A New Architecture for Conversion of Multiple Digits Double Base Number System (DBNS) to Single Digit DBNS with Bases 2 and 5 using the Look - up Table (LUT) Technique

Satrughna Singha
Department of Computer Science and Engineering
Maulana Abul Kalam Azad University of Technology
Haringhata, Nadia, West Bengal, India

Amitabha Sinha
Department of Microelectronics and VLSI Technology
Maulana Abul Kalam Azad University of Technology
Haringhata, Nadia, West Bengal, India .

***Abstract*—In this paper a new architecture for conversion of multiple digits double base number system (DBNS) into a single digit double base number system with bases 2 and 5 using the Look-up Table (LUT) approach has been discussed. Among the other non-binary number systems, DBNS is becoming popular for their competences of performing multiplication operation efficiently. In-spite of the various advantages of DBNS, the major problem which is connected with the DBNS is the transformation from multiple digits double base number system (DBNS) into a single digit double base number system. Putting up this matter in view this paper represents a new transformation scheme from multiple digits or multiple index pairs DBNS into a single digit or single index pair DBNS with bases 2 and 5 using the Look-up Table (LUT) technique. In this paper some important properties of the DBNS with bases 2 and 5 have been discussed. Also the DBNS index calculus with bases 2 and 5 has been discussed.**

Keywords—Double Base Number System (DBNS), Look-up Table (LUT), Field Programmable Gate Arrays (FPGAs), Near-canonic Double Base Number Representation (NCDBNR),Index Calculus.

I. Introduction

Nowadays the non-binary number systems like Double Base Number System (DBNS) [1] [2], Residue Number System (RNS) [3], Redundant Complex Number System (RCNS) [4], Complex Logarithmic Number System (CLNS) [5] [6] and Fermat Number System (FNS) [7], are gaining attractions to many researchers due to their abilities of efficient control of arithmetic computations. Digital signal and image processing applications [8] [9] are compute intensive and demand high performance arithmetic operations in real-time [10]. The Double Base Number System (DBNS) with the bases 2 and 5 having the form:

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

where $d_{i,j} \in \{0,1\}$ and i and j are integers and denoted as binary and quinary exponents [1] [11] respectively. The form (1) will be indicated as a double base number system (DBNS). Obviously, the binary number system is an exceptional case of the above illustration. The double base number system is very helpful for signal processing applications. The signal processing application [16] [20] consists of repeated multiplication and addition operations. The multiplication operation is efficiently carried out using DBNS. A number of schemes exist to transform the multiple digits DBNS into a single digit DBNS. To reduce the time complexity, the "Look-Up Table" (LUT) [13] [14] approach has been proposed. However, when the dynamic range is large, the look-up table approach is inefficient [3]. It can also be shown that due to the increase of the dynamic range, there is exponential increase in the size of the look-up table. Observing these matters in view, a novel architecture has been presented for the transformation of multiple digits DBNS into a single digit DBNS with bases 2 and 5. The new concept DBNS using bases 2 and 5 has been reported in [11]. Gradually the DBNS are becoming more popular and attractive. Table 1 describes a DBNS table where i and j both extend from 0 to 3.

Table 1
The DBNS Table for i and j extending from 0 to 3

i / j	0	1	2	3
0	1	5	25	125
1	2	10	50	250
2	4	20	100	500
3	8	40	200	1000

II. The DBNS Properties

In this segment some important properties of the double base number system related to the addition and multiplication operations have been discussed. It is very difficult to find out the canonic double base number representation (CDBNR) of a given number. A greedy algorithm has been proposed [1] [12] to find out a near-canonic double base number representation (NCDBNR). The process of detecting the NCDBNR performs an important role in executing the basic arithmetic operations.

A. The Different Rules of DBNS Addition and Multiplication

In this section the different rules of DBNS addition and multiplication [11] have been presented. Also some examples have been shown for describing the various rules. The addition operation can be performed by covering the corresponding DBNS mappings. The image (D_z (i, j)) of the DBNS map of the number z = x + y can be achieved by using the rules:

(i) $D_z(i+1, j) = D_x(i, j) + D_y(i, j)$.................................Rule 1
(ii) $D_z(i, j+1) = D_x(i, j) + D_y(i+2, j)$............................Rule 2
(iii) $D_z(i+2, j) = D_x(i+1, j) + D_y(i+1, j)$........................Rule 3
(iv) $D_u(i+3, j) + D_v(i+1, j) = D_x(i, j) + D_y(i, j+1) + D_z(i+2, j)$..............................Rule 4
(v) $D_u(i+5, j) = D_x(i+1, j) + D_y(i, j+1) + D_z(i, j+2)$..Rule 5

Let x and y be two integers which have been denoted in the form of CDBNR. The CDBNR of their addition is z which has been an n-tuple of the elements $\{2^{i+1}5^{j} = 2^i5^j + 2^i5^j\}$ for Rule1. The illustration of the multiplication result can be reduced using the Arithmetic Ready Double Base Number Representation (ARDBNR) by the help of Rule 6 and Rule 7 after multiplication.

(vi) $D_u(i, j+4) = D_x(i+2, j+2) + D_y(i, j+3) + D_z(i+4, j+2)$..Rule 6
(vii) $D_u(i, j+3) = D_x(i+2, j+1) + D_y(i, j+2) + D_z(i+4, j+1)$..Rule 7

Let us examine the addition operation of the two numbers 160 and 320 using the above mentioned rules.

Table 2
The DBNS illustration of the number 160

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

Table 3
The DBNS illustration of the number 320

i / j	0	1	2	3	4
0					
1			50	250	
2		20			
3					
4					

Table 4
The DBNS illustration of the number 480 (160+320)

i /j	0	1	2	3	4
0					
1		10	50		
2		20			
3					
4			400		

Table 5
The DBNS illustration of the number 480 (160+320)

i / j	0	1	2	3	4
0					
1		10		250	
2		20			
3			200		
4					

Table 6
The DBNS illustration of the number 480 (160+320)

i / j	0	1	2	3	4
0		5	25		
1		10			
2					
3		40			
4			400		

The DBNS illustration of the number 480 has three tables. So the applications of the reduction rules of the above example have been used depending on the solution of the purely exponential Diophantine equation [1] [18]. The Diophantine equation provides a result which is not optimal. So the final solution can be obtained from the Pillai Equation [1] [18] and by using the above mentioned rules. The final solution has been shown in Table 7.

Table 7
The Final DBNS Solution of the Number 480

i / j	0	1	2		3	4
0						
1						
2						
3						
4		80	400			

III. The DBNS Index Calculus

A number x has been represented as a triple (s_x, b_x, q_x), where s_x is the sign bit, b_x and q_x are integers such that $s_x2^{bx}5^{qx}$ is an adequate estimate of x. More accurately, if $\in$ is the error permitted then $|x - s_x2^{bx}5^{qx}| < \in$. In case of multiplication and division operations the complexity measured is low, namely if $x = (s_x, b_x, q_x)$ and $y = (s_y, b_y, q_y)$ then

$x.y = ((s_x + s_y) \bmod 2, b_x + b_y, q_x + q_y)$ ……… (2)

$x/y = ((s_x + s_y) \bmod 2, b_x - b_y, q_x - q_y)$ .... (3)

The execution of addition and subtraction using the index calculus can be completed using the identities:

$2^a5^b + 2^c5^d = 2^a5^b (1 + 2^{c-a}5^{d-b})$

$\approx 2^a5^b\ \Phi (c-a, d-b)$... (4)

$2^a5^b - 2^c5^d = 2^a5^b (1 - 2^{c-a}5^{d-b})$

$\approx 2^a5^b\ \Psi (c-a, d-b)$... (5)

We will of course pre-compute and store the functions having the approximation of $\Phi(x, y)$ and $\Psi(x, y)$ where $x = c-a$ and $y = d-b$ and $\Phi (x, y)$ and $\Psi (x, y)$ can be expressed as:

$\Phi(x, y) = 1 + 2^x5^y \approx 2^e5^f$ .... (6)

$\Psi(x, y) = 1 - 2^x5^y \approx 2^e5^f$ .... (7)

The addition or subtraction of two numbers has now been described by the following two steps: (1) Find the corresponding element (e, f) in the table and (2) Add (a, b) with (e, f) and finally the values (α, β) have been generated. So the equations (4) and (5) can be rewritten as:

$2^a5^b + 2^c5^d = 2^a5^b (1 + 2^{c-a}5^{d-b})$

$\approx 2^a5^b\ \Phi (c-a, d-b) \approx 2^a5^b . 2^e5^f \approx 2^{a+e}5^{b+f} \approx 2^\alpha 5^\beta$ .... (8)

$2^a5^b - 2^c5^d = 2^a5^b (1 - 2^{c-a}5^{d-b})$

$\approx 2^a5^b\ \Psi (c-a, d-b) \approx 2^a5^b . 2^e5^f \approx 2^{a+e}5^{b+f} \approx 2^\alpha 5^\beta$ .... (9)

IV. The DBNS FIR Filter Simulator

The Inner Product Step Processor (IPSP) [1] [14] [15] has been used as the main hardware circuit for DSP processors. A simulator of the IPSP has been made possible by using the DBNS. The DBNS is useful to validate the operation of the simulator mainly for the word reduction of the ROM [17]. This can also be used to decrease the number of adders for binary and exponent summation operations. A simulator of the IPSP can also be implemented by the help of the DBNS with bases 2 and 5. The simulation examines the IPSP in a multi-tap systolic FIR filter [21] [22] structure using the DBNS with bases 2 and 5 [16]. The data transformation ROM table has been created from a custom integer programming file which chooses binary and quinary exponents a and b on the basis of reducing the error $\in = |x - 2^a5^b|$. The first ten records of the table for integer values of x with ten bits of precision have been shown in table 8.

Table 8
The first ten records of the data transformation ROM

x	a (DB)	b (DQ)
1	0	0
2	1	0
3	55	-23
4	2	0
5	0	1
6	56	-23
7	33	-13
8	3	0
9	38	-15
10	1	1

V. Experimental Results and Analysis

In this segment an architecture of a single index pair DBNS transformation from multiple index pairs (indices) DBNS using index calculus has been described [1]. Including the architecture of multiple index pairs DBNS, the architecture for transforming multiple digits DBNS into a single digit DBNS has also been shown. Clearly, a number X with two index pairs (a, b) and (c, d) has been represented as $X = 2^a5^b + 2^c5^d = 2^a5^b (1 + 2^{c-a}5^{d-b}) = 2^a5^b (\Phi(c-a, d-b)) = 2^a5^b . 2^e5^f$ where, $2^e5^f = 1 + 2^{c-a}5^{d-b} = \Phi(c-a, d-b)$ and $(c-a) = e$ and $(d-b) = f$. Therefore X can be stated as $X = 2^{a+e}5^{b+f} = 2^\alpha 5^\beta$. To calculate the values of α and β it is very important to calculate the value of $\Phi(c-a, d-b)$ which has been kept in an LUT [19]. The procedure can be done by calculating the values of (c-a) and (d-b) by the subtractors. Fig. 1 represents the architecture for multiple index pairs (indices) to single index pair DBNS transformation.

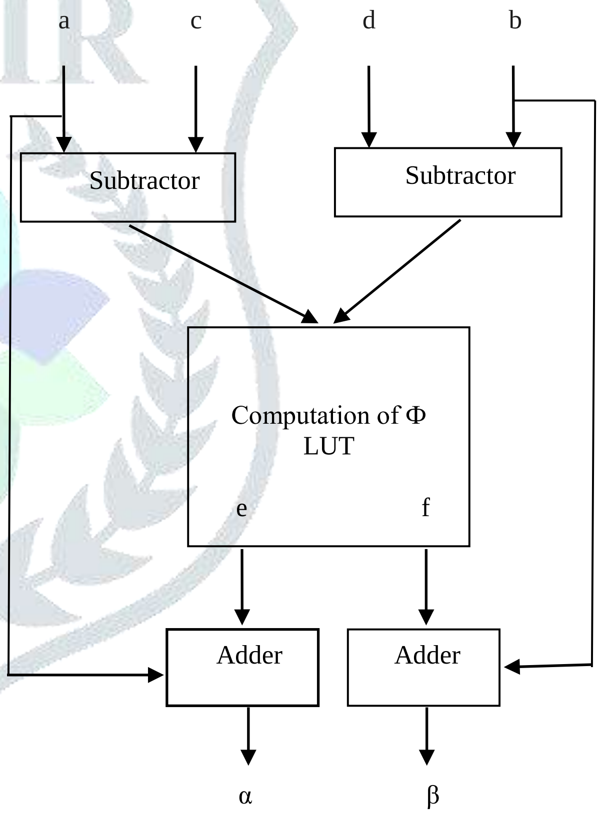

Fig. 1: The Block Diagram of the Architecture of Multiple Index Pairs DBNS into Single Index Pair DBNS.

The block diagram of the complete architecture for transformation for a given data into a single index pair DBNS data has been shown in Fig. 2.

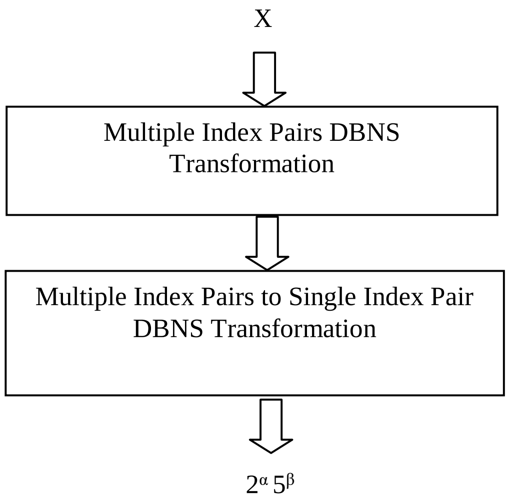

Fig. 2: The Block Diagram of the Complete Architecture for Transformation for a given Number into a Single Index Pair DBNS

The presented architecture can be realized and validated on Xilinx Virtex IV FPGA. Table 9 illustrates the DBNS transformation data for (α, β) of Fig. 1 for single index pair DBNS.

Table 9
The DBNS Transformation Table for (α, β) from Fig. 1 for Single Index Pair DBNS from Multiple Index Pairs DBNS

Sl. No	c-a	d-b	Φ(c-a, d-b)	e	f
1	0	0	2	1	0
2	0	1	6	56	-23
3	1	0	3	55	-23
4	1	1	11	36	-14
5	2	1	21	30	-11
6	1	2	51	-57	27
7	2	0	5	0	1
8	0	2	26	-51	24
9	2	2	101	74	-29
10	0	3	126	-65	31
11	3	0	9	38	-15
12	3	1	41	-55	26
13	3	2	201	-62	30

Now the addition operation of the two DBNS numbers can be described by the following two steps: (1) Find the corresponding element (e, f) from the LUT and (2) Add exponent values (a, b) with (e, f) and finally the exponent values (α, β) have been generated. Then the final single index pair DBNS ($2^{\alpha} 5^{\beta}$) has been produced.

VI. Conclusions

In this paper a new architecture for transformation of a given multiple digits double base number system (DBNS) to a single index pair (i, j) (single digit i.e., powers of 2 & 5) DBNS data has been introduced. In this method the multiple index pairs (indices) are transformed into a single index pair using the look-up table and adders. So only the adders/subtractors and look-up table (LUT) are the primary hardware circuits needed to implement the above mentioned proposed architecture. The comprehensive study in this perspective can be completed in the future work.

References

[1] V. S. Dimitrov, G. A. Jullien, W. C. Miller, (1999), Theory and applications of the double-base number system, IEEE Trans. on Computers, vol. 48, no. 10, pp.1098- 1106.
[2] V. S. Dimitrov, S. Sadeghi-Emamchaie, G. A. Jullien, W. C. Miller, (1996), A near canonic double-based number system (DBNS) with applications in digital signal processing. Proceedings SPIE conference on advanced signal processing, August.
[3] H. Garner. The Residue Number System, IRE Tran. on Electronic Computers, June 1959.
[4] T. Aoki, H. Amanda, T. Higuchi, Real/ Complex Reconfigurable Arithmetic using Redundant Complex Number Systems, Proc. 13th IEEE Symp. on Comp. Arithmetic (ARITH'97), pp. 200 – 207, 1997.
[5] J. Duprat, Y. Herrerous, S. Kla, New Redundant Representation of Complex Numbers and Vectors", IEEE trans. On Computers, Vol. 42, no. 7, pp. 817 – 824, July 1993.
[6] David Lewis, Complex Logarithmic Number System Arithmetic using High-Radix Redundant CORDIC Algorithm, Technical Report, University of Toronto, Ontario, Canada.
[7] Peterson, Ivars, Cracking Fermat Numbers, Science News (March 1): 163, No. 9, 2003. Available at http:// www.sciencenews.org.
[8] Barkley Design Tech. Inc., Evaluating DSP Processors Performance. Http://www.bdti.com
[9] B.G. Lee, (1984), A new algorithm to compute the discrete cosine transforms. IEEE trans. on acoustics, speech and signal processing, vol. Assp-32, pp.1243- 1245, December.
[10] S. Singha and A. Sinha, "Survey of Various Number Systems and Their Applications", International Journal of Computer Science and Communication, Volume-1, Number-1, pp. 73-76, January 2010, Serials Publications, India.
[11] S. Singha and A. Sinha, " Theory and Applications of the Double-Base Number System Using Bases 2 and 5", International Journal of Computer Science and Communication, Volume-1, Number-1, pp. 97-100, January 2010, Serials Publications, India.
[12] S. Singha, A. Ghosh and A. Sinha, "A New Architecture for FPGA based Implementation of Conversion of Binary to Double Base Number System (DBNS) using Parallel Search Technique", ACM SIGARCH Computer Architecture News, Volume 39, Issue 5, pp. 12-18, December 2011, ACM New York, USA.
[13] S. Singha and A. Sinha, "Survey on Reconfigurable Architecture and Computing", International Journal of Electronics Engineering, Volume 9, Issue 1, pp. 165 - 171, January - June 2017, Serials Publications, India.
[14] M. Mukherjee, A. Sinha, A novel architecture for conversion of binary to single digit double base numbers, ACM Sigarch Computer Architecture News, vol. 38, no. 5, December, 2010.
[15] A.Sinha, et al., (2001). Re-configurable parallel architecture for signal/image processing applications, proc. Embedded systems conference, stuttgart, germany, october 9th -11th.
[16] A.Sinha, P. Sinha, K. Newton, K. Mukherjee, multi based number systems for performance enhancement of digital signal processors. Filed for U.S. patent. (U.S. pat. Appl. No.11/488,138), published in U.S. patent

documents serial no.488138, U.S. class at publication 708/620, int'l class: g06f 7/5220060101 g06f007/52, 2006.

[17] Christophe Bobda, Introduction to Reconfigurable Computing: Architectures, Algorithms And Applications, Published by Springer, P.O. Box 17, 3300 Aa Dordrecht, The Netherlands, 2007.

[18] 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.

[19] R. Muscedere, G. A. Jullien, V. S. Dimitrov, W. C. Miller, Efficient techniques for binary-to- multi-digit multidimensional logarithmic number system conversion using range-addressable look-up tables. IEEE Trans. on Computers, vol. 64, no. 3, pp. 257-269, March, 2005.

[20] F. J. Rammig, "A concept for the editing of hardware resulting in an automatic hardware-editor," in proceedings of 14th design automation conference, New Orleans, 1977, pp. 187–193.

[21] G. Estrin and R. Turn, "Automatic Assignment of Computations in a Variable Structure Computer System," IEEE transaction on Electronic Computers, vol. 12, no. 5, pp. 755–773, 1963.

[22] J. Eskritt, R. Muscedere, G. A. Jullien, V. S. Dimitrov, W. C. Miller, (2000), A 2-digit DBNS filter architecture, Proceedings SIPS workshop (Lafayette, L A), October.