Lab 3 – Ripple Carry Adder

Name : Chirath A.A.I

Index No : 210086E

- Lab Tasks :

The results of this lab were designing a Half Adder, utilizing the Half Adder to develop a Full Adder, and then using the Full Adder to design the 4-bit Ripple Carry Adder. Finally, we must use the BAYSIS 3 development board to simulate and test the RCA's performance.

- Truth Tables and Boolean Expressions:

1). Half Adder

A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Sum(S) → $A \oplus B$

Carry(C) → $A \cdot B$

2). Full Adder

A	B	C_{in}	S	C_{out}
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

Sum(S) → $A'B'C_{in} + A'BC_{in}' + AB'C_{in}' + ABC_{in}$

$= (A'B' + AB)\, C_{in} + (A'B + AB')\, C_{in}'$

$= (A \oplus B)'\, Cin + (A \oplus B)\, Cin'$

$= (A \oplus B) \oplus Cin$

$= A \oplus B \oplus Cin$

Carry Out(C_{out}) → $A'BC_{in} + AB'C_{in} + ABC_{in}' + ABC_{in}$

$= (B'C_{in} + BC_{in}')A + BC_{in}(A' + A)$

$= A(B \oplus C_{in}) + BC_{in}(1)$

$= A(B \oplus C_{in}) + BC_{in}$

VHDL Files: -

1). Half Adder

```
19  ----------------------------------------------------------------------------------
20
21
22  library IEEE;
23  use IEEE.STD_LOGIC_1164.ALL;
24
25  -- Uncomment the following library declaration if using
26  -- arithmetic functions with Signed or Unsigned values
27  --use IEEE.NUMERIC_STD.ALL;
28
29  -- Uncomment the following library declaration if instantiating
30  -- any Xilinx leaf cells in this code.
31  --library UNISIM;
32  --use UNISIM.VComponents.all;
33
34  entity HA is
35      Port ( A : in STD_LOGIC;
36             B : in STD_LOGIC;
37             S : out STD_LOGIC;
38             C : out STD_LOGIC);
39  end HA;
40
41  architecture Behavioral of HA is
42
43  begin
44  S <= A XOR B;
45  C <= A AND B;
46  end Behavioral;
47
```

2). Full Adder

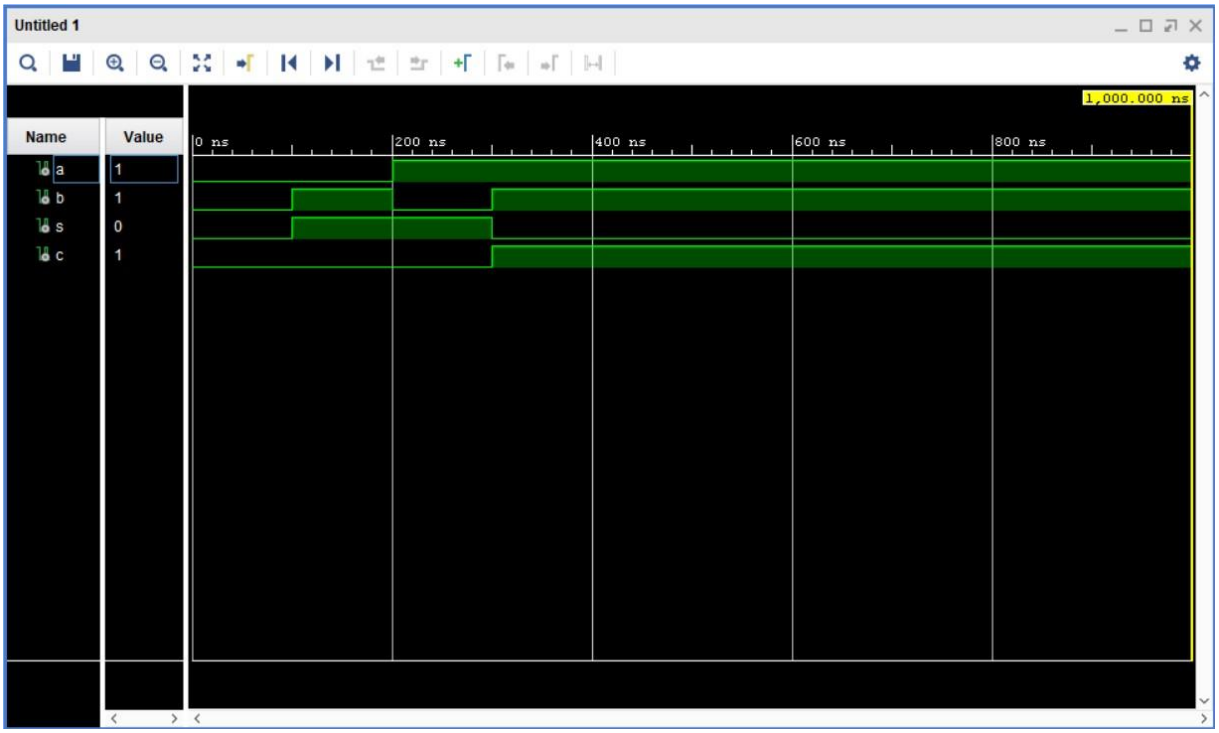
```
21
22  library IEEE;
23  use IEEE.STD_LOGIC_1164.ALL;
24
25  -- Uncomment the following library declaration if using
26  -- arithmetic functions with Signed or Unsigned values
27  --use IEEE.NUMERIC_STD.ALL;
28
29  -- Uncomment the following library declaration if instantiating
30  -- any Xilinx leaf cells in this code.
31  --library UNISIM;
32  --use UNISIM.VComponents.all;
33
34  entity FA is
35      Port ( A : in STD_LOGIC;
36             B : in STD_LOGIC;
37             C_in : in STD_LOGIC;
38             S : out STD_LOGIC;
39             C_out : out STD_LOGIC);
40  end FA;
41
42  architecture Behavioral of FA is
43  component HA
44   port (
45   A: in std_logic;
46   B: in std_logic;
47   S: out std_logic;
48   C: out std_logic);
49  end component;
50
51  SIGNAL HA0_S, HA0_C, HA1_S, HA1_C : std_logic;
52
53  begin
54  HA_0 : HA
55   port map (
56   A => A,
57   B => B,
58   S => HA0_S,
59   C => HA0_C
60  );

61  HA_1 : HA
62   port map (
63   A => HA0_S,
64   B => C_in,
65   S => HA1_S,
66   C => HA1_C
67  );
68  S <= HA1_S;
69  C_out <= HA0_C OR HA1_C;
70
71  end Behavioral;
```

3). 4-Bit RCA

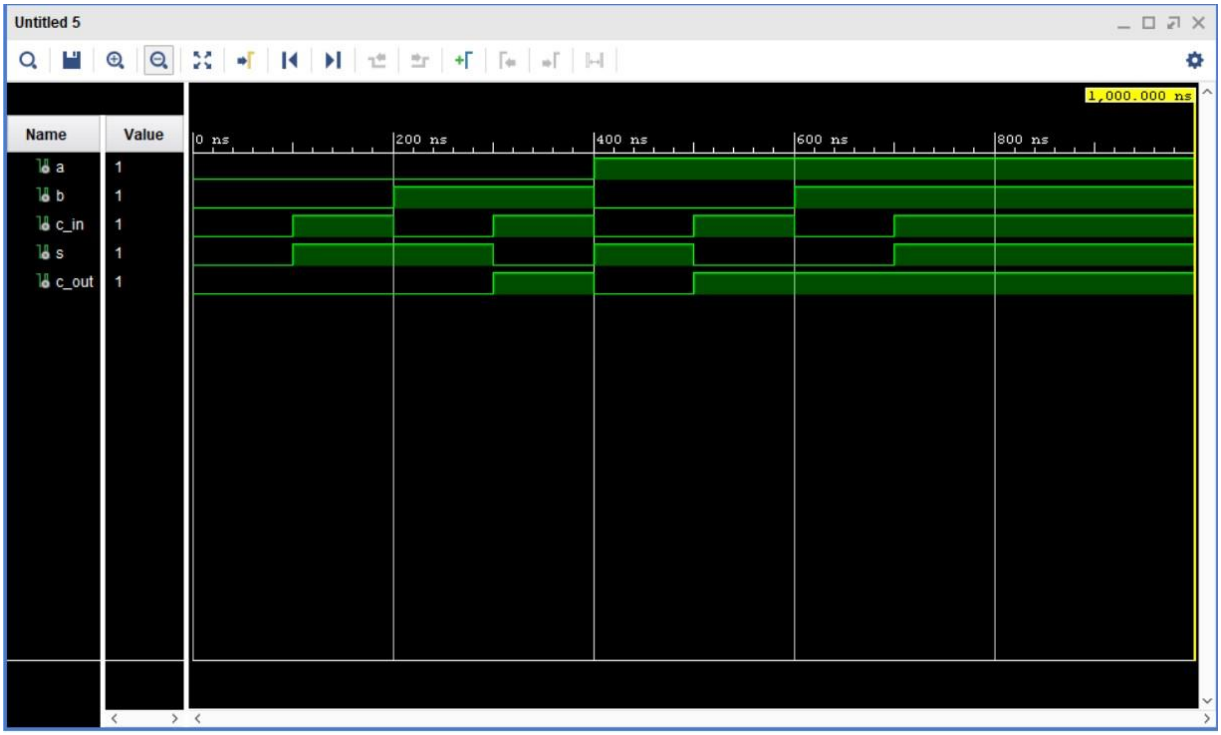
```
21
22  library IEEE;
23  use IEEE.STD_LOGIC_1164.ALL;
24
25  -- Uncomment the following library declaration if using
26  -- arithmetic functions with Signed or Unsigned values
27  --use IEEE.NUMERIC_STD.ALL;
28
29  -- Uncomment the following library declaration if instantiating
30  -- any Xilinx leaf cells in this code.
31  --library UNISIM;
32  --use UNISIM.VComponents.all;
33
34  entity RCA_4 is
35      Port ( A0 : in STD_LOGIC;
36             A1 : in STD_LOGIC;
37             A2 : in STD_LOGIC;
38             A3 : in STD_LOGIC;
39             B0 : in STD_LOGIC;
40             B1 : in STD_LOGIC;
41             B2 : in STD_LOGIC;
42             B3 : in STD_LOGIC;
43             C_in : in STD_LOGIC;
44             S0 : out STD_LOGIC;
45             S1 : out STD_LOGIC;
46             S2 : out STD_LOGIC;
47             S3 : out STD_LOGIC;
48             C_out : out STD_LOGIC);
49  end RCA_4;
50
51  architecture Behavioral of RCA_4 is
52  component FA
53  port (
54      A: in std_logic;
55      B: in std_logic;
56      C_in: in std_logic;
57      S: out std_logic;
58      C_out: out std_logic);
59  end component;

60
61  SIGNAL FA0_S, FA0_C, FA1_S, FA1_C, FA2_S, FA2_C, FA3_S, FA3_C  : std_logic;
62
63  begin
64  FA_0 : FA port map (
65       A => A0,
66       B => B0,
67       C_in => '0', -- Set to ground
68       S => S0,
69       C_Out => FA0_C
70  );
71  FA_1 : FA port map (
72       A => A1,
73       B => B1,
74       C_in => FA0_C,
75       S => S1,
76       C_Out => FA1_C
77  );
78  FA_2 : FA port map (
79       A => A2,
80       B => B2,
81       C_in => FA1_C,
82       S => S2,
83       C_Out => FA2_C
84  );
85  FA_3 : FA port map (
86       A => A3,
87       B => B3,
88       C_in => FA2_C,
89       S => S3,
90       C_Out => C_out
91  );
92  end Behavioral;
```

Timing Diagrams : -

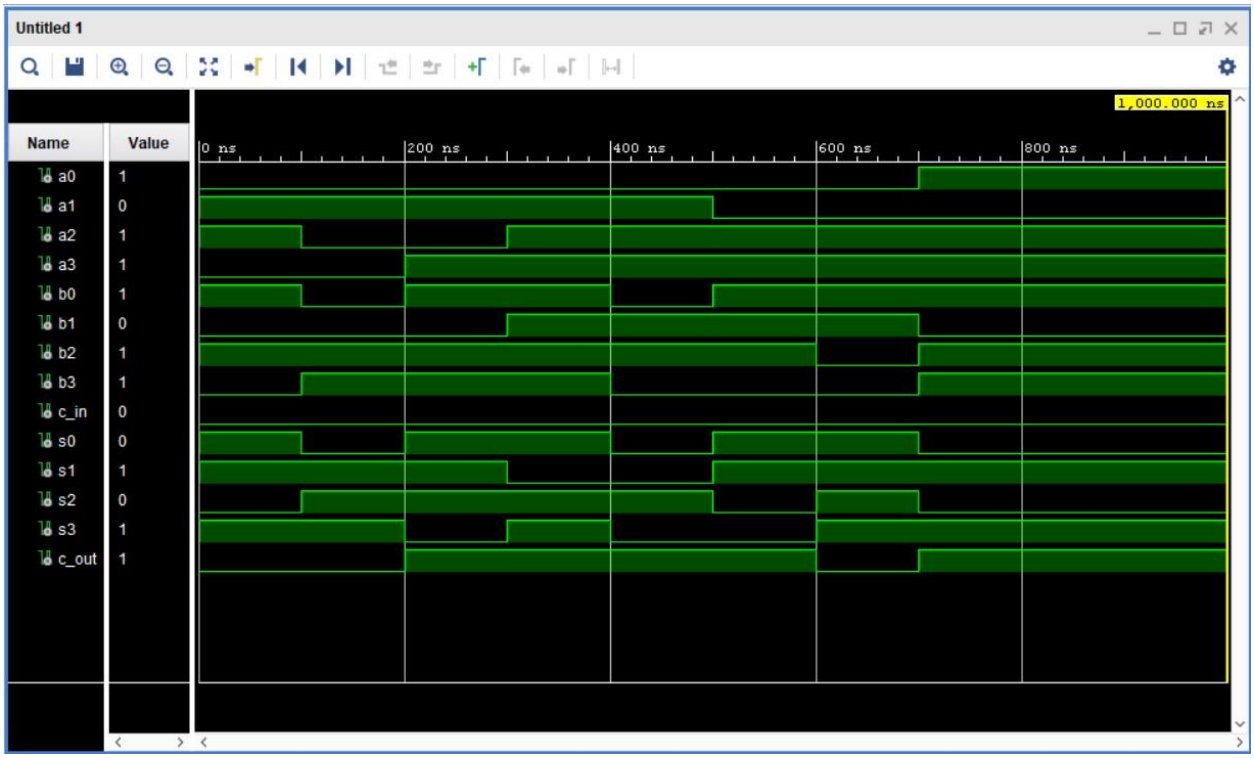
1). Half Adder



2). Full Adder



3). 4-Bit RCA


Untitled 1
Name
Value
a0 1
a1 0
a2 1
a3 1
b0 1
b1 0
b2 1
b3 1
c_in 0
s0 0
s1 1
s2 0
s3 1
c_out 1
0 ns
200 ns
400 ns
600 ns
800 ns
1,000.000 ns

Test Bench Files: -

1). Half Adder

```
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;

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

architecture Behavioral of TB_HA is
COMPONENT HA
    PORT( A, B : IN STD_LOGIC;
          S, C : OUT STD_LOGIC);
END COMPONENT;

signal a, b, s, c : std_logic;

begin
UTT: HA PORT MAP(
    A => a,
    B => b,
    S => s,
    C => c
);
process
 begin
 a <= '0'; -- set initial values
 b <= '0';
 WAIT FOR 100 ns; -- after 100 ns change inputs
 b <= '1';
 WAIT FOR 100 ns; --change again
 a <= '1';
 b <= '0';
 WAIT FOR 100 ns; --change again
 b <= '1';
 WAIT;
end process;
end Behavioral;
```

2). Full Adder

```
21
22  library IEEE;
23  use IEEE.STD_LOGIC_1164.ALL;
24
25  -- Uncomment the following library declaration if using
26  -- arithmetic functions with Signed or Unsigned values
27  --use IEEE.NUMERIC_STD.ALL;
28
29  -- Uncomment the following library declaration if instantiating
30  -- any Xilinx leaf cells in this code.
31  --library UNISIM;
32  --use UNISIM.VComponents.all;
33
34  entity TB_FA is
35  --  Port ( );
36  end TB_FA;
37
38  architecture Behavioral of TB_FA is
39  COMPONENT FA
40      PORT( A, B, C_in : IN STD_LOGIC;
41            S, C_out   : OUT STD_LOGIC);
42  END COMPONENT;
43
44  signal a, b, c_in, s, c_out : std_logic;
45
46  begin
47  UTT: FA PORT MAP(
48      A => a,
49      B => b,
50      C_in => c_in,
51      S => s,
52      C_out => c_out
53  );
54
55  process
56  begin
57   a <= '0'; -- set initial values
58   b <= '0';
59   c_in <= '0';
60   WAIT FOR 100 ns; -- after 100 ns change inputs
61   c_in <= '1';
62   WAIT FOR 100 ns; --change again
63   b <= '1';
64   c_in <= '0';
65   WAIT FOR 100 ns; --change again
66   c_in <= '1';
67   WAIT FOR 100 ns; --change again
68   a <= '1';
69   b <= '0';
70   c_in <= '0';
71   WAIT FOR 100 ns; --change again
72   c_in <= '1';
73   WAIT FOR 100 ns; --change again
74   b <= '1';
75   c_in <= '0';
76   WAIT FOR 100 ns; --change again
77   c_in <= '1';
78   WAIT;
79  end process;
80
81  end Behavioral;
```

3). 4 – Bit RCA

```
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 TB_4_RCA is
--  Port ( );
end TB_4_RCA;

architecture Behavioral of TB_4_RCA is
COMPONENT RCA_4
    PORT( A0, A1, A2, A3, B0, B1, B2, B3, C_in : IN STD_LOGIC;
          S0, S1, S2, S3, C_out                : OUT STD_LOGIC);
END COMPONENT;

signal a0, a1, a2, a3, b0, b1, b2, b3, c_in, s0, s1, s2, s3, c_out : std_logic;

begin
UTT: RCA_4 PORT MAP(
    A0 => a0,
    A1 => a1,
    A2 => a2,
    A3 => a3,
    B0 => b0,
    B1 => b1,
    B2 => b2,
    B3 => b3,
    C_in => c_in,
    S0 => s0,
    S1 => s1,
    S2 => s2,
    S3 => s3,
    C_out => c_out
);

 process
 begin
  a0 <= '0';
  a1 <= '1';
  a2 <= '1';
  a3 <= '0'; -- set initial values
  b0 <= '1';
  b1 <= '0';
  b2 <= '1';
  b3 <= '0';
  c_in <= '0';
  WAIT FOR 100 ns; -- after 100 ns change inputs
  a2 <= '0';
  b0 <= '0';
  b3 <= '1';
  WAIT FOR 100 ns; --change again
  a3 <= '1';
  b0 <= '1';
  WAIT FOR 100 ns; --change again
  a2 <= '1';
  b1 <= '1';
  WAIT FOR 100 ns; --change again
  b0 <= '0';
  b3 <= '0';
  WAIT FOR 100 ns; --change again
  b0 <= '1';
  a1 <= '0';
  WAIT FOR 100 ns; --change again
  b2 <= '0';
  WAIT FOR 100 ns; --change again
  a0 <= '1';
  b1 <= '0';
  b2 <= '1';
  b3 <= '1';
  WAIT;
 end process;

 end Behavioral;
```

Discussion :

- Why some of the input combinations results in outputs that cannot be represented using LED LD0-LD3?

  Inputs ranging from 0 to 15 can be provided from SW0 to SW3 or SW12 to SW15. As a result, the output will fall between 0 and 30. However we can only indicate outputs in range of 0 to 15 using LD0 - LD3.

- Discuss the role of LD15

  The aim of LD15 is to show the overflow that we obtain from the numbers in the range of 16 to 30, as we can only output values from 0 to 15.

Conclusions :

Fundamental circuits are used to construct even the most complicated circuits. In this lab, the Half Adder was utilized to create a Full Adder and a 4-Bit Ripple Carry Adder. Furthermore, we bundled our IP for next developments.