

Computer Basics



Electronic computers have changed dramatically over their 50 history, but a few basic principles characterize all computers

© University of Washington, 2001



A Computer Is ...

- ❖ Computers process information by deterministically following instructions, called *executing* instructions
 - ❖ Unlike humans, computers follow instructions *exactly*
 - ❑ Computers have no imagination or creativity
 - ❑ Computers have no intuition
 - ❑ Computers are literal, with no sense of irony, subtlety, proportion, ...
 - ❑ Computers don't joke, they're not vindictive or cruel
 - ❑ Computers are not purposeful
- ... computers only execute instructions

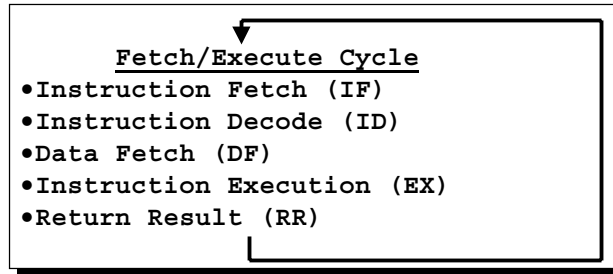
If a computer has any useful characteristics, it's because someone has programmed it -- given it the instructions -- to behave usefully

© University of Washington, 2001

**FIT
100**

Interpreting Instructions

- ❖ To perform instructions, a computer's hardware implement a process called the *fetch/execute cycle*



- ❖ The F/E Cycle is an unending process

© University of Washington, 2001

**FIT
100**

An Analogy ...

- ❖ At the Nenana Ice Classic people pay \$2 to guess when the ice will “break up” on the Tananah River

The person who processes entries at the Ice Classic headquarters works just like the Fetch/Execute Cycle

Sello, Nenana!

*I'm sure breakup will be on
*** May 4 at 12:04PM ***
Find \$2 enclosed.*

*Yours, Frost 7. Snowman
61 River St.
Circle AX*

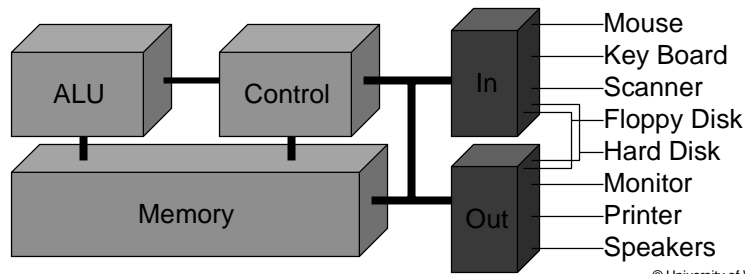
Get next entry (IF)
Find date/time (ID)
Get card's date/time (DC)
Enter Name (EX)
Return Card (RR)

© University of Washington, 2001

**FIT
100**

Anatomy Of A Computer

- ❖ A computer is composed of five components ...
 - ❑ Arithmetic/Logic Unit (ALU) -- the part that “computes”, e.g. +
 - ❑ Control -- the part that follows the Fetch/Execute Cycle of the program and tells the ALU what to compute
 - ❑ Memory -- where data, programs are kept while computing
 - ❑ Input -- ports to peripheral devices from which data comes
 - ❑ Output -- ports to peripheral devices to which data goes



© University of Washington, 2001

**FIT
100**

Memory

- ❖ The memory is passive, storing programs and data

address:	0	1	2	3	4	5	6	7
value:	R	2	D	2	100	*	R	W

← byte →

- ❖ Memory is like a series of “byte-size” boxes -- each has an address and some contents called its value
- ❖ Memory is called RAM for “random access memory” because the control can access any random location in the memory

© University of Washington, 2001

**FIT
100**

Control Rules!

- ❖ The control follows through the instructions, executing them by telling other parts what to do
- ❖ The instructions come from the program stored in the memory

The instructions are expressed in a machine language that the control can understand. A typical machine instruction is

add 884, 1004, 6618

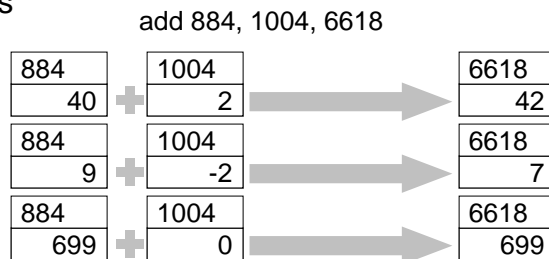
which means “add the number in memory location 884 to the number in memory location 1004, and put the result in memory location 6618”

© University of Washington, 2001

**FIT
100**

Emphasis ...

- ❖ The instruction add 884, 1004, 6618
- ❖ Does not add 884, 1004 and 6618 together -- we can figure that out with a calculator ... it adds whatever numbers are stored in those memory locations
- ❖ Different numbers in those locations produce different results

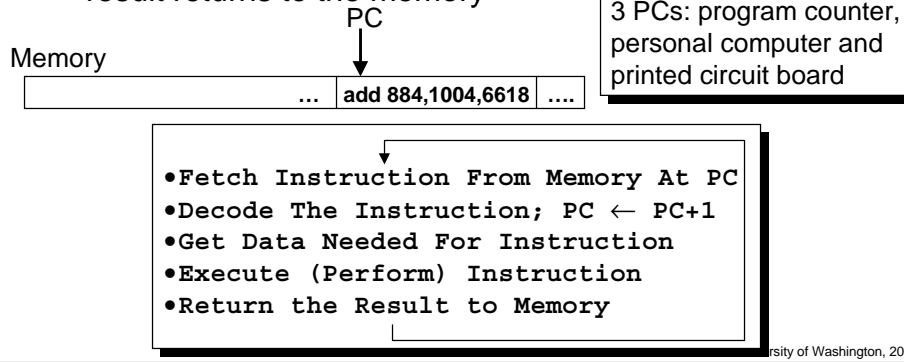


© University of Washington, 2001

**FIT
100**

Following Instructions

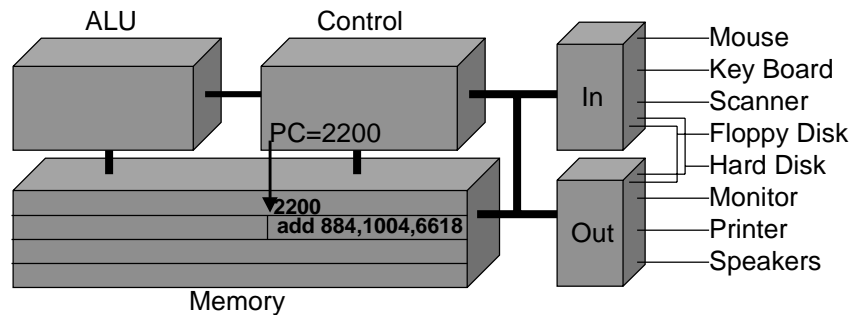
- ❖ The control keeps its place in the program using a program counter or PC ... a better name would be "instruction pointer"
- ❖ The control also prepares for data fetches from and result returns to the memory



**FIT
100**

The Fetch/Execute Process

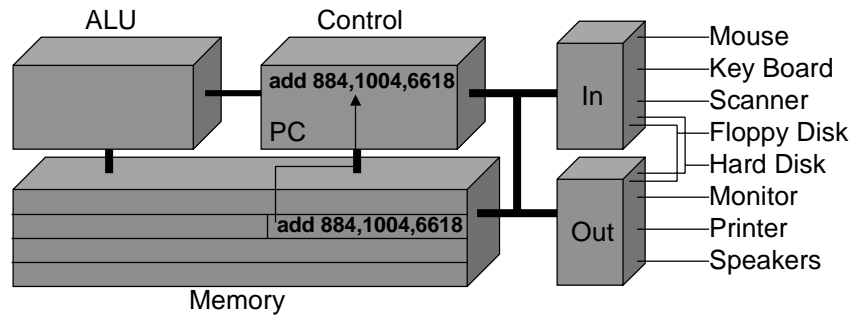
- ❖ Just before instruction fetch



**FIT
100**

Instruction Fetch

- ❖ Get instruction at the memory location PC

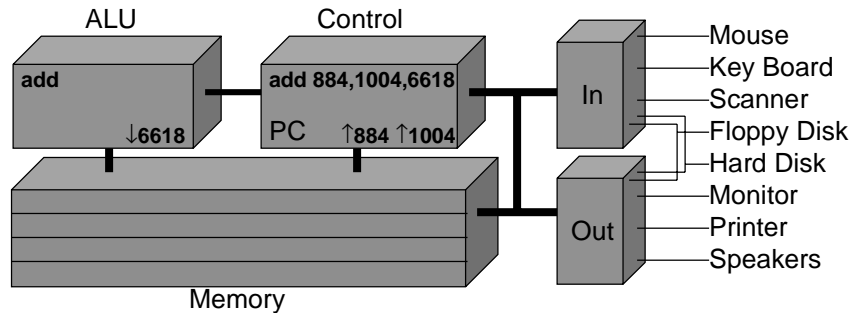


© University of Washington, 2001

**FIT
100**

Instruction Decode

- ❖ Analyze instruction and set up later steps
 - Specify ALU operation (add)
 - Specify addresses to fetch (884,1004) and to store (6618)

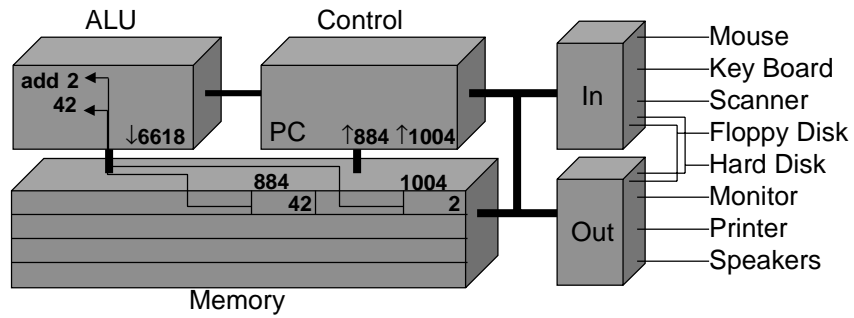


© University of Washington, 2001

**FIT
100**

Data Fetch

- ❖ Move values stored at fetch-addresses to ALU for processing

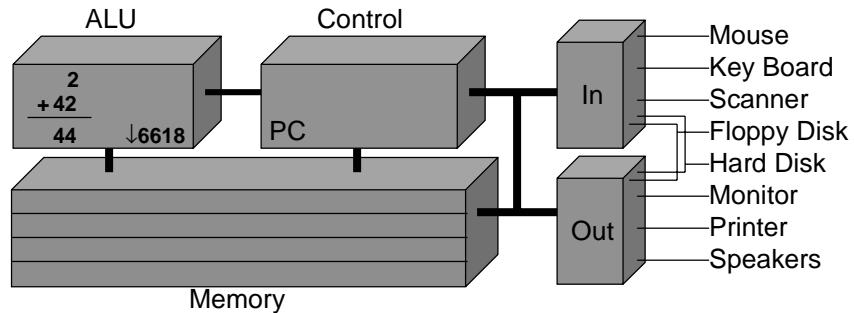


© University of Washington, 2001

**FIT
100**

Execute

- ❖ The operation of the instruction (add) is performed

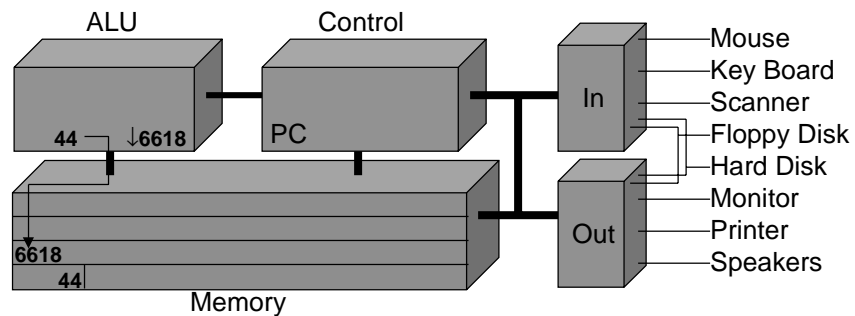


© University of Washington, 2001

**FIT
100**

Result Return

- ❖ The result is returned to memory to the address specified in the instruction



© University of Washington, 2001

**FIT
100**

The PC's PC

- ❖ After the instruction has been fetched and executed, the next instruction in sequence is fetched at PC+1
- ❖ This scheme *should* cause the computer to run through memory executing all instructions once and “fall off the end of memory”
- ❖ Computers have machine instructions to branch and jump, i.e. go to some instruction other than the next
- ❖ Jump and Branch change the PC after increment
- ❖ Programs generally repeat many instructions

© University of Washington, 2001

**FIT
100**

The Numbers, Please

- ❖ A memory location can store one byte of information (8 bits), enough for a keyboard character
- ❖ A “normal” whole number (integer) uses 4 bytes
- ❖ A machine instruction uses 4 bytes
- ❖ Units of memory size are ...
 - ❑ KB, kilobyte, 1024 bytes ... just over a thousand bytes, a “K”
 - ❑ MB, megabyte, 1,048,576 bytes ... just over a million bytes, a “meg”
 - ❑ GB, gigabyte, 1,073,741,824 bytes ... just over a billion bytes, a “gig”
 - ❑ TB, terabyte, 1,099,511,627,776 bytes ... just over a trillion bytes

© University of Washington, 2001

**FIT
100**

Strange Numbers

- ❖ Why do computers use such strange numbers???
 - ❑ These numbers are powers of 2
 - + $2^{10} = 1,024$ call it a thousand
 - + $2^{20} = 1,048,576$ call it a million
 - + $2^{30} = 1,073,741,824$ call it a billion
 - + $2^{40} = 1,099,511,627,776$ call it a trillion
- ❖ When you buy a megabyte of memory it's as if you get 48,576 bytes for free!

© University of Washington, 2001

**FIT
100**

The Pace Of Computing

- ❖ Computers use electronic clocks to pace the F/E cycle
- ❖ If the computer goes around the F/E cycle once per tick, then the rate of the clock (“ticks/second”) gives the number of instructions executed per second

- ❖ hertz measures “cycles per second”
- ❖ 500MHz, specifies “500 million cycles per second”

In actuality the “one instruction per clock cycle” rule is only an approximation ... modern computers are very much more complicated.

**FIT
100**

Summary

- ❖ Computers deterministically execute instructions to process information
- ❖ Computers have five parts: ALU, Control, Memory, Input and Output
- ❖ The control implements a process called the Fetch/Execute Cycle
- ❖ The fetch/execute cycle is a fundamental method of deterministically performing operations, and the idea is used many places in computation ...