BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI K. K. BIRLA Goa Campus
First Semester 2020-2021
CS F342 Computer Architecture
Practice Lab, 25th August 2020

INTRODUCTION TO VERILOG

Implement the following design using verilog HDL in Icarcus Verilog simulator and GTKWave tool.

Problem statement

Design a 4-bit two's complement ripple carry adder/subtractor circuit using full adders and test it using given test bench.

Download and add ripple_carry_adder_subtractor.v to your project. Edit the file by adding the relevant code to simulate 4-bit ripple carry adder and subtractor.

Theory

An adder/subtractor is an arithmetic combinational logic circuit which can add/subtract two N-bit binary numbers and output their N-bit binary sum/difference, a carry/borrow status bit, and if needed an overflow status bit. If we choose to represent signed numbers using 2's complement, then we can build an adder/subtractor from a basic adder circuit.

Sum/Difference

In the case of addition, the adder/subtractor would behave exactly as the basic adder circuit. Subtraction, however, is different. Here the number to be subtracted is negated, i.e we make use of the algebraic rule $A-B = A+(-B)$. The negation of a number B in 2's complement is $not(B)+1$. We can achieve this computation by giving the basic adder the numbers A and $not(B)$ then setting the initial carry input to the basic adder to be 1.

Carry/Borrow
When adding, the carry output of the basic adder circuit is used directly. For subtraction, the borrow is the negation of the carry.

Overflow
The detection of an overflow is the same for both addition and subtraction: an overflow has occurred if and only if the basic adder's two most significant carry outputs are different. [today's lab problem is not storing overflow.]

Example
As an example imagine we have four full adders. With these we can easily build a 4-bit ripple carry adder to calculate A+B. We just link up the four half adders, the carry out of one becomes the carry in of another, and set the first carry in to 0.

Now, if we want to calculate A+(-B) then we can use the same half adders if we invert the bits of B and set the first carryin to 1. This would add A and the 2's complement of B giving us our subtraction.

To build our 4-bit adder/subtractor circuit using a 4-bit ripple carry adder we need, along with the two input numbers toadd/subtract (A and B), an input telling us if we are to perform an addition or subtraction operation which we can call Op.

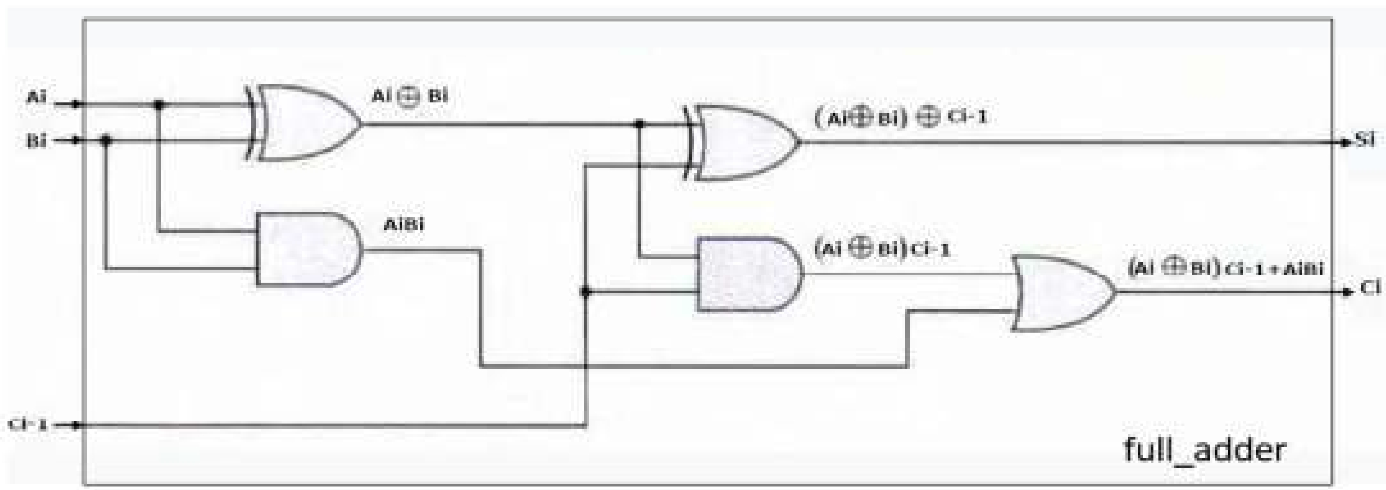
Now, if we look at the definition of the XOR gate we see that B xor 0 = B and B xor 1 = not(B). We can use this in our negation of B if we set Op to 0 when adding and to 1 when subtracting. This in conjunction with setting the initial carry input of the ripple carry adder to Op gives us what we need. The following table shows how this works:

Op	Operand1	Operand2	Initial carry	Result
0	A	B	0	A+B
1	A	Not(B)	1	A-B

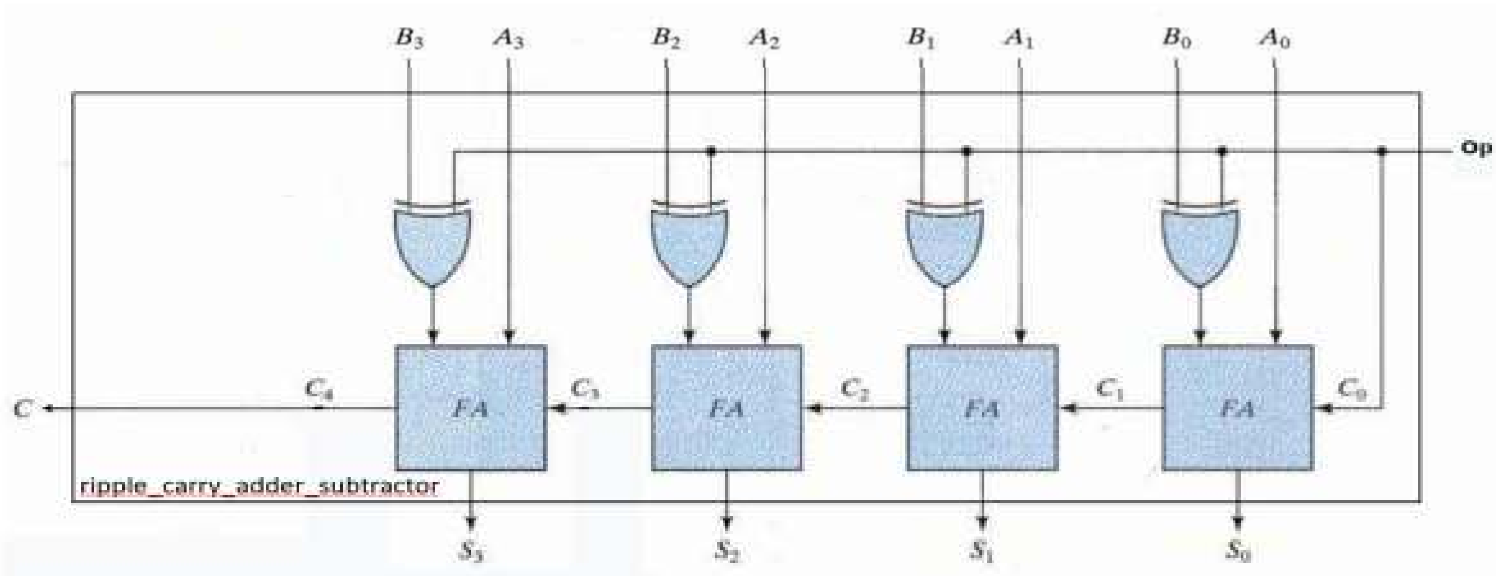
The carry/borrow status bit output can also make use of the XOR gate. If C is the carry output of the ripple carry adder, then we can set the carry/borrow output of the adder/subtractor to be C xor Op.

The overflow status bit output is handled easily, as it is C3 xor C4 when C3 and C4 are the respective carry outputs of the third and fourth half adder respectively. [not included in this lab]

Circuit for Full Adder Module



Circuit for Ripple Carry Adder Subtractor Module



Output Waveform