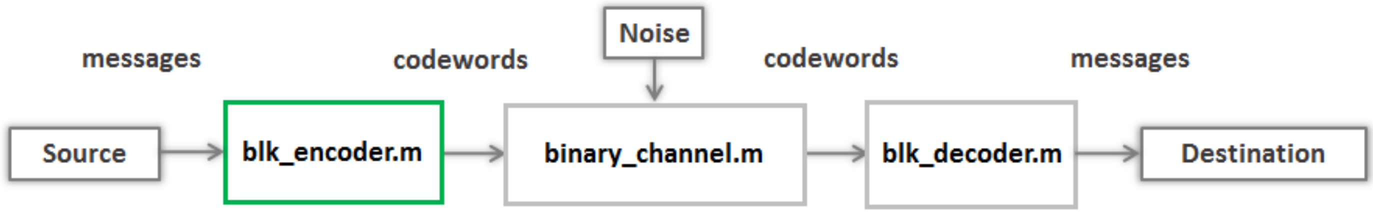
Courseware Course Info Course Outline Grading Scheme Instructors Resources Discussion Progress

LAB 10 - TASK 1 (8,4,3) BLOCK ENCODER (1/1 point)

In this task, you will write code implementing the MATLAB function, blk_encoder.m, which takes a four bit message and produces an eight bit codeword using the (8,4.3) code discussed in lecture.



```
messages = gen_messages(4); % cell array of all four bit messages
codewords = cell(16,1);     % cell array to hold codewords

for i=1:16, % loop over all messages
    msg = messages{i}; % take the i'th message
    %disp(msg);
    % Modify the code below to return the 8 bit codeword corresponding to
    % the 4 bit message in msg by
    %  1. calculating the 4 parity bits
    %  2. concatenating the parity bits to the message bits
    p1 = xor(msg(1),msg(2));
    p2 = xor(msg(3),msg(4));
    p3 = xor(msg(1),msg(3));
    p4 = xor(msg(2),msg(4));
    codeword = [msg p1 p2 p3 p4];
```

Correct

```
parity = xor(msg([1 3 1 2]),msg([2 4 3 4]));
codeword = [msg parity];
```

Congratulations! All codewords are correct.

Check Reset Save Hide Answer *You have used 1 of 10 submissions*

INSTRUCTIONS

Step 1: Run the code as presented

The code above creates a 16x1 cell array called **messages**, which contains all 16 possible four bit messages, e.g. [0 0 0 0], [0 0 0 1], [0 0 1 0], etc. A cell array is similar to a regular MATLAB array or vector, except that each element is called a "cell" which you can think of as being a container to hold an arbitrary data element. Here, each cell contains one 1x4 element array of 1's and/or 0's. The code also initializes a 16x1 cell array called **codewords** to store the corresponding codewords.

The **for** loop cycles through each of the possible messages. It first extracts each message stored in the cell array **messages** as a 1x4 vector of binary digits called **msg**. Note that we use braces {} to select individual cells from a cell array, rather than parentheses (). The next line creates an eight bit vector **codeword** from **msg** by appending four extra bits. Here the extra bits are all zeros. In the next step, it is your job to replace this line with your own code that appends the four parity bits of the (8,4,3) code discussed in lecture to create a valid codeword. The final line stores the vector **codeword** in the cell array **codewords** for later checking.

After the for loop has created and stored codewords for each of the 16 messages, it passes the cell arrays **messages** and **codewords** to the function **check_blk_encoding**, which checks to see whether each codeword was correctly computed from the corresponding message.

Click on the **Run Code** button. MATLAB will return a message indicating that one of the codewords is incorrect. Even though there are many incorrect codewords, the function check_blk_encoder only indicates the first one it encounters.

Step 2: Implement the (8,4,3) encoder

To complete this task, you should replace the line under the comments starting with

```
% Modify the code below
```

with your code to append the four parity bits of the (8,4,3) code discussed in the lecture videos.

Hints:

1. If you have two bits **a** and **b**, one way of determining the parity of [**a b**] is to use the MATLAB exclusive-or function **xor(a,b)**. Note that if a and b are vectors with the same size, then **xor** returns the a vector of the same size, where each element is the **xor** of the corresponding elements in **a** and **b**.

2. Another possible way to determe the parity of a set of bits is to sum them together and determine whether the sum is even or odd. One way to determine whether a number is even or odd is to take the remainder or modulus after division by two. If you choose to use this technique, the MATLAB functions **sum**, **rem** and/or **mod** may be useful.

Step 2: Submit your work

If you have correctly computed all the codewords, then running the MATLAB code should return a message indicating all codewords are correct.

Once you have completed your work, click on the **Check** button to submit your answer.

Help

EdX offers interactive online classes and MOOCs from the world's best universities. Online courses from MITx, HarvardX, BerkeleyX, UTx and many other universities. Topics include biology, business, chemistry, computer science, economics, finance, electronics, engineering, food and nutrition, history, humanities, law, literature, math, medicine, music, philosophy, physics, science, statistics and more. EdX is a non-profit online initiative created by founding partners Harvard and MIT.

Terms of Service and Honor Code

Privacy Policy (Revised 4/16/2014)

About & Company Info

About

News

Contact

FAQ

edX Blog

Donate to edX

Jobs at edX

Follow Us

Twitter

Facebook

Meetup

LinkedIn

Google+