Computer Organisation and Architecture Lab

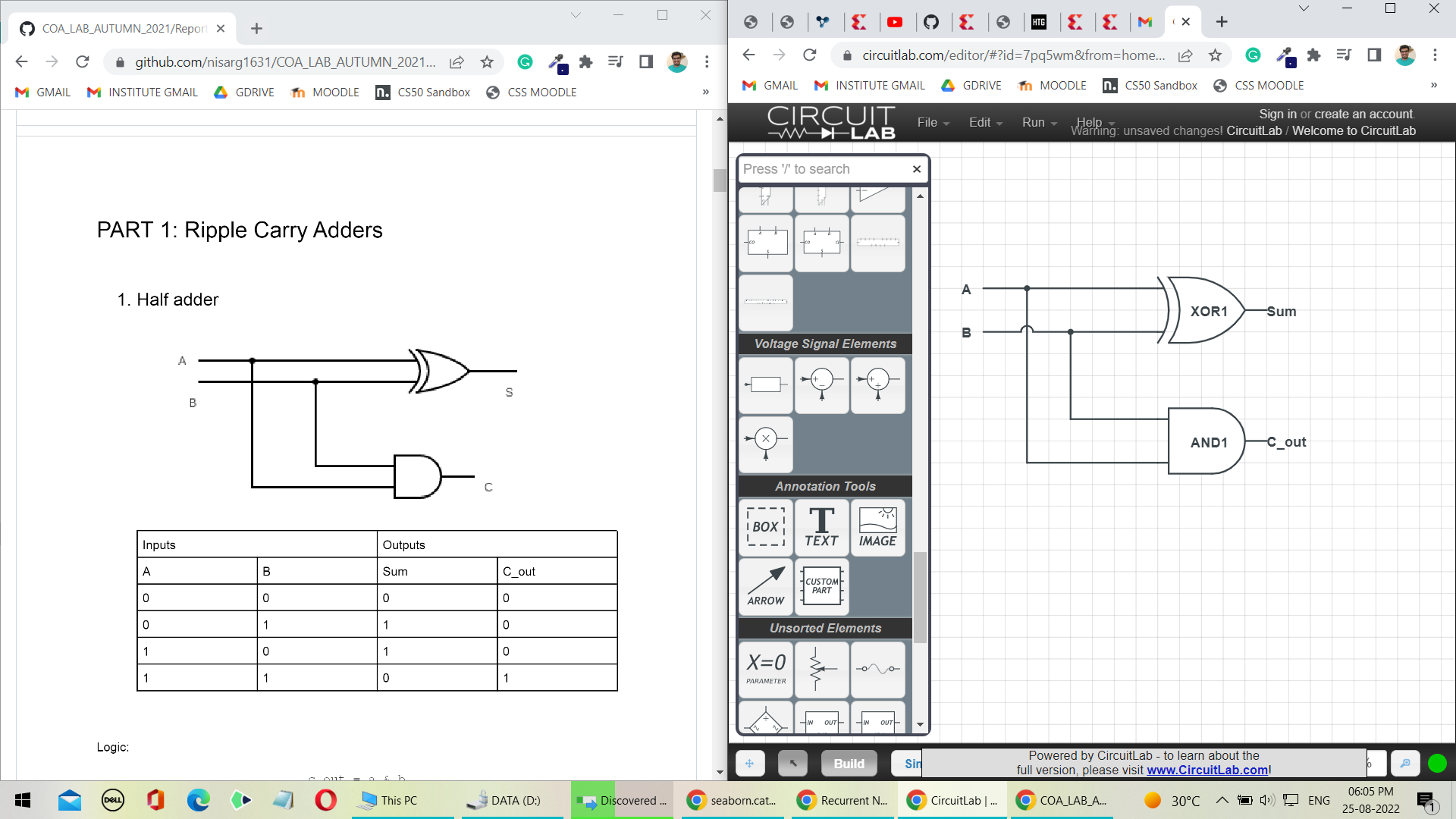
Group number: 65

Members: Pranav Mehrotra ( 20CS10085)

Saransh Sharma (20CS30065)

Q1) RCA using Verilog

1. Half Adder



c

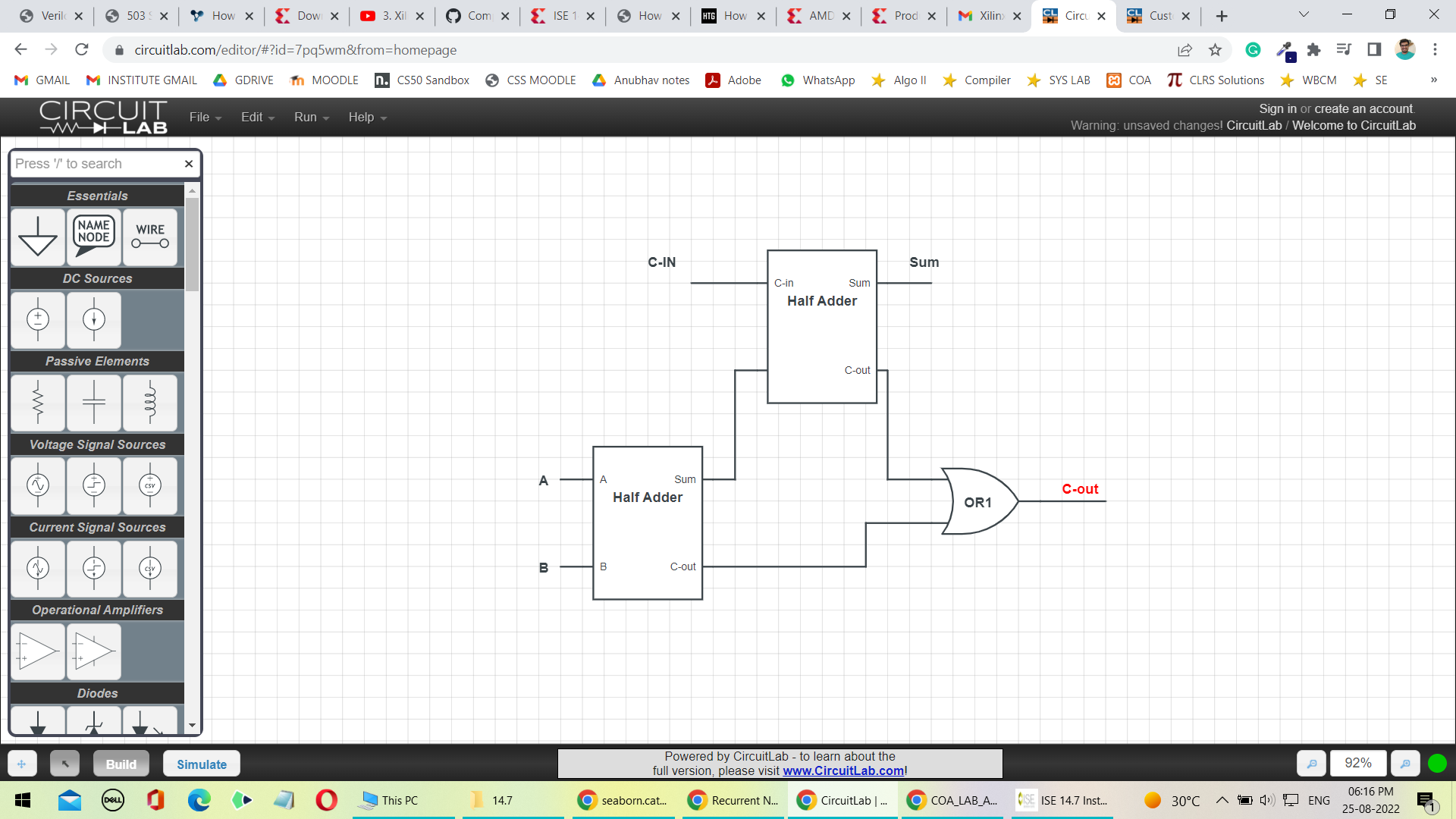
s

a

b

|  |  |  |  |
| --- | --- | --- | --- |
| Truth Table of a Half Adder | | | |
| Input | | Output | |
| A | b | s (Sum) | c (carry) |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |

1. Full Adder



c

s

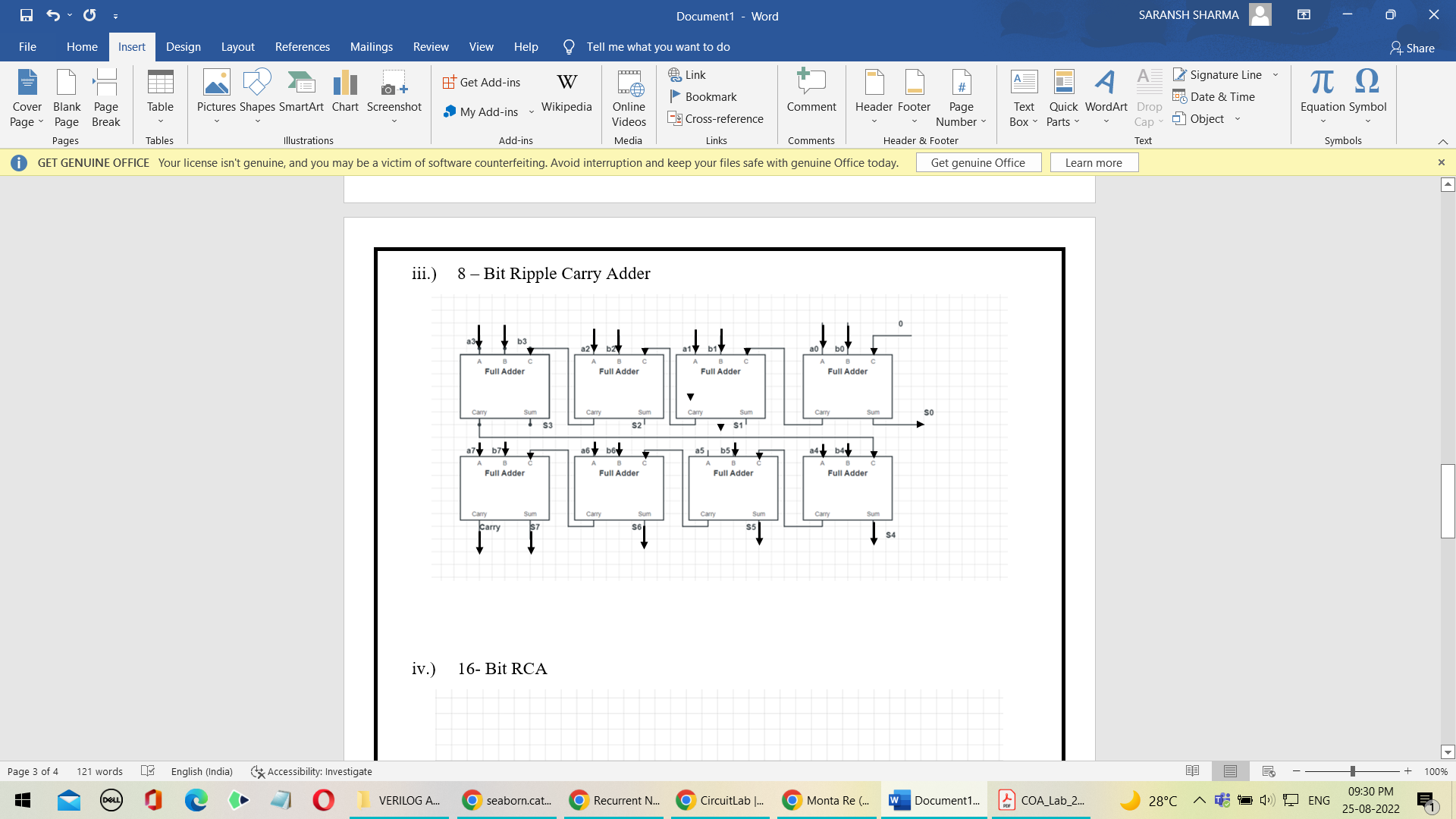
b

a

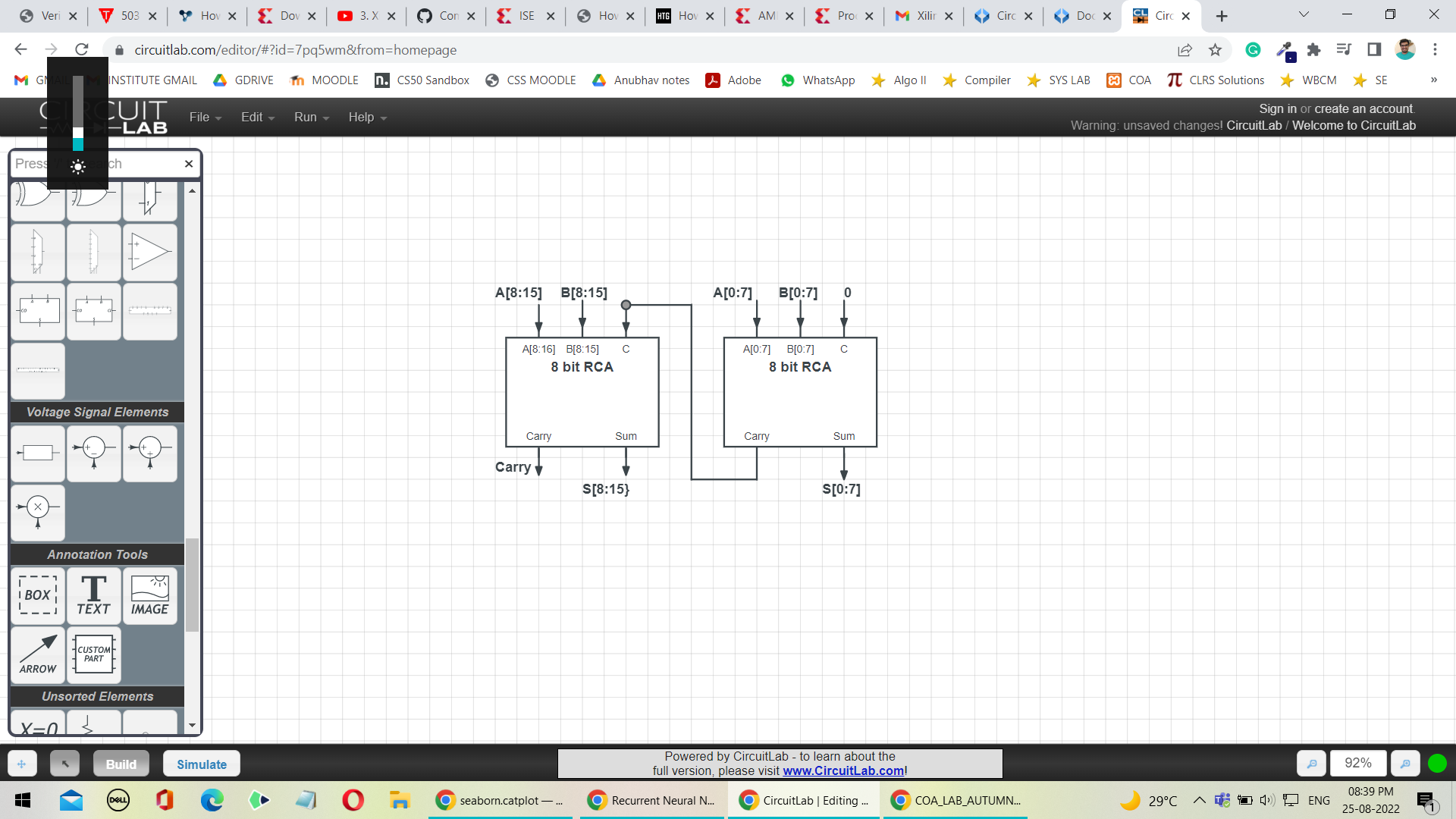
C0

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Truth Table of a Full Adder | | | | |
| Input | | | Output | |
| a | b | c0 | s | C |
| 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 |

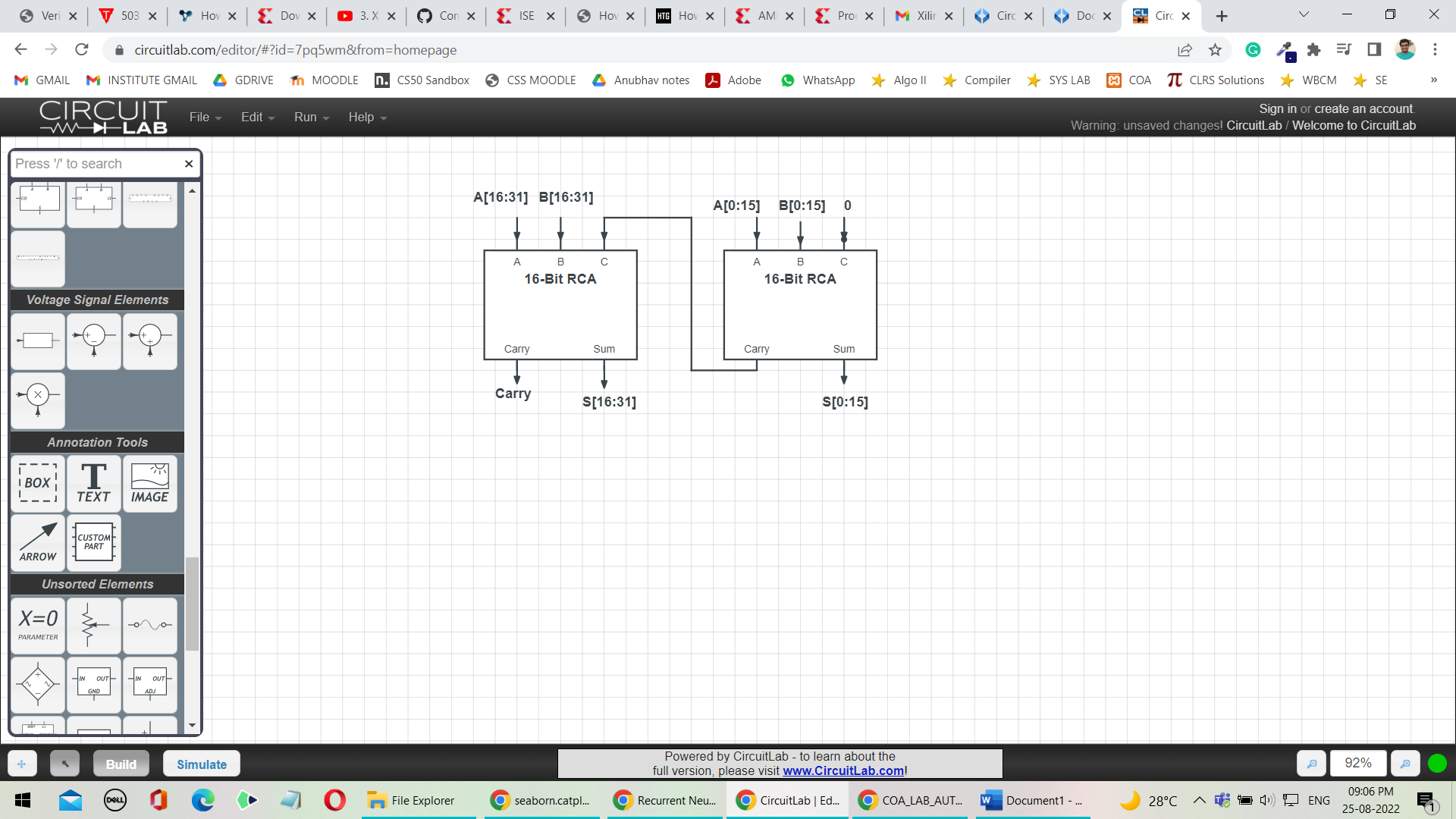
1. 8 – Bit Ripple Carry Adder



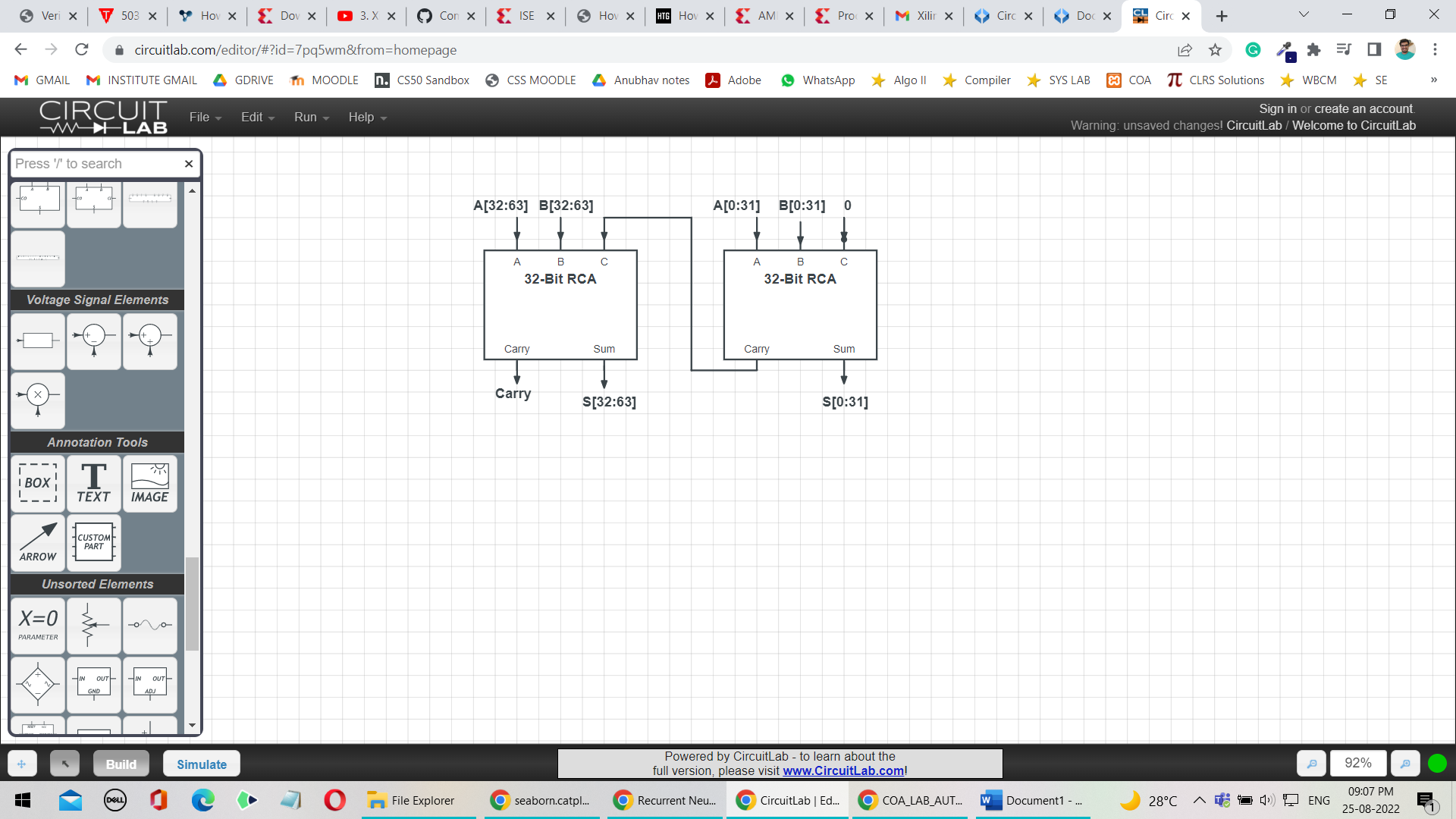
1. 16- Bit RCA



1. 32-Bit RCA



1. 64- Bit RCA



Question: How can you use the above circuit, to compute the difference between two n-bit numbers?

Answer: Ripple Carry adder with input a and b and initial carry bit c0 actually computes the binary sum of a and b, also adding c0 to the LSB of the sum. Thus, c­0 is set to 0 so that RCA adds a and b bitwise and output the result.

For finding difference between two number a and b i.e., a-b we will convert the subtraction problem into addition problem:

i.e., a – b = a + (-b) where (-b) is the 2’s complement of b.

we know that 2’s complement of any number can be calculated by computing 1’s complement of that number and then adding 1 to the LSB of the 1’s complement. That is, (-b) = (~b) + 1 where (~b) is the 1’s complement of b

let a = 100 (4) and b = 010 (2) and n=3

therefore, a – b = 100 – 010

-b = 101 (= ~b) + 1 = 110

a – b = 100 + (110) = 010 (carry\_out = 1)

To calculate 1’s complement of a number, we simply have to flip all the bits of that number. That is, we pass every bit of b to a NOT gate to flip the bits of the b. Therefore, whenever we need to calculate difference between a and b i.e., a-b we would first calculate 1’s compliment of b by taking NOT of every bit of b with 1. Since we also need to add 1 to the 1’s complement calculated, we would pass this 1 as c0 to the RCA. Therefore, upon passing a, (~b) and 1 as input to RCA, the n-bit RCA can calculate the difference between a and b.