Calvin Passmore ECE 6470

Simple MIPS

Specification

I propose to design and implement a verify simple Microprocessor without Interlocked Pipelined Stages (MIPS). This means that each stage of the processor happens within one clock cycle, and the clock will run slowly. I also propose using a small set of instructions to simplify the assembly code. The instructions I propose to support are shown in the below table.

Name	**Meaning**
NOOP	Do nothing
Load	Load from memory into register
Load Number	Load a scalar into the register
Store	Store from the register into memory
Add	Add two registers
Subtract	Subtract two registers
XOR	XOR two registers into a single register
And	AND individual bits in two registers into one register
Jump	Jump to address
Jump if non 0	Jump to address if the given register is not 0
Push	Push the register value onto the stack
Pop	Pop the top value off the stack

I propose to use a small number of registers: one each for the current address, current instruction, stack address, and return value register, as well as 12 general purpose registers resulting in a total of 16 registers.

All instructions will be 32 bits long in the format shown in the table below. An example of using the ADD instruction would be (ADDOPCODE)(ReturnReg)(GenReg0)(GenReg1). This instruction would add the values in general purpose registers 0 and 1 and put the result in the return value register. An example of Load would be (LOADOPCODE)(GenReg0)(Unused)(MemoryAddress). This instruction would load the value at the memory address into the general purpose register 0.

4 bits	8 bits	10 bits	10 bits
OPCODE	Destination	Source 1	Source 2

Team Members

I will be working on this project alone.

Estimated Tasks

- Create the instruction fetch block
- Create the info prep block (getting register values ready for the ALU)
- Create the ALU block
- Create the memory write block
- Simulate the processor with assembly code
- Debug the processor and re-simulate
- Synthesis
- Placement & Routing

Projected Timeline

Instruction Fetch Block * Info Prep Block

- 3/21 - 3/23

ALU & Memory

- 3/23-3/30

Synthesis

- 3/31 - 4/4

Placement & Routing

- 4/5 - 4/11

Buffer Time

- 4/12 - 4/25