Rodrigo Becerril Ferreyra

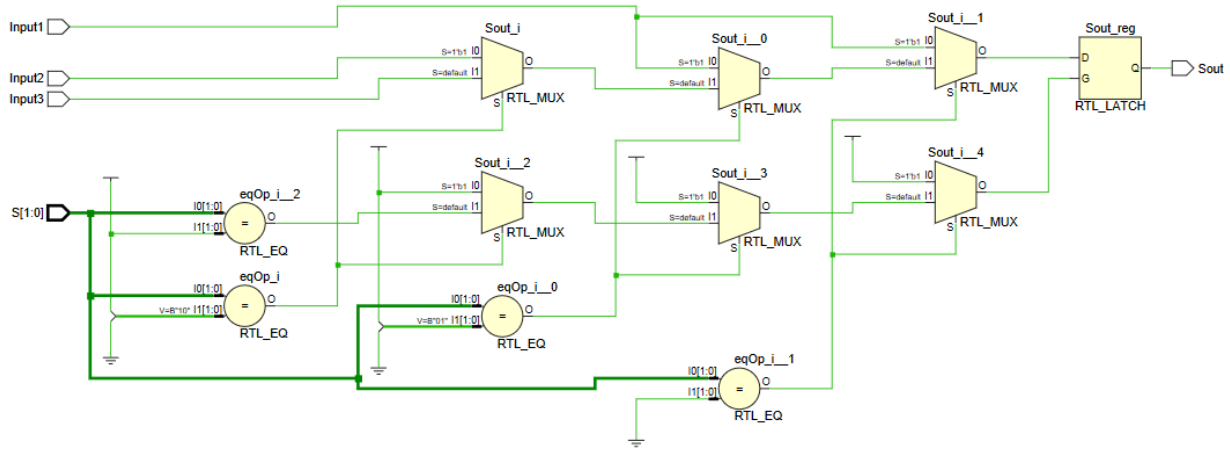
Dr. Gevik Sardarbegian

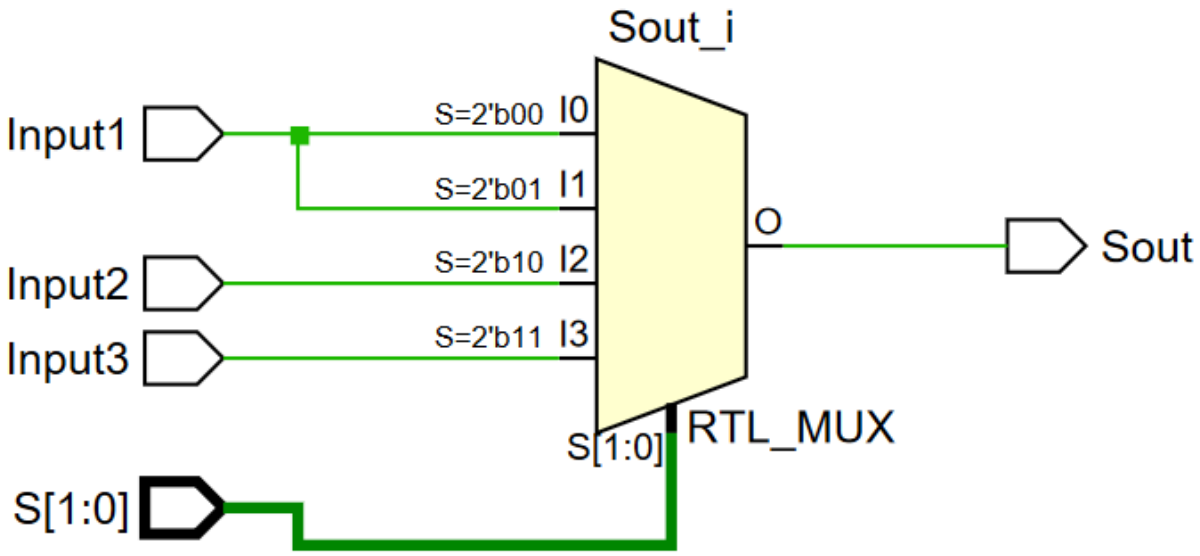
CECS 440 Section 02

26 October 2021

Lab 4

This lab tasked us with creating a 16-bit Arithmetic Logic Unit (ALU) from 16 different 1-bit ALUs that can add, subtract, OR, or AND two bits. Both ALU modules were verified with the code I provided below. The full adder module was verified last lab. I did not verify the multiplexer, AND gate, or OR gate modules because they are very simple. To verify these, I simply looked at the block diagram that was created using the Elaborated Design feature of Vivado. Note that I performed modifications on the multiplexer module, because the block diagram looked very off, and I could not guarantee that it works as intended. Additionally, your version uses RTL_LATCHES. These latches should be avoided at all costs, unless the designer explicitly creates one. Latches mean that there is likely a case statement that does not consider all options. Here are the block diagram for your mux module and mine:





Please note that high-quality vector .pdf files of these images are available in the Lab submission. The following source files are also available there.

full_adder.vhd

```
----------------------------------------------------------------------------------
-- Company: California State University, Long Beach
-- Engineer: Rodrigo Becerril Ferreyra
--
-- Create Date: 10/12/2021 11:54:07 PM
-- Design Name: full_adder
-- Module Name: full_adder
-- Project Name: CECS 440 Lab 3
-- Description: A simple 1-bit wide full adder.
--
-- Dependencies:
--
-- Revision: 1
----------------------------------------------------------------------------------

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.NUMERIC_STD.ALL;

entity full_adder is
    port
    (
        a:    in  std_logic;
        b:    in  std_logic;
        cin:  in  std_logic;
        sum:  out std_logic;
        cout: out std_logic
    );
end full_adder;

architecture Behavioral of full_adder is
begin

    -- two simple assign statements for a one-bit adder
    sum <= a xor b xor cin;
    cout <= (a and b) or (b and cin) or (cin and a);

end Behavioral;
```

and_gate.vhd

```
Library ieee;
Use ieee.std_logic_1164.all;
Use ieee.std_logic_unsigned.all;

Entity and_gate is
port(
            In1          :     in    std_logic;
            In2          :     in    std_logic;
            Sout   :     out   std_logic
);
End;

Architecture behavior of and_gate is

Begin
-- In1  In2  Sout
--  0    0    0
--  0    1    0
--  1    0    0
--  1    1    1

-- and gate logic
      Sout <= In1 and In2;

End;
```

or_gate.vhd

```
----------------------------------------------------------------------------------
-- Company: California State University, Long Beach
-- Engineer: Rodrigo Becerril Ferreyra
--
-- Create Date: 10/25/2021 11:46:06 AM
-- Module Name: or_gate - Behavioral
-- Project Name: Lab 4
----------------------------------------------------------------------------------

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
--use IEEE.NUMERIC_STD.ALL;

entity or_gate is
    port
    (
        In1 : in  std_logic;
        In2 : in  std_logic;
        Sout: out std_logic
    );
end or_gate;

architecture Behavioral of or_gate is
begin

    Sout <= In1 or In2;

end Behavioral;
```

MUX31.vhd

```
----------------------------------------------------------------------------------
-- Company: California State University, Long Beach
-- Engineer: Rodrigo Becerril Ferreyra

-- Module Name: MUX31 - Behavioral
-- Project Name: Lab 4
-- Description: I improved the mux block (is now an actual mux).
--   The last version had an RTL_LATCH in it, which I hear is something to avoid
----------------------------------------------------------------------------------

library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;

entity MUX31 is
    -- define ports
    port
    (
        Input1: in  std_logic;
        Input2: in  std_logic;
        Input3: in  std_logic;
        S:      in  std_logic_vector(1 downto 0);
        Sout:   out std_logic
    );
end MUX31;

architecture Behavioral of MUX31 is
begin
    process (S, Input1, Input2, Input3)
    begin
        case S is
            when  "00"  => Sout <= Input1;
            when  "01"  => Sout <= Input1;
            when  "10"  => Sout <= Input2;
            when  "11"  => Sout <= Input3;
            when others => Sout <= 'Z';
        end case;
    end process;

end Behavioral;
```

ALU1bit.vhd

```
----------------------------------------------------
------------------------------
-- Company: California State University, Long Beach
-- Engineer: Rodrigo Becerril Ferreyra
--
-- Create Date: 10/25/2021 11:31:08 AM
-- Module Name: ALU1bit - Behavioral
-- Project Name: Lab 4
----------------------------------------------------
------------------------------

library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
use IEEE.NUMERIC_STD.ALL;

-- assign top-level ports
entity ALU1bit is
    port(
            A       :   in  std_logic;
            B       :   in  std_logic;
            Cin     :   in  std_logic;
            S       :   in  std_logic_vector(1
downto 0);
            Sout    :   out std_logic;
            Cout    :   out std_logic
    );
end ALU1bit;

Architecture behavior of ALU1bit is
    -- intstantiate full_adder module
    component full_adder
        port
        (
            a       :   in  std_logic;
            b       :   in  std_logic;
            cin     :   in  std_logic;
            sum     :   out std_logic;
            cout    :   out std_logic
        );
    end component;

    -- intstantiate MUX31 module
    component MUX31
        port
        (
            Input1  :   in  std_logic;
            Input2  :   in  std_logic;
            Input3  :   in  std_logic;
            S       :   in  std_logic_vector(1
downto 0);
            Sout    :   out std_logic
        );
    end component;

    -- intstantiate and_gate module
    component and_gate
        port
        (
            In1     :   in  std_logic;
            In2     :   in  std_logic;
            Sout    :   out std_logic
        );
    end component;

    -- intstantiate or_gate module
    component or_gate
    port
    (
        In1     :   in  std_logic;
        In2     :   in  std_logic;
        Sout    :   out std_logic
    );
    end component;

    -- signal definitions
    signal Sout_full_adder: std_logic;
    signal Sout_and_gate  : std_logic;
    signal Sout_or_gate   : std_logic;
    signal Xor_out        : std_logic;
    signal Cin_muxed      : std_logic;

begin

    -- assign statements
    -- flips B if S(0) is 1
    Xor_out <= B xor S(0);
    -- If S(0) is set (subtract mode), then a 1
should always go into the
    --   Cin port on the full adder (two's
compliment is flip all bits
    --   then add 1). If S(0) is cleared, then
simply pass Cin into
    --   the Cin port in full_adder.
    --with S(0) select Cin_muxed <= '1' when '1',
Cin when '0', 'Z' when others;

    fulladderblock: full_adder
    port map
    (
        a => A,
        b => Xor_out,
        cin => Cin,
        sum => Sout_full_adder,
        cout => Cout
    );

    andgateblock: and_gate
    port map
    (
        In1 => A,
        In2 => B,
        Sout => Sout_and_gate
    );

    orgateblock: or_gate
    port map
    (
        In1 => A,
        In2 => B,
        Sout => Sout_or_gate
    );

    muxblock: MUX31
    port map
    (
        Input1 => Sout_full_adder,
        Input2 => Sout_and_gate,
        Input3 => Sout_or_gate,
        S => S,
        Sout => Sout
    );

end;
```

ALU16bit.vhd

```
----------------------------------------
----------------------------------------
----
-- Company: California State
University, Long Beach
-- Engineer: Rodrigo Becerril Ferreyra
--
-- Create Date: 10/25/2021 01:23:54 PM
-- Module Name: ALU16bit - Behavioral
-- Project Name: Lab 4
----------------------------------------
----------------------------------------
----

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
--use IEEE.NUMERIC_STD.ALL;

-- define top-level ports for ALU16bit
entity ALU16bit is
    port
    (
        A:    in  std_logic_vector(15
downto 0);
        B:    in  std_logic_vector(15
downto 0);
        S:    in  std_logic_vector(1
downto 0);
        Cout: out std_logic;
        Sout: out std_logic_vector(15
downto 0)
    );
end ALU16bit;

architecture Behavioral of ALU16bit is
    -- ALU1bit declaration
    component ALU1bit
        port(
                A       :   in
std_logic;
                B       :   in
std_logic;
                Cin     :   in
std_logic;
                S       :   in
std_logic_vector(1 downto 0);
                Sout    :   out
std_logic;
                Cout    :   out
std_logic
        );
    end component;

    -- cin-cout inbetween wires
    signal inbetween:
std_logic_vector(14 downto 0);
begin

    Sum0: ALU1bit
    port map
    (
        A => A(0),
        B => B(0),
        Cin => S(0),
        S => S,
        Cout => inbetween(0),
        Sout => Sout(0)
    );

    Sum1: ALU1bit
    port map
    (
        A => A(1),
        B => B(1),
        Cin => inbetween(0),
        S => S,
        Cout => inbetween(1),
        Sout => Sout(1)
    );

    Sum2: ALU1bit
    port map
    (
        A => A(2),
        B => B(2),
        Cin => inbetween(1),
        S => S,
        Cout => inbetween(2),
        Sout => Sout(2)
    );

    Sum3: ALU1bit
    port map
    (
        A => A(3),
        B => B(3),
        Cin => inbetween(2),
        S => S,
        Cout => inbetween(3),
        Sout => Sout(3)
    );

    Sum4: ALU1bit
    port map
    (
        A => A(4),
        B => B(4),
        Cin => inbetween(3),
        S => S,
        Cout => inbetween(4),
        Sout => Sout(4)
    );

    Sum5: ALU1bit
    port map
    (
        A => A(5),
        B => B(5),
        Cin => inbetween(4),
        S => S,
        Cout => inbetween(5),
        Sout => Sout(5)
    );

    Sum6: ALU1bit
    port map
    (
        A => A(6),
        B => B(6),
        Cin => inbetween(5),
        S => S,
        Cout => inbetween(6),
        Sout => Sout(6)
    );

    Sum7: ALU1bit
    port map
    (
        A => A(7),
        B => B(7),
        Cin => inbetween(6),
        S => S,
        Cout => inbetween(7),
        Sout => Sout(7)
    );

    Sum8: ALU1bit
    port map
    (
        A => A(8),
        B => B(8),
        Cin => inbetween(7),
        S => S,
        Cout => inbetween(8),
        Sout => Sout(8)
    );

    Sum9: ALU1bit
    port map
    (
        A => A(9),
        B => B(9),
        Cin => inbetween(8),
        S => S,
        Cout => inbetween(9),
        Sout => Sout(9)
    );

    Sum10: ALU1bit
    port map
    (
        A => A(10),
        B => B(10),
        Cin => inbetween(9),
        S => S,
        Cout => inbetween(10),
        Sout => Sout(10)
    );

    Sum11: ALU1bit
    port map
    (
        A => A(11),
        B => B(11),
        Cin => inbetween(10),
        S => S,
        Cout => inbetween(11),
        Sout => Sout(11)
    );

    Sum12: ALU1bit
    port map
    (
        A => A(12),
        B => B(12),
        Cin => inbetween(11),
        S => S,
        Cout => inbetween(12),
        Sout => Sout(12)
    );

    Sum13: ALU1bit
    port map
    (
        A => A(13),
        B => B(13),
        Cin => inbetween(12),
        S => S,
        Cout => inbetween(13),
        Sout => Sout(13)
    );

    Sum14: ALU1bit
    port map
    (
        A => A(14),
        B => B(14),
        Cin => inbetween(13),
        S => S,
        Cout => inbetween(14),
        Sout => Sout(14)
    );

    Sum15: ALU1bit
    port map
    (
        A => A(15),
        B => B(15),
        Cin => inbetween(14),
        S => S,
        Cout => Cout,
        Sout => Sout(15)
    );

end Behavioral;
```

ALU1bit_tb.vhd

```
----------------------------------------
----------------------------------------
-- Company: California State University,
Long Beach
-- Engineer: Rodrigo Becerril Ferreyra
--
-- Create Date: 10/25/2021 12:37:32 PM
-- Module Name: ALU1bit_tb - Behavioral
-- Project Name: Lab 4
----------------------------------------
----------------------------------------

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
--use IEEE.NUMERIC_STD.ALL;

entity ALU1bit_tb is
end ALU1bit_tb;

architecture Behavioral of ALU1bit_tb is

    -- declare unit under test
    component ALU1bit
        port
        (
            A       :   in  std_logic;
            B       :   in  std_logic;
            Cin     :   in  std_logic;
            S       :   in
std_logic_vector(1 downto 0);
            Sout    :   out std_logic;
            Cout    :   out std_logic
        );
    end component;

    -- declare signals
    signal A_tb: std_logic := '0';
    signal B_tb: std_logic := '0';
    signal Cin_tb: std_logic := '0';
    signal S_tb: std_logic_vector(1
downto 0) := "00";
    signal Sout_tb: std_logic;
    signal Cout_tb: std_logic;
    constant tick: time := 100 ns;

begin

    -- instantiate ALU1bit as uut
    uut: ALU1bit
    port map
    (
        A => A_tb,
        B => B_tb,
        Cin => Cin_tb,
        S => S_tb,
        Sout => Sout_tb,
        Cout => Cout_tb
    );

    -- process to change A
    process
    begin
        wait for tick;
        A_tb <= not A_tb;
    end process;

    -- process to change B
    process
    begin
        wait for (tick*2);
        B_tb <= not B_tb;
    end process;

    -- process to change Cin
    process
    begin
        wait for (tick*4);
        Cin_tb <= not Cin_tb;
    end process;

    -- process to change S(0)
    process
    begin
        wait for (tick*8);
        S_tb(0) <= not S_tb(0);
    end process;

    -- process to change S(1)
    process
    begin
        wait for (tick*16);
        S_tb(1) <= not S_tb(1);
    end process;

end Behavioral;
```

ALU16bit_tb.vhd

Results:

S1	S0	A	B	Sout	Cout
0	0	150	260	410	0
0	0	-25	65	40	1
0	0	2	2	4	0
0	1	550	320	230	1
0	1	25	60	-35	0
0	1	-2	16	-18	1
1	0	64	256	0	0
1	0	2000	3	0	0
1	0	5342	968	218	0
1	1	12	7	15	1
1	1	0	2345	2345	0
1	1	-1	5	-1	1