

Part 1

Lab 4

$(u-v)(v-w)(w-u)$

000

→

$V, V,$
00

001

→

11

010

→

11

011

→

00

100

→

10

101

→

01

110

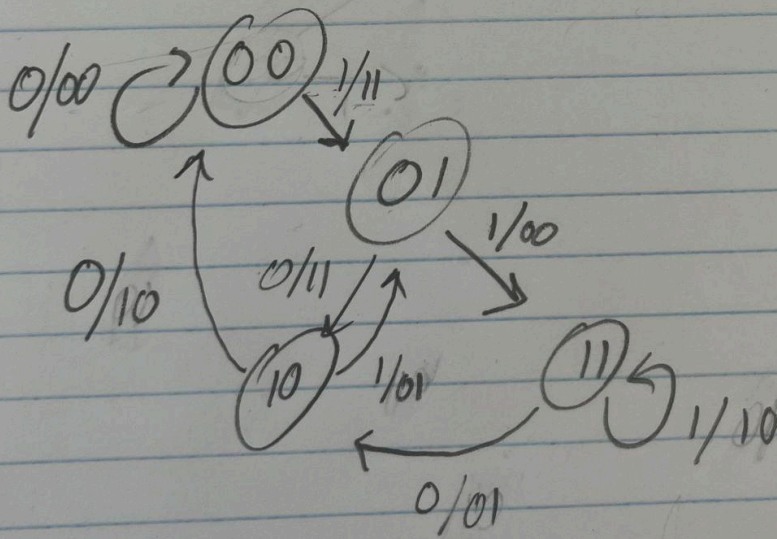
→

01

111

→

10



```
Editor - \\samba2.engr.scu.edu\jlanders\ECC\Desktop\ELEN 142\Lab4\ConvEncode.m
ConvEncode.m x Lab4.mlx x +
1 function [encoded] = ConvEncode(msgBits)
2
3     currentstate = 0;
4     output = [];
5
6     for i = 1:length(msgBits)
7
8         input = msgBits(i);
9
10        switch currentstate
11            case 0
12                if input == 1
13                    nextstate = 1;
14                    output = [output, 1, 1];
15                else
16                    nextstate = 0;
17                    output = [output, 0, 0];
18                end
19            case 1
20                if input == 1
21                    nextstate = 3;
22                    output = [output, 0, 0];
23                else
24                    nextstate = 2;
25                    output = [output, 1, 1];
26                end
27            case 2
28                if input == 1
29                    nextstate = 1;
30                    output = [output, 0, 1];
31                else
32                    nextstate = 0;
33                    output = [output, 1, 0];
34                end
35            case 3
36                if input == 1
37                    nextstate = 3;
38                    output = [output, 1, 0];
39                else
40                    nextstate = 2;
41                    output = [output, 0, 1];
42                end
43        end
44
45        currentstate = nextstate;
46        encoded = output;
47    end
```

Part 2

```

function [Distance] = HammingDist(msgBits, encBits)

    difference = xor(msgBits, encBits);
    Distance = 0;

    for i = 1:length(difference)
        if difference(i) == 1
            Distance = Distance + 1;
        end
    end
end

```

Live Editor - \\samba2.engr.scu.edu\jlanders\ECC\Desktop\ELEN 142\Lab4\Lab4.mlx

ConvEncode.m Lab4.mlx HammingDist.m

```

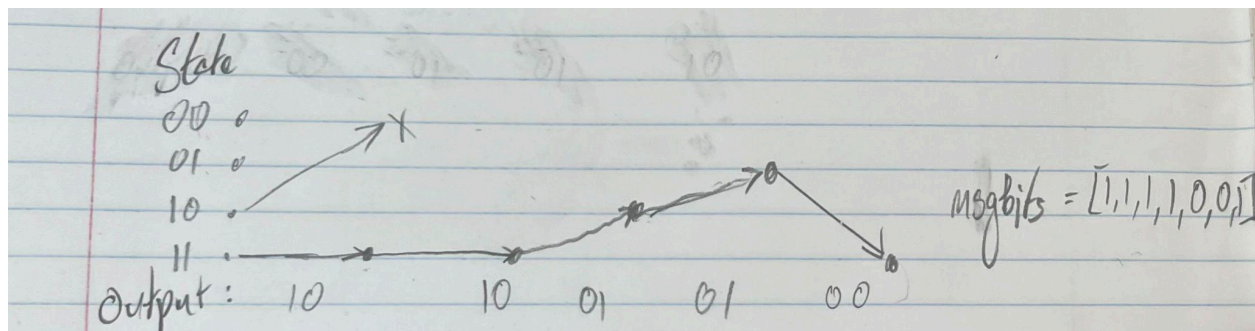
1 clear all;
2 msgBits = [0,1,0,1,1,1,0,0,1,0,1,0,0,0,1];
3 encoded = ConvEncode(msgBits);
4
5 msgBits = [0,1,0,1,1,1,0,0,1,0,1,0,0,0,1];
6 encBits = [0,1,0,0,0,0,0,0,1,0,1,0,0,0,1];
7 Distance = HammingDist(msgBits, encBits);
8
9 BER = Distance/length(msgBits);

```

Workspace

Name	Value
ans	1
BER	0.2000
decoded	1x15 double
Distance	3
encBits	1x15 double
encoded	1x30 double
msgBits	1x15 double
myTrellis	1x1 struct
next_states	[0,1,2,3;0,1,2,3]
state_outp...	[0,3;3,0;2,1;1,2]
worked	1

Part 3



```

myTrellis.numInputSymbols = 2;
myTrellis.numOutputSymbols = 4;
myTrellis.numStates = 4;

next_states = [0, 1;
               2, 3;
               0, 1;
               2, 3]

state_outputs = [0, 3;
                 3, 0;
                 2, 1;
                 1, 2]

myTrellis.nextStates = next_states;
myTrellis.outputs = state_outputs;
istrellis(myTrellis)

decoded = vitdec(encoded, myTrellis, 1, 'trunc', 'hard');
|
if decoded == msgBits
    worked = 1;
else
    worked = 0;
end

```

Part 4

1. What did you observe from the experiments?
2. If you could expand this experiment how would you do it?
3. How does the experiment relate to the course topic?
4. What did you learn from the experiment?