Lab-3 Q2 Report

Balasubramanian P - 200050103

January 2022

1 Question

Design a 4-bit ripple carry adder. ‘a’ and ‘b’ are the two 4-bit unsigned numbers that are to be added. ‘cin’ is a single-bit input carry that should be added with ‘a’ and ‘b’. ‘sum’ is the 4-bit unsigned addition output and ‘cout’ is the single-bit output carry. This can be designed only in a structural way using only instances of the 1-bit full-adder designed in the previous question (i.e. question 1 of assignment 3.) No other component or gate is permitted.

```
entity FourbitRipCarAdd 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. For adding two 1-bits and a carry we use a full adder, so to add two 4-bit numbers we need four 1-bit full adders.
2. The design of the adder will be logically similar to the addition process we follow where the carry in one column is added to the next column/ bit.
3. The carry output ’cout’ from one full adder us passed on to the next full adder as ’cin’.

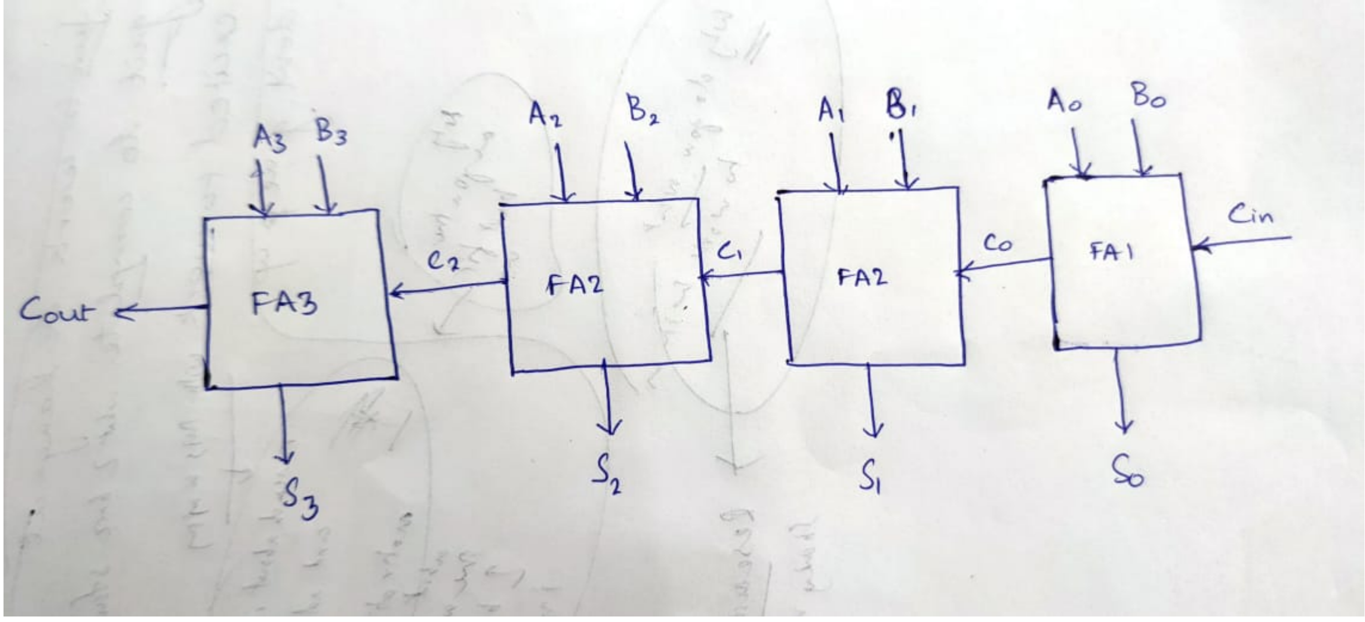


Figure 1: FourbitRipCarAdd Design

3 Details about Execution

- I used the Full Adder built in the previous question as the primary component.
- Used four full adder instances 4-bit adder component. (as shown in Fig 1)

4 Final Design

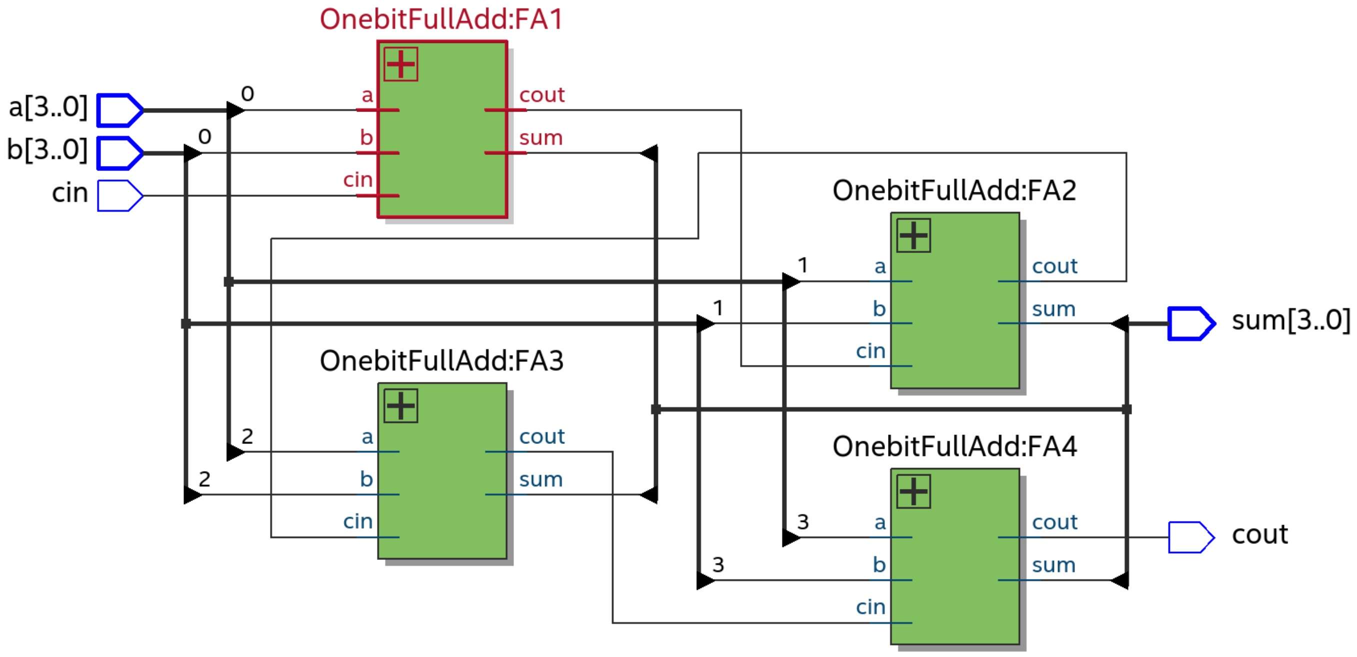


Figure 2: 4-bit Ripple Carry Adder

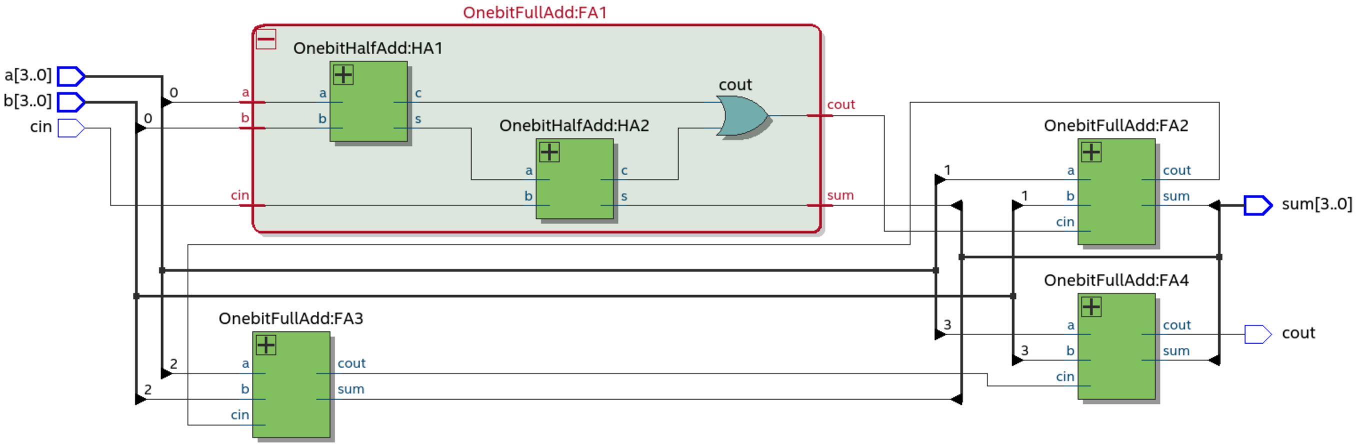
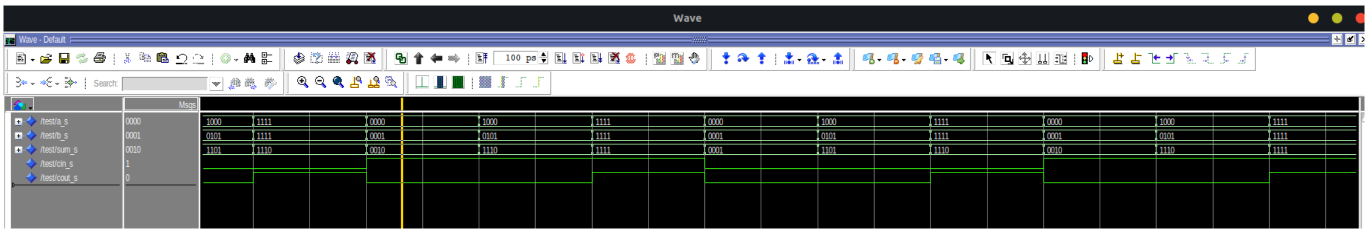


Figure 3: 4-bit Ripple Carry Adder

5 Waveform Outputs



```
/test/a_s     0000  1000 1111 0000 1000 1111 0000 1000 1111 0000 1000 1111
/test/b_s     0001  0101 1111 0001 0101 1111 0001 0101 1111 0001 0101 1111
/test/sum_s   0010  1101 1110 0010 1110 1111 0001 1101 1110 0010 1110 1111
/test/cin_s   1
/test/cout_s  0
```

Figure 4: waveform image 1