Faculty of Engineering

DEPARTMENT of ELECTRICAL AND ELECTRONIC ENGINEERING

EENG447	Instructor:
Digital IC Design	G. YEMİŞCİOĞLU
Midterm EXAMINATION	Duration: 90 minutes
November 16, 2017	Number of Questions: 4

Good Luck

STUDENT'S	
NUMBER	
NAME	SOLUTIONS
SURNAME	
GROUP NO.	

Question	Achieved	Points
1		15
2		15
3		20
4		20
5		15
6		15
TOTAL		100

Read the following instructions carefully:

1. Calculators are allowed.
2. Switch off mobile phones and do not borrow any stationery from your friends.
3. In your solutions, show all details you claim credit for.

Question 1:

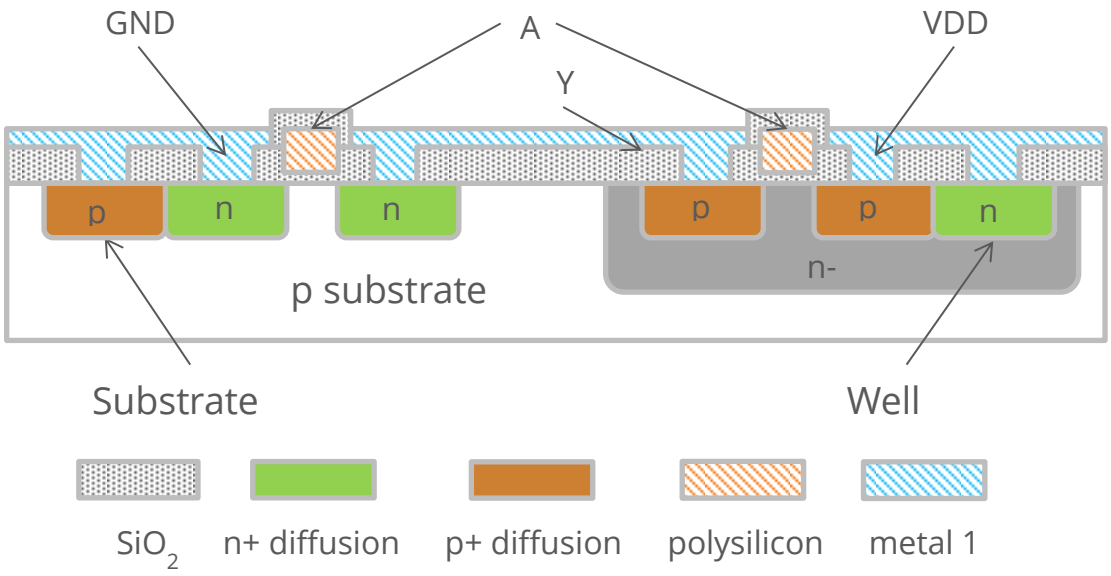
(a) What is Moore's Law? Discuss how it is affecting IC design industry.

(**3 pts**)

- The observation made in 1965 by Gordon Moore
- The number transistor that could be placed in an integrated circuit will double approximately every 12-18 months.
- As the transistor sizes are shrinking to atomic levels, the number of transistors inside a chip is increased, this has an effect on the speed, performance, power consumption, functionality and also reliability of the chip.

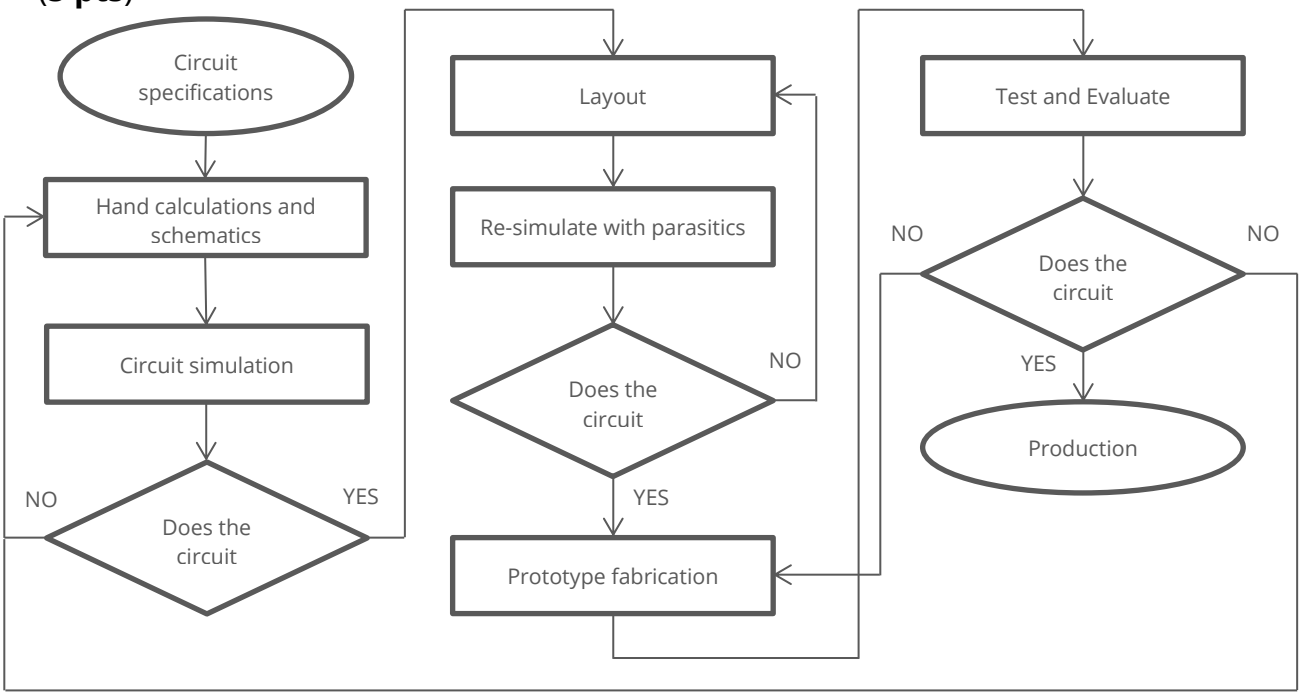
(b) Draw and label the terminals and substrate doping of an nMOS and pMOS transistor cross-section forming a CMOS inverter, including the well(s).

(**7 pts**)



(c) Draw the typical IC design flow and explain briefly each of the steps required.

(**5 pts**)



Question 2:

Consider the And-OR-Inverter (AOI) cell shown in **Figure 1**. Derive the CMOS complex gate that implements this cell.
(**15 pts**)

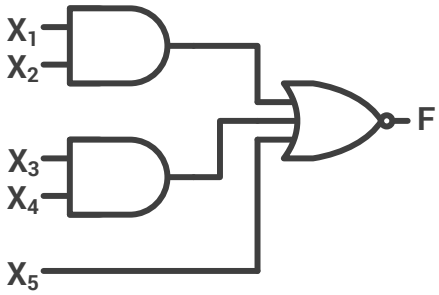
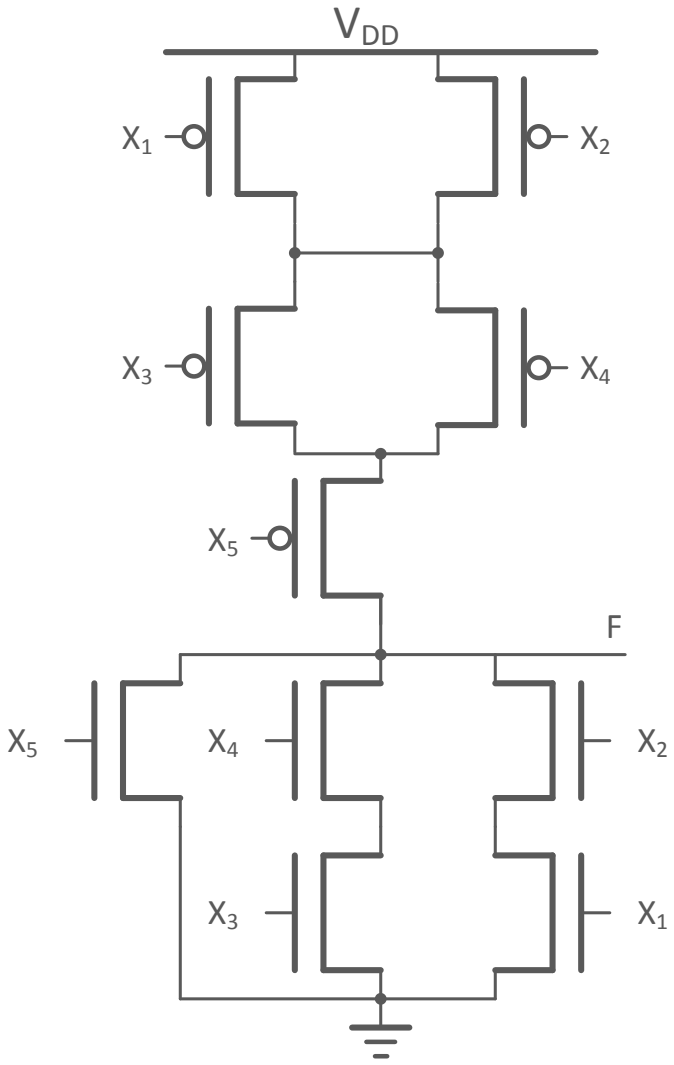


Figure 1: AOI Cell.

$T1 = X_1.X_2$
$T2 = X_3.X_4$
Pull-Up Network
$F = (X_1.X_2 + X_3.X_4 + X_5)'$
$F = (X_1'+X_2') (X_3'+X_4').X_5'$
$F = (X_1'X_3'+X_1'X_4'+X_2'X_3'+X_2'X_4')X_5'$

Pull-Down Network
$F = (X_1.X_2) +(X_3.X_4)+X_5'$

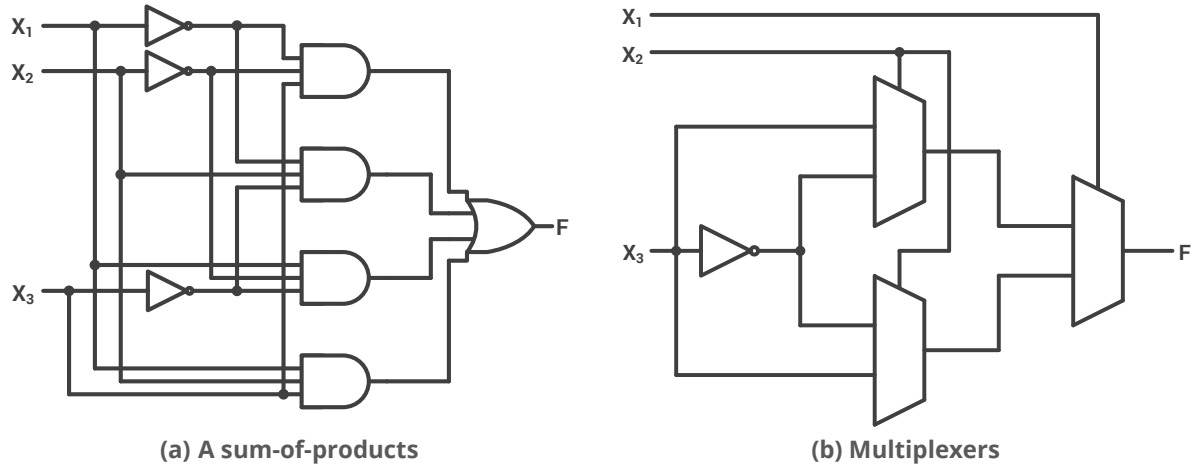


Question 3:

Consider the circuit shown in **Figure 2 (a).**

(a) Show the truth table for the logic function ***F***.
(b) If each gate in the circuit is implemented as a CMOS gate, how many transistors are needed?
(c) Show that the circuit in **Figure 2 (b)** is functionally equivalent to the circuit in **Figure 2 (a).**
(d) How many transistors are needed to build this CMOS circuit?
(e) Which function do these circuits perform?
(f) Implement the function with a minimum number of transistors.

(**20 pts**)



(a) A sum-of-products **(b) Multiplexers**

Figure 2: CMOS circuit.

a)

$T1 = X_1'X_2'X_3$, $T2 = X_1'X_2X_3'$, $T3 = X_1X_2'X_3'$, $T4 = X_1X_2X_3$, $F = T_1+T_2+T_3+T_4$

X_1	X_2	X_3	X_1'	X_2'	X_3'	T1	T2	T3	T4	**F**
0	0	0	1	1	1	0	0	0	0	**0**
0	0	1	1	1	0	1	0	0	0	**1**
0	1	0	1	0	1	0	1	0	0	**1**
0	1	1	1	0	0	0	0	0	0	**0**
1	0	0	0	1	1	0	0	1	0	**1**
1	0	1	0	1	0	0	0	0	0	**0**
1	1	0	0	0	1	0	0	0	0	**0**
1	1	1	0	0	0	0	0	0	1	**1**

$F = X_1 \oplus X_2 \oplus X_3$

b)

AND gate has 8 transistors = 8 x 4 AND gate = 32 transistors

OR gate has 10 transistors = 10 x 1 Or gate = 10 transistors

Inverter has 2 transistors = 2 x 3 Inverters = 6 transistors

Total = **48 transistors**

c)

Mux1 = $X_2'X_3 + X_2X_3'$

Mux2 = $X_2' X_3' + X_2 X_3$

Mux2 = X_1' Mux1+ X_1 Mux2

= $\mathbf{X_1'X_2'X_3 + X_1'X_2X_3' + X_1X_2' X_3' + X_1X_2 X_3}$

= **T1 + T2 + T3 + T4**

d)

Mux has 2 TG and 1 Inverter

TG has 2 transistors = 2 x 2 TG = 4 transistors

Inverter has 2 transistors = 2 x 1 Inverter = 2 transistors.

Total = 6 transistors

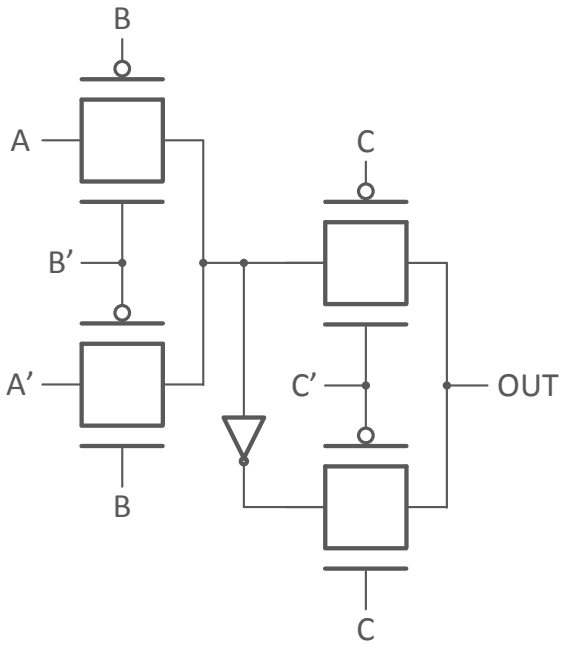
3 x Muxes = 18 transistors

1 x Inverter = 2 transistors

Total = **20 transistors**

e) **XOR**

f)



Question 4:

(a) Derive a CMOS complex gate for the logic function

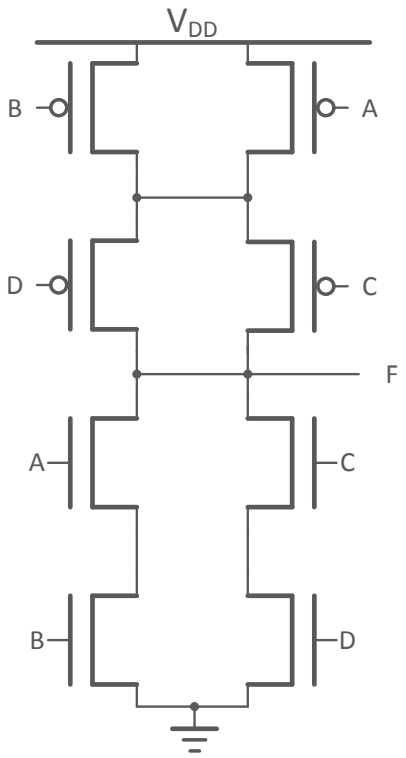
F (A, B, C, D) = Σ (0, 1, 2, 4, 5, 6, 8, 9, 10).

(**15 pts**)

AB \ CD	00	01	11	10
00	1	1	0	1
01	1	1	0	1
11	0	0	0	0
10	1	1	0	1

F = B'D' + A'C' + B'C' + A'D'

F' = AB + CD



(b) Analyse the CMOS logic circuit shown in **Figure 3** and write the expression for output **F**.

(**5 pts**)

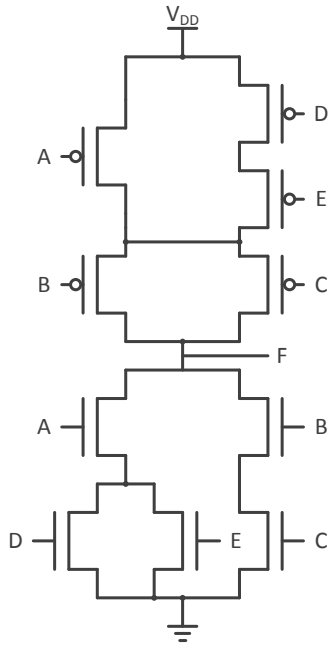
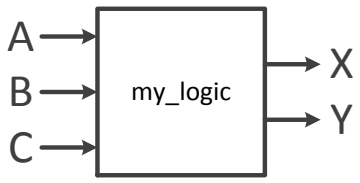


Figure 3: CMOS Transistor Logic

F = (A'+D'E')(B'+C')
F' = A(D+E)+(B.C)

Question 5:

The specification for a VHDL entity may be represented as a set of input and output signals, shown by the symbol in **Figure** 4 **(a)**, and an internal behaviour as defined by the truth table in **Figure** 4 **(b)**. Write a complete, commented VHDL code that implements the following functions specified.



(a)

A	B	X
0	0	1
0	1	0
1	0	1
1	1	1

B	C	Y
0	0	0
0	1	1
1	0	1
1	1	0

(b)

Figure 4: Logic Block Diagram & Truth Table

(**15 pts**)

```
library ieee;
use ieee.std_logic_1164.all;

-- Entity declaration.
ENTITY my_logic IS
   PORT(
      A,B,C : IN BIT;      -- input declarations BIT type
        X,Y : OUT BIT;     -- output declarations BIT type
   );
-- behavioural of circuit
ARCHITECTURE behave OF my_logic IS
BEGIN
   X <= NOT A AND NOT B OR A AND NOT B OR A AND B; -- X function
   Y <= B XOR C;                                   -- Y function
END behave;
```

Question 6:

Consider the following two VHDL code fragments:

```
PROCESS
BEGIN
   WAIT UNTIL (Clock'EVENT AND Clock='1');
     IF reset='1' THEN
        Q2 <= '0';
     ELSE
        Q2 <= D;
     END IF;
END PROCESS;
```

```
PROCESS (Reset, Clock)
BEGIN
   IF reset='1' THEN
      Q3 <= '0';
   ELSEIF (Clock'EVENT AND Clock='1') THEN
      Q3 <= D;
   END IF;
END PROCESS;
```

How do these two code fragments differ?
(**15 pts**)

In the leftmost process, the process begins and then waits for a rising clock edge ("WAIT UNTIL…). At the rising clock edge, it checks first for an active reset signal, and if it finds ones it stores 0 and sends that 0 through to output Q2. If reset is low it stores D and sends that D value through to output Q2. Thus it specifies a rising-edge triggered D flip flop with synchronous reset.

In the rightmost process, the process is "sensitive" to Reset and Clock, meaning that a change on one of those signals activates the process. If Reset activated the process, it stores 0 and sends that 0 through to output Q3. If a clock event activated the process and that event was a rising clock edge, it stores D and sends that D value through to output Q3. Thus it specifies a rising-edge triggered D flip-flop with asynchronous reset.