COMPUTER ORGANISATION AND ARCHITECTURE

INTRODUCTION TO VERILOG PROGRAMMING

VERILOG ASSIGNMENT-1

GROUP 09

Atishay Jain (20CS30008)

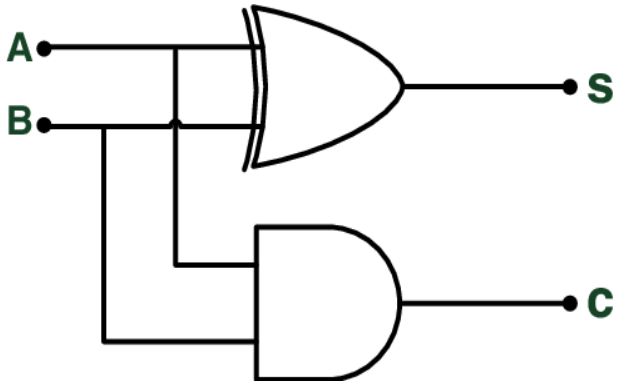
Abothula Suneetha (20CS10004)

Note: In this assignment, we have created one parent module for 64-bit Ripple carry adder. In this module, we have used the other modules of 32-bit RCA. Then in 32-bits RCA, we are using modules of 16-bit RCA. Similarly, in 16-bit RCA, we have used modules of 8-bit RCA. 8-bit RCA uses 8 Full Adder modules and Full Adder uses two Half adder modules.

For checking correct working of any of these modules, we have created separate test-bench for all of these, where we have used several test cases.

1) RIPPLE CARRY ADDERS

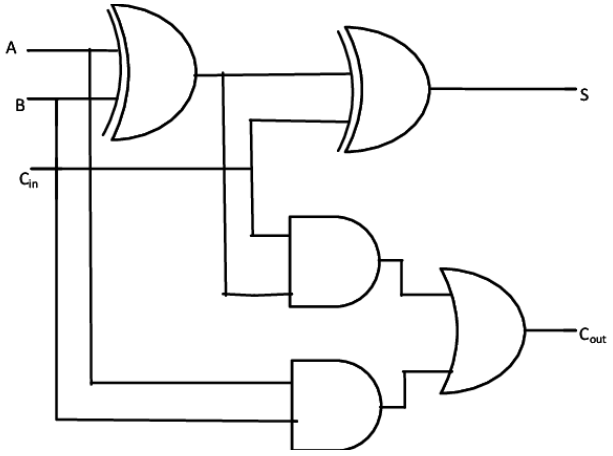
(a) **Half Adder**: A Half Adder is a combinational circuit, which takes in two input bits, a and b, and produces the sum bit, s and the carry-out bit, c.



Truth Table for Half Adder:

a	b	sum(s) = a ^ b	Carry(c) = a & b
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

(b) **Full Adder**: A Full Adder is a combinational circuit, which takes in three input bits, a, b, and in addition a carry-in bit (cin), and produces the sum bit, s and the carry-out bit (cout)



Truth Table for Full Adder:

a	b	carry-in (cin)	sum(s) = a^b^cin	cout = (a&b) | (cin&(a^b))
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

(c) **The longest delays in the circuits**:

8-bit Ripple Carry Adder: 3.471ns (Levels of logic = 6)
16-bit Ripple Carry Adder: 6.167ns (Levels of Logic = 10)
32-bit Ripple Carry Adder: 11.559ns (Levels of Logic = 18)
64-bit Ripple Carry Adder: 22.343ns (Levels of Logic = 34)

(d) How can you use the above circuit, to compute the difference between two n-bit numbers?

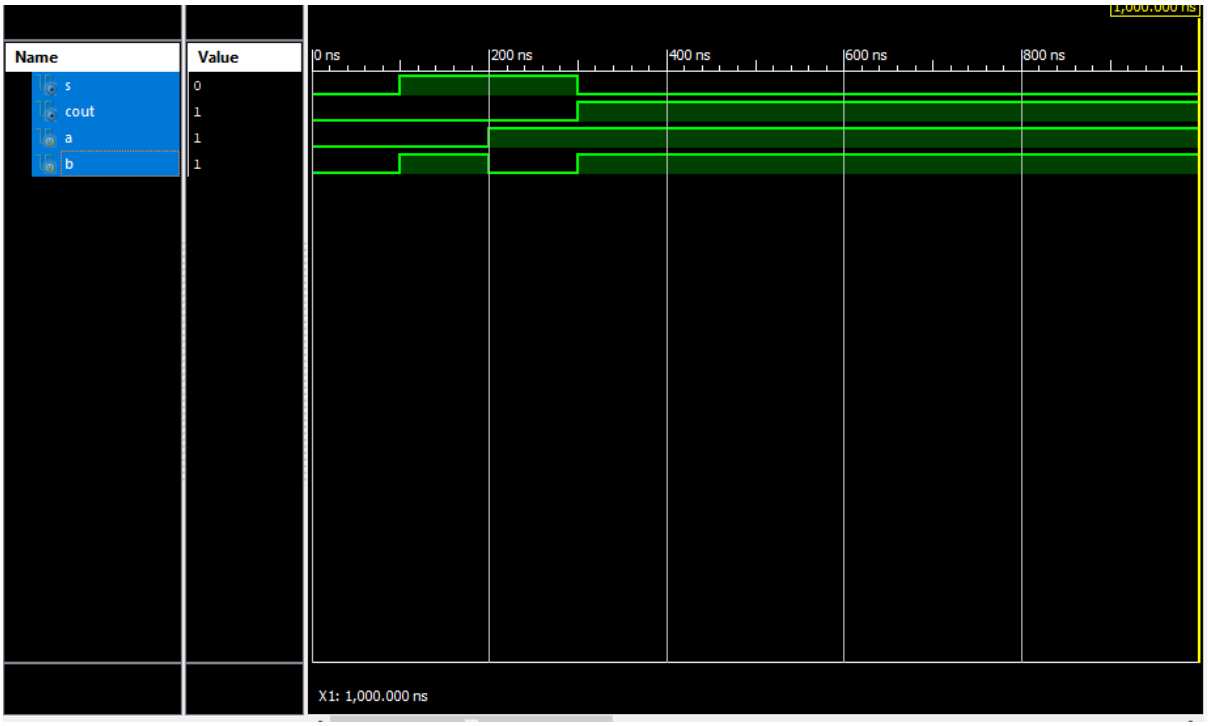
To calculate difference of two numbers say a and b, we can write it as follows.

a – b = a + (-b)
= a + (2's complement of b)
= a + (1's complement of b) + 1

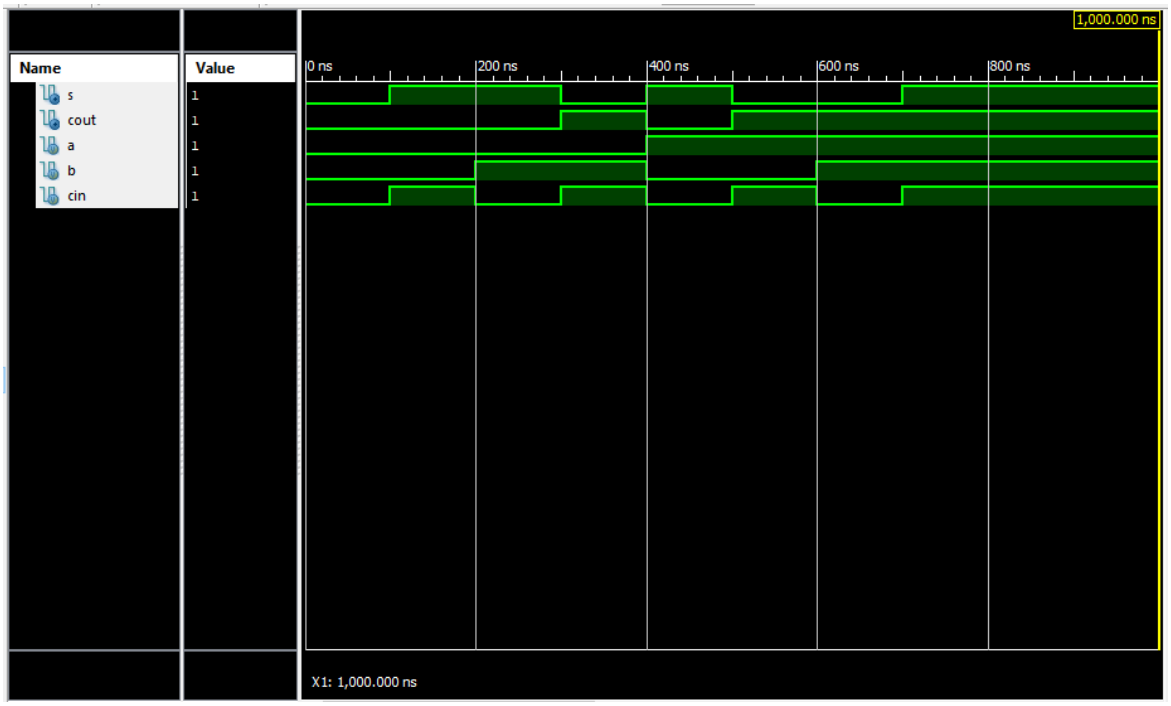
So, it can be written as RCA (a, 1's complement of b, 1) where 1 is the carry in and a, 1's complement of b are inputs. This way, we get a + (1's complement of b) + 1 using an RCA. In the circuit we can put a switch which will be connected to the carry in of the adder and also with XOR gates with each of the input bits of b. When the switch is turned on it will make the carry in 1 and simultaneously flip all the bits of b (XOR with 1 flips the bit value). This way we will get a-b in output. When the switch is off it will not change anything and we will get a+b in output. In order to perform a subtraction of two n bit numbers, we can ripple in a series of n full adders to make an n bit RCA, and use it to perform the addition of a + (1s complement of b) + 1 by setting the input carry to 1 instead of 0 and complementing bits of b using NOT gates. Thus, subtraction can be performed using addition using the 2s complement method. We do not have to perform two additions to add the extra 1 as it can be passed as the carry.

Simulations Screenshots:

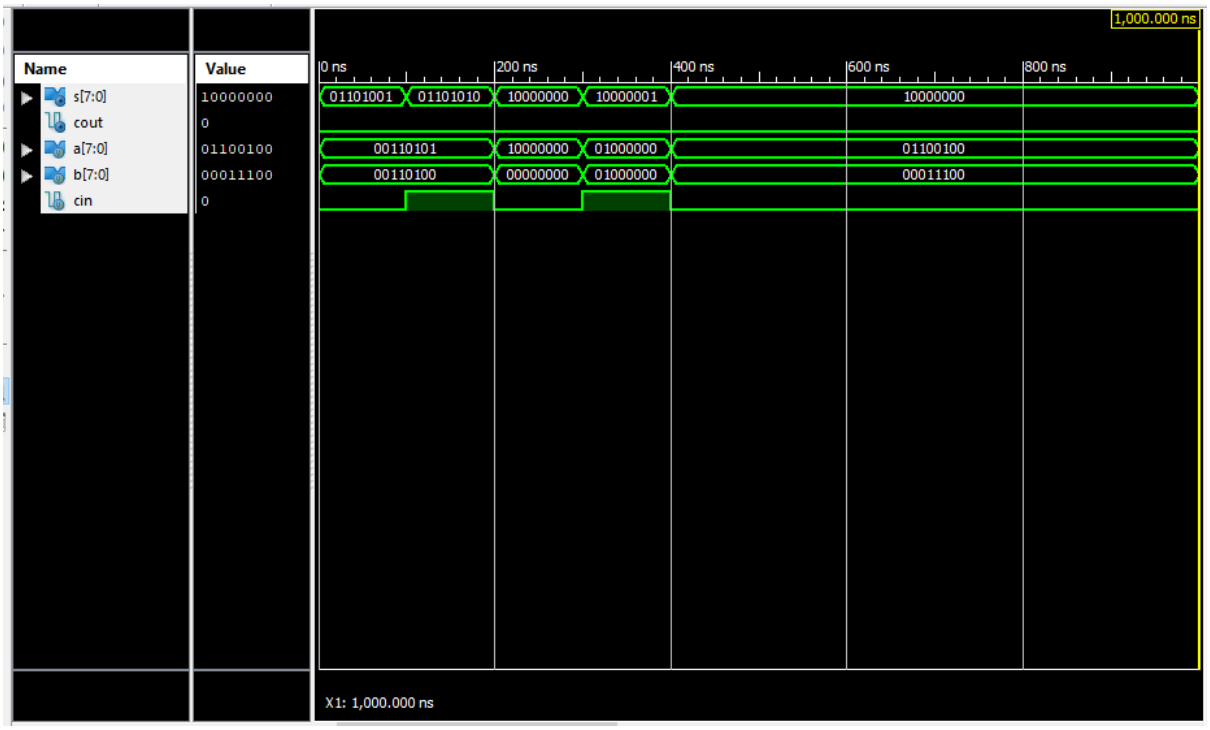
- Half Adder:



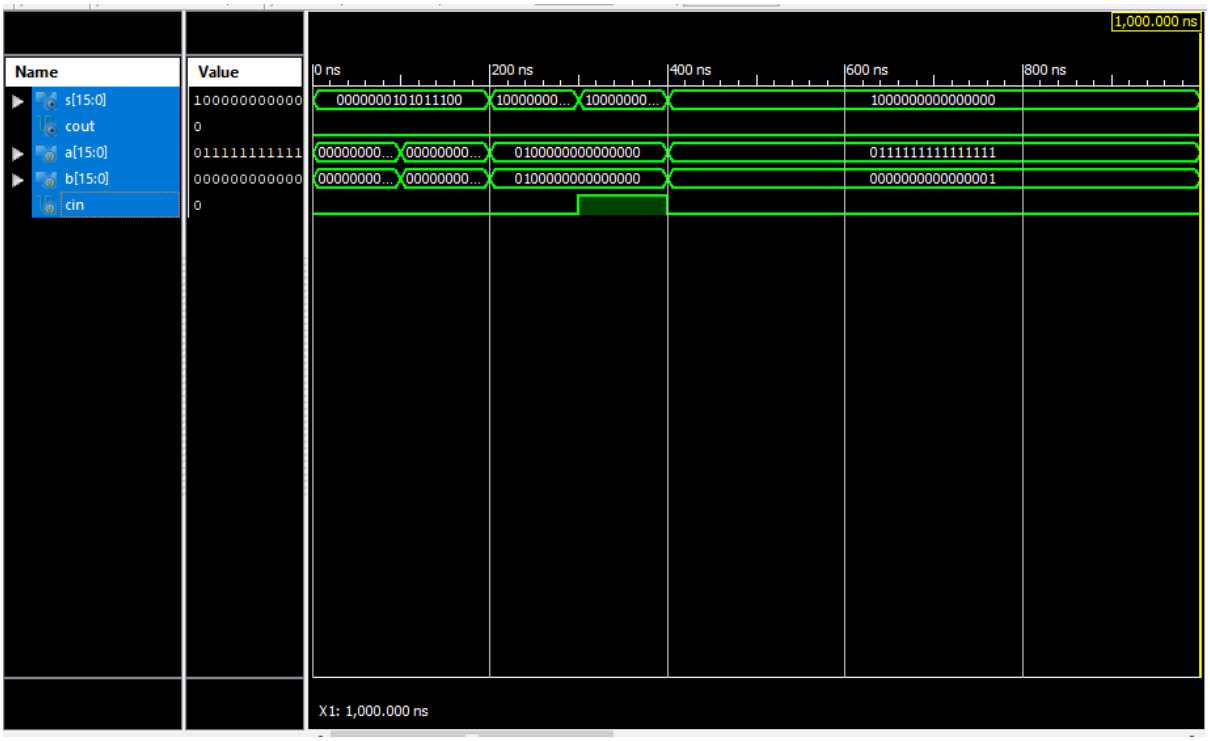
- Full Adder



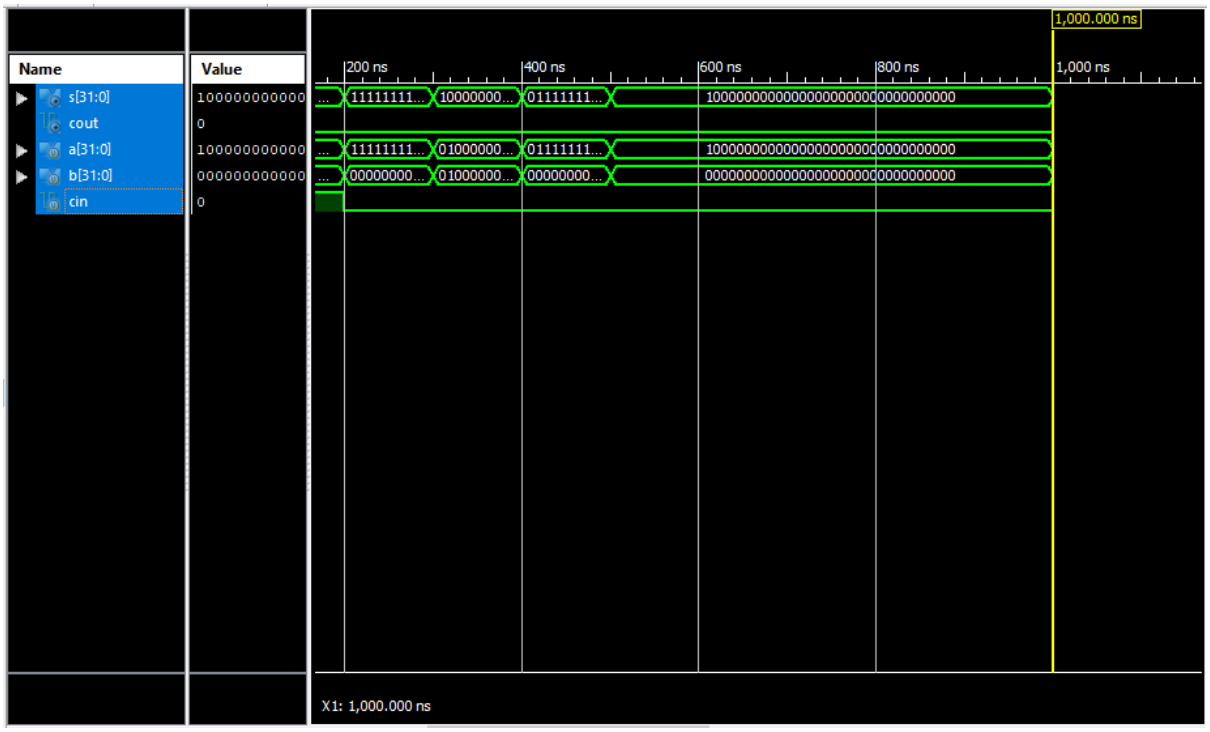
- RCA 8-bit



- RCA 16-bit



- RCA 32-bit



- RCA 64-bit