

Lab Report 2

Half, Full and 4 Bit Adders

ECE 238L

Kenneth Cox

March 5, 2021

Date Performed: Feb 18, 2021
Instructor: Professor Hamke

1 Items

Half Adder

Half Adder

4 Bit Adder

2 Feedback

The VHDL Videos in the lab assignments folder really helped me refactor this code. It was probably the most useful resource on VHDL I have come across

```

20
21
22 library IEEE;
23 use IEEE.STD_LOGIC_1164.ALL;
24
25
26 entity halfadd is
27     Port ( a,b : in STD_LOGIC;
28           sum,cout : out STD_LOGIC);
29 end halfadd;
30
31 architecture Behavioral of halfadd is
32
33 begin
34
35     sum <= (A and not(B)) or (not(A) and B);
36     cout <= A and B;
37
38
39 end Behavioral;
40

```

Figure 1: source

```

entity lab2_tb is
end lab2_tb;

architecture bench of lab2_tb is

component halfadd
    Port ( a,b : in std_logic;
          sum, cout : out std_logic);
end component;

signal a_tb, b_tb: std_logic;
signal cout_tb, sum_tb : std_logic;

begin

    uut:halfadd port map( a => a_tb,|
                        b => b_tb,
                        sum => sum_tb,
                        cout => cout_tb);

stimulus: process
begin
    A_tb <='0'; B_tb <= '0'; wait for 100 ns;
    A_tb <='0'; B_tb <= '1'; wait for 100 ns;
    A_tb <='1'; B_tb <= '0'; wait for 100 ns;
    A_tb <='1'; B_tb <= '1'; wait for 100 ns;
    wait;
end process;
end bench;

```

Figure 2: test bench

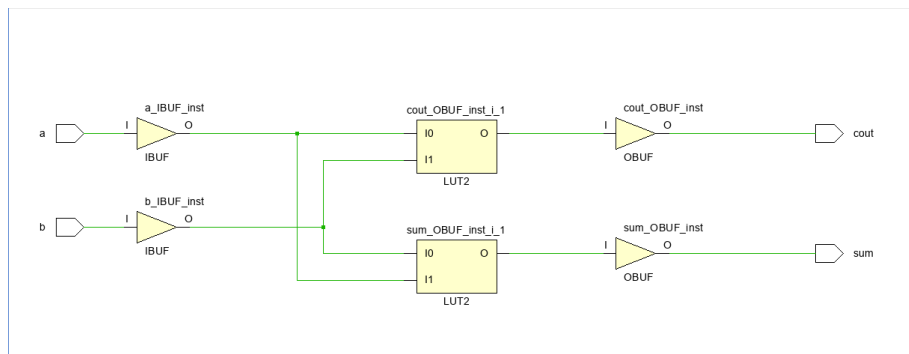


Figure 3: Schematic

Cell Properties

■ cout_OBUF_inst_i_1

I1	I0	O=I0 & I1	
0	0	0	
0	1	0	
1	0	0	
1	1	1	

Figure 4: Truth Table

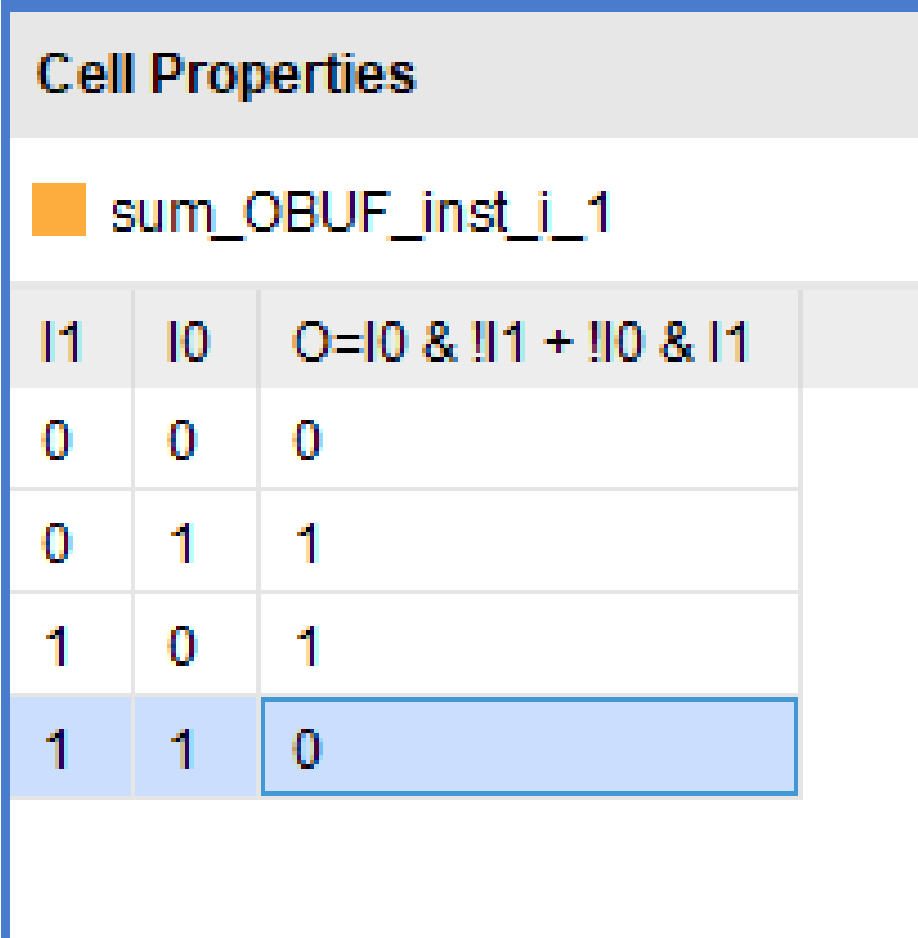


Figure 5: Truth Table cont.

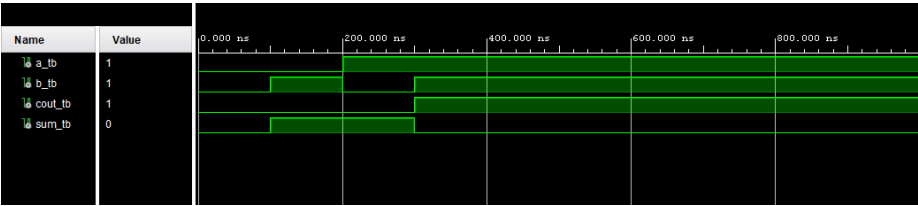


Figure 6: Truth Table

```

9  -----
0
1  library IEEE;
2  use IEEE.STD_LOGIC_1164.ALL;
3
4  entity fulladder is
5      Port ( A : in STD_LOGIC;
6            B : in STD_LOGIC;
7            Cin : in STD_LOGIC;
8            sum : out STD_LOGIC;
9            Cout : out STD_LOGIC);
0  end fulladder;
1
2  architecture behavioral of fulladder is
3
4      begin
5
6          sum <= A XOR B XOR Cin ;
7          Cout <= (A AND B) OR (Cin AND A) OR (Cin AND B) ;
8
9  end behavioral;
0

```

Figure 7: source

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;

-- Uncomment the following library declaration if instantiating
-- any Xilinx leaf cells in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity bench is
-- Port ( );
end bench;

architecture testbench of bench is

component fulladder
    port ( a, b, cin : in std_logic;
          sum, cout : out std_logic);
end component;

signal a_tb, b_tb, cin_tb : std_logic;
signal sum_tb, cout_tb : std_logic;

begin

    uut:fulladder port map( a => a_tb,
                           b => b_tb,
                           cin => cin_tb,
                           sum => sum_tb,
                           cout => cout_tb);

    stimulus: process

    begin
        A_tb <='0'; B_tb <= '0'; Cin_tb <= '0'; wait for 100 ns;
        A_tb <='0'; B_tb <= '0'; Cin_tb <= '1'; wait for 100 ns;
        A_tb <='0'; B_tb <= '1'; Cin_tb <= '0'; wait for 100 ns;
        A_tb <='0'; B_tb <= '1'; Cin_tb <= '1'; wait for 100 ns;
        A_tb <='1'; B_tb <= '0'; Cin_tb <= '0'; wait for 100 ns;
        A_tb <='1'; B_tb <= '0'; Cin_tb <= '1'; wait for 100 ns;
        A_tb <='1'; B_tb <= '1'; Cin_tb <= '0'; wait for 100 ns;
        A_tb <='1'; B_tb <= '1'; Cin_tb <= '1'; wait for 100 ns;
        wait;
    end process;
end testbench;

```

Figure 8: test bench

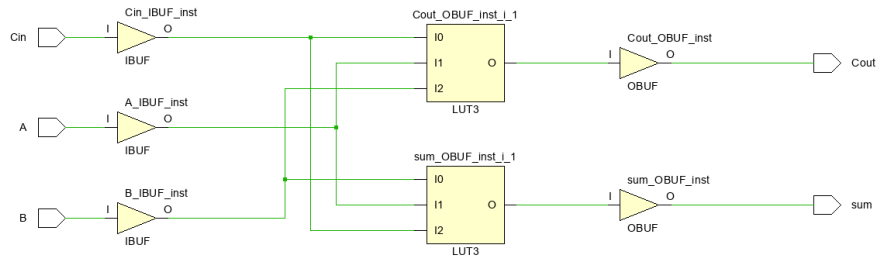


Figure 9: Schematic

Cell Properties			
sum_OBUF_inst_i_1			
I2	I1	I0	O=I0 & !I1 & !I2 + !I0 & I1 & !I2 + !I0 & !I1 & I2 + I0 & I1 & I2
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Figure 10: Truth Table

Cell Properties				
Cout_OBUF_inst_i_1				
I2	I1	I0	O=I0 & I1 + I0 & I2 + I1 & I2	
0	0	0	0	
0	0	1	0	
0	1	0	0	
0	1	1	1	
1	0	0	0	
1	0	1	1	
1	1	0	1	
1	1	1	1	

Figure 11: Truth Table cont.

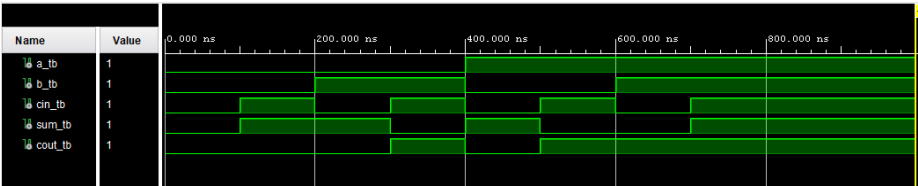


Figure 12: Truth Table

```

-----
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity fourbitadder is
Port ( a : in STD_LOGIC_VECTOR (3 downto 0);
      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 fourbitadder;

architecture Behavioral of fourbitadder is

component fulladder
Port ( a : in STD_LOGIC;
      b : in STD_LOGIC;
      cin : in STD_LOGIC;
      sum : out STD_LOGIC;
      cout : out STD_LOGIC);
end component;

signal carry1,carry2,carry3: STD_LOGIC;

begin

fa0: fulladder port map( a(0), b(0), cin, sum(0), carry1);
fa1: fulladder port map( a(1), b(1), carry1, sum(1), carry2);
fa2: fulladder port map( a(2), b(2), carry2, sum(2), carry3);
fa3: fulladder port map( a(3), b(3), carry3, sum(3), Cout);

end Behavioral;

```

Figure 13: source

```

1  LIBRARY ieee;
2  USE ieee.std_logic_1164.ALL;
3
4  ENTITY Tb_4_Adder IS
5  END Tb_4_Adder;
6
7  ARCHITECTURE behavior OF Tb_4_Adder IS
8
9
10 COMPONENT fourbitadder
11 PORT(
12   a : IN std_logic_vector(3 downto 0);
13   b : IN std_logic_vector(3 downto 0);
14   Cin : IN std_logic;
15   sum : OUT std_logic_vector(3 downto 0);
16   Cout : OUT std_logic
17 );
18 END COMPONENT;
19
20 signal a : std_logic_vector(3 downto 0) := (others => '0');
21 signal b : std_logic_vector(3 downto 0) := (others => '0');
22 signal Cin : std_logic := '0';
23
24 signal sum : std_logic_vector(3 downto 0);
25 signal Cout : std_logic;
26
27 BEGIN
28
29 uut: fourbitadder PORT MAP (
30   A => a,
31   B => b,
32   Cin => Cin,
33   sum => sum,
34   Cout => Cout);
35
36 stim_proc: process
37   begin
38     a <= "0110"; b <= "1100"; wait for 100 ns;
39     a <= "1111"; b <= "1100"; wait for 100 ns;
40     a <= "0110"; b <= "0111"; wait for 100 ns;
41     a <= "0110"; b <= "1110"; wait for 100 ns;
42     a <= "1111"; b <= "1111"; wait for 100 ns;
43
44     wait;
45
46   end process;
47
48 END;
49

```

Figure 14: test bench

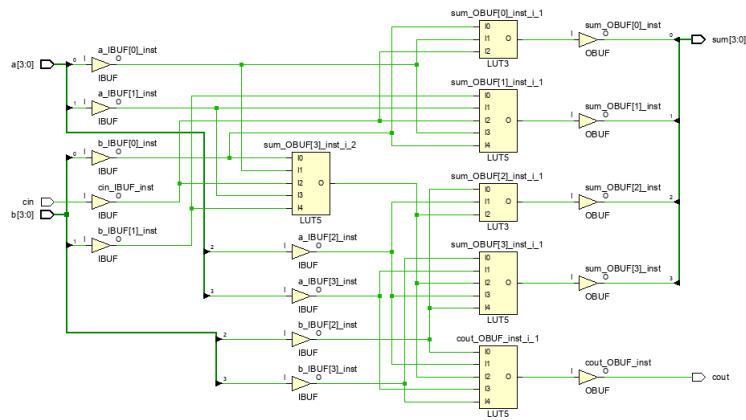


Figure 15: Schematic

Cell Properties						
■ cout_OBUF_inst_1						
I4	I3	I2	I1	I0	O= $I0 \& I1 \& I3 + I1 \& I3 \& I4 + I1 \& I2 \& I3 + I0 \& I1 \& I3 \& I4 + I0 \& I2 \& I3 + I0 \& I2 \& I3 \& I4 + I1 \& I2 \& I3 \& I4 + I1 \& I2 \& I3 \& I4$	
0	0	0	0	0	0	0
0	0	0	0	1	0	
0	0	0	1	0	0	
0	0	0	1	1	0	
0	0	1	0	0	0	
0	0	1	0	1	0	
0	0	1	1	0	0	
0	0	1	1	1	0	
0	1	0	0	0	0	
0	1	0	0	1	0	
0	1	0	1	0	0	
0	1	0	1	1	1	
0	1	1	0	0	0	
0	1	1	0	1	1	
0	1	1	1	0	1	
0	1	1	1	1	1	
1	0	0	0	0	0	
1	0	0	0	1	0	
1	0	0	1	0	0	
1	0	0	1	1	1	
1	0	1	0	0	0	
1	0	1	0	1	1	
1	0	1	1	0	1	
1	0	1	1	1	1	
1	1	0	0	0	1	
1	1	0	0	1	1	
1	1	0	1	0	1	
1	1	0	1	1	1	
1	1	1	0	0	1	
1	1	1	0	1	1	
1	1	1	1	0	1	
1	1	1	1	1	1	

Figure 16: Truth Table

Cell Properties			
sum_OBUF[0]_inst_i_1			
I2	I1	I0	O=I0 & !I1 & !I2 + !I0 & I1 & !I2 + !I0 & !I1 & I2 + I0 & I1 & I2
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Figure 21: Truth Table

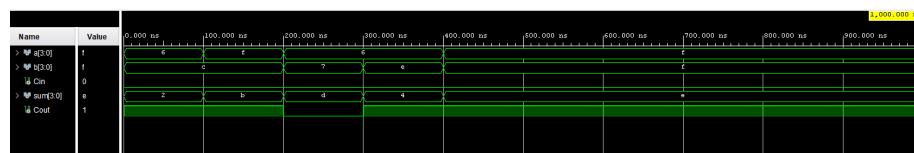


Figure 22: Truth Table