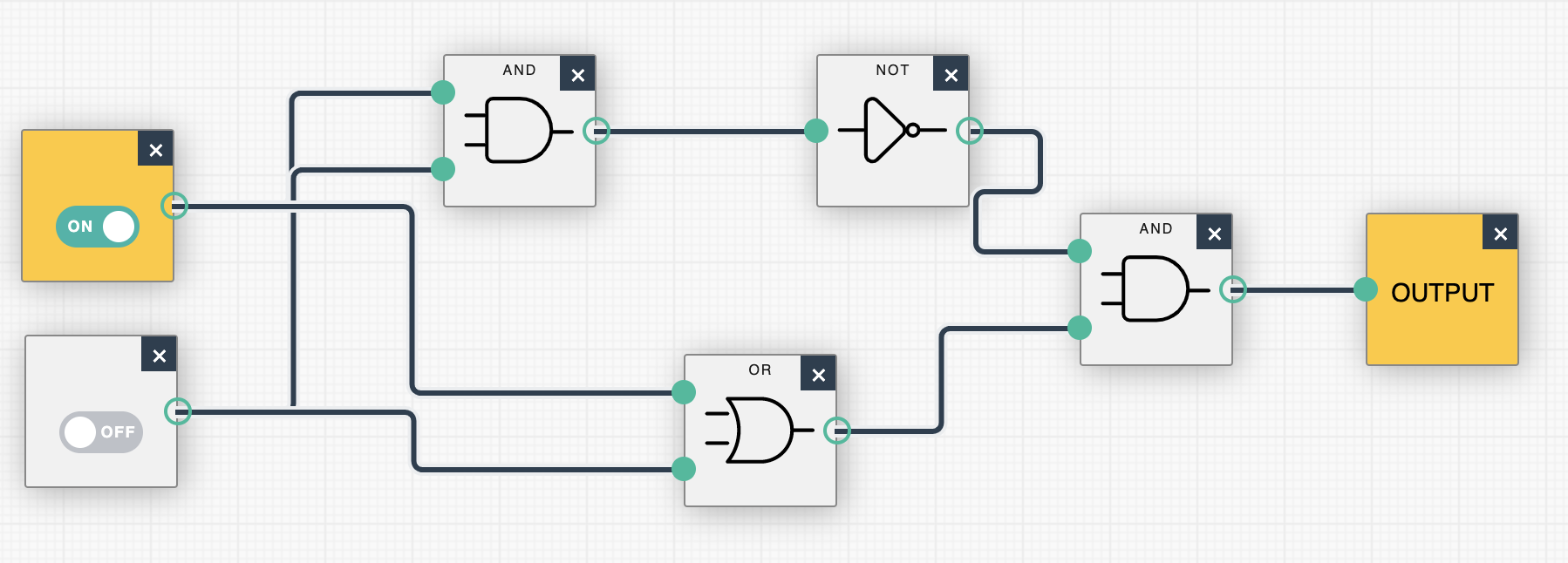
Logic Tables :- (1 = TRUE, 0 = FALSE)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A** | **B** | **AND** | **OR** | **XOR** |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 | 0 |

**Q6.** Can we implement XOR gate using only NOT, AND, and OR gates? If yes, how?

**A6.**

Yes :-



[Simulations created using Logic gate simulator on academo.org]

**Q7.** How many bits would you need to represent 2018 in binary? How many bytes is that?

**A7.**

201810 = 111111000102 = 11 bits = 1.375 bytes

**Q8.** Write “Programming Languages” (excluding the quotes) using ASCII code.

**A8.**

Programming Languages using ASCII :-

80 114 111 103 114 97 109 109 105 110 103 32 76 96 110 103 117 97 103 101 115

**Q9.** What were the shortcomings of Assembly languages?

**A9.**

* Assembly language is prone to bugs.
* Difficult to understand the code
* It is not portable as it varies from architecture to architecture.

**Q10.** What were the advantages and limitations of ALGOL?

**A10.**

Advantages of ALGOL :-

* ALGOL 58 introduced the idea of data types, it also added the functionality of compound statements.
* Upon its inception, ALGOL introduced block structure which consisted of data and other instruction statements in the same structure like an entire program.
* ALGOL became the first language to apply nested function definitions with lexical scope.

Disadvantages of ALGOL :-

* It lacked standardized Input/Output making the portability come with a lot of hassle.
* ALGOL 60 had certain feature that were too flexible, making it tough to understand and inefficient to implement.

**Q11.** What is the difference between parameters and arguments of a procedure? Explain with appropriate examples.

**A11.**

A parameter is the placeholder element while defining a method which is used only in the scope of that method, whereas an argument is the actual value we pass in the parameter when calling the method.

For example :-

**“**

void FunctionABC (int x) {

x = x+10;

cout<< x;

}

int main (){

Int a;

a = 5;

FunctionABC(a);

return 0;

}

**”,** where x is a parameter and a is an argument.

**Q12.** Write a recursive factorial function/method in C++ or Java (No limitation on number of lines.

**A12.**

int fact(int x){

if (x==0 || x==1){

return 1;

}

else {

return x\*fact(x-1);

}

}

**Q13.** Explain Language Syntax and Language Semantics in your own words.

**A13.**

Language syntax is the way/rules of writing a statement. For example:-

A for loop starts with a “for” followed by three statements in parenthesis being initialisation, condition and variable updation.

Language semantic is what the sequence of statements actually mean. For example:-

A for loop is executed by initialising the variable present in the for statement followed by checking if the value of variable satisfies the condition then the statements followed by the for loop are executed and in the end variable updates and starts back from the condition checking step.

**Q14.** Explain how Java codes are compiled and then interpreted.

**A14.**

A java code when run, first goes to the Java Compiler where it is converted into java bytecode. Java bytecode is platform independent and then it goes to the Java Virtual Machine (JVM) where it interprets the bytecode and executes. This process however is sometimes time consuming, that’s why modern JVM’s convert the bytecode into native machine code and execute the machine code instead of interpreting the bytecode.

**Q15.** Explain the following terms:

a. Syntactic sugar

b. API

**A15.**

1. **Syntactic sugar -** Syntactic sugar are the shortcuts provided by a programming language to make it easier to read/write without adding any additional functionality to it which wasn’t already there. For example :-

“ x=x+10” can also be written as “x+=10”

1. **API -** API is the abbreviated form of Application Programming Interface. API is basically a set of protocols and routes designed for two softwares to interact with each other.

**Q16.** Do you think that C programming language is a successful programming language even though it’s not the most popular language today? Explain your reasonings.

**A16.**

Yes, C programming language is a successful language despite not being popular in the modern day and age for the following reasons :-

1. Fast - Modern CPUs are all built on C and the only way to go faster is to use Assembly Language.
2. Simplicity - C lacks constructors, destructors, name manglings, garbage collectors, etc. which makes the performance easier to predict and makes the build times less.
3. Versatile - Most kernels are written in C, making it the best language to get to the core. C doesn’t require huge systems making it a good option to program IOT devices.
4. For dynamic power, C can always be linked with Lua which provides it with abstractions and flexible programming while running it.
5. Old - C has been around for a long time and is still in use which means so many programmers have assured perfection in this language.
6. C is the best language to start off one’s programming journey due to its simplicity and ease of use.