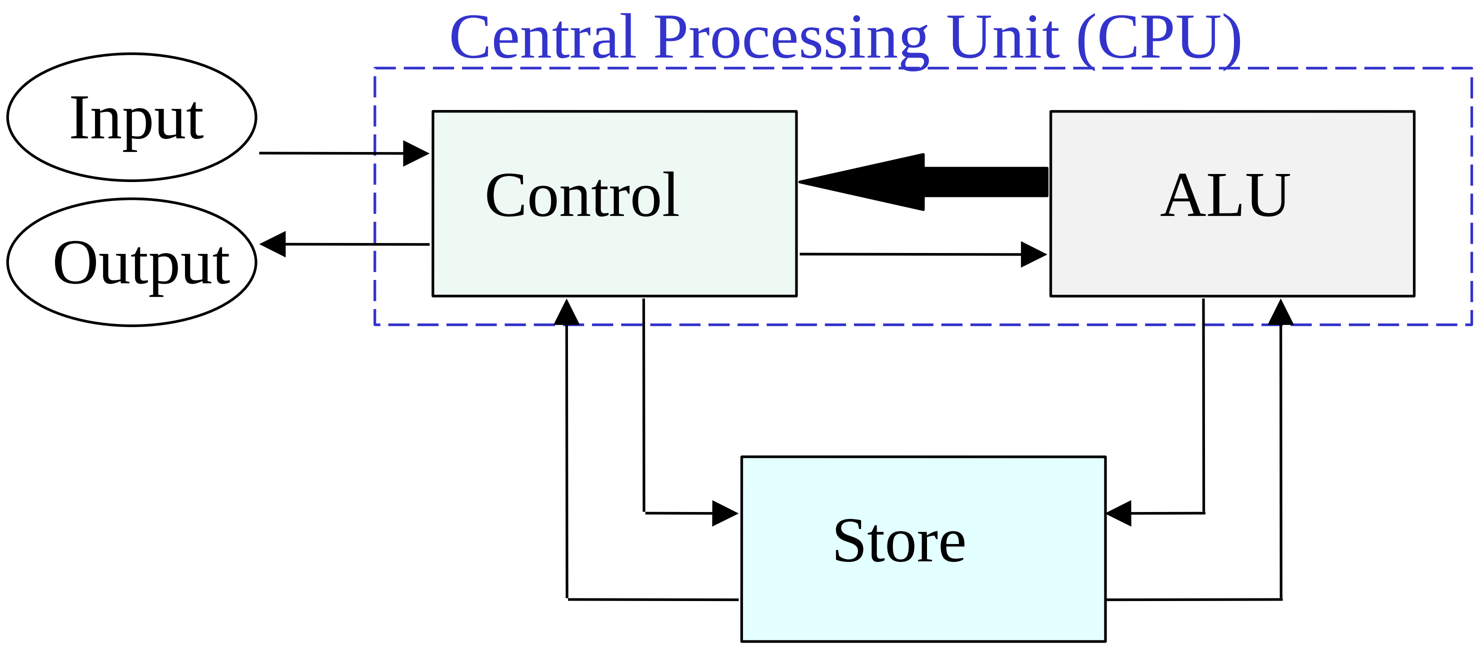
CM12002
Computer Systems Architectures

Fabio Nemetz

This lecture:

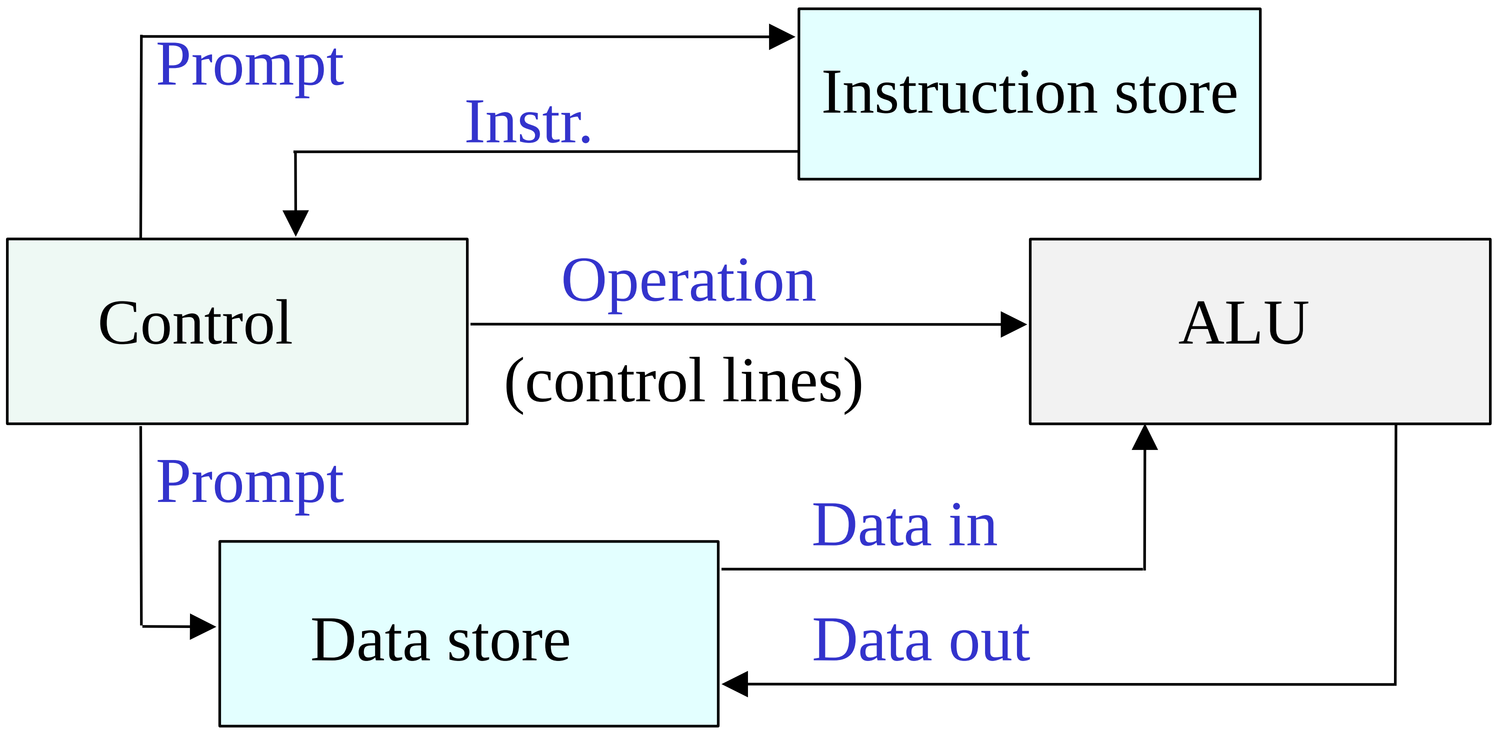
- The Harvard Architecture
- An example, using Arduino

The von Neumann Architecture



The Harvard Architecture

This maintains separate stores for data and instructions.
Thus the CPU can access instructions and data simultaneously.



The Harvard Architecture

Harvard architecture useful either:

- In special-purpose devices like microcontrollers* and signal processors (e.g., data from sensors) with instructions stored in ROM**.
- In sophisticated processors which can exploit the parallel fetching of code and data.

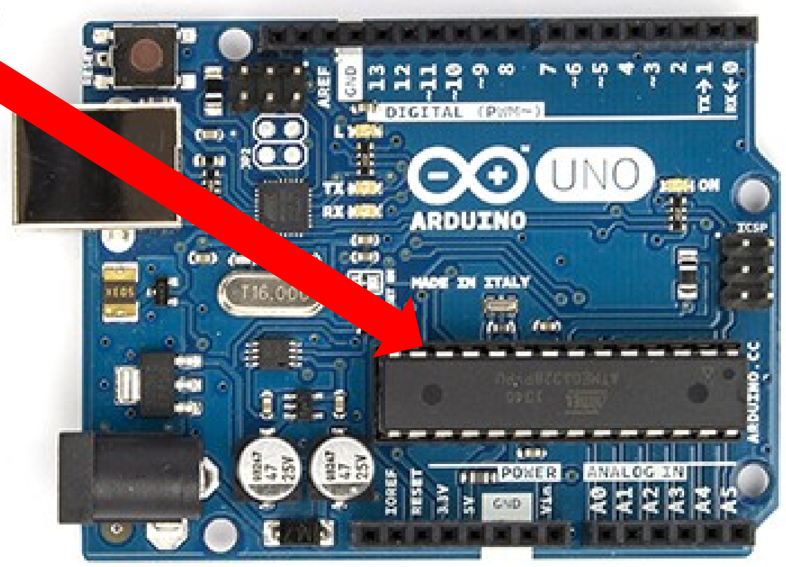
***Microcontroller**: small computer on a single integrated circuit containing a processor core, memory, and programmable input/output peripherals (basically **one device with ALU, control unit, I/O, memory**)

**ROM: Read Only Memory vs RAM: Random Access Memory

Arduino UNO

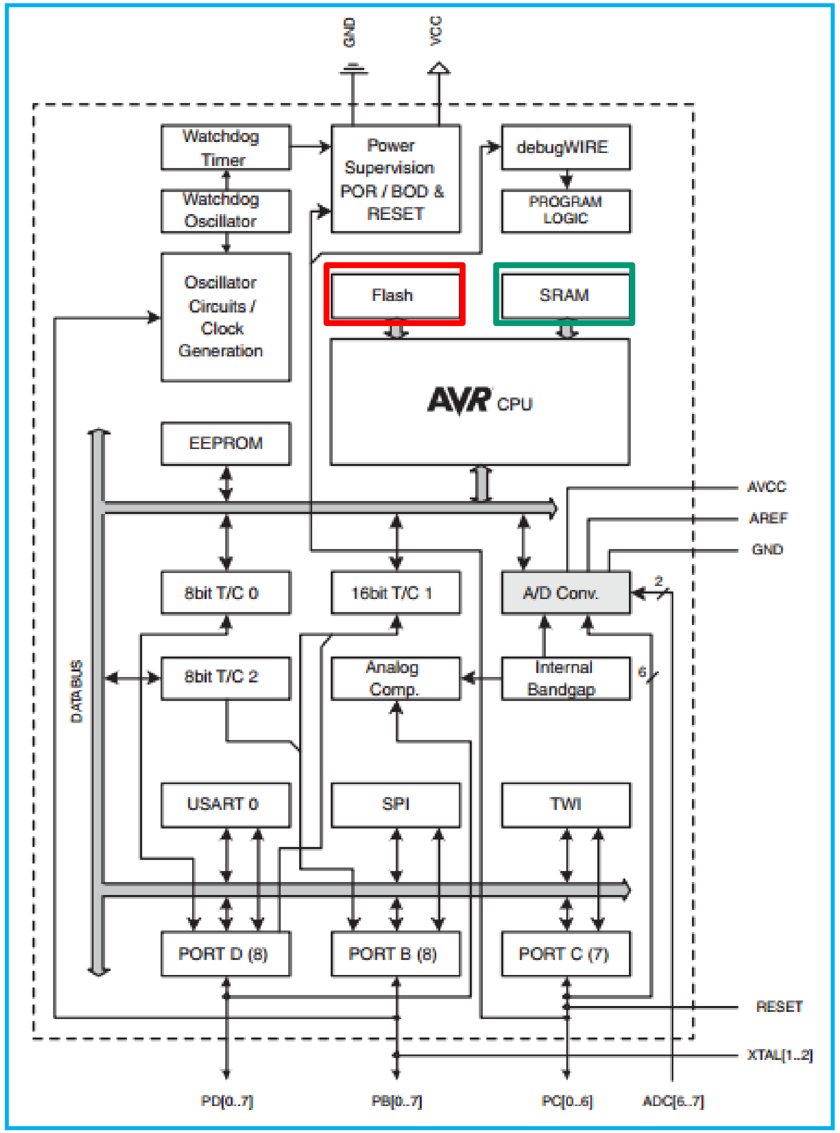
- Microcontroller: ATmega 328

- Microcontroller ATmega328 is the one used by the Arduino UNO board



Arduino UNO

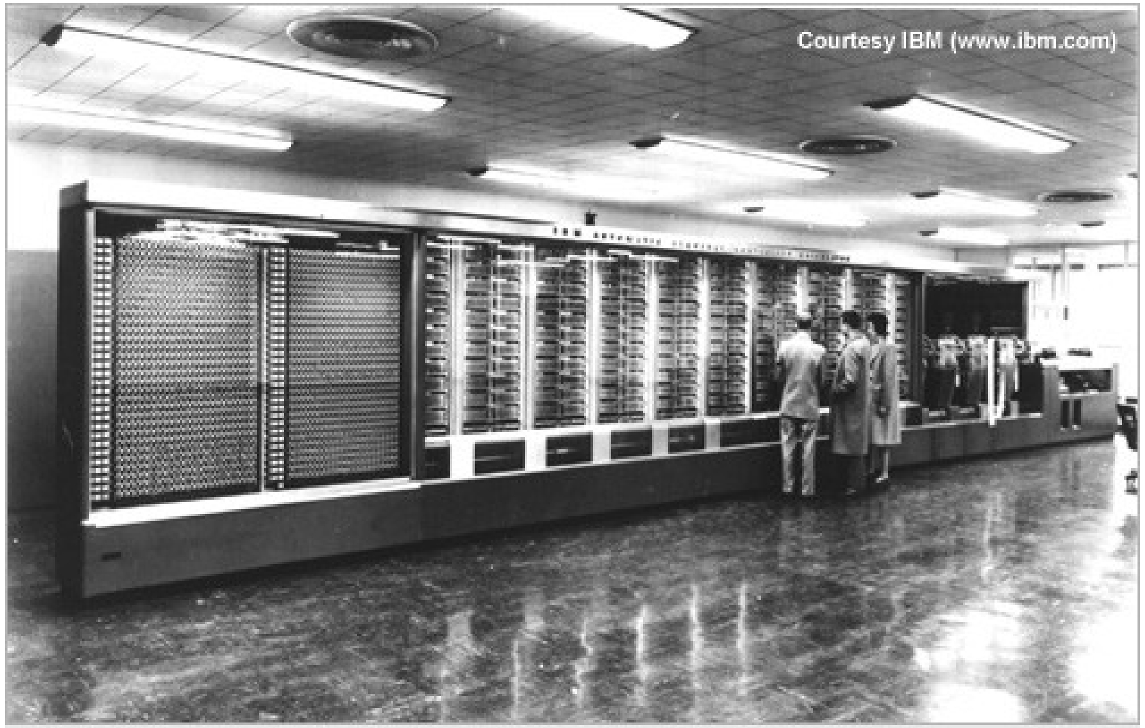
- Microcontroller: ATmega 328
- Microcontrollers are designed for embedded applications.
- An embedded processor typically has a well-defined task that it must perform reliably and efficiently, and at minimal cost.
- **Harvard Architecture**: good match for embedded applications
- Instructions: Flash memory (32Kb*)
- Data: SRAM** (2Kb)



*1Kb = 1024 bytes (1 byte = 8 bits)
This will be covered later during the unit

***SRAM= static RAM (no-need to refresh like DRAM)*

The Harvard Mark I (Harvard Architecture)



The Harvard Mark I

1944

Break

Question 1

- What is a software bug?

Answer

- A software bug is an error, flaw, failure, or fault in a computer program or system that causes it to produce an incorrect or unexpected result, or to behave in unintended ways.
- So “debug” means finding and fixing “bugs”

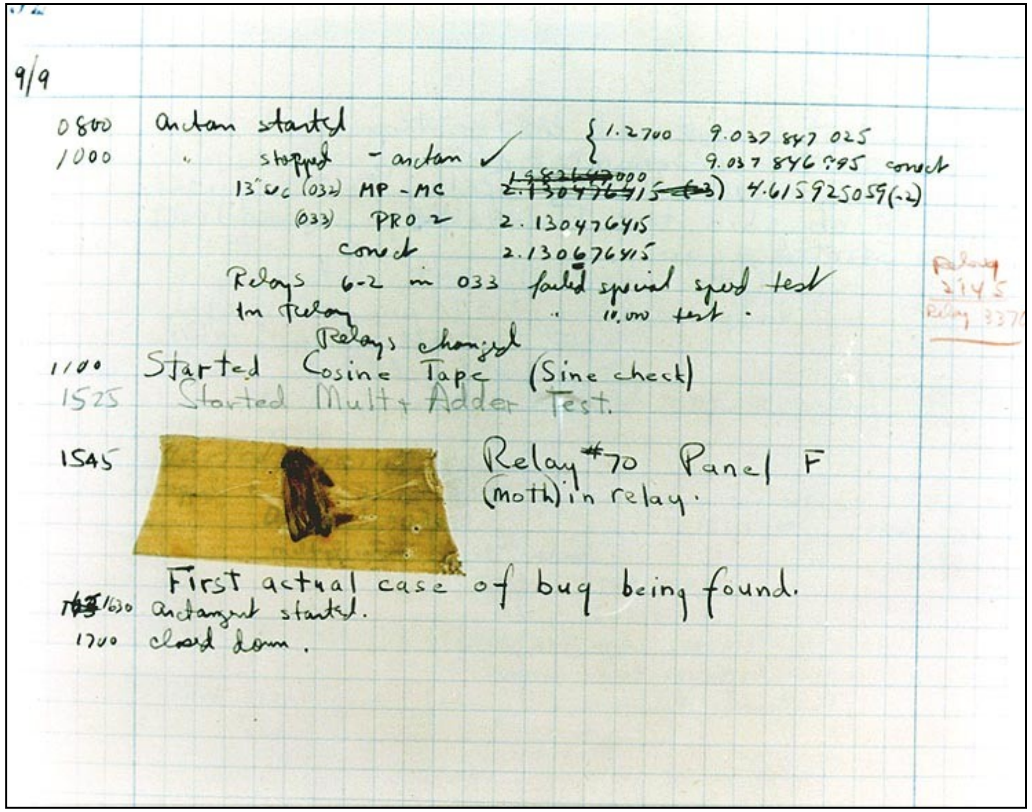
Question 2

- Why is it called “bug”???

The Harvard Mark II

The origin of the engineering term “bug” **in software**

1947: operators traced an error in the Mark II to a moth trapped in a relay



Grace Hopper

Mark II Log Book https://www.nationalgeographic.org/thisday/sep9/worlds-first-computer-bug/

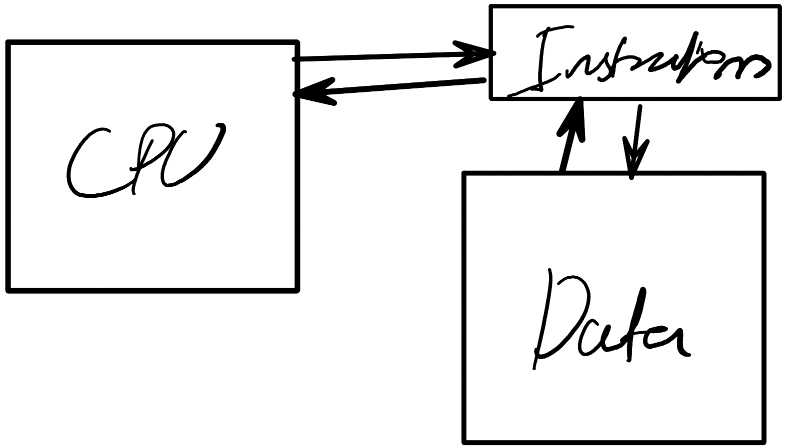
Modified Harvard Architectures

Modifications to the original Harvard Architecture:

- Provide a data pathway from the instruction store to the data store, so only one loading mechanism from store to CPU is required.

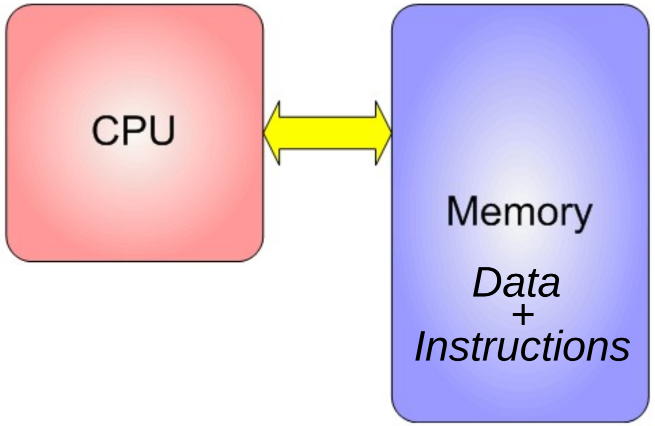
- Have separate caching* for code and data, but a single store, so that the processor can function in either Harvard or Von Neumann mode.

*cache: memory for code and data directly accessible to CPU

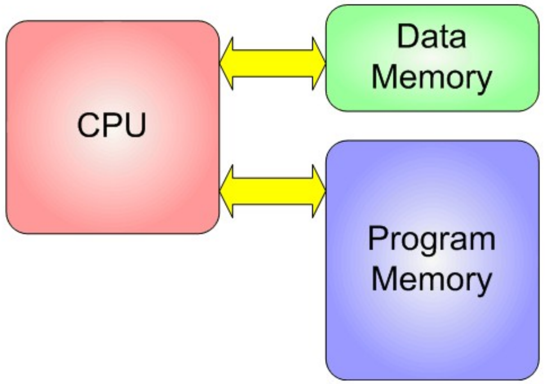


Von Neumann v Harvard

Von Neumann Architecture



Harvard Architecture



flexibility

performance

Store = memory

Back to the von Neumann Architecture

The components of the von Neumann architecture

- An arithmetic and logic unit (ALU) -- for performing actual computations
- A data store, or memory.
- A control unit --- for executing programs (retrieving data from memory and I/O, sending instructions to the ALU, etc.)
- Input (keyboard, external storage, network connection, etc.)
- Output (monitor, external storage, network connection, etc.)
- *Connections* between these components (buses).

The von Neumann architecture and this course

In this course, we will study the operation of each component of the von Neumann architecture in more detail:

- ALU: circuits for addition and logical operations
- Control unit: instruction coding and execution, basic control operations (branching, subroutines), circuits for control (latches)
- Store: digital representation of data, latches for storage
- I/O: different methods for controlling I/O (polling, interrupts, direct memory access)

Von Neumann machine characteristics

The characteristics of the machine we have constructed are:

- Discrete (i.e. ‘digital’) data;
- Storage of data in exact form, for a possibly indefinite time;
- Instructions and data stored in same form (making assemblers and compilers possible) and in the same place (compare with **Harvard** architecture).
- Processing capabilities, e.g. arithmetic, in ALU;
- Control to enable automation (selection of data and operations, their execution and sequencing); and
- Communications with the outside world via I/O devices.

Questions/reflection/research:

- Describe the von Neumann Architecture
- What are the main drawbacks of the von Neumann architecture?
- How can they be overcome?

Next time:

Parallel Architectures.