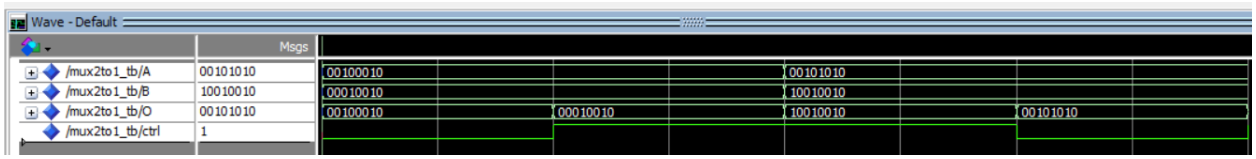
Case Study

Bagian 1

1. Buatlah entity multiplexer 2 to 1. Entity tersebut akan memiliki pin data in A, B yang masing-masing berukuran 8 bit, pin control berukuran 1 bit dan pin output O berukuran 8 bit.

```
library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.numeric_std.all;

entity mux2to1 is
    port (
        A, B : IN std_logic_vector (7 downto 0);
        control : IN std_logic;
        O : OUT std_logic_vector (7 downto 0)
    );
end entity mux2to1;

architecture rtl of mux2to1 is

begin

    mux: process(A, B, control)
    begin
        if control = '0' then
            O <= A;
        else
            O <= B;
        end if;
    end process mux;

end architecture rtl;
```

2. Simulasikanlah rangkaian multiplexer yang anda buat menggunakan modelsim, sertakan screenshot.



Bagian 2

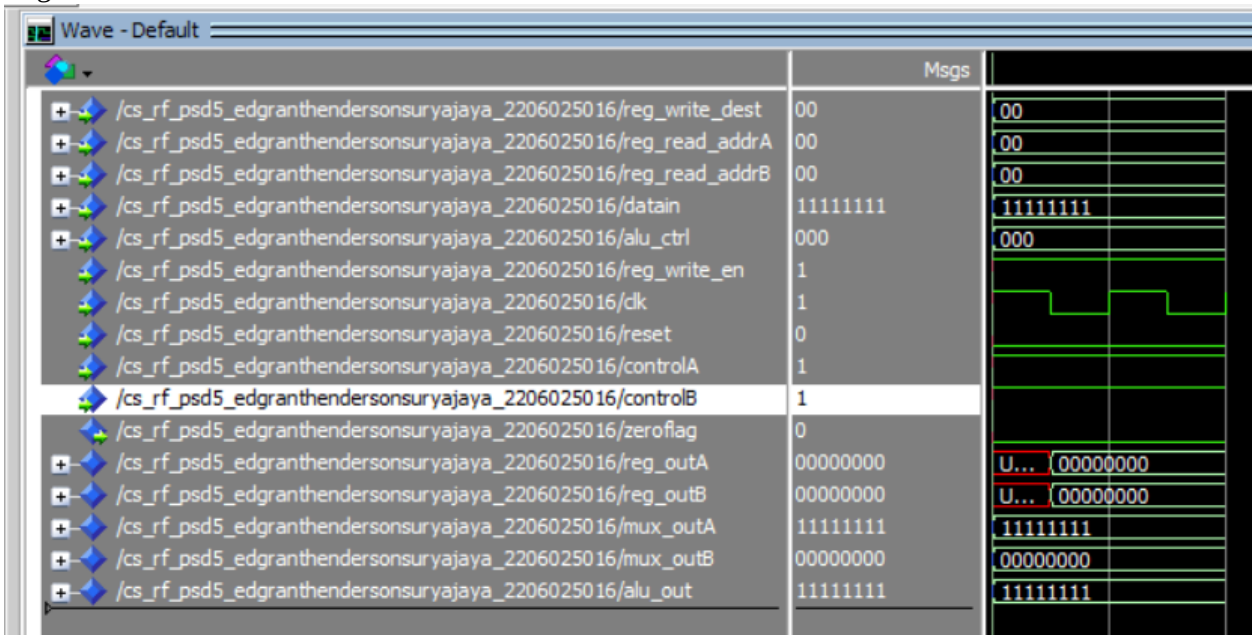
1. Kode

```
library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.numeric_std.all;

entity CS_RF_PSD5_EdgrantHendersonSuryajaya_2206025016 is
    port (
        -- input
        reg_write_dest : IN std_logic_vector (1 downto 0);
        reg_read_addrA, reg_read_addrB : IN std_logic_vector (1 downto 0);
        datain : IN std_logic_vector (7 downto 0);
        alu_ctrl : IN std_logic_vector (2 downto 0);

        -- other input
        reg_write_en : IN std_logic;
        clk, reset : IN std_logic;
        controlA, controlB : IN std_logic;
        zeroflag : OUT std_logic
    );
end entity CS_RF_PSD5_EdgrantHendersonSuryajaya_2206025016;

architecture rtl of CS_RF_PSD5_EdgrantHendersonSuryajaya_2206025016 is
    component register_ED is
        port (
            clk, reset      : IN std_logic;
            reg_write_en    : IN std_logic := '0';
            reg_write_dest  : IN std_logic_vector (1 downto 0) := (others => '0');
            reg_write_data  : IN std_logic_vector (7 downto 0) := (others => '0');
            reg_read_addrA  : IN std_logic_vector (1 downto 0) := (others => '0');
            reg_read_addrB  : IN std_logic_vector (1 downto 0) := (others => '0');

            reg_read_dataA  : OUT std_logic_vector (7 downto 0);
            reg_read_dataB  : OUT std_logic_vector (7 downto 0)
        );
    end component register_ED;

    component mux2to1 is
        port (
            A, B : IN std_logic_vector (7 downto 0);
            control : IN std_logic;
            O : OUT std_logic_vector (7 downto 0)
        );
    end component mux2to1;

    component alu_ED is
        port (
            a, b        : IN std_logic_vector (7 downto 0);
            alu_control : IN std_logic_vector (2 downto 0);
            alu_result  : OUT std_logic_vector (7 downto 0);
            zeroflag    : OUT std_logic
        );
    end component alu_ED;

    signal reg_outA, reg_outB : std_logic_vector (7 downto 0);
    signal mux_outA, mux_outB : std_logic_vector (7 downto 0);
    signal alu_out : std_logic_vector (7 downto 0);
begin
```

```
    REG: register_ED port map (
        clk => clk,
        reset => '0',
        reg_write_en => reg_write_en,
        reg_write_dest => reg_write_dest,
        reg_write_data => alu_out,
        reg_read_addrA => reg_read_addrA,
        reg_read_addrB => reg_read_addrB,

        reg_read_dataA => reg_outA,
        reg_read_dataB => reg_outB
    );

    MUXA: mux2to1 port map (
        A => reg_outA,
        B => datain,
        control => controlA,
        O => mux_outA
    );
    MUXB: mux2to1 port map (
        A => reg_outB,
        B => "00000000",
        control => controlB,
        O => mux_outB
    );
    ALU: alu_ED port map (
        mux_outA,
        mux_outB,
        alu_ctrl,

        alu_out,
        zeroflag
    );

    -- initialize register jadi 0
    initialize: process(clk, reset)
    begin
        if reset = '1' then
            reg_outA <= "00000000";
            reg_outB <= "00000000";
        elsif falling_edge(clk) then
            if (reg_outA = "UUUUUUUU") then
                reg_outA <= "00000000";
            end if;
            if (reg_outB = "UUUUUUUU") then
                reg_outB <= "00000000";
            end if;
        end if;
    end process initialize;
end architecture rtl;
```

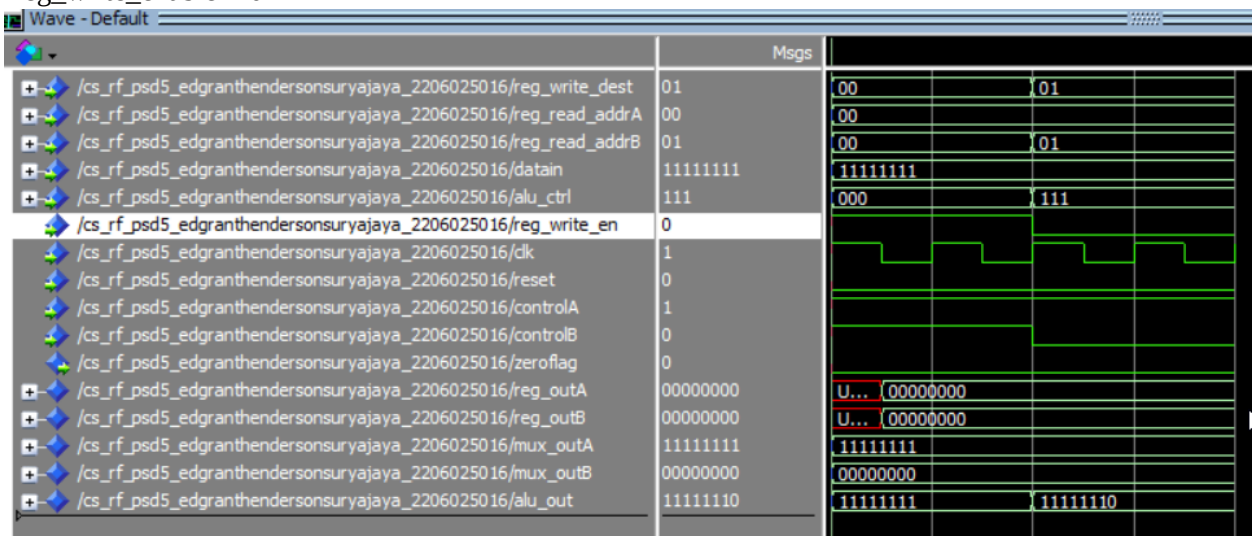
2. Simulasikanlah rangkaian yang telah anda buat dengan input sebagai berikut

a. Alu_ctrl = 000
Mux_ctrl_A = 1
Mux_ctrl_B = 1
Reg_write_dest = 00
Reg_read_addr = 00
Reg_read_addr = 00
Reg_write_enable = 1



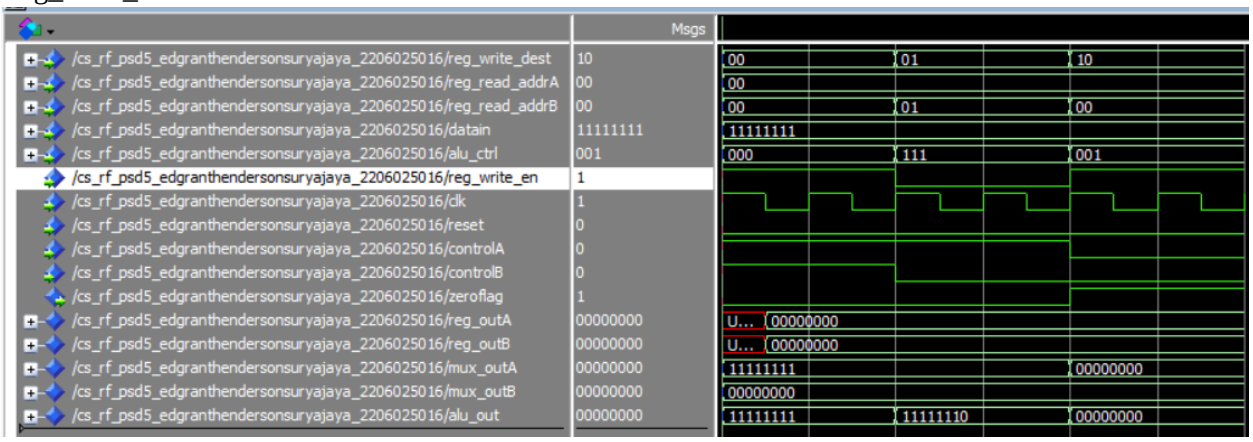
dataIn tersimpan di MUX out A. Itu menjadi salah satu operand ALU, jadinya ALU menambahkan dataIn dengan 0, menghasilkan data IN yang disimpen di adress 00

b. Alu_ctrl = 111
Mux_ctrl_A = 1
Mux_ctrl_B = 0
Reg_write_dest = 01
Reg_read_addr = 00
Reg_read_addr = 01
Reg_write_enable = 0



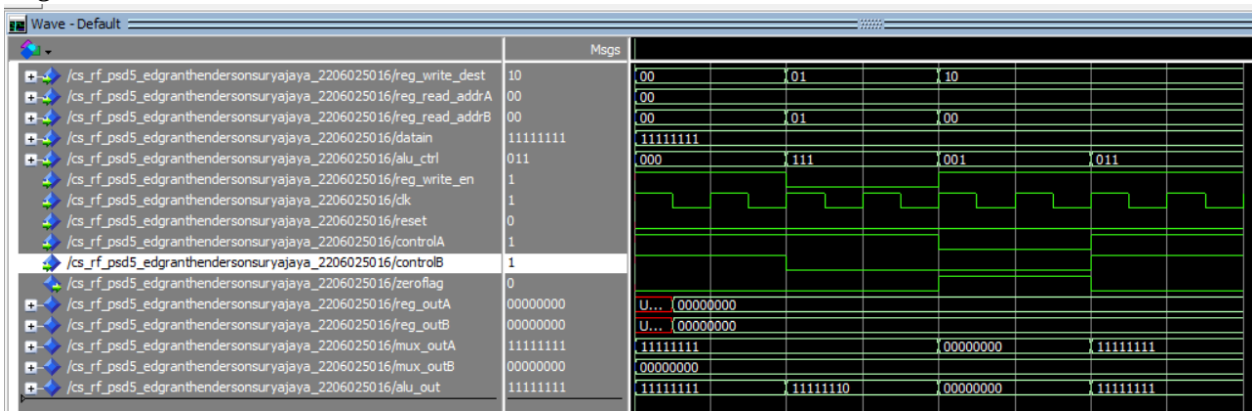
Operasi yang dilakukan adalah decrement, data ALU berasal dari register B dan dataIn. Register menyimpan pada address 01, tetapi karena write enablenya 0, data tidak bisa ditulis jadinya 01 masih kosong.

c. Alu_ctrl = 001
Mux_ctrl_A = 0
Mux_ctrl_B = 0
Reg_write_dest = 10
Reg_read_addr = 00
Reg_read_addr = 00
Reg_write_enable = 1



Operasi apakah yang dilakukan adalah pengurangan, data ALU berasal dari address 00 yang berisi 11111111, karena kedua nilai sama, jadi hasilnya 00000000. Perhitungan disimpan pada address 10

d. Alu_ctrl = 011
Mux_ctrl_A = 1
Mux_ctrl_B = 1
Reg_write_dest = 10
Reg_read_addr = 00
Reg_read_addr = 00
Reg_write_enable = 1



Operasi yang dilakukan adalah OR, data berasal DataIN dan 00000000 karena control mux adalah 1 untuk kedua mux. Oleh karena itu 00000000 OR 11111111 hasilnya adalah 11111111. Perhitungan disimpan pada address 00.

3. Cantumkanlah hasil sintesis dari rangkaian yang anda buat (Soal Bonus)
Quartus gak mau compile

4. Berilah analisis terhadap percobaan yang telah anda lakukan

pada program ini kita mengunakan structural style dengan cara menggunakan component dari entity lain. Menggunakan ALU, MUX, dan Register untuk membuat computer sederhana. Terdapat masalah ALU saya tidak bekerja, ternyata ALU saya belum dicompile, setelah dicompile semua rangkaian berjalan dengan baik. Saya juga berinisiatif menambahkan inisialisasi bebeara signal, untuk tidak menimbulkan undifined behaviour

5. Berilah kesimpulan terhadap percobaan yang telah anda lakukan

- Structural style menggunakan blok program lain untuk membaut program secara modular
- Semua vhdl harus dicompile, bukan hanya parrentnya
- Quartus jelek

Kerjain bagian 1 nya jangan lama-lama ya.., bagian 2 bakal butuh waktu lama