ECE 409 - ERROR CORRECTING CODES - INVESTIGATION 21
INTRODUCTION TO BLOCK CODES - PART I

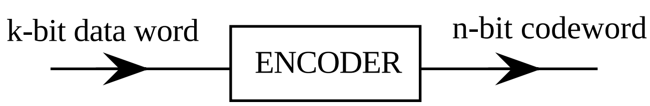
SUMMER 2004 **A.P. FELZER**

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.

From our results in ECE 405 we know that despite waveshaping to reduce intersymbol interference, despite matching filters to minimize the affects of noise and despite equalizers to minimize the affects of channel induced phase shifts and attenuation there are always going to be errors. The objective of error correcting binary codes is to add extra bits to codes in order to help identify and correct errors in received signals. There are two kinds of error correcting codes - block codes and convolution codes. The objective of this and the next Investigations is to introduce block codes.

Note that general coding theory also includes the topics of compression and encryption. The goal of compression is to eliminate "redundant" code words and reduce wordlengths like we did in companding in ECE 405. The goal of encryption is to make messages unintelligible to all but the intended recipient.

1. In **block codes** of size (n, k) transmitted signals are grouped together in blocks of size n consisting of k data bits and n-k bits added to detect errors in the received signals as follows

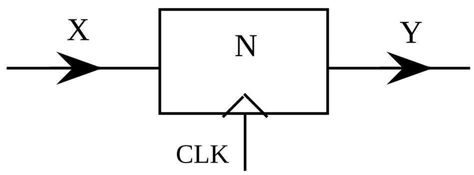


   We call k/n the **rate** of the code. How many valid codewords - codewords without any errors - are there in a block code of size (n,k)

2. A very simple block code is obtained by adding a **parity bit** - an extra bit added to the data bits to make the total number of ones even (or odd). Add a ninth bit to each of the following words to make the total number of ones even. We call this **even parity**
   a. A=10011000
   b. B=11001010

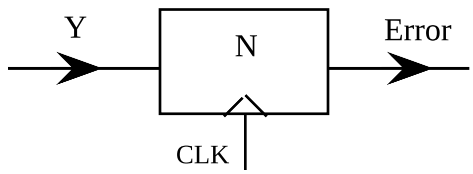
3. Suppose a transmitted code word has even parity
   a. How will a single error affect the received signal
   b. How will two or more errors affect the received signal

4. Design a serial even parity code word generator as follows



   that adds a parity bit to every 8 bits of data. Hint - use an exclusive-or gate.

5. Design a serial code word error detector as follows



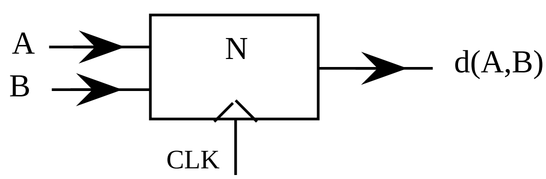
for 9 bit code words with even parity

6. As we've seen in the last series of problems adding a parity bit to generate code words with even parity enables us to detect if there's been an odd number of errors because code words with an odd number of ones are not valid. In order to generalize on this result so we can detect more than one error we define the **Hamming distance d(A,B)** between two words like the following

$$A = 10110 \quad \text{and} \quad B = 00101 \quad \text{with} \quad d(A,B) = 3$$

to be the total number of bits where A and B are different. **Memorize** the definition of Hamming distance. Then find the Hamming distances between
   a. A=10101010 and B=01100011
   b. C=01011001 and D=01101011

7. Design a serial circuit N as follows



to find the Hamming distance d(A,B) between two 8-bit code words A and B

8. We define the **Hamming distance of a code** as the minimum possible distance $d_{\min}$ between any two valid codewords in the code. **Memorize** this definition. Explain why adding a single parity bit gives us a code with Hamming distance equal to two.

9. Suppose the Hamming distance of a particular code is $d_{\min} = 2$ and that in particular the codewords $A = 10110$ and $B = 10000$ are a distance two apart. Then we can get from A to B by changing one bit at a time as follows

10110 ●
10010 ○
10000 ●

where we have used solid dots to indicate valid codewords. Generalizing on this notation we draw

● ○ ●

to represent any code with a Hamming distance equal to two. Given the following codes

Code 1: ● ●
Code 2: ● ○ ○ ○ ●

   a. What's the Hamming distance between codewords in Code 1

b. What's the Hamming distance between codewords in Code 2
c. Draw a diagram like those above for a code with Hamming distance 5

10. Explain why we can detect single errors with codes having Hamming distances equal to two

11. Explain why we can detect up to two errors in codes with Hamming distances equal to three

12. Generalize on the result of Problem (11) for codes with Hamming distance d

13. Suppose we receive a codeword c in a code with a Hamming distance d. Then find and explain the value of P in the following expression

    P = Prob[c has no errors] + Prob[c has a detectable error] + Prob[c has an undetectable error]

14. Find out about and explain how the (7, 4) Hamming code works. Do some examples to illustrate