IS1200 Lab 4

Questions for Assignment 1 (ALU)

• Explain how each of the ALU functions are defined. In particular, you need to be able to explain how subtraction works, including the use of two's complement.

Explain

• How did you implement the logic for the Zero output port? Did you consider any alternatives? Be prepared to explain your design choices.

Massive bitwise NOR that is only true when all the output bits are 0.

• What is the purpose of the ALU? Why are several functions grouped together into one component?

Its primary purpose is to perform arithmetic and logical operations on binary data. Functions are grouped together to optimize:

Speed (no need to transfer between different units)

Cost (less components)

Simplicity (reduced complexity, making it easier to manufacture and maintain)

Questions for Assignment 2 (Register File)

• Explain if the read operation or the write operation, or both operations are clocked (updated at the clock edge). Why is it implement this way?

Reads are performed without a clock signal for speed and immediacy, while writes are synchronized with a clock signal for control and coordination within the system.

• Explain the semantics of reading from and writing to $0, and how you implemented this behavior.

We did not implement writing to $0 since it should always be 0, therefore we also did not connect it to a clock signal. The only thing we did was add a register for $0 and connect it to the correspondning mux port.

• How many bits of data can this register file store? If the address width was the same size as for a complete 32-bits MIPS processor, how many bits would in such a case such register file store?

32 bits per register * 8 registers = 256 bits

Questions for Assignment 3 (Control Unit)

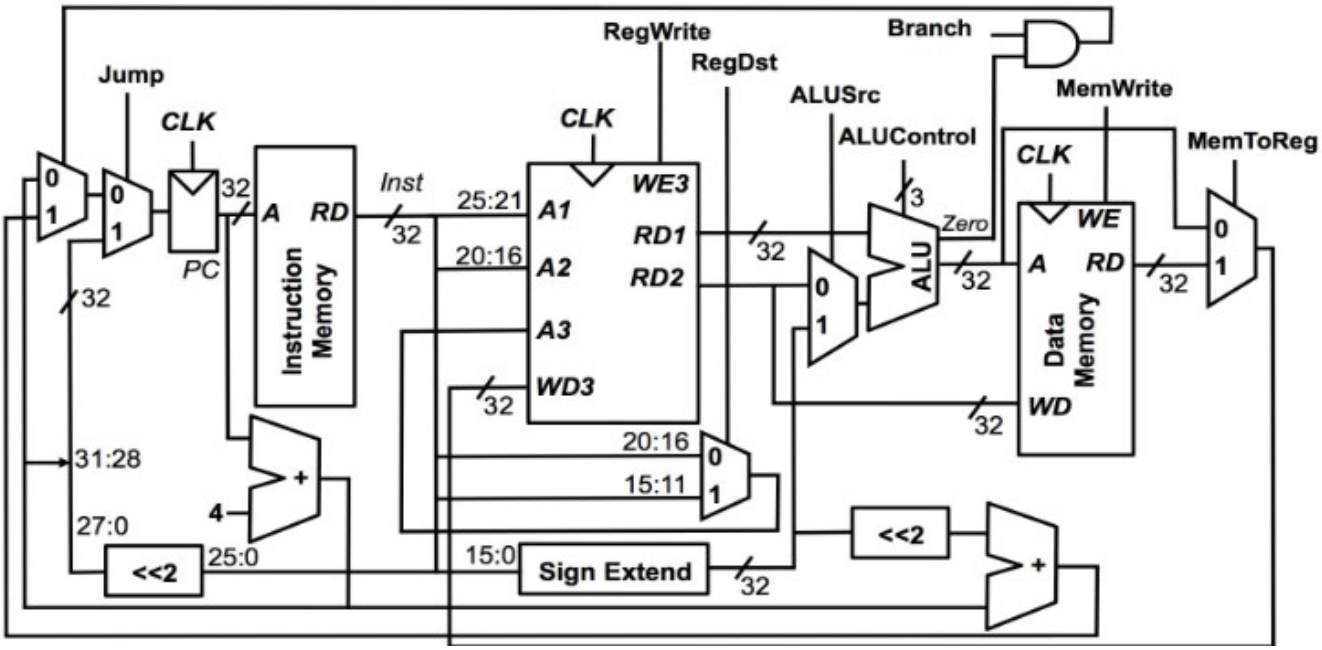
• *Explain how you have implemented the control signals for the beq instruction. Why is this a correct solution?*

ALUControl is 110 when beq instruction since the ALUOp is 01 which corresponds to ?1.

ALUOp	funct	ALUControl
00	?	010 (add)
?1	?	110 (subtract)
1?	100000 (add)	010 (add)
1?	100010 (sub)	110 (subtract)
1?	100100 (and)	000 (and)
1?	100101 (or)	001 (or)
1?	101010 (slt)	111 (set less than)

Instr	op	RegWrite	RegDst	ALUSrc	Branch	MemWrite	MemToReg	Jump	ALUOp
R-Type	000000	1	1	0	0	0	0	0	10
lw	100011	1	0	1	0	0	1	0	00
sw	101011	0	?	1	0	1	?	0	00
beq	000100	0	?	0	1	0	?	0	01
addi	001000	1	0	1	0	0	0	0	00
j	000010	0	?	?	?	0	?	1	??

• *Be prepared to explain why the RegDst control signal or the AluSrc signal is hooked up to certain signals. You should be prepared to explain this using the following figure.*



Questions for Assignment 4 (Data path)

• Explain how the bit selection works for the alternatives that are controlled by the RegDst control signal. Which instructions are using what logic and why?

All instructions are using bit selection to get the corresponding bits we're after at stages of the circuit.

• Explain how the beq instruction is implement, how the address is calculated, and how the signals are controlled by the control unit.

Branch target address:

BTA = PC + 4 + signext(imm) * 4

Questions for Assignment 5 (Factorial Function in Assembler)

• *Show and explain how the factorial function works for arbitrary input value n (the teaching assistant will give you the value that you should test). Be prepared so that you know how to change the input value easily.*

Show and explain

• *Explain how you implemented unconditional jumps in your program.*

beq $0, $0, SOME_ADDRESS

checking if zero is equal to zero is always true