**Laboratory Report Cover Sheet**

**18ECE206J ADVANCED DIGITAL SYSTEMS DESIGN**

**Fourth Semester, 2021-22 (Even semester)**

SRM Institute of Science and Technology College of Engineering and Technology

Department of Electronics and Communication Engineering

**Name :**

**Register No. :**

**Day/ Session :**

**Venue :**

**Title of Experiment :**

**Date of Conduction :**

**Date of Submission :**

|  |  |  |
| --- | --- | --- |
| **Particulars** | **Max. Marks** | **Marks**  **Obtained** |
| Pre lab and Post lab | 10 |  |
| Lab Performance | 20 |  |
| Simulation and results | 10 |  |
| Total | 40 |  |

**REPORT VERIFICATION**

**Staff Name : Signature :**

**2. Implement Reed-Muller expressions using logic gates**

**Aim:** To design and implement Reed-Muller expressions using Logic gates

**Software requirement:**  Logisim

Theory: Reed-Muller logic is an algebraic technique for logic circuit design based on AND and Exclusive-OR (Modulo2arithmetic) operations. There is no universal definition and name for Reed- Muller logic. In the literature, it can be found that many different names are used, such as exclusive OR-switching function, EXOR 1ogic, Reed-Muller expansion of Boolean functions, Reed-Muller 1ogic, Reed-Muller algebraic techniques, Reed-Muller representation (expansion, form) of Boolean 1ogic, modulo-2 expressions, AND-EXOR expression, Reed-Muller polynomial. All these names are often used to describe the same thing, that is, at first, a logic function is represented in exclusive OR sum of products instead of inclusive OR sum of products, and then, the function is minimized by employing certain rules based on two basic operations, AND and Exclusive OR.

Let us illustrate Reed Muller Expansion technique with the following example

F(A,B,C,D)=∑m(5,7,10,15)

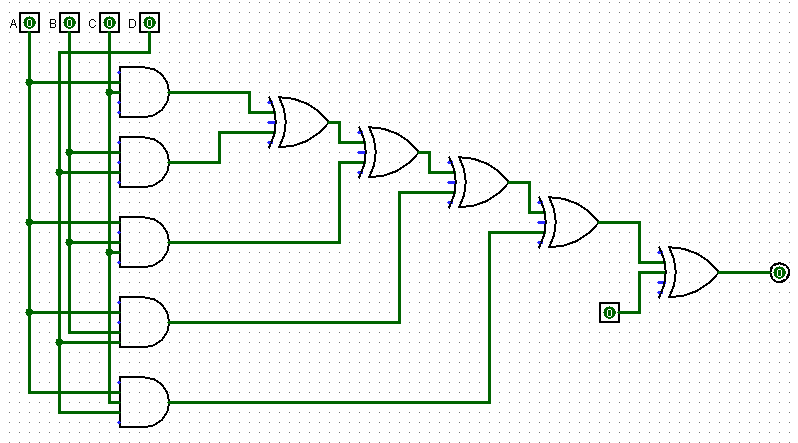
Solution: F(A,B,C,D)=A’BC’D+A’BCD+AB’CD’+ABCDs

F(A,B,C,D) =(1⊕*A*)*B*(1⊕*C*)*D*⊕(1⊕*A*)*BCD*⊕*A*(1⊕*B*)*C*(1⊕*D*)⊕*ABCD*

=*BD*⊕*BCD*⊕*ABD*⊕*ABCD*⊕*BCD*⊕*ABCD*⊕*AC*⊕*ACD*⊕*ABC*⊕*ABCD*⊕*ABCD*

=*AC*⊕*BD*⊕*ABC*⊕*ABD*⊕*ACD*⊕*BCD*⊕*BCD*⊕*ABCD*⊕*ABCD*⊕*ABCD*⊕*ABCD*

= *AC* ⊕*BD* ⊕*ABC* ⊕*ABD* ⊕*ACD* ⊕0



**Fig.1 Implementation using Reed-Muller Expansion**

**Pre-lab questions:**

* 1. Mention the similarities between SOP and Polynomial forms.
  2. State the rules for local transformations using EXOR gate.
  3. Give the set of rules that apply on the algebraic operations using exclusive OR operation.

1. Mention the advantages of Reed Muller Expansion technique.

**Post-Lab questions:**

1. Simplify the logic expression using Reed Muller Expansion technique f(A,B,C)=∑m(1,3,5,7)
2. Explain why it is possible to replace OR with EXOR in SOP expressions. Justify
3. What are the Reed-Muller terms which will give Karnaugh maps which contain the maximum number of 1’s which cannot be further simplified.
4. Simplify the logic expression using Reed Muller Expansion technique f(w,x,y,z)=∑m(0,4,6,11,13,15) and verify the output using logisim software.

**Output :**

**Result:**