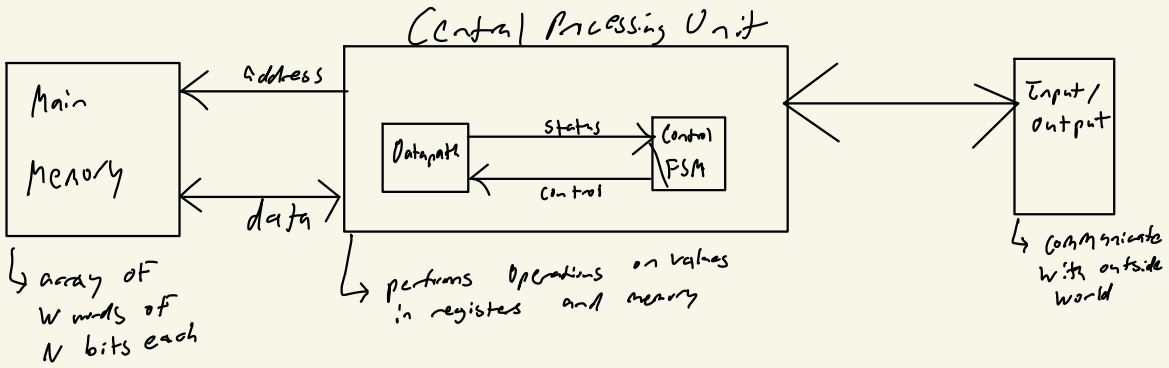
# Designing an Instruction Set

Ch 9

# Von Neuman Model




- Express program as a seq of coded instructions
  - ↳ memory holds data and instructions → binary data
  - ↳ CPU fetches, interprets, and executes successive instructions of the program

## Instructions

- each instruction has a
- opcode to specify the operation to be performed
- source operands and destination for the result
- in the von Neumann, instructions are executed sequentially

Program Counter
↳ address of instruction to be executed

↳ ① Fetch instruction → ② decode instruct.
② → ③ read src operands
③ → ④ execute
④ → ⑤ write dst operands → ⑥ compute next PC

# Instruction Set Architecture

- contract between soft./hard.
  - functional def. of operations and storage locations
    - precise description of how soft. can invoke/access them
  - ISA specifies what hardware does (not implementation)

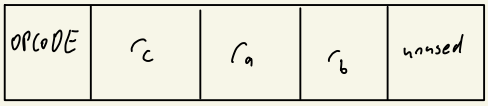
- design
  - how many operations?
  - what type/how much storage
  - how to encode instructions?
  - how to future proof?

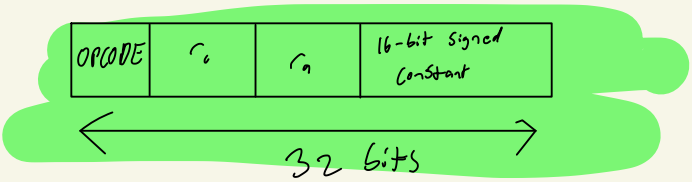
} optimize quantitatively against benchmark of common case (usage)

# Beta ISA

- Storage
  - processor: 32 registers
  - main memory: 32-bit byte addresses
    - 4 bytes/word implies addresses are multiples of 4

- Instruction Format:




  - register and constant format
    - Loads/stores → access memory
    - Branches/Jumps → change PC