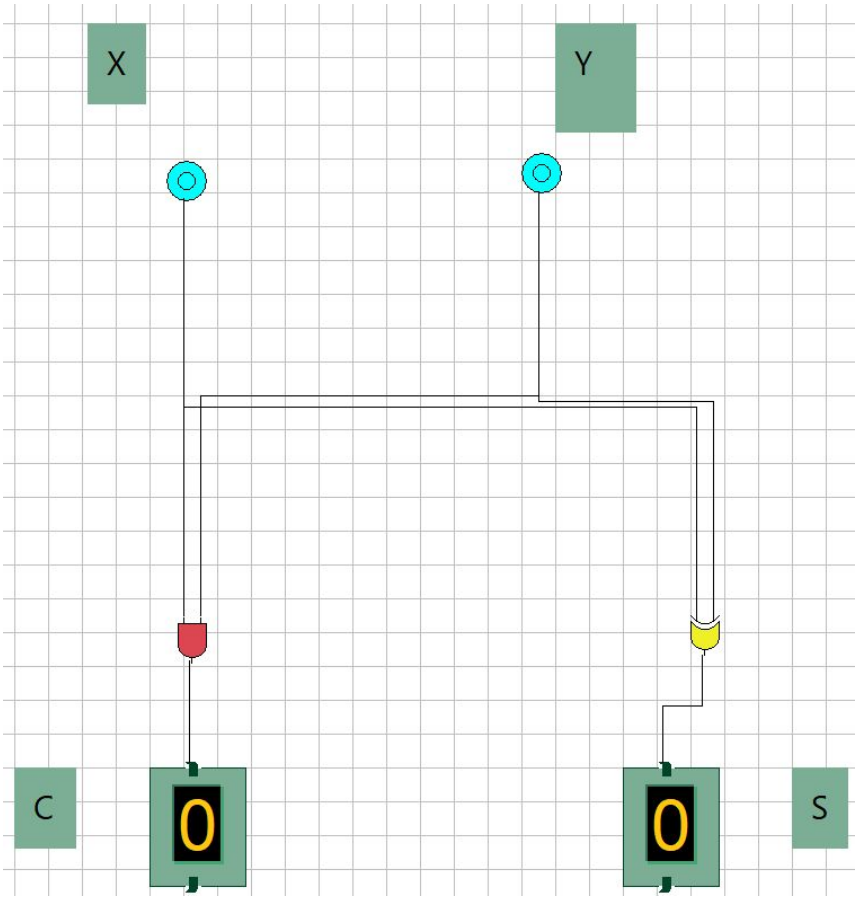
Elementary Adders

Group - 11

An **Adder** is a device that can add two binary digits. It is a type of digital circuit that performs the operation of additions of two numbers. It is mainly designed for the addition of binary numbers, but they can be used in various other applications like binary code decimal, address decoding, table index calculation, etc.

1. Half - Adders :

There are two inputs and two outputs in a **Half Adder**. Inputs are labelled as X and Y, and output as S (for sum) and C (for carry).



Delay calculation :

Assuming same propagation delay through all basic gates such as AND, OR and NOT gates as δ.

So the delay for carry is δ as only an AND gate is required to calculate carry.

The delay for sum is 3δ as an XOR gate is required and it has three levels of NOT, AND and OR gates.

2. Full - Adders :

The full adder circuit has three inputs: A, B and C, with this logic circuit, two bits can be added together, taking a carry from the next lower order of magnitude, and sending a carry to the next higher order of magnitude.

The first two inputs are A and B and the third input is an input carry as C-IN.

We can make a full adder using half adder components as shown below in the circuit diagram.

2 Half Adders and a OR gate is required to implement a Full Adder.

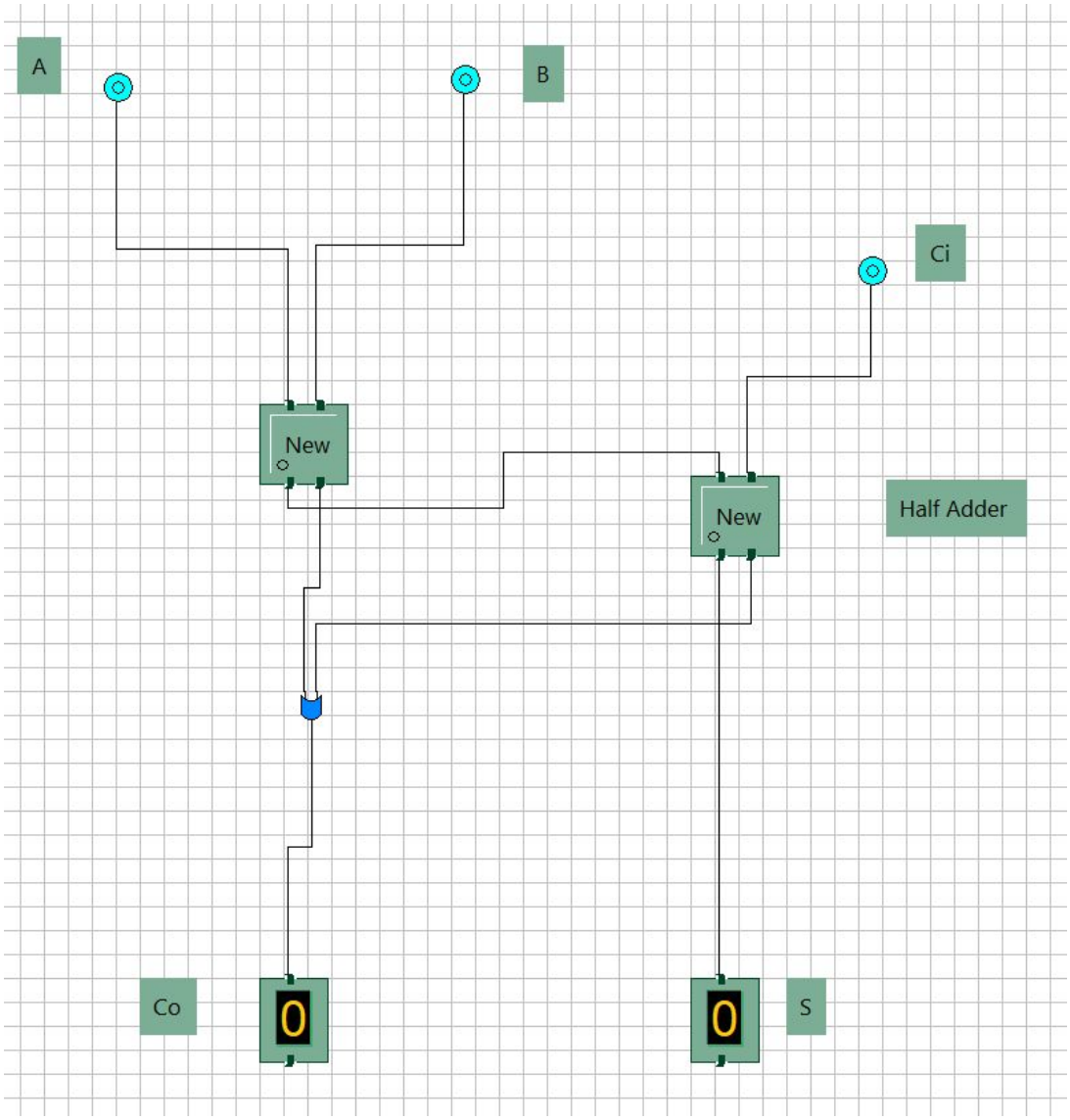
Delay calculation :

Assuming same propagation delay through all basic gates such as AND, OR and NOT gates as δ.

So the delay for carry is 5δ:

3δ for computing sum from first half adder, another δ from computing carry from second half adder and finally one last δ from last OR gate.

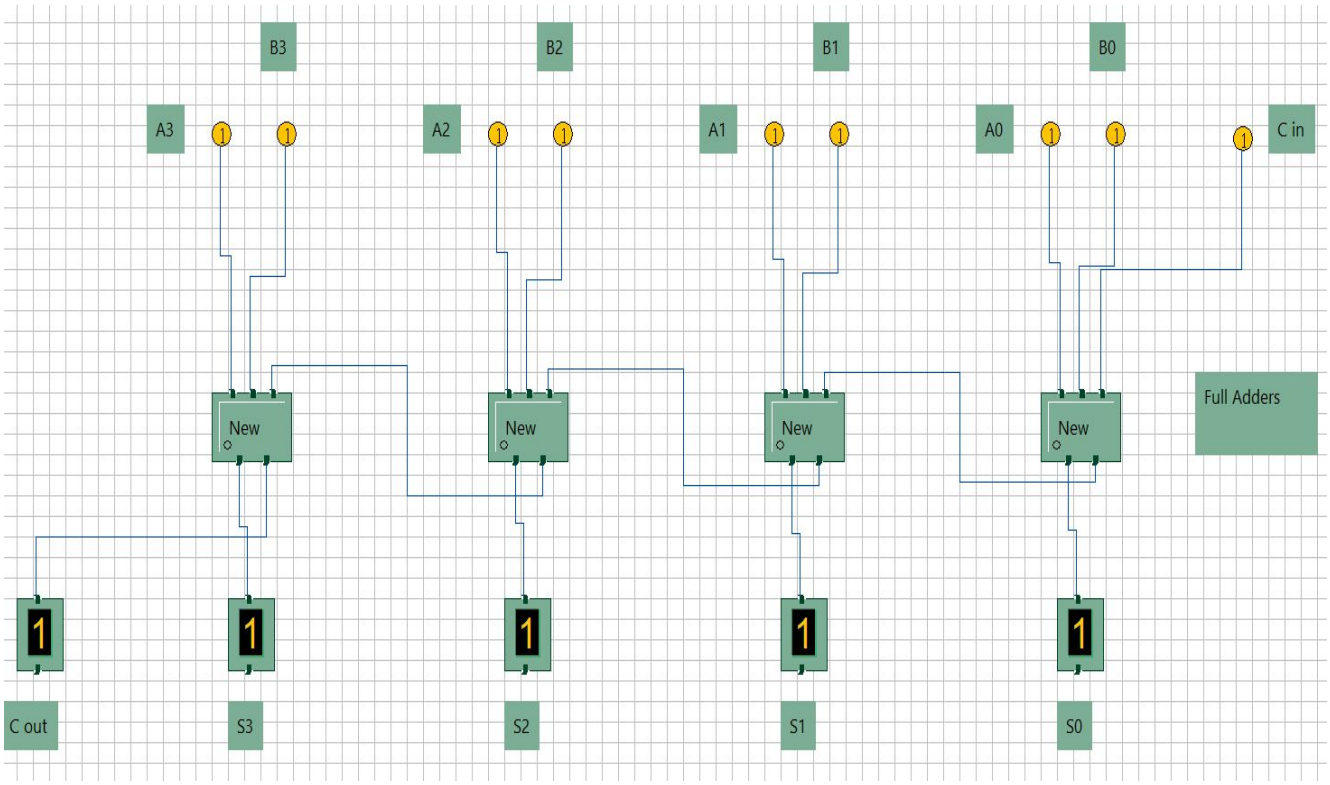
The delay for sum is 6δ as there are two half adders and each half adder takes 3δ for sum (3δ + 3δ).



3. Ripple Carry Adder :

In the ripple carry adder, we add two
A n-bit ripple carry adder is created by cascading n full adders.

The sum of the most significant bit is only available after the carry signal has rippled through the adder from the least significant stage to the most significant stage. As a result, the final sum and carry bits will be valid after a considerable delay.
Following is a 4-bit Ripple carry adder :



Delay calculation :

Assuming same propagation delay through all basic gates such as AND, OR and NOT gates as δ.
Delay for sum for a Full Adder is 6δ.
Delay for carry for a Full Adder is 5δ.
For a n-bit ripple carry adder :

Delay for S_0 is 6δ. For C_1 it is 5δ. For S_1 it is delayed for C_1 and delay for computing S_1, so it is 5δ+3δ = 8δ

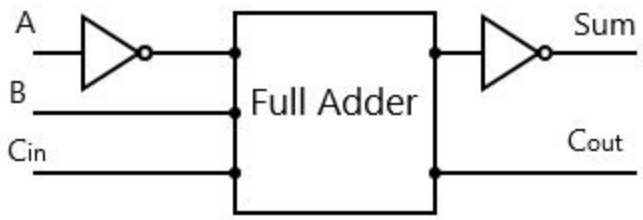
So the overall delay for S_{n-1} is 2(n-1)δ+3δ because of C_{n-1} and 3δ to compute S_{n-1} using that. So the total delay (S_{n-1}) is (2n+4)δ. Similarly delay for C_n is (2n+3)δ.
So the total delay of RCA with n bits is (2n+4)δ, which is O(n).

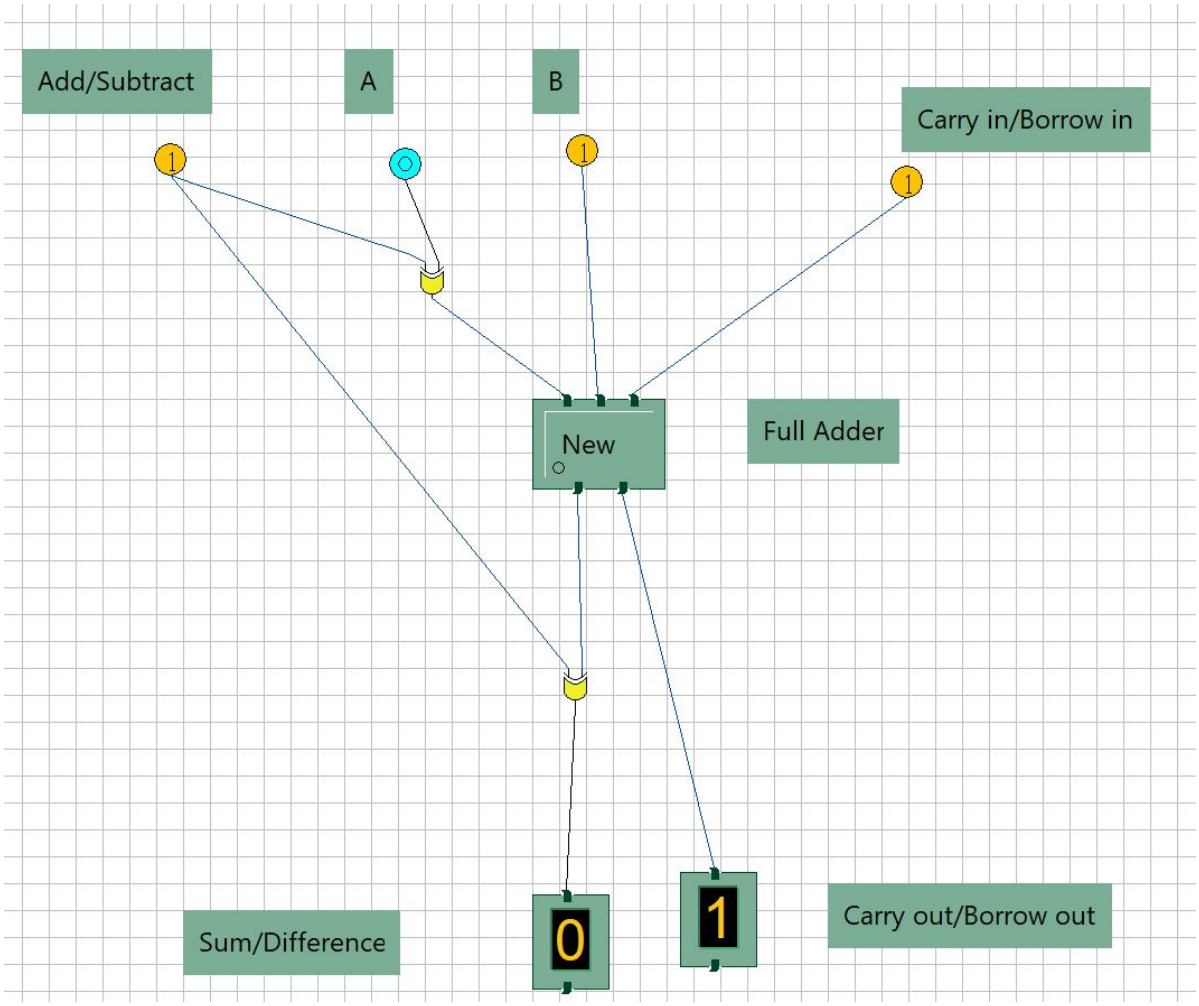
4. Full Adder/Subtrator :

A Full Adder/Subtractor is one which is capable of both addition and subtraction of binary numbers in one circuit itself. The operation being performed depends upon the binary value the control signal holds.

The sum bit in Full Subtractor is the same as in Full Adder which is (if inputs are A,B,C) A⊕B⊕C. But the borrow bit(which used to be sum in Full Adder) differs. The Borrow bit in Full Subtractor is A'B+A'C+BC. So to implement full subtractor using full adder, we can insert a NOT gate after A to change the borrow bit to A'B+A'C+BC but the sum bit becomes A'⊕B⊕C. So if we put a NOT gate before sum it becomes A⊕B⊕C, which is the required output.

To implement both adder and subtractor, we use XOR gates instead of NOT gates. If the control signal is 1, XOR gate works as NOT gate, else it works as a short circuit, working as a normal adder.





Delay Calculation:

We will add the extra delay due to the introduction of XOR gates to the delay of full adder, which we already computed. We introduced an XOR gate before A, so we will add 3δ(delay of XOR) to both carry and sum. There is an extra XOR gate before sum, so we need to add an extra 3δ to sum.

Overall, delay of sum = 6δ(from full adder) + 3δ*2(from 2 XOR gates)
Delay of carry = 5δ(from full adder) + 3δ(from XOR gate after A).
Finally,

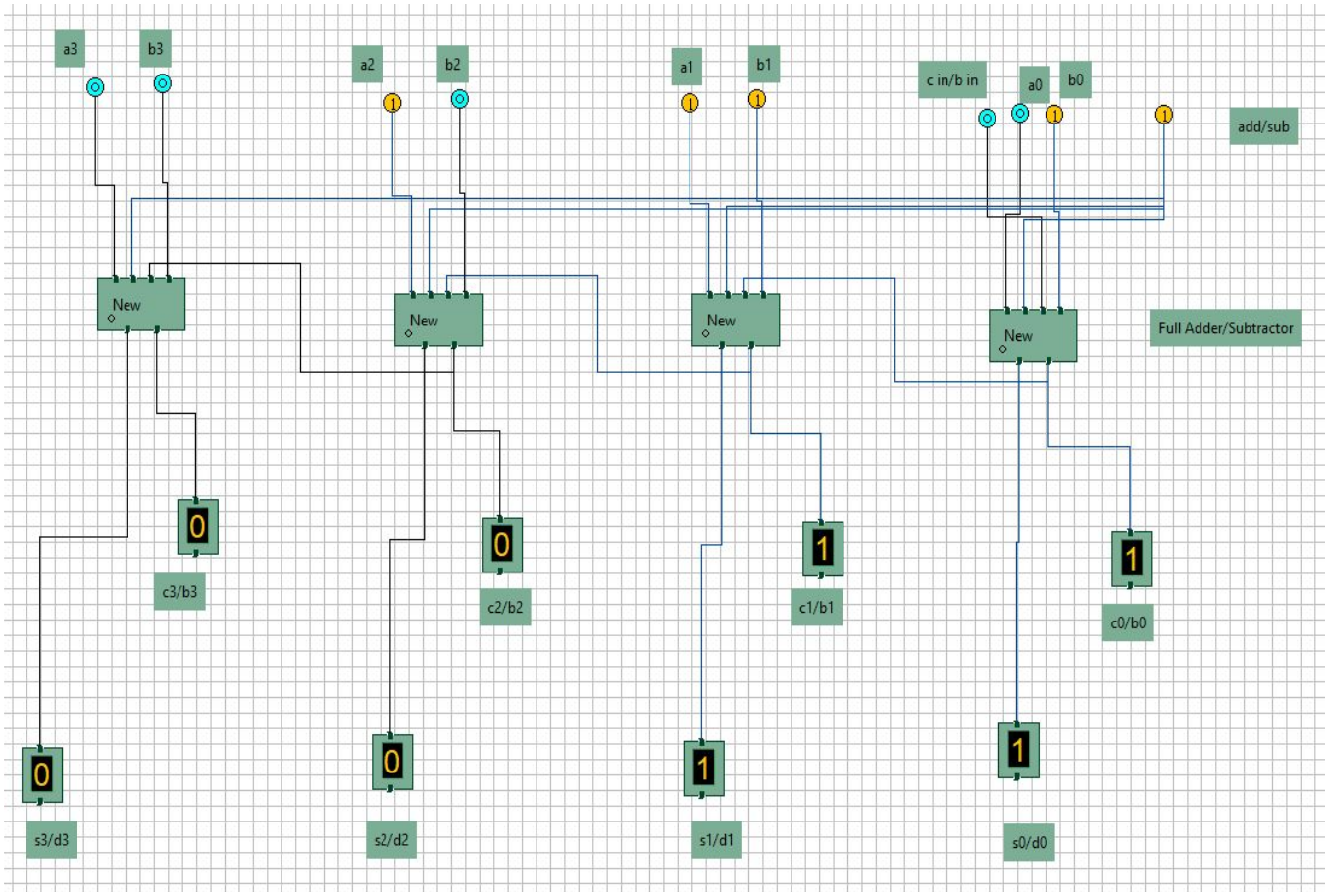
Delay of sum = 12δ and Delay of carry = 8δ.

5. Ripple Carry Adder/ Subtractor :

Ripple carry adder/subtractor is designed to add/subtract two bit vectors to generate their sum/difference vector and the carry out.

An n-bit ripple adder/subtractor is created by cascading n full adder/subtractors.

Ripple carry adder/subtractor has a control bit, which decides whether to add/subtract the given two bit vectors. This works extremely similar to ripple carry adder, except that when it's subtracting, it passes borrow rather than pass carry like the ripple carry adder.



Delay Calculation:

Assuming same propagation delay through all basic gates such as AND, OR and NOT gates as δ.

Delay for sum for a Full Adder/Subtractor is 12δ.

Delay for carry for a Full Adder/Subtractor is 8δ.

For a n-bit ripple carry adder/subtractor :

Delay for S_0 is 12δ. For C_1 it is 8δ. For S_1 it is delayed for C_1 and delay for computing S_1, so it is 8δ+6δ = 14δ.

Delay for C_n is (2n+6)δ by :

- 3δ while calculating A' from A for the first adder;
- 2δ at each step for calculating carry upto nth adder, so total 2(n-1)δ;
- 5δ for calculating carry for first adder (C0).

So the overall delay for S_{n-1} is 2(n-1)δ+6δ because of C_{n-1} and 3δ due to the second XOR of the full adder/subtractor and 3δ to compute S_{n-1} using that. So the total delay (S_{n-1}) is (2n+10)δ.

So the total delay of RCA/S with n bits is (2n+10)δ, which is O(n).

Contributors :

Divyansh Bhatia	-	19CS10027
Esha Manideep Dinne	-	19CS10030
Aryan Agarwal	-	19CS30005
Amartya Mandal	-	19CS10009
ASRP Vyahruth	-	19CS10002

Work Distribution :

Esha Manideep Dinne	-	Half Adder
Divyansh Bhatia	-	Full Adder
Amartya Mandal	-	Ripple Carry Adder
ASRP Vyahruth	-	Full Adder/Subtractor
Aryan Agarwal	-	Ripple Carry Adder/Subtractor

Everyone wrote their respective parts of the theory and everybody contributed in calculating the delays.

********** Thank You **********