Computer Systems and Networks

ECPE 170 – Jeff Shafer – University of the Pacific

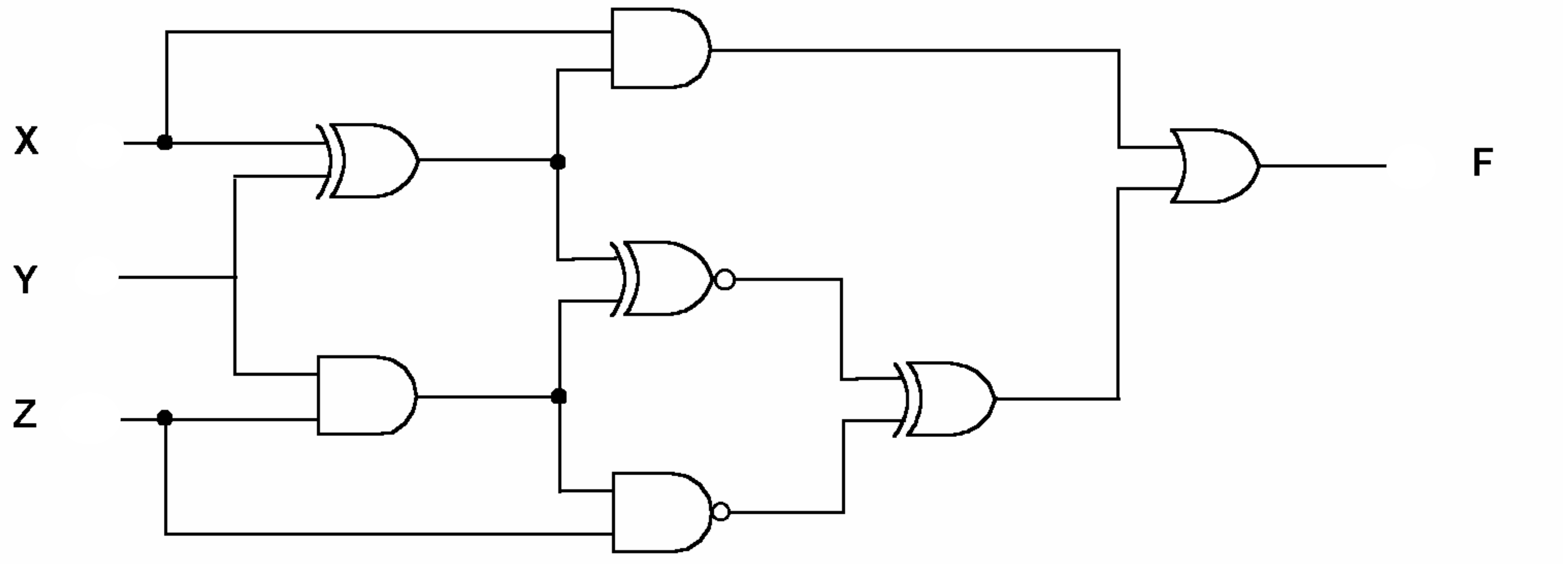
Design of a Simple Computer

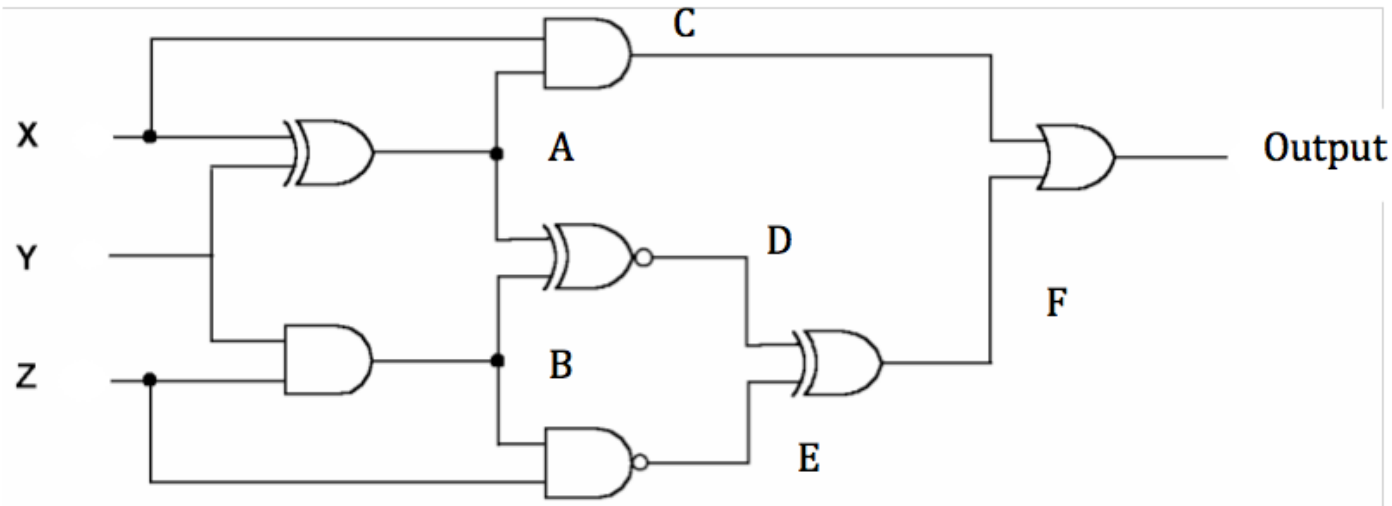
Schedule

- **Today**
  - Simple computer organization + **Quiz 2**
- **Monday 6th**
  - Simple computer organization (continued)
  - Exam review
- **Wednesday 8th - Exam 1**
  - Exam covers all of Chapters 2 and 3
  - Study: Homework, Quizzes, review slides
- **Friday 10th**
  - Introduce new machine architecture – MARIE – and assembly programming language

Homework 5 – 3.34

- **Find the truth table that describes the circuit:**





x	Y	z	A	B	C	D	E	F	Output
0	0	0	*0*	*0*	*0*	*1*	*1*	*0*	**0**
0	0	1	*0*	*0*	*0*	*1*	*1*	*0*	**0**
0	1	0	*1*	*0*	*0*	*0*	*1*	*1*	**1**
0	1	1	*1*	*1*	*0*	*1*	*0*	*1*	**1**
1	0	0	*1*	*0*	*1*	*0*	*1*	*1*	**1**
1	0	1	*1*	*0*	*1*	*0*	*1*	*1*	**1**
1	1	0	*0*	*0*	*0*	*1*	*1*	*0*	**0**
1	1	1	*0*	*1*	*0*	*0*	*0*	*0*	**0**

Tip 1: The little circle on a gate typically represents negation (of either the input or output, depending on location). Thus, this circuit contains a negated AND (NAND) gate, as well as a negated exclusive-OR gate.

Tip 2: Don't try to do it all in your head! Add intermediate points (A-F) to see data progressing through circuit.

Measures of Capacity and Speed

- Kilo- (K) = 1 thousand = 10^3 and 2^{10}
- Mega- (M) = 1 million = 10^6 and 2^{20}
- Giga- (G) = 1 billion = 10^9 and 2^{30}
- Tera- (T) = 1 trillion = 10^{12} and 2^{40}
- Peta- (P) = 1 quadrillion = 10^{15} and 2^{50}
- Exa- (E) = 1 quintillion = 10^{18} and 2^{60}
- Zetta- (Z) = 1 sextillion = 10^{21} and 2^{70}
- Yotta- (Y) = 1 septillion = 10^{24} and 2^{80}

Whether a metric refers to a power of ten or a power of two typically depends upon what is being measured.

Introduction

- **Chapter 4 in textbook**
- Course to date
  - Chapter 2 – **Representing** numbers/letters in a computer-friendly format
  - Chapter 3 - Creating **digital circuits** that implement Boolean functions and store data
- Next goal
  - Combine these basic components to build a simple (but functional) computer
  - Program that computer (in **assembly language**)

CPU Basics

- Steps to run a program?
  - **Fetch** instruction from memory
  - **Decode** instruction to determine operation
  - **Execute** instruction
- Many components are needed to accomplish this

CPU Basics

- Two main components: datapath and control unit
- **Datapath**
  - Arithmetic-logic unit
  - Storage units (registers)
  - Connected by a data bus that also reaches main memory
- **Control Unit**
  - Responsible for sequencing operations
    - What does hardware do first?
    - What does hardware do second?
    - What does the ALU do?

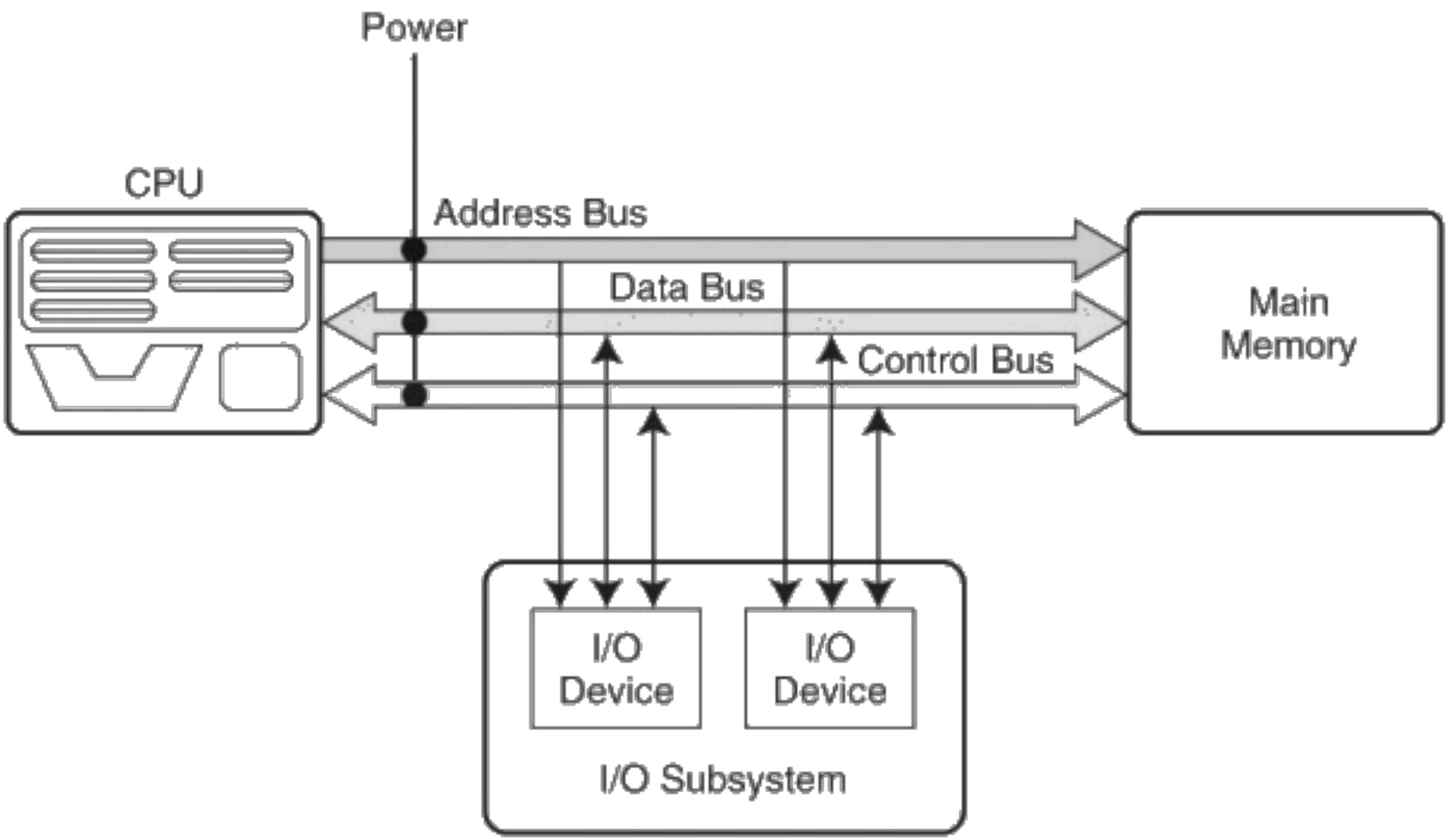
Registers

- Registers hold data that can be readily accessed by the CPU
  - **Much faster** than main memory
- Implemented using D flip-flops
  - A 32-bit register requires 32 D flip-flops

Data Bus

- Data bus moves data between CPU components
  - A bus is a **set of wires** (8, 16, 32, 64, …)
  - One bit per wire per clock cycle
- Bus components
  - **Data lines**
    - Move data
  - **Address lines**
    - Determine location of data (either source or destination)
  - **Control lines**
    - Determine direction of data flow

Example Bus


Power
CPU
Address Bus
Data Bus
Control Bus
Main Memory
I/O Device
I/O Device
I/O Subsystem

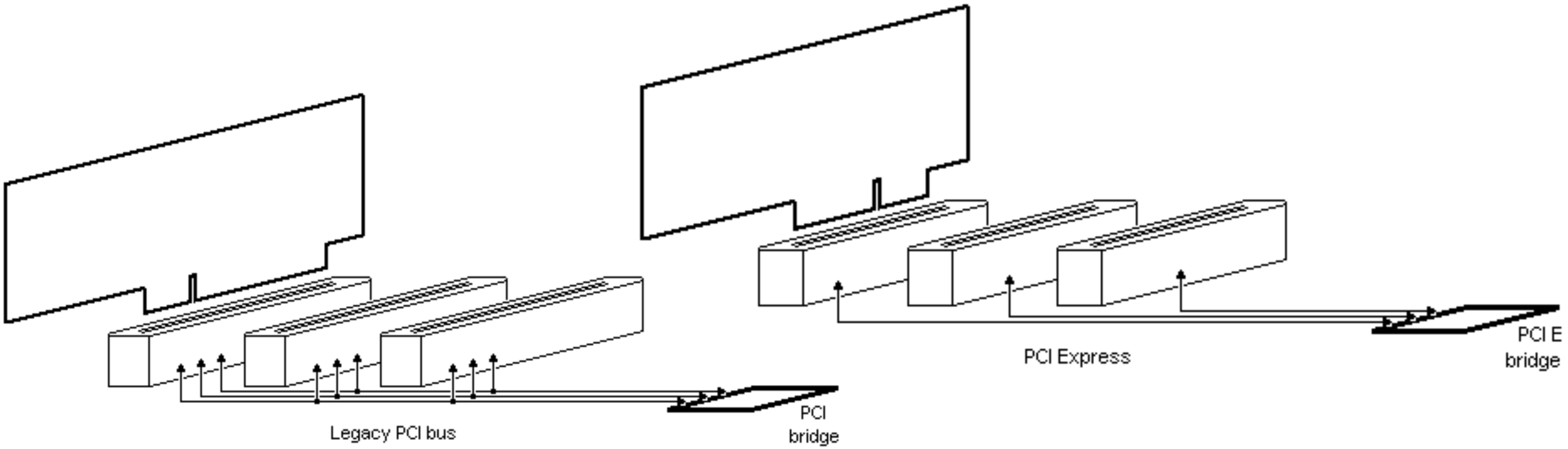
Point-to-Point vs Multipoint

Multipoint Bus

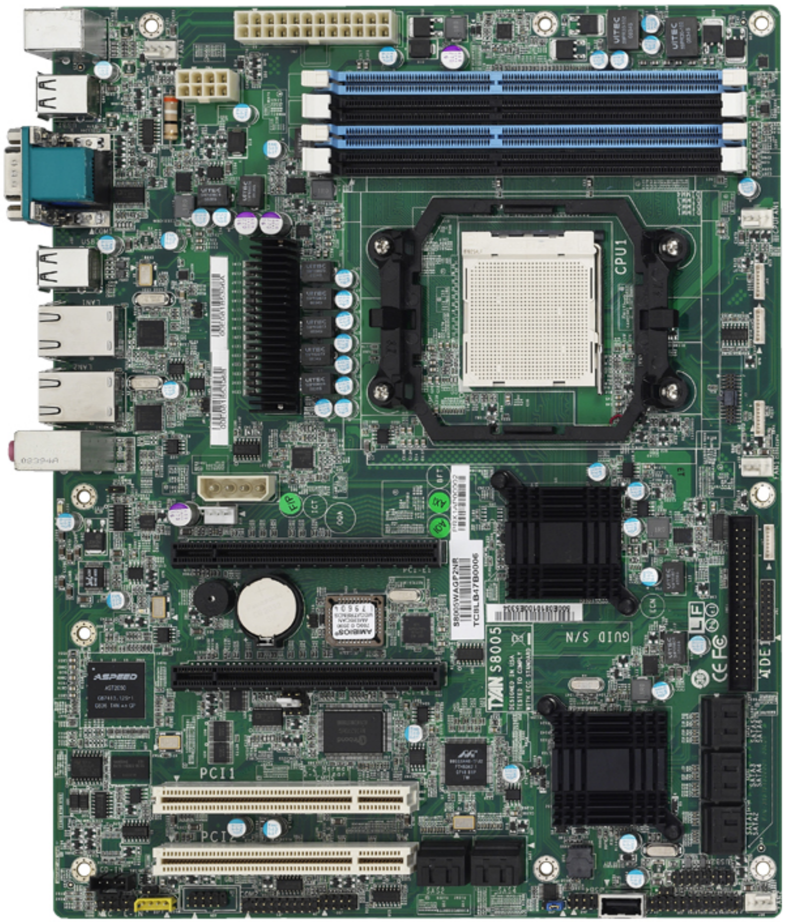
- Connect two components via **shared** wires
- Example: PCI bus

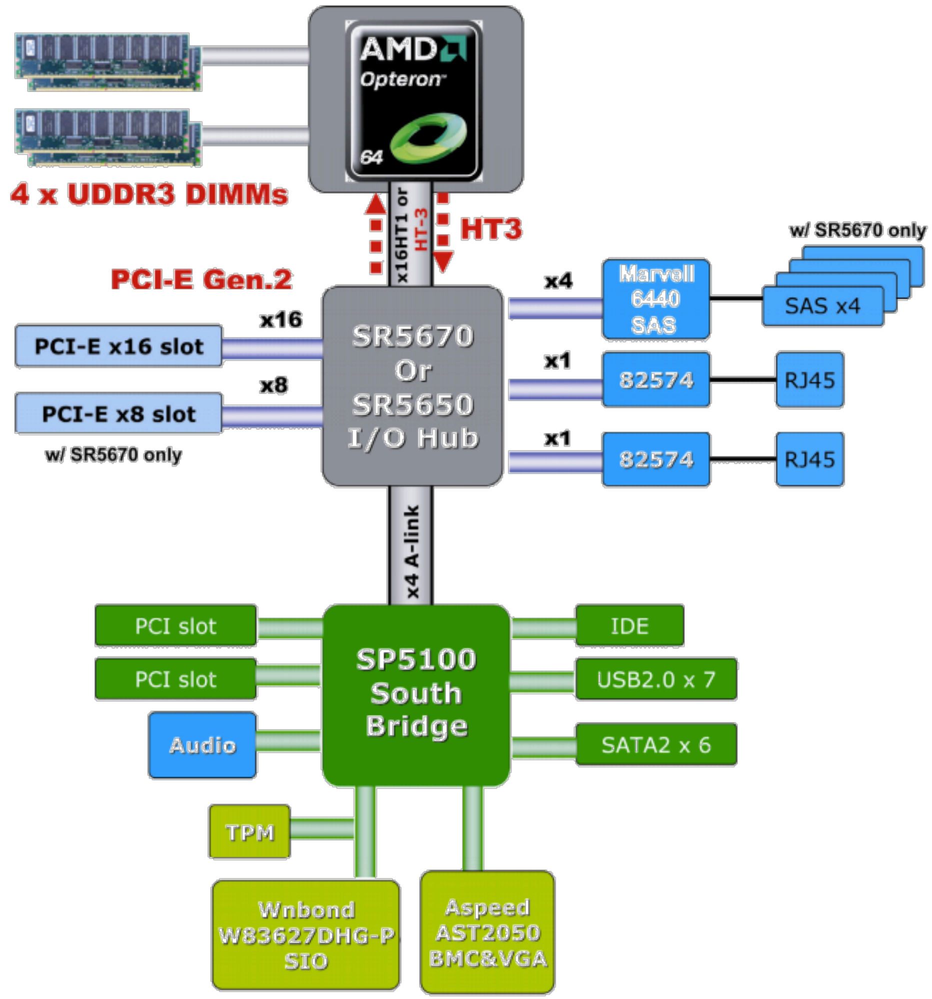
Point-to-Point Bus

- Connect multiple components via **dedicated** wires
- Example: PCI-e bus



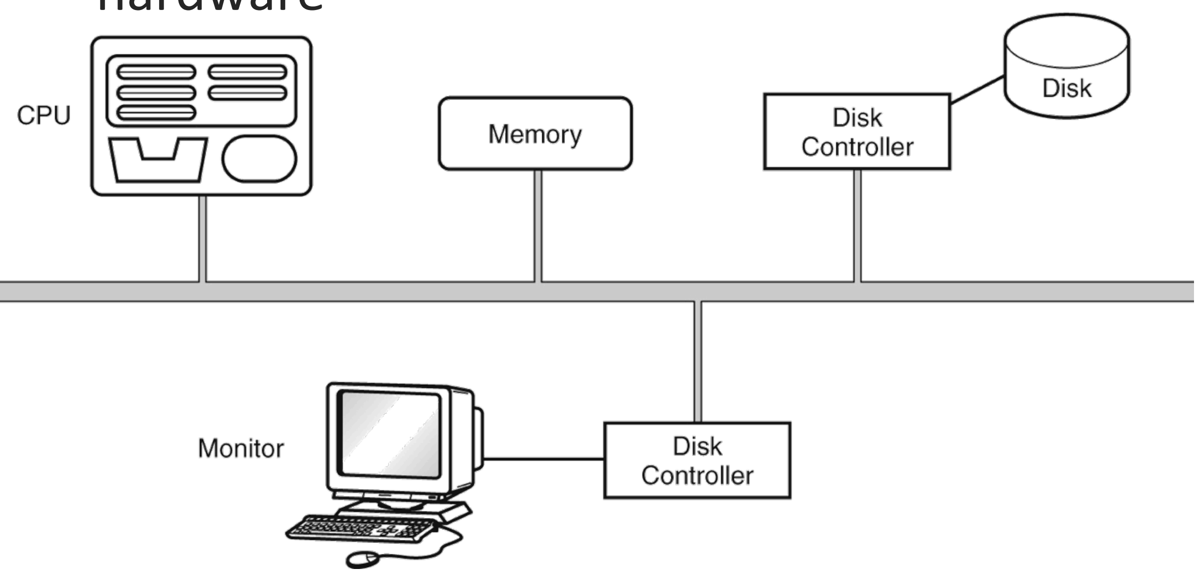
Modern AMD Opteron System





Multipoint Bus

- Multipoint bus is a shared resource
  - When can I access this shared resource?
  - What can others access it?
  - Controlled through protocols implemented in hardware



Clocks

- Computer components must be carefully synchronized
  - Use a **clock** (think of a "drummer", rather than a "watch")
- Fixed number of clock cycles are required to carry out each data movement or computational operation
- Clock frequency determines the speed with which all operations are carried out.
  - Measured in megahertz or gigahertz
  - Clock cycle time is the reciprocal of clock frequency
    - An 800 MHz clock has a cycle time of 1.25 ns.

Clocks

- **Clock speed does not (*directly*) equal CPU performance!**
- CPU time required to run a program:

$$\text{CPU Time} = \frac{\text{seconds}}{\text{program}} = \frac{\text{instructions}}{\text{program}} * \frac{\text{avg.cycles}}{\text{instruction}} * \frac{\text{seconds}}{\text{cycle}}$$

- How can we decrease CPU time? Many ways!
  - Reduce the number of instructions in a program
  - Reduce the number of cycles per instruction
  - Reduce the number of nanoseconds per clock cycle

The Input/Output Subsystem

- A computer communicates with the outside world through its input/output (I/O) subsystem
- Two different ways I/O devices can function
  - **Memory-mapped**: the I/O device behaves like main memory from the CPU's point of view.
  - **Instruction-based**: the CPU has a specialized I/O instruction set
- Modern devices are typically memory-mapped
  - But CPUs still have legacy I/O instructions…

Interrupts

- High priority events (requiring immediate handling) can alter normal program flow
  - I/O requests
  - Arithmetic errors (division by 0)
  - Invalid instructions
- CPU is notified of the high-priority event via an **interrupt**
  - Nonmaskable interrupts are high-priority interrupts that cannot be ignored
- Each interrupt is associated with a procedure (subroutine) that tells the CPU what to do
  - Copy data from the NIC?
  - Give the video card a new frame to display?