24. RISC Pipelining

Chapter 15
sections 15.5
MIPS Instruction Set

- The MIPS instruction set was designed for pipeline execution
  - MIPS instructions are the same length. x86 instructions vary from 1 byte to 17 bytes and pipelining is much more challenging.
  - MIPS has only a few instruction formats, with the source operand being located in the same place in each instruction.
  - Memory operands only appear in loads or stores in MIPS.
  - Operands must be aligned in memory.
MIPS Instructions

MIPS instructions classically take five steps:

- Instruction Fetch (IF)
- Instruction Decode & Register Fetch (ID)
- Execution / Effective Address (EX)
- Memory Access (MEM)
- Write Back Results (WO)
Non-pipelined RISC Processor (MIPS)
5 stages
Instruction Fetch (IF)

- **IF (Instruction Fetch)**
  \[
  IR \leftarrow M[PC] \\
  NPC \leftarrow PC + 4
  \]

- IR: Instruction Register
- M: Memory
- PC: Program Counter
- NPC: Next Program Counter

- Fetch the instruction
- Update program counter
Instruction Decode (ID)

- **ID (Instruction Decode/Register Fetch)**
  - A ← Regs[rs]
  - B ← Regs[rt]
  - Imm ← sign-extend immediate field of IR

- Decoding can be done in parallel with reading registers

- In an aggressive implementation the branch can be completed at the end of this stage as we will see later
Execution (EX)

- Performs one of the following:

  - Memory Reference
    - ALUOutput $\leftarrow A + \text{immediate}$

  - Register-Register ALU instruction
    - ALUOutput$\leftarrow A \text{ opcode } B$

  - Register-Immediate ALU instruction
    - ALUOutput$\leftarrow A \text{ opcode } \text{Imm}$

  - Branch
    - ALUOutput$\leftarrow \text{NPC} + (\text{Imm} \ll 2)$
    - Cond $\leftarrow (A==0)$
Memory Access (MEM)

- Memory Reference
  - LMD <- Mem[ALUOutput] ; load from memory
  - Mem[ALUOutput] <- B ; store to memory

- Branch
  - if (cond) PC <- ALUOutput
Write Back (WB)

- Register-Register ALU
  - Regs[rd] \rightarrow \text{ALUOutput}

- Register-Immediate ALU
  - Regs[rt] \rightarrow \text{ALUOutput}

- Load instruction
  - Regs[rt] \rightarrow \text{LMD}
WINMIPS64
WinMIPS64

- WinMIPS64 is an instruction set simulator

- You can:
  - Load MIPS programs
  - Execute one cycle at a time
  - Visualize the pipeline

- You can download the software and read the documentation here:
  - [http://indigo.ie/~mscott/](http://indigo.ie/~mscott/)
WinMIPS64 Example

• Open notepad, copy the following into a new notepad file, and save it as test.s

• Copy WinMIPS64 from the CS 430 public folder

• Open WinMIPS64

• From WinMIPS64, open test.s

• Step through the program using F7