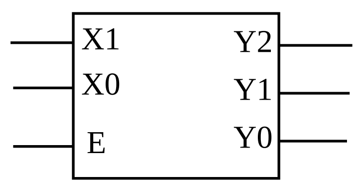
ECE 204 - COMBINATIONAL BUILDING BLOCKS - INVEST 13
DECODERS AND ENCODERS

FALL 2003 **A.P. FELZER**

To do "well" on this investigation you must not only get the right answers but must also do neat, complete and concise writeups that make obvious what each problem is, how you're solving the problem and what your answer is. You also need to include drawings of all circuits as well as appropriate graphs and tables.

In the last Investigation we looked at some of the basic properties of the inputs and outputs of logic circuits. The objective of this and the next several Investigations is to look at some basic MSI (Medium Scale Integration) logic circuits made from 20 to 100 gates that are very useful in many applications. In this investigation we look at encoders and decoders for translating data from one code to another like from signed magnitude to 2's complement.

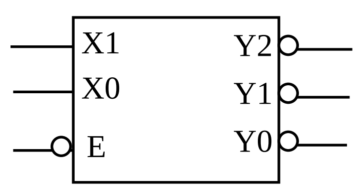
1. We begin with a review problem. Given the following logic circuit



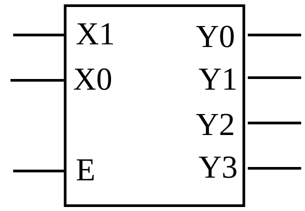
with truth table as follows

E	**X1**	**X0**	**Y2**	**Y1**	**Y0**
0	x	x	0	0	0
1	0	0	0	1	0
1	0	1	1	0	1
1	1	0	0	1	0
1	1	1	1	0	1

Find the truth table when the enable and outputs are active low as follows



2. **Decoders** and **encoders** are very important logic circuits that convert data from one code to another. Binary decoders in particular are logic circuits like the following 2-to-4 decoder



with n=2 inputs and $N=2^n = 2^2=4$ outputs and the following Truth Table

E	X1	X0	Y0	Y1	Y2	Y3
0	x	x	0	0	0	0
1	0	0	1	0	0	0
1	0	1	0	1	0	0
1	1	0	0	0	1	0
1	1	1	0	0	0	1

Describe what this decoder is doing. **Memorize** this relationship.

3. One possible application of a 2-to-4 decoder like the one in Problem (2) is a traffic light controller with Y0 ... Y3 controlling the lights on Main Street as follows

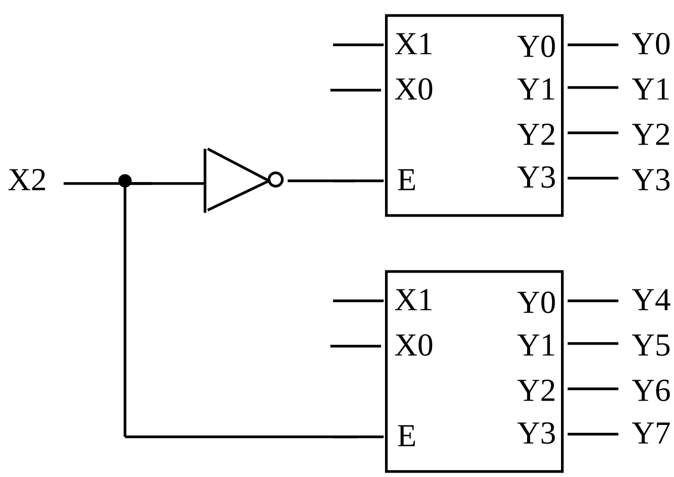
   Y0 - Green Light and WALK signal
   Y1 - DON'T WALK signal
   Y2 - Yellow Light
   Y3 - Red Light

   as the input X_1X_0 goes from 00 to 11. Come up with your own application of a decoder that is different from those of your friends.

4. Draw a realization of the 2-to-4 decoder in Problem (2) from AND gates and INVERTERS. You can use gates with more than two inputs.

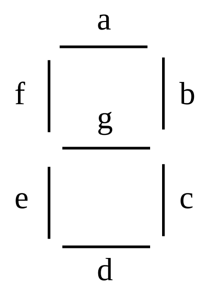
5. Suppose the 2-to-4 decoder in Problem (2) is changed so that now the enable and the outputs are active low
   a. Draw the chip diagram
   b. Draw the logic diagram to realize the circuit
   c. Write out the truth table

6. One of the really useful features of MSI circuits like decoders is that they can be put together to form larger decoders. Given the following 3-to-8 decoder made from two 2-to-4 decoders



   a. Write the truth table
   b. Describe what's going on in this logic circuit

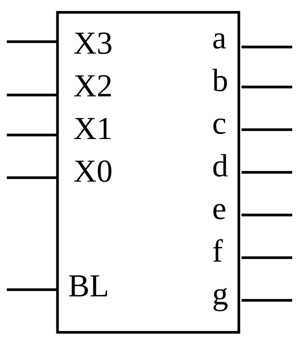
7. The objective of this problem is to introduce seven-segment displays. If we look closely at digital watches and clocks we see seven-segment displays as follows

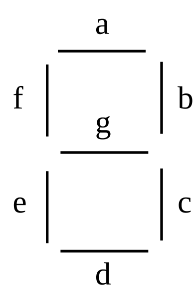


with segments (lines) that light up to form the different numbers. Which of the segments light up to form the numbers

a. 3
b. 5

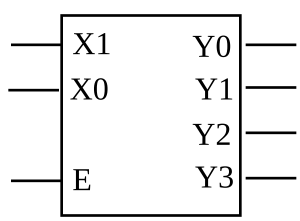
8. Up to now we've only been working with binary decoders that have one output asserted at a time. But more general decoders like the following BCD to seven-segment decoder



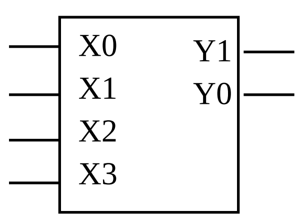


typically have many outputs asserted at a time. Verify this fact by writing out the truth table for the seven-segment decoder above that tells the display which segments to turn ON and which to keep OFF for each BCD number from 0 to 9. Note that when the input BL=0 the display is *blanked* - all segments are OFF. And when BL=1 the BCD input is decoded. Also note that there are many other useful decoders like BCD-to-Binary and Binary-to-BCD.

9. As we've seen above the output of a binary decoder like the following



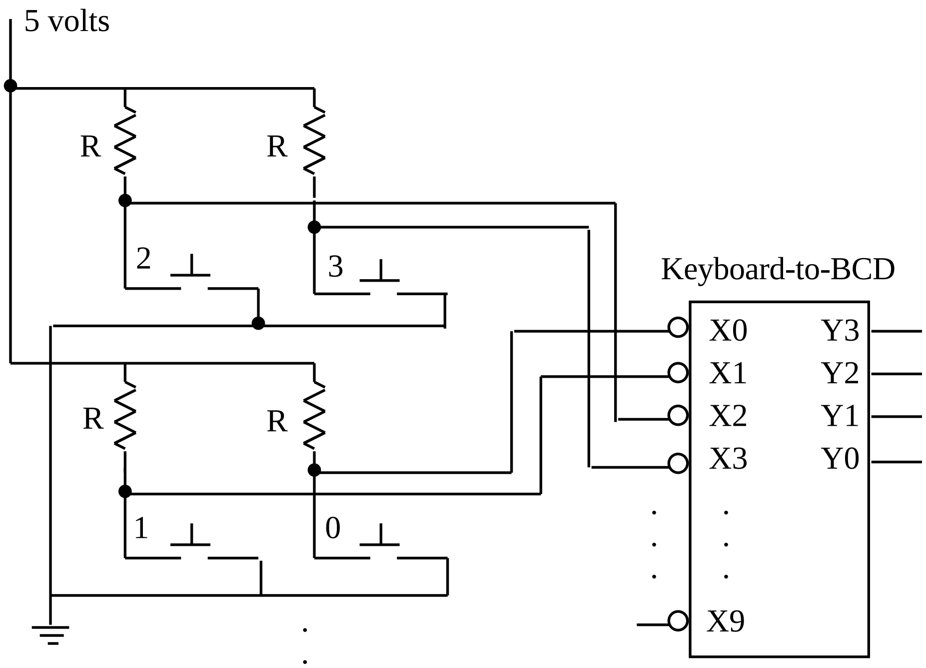
tells us the value of the binary number at the input. The corresponding binary **encoder** as follows



does just the opposite. It tells us the binary code of the selected input.

a. Write the Truth Table of the binary encoder if only one input can be selected at a time
b. Realize your encoder with OR gates

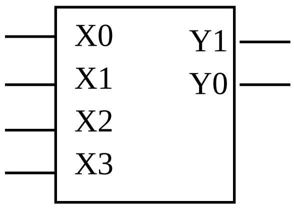
10. Explain the operation of the the following circuit



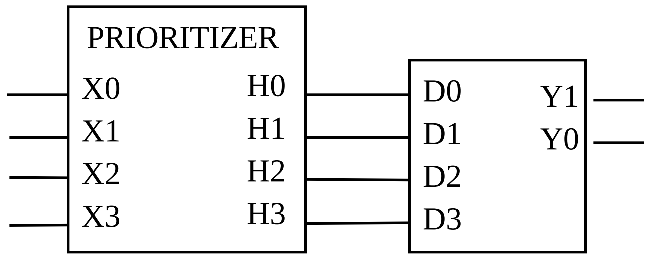
consisting of a calculator keyboard connected to a keyboard-to-BCD encoder.

11. In the previous two problems we've been assuming that only one input would be selected at a time. But there are many situations in real applications where more than one input is "competing for attention" like when several users are trying to access a shared printer at the same time. One way to handle this situation is with **priority encoders**. When more than one input is selected the priority encoder encodes the one with the *highest priority*.

Let's suppose the following is a priority encoder



with X0 having the highest priority and X3 the lowest. The trick in designing such a circuit is to first design a "prioritizer" and then connect it to a "regular" encoder as follows



with truth table as follows

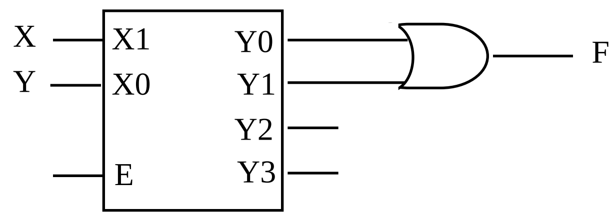
X0	X1	X2	X3	H0	H1	H2	H3
1	x	x	x	1	0	0	0
0	1	x	x	0	1	0	0
	⋮				⋮		

a How can you tell from the truth table that X0 has the highest priority
b. How can you tell from the truth table that X3 has the lowest priority
c. Design the prioritizer
d. Describe in words the overall operation of the circuit

12. The objective of this problem is to show how decoders can be used to realize general logic equations. Suppose for example that we want to realize the following logic equation

$$F = X' \cdot Y' + X \cdot Y'$$

a. Write the Truth Table for F
b. Make use of your truth table in part (a) to explain why F can be realized with a 2-to-4 decoder as follows



c. Use a 2-to-4 decoder with active high inputs and outputs to realize

$$F = X \cdot Y' + X' \cdot Y$$

d. Repeat part (c) if the input of the decoder is active high and the output active low. **Memorize** the results of this problem