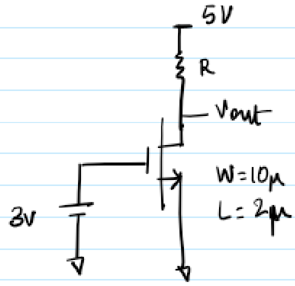
EE 330 Fall 2024
Homework 7 Solutions

Problem 1

part 1 and part 2



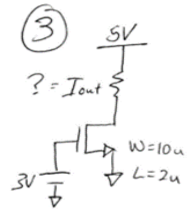
Assume nmos in saturation

$$I_D = \frac{\mu_0 C_{ox}}{2}\frac{W}{L}(V_{GS}-V_{TN})^2$$

$$I_D = \frac{100\mu A}{2} \times \frac{10\mu}{2\mu}(3V-0.75)^2 = \boxed{1.266\,mA}$$ 2

$$V_{out} = V_{DD} - I_D R = 5 - 1.266\times10^{-3}\times 1\times10^{3} = \boxed{3.744\,V}$$ 1

Part 3



Assume it's in saturation

$I_D = 1.27mA$

$V_{out} = 5V - (1.27mA \cdot 5k\Omega) = -1.35V$

$V_{out} \geq V_{GS} - V_{TN}$ ✗ fails to be saturation

but

$V_{GS} > V_{TN}$ ✓

Let's use linear mode equation.

$$I_D = \frac{100uA/v^2}{2}\left(\frac{10u}{2u}\right)(V_{GS}-V_{TN}-V_{out})V_{out}$$

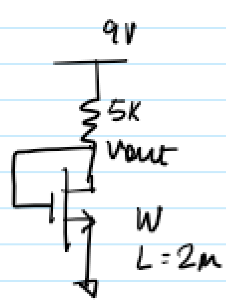
$$I_D = \frac{100uA/v^2}{2}\left(\frac{10u}{2u}\right)(3V-0.75V-V_{out})V_{out}$$

$$I_D = \frac{5V-V_{out}}{5k\Omega} \Rightarrow \frac{100uA/v^2}{2}\left(\frac{10u}{2u}\right)(3v-0.75-V_{out})V_{out} = \frac{5V-V_{out}}{5K}$$

$\Rightarrow V_{out} = 0.225$, $V_{out} = 2.825$ ← If we use this Nmos will be in saturation

$$I_{out} = \frac{V_{DD}-V_{out}}{5k\Omega} = \frac{5-2.825}{5k\Omega} = \boxed{0.435\,mA}$$

2.)



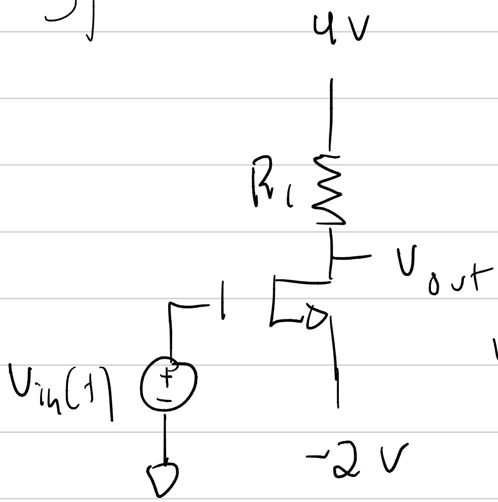
$V_{out} = V_{GS} = 3V$ (Assume saturation)

$$I_D = \frac{V_{DD}-V_{out}}{5K} = \frac{6}{5K} = 1.2\,mA$$

$$I_D = \frac{\mu_0 C_{ox}}{2}\frac{W}{L}(V_{GS}-V_T)^2 = 1.2\times10^{-3}$$

$$\Rightarrow W = \frac{1.2\times10^{-3}\times 2\times10^{-6}}{\frac{100\times10^{-6}}{2}\times(3-0.75)^2} = 9.48\,\mu m$$

3)



$W = 12\mu$, $L = 2\mu$

$V_{GS} = V_G - V_S = 0 - (-2V) = 2V$

$V_T = 0.75\ V$

$I_{DQ} = \frac{\mu C_{ox}}{2} \frac{W}{L} \times (V_{GS} - V_T)^2$

$= \frac{100\mu}{2} \frac{12}{2} \times (2 - 0.75)^2$

$= 468.75\ \mu A$

For saturation

$V_{DS} \geq V_{GS} - V_T$

$\to V_{DS} = V_D - V_{SS} = (4 - I_D R) - (-2) = 6 - I_D R$

$\to 6 - I_D R \geq 2 - 0.75$

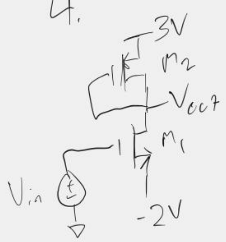
$I_D R \leq -(1.25 - 6)$

$R \leq \frac{4.75}{I_D} = \frac{4.75\ V}{468.75\ \mu A}$

$R \leq 10.13\ k\Omega$

[redacted]

4.



$I_{D1} = I_{D2}$ $\quad (V_{SG} - |V_{Tp}|)^2$

$$\frac{\mu_p C_{ox}}{2} \frac{W_2}{L_2}(3 - V_{out} - 0.75)^2 = \frac{\mu_n C_{ox}}{2} \frac{W_1}{L_1}(2 - 0.75)^2$$

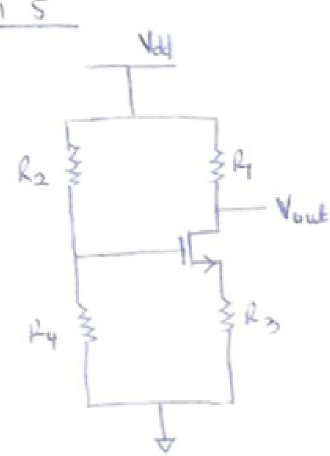
$V_{out} = 1.68$ $\quad$ ~~$V_{out} = 2.81$~~

For sat. $V_{SG} > V_T$, SO

pmos

5.

Problem 5



From the process parameters

$V_{th} = 0.76$, $\frac{W}{L} = \frac{3}{0.6}$

$\frac{\mu_n C_{ox}}{2} = 57.8 \times 10^{-6}\ A/V^2$

$\rightarrow$ $V_G = V_{dd} * \frac{90k}{(90+10)k} = \frac{90}{100} \times 10 = 9V$

$\Rightarrow$ Assuming operation in saturation,

$$I_D = \frac{\mu_n C_{ox}}{2} \frac{W}{L} (V_{GS} - V_T)^2 = 57.8 \times 10^{-6} \times \frac{3}{0.6} (9 - V_S - 0.76)^2$$

$$I_D = 2.89 \times 10^{-4} (8.24 - V_S)^2$$

Since $I_D = I_S = \frac{V_S}{R_3} = \frac{V_S}{2 \times 10^3} = 2.89 \times 10^{-4} (8.24 - V_S)^2$

$$\Rightarrow V_S = 0.578(8.24 - V_S)^2 = 0.578(67.898 - 16.48V_S + V_S^2)$$

$$V_S = 39.25 - 9.53V_S + 0.578V_S^2$$

$$0 = 39.25 - 10.53V_S + 0.578V_S^2$$

Using quadratic formula, $\frac{-b \pm \sqrt{b^2 - 4ac}}{2a} = V_S$

$$V_S = \begin{cases} 12.99 \\ 5.23 \end{cases}$$

→ The correct answer cannot be 12.99, since V_{dd} is only 10v

$\Rightarrow$ For $V_S = 5.23\,v$, $I_D = 2.89 \times 10^{-4}(8.24 - 5.23)^2 = \underline{2.62\,mA}$

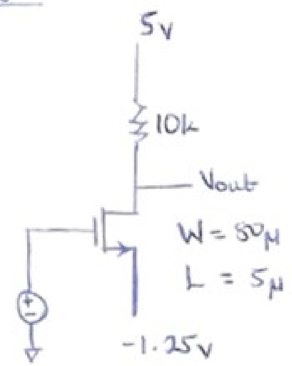
$\Rightarrow V_{out} = R_4 \times I_D = 5000 \times 2.62 \times 10^{-3} = \underline{13.1\,v}$

$$V_{DS} = 13.1 - 5.23 = 7.87\,v$$

$$V_{GS} = V_G - V_S = 9 - 5.23 = 3.77\,v$$

Therefore, $V_{DS} > V_{GS} - V_T$ and the circuit is in saturation

Problem 6



(a)

$V_{gs} = V_{IN} - V_S = 0 - (-1.25) = 1.25V$

$V_{DS} = V_D - V_S = V_D - (-1.25) = V_D + 1.25$

Since $V_{DS} > V_{gs} - V_T$, the circuit is operating in saturation

$\Rightarrow I_D = \frac{\mu_n C_{ox}}{2} \frac{W}{L} (V_{gs} - V_T)^2 = \frac{100 \times 10^{-6}}{2} \frac{80}{5} (1.25 - 0.75)^2 = 200 \times 10^{-6} A$

$\Rightarrow V_{out} = V_{DD} - (I_D \times 10k) = 5 - (200 \times 10^{-6} \times 10 \times 10^{3})$

$V_{out} = 5 - 2 = 3V$

(b) If $V_{IN} = 0.1v$, $V_{gs} = 0.1 + 1.25 = 1.35$

$\rightarrow I_D = \frac{100 \times 10^{-6}}{2} \frac{80}{5} (1.35 - 0.75)^2 = 288 \times 10^{-6} A$

$\rightarrow V_{out} = 288 \times 10^{-6} \times 10k = \underline{2.88v}$

$\Rightarrow$ Thus, the output wave form will be going between 2V and 2.88V

(c) $\text{Gain} = \frac{2.88}{0.1} = \underline{28.8}$

7.

Consider the C_{ox} given at the top of the assignment $C_{ox} = 4\ fF/\mu m^2$

$$C_P = C_{ox} * W_p * L_P = 4 * 20 * 2 = 160\ fF$$

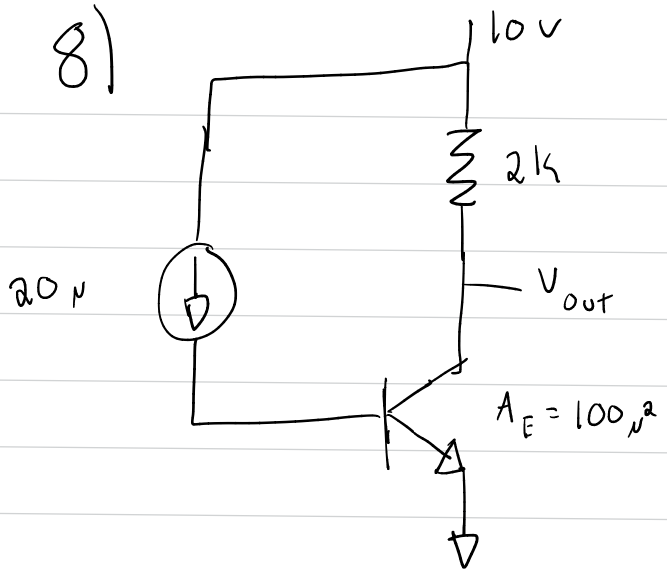
$$C_n = C_{ox} * W_n * L_n = 4 * 5 * 1 = 20\ fF$$

$$C_{total} = 160 + 20 = 180\ fF$$

$$R_{LH} = R_P = \frac{1}{\mu_{cox_p}\left(\frac{W_p}{L_p}\right)(V_{GS} - V_t)} = \frac{1}{75 * 10^{-6} * (\frac{20}{2})(1.5 - 0.5)} = 1.33K\Omega$$

$$R_{HL} = R_N = \frac{1}{\mu_{cox_n}\left(\frac{W_n}{L_n}\right)(V_{GS} - V_t)} = \frac{1}{350 * 10^{-6} * (\frac{5}{1})(1.5 - 0.5)} = 571.4K\Omega$$

8)



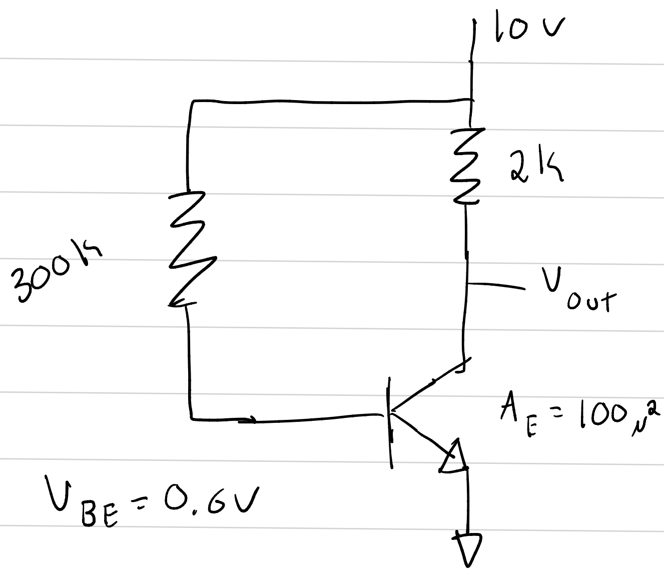
$J_S A = 10^{-12}$ A

$\beta_p = 30$, $\beta_n = 100$

$i_c = \beta_n i_b$, $i_b = 20\mu$ A

$i_c = 100 \cdot 20\mu = 2$ mA

$V_{out} = 10 - 2k \cdot 2$ mA $= 6$ V



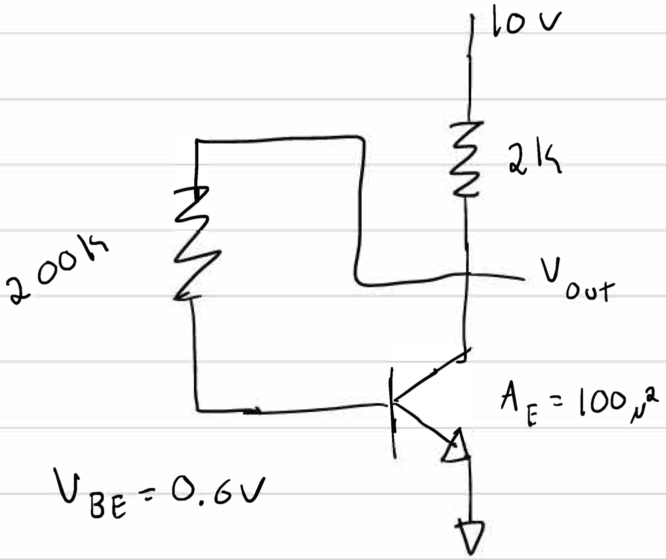
$V_{BE} = 0.6V$

$i_c = \beta_n i_b$, $i_b = \frac{10 - 0.6}{300k}$

$= 31.3\ \mu A$

$i_c = 100 \cdot 31.3\mu = 3.13$ mA

$V_{out} = 10 - 2k \cdot 3.13$ mA $= 3.73$ V



$$i_b = \frac{V_{out} - 0.6}{200k}, \quad i_c = \beta_n i_b$$

$$V_{out} = 10 - i_e \cdot 2k$$

$$i_e = i_b + i_c$$

$$i_b = 23.4\ \mu A$$

$$V_{out} = 5.28\ V$$

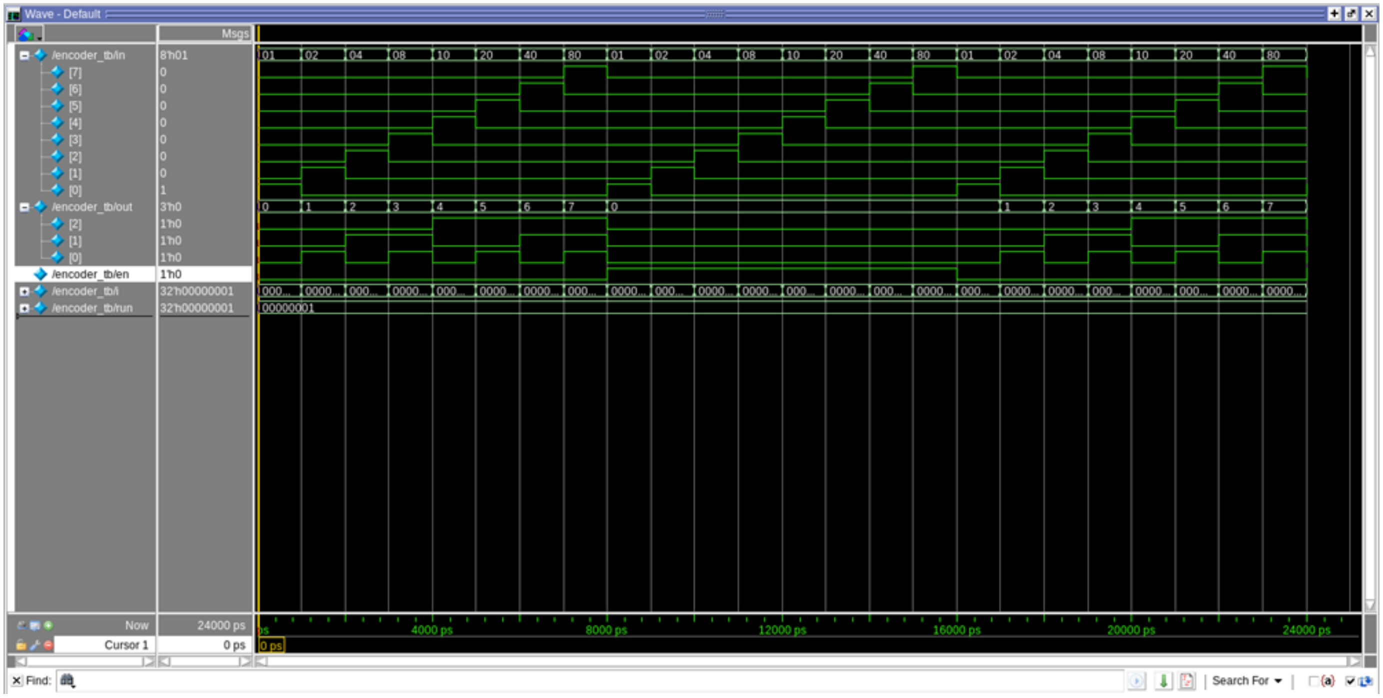
9, 10)

9&10)

```
 //encoder

`timescale 1ns/1ps                                          //nice timescale for
simulation


module encoder_8to3(in, out, en);                           //instantiate module

input [7:0] in;                                             //instantiate input, 8 bits

input en;                                                   //enable pin

output reg [2:0] out;                                       //instantiate output, 3 bits

always @ (*) begin                                          //any time an input changes, execute
```

```
        if (en) begin                                   //active low enable, when en is high

                out = 0;                        //set output to zero

        end

        else begin                                      //if enabled

                case (in)                       //case maps inputs to output

                        8'b00000001 : out = 3'b000;     //only anticipates one input at a time

                        8'b00000010 : out = 3'b001;

                        8'b00000100 : out = 3'b010;

                        8'b00001000 : out = 3'b011;

                        8'b00010000 : out = 3'b100;

                        8'b00100000 : out = 3'b101;

                        8'b01000000 : out = 3'b110;

                        8'b10000000 : out = 3'b111;

                endcase

        end

end

endmodule


//encoder testbench

`timescale 1ns/1ps                                                      //nice timescale for
simulation


module encoder_tb();                                            //instantiate testbench


reg [7:0] in;                                   //instantiate input

reg en;                                         //enable pin

wire [2:0] out;                                 //instantiate output
```

```
integer i, run;                                         //used within testbench loop


encoder_8to3 DUT(.in(in),.out(out),.en(en));      //instantiate DUT

initial begin                                           //start test

    run = 1;                                            //used to force while loop to run
    forever

    in = 0;                                             //input initial value

    en = 0;                                             //enable initial value

    i = 0;                                                  //iteration variable

    while (run) begin                                   //force to loop forever

        if (i > 7) begin                                //if iteration variable exceeds seven

            en = ~en;                                   //toggle enable

            i = 0;                  //set iteration variable to zero

            in = 1;                 //set input to one to avoid negative exponent error

            #1;                     //wait one time unit

            i = 1;                  //set iteration variable to one, enter back into loop

        end

        in = in + (2 ** i);         //add 2^n to input

        in = in - (2 ** (i-1));     //subtract 2^(n-1) from input, above if avoid 2^-1

        i = i + 1;                  //increment iteration variable

        #1;                         //wait one time unit

    end                             //above while loop mimics 8 bit shift register which

end                                     //feeds back into itself. Has one active bit,
rest are zero

endmodule
```



/decoder implementation

```
`timescale 1ns/1ps                          //nice simulation timescale

module decoder_3to8(in, out, en);           //instantiate module
input [2:0] in;                             //instantiate input, 3 bits
input en;                                   //enable pin
output reg [7:0] out;                       //instantiate output, 7 bits
always @ (*) begin                          //any time an input changes, execute
        if (en) begin                       //active low enable, when en is high
                out = 0;                //set output to zero
        end
```

```
        else begin                                      //if enabled

        case (in)                                       //case statement maps input
        to output

            3'b000 : out = 8'b00000001;

            3'b001 : out = 8'b00000010;

            3'b010 : out = 8'b00000100;

            3'b011 : out = 8'b00001000;

            3'b100 : out = 8'b00010000;

            3'b101 : out = 8'b00100000;

            3'b110 : out = 8'b01000000;

            3'b111 : out = 8'b10000000;

        endcase

    end

end

endmodule

//decoder testbench

`timescale 1ns/1ps                                      //nice simulation timescale


module decoder_tb();                                    //instantiate testbench

reg [2:0] in;                                   //instantiate input, 3 bits

reg en;                                         //enable pin

wire [7:0] out;                                 //instantiate output, 8 bits

decoder_3to8 DUT(.in(in),.out(out),.en(en));        //instantiate DUT

    initial in = 0;                             //set initial values

    initial en = 0;

    always #1 in[0] = ~in[0];               //toggle at timed intervals
```

```
        always #2 in[1] = ~in[1];

        always #4 in[2] = ~in[2];

        always #8 en = ~en;

endmodule
```