Lab 3 – Ripple Carry Adder

CS1050 Computer Organization and Digital Design

Dept. of Computer Science and Engineering, University of Moratuwa

Name : **Dissanayake D.M.A.K.**
Index No. : **2202135N**
Group : **44**

Lab Task: Design and Testing of a 4-bit Ripple Carry Adder (RCA)

Objective: To design and validate the functionality of a 4-bit Ripple Carry Adder using basic logic components.

Procedure:

- **Determine the Boolean Expressions for Half Adder (HA) and Full Adder (FA) and get the Simplified Expression.**
- **Truth Table for HA and FA:** Create a truth table for Half Adder (HA) and Full Adder (FA) based on the Boolean expressions derived in step 1.
- **Circuit Design for HA:** Develop a circuit diagram for the Half Adder (HA) based on the simplified Boolean expression.
- **Simulation of HA Circuit:** Test the functionality of the Half Adder (HA) by simulating the circuit with the all four input combinations.
- **Circuit Design for FA using HA:** Utilize the Half Adder (HA) as a building block to design a Full Adder (FA) then develop a circuit diagram for the Full Adder (FA) using the Half Adder.
- **Simulation of FA Circuit:** Test the functionality of the Full Adder (FA) by simulating the circuit with all of eight input scenarios and then create a symbol representation for the Full Adder.

- **Circuit Design for 4-bit RCA:** Construct a 4-bit Ripple Carry Adder using the Full Adder (FA) as the basic building block and connect the Full Adders in a cascading fashion to form the 4-bit Ripple Carry Adder.
- **Simulation of 4-bit RCA Circuit:** Validate the functionality of the 4-bit Ripple Carry Adder by simulating the circuit with a set of test inputs (e.g., 8 different input combinations).

- **Testing on BASYS3:** Implement the designed 4-bit Ripple Carry Adder on the BASYS3 FPGA board and verify its performance by testing with real hardware.

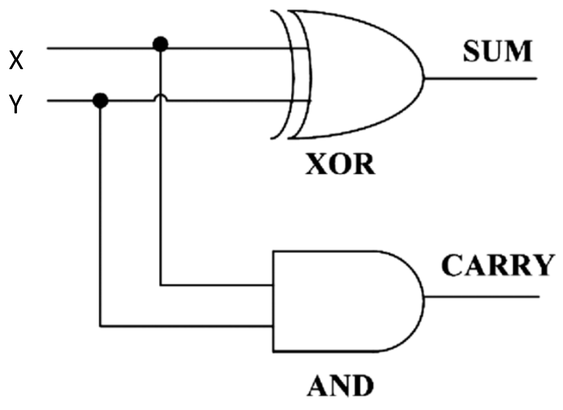
Truth Tables and the Boolean Expressions

Half Adder

Inputs		Outputs	
X	Y	Carry (C)	Sum (S)
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

$$\boldsymbol{S} = \boldsymbol{X} \oplus Y$$

$$\boldsymbol{C} = \boldsymbol{X}.\boldsymbol{Y}$$



circuit design for Half Adder

Full Adder

Inputs			Outputs	
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

$\boldsymbol{Logical\ Expression\ for\ S}$:

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

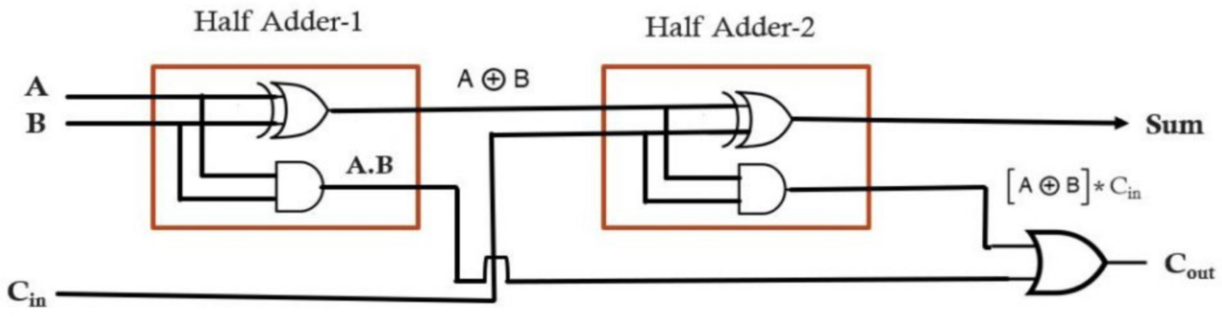
$= C\_in\,(A'\,B' + A\,B) + C\_in'\,(A'\,B + A\,B')$

$= C\_in \oplus (A \oplus B) = (1,2,4,7)$

$\boldsymbol{Logical\ Expression\ for\ C\_out}$:

$= A'\,B\,C\_in + A\,B'\,C\_in + A\,B\,C\_in' + A\,B\,C\_in$

$= A\,B + B\,C\_in + A\,C\_in = (3,5,6,7)$



Circuit design for Full adder

VHDL Files of Design Source, Simulation Files and Timing Diagrams

Half Adder - Behavioral

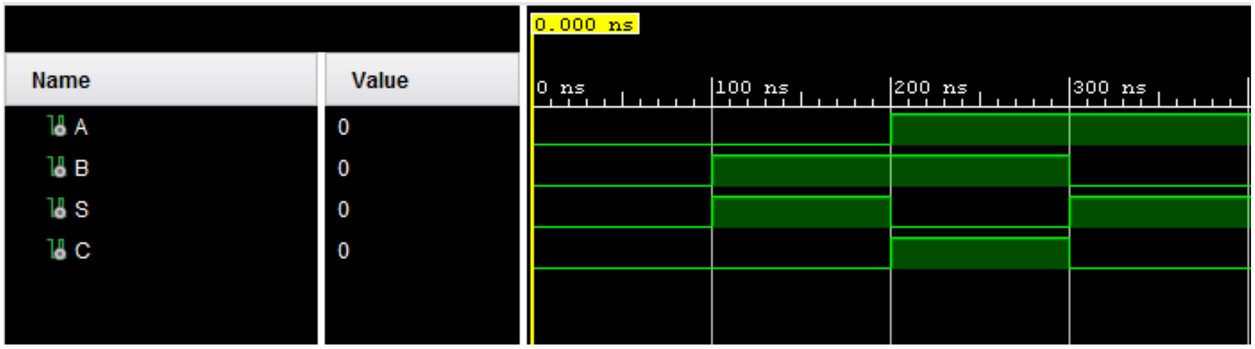
```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 02/13/2024 01:11:37 PM
-- Design Name:
-- Module Name: HA - Behavioral
-- Project Name: Ripple Carry Adder
-- Target Devices:
-- Tool Versions:
-- Description:
--
-- Dependencies:
--
-- 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

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 02/13/2024 01:17:35 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 100 ns;
   B <= '1';
   WAIT FOR 100 ns;
   A <= '1';
   WAIT FOR 100 ns;
   B <= '0';
   WAIT;

end process;


end Behavioral;
```

Half Adder – Timing Diagram



Full Adder - Behavioral

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 02/13/2024 01:34:23 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;
```

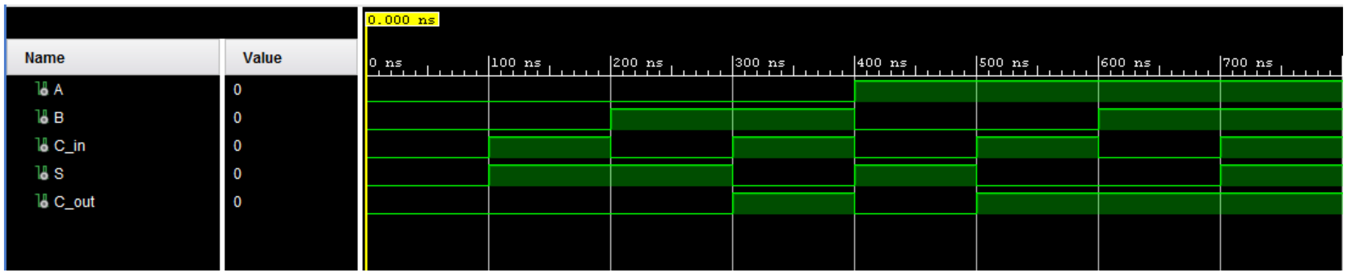
Full Adder – Test Bench File

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 02/13/2024 01:44:57 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,
              C_in => C_in,
                 S => S,
             C_out => C_out
);

 process
 begin
   A       <= '0';
   B       <= '0';
   C_in    <= '0';
   WAIT FOR 100 ns;
   C_in   <= '1';
   WAIT FOR 100 ns;
```

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

end process;

end Behavioral;
```

Full Adder – Timing Diagram



Ripple Carry Adder - Behavioral

```
----------------------------------------------------------------------------------
-- Company: 
-- Engineer: 
-- 
-- Create Date: 02/13/2024 02:39:52 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 => '0', -- Set to ground 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 File

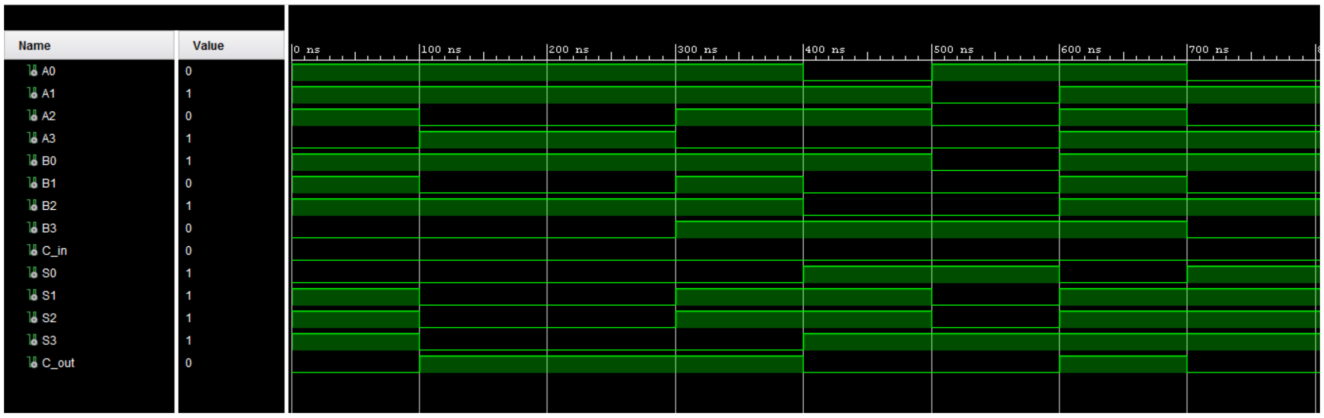
```
----------------------------------------------------------------------------------
-- Company: 
-- Engineer: 
-- 
-- Create Date: 02/13/2024 02:55:35 PM
-- Design Name: 
-- Module Name: TB_4_RCA - Behavioral
-- Project Name: 
-- Target Devices: 
-- Tool Versions: 
-- Description: 
-- 
-- Dependencies: 
-- 
-- 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,
   S0 => S0,
   S1 => S1,
   S2 => S2,
   S3 => S3,
   C_out => C_out
);

 process
 begin
   A0 <= '1';
   A1 <= '1';
   A2 <= '1';
   A3 <= '0';

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

   C_in <= '0';

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

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

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

```
A3 <= '1';

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

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

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

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

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

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

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

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

B0 <= '1';
B1 <= '1';
```

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

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

    WAIT;

end process;

end Behavioral;
```

Ripple Carry Adder – Timing Diagram



Name	Value
A0	0
A1	1
A2	0
A3	1
B0	1
B1	0
B2	1
B3	0
C_in	0
S0	1
S1	1
S2	1
S3	1
C_out	0

Discussion

Output Representation Challenges:

During the analysis of the LED outputs (LD0-LD3), it was observed that certain input combinations might result in outputs that cannot be accurately represented by these LEDs. This limitation arises from factors such as overflow, handling negative numbers, and the absence of error detection indicators. To address this, LD15 was introduced to indicate overflow or carry-out from the most significant bit.

Role of LD15:

LD15 plays a crucial role in indicating situations where the 4-bit representation is inadequate. It serves as an overflow indicator, providing valuable information about scenarios where the LED outputs alone may not fully represent the arithmetic result.

Conclusion:

- The successful design and implementation of the Half Adder, Full Adder, and 4-Bit RCA were achieved.
- The ultimate objective of constructing a 4-bit RCA using Full Adders, themselves constructed with Half Adders, was accomplished.
- The lab experience provided a clear understanding of the hierarchical approach to building a 4-bit RCA (constructed with Full Adders, which, in turn, were built with Half Adders).

-End-