Lab 6

Arithmetic Unit

CS1050 Computer Organization and Digital Design

RATHNAYAKE R.M.I.B
220526N

Group 41

Introduction:

- **Task**

  We have to implement a 4-bit arithmetic unit that can store two numbers in registers and sum them up. This Arithmetic Unit (AU) will be implemented using a prebuilt ripple carry adder (RCA) and a slow clock.

  First, we implemented two 4-bit registers using D-flip flops, which has an enable switch and a clock input. Then those registers are connected to the RCA and form an AU. Finally, we verified the functionality of AU via simulating on the BYASIS 3 board.

- **Implementation of decoding:**

  - 1-to-2 Decoder

Input	Output	
I	Y_1	Y_0
0	0	1
1	1	0

$$Y_0 = \bar{I}$$

$$Y_1 = I$$

  - Boolean expression of the Zero flag

$$Zero = not\ (RCAOutput(0) + RCAOutput(1) + RCAOutput(2) + RCAOutput(3) + carryOut\ )$$

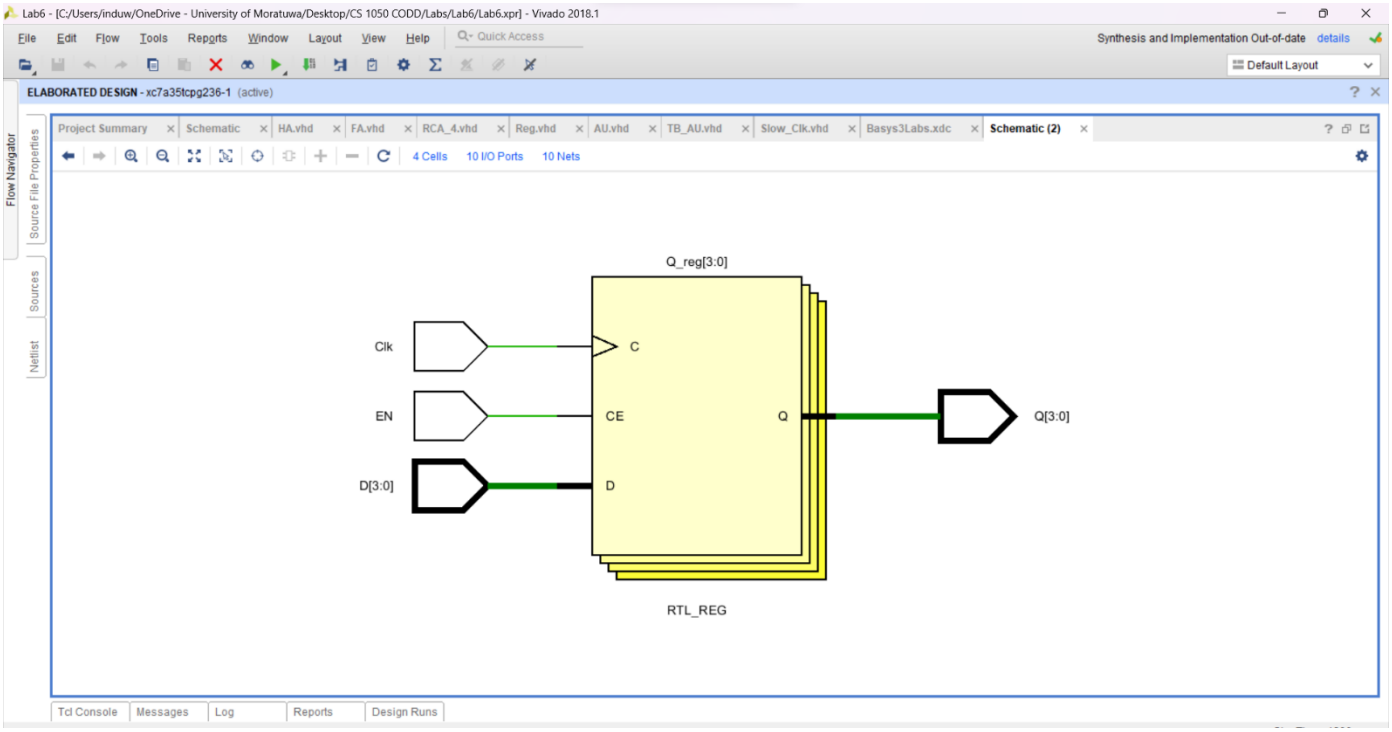
1. 4-bit register

- **Design source file**

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 03/15/2024 10:36:38 AM
-- Design Name:
-- Module Name: Reg - 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 Reg is
    Port ( D : in STD_LOGIC_VECTOR (3 downto 0);
```

```
            EN : in STD_LOGIC;
            Clk : in STD_LOGIC;
            Q : out STD_LOGIC_VECTOR (3 downto 0));
end Reg;

architecture Behavioral of Reg is

begin
    process (Clk) begin
        if (rising_edge(Clk)) then -- respond when clock rises
            if En = '1' then -- Enable should be set
                Q <= D;
            end if;
        end if;
    end process;

end Behavioral;
```

- **Elaborated design schematic**



2. Arithmetic Unit

- **Design source file**

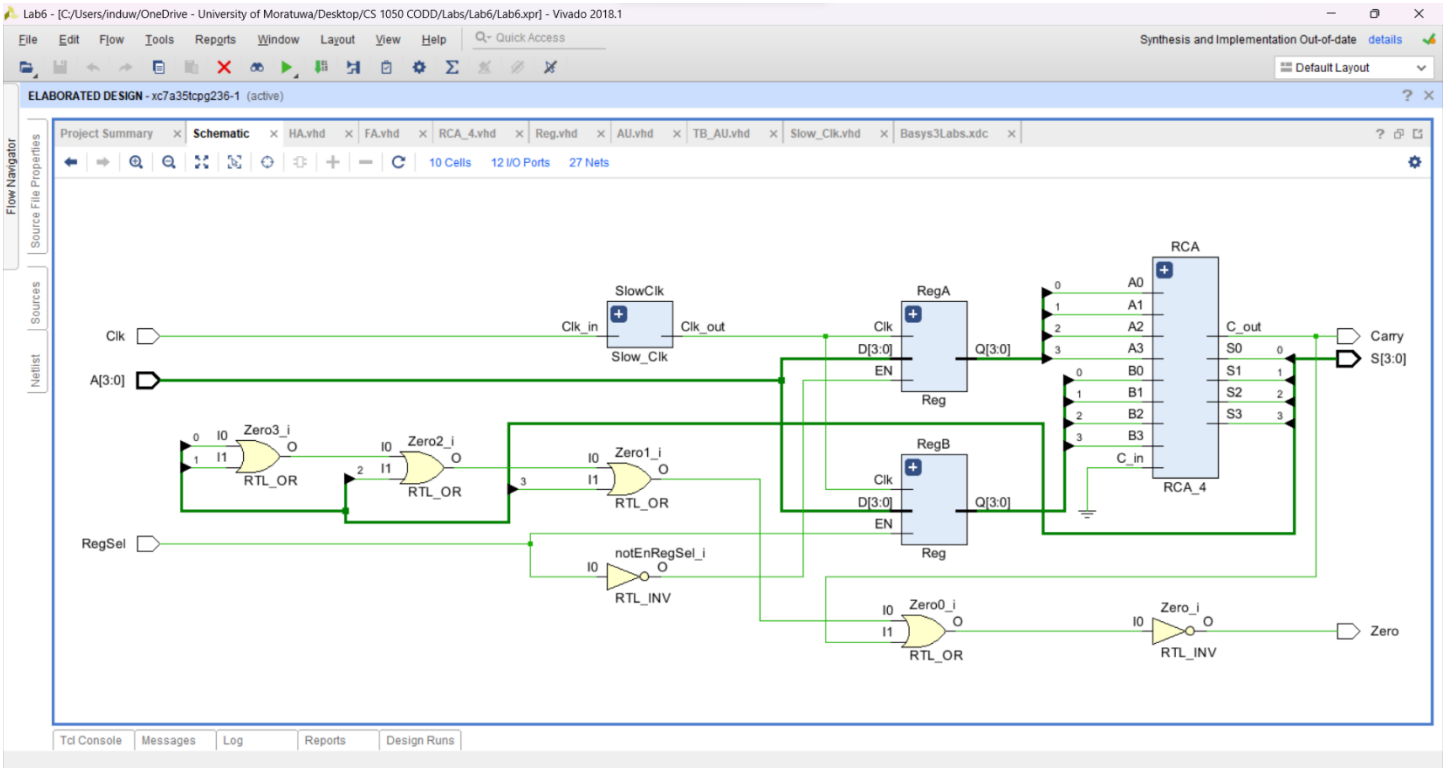
```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 03/15/2024 10:52:44 AM
-- Design Name:
-- Module Name: AU - Behavioral
-- Project Name:
-- Target Devices:
-- Tool Versions:
-- Description:
--
-- Dependencies:
--
-- Revision:
-- 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 AU is
    Port ( A : in STD_LOGIC_VECTOR (3 downto 0);
           RegSel : in STD_LOGIC;
           Clk : in STD_LOGIC;
           S : out STD_LOGIC_VECTOR (3 downto 0);
           Zero : out STD_LOGIC;
           Carry : out STD_LOGIC);
end AU;

architecture Behavioral of AU is
    component Reg
            port( D : in STD_LOGIC_VECTOR (3 downto 0);
               EN : in STD_LOGIC;
               Clk : in STD_LOGIC;
               Q : out STD_LOGIC_VECTOR (3 downto 0));
    end component ;

    component 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 component;

    component Slow_Clk is
```

```
        Port ( Clk_in : in STD_LOGIC;
               Clk_out : out STD_LOGIC);
    end component;


signal SlowClkOut : std_logic;
signal RegAOutput : STD_LOGIC_VECTOR (3 downto 0);
signal RegBOutput : STD_LOGIC_VECTOR (3 downto 0);

signal RCAOutput : STD_LOGIC_VECTOR (3 downto 0);
signal carryOut : std_logic;
signal notEnRegSel : STD_LOGIC;
signal EnRegSel : STD_LOGIC;

begin
EnRegSel <= RegSel;
notEnRegSel <= not RegSel;

SlowClk : Slow_Clk
    port map(
        Clk_in => Clk,
        Clk_out => SlowClkOut);

RegA : Reg
    port map(
        D => A,
        EN => notEnRegSel,
        Clk =>SlowClkOut,
        Q => RegAOutput
        );

RegB : Reg
    port map(
        D => A,
        EN => EnRegSel,
        Clk =>SlowClkOut,
```

```
        Q => RegBOutput
        );

 RCA : RCA_4
    port map(
       A0 => RegAOutput(0),
       A1 => RegAOutput(1),
       A2 => RegAOutput(2),
       A3 => RegAOutput(3),

       B0 => RegBOutput(0),
       B1 => RegBOutput(1),
       B2 => RegBOutput(2),
       B3 => RegBOutput(3),

       C_in => '0',
       S0 => RCAOutput(0),
       S1 => RCAOutput(1),
       S2 => RCAOutput(2),
       S3 => RCAOutput(3),
       C_out => carryOut);

S <= RCAOutput ;
Carry <= carryOut;

Zero <= not(RCAOutput(0) or RCAOutput(1) or RCAOutput(2) or RCAOutput(3)
or carryOut );

end Behavioral;
```

- **Elaborated design schematic**



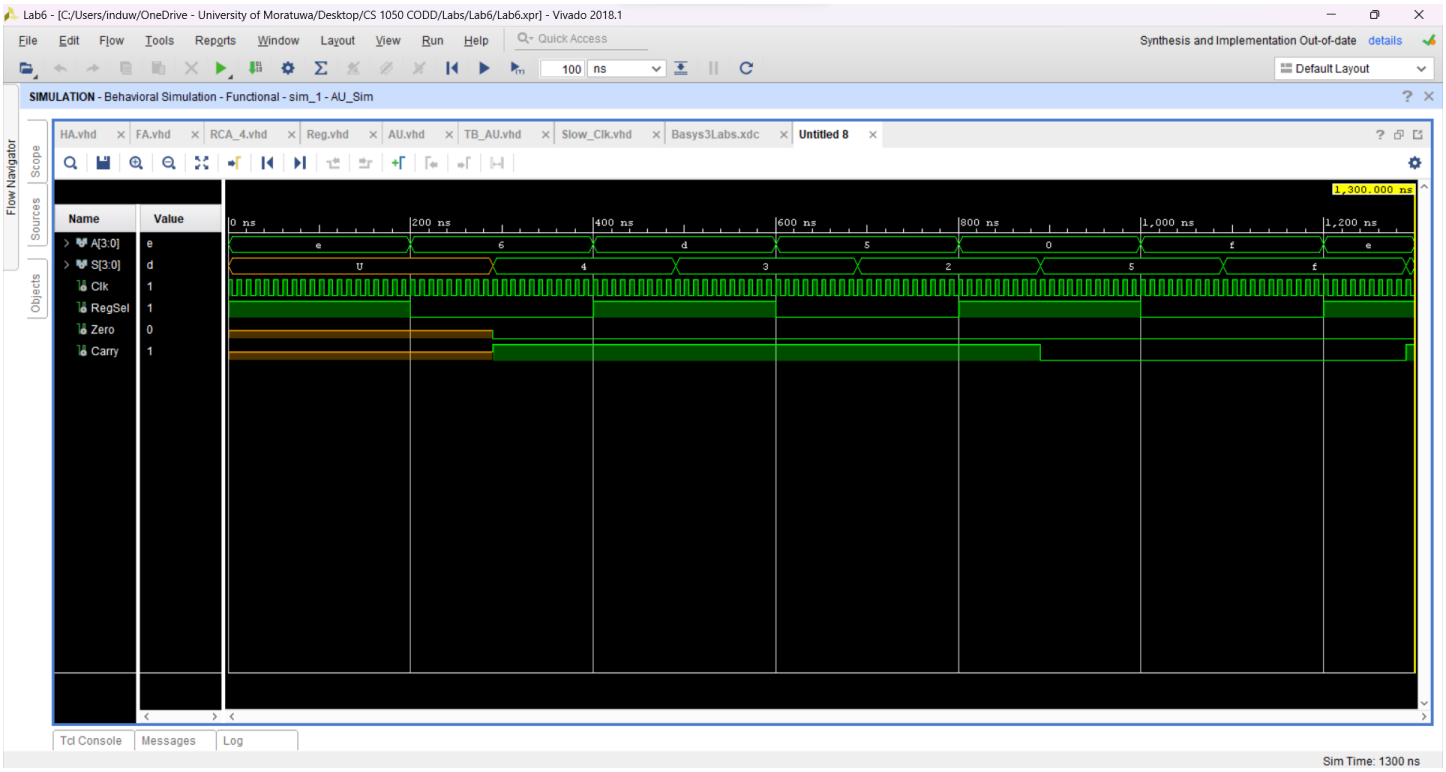
- **Simulation source file**

```
----------------------------------------------------------------------------------
--------
-- Company:
-- Engineer:
--
-- Create Date: 03/15/2024 12:24:57 PM
-- Design Name:
-- Module Name: AU_Sim - Behavioral
-- Project Name:
-- Target Devices:
-- Tool Versions:
-- Description:
--
-- Dependencies:
--
-- Revision:
-- Revision 0.01 - File Created
-- Additional Comments:
```

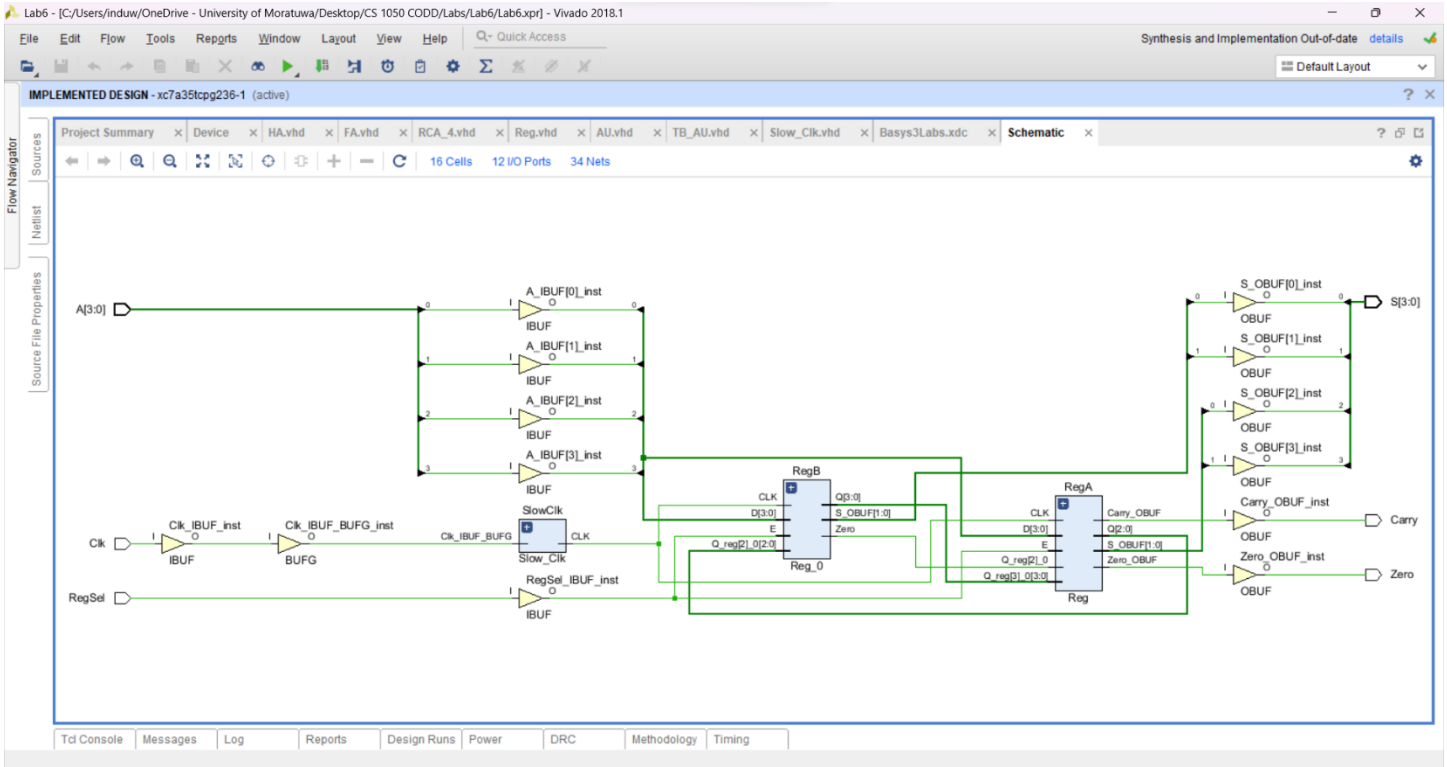
```
--
----------------------------------------------------------------------------
--------


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 AU_Sim is
--  Port ( );
end AU_Sim;

architecture Behavioral of AU_Sim is
component AU is
    Port ( A : in STD_LOGIC_VECTOR (3 downto 0);
           RegSel : in STD_LOGIC;
           Clk : in STD_LOGIC;
           S : out STD_LOGIC_VECTOR (3 downto 0);
           Zero : out STD_LOGIC;
           Carry : out STD_LOGIC);
end component;

signal A, S : STD_LOGIC_VECTOR (3 downto 0);
signal Clk: std_logic := '0';
signal RegSel, Zero, Carry : STD_LOGIC;

begin
UUT : AU  port map(
        A => A,
        S => S,
        Clk => Clk,
```

```
        RegSel => RegSel,
        Zero => Zero,
        Carry => Carry);

process begin
    Clk <= not Clk;
    wait for 5ns;
end process;

process begin

    RegSel <= '1';
    A<= "1110";
    wait for 200ns;

    RegSel <= '0';
    A<= "0110";
    wait for 200ns;

    RegSel <= '1';
    A<= "1101";
    wait for 200ns;

    RegSel <= '0';
    A<= "0101";
    wait for 200ns;

    RegSel <= '1';
    A<= "0000";
    wait for 200ns;

    RegSel <= '0';
    A<= "1111";
    wait for 200ns;

end process;
end Behavioral;
```

- **Time diagram**



- **Implemented design schematic**



- **Constraints file**

```
## Clock signal
set_property PACKAGE_PIN W5 [get_ports Clk]
     set_property IOSTANDARD LVCMOS33 [get_ports Clk]
     create_clock -add -name sys_clk_pin -period 10.00 -waveform {0 5} [get_ports Clk]

## Switches
set_property PACKAGE_PIN V17 [get_ports {A[0]}]
     set_property IOSTANDARD LVCMOS33 [get_ports {A[0]}]
set_property PACKAGE_PIN V16 [get_ports {A[1]}]
     set_property IOSTANDARD LVCMOS33 [get_ports {A[1]}]
set_property PACKAGE_PIN W16 [get_ports {A[2]}]
     set_property IOSTANDARD LVCMOS33 [get_ports {A[2]}]
set_property PACKAGE_PIN W17 [get_ports {A[3]}]
     set_property IOSTANDARD LVCMOS33 [get_ports {A[3]}]
set_property PACKAGE_PIN R2 [get_ports {RegSel}]
     set_property IOSTANDARD LVCMOS33 [get_ports {RegSel}]

## LEDs
set_property PACKAGE_PIN U16 [get_ports {S[0]}]
     set_property IOSTANDARD LVCMOS33 [get_ports {S[0]}]
set_property PACKAGE_PIN E19 [get_ports {S[1]}]
     set_property IOSTANDARD LVCMOS33 [get_ports {S[1]}]
set_property PACKAGE_PIN U19 [get_ports {S[2]}]
     set_property IOSTANDARD LVCMOS33 [get_ports {S[2]}]
set_property PACKAGE_PIN V19 [get_ports {S[3]}]
     set_property IOSTANDARD LVCMOS33 [get_ports {S[3]}]
set_property PACKAGE_PIN P1 [get_ports {Carry}]
     set_property IOSTANDARD LVCMOS33 [get_ports {Carry}]
set_property PACKAGE_PIN L1 [get_ports {Zero}]
     set_property IOSTANDARD LVCMOS33 [get_ports {Zero}]
```

Conclusion

A 4-bit Arithmetic Unit can be created using small components, which are a 1-to-2 decoder, two registers (D flip-flops), and a RCA.

We have identified the following components of an AU:

Zero flag – helps in conditional branching.

Registers – use to store the data.

RegSel input (implemented using a 1-to-2 decoder) - selects which register to load.

Also, this lab concludes with the introduction of how the processor does the arithmetic operation.