Lab 3 – Ripple Carry Adder

CS1050 Computer Organization and Digital Design

Name: Oshadi Saumya Perera

Index No: 200458M

Lab Task

- In this lab we will design a 4-bit Ripple Carrier Adder (RCA). We will use 2 Half Adders (HA) to build a Full Adder (FA) and 4 Full Adders to build the RCA.
- We will create an FA symbol for future use of FA.
- Also we will learn to build more complex components using many basic components.
- Verify their functionality via simulation.

Truth tables and steps involved in simplifying the Boolean expressions

Half Adder

Truth Table

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

K-map (S)

A \ B	0	1
0	0	1
1	1	0

$S = A.B' + A'.B = A \oplus B$

K-map (C)

A \ B	0	1
0	0	0
1	0	1

$C = A.B$

Full Adder

Truth Table

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

K-map (S)

AB / Cin	00	01	11	10
0	0	1	0	1
1	1	0	1	0

$$S = A'.B'.Cin + A'.B.Cin' + A.B.Cin + A.B'.Cin'$$

$$S = B'.(A'.Cin + A.Cin') + B.(A'.Cin' + A.Cin)$$

$$(A'.Cin + A.Cin')' = (A + Cin').(A' + Cin)$$

$$(A'.Cin + A.Cin')' = A.A' + A.Cin + Cin'.A' + Cin'.Cin$$

$$(A'.Cin + A.Cin')' = 0 + A.Cin + Cin'.A' + 0$$

$$(A'.Cin + A.Cin')' = A.Cin + Cin'.A'$$

$$S = B'.(A'.Cin + A.Cin') + B.(A'.Cin + A.Cin')'$$

$$S = B \oplus (A'.Cin + A.Cin')$$

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

$$S = A \oplus B \oplus Cin$$

K-map (C_out)

AB / Cin	00	01	11	10
0	0	0	1	0
1	0	1	1	1

$$Cout = A.B + B.Cin + A.Cin$$

Full Adder (2 HA's)

- First Half Adder

$$Sh1 = A \oplus B$$

$$Ch1 = A.B$$

- Full Adder

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

$$S = Sh1 \oplus Cin$$

$$Cout = A.Cin + A.B + B.Cin$$

$$Cout = A.B + Cin.(A + B)$$

$$Cout = A.B + Cin.\big((A \oplus B) + A.B\big)$$

$$Cout = A.B + Cin.(A \oplus B) + Cin.(A.B)$$

$$Cout = A.B.(1 + Cin) + Cin.(A \oplus B)$$

$$Cout = A.B.1 + Cin.(A \oplus B)$$

$$Cout = A.B + Cin.(A \oplus B)$$

$$Cout = Ch1 + Cin.Sh1$$

$$Cout = Ch1 + Ch2$$

VHDL design source codes

- Half Adder design source code (HA.vhd)

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 05/26/2022 10:53:17 PM
-- Design Name:
-- Module Name: HA - Behavioral
-- Project Name:
-- Target Devices:
-- Tool Versions:
-- Description:
--
-- Dependencies:
--
-- Revision:
-- Revision 0.01 - File Created
-- Additional Comments:
--
----------------------------------------------------------------------------------
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 HA is
   Port ( A : in STD_LOGIC;
          B : in STD_LOGIC;
          S : out STD_LOGIC;
          C : out STD_LOGIC);
end HA;
architecture Behavioral of HA is
begin
   S <= A XOR B;
   C <= A AND B;
end Behavioral;
```

- Half Adder Test bench code (TB_HA.vhd)

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 05/26/2022 11:03:34 PM
-- Design Name:
-- Module Name: TB_HA - Behavioral
-- Project Name:
-- Target Devices:
-- Tool Versions:
-- Description:
--
-- Dependencies:
--
-- Revision:
-- Revision 0.01 - File Created
-- Additional Comments:
--
----------------------------------------------------------------------------------
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_HA is
--  Port ( );
end TB_HA;

architecture Behavioral of TB_HA is
component HA
   PORT(
      A : in STD_LOGIC;
      B : in STD_LOGIC;
      S : out STD_LOGIC;
      C : out STD_LOGIC
   );
end component;
signal a, b, s, c : STD_LOGIC;
begin
UUT : HA
   PORT MAP(
      A => a,
      B => b,
      S => s,
      C => c
   );
   process
   begin
      a <= '0';
      b <= '0';
      wait for 10ns;
```

```
        b <= '1';
        wait for 10ns;

        a <='1';
        b <= '0';
        wait for 10ns;

        b <= '1';
        wait;
    end process;
end Behavioral;
```

- Full Adder design source code (FA.vhd)

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 05/26/2022 11:45:25 PM
-- Design Name:
-- Module Name: FA - Behavioral
-- Project Name:
-- Target Devices:
-- Tool Versions:
-- Description:
--
-- Dependencies:
--
-- Revision:
-- Revision 0.01 - File Created
-- Additional Comments:
--
----------------------------------------------------------------------------------
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 FA is
   Port ( A : in STD_LOGIC;
          B : in STD_LOGIC;
          C_in : in STD_LOGIC;
          S : out STD_LOGIC;
          C_out : out STD_LOGIC);
end FA;

architecture Behavioral of FA is
component HA
   PORT(
      A : in STD_LOGIC;
      B : in STD_LOGIC;
      S : out STD_LOGIC;
      C : out STD_LOGIC
   );
end component;
signal HA0_S, HA0_C, HA1_S, HA1_C : STD_LOGIC;
begin
HA_0 : HA
   PORT MAP(
      A => A,
      B => B,
      S => HA0_S,
      C => HA0_C
   );
```

```
HA_1 : HA
   PORT MAP(
      A => HA0_S,
      B => C_in,
      S => HA1_S,
      C => HA1_C
   );
C_out <= HA0_C OR HA1_C;
S <= HA1_S;
end Behavioral;
```

- Full Adder Test bench code (TB_FA.vhd)

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 05/27/2022 12:20:00 AM
-- Design Name:
-- Module Name: TB_FA - Behavioral
-- Project Name:
-- Target Devices:
-- Tool Versions:
-- Description:
--
-- Dependencies:
--
-- Revision:
-- Revision 0.01 - File Created
```

```
-- Additional Comments:
--
----------------------------------------------------------------------------------
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_FA is
--  Port ( );
end TB_FA;

architecture Behavioral of TB_FA is
component FA
   PORT(
      A : in STD_LOGIC;
      B : in STD_LOGIC;
      C_in : in STD_LOGIC;
      S : out STD_LOGIC;
      C_out : out STD_LOGIC
   );
end component;
signal a, b, c_in, s, c_out : STD_LOGIC;
begin
```

```
UUT : FA
   PORT MAP(
      A => a,
      B => b,
      C_in => c_in,
      S => s,
      C_out => c_out
   );

   process
   begin
      a <= '0';
      b <= '0';
      c_in <= '0';
      wait for 100ns;

      c_in <= '1';
      wait for 100ns;

      b <= '1';
      c_in <= '0';
      wait for 100ns;

      c_in <= '1';
      wait for 100ns;

      a <= '1';
      b <= '0';
      c_in <= '0';
```

```
        wait for 100ns;

        c_in <= '1';
        wait for 100ns;

        b <= '1';
        c_in <= '0';
        wait for 100ns;

        c_in <= '1';
        wait;

    end process;
end Behavioral;
```

- Ripple Carry Adder design source code (RCA_4.vhd)

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 05/27/2022 01:18:57 AM
-- Design Name:
-- Module Name: RCA_4 - Behavioral
-- Project Name:
-- Target Devices:
-- Tool Versions:
-- Description:
--
-- Dependencies:
--
-- Revision:
-- Revision 0.01 - File Created
-- Additional Comments:
--
----------------------------------------------------------------------------------
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 RCA_4 is
    Port ( A0 : in STD_LOGIC;
           A1 : in STD_LOGIC;
           A2 : in STD_LOGIC;
           A3 : in STD_LOGIC;
           B0 : in STD_LOGIC;
           B1 : in STD_LOGIC;
           B2 : in STD_LOGIC;
           B3 : in STD_LOGIC;
           C_in : in STD_LOGIC;
           S0 : out STD_LOGIC;
           S1 : out STD_LOGIC;
           S2 : out STD_LOGIC;
           S3 : out STD_LOGIC;
           C_out : out STD_LOGIC);
end RCA_4;

architecture Behavioral of RCA_4 is
    component FA
        port (
            A: in std_logic;
            B: in std_logic;
            C_in: in std_logic;
            S: out std_logic;
            C_out: out std_logic
        );
    end component;
```

```
    SIGNAL FA0_S, FA0_C, FA1_S, FA1_C, FA2_S, FA2_C, FA3_S, FA3_C  : std_logic;

begin
    FA_0 : FA
      port map (
         A => A0,
         B => B0,
         C_in => C_in,
         S => S0,
         C_Out => FA0_C
      );

     FA_1 : FA
      port map (
          A => A1,
          B => B1,
          C_in => FA0_C,
          S => S1,
          C_Out => FA1_C
       );

     FA_2 : FA
      port map (
         A => A2,
         B => B2,
         C_in => FA1_C,
         S => S2,
         C_Out => FA2_C
```

```
    );

    FA_3 : FA
     port map (
       A => A3,
       B => B3,
       C_in => FA2_C,
       S => S3,
       C_Out => C_out
     );
end Behavioral;
```

- Ripple Carry Adder Test bench code (TB_4_RCA.vhd)

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 05/27/2022 05:00:59 PM
-- Design Name:
-- Module Name: TB_4_RCA - Behavioral
-- Project Name:
-- Target Devices:
-- Tool Versions:
-- Description:
--
-- Dependencies:
--
```

```
-- Revision:
-- Revision 0.01 - File Created
-- Additional Comments:
--
----------------------------------------------------------------------------------
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 : in STD_LOGIC;
         A1 : in STD_LOGIC;
         A2 : in STD_LOGIC;
         A3 : in STD_LOGIC;
         B0 : in STD_LOGIC;
         B1 : in STD_LOGIC;
         B2 : in STD_LOGIC;
         B3 : in STD_LOGIC;
```

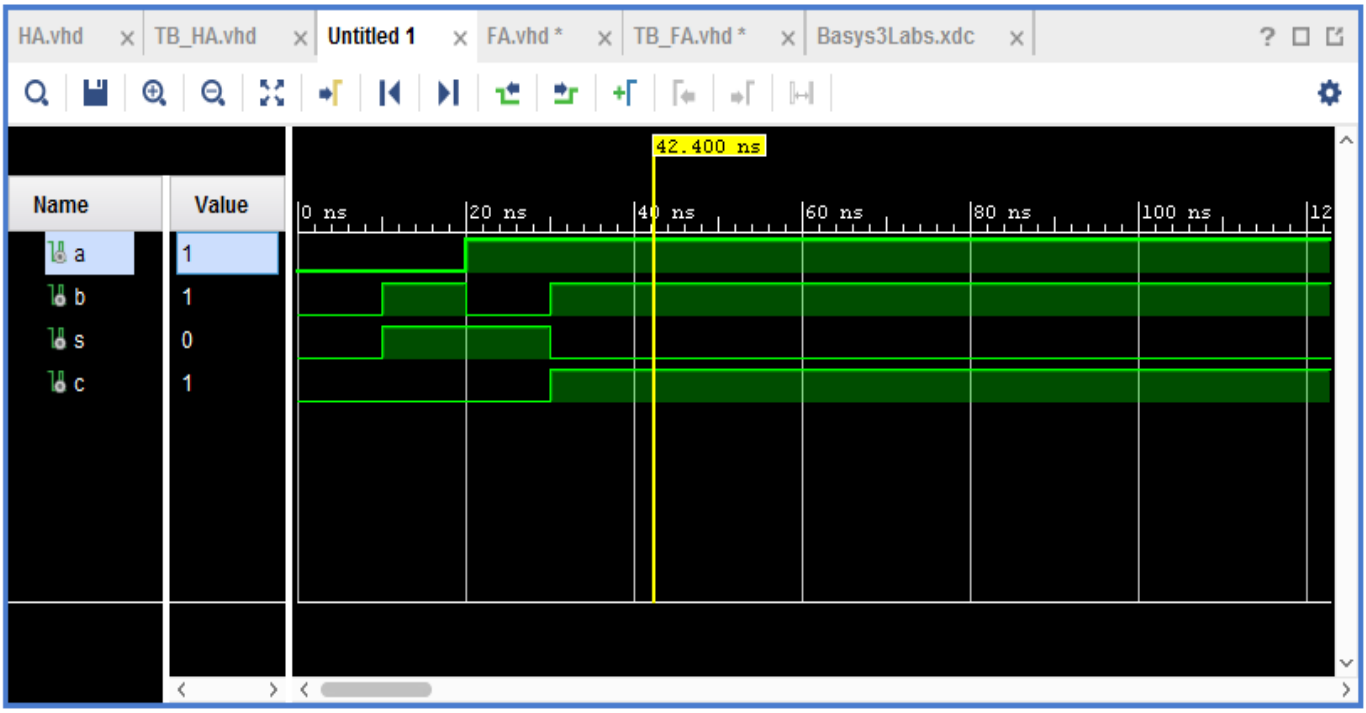
```
        C_in : in STD_LOGIC;
        S0 : out STD_LOGIC;
        S1 : out STD_LOGIC;
        S2 : out STD_LOGIC;
        S3 : out STD_LOGIC;
        C_out : out STD_LOGIC
    );

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

begin
  UUT : 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 <= '1';
   a1 <= '0';
   a2 <= '1';
   a3 <= '0';
   b0 <= '0';
   b1 <= '0';
   b2 <= '0';
   b3 <= '0';
   c_in <= '0';
   wait for 100ns;

   a0 <= '1';
   a1 <= '1';
   a2 <= '1';
   a3 <= '1';
   b0 <= '0';
   b1 <= '0';
   b2 <= '0';
   b3 <= '0';
   c_in <= '0';
   wait for 100ns;

   a0 <= '0';
   a1 <= '1';
   a2 <= '0';
   a3 <= '1';
```

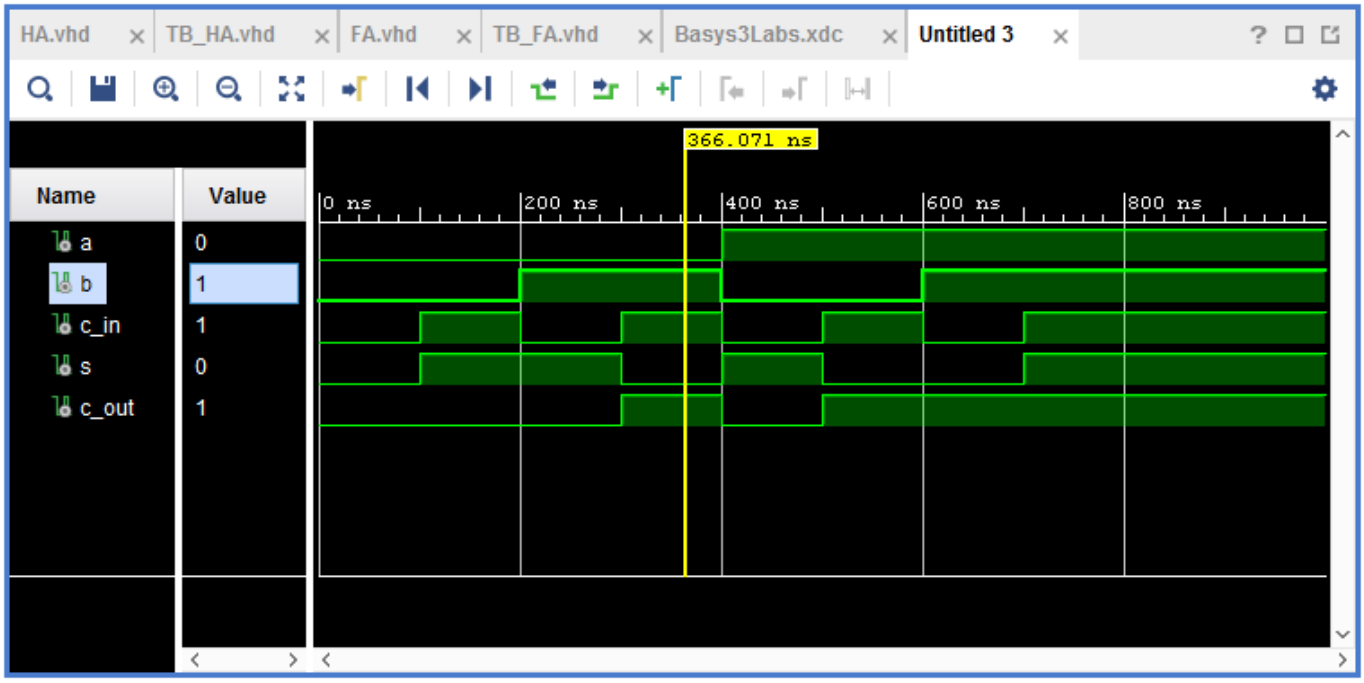
```
        b0 <= '1';
        b1 <= '1';
        b2 <= '1';
        b3 <= '1';
        c_in <= '0';
        wait;
    end process;

end Behavioral;
```

Timing Diagrams

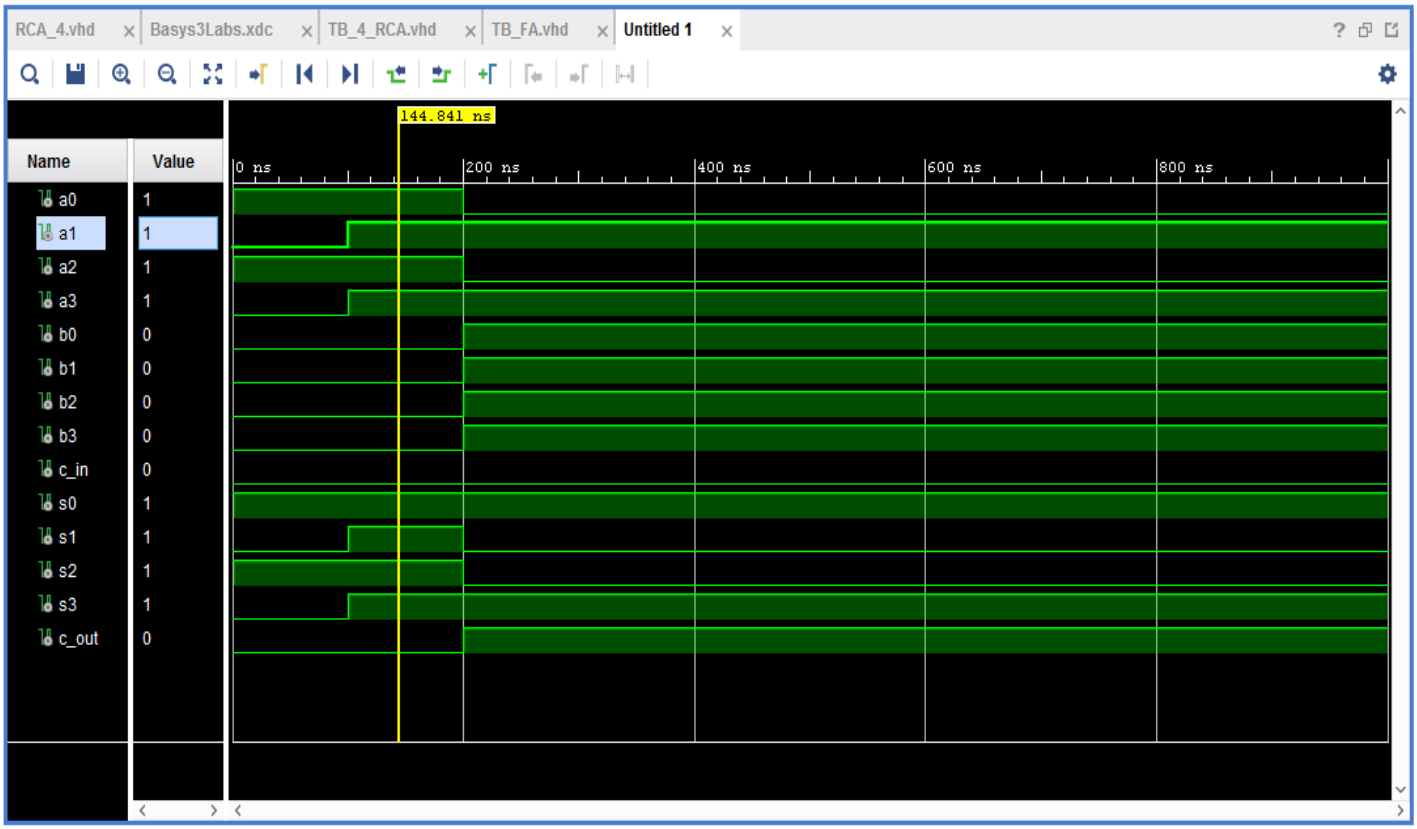
- Half Adder



- Full Adder



- Ripple Carry Adder



Discuss why some of the input combinations results in outputs that cannot be represented using LED LD0-LD3.

In 4-bit RCA there are two types of outputs, sum of each FA (S0, S1, S2, and S3) and carry bit (C_out). We use LED LD0-LD3 to represent sum of each FA and LD15 to represent carry bit.

Discuss the role of LD15.

LD15 is used to connect carry bit (C_out) of 4-bit RCA.

Conclusion

We can develop an 8-bit Ripple Carry Adder using two 4-bit Ripple Carry Adders. Also we can extend RCA to support subtraction.