Nem Negash

CMPE 415

Prof. Tinoosh Mohsenin

April 18, 2021

HW6 Report

Problem 1:

a) 100 100 011 010

Binary Value	1	0	0	1	0	0	0	1	1	0	1	0
EEC Code	e_{12}	e_{11}	e_{10}	e_9	e_8	e_7	e_6	e_5	e_4	e_3	e_2	e_1

$e_1 = e_3 \wedge e_5 \wedge e_7 \wedge e_9 \wedge e_{11} = 0\wedge1\wedge0\wedge1\wedge0 = 0$
$e_2 = e_3 \wedge e_6 \wedge e_7 \wedge e_{10} \wedge e_{11} = 0\wedge0\wedge0\wedge0\wedge0 = 0$
$e_4 = e_5 \wedge e_6 \wedge e_7 \wedge e_{12} = 1\wedge0\wedge0\wedge1 = 0$
$e_8 = e_9 \wedge e_{10} \wedge e_{11} \wedge e_{12} = 1\wedge0\wedge0\wedge1 = 0$

Computed: 0000
Received: 0110

Error Bit Location = computed ^ received = 0000^0110 = 0110 = e6

Corrected Code: 100 100 **1**11 010

b) 000 110 111 000

Binary Value	0	0	0	1	1	0	1	1	1	0	0	0
EEC Code	e_{12}	e_{11}	e_{10}	e_9	e_8	e_7	e_6	e_5	e_4	e_3	e_2	e_1

$e_1 = e_3 \wedge e_5 \wedge e_7 \wedge e_9 \wedge e_{11} = 0\wedge1\wedge0\wedge1\wedge0 = 0$
$e_2 = e_3 \wedge e_6 \wedge e_7 \wedge e_{10} \wedge e_{11} = 0\wedge1\wedge0\wedge0\wedge0 = 1$
$e_4 = e_5 \wedge e_6 \wedge e_7 \wedge e_{12} = 1\wedge1\wedge0\wedge0 = 0$
$e_8 = e_9 \wedge e_{10} \wedge e_{11} \wedge e_{12} = 1\wedge0\wedge0\wedge0 = 1$

Computed: 1010
Received: 1100

Error Bit Location = computed ^ received = 1010 ^1100 = 0110 = e6

Corrected Code: 000 110 **0**11 000

c) 111 011 011 101

Binary Value	1	1	1	0	1	1	0	1	1	1	0	1
EEC Code	e_{12}	e_{11}	e_{10}	e_9	e_8	e_7	e_6	e_5	e_4	e_3	e_2	e_1

$e_1 = e_3 \wedge e_5 \wedge e_7 \wedge e_9 \wedge e_{11} = 1\wedge1\wedge1\wedge0\wedge1 = 0$
$e_2 = e_3 \wedge e_6 \wedge e_7 \wedge e_{10} \wedge e_{11} = 1\wedge0\wedge1\wedge1\wedge1 = 0$
$e_4 = e_5 \wedge e_6 \wedge e_7 \wedge e_{12} = 1\wedge0\wedge1\wedge1 = 1$
$e_8 = e_9 \wedge e_{10} \wedge e_{11} \wedge e_{12} = 0\wedge1\wedge1\wedge1= 1$

Computed: 1100
Received: 1101

Error Bit Location = computed ^ received = 1100 ^ 1101 = 0001 = e1

Corrected Code: 111 011 011 10**0**

Problem 2:
[5 pts] What is the length of total code word? Which bits are check bits and which one are data bits? Compute the ECC bits for 4-bit data 0110 and write the complete code word.

D_4	D_3	D_2	E_4	D_1	E_2	E_1

Bit_position	7	6	5	4	3	2	1
Bits	0	1	1	E_4	0	E_2	E_1

$e_1 = d_3 \wedge d_5 \wedge d_7 = 0\wedge1\wedge0 = 1$
$e_2 = d_3 \wedge d_6 \wedge d_7 = 0\wedge1\wedge0 = 1$
$e_4 = d_5 \wedge d_6 \wedge d_7 = 1\wedge1\wedge0 = 0$

Bit_position	7	6	5	4	3	2	1
Bits	0	1	1	0	0	1	1

Complete Code word = 0110011.

[20 pts] Write the verilog for the module that checks if there is an error in the received data. The verilog module has the received word as input and has two outputs error and error-bit. If there is any error, then the verilog module must locate the error bit and make the error signal to be 1 and send out the location of error-bit. Otherwise the error signal remains zero.

```
module decoder(

    input [6:0] rec_word,
    output [2:0] error_bit,
    output error_check
    );

    wire [2:0] computed;
    wire [2:0] received;

    //received
    assign received = {rec_word[3],rec_word[1],rec_word[0]};

     //computed
     assign computed[0] = rec_word[2] ^ rec_word[4] ^ rec_word[6];
     assign computed[1] = rec_word[2] ^ rec_word[5] ^ rec_word[6];
     assign computed[2] = rec_word[4] ^ rec_word[5] ^ rec_word[6];

    //get error bits
     assign error_bit = computed ^ received;

    //check if there is an error
     assign error_check = error_bit[0] | error_bit[1] | error_bit[2];

    endmodule
```

[15 pt] Write a testbench that can test the block with these input values. First write down what is the error bit location if there was an error.

- o 1111000
- o 0110111
- o 1000111
- o 0111010

```
module decoder_tb();

reg [6:0] rec_word;
wire error_check;
wire [2:0] error_bit;

decoder DUT(.rec_word(rec_word),.error_check(error_check),.error_bit(error_bit));

initial
    begin
        rec_word = 7'b1111000;

        #50
        rec_word = 7'b0110111;

        #50
        rec_word = 7'b1000111;

        #50
        rec_word = 7'b0111010;

    end
endmodule
```