Design and Implementation of RISC-V 5-stage Pipelined CPU

This project is part of the Computer Architecture (2) course

The processor will support a core set of RISC-V instructions and will be enhanced to incorporate advanced features and functionalities

The project consisted of two phases, the first was implementing the modules and the pipeline stages without any performance features. In the second phase, the performance features are added to eliminate data dependencies and handle hazards, improve branching, and handle exceptions.

No.	NAME	FORMAT	MNEMONIC	Description (in Verilog)	OPCODE/ FUNCT3/FUNCT7 or IMM in HEX
1	Add word	R	addw	R[rd] = R[rs1] + R[rs2]	33/1/20
2	Add Immediate word	I	addiw	R[rd] = R[rs1] + imm	13/0
3	And	R	and	R[rd] = R[rs1] & R[rs2]	33/7/00
4	And Immediate	I	andi	R[rd] = R[rs1] & imm	1B/6
5	Branch On Equal	SB	beq	if(R[rs1]==R[rs2]) PC=PC+{imm, 1b'0}	63/0
6	Branch On Not Equal	SB	bne	if(R[rs1]!=R[rs2]) PC=PC+{imm, 1b'0}	63/1
7	Jump And Link	UJ	jal	R[rd]=PC+4;PC= PC +{imm, 1b'0}	6F
8	Jump And Link Register	I	jalr	R[rd]=PC+4;PC= R[rs1] +imm	67/0
9	Load Half Word [1]	I	lh	R[rd]={48'bM[](15), M[R[rs1]+imm](15:0)}	03/2
10	Load Upper Imm.	U	lui	R[rd] = {imm, 12'b0}	38
11	Load Word	I	lw	R[rd] = {M[R[rs1]+imm](31:0)}	03/0
12	xor	R	xor	R[rd] = R[rs1] ^ R[rs2]	33/3/00
13	Or	R	or	R[rd] = R[rs1] | R[rs2]	33/5/00
14	Or Immediate	I	ori	R[rd] = R[rs1] | imm	13/7
15	Set Less Than	R	slt	R[rd] = (R[rs1] < R[rs2]) ? 1 : 0	33/0/00
16	Shift Left	R	sll	R[rd] = R[rs1] << R[rs2]	33/4/00
17	Shift Right	R	srl	R[rd] = R[rs1] >> R[rs2]	33/2/00
18	Store Byte	S	sb	M[R[rs1]+imm](7:0) = R[rs2](7:0)	23/0
19	Store Word	S	sw	M[R[rs1]+imm](31:0) = R[rs2](31:0)	23/2
20	Subtract	R	sub	R[rd] = R[rs1] - R[rs2]	33/6/00

CORE INSTRUCTION FORMATS

	31 27	26 25	24 20	19 15	14 12	11 7	6 0
R	funct7		rs2	rs1	funct3	rd	Opcode
I	imm[11:0]			rs1	funct3	rd	Opcode
S	imm[11:5]		rs2	rs1	funct3	imm[4:0]	opcode
SB	imm[12|10:5]		rs2	rs1	funct3	imm[4:1|11]	opcode
U	imm[31:12]					rd	opcode
UJ	imm[20|10:1|11|19:12]					rd	opcode

Core Processor Design:

- PC
- Instruction Memory (64K x 1 byte)
- Control Unit
- Immediate Gen
- Register File
- ALU
- Data Memory (8K x 1 byte)

Pipeline Management

- IF stage
- ID stage
- EXE stage
- MEM stage
- WB stage

Performance Features

- Dynamic branch prediction (bit predictor to enhance branching accuracy)
- Hazard detection unit for detecting control hazards
- Forwarding unit to manage dependencies between instructions (ALU-ALU and MEM-ALU forwarding)
- Exception detection unit

Codes in the repository contain the core design modules, additional modules, pipeline stages, modules added for phase2, and their corresponding testbenches