Introduction

This project is a design of a 10-bit Custom RISK-V Microprocessor including the ALU, Register File and Main Control Unit. Our design has 10 bits as instruction code, 20 bits as Data bits and 16 bits as addresses of RAM/ROM. In this design, we are implementing 16 different operations for our CPU. There are three types of instructions which are R-type, I-type and J-Type. Most of the operations require 2 operands. There are 5 arithmetic operations and 6 logical operations along with 1 branch and 1 jump operation. The other two operations are load and store.

R-Type instructions

The instruction format for R-type is given below

Opcode	RS	RT	RD
4	2	2	2

We need 4 bits for opcode as we have 16 different operations. We have 2 bits for the register address as the number of registers in our register file is 4.

I-Type instructions

The instruction format for I-type is given below

Opcode	RS	RT	Immediate
4	2	2	2

J-Type instructions

The instruction format for J-type is given below

Opcode	Address
4	6

Function table for Main Control Unit:

Operation	Opcode	Reg Dst	Jump	Branch	Mem Read	Mem to Reg	ALU OP	Mem Write	ALU Source	Reg Write
Do Nothing	0000	0	0	0	0	0	0000	0	0	0
Load	0001	1	0	0	1	1	0101	0	1	1
Store	0010	0	0	0	0	0	0101	1	1	0
Add	0011	0	0	0	0	0	0101	0	0	1
Sub	0100	0	0	0	0	0	0110	0	0	1
Addi	0101	1	0	0	0	0	0101	0	1	1
Subi	0110	1	0	0	0	0	0110	0	1	1
And	0111	0	0	0	0	0	0001	0	0	1
Or	1000	0	0	0	0	0	0010	0	0	1
Xor	1001	0	0	0	0	0	0100	0	0	1
Not	1010	0	0	0	0	0	0011	0	0	1
Shift Left	1011	1	0	0	0	0	1000	0	0	1
Shift Right	1100	1	0	0	0	0	1001	0	0	1
beq	1101	0	0	1	0	0	0000	0	0	0
slt	1110	0	0	0	0	0	0111	0	0	1
jump	1111	0	1	0	0	0	0000	0	0	0

The Instruction Set Architecture (ISA):

Type of Instruction	Operation	Opcode	Example	Description
No Operation	No Operation	0000	nop	No Operation
R-Type	add	0011	add r3,r2,r1	r3←r2+r1
	sub	0100	sub r3,r2,r1	r3←r2-r1
	and	0111	and r3,r2,r1	r3←r2&r1
	Or	1000	or r3,r2,r1	r3←r2^r1
	Xor	1001	xor r3,r2,r1	r3←r2⊕r1
	Not	1010	not r3,r2,0	r3 ← !(r2)
	slt	1110	slt r3,r2,r1	if r1 < r2, r3 → 1 else r3→ 0
I-Type	sw	0010	sw r1, r0, 5	M[r0 + 5] ← r1
	lw	0001	lw r1, r0, 10	M[r0 + 10] → r1
	Shift Left	1011	sll r3,r1,2	r3 ← r1 << 2
	Shift Right	1100	srl r3,r1,2	r3 ← r1 >> 2
	addi	0101	addi r1,r2,10	r1←r2 + 10
	subi	0110	subi r1, r2, 10	r1← r2 - 10
	beq	1101	beq r1, r0, 3	If r1 == r0, PC ← PC + 1 + 3
J-Type	jmp	1111	jmp 15	PC ←15