6.16 Actel Corporation manufactures an FPGA family called Act 1, which has the multiplexer based logic block illustrated in Figure P6.1. Show how the function

$f = w_2\overline{w_3} + w_1w_3 + \overline{w_2}w_3$ can be implemented using only one Act 1 logic block.

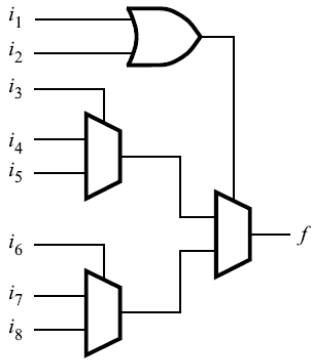
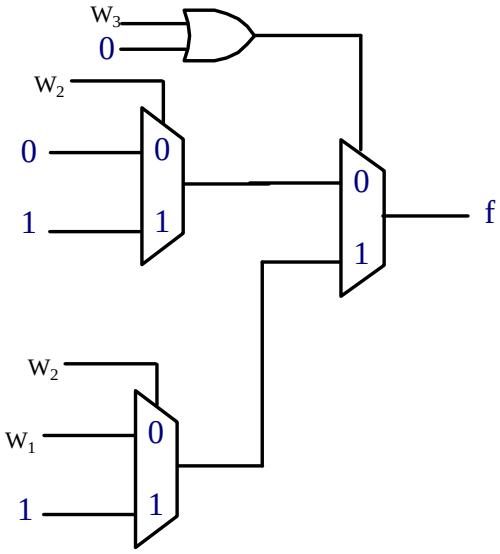


Figure P6.1 The Actel Act 1 logic block.

Solution:

$f = w_2\overline{w_3} + w_1w_3 + \overline{w_2}w_3$

$\Rightarrow f = \overline{w_3}(w_2) + w_3(w_1 + \overline{w_2})$



6.17 Show how the function $f = w_1\overline{w_3} + \overline{w_1}w_3 + w_2\overline{w_3} + w_1\overline{w_2}$ can be realized using Act 1 logic blocks. Note that there are no NOT gates in the chip; hence complements of signals have to be generated using the multiplexers in the logic block.

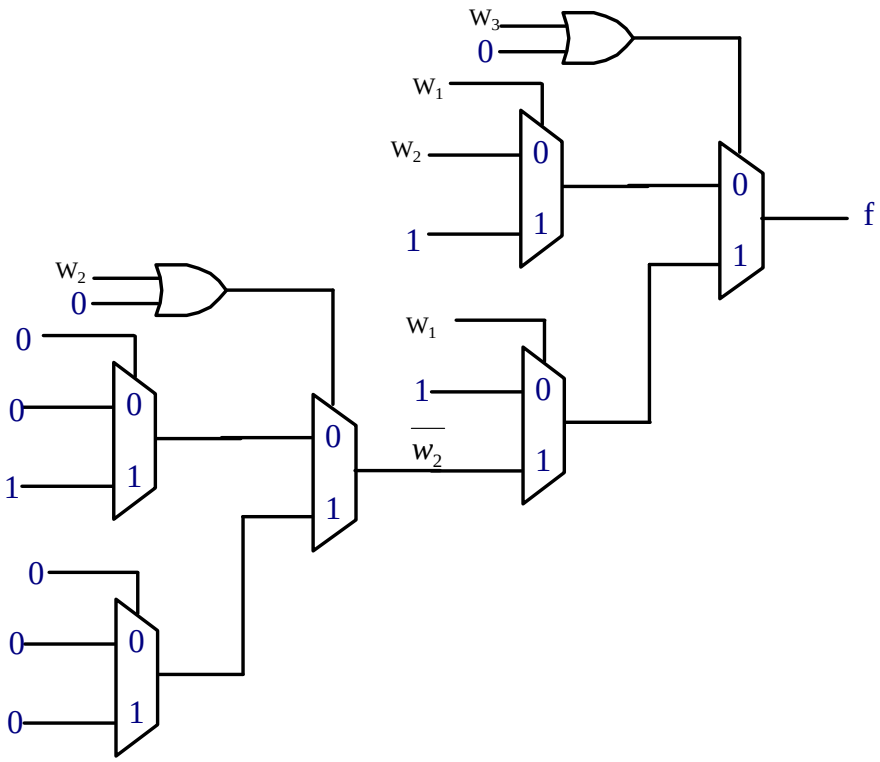
Solution:

Using Shannon expansion in term of w_3

$f = w_1\overline{w_3} + \overline{w_1}w_3 + w_2\overline{w_3} + w_1\overline{w_2}$

$\Rightarrow f = \overline{w_3}(w_1 + w_2) + w_3(\overline{w_1} + w_1\overline{w_2})$

$\Rightarrow f = \overline{w_3}(w_1 + \overline{w_1}w_2) + w_3(\overline{w_1} + w_1\overline{w_2})$



6.18 Consider the VHDL code in Figure P6.2. What type of circuit does the code represent? Comment on whether or not the style of code used is a good choice for the circuit that it represents.

Solution:

The code in Figure P6.2 is a 2-to-4 decoder with an enable input. It is not a good style for defining this decoder. The code is not easy to read. It is better to use the style in Figures 6.30 or 6.46.

6.21 Using a selected signal assignment, write VHDL code for a 4-to-2 binary encoder.

Solution:

```
ENTITY encoder_4to2 IS
   PORT ( w : IN STD_LOGIC_VECTOR(3 DOWNTO 0);
            y : OUT STD_LOGIC_VECTOR(1 DOWNTO 0));

END encoder_4to2;
ARCHITECTURE Behavior OF encoder_4to2 IS
BEGIN
   WITH w SELECT
   y <="00" WHEN "0001",
         "01" WHEN "0001",
         "10" WHEN "0001",
         "11" WHEN OTHERS;

END Behavior;
```

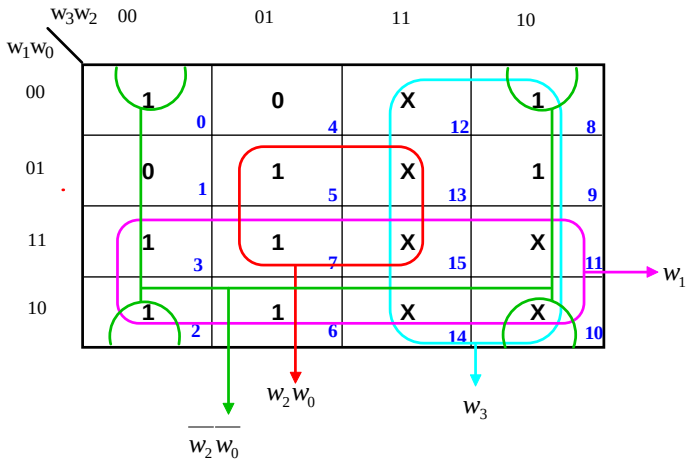
6.29 Derive minimal sum-of-products expressions for the outputs *a*, *b*, and *c* of the 7-segment display in Figure 6.25.

Solution:

The truth table of 7-segment display is shown below:

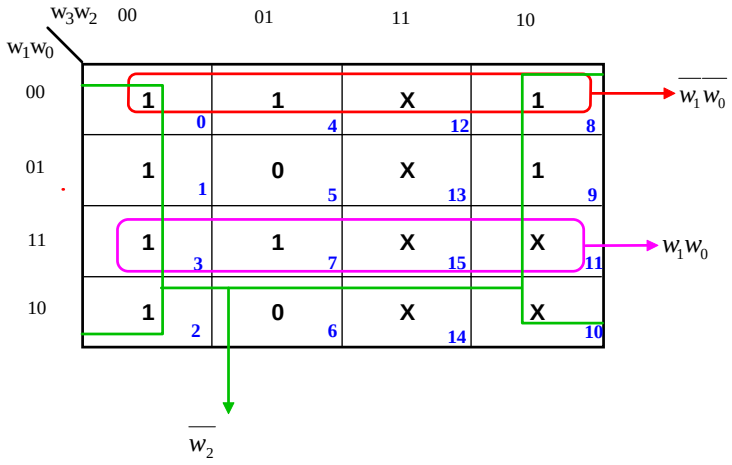
w_3	w_2	w_1	w_0	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	1	1	1	0	0	1
0	1	0	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	1	0	1	1
0	1	1	0	1	0	1	1	1	1	1
0	1	1	1	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	1	0	1	1

$$a(w_3,w_2,w_1,w_0)=\sum m(0,2,3,5,6,7,8,9)+d(10,11,12,13,14,15,16)$$



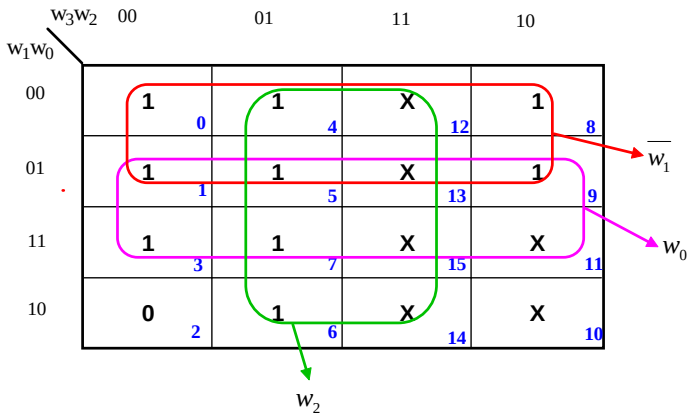
$$a = w_3 + w_1 + w_2w_0 + \overline{w_2}\,\overline{w_0}$$

$$b(w_3,w_2,w_1,w_0)=\sum m(0,1,2,3,4,7,8,9)+d(10,11,12,13,14,15,16)$$



$$b = \overline{w_2} + w_1w_0 + \overline{w_1}\,\overline{w_0}$$

$$c(w_3, w_2, w_1, w_0) = \sum m(0,1,3,4,5,6,7,8,9) + d(10,11,12,13,14,15,16)$$

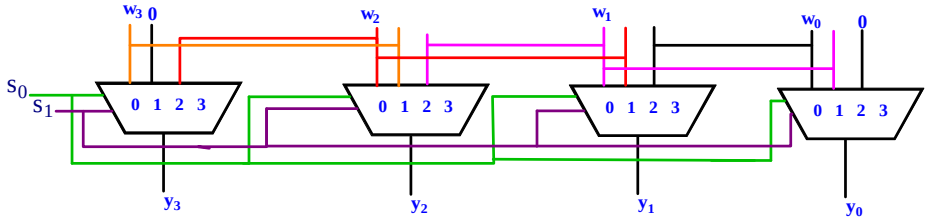


$$c = \overline{w_1} + w_0 + w_2$$

6.31 Design a shifter circuit, similar to the one in Figure 6.55, which can shift a four-bit input vector, $W = w_3w_2w_1w_0$, one bit-position to the right when the control signal *Right* is equal to 1, and one bit-position to the left when the control signal *Left* is equal to 1. When *Right* $=$ *Left* $=$ 0, the output of the circuit should be the same as the input vector. Assume that the condition *Right* $=$ *Left* $=$ 1 will never occur.

Solution: let, S_1 select= right shift and S_0 select=left shift

s_1	s_0	y_3	y_2	y_1	y_0
0	0	w_3	w_2	w_1	w_0
0	1	0	w_3	w_2	w_1
1	0	w_2	w_1	w_0	0
1	1	X	X	X	X



6.36 Figure 6.21 shows a block diagram of a ROM.Acircuit that implements a small ROM, with four rows and four columns, is depicted in Figure P6.3. Each X in the figure represents a switch that determines whether the ROM produces a 1 or 0 when that location is read.

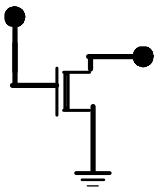
(a) Show how a switch (X) can be realized using a single NMOS transistor.

(b) Draw the complete 4×4 ROM circuit, using your switches from part (*a*). The ROM should be programmed to store the bits 0101 in row 0 (the top row), 1010 in row 1, 1100 in row 2, and 0011 in row 3 (the bottom row).

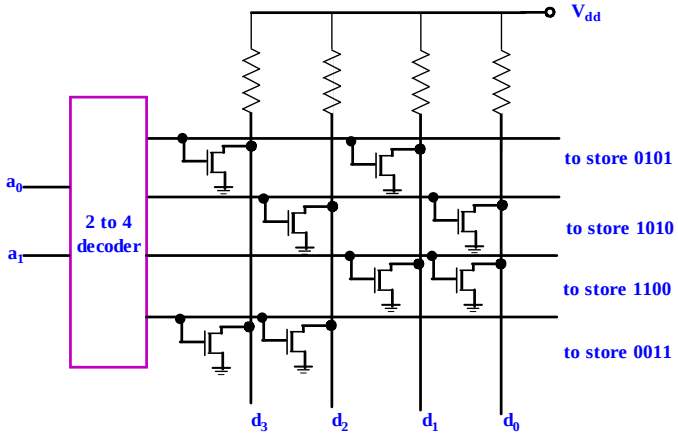
(c) Show how each (X) can be implemented as a programmable switch (as opposed to providing either a 1 or 0 permanently), using an EEPROM cell as shown in Figure 3.64. Briefly describe how the storage cell is used.

Solution:

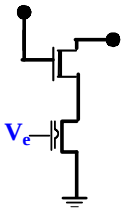
(a) Each ROM location that should store a 1 requires no circuitry, because the pull-up resistor provides the default value of 1. Each location that store a 0 has the following cell



(b)



(c) Every location of the ROM contains the following cell



If a location should store a 1, the corresponding EEPROM transistor is programmed to be turned off. But if the location should store 0, the EEPROM transistor is left unprogrammed.

6.37 Show the complete circuit for a ROM using the storage cells designed in Part (*a*) of problem 6.36 that realizes the logic functions

$$d_3 = \overline{a_0 \oplus a_1}$$

$$d_2 = a_0 \oplus a_1$$

$$d_1 = a_0 a_1$$

$$d_0 = a_0 + a_1$$

Solution:

a_1	a_0	d_3	d_2	d_1	d_0
0	0	0	1	0	0
0	1	1	0	0	1
1	0	1	0	0	1
1	1	0	1	1	1