Concordia University

Laboratory Report

COEN - 316

Lab – 2

Register File

Submission Date: October 24, 2019

Prepared by

Salman Rahman 27853815

Name Student ID

“I certify that this submission is my original work and meets the Faculty's Expectations of Originality”

Contents

[Figures 2](#_Toc22775246)

[ ModelSim Simulation of Register file: (please zoom in the figures if required) 3](#_Toc22775247)

[ RTL schematic and Tech schematic 8](#_Toc22775248)

[ ALU.UCF file 9](#_Toc22775249)

[ VHDL Code 10](#_Toc22775250)

# Figures

[Figure 1 Simulation of register file from 0 to 16 ns according to the given do file in the lab manual, in this case the register file was reset using the asynchronous reset signal and different registers are being loaded with the data from the din input signal 3](#_Toc22775217)

[Figure 2 Simulation of register file from 16 to 38 ns, during this time every register is being saved with data at every clock cycle 3](#_Toc22775218)

[Figure 3 Simulation of register file from 38 to 82ns, during this time registers up to register 16 in the register file was loaded with data from the din input at every rising clock edge 4](#_Toc22775219)

[Figure 4 Simulation of register file from 82 to 126ns, during this time registers up to register 27 in the register file was loaded with data from the din input at every rising clock edge 4](#_Toc22775220)

[Figure 5 Simulation of 126 to 172ns is shown in the figure. All the 32 registers are loaded with data at 32 clock rising edges. The read\_a and read\_b values are changed to see the proper output at the out ports. 5](#_Toc22775221)

[Figure 6 RTL schematic of the vhdl code 8](#_Toc22775222)

[Figure 7 Tech schematic of the synthesis 8](#_Toc22775223)

* ModelSim Simulation of Register file: (please zoom in the figures if required)

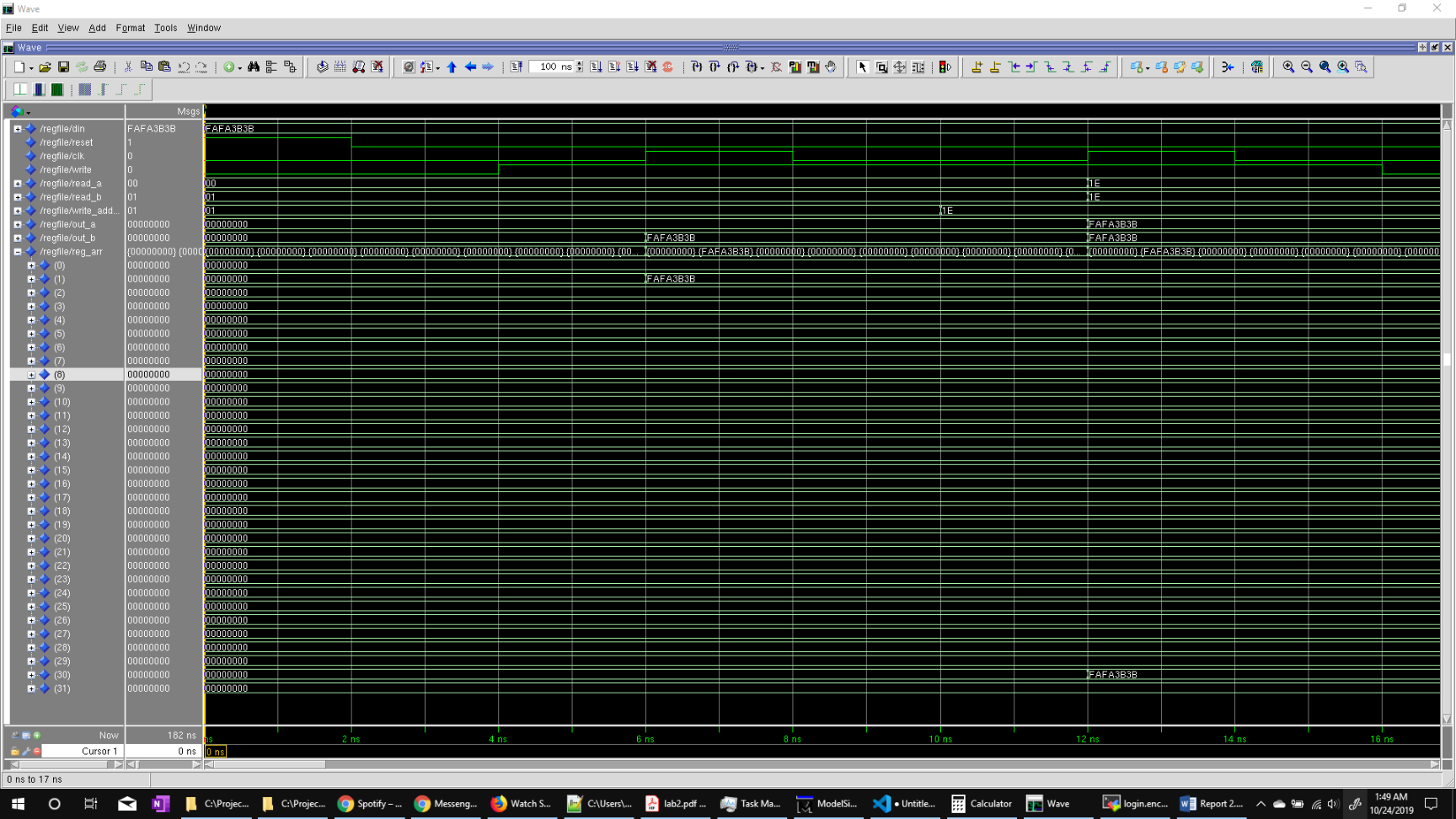


Figure 1 Simulation of register file from 0 to 16 ns according to the given do file in the lab manual, in this case the register file was reset using the asynchronous reset signal and different registers are being loaded with the data from the din input signal

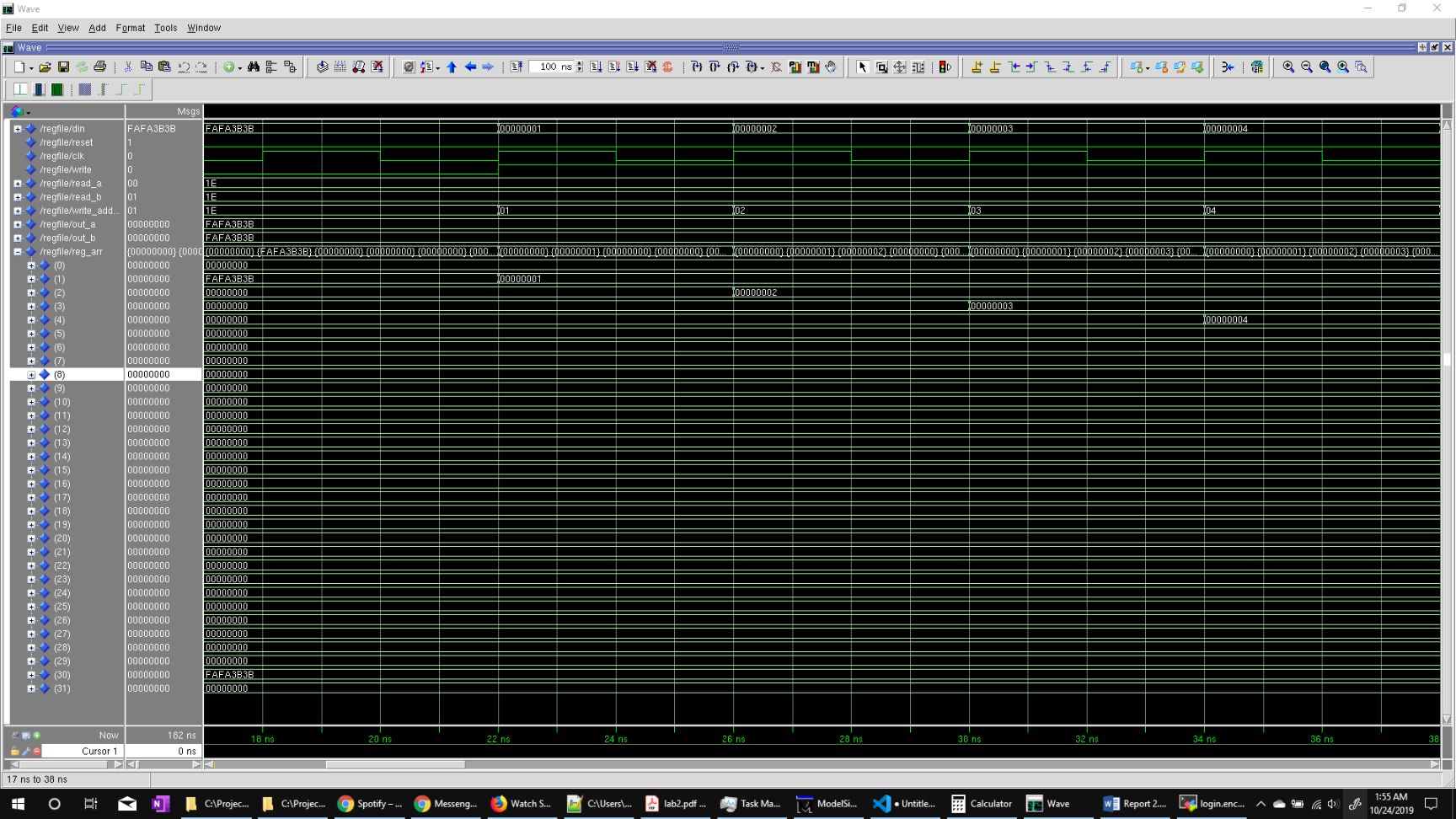


Figure 2 Simulation of register file from 16 to 38 ns, during this time every register is being saved with data at every clock cycle

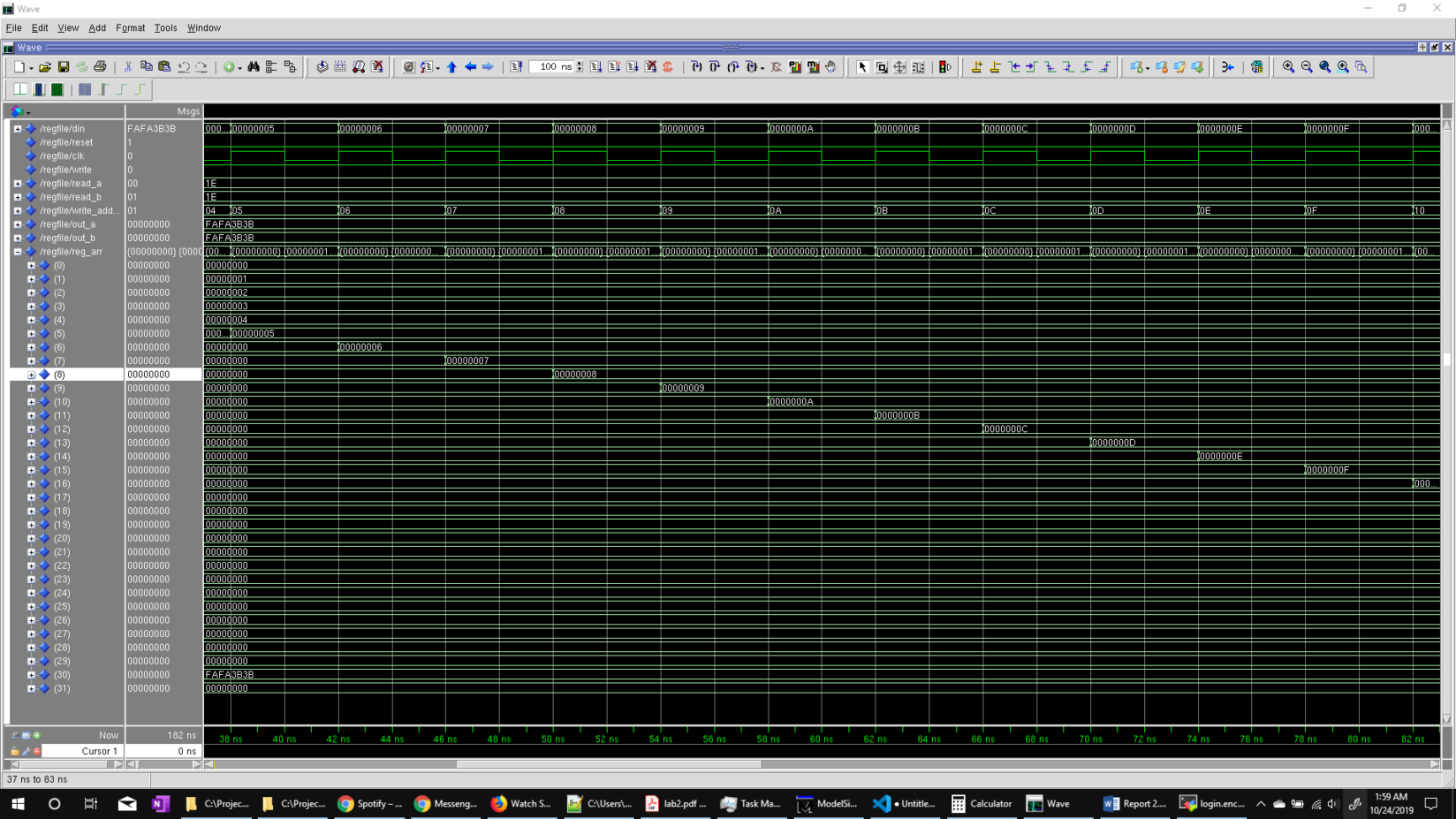


Figure 3 Simulation of register file from 38 to 82ns, during this time registers up to register 16 in the register file was loaded with data from the din input at every rising clock edge

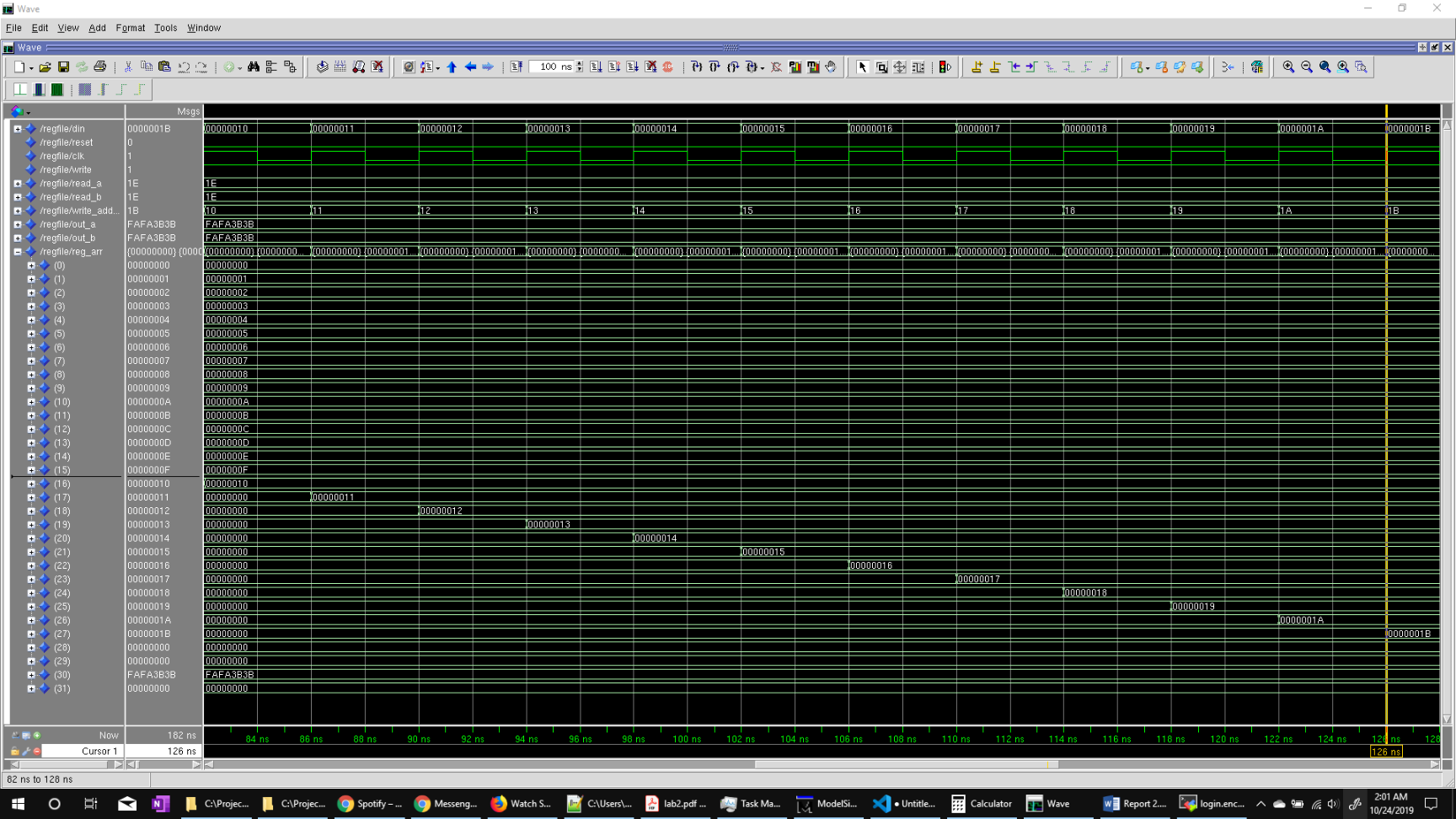


Figure 4 Simulation of register file from 82 to 126ns, during this time registers up to register 27 in the register file was loaded with data from the din input at every rising clock edge

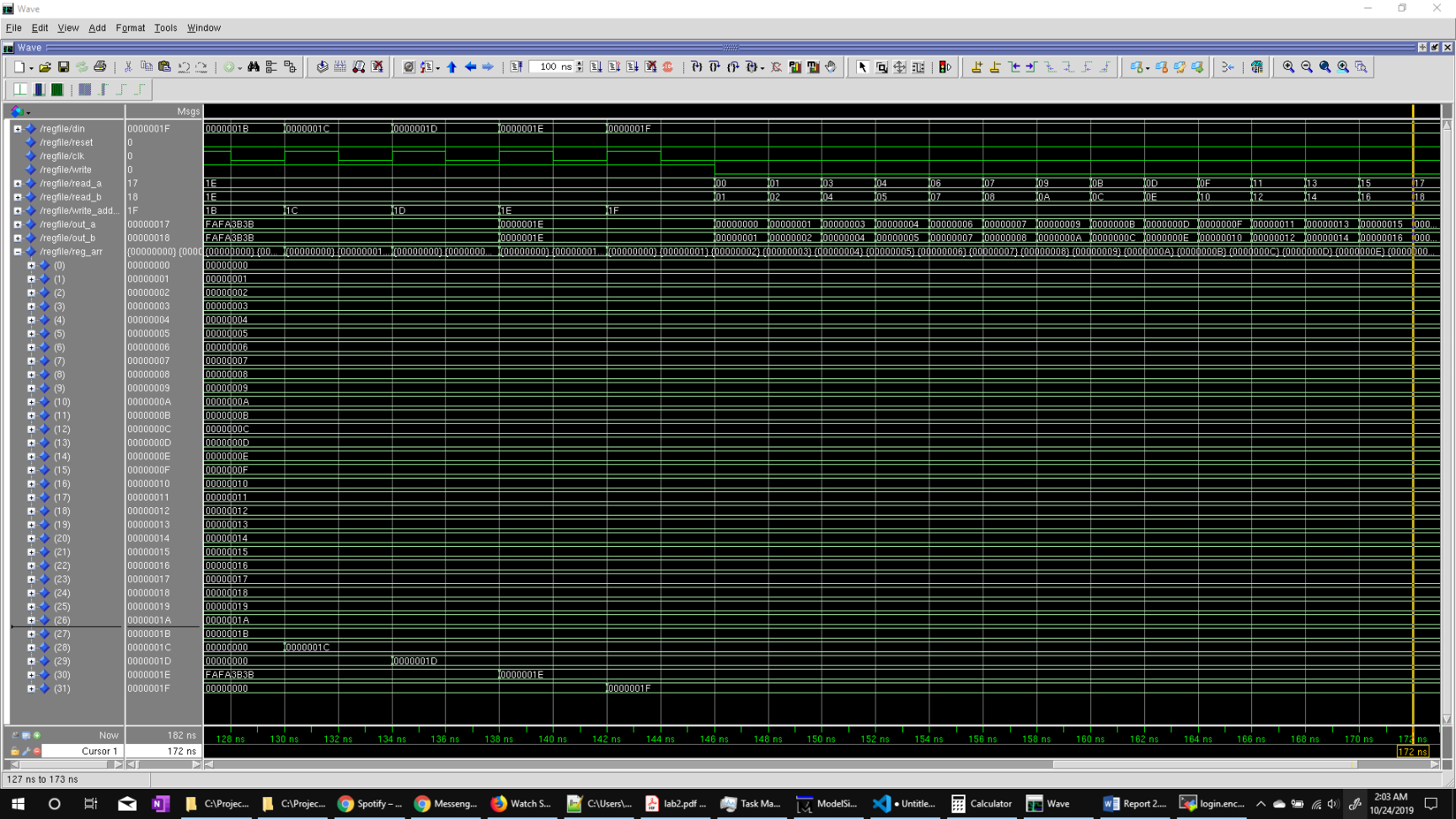


Figure 5 Simulation of 126 to 172ns is shown in the figure. All the 32 registers are loaded with data at 32 clock rising edges. The read\_a and read\_b values are changed to see the proper output at the out ports.

* Precision log file

# Info: [9566]: Logging session transcript to file /nfs/home/s/sal\_rahm/316/fpga\_adv/lab2/precision.log

// Precision RTL Synthesis 64-bit 2016.1.0.15 (Production Release) Wed Jun 8 09:35:56 PDT 2016

//

// Copyright (c) Mentor Graphics Corporation, 1996-2016, All Rights Reserved.

// Portions copyright 1991-2008 Compuware Corporation

// UNPUBLISHED, LICENSED SOFTWARE.

// CONFIDENTIAL AND PROPRIETARY INFORMATION WHICH IS THE

// PROPERTY OF MENTOR GRAPHICS CORPORATION OR ITS LICENSORS

//

// Running on Linux sal\_rahm@grace.encs.concordia.ca #1 SMP Fri Sep 20 08:24:10 CDT 2019 3.10.0-1062.1.2.el7.x86\_64 x86\_64

//

// Start time Thu Oct 24 02:16:52 2019

# -------------------------------------------------

# Info: [9566]: Logging session transcript to file /nfs/home/s/sal\_rahm/316/fpga\_adv/lab2/precision.log

# COMMAND: new\_project -name project\_1 -folder /nfs/home/s/sal\_rahm/316/fpga\_adv/lab2 -createimpl\_name lab2

# Info: [9574]: Input directory: /nfs/home/s/sal\_rahm/316/fpga\_adv/lab2

# Info: [9569]: Moving session transcript to file /nfs/home/s/sal\_rahm/316/fpga\_adv/lab2/precision.log

# Info: [9555]: Created project /nfs/home/s/sal\_rahm/316/fpga\_adv/lab2/project\_1.psp in folder /nfs/home/s/sal\_rahm/316/fpga\_adv/lab2.

# Info: [9531]: Created directory: /nfs/home/s/sal\_rahm/316/fpga\_adv/lab2/lab2.

# Info: [9554]: Created implementation lab2 in project /nfs/home/s/sal\_rahm/316/fpga\_adv/lab2/project\_1.psp.

# Info: [9575]: The Results Directory has been set to: /nfs/home/s/sal\_rahm/316/fpga\_adv/lab2/lab2/

# Info: [9566]: Logging project transcript to file /nfs/home/s/sal\_rahm/316/fpga\_adv/lab2/lab2/precision.log

# Info: [9566]: Logging suppressed messages transcript to file /nfs/home/s/sal\_rahm/316/fpga\_adv/lab2/lab2/precision.log.suppressed

# Info: [9550]: Activated implementation lab2 in project /nfs/home/s/sal\_rahm/316/fpga\_adv/lab2/project\_1.psp.

new\_project -name project\_1 -folder /nfs/home/s/sal\_rahm/316/fpga\_adv/lab2 -createimpl\_name lab2

# COMMAND: add\_input\_file {../../32-bit-CPU/32-bit-register.vhd}

add\_input\_file {../../32-bit-CPU/32-bit-register.vhd}

# COMMAND: setup\_design -manufacturer Xilinx -family "VIRTEX-II Pro" -part 2VP30ff896 -speed -7

# Info: [15298]: Setting up the design to use synthesis library "xcv2p.syn"

# Info: [575]: The global max fanout is currently set to 10000 for Xilinx - VIRTEX-II Pro.

# Info: [15324]: Setting Part to: "2VP30ff896".

# Info: [15325]: Setting Process to: "7".

# Info: [7512]: The place and route tool for current technology is ISE.

setup\_design -manufacturer Xilinx -family "VIRTEX-II Pro" -part 2VP30ff896 -speed -7

# COMMAND: setup\_design -frequency 100 -max\_fanout=10000

# Info: [575]: The global max fanout is currently set to 10000 for Xilinx - VIRTEX-II Pro.

setup\_design -frequency 100 -max\_fanout=10000

# COMMAND: compile

# Info: [3022]: Reading file: /CMC/tools/mentor/precision/Mgc\_home/pkgs/psr/techlibs/xcv2p.syn.

# Info: [634]: Loading library initialization file /CMC/tools/mentor/precision/Mgc\_home/pkgs/psr/userware/xilinx\_rename.tcl

# Info: XILINX

# Info: [40000]: vhdlorder, Release 2016a.7

# Info: [40000]: Files sorted successfully.

# Info: [40000]: hdl-analyze, Release RTLC-Precision 2016a.7

# Info: [42502]: Analyzing input file "/nfs/home/s/sal\_rahm/316/fpga\_adv/lab2/../../32-bit-CPU/32-bit-register.vhd" ...

# Info: [659]: Top module of the design is set to: regfile.

# Info: [657]: Current working directory: /nfs/home/s/sal\_rahm/316/fpga\_adv/lab2/lab2.

# Info: [40000]: RTLC-Driver, Release RTLC-Precision 2016a.7

# Info: [40000]: Last compiled on Jun 2 2016 06:11:46

# Info: [44512]: Initializing...

# Info: [44504]: Partitioning design ....

# Info: [40000]: RTLCompiler, Release RTLC-Precision 2016a.7

# Info: [40000]: Last compiled on Jun 2 2016 06:47:43

# Info: [44512]: Initializing...

# Info: [44522]: Root Module work.regfile(register\_file\_arch): Pre-processing...

# Info: [45251]: Built-in hardware memory core inferred for variable ': regfile.reg\_arr depth = 32, width = 32'.

# Info: [44523]: Root Module work.regfile(register\_file\_arch): Compiling...

# Info: [44842]: Compilation successfully completed.

# Info: [44856]: Total lines of RTL compiled: 42.

# Info: [44835]: Total CPU time for compilation: 0.0 secs.

# Info: [44513]: Overall running time for compilation: 0.0 secs.

# Info: [657]: Current working directory: /nfs/home/s/sal\_rahm/316/fpga\_adv/lab2/lab2.

# Info: [15330]: Doing rtl optimizations.

# Info: [660]: Finished compiling design.

compile

# COMMAND: synthesize

# Info: [657]: Current working directory: /nfs/home/s/sal\_rahm/316/fpga\_adv/lab2/lab2.

# Info: [20013]: Precision will use 3 processor(s).

# Info: [15002]: Optimizing design view:.work.regfile.register\_file\_arch

# Info: [12035]: -- Running timing characterization...

# Info: [8048]: Added global buffer BUFGP for Port port:clk

# Info: [3027]: Writing file: /nfs/home/s/sal\_rahm/316/fpga\_adv/lab2/lab2/regfile.edf.

# Info: [3027]: Writing file: /nfs/home/s/sal\_rahm/316/fpga\_adv/lab2/lab2/regfile.ucf.

# Info: [12045]: Starting timing reports generation...

# Info: [12046]: Timing reports generation done.

# Info: [12048]: POST-SYNTHESIS TIMING REPORTS ARE ESTIMATES AND SHOULD NOT BE RELIED ON TO MAKE QoR DECISIONS. For accurate timing information, please run place-and-route (P&R) and review P&R generated timing reports.

# Info: [660]: Finished synthesizing design.

# Info: [11019]: Total CPU time for synthesis: 1.1 s secs.

# Info: [11020]: Overall running time for synthesis: 2.2 s secs.

synthesize

# COMMAND: save\_project

# Info: [9562]: Saved implementation lab2 in project /nfs/home/s/sal\_rahm/316/fpga\_adv/lab2/project\_1.psp.

save\_project

# COMMAND: close\_project -discard

# Info: [9530]: Closed project: /nfs/home/s/sal\_rahm/316/fpga\_adv/lab2/project\_1.psp.

close\_project -discard

# COMMAND: exit

exit

* RTL schematic and Tech schematic

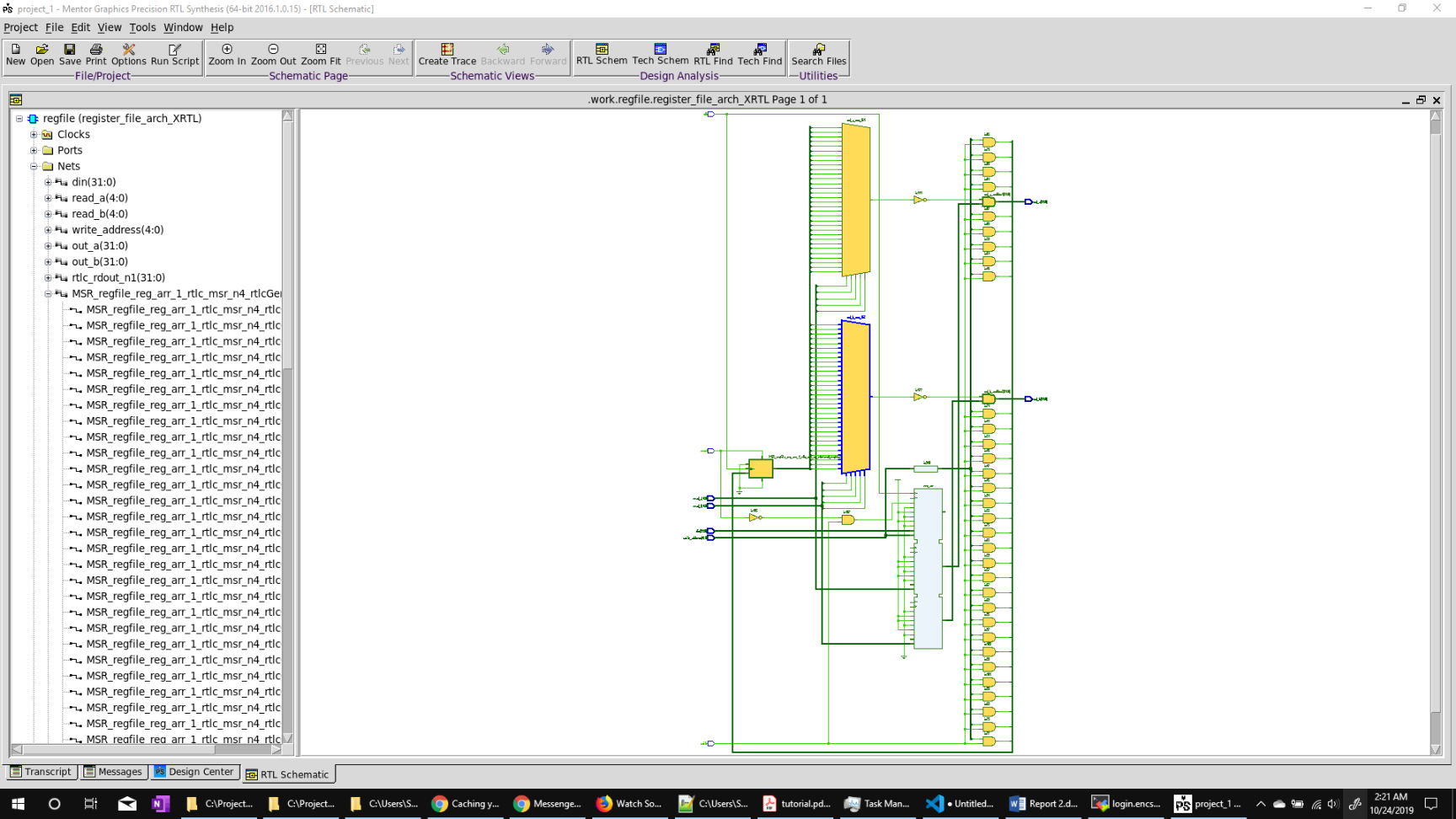


Figure 6 RTL schematic of the VHDL code

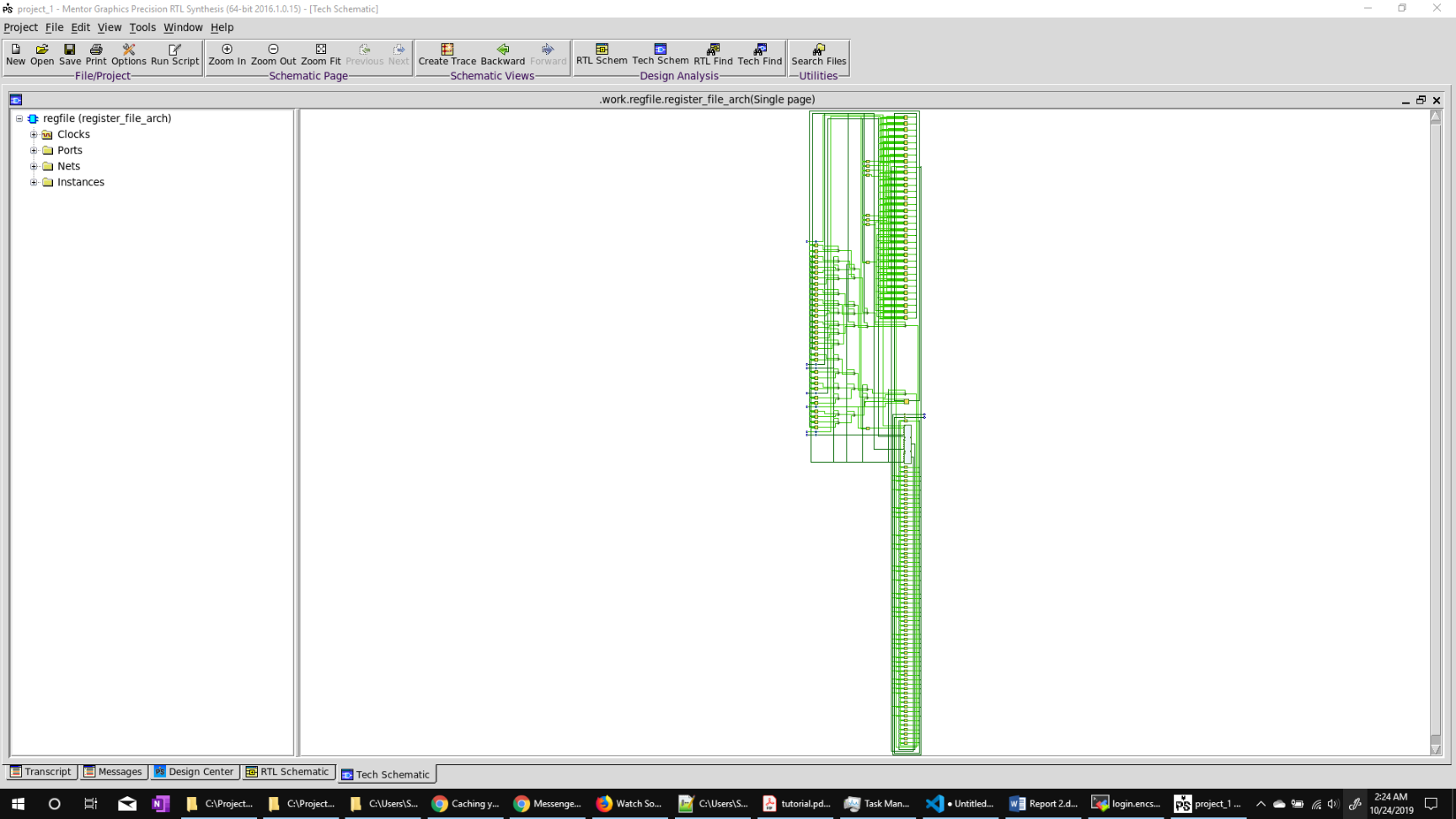


Figure 7 Tech schematic of the synthesis

* ALU.UCF file

# Precision RTL Synthesis 64-bit 2016.1.0.15 (Production Release) Wed Jun 8 09:35:56 PDT 2016

CONFIG STEPPING="0";

NET out\_a (0) LOC = T6;

NET out\_a (1) LOC = V1;

NET out\_a (2) LOC = R3;

NET out\_a (3) LOC = R5;

NET out\_a (4) LOC = T2;

NET out\_a (5) LOC = P4;

NET out\_a (6) LOC = R7;

NET out\_a (7) LOC = P2;

* VHDL Code

-- 32 x 32 register file

-- two read ports, one write port with write enable

library IEEE;

use IEEE.std\_logic\_1164.all;

use IEEE.std\_logic\_unsigned.all;

entity regfile is

port(

din : in std\_logic\_vector(31 downto 0);

reset : in std\_logic;

clk : in std\_logic;

write : in std\_logic;

read\_a : in std\_logic\_vector(4 downto 0);

read\_b : in std\_logic\_vector(4 downto 0);

write\_address : in std\_logic\_vector(4 downto 0);

out\_a : out std\_logic\_vector(31 downto 0);

out\_b : out std\_logic\_vector(31 downto 0)

);

end regfile ;

architecture register\_file\_arch of regfile is

-- 32 registers in a register file, thus 0 to 31 in reg\_arr

type register\_array is array(0 to 31) of std\_logic\_vector(din'length-1 downto din'right);

signal reg\_arr : register\_array;

begin

read: process(read\_a, read\_b, reg\_arr)

begin

-- converting std vector to integer - to find the appropriate register to access

out\_a <= reg\_arr(CONV\_INTEGER(read\_a));

out\_b <= reg\_arr(CONV\_INTEGER(read\_b));

end process;

reg\_file\_update: process(clk, reset)

begin

if(reset = '1') then

reg\_arr <= (others => (others => '0'));

elsif(clk'event and clk = '1' and write = '1') then

reg\_arr(CONV\_INTEGER(write\_address)) <= din;

end if;

end process;

end register\_file\_arch ;