University of Virginia

Department of Physics

Physics 606: How Things Work II

Lecture #22 Slides:

Computers

Computers

Question:

Today, the fastest PCs run at roughly 1.5 GHz. Someday, computers may run at 1,000,000 GHz. Compared to present computers, those high-speed ones would have to be

1. much larger.
2. much smaller.
3. about the same size.

Observations About Computers

- They respond to inputs with various outputs
- They handle all kinds of information
- Information is measured in bits and bytes
- Some information is lost when power fails
- Computers work extremely quickly
- They follow instructions perfectly

Analog Representation

- A number is represented by a physical quantity
  - Current
  - Voltage
  - Magnetization
- Number is proportional to the physical quantity
- Precision is determined by the quantity itself

Digital Representation

- A numbers is represent by physical quantities
- Physical quantities take on discrete values
- These values represent pieces of the number
- Precision is determined by number of quantities

Binary Representation

- Each physical quantity has two values
  - One value is defined as a “1”
  - The other value is defined as a “0”
- Each quantity represents one information bit
- A number is represented by several bits
- The more bits, the more precision
- Bits are relatively immune to noise

Example: 19

- Five bits can represent number from 0 to 31
- 19 is represented by the bits: 10011
- Each bit represents a power of 2
- $1 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 19$

Representing Non-Numbers

- Bits or groups of bits are assigned to objects
  - Characters
  - Colors
  - Days of the week
- 8 bits (a byte) can distinguish 256 objects
- Two bytes can distinguish 65,536 objects

Quantities Representing Bits

- Current
- Magnetization
- Charge
- Optical properties
- Light
- Radio Waves
- Sound

Computers & Bits

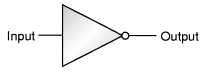
- Computation: currents
- Memory: charge
- Disk Drives: magnetization
- CDROM/DVDROM: optical properties

Computing

- Computers perform logical operations with bits
- Complicated operations based on simple ones
- Simplest operations: inversion & not-and
- Any function can be realized from these two

Inverter

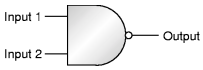
- Takes one input bit, provides one output bit
- Output bit is inverse of input bit



Input	Output
1	0
0	1

Not-And (NAND)

- Takes two input bits, provides one output bit
- Output is inverse of logical “and” of input bits



Input 1	Input 2	Output
1	1	0
1	0	1
0	1	1
0	0	1