

Akram_10_15_2021_Lab_6_Register_File

Friday, October 15, 2021 5:34 PM

Register File Report,
Ismail Akram,
10/17/2021,
CSC343 Prof Gertner

Table of contents
Objective
Description
Screenshots
Simulation
Conclusion

Objective:

Goal: Add-Sub unit with Register file (3-port RAM)

Description of Specifications, and Functionality

- Instruction Register for our integrated 3-port RAM and N-Bit ADD/SUB

Screenshots of VHDL code

```

LIBRARY IEEE;
USE IEEE.STD_LOGIC.all;
USE IEEE.STD.TEXT.all;
USE STD.TEXT.ALTERA.all;
USE STD.TEXT.ALTERA_COMPONENTS.all;

ENTITY Akram_10_15_2021_Lab_6_Register_file IS
  PORT
    (
      -- Akram_Op : IN STD_LOGIC; -- Operator
      -- Akram_rd : IN STD_LOGIC; -- RD
      -- Akram_data : IN STD_LOGIC_VECTOR (31 DOWNTO 0); -- data-write
      -- Akram_lraddress : IN STD_LOGIC_VECTOR (4 DOWNTO 0);
      -- Akram_rraddress_A : IN STD_LOGIC_VECTOR (4 DOWNTO 0);
      -- Akram_rraddress_B : IN STD_LOGIC_VECTOR (4 DOWNTO 0);

      Akram_wren : IN STD_LOGIC; -- '1';
      Akram_rden : IN STD_LOGIC_VECTOR (31 DOWNTO 0); -- data1
      Q_b : OUT STD_LOGIC_VECTOR (31 DOWNTO 0); -- data2
      Akram_Sum_rd : OUT STD_LOGIC_VECTOR (31 DOWNTO 0); -- sum
      Akram_LR : IN STD_LOGIC_VECTOR (31 DOWNTO 0) -- Instruction register
    );
END ENTITY Akram_10_15_2021_Lab_6_Register_file;

ARCHITECTURE SYN OF Akram_10_15_2021_Lab_6_Register_file IS
  SIGNAL sub_wire0 : STD_LOGIC_VECTOR (31 DOWNTO 0); -- Q_A
  SIGNAL sub_wire1 : STD_LOGIC_VECTOR (31 DOWNTO 0); -- Q_B
  SIGNAL sub_wire2 : STD_LOGIC_VECTOR (31 DOWNTO 0); -- sum
  SIGNAL Signal_Cf, signal_Cov, signal_ZeroF : STD_LOGIC; -- Flags
  SIGNAL Akram_LR_OP : STD_LOGIC; -- IR Operator, Instruction Register Opcode Bits [26:31] = 000000 encode addition operation, 000001 encode subtraction operation
  SIGNAL Akram_LR_rd : STD_LOGIC_VECTOR (4 DOWNTO 0); -- Read Instruction Register Bits [16:20] = source index ranging from [00000] to [11111]
  SIGNAL Akram_LR_daddress_A : STD_LOGIC_VECTOR (4 DOWNTO 0); -- Read2, Instruction Register Bits [11:15] = source register index ranging from [00000] to [11111]
  SIGNAL Akram_LR_daddress_B : STD_LOGIC_VECTOR (4 DOWNTO 0); -- source register index ranging from [00000] to [11111]

  COMPONENT Akram_10_15_2021_N_bit_Add_Sub_flags IS -- add/sub unit
    GENERIC
      N : Integer := 32;
    PORT
      A : IN STD_LOGIC_VECTOR (N-1 DOWNTO 0);
      B : IN STD_LOGIC_VECTOR (N-1 DOWNTO 0);
      C : OUT STD_LOGIC_VECTOR (N-1 DOWNTO 0);
      CF, OF, Negf, Zerof : INOUT STD_LOGIC; -- carry flag, overflow, negative, zero
  END COMPONENT;

  COMPONENT Akram_Register_N_VHDL IS
    GENERIC
      N : Integer := 32;
    PORT
      AKRA_Clk : IN STD_LOGIC; -- clock --
      AKRA_wren : IN STD_LOGIC; -- write enable --
      AKRA_rden : IN STD_LOGIC; -- read enable --
      AKRA_data : IN STD_LOGIC_VECTOR (N-1 DOWNTO 0); -- data input --
      AKRA_sum : OUT STD_LOGIC_VECTOR (N-1 DOWNTO 0); -- output --
  END COMPONENT;

BEGIN
  sub_wire0 <= sub_wire1(31 DOWNTO 0); -- data1
  sub_wire1 <= sub_wire0(31 DOWNTO 0); -- data2
  Akram_Sum_rd <= Signal_Sum(31 DOWNTO 0);

  Akram_LR_OP <= AKRAM_LR(25 DOWNTO 23); -- 0: addition or 1: subtraction
  Akram_LR_rd <= AKRAM_LR(23 DOWNTO 19);
  Akram_LR_daddress_A <= Akram_LR(18 DOWNTO 14);
  Akram_LR_daddress_B <= Akram_LR(14 DOWNTO 10);
  Akram_LR <= "00000000000000000000000000000000"; -- allzero

  GENERIC MAP (
    address_act_A => "NONE",
    address_act_B => "CLOCK"
  );

```

Task Compilation Status

- Compile Design: 0.00s
- Analysis & Synthesis: 0.04s
- Filter (Pave & Route): 0.04s
- Assembler (Generate programming files): 0.00s
- TimeQuest Timing Analysis: 0.00s
- EDA Netlist Writer: 0.00s
- Edit Settings: 0.00s
- Program Device (Open Programmer): 0.00s

Messages

Type ID Message

- 332140 No recovery paths to report
- 332140 No removal paths to report
- 332140 No setup paths to report
- 332140 No hold paths to report
- 33202 Design is not fully constrained for setup requirements
- 33202 Design is not fully constrained for hold requirements
- Quartus Prime TimeQuest Timing Analyzer was successful. 0 errors, 6 warnings
- 2930000 quartus prime full compilation was successful. 0 errors, 88 warnings

Fig 1. Register File VHDL code (using 3-port RAM, N-Bit ADD/SUB and N-Bit Register as components). I've included very detailed comments and explanation in the code (seen in screenshot). Data is being fed in via .mif file (line 77 and 123).

Fig 1a. VHDL code cont

Fig. 1c VHDL code cont.

Quartus Prime Lite Edition - C:/Users/itz/OneDrive/Collage/CSC342_3 Computer Organization/Labs/Lab 6/Akram_10_15_2021_Lab_6_Register_File - Akram_10_15_2021_Lab_6_Register.vhd

```

1 library IEEE;
2 use IEEE.STD.TEXT.TEXT;
3 use IEEE.STD.TEXT.TEXT;
4 entity Akram_Register_N_VHDL is
5 generic(N : integer := 32);
6 port(Clk : in STD.TEXT.TEXT; -- clock --
7       Akra_Wren : in STD.TEXT.TEXT; -- write enable --
8       Akra_Chen : in STD.TEXT.TEXT; -- chip enable --
9       Akra_Din : in STD.TEXT.TEXT_VECTOR(N-1 downto 0); -- data input --
10      Akra_Q : out STD.TEXT.TEXT_VECTOR(N-1 downto 0); -- output --
11 end Akram_Register_N_VHDL;
12 architecture arch of Akram_Register_N_VHDL is
13 begin
14   storage : STD.TEXT.TEXT_VECTOR(N-1 downto 0);
15
16   process (Akra_Clk)
17   begin
18     if rising_edge(Akra_Clk) and Akra_Wren = "1"
19     then storage <= Akra_Data;
20   end process;
21
22   process (Akra_Din, Akra_Chen, storage)
23   begin
24     if Akra_Chen = "1"
25     then Akra_Q <= storage;
26     else
27       if Akra_Din >= others then Akra_Q <= others;
28     end if;
29   end process;
30
31   process (Akra_Wren, Akra_Chen, storage)
32   begin
33     if Akra_Wren = "1" and Akra_Chen = "1"
34     then Akra_Q <= storage;
35     else
36       if Akra_Wren = others then Akra_Q <= others;
37     end if;
38   end process;
39 end arch;

```

Tasks	Compilation
	Task
✓	> ▶ 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

Type ID Message

- 332140 No Recovery paths to report
- 332140 No Removal paths to report
- 332140 No Initalization/Writeback paths to report
- 332102 Design is not fully constrained for setup requirements
- 332102 Design is not fully constrained for hold requirements
- Quartus Prime TimeQuest Timing Analyzer was successful. 0 errors, 6 warnings
- 293000 Quartus Prime Full Compilation was successful. 0 errors, 88 warnings

System Processing (126)

Fig 2. N-Bit Register from prior lab (Lab 3), already explained. This will be used as a component for the Register File (this lab)

Quartus Prime Lite Edition - C:/Users/itz/OneDrive/Collage/CSC342_3 Computer Organization/Labs/Lab 6/Akram_10_15_2021_Lab_6_Register_File - Akram_10_15_2021_Lab_6_Register.vhd

```

1 library IEEE;
2 use IEEE.STD.TEXT.TEXT;
3 use IEEE.STD.TEXT.TEXT;
4 entity Akram_9_19_2021_N_8Bit_Add_Sub_Flags is
5 generic(N : integer := 32);
6 port(C, B : in STD.TEXT.TEXT_VECTOR(N-1 downto 0);
7       Op : in STD.TEXT.TEXT; -- operator --
8       CF, OFV, NegF, ZeroF : out STD.TEXT.TEXT); -- carry flag, overflow, negative, zero
9 end Akram_9_19_2021_N_8Bit_Add_Sub_Flags;
10
11 architecture BEHAV_NBASF of Akram_9_19_2021_N_8Bit_Add_Sub_Flags is
12 begin
13   process(Bx, B, Op)
14   begin
15     for i in 0 to n-1 loop
16       Bx(i) := B(i) xor Op;
17     end loop;
18   end process;
19
20   CF <= Op;
21   Sum := Bx(n-1);
22   Temp_Sum := C(n-1) xor C(n-1 downto 0);
23   C(n-1) <= (A and Bx) or ((n-1 downto 0) and (A xor Bx));
24   OFV <= C(n-1) xor C(n); -- overflow flag
25   NegF <= Temp_Sum(n-1); -- negative flag
26   ZeroF <= not(Or_reduce(Temp_Sum)); -- zero flag
27 end BEHAV_NBASF;

```

Tasks	Compilation
	Task
✓	> ▶ 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

Type ID Message

- 332140 No Recovery paths to report
- 332140 No Removal paths to report
- 332140 No Initalization/Writeback paths to report
- 332102 Design is not fully constrained for setup requirements
- 332102 Design is not fully constrained for hold requirements
- Quartus Prime TimeQuest Timing Analyzer was successful. 0 errors, 6 warnings
- 293000 Quartus Prime Full Compilation was successful. 0 errors, 88 warnings

System Processing (126)

Fig 3. N-Bit ADD/SUB, already verified by you. I'm using this as a component for this lab to pass my outputs from 3-port RAM into this, then operating (addition/subtraction) them out and writing the result into a third address Akram_sum_rd.

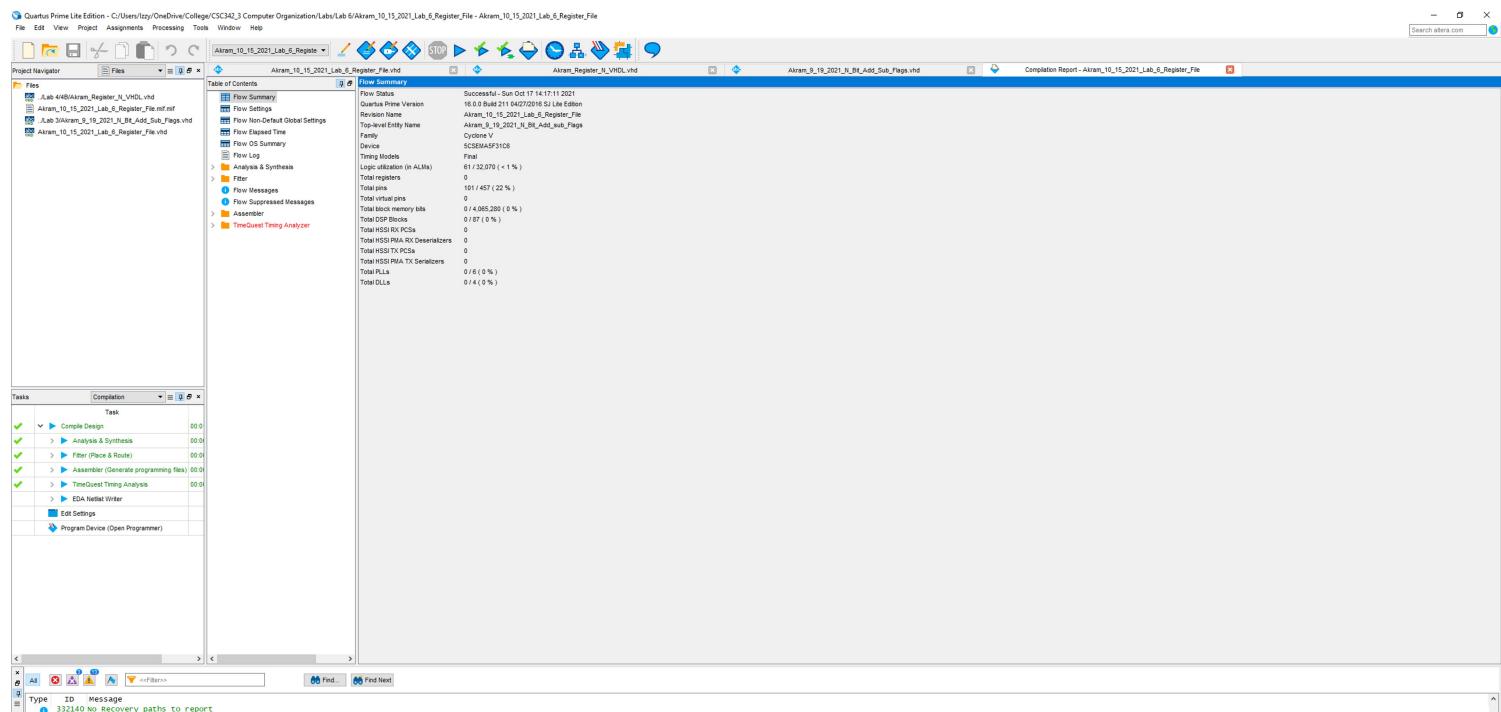


Fig 4. Compilation successful.

Simulation

ModelSim ALTERA STARTER EDITION 10.4d

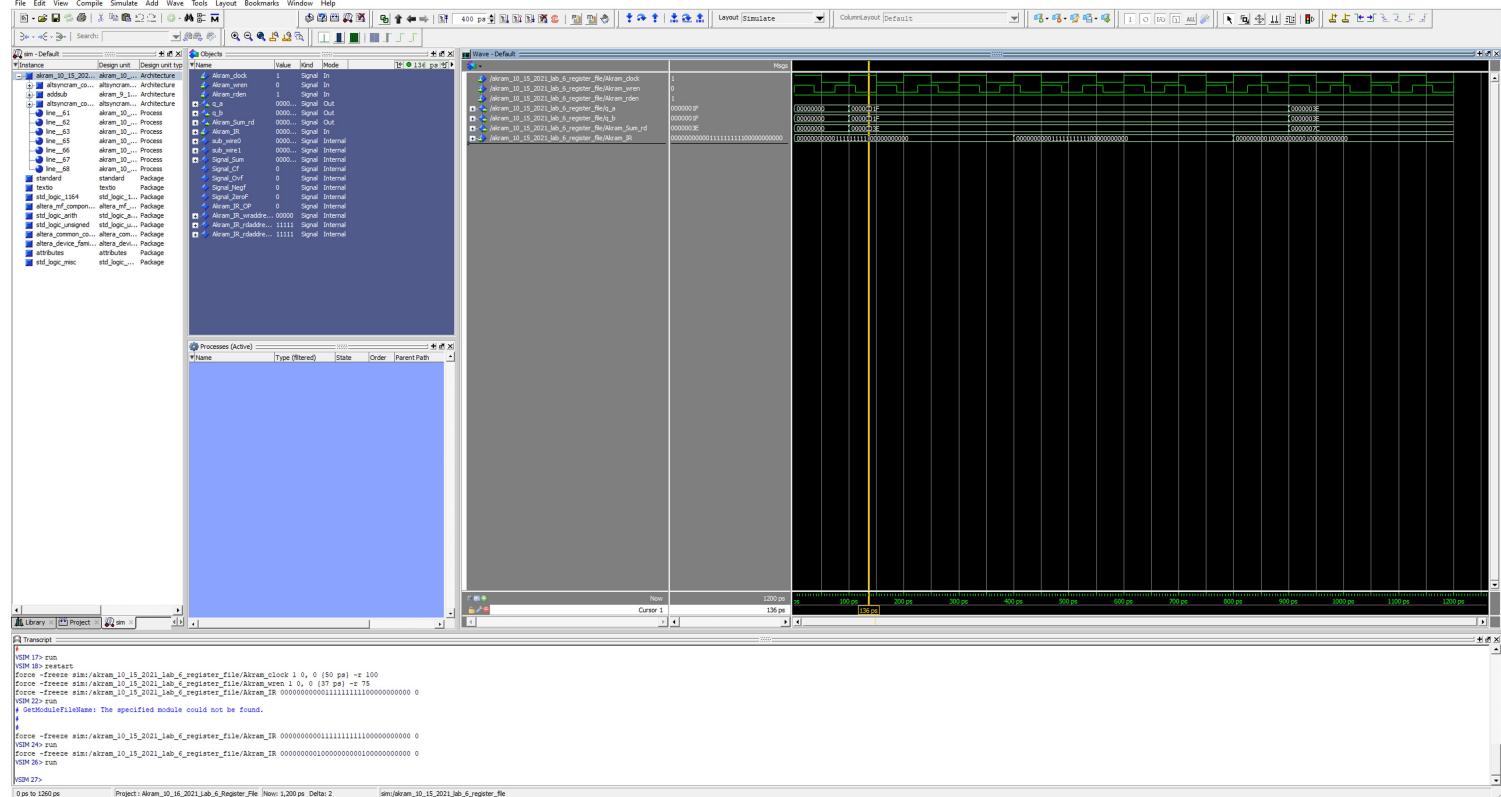
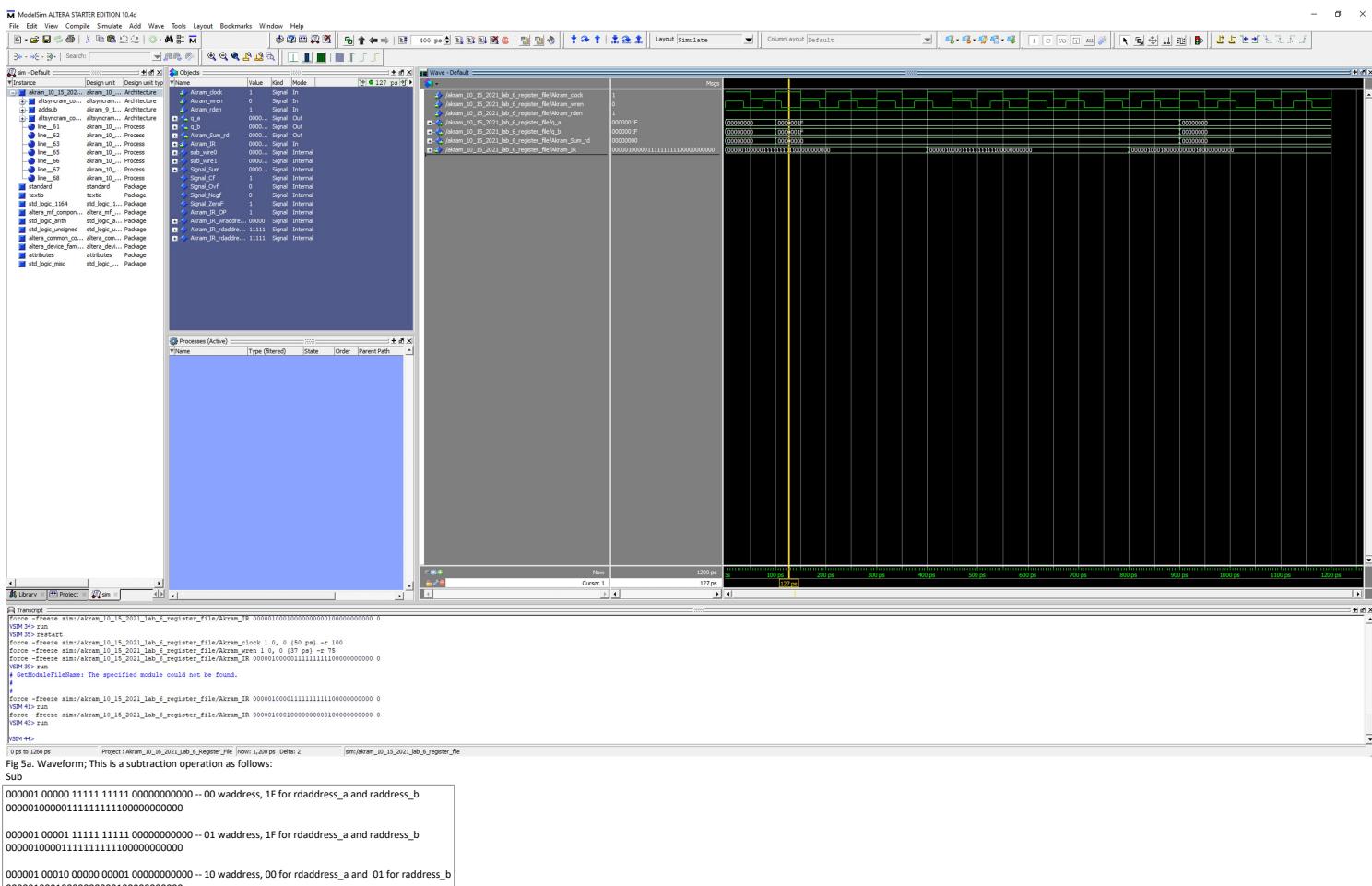


Fig 5. Waveform; data from mif file is fed in, and IR is fed in manually. This is an addition operation as follows

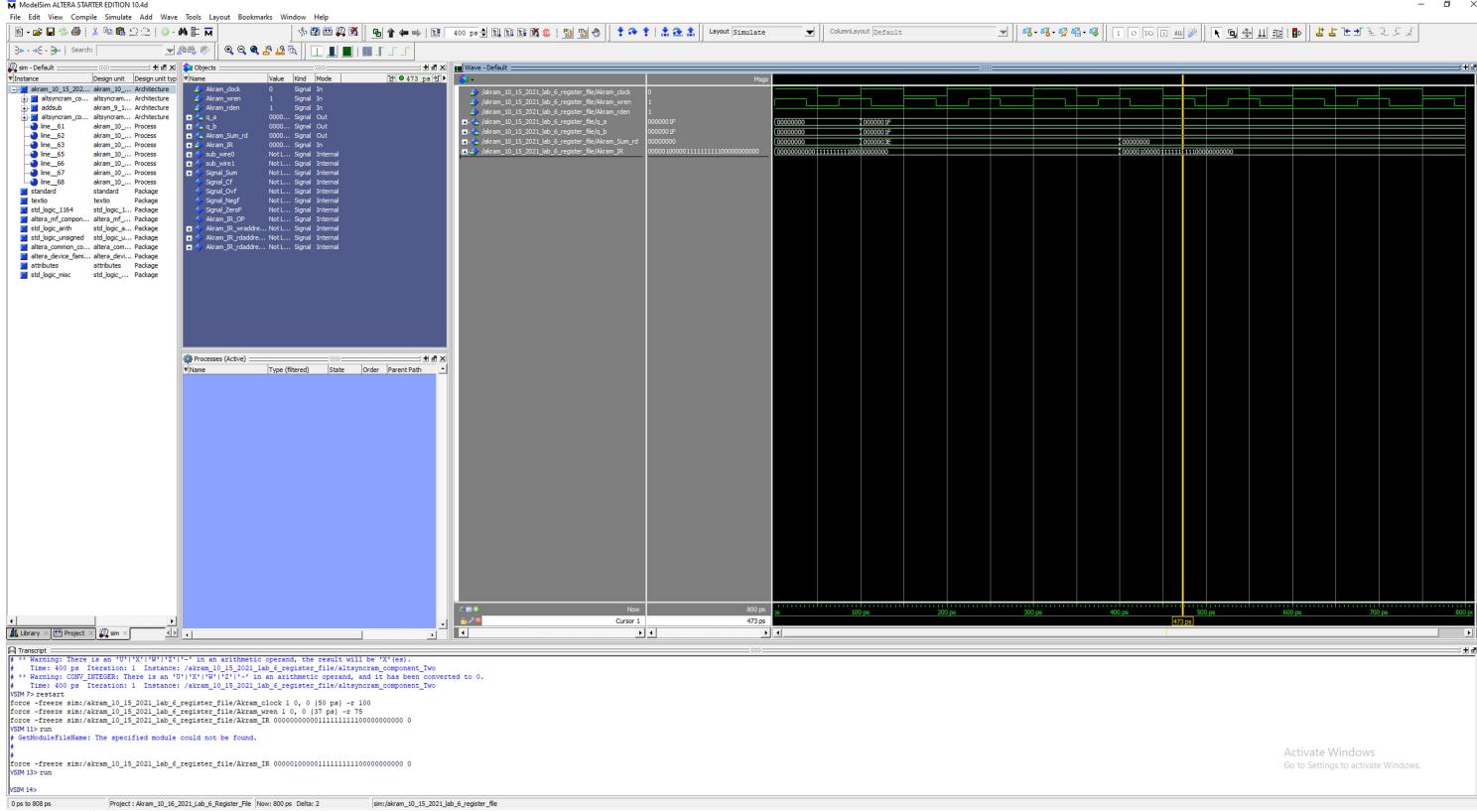
```
Add  
0000000000 11111 11111 0000000000 -- explanation: 00 waddress, 1F for rdaddress_a and raddress_b  
00000000000111111111000000000000 - IR  
  
0000000001 11111 11111 0000000000 -- explanation: 01 waddress, 1F for rdaddress_a and raddress_b  
0000000001111111110000000000 - IR  
  
000000 00010 00000 00001 0000000000 -- explanation: 10 waddress, 00 for rdaddress_a and 01 for raddress_b  
0000000100000000010000000000 - IR
```

Which results in $1F + 1F = 3E$ for Akram_Sum_rc
Shown in video demo



Which results in $1F - 1F = 0$ for Akram_Sum_rc
Shown in video demo

Shown in video demo



Conclusion

QUESTION
I learned how to use an Instruction Register and implement that to my integrated 3-port RAM from a prior lab. So now instead of manually forcing read and writes, I can feed in an Instruction Register. Also, I learned that I can feed in my data through a .MIF file into my ADD/SUB and then write that sum into a 3rd register.