

Performance Analysis of FPGA Based MAC Unit using DBTNS Multiplier & TRNS Adder for Signal Processing Algorithm

¹Aniruddha Ghosh, ²Amitabha Sinha

¹Calcutta Institute Of Technology, Uluberia, Howrah, W.B, India;

²Birbhum Institute of Engineering & Technology, Suri, Birbhum, W.B, India;

g_aniruddha2003@yahoo.co.in; becas_amits83@hotmail.com;

ABSTRACT

Digital signal processing (DSP) algorithms are actually nothing but sum of product. So, they are computationally intensive. All the mathematical tasks related to DSP algorithm are based on multiplication and addition. So, the implementations of DSP algorithms-based applications extensively require multiplier and adder. Due to extensive use of multiplication and addition operation, speed up cannot be achieved. Designing the high-performance adder and multiplier are primary objective for implementing high performance signal processing applications. This hindrance can be removed by Multiply-Accumulate Unit (MAC). The special feature of a MAC unit is its ability to perform single cycle multiplication and addition operation.

The performance of MAC Unit can be improved by using non-binary number system. Ternary value logic (TVL) has the ability to offer several advantages over conventional binary number system like reduce chip area, reduce overall delay. TVL can switch between three levels, such levels denoted by 0, 1, 2. As Residue Number Systems (RNS) can perform "carry free" arithmetic operations, high performance adder can be implemented using RNS in TVL domain (TRNS). Partial product free multiplication can be implemented by using Double Base Number System in TVL domain (DBTNS). Double Base Ternary Number System (DBTNS) multiplier can perform better than conventional TVL multiplier. A MAC unit is used to perform the multiplication and accumulator operations together to avoid unnecessary overhead on the processor in terms of processing time and the on-chip memory requirements.

Keeping in view of these issues, a new architecture is proposed for implementing high performance MAC unit for DSP applications. In this paper, a new approach of designing MAC unit is instigated using DBTNS multiplier and TRNS adder. A major bottleneck of implementing this architecture is the complexity involved in converting TVL to DBTNS in initial stage and converting TRNS to TVL in final stage. The performance of Ternary Residue Number Systems (TRNS) system depends on selection of moduli because selection of moduli is not properly maintained then it will affect system speed, dynamic range and hardware complexity. Proposed MAC unit is mapped on field programmable gate array (FPGA) for analysis its performance.

Keywords: Ternary Logic Value (TVL), Trit, Ternary Residue Number Systems (TRNS), Double Base Ternary Number System (DBTNS), DBTNS Multiplier, TRNS Adder, Multiply-Accumulate Unit (MAC), FPGA, DSP Algorithms.

1 Introduction

The numeral systems which support three level of switching is termed as Ternary Value Logic (TVL) whose base is 3 and each ternary digit is termed as trit[1]. Volume of Information stored in a trit is

log₃. In Ternary Value Logic (TVL), 0, 1, and 2 are used to represent all numbers [2]. Almost all the Digital Signal Processing applications are mainly nothing but sum of product. In the current scenario, high speed processors having dedicated hardware is one of the main concerns. The enhancement can be made by designing high speed multiplications and additions unit [3]. Multiply-Accumulate (MAC) unit has the capabilities to handle the high-performance digital processing system for Digital Signal Processing (DSP) algorithms [4][5][6]. Significant building blocks of a MAC unit are multiplier, adder and accumulator [5]. In a MAC unit, initially, input data are multiplied then they are added with previously stored accumulator data. Due to extensive use of portable electronic systems like laptop, calculator, mobile etc., and the low power devices have become very popular in today's world [7]. Main aim of VLSI design is to implement low power and high-throughput circuitry design [8][9]. So, implementation of fast and efficient MAC unit is the key objective for real-time signal processing system [10]. This speed can be achieved by enhancing the speed of basic modules of MAC unit by making multiplier unit and adder unit fast. Non-weighted and non-binary number system can help to implement such fast unit [6]. The Residue Number System (RNS) [11] is a non-weighted number system. RNS breaks a large number into the set of smaller number depending on moduli set [12]. Moduli set must be relatively prime to achieve maximum dynamic range [13]. It is shown in various studies that RNS can able to handle fault tolerant, detect and correct the fault [14]. In recent days Residue Number Systems (RNS) [11][15] is attractive owing their competencies of performing carry free addition. So, high performance adder can be implemented using Ternary Residue Number System (TRNS) i.e. RNS using TVL. This enhancement in speed is achieved due to concurrent operations on the moduli. There is another non-weighted number system present, namely, Double Base Number System (DBNS) [16] which can able to perform partial product free multiplication. So, Double Base Ternary Number System (DBTNS) multiplier can help to reduce the complexity of multiplication in compare with the conventional Ternary multiplier. But major bottleneck is the extraction of indices ([i, j] pair) [16][17] when converting ternary number to double base number. For implementing DBTNS conversion, LUT based approach have been adopted. But when dynamic range increases, LUT based approach become incapable to break the complexity as the LUT size increase exponentially. Partial product free multiplication can be performed by DBTNS multiplier so high speed multiply accumulate (MAC) [4][5] units can be implemented using DBTNS multiplier. Keeping these issues in view, this paper presents a new architecture for efficient implementation of MAC units exploiting the potentials of Ternary Residue Number System (TRNS) Adder and Double Base Ternary Number System (DBTNS) Multiplier using Ternary Value Logic (TVL). Performance analysis of a MAC unit using such a scheme clearly indicates the novelty of the architecture. The architecture was implemented and validated on Xilinx Virtex FPGA [16] [17] [18].

2 Review of TVL and RNS

2.1 Review Of Ternary Value Logic (TVL)

In binary system, logic levels are restricted in two state either 0 or 1. So less amount of information is carried by binary logic. To overcome this restriction, the alternative can be suggested as multivalued logic system [2]. But one cannot use unlimited logic level as limitation depends on use of technology [3]. There are three switching state presents in ternary value logic system [19]. Although TVL most often refers to a system in which all whole numbers can be represented by the three switching states i.e 0, 1, and 2 and each ternary digit is denoted by trit [3][19]. In TVL system amount of information can be stored more than binary system. TVL system has some capabilities to enhance processor capacity, accurate processing of signal, less memory required in compare with binary system [2][20]. The

arithmetic operations of TVL Systems are Compliment Operation, Addition, Subtraction, multiplication and division [21][22].

2.2 Review Of Double Base Ternary Number Systems (DBTNS)

An integer can be represented as a sum of mixed powers of two integers, two (2) and three (3) respectively. Mathematically representing technique is termed as Double Base Number System i.e. DBNS [16][23][24]. In the Double-Base number system, an integer, x, can be represented in (1).

$$x = \sum_{i,j} d_{i,j} 2^i 3^j, \quad \text{where } d_{i,j} = \{0, 1\} \tag{1}$$

Form (1), a given binary number can be converted into DBNS as number of (i, j) pair. These are also referred to as DBNS indices [18][25][22]. In the Double-Base number system, when x, is a ternary number then x can be expressed as in (2) which is well elaborated in [24].

$$x = \sum_{i,j} d_{i,j} 2^i 3^j, \quad \text{where } d_{i,j} = \{0, 1, 2\}. \tag{2}$$

These indices (i, j) are in ternary number system. So conversion of a ternary number into DBNS as number of (i, j) pair in TVL domain [19] can be termed as Double Base Ternary Number Systems (DBTNS). Table 1 is representing DBTNS table where trit length of indices (i, j) is 1 and in table 2, trit length of indices (i, j) is 2. Dynamic range for 1 trit is 91 and 2 trit is 5028751.

Table 1. DBTNS Table for i, j → 1 trit

i \ j	0	1	2
0	0001	0010	0100
1	0002	0020	0200
2	0011	0110	1100

Table 2: DBTNS Table for i, j → 2 trit

i, j	00	01	02	10	11	12	20	21	22
00	000000 000001	000000 000010	000000 0000100	000000 0001000	000000 0010000	000000 0100000	000000 1000000	0000001 0000000	0000010 0000000
01	0000000 0000002	0000000 0000020	0000000 0000200	0000000 0002000	0000000 0020000	0000000 0200000	0000000 2000000	0000002 0000000	0000020 0000000
02	0000000 0000011	0000000 0000110	0000000 0001100	0000000 0011000	0000000 0110000	0000000 1100000	0000001 1000000	0000011 0000000	0000110 0000000
10	0000000 0000022	0000000 0000220	0000000 0002200	0000000 0022000	0000000 0220000	0000000 2200000	0000002 2000000	0000022 0000000	0000220 0000000
11	0000000 0000121	0000000 0001210	0000000 0012100	0000000 0121000	0000000 1210000	0000001 2100000	0000012 1000000	0000121 0000000	0001210 0000000
12	0000000 0001012	0000000 0010120	0000000 0101200	0000000 1012000	0000001 0120000	0000010 1200000	0000101 2000000	0001012 0000000	0010120 0000000
20	0000000 0002101	0000000 0021010	0000000 0210100	0000000 2101000	0000002 1010000	0000021 0100000	0000210 1000000	0002101 0000000	0021010 0000000
21	0000000 0011202	0000000 0112020	0000000 1120200	0000001 1202000	0000011 2020000	0000112 0200000	0001120 2000000	0011202 0000000	0112020 0000000
22	0000000 0100111	0000000 1001110	0000001 0011100	0000010 0111000	0000100 1110000	0001001 1100000	0010011 1000000	0100111 0000000	1001110 0000000

2.3 Review Of Ternary Residue Number Systems (RNS)

One of the well-known non-weighted number systems is residue number system (RNS) [11]. A number, X , can be represented in RNS [26] as $X = (x_1, x_2, x_3, \dots, x_N)$ where $x_i = X \text{ modulo } m_i$; where x_i is the i -th residue digit, the i -th modulus is denoted by m_i and all m_i should be mutually prime numbers. The maximum number of different values can be represented by dynamic range, M , as show in (3).

$$M = \prod_{i=0}^N m_i \text{ and } X < M \quad (3)$$

For signed RNS, any integer in the range of $(-M/2, M/2]$, has a unique RNS N tuple representation can be shown by (4).

$$\begin{aligned} x_i &= (X \text{ modulo } m_i) \text{ for } X > 0, \\ &= ((M - |X|) \text{ mod } m_i) \text{ for } X < 0 \end{aligned} \quad (4)$$

One of the major issue related RNS is to select moduli set. To get the maximum dynamic range, moduli set should be relative prime [12][13]. Irregular choice of unbalanced moduli sets prompts inefficient architectures wherein the biggest modulus is too much prevailing regarding both cost and execution. An example of a moduli set with good balance is $\{r^n - 2, r^n - 1, r^n\}$ where r denotes radix or base. So, a good balanced modulus set for TVL is $\{3^n - 2, 3^n - 1, 3^n\}$ [14].

3 Architecture of Proposed MAC Unit Using DBTNS Multiplier and TRNS Adder

DSP algorithms are based on sum of product. So, MAC unit is the best solution for implementing signal processing algorithm because multiplication and accumulation operation can be performed in single cycle by MAC unit [4][5][6]. Main building blocks of a conventional MAC unit are multiplier and accumulator which is used for storing the sum of the previous consecutive products [5][27]. So, multiplier, adder and accumulator are required for implementing MAC unit. Single cycle multiplication and accumulation can be done using this MAC unit [5]. In the architecture of the proposed MAC unit, there are two input $h(n)$ and $x(n)$ which is in TVL. Initially, they are converted into DBTNS and $x(n)$ and $h(n)$ are multiplied using DBTNS multiplier. Then the TVL based product is converted in RNS to perform the carry free addition in TVL domain. The proposed architecture of DBTNS - TRNS mixed base Multiply-Accumulate (MAC) Unit is depicted in the figure 1. So, the following modules are required for implementation of proposed DBTNS - TRNS mixed base MAC unit.

- A. Integer to TVL Conversion
- B. DBTNS Conversion
- C. DBTNS Multiplier
- D. TVL – RNS Conversion
- E. TRNS Adder
- F. Ternary Registrar (TReg)

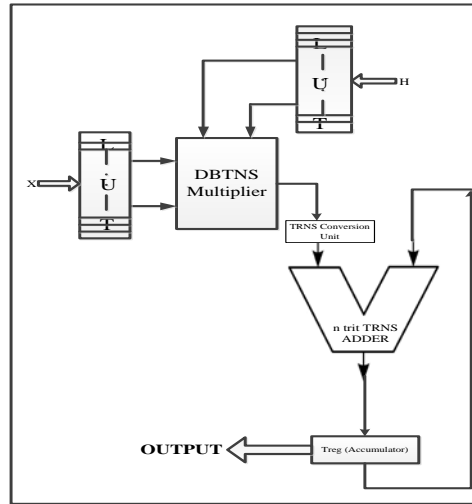


Figure 1. DBTNS-TRNS mixed base MAC Unit

3.1 Integer to TVL Conversion Unit

The conversion of Integer to Ternary Value Logic is carried out by this unit. The approach is totally Look Up Table i.e. LUT [22][28] based. The approach is depicted in figure 2.

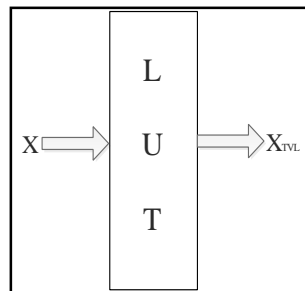


Figure 2. LUT Based TVL Conversion Unit

3.2 DBTNS Conversion Unit

The conversion of Ternary Number to Double Base Ternary Number System is carried out by this unit. The approach is totally Look Up Table (LUT) based [23]. In DBTNS, there are two bases, one is 2 and another is 3 and the number is represented in terms of power 2 and 3 i.e. $X=2^i .3^j$ where these indices (i, j) are in ternary number system [19][28]. Here, the values of i and j are stored in different location of LUT as shown in figure 3. These i and j are used in the consecutive steps.

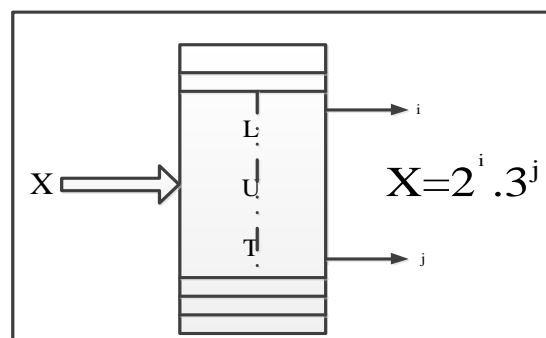


Figure 3. LUT Based DBTNS Conversion Unit

3.3 DBTNS Multiplier Unit

Suppose, X_1 and X_2 are two ternary numbers. In DBTNS, $X_1 = 2^{i_1} \cdot 3^{j_1}$ and $X_2 = 2^{i_2} \cdot 3^{j_2}$. Now, $Z = X_1 \cdot X_2$ then $Z = 2^{(i_1+i_2)} \cdot 3^{(j_1+j_2)}$. The architecture of DBTNS Multiplier [17][28] is depicted in the figure 4. The operation of DBTNS is vividly discussed in [24]. In the architecture of DBTNS multiplier [24], 'n' is the trit length of indices, 'N' is the trit length of ternary equivalent of power of 2 i.e. $2^{(i_1+i_2)}$ and 'M' is the trit length of product. For implementing this architecture, Ternary Adder, Barrel Shifter and LUT are required [24][28][29]. Ternary Ripple Carry Adder (TRCA) is used to add the indices. The ternary equivalent of power of 2 i.e. $2^{(i_1+i_2)}$ is kept in the LUT. This stored data is passed through a barrel shifter as it has ability to perform multi-trit shifting in a single cycle. The amount of shift is defined by the power of 3 i.e. $(j_1 + j_2)$. The multiplied result can be collected from the barrel shifter. Performance analysis of DBTNS multiplier is depicted in the table 3 [24].

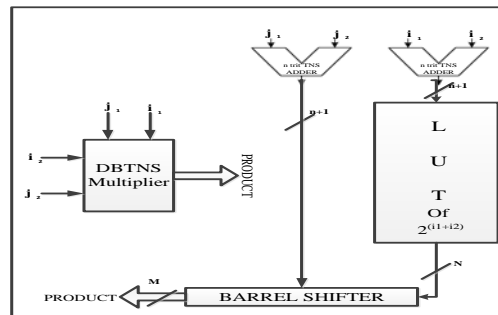


Figure 4. DBTNS Multiplier Unit

Table 3. Data Table of DBTNS Multiplier

INPUT INDEX TRIT LENGTH (i, j) (n)	TNS ADDER		LUT DATA RANGE	BARREL SHIFTER INPUT DATA TRIT LENGTH (N)	MAXIMUM SHIFT	PRODUCT (M)
	OUTPUT TRIT LENGTH (n+1)	OUTPUT DATA RANGE				
1	2	00 to 11	2^0 to 2^4	3	4	7
2	3	000 to 121	2^0 to 2^{16}	11	16	27
3	4	0000 to 1221	2^0 to 2^{52}	33	52	85

3.4 TRNS Conversion Unit

A ternary number X can be represented by residue number system (RNS) in TVL domain [14]. A number, X , can be represented in RNS [26] as $X = (x_1, x_2, x_3, \dots, x_N)$ where $x_i = X \text{ modulo } m_i$; where x_i is the i -th residue digit, the i -th modulus is denoted by m_i and all m_i should be mutually prime numbers. The maximum number of different values can be represented by dynamic range, M , which can be expressed by equation (3). The moduli set $\{r^n - 2, r^n - 1, r^n\}$ are selected for the proposed architecture where r denotes radix or base and for TVL system, the value of r is 3 and $n = 1, 2, 3, 4$ trit. X is a $2n$ trit TVL number. When TRNS conversion for modulus 3^n is performed on X , the generated output is X_{m0} and the computed output is as $X_{m0} = X(n-1 \text{ downto } 0)$. The TRNS conversion for modulus 3^n is shown figure – 5. The TRNS conversion for modulus $(3^n - 1)$ is shown figure 6. Initially, n trit least significant digit is added with n trit most significant digit. As considered earlier, X is a $2n$ trit TVL number. The operation is represented by equation (5).

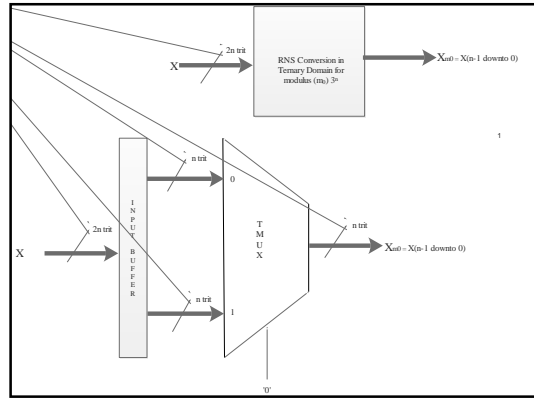


Figure 5. TRNS conversion for modulus 3^n

$$\begin{aligned}
 X_{m1} &= X(2n-1 \text{ downto } n) + X(n-1 \text{ downto } 0) && \text{when carry out} = 0; \\
 &= X(2n-1 \text{ downto } n) + X(n-1 \text{ downto } 0) + '1' && \text{when carry out} = 1; \\
 &= X(2n-1 \text{ downto } n) + X(n-1 \text{ downto } 0) + '2' && \text{when carry out} = 2;
 \end{aligned}
 \tag{5}$$

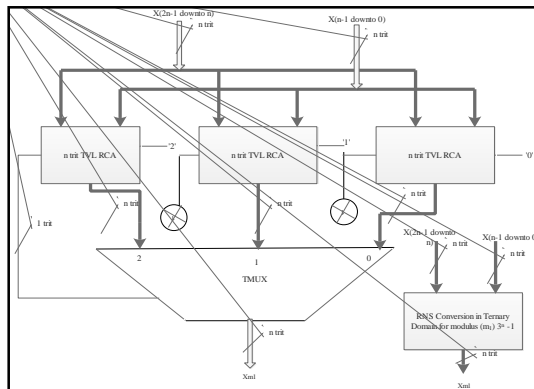


Figure 6. TRNS conversion for modulus 3^n-1

The TRNS conversion for modulus $(3^n - 2)$ is shown figure 7. The operation of this module is described by equation (6).

$$\begin{aligned}
 X_{m2} &= \{2X(2n-1 \text{ downto } n)\} + X(n-1 \text{ downto } 0) && \text{when carry out} = 0; \\
 &= \{2X(2n-1 \text{ downto } n)\} + X(n-1 \text{ downto } 0) + '2' && \text{when carry out} = 1; \\
 &= \{2X(2n-1 \text{ downto } n)\} + X(n-1 \text{ downto } 0) + '4' && \text{when carry out} = 2;
 \end{aligned}
 \tag{6}$$

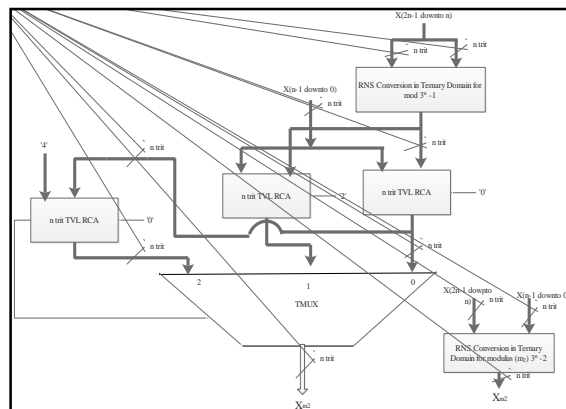


Figure 7. TRNS conversion for modulus 3^n-2

3.5 Multi-trit TRNS Adder

The multi-trit TRNS adder [14][26] is implemented based on carry free RNS adder in ternary value logic (TVL). TRNS adder is implemented using ternary adder, ternary subtractor, ternary logic comparator and ternary multiplexer. TRNS adder is shown in figure 8. To perform RNS addition in ternary value logic can be performed as in equation (7).

$$\begin{aligned} \text{SUM}_{\text{RNS}} &= A_i + B_i && \text{if } A_i + B_i < m_i \\ &= A_i + B_i - m_i && \text{otherwise} \end{aligned} \quad (7)$$

where A_i, B_i are n trit input data and m_i is the i -th modulus and all m_i are mutually prime numbers. Whether the added data ($A_i + B_i$) is lesser or greater that decision is made by logic comparator. Depending on the decision of logic comparator, operation of demux and mux is decided which one have to pass and which have to restrict. Trit length analysis of TRNS adder is described in the table 4 for different trit length of moduli.

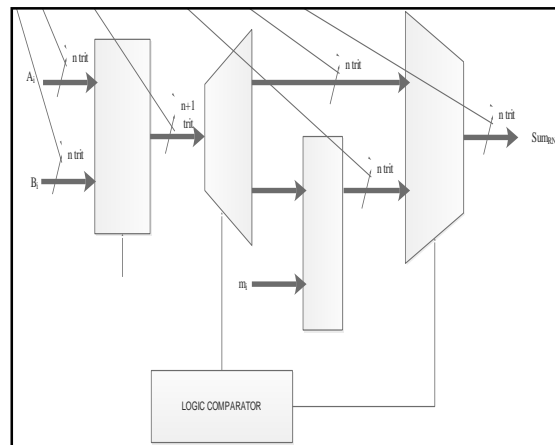


Figure 8. TRNS Adder

Table 4. Data Table of TRNS Adder

INPUT TRIT LENGTH (n)	TVL ADDER		MODULI SET { $r^n - 2, r^n - 1, r^n$ } r: RADIX = 3 (for TVL)	DYNAMIC RANGE (M)	OUTPUT TRIT LENGTH (same as input trit length i.e. n)
	OUTPUT TRIT LENGTH (n+1)	OUTPUT DATA RANGE			
1	2	00 to 11	{1, 2, 3}	6	1
2	3	000 to 121	{7, 8, 9}	504	2
3	4	0000 to 1221	{25, 26, 27}	17,550	3
4	5	00000 to 12221	{79, 80, 81}	5,11,920	4

4 Principle Of Operation Of DBTNS - TRNS Mixed Base MAC Unit For FIR Filter

Multiply-accumulate operation can be performed in single cycle by MAC unit [4][5]. The architecture of the proposed MAC unit is shown in figure 9. There are two inputs, $h(n)$ & $x(n)$ in a MAC unit. Initially they are converted into TVL system of n trit and these TVL converted values are passed through DBTNS converter to generate indices. Then these inputs are passed through DBTNS multiplier to perform multiplication. In the proposed architecture, there are 5 (five) LUTs, among these 5 LUTs, LUT-1 & LUT-

3 are used for integer to ternary number, LUT-2 & LUT-4 are used for converting TNS to double base ternary number and LUT-5 is used to convert ternary to integer. Initially, $x(n)$ and $h(n)$ are converted into DBTNS i.e. $x(n) = 2^{i_1} \cdot 3^{j_1}$ and $h(n) = 2^{i_2} \cdot 3^{j_2}$. The indices of 2 & 3 are passed through DBTNS Multiplier unit and multiplied data is added with zero which is initially stored in an accumulator. Trit length analysis of DBTNS multiplier is cited in the table – 3. Then, this multiplied data is converted into ternary RNS i.e. TRNS of n trit. The moduli set $\{3^n - 2, 3^n - 1, 3^n\}$ are selected for the proposed architecture and $n = 1, 2, 3, 4$ trit. In the proposed architecture, there are 3 (three) LUTs, among these 3 LUTs, LUT-1 & LUT-2 are used for integer to ternary number and LUT-3 is used to convert TRNS to TVL. The multiplied data in TVL is denoted by X . After TRNS conversion X is represented by $\{X_{m_2}, X_{m_1}, X_{m_0}\}$. Here LUT is used for storing the inputs of FIR filter [30][31][32]. Here we are considering the number of tapping is four. FIR algorithm can be implemented using MAC unit. FIR filter can be represented by the equation (8).

$$y(n) = \sum_{k=0}^{N-1} x(n-k) \cdot h(k) \tag{8}$$

The operations of FIR Algorithm are taken place as follows:

- a) To check FIR filter algorithm, filter coefficients ($h(n)$) are stored in a LUT in TVL system. Another input, $x(n)$ is given from another source. $x(n)$ is also in TVL system.
- b) When Program Counter (PC) [6] gets clock, its starts to point an address of LUT, then a data (i.e $h(n)$) sends for performing arithmetic operation with $x(n)$.
- c) The multiplied results are converted in TRNS depending on the moduli set $\{3^n - 2, 3^n - 1, 3^n\}$ where $n = 1, 2, 3, 4$. Trit length analysis of TRNS adder is cited in the table 4 for different trit length of moduli.
- d) RNS addition is performed on these TRNS data.
- e) Initially, Ternary Registrar (TReg) [33] data is zero then TReg starts updating from the very next clock.
- f) After completion of operation, stored data of TReg is converted to TVL data.

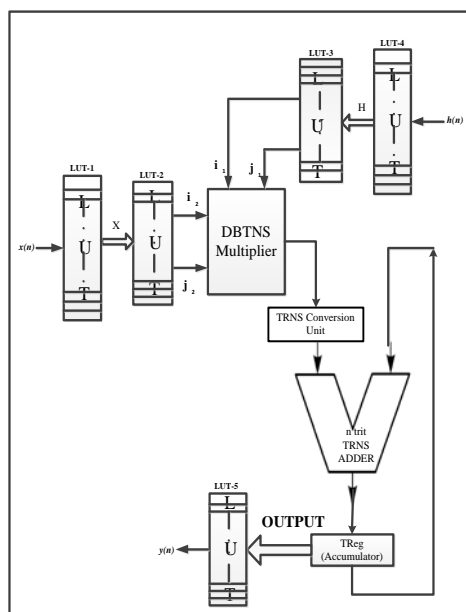


Figure 9. Block diagram of a proposed DBTNS - TRNS mixed base MAC unit

5 Performance Analysis of Proposed DBTNS - TRNS Mixed Base MAC Unit

DBTNS – TRNS based MAC unit is implemented using TVL conversion Unit, DBTNS conversion unit, DBTNS Multiplier Unit, TRNS Conversion Unit, TRNS Adder Unit and Ternary Registrar Unit. The TVL conversion is performed by LUT [3][8] based approach. So over all time complexity [6][20] depends on the LUT size. To implement FIR filter using TRNS MAC unit, total delay can be represented as

(n - trit LUT access time for integer to TVL conversion + n - trit LUT access time for TVL to DBTNS conversion + time taken by DBTNS multiplier + time taken by TRNS conversion + time taken by TRNS Adder + n - trit LUT access time for TNS to integer conversion).

DBTNS multiplier is implemented using ternary Ripple Carry Adder (TRCA) and ternary barrel shifter. Due to single cycle multitrit shifting ability, choosing of barrel shifter can perform. So the time taken by DBTNS multiplier can be calculated as

(time taken by n – trit ternary RCA + time taken by ternary Barrel Shifter)

TRNS Conversion Unit is implemented using ternary Ripple Carry Adder (TRCA) and ternary multiplexer. So the time taken by TRNS conversion unit [14] can be calculated as

(time taken by n – trit ternary RCA + time taken by ternary Multiplexer).

TRNS Adder Unit is implemented using ternary Ripple Carry Adder (TRCA), ternary subtractor and ternary multiplexer and de-multiplexer. So time taken by TRNS Adder Unit can be calculated as

(time taken by n – trit TRCA + time taken by ternary De-multiplexer + time taken by n – trit ternary Subtractor + time taken by ternary Multiplexer).

If the number of trit of input data of FIR filter [4][27][29][30] are changed then execution time is also varied. The synthesis report of 8 tap FIR filter with change of trit is shown in the table 5. The relation between number of LUTs and number of trit, maximum frequency and trit and execution time and trit is shown in the figure 10.

Table 5. Synthesis report of 8 tap FIR filter using DBTNS - TRNS mixed based MAC unit with change of Trit

Sl. No.	Moduli	Input Index Trit Length (i, j)	Synthesis Report				Number of Slice LUTs
			Minimum period	Maximum Frequency	Minimum input arrival time before clock	Maximum output required time after clock	
1.1	7, 8, 9	1	2.439ns	410.071MHz	7.399ns	0.609ns	169 out of 46560
1.2		2	2.439ns	410.071MHz	9.525ns	0.609ns	203 out of 46560
1.3		3	2.439ns	410.071MHz	9.805ns	0.609ns	209 out of 46560
2.1	25, 26, 27	1	2.775ns	360.386MHz	9.390ns	0.595ns	224 out of 46560
2.2		2	2.775ns	360.386MHz	10.979ns	0.595ns	269 out of 46560
2.3		3	2.775ns	360.386MHz	11.809ns	0.595ns	273 out of 46560
3.1	79, 80, 81	1	3.232ns	309.377MHz	11.875ns	0.604ns	285 out of 46560
3.2		2	3.232ns	309.377MHz	12.770ns	0.604ns	366 out of 46560
3.3		3	3.232ns	309.377MHz	13.504ns	0.604ns	379 out of 46560

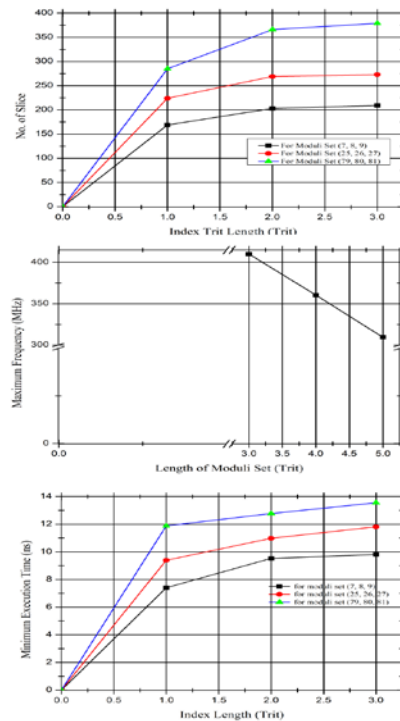


Figure 10. Complexity analysis of 8-tap FIR Filter using DBTNS – TRNS mixed base MAC unit with the change of trit

6 Conclusion

The multivalued logic approach i.e TVL which offers several benefits over existing binary digital system [3][24][34][35]. In this paper, a new architecture for MAC units in ternary value logic (TVL) domain has been proposed for implementing DSP algorithm like FIR algorithm [20][27] using DBTNS multiplier and TRNS adder. TRNS adder can perform carry free addition in ternary domain efficiently and partial product free multiplication operations can be performed by DBTNS multiplier efficiently. Since TRNS adder is efficient compared to conventional TVL adder and DBTNS multipliers are efficient compared [6][28] to conventional ternary multiplier. The novelty of the proposed MAC unit is justified by analyzing the experimental results. The architecture was validated on Xilinx FPGA [18][36][37] and the detailed analysis and studies of different modules of the proposed units have been simulated using Xilinx ISE version 12.3. A detailed study shows improvements on other DSP algorithms [4] like speech processing, high quality sound systems, adaptive echo cancellation, solar signal processing, military applications etc where in addition to high speed, high precisions are also required [34][38] [39].

REFERENCES

- [1] Chung-Yu-Wu., "Design& application of pipelined dynamic CMOS ternary logic & simple ternary differential logic" IEEE journal on solid state circuits, 28: 895-906, 1993.
- [2] S.L. Hurst, "Multiple-valued logic--its status and its future," IEEE Transactions on Computers, vol. C-33, no. 12, pp. 1160-1179, December 1984.
- [3] Reto Zimmermann "Lecture notes on Computer Arithmetic: Principles,Architectures and VLSI Design," Integrated System Laboratory, Swiss Federal Institute of Technology (ETH) Zurich, Mar,16, 1999. URL http://www.iis.ee.ethz.ch/zimmi/publications/comp_arith_notes.ps.gz
- [4] Sanjit K.Mitra "Digital Signal Processing",A Wiley-Inter science Publication,1999.

- [5] Kai Hwang (Purdue University) and Faye A. Briggs (Rice University), "COMPUTER ARCHITECTURE AND PARALLEL PROCESSING", International Edition 1985.
- [6] J. P. Hayes, "Computer Organization", (3rd edition), McGraw-Hill, 1998.
- [7] R.Mariani, F.Pessolano & R.Saletti ' A new CMOS ternary logic design for low power low voltage circuit' Tutorial University of Pisa, Italy.
- [8] Radanovic M. Syrzycki., "Current-mode CMOS adders using multiple-valued logic", Canadian Conference on Electrical and Computer Engineering, 190-193, 1996.
- [9] Gonzalez F, Mazumder P., "Multiple-valued signed digit adder using negative differential resistance devices." *IEEE Trans. on Computers.* 47: 947 - 959, 1998.
- [10] Wei Wang, M.N.S. Swamy, and M.O. Ahmad, "Moduli Selection in RNS for Efficient VLSI Implementation", *IEEE Press, New York*, pages. IV-512 ~ 515, May, 2003.
- [11] Chao-Lin Chiang and Lennart Johnsson, "Residue Arithmetic and VLSI" Presented at 1983 IEEE Internal Conference on Computer Design: VLSI in Computers (ICCCD'83), New York, Oct. 31 - Nov 3, 1983.
- [12] Eep Setiaarif, Pepe Siy, "A New Moduli Set Selection Technique To Improve Sign Detection And Number Comparison In Residue Number System (RNS)", *NAFIPS 2005 - 2005 Annual Meeting of the North American Fuzzy Information Processing Society*, *IEEE Press, New York*, pages 766 ~ 768, 2005.
- [13] Abdallah, M. and A. Skavantzou, "On multi moduli residue number systems with moduli of forms $\{ra, rb-1, rc+1\}$ ", *IEEE Trans. Circuits Syst. I: Regular Paper*, Vol.52, pages 1253-1266, 2005.
- [14] M. Hosseinzadeh and K. Navi, "A New Moduli Set for Residue Number System in Ternary Valued Logic", *Journal of Applied Sciences*, Vol. 7(23), pages 3729-3735, 2007.
- [15] Mandyam S. and Stouraitis T. "Efficient Analog to Residue Conversion Schemes". *IEEE International Symposium on Circuits and Systems*. New Orleans, USA, pages 2885-2888, May 1990.
- [16] V. S. Dimitrov, G. A. Jullien, W. C. Miller, *Theory and Applications of the Double-Base Number System*. *IEEE Trans. Computers*, Vol. 48, 10, pp.1098-1106, 1999.
- [17] Satrughna Singha, Aniruddha Ghosh and Amitabha 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, ACM New York, USA, December 2011. DOI:10.1145/2093339.2093343.
- [18] R. Tessier and W. Bursleson, *Reconfigurable computing for digital signal processing: A survey*, *Journal of VLSI Signal Processing*, Vol 28, no.7-27, pp 7-27, 2001.
- [19] Yoeli M, Rosenfeld G., "Logical Design of ternary switching circuits." *IEEE Trans Computer.*, 14: 19-29, 1965.
- [20] Parhami, B., "Computer Arithmetic: Algorithms and Hardware Designs", 1st Edn., Oxford University Press, Oxford, UK., 2001, ISBN: 0-19-512583-5.
- [21] K.C.Smith 'Multiple Valued Logic: A tutorial & application' *IEEE Tran. Computer* Vol.21 p.p.17-21 April 1988.
- [22] Satrughna Singha and Amitabha Sinha, "Survey of Various Number Systems and Their Applications", *International Journal of Computer Science and Communication*, Volume-1, Number-1, pp. 73-76, Serials Publications, Kurukshetra University, Haryana, India, January 2010.
- [23] V. S. Dimitrov, S. Sadeghi-Emamchaie, G. A. Jullien, 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.

- [24] Aniruddha Ghosh and Amitabha Sinha, "FPGA Implementation of MAC Unit for Double Base Ternary Number System (DBTNS) and its Performance Analysis", International Journal of Computer Applications, Volume 181, Issue 14, pp. 9-22, September 2018, Foundation of Computer Science (FCS), NY, USA, DOI:10.5120/ijca2018917785.
- [25] R. Muscedere, V. S. Dimitrov, G. A. Jullien, W. C. Miller, 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.
- [26] Fred J. Taylor, "Residue Arithmetic: A Tutorial with Examples", IEEE Trans. on Computer, pp. 50~62, May 1984.
- [27] Conway, R. and J. Nelson, "Improved RNS FIR filter architectures. IEEE Trans", Circuits Syst. II: Express Briefs, Vol. 51, pages 26-28, 2004.
- [28] Aniruddha Ghosh, Satrughna Singha and Amitabha Sinha, "A New Architecture for FPGA Implementation of A MAC Unit for Digital Signal Processors using Mixed Number System", ACM SIGARCH Computer Architecture News, Volume 40, Issue 2, pp. 33-38, ACM New York, USA, May 2012. DOI:10.1145/2234336.2234342.
- [29] 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 7/52 20060101 G06F007/52, 2006.
- [30] J. Eskritt, R. Muscedere, G. A. Jullien, V. S. Dimitrov, W. C. Miller, A 2-Digit DBNS Filter Architecture. Proceedings SiPS Workshop (Lafayette, L A), October, 2000.
- [31] G. A Jullien, V. S. Dimitrov, B. Li, W. C Miller, A. Lee, M. Ahmadi, A Hybrid DBNS Processor for DSP Computation. Proceedings International Symposium on Circuits and Systems, 1999.
- [32] Nevio Benvenuto, Lewis E. Franks, and F. S. Hill, Jr. "Realization of Finite Impulse Response Filters Using Coefficients +1, 0 and -1" IEEE Transactions on Communications, vol. COMM-33, no. 10, October 1985.
- [33] Dhande P, Ingole VT., "Design of clocked ternary S-R and D flip-flop based on simple ternary gates.", International journal on software engineering and knowledge engineering, 15: 411417, 2005.
- [34] Gonzalez F, Mazumder P., "Multiple-valued signed digit adder using negative differential resistance devices." IEEE Trans. on Computers. 47: 947 - 959, 1998.
- [35] Radanovic M. Syrzycki., "Current-mode CMOS adders using multiple-valued logic", Canadian Conference on Electrical and Computer Engineering, 190-193, 1996.
- [36] Alireza Kaviani and Stephen Brown,"HYBRID FPGA ARCHITECTURE", FPGA.96, Monterey, CA, Feb.1996, pp. 1-7.
- [37] C.Rozon 'On the use of VHDL as a Multivalued Logic Simulator' Proc. ISMVL 1996,p.p.110-115.
- [38] B. G. Lee, A new Algorithm to compute the discrete cosine transforms. IEEE Trans on Acoustics, speech and signal Processing, vol. ASSP-32, pp.1243- 1245, December, 1984.
- [39] Ping Wah Wong, "Fully Sigma-Delta Modulation Encoded FIR Filters" IEEE Transactions on Signal Processing, vol. 40, no. 6, pp. 1605-1610, June 1992.