Lab 3

Ripple Carry Adder

CS1050 Computer Organization and Digital Design

RATHNAYAKE R.M.I.B
220526N

Group 41

Task:

As the task, we had to implement Half Adder using basic logic gates. Using those pre-built Half Adders, we implemented the Full Adder. Then, using those Full Adders, a Ripple Carry Adder was built and run simulations for several inputs. Then finally, this Ripple Carry Adder tests on the BYSYS3 board.

1. Half Adder

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

Boolean Expressions:

$S = A \oplus B$

$C = A.B$

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

Boolean Expressions:

$S = (A \oplus B) \oplus C\_in$

$C\_out = A.B + C\_in.(A \oplus B)$

Derive these Boolean Expressions:

- K map for "S"

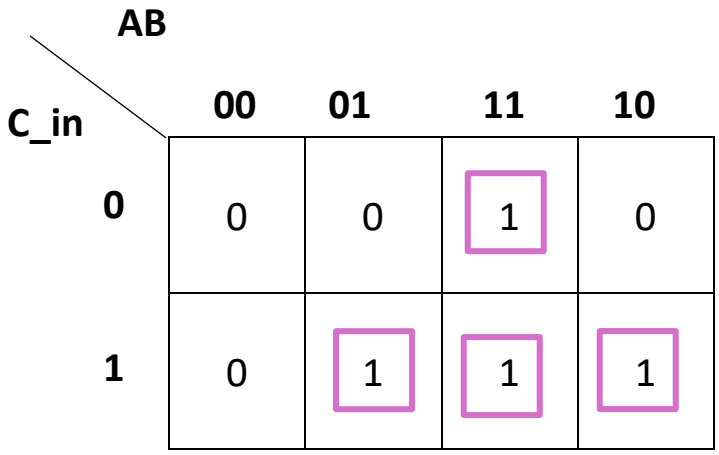
C_in \ AB	00	01	11	10
0	0	1	0	1
1	1	0	1	0

$S = A'.B'.C\_in + A'.B.C\_in' + A.B.C\_in + A.B'.C\_in'$

$= (A'.B' + A.B).C\_in + (A'.B + A.B').C\_in'$

$= (A \oplus B) \oplus C\_in$

- K map for “C_out”



C_out = A.B. .C_in’ + A’.B.C_in + A.B’.C_in + A.B.C_in

= A.B (C_in + C_in’).+ (A’.B + A.B’).C_in

= A.B + (A ⊕ B).C_in

(We use this way to combine those 1’s to easily get this "A.B. + (A ⊕ B).C” equation.)

VHDL FILES

1. HA

```
----------------------------------------------------------------------------
--------
-- Company:
-- Engineer:
--
-- Create Date: 02/13/2024 02:33:43 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;
```

```
----------------------------------------------------------------------------
--------
-- Company:
-- Engineer:
--
-- Create Date: 02/13/2024 02:49:42 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, B : IN STD_LOGIC;
          S, C : OUT STD_LOGIC);
END COMPONENT;

SIGNAL A,B : std_logic;
SIGNAL 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 100ns;
        B <= '1';

        WAIT FOR 100ns;
        A <= '1';
        B <= '0';

        WAIT FOR 100ns;
        B <= '1';

        WAIT;
end process;

end Behavioral;
```

2. FA

```
----------------------------------------------------------------------------
--------
-- Company:
-- Engineer:
--
-- Create Date: 02/13/2024 03:14:39 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);

S <= HA1_S;
C_out <= HA0_C OR HA1_C;

end Behavioral;
```

```
----------------------------------------------------------------------------
--------
-- Company:
-- Engineer:
```

```
--
-- Create Date: 02/13/2024 03:51:49 PM
-- 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, B, C_in : IN STD_LOGIC;
          S, C_out : OUT STD_LOGIC);
END COMPONENT;
```

```
signal A, B, C_in : std_logic;
signal S, C_out : std_logic;

begin
UUT: FA PORT MAP(
    A => A,
    B => B,
    S => S,
    C_in => C_in,
    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;
```

3. RCA

```
----------------------------------------------------------------------------
--------
-- Company:
-- Engineer:
--
-- Create Date: 02/13/2024 05:26:34 PM
-- 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;
```

```
----------------------------------------------------------------------------
--------
-- Company:
-- Engineer:
--
```

```
-- Create Date: 02/13/2024 05:49:17 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, A1, A2, A3, B0, B1, B2, B3, C_in : IN STD_LOGIC;
          S0, S1, S2, S3, C_out : OUT STD_LOGIC);
END COMPONENT;
```

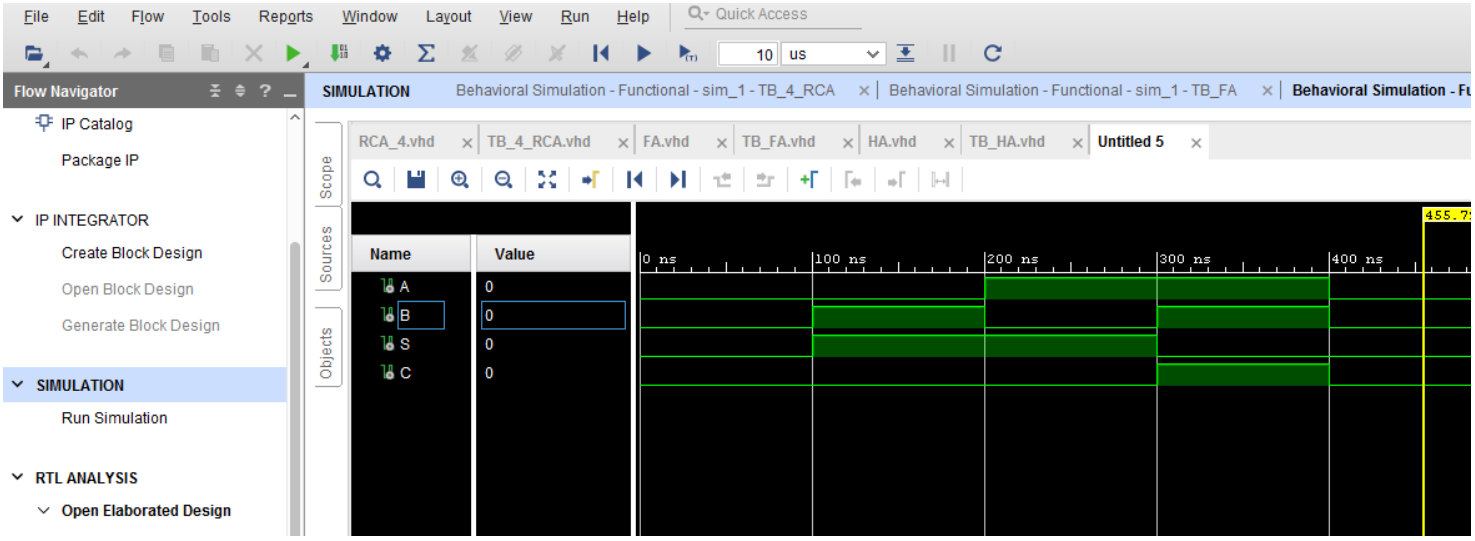
```
SIGNAL A0, A1, A2, A3, B0, B1, B2, B3, C_in : std_logic;
SIGNAL S0, S1, S2, S3, C_out: std_logic;

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,
    C_out => C_out,
    S0 => S0,
    S1 => S1,
    S2 => S2,
    S3 => S3
);

process
begin
       A0 <= '0';
       A1 <= '1';
       A2 <= '1';
       A3 <= '1';
       B0 <= '0';
       B1 <= '1';
       B2 <= '1';
       B3 <= '0';
       C_in <= '0';

       WAIT FOR 100ns;
       A0 <= '1';
       A1 <= '0';
       A2 <= '1';
       A3 <= '1';
       B0 <= '1';
       B1 <= '0';
```

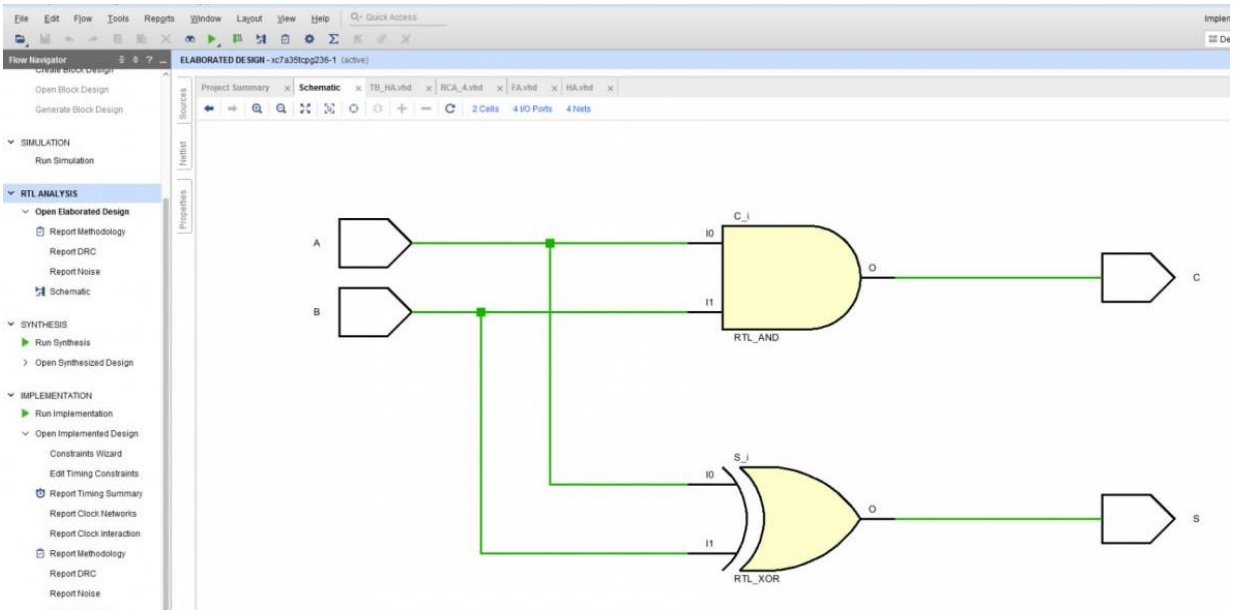
```
        B2 <= '1';
        B3 <= '0';

        WAIT FOR 100ns;
         A0 <= '1';
         A1 <= '1';
         A2 <= '0';
         A3 <= '1';
         B0 <= '1';
         B1 <= '0';
         B2 <= '1';
         B3 <= '0';

      WAIT FOR 100ns;
         A0 <= '1';
         A1 <= '1';
         A2 <= '1';
         A3 <= '1';
         B0 <= '1';
         B1 <= '1';
         B2 <= '1';
         B3 <= '0';

-------------Other 4 Combinations---------------------------

        WAIT FOR 100ns;
         A0 <= '0';
         A1 <= '0';
         A2 <= '0';
         A3 <= '0';
         B0 <= '1';
         B1 <= '1';
         B2 <= '1';
         B3 <= '1';

        WAIT FOR 100ns;
         A0 <= '1';
         A1 <= '1';
         A2 <= '0';
         A3 <= '0';
```

```
        B0 <= '1';
        B1 <= '1';
        B2 <= '0';
        B3 <= '0';

        WAIT FOR 100ns;
        A0 <= '1';
        A1 <= '1';
        A2 <= '1';
        A3 <= '1';
        B0 <= '1';
        B1 <= '1';
        B2 <= '1';
        B3 <= '1';

        WAIT FOR 100ns;
        A0 <= '1';
        A1 <= '1';
        A2 <= '0';
        A3 <= '0';
        B0 <= '0';
        B1 <= '0';
        B2 <= '1';
        B3 <= '0';

-------GET BACK TO ALL 0------

        WAIT FOR 100ns;
        A0 <= '0';
        A1 <= '0';
        A2 <= '0';
        A3 <= '0';
        B0 <= '0';
        B1 <= '0';
        B2 <= '0';
        B3 <= '0';

       WAIT;
end process;
end Behavioral;
```

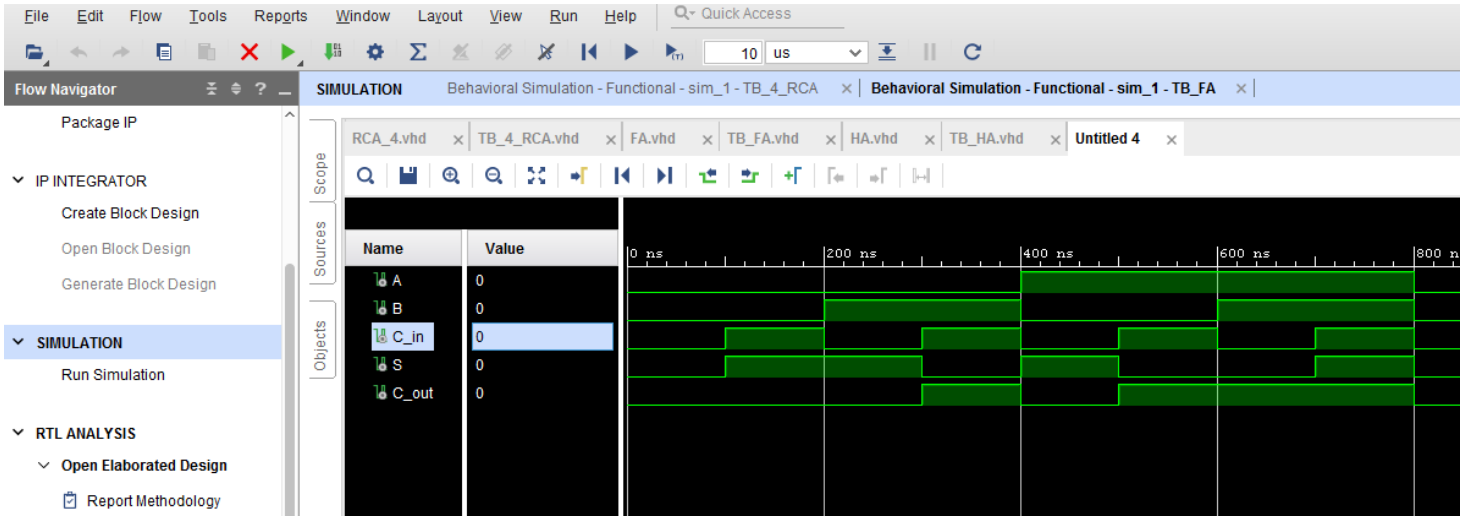
Time Diagrams

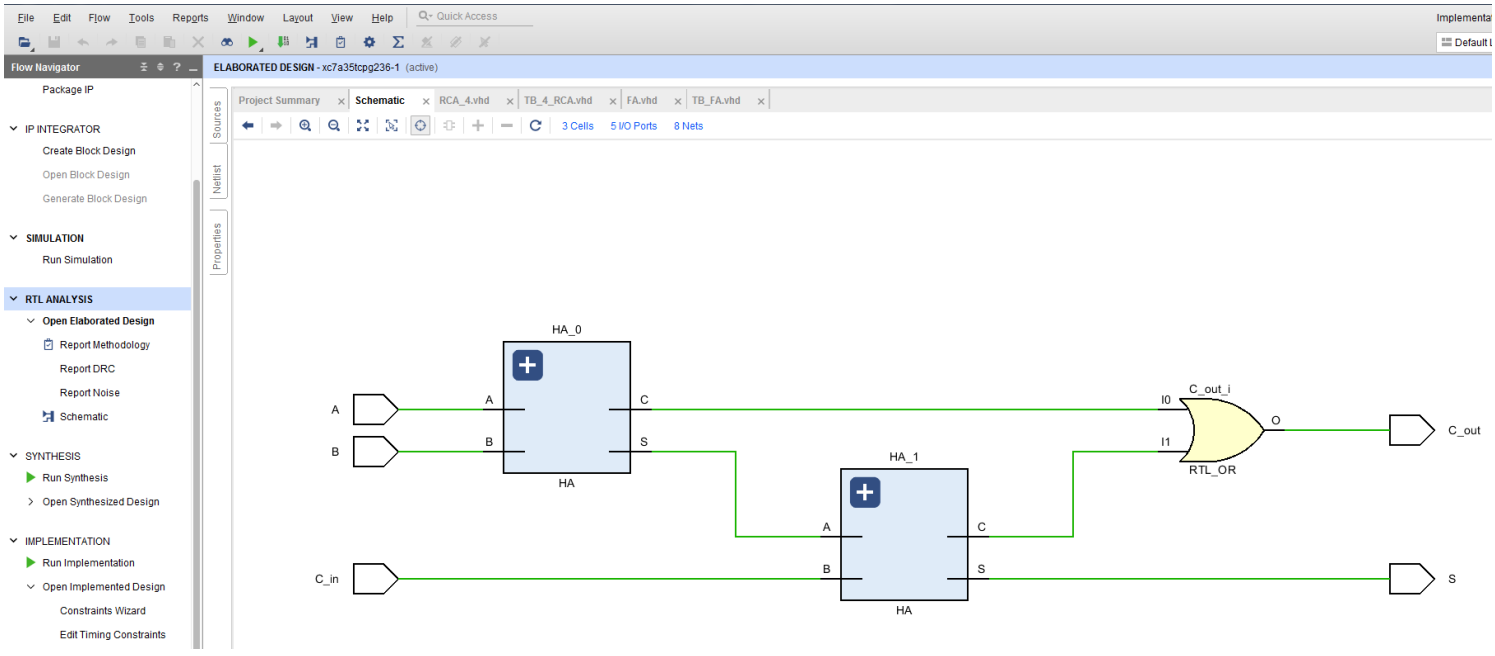
1. HA Time Diagram



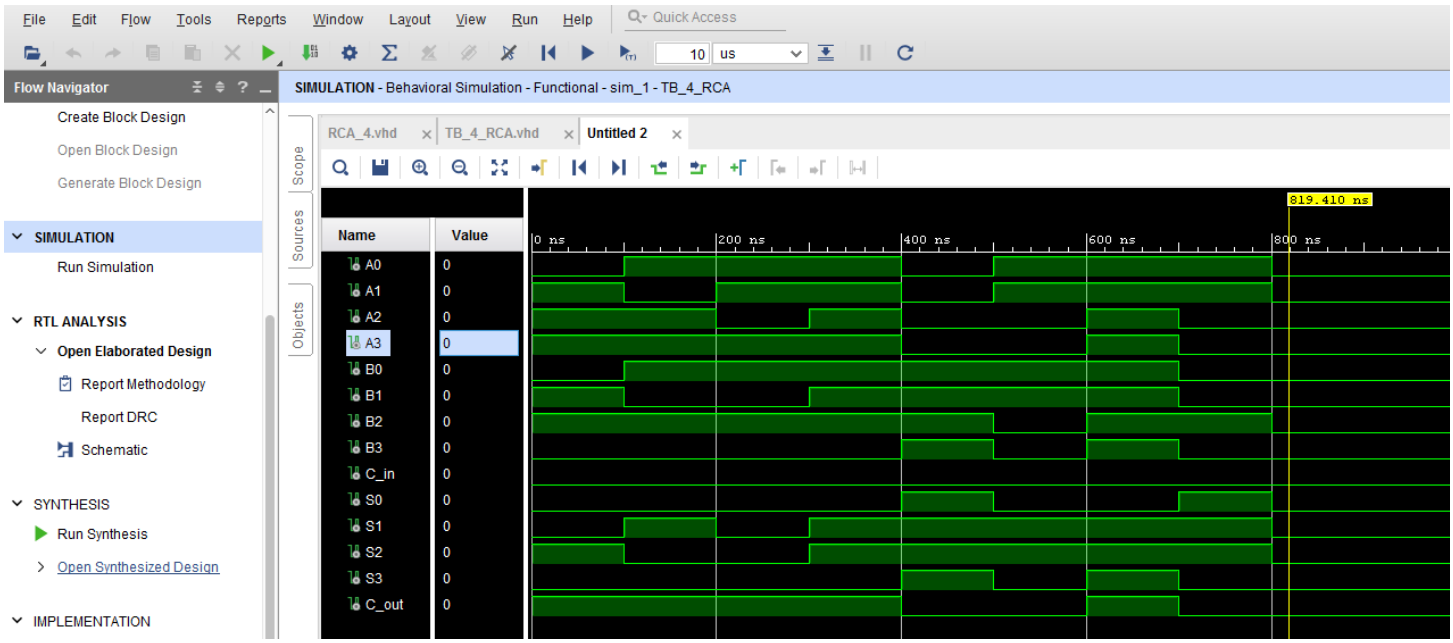


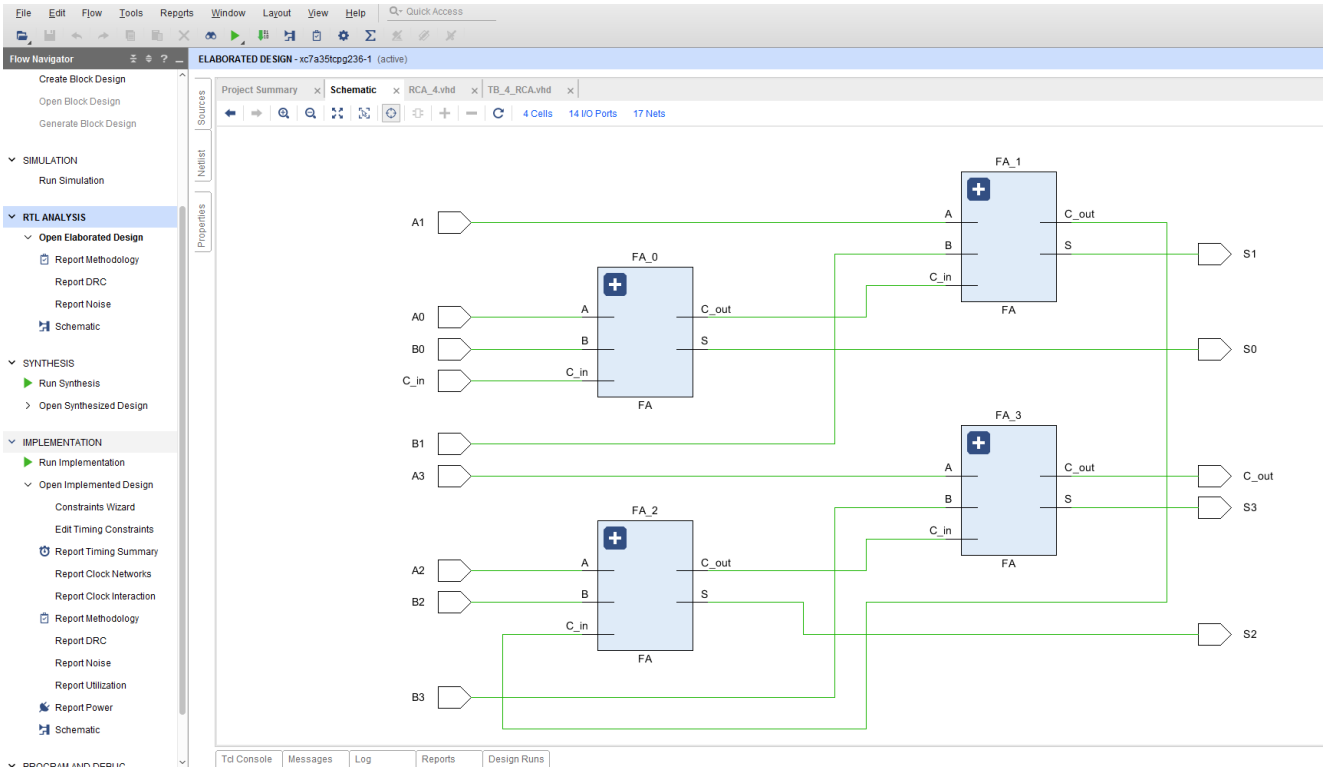
2. FA Time diagram





3. RCA Time Diagram





Discussion:

- Why some of the input combinations results in outputs that cannot be represented using LED LD0-LD3. Discuss the role of LD15?

LED: LD0-LD3 is used to indicate the sum of two 4-bit binary numbers.

i.e.

```
  0010
+ 1010
------
  1100
```

- We can indicate this output by those LED's.

But the addition of,

```
  1111
+ 1010
------
 (1) 1001
------
```

- This “1001” is the output on S0-S3, which is represented by LED: LD0-LD3. This is an incorrect response. The correct output is "**1** 1001," which is a 5-bit binary number, and this overflow carry-bit can be represented using the LD15.

Overflow carry bit.

Conclusion:

Since we first started with basic logic gates and combined those basic gates, we can implement more complex components. Creating small Boolean circuits is much easier, and combining those circuits helps create more complex circuits in less time. As well, we can import these modules for the very advanced circuit-building process.