MIPS Semantics

Principles

1. Every instruction takes five cycles to execute
2. There is no overlap between instructions in this implementation
3. There are separate instruction and data memories (Harvard Architecture)
4. The only control instruction we are implementing is BEQ (branch to PC+4+Imm if indicated register is zero)
5. For instruction formats and mnemonics, refer to

   http://max.cs.kzoo.edu/cs230/Resources/MIPS/MachineXL/InstructionFormats.html

6. Note the following:
   rs is bits 21-25 of the instruction register
   rt is bits 16-20 of the instruction register
   rd is bits 11-15 of the instruction register
   the immediate field is bits 01-15 of the instruction register
   the opcode is bits 26-31 of the instruction register
   the function code is bits 0-5 of the instruction register

Note the overlap. Some instructions use the lower 16 bits as an immediate operand, while others use the bits for a second operand register number (rt) and a function code (for alu operations).

Cycle by Cycle

Cycle 1: Instruction fetch (IF). Apples to all instructions

IR <- InstructionMemory[PC]
NPC <- PC + 4

Cycle 2: Instruction decode (ID)/register fetch for all instructions. For ALU operations:

A <- Registers[rs]
B <- Registers[rt]
Imm <- sext(immediate field of IR)

Cycle 3: Execute (EX) ALU op/ compute effective address (load/store/jump)

For memory reference operations:

ALU Output <- A + Imm

For register-register ALU operations:

ALU Output <- A func B

Where func is specified by the opcode and function code. For Register-immediate ALU operations:

ALU output <- A op Imm

Where op is specified by the opcode. For branch instructions

ALU output <- A + (Imm >>2)
Zero <- (A == 0)

Cycle 4: Memory access (MEM)/branch completion

For load instruction:

LMD <- DataMemory[ALU output]

For store instruction:

DataMemory[ALU output] <- B

For all other instructions nothing happens in this cycle.

Cycle 5: Write back (WB):

For ALU operations (reg-reg or reg-imm):

R*egisters[rd] <- ALU output*

For Load:

Regs[rd] <- LMD

For all instructions except taken branch

PC <- NPC

For taken branch

PC <- ALU output