Project 3: Designing a 32-bit CPU

Adam Sumner - A20283081, **Contribution** - 25%
Bobby Unverzagt - A2028923, **Contribution** - 25%
Emilie Woog - A20265269, **Contribution** - 25%
Nash Kaminski - A20283999, **Contribution** - 25%

ECE 485

December 5th, 2015

1 Introduction

This goal of this project is to design a stripped down version of the MIPS processor. The processor will be a 32-bit version of the processor discussed in class and the text book, however, its instruction set will be a small subset of the MIPS processor's full capability.

1.1 Background Information

MIPS is a reduced instruction set computer (RISC) instruction set architecture (ISA). It defines three types of instruction types: R (register), I (Immediate), and J (Jump). For the implementation that this project is focused on, only R and I instructions will be executed. R type instructions are the most common form of instructions. The format for an r-type instruction is:

`Bits[31:26]`	`Bits[25:21]`	`Bits[20:16]`	`Bits[15:11]`	`Bits[10:6]`	`Bits[5:0]`
opcode	Rs	Rt	Rd	shamt	funct

For this instruction, the opcode field is always 000000_2, while the function code `funct` is used to determine which instruction is to be carried out. Rs

and Rt are the two registers in which the operation reads and Rd is the destination of the result. Some instructions require a shift amount (`shamt`), so it is specified explicitly.

The I type instruction involves an immediate value, so the instruction format must accommodate this. The format of this type of instruction is:

Bits[31:26]	Bits[25:21]	Bits[20:16]	Bits[15:0]
opcode	Rs	Rt	immediate

For this isntruction, the op code field is used to define the specific instruction, Rs is the register in which the operation acts on along with the immediate value as the other operand. Rt is the destination register in which the result is stored.

2 Design

2.1 Instruction Set

Table 1 shows the instructions that are implemented in the CPU with the respective OpCode and Function Field for each instruction.

OpCode[31:26]	Function Field [5:0]	Instruction	Operation
100011_2	--	lw	lw $t3, 200($s2)
101011_2	--	sw	sw $t4, 100($t3)
000000_2	100000_2	add	add $s3, $t2, $s2
000000_2	110000_2	sub	sub $s3, $t2, $s2
000100_2	--	beq	beq $s5, $s2, 500
000000_2	000001_2	nand	tbd
000010_2	--	andi	tbd
000000_2	000001_2	or	tbd
000011_2	--	ori	tbd

Table 1: CPU Instruction Set

2.2 Memory

For this project, it seemed unnecessary to implement memory of 4GB (2^{32}). It was chosen to use an array of 256 words instead. If need be, this memory size could be upgraded easily, so this choice does not hinder performance on the actual design of the CPU.

2.3 DataPath

2.4 Control

3 Analysis

4 Simulation Results

5 Conclusion

The design and implementation of a 32-bit CPU was a success. An architecture was designed and a behavior was successfully put into practice. All requested functionality was achieved. This 32-bit CPU can now be used in further projects.