Guided Reading in CODE

The reading in **Code** builds up to how we can use binary and transistors to build a computer. Essentially the history of representing information using alternative methods such as Morse code and Braille, were technology breakthroughs. Automating alternative information representations with electricity continued this progress.

Start reading...

Chapters 1-10 describe basic concepts and history building up to digital logic.

Codes are any system of communicating information where traditional media are impractical or impossible. Media such as spoken word or hand signals rely on unbroken line-of-sight. Written words, while useful for communicating large amounts of data, must be physically carried from place to place.

Many forms of coded information have been invented to bridge these obstacles. The main advantage of a well constructed code, is its ability to be transmitted through, and received and interpreted by machines. Machines have advantages over human eyes and ears in that, when well engineered, they make fewer mistakes, and when powered by electricity, they can transmit signals far faster than a human could ever travel.

A major leap forward in this trend toward the use of coded machine signals was the telegraph relay. Through a clever, but relatively simple implementation of electromagnets, telegraph relays were arguably the first "processors." Their process was simple: receive signal on one end, and transmit the same signal on the other end. But for the first time, incoming electrical signals were flipping electrical switches. The introduction of logic gates to circuitry took that concept of incoming signals flipping switches which affect outgoing signals, amplified its complexity by several orders of magnitude, and that became the groundwork for modern computing.

Chapter 11

1. Briefly describe the four gates explained in Chapter 11. Describe the behavior of each gate based on the input values to the gate.

   Logic gates are specialized switches which take two incoming electrical signals (or inputs) and return a single outgoing signal (output). The input and output are either "ON" (current is running through the wire) represented by 1 in binary, or "OFF" (current is not running) represented by 0. The four logic gates are:

AND - Returns a 1 output if and only if both inputs are 1. Returns a 0 output otherwise.

OR - Returns a 0 output if and only if both inputs are 0. Returns a 1 output otherwise.

NAND - Returns a 0 output if and only if both inputs are 1. Returns a 1 output otherwise. A NAND gate is built by connecting an AND gate to an inverter, which reverses its output.

NOR - Returns a 1 output if and only if both inputs are 0. Returns a 0 output otherwise. A NOR gate is built by connecting an OR gate to an inverter, which reverses its output.

Similar to logic gates are inverter and buffers. Although similar in their construction, these switches return a single output from a single input, rather than two inputs:

Inverter (NOT) - Returns a 1 output if the input is 0. Returns a 0 output if the input is 1. This is used to create the NAND and NOR gates from the AND and OR gates respectively.

buffer - Returns a 1 output if the input is 0. Returns a 0 output if the input is 1. This doesn't alter the code, but it can be used to strengthen a weak signal over (relatively) long distances, much like the telegraph relay described in Chapter 6.

Chapter 12

1. A half adder is built from how many "sub components" and has how many inputs and outputs?
2. A full adder is built from how many "sub components" and has how many inputs and outputs?
3. How many total inputs and outputs are there for an 8-bit adding circuit?
4. How many total transistors are needed for the ripple version of the 8-bit adder?

Chapter 13

1. Design the following circuits with Logisim and post a snapshot of the circuit (a jing video of showing the circuit in action would be nice).
    a. Half Adder and Full Adder
    b. 8 Bit Adder
    c. 8 Bit Adder with the Ones' Complement circuit and the Sub input to the circuit for subtraction

Chapter 14

1. Design the following circuits with Logisim and post a snapshot of the circuit (a jing video of showing the circuit in action would be nice).
    a. Adding machine (page ~~159~~**168**). A bit of a challenge

   b. Edge Triggered D FlipFlop (page ~~163~~**172**)
   c. 8 Bit ripple counter (use Logisim D FF, page ~~168~~**177**)

Chapter 15

1. Define byte.
2. What is the range of an unsigned byte? A signed two's complement byte?
3. What does the X in the following binary number represent 1010**X**10101?
4. What is the base 16 number system called?
5. Why is base 16 a good choice for representing binary numbers?
6. What base 16 number is 101011000011? What is an easy way to determine this without using a calculator?
7. What does the X represent in the following base 16 number D4A**X**F0?
8. The book describes how to convert decimal to base 2 or base 16. Research and describe a technique that allows conversion of decimal to any base (2-36).

Chapter 16

1. What do the prefix names kilo, mega, giga, tera, and peta represent in bytes and what were the Greek origins of the words?
2. At the end of Chapter 16 the author reminds the reader about something that is "very important," what is this?