COEN 316 – Computer Architecture and Design **Jafar Abbas**
LAB 2 – Register File 26346650
October 26, 2017

Objectives

The purpose of this lab is to design, synthesize, simulate and test a 32x32 Register File which handles reading two registers simultaneously, synchronous writing to any register and an asynchronous reset. The register file will be built upon and used as part of a more complex CPU design in future labs.

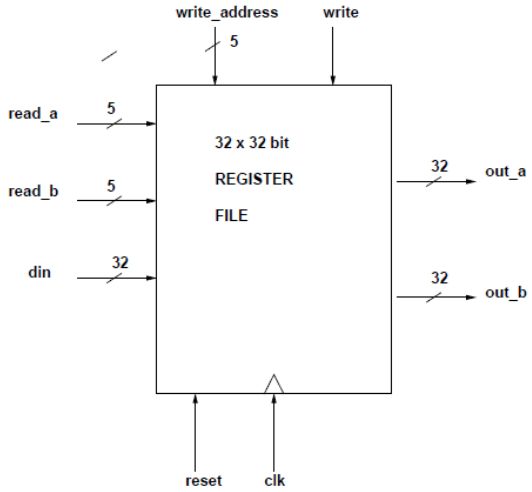


Figure 1: Register file block diagram.

Procedure

This lab follows the procedure outlined in Parts I, II, III, and IV of the tutorial "Digital Logic Simulation and Synthesis Using Modelsim, Precision RTL, and Xilinx ISE". The end-goal is to simulate, synthesize and download to the Xilinx FPGA board the 32-bit ALU described below in the LAB 2 manual.

Requirements

1) Implement the Register File using any style of VHDL
2) Asynchronous reset
3) Asynchronous reading of 2 registers
4) Synchronous write to any register via write and datain

Results

Entity for regfile:

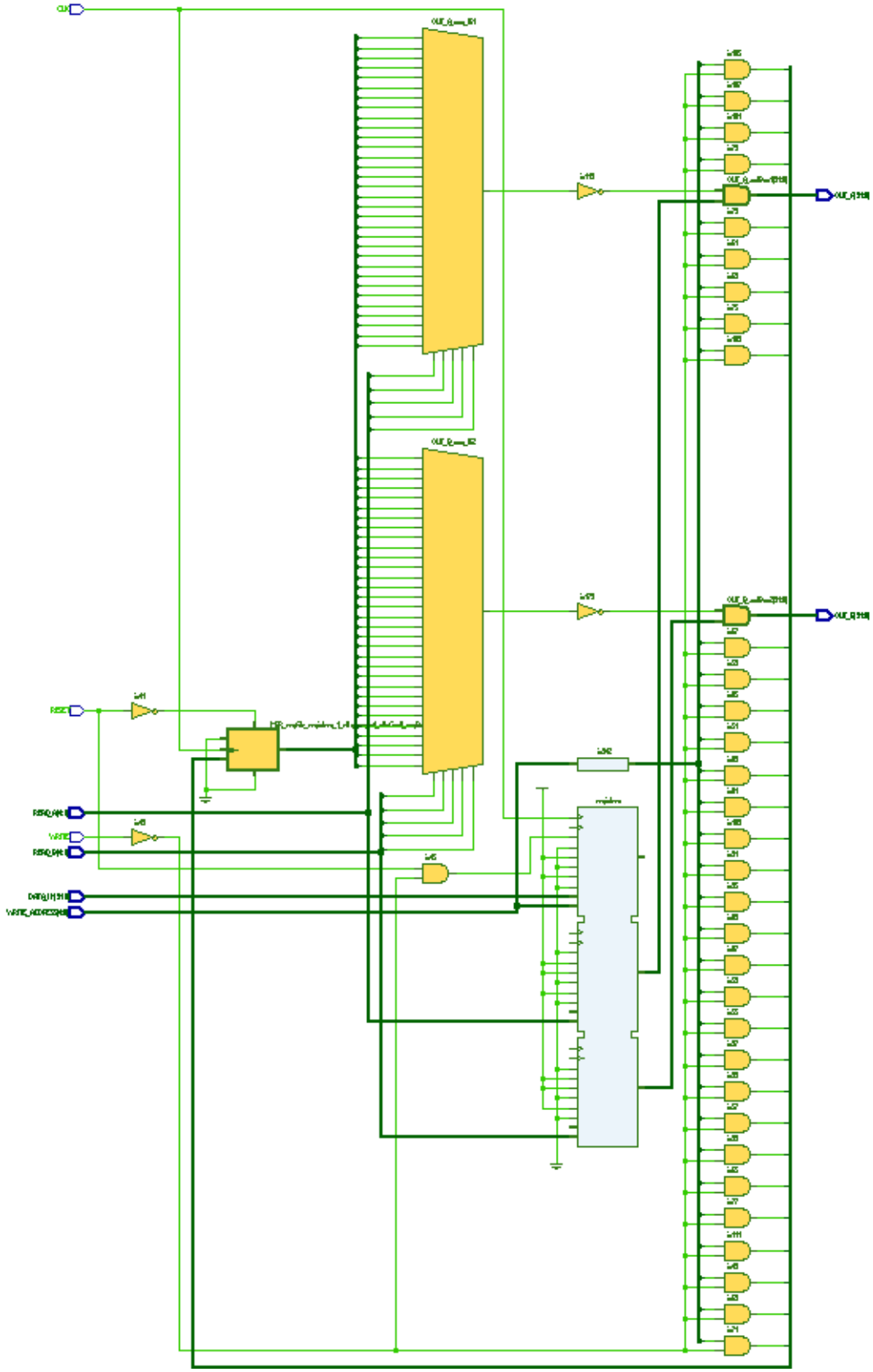
```
library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_unsigned.all;
use ieee.numeric_std.ALL;

-- Register implementation based on my stack from COEN 313
-- Assumptions:
--              Inverted outputs
--              WRITE simply overwrites data


entity regfile is

    port(
    --inputs
        DATA_IN : 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);

    --outputs
        OUT_A : out std_logic_vector(31 downto 0);
        OUT_B : out std_logic_vector(31 downto 0));

end regfile ;
```

Architecture of regfile:

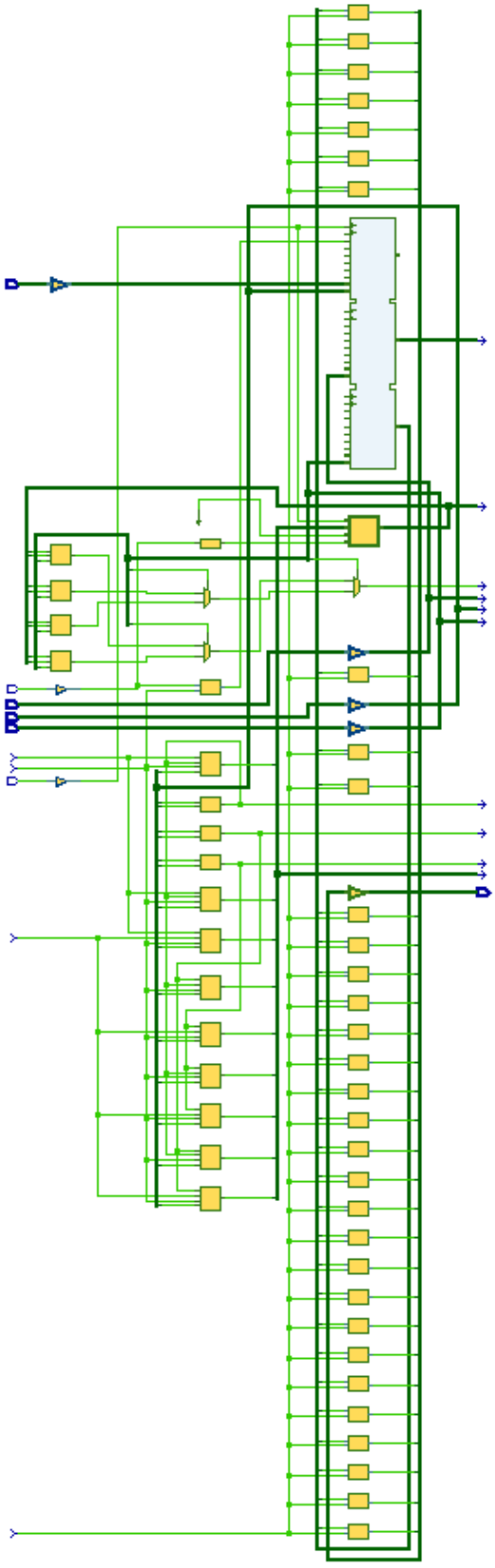
I used an array of 32 32bit vectors to represent my registers. read_a and read_b addresses are converted to integers and used as index to my array to access the register they represent.

```
architecture lab2 of regfile is

    type mem_type is array(31 downto 0) of std_logic_vector(31 downto 0);

    signal registers : mem_type := (others => (others => '0'));

    constant top_ptr : integer:= 0;
    constant bot_ptr : integer:= 31;

    begin

        OUT_A <= registers(to_integer(unsigned(READ_A)));
        OUT_B <= registers(to_integer(unsigned(READ_B)));

        --Reset/Writefunctionality (in order of priority)
        func : process(CLK, WRITE, RESET)

            variable temp_index: integer := 0;

            begin

                -- Asynchronous reset
                -- Resets EVERYTHING, stack registers & shadow registers
                if (RESET = '0') then

                        for i in top_ptr to bot_ptr loop
                            registers(i) <= (others => '0');
                        end loop;

                -- Synchronous with rising edge of CLK
                elsif (rising_edge(CLK)) then

                    -- WRITE (force)
                    -- This function will overwrite data at register[write_address]
                    if (WRITE = '0') then

                        temp_index := to_integer(unsigned(WRITE_ADDRESS));

                        -- check if write_address corresponds to a valid entry in regfile
                        if ((bot_ptr >= temp_index) and (temp_index >= top_ptr)) then

                            registers(temp_index) <= DATA_IN;

                        end if;

                    else
                        -- do nothing
                    end if;
                else
                    -- do nothing
                end if;
        end process;

end lab2;
```

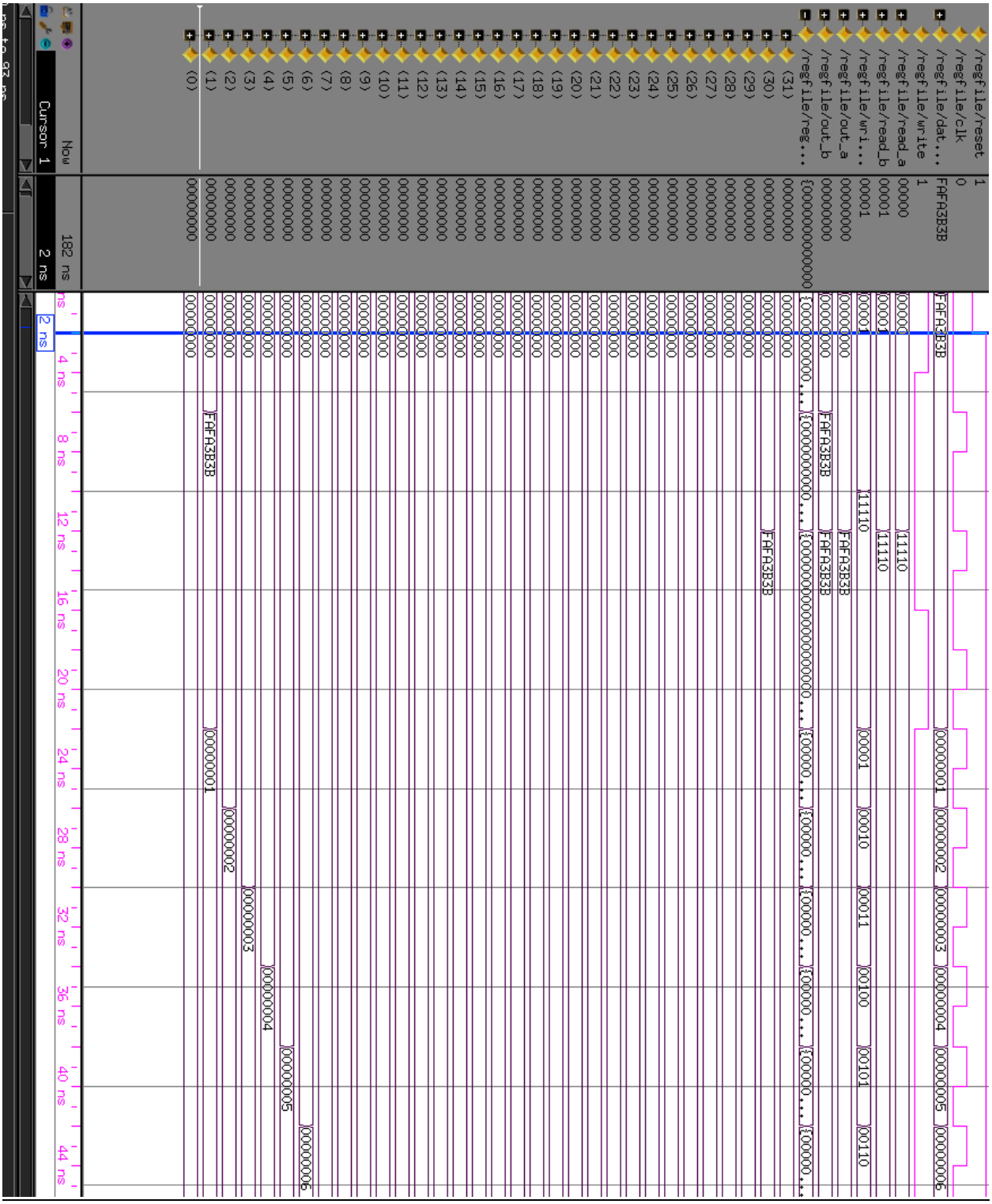
Synthesized circuit:

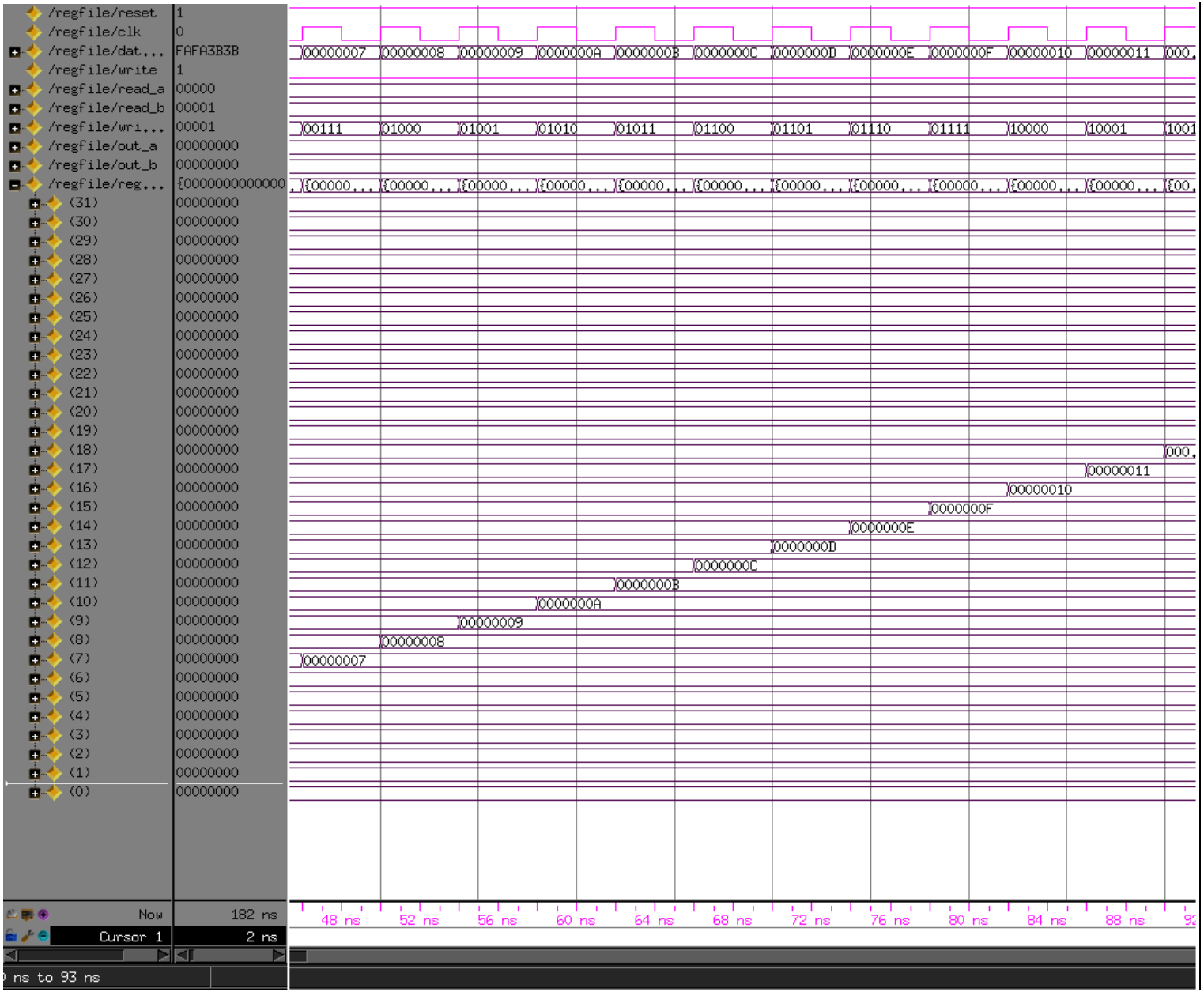


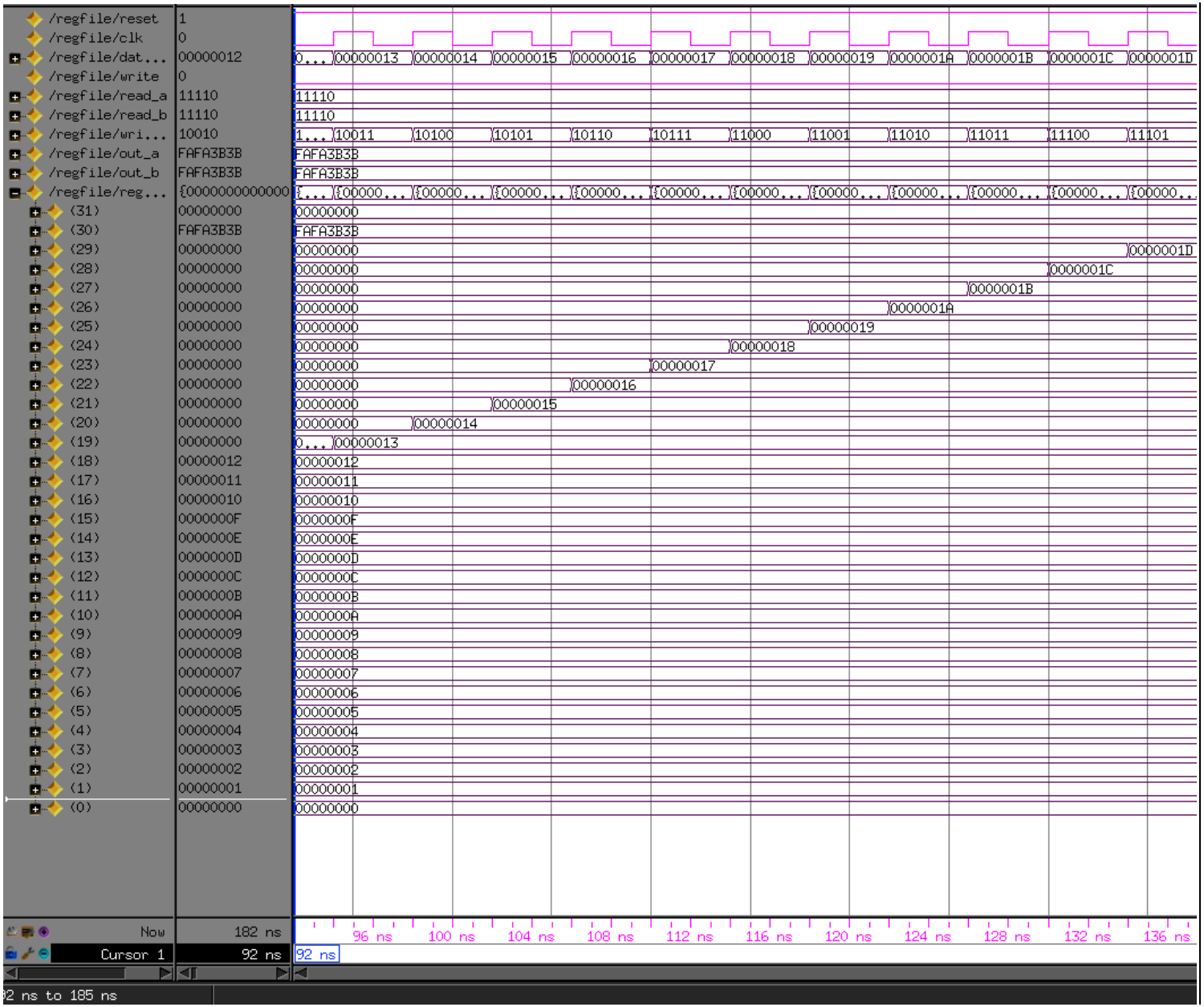
Technology mapped circuit:

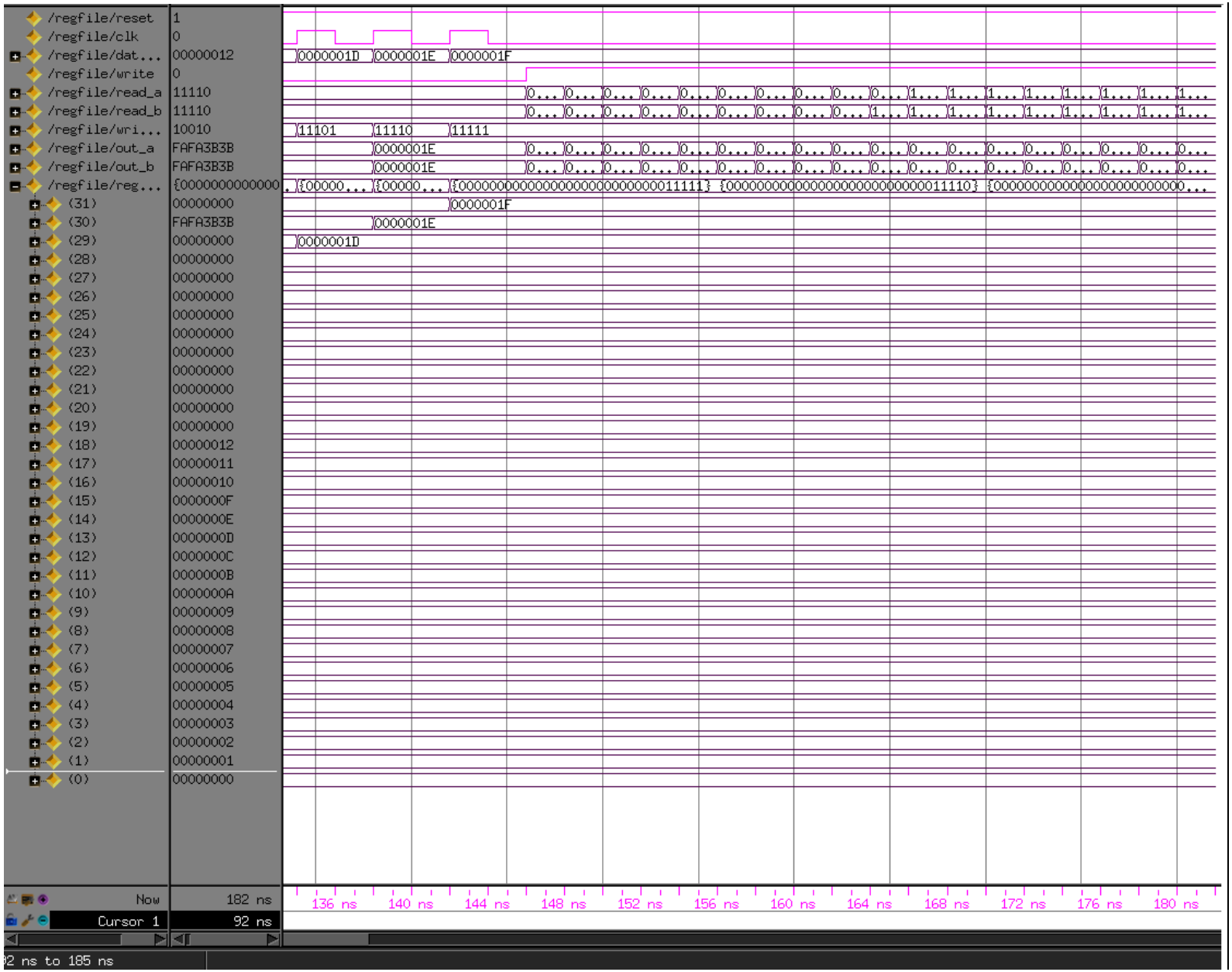


Simulation Results




/regfile/reset
/regfile/clk
/regfile/dat...
/regfile/write
/regfile/read_a
/regfile/read_b
/regfile/wri...
/regfile/out_a
/regfile/out_b
/regfile/reg...
(31)
(30)
(29)
(28)
(27)
(26)
(25)
(24)
(23)
(22)
(21)
(20)
(19)
(18)
(17)
(16)
(15)
(14)
(13)
(12)
(11)
(10)
(9)
(8)
(7)
(6)
(5)
(4)
(3)
(2)
(1)
(0)
1
0
FAFA3B3B
1
00000
00001
00001
00000000
00000000
{0000000000000
00000000
Now
182 ns
Cursor 1
2 ns
ns to 93 ns
00000007
00000008
00000009
0000000A
0000000B
0000000C
0000000D
0000000E
0000000F
00000010
00000011
00111
01000
01001
01010
01011
01100
01101
01110
01111
10000
10001
48 ns
52 ns
56 ns
60 ns
64 ns
68 ns
72 ns
76 ns
80 ns
84 ns
88 ns


/regfile/reset
/regfile/clk
/regfile/dat...
/regfile/write
/regfile/read_a
/regfile/read_b
/regfile/wri...
/regfile/out_a
/regfile/out_b
/regfile/reg...
(31)
(30)
(29)
(28)
(27)
(26)
(25)
(24)
(23)
(22)
(21)
(20)
(19)
(18)
(17)
(16)
(15)
(14)
(13)
(12)
(11)
(10)
(9)
(8)
(7)
(6)
(5)
(4)
(3)
(2)
(1)
(0)
Now
182 ns
Cursor 1
92 ns
96 ns
100 ns
104 ns
108 ns
112 ns
116 ns
120 ns
124 ns
128 ns
132 ns
136 ns
92 ns to 185 ns


/regfile/reset
/regfile/clk
/regfile/dat...
/regfile/write
/regfile/read_a
/regfile/read_b
/regfile/wri...
/regfile/out_a
/regfile/out_b
/regfile/reg...
(31)
(30)
(29)
(28)
(27)
(26)
(25)
(24)
(23)
(22)
(21)
(20)
(19)
(18)
(17)
(16)
(15)
(14)
(13)
(12)
(11)
(10)
(9)
(8)
(7)
(6)
(5)
(4)
(3)
(2)
(1)
(0)
1
0
00000012
0
11110
11110
10010
FAFA3B3B
FAFA3B3B
00000000
00000000
FAFA3B3B
00000000
00000000
00000000
00000000
00000000
00000000
00000000
00000000
00000000
00000012
00000011
00000010
0000000F
0000000E
0000000D
0000000C
0000000B
0000000A
00000009
00000008
00000007
00000006
00000005
00000004
00000003
00000002
00000001
00000000
0000001D
0000001E
0000001F
11101
11110
11111
Now
182 ns
Cursor 1
92 ns
136 ns
140 ns
144 ns
148 ns
152 ns
156 ns
160 ns
164 ns
168 ns
172 ns
176 ns
180 ns
2 ns to 185 ns

Configuration file (I/O mapping):

Not implemented in this lab due to lack of inputs&outputs

Precision.log file:

```
# ------------------------------------------
# Info: [9566]: Logging session transcript to file /nfs/home/j/j_abba/Modelsim/LAB2/FPGA_ADV/precision.log
# COMMAND: new_project -name regfile -folder /nfs/home/j/j_abba/Modelsim/LAB2/FPGA_ADV -createimpl_name regfile_impl_1
# Info: [9574]: Input directory: /nfs/home/j/j_abba/Modelsim/LAB2/FPGA_ADV
# Info: [9569]: Moving session transcript to file /nfs/home/j/j_abba/Modelsim/LAB2/FPGA_ADV/precision.log
# Info: [9555]: Created project /nfs/home/j/j_abba/Modelsim/LAB2/FPGA_ADV/regfile.psp in folder /nfs/home/j/j_abba/Modelsim/LAB2/FPGA_ADV.
# Info: [9531]: Created directory: /nfs/home/j/j_abba/Modelsim/LAB2/FPGA_ADV/regfile_impl_1.
# Info: [9554]: Created implementation regfile_impl_1 in project /nfs/home/j/j_abba/Modelsim/LAB2/FPGA_ADV/regfile.psp.
# Info: [9575]: The Results Directory has been set to: /nfs/home/j/j_abba/Modelsim/LAB2/FPGA_ADV/regfile_impl_1/
# Info: [9566]: Logging project transcript to file /nfs/home/j/j_abba/Modelsim/LAB2/FPGA_ADV/regfile_impl_1/precision.log
# Info: [9566]: Logging suppressed messages transcript to file /nfs/home/j/j_abba/Modelsim/LAB2/FPGA_ADV/regfile_impl_1/precision.log.suppressed
# Info: [9550]: Activated implementation regfile_impl_1 in project /nfs/home/j/j_abba/Modelsim/LAB2/FPGA_ADV/regfile.psp.
new_project -name regfile -folder /nfs/home/j/j_abba/Modelsim/LAB2/FPGA_ADV -createimpl_name regfile_impl_1
# COMMAND: add_input_file {../Code/regfile.vhd}
add_input_file {../Code/regfile.vhd}
# COMMAND: setup_design -manufacturer Xilinx -family "VIRTEX-II Pro" -part 2VP20ff896 -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: "2VP20ff896".
# 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 2VP20ff896 -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/j/j_abba/Modelsim/LAB2/FPGA_ADV/../Code/regfile.vhd" ...
# Info: [659]: Top module of the design is set to: regfile.
# Info: [657]: Current working directory: /nfs/home/j/j_abba/Modelsim/LAB2/FPGA_ADV/regfile_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.regfile(lab2): Pre-processing...
# Info: [45251]: Built-in hardware memory core inferred for variable ': regfile.registers depth = 32, width = 32'.
# Info: [44523]: Root Module work.regfile(lab2): Compiling...
# Info: [45308]: "/nfs/home/j/j_abba/Modelsim/LAB2/FPGA_ADV/../Code/regfile.vhd", line 70: The comparison operator has been optimized to constant 0.
# Info: [44842]: Compilation successfully completed.
# Info: [44856]: Total lines of RTL compiled: 69.
# 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/j/j_abba/Modelsim/LAB2/FPGA_ADV/regfile_impl_1.
# Info: [15330]: Doing rtl optimizations.
# Info: [660]: Finished compiling design.
compile
# COMMAND: synthesize
# Info: [657]: Current working directory: /nfs/home/j/j_abba/Modelsim/LAB2/FPGA_ADV/regfile_impl_1.
# Info: [20013]: Precision will use 3 processor(s).
# Info: [15002]: Optimizing design view:.work.regfile.lab2
# Info: [12035]: -- Running timing characterization...
# Info: [8048]: Added global buffer BUFGP for Port port:CLK
# Info: [3027]: Writing file: /nfs/home/j/j_abba/Modelsim/LAB2/FPGA_ADV/regfile_impl_1/regfile.edf.
# Info: [3027]: Writing file: /nfs/home/j/j_abba/Modelsim/LAB2/FPGA_ADV/regfile_impl_1/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
# Info: [660]: Finished synthesizing design.
# Info: [11019]: Total CPU time for synthesis: 1.2 s secs.
# Info: [11020]: Overall running time for synthesis: 1.9 s secs.
```

I got no warnings or errors!

Conclusion

In conclusion, in this lab I successfully implemented a 32x32 Register File. The code compiled and simulated using Modelsim which allowed confirming its correctness. The RTL schematic was generated to obtain a physical representation of the entity and using the precision log we were able to identify possible mistakes and errors (it was clean).

The given DO file in this lab didn't properly test the reset function, since it resets before even filling the register file. I did actually test this manually and it worked properly although that's not shown in my screenshots.

Since the FPGA board has very limited number of inputs, in this lab we did not actually map any inputs and physically test the design on the board. Although, I still generated a bit-file with the Xilinx tools to make sure all is well.

Regardless, in the future it will be easy to instantiate my 32x32 register file and use it as part of a larger design of a CPU.