APPLICATION OF VEDIC SUTRAS IN BINARY ARITHMETIC AND BINARY LOGIC

Muthuselvi Prabhu

Abstract

The binary number system is the perfect numbering system for digital electronics. Manipulation of binary numbers is an essential part of digital systems. Binary logic is different from binary arithmetic. It consists of binary variables and a set of logical operations. The most important part of the binary logic is De Morgan's Theorem. This paper shows how ancient Vedic sutras can be applied in binary arithmetic and binary logic, which is an essential part of the digital system.

Objectives

Manipulation of different base numbers using the sutras, *By Osculation* and *The Remainders by the Last Digit*

Complement of binary numbers using the sutra, *All from 9 and the Last from 10*

De Morgan's law and the complement of a function using the sutra, *Transpose and Apply*

Canonical and standard form under the sutra, *The Product of the Sum is the Sum of the Products*

Three variable circuit simplifications using the sutra, *Vertically and Crosswise*

Manipulation of Numbers

As a human being, we practise numbers with the decimal number system, but machines work only with binary. The conversion of decimal numbers into different base numbers can be done using sutra *The Remainders by the Last Digit.*

To convert to binary numbers, divide the number by 2, write the quotient on the left side and write the remainder under the quotient and repeat this until we get the quotient of 0. The remainder gives the binary equivalent of the given number. To convert to octal and hexadecimal numbers, divide the number by 8 and 16 repeating the same process as above.

Example Convert 255_{10} to binary,

0	1	3	7	15	31	63	127	255
1	1	1	1	1	1	1	1	

$255_{10} = 11111111_2$

Example Convert 609_{10} to octal,

0	1	9	76	609
1	1	4	1	

$609_{10} = 1141_8$

Example Convert 794_{10} to hexadecimal,

0	3	49	794
3	1	10	

In hexadecimal 10 is written as A and so $794_{10} = 31A_{16}$

Since we are familiar with decimal numbers, it is important to know the conversion of different base numbers into decimals and this can be done using the sub-sutra, *By Osculation*.

To convert binary to decimal, 2 is used as the osculator and the osculation is done from left to right to get the decimal equivalent. To convert octal to decimal, 8 is used as the osculator and to convert hexadecimal to decimal, 16 is used as the osculator, osculate from left to right to get the decimal equivalent.

Example Convert 100101001_2 to decimal,

E = 2

1	0	0	1	0	1	0	0	1
	2	4	9	18	37	74	148	297

$100101001_2 = 297_{10}$

Example Convert 3601_8 convert to decimal

E = 8

3	6	0	1
	30	240	1921

$3601_8 = 1921_{10}$

Example Convert $3B2F_{16}$ to decimal

E = 16

3	B	2	F
	59	946	15151

$3B2F_{16} = 15151_{10}$

2's Complement

Complements play an important role in binary arithmetic. In decimals, we use the sutra, *All from 9 and the Last from 10,* to find the complement of a number. In binary, the number before the base number 10 is 1, so *All from 1 and Last from 10_2* can be used to find the complement of a binary number.

Example Find 2's Complement of $1001\ 1111_2$

2's Complement - 0110 0001

Example Find 2's complement of $1100\ 1010_2$

2's Complement - 0011 0110 (don't apply the sutra to the last 0's like decimals)

Binary Logic

Binary logic is different from binary arithmetic as it deals with variables that assume discrete values and with operators that assume logical meaning. The basic operations in binary logic are NOT, AND and OR. Complex digital logic circuits can be built using a few types of basic circuits called "Gates". NOT, AND and OR are the basic logic gates.

Boolean Algebra and De Morgan's Law

Boolean binary algebra is used as a mathematical tool to describe and design complex binary logic circuits. De Morgan's theorem is an essential part of Boolean Algebra and helps to simplify algebra.

De Morgan's Law:

$$\overline{A+B} = \bar{A}\,.\,\bar{B}$$

$$\overline{A\,.\,B} = \bar{A}+\bar{B}$$

In the first law, OR is converted into AND and in the second law AND is converted into OR, hence it comes under sutra *Transpose and Adjust.*

Complement of a Function

The complement of a function can be obtained by using the *Transpose and Adjust* sutra for any number of variables. To find a complement, convert AND into OR, OR into AND, and convert prime into not prime, not prime into prime.

For example, find the complement of the functions ,

$$F_1 = x^1yz^1 + x^1y^1z$$

$$F_1^1 = (x + y^1 + z)\,(x + y + z^1)$$

Canonical and Standard Forms

A binary variable may be either in its normal form or in its complement form. Let us consider 2 variables *x* and *y*. The following table shows the representation of Min terms and Max terms.

X	Y	Min terms	Max terms
0	0	$m_0 = x^1\,y^1$	$M_0 = x + y$
0	1	$m_1 = x^1\,y$	$M_1 = x + y^1$
1	0	$m_2 = x\,y^1$	$M_2 = x^1 + y$
1	1	$m_3 = x\,y$	$M_3 = x_1 + y_1$

The above table shows that the Min terms and Max terms are complements of each other. If there are n Boolean variables, then there will be 2^n Min terms and 2^n Max terms. Boolean functions are expressed as a sum of Min terms (standard product) or product of Max terms (standard sum). The product of Max terms is equal to the sum of Min terms, which is under the sutra, *The Product of the Sum is the Sum of the Products*

Let us consider a function with 3 variables A, B and C. Below is the truth table.

	A	B	C	0	1
0	0	0	0		X
1	0	0	1	X	
2	0	1	0		X
3	0	1	1	X	
4	1	0	0		X
5	1	0	1		X
6	1	1	0	X	
7	1	1	1	X	

Product of Max terms = $(A + B + \bar{C})\ (A + \bar{B} + \bar{C})\ (\bar{A} + \bar{B} + C)\ (\bar{A} + \bar{B} + \bar{C})$

Sum of Min terms = $\bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + A\bar{B}\bar{C} + A\bar{B}C$

Product of Max terms = Sum of Min terms

$$(A + B + \bar{C})\ (A + \bar{B} + \bar{C})\ (\bar{A} + \bar{B} + C)\ (\bar{A} + \bar{B} + \bar{C}) = \bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + A\bar{B}\bar{C} + A\bar{B}C$$

Here, we take the product of Max terms for simplification. Simplify the first two and the last two terms using the sutra, *Vertically and Crosswise*.

$$
\begin{array}{c}
A \;+\; B \;+\; \bar{C} \\
A \;+\; \bar{B} \;+\; \bar{C} \\
\hline
A + A(B+\bar{B}) + A\bar{C} + (B\bar{B}) + \bar{C}(B+\bar{B}) + \bar{C} = A + \bar{C}
\end{array}
$$

$$
\begin{array}{c}
\bar{A} \;+\; \bar{B} \;+\; C \\
\bar{A} \;+\; \bar{B} \;+\; \bar{C} \\
\hline
\bar{A} + \bar{A}\bar{B} + \bar{A}(C+\bar{C}) + \bar{B} + \bar{B}(C+\bar{C}) + C\bar{C} = \bar{A} + \bar{B}
\end{array}
$$

Product of sum

$$(A + B + \bar{C})\ (A + \bar{B} + \bar{C})\ (\bar{A} + \bar{B} + C)\ (\bar{A} + \bar{B} + \bar{C}) = (A + \bar{C})(\bar{A} + \bar{B})$$

Conclusion

This paper shows how the Vedic sutras are applied in binary arithmetic and binary logic. For circuit simplification, the K-map method or Quine-McCluskey methods are used, which uses only the sum of Max terms. Here, we simplified the product of Min terms of 3 variables using *Vertically and Crosswise*, which gives the result as a sum of product. The future focus of this paper is how to apply the Vedic sutras for circuit simplification using more than 3 variables.

References

[1] Bharati Krishna Tirthaji Maharaja, (1994). Vedic Mathematics. Delhi: Motilal Banarasidas

[2] Kenneth R Williams, (2006). Discover Vedic Mathematics. Delhi: Motilal Banarasidas

[3] M.Morris Mano, (2004). Digital Logic and Computer Design. Pearson