Homework 5 Spring 2018 TA: Robert Buckley

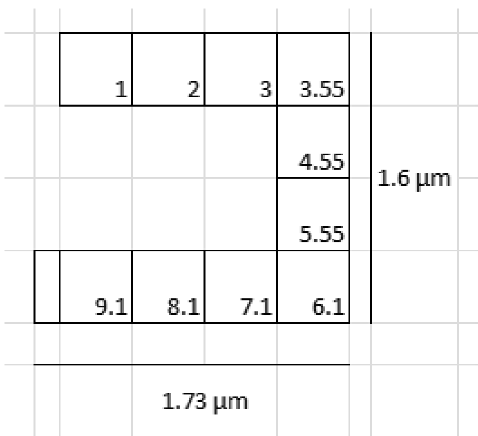
Problem 1:

Sheet Resistance of poly is 317.1 Ω/■ => $\frac{3000}{317.1} = 9.46$

Min Poly width in resistor is 0.1 μm and min spacing is 0.1 μm.

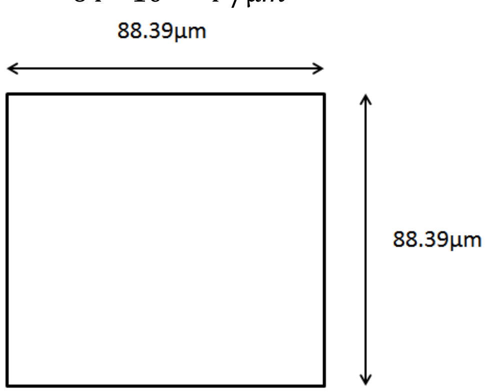
Due to the small number of squares this will be fine (last 1/3rd of a seq

$$4 + 3.3 + 1 + (2 * 0.55) = 9.4\blacksquare = 2980\Omega$$

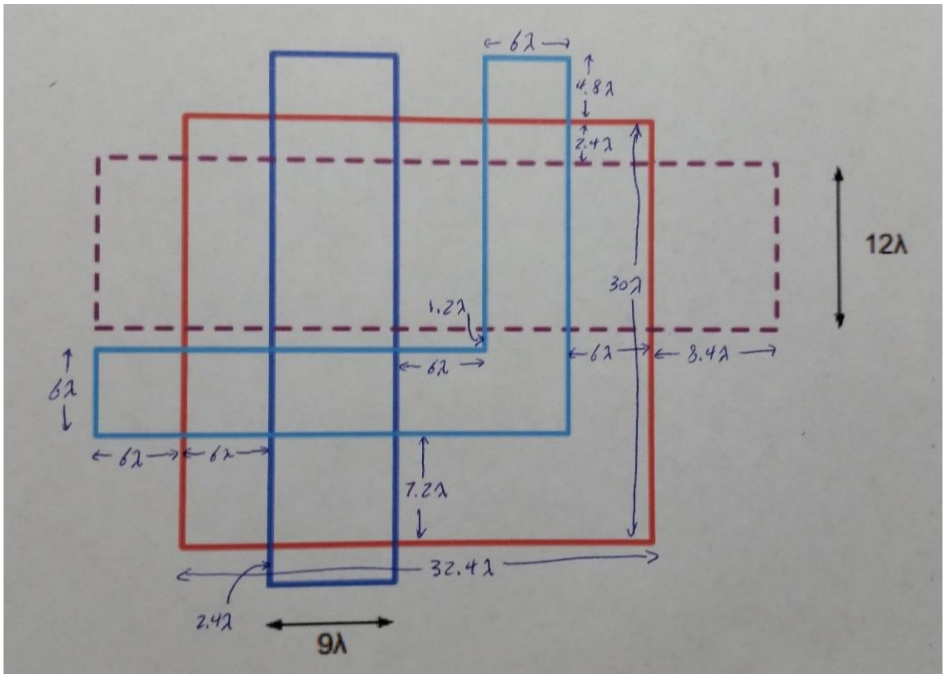


Problem 2: Using poly and Metal1 => 64 aF/μm^2

$$A_C = \frac{0.5 * 10^{-12}\,F}{64 * 10^{-18}\,F/\mu m^2} = 7812.5\ \mu m^2 \Rightarrow \sqrt{A_C} = 88.39\mu m$$



Problem 3:



$\lambda 2 = 0.1^2 = 0.01$

$C_{32} = A_{32} * 38\frac{aF}{\mu m^2} = \lambda^2 * (6 * 9) * C_{\mu m} = 8.64\,aF$

$C_{31} = A_{31} * 16\frac{aF}{\mu m^2} = \lambda^2 * (12 * 6) * C_{\mu m} = 1.92\,aF$

$C_{3P} = A_{3P} * 10\frac{aF}{\mu m^2} = \lambda^2 * (6 * 6 + 6 * 12 + 6 * 1.2) * C_{\mu m} = 11.52aF$

$C_{3S} = A_{3S} * 19\frac{aF}{\mu m^2} = \lambda^2 * (6 * 6 + 6 * 4.8) * C_{\mu m} = 12.31aF$

$C_{21} = A_{21} * 44\frac{aF}{\mu m^2} = \lambda^2 * (12 * 9) * C_{\mu m} = 47.52\,aF$

$C_{2P} = A_{2P} * 18\frac{aF}{\mu m^2} = \lambda^2 * (6 * 12 + 12 * 18) * C_{\mu m} = 51.84\,aF$

$C_{2S} = A_{2S} * 19\frac{aF}{\mu m^2} = \lambda^2 * (6 * 12 + 12 * 8.4) * C_{\mu m} = 32.83\,aF$

$C_{1P} = A_{1P} * 64\frac{aF}{\mu m^2} = \lambda^2 * (30 * 9) * C_{\mu m} = 105.3\,aF$

$C_{1S} = A_{1S} * 39\frac{aF}{\mu m^2} = \lambda^2 * (2.4 * 9 + 4.8 * 9) * C_{\mu m} = 25.27\,aF$

$C_{PS} = A_{PS} * 103\frac{aF}{\mu m^2} = \lambda^2 * (30 * 32.4) * C_{\mu m} = 1001.16\,aF$

Problem 4:

$$R(320) = 4534 * \left(1 + (320 - 250) * \left(\frac{1200}{10^6}\right)\right) = 4914\ \Omega$$

Problem 5:

Using the resistivity calculator at http://cleanroom.byu.edu/ResistivityCal, the resistivity equals

ρ = 22.34 Ω-cm = 223400 Ω-um

$$R = \rho * \left(\frac{L}{W * t}\right) = 223.4 * 10\text{^}3 * \left(\frac{50}{2 * t}\right) = \frac{5.59 * 10^6}{t}\ \Omega$$

Problem 6:

Value of combination is $R_T = R_1 + R_2$

Substitution and algebra yield $R_T = (R_1 + R_2) * (1 + \frac{\Delta T}{10^6} * \left(\frac{R_1}{R_1+R_2} * TCR_1 + \frac{R_2}{R_1+R_2} * TCR_2\right)$

This matches the form of the original equation if $TCR_T = \left(\frac{R_1}{R_1+R_2} * TCR_1 + \frac{R_2}{R_1+R_2} * TCR_2\right)$

=> $TCR_T = 133.33\ ppm/°C$

The TCR is $\frac{1400}{133.33} = 10.5$ times less than just an n+ doped resistor.

Problem 7:

$$R = R_s * \left(\frac{L}{W}\right) = 7.7 * \left(\frac{100}{1}\right) = 770\ \Omega$$

$$C_{PS} = \left(103 \frac{aF}{\mu m^2}\right) * (100 * 1) = 10300\ aF$$

$$C_{P2} = \left(64 \frac{aF}{\mu m^2}\right) * (100 * 1) = 6400\ aF$$

Problem 8:

$$I_d = J_s A \left(e^{\frac{V_D}{V_T}} - 1\right) = \left(50um^2 * \frac{10^{-15} A}{um^2}\right) * \left(e^{\frac{0.5V}{26mV}} - 1\right) = 78.2\ uA$$

$$I_d = J_s A \left(e^{\frac{V_D}{V_T}} - 1\right) = \left(50um^2 * \frac{10^{-15} A}{um^2}\right) * \left(e^{\frac{0.6V}{26mV}} - 1\right) = 3.40\ mA$$

Problem 9:

$$I_d = \left(\frac{10^{-15}A}{um^2} * 200\ um^2\right)\left(e^{\frac{V_D}{26mV}} - 1\right)$$

$$V_x - V_R - V_D = 10V - I_d(2k\Omega) - V_D = 0$$

Solve system of equations: $I_D = 4.71\ mA$

Problem 10:

$$I_d = \left(\frac{10^{-15}A}{um^2} * 200\ um^2\right)\left(e^{\frac{V_D}{26mV}} - 1\right)$$

$$V_x - V_R - V_D = 520mV - I_d(2k\Omega) - V_D = 0$$

Solve system of equations: $I_D = 41.1\ uA$

Problem 11 and 12:

Half Adder Code

```
`timescale 1ns / 1ps

module Halfadder (iA, iB, oSum, oCarry);
  input iA,iB; //defining inputs
  output oSum, oCarry; //defining outputs

  assign oSum = iA^iB; //sum is equal to A xor B
  assign oCarry = iA & iB; //carry is equal to A and B

endmodule
```

Half Adder Test Bench

/home/falegria/ee330/Half_adder_tb.v - Default

```
`timescale 1ns / 1ps
module half_adder_tb ();
  reg A,B;
  wire Sum, Carry;
  Halfadder uut (.iA(A), .iB(B), .oSum(Sum), .oCarry(Carry)); //unit under

initial
begin //begin setting bits (basically recreating the truth table)


    A = 1'b0;
    B = 1'b0;
  end
  initial
  begin
    #10//timestamp to wait 35ns

    A = 1'b0;
    B = 1'b1;

    #10 //timestamp to wait 35ns

    A = 1'b1;
    B = 1'b0;

    #10 //timestamp to wait 35ns

    A = 1'b1;
    B = 1'b1;

   #10 ;//timestamp to wait 35ns
  end
endmodule
```

Half Adder Output