Lab Report5

1. ALU

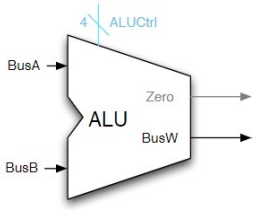


Fig 1.1 ALU

1.1 Test Result

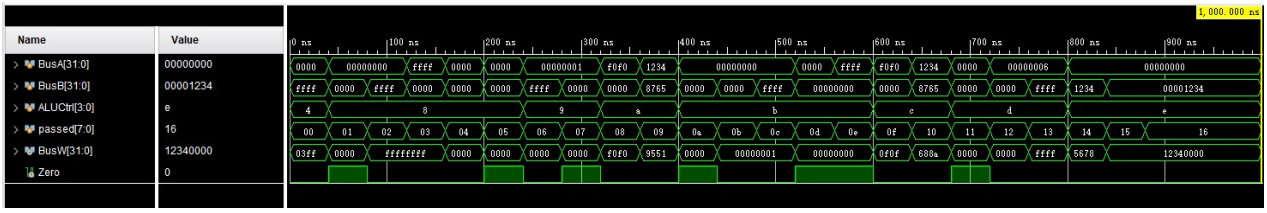


Fig 1.2 ALU test result

SRL 0xFFFF1234,6 passed

ADDU 0,0 passed

ADDU 0,-1 passed

ADDU -1,0 passed

ADDU FF,1 passed

SUBU 0,0 passed

SUBU 1,-1 passed

SUBU 1,1 passed

XOR 0xF0F0F0F0,0x0000FFFF passed

XOR 0x12345678,0x87654321 passed

SLTU 0,0 passed

SLTU 0,1 passed

SLTU 0,-1 passed

SLTU 1,0 passed

SLTU -1,0 passed

NOR 0xF0F0F0F0,0x0000FFFF passed

NOR 0x12345678,0x87654321 passed

SRA 0x00000001,3 passed

SRA 0x00001234,6 passed

SRA 0xFFFF1234,6 passed

LUI 0x12345678 passed

LUI 0x00001234 passed

All tests passed

1.2 Synthesis Log

C:/Users/songz/lab5/lab5.runs/synth_2/ALU.vds

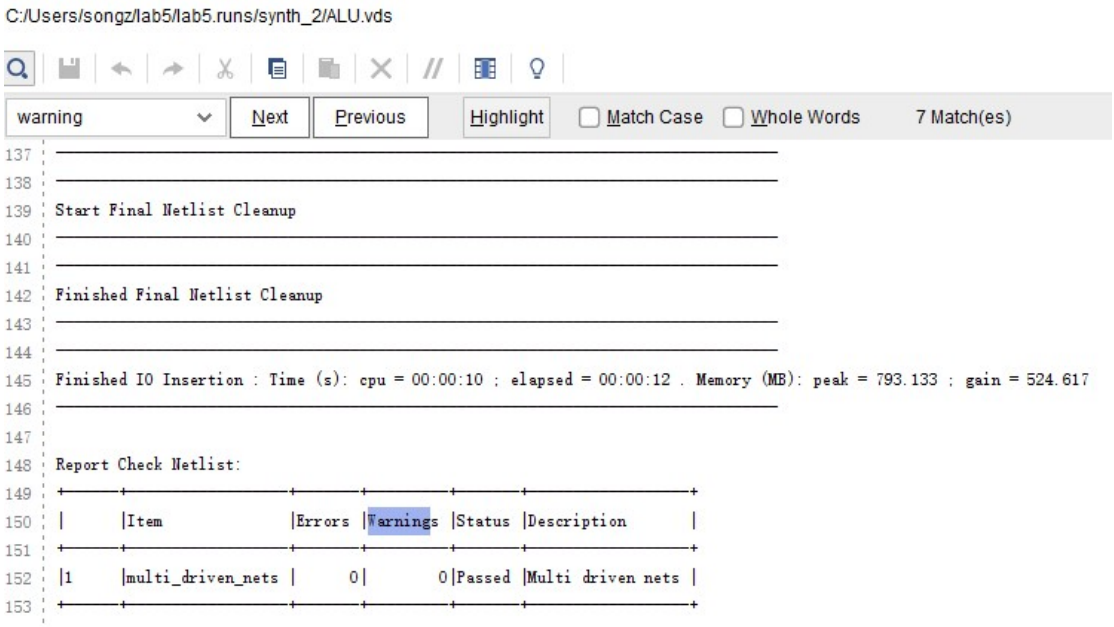


Fig 1.2 Synthesis Report

2. ALU Control

2.1 Synthesis Report

C:/Users/songz/lab5/lab5.runs/synth_2/ALUControl.vds

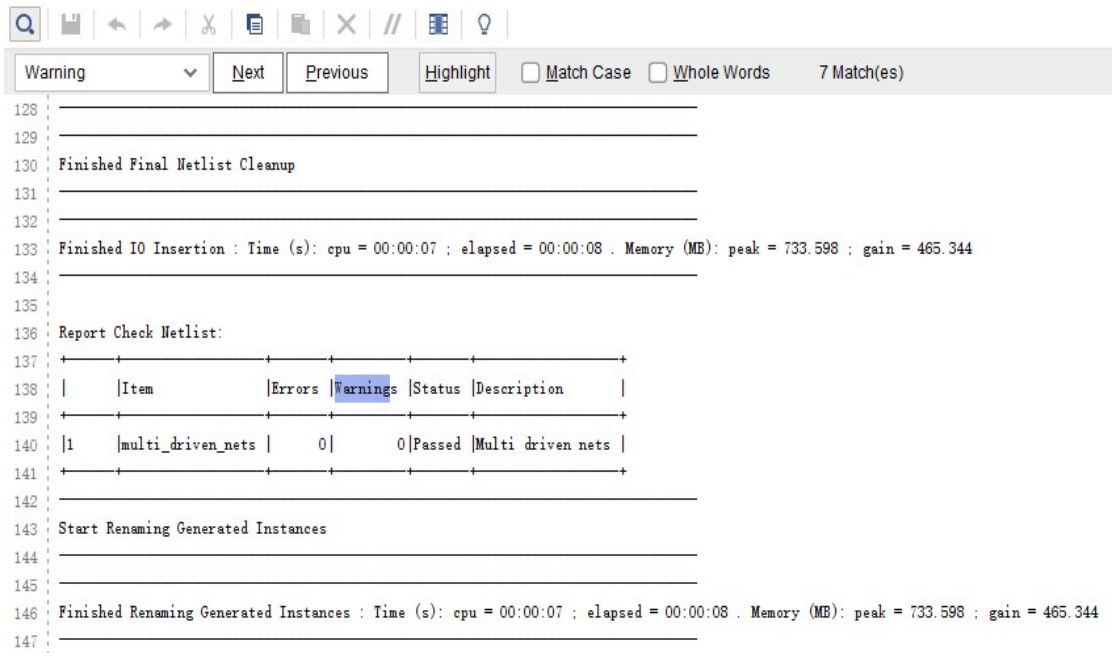


Fig 2.1 Synthesis Report

2.2 Test Result

SLL Instruction passed

SRL Instruction passed

SRA Instruction passed

ADD Instruction passed

ADDU Instruction passed

SUB Instruction passed

SUBU Instruction passed

AND Instruction passed

OR Instruction passed

XOR Instruction passed

NOR Instruction passed

SLT Instruction passed

SLTU Instruction passed

ANDI Instruction passed

ORI Instruction passed

ADDI Instruction passed

SUBI Instruction passed

SLTI Instruction passed

ADDIU Instruction passed

SUBIU Instruction passed

XORI Instruction passed

SLTU Instruction passed

NORI Instruction passed

LUI Instruction passed

All tests passed

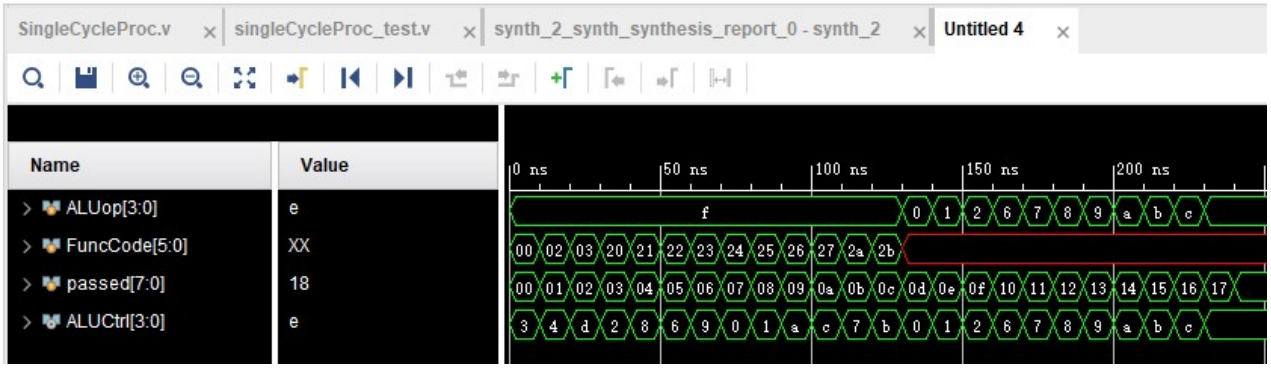


Fig 2.2 ALU Control Test Control

3. Single Cycle Control

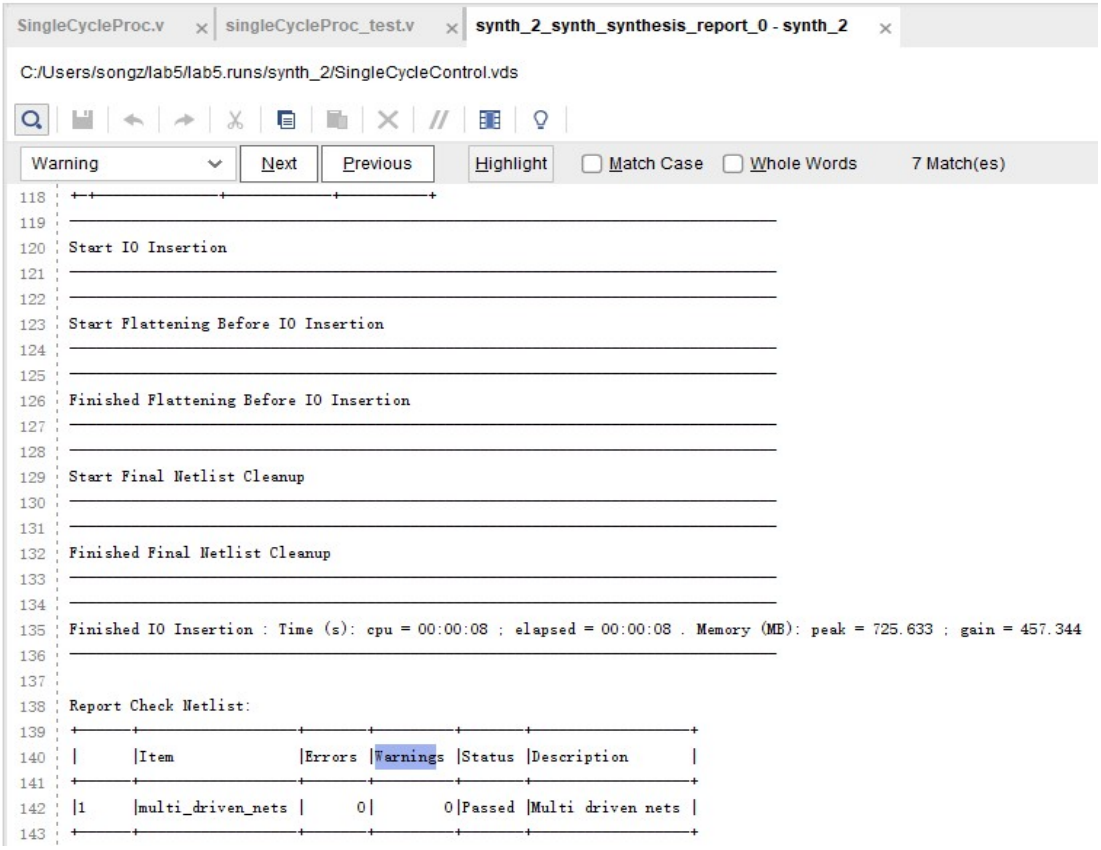


Fig 3.1 Synthesis Report

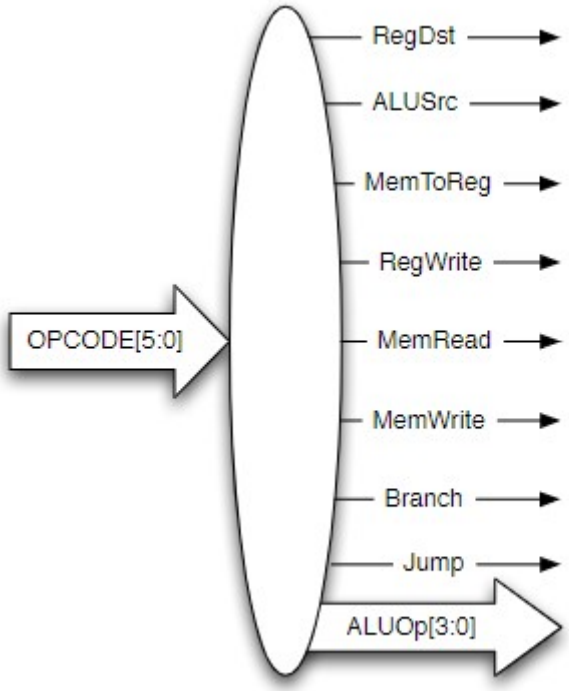


Fig 3.2 Single Cycle Controller Schematic

4. Single Cycle Processor

4.1 Synthesis Report

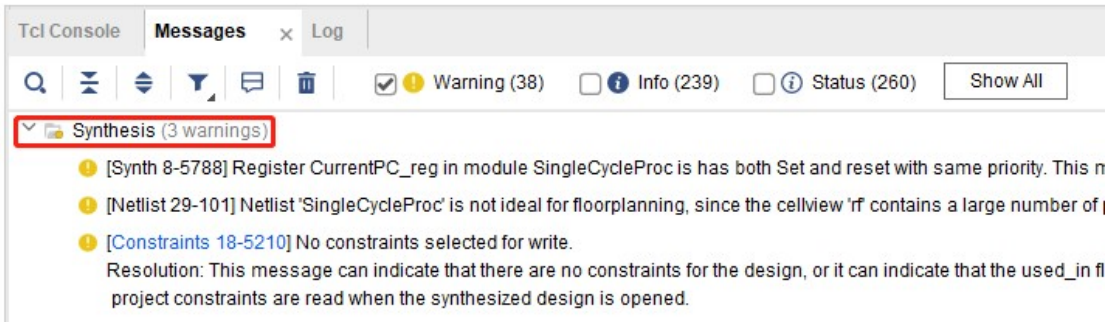


Fig 4.1 Synthesis Report

4.2 Test Result

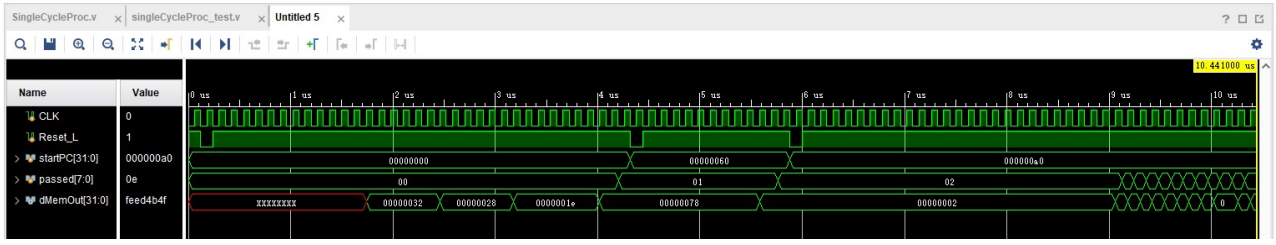


Fig 4.2 Single Cycle Processor Waveform

```
run all
                Results of Program 1 passed
                Results of Program 2 passed
              Result 1 of Program 3 passed
              Result 2 of Program 3 passed
              Result 3 of Program 3 passed
              Result 4 of Program 3 passed
              Result 5 of Program 3 passed
              Result 6 of Program 3 passed
              Result 7 of Program 3 passed
              Result 8 of Program 3 passed
              Result 9 of Program 3 passed
             Result 10 of Program 3 passed
             Result 11 of Program 3 passed
             Result 12 of Program 3 passed
All tests passed
```

Fig 4.3 Output Log

5. Questions

(a) Explain why we designed the data memory such that reads happen on the positive edge of the clock, while writes happen on the negative edge of the clock.

Because address bus in reading, and writing bus in writing, are separate. In this way, memory read and write can happen in one cycle.

(b) Take a look at the test programs provided in the instruction memory Verilog file (only test pro-grams 1 through 3) and briefly explain what each of them do and what instructions they test.

Test program1: To sum a array whose starting address is $a0, the size is $a1. Use loop to sum words in array and store the sum.

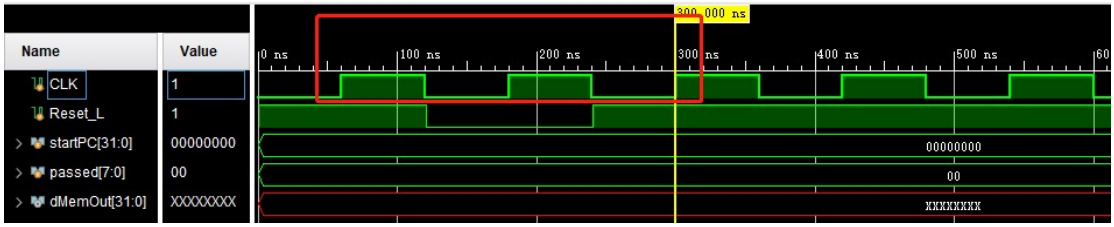
Test program2: Do some arithmetic computations and store the final result into data memory address[32].

Test program3: Do some immediate numbers calculations, and store each result in different data memory addresses. At last, load each data.

(c) If you wanted to support the bne instruction, what modifications would you have to do to your design? Include changes to all affected sub-components as well.

First, ALU Control component needs to be modified, a new case for BNE should be added. Opcode is like BEQ case. Output of controller should add a wire of BNE. Plus the wire connections near final PC needs to be modified as well.

(d) What clock rate did the synthesis process estimate your overall design would run at? Explain why this design is inefficient and provide suggestions for improvement.



Clk cycle is 120 ns. So clock rate is 8.33*10^6 Hz. This rate is so low for a processor. Thus this single cycle is inefficient.