Introduction to Digital Logic

ECGR2181
Lecture Notes 1

Reading: syllabus, Chapter 1
Computer System: Layers of Abstraction

Application Program
Language
Operating System

Instruction Set Architecture
(and I/O Interfaces)

Microarchitecture
Logic Gates
Circuits
Devices

Software
Hardware
Big Idea #1: Transformations Between Layers

How do we solve a problem using a computer? A systematic sequence of transformations between layers of abstraction.

- **Problem**
- **Algorithm**
- **Program**
- **Instr Set Architecture**

**Software Design:** choose algorithms and data structures

**Programming:** use language to express design

**Compiling/Interpreting:** convert language to machine instructions
Deeper and Deeper...

Processor Design: choose structures to implement ISA

Logic/Circuit Design: gates and low-level circuits to implement components

Process Engineering & Fabrication: develop and manufacture lowest-level components
Many Choices at Each Layer

Solve a system of equations

- Red-black SOR
- Gaussian elimination
- Jacobi iteration
- Multigrid

Tradeoffs:
cost
performance
power
(etc.)

- FORTRAN
- C
- C++
- Java

- Sun SPARC
- Intel x86
- Compaq Alpha

- Pentium II
- Pentium III
- AMD Athlon

- Ripple-carry adder
- Carry-lookahead adder

- CMOS
- Bipolar
- GaAs
How do we represent data in a computer?

At the lowest level, a computer is an electronic machine.
  - works by controlling the flow of electrons

Easy to recognize two conditions:
  1. presence of a voltage – we’ll call this state “1”
  2. absence of a voltage – we’ll call this state “0”

Could base state on value of voltage, but control and detection circuits more complex.
  - compare turning on a light switch to measuring or regulating voltage

We’ll see examples of these circuits in later chapters.
Computer is a binary digital system.

Digital system:
- finite number of symbols

Binary (base two) system:
- has two states: 0 and 1

Basic unit of information is the *binary digit*, or *bit*.

Values with more than two states require multiple bits.
- A collection of two bits has four possible states:
  00, 01, 10, 11
- A collection of three bits has eight possible states:
  
  A *collection of n bits has $2^n$ possible states.*
Basic Logic Gates

**NOT**

\[ \overline{A} \]

**OR**

\[ A + B \]

**AND**

\[ AB \]

**NOR**

\[ \overline{A + B} \]

**NAND**

\[ \overline{AB} \]
# Building a Truth Table

<table>
<thead>
<tr>
<th>AND</th>
<th>OR</th>
<th>NOT</th>
</tr>
</thead>
</table>
Inverter (NOT Gate)

<table>
<thead>
<tr>
<th>In</th>
<th>Out</th>
</tr>
</thead>
<tbody>
<tr>
<td>0V</td>
<td>2.9V</td>
</tr>
<tr>
<td>2.9V</td>
<td>0V</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>In</th>
<th>Out</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
Electronics Packaging

- There are several packaging technologies available that an engineer can use to create electronic devices.
- Some are suitable for inexpensive toys but not miniature consumer products, and some are suitable for miniature consumer products but not inexpensive toys.
- These packages have metal leads that are the conductive wire that connect electricity from the outside world to the silicon inside the package.
- Leads between packages are connected with small copper traces on a printed circuit board (PCB), and the package leads are soldered to the PCB.
Examples of Electronics Packages

Dual In-line Package (DIP) Older technology, requires the metal leads to go through a hole in the printed circuit board.

Dual Flat Pack (DFP) - A fairly recent technology, metal leads solder to the surface of the printed circuit board.
Examples of Electronics Packages

Quad Flat Pack (QFP) - like the Dual Flat Pack, except here are metal leads are on four sides.

Ball Grid Array (BGA) - The connections to the component are on the bottom of the chip, and have balls of solder on these connections.
Using these Components
The End Products
Before Next Class . . . .

• Visit the class website
  – Homework 1 will be posted
  – Transparencies will be posted
• Read Chapter 1 and 2