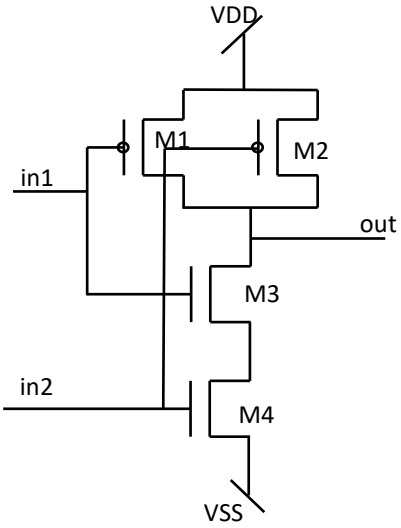
18th International Microelectronics Olympiad

PROBLEMS

(10 points for each problem)

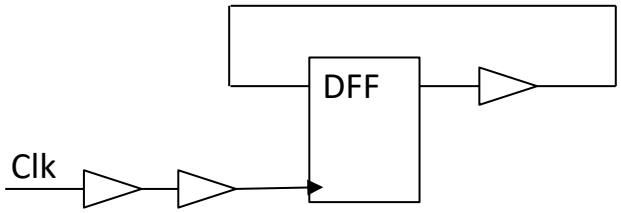
Problem 1.

Calculate the Vsp of the given circuit when the circuit switches from in1=0, in2=0 state into in1=1,in2=1 state if Wp(M1)=4u, Wp(M2)=5u, Wn(M3)=Wn(M4)=3u, VSS=0V, VDD=1.2V, Vth(n)=0.4V, Vth(p)=0.6V μ_nCox=60 μ_p Cox=20, Ln=Lp=0.03u.



Problem 2.

Determine the minimum period of the clock pulse of the given circuit if T_{pBuf}=15ps, $T_{clk>Q}(DFF)$=80ps, T_{hd}=-20ps, T_{su}=25ps.



Problem 3.

in linear mode the resistance of an n-MOS transistor with an induced current is 2.5 kΩ. How will the width-to-length (W/L) ratio of a transistor change if the resistance is doubled?
The I-V characteristic of the transistor in linear mode is $I_{DS} = \mu_n C_{ox} (W/L) (V_{GS}- Vt) V_{DS}$.

Problem 4.

To ensure the same specific capacitance C_0, how will the thickness (d_{SiO2}= 1 nm) of a MOS capacitor insulator change in case of using a high-k insulator (Hf O_2)?
Assume ε_{SiO2} =4 and $\varepsilon_{Hf\,O2}$=20.

Problem 5.

Construct a Galois LFSR circuit with feedback polynomial $d(x)=1+x^3+x^4+x^5+x^8$. Determine the sequence of states of this circuit, starting with the state S0 (1000 0000). How many states does this circuit have? Describe the circuit in Verilog.

Problem 6.

Design a circuit of BCD counter using a 5-bit Johnson's counter and an encoder.

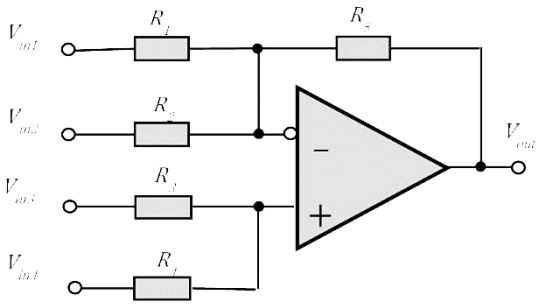
Problem 7.

Draw the circuit of a low-pass filter inverting amplifier on OpAmp, select and calculate the elements of the circuit, if known:
Input resistance: R_{IN} =1k

Gain coefficient: Ku = -5
Cutoff frequency: Fc= 1MHz

Problem 8.

Calculate the value of the output voltage for the combinational adder/subtractor circuit if V_{IN1} = 1V, V_{IN2} = 2V, V_{IN3} = 4V, V_{IN4} = 1V, $R_1 = R_2 = R_4$ = 1k, $R_3 = R_5$ = 2k, OpAmp is with double supply, the amplifier and input voltage sources are ideal.



Problem 9.

The following truth table is given.
Truth table:

x_3	x_2	x_1	x_0	y
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	1
1	1	1	1	1

It is required to:
1. Construct the canonical disjunctive normal form of the function $y=f(x_0,x_1,x_2,x_3)$
2. Reduce the canonical disjunctive normal form of the function with the help of the Karnaugh map, getting the form of the function with NAND basis
3. Draw the corresponding combinational circuit.

Problem 10.

The wave algorithm connecting the given point A to point B is implemented in the discrete routing field, in which only the length of the connection is taken into account. The implementation of the wave algorithm and the result are given in the figure.
It is required to connect a given point A to point B using a wave algorithm with penalty units, in which the length of the connection is accessed 1 unit, the adjacency to an already taken interconnect (shown in dark color) is accessed 1 unit, and the adjacency of interconnect and contact square (shown by the circle) is 2 units.

6	5	⊗					
5	4				⊗		
4	3	2	3	4	5	6	
3	2	1	2	⊗		7**B**	
2	1	A0	1		7		
4	2	1	2	⊗	6	7	11
5	4	2	3	4	5	6	10
6	5	4	5	6	7	8	9

Problem 11.
Solve the functional equation $f(x-3)+Cf(7-x)=x^3$.

Here f is continuous on the real axis function which should be determined and C is constant. For what C the problem has a solution? Is it possible to solve this problem with given function g instead of x^3? Is this problem always solvable?

Problem 12.
Compare the values of integrals (answer should be argumentative):

$$\int_0^{\frac{\pi}{2}} \frac{dx}{1+\tan^{2022} x} \quad \text{and} \quad \int_0^{\frac{\pi}{2}} \frac{dx}{1+\tan^{2023} x}.$$

Problem 13.
The given combinational circuit has N lines that can be in three different states independently of each other.

1. A line is in the "fault-free" state
2. A line is in the state "stuck-at-0", i.e. it has a fault of type "stuck-at-0"
3. A line is in the state "stuck-at-1", i.e. it has a fault of type "stuck-at-1".

It is supposed that the circuit may work properly, i.e. all its lines are fault-free, or it may have multiple stuck-at faults, i.e. two or more number of lines may have stuck-at-1 faults.
Find the overall number of multiple (not including single) stuck-at-1 faults.

Problem 14.
The given combinational circuit has N lines that can be in three different states independently of each other:

1. A line is in the fault-free state
2. A line is in the state "stuck-at-0", i.e. it has a fault of type "stuck-at-0"
3. A line is in the state "stuck-at-1", i.e. it has a fault of type "stuck-at-1".

It is supposed that the circuit may work properly, i.e. all its lines are fault-free, or it may have multiple stuck-at faults, i.e. two or more number of lines have stuck-at-0 faults. Please find the overall number of multiple (not including single) stuck-at-0 faults.

Problem 15.
If the concentration of acceptors in silicon $N_a=2{,}3\cdot10^{13}cm^{-3}$, while the concentration of donors $N_d=2{,}2\cdot10^{13}cm^{-3}$, find the specific conductivity at a temperature of 300^oK.

Problem 16.
If at a temperature of 20°C in a semiconductor sample in the form of a parallelepiped with dimensions of 0,25x0,25x0,25cm, the concentration of charge carriers is 10^{15} cm^{-3}, a voltage of 20V is applied to thin sides. Mobility is equal $500cm^2/V\cdot s$. Find the current passing through the sample.

Problem 17.
Write a function that takes an array of positive integers as input and finds the longest decreasing subsequence within the array. A subsequence is defined as a sequence of elements from the original array where the elements are not necessarily adjacent but are in decreasing order. The function should return the longest decreasing subsequence.
For example: for the input array [5, 9, 3, 4, 8, 10, 6], the output should be [9, 8, 6].

Problem 18.
Fill in the missing part of the following code in the marked area to make it work as expected.

```
#include <iostream>
#include <vector>

using namespace std;
struct Product {
   string name;
   double price;
   int quantity;
};

template <typename C>
void processProducts(C c, const vector<Product>& products) {
   for (const auto& product : products) {
      c(product);
   }
}

template <typename C>
void call(C c, Product product) {
   if (product.price > 10) {
      c("Expensive ", product.name, product.price, product.quantity);
   } else {
      c("Cheap ", product.name, product.price, product.quantity);
   }
}
// ---------- DO NOT TOUCH THE CODE ABOVE ----------
/*

   Your code goes here.

*/
// ---------- DO NOT TOUCH THE CODE BELOW ----------
int main() {
   vector<Product> products = {
      {"Apple", 12, 10},
      {"Banana", 6, 8},
      {"Orange", 8, 12}
   };
   processProducts(productInfo(
      [](const string& type, const string& name, double price, int quantity) {
         cout << "Type: " << type << "Product: " << name << ", Price: $" << price << ", Quantity: " <<
quantity << endl;
      }
   ), products);
}
```