1. Design a circuit to convert a 3-bit Greys code number into standard 3-bit binary. Start by writing down the truth table, clearly labeling the input and output bits. Solve the logic for each output bit (using a K-map unless the logic is trivial) and write down the boolean algebra expression for each output bit. *You do not need to draw this out using gates.* Note: this is a bonus K-map problem from last week...

2. You have three switches, a battery, a light bulb, and an infinite supply of wire. Devise a way to realize the logic expression \( A \cdot (B + C) \), where the switch positions are the inputs and the light bulb is the “output.” [Steck, Problem 11.2]

3. Design a 4-input (digital) multiplexer from ordinary logic gates. Your circuit should also include an ENABLE line so that the output will always be LO unless ENABLE is HI. Show a high-level schematic of the inputs and outputs, and then show an implementation using ordinary gates. You can use multi-input AND/OR gates along with inversion circles if you wish.

4. Design a 2-bit decoder using gates. This device will have 2 input lines for the address value, and four output lines. One and only one output line will be HI depending upon the input address. Also include an ENABLE line so that the output will always be LO unless ENABLE is HI. Hint: You don’t need a K-map for this. Use multi-input AND/OR gates and inversion circles to simplify your diagram if you wish.

5. Look up the 74139, which is a 2:4 decoder/DEMUX with two sections per package. Show how five of these decoders can be combined to produce a 4:16 decoder. Each section of the 74139 has an ENABLE input, make sure your composite device has one also.

6. Design a circuit to indicate whether a 4-bit integer is prime. Start with a truth table, then implement the necessary logic using an 8:1 (8-input) multiplexer.

---

1Look this up on Wikipedia if you missed this in class