

## Akram\_10\_6\_2021\_Lab\_5\_Integration

Wednesday, October 13, 2021 2:56 PM

```

library IEEE;
use IEEE.STD.TEXT.all;
use IEEE.STD.TEXT.TEXT.all;
use IEEE.STD.TEXT.TEXT.all;
entity Akram_10_13_2021_N_Bit_Add_Sub_Flags is
    generic (n : integer := 32);
    port (A, B : in std_logic_vector (n-1 downto 0);
          Op : in std_logic; -- Operator
          C : out std_logic_vector (n-1 downto 0);
          CF, Ovf, Negf, Zerof : inout std_logic); -- carry flag, overflow, Negative, zero
end entity;
architecture BEHAVIORAL of Akram_10_13_2021_N_Bit_Add_Sub_Flags is
begin
    process (A, B)
        begin
            for i in 0 to n-1 loop
                Bx(i) <= B(i) xor Op;
            end loop;
            C(0) <- Op;
            Sum <- A xor Bx var C(n-1 downto 0);
            Temp_Sum <- (A and Bx) or ((n-1 downto 0) and (A xor Bx));
            Ovf <- C(n-1) xor C(0); -- overflow flag
            Negf <- Temp_Sum(n-1) and (Temp_Sum(0)); -- negative flag
            Zerof <- not (or_reduce (Temp_Sum)) - zero_flag;
        end process;
    end architecture;
end;

```

Task List:

- Compile Design (0.0)
- Analysis & Synthesis (0.0)
- Filter (Place & Route) (0.0)
- Assembler (Generate programming files) (0.0)
- TimeQuest Timing Analysis (0.0)
- EDA Netlist Writer
- Edit Settings
- Program Device (Open Programmer)

Messages:

- Generated archive 'C:/Users/Izzy/OneDrive/College/CSC342\_3 Computer Organization/Labs/Lab 5/Akram\_10\_13\_2021\_Lab\_5\_Integration.qar'
- Generated report 'Akram\_10\_13\_2021\_Lab\_5\_Integration.archive.rpt'
- 23030 evaluation of tcl script c:/Altera/lite/16.0/quartus/common/tcl/apps/qpm/qar.tcl was successful.
- Quartus Prime Shell was successful. 0 errors, 0 warnings

Fig 1. Using my adder from previous lab (Sept 19)

```

library IEEE;
use IEEE.STD.TEXT.all;
use altera_mf.alteramf_components.all;
entity Akram_10_13_2021_Lab_5_Integration is
    port (
        Akram_d0 : in STD_LOGIC; -- Operator
        Akram_d1 : in STD_LOGIC := '1';
        Akram_d2 : in STD_LOGIC_VECTOR (31 DOWNTO 0); -- data-write
        Akram_daddress_a : in STD_LOGIC_VECTOR (4 DOWNTO 0); -- read1
        Akram_daddress_b : in STD_LOGIC_VECTOR (4 DOWNTO 0); -- read2
        Akram_dwrite : in STD_LOGIC_VECTOR (4 DOWNTO 0); -- write
        Akram_dmen : in STD_LOGIC_VECTOR (31 DOWNTO 0); -- data1
        q_a : out STD_LOGIC_VECTOR (31 DOWNTO 0); -- data1
        q_b : out STD_LOGIC_VECTOR (31 DOWNTO 0); -- data2
        q_m : out STD_LOGIC_VECTOR (31 DOWNTO 0); -- data2
        rd : out STD_LOGIC_VECTOR (31 DOWNTO 0); -- data2
    );
end entity;
architecture SYN of Akram_10_13_2021_Lab_5_Integration is
begin
    sub_wire0 : STD_LOGIC_VECTOR (31 DOWNTO 0) := q_a;
    sub_wire1 : STD_LOGIC_VECTOR (31 DOWNTO 0) := q_b;
    sub_wire2 : STD_LOGIC_VECTOR (31 DOWNTO 0) := q_m;
    signal signal_cf, signal_ovf, signal_negf, signal_zerof : STD_LOGIC;
    component Akram_9_19_2021_N_Bit_Add_Sub_Flags is -- add/sub unit
        generic (n : integer := 32);
        port (A, B : in std_logic_vector (n-1 downto 0);
              sum : out std_logic_vector (n-1 downto 0);
              CF, Ovf, Negf, Zerof : inout std_logic); -- carry flag, overflow, Negative, zero
    end component;
    begin
        sub_wire0 <= sub_wire1 & sub_wire2;
        sub_wire1 <= Akram_d0 & Akram_d1;
        Akram_d2 <= signal_zerof;
        altSyncram_component : altSyncram
            generic map (numReadPorts => 32,
                         numWritePorts => 1,
                         outputPort => "DYNAMIC_PORT",
                         outdataAcLR_b => "NONE",
                         outdataAcLR_a => "NONE",
                         powerUpReset => "PWRUPRESET",
                         powerUpUninitialized => "FALSE",
                         ramControlType => "HIF",
                         initFile => "Akram_10_13_2021_Lab_5_Integration.mif", -- HIF
                         ipm_type => "altSyncram",
                         numWritePorts => 32,
                         widthA => 32,
                         widthB => 32,
                         widthM => 32,
                         numByteenaA => 1
            )
            port map (
                address_a => Akram_daddress_a,
                address_b => Akram_daddress_b,
                clock0 => Akram_clock,
                clock1 => Akram_clock
            );
    end;
end;

```

Task List:

- Compile Design (0.0)
- Analysis & Synthesis (0.0)
- Filter (Place & Route) (0.0)
- Assembler (Generate programming files) (0.0)
- TimeQuest Timing Analysis (0.0)
- EDA Netlist Writer
- Edit Settings
- Program Device (Open Programmer)

Messages:

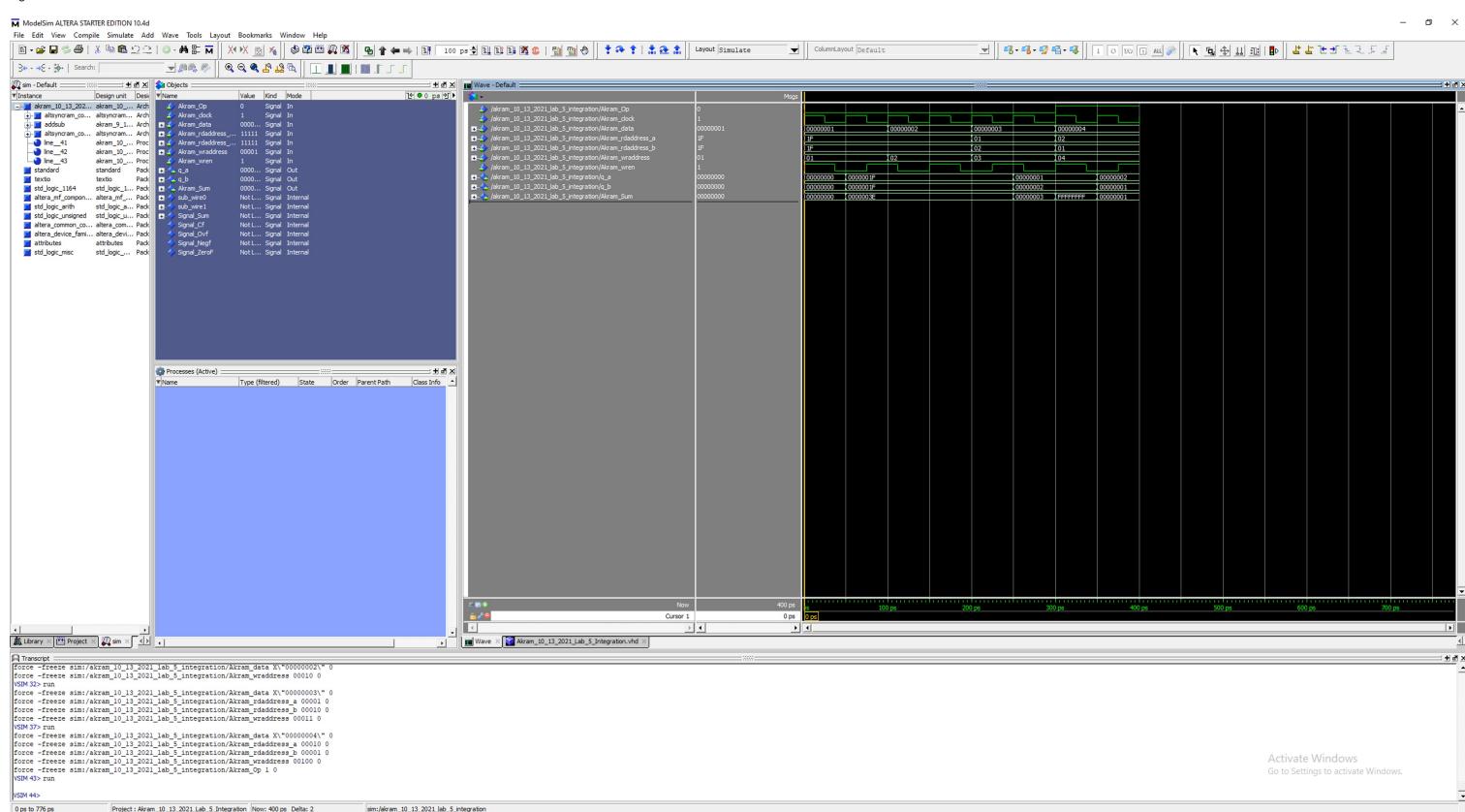
- 332140 No Removal paths to report
- 332148 Timing Requirements not met
- 332146 worst-case min/max pulse-width slack is -3.174
- 332105 Delays 100% fully constrained for setup requirements
- 332102 Delays 100% not fully constrained for hold requirements
- 293000 Quartus Prime Timestep Timing Analyzer was successful. 0 errors, 6 warnings
- 293000 Quartus Prime Full Compilation was successful. 0 errors, 13 warnings

Fig 2. Modifying my 3-port to accommodate adder (integration), full code is in QAR file.

The screenshot shows the Quartus Prime software interface with the following details:

- Project Navigator:** Shows the project structure with files like "Lab 3/Akram\_9\_19\_2021\_N\_BT\_Add\_Sub\_Flags.vhd", "Akram\_10\_13\_2021\_lab\_5\_integration.mif", and "Akram\_10\_13\_2021\_lab\_5\_integration.vhd".
- File List:** Lists files such as "Akram\_10\_13\_2021\_lab\_5\_integration.vhd", "Completion Report - Akram\_10\_13\_2021\_lab\_5\_integration", and "Akram\_9\_19\_2021\_N\_BT\_Add\_Sub\_Flags.vhd".
- Code Editor:** Displays the VHDL code for "Akram\_10\_13\_2021\_lab\_5\_integration.vhd". The code defines a component "Akram\_9\_19\_2021\_N\_BT\_Add\_Sub\_Flags" with various ports and internal logic.
- Task Manager:** Shows the build status for tasks like "Compile", "Analysis & Synthesis", "Filter (Place & Route)", "Assembler (Generate programming file)", "TimeQuest Timing Analysis", "EDA Netlist Writer", and "Program Device (Open Programmer)".
- IP Catalog:** Lists available IP cores including Basic Functions, DSP, Interface Protocols, Memory Interfaces and Controllers, Processors and Peripherals, and University Program.

**Fig 2a.** Best of the VHDL code



**Fig 3. Waveform Simulation**