Lab-3 Q3 Report

Balasubramanian P - 200050103

January 2022

1 Question

Design a 4-bit ripple carry adder-subtractor. ‘a’ and ‘b’ are the two 4-bit unsigned numbers that are to be added. Here ‘cin’ is not an input carry like in the previous question but it is a mode selection bit. If cin=0 then ‘a’ and ‘b’ should be added (a+b). If cin=1 then ‘b’ should be subtracted from ‘a’ (a-b). ‘sum’ is the 4-bit unsigned addition or subtraction output and ‘cout’ is the single-bit output carry. This can be designed only in a structural way using the 4-bit ripple carry adder designed in the previous question (i.e. question 2 of assignment 3) and basic gates. At least one instance of the 4-bit ripple carry adder must be used along with any other basic gates necessary.

```
entity FourbitRipCarAddSub is
    port ( a, b : in std_logic_vector (3 downto 0); cin: in std_logic;
    sum : out std_logic_vector (3 downto 0); cout: out std_logic);
end entity;
```

2 Approach

1. We have a 4-bit adder from the previous question. So we need to think about performing binary subtraction by doing addition.
2. And we know how to do that!! 2’s complement! We take the two’s complement of the number to be subtracted from the other number and then add it to get the final answer.
3. The initial carry ’cin’ determines where the operation to be performed is to add or subtract.

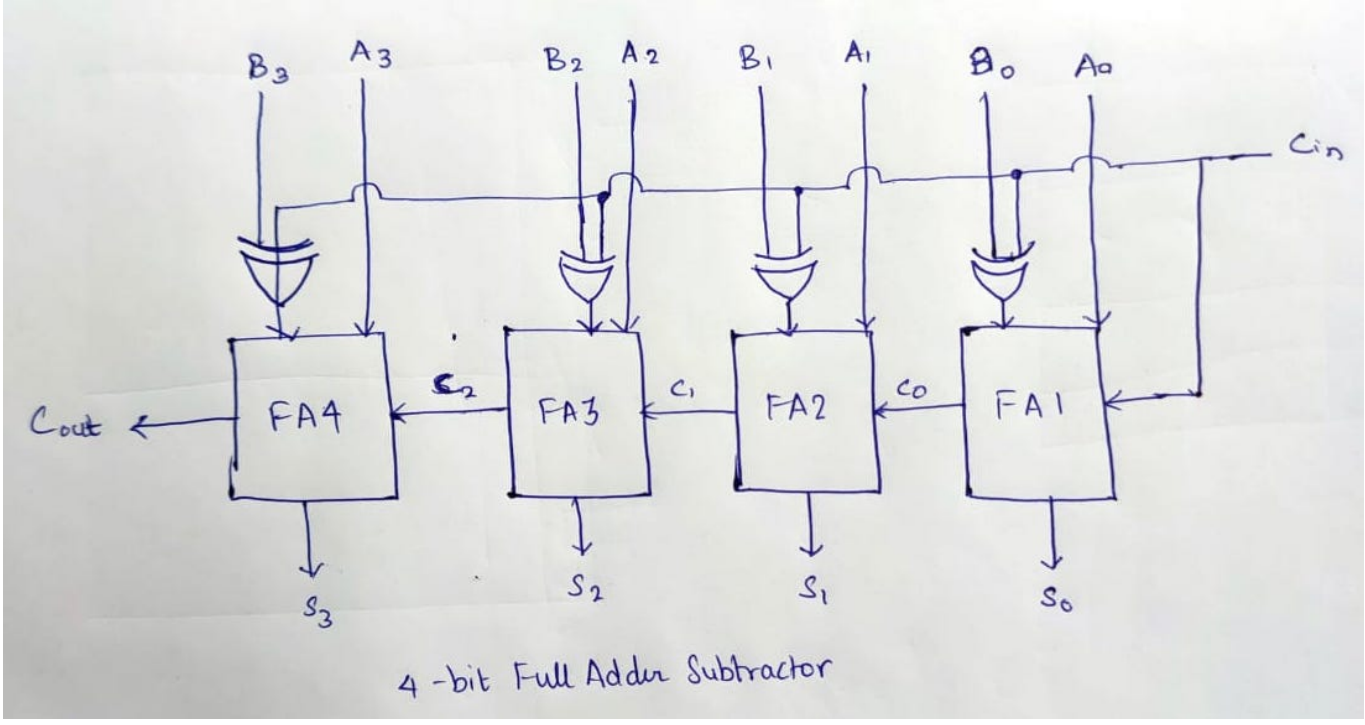

Figure 1: FourbitRipCarAddSub Design

3 Details about Execution

- I used the 4-bit adder built in the previous question as the primary component.
- When the inputs are passed in, the bits of input ’b’ are taken an xor with the input ’cin’.

4 Final Entity

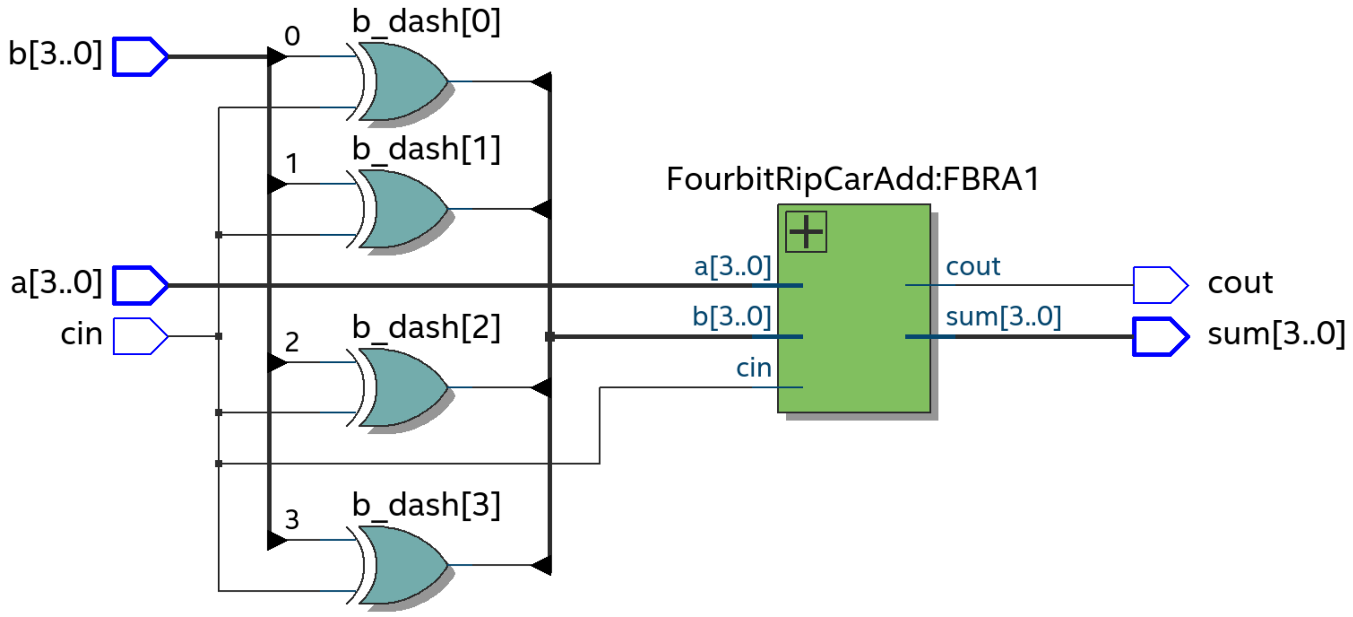

Figure 2: 4-bit Ripple Carry Adder-Subtractor

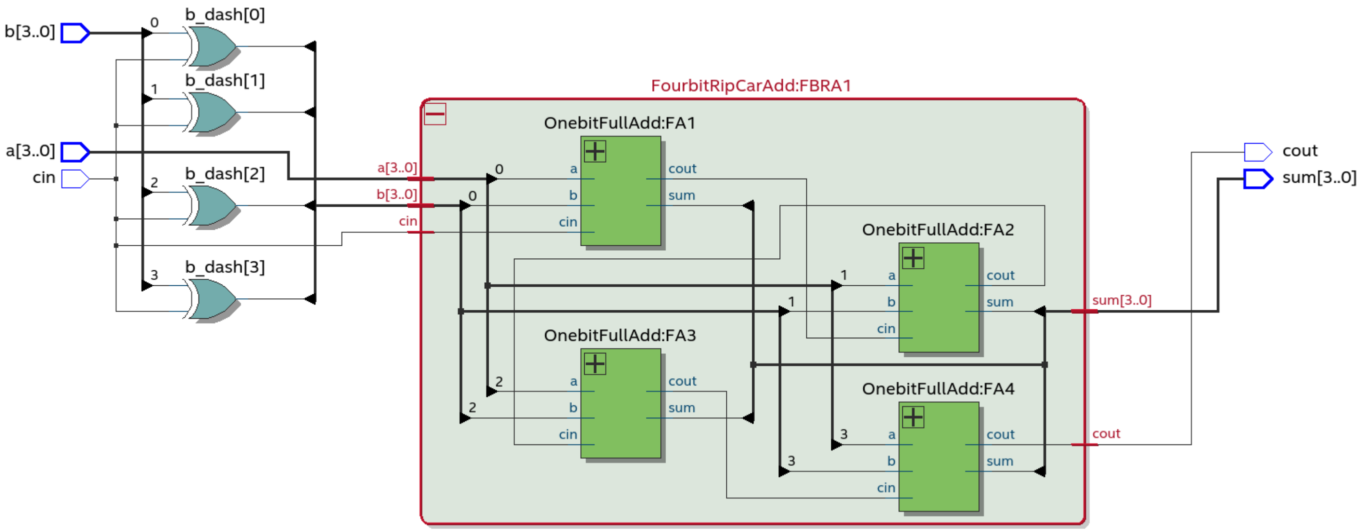

Figure 3: 4-bit Ripple Carry Adder-Subtractor Expanded

5 Waveform Outputs

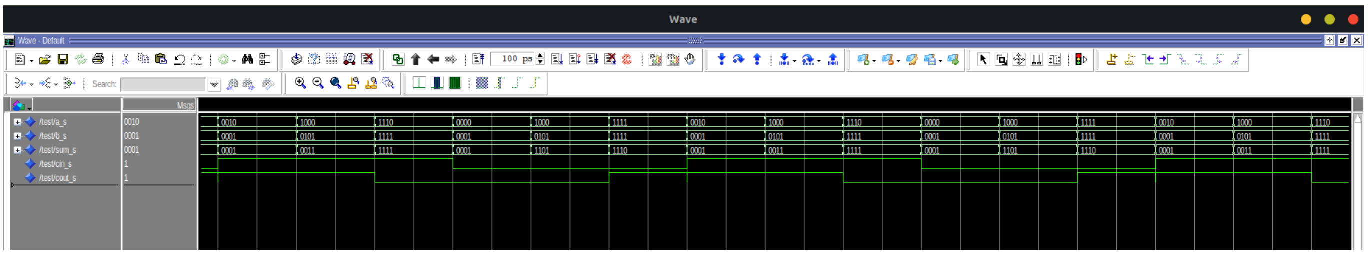

Figure 4: waveform image 1