Lab Work 2 (Part 3)

Other Combinational Logic Blocks

In this section, we are going to look at other commonly-used combinational logic circuits (decoder, encoder, multiplexer and demultiplexer.

Decoder

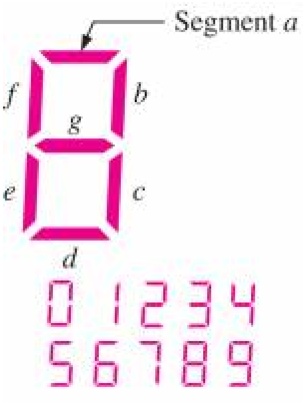
A decoder is supposed to detect/determine specific bit combinations (code). For an n-bit binary code, there can be up to 2^n combinations (thus, as many outputs). A common decoder is usually known as an $n to 2^n$ decoder (n input, 2^n output).

Truth Table for a 3-8 Decoder (active HI output):

A_2	A_1	A_0	Y_0	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

7-segment Display

This is a 'classic' device used to display decimal digits 0 to 9. It consists of 7 LED segments (hence the name) arranged as such to enable it display decimal digits (***Note****: It can also been used to display certain alphabets/letters*). To display binary or BCD value, a decoder is required to 'convert' the value into an output that can be used to drive the 7-segment.

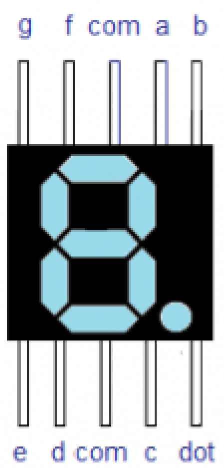


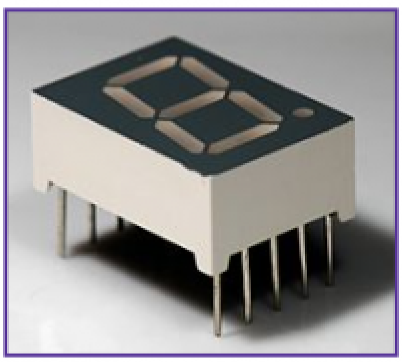
Disclaimer: *The image above is extracted from resources available for Digital Fundamentals 11th Edition (Global Edition)*

Truth Table for a 7-segment Decoder (active HI output):

A_3	A_2	A_1	A_0	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	1	1	1	0	0	1
0	1	0	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	1	0	1	1
0	1	1	0	1	0	1	1	1	1	1
0	1	1	1	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	1	0	1	1

There are two types of 7-segments: common anode and common cathode. A common anode 7-segment has the anode terminals of all LED in the package connected the the COM (common) pin. The same goes for common cathode.





Disclaimer: *The image above is taken from http://www.circuitstoday.com/interfacing-seven-segment-display-to-8051. I will remove the image if the copyright owner asks me to do so.*

Encoder

An encoder does the *inverse* decoder function. Therefore, it usually accepts a group of bits that has only 1 bit active at a time to represent a specific value or pattern. This can be converted by the encoder into a coded format (e.g. binary or BCD).

Notice that an encoded form usually has lower number of bits, so it can also be seen as a 'compression' function.

Truth Table for a BCD Encoder:

Decimal Digit	A_3	A_2	A_1	A_0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

Multiplexer

A multiplexer is a selection logic block. Multiple input lines can be selected to drive a single output line. Since the selector signal is in binary form, a multiplexer is usually found as a 2^n to 1 selection block (where n is the number of selector bits and $n>0$).

Truth Table for a 4-1 Multiplexer:

Selector Bits		**Output**
S_1	S_0	Y
0	0	D_0
0	1	D_1
1	0	D_2
1	1	D_3

Note that in the above truth table, data inputs (D_3-D_0)have been 'compressed'. If each of the 4 input bits is listed with all possibilities, we would need a 64-row table!

Demultiplexer

A demultiplexer is the inverse of a multiplexer (*duh!*). A single source signal can be routed to any one of multiple output lines, depending on the selector signal. One thing should be noted here is that a decoder can actually be used as a demultiplexer!

Things To Do

THING 1 Build a truth table for 2-4 decoder. Build the logic circuit and verify.

THING 2 Get the Boolean expression for a 7-segment decoder assuming we need to display **the numbers 0 to 3 only**. Build the logic circuit and verify.

THING 3 Build the logic circuit for a 4-1 multiplexer and verify.

THING 4 (Optional) Repeat **THING 2** so that it can display the full range 0-9.

THING 5 (Optional) Build a logic circuit to implement 1-4 demultiplexer. Feed a 1kHz TTL signal to the input. Verify that the output is visible on the selected output channel. ***Hint**: The circuit is available in the textbook!*

THING X (Optional) Consider using tristate buffer(s) in (de-)multiplexer circuits. What is the (dis-)advantage(s) of using this implementation?