Buletinul Ştiinţific al Universităţii "Politehnica" din Timişoara

Seria ELECTRONICĂ şi TELECOMUNICAŢII
TRANSACTIONS on ELECTRONICS and COMMUNICATIONS

Tom 53(67), Fascicola 2, 2008

Binary to Triple Base Number Conversion System- An Efficient Techniques to Convert Binary Number to Triple Base Number.

Amitabha Sinha[1] , Subhashis Maitra[2], Pavel Sinha[3], Ken Newton[4] and Kishanu Mukherjee[5]

Abstract - High computational complexity is an important drawback of different signal processing algorithms and face many challenges in real-time applications .To enhance the speed of different arithmetic units in general and multiplications and additions in particular are therefore the most important issues in the current research areas . Double based number systems (DBNS)[1][11] gains its popularity for their capabilities of handling arithmetic operations efficiently. Even though DBNS schemes exhibit reasonably good performance for 8 bit multiplication , they are not efficient for higher bits that is they are not efficient to cover a large range of numbers .Here we introduce a new concept "Triple based Number Systems (TBNS)[2][3][4][5][12][15] for performance enhancement of the multiplier of the digital signal Processors. The principle of conversion of binary number to TBNS has been dealt here clearly. This number system has been dealt efficiently with in details and a comparison between TBNS and Double Base Number Systems (DBNS) clearly indicate the advantages of the former in terms of speed, hardware complexity and power dissipation. Different architectural models have been proposed.

Index terms: DSP , TBNS, DBNS , LUT.

I. INTRODUCTION

Digital signal processing require very high speed processing on signal data in real-time with a high degree of accuracy and flexibility and low power consumption.

The signal processing algorithms face many challenges in real-time applications because of their high computational complexity. Since most of the DSP algorithms are based on multiplication and additions (some of them dealt with addition efficiently and some other dealt with multiplication), the enhancement of speed of the arithmetic units are the most important issues in the design of the architecture of current signal processor units. To improve the performance of adders and subtractors , a number of well known schemes have been proposed[7][8][9]10][13][14]. How binary number can be converted into TBNS form have been dealt here in an efficient way. After converting a given binary number to TBNS, addition & multiplication need to be performed since these two basic operations are primarily required for most of the signal processing applications. The indices ([i, j, k]) extracted at the time of conversion are used for addition and multiplication in TBNS. Performance of the ALU has been enhanced greatly by introducing an efficient multiplication scheme " TBNS " which is an augmentation of the concept " double based Number Systems".

II. TBNS THEORY

The Triple Base Number System (TBNS) is a special way of representing integers as a sum of mixed powers of two(2) , three(3) and five(5) which is known as three integers. In TBNS , we represent integers in the form as shown in equation-1.

$$x = \sum_{i,j,k} d_{i,j,k}\ 2^i 3^j 5^k \qquad (1)$$

where $d_{i,j,k} = \{0,1\}$

Figure-1 depicts a TBNS table where i, j and k range from 0 to 2. From the expression it is clear that a given binary number when converted into TBNS system can be represented as a number of (i,j,k) pairs.

[1] , [5]West Bengal University of Technology, Salt Lake,Calcutta-700064, India, e-mail: amitabha.sinha @wbut.ac.in
[2]Kalyani Government Engineering College , Kalyani , Nadia, West Bengal, India, e-mail: subhashis07@yahoo.co.in
[3]Concordia University, Montreal, Canada ,e-mail: pavel_sinha@yahoo.com
[4]ESP microDesign , Kutztown, Pennsylvania, U.S.A. , e-mail:knewton@fast.net

These are also referred to as TBNS indices. Greedy algorithm[2] is an iterative approach for computing these indices.

III. CONVERSION USING BOTH BST AND RANGE TABLE SEARCH (HYBRID APPROACH)

The range Table for an 8 bit binary number is shown in Table 1.From this table we see that the co-ordinates in TBNS table depends upon the position of 1's in binary data. From Table 1, we see that when D_7 , the number must be greater than or equal to 128.

This implies that, the number must be larger than 100 but may or may not be larger than 150 , 160 and 225. Hence the first (i,j,k) pair or the coordinate in the TBNS table will be (2,0,2) or (1,1,2) or (2,2,1) or (0,2,2) . Then the number whose coordinate is evaluated is subtracted from the input binary number and the result is again computed with help of range table. This is Greedy Algorithm.

Let us take the example of 215. Its binary representation is 11010111. As D7 = 1, it is compared with (100,150,180,225,300). The first coordinate in terms of [i,j,k] becomes (2,2,1) (coordinate 180) in the TBNS table. Then (10110100) is subtracted and the result becomes 00100011 (35). Since D7 = 0, D6 = 0, D5=1, the no. is compared with 30 or 36 or 45. The co-ordinate in second iteration becomes (1,1,1) (co-ordinate of 30). 30(00011110) is subtracted and the result is 00000101 (5). Hence D7=0, D6=0, D5=0, D4=0, D3=0, D2 = 1, the no. is compared to 4 or 5 or 6. The coordinate in the 3rd iteration becomes (0,0,1) (Co-ordinate of 5). Finally 5 is subtracted and the result be 00000000 thereby ending the conversion process.

IV. ARCHITECTURE OF BINARY TO TBNS CONVERTER

It is clear from the analysis of 3*3*3 TBNS table that maximum of three 3 (i,j,k) pairs are needed to represent an 8 bit binary number. For signal processing applications since the sampled data will arrive at regular interval , pipelined architecture will be the best suitable to exploit the parallelism features. So, a maximum of three(3) Conversion Processing Element (CPE) are employed which are connected in cascade. The block diagram of such a configuration is shown in Figure-2.

Suppose, the time to extract one (i,j,k) pair is T. So, the first binary data will take 3T to represent a TBNS based no. When the partial conversion data for the first input data enters into second stage, the second input binary data enters into the second stage, the second input binary data enters into the first stage of the pipeline and so on. So, after 3T , each binary data will effectively take T to represent the corresponding TBNS base number.

(i,j)		------→ k 0	1	2
		1	5	25
(0,0)	1	1	5	25
(1,0)	2	2	10	50
(1,1)	3	3	15	75
(2,0)	4	4	20	100
(1,1)	6	6	30	150
(0,2)	9	9	45	225
(2,1)	12	12	60	300
(1,2)	18	18	90	450
(2,2)	36	36	180	900

Fig. 1.TBNS table for (i,j,k) ranging from 0 to 2.

Table 1. Range for 8-BIT Number .

Sr. No.	8-bit Data D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Number(N)	(i,j,k)
1	1	X	X	X	X	X	X	X	128≤N	100 or 150 or 180 or 225 or 300
2	0	1	X	X	X	X	X	X	64≤N<128	50 or 60 or 75 or 90 or 100
3	0	0	1	X	X	X	X	X	32≤N<64	25 or 30 or 36 or 45 or 50
4	0	0	0	1	X	X	X	X	16≤N<32	15 or 18 or 20 or 25 or 30
5	0	0	0	0	1	X	X	X	8≤N<16	6 or 9 or 10 or 12 or 15
6	0	0	0	0	0	1	X	X	4≤N<8	4 or 5 or 6
7	0	0	0	0	0	0	1	X	2≤N<4	2 or 3
8	0	0	0	0	0	0	0	1	N=1	1
9	0	0	0	0	0	0	0	0	N=0	

A micro programmed Control unit associated with each CPE to reduce both the hardware and time complexity. The control unit plays a big role in the Binary to TBNS conversion operation. It stores the TBNS based numbers of the 3*3*3 TBNS table in the binary format in the control memory and uses them when signalled by the PE. It is to be noted that the

(i,j,k) pair of a number is stored either in a LUT or in the control memory.

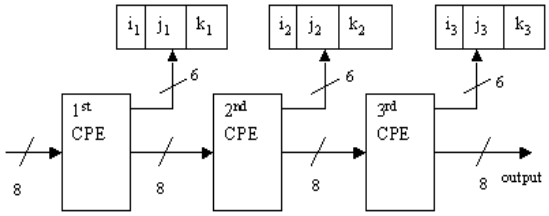

Fig.2. Binary to DBNS Converter.

V. ARCHITECTURE OF CONVERSION PROCESSING ELEMENT (CPE) USING HYBRID APPROACH.

First the data is passed through an 8:3 priority encoder, whose inputs are $D_7 - D_0$ an outputs are Y_2, Y_1, Y_0 and V (Valid bit). The output is shown in Table-II.

Now the three bit output is sent to Control Unit. It checks the conditions and sends the number to compare with incoming data. Here the Control Unit applies the BST algorithm.

Suppose a binary input (X) is encountered by the PE for which $D_7 = 0$ and $D_6=1$, so the number is between 63 and 128. So the output of the encoder is 001 and V=1. The control unit checks the encoder output and sends 90 to the input of 1st comparator for 1st comparison and 75 and 100 to the input of the output multiplexer. If X>90 then lower input of the output multiplexer is enabled and X is compared to 100. If X>100 then the coordinate of 100 is the 1st pair of (i,j,k) of X. If X<90 in the 1st comparison, then upper input of the output multiplexer is enabled and X is compared to 75. If X>75 then 75 is the 1st pair of (i,j,k) . If X<75, Control Unit will send 60 to the input of the 2nd comparator for 1st comparison and 50 and 75 to the input of the output multiplexer. Then same method for comparator1 is repeated. So here at least 2 and at most 4 comparisons are required to extract a pair of (i,j,k) . Then the subtraction is done and the result is sent to the next PE . If zero is encountered, it is easily checked by the valid bit of the priority encoder.

Table 2. Input and Output of the Priority Encoder.

Sr. No.	8-bit Data								Output of the Priority Encoder			
	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Y_2	Y_1	Y_0	V
1	1	X	X	X	X	X	X	X	0	0	0	1
2	0	1	X	X	X	X	X	X	0	0	1	1
3	0	0	1	X	X	X	X	X	0	1	0	1
4	0	0	0	1	X	X	X	X	0	1	1	1
5	0	0	0	0	1	X	X	X	1	0	0	1
6	0	0	0	0	0	1	X	X	1	0	1	1
7	0	0	0	0	0	0	1	X	1	1	0	1
8	0	0	0	0	0	0	0	1	1	1	1	1
9	0	0	0	0	0	0	0	0	X	X	X	0

VI. TIME COMPLEXITY ANALYSIS

To analyse the Time Complexity using the hybrid approach, the following parameters are defined.

Let us assume that,

Memory access time (time to send data from CU to PE included) = t_a

Comparison time = t_c

Delay of the priority encoder = t_e

Delay of multiplexer = t_m

Subtraction time = t_s

So the total time to compute (i,j,k) pairs = $t_H = t_e + 2(t_a + (t_c + t_m + t_c)) + t_s = t_e + 2t_a + 2t_m + 4t_c + t_s$

Figure-3 depicts the architecture of the conversion processing element to convert binary number to TBNS.

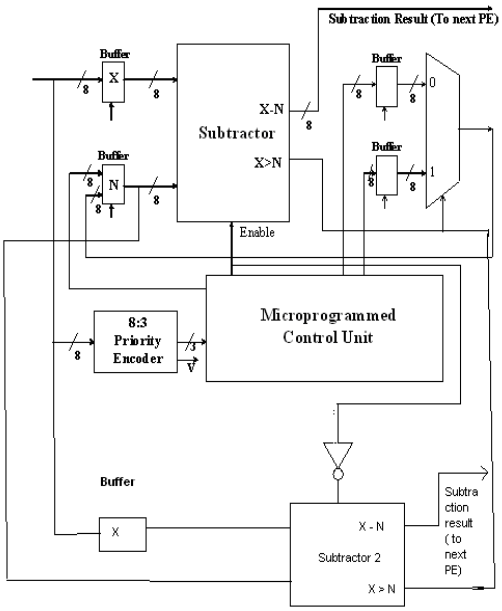

Fig..3. Conversion Processing Element (CPE)

VII. COMPARISON OF THE PROPOSED ARCHITECTURE WITH THE CONVENTIONAL MULTIPLIER.

Figure 4 and 5 show the architecture of a conventional and TBNS multiplier units respectively.

Let us consider fig.4. , where it is assumed that two 8-bit data are to be multiplied.

Let , Propagation delay in the register = t_p ,

Delay in the and gate array = t_{and} ,

Delay in the adder = t_{add}

So total time required to perform the multiplication = $(t_p + t_{and} + t_{add})x\ 8$, since 8-pulses are required to shift the data in the shift register.

Now considering fig.5. , the total time required to perform the multiplication = $T_{CPE} + T_{EXADD} + T_{RCPE}$, where T_{CPE} ,T_{EXADD} and T_{RCPE} are the delay in the CPE , exponent addition and the reverse CPE units respectively.

So it is clear that the time required for two bit multiplication using the proposed architecture is very much less than that required using the conventional one. Though the hardware required for the proposed architecture is more but the same unit can be used both for addition and multiplication.

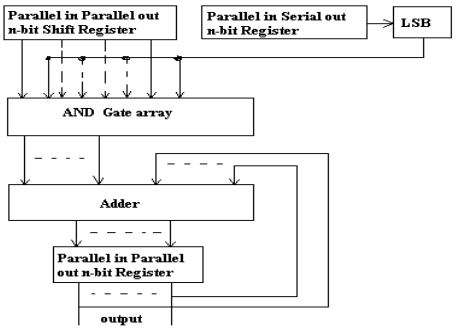

Fig.4. n-bit multiplier.

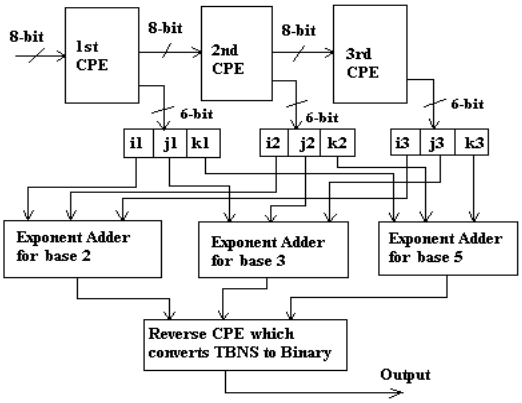

Fig.5. 8-bit TBNS multiplier .

VIII. CONCLUSSION

The aim of this project was to study the suitability of TBNS for implementing a class of signal processing algorithms and to present a number of architecture using spatial and temporal parallelism. A new concept "Triple based Number Systems (TBNS) have been introduced here and a detailed analysis regarding the efficiency of this number system was given. A details analysis on time-complexity of TBNS shows the advantages of TBNS. From the architecture it is clear that TBNS shows its popularity in terms of speed, hardware complexity and power dissipation. Different architectural models have been proposed and a design methodology with small design steps has been used successfully. In telecommunication, Digital signal processing and image processing area 16, 32 or 64-bit data is very natural. For these data, time complexity will be increased due to greater number of comparisons and memory accesses. To keep the time complexity in reasonable limit, hardware complexity should be increased. However, for higher number of data bits, there will be no major change in the architecture of a PE . The priority encoder will be a m: log2m priority encoder. But if the same range of DBNS Table (i=0-3 & j=0-3) is used, a large number of PEs will be required . So, to limit the number of PE, the range of DBNS table must be increased. But to keep the number of PE less, the number of locations in the control memory has to be increased. For 64-bit data the total number of locations for all the PEs will be infinite. So, there is a limit to the number of input data bits in case of DBNS system. There must be a trade-off in between the number of input data bits and number of memory locations required. The TBNS concept can be used for computing the following Digital Signal Processing functions Efficiently.

1) Fast Fourier Transform (FFT)
2) Discrete Cosine transform (DCT)
3) Wave Let Transform
4) To reduce the complexity involved in CDMA Systems, such as Viterbi Decoder & Reed-Solomon Encoder.

REFERENCES

[1] Vassil S Dimitrov, Graham A. Jullien and William C. Miller, Theory and Application of the Double Base Number System, IEEE Transaction on Computers, Vol 48, No.10,Oct. 1999.

[2] A.Avizienis, "Signed-digit Number Representation for Fast Parallel Arithmetic", IRE Trans. Electronic Computer , pp-389- 400, September , 1961.

[3] F.J.Taylor,R.Gill,J.Joseph and J.Radke, A 20-bit Logarithmic Number System Processor, IEEE Trans. Computers,vol-37,pp-190- 200,1988.

[4] D.K.Banerji , J.A. Brzozowski , "Sign Detection in Residue Number Systems", IEEE Transaction , issue:4, C-18, pp-313-320, April,1969.

[5] Jean-Claude Bajard, Laurent-Stephane Didier and P.Kornerup, An RNS Montogomery modular multiplication algorithm, IEEE Trans. On Computers, vol.-47, pp-766-76,July 1998.

[6] Jean-Claude Bajard, Laurent-Stephane Didier and P.Kornerup, Modular Multiplication and Base Extensions in Residue Number System, IEEE Symposium on Computer Arithmatic(2001) , 11-13, June , 2001 Page(s):59 – 65.

[7] A.P.Shenoy and R.Kumaresa, Fast Base Extension using a redundant modulus in RNS , IEEE Trans. On Computers, 38(1989),pp-292-297.

[8] Mary Jane Irwin ,"CSE 575, Computer Arithmetic" , Spring 2003.

[9] V. Paliouras, T. Stouraitis , "Low-power properties of the logarithmic number system" , Proceedings of 15th IEEE Symposium on Computer Arithmetic, 2001. 11-13 June 2001 , Page(s):229 – 236.

[10] J. Eskritt, R. Muscedere, G.A. Jullien, V.S. Dimitrov and W.C. Miller, "A 2-Digit DBNS Filter Architecture", Proceedings SiPS Workshop (Lafayette, LA), October 2000.

[11] V. Dimitrov, Saeid Sadeghi-Emamchaie, G.A. Jullien and W.C. Miller, "A Near Canonic Double-Based Number System (DBNS) with Applications in Digital Signal Processing", Proceedings SPIE Conference on Advanced Signal Processing, August 1996.

[12] G. A. Jullien, V. S. Dimitrov, B. Li, W. C. Miller, A. Lee, and M. Ahmadi, "A Hybrid DBNS Processor for DSP Computation", Proceedings International Symposium on Circuits and Systems, 1999.

[13] R. Muscedere, V.S. Dimitrov, G.A. Jullien, W.C. Miller and M. Ahmadi, "On Efficient Techniques for Difficult Operations in One and Two-digit DBNS Index Calculus", Proceedings 34th Asilomar Conference on Signals, Systems and Computers, November 2000.

[14] R. Tessier and W. Burleson, "Reconfigurable computing for digital signal processing: A survey", Journal of VLSI Signal Processing, Vol28, no.7-27, pp 7-27, 2001.

[15] A. Sinha, et al., "Re-configurable Parallel Architecture for Signal/Image Processing Applications", Proc. Embedded Systems Conference, Stuttgart, Germany, October 9th -11th, 2001.