IJCSC International Journal of Computer Science & Communication Vol. 1, No. 1, January-June 2010, pp. 97-100

Theory and Applications of the Double-Base Number System Using Bases 2 and 5

Satrughna Singha[1], Supriya Chakraborty[2] & Amitabha Sinha[3]

[1]Department of Computer Science and Engineering, JIS College of Engg., Kalyani, India
[2]Department of Computer Science and Engineering, JIS College of Engg., Kalyani, India
[3]Department of Information Technology, West Bengal University of Technology, India
Email: [1]satrughna.singha@gmail.com, [2]supriya.k6@gmail.com

ABSTRACT

In this paper, we study some of the main properties of the double base number system (DBNS), using bases 2 and 5. A simple geometric interpretation provides the implementation of the arithmetic operations and the addition and multiplication rules of this number systems have been introduced. It can also be shown that the double base number system is used in digital signal processing. Also, the sparseness of the DBNS representation has been emphasized.

Keywords: Double-base Number System, Canonic Double-base Number Representation, Near-canonic Double-base Number Representation.

1. INTRODUCTION

A number system using bases 2 and 5, allowing as digits 0, 1 and requiring O(log N) nonzero digits is the double base number system, i.e., a representation having the form of (1)

$$x = \sum_{i,j} d_{i.j} 2^i 5^j \qquad (1)$$

Therefore, the binary number system is a special case of the above representation. Signal processing algorithms are computationally intensive and therefore, the major issues have been the enhancement of speed of the arithmetic units in general and multiplications and additions in particular. To improve the performance of adders and subtractors, a number of well known schemes have been proposed [1]. Recently a new concept "Double Based Number Systems (DBNS)" [3] have been reported and are becoming attractive for their capabilities of performing multiplication operations efficiently. Table 1 depicts a DBNS table where i and j both range from 0 to 4.

Table 1
DBNS Table for *i* and *j* Ranging from 0 to 4

i / *j*	0	1	2	3	4
0	1	5	25	125	625
1	2	10	50	250	1250
2	4	20	100	500	2500
3	8	40	200	1000	5000
4	16	80	400	2000	10000

In this paper we will deal with canonic (minimal number of nonzero digits). In many applications, the computational complexity of algorithms depends upon the number of zeros of the input data in the corresponding number system [2], [4], [5]. Arithmetic operations in this number system do not guarantee that the results are obtained in the minimal. The canonic number system provides very fast addition and multiplication.

2. PROPERTIES OF THE DBNS

In this section the various properties of the double-base number system have been discussed. The problem of finding the canonic DBNS representation of a given integer is a difficult problem. V. Dimitrov and G. A. Jullien proposed a greedy algorithm [8] which provides the so-called near-canonic double-base number representation (NCDBNR) [7].

Definition 1: The representation of a given integer x into the form:

$$x = \sum_{i,j} d_{i,j} 2^i 5^j, d_{i,j} \in \{0,1\} \qquad (2)$$

will be referred to as a double-base number system(DBNS), using bases 2 and 5. The representation of a given integer as a sum of minimal number 2- integers (numbers of the form $2^i 5^j$,smaller than or equal to x) will be referred to as the canonic double-base number representation(CDBNR) [14].

2.1. Arithmetic Ready Transformations

The method of finding the NCDBNR plays an important role in performing basic arithmetic operations. It can be shown that the nonzero digits be nonconsecutive in our mapping representation; this allows addition to be mapped to a simple Boolean operation. Based on this requirement, we provide the following definition:

Definition 2: A DBNR[14] has no consecutive nonzero digits is defined as an addition ready DBNR(ADRBNR).

2.2. Generalized Reduction

We can generalize the reduction problem using the purely exponential Diophantine equation, [9], [11], [12] (3), where $l < k$.

$$2^{i_1}5^{j_1} + 2^{i_2}5^{j_2} + ... + 2^{i_k}5^{j_k} = 2^{m_1}5^{n_1} + 2^{m_2}5^{n_2} + ...2^{m_l}5^{n_l} \quad (3)$$

Theorem 1: The Diophantine equation[23] $x + y = z$, where GCD(x, y, z) = 1 and x, y, z are 6-integers(that is have the form $2^{x_1}3^{x_2}5^{x_3}7^{x_4}11^{x_5}13^{x_6}$, with $x \geq 0$, $i = 1, 2, 3, 4,$ 5, 6) has exactly 545 solutions [14].

Proof. See [9].

In our case, $x_2 = x_4 = x_5 = x_6 = 0$ and the only solutions of $x + y = z$ are (1, 2, 3), (1, 3, 4), and (1, 8, 9). Therefore, these represent the only three cases where we can replace two active cells with one. An interesting possibility for reducing the active cells follows from the Pillai equation[13]:

$$2^a \pm 2^b = 5^c \pm 5^d \quad (4)$$

Pillai was able to solve all the above four equations, excluding the equation $2^a - 2^b = 5^c - 5^d$ on which he conjectured that the only solution is (7, 3, 3, 1).The conjecture was proven by Stroeker and Tijdeman [13]. For our purposes, only (5) and (6) are relevant:

$$2^a - 2^b = 5^c + 5^d \quad (5)$$

$$2^a + 2^b = 5^c - 5^d \quad (6)$$

Following [13], the solutions of (5) are (2, 1, 0, 0), (3, 1, 1, 0), (5, 1, 1, 2) while (6) has solutions (1, 1, 1, 0) and (4, 2, 2, 1).

3. DBNS-MAP ADDITION AND MULTIPLICATION

In this section the various rules of DBNS-map addition and multiplication methods have been discussed.

3.1. Addition

Let x and y be two integers in the CDBNR. If x and y contain the element 2^i5^j, then the element $2^{i+1}5^j$ does not exist. Actually we perform a reduction into minimal form.

Definition 3: $I_x(i, j)$ Is the DBNS map of the integer x, represented in the ARDBNR.

The image ($I_z(i, j)$ of the DBNS map of the number $z = x + y$ can be obtained using:

$$I_z(i+1, j) = I_x(i, j) \text{ AND } I_y(i, j) \quad \text{Rule (1)}$$

$$I_z(i, j) = I_x(i, j) \text{ XOR } I_y(i, j) \quad \text{Rule (2)}$$

It is to be noted that using the ARDBNR if $I_x(i, j) = I_y(i, j) = 1$, then $I_x(i+1, j) = I_y(i+1, j) = 0$,

$$I_z(i, j+1) = I_z(i, j) \text{ AND } I_z(i+2, j) \quad \text{Rule (3)}$$

$$I_z(i+2, j) = I_z(i+1, j) \text{ AND } I_z(i+1, j) \quad \text{Rule (4)}$$

$$I_z(i+3, j) AND I_z(i+1, j) = I_z(i, j) \text{ AND } I_z(i, j+1) \; AND \; I_z(i+2, j) \quad \text{Rule (5)}$$

$$I_z(i+5, j) = I_z(i+1, j) \text{ AND } I_z(i,j+1) \text{ AND } I_z(i, j+2) \quad \text{Rule (6)}$$

Let us consider the addition of the numbers 136 and 525 using the proposed technique.

The addition operation is presented in Table 2.

Table 2
The DBNS Representation of the Number 136

i / j	0	1	2	3	4
0	1			125	
1		10			
2					
3					
4					

Table 3
The DBNS Representation of the Number 525.

i / j	0	1	2	3	4
0		5			
1					
2		20		500	
3					
4					

Table 4
The DBNS Representation of the Number 616(525 + 136)

i / j	0	1	2	3	4
0	1				
1		10	50		
2			100	500	
3					
4					

Table 5
The DBNS Representation of the Number 616(525 + 136).

i / j	0	1	2	3	4
0	1				
1			50	250	
2		20	100		
3		40	200		
4					

Table 6
The DBNS Representation of the Number 616(525 + 136)

i / j	0	1	2	3	4
0	1		25	125	
1		10			
2				500	
3					
4					

The DBNS representation of the number 616 has three tables. So the example is selected so that the application of the reduction rules based on the solution of (3) gives a result which is not optimal. So from the Pillai Equation [10], and using the above rules the final map is given in Table 7.

Table 7
The Final Map of the Number 661 from Pillai Equation

i / j	0	1	2	3	4
0	1				
1					
2		20			
3					
4					
5					
6					
7		640			

3.2. Multiplication

Let x and y be integers, represented by DBNS maps in the CDBNR[23]. The CDBNR of their product, z, is an n tuple of the elements{ $2^{i_z}5^{j_z} = 2^{i_x+i_y}5^{j_x+j_y}$ },where the $\{i_x, j_x\}$ and $\{i_y, j_y\}$ are the 2 integer index locations of the active cells in the CDBNRs of x and y. Let us consider the multiplication of the numbers 211 and 107, represented via their DBNS maps as shown in Table 4.The representation of the multiplication result is shown in Table 4 and the ARDBNR reduction, using Rule(7) and Rule (8).

$$I_z(i, j+4) = I_z(i+2, j+2) \text{ And } I_z(i, j+3) \text{ AND } (i+4, j+2) \quad \text{Rule (7)}$$

$$I_z(i, j+3) = I_z(i+2, j+1) \text{ AND } I_z(i, j+2) \text{ AND } (i+4, j+1) \quad \text{Rule (8)}$$

Now we consider the example of DBNS multiplication process.

Table 8
The DBNS Representation of the Number 211

i / j	0	1	2	3	4
0	1				
1		10			
2					
3			200		
4					
5					
6					
7					

Table 9
The DBNS Representation of the Number 107

i / j	0	1	2	3	4
0			25		
1	2				
2					
3					
4		80			
5					
6					
7					

Table 10
The DBNS Representation of the Number 22577
(211*107 = 22577)

i / j	0	1	2	3	4
0					625
1	2			250	
2			100		
3		40			2500
4			400		
5		160			
6			1600	8000	
7					
8			6400		

The example is selected so that the application of the reduction rules based on the solution of (3) gives a result which is not optimal. So from the Pillai Equation [10], (4),the final map is given in Table 7. The Reduction Rules Applied Here are Rule (7) and Rule (8).

Table 11
The Reduction Rules Applied here are Rule (7) and Rule (8)

i / j	0	1	2	3	4	5	6
0							15625
1	2		50				
2				500			
3							
4							
5							
6							
7							
8			6400				

4. CONCLUSIONS

In this paper, we have presented the theory of a double base number system using bases 2 and 5 as a technique for representing numbers that allows low complexity arithmetic operations using a variety of implementation media. We have provided a method for arithmetic operations using symbolic substitution. For the addition in the DBNS, we simply have to ensure that there are no consecutive active cells lying in any column on the DBNS map . For multiplication, it can be shown that 2D shifts and additions will be required. We also seek representations with minimal active cells located large 2D hamming distances from each other. Symbolic substitution provides the ability to seek such representation. Our approach produces near canonic forms for the output representation.

REFERENCES

[1] A. Borodin and P. Towari, "On the Decidability of Sparse Univariate Polynomial Interpolation", *Computational Complexity*, **1**, pp. 67-90, 1991.

[2] P. Monotgomery, "A Survey of Modern Integer Factorization Algorithms", *CWI Quarterly*, **7**, no.4, pp. 337-366, 1994.

[3] M.D. Ercegovac, T.Lang, J.G.Nash, and L.P. Chow, "An Area Time Efficient Binary Divider", *IEEE Int'l. Conf. Computer Design*, pp. 645-648, Rye Brook, N.Y.Oct.1987.

[4] P.Kotnerup,"Computer Arithmetic Exploiting Redundancy in Number Representations," *Proc.ASAP '95*,Strasbourg, France, 1995.

[5] A. Avizienis, "Signed-Digit Number Representation for Fast Parallel Arithmetic," *IRE Trans Electronic Computers*, **10**, pp.389–400, 1961.

[7] G.W. Reitwiesner, "Binary Arithmetic,"*Advances in Computing*, **1**, pp231- 308,1960.

[8] G.Hardy, Ramanujan. Cambridge Univ. Press, 1940.

[9] B.M.M.de-Weger, "Algorithms For Diophantine Equations," *CWI Tracts-Amsterdam*, **65**, 1989.

[10] S.S.Pillai, "On the Equation $2^a - 2^b = 3^c - 3^d$," *Bulletin of The Calcutta Math. Soc.*, **37**, pp.15-20, 1945.

[11] A.J.Brentjes, "Multi-Dimensional Continued Fraction Algorithms," *Mathematical Centre Tracts*, Amsterdam, **145**, 1981.

[12] C.Y.Chen, C.-C. Chang, and W.-P. Yang, "Hybrid Method for Modular Exponentiation with Precomputations," *IEE Electronics Letters*, **32**, No. 6, pp.540-541, 1996.

[13] R.J. Stroker and R. Tijdeman, "Diophantine Equations," *Computational Methods in Number Theory, H Lenstra and R. Tijdeman, eds., Math. Centre Tracts-Amsterdam*, **155**, pp. 321-369, 1987.

[14] Vassil S. Dimitrov, Graham A. Jullien, William C. Miller, "Theory and Applications of the Double-Base Number System", *IEEE Transcations on Computers*, **48**, No. 10, 1999.