Project: Design of a Pipelined Mini-RISCV Processor

In this project you will build a pipelined processor based on the following instruction set.

R-type: add, sub, and, or, xor, slt, sltu, ror

I-type: addi, andi, ori, xori, slti, sltiu, rori, ld

S-type: st

SB-type: beq, bne

JL-type: jal

JR-type: jr

The above instructions are a variation of the RISC-V instruction set discussed in class. The following changes have been made:

1. The size of the instruction is **16 bits instead of 32**.
2. The width of the datapath is **16 bits**.
3. The register file is 8 16-bit registers.
4. R-format instructions are similar to RISC-V structure with respect to number of operands. However, the "Func" fields are of different sizes.
5. I-format instructions have similar fields as in RISC-V. However, the immediate is only **five bits long**. Therefore, you need to do appropriate extension (ie. 16-bit Immediate generation unit) in order to obtain sixteen bits immediate.
6. The only Shift instruction available is "rotate right" in both R- and I-type formats. In R-type, the content of Rs1 is the data to be shifted and the content of Rs2 represents the shift amount. Only the least significant 4 bits of the content of Rs2 are used for the shift amount; other bits are ignored. Similarly, for I-type, the fifth bit of the immediate is ignored.
7. Assume the size of the PC register is 16 bits.
8. Assume that data and instruction memories are word-addressable, where the word size is 16 bits.
9. As shown in the table of the instruction format below, there are different immediate sizes depending on the format type: I- and SB-type are 5 bits, and JL-type is 10 bits.
10. Control instructions (i.e. beq, bne, jal) use PC-relative addressing as explained in class. The relative address corresponds to the number of instructions from the control instruction.
11. Use the instruction formats shown below along with the opcodes and functions given in the table below.

R-type	F2b	Rs2	Rs1	F2a	Rd	Opcode
	2-bit	3-bit	3-bit	2-bit	3-bit	3-bit
I-type	Imm[4:0]		Rs1	F2a	Rd	Opcode
	5-bit		3-bit	2-bit	3-bit	3-bit
SB-type	Imm[4:3]	Rs2	Rs1	F2a	Imm[2:0]	Opcode
	2-bit	3-bit	3-bit	2-bit	3-bit	3-bit
JL-format	Imm[9]Imm[3:0]Imm[8:4]				Rd	Opcode
	10-bit				3-bit	3-bit
JR-format	N.U.		Rs1	N.U.		Opcode
	5-bit		3-bit	5-bit		3-bit

Format	Instruction	Opcode	F2a	F2b
R-type	add	000	00	00
	sub	000	01	00
	and	000	10	00
	or	000	11	00
	xor	000	00	01
	slt	000	01	01
	sltu	000	10	01
	ror	000	11	01
I-type	ld	001	00	n.a.
	addi	001	01	n.a.
	andi	001	10	n.a.
	ori	001	11	n.a.
	xori	010	00	n.a.
	slti	010	01	n.a.
	sltiu	010	10	n.a.
	rori	010	11	n.a.
S-type	st	011	n.a.	n.a.
SB-type	beq	100	n.a.	n.a.
	bne	101	n.a.	n.a.
JL-type	Jal	110	00	n.a.
	Lui	110	01	n.a.
J-type	jr	111	n.a.	n.a.

Part I: Design of the ALU.

This part consists of designing the 16-bit ALU. You may use the ALU from the one of the previous courses without redesigning it. Just make the necessary changes to make them fit the project's requirements. The main output of the ALU is the 16-bit result. Feel free to output more signals as you see fit.

Part II: Design of the Datapath

Using the ALU from Part I, design the pipelined datapath based on the instruction set provided above.

Besides the ALU, several main components should be designed to make up the datapath of the processor; mainly register file, instruction and data memories. Other components are needed in order to connect the main parts such as busses, pipeline registers, and multiplexers.