What is a Computer?

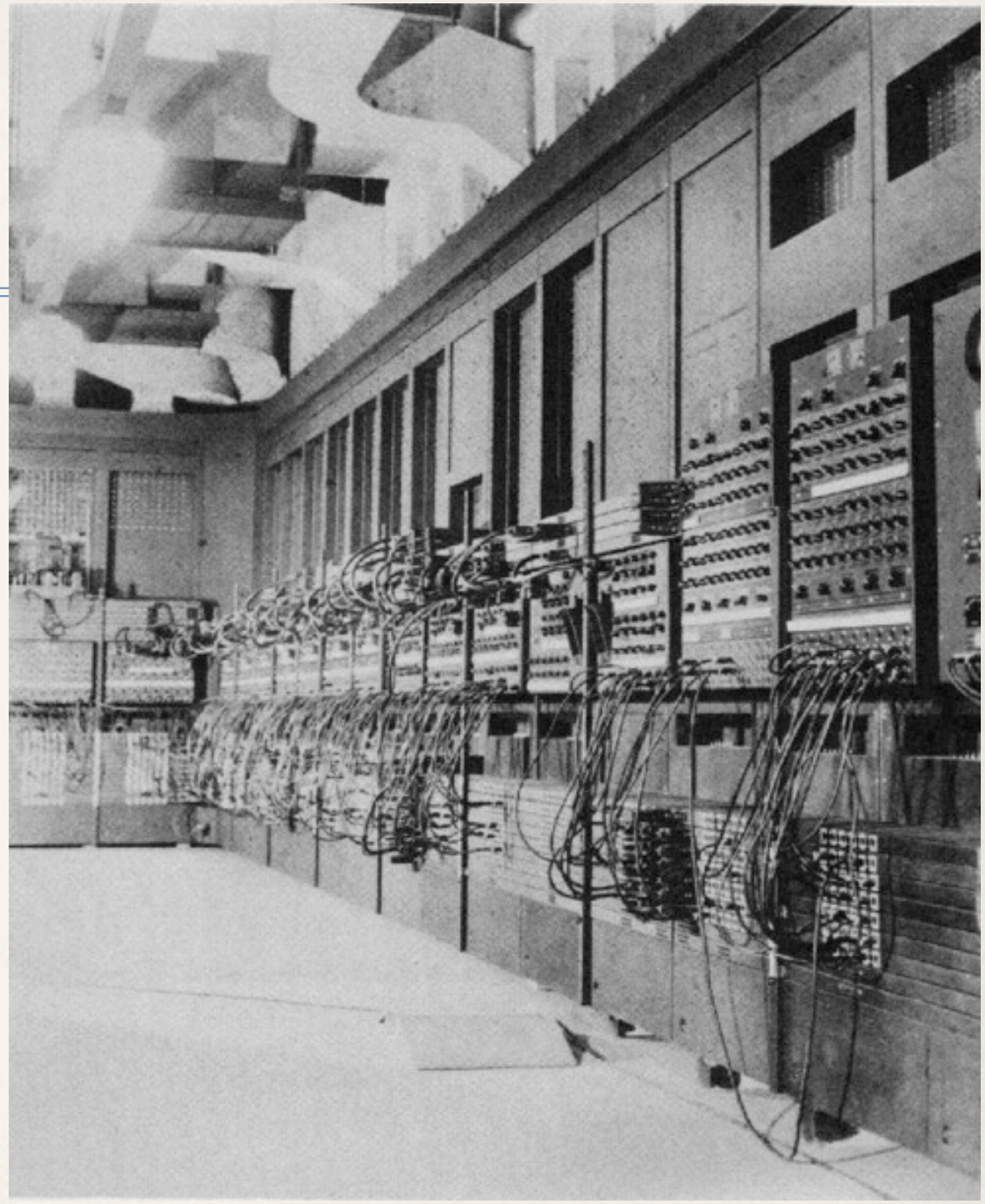
- According to Webster's Dictionary:
  - a computer is "an electronic machine which, by means of stored instructions and information, performs rapid, often complex calculations or compiles, correlates, and selects data."
- Essentially a computer is a machine that manipulates numbers and characters with high speed and precision.

The Development of Computers

- 1000 B.C. - Abacus
- Early 1940's:
  - Harvard Mark 1 (electro-mechanical)
  - Atanasoff-Berry-Computer (ABC) machine
  - Electronic Numeric Integrator and Calculator (ENIAC)

The ENIAC

- One of the first electronic digital computers developed by the military to compute ballistic firing tables in World War II.
- Over thirty tons
- 19,000 vacuum tubes, 1,500 relays, and hundreds of thousands of resistors, capacitors, and inductors
- Consumed almost 200 kilowatts

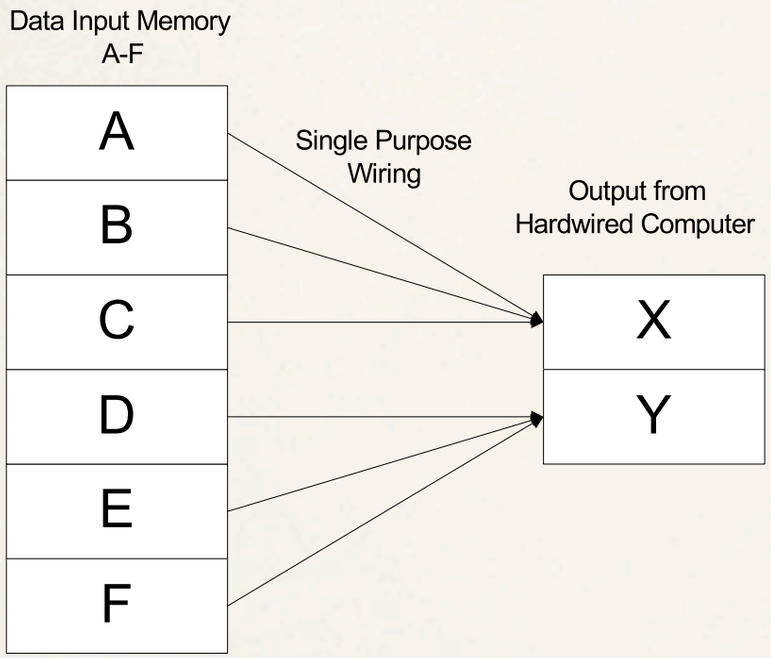


Compare to today: iPhone

- Multipurpose cell phone / personal digital assistant. Odds are... there's an app for it.
- 4.8 ounces
- 512MB RAM, 16-32GB SSD
- 1GHz CPU
- ~40 million transistors

Hardwired vs. “Softwired”

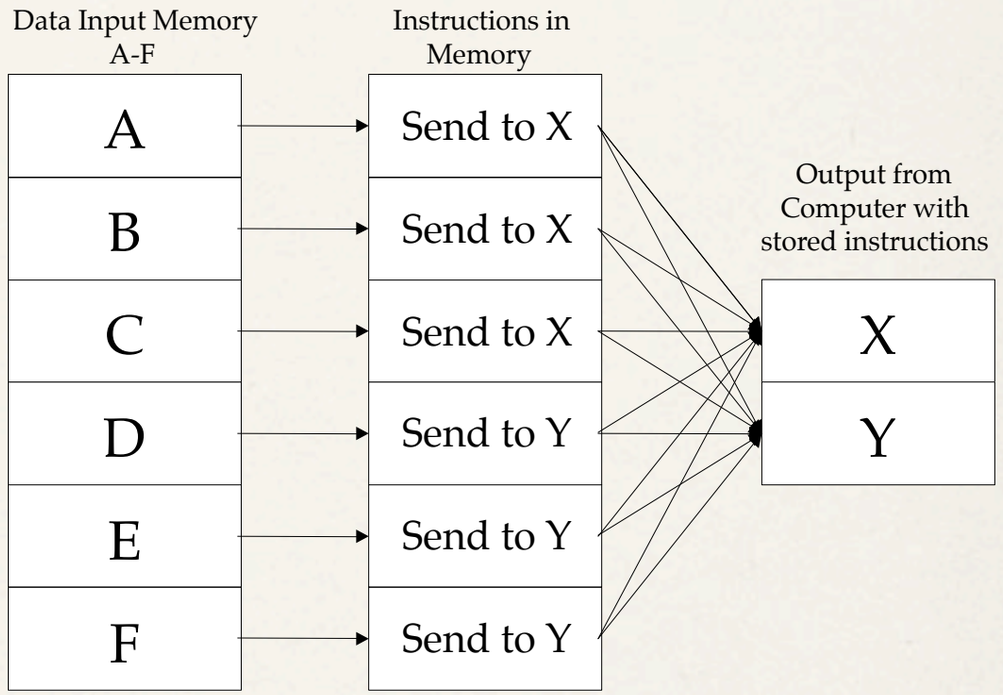
Hardwired



$$X = A + B + C$$

$$Y = D + E + F$$

“Softwired”



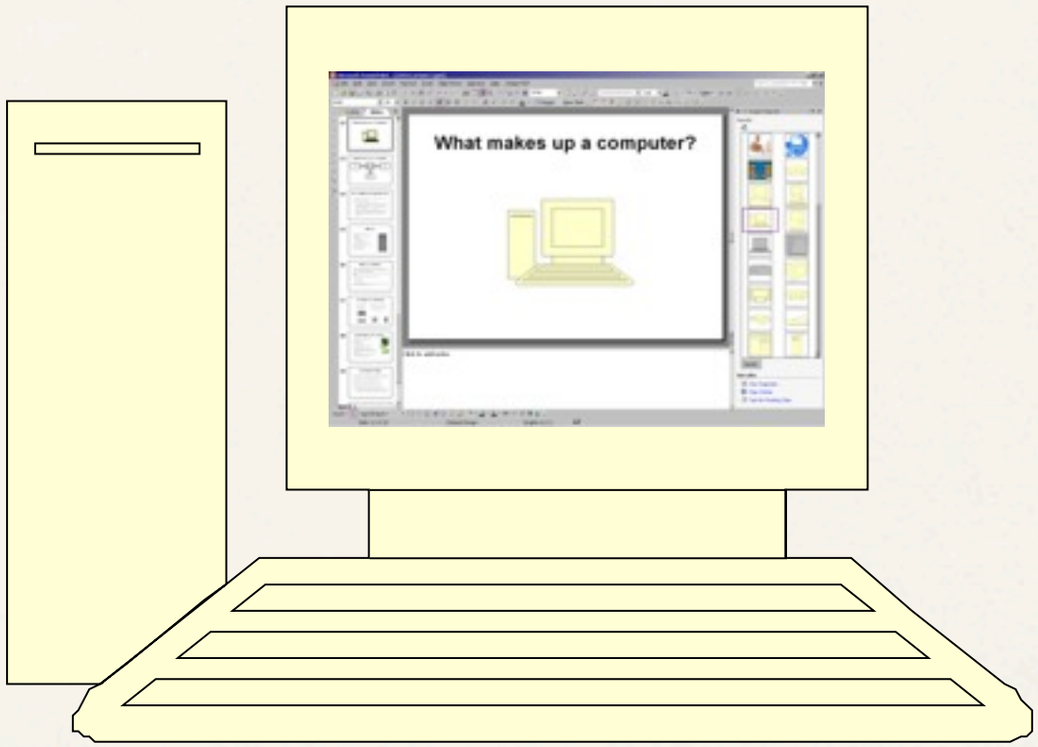
$$X = A + B + C$$

$$Y = D + E + F$$

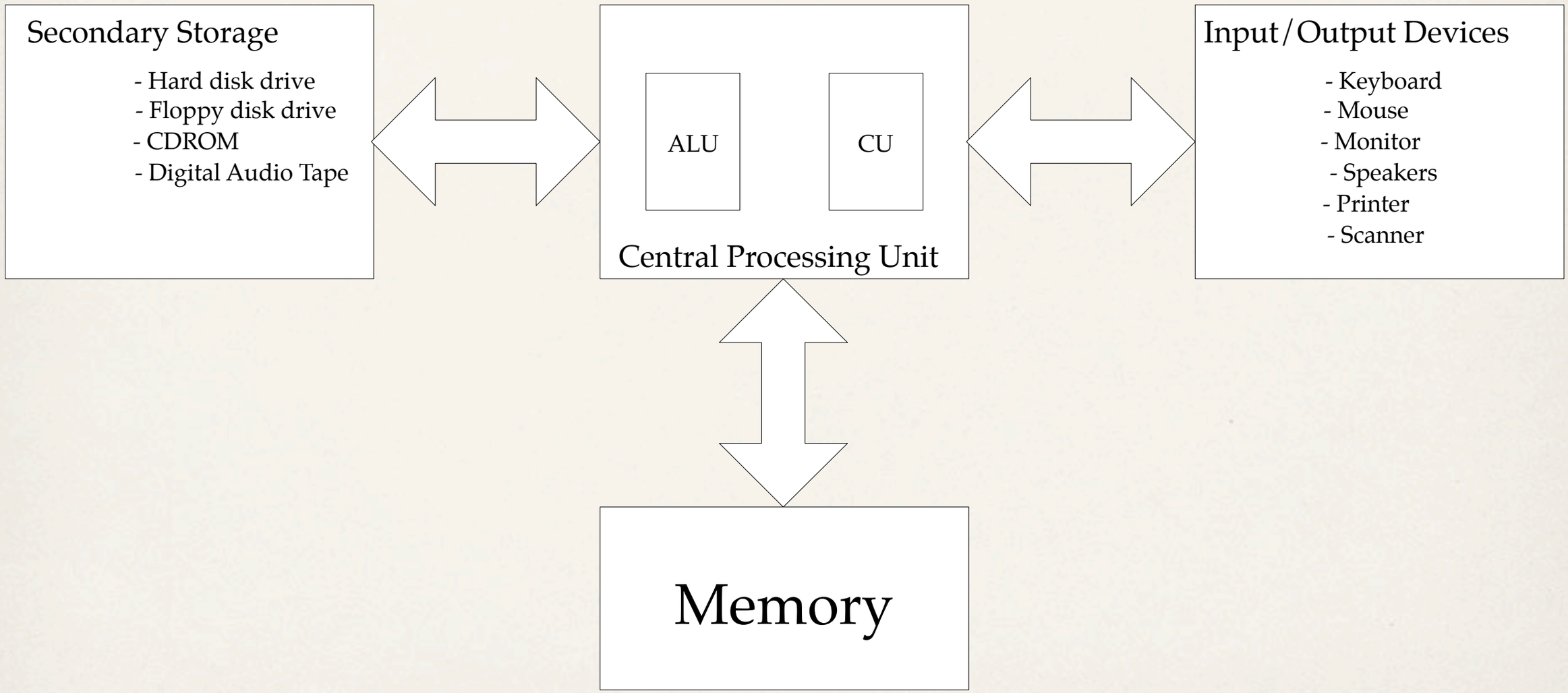
Timeline of Computers

Electronic Computers	Batch Processing	Time-Sharing Systems	Personal Computers	Powerful Portable Computers
1945	1950	1965	1975	1981 present
Vacuum Tubes Machine Language Programming ENIAC	Transistors Magnetic Core Memory Assemblers Compilers UNIVAC 1 IBM 704	Integrated Circuit Technology Operating System Software Teleprocessing	Supercomputers Microcomputers Workstations IBM PC (desktop)	Laptops Palmtops Personal Digital Assistants Wearable Computing

What makes up a computer?



What makes up a computer?



The Central Processing Unit (CPU)

- The component that transforms data from one form to another
- It is composed of two sections:
  - the arithmetic/logic unit (ALU) which performs arithmetic operations (e.g. +,-) and logical comparisons (e.g. >, <, =), and
  - the control unit (CU) which decodes machine instructions and sends signals to other components telling them what operations to perform.

Memory

- Computer memory is comparable to a collection of mailboxes.
- Each memory cell has two components:
  - its address, and
  - its contents, or value.
- A memory cell has one value and is never empty.
- Also called Random-Access-Memory (RAM)
- Volatile memory

0	LDA 14
1	ADD 15
2	STA 14
3	HLT
4	153
5	'A'
6	6.5
7	0.00014
8	gray pixel
9	'!'
10	0
11	0
12	0
13	0
14	10
15	7
…	

Memory continued

- The purpose of memory is to store information that is to be processed or store instructions on how to process information.
- Information is stored in memory in bits, or binary digits.
- Each bit can hold one of two states:
  - On or Off,
  - One or Zero,
  - True or False.
- A grouping of bits are called a “word”.
- Typically 8-bits are called 1-Byte.

How a Basic Computer Works

- Needs a set of basic instructions

  for example: Start, Stop, Load, Save, Add, Branch, etc.

- Instructions must be given in a form the computer understands:
  - machine language

How a Basic Computer Works (cont'd)

- For each basic command, an operation code is required.

Commands	Operation Code
Start	0000
Stop	0001
Add	0010
Load	0011
Save	0100
Branch if equal	0101
Brach if not equal	0110

How a Basic Computer Works (cont'd)

- For each basic command, an operation code is required.

Commands	Operation Code
Start	0000
Stop	0001
Add	0010
Load	0011
Save	0100
Branch if equal	0101
Brach if not equal	0110

Notice the order?
These operation codes are sequentially counting in binary:
0 - ?

How a Basic Computer Works (cont'd)

- Using our operation code, if we wanted to write a program using our Basic Computer it would resemble:

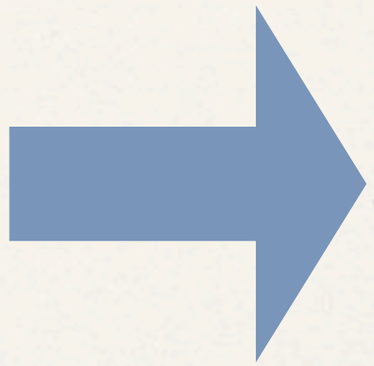
```
0000
0011 1110
0010 1111
0100 1110
0001
```

Assumes two numbers (data) stored at locations 14 (1110) and 15 (1111).

How a Basic Computer Works (cont'd)

- Using our operation code, if we wanted to write a program using our Basic Computer it would resemble:

```
0000
0011 1110
0010 1111
0100 1110
0001
```

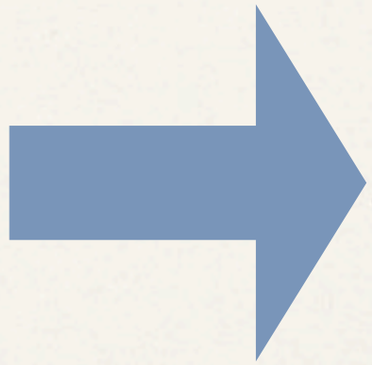


Assumes two numbers (data) stored at locations 14 (1110) and 15 (1111).

How a Basic Computer Works (cont'd)

- Using our operation code, if we wanted to write a program using our Basic Computer it would resemble:

```
0000
0011 1110
0010 1111
0100 1110
0001
```



```
Start
Load 14
Add  15
Save 14
Stop
```

Assumes two numbers (data) stored at locations 14 (1110) and 15 (1111).

How a Basic Computer Works (cont'd)

- The usage of operation code (op-codes) is not human friendly, but this is the language the computer understands (machine language).
- To make machine language more programmer friendly, short words (mnemonics) were used to represent the operation code.

How a Basic Computer Works (cont'd)

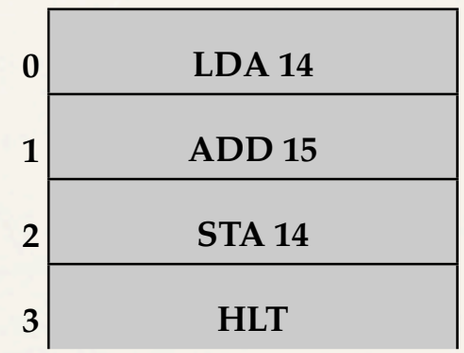
- For each basic command, an operation code is required and a mnemonic

Commands	Operation Code	Mnemonic
Start	0000	ORG
Stop	0001	HLT
Add	0010	ADD
Load	0011	LDA
Save	0100	STA
Branch if equal	0101	BEQ
Brach if not equal	0110	BNE

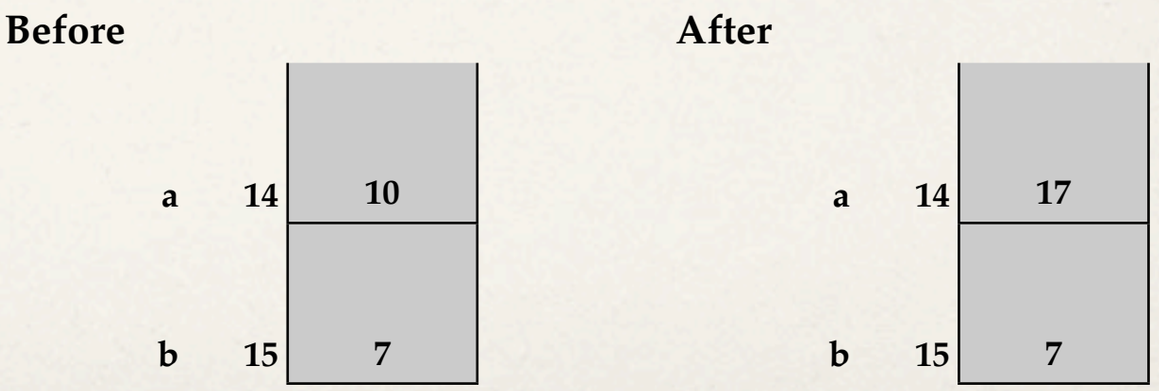
CPU-Memory Interaction

A computer transforms information from one form to another by a list of instructions called a program.

The program shown in memory is loading the value stored in address 14. Then adding it to the value stored in address 15. The result is then stored into address 14.



a = a + b;



Input/Output (I/O) Devices

- Allows interaction with the computer
- Input devices:
  - Keyboard, mouse, gamepad
- Output devices:
  - Monitor, printer, speakers
- Until recently most I/O with computers was done completely with text.
- The Graphical User Interface (GUI) is common now.

Secondary Storage

- Secondary storage is required to provide a permanent copy of information incase of a power loss, or for later retrieval.
- Also, systems typically store more information than will fit in memory.
- Typical secondary storage includes:
  - Hard disk drives, floppy disk drives, CDROMs, DVDROMs, USB keys, and magnetic tapes.

The Human-Computer Analogy

The Human-Computer Analogy

- "The Brain" → CPU

The Human-Computer Analogy

- "The Brain" → CPU
- "The Senses" → I/O Devices

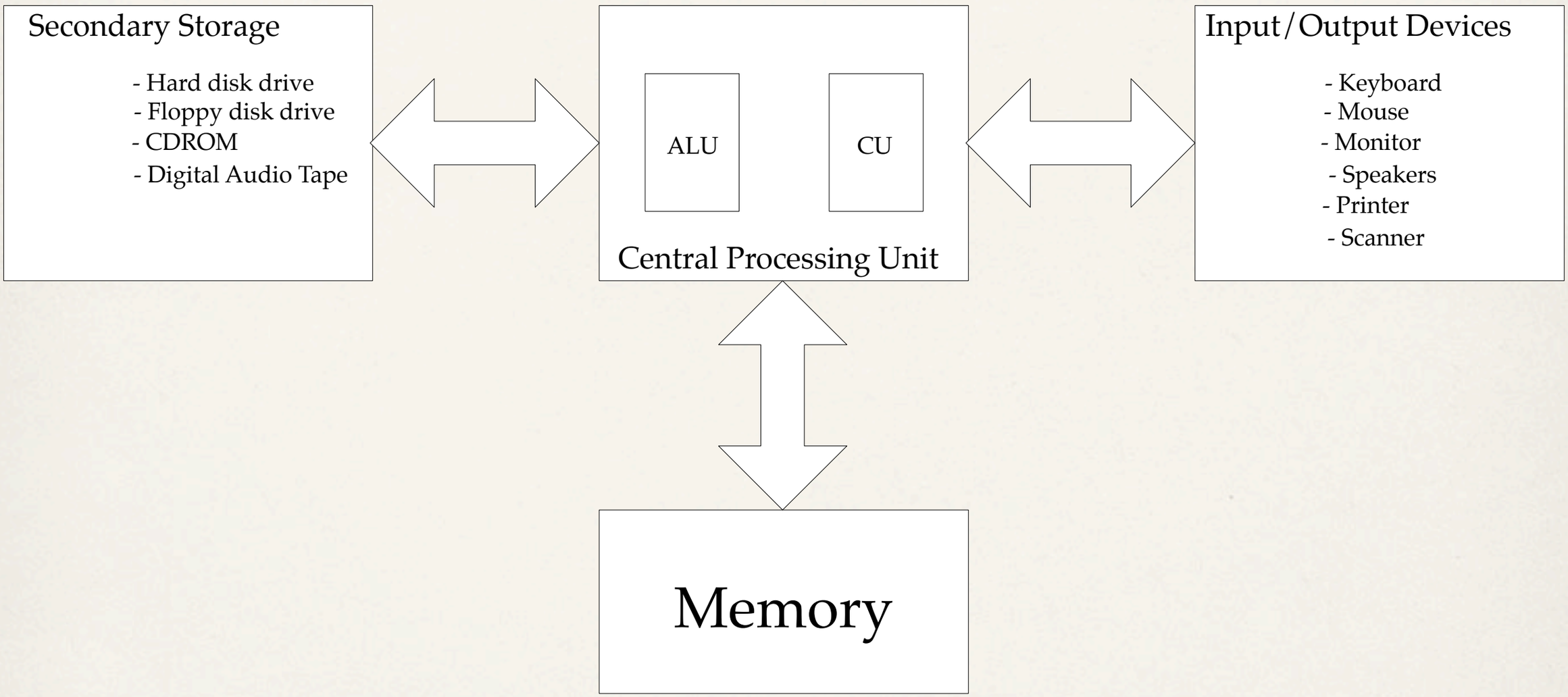
The Human-Computer Analogy

- "The Brain" → CPU
- "The Senses" → I/O Devices
- "Short Term Memory" → Memory

The Human-Computer Analogy

- "The Brain" → CPU
- "The Senses" → I/O Devices
- "Short Term Memory" → Memory
- "Belly" → Secondary Storage

Recall: What makes up a computer?



Software

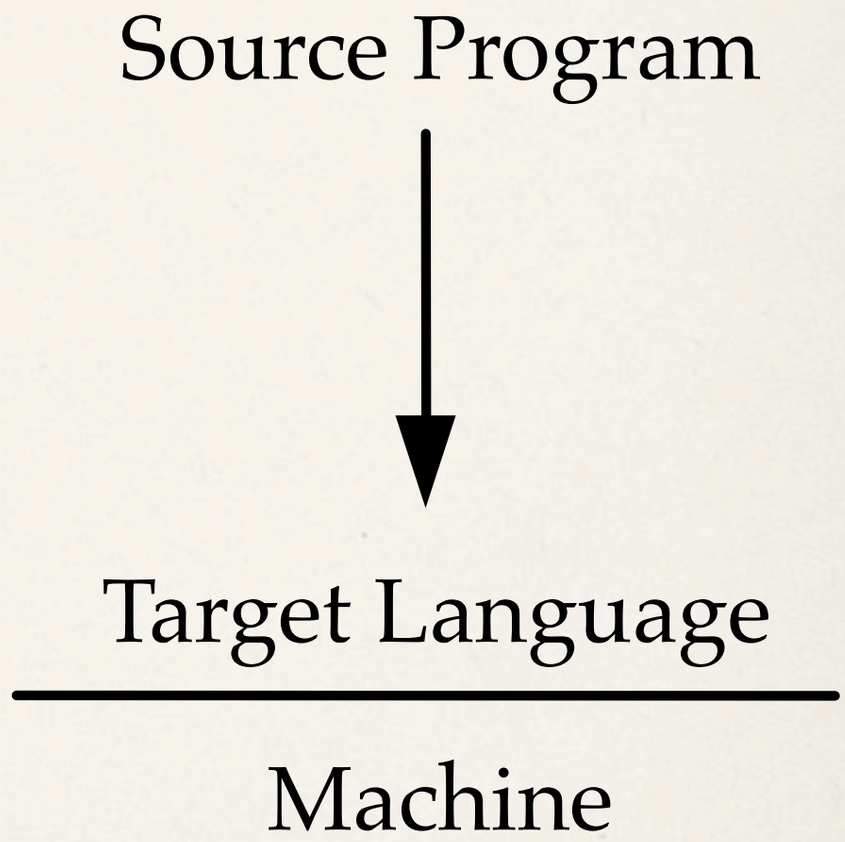
- The list of instructions required to operate the computer.
- Operating System (OS): a collection of programs that control the interaction of the user and the computer hardware.
- Application: a program which assists in accomplishing a specific task, such as word processing.

Programming

- Assembling the list of instructions for the computer execute.
- The developer very rarely writes in the language directly understood by the computer (machine language).
- Normally, the source program is translated from a high-level language (such as C) for the computer to execute.

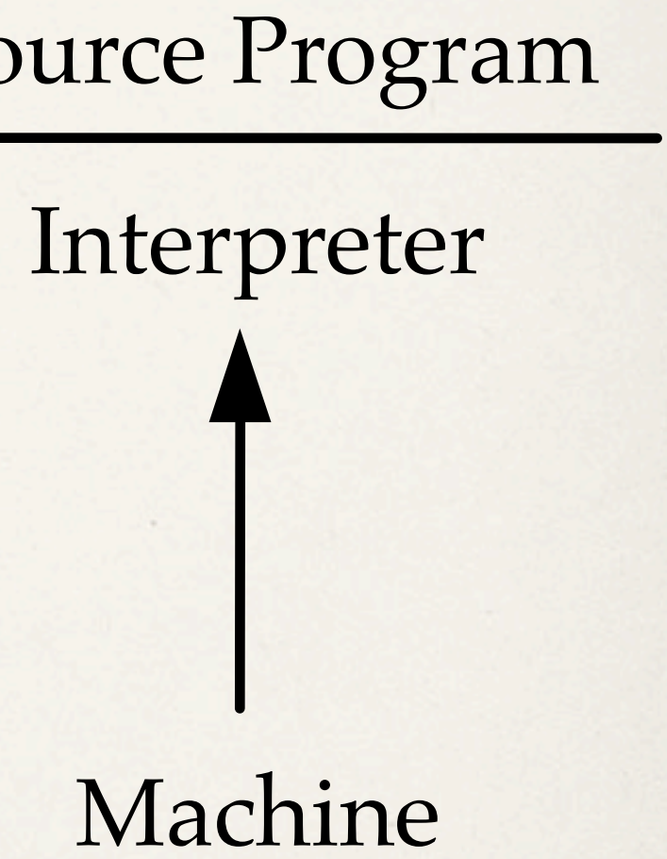
Program Translation: the Compiler

- Compiler
  - language is brought down to level of machine
  - translates *source* program into *target* language (machine language)
  - translates once and keeps target language copy



Program Translation: the Interpreter

- Interpreter
  - machine is brought up to level of the language
  - behaves as though machine runs program source natively
  - executes source program line by line every time program is run



Compiler vs. Interpreter

- Compilation can be more efficient because it stores a machine language version of the source program. An interpreter must translate the program each time it is run.
- Interpretation can be more flexible as changes or corrections to source programs can be implemented on the fly. A compiler must re-translate the entire source program each time a change is made.