Decoders:
- Input: A binary code. (The code can be thought of as a binary number.)
- Output: A single output corresponding to that code becomes True.

Output Equations: (Neglecting E.)
\[ \begin{align*}
Z_0 &= \overline{A_2}A_1A_0, \\
Z_1 &= \overline{A_2}A_1\overline{A_0}, \\
Z_2 &= \overline{A_2}A_1\overline{A_0}, \\
Z_3 &= \overline{A_2}A_1\overline{A_0}, \\
Z_4 &= A_2A_1\overline{A_0}, \\
Z_5 &= A_2A_1\overline{A_0}, \\
Z_6 &= A_2A_1\overline{A_0}, \\
Z_7 &= A_2A_1\overline{A_0}
\end{align*} \]
(Note: Each equation is a minterm.)
(How do you draw the output circuits?)

What is E(L)?
- E is an enable:
  - When E = 1, the decoder functions normally.
  - When E = 0, all of the outputs are 0.
  - The bubble on the diagram signifies active low.
  - E allows a chip to output all 0’s.
  - E can be used to prevent a chip from interfering with other operations.

-Decoders come in a variety of sizes including: 2-to-4, 3-to-8, 4-to-16
- We can create bigger decoders from smaller ones by using the enable.
  Example: Create a 3-to-8 decoder using two 2-to-4 decoders.

Note: By adding OR gates, we can even retain the Enable function.
**Encoders:**

Encoder:
- **Input:** Only one of the inputs may be True at one time.
- **Output:** A binary code/number corresponding to the True input.

![8-to-3 Encoder Diagram]

Output Equations:
\[
C_0 = I_7 + I_5 + I_3 + I_1 \\
C_1 = I_7 + I_6 + I_3 + I_2 \\
C_2 = I_7 + I_6 + I_5 + I_4
\]

Priority Encoder:
- **Input:** Any number of inputs may be True at one time.
- **Output:** A binary code/number corresponding to the highest priority input

![8-to-3 Priority Encoder Diagram]

What is W?
- W is True when any input is True.
- W is False when all inputs are False.
- W enables differentiation between I_0 and no input.

Output Equations can be determined by the use of K-maps.

Why is information “encoded?”
- Efficiency - More information is stored in fewer bits.
- Creates a reduction in the number of inputs and/or outputs required.
Multiplexers (MUX’s):
- Input: A number of signals and control inputs.
- Output: One of the input signals, as selected by the control inputs.

Output Equation: 
\[ Z = S_1S_0D_3 + S_1\overline{S_0}D_2 + \overline{S_1}S_0D_1 + \overline{S_1}\overline{S_0}D_0 \]

How do we implement the MUX? (Four 3-input ANDs and a 4-input OR.)

Example: Show the Truth Table and Voltage Table for a 4-input MUX where:
D_3, D_2, D_1, and D_0 are active low and S_1, S_0, and Y are active high.

<table>
<thead>
<tr>
<th>Truth Table</th>
</tr>
</thead>
<tbody>
<tr>
<td>D_3</td>
</tr>
<tr>
<td>*</td>
</tr>
<tr>
<td>*</td>
</tr>
<tr>
<td>*</td>
</tr>
<tr>
<td>*</td>
</tr>
<tr>
<td>*</td>
</tr>
<tr>
<td>*</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Voltage Table</th>
</tr>
</thead>
<tbody>
<tr>
<td>D_3</td>
</tr>
<tr>
<td>*</td>
</tr>
<tr>
<td>*</td>
</tr>
<tr>
<td>*</td>
</tr>
<tr>
<td>*</td>
</tr>
<tr>
<td>*</td>
</tr>
<tr>
<td>*</td>
</tr>
<tr>
<td>H</td>
</tr>
<tr>
<td>L</td>
</tr>
</tbody>
</table>

Note: Wildcards (*) are used to keep the tables to a reasonable number of rows.
Without wildcards, this table would take \(2^6 = 64\) rows.

Note 2: After grouping and counting in the Quartus waveform file, you can use the invert function on active-low inputs to put them in the right order.
Demultiplexer:
- Input: A single input signal and control inputs.
- Output: The input becomes the output on the line selected by the control inputs.

Output Equations:
\[
L_0 = \overline{A}S_1S_0, \quad L_1 = \overline{A}S_1S_0, \quad L_2 = AS_i\overline{S}_0, \quad L_3 = AS_iS_0
\]

Why do we care about MUX/DEMUX’s?
- They enable the use of a single line send to/from multiple sources.

Implementing Functions using a MUX:
- A multiplexer can be used to directly implement functions

Example #1: \( f = \overline{a}b + \overline{b}c + bc \)
- Select MUX size (4-input MUX)
- Select inputs for selector lines (‘a’ and ‘b’)
- Assign inputs to other input lines

\[
\begin{align*}
(ab)_2 = 00 & \rightarrow f = 1*0 + 1*\overline{c} + 0*c = \overline{c} \\
(ab)_2 = 01 & \rightarrow f = 1*1 + 0*\overline{c} + 1*c = 1 \\
(ab)_2 = 10 & \rightarrow f = 0*0 + 1*\overline{c} + 0*c = \overline{c} \\
(ab)_2 = 11 & \rightarrow f = 0*1 + 0*\overline{c} + 1*c = c
\end{align*}
\]

What do we connect to D1? => +5 Volts.
Example #1 (continued):

It is sometimes easier using a K-Map:

<table>
<thead>
<tr>
<th>ab \ c</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>01</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Just look at the output for each combination of ‘a’ and ‘b.’

Example #2: \( f = \overline{a}bcd + a\overline{c}d + bc\overline{d} \)

- Select MUX size (4-input MUX)
- Select inputs for selector lines (‘a’ and ‘b’)
- Assign inputs to other input lines

\[
(ab)_2 = 00 \rightarrow f = 1*0*c*d + 0*\overline{c}*d + 0*c*\overline{d} = 0
\]

\[
(ab)_2 = 01 \rightarrow f = 1*1*c*d + 0*\overline{c}*d + 1*c*\overline{d} = cd + cd = c
\]

\[
(ab)_2 = 10 \rightarrow f = 0*0*c*d + 1*\overline{c}*d + 0*c*\overline{d} = \overline{c}d
\]

\[
(ab)_2 = 11 \rightarrow f = 0*1*c*d + 1*\overline{c}*d + 0*c*\overline{d} = \overline{c}d + cd = c \oplus d
\]

Example #3:

DCBDCACABDCBAZ

- Use a 4-input MUX w/ Enable
- Find a common variable and attach it to the enable

\[
\]
Example #3 (continued):
- Determine other inputs (equations assume C is False)

\[(AD)_2 = 00 \rightarrow Z = (1 \cdot \overline{B} \cdot 0 + 0 \cdot B + 0 \cdot 0 + \overline{B} \cdot 1) = \overline{B}\]

\[(AD)_2 = 01 \rightarrow Z = (1 \cdot \overline{B} \cdot 1 + 0 \cdot B + 0 \cdot 1 + \overline{B} \cdot 0) = \overline{B}\]

\[(AD)_2 = 10 \rightarrow Z = (0 \cdot \overline{B} \cdot 0 + 1 \cdot B + 1 \cdot 0 + \overline{B} \cdot 1) = B + \overline{B} = 1\]

\[(AD)_2 = 11 \rightarrow Z = (0 \cdot \overline{B} \cdot 1 + 1 \cdot B + 1 \cdot 1 + \overline{B} \cdot 0) = 1\]

**Tri-State Buffers:**
- Have three states rather than two: 0, 1, and High-Z (Z = impedance)
  - If \(E = 0\), \(B\) is in High-Z
  - If \(E = 1\), \(B = A\)

\[\text{Ohm’s Law: } V = IZ \rightarrow I = \frac{V}{Z}, \text{ so as } Z \rightarrow \infty, I \rightarrow 0 \text{ (open circuit)}\]

- The outputs of tri-state buffers can be linked together so they share a wire
  Example: A MUX built from tri-state buffers

Note: A set of outputs tied together is called a “bus.”
7-Segment LED and Decoder:

Designing the decoder:
- The ‘g’ LED turns on for digits 2, 3, 4, 5, 6, 8, and 9
- BCD does not allow 1010 through 1111 => These are “don’t cares.”

K-Map

```
<table>
<thead>
<tr>
<th>A_3 A_2</th>
<th>A_1 A_0</th>
<th>00</th>
<th>01</th>
<th>11</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>01</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>1</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>
```

\[ g = A_3 + A_2 A_1 + A_2 A_1 + A_1 A_0 \]

- We can do the same thing for each of the other 6 outputs.

For useful examples, see figures from Lam, O’Malley, and Arroyo posted online.