UNIVERSITY OF BIRMINGHAM

Computer Systems
CPU, Memory and Program Execution

Lecture Objectives

To develop fundamental understanding of computer **architecture & organization, instruction sets** and **assembly language** programming.

Lecture Outline

- Anatomy of a Computer System
  - Memory, Control Unit, ALU, Input/Output
- How a program is executed
  - The Fetch/Execute Cycle
- Computer's View of Software

Before we start …
What is a computer?

A computer is a complex system (machine) that can be instructed to carry out sequences of arithmetic or logical operations automatically via computer programming…

So …
How do Computers Execute Programs?

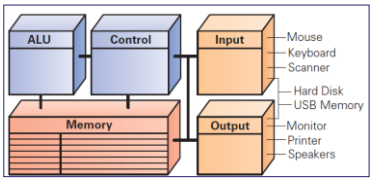
Requirements of a Computing Device

- Load the program from some external device/memory
- – interface to the outside world
- Process instructions in the correct order – mechanism to keep track of progress, local storage and decoding of instructions
- Access pieces of data in accordance with the program's instructions – local storage of data
- Perform computations – a calculation "engine"
- Take decisions according to the results of the computations – mechanism for control
- Send the results of the computations to some external device – interface to the outside world

Anatomy of a Computer System

All computers, regardless of their implementing technology, have five basic parts or subsystems:

1) Memory,
2) Control unit,
3) Arithmetic/logic unit (ALU),
4) Input unit, and
5) Output unit



1) Memory

Memory stores both the **program** while it is running and the **data** on which the program operates:
Properties of memory:

- **Discrete locations**
  - Memory is organized as a sequence of discrete locations
  - In modern memory, each location is composed of 1 byte (8 bits)
- **Addresses**
  - Every memory location has an address, whole numbers starting at 0
- **Values**
  - Memory locations record / store values

1) Memory

- **Finite Capacity**
  - Memory locations have a finite capacity (limited size),
  - Data may not "fit" in the memory location

0	1	2	3	4	5	6	7	8	9	10	11	
100	T	h	a	N	K	$	*	4	~~b~~	d	a	...

Figure 9.3 Diagram of computer memory illustrating its key properties.

Address of location is displayed above the box and the contents of location is shown in the box

Blocks of four/eight bytes are used as a unit so often that they are called **memory words**

Random Access Memory (RAM)

- Computer memory is called Random Access Memory (RAM)
  - "Random access" is out-of-date and simply means that the computer can refer to the memory locations in any order
- RAM is measured in megabytes (MB), gigabytes (GB), terabytes (TB), ...
- Lots of memory is needed to handle the space required of programs and data

2) Control Unit

- The control unit of a computer is where the Fetch/ Execute Cycle occurs
  - Its circuitry fetches an instruction from memory and performs the other operations of the Fetch/Execute Cycle on it
  - A typical machine instruction has the form **ADD 4000, 2000, 2080**
  - Looks like those three numbers should be added together
- What it really means is that whatever numbers are stored in memory locations 2000 and 2080 be added together, and the result be stored in location 4000

Illustration of a Single Instruction

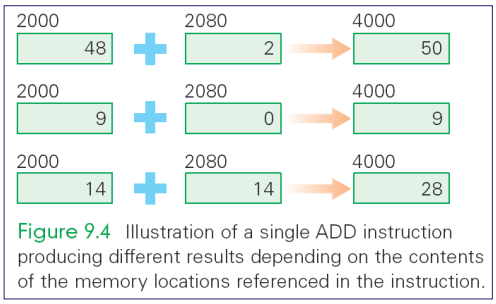


Figure 9.4 Illustration of a single ADD instruction producing different results depending on the contents of the memory locations referenced in the instruction.

3) Arithmetic/Logic Unit (ALU)

- "Does the math"
- A circuit in the ALU can add/subtract two numbers
- The circuit uses logic gates or simpler circuits that implement operations like AND and OR
- There are also circuits for multiplying, for comparing two numbers, etc.
- The ALU carries out each machine instruction with a separate circuit

4 & 5) Input / Output Units

- These two components are the wires and circuits through which information moves into and out of a computer
- A computer without input or output is useless (so to speak, as you cannot get information in/out of it)

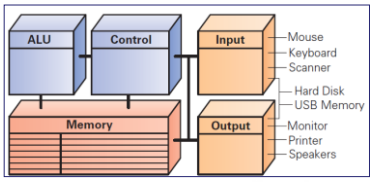
The Peripherals

- Peripherals connect to the computer input/output (I/O) ports
- They provide input or receiving its output
- They are not considered part of the computer:
  - They are only specialized gadgets that encode or decode information between the computer and the physical world.
  - Some examples are Keyboard, Mouse, Monitor, USB, Hard Disks, Network Cards etc.

Anatomy of a Computer System – Recap!

All computers, regardless of their implementing technology, have five basic parts or subsystems:

1) Memory,
2) Control unit,
3) Arithmetic/logic unit (ALU),
4) Input unit, and
5) Output unit



How Computers Execute Programs?

Here is a simple computer program:

```
a = int(input("Enter a number:"))
b = 1
for i in range (1, a+1):
      b = b*i

print (a,"! = ",b)

What is the program computing ?
```

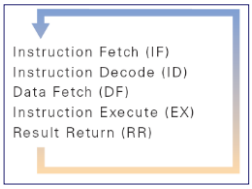
Online Python compiler - https://www.online-python.com/

Executing Programmes

A program is **loaded** into memory and address of first instruction is placed in PC (Program Counter)

The following Fetch / Execute cycle is then followed:

1) Instruction Fetch (IF)
2) Instruction Decode (ID)
3) Data / Operand Fetch (DF/OF)
4) Instruction Execution (EX)
5) Result Return / Store (RR/ST)



ADD 800, 428, 884

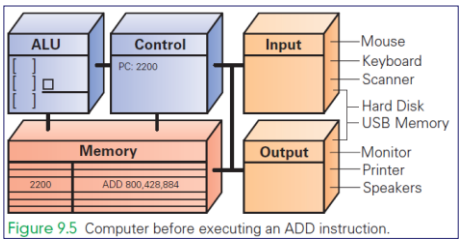


Figure 9.5 Computer before executing an ADD instruction.

ADD the values found in memory locations 428 and 884 and store the result in location 800

Instruction Fetch (IF)

- Execution begins by moving the instruction at the address given by the PC (2200) from memory to the control unit
- Bits of instruction are placed into the decoder circuit of the Control Unit
- Once instruction is fetched, the PC can be readied for fetching the next instruction

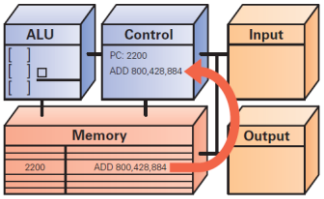


Figure 9.6 Instruction Fetch: Move instruction from memory to the control unit.

Program Counter

Instruction Decode (ID)

- Decoder determines what operation the ALU will perform (ADD), and sets up the ALU
- Decoder finds the memory address of the instruction's data (source operands)
- Decoder finds the destination address for the Result Return step and places the address in the RR circuit

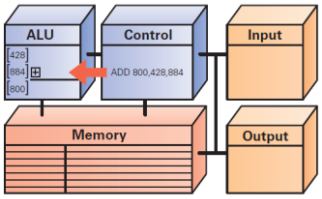


Figure 9.7 Instruction Decode: Pull apart the instruction, set up the operation in the ALU, and compute the source and destination operand addresses.

Instruction Decode (ID)

- Decoder finds the memory address of the instruction's data (source operands)
  - Most instructions operate on two data values stored in memory (like ADD), so most instructions have addresses for two source operands
  - These addresses are passed to the circuit that fetches them from memory during the next step
- Decoder finds the destination address for the Result Return step and places the address in the RR circuit

Data Fetch (DF)

- The data values to be operated on are retrieved from memory
- Bits at specified memory locations are copied into locations in the ALU circuitry
- Data values remain in memory (they are not destroyed)

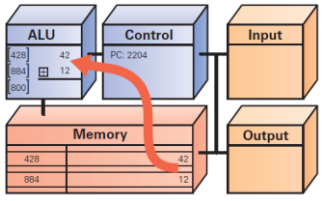


Figure 9.8 Data Fetch: Move the operands from memory to the ALU.

Instruction Execution (EX)

At this stage, the answer is still stored in CPU and has not been written into Memory

- For the ADD instruction, the addition circuit adds the two source operands together to produce their sum
- Sum is held in the ALU circuitry
- This is the actual computation

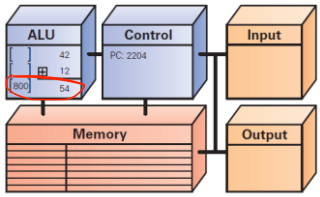


Figure 9.9 Instruction Execute: Compute the result of the operation in the ALU.

Result Return (RR)

- RR returns the result of EX to the memory location specified by the destination address.
- Once the result is stored, the cycle begins again

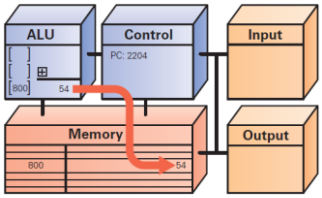


Figure 9.10 Result Return: Store the result from the ALU into the memory at the destination address.

Many Simple Operations ISA: instruction set Architecture

- Computers "**know**" very few instructions
- The decoder hardware in the controller recognizes, and the ALU performs, only about 100 or so different instructions (with a lot of duplication)
- Everything that computers do must be reduced to some **combination** of these primitive, hardwired instructions
- ADD is representative of the complexity of computer instructions…some are **slightly simpler**, some **slightly more complex.**
- Computers achieve success at what they can do with speed i.e. **millions/billions** of instructions per second

(Clock Speed in MHz/GHz)

Computer's View of Software

- A computer "sees" software as a long sequence of 4-byte (32-bit) groups of bits (0's and 1's)

```
. . . 10001111 10010100 00000011 01110100
      10001111 10011000 00000001 10101100
      00000010 10011000 10100000 00100000  ↔  ADD 20, 20, 24
      10101111 10010100 00000001 10010000 . . .
```

This binary object file can be hundreds of thousands to millions of words long.

- **Once installed, the computer runs the software by:**
  - Copying the binary instructions into the RAM
  - Executing (running) them using the Fetch/Execute Cycle.
- **It does whatever the instructions tell it to do!**

Summary

- All computers, regardless of their implementing technology, have five basic parts or subsystems:
  - The memory, a very long sequence of bytes, each with an address, stores the program and data while the program is running.
  - The ALU does the actual computing.
  - The Control Unit controls the activity of all other units in the computer system, giving them directions.
  - The input and output units are the interfaces for the peripheral devices connected to the computer.
- A program is executed according to the Fetch/Execute Cycle:
  - The repeating process fetches each instruction (indicated by the PC), decodes the operation, retrieves the data, performs the operation, and stores the result back into the memory.