2018-02-06

Class Information

- Professor: Chris Terman(cjt@mit.edu)
- Class on implementation of digital systems
- Code to 6.004 lab = 775533

Introduction to Information

- Technology is organized using abstraction along a stack of concepts
  - 6.004 goes from digital circuits to cloud computing
- Digital systems are characterized as the flow of "information" through a circuit
  - But how do we define it?
    - **Information** = data communicated or received that reduces the ambiguity in a solution space
      - Data differs from information in that information has to provide specificity towards some kind of end result

Mathematical formulation

Given a discrete random variable X with possible states $x_i \in \{x_1, x_2, x_3, ...\}$ with each state having probibility $p_i \in \{p_1, p_2, p_3, ...\}$

The amount of information passed by specifying that the x_nth state is the actual state is

$$I(n) = log_2\left(\frac{1}{p_n}\right)$$

If you go from N equally possible states to M possible states, the probability that the actual state would be within the N elements is $p = \frac{N}{M}$.

So, for that situation,

$$I(n) = log_2\left(\frac{M}{N}\right)$$

Entropy

- **Entropy** = the average of each information value for each state in the state sapce, weighted by its its probibility

  - Represents the "average" number of bits you need to send the actual data

Encoding

- **Encoding** = a mapping between bit strings and values in the state space
  - **Fixed-length encoding** = a mapping where each bitstring that maps to a state is the same length
    - *e.g.* ASCII
  - **Variable-length encoding** = a mapping where bitstrings that map to certain states can vary by length
    - *e.g.* UTF-8
    - *How do you choose bit tokens so a bitstream can be unambiguously represented?*

Binary tree representation of encodings

- Unambiguous codes can be written as a binary tree
  - Fixed-length encodings have all end states the same distance down the tree
  - Variable-length encodings have end states that vary in difference from the root of the tree

A note on negative integer representation

- In this class, we use two's complement encoding

Huffman Codes

- **Huffman codes** = scheme for creating encodings where the bit tokens for low-information(high probability) states are short and the bit token for high-information(low-probability) states are long
- Algorithm:
  1. Create tree with two elements with states with smallest p_i
  2. Merge that subtree with the element with the next smallest p_i
  3. Terminate once all states are assigned a place

Error Detection

- **Hamming distance** = the number of digits for which two bit strings of equal length are different
  - If a single bit flips, the hamming distance increases by one

- **Even-parity single bit check** = add a bit to the end of the

---

2018-02-08

Physical Encoding of Bits

- We've talked about how to use bits in an abstract sense
  - Now, we're interested in how we should represent them in physical systems

What Do We Want in a Physical Bit System?

1. Cheap & Small
   - We will need a **lot** of them
2. Stable
   - State doesn't change unless we tell it to change
3. Ease of manipulation
   - Operations like access, inverse, moving, *etc*

- In addition, we should use things that are studdied in EECS
  - *e.g.*
    - Voltages(what we'll use)
    - Currents
    - Phase
    - Frequency

Example: Voltage Representing a Grayscale Image

- Let each pixel have an analog value between 0V and 1V
  - *Question: how much information does one pixel's value contain?*
- **Operation blocks** = a bundle of functionality that maps one pixel voltage to a resultant voltage
  - Repeated application of operation blocks using analog signals will have a compounding error effect in the presence of ANY noise
    - The solution is digital systems
      - This abstraction permits our systems to tolerate reasonable amounts of noise and physical imperfection

What Causes Noise?

- Things like

  - Parasitic resistance
  - Inductance
  - Capacitance
  - IR drop
  - External EM fields

Digital Processing Element

- **Digital processing element** = a **combinational device** with
  - One or more **digital inputs**
  - One or more **digital outputs**
  - A **functional specification**
    - **Functional specification** = essentially a truth-table that defines how the element maps inputs to outputs
  - A **timing specification**
    - **Timing specification** = the amount of time it takes for digital processing to take place
      - Most modern logic gates take no more than $10ps$
- A device is a digital processing element
  1. Each element is a digital processing element
  2. Every input is connected to exactly one output or to some vast supply of constant voltage(1 or 0)
  3. Must not contain cycles
     - *Why?*

Attempts at a Solution

Attempt 1

- Central idea is to create a threshold V_{TH} such that
  - If $V < V_{TH}$, then it is a 0
  - If $V > V_{TH}$, then it is a 1
- Downside
  - It is nearly impossible to distinguish $V + \epsilon$ from $V - \epsilon$

Attempt 2

- Central idea is to create two thresholds V_L and V_H such that
  - If $V < V_L$, then it is a 0
  - If $V < V_H$, then it is a 1
- Is a much better system because there is no point where a small noise disturbance can sway the result
- Downside

- The middle range is ambiguous

Attempt 3

- Central observation is that input processing is the only time when bit interpretation needs to happen
  - Thus, we need a wider margin for input margins than for output margins
  - If digital signal represents and output, then
    - If $V < V_{OL}$, then it is a 0
    - If $V > V_{OH}$, then it is a 1
  - If digital signal represents and input, then
    - If $V < V_{IL}$, then it is a 0
    - If $V > V_{IH}$, then it is a 1

Buffers

- **Buffer** = a digital processing element that takes in an analog signal and outputs a digital voltage
- **Voltage Transfer Characteristic(VTC)** = a graph of input voltage vs inte
  - Note
    1. The VTC behavior in the "forbidden zone" can be anything, as those aren't valid function inputs anyways
    2. The forbidden zone in the VTC must be taller than it is wide
       - Specifically, $V_{OH} - V_{OL} > V_{IH} - V_{IL}$
       - Result
         1. Combinational device must have **gain** > 1
         2. Combinational device must be nonlinear

2018-02-13

CMOS Technology

MOSFETs

- **MOSFETs(Metal-Oxide-Semiconductor Field-Effect Transistor)** = four-terminal device that allows current to flow between **diffusion terminals** if **gate terminal** voltage is high enough
  - **Diffusion terminals** = the logical input and output of a MOSFET
    - **Drain** = the terminal where an input signal is placed

    - * **Source** = the output terminal which either receives the source signal or doesn't, depending on the value at the gate terminal
  - **Gate terminal** = the terminal that lies right above the channel of Silicon that seperates the drain and source
    - * Placing a positive voltage on the gate creates attractive force that permits electrons to flow from the drain to the source
  - **Bulk terminal** = connection to either ground or V_{CC} depending on what type of FET

NFET vs PFET

- **NFET** = diffusion terminals are of N type and substrate is of P type
  - Bulk should connect to ground
- **PFET** = diffusion terminals are of P type and substrate is of N type
  - Bulk should connect to V_{CC}

"Complementary MOS"

- Two MOSFET circuits are complementary when exactly one circuit conducts for any given permutation of inputs

CMOS and Inversion

- Generally, MOS circuits are designed with a **pull-up circuit** and a **pull-down circuit**
  - **Pull-up circuit** = composed of PFETs which conduct on 0
  - **Pull-down circuit** = composed of NFETs which conduct on 1

Propogation Delay and Contamination Delay

- **Propogation delay(T_{PD})** = the maximum amount of time a FET device will take to convert valid input to valid output
- **Contamination delay(T_{CD})** = the minimum amount of time a FET device will take to produce an invalid output once an input becomes invalid
  - Measure of gain?
  - Why minimum?

2018-02-15

Combinational Logic

- **Standard Cell Library** = the collection of FET circuits that are used to create higher level components
  - Think *standard library* in a programming language like C++
- The purpose of this lecture is to begin using a small standard cell library to begin synthesizing logical systems

Functional Specifications

- There are many ways to describe how a logical element should behave
  - "Word description" just means that you describe the relationship between
  - **Truth table** = an exhaustive listing of each permutation of inputs and the corresponding outputs
  - **Boolean expression** = a formula using boolean algebra operations(AND, OR, NOT)

$$Y = \overline{A} \cdot B + C$$

    * **Sum-of-Products** = a boolean expression composed of several sub-expressions, connected by OR's, that are themselves composed of compund AND's
      · For ANY truth table, you can represent the function by a sum-of-products with one term for each 1 in the output column of the truth table

Logical Universality

- If a set of gates can be used to encode any logical computation, the set is called **logically universal**
  - *e.g.* NAND is logically universal

Demorgan's Laws

$$\overline{A} \cdot \overline{B} = \overline{A + B}$$
$$\overline{A} + \overline{B} = \overline{A \cdot B}$$

Leniency

- What is leniency?

Multiplexer

- **Multiplexer** = a logical element that has a control bus which specifies what input to forward out to and output