Experiment 7
Multiplexers and Demultiplexers

Eng. Mai Z. Alyazji

November, 2016
Objectives:
To be familiar with Multiplexers and Demultiplexers
To know how to implement functions using Multiplexers.

Background:

Multiplexers
A multiplexer is a combinational circuit that selects binary information from one of many input lines and directs it to a single output line. The selection of a particular input line is controlled by a set of selection lines. Normally, there are $2^n$ input lines and $n$ selection lines whose bit combinations determine which input is selected.

A two-to-one-line multiplexer connects one of two 1-bit sources to a common destination, as shown in Figure 1. The circuit has two data input lines, one output line, and one selection line $S$. When $S = 0$, the upper AND gate is enabled and $I_0$ has a path to the output. When $S = 1$, the lower AND gate is enabled and $I_1$ has a path to the output. The multiplexer acts like an electronic switch that selects one of two sources.

![Figure 1](image_url)

Figure 1 2x1 Multiplexer

$$F = S' I_0 + S I_1$$
Figure 2 shows the Multiplexer in Logisim. Note that it has a bit acts as enable, when it is 0 the output is invalid, try it!

**Implementing functions with Multiplexer**

We will now show a method for implementing a Boolean function of \( n \) variables with a multiplexer that has \( n - 1 \) selection inputs. The first \( n - 1 \) variables of the function are connected to the selection inputs of the multiplexer. The remaining single variable of the function is used for the data inputs. If the single variable is denoted by \( z \), each data input of the multiplexer will be \( z, z', 1, \) or \( 0 \). To demonstrate this procedure, consider the Boolean function

\[
F(x, y, z) = \sum (1, 2, 6, 7)
\]
This function of three variables can be implemented with a four-to-one-line multiplexer. The two variables x and y are applied to the selection lines in that order; x is connected to the S₁ input and y to the S₀ input. The values for the data input lines are determined from the truth table of the function as shown in Figure 3. When xy = 00, output F is equal to z because F = 0 when z = 0 and F = 1 when z = 1. This requires that variable z be applied to data input 0. The operation of the multiplexer is such that when xy = 00, data input 0 has a path to the output, and that makes F equal to z. In a similar fashion, we can determine the required input to data lines 1, 2, and 3 from the value of F when xy = 01, 10, and 11, respectively. This particular example shows all four possibilities that can be obtained for the data inputs.

As a second example, consider the implementation of the Boolean function

\[ F(A, B, C, D) = \sum (1, 3, 4, 11, 12, 13, 14, 15) \]

This function is implemented with a multiplexer with three selection inputs as shown in Figure 4. Note that the first variable A must be connected to selection input S₂ so that A, B, and C correspond to selection inputs S₂, S₁, and
$S_0$, respectively. The values for the data inputs are determined from the truth table listed in the figure. The corresponding data line number is determined from the binary combination of ABC. For example, the table shows that when $ABC = 101$, $F = D$, so the input variable $D$ is applied to data input 5. The binary constants 0 and 1 correspond to two fixed signal values. When integrated circuits are used, logic 0 corresponds to signal ground and logic 1 is equivalent to the power signal, depending on the technology (e.g., 3 V).

<table>
<thead>
<tr>
<th>$A$</th>
<th>$B$</th>
<th>$C$</th>
<th>$D$</th>
<th>$F$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Figure 4 Implementing a four-input function with a multiplexer
Constructing 4-1 Multiplexers out of 2-1 Multiplexers

We can construct a 4-1 MUX by using three 2-1 MUXes. We connect the least significant bit of the Selection Bits to the first level MUXes, and connect the most significant bit of the Selection Bits to the second level MUX, See Figure 5.

![Figure 5 Constructing 4-1 Multiplexers out of 2-1 Multiplexers](image)

Demultiplexers

The demultiplexer is a circuit that receives information from a single Line and directs it to one of $2^n$ possible output lines, the selection of a specific output is controlled by the bit combination of n selection lines. Figure 6 shows demultiplexers in Logisim.

![Figure 6 Demultiplexer](image)
Lab work using circuit maker program:

Equipment’s required:

IC’s 74LS157 (Quad 2-line to 1-line data multiplexers), 74LS151 (8-input multiplexer), 74LS04 (Hexa NOT).
The Datasheets of the IC’s.

Part I: Constructing 4-1 Multiplexers out of 2-1 Multiplexers

Use two 74LS157 IC to construct a 4-1 Multiplexer.
Part II:

Implementing functions using Multiplexers

Using 74LS151 IC implement the following function
\[ F(A,B,C,D) = \sum(0,2,4,5,8,10,11,15) \]

Exercises:
Using Logisim

- Build a 8 input multiplexer using two 4 input multiplexer and one 2 input multiplexer.
- Implement the following function using multiplexers:
  \[ F(x,y,z) = \prod(0,1,5,7) \]
  \[ F(A,B,C,D) = \sum(3,4,6,8,10,12,13,14) \]