Lecture 17 - State Machines cont'd & von Neumann Arch.

The von Neumann computer

- historically, the first computers were *fixed program* computers
- Recall the machines I walked you through in the first lecture. Many of those had *hard-wired* programs
- Another way to put it is that they were not *programmable*
- This clearly raises issues. We'd like our machines to be truly general purpose, in that the operations they perform can change over time
- This led to the development of the *stored program computer* (or the von Neumann computer)
- In a von Neumann machine, programs (instructions for the computer) are stored in read/write memory along with data

Basic von Neumann architecture

- A von Neumann architecture consists of three main parts: memory that is read/write, Input/output devices (I/O devices), and a central processing unit (CPU) which has the job of actually executing instructions in the machine
- the CPU consists of two primary subunits: the control and the processing unit
- The simplest processing unit consists of an *arithmetic logic unit* (ALU), which performs calculations on data. Remember, we built a small ALU before!
- Which operations the ALU performs is determined by sequencing carried out by the control unit
- A processing unit will also contain a *register file*, which is just a clump of registers used for temporary storage which the ALU can operate on
- The ALU can get data from the control unit, and may send addresses to the control unit for branches or jumps (control flow)
- Memory you can think of as a storage array and two primary registers, the **MDR (memory data reg.) and the MAR (memory addr reg)**
- MAR is k-bits wide (address width) and MDR is >= j bits, where j is the addressability of the machine
- We also have I/O devices attached to the machine: e.g. keyboard, mouse, disk, video, printer, etc. They receive commands from the CPU. They can also send data and status back to the CPU. Device sends **interrupt signals** to the CPU to notify that it's done something