Concordia University

Laboratory Report

COEN - 316

Lab – 4

CPU Datapath

Submission Date: November 21, 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](#_Toc25286845)

[ Objective 3](#_Toc25286846)

[ Introduction 3](#_Toc25286847)

[ Conceptual Diagram of datapath 5](#_Toc25286848)

[ Conclusion 5](#_Toc25286849)

[ Results 7](#_Toc25286850)

[o ModelSim Simulation of Next Address Unit: (please zoom in the figures if required) 7](#_Toc25286851)

[o Precision log file 8](#_Toc25286852)

[o RTL schematic and Tech schematic 10](#_Toc25286853)

[o ALU.UCF file 11](#_Toc25286854)

[o VHDL Code 12](#_Toc25286855)

# Figures

[Figure 1 Instructions format of MIPS 4](#_Toc25286856)

[Figure 2 Conceptual diagram of the designed datapath 5](#_Toc25286857)

[Figure 3 Simulation of next address unit is shown with different sets of inputs to test all the possible cases 7](#_Toc25286858)

[Figure 4 Screenshot of tester do file 7](#_Toc25286859)

[Figure 5 RTL schematic of the VHDL code 10](#_Toc25286860)

[Figure 6 Tech schematic of the synthesis 10](#_Toc25286861)

* Objective

The objective of the lab was to design the datapath using all the components created in the previous labs which are the ALU, REGFILE and the NEXT-ADDRESS unit. To complete the data path two more components were required are I-CACHE and D-CACHE

* Introduction

The datapath of the processor handles the execution of the instructions. The datapath designed during the lab consists of the following components:

1. PC
2. I-CACHE
3. reg\_des – Mux
4. REGFILE
5. alu\_src – Mux
6. ALU – Mux
7. D-Cache
8. reg\_in\_src Mux
9. Sign EXTEND block
10. NEXT-ADDRESS

The designed datapath is capable of executing 20 different instructions of three different types as follows:

1. J-Format instruction (format shown in figure below)
   1. Jump – j there
   2. Jump register – jr rs
2. I-Format instruction (format shown in figure below)
   1. Load upper immediate – lui rt, immediate
   2. Set less than immediate – slti rt,rs,immediate
   3. Add immediate – addi rt,rs,immediate
   4. AND immediate – andi rt,rs,immediate
   5. OR immediate – ori rt, rs,immediate
   6. XOR immediate – xori rt,rs,immediate
   7. Load word – lw rt, immediate(rs)
   8. Store word – sw rt, immediate(rs)
   9. Branch less than 0 – bltz rs, there
   10. Branch equal – beq rs, rt, there
   11. Branch not equal 0 – bne rs, rt, there
3. R-Format instruction (format shown in figure below)
   1. Addition– add rd, rs,rt
   2. Subtraction – sub rd, rs, rt
   3. Set less than – slt rd, rs, rt
   4. AND (logical) – and rd, rs, rt
   5. OR (logical) – or rd, rs, rt
   6. XOR (logical) – xor rd, rs, rt
   7. NOR (logical) – nor rd, rs, rt

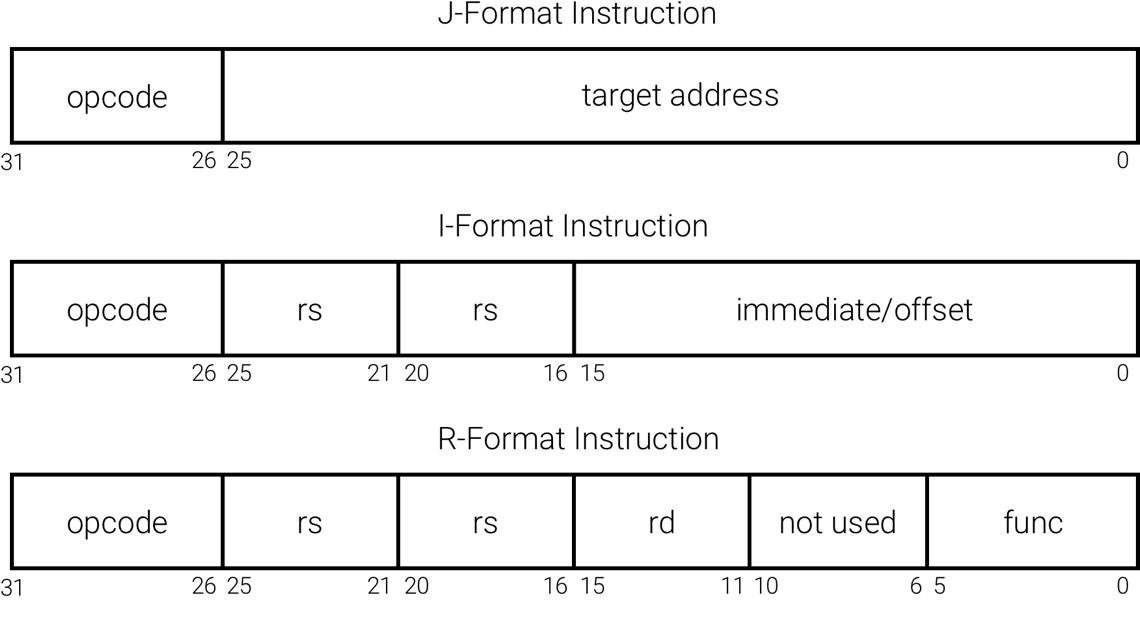


Figure 1 Instructions format of MIPS

* Conceptual Diagram of datapath

The conceptual diagram of the data path was created using the diagram provided in the lab manual. The control signals and their bus width is shown in the figure below.

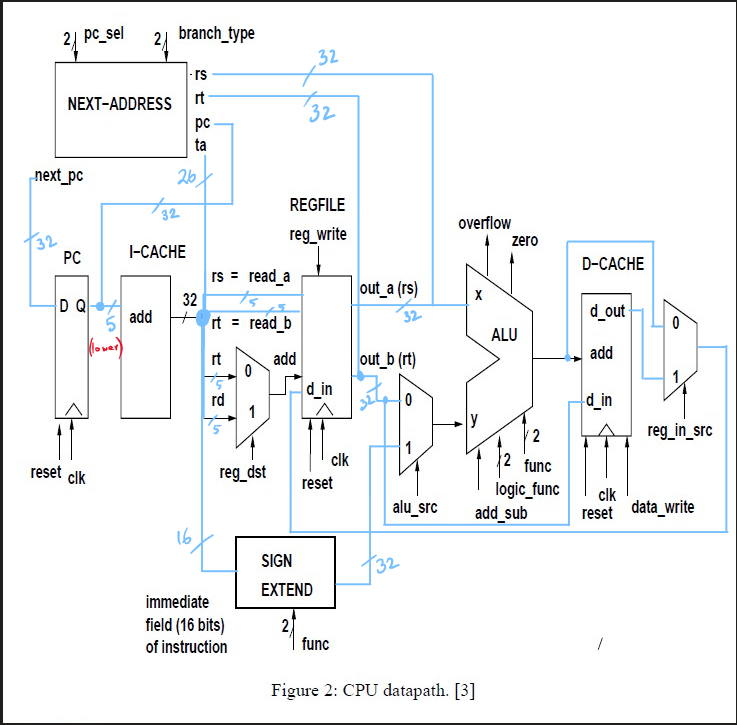


Figure 2 Conceptual diagram of the designed datapath

* Conclusion

In conclusion, the data path was successfully designed and simulated during the lab which made use of all the components created in the previous labs. The datapath successfully executed twenty different instruction of different formats. A total of ten control signals were required to implement the datapath.

* Results
  + ModelSim Simulation of Next Address Unit: (please zoom in the figures if required)

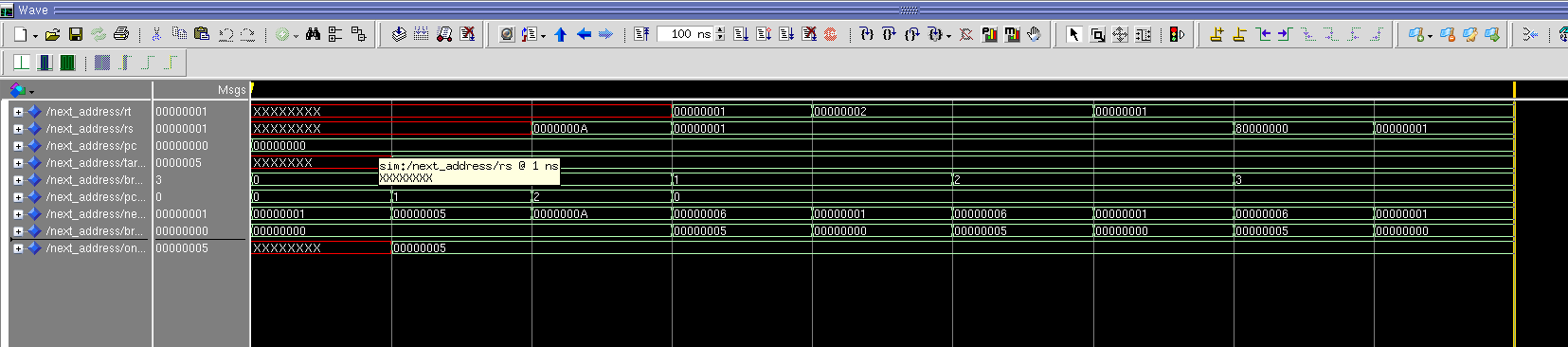


Figure 3 Simulation of next address unit is shown with different sets of inputs to test all the possible cases

The do file is shown below that was used to test the next\_address unit:

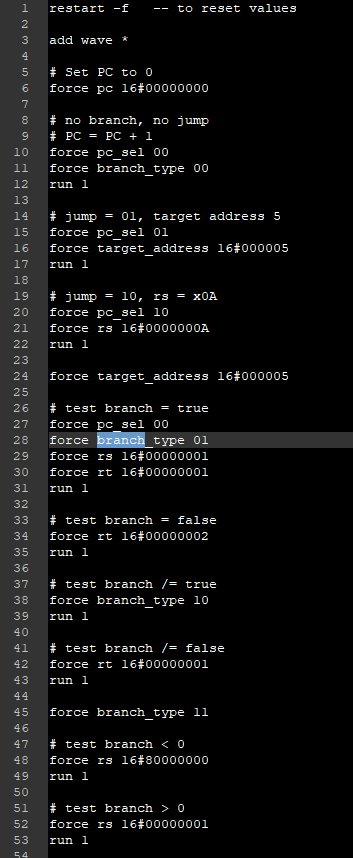


Figure 4 Screenshot of tester do file

The output of the next\_address unit was as expected according to our conceptual diagram.

* + Precision log file

# Info: [9566]: Logging session transcript to file /nfs/home/s/sal\_rahm/316/fpga\_adv/lab3/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@poise.encs.concordia.ca #1 SMP Thu Oct 17 14:34:42 CDT 2019 3.10.0-1062.4.1.el7.x86\_64 x86\_64

//

// Start time Thu Nov 7 03:14:10 2019

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

# Info: [9566]: Logging session transcript to file /nfs/home/s/sal\_rahm/316/fpga\_adv/lab3/precision.log

# COMMAND: new\_project -name lab3 -folder /nfs/home/s/sal\_rahm/316/fpga\_adv/lab3 -createimpl\_name lab3\_impl\_1

# Info: [9574]: Input directory: /nfs/home/s/sal\_rahm/316/fpga\_adv/lab3

# Info: [9569]: Moving session transcript to file /nfs/home/s/sal\_rahm/316/fpga\_adv/lab3/precision.log

# Info: [9555]: Created project /nfs/home/s/sal\_rahm/316/fpga\_adv/lab3/lab3.psp in folder /nfs/home/s/sal\_rahm/316/fpga\_adv/lab3.

# Info: [9531]: Created directory: /nfs/home/s/sal\_rahm/316/fpga\_adv/lab3/lab3\_impl\_1.

# Info: [9554]: Created implementation lab3\_impl\_1 in project /nfs/home/s/sal\_rahm/316/fpga\_adv/lab3/lab3.psp.

# Info: [9575]: The Results Directory has been set to: /nfs/home/s/sal\_rahm/316/fpga\_adv/lab3/lab3\_impl\_1/

# Info: [9566]: Logging project transcript to file /nfs/home/s/sal\_rahm/316/fpga\_adv/lab3/lab3\_impl\_1/precision.log

# Info: [9566]: Logging suppressed messages transcript to file /nfs/home/s/sal\_rahm/316/fpga\_adv/lab3/lab3\_impl\_1/precision.log.suppressed

# Info: [9550]: Activated implementation lab3\_impl\_1 in project /nfs/home/s/sal\_rahm/316/fpga\_adv/lab3/lab3.psp.

new\_project -name lab3 -folder /nfs/home/s/sal\_rahm/316/fpga\_adv/lab3 -createimpl\_name lab3\_impl\_1

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

add\_input\_file {../../32-bit-CPU/next\_address.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/lab3/../../32-bit-CPU/next\_address.vhd" ...

# Info: [659]: Top module of the design is set to: next\_address.

# Info: [657]: Current working directory: /nfs/home/s/sal\_rahm/316/fpga\_adv/lab3/lab3\_impl\_1.

# 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.next\_address(next\_address\_arch): Pre-processing...

# Info: [44523]: Root Module work.next\_address(next\_address\_arch): Compiling...

# Info: [44846]: Rebalanced Expression Tree...

# Info: [44842]: Compilation successfully completed.

# Info: [44841]: Counter Inferencing === Detected : 1, Inferred (Modgen/Selcounter/AddSub) : 0 (0 / 0 / 0), AcrossDH (Merged/Not-Merged) : (0 / 0), Not-Inferred (Acrossdh/Attempted) : (0 / 0), Local Vars : 1 ===

# Info: [44856]: Total lines of RTL compiled: 57.

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

# Info: [44513]: Overall running time for compilation: 1.0 secs.

# Info: [657]: Current working directory: /nfs/home/s/sal\_rahm/316/fpga\_adv/lab3/lab3\_impl\_1.

# 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/lab3/lab3\_impl\_1.

# Info: [15002]: Optimizing design view:.work.next\_address.next\_address\_arch

# Info: [3027]: Writing file: /nfs/home/s/sal\_rahm/316/fpga\_adv/lab3/lab3\_impl\_1/next\_address.edf.

# Info: [3027]: Writing file: /nfs/home/s/sal\_rahm/316/fpga\_adv/lab3/lab3\_impl\_1/next\_address.ucf.

# Info: [660]: Finished synthesizing design.

# Info: [11019]: Total CPU time for synthesis: 0.8 s secs.

# Info: [11020]: Overall running time for synthesis: 1.0 s secs.

synthesize

# COMMAND: exit -force

# Warning: [9526]: Discarded unsaved work in implementation lab3\_impl\_1.

# Info: [9530]: Closed project: /nfs/home/s/sal\_rahm/316/fpga\_adv/lab3/lab3.psp.

close\_project -discard

exit -force

* + RTL schematic and Tech schematic

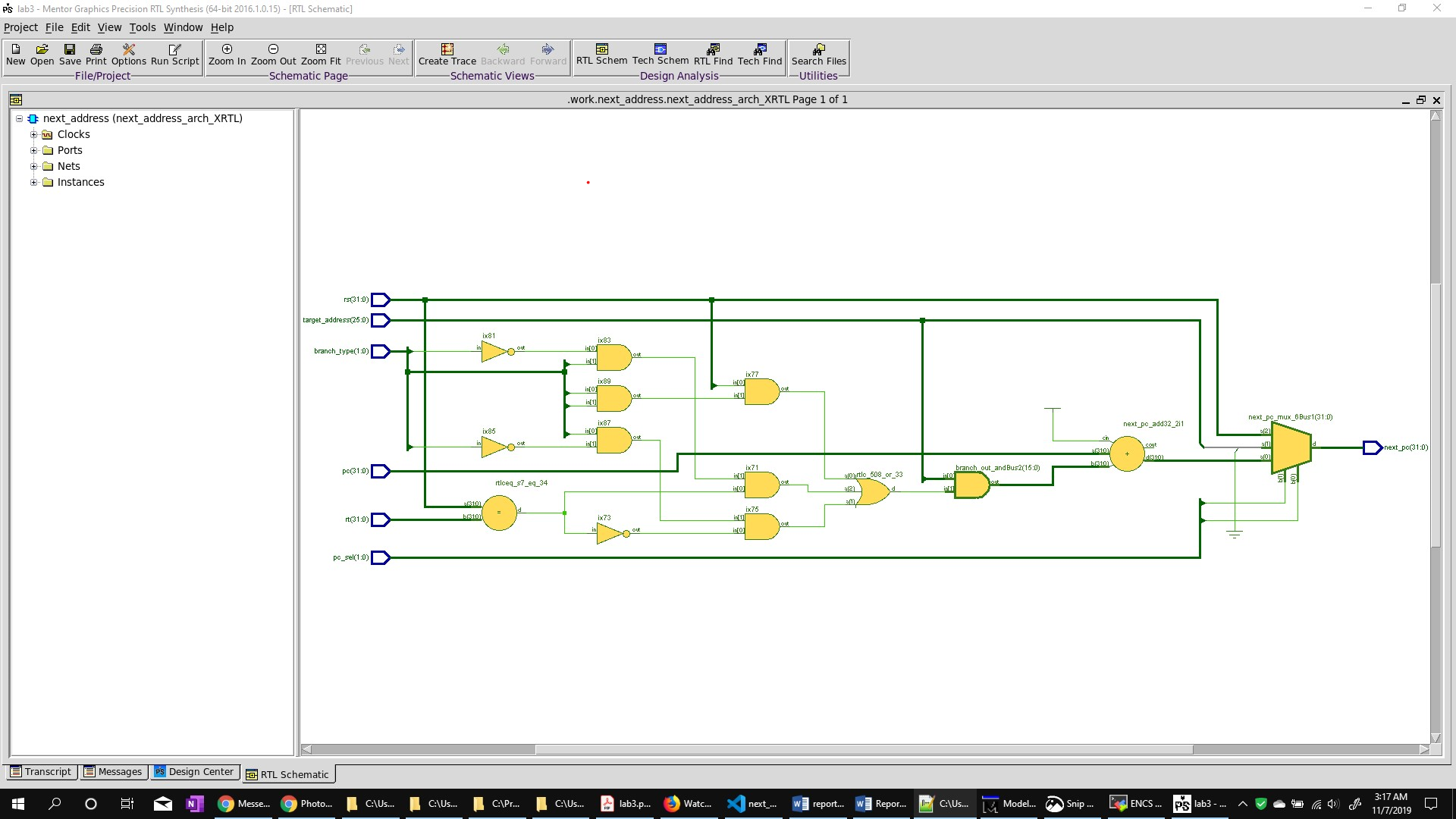


Figure 5 RTL schematic of the VHDL code

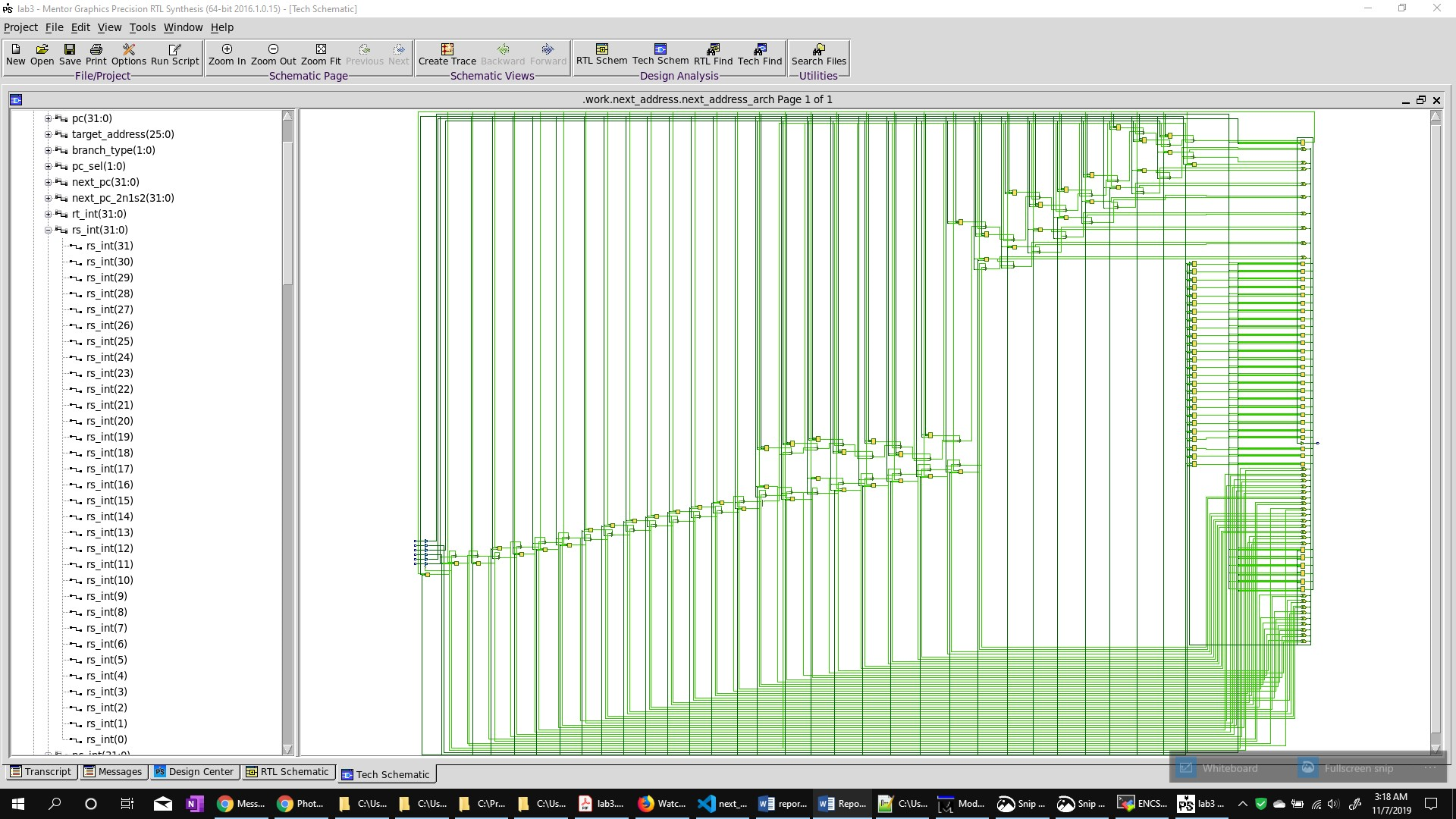


Figure 6 Tech schematic of the synthesis

* + 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 next\_pc (0) LOC = T6;

NET next\_pc (1) LOC = V1;

NET next\_pc (2) LOC = R3;

NET next\_pc (3) LOC = R5;

NET next\_pc (4) LOC = T2;

NET next\_pc (5) LOC = P4;

NET next\_pc (6) LOC = R7;

NET next\_pc (7) LOC = P2;

* + VHDL Code

-- next address calculation unit

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_signed.all;

entity next\_address is

port(

    rt, rs          : in std\_logic\_vector(31 downto 0);

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

    target\_address  : in std\_logic\_vector(25 downto 0);

    branch\_type     : in std\_logic\_vector(1 downto 0);

    pc\_sel          : in std\_logic\_vector(1 downto 0);

    next\_pc         : out std\_logic\_vector(31 downto 0)

);

end next\_address;

architecture next\_address\_arch of next\_address is

    signal branch\_out: std\_logic\_vector(next\_pc'length-1 downto next\_pc'right);

    signal on\_branch\_sign\_ext: std\_logic\_vector(next\_pc'length-1 downto next\_pc'right);

    begin

        on\_branch\_sign\_ext((target\_address'length-1)-(5+5) downto target\_address'right) <= target\_address((target\_address'length-1)-(5+5) downto target\_address'right);

        on\_branch\_sign\_ext(next\_pc'length-1 downto target\_address'length-(5+5)) <= (others => target\_address((target\_address'length-1)-(5+5)));

        branch\_select: process(on\_branch\_sign\_ext, rs, rt, branch\_type)

        begin

            branch\_out <= (others=>'0');

            case branch\_type is

                when "00" =>

                    branch\_out <= (others=>'0');

                when "01" =>

                    if(rs = rt) then

                        branch\_out <= on\_branch\_sign\_ext;

                    end if;

                when "10" =>

                    if(rs /= rt) then

                        branch\_out <= on\_branch\_sign\_ext;

                    end if;

                when "11" =>

                    if(rs < 0) then

                        branch\_out <= on\_branch\_sign\_ext;

                    end if;

                when others =>

                    branch\_out <= (others=>'U');

            end case;

        end process;

        pc\_select: process(pc, branch\_out, target\_address, rs, pc\_sel)

        begin

            case pc\_sel is

                when "00" =>

                    next\_pc <= pc + branch\_out + 1;

                when "01" =>

                    next\_pc <= "000000" & target\_address;

                when "10" =>

                    next\_pc <= rs;

                when others =>

                    next\_pc <= (others => 'U');

                    --  11 is unused

            end case;

        end process;

end next\_address\_arch;