Midterm

Format

- 20 - 25 questions
- Multiple choice
- True/false
- Matching
- Fill in the blanks
- Short answer questions
- 1 hour 15 minutes
    - Whole class time

Notes

- One sided A4 cheat sheet
- MIPS reference sheet

Topics

- Assembler directives
    - `.asciiz`
    - `.data`
    - `.word`
    - `.half`
    - `.byte`
- Register range with `n` bits
    - Unsigned
        - $2^n - 1$
    - Signed
- Symbol tables
- One pass assembler
    - May use more memory
- Two pass assembler
    - First pass
    - Second pass
- Hi and lo registers
    - Hi
    - Lo

- Data movement
  - `mthi`
  - `mtlo`
  - `mfhi`
  - `mflo`
- Data addressability
- Logical Instructions
  - `sll`
    - Specify how far to move
    - Appends zeros to the end (right side)
    - Cut off same amount at start
    - Shifting left by one multiplies the value by two
  - `srl`
    - Specify how far to move
    - Appends zeroes to the start (left side)
    - Cut off same amount at end
  - `sra`
    - Adds prefix s zeroes or ones depending on the signed arithmetic value
    - Start with 1
      - Put 1s
    - Start with 0
      - Put 0s
  - `sla`
- Endian
  - Big endian
    - Start at most significant bit
    - Memory going down
      - DE
      - AD
      - BE
      - EF
  - Little endian
    - Start at least significant bit
    - Memory going down
      - EF
      - BE
      - AD
      - DE
- Memory addressing

    - Example
        - A memory system has 64 GB of storage
        - Memory system has 34 address pins
        - What can you infer about its addressability
            - 64 GB = $2^6 * 2^30$ Bytes = 2^{36} Bytes
            - With 34 pins you can address 2^{34} locations
            - We cannot address all bytes individually
            - We can address 2^{34} locations or 2^{36} bytes divided by 4
            - Therefore we can address 4 bytes, which is 1 word
            - Therefore, this memory is word addressable
            - Other types of addressability can be byte, half-word, word, or double word addressability
- Jump instruction
    - Computing target from jump location
    - Computer jump location from target