ECEn 528
Study Guide – Instruction Set Architecture

- Read Sections A.1-A.11 of H&P
  - Things to focus on
    - The different options while designing instruction sets.

  - Clarifications
    - The authors of the textbook were at the heart of the RISC/CISC debates, and are a bit partisan in the way that they state features "we would expect in a modern ISA".
    - Similarly, the creators of the multi-media instructions would likely disagree strongly with H&P's assessment in section A.8 that SIMD instructions are a failure.
      - The example is misleading, because it depends upon the format of the data.
    - Don't worry too much about the details of the MIPS instruction set in section A.9

  - Answer the following questions:

    1. Why are stack and accumulator-based machines rare now?

Registers are faster than memory and more efficient for a compiler to use than other forms of internal storage

    2. What are the pros and cons of having a large number of addressing modes?

More modes allows for fewer instructions, but increase complexity and may increase average CPI

    3. What are the advantages and disadvantages of using conditions codes vs. condition registers for branches?

Using condition codes is good because the condition is sometimes set for free, but bad because CC is extra state and therefore constrains the ordering of instructions.
Using condition registers is good because it's simple but bad because it uses up a register.

4. How might you go about supporting 32-bit immediates in a 32-bit fixed-encoding instruction set? How about a variable-encoding instruction set?

Set low 16, set high 16, add. Or set low 8, shift, set low 8, shift, etc.
Simply extend the instruction

5. Why might it be useful to have different instruction formats for different classes of instructions?

Smaller instruction sizes can reduce overall code size, and different formats with different operands require different logic for processing, so splitting the formats into different instructions can simplify the logic