Kwame Nkrumah University of Science and Technology, Kumasi, Ghana.
College of Engineering

Department of Computer Engineering

COE 381: Microprocessors

Microprocessor Circuit Design

Group Work By Computer Engineering 3
Group 1

October 3, 2019

GROUP MEMBERS

NAME	INDEX NUMBER	SIGNATURE
Akudbilla Daniel	9348417	
Gyimah Francis	9343117	
Doe Godfred	9348517	
Owusu Amponsah Clifford	9346517	
Appiah Augustine	9339917	
Dunyo Kwame John	9348017	

TEAMWORK AND PARTICIPATION

The entire team worked together over 7 hours every week. This document shows the process we went through in designing the CPU(Microprocessor).

ABOUT CEMGA

CEMGA is a 16 bit microprocessor, based on the RISC architecture, Unicyle and Harvard architecture. The entire CPU was designed using logisim.

ARCHITECTURE

The architecture of this microprocessor is based on the MIPS architecture, which is the most popular implementation of the RISC architecture and inspired by the Intel 8086 microprocessor design.

The processor is designed to be as simple as possible and hence does not include things like pipeline, branch prediction, cache among others.

The overall project is divided into 4 main parts.

- The ALU ⟶ Responsible for all the mathematical and logical operations of the processor
- The Control Unit ⟶ Responsible for creating control flags throughout the processor, according to the current running instructions
- Register Banks ⟶ Registers for the CPU to store information.
- The CPU ⟶ The final module, which is an assembly of all the other modules.

THE ALU

The ALU is designed to be as simple as possible. It accepts two input data, both being 16 bit in width, and depending on the instruction given it, an output is generated.

The ALU basically performs 9 main operations, which are

1. Or operation
2. Not operation
3. And operation
4. Xor operation
5. Addition operation
6. Subtraction operation
7. Multiplication operation
8. Division operation
9. Move operation, to move the next immediate instruction to the registers.

THE CONTROL UNIT

The control unit is the heart of the entire CPU. It's main operation is to generate control signals and flags throughout the CPU to control when certain operations are performed.

The CU(Control Unit) performs the following operations

- It sets flag for the immediate present, i.e, next instruction to be executed
- Sets the status flag for the entire CPU
- It sets signal to allow reading and writing from the register
- It sets signal to control the jumping operation of the PC (determining which instruction to move to next)
- It generates signal to either switch on or off the entire CPU.

THE REGISTER BANKS

The register banks is a combination of 4 registers, grouped together to act as the memory of the entire CPU.It also acts the accumulator for the ALU. The 4 registers are named A, B, C and D.

THE CPU

The CPU is the assembly of the all the other parts. It is in the CPU where all the operations of the microprocessor are performed. A sequence of hexadecimal instructions are fed into the instruction memory of the CPU which determines from which the CU coordinates all the other parts of the microprocessor.

THE INSTRUCTION SET

The microprocessor instruction set is divided into 3 main parts. The first 4 bits is the opcode, this shows the instruction to be executed.

The second 4 bits selects the register which will be written to or read from. The first 2 bits of the 4 bits for the register selects which register will be written to as well as read from the register input and output respectively. The other 2 bits selects an additional register we would want to also read from the register 2 output.

The last 8 bits of the 16 bits instructions also shows the location of the next instruction which the program counter(PC) would point to.

The table below shows the Instruction set of the opcode of our microprocessor.

OPCODE	INSTRUCTION	MEANING
0000	OR	Or operation
0001	NOT	Not operation
0010	AND	And operation
0011	XOR	Xor operation
0100	ADD	Addition operation
0101	SUB	Subtraction operation
0110	MULT	Multiplication operation
0111	DIV	Division operation
1000	MOV	Move operation
1001	MOV	Move operation
1010	LOAD	Read from memory
1011	STORE	Write to memory
1100	JMPZ	Jump if zero
1101	JMPN	Jump if negative
1110	JMPP	Jump if positive
1111	HALT	Stop the entire CPU