Rajshahi University of Engineering and Technology
Course Title: Sessional Based on CSE 2203
Course Code: CSE 2204
Lab Final Report

Submitted to

Dr. Boshir Ahmed
Professor,
Computer Science and Engineering, RUET

Md. Zahirul Islam
Lecturer,
Computer Science and Engineering, RUET

Submitted by

Mrittika Roy
Roll: 1803175
Section: C
Computer Science and Engineering, RUET

Date of Submission: 25.08.2021

Experiment No: 1

Name of the Experiment:
F (A, B, C, D) = SOP (1, 2, 3, 6, 9,11).

Objectives:

- To know about SOP
- To know about minterms
- To know about the use of minterms
- To know the implementation of logic function using canonical form

Theory:
The full form of SOP is ‘Sum of Products’. Boolean functions expressed as a sum of minterms or product of maxterms are said to be in canonical form. A Boolean function can be expressed, canonically, as a sum of minterms, where each minterm corresponds to a row (of the function's truth table) whose output value is 1. The canonical form of a boolean expression is unique.

Experimental Result Analysis:

i. Circuit:

Using Drag and Drop:

1. F(A, B, C, D) = SOP(1,2,3,6,9,11) (Using Drag and Drop)

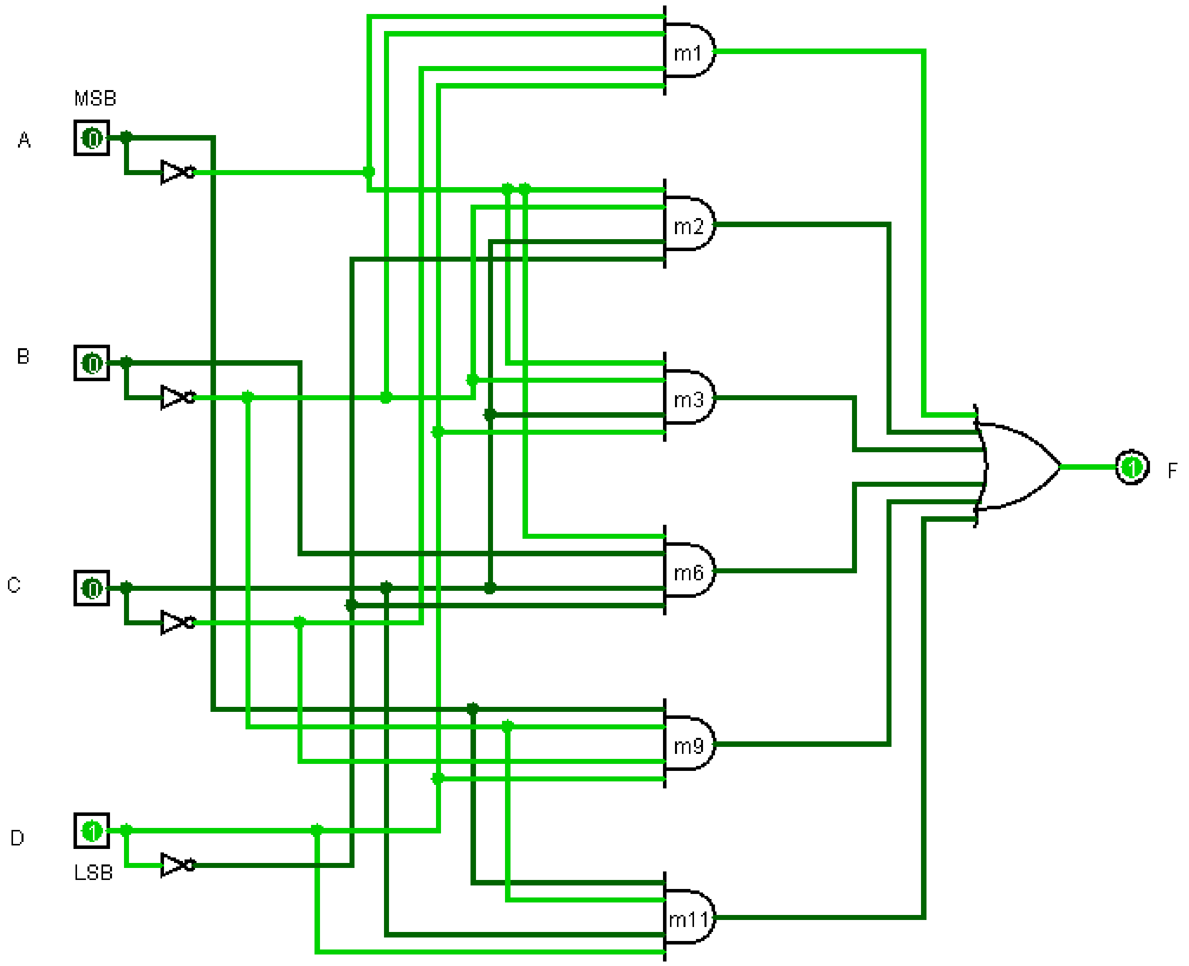
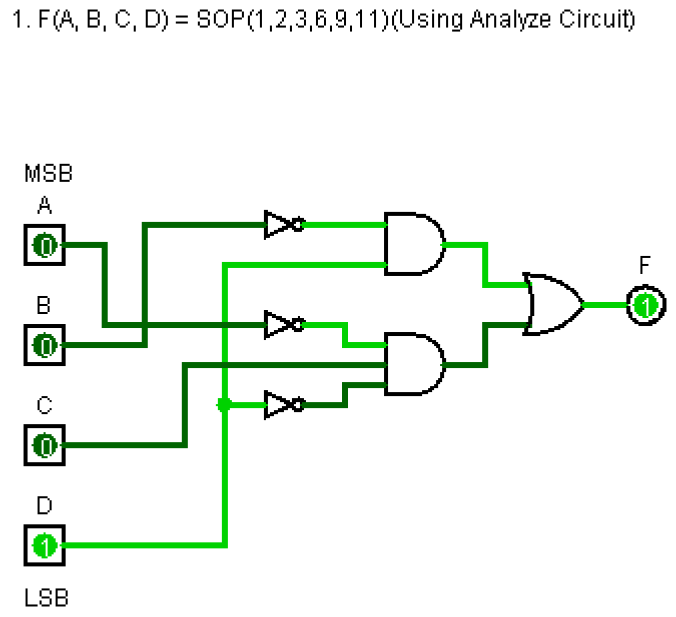


Figure: Circuit Diagram for the Given Function Using Drag and Drop

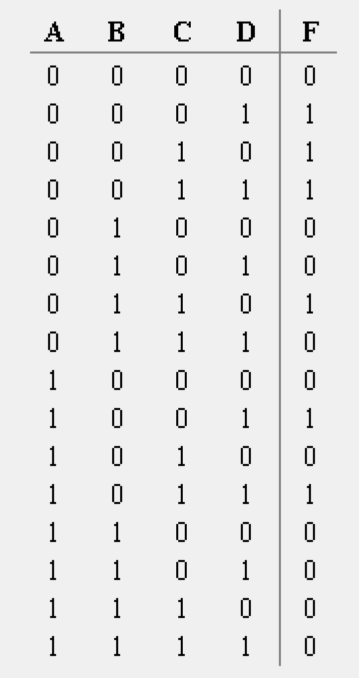
Using Analyze Circuit:



1. F(A, B, C, D) = SOP(1,2,3,6,9,11)(Using Analyze Circuit)

Figure: Circuit Diagram for the Given Function Using Circuit Analysis

ii. Truth Table:



A	B	C	D	F
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

Conclusion:

From the above experiment we can conclude that we can implement any logic function using the sum of minterms.

Experiment No: 2

Name of the Experiment:
Simplified Version of F.

Objectives:

- To know about the methods of simplification of Boolean Functions
- To know about Boolean Algebra
- To Know about Karnaugh Map

Theory:
Simplification of Boolean Expression means simplifying the given expression with fewer numbers of terms and operations keeping the output as before. The process allows the implementation of a simpler circuit compared to its original form. There are two ways to simplify the boolean expression. They are: a. Using Boolean algebra and b. Using Karnaugh Map. Boolean algebra uses many laws to simplify the expression. But K-map(Karnaugh map) uses the graphical technique to simplify the given expression.
In this experiment, the given expression of F in Experiment 1 is simplified using K-map. The given expression is:
F (A, B, C, D) = SOP (1, 2, 3, 6, 9,11)
Or, F (A, B, C, D) =A’B’C’D + A’B’CD’ + A’B’CD + A’BCD’ + AB’C’D + AB’CD
Using the K-map we get,

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

So the simplified expression is:
F = B’D + A’CD’

Experimental Result Analysis:

i. Circuit:

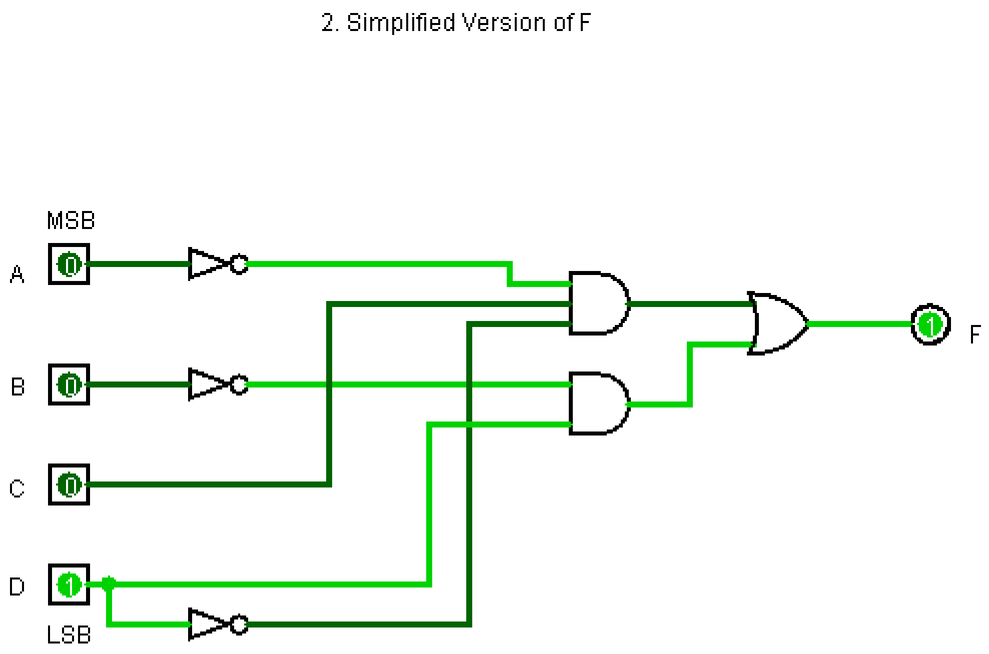
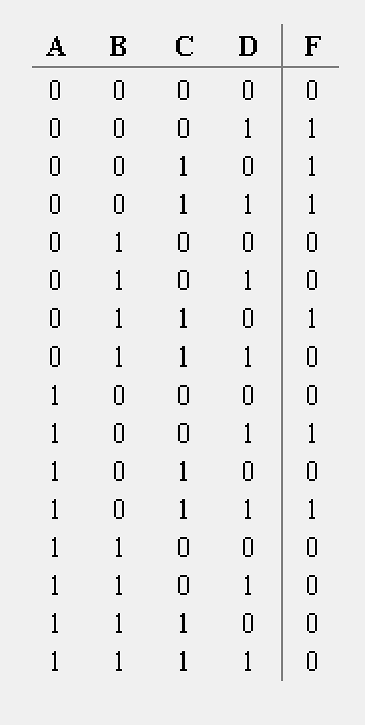


Figure: Circuit Diagram for the Simplified Version of Given Function

ii. Truth Table:



A	B	C	D	F
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

Conclusion:

From this experiment, it is verified that the circuit diagram of the simplified version of F is identical with the circuit diagram of Experiment 1 using the analyze circuit approach. So we can conclude that k-map provides the simplest version of any Boolean expression.

Experiment No: 3

Name of the Experiment:
Designing a counter that counts 0, 3, 2, 1, 4, 0 by using different type flip-flops.(D and T flip flops are used).

Objectives:

- To know how counter works
- To know how to make the circuit excitation table
- To know how to implement the circuit from the circuit excitation table
- To know how to design a counter that counts customized counting sequence

Theory:
A counter is a sequential circuit that can count a specific sequence of numbers or states. Counters can be implemented using various types of flip flops.To count a n bit number, n numbers of flip flops are needed. A n bit counter can count maximum 2^n numbers of states or numbers. But the counting sequence and number of counting states can be customized as per the need.
In this experiment, the given counting sequence is 0, 3, 2, 1, 4, 0. To count the sequence, a 3 bit counter is needed. Because the maximum number in this sequence is 4 and to represent 4 in the binary number system, 3 bits are needed. A 3 bit counter can count 000 to 111. But here there are some unused states. They are: 5, 6 and 7. While implementing the counter for the given sequence, the unused states are considered as don't care conditions.

Experimental Result Analysis:
i. State transition diagram:

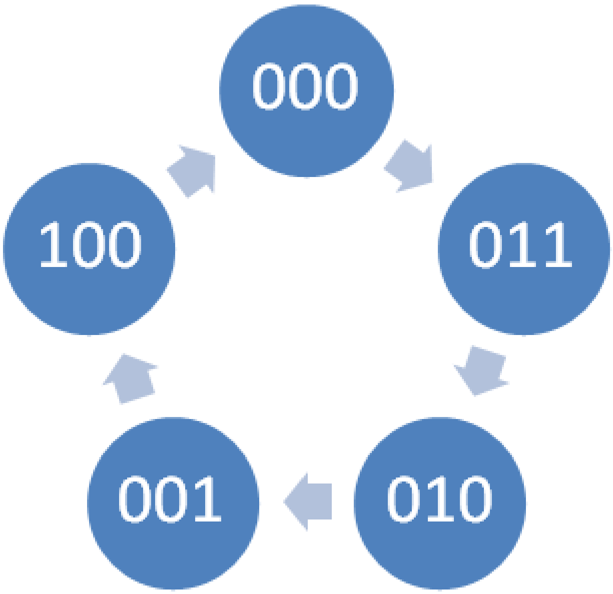


Figure: State Transition Diagram for the Given Counting Sequence

ii. Circuit Excitation Table:

Present State			Next State			Flip-Flops		
Q_C	Q_B	Q_A	Q_{C+1}	Q_{B+1}	Q_{A+1}	D_C	T_B	D_A
0	0	0	0	1	1	0	1	1
0	1	1	0	1	0	0	0	0
0	1	0	0	0	1	0	1	1
0	0	1	1	0	0	1	0	0
1	0	0	0	0	0	0	0	0

iii. Necessary K-maps for Circuit Implementation:

For D_C:

BA C	**00**	**01**	**11**	**10**
0	0	1	0	0
1	0	X	X	X

$D_C = B'A$

For T_B:

BA C	**00**	**01**	**11**	**10**
0	1	0	0	1
1	0	X	X	X

$T_B = C'A'$

For D_A:

BA C	00	01	11	10
0	1	0	0	1
1	0	X	X	X

$D_A = C'A'$

iv. Circuit:

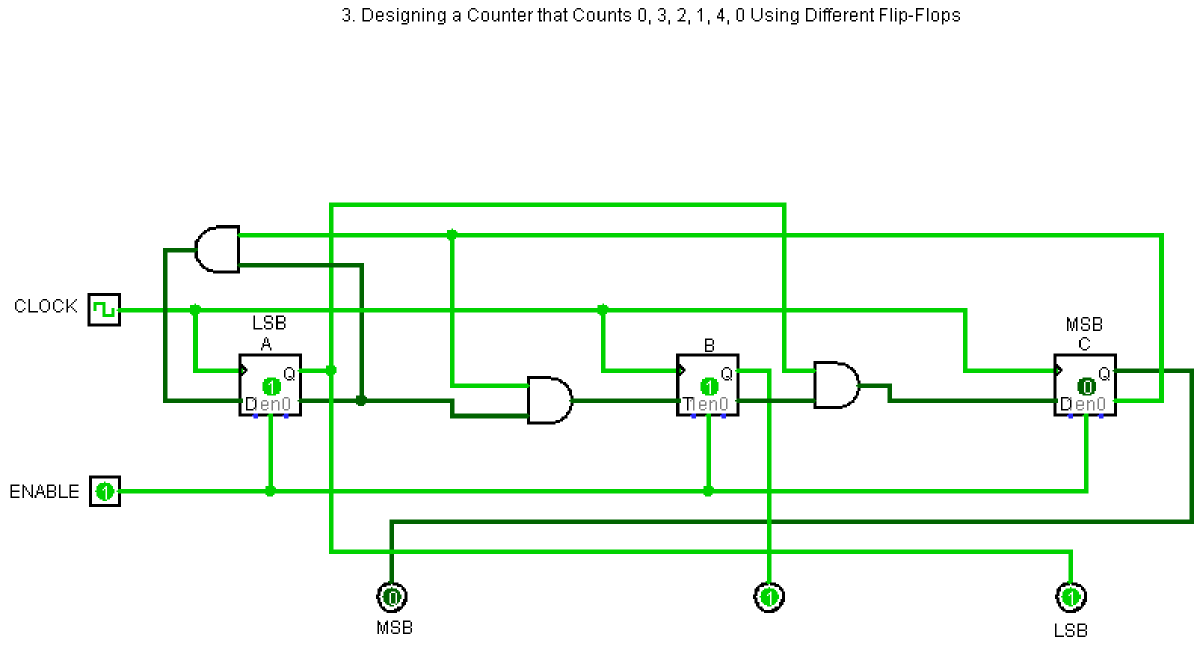

3. Designing a Counter that Counts 0, 3, 2, 1, 4, 0 Using Different Flip-Flops
CLOCK
ENABLE
LSB
A
Q
D en 0
B
Q
T en 0
MSB
C
Q
D en 0
MSB
LSB

Figure: Circuit Diagram for the Counter that Counts the Given Sequence

Conclusion:

In the above experiment, A counter is designed that counts 0, 3, 2, 1, 4, 0. Here two D flip flops and one T flip flop are used to implement the circuit. In this experiment, a synchronous counter is implemented to avoid propagation delay. The implemented counter counts the every given state sequentially and it changes its state at each clock pulse.