ECE 301 Digital Electronics

Multiplexers

Revised Fall 2013

K. Hintz & C. Lorie

OBJECTIVE

To understand the operation of an 8x1 multiplexer (MUX) and design and verify the correct operation of a 4x2 MUX.

PREPARATION

Design and construct the circuits.

PROCEDURE

1. With reference to the datasheet for a 74LS151, verify that the eight-input MUX behaves in accordance with the truth table that is included on the first page of the datasheet.

2. A two by four (2x4) input multiplexer has two sets of inputs labeled *A* and *B*, each of which is 4-bits wide. There is one 4-bit wide output.
   - When the selector input, $A/\overline{B}$, pronounced "A B-bar" is set to a logic *1*, then the four *A* inputs, A_1 through A_4 are routed to the four output bits, O_1 through O_4.
   - When the selector input, $A/\overline{B}$ is set to a logic *0*, then the four *B* inputs, B_1 through B_4 are routed to the four output bits, O_1 through O_4.

note: The identifier for the selector input, $A/\overline{B}$, is a standard terminology which incorporates the polarity of the activating signal in the identifier. That is, when the selector is asserted as a logic *1*, the *A* input is passed to the output. In some cases it may be desirable to use what is called negative logic where the *A* input is asserted as a zero, that is a zero on the selector input would cause the *A* inputs to pass to the output. In this case, the proper identifier for the selector input would be $\overline{A}/B$. This type of identifier avoids any misinterpretation about the polarity of the actuating signal.

   a. Design a 4x2 MUX by specifying its truth table.
   b. Design a circuit to implement the truth table of the 4x2 MUX.
   c. Build and verify the correct operation of the MUX by comparing the truth table to the behavior of the circuit.
      - Apply all ones to the *A* data inputs.
      - Apply all zeros to the *B* data inputs.
      - Apply ones to the even numbered *A* data inputs, and zeros to the odd numbered *A* data inputs. Simultaneously apply ones to the odd numbered *B* data inputs, and zeros to the even numbered *B* data inputs. Verify that each output bit toggles when the $A/\overline{B}$ selector input is toggled (changed from one value to another)