EE476
Cad 2 – Register File
Yexuan Chen
Yuxiang Chen
Xinyi Chang

1. Introduction:

The purpose of this mini project is to design a register file with 13 wordlines and 16 bitlines. The write data and the write address signal come in after the rising edge of the clock. Then, the register file write in the data at the rising edge. Finally, the read ports will read the data when the read address is selected. This performs the write before read operation.

2. Analytical Approach

In order to build a complete register file, we need to build a decoder for the write word line and two decoders to select the read address. And for the main part of the register file, we choose the latch based register file. It uses one b-latch to store the write data and it is connected with 13 a-latches. And the data at the selected wordline will be written into the a latch at the following clock rising edge. The reason we choose the latch based register is because it efficiently reduces the redundant usage of b-latches and also reduces the area of the layout.

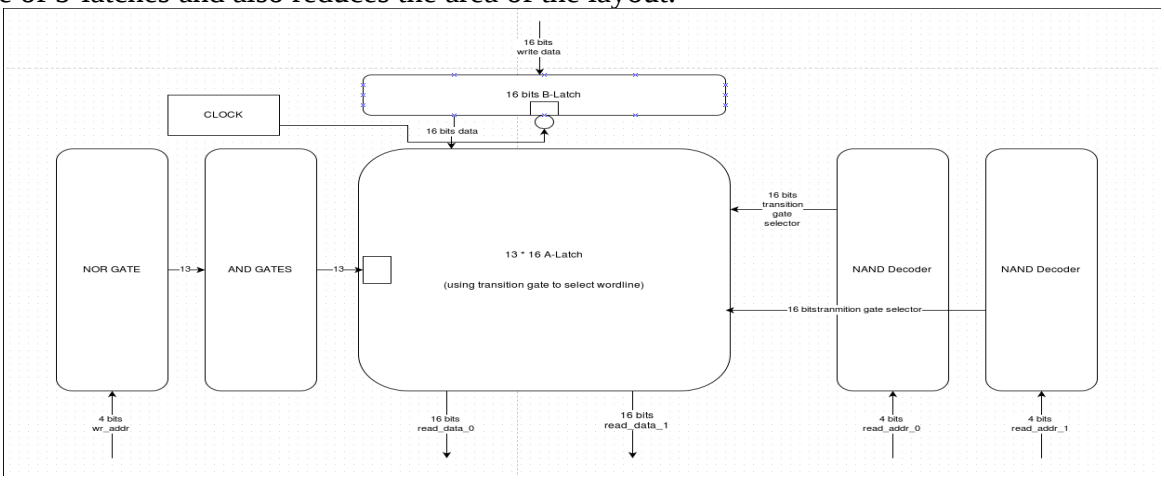

Figure 2.1 High Level Block Diagram

3. Schematic

Figure 3.1 shows the two decoders for the register file and figure 3.2 shows the schematic design of the register file. And Figure 3.3 shows the schematic simulation of the register file, it shows the complete write before read operation. And it also verifies that our design can read and write to each of our 13 registers.

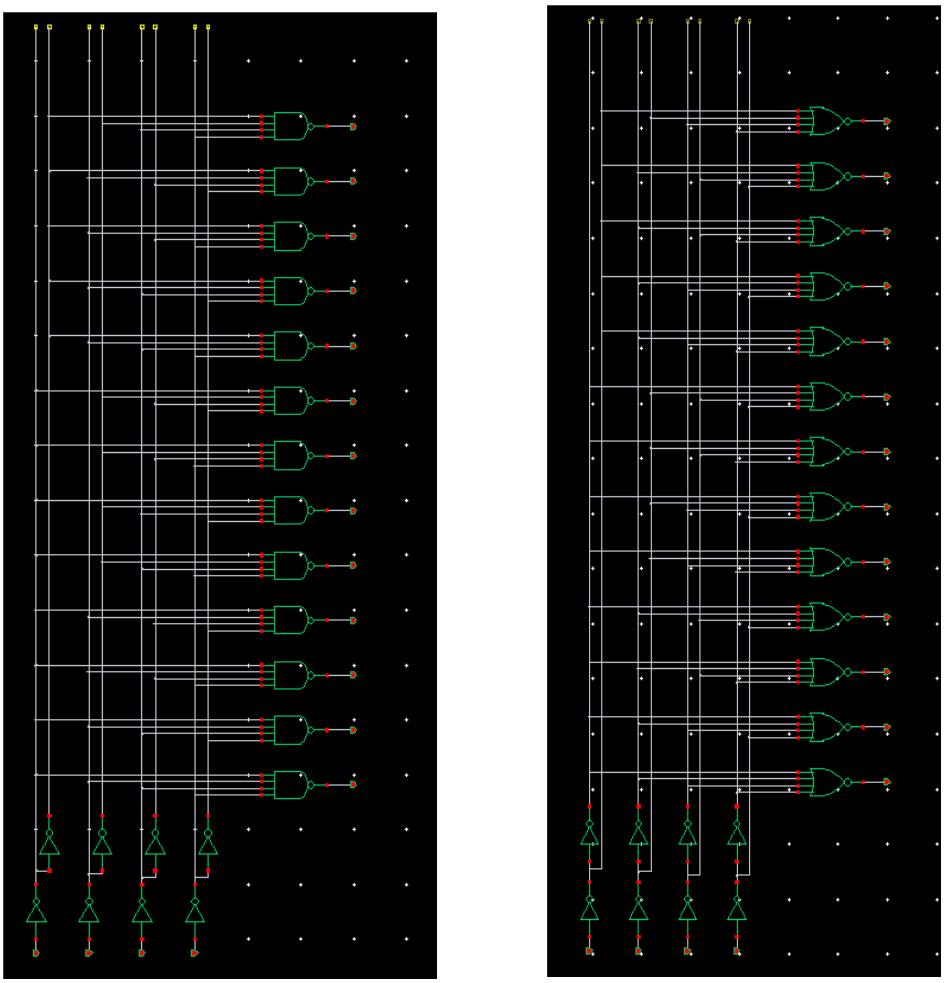

Figure 3.1 – Decoder for the Register File

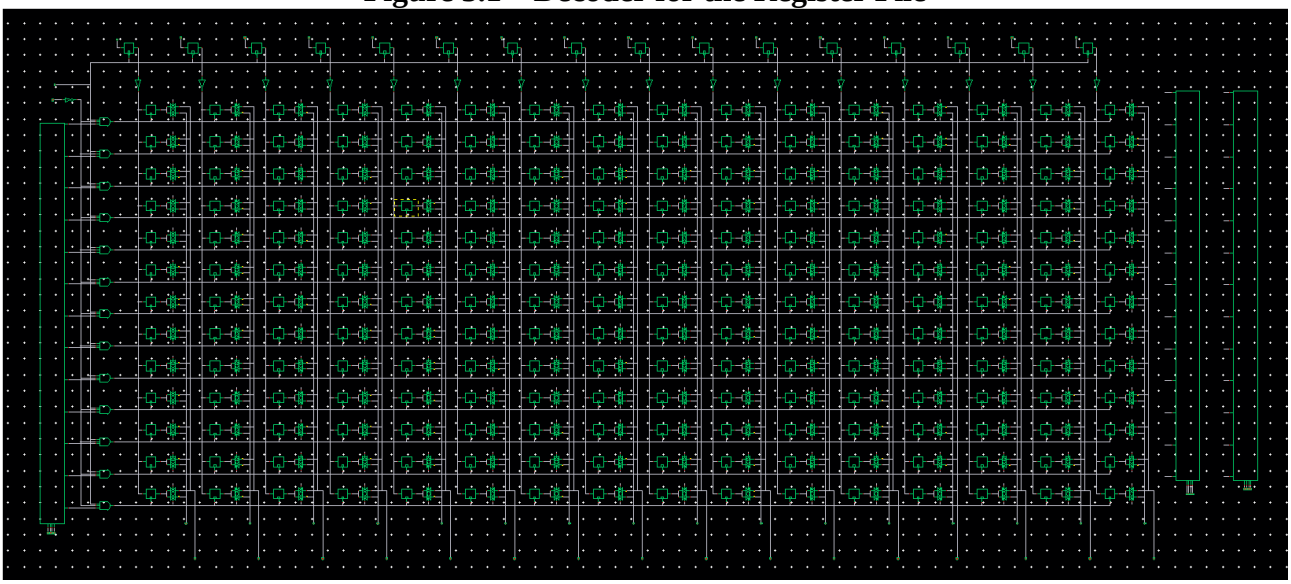

Figure 3.2 – schematic design of the Register File

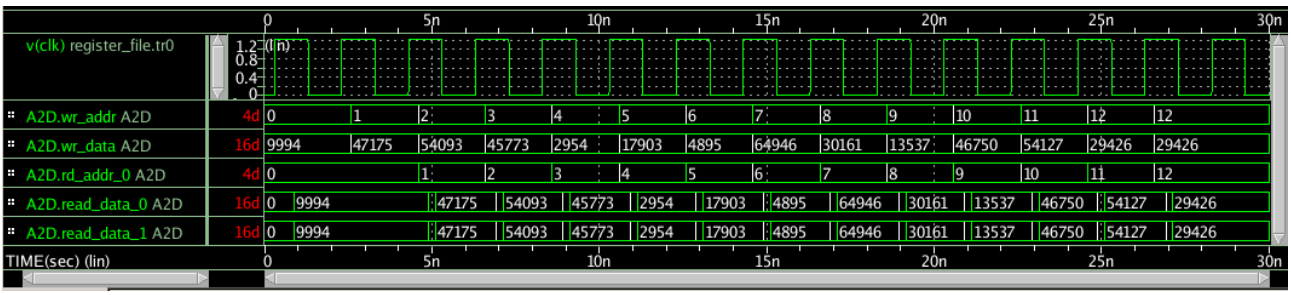

Figure 3.3 – simulation waveform

4. Layout

Figure 4.1 shows the layout design of the register file. It also pass the DRC and LVS.

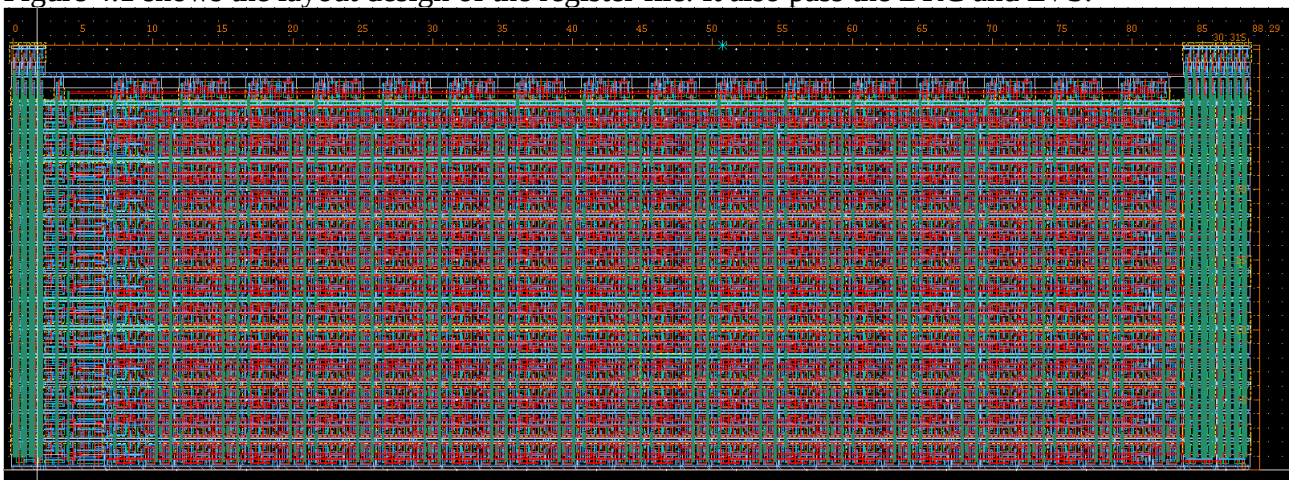

Figure 4.1 – layout design of the register file

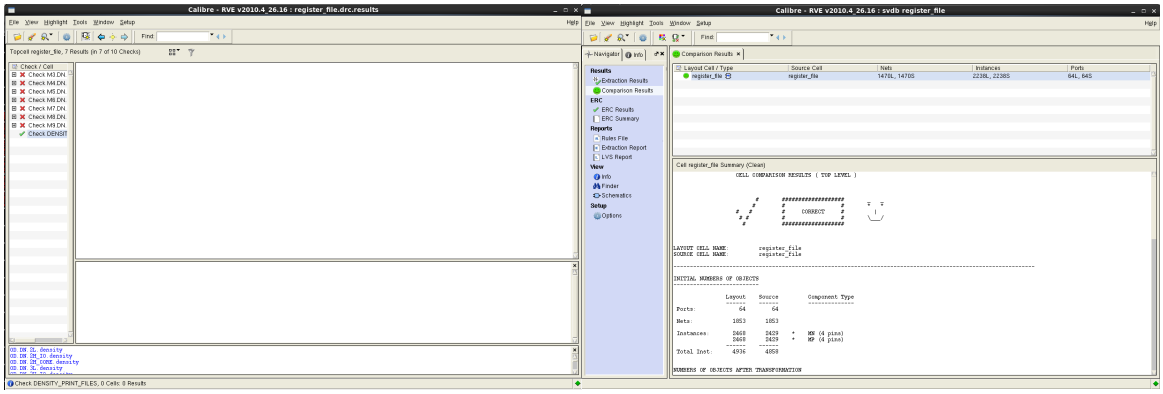

Figure 4.2 DRC AND LVS

5. Verification

(a): In order to verify that we can write and read to all the registers, we design an experiment as follows:

The register select one word line and input data in every cycle, and we use 13 clock cycles to write data to registers to every word line. And we also performs the read operation in the rising edge after the data is written into the registers. As shown on Figure 3.2, we successfully write and read the data to every registers.

(b): In order to show that our design meet the “write before read” requirement, we performs a “write before read” operation for one cycle, as shown in Figure 5.1. On the figure, we can see that the the data was written into the register file after the first rising edge, then at the second rising edge, the data can be read in the register file. Then, after the read address is selected, the data at the selected word line is read at the output.

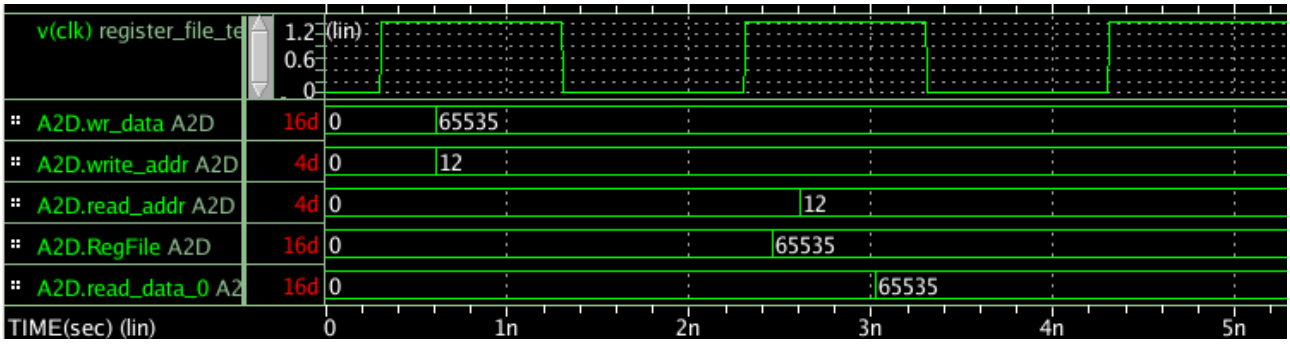

Figure 5.1 – Timing Diagram from the schematic simulation

6. Measurements

(a): The critical path for the schematic design happens when read operation and white operation happening at same "write before read" cycle at the same world line. The detail critical path is shown in figure 6.1(Starting from the decoder to read_1_0)

(b): The critical path for the layout design is same as schematic design.

(c): The energy dissipated in one write operation is **0.18 pJ**
The power dissipated in one write operation is **0.46 mW**
(d): The energy dissipated in one read operation is **0.79 pJ**
The power dissipated in one write operation is **0.46 mW**
(e): Layout width = 88.29
Layout height = 30.5
Layout area = 2693

Aspect-ratio = 2.9 : 1

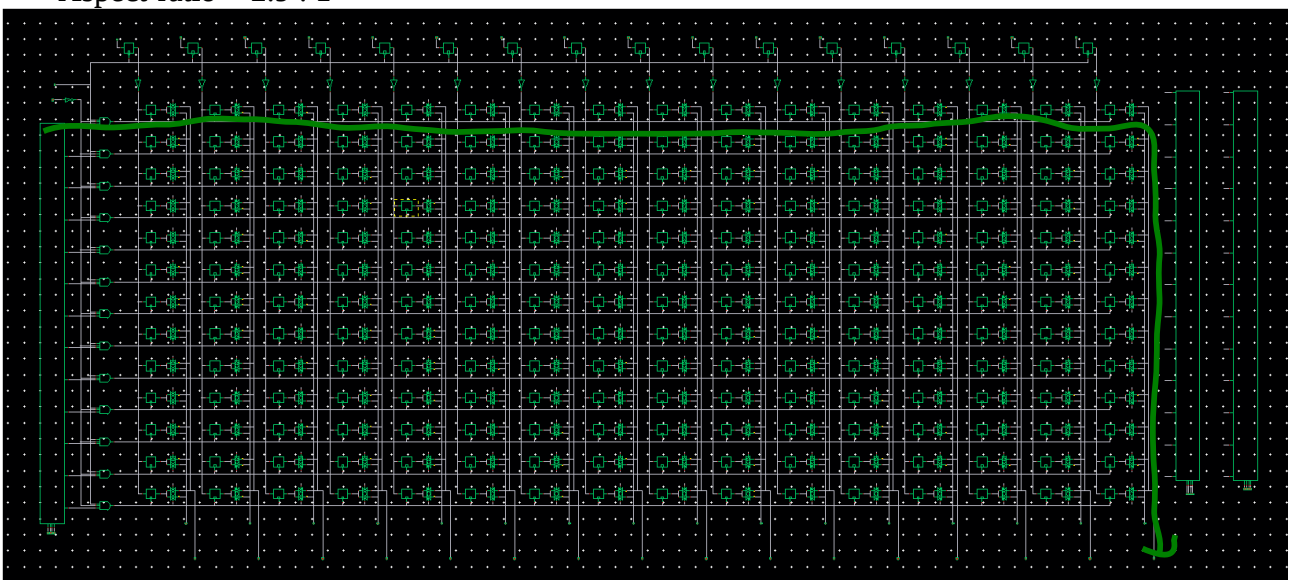

Figure 6.1 Critical Path

7. Directory

/home/projects/ee477/yuxiangc/cad2/Results
Schematic:
Control file: register_file.ctl
Netlist file: register_file.ckt
Simulation: register_file.tr0
Measurement: register_file_test.mt0
Layout:

Control file: register_file_layout.ctl
Netlist file: register_file_layout.ckt
Simulation: register_file_layout.tr0
Pex: register_file.pex.netlist / register_file.pex.netlist.REGISTER_FILE.pxi
drc: drc
lvs: lvs