Dr. B. R. Sastry,T.Sireesha,B.Gowri Sivanandhini,S.Natarajan / International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622
www.ijera.com Vol. 2, Issue 1,Jan-Feb 2012 pp.538-544

High speed pipelined architecture for cyclic convolution based on FNT

Dr. B. R. Sastry, FIETE, MIEEE, MISTE, SMCSI[*], **T.Sireesha** AMIETE, M.E.(D.S)[**],
B.Gowri Sivanandhini M.E. (D.S)[***], **S.Natarajan** (M.Tech) (E.S)[****]

[*](Department of Computer Science & Engineering, Aurora's Engineering college, Bhongir)
[**](Department of Electronics & Communication Engineering, Aurora's Engineering college, Bhongir)
[***] (Department of Electronics & Communication Engineering, Aurora's Engineering college, Bhongir)
[****] (Department of Electronics & Communication Engineering, Aurora's Engineering college, Bhongir)

Abstract-

This paper presents high speed pipelined architecture for cyclic convolution based on Fermat Number Transform (FNT) in the diminished-1 number system. A code conversion method without addition (CCWA) and a butterfly operation method without addition (BOWA) were used to perform the FNT and its inverse (IFNT). The point wise multiplication in the convolution is accomplished by modulo 2^n+1 partial product multipliers (MPPM) and output partial products which are inputs to the IFNT. Thus modulo 2^n+1 carry propagation additions are avoided in the FNT and the IFNT except their final stages and the modulo 2^n+1 multiplier.

Buffers were used after each and every block in order to store the results of previous block. Introduction of buffers after each block has enabled the architecture to fetch the inputs and outputs at every clock cycle. Compared with the parallel architecture, the proposed one has better throughput and speed

1. INTRODUCTION

The cyclic convolution based on FFT is a widely used operation in signal processing, which needs to be performed in a complex domain even if both the sequences to be performed would be real. Additionally, the dynamic range of the numbers varies widely so that one need to use floating point numbers to avoid scaling and quantization problems. Some architecture for efficient cyclic convolution has been developed to overcome the problems based on Number Theory Transform (NTT). They replace the complex domain with a finite field or a finite residue ring and can be defined by the FFT-like formula. All arithmetic operations are performed modulo *m* and the convolution results are exact without rounding errors. When the modulus in NTT is a Fermat number ($F_t = 2^{2^t}+1$, the *t*th Fermat; *t* is an integer), the NTT turns into the Fermat Number Transform (FNT). The multiplication in the FNT and its inverse (IFNT) can be converted into bit shifts when the transform kernel is 2 or its integer power. Though the modulus of the FNT has a strict relationship with its maximum transform, the cyclic convolution based on FNT is more attractive than the conventional method in some areas.

Most cyclic convolution architectures based on FNT are implemented for the operands in the diminished-1 representation. Thus the first stage includes code conversion (CC) stage which converts the normal binary numbers into their diminished-1 representation. Other arithmetic operations include modulo 2^n+1 negation, addition, subtraction, multiplication operations in the diminished-1 number system. These operations constitute the butterfly operation (BO) which is the most important element in the FNT. The CC and the BO are both mainly composed of modulo 2^n+1 adders of which the fastest one in the diminished-1 number system is proposed by Vergos so far. The fast modulo 2^n+1 adder involves the carry-propagation addition computation and is used in the recent FNT implementations.

In this paper, a code conversion method without addition (CCWA) and a butterfly operation method without addition (BOWA) which take full advantage of the carry-save adder are proposed to accomplish the cyclic convolution with the unity root 2 or its integer power. The modulo 2^n+1 partial product multiplier (MPPM) is used to accomplish the point wise multiplication so that the final carry-propagation addition of two partial product in the multiplier is avoided.

In order to have the pipelined architecture, we modified the parallel architecture [1] by adding buffers after every stage.

Since FNT includes CCWA and BOWA, two buffers were placed in FNT i.e one buffer after CCWA and another after BOWA stage respectively. Similarly, another 2 buffers were placed in IFNT block. Hence a total of 4 buffers were used in the pipelined architecture for cyclic convolution based on FNT operation.

The rest of this paper is organized as follows: the foundations of cyclic convolution based on FNT are formulated in the next section. The important operations in cyclic convolution are presented in section 3. In section 4, the pipelined architecture for cyclic convolution based on FNT is illustrated.

Comparative results that show the efficiency of the proposed architecture against the parallel architecture are presented in section 5 Finally conclusions were presented in section 6.

2. Foundations

The cyclic convolution via the FNT is composed of the FNTs, the point wise multiplications and the IFNT. FNTs of two sequences {*ai*} and {*bi*} will produce two sequences {*Ai*} and {*Bi*}. Modulo 2*n*+1 multipliers are employed to accomplish the point wise multiplication between {*Ai*} and {*Bi*} and produce the sequence {*Pi*}. The final resulting sequence {*pi*} can be obtained by taking the inverse FNT of the product sequence {*Pi*}. Each element in the {*pi*} is in the diminished-1 representation.

The FNT of a sequence of length *N*
{x_i }(i=0,1,… , N-1) is defined as [3]:

$$X_i = \sum_{i=0}^{N-1} x_i \alpha_N^{\langle ik \rangle} \bmod F_t \quad \text{k=0,1,…,N-1)} \quad (1)$$

Where $F_t = 2^{2^t} + 1$, the *t*th Fermat; *N* is a power of 2 and α is an *N*th root unit (*i.e.* $\alpha_N^N \bmod F_t = 1$ and $\alpha_N^N \bmod F_t \neq 1$, 1≤m<N). The notation <*ik*> means *ik* modulo *N*.

The inverse FNT is given by $x_i = \frac{1}{N} \sum_{k=0}^{N-1} X_k \alpha_N^{-\langle ik \rangle} \bmod F_t$ (i=0,1,..,N-1) (2)

where 1/*N* is an element in the finite field or ring of integer and satisfies the following condition:

$$(N.1/N) \bmod F_t = 1 \quad (3)$$

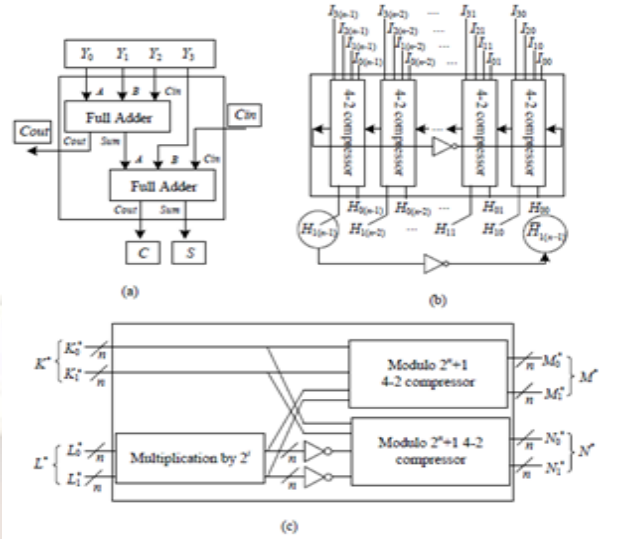

Fig. 1. Elementary operations of FNT architecture with unity root 2 (a) 4-2 compressor (b) Modulo 2^n+14-2 compressor (c) Butterfly operation without addition

Parameters α, F_t , *N* must be chosen carefully. In this paper, we choose α=2, $F_t = 2^{2^t} + 1$ and $N = 2.2^t$ where *t* is an integer.

3. Important operations in cyclic Convolution based on FNT

Important operations of the cyclic convolution based on FNT with the unity root 2 include the CCWA, the BOWA and the MPPM. The CCWA and the BOWA both consist of novel modulo 2^n+1 4-2 compressors mainly which are composed of the 4-2 compressor introduced by Nagamatsu [10]. The 4-2 compressor, the novel modulo 2^n+1 4-2 compressor and the BOWA are shown in Fig. 1. In the figure, "*X**" denotes the diminished-1 representation of *X*, i.e. $X^* = X - 1$.

Introduction of buffers after each stage has enabled the architecture to fetch the outputs at every clock cycle.

3.1 Code conversion without addition

In CC stage, normal binary numbers (NBCs) were converted into their diminished-1 representation as shown in Fig.1(a)&1(b). It is the first stage in the FNT. To reduce the cost, we propose the CCWA that is performed by the modulo 2^n+1 4-2 compressor. Let *A* and *B* represent two operands whose widths are no more than 2*n* bits. We define two new variables:

$$\begin{cases} A = 2^n A_H + A_L \\ B = 2^n B_H + B_L \end{cases} \quad (4)$$

and

$$\begin{cases} M_0 = (2^n - 1) - A_H = \overline{A}_H \\ M_1 = (2^n - 1) - B_H = \overline{B}_H \\ M_2 = (2^n - 1) - B_L = \overline{B}_L \end{cases} \quad (5)$$

If the subsequent operation of CC is modulo 2^n+1 addition, assign A_L, M_0, B_L and M_1 to I_0, I_1, I_2, I_3 in the modulo 2^n+1 4-2 compressor respectively. I_0, I_1, I_2, I_3 are defined as follows:

$$\begin{cases} I_0 = I_{0(n-1)} I_{0(n-2)} \ldots I_{01} I_{00} \\ I_1 = I_{1(n-1)} I_{1(n-2)} \ldots I_{11} I_{10} \\ I_2 = I_{2(n-1)} I_{2(n-2)} \ldots I_{21} I_{20} \\ I_3 = I_{3(n-1)} I_{3(n-2)} \ldots I_{31} I_{30} \end{cases} \quad (6)$$

We obtain the sum vector H_0^* and carry vector H_1^* in the diminished-1 number system. The most significant bit of H_1^* is complemented and connected back to its least significant bit in order to get the diminished-1 results

$$\begin{cases} H_0^* = H_{0(n-1)} H_{0(n-2)} \ldots H_{01} H_{00} \\ H_1^* = H_{1(n-2)} \ldots H_{11} H_{10} \overline{H}_{1(n-1)} \end{cases} \quad (7)$$

The result of modulo 2^n+1 addition of *A** and *B** is equal to the result of modulo 2^n+1 addition of H_0^* and H_1^* . In this way, *A* and *B* are converted into their equivalent diminished-1 representations H_0^* and H_1^* .

Let $\left|A^* + B^*\right|_{2^*+1}$, $\left|\overline{A^*}\right|_{2^*+1}$, $\left|A^* - B^*\right|_{2^*+1}$ and $\left|A^* \times 2^i\right|_{2^*+1}$ denote modulo 2^n+1 addition, negation, subtraction and multiplication by the power of 2 respectively. The CCWA for subsequent modulo 2^n+1 addition can be described as follows:

$$\left|A^* + B^*\right| = \left|A_L + M_0 + B_L + M_1\right|_{2^*+1} = \left|H_0^* + H_1^*\right|_{2^*+1} \quad (8)$$

If the subsequent operation is modulo 2^n+1 subtraction, we assign A_L, M_0, M_2 and B_H to I_0, I_1, I_2, I_3 respectively. Then H_0^* and H_1^* in the modulo 2^n+1 4-2 compressor constitute the result of the CCWA. The conversion is described as follows:

$$\left|A^* - B^*\right|_{2^*+1} = \left|A - B\right|_{2^*+1} = \left|A + \overline{B}\right|_{2^*+1} = \left|A_L + M_0 + M_2 + B_H\right|_{2^*+1} = \left|H_0^* + H_1^*\right|_{2^*+1} \quad (9)$$

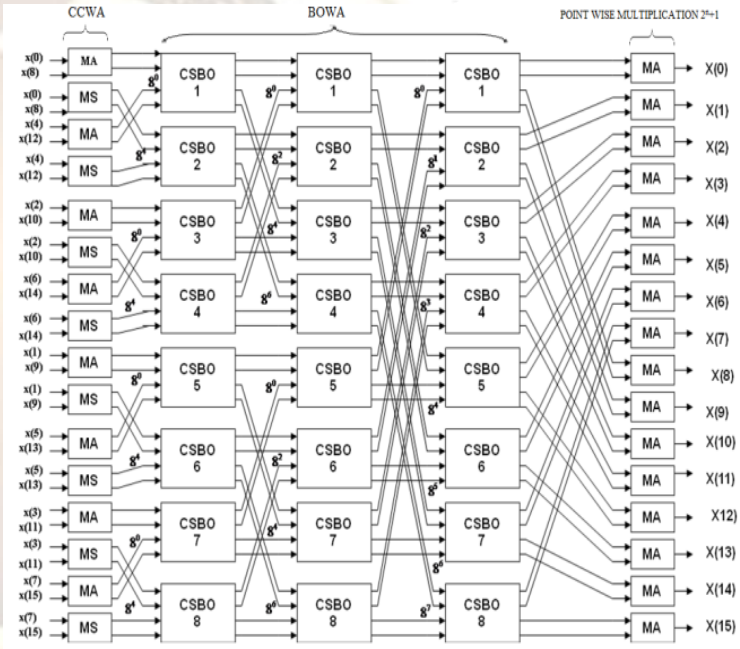

Fig.2 Internal architecture of FNT which includes ccwa,bowa and point wise multiplication 2^n+1

After CCWA, we obtain the result consisting of two diminished-1 numbers. The result also includes the information of modulo 2^n+1 addition or subtraction in the first stage of BO.

3.2 Butterfly operation without addition

After the CCWA, we obtain the results of modulo 2^n+1 addition and subtraction in the diminished-1 representation. Each result consists of two diminished-1 values. The subsequent butterfly operation involves four operands. The proposed BOWA involves two modulo 2^n+1 4-2 compressors, a multiplier and two inverters as shown in Fig. 1(c). The multiplication by an integer power of 2 in the diminished-1 number system in the BOWA is trivial and can be performed by shifting left the low-order *n-i* bits of the number by *i* bit positions then inversing and circulating the high order *i* bits into the *i* least significant bit positions [7].

Thus the BOWA can be performed without the carry-propagation chain so as to reduce the delay and the area obviously. *K**, *L**, *M**, *N** are corresponding to two inputs and two outputs of previous BO in the diminished-1 number system respectively and given by

$$\begin{cases} M^* = \left|M_0^* + M_1^*\right|_{2^n+1} = \left|K_0^* + K_1^* + L_0^* \times 2^i + L_1^* \times 2^i\right|_{2^n+1} = \left|K^* + L^* \times 2^i\right|_{2^n+1} \\ N^* = \left|N_0^* + N_1^*\right|_{2^n+1} = \left|K_0^* + K_1^* - L_0^* \times 2^i - L_1^* \times 2^i\right|_{2^n+1} = \left|K^* - L^* \times 2^i\right|_{2^n+1} \\ \qquad = \left|K^* + \overline{L^* \times 2^i}\right|_{2^n+1} \end{cases} \quad (10)$$

Where $K^* = \left|K_0^* + K_1^*\right|_{2^n+1}, L^* = \left|L_0^* + L_1^*\right|_{2^n+1}$

3.3 Modulo 2^n+1 partial product multiplier

In the proposed pipelined architecture for cyclic convolution based on FNT, the BOWA can accept four operands in the diminished-1 number system. Every point wise multiplication will produce one partial product output. The operation can be accomplished by taking away the final modulo 2^n+1 adder of two partial products in the multiplier. Thus the final modulo 2^n+1 adder is omitted and the modulo 2^n+1 partial product multiplier is employed to reduce the delay and the area.

4. Pipelined architecture for cyclic Convolution

Based on the CCWA, the BOWA and the MPPM, we design the whole pipelined architecture for the cyclic convolution based on FNT as shown in Fig. 2. It includes the FNTs, the point wise multiplication and the IFNT mainly. FNTs of two input sequences {*ai*} and {*bi*} produce two sequences {*Ai*} and {*Bi*} (*i*=1, 2, …*N*- 1). Sequences {*Ai*} and {*Bi*} are sent to *N* MPPMs to accomplish the point wise multiplication and produce *N* pairs of partial products.

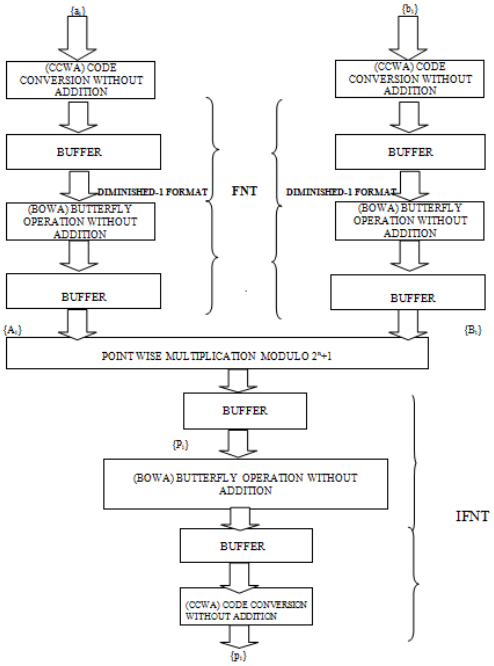

Fig. 3. Pipelined architecture for the cyclic convolution based on FNT

Buffers were used after each and every stage in pipelined architecture in order to store the results of a sequence. Introduction of buffers after each stage has enabled the architecture to fetch the outputs at every clock cycle. Finally the IFNT of the partial products are performed to produce the resulting sequence {*pi*} of the cyclic convolution.

The efficient FNT structure involves $\log_2 N$+1 stages of operations. The original operands are converted into the diminished-1 representation in the CCWA stage, containing the information of modulo 2^n+1 addition or subtraction in the first butterfly operation stage of the previous FNT structure. Then the results are sent to the next stage of BOWA. After $\log_2 N$+1stages of BOWAs, the results composed of two diminished-1 operands are obtained. The final stage of FNT consists of modulo 2^n+1 carry-propagation adders which are used to evaluate the final results in the diminished-1 representation. The CCWA stage, the BOWA stage and the modulo 2^n+1 addition stage in the FNT involves *N*/2 couples of code conversions including

the information of modulo 2^n+1 addition and subtraction, $N/2$ butterfly operations and $N/2$ couple of modulo 2^n+1 additions respectively.

From the definition of FNT and IFNT in section 2, the only difference between the FNT and the IFNT is the normalization factor $1/N$ and the sign of the phase factor α_N. If ignoring the normalization factor $1/N$, the above formula is the same as that given in the FNT except that all transform coefficients $\alpha_N^{\langle ik \rangle}$ used for the FNT need to be replaced by $\alpha_N^{-\langle ik \rangle}$ for the IFNT computation.

5. Comparison and results

Here we compare the results between parallel architecture [1] and the proposed pipelined architecture. We can clearly observe the improvement with regard to time and area over the earlier proposed architecture[1]. Table 1 defines the comparison between parallel and pipelined architecture with regard to fetching of input sequence Ai,Bi. Table 1 also describes the time taken by parallel and pipelined architectures to generate 1st,2nd and 3rd outputs.

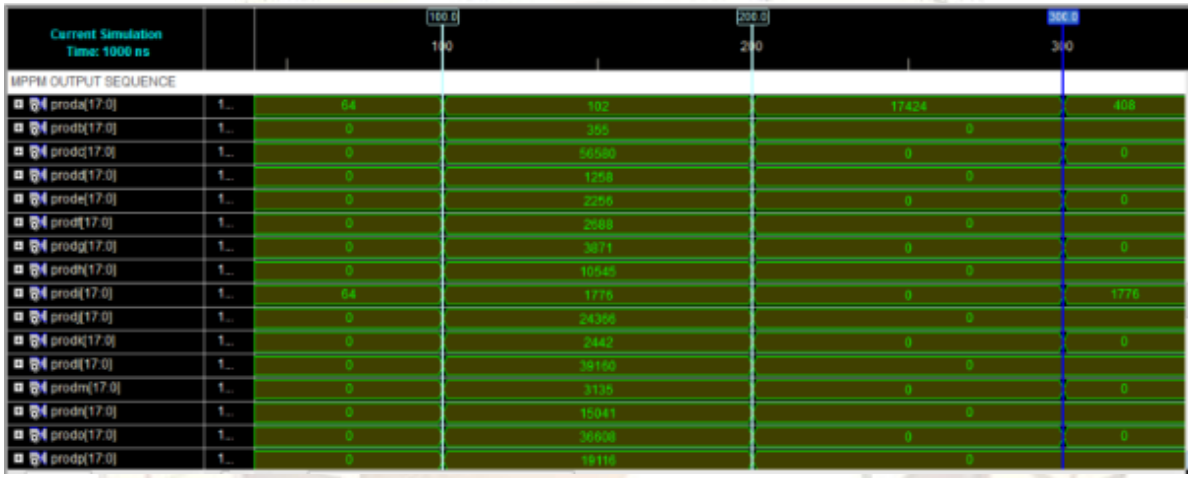

Fig.5.1.Output Sequence from IFNT using parallel architecture

In parallel architecture Fig.5.1, inputs were fetched after every 100nsec. Even outputs were also observed after every 100nsec. Hence one has to wait for 100nsec in order to view the next sequence output

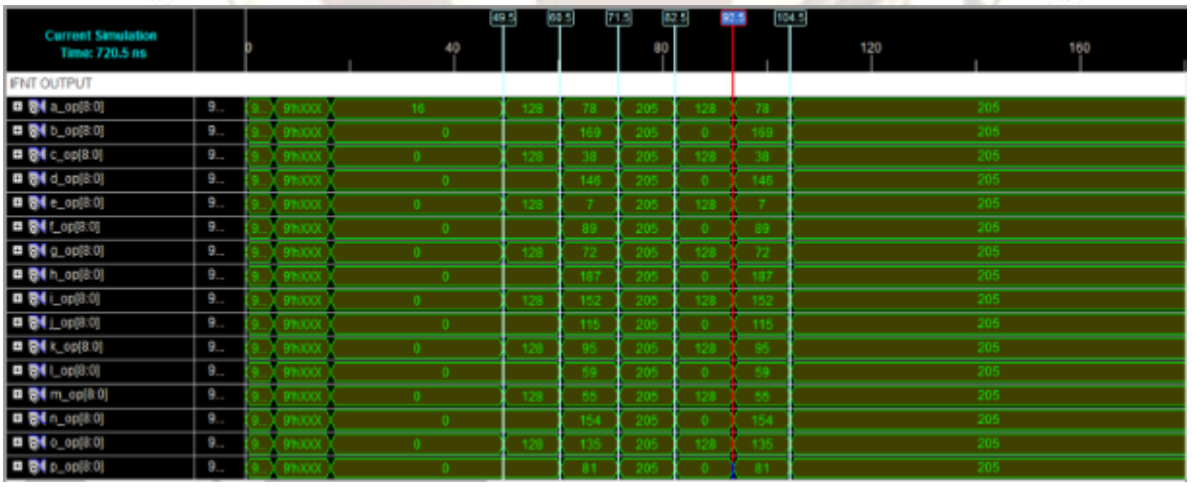

Fig.5.2 Output Sequence from IFNT using pipelined architecture

From Figures.5.1 and 5.2, it is clear that , in parallel architecture outputs were generated after every 100ns i.e 1st sequence output at 100ns, 2nd sequence output at 200ns, 3rd sequence output at 300ns respectively. Where as in pipelined architecture, outputs were generated after every 11nsec i.e 1st sequence output is observed at 49.5ns,2nd sequence output is observed at 60.5ns,3rd sequence output is observed at 71.5ns repectively. Hence with the implementation of pipelined architecture, it is very much clear that the speed has been increased during the generation of the outputs

Table 1

	For parallel architecture	For pipelined architecture
Time taken to fetch the input sequence Ai	100ns	11ns
Time taken to fetch the input sequence Bi	100ns	11ns
Time taken to generate the output sequence for MPPM	100ns	30ns
Time taken to generate the 1st output sequence for IFNT	100ns	49.5ns
Time taken to generate the 2nd output sequence for IFNT	200ns	60.5ns
Time taken to generate the 3rd output sequence for IFNT	300ns	71.5ns

6 Conclusions

A pipelined architecture for the cyclic convolution based on FNT is proposed. The FNT and the IFNT are accomplished by the CCWA and the BOWA mainly. The point wise multiplication is performed by the modulo $2n+1$ partial product multiplier. Thus there are very little modulo 2^n+1 carry-propagation addition compared to the existing cyclic convolution architecture.

Buffers were used after each and every stage in pipelined architecture in order to store the results from the previous block.

Introduction of buffers after each stage has enabled the architecture to fetch the outputs at every clock cycle. Though the operations performed in the pipelined architecture were same as in parallel architecture, we can observe convincing results when the buffers were placed after every stage.

References

[1] Jian Zhang and Shuguo Li "High speed parallel architecture for cyclic convolution based on FNT", IEEE Computer Society Annual Symposium on VLSI, 2009,pp. 199-204

[2] C. Cheng, K.K. Parhi, "Hardware efficient fast DCT based on novel cyclic convolution structures", *IEEE Trans. Signal processing*, 2006, 54(11), pp. 4419- 4434

[3]H.C. Chen, J.I. Guo, T.S. Chang, *et al.*, " A memory efficient realization of cyclic convolution and its application to discrete cosine transform", *IEEE Trans. Circuit and system for video technology*, 2005, 15(3), pp. 445-453

[4] R. Conway, "Modified Overlap Technique Using Fermat and Mersenne Transforms", *IEEE Trans. Circuits and Systems II: Express Briefs*, 2006, 53(8), pp.632 – 636

[5] A. B. O'Donnell, C. J. Bleakley, "Area efficient fault tolerant convolution using RRNS with NTTs and WSCA", *Electronics Letters*, 2008, 44(10), pp.648-649

[6]H. H. Alaeddine, E. H. Baghious and G. Madre *et al.*, "Realization of multi-delay filter using Fermat number transforms", *IEICE Trans. Fundamentals*, 2008, E91A(9), pp. 2571-2577

[7] N. S. Rubanov, E. I. Bovbel, P. D. Kukharchik, V. J. Bodrov, "Modified number theoretic transform over the direct sum of finite fields to compute the linear convolution", *IEEE Trans. Signal Processing*, 1998, 46(3), pp. 813-817

[8] T. Toivonen, J. Heikkila, "Video filtering with fermat number theoretic transforms using residue number system", *IEEE Trans. circuits and systems for video*
technology, 2006, 16(1), pp. 92-101

[9] L. M. Leibowitz, "A simplified binary arithmetic for the Fermat number transform," *IEEE Trans. Acoustics Speech and Signal Processing*, 1976, 24(5):356-359

[10] H. T. Vergos, C. Efstathiou, D. Nikolos, "Diminishedone modulo 2*n* + 1 adder design", *IEEE Trans. Computers*, 2002, 51(12), pp. 1389-1399
[11] M. Nagamatsu, S. Tanaka, J. Mori, et. al. "15-ns 32 × 32-b CMOS multiplier with an improved parallel structure", IEEE Journal of Solid-State Circuits, 1990, 25(2), pp. 494-497.
[12] C. Efstathiou, H. Vergos, G. Dimitrakopoulos, et al., "Efficient diminished-1 modulo 2^n + 1 multipliers", IEEE Trans. Computers, 2005, 54(4), pp. 491-496.
[13] A. Tyagi, "A reduced-area scheme for carry-select adders", IEEE Trans. Computers, 1993, 42(10), pp. 1163-1170.
[14] R. Zimmermann, "Efficient VLSI implementation of modulo (2n ± 1) addition and multiplication", Proc. 14th IEEE Symp. Computer Arithmetic, 1999, pp. 158-167.

Author Biographies

Dr. B. R. Sastry, FIETE, MIEEE, MISTE, SMCSI is a Professor & Dean (Academics) in Aurora's Engineering College, Bhuvanagiri, Nalgonda Dist. He has 30 years of teaching experience in the field of Computer Science in Industry and Engineering and Post-graduate colleges affiliated to various universities. He Guided five Ph.D students in the field of Computer Science.

Ms. T. Sireesha, AMIETE, M.E. (DS) is an Associate professor & HoD (E.C.E-Dept) in Aurora's Engineering College, Bhuvanagiri, Nalgonda Dist. She has 10 years of teaching experience in the area Electronics and Communication Engineering. Her area of research was in the field of VLSI System Design.

Ms. B. Gowri sivanandhini M.E. (DS) is an Associate professor in E.C.E-Dept at Aurora's Engineering College, Bhuvanagiri, Nalgonda Dist. She has 10 years of teaching experience in the area Electronics and Communication Engineering. Her area of research was in the field of Fault Detection in Digital systems.

Mr. S.Natarajan is pursuing his M.Tech Embedded Systems at Aurora's Engineering College, Bhuvanagiri, Nalgonda Dist. His area of interests are Embedded Systems and VLSI.