Microarchitecture

- Implements the ISA (instruction set architecture)
  - An ISA specifies instructions that the CPU provides
- Microprogram
  - Program within the CPU
  - Think of each ISA instruction as a function implemented by the microprogramming
  - The master microprogram is an infinite loop which figures out what to execute next and executes it
- IJVM
  - Example ISA written for this book
  - Only implements integer instructions
  - Mic-1 is the example microarchitecture

- Instructions
    - Instructions contain fields
        - opcode (operation code) - identifies the instruction (ADD, NOT, etc)
        - Operand(s) - variable(s) on which to operate
        - One binary number, different sections mean different things
- Data path (Fig 4-1)
    - Contains the ALU, its inputs, and its outputs
- MR-1 ALU (Fig 3-18)
    - 6 control lines
    - 2 inputs (buses A + B)
    - B input can be from one of the 9 registers attached to the bus
    - A input is register H
    - Loading H
        - Pass the value to load through B
        - Use addition, except with ENA set to 0 to force A to 0

- ALU control lines (fig 4-2)
- Data path timing (fig 4-3)
  - Steps
    - Control registers set up
    - Registers are loaded onto B bus
    - ALU and shifter operate
    - Results propagate on C bus back to registers
  - Timing within a clock cycle is implicit, timed by the propagation delay of the circuits

- Memory operations
  - 32-bit addressable memory port
    - Controlled by MAR (memory address register) and MDR (memory data register)
      - Put the address of the word you want in MAR
      - Value is placed in MDR
  - 8-bit byte-addressable memory port
    - Controlled by PC register, reads 1 byte into lower order byte of MBR
  - MAR contains word addresses, PC contains byte addresses
    - Putting a 2 in PC and starting a memory read will read byte 2 from memory and put it in the low order bits of MBR
    - Putting a 2 in MAR and starting a memory read will read bytes 8-11 from memory and put them in MDR