An Apriori Analysis of Logic Gates (XOR / XNOR) In Reducing the Degree of Complexity in Complex Circuits and Increasing the Efficiency in Decision Making Ways

Dasaka VSS Subrahmanyam[1], K. Venkatesh Sharma[2], and M. Mohan Veer[3]
[1]*Dept. of CSE, Keshav Memorial Engineering College, Hyderabad, Telangana*
[2]*Dept. of CSE, CVR Engineering College, Hyderabad, Telangana*
[3]*Dept. of CSE, Keshav Memorial Engineering College, Hyderabad, Telangana*

***Abstract*—More complex circuits are necessitated to handle more complex logical operations in decision making. To reduce the complexity of operations in logic circuits, more efficient circuits are to be designed. The role of XOR or XNOR gates is to be analyzed in order to reduce the complexity of enhanced logic circuits. The impact of XOR or XNOR gates in logic circuits is to be analyzed as their number is increasing proportionately. Apriori analysis is to be performed in analyzing the nature of operations XOR or XNOR gates.**

***Index Terms* — Apriori, Complexity, Circuits, Decision Trees, Logic Gates, Mathematical Modeling, XNOR, XOR.**

I. INTRODUCTION

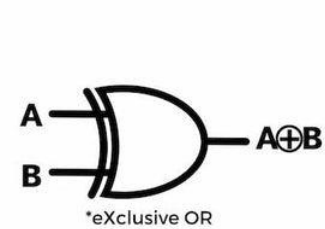
Logic gate provide the basic building blocks of logical operations including addition, subtraction, multiplication, division, and modulus operations. Especially XOR / XNOR gates are used not only in probabilistic computations but also used as decision making tools. XOR gates are used as controlled inverters [2]. XOR is a derivative of OR gate and is used in specific conditions such as either of the two inputs should be high. XNOR is a complement of XOR gate and is used in specific conditions such as the two inputs should be either high or low. Logical circuits become complex as the number inputs kind of XOR / XNOR gates are increasing proportionately. Handling these gates with a minimum number of inputs is easy but analyzing these kinds of gates with a greater number of inputs is difficult [2]. In such conditions an apriori analysis is needed in order to know their impact on the entire logic circuit. Here in this paper, the behavior of complex circuits with XOR / XNOR gates with a greater number of input variables is examined and, in each case, useful output cases are verified accordingly. Thus, the output behavior of logic circuits of XOR / XNOR gates with any number of input variables can be computed and the number of useful output cases can be analyzed [2]. This paper is aimed at reducing the degree of complexity of complex circuits especially with XOR / XNOR gates. The speed of logical operations will be advanced further once the degree of complexity of logic circuits was reduced [2].

A mathematical modeling is needed to formulate the number of useful and non-useful outputs is generated. The same procedure is used for both XOR / XNOR gates, but with different mechanisms [1], as per their different behaviors. Different cases of XOR / XNOR gates with two input variables, three input variables etc., are examined and their behavior is analyzed at every stage. These kinds of circuits are not only useful in digital circuit operations but also in arithmetic operations and data storage [2].

All these logic gates satisfy properties such as commutativity, identity, associativity, complement, idempotent, symmetry, equivalence, realization using other logic gates, using in error detection and correction techniques [1]. Especially, XOR gates are used to determine the parity [2] of a binary number (the o/p of an gate XOR is 1 if the number of 1's in a binary number is odd and the o/p is 0 if the number of 1's in a binary number is even [1]). An analysis of XOR gates in logic circuits, in order to know to know their nature of behavior, with more input variables is necessitated. An apriori analysis can be performed on these gates because of their property of associativity. This property allows any number of input variables irrespective of their input order [2]. It makes the possibility making various logical operations.

II. ANALYSIS AND DESIGN OF LOGIC GATES: MATHEMATICAL MODELING OF GATES

An XOR Gate with two inputs:



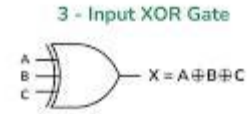
2 input XOR gate

A	B	A⊕B
0	0	0
0	1	1
1	0	1
1	1	0

Figure: 1

In the above Truth table, the number of inputs is 2, son the number decision making ways is 2^2 (the number of decision-making ways is 2^n [1] if there are n inputs). The number of useful decisions is 2, out of the total 4 rows (2 are for keeping decisions True and 2 are for keeping decisions False). These two decision making rows (of True) {(1 0), (0 1)} form a unit matrix of size 2 x 2. The remaining rows (of keeping decisions False) is zero. For a 2-input gate a 2 x 2 unit matrix is formed. Thus, out of 4 rows, half rows are of type True and the remaining half are of type False.

An XOR Gate with 3 inputs with its Truth table:



Truth Table

Input A	Input B	Input C	X = A⊕B⊕C
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Figure: 2

In the above Truth table, the number of inputs is 3, the number decision making ways is 2^3.i.e., 8 The number of ways of keeping decisions True is four. Out of these 8 rows, three rows {(1 0 0), (0 1 0), (0 0 1)} form a unit matrix of type 3 x 3 and one more True decision making row {(1 1 1)} is remained. The remaining number of ways of getting decisions False is four. Thus for a 3-input XOR gate a 4 x 3 matrix is formed, if all True decisions are considered. A 4 x 3 matrix is formed with the remaining decisions making False. Thus out of 8 rows, half rows are of type True and the remaining half are of type False.

An XOR Gate with 4 inputs with its Truth table:

Let the input variables be A, B, C and D. Its Truth table is formed as shown below:

Table: 1

A	B	C	D	o/p
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

In this case, the Truth table has 16 decision making rows. The number of ways of keeping decisions True is 8. Out of these 8 rows, four rows {(1 0 0 0), (0 1 0 0), (0 0 1 0), (0 0 0 1)} form a unit matrix of type 4 x 4 and four other True decision making rows {(0 1 1 1), (1 0 1 1), (1 1 0 1), (1 1 1 0)} form another 4 x 4 matrix (not a unit matrix). Thus an 8 x 4 matrix

is formed by considering all True decision-making rows. The remaining number of ways of getting decisions False is 8. They form another 8 x 4 matrix. In the 4 x 4 matrix of True decision makings, {(0 1 1 1), (1 0 1 1), (1 1 0 1), (1 1 1 0)}, the second row is formed by adding (0 1 0 0) to the first row. The third row is formed by adding (0 0 1 0) to the second row. The fourth row is formed by adding (0 0 0 1) to the third row. Thus, for a 4-input XOR gate an 8 x 4 matrix (includes a 4 x 4-unit matrix) is formed. These matrices are of True decision rows. All the remaining rows of False decision making formed another 8 x 4 matrix. Thus, out of 16 rows, half rows are of type True and the remaining half are of type False.

An XOR Gate with 5 inputs with its Truth table:

Let the input variables be A, B, C, D and E. Its Truth table is formed as shown below:

Table: 2

A	B	C	D	E	o/p
0	0	0	0	0	0
0	0	0	0	1	1
0	0	0	1	0	1
0	0	0	1	1	0
0	0	1	0	0	1
0	0	1	0	1	0
0	0	1	1	0	0
0	0	1	1	1	1
0	1	0	0	0	1
0	1	0	0	1	0
0	1	0	1	0	0
0	1	0	1	1	1
0	1	1	0	0	0
0	1	1	0	1	1
0	1	1	1	0	1
0	1	1	1	1	0
1	0	0	0	0	1
1	0	0	0	1	0
1	0	0	1	0	0
1	0	0	1	1	1
1	0	1	0	0	0
1	0	1	0	1	1
1	0	1	1	0	1
1	0	1	1	1	0
1	1	0	0	0	0
1	1	0	0	1	1
1	1	0	1	0	1
1	1	0	1	1	0
1	1	1	0	0	1
1	1	1	0	1	0
1	1	1	1	0	0
1	1	1	1	1	1

In this case, the Truth table has 32 decision making rows. The number of ways of keeping decisions True is 16. Out of these 16 rows, five rows {(1 0 0 0 0), (0 1 0 0 0), (0 0 1 0 0), (0 0 0 1 0), (0 0 0 0 1)} form a unit matrix of type 5 x 5 and the remaining are six other True decision making rows. The remaining True decision-making rows form another 11 x 5 matrix (not a unit matrix). Thus a 16 x 5 matrix is formed by considering all the True decision-making rows. The remaining number of ways of getting decisions False is 16. They form another 16 x 5 matrix. Thus, for a 5-input XOR gate a 16 x 5 matrix (includes a 5 x 5 unit matrix) is formed. These matrices are of True decision rows. All the remaining rows of False decision making formed another 16 x 5 matrix.

Thus, out of 32 rows, half rows are of type True and the remaining half are of type False.

An XOR Gate with 6 inputs with its Truth table:

In this case, the Truth table has 64 decision making rows. The number of ways of keeping decisions True is 32. Out of these 32 rows, six rows form a unit matrix of type 6 x 6 and the remaining are 26 other True decision-making rows. These 26 True decision-making rows form another 26 x 6 matrix (not a unit matrix). Thus a 32 x 6 matrix is formed by considering all the True decision-making rows. The remaining number of ways of getting decisions False is 32. They form another 32 x 6 matrix. Thus, for a 6-input XOR gate a 32 x 6 matrix (includes a 6 x 6-unit matrix) is formed. These matrices are of True decision rows. All the remaining rows of False decision making formed another 32 x 6 matrix. Thus, out of 64 rows, half rows are of type True and the remaining half are of type False.

IV. ANALYSIS OF XOR LOGIC GATES

Table: 3

no. of i/p	total no. of rows (tr)	true decision rows (tdr)	false decision rows (fdr)	one unit matrix	true rows other than unit matrix	true decision row matrix (includes unit matrix)	false decision row matrix
2	4	2	2	2x2	0	2x2	2x2
3	8	4	4	3x3	1	4x3	4x3
4	16	8	8	4x4	4	8x4	8x4
5	32	16	16	5x5	11	16x5	16x5
6	64	32	32	6x6	26	32x6	32x6
7	128	64	64	7x7	57	64x7	64x7
8	256	128	128	8x8	120	128x8	128x8
9	512	256	256	9x9	247	256x9	256x9
10	1024	512	512	10x10	502	512x10	512x10

V. MATHEMATICAL MODELING

From the above analysis, the following modeling can be drawn, on the basis of the number of inputs to a single XOR gate.

For a given XOR gate with n inputs:

i)total number of decision rows: 2^{n}

ii) total number of True decision rows: 2^{n-1}

iii) total number of false decision rows: 2^{n-1}

iv) number unit matrices formed is one, of type n x n

iv) a matrix of True decision rows: of type (tr/2) x n (includes one unit matrix of type n x n)

v) a matrix of False decision rows: of type (tr/2) x n

vi) total number of rows = true decision rows + false decision rows i.e., tr = tdr + fdr (of each 50%)

vii) a series of True decision rows (excluding unit matrix of type nxn) is formed in the following way:

0, 1, 4, 11, 26, 57, 120, 247, 502, ……..

- ⇨ (0), (1), (4), (11), (26), (57), (120), (247), (502), ……..
- ⇨ $(2^{2-1} - 2)$, $(2^{3-1} – 3)$, $(2^{4-1} - 4)$, $(2^{5-1} - 5)$, $(2^{6-1} – 6)$, $(2^{7-1} - 7)$, $(2^{8-1} - 8)$, $(2^{9-1} - 9)$, $(2^{10-1} - 10)$, $(2^{11-1} - 11)$, ……
- ⇨ i.e., $(2^{n-1} - n)$, where n is the number of inputs variables.

Thus, the number of True decision makings can be found from an n input XOR gate (excluding unit matrix of type n x n i.e., without considering n rows of unit matrix) = $(2^{n-1} - n)$.

viii) Only For True Decision-Making Rows (other than ways formed by unit matrix):

For a 2-input XOR gate:

Only 1-input way is valid.

There are two ways of selecting True decision.

Because of its even number of inputs, and there is no way of selecting two high choices.

For a 3-input XOR gate:

Only inputs with 1-input, and 3-input are valid.

There is only one choice of selecting a True decision, when all inputs are high. Rows with 2-input ways are not valid.

For a 4-input XOR gate:

Only inputs with 1-input, and 3-input are valid.

In addition to the unit matrix of type 4x4, another matrix of type 4x4, of True decisions, is formed with rows {(0 1 1 1), (1 0 1 1), (1 1 0 1), (1 1 1 0)}, all diagonal elements of the matrix are 0's and both lower triangle and upper triangles are only 1's. Its nature is quite opposite to a unit matrix. There is no way of getting a True decision, in case of all high inputs.

For a 5-input XOR gate:

Only inputs with 1-input, 3-input and 5-input are valid.

There are ten ways with 3-inputs. Only one-way of selection with all five inputs.

For a 6-input XOR gate:

Only inputs with 1-input, 3-input and 5-input are valid.

There are twenty 3-input ways and six 5-input ways.

For a 7-input XOR gate:

Only inputs with 1-input, 3-input, 5-input and 7-input are valid. There are thirty-five 3-input ways, twenty-five 5-input ways and one 7-input way.

The same can be extended to any number of inputs to XOR / XNOR gates.

ix) In XOR gates, only odd number of inputs are to be validated for decision making. But in computer science binary tree data structures are more frequently used for decision making domains. More pruning techniques are needed to reduce tree type data structures into binary tree data structures. It can be achieved upon considering Baye's concepts into account. The efficiency of decision-making techniques is mainly depended upon the techniques involved in logic according to the type of data considered and the predicted degree of accuracy. Either XOR or XNOR can be considered for proper decision making in data science and computer Networks. One of the major applications is that it makes advanced predictions in Reuters, which plays a prominent role in providing protection to Network security for cyber-attacks.

VI. CONCLUSION

The same procedure is applicable to even XNOR gates but with different mechanism, by nature of its definition. An apriori analysis of these gates provide the ways of selecting proper decision making and also reduces the complexity of the concerned networks. These are very much practical in their operations when the number of input variables are increasing proportionately. Also useful in Industrial Internet of Things and Network Security where decision making with number of input variables is involved. They provide more applications when these are combined with probability domains. Analysis of XOR / XNOR gates is also very much useful in data science, decision making trees and artificial intelligence domains of computer science.

REFERENCES

[1] M. Morris Mano, Michael D Ciretti, *Digital Design*: Pearson Printice Hall, 4th edition, ch.2, pp.36-63.

[2] Charles H. Roth, Jr, Larry L Kinny, *Fundamentals of Logic Design:* Cengage Learning, ch.2.3,4 pp.27-115.