Boolean Logic

Tom Kelliher, CS 220

Chapter 1, Textbook

- The truth table for a Boolean function of n variables has how many rows?
- Name two ways of representing a Boolean function.
- Correspondence between truth values, bits, and voltages in circuits:

  false is represented by the bit '0' and 0 volts (Gnd or ground)

  true is represented by the bit '1' and 3.3 or 5 volts, depending upon the logic family used. These voltages are often represented by the symbols V_{cc} or V_{dd}.
- Given a Boolean function, construct its truth table. For example, $f(a, b, c) = ab + ac + bc$.
- Given a truth table, construct the canonical representation of the corresponding Boolean function. For example, start with the truth table of Figure 1.3 on pg. 11 (1st ed., Figure 1.1 on pg. 9).
- Using only two-input AND gates (as many as you need), implement a four-input AND gate.
- Using only two-input OR gates (as many as you want), implement an eight-input OR gate.
- Implement the majority function: $f(a, b, c) = ab + ac + bc$.

  Write an HDL program implementing this function.

Chapter 1, Slides

- Slide 3: Note additional examples of truth tables.
- Slide 4: Note the use of function call notation, but a direct correspondence to a gate or HDL implementation. These will be useful when you implement gates in HDL.

  De Morgan's law: $\overline{A + B} = \overline{A} \cdot \overline{B}$.

  How do we get to $A + B = \overline{\overline{A} \cdot \overline{B}}$?
- Slide 5: It looks like the 2 Mux has an implementation with four product terms. It turns out that only two product terms are needed. How? How can we confirm that the resulting implementation is correct?

  How is a 4 Mux implemented from 2 Muxes?

- Slide 10: The gate input assignment can be confusing. Remember this:
  `formal = actual`
  where `formal` is a pin and `actual` is an external wire.