Computer Science 210

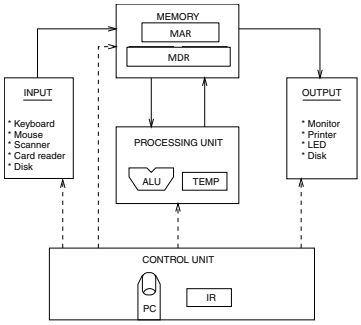
Computer Systems 1

Lecture 10

The von Neumann Computer

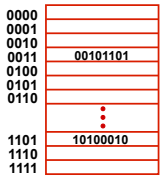
Credits: Slides prepared by Gregory T. Byrd, North Carolina State University

Von Neumann Model



Memory

- 2^k x m array of stored bits
- Address
  - unique (k-bit) identifier of location
- Contents
  - m-bit value stored in location (register)



- Basic Operations:
- **LOAD**
  - read a value from a memory location
- **STORE**
  - write a value to a memory location

Interface to Memory

- How does processing unit get data to/from memory?
- MAR: Memory Address Register
- MDR: Memory Data Register

- To LOAD a location (A):
  1. Write the address (A) into the MAR
  2. Send a "read" signal to the memory
  3. Read the data from MDR
- To STORE a value (X) to a location (A):
  1. Write the data (X) to the MDR
  2. Write the address (A) into the MAR
  3. Send a "write" signal to the memory

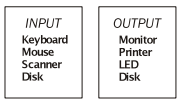
Processing Unit

- Functional Units
  - ALU = Arithmetic and Logic Unit
  - could have many functional units. some of them special-purpose (multiply, square root, ...)
  - LC-3 performs ADD, AND, NOT
- Registers
  - Small, temporary storage
  - Operands and results of functional units
  - LC-3 has eight registers (R0, ..., R7), each 16 bits wide
- Word Size
  - number of bits normally processed by ALU in one instruction
  - also width of registers
  - LC-3 is 16 bits

Input and Output

Devices for getting data into and out of computer memory

Each device has its own interface, usually a set of registers like the memory's MAR and MDR



- LC-3 supports keyboard (input) and monitor (output)
- keyboard: data register (KBDR) and status register (KBSR)
- monitor: data register (DDR) and status register (DSR)

Some devices provide both input and output
- disk, network

Program that controls access to a device is usually called a *driver*.

Control Unit

Orchestrates execution of the program

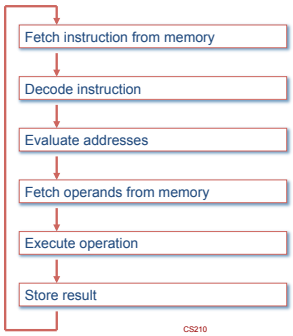
Instruction Register (IR) contains the *current instruction*

Program Counter (PC) contains the *address* of the next instruction to be executed

Control unit:

- reads an instruction from memory
  - the instruction's address is in the PC
- interprets the instruction, generating signals that tell the other components what to do
  - an instruction may take many *machine cycles* to complete

Instruction Processing



Instruction

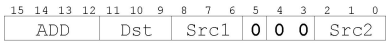
- The instruction is the fundamental unit of work.
- Specifies two things:
  - ***opcode***: operation to be performed (e.g. ADD)
  - ***operands***: data/locations to be used for operation

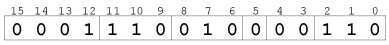
- An instruction is encoded as a **sequence of bits** *(Just like data!)*
  - Often, but not always, instructions have a fixed length, such as 16 or 32 bits.
  - Control unit interprets instruction: generates sequence of control signals to carry out operation.
  - Operation is either executed completely, or not at all.

- A computer's instructions and their formats is known as its ***Instruction Set Architecture (ISA)***.

Example: LC-3 ADD Instruction

- LC-3 has 16-bit instructions.
  - Each instruction has a four-bit opcode, bits [15:12].
- LC-3 has eight *registers* (R0-R7) for temporary storage
  - Sources and destination of ADD are registers

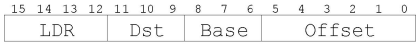


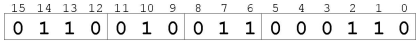


"Add the contents of R2 to the contents of R6, and store the result in R6."

Example: LC-3 LDR Instruction

- Load instruction – reads data from memory
- Base + offset mode:
  - add offset to base register – result is memory address
  - load from memory address into destination register





"Add the value 6 to the contents of R3 to form a memory address. Load the contents of that memory location to R2."