

CMPEN 270 Spring 2022

Design Project #2 – 5% of Final Grade

Instructions

1. **All submitted work must be independent. All college and university guidelines regarding academic integrity will be strictly enforced.**
2. **VERIFY the contents of any submissions is included in the Gradescope course drop box. No credit will be given for blank or corrupt submissions.**

Due Friday, 3/25/2022 @11:59pm

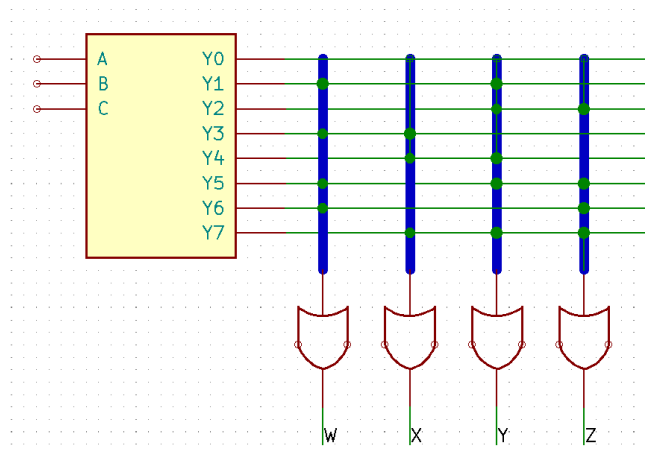
Read Only Memory (ROM) is a type of storage common in computers, video games, and other digital systems. This memory is hard-wired, in other words once it's programmed its contents cannot be changed. An example is a video game cartridge, which stores a program (instructions) to be run on a gaming system and isn't modified once manufactured.



A common method of implementing a ROM is to use an address decoder (1-hot for example) and a set of OR gates. The address is fed to the decoder inputs. Each output of the decoder has a *possible* connection to each OR gate, and we use as many OR gates as we want to store the bits contained in each address. Often this is shown as many inputs in a wire bus (which appears as one thick wire but is actually multiple individual wires).

When a customer wants a ROM programmed, they supply the manufacturer with a program, usually in the form of a truth table with addresses as the inputs, so that the manufacturer knows which connections to metalize between the decoder outputs and the OR gates. An example is shown below that uses a 1-hot decoder. A junction or dot shows a connection being made from a decoder output to the bus, which connects to the OR gate inputs. Can you verify that this circuit diagram would produce the truth table shown? Work through for your own understanding.

Address (ABC)	Output (WXYZ)
000	0000
001	1010
010	0011
011	1100
100	0110
101	1011
110	1001
111	0111



Your role is to develop your own ROM that could display an 8-letter word or phrase in all caps*. Some examples are FOOTBALL, TAILGATE, etc. Please verify that it's a word by using the SCRABBLE dictionary (<https://scrabble.hasbro.com/en-us/tools#dictionary>). Each address of the ROM will contain the 7-bit ASCII code of that letter. The 7-bits could be displayed using OUT6 to OUT0 on the DEB, for example. More details and an ASCII chart could be found in Lecture Set 2.

*In fact, your ROM when finished could be tested with an alphanumeric display! But it's only required to demonstrate the correct ASCII bits for this project.

1) Assuming 1-hot logic, what would be the largest OR gate needed to represent any of the output bits? HINT: this is a significant number. To save time in your design, you could use smaller gates, constant '1's or '0's, in some output bits. High input gates can be constructed in many ways, including using multiplexers, multiple smaller gates, etc. There is no one correct approach.

2) In your DEB (and in many decoders) the 74HC138N uses a 0-cold logic instead of a 1-hot. How would you modify or switch out the OR gates to work with your design?

3) Submit your answers to questions (1) and (2) and a written explanation of how you approached the design – about one page NEATLY handwritten or typed. Describe any methods used and include a schematic of the circuit design.

4) Build the circuit and take a video demo of your working prototype. A working demo shall be submitted via Gradescope along with your write-up. This is most easily done using a Digital Evaluation Board (DEB).

You may check your progress at any time, during office hours or lab. Experimentation is encouraged!

Although you may find and use IC's supplied by the lab, we do not guarantee that parts will always be available. You are responsible for obtaining your own hardware if needed, which is best with a DEB-1002. Please see more info in Lecture Set 1 or syllabus for acquiring necessary course materials.