Project 2: Single Cycle Processor

Group 5: Yeqin Mo, Shuyue Zheng, Fatemeh Pishdadian

Summary of the Main Control unit:

Op code	000000	100011	101011	000100	00010	000111	001000
	R-type	lw	sw	beq	bne	bgtz	addi
RegWrt	1	1	0	0	0	0	1
ALUsrc	0	1	1	0	0	0	1
RegDst	1	0	x	x	x	x	0
MemtoReg	0	1	x	x	x	x	0
MemWrt	0	0	1	0	0	0	0
Branch	0	0	0	1	1	1	0
Jump	0	0	0	0	0	0	0
Extop	x	1	1	x	x	x	1
ALUop<2>	1	0	0	0	0	0	0
ALUop<1>	0	0	0	0	0	0	0
ALUop<0>	0	0	0	1	1	1	0

Summary of the ALU Control unit:

Operation	Op code	ALUop	Funct	ALU action	ALUctr
add	000000	100	100000	add	0000
addu	000000	100	100001	addu	0001
sub	000000	100	100010	sub	0010
subu	000000	100	100011	subu	0011
and	000000	100	100100	and	0100
or	000000	100	100101	or	0101
sll	000000	100	000000	sll	0110
slt	000000	100	101010	slt	1000
sltu	000000	100	101011	sltu	1001
addi	001000	000	xxxxxx	add	1010
lw	100011	000	xxxxxx	add	1010
sw	101011	000	xxxxxx	add	1010
beq	000100	001	xxxxxx	sub	1011
bne	000101	001	xxxxxx	sub	1011
bgtz	000111	001	xxxxxx	sub	1011

Waveforms

Simulation waveforms of three test program are shown in Figure 1, Figure 2, and Figure 3 respectively.

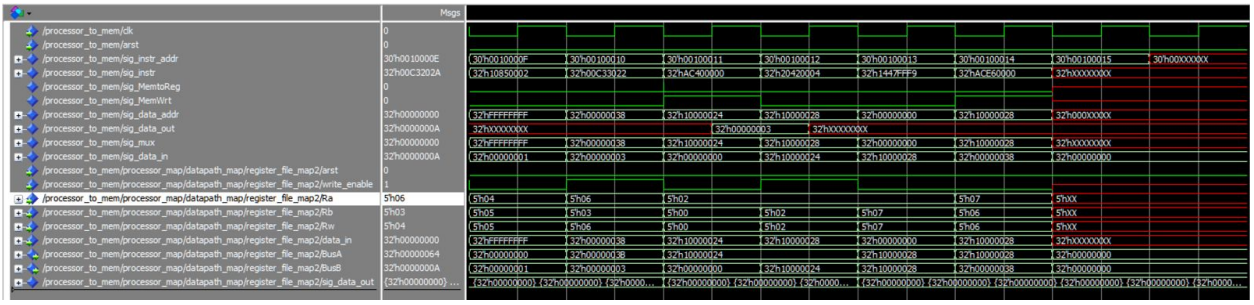

Figure 1: Waveforms of bills_branch

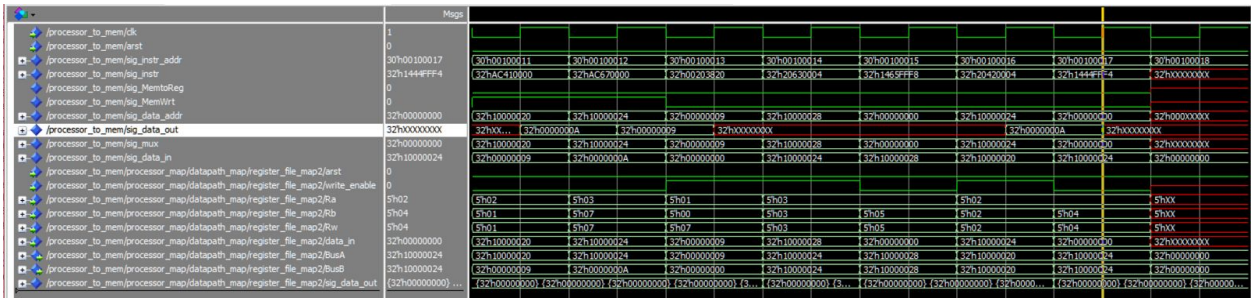

Figure 2: Waveforms of sorted_corrected_branch

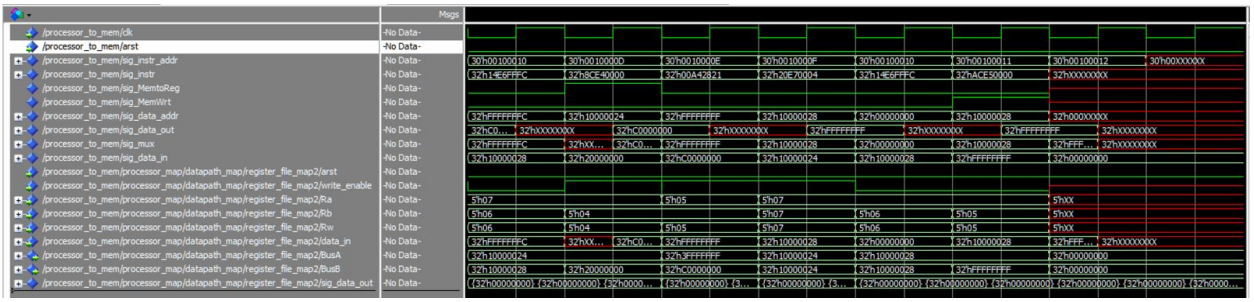

Figure 3: Waveforms of unsigned_sum

Major Challenges & Solutions

1. Branch control signal:

When we tested the processor to memory connection file, we found the bne instruction has a problem with sending control signal to instruction fetch part. The corrected instruction didn't do the jump step after the bne, our test jumped to a different address however. This error reveals the wrong branch control signal, which was an and signal between branch and zero.

For the solution, we designed a new structure for the output for this control signal. As shown in Figure 4, we used a 3 to 1 mux to choose the instruction control signals of beq, bne, and bgtz. Each of them came from the and operation among the signals of branch, zero, and the msb of alu result. The updated design gave us the correct result according to simulation.

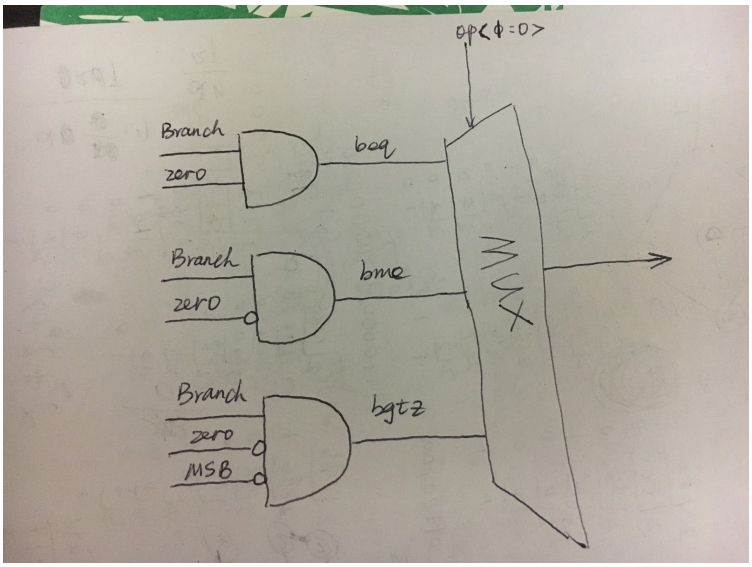

Figure 4: Branch Control Signal Design

2. Timing problem:

Register file: At the beginning, we use the rising edge to write data into the register file. But the result is that we cannot write data into register successfully because of dec_n latency. So we invert the clock and using falling edge to write data to register. So we ensure that when the falling edge arrives, Rw has been ready.

Data memory access: When we test lw instruction, we found that if we use the falling edge to write data to memory, there is a timing problem for lw instruction. That is, the data which you want to load in this cycle will appear in next cycle (one cycle delay). This made our program test a mass. Then we tested lw instruction separately and found that if we use rising edge instead, there would be only half cycle delay for lw instruction. We fixed the data memory access latency by using the rising edge of the clock.