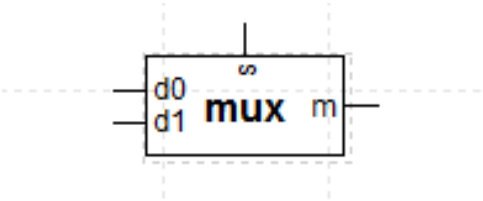
ENTITY DEFINITIONS AS COMPONENTS

In this Lab you will learn how to use DesignWorks to take a schematic and implement its entity definition as a "package" that can then be placed in a parts library. It then can be used as a component in the construction of other circuits.

Consider the following behavioural description:

ENTITY DEFINITION:



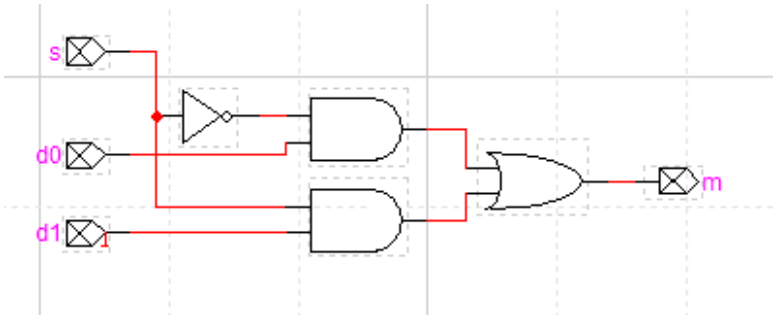
FUNCTIONAL SPECIFICATION:

$$m = d0 \cdot \overline{s} + d1 \cdot s$$

This device is an example of a simple digital switch, called a "2×1 multiplexer" because `m = d0` or `m = d` depending on the value of `s`.

In the following steps, you will construct a package representing this entity and use it to construct some simple circuits

1. If you have not already done so, launch DesignWorks and open a new "Generic Simulation" window.
2. Create a new library called "mylib.clf" in the Parts Palette as follows:
   (a) Place the mouse cursor anywhere in the Parts Palette (small window at the top right of the screen) and right-click the mouse.
   (b) Select "New Lib" from the resulting menu.
   (c) In the "Save in" textbox of the "Save Library As" window, select the directory path where the new library will be stored. This should be in your home directory (U:).
   (d) Type the library name to be used ("mylib.clf") in the File Name area of the "Save Library As" window, and click on "Save".
   (e) Check in the list of libraries at the top of the Parts Palette to confirm that your library, called "mylib.clf" is listed among the libraries available.
3. Build the following circuit using an OR-2, two AND-2's, and a NOT gate:



To define the "port connectors" (the pentagonal symbols on the input and output signal lines) proceed as follows:

(a) From the "Pseudo Devices.clf" library select the "Port In" component and connect one to each of the input signal lines of your circuit.

(b) From the same library select the "Port Out" component and connect one to each of the output signal lines of your circuit.

You may need to rotate a port before connecting it. To do this, place your mouse cursor over the port and right click the mouse. From the menu list select "Flip Horizontal", "Rotate Left" or "Rotate Right".

(c) The input and output ports must now be associated with the corresponding ports of the entity as shown above. To do this, you must first name each port:

i. Select the "pencil icon" from the toolbar.

ii. Place the mouse cursor over each port and click once. An empty text field will appear. In the field provided, enter the name shown in the entity definition.

4. Do not include any input-output devices unless you wish to test your circuit first. In that case, delete the input/output components afterwards before proceeding.

5. When you have named all the ports, save the resulting schematic in a file named "mux.cct".

6. To create a package corresponding to the entity whose behaviour is defined by this schematic:

(a) From the "File" pulldown menu select "New" and in the Design Template window that opens, select the tab labelled "Other Actions."

(b) From the "Other Actions" menu, select "Create a Simulation Model using the Model Wizard."

(c) Click on the "Browse" button that is displayed in the Simulation Model Wizard window and find the file "mux.cct" that you just saved above. Select it, and click on "Open." You will return to the Simulation Model Wizard window.

(d) Click on "Next" to open the "Model Info" window. Select "Internal Structural Circuit" as the model type.

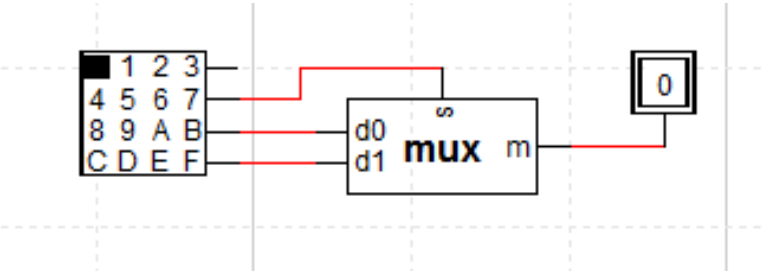
(e) Click on "Next" to open a window that will define pin locations. Four list boxes will be displayed with the names of the input ports from your schematic (`d0, d1, s`) in the left list box and the name of your output port `m` in the right list box. The top and bottom list boxes will be empty.

(f) Each list box represents a side of the entity definition rectangle. You can move ports to a different side of the entity by dragging them to a different list box.

(g) The position of the labels in the list boxes reflects the order of the ports on the entity. You can reorder the ports by dragging them to different positions within the list box. Arrange the labels of the input pins in the left box so that "`d0`" is above "`d1`".

(h) Click "Next." At this point your entity can be stored as a package in a library. Click on "Open Library" and select "mylib.clf." Then click on "Finish."

7. Select the library "mylib.clf" from the Parts Palette window. You should observe a list containing one part, called "mux." This is the part you have just created. In a new design window, select the mux component from "mylib.clf" and place it in the centre of the design window. Test your component by attaching a "hex keyboard (wo/STB)" and a binary probe tfrom the "Simulation IO.clf" library and connect them ias shown in the following circuit:



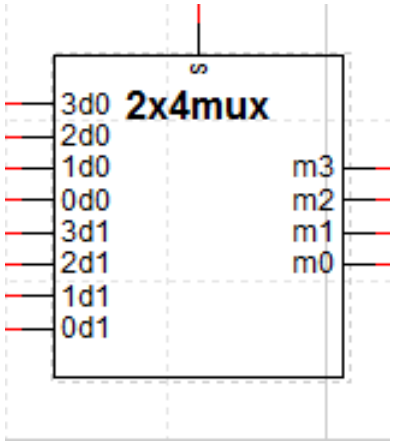
By successively clicking on the values 0 through 7 of the hex keyboard, verify that the observed behavior is:

s	d0	d1	m
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

From this function table, verify that the function select table is:

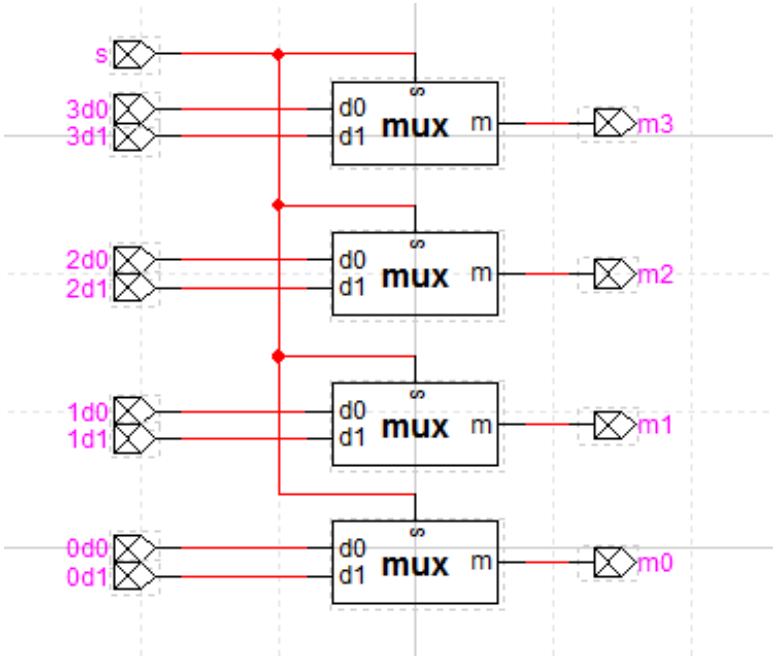
s	function
0	m = d0
1	m = d1

8. A 2 × 4 multiplexer is one that can select between two 4-bit input bus ports to deliver a value to the 4-bit output port `m`. The entity definition for this component is:

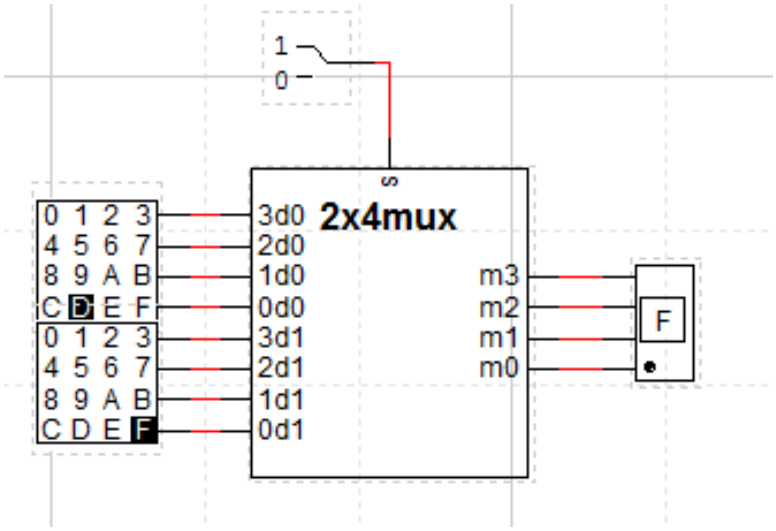


The signal ports making up the 4-bit data input ports are (`3d0, 2d0, 2d0, 0d0`) and (`3d1, 2d1, 1d1, 0d1`). The 4-bit data output port is (`m3, m2, m1, m0`). The functional specification is given by the same function select table as for the 2×1 MUX.

9. A 2 × 4 MUX can be constructed using four 2 ×1 MUXes. Using the 2 × 1 MUX component you just placed in your library, construct the following schematic in a new design window:



10. Package this circuit as a component called "`2x4mux`" following the same steps that were used for the 2x1 MUX and store it in your parts library. Then construct the following test circuit:



11. Using the test circuit, verify that output of the multiplexer is either the hexadecimal value from one hex keyboard or the other, depending on the value of the select input,"`s`.