IMPLEMENTATION OF CYCLIC CONVOLUTION BASED ON FNT

A.Laxman[1], A. Vamshidhar Reddy[2], L.Prakash[3], T.Satyanarayana[4]

[1] *Asst.Prof.,ECE Department, MAHAVEER INST.OF SCIENCE & TECH,, AP, India,* ***amgothlaxman@gmail.com***
[2] *Assoc.Prof.,ECE Department, RRS COLLEGE OF ENG. & TECH.,AP,India,****avamshireddy@gmail.com***
[3] *Asst.Prof.,ECE Department, DVR COLLEGE OF ENG. & TECH., AP, India,* ***laudiya.prakash@gmail.com***
[4] *Asst.Prof.,ECE Department, DVR COLLEGE OF ENG. & TECH, AP, India ,* ***satyant234@gmail.com***

Abstract

Cyclic convolution is also known as circular convolution. It is simpler to compute and produce less output samples compared to linear convolution. There are many architectures for calculating cyclic convolution of any two signals. Implementation using Fermat Number Transform (FNT) is one of them. Fermat Number is a positive integer of the form $F_n = 2^{2^n} + 1$ *where n is a nonnegative integer.The basic property of FNT is that they are recursive.*

This paper presents a cyclic convolution based on Fermat Number Transform(FNT) in the diminished-1 number system.A Code Convolution method Without Addition(CCWA) and a Butterfly Operation method Without Addition(BOWA) are proposed to perform the FNT and its inverse(IFNT) except their final stages in the convolution.The pointwise 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 Modulo*2^n*+1 multiplier.The execution delay of the parallel architecture is reduced evidently due to the decrease of Modulo* 2^n*+1 carry propagation addition.compared with the existing cyclic convolution architecture,the proposed one has better throughput performance and involves less hardware complexity.Synthesis results using 130nm CMOS technology demonstrate the superiority of the proposed architecture over the reported solution.*

Index Terms: *FERMAT NUMBER THEORETIC TRANSFORM, BUTTERFLY ARCHITECTURE, PARALLEL ARCHITECTURE FOR CYCLIC CONVOLUTION, and COMPARISON AND RESULTS*

--***--

1. FERMAT NUMBER THEORETIC TRANSFORM

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+1multipliers 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 {xi} (i= 0,1, …N- 1) is defined as

$$\text{Xk=} \sum_{i=0}^{N-1} x_i \alpha_N^{(ik)} \bmod(n) F_t (k = 0,1....N-1)$$

where Ft=22t+1, the tth Fermat; N is a power of 2 and α is an Nth root unit (i.e. αNN mod Ft=1 and α MN mod Ft ≠1 ,1≤m <N). The notation < ik > means ik modulo N.
The inverse FNT is given by

$$\text{x}_t = \frac{1}{N} \sum_{k=0}^{N-1} x_k \alpha_N^{-(ik)} \bmod F_t (i = 0,1....N-1)$$

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

(N.1/N) mod F_t =1

Parameters α, *Ft*, *N* must be chosen carefully and some conditions must be satisfied so that the FNT possesses the cyclic convolution property. In this project, α=2, Ft=2^{2t}+1 and N=2.2^t where t is an integer.

1.1 Important operations

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 2n+1 4-2 compressors mainly which are composed of the 4-2 compressor introduced by Nagamatsu. The 4-2 compressor, the novel modulo 2n+1 4-2 compressor and the BOWA are shown in Fig.4.1. In the figure, "X*" denotes the diminished-1 representation of X, i.e. X* =X-1.

1.2 Code conversion without addition

The CC converts normal binary numbers (NBCs) into their diminished-1 representation. It is the first stage in the FNT. Delay and area of CC of a 2n-bit NBC are no less than the ones of two n-bit carry propagation adders.

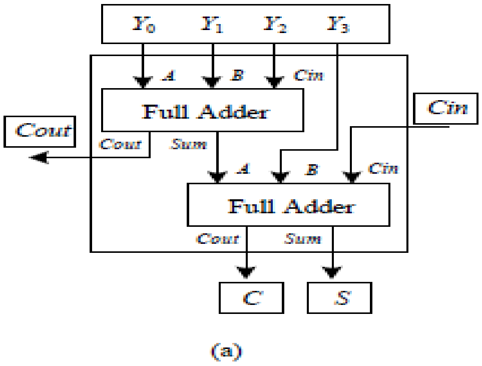


Fig-1 Elementary operations of FNT architecture with unity root 2, (a) 4-2 compressor

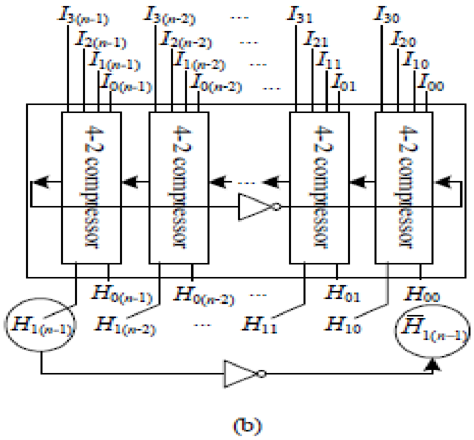


Fig-2 modulo 2n+1 4-2 compressor

To reduce the cost, we propose the CCWA that is performed by the modulo 2n+1 4-2 compressor. Let A and B represent two operands whose widths are no more than 2n bits. We define two new variables:

A = 2n AH +AL
B = 2n BH +BL
M0 = (2n - 1) - AH = ĀH
M1 = (2n - 1) - BH = BH
M2 = (2n - 1) – BL = BL

If the subsequent operation of CC is modulo 2n+1 addition, assign AL, M0, BL and M1 to I0, I1, I2, I3 in the modulo 2n+1 4-2 compressor respectively. I0, I1, I2, I3 are defined as follows:

I0 = I0(n-1) I0(n-2)…….I01I00
I1 = I1(n-1) I1(n-2)…….I11I10
I2 = I2(n-1) I2(n-2)…….I21I20
I3 = I3(n-1) I3(n-2)…….I31I30

We obtain the sum vector H_O^* 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. That is to say

H_0^* = H0(n-1) H0(n-2)……..H01H00

H_1^* = H1(n-2) ……..H11H10 H1(n-1)

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

Let │A* +B*│2n+1, │Ā*│2n+1, │A* - B*│2n+1, and│A* +2i│2n+1 denote modulo 2n+1 addition, negation, subtraction and multiplication by the power of 2 respectively which are proposed by Leibowitz originally.

If the subsequent operation is modulo 2n+1 subtraction, we assign AL, M0, M2 and BH to I0, I1, I2, I3 respectively. Then H_O^* and H_1^* in the modulo 2n+1 4-2 compressor constitute the result of the CCWA.

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

1.3 Diminished-One modulo 2n +1 Adder Design

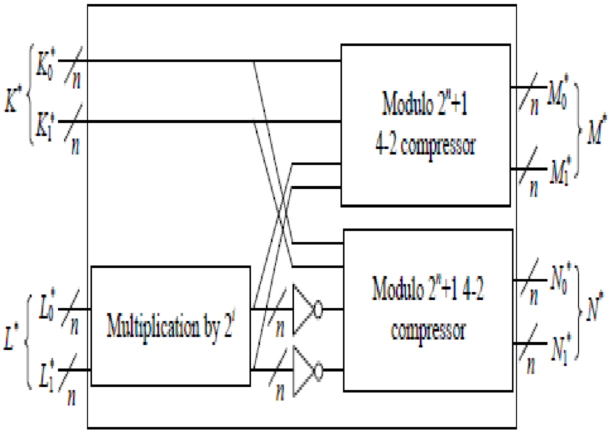
Modulo 2 +1 adders are also utilized as the last stage adder of modulo 2 +1multipliers. Modulo 2 +1multipliers find applicability in pseudorandom number generation, cryptography, and in the Fermat number transform, which is an effective way to compute convolutions Leibowitz has proposed the diminished-one number system. In the diminished-one number system, each number X is represented by X =X-1. The representation of 0 is treated in a special way. Since the adoption of this system leads to modulo 2 +1 adders and multipliers of n bits wide operands, it has been used for many residue number system implementations; Efficient VLSI implementations of modulo 2 +1 adders for the diminished-one number system have recently been presented. The adders although fast, are, according to the comparison presented, still slower than the fastest modulo 2 adders or the fastest modulo 2 -1 adders

In this project, we derive two new design methodologies for modulo 2 +1 adders in the diminished-one number system. The first one leads to traditional Carry Look-Ahead (CLA), while the second to parallel-prefix adder architectures. Using implementations in a static CMOS technology, we show that the proposed CLA adder design methodology leads to more area and time efficient implementations than those presented, for small operand widths. For wider operands, the proposed parallel-prefix design methodology leads to considerably faster adder implementations than those presented and as fast as the integer or the modulo 2 +1 architecture presented.

2. BUTTERFLY ARCHITECTURE

2.1 Butterfly operation without addition

After the CCWA, we obtain the results of modulo 2n+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 2n+1 4-2 compressors, a multiplier and some inverters.



.

Fig-3 Butterfly operation without addition

The multiplication by an integer power of 2 in the diminished-1 number system in the BOWA is trivial and can be performed by left shifting 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. 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

M*= $|M_0^* + M_1^* \;|_{\cdot 2^n+1}$=

$|K_0^* + K_1^* + \; L_0^* \, X\, 2^t + \; L_1^* \, X\, 2^t|_{\cdot 2^n+1}$=

$|K_{.}^* + \; L_0^* \, X\, 2^t|_{\cdot 2^n+1}$

N*= $|N_0^* + N_1^* \;|_{\cdot 2^n+1}$=

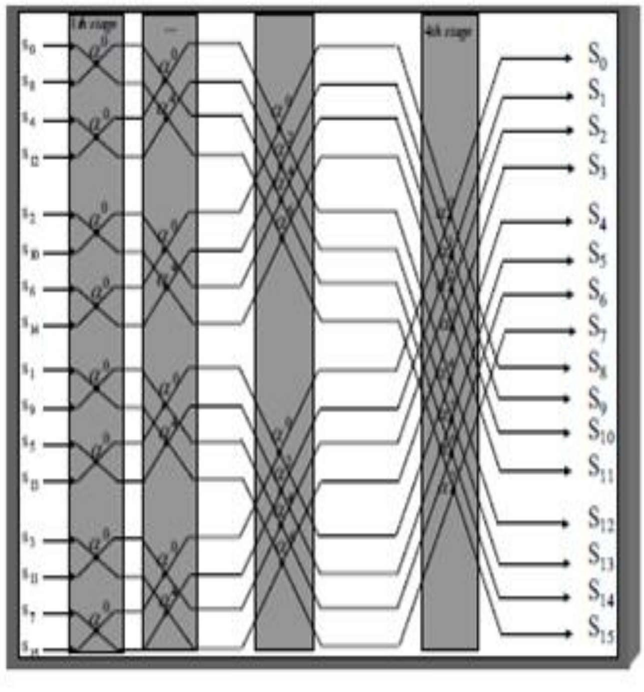
$|K_0^* + K_1^* - \; L_0^* \, X\, 2^t - \; L_1^* \, X\, 2^t|_{\cdot 2^n+1}$=

$|K_{.}^* - \; L_0^* \, X\, 2^t|_{\cdot 2^n+1}$=$|K_{.}^* + \; L_0^* \, X\, 2^t|_{\cdot 2^n+1}$

Where $K_{.}^* = |K_0^* + K_1^*|_{\cdot 2^n+1}$, $L_{.}^* = |L_0^* + L|_{\cdot 2^n+1}$

2.2 The FNT Architecture

In the previous sections, we have presented the reconfiguration at a rather low level. The Butterfly constitutes a high parameterized function level. The fact to have this parameterized function allows designing a reconfigurable operator who's Butterfly forms the highest level operator. Figure 4 depicts the global reconfigurable operator. Over C it is called FFT and over GF (Ft) is called FNT. This architecture has been validated by software. A simple test of calculation of FFT and IFFT, showed the validity of this structure.



.

Fig-4 The architecture of FNT operator

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

2.3 Modulo 2n+1 Partial Product Multiplier

For the modulo 2n+1 multiplier proposed by Efstathiou, there are n+3 partial products that are derived by simple AND and NAND gates. An FA based Dadda tree that reduces the n+3 partial products into two summands is followed. Then a modulo 2n+1 adder for diminished-1 operands is employed to accept these two summands and produce the required product.
In the proposed parallel architecture for cyclic convolution based on FNT, the BOWA can accept four operands in the diminished-1 number system. Every point wise multiplication only needs to produce two partial products rather than one product. The operation can be accomplished by taking away the final modulo 2n+1 adder of two partial products in the multiplier. Thus the final modulo 2n+1 adder is omitted and the modulo 2n+1 partial product multiplier is employed to save the delay and the area.

3. PARALLEL ARCHITECTURE FOR CYCLIC CONVOLUTION

Based on the CCWA, the BOWA and the MPPM, we design the whole parallel architecture for the cyclic convolution based on FNT as shown in Fig.6.1. 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. Then the IFNT of the partial products are performed to produce the resulting sequence {pi} of the cyclic convolution.

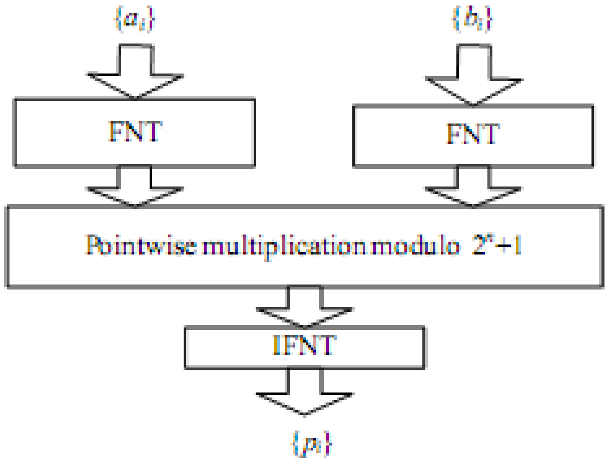


Fig-5 Parallel architecture for the cyclic convolution based on FNT

In the architecture, the radix-2 decimation-in-time (DIT) algorithm which is by far the most widely used algorithm is employed to perform the FNT and the IFNT

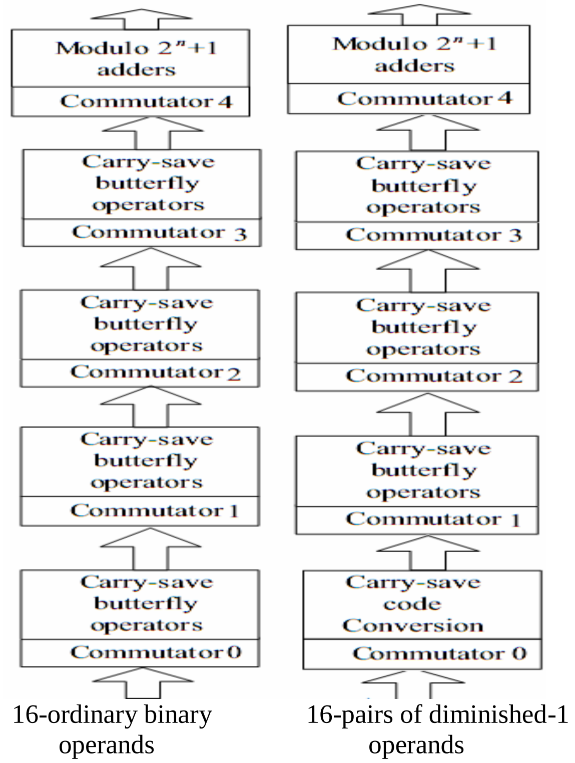

Modulo 2^n+1 adders
Commutator 4
Carry-save butterfly operators
Commutator 3
Carry-save butterfly operators
Commutator 2
Carry-save butterfly operators
Commutator 1
Carry-save butterfly operators
Commutator 0
Modulo 2^n+1 adders
Commutator 4
Carry-save butterfly operators
Commutator 3
Carry-save butterfly operators
Commutator 2
Carry-save butterfly operators
Commutator 1
Carry-save code Conversion
Commutator 0
16-ordinary binary operands
16-pairs of diminished-1 operands

Fig-6 Structures for FNT and IFNT (Ft=28+1)

The efficient FNT structure involves log2N+ 1 stages of operations. The original operands are converted into the diminished-1 representation in the CCWA stage, containing the information of modulo 2n+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 log2N-1 stages of BOWAs, the results composed of two diminished-1 operands are obtained. The final stage of FNT consists of modulo 2n+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 2n+1 addition stage in the FNT involves N/2 couples of code conversions including the information of modulo 2n+1 addition and subtraction, N/2 butterfly operations and N/2 couple of modulo 2n+1 addition 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 αN. If ignoring the normalization factor 1/N, the above formula is the same as that given in the FNT except that all transform coefficients αN ik used for the FNT need to be replaced by αN-(ik) for the IFNT computation. The proposed FNT structure can be used to complete the IFNT as well with little modification as shown in

Fig. 6.2(b). After the IFNT of N-point bit reversed input data, the interim results are multiplied by 1/N in the finite field or ring. Then x[j] and x[j+N/2] (j=1,2,…,N/2-1) exchange their positions to produce the final results of the IFNT in natural order. Our architecture for the cyclic convolution gives a good speed performance without requiring a complicated control. Furthermore, it is very suitable for implementation of the overlap-save and overlap adds techniques which are used to reduce a long linear convolution to a series of short cyclic convolutions.

3.1 Novel parallel architecture for FNT

Since the FNT has a mathematical algorithm similar to the FFT, an FFT-type structure can be applied to perform a fast FNT.The radix-2 decimation-in-time (DIT) fast algorithm which is by far the most widely used FFT algorithm is employed. With the input data sequence stored in bit –reversed order and the CSBO performed in place, the resulting FNT sequence is obtained in natural order. The novel architecture is shown in Fig. in the case the transform length is 8 and the modulus is 24+1.in Fig. , MA and MS denote “modulo 2n+1 addition” and “modulo 2n+1 subtraction” respectively. The novel parallel N-point FNT architecture with the unity root 2 is composed of one stage of CSCC,log2N-1 stages of CSBO and one stage of modulo 2n+1 carry propagation addition. The final stage is used to evaluate the final results, each of which is a diminished-1 value.

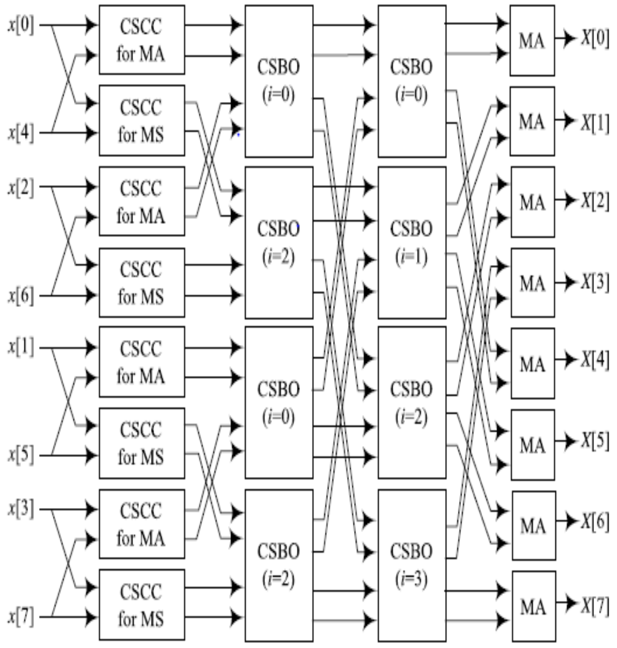


Fig-7 Novel architecture for 8-point FNT with Modulus 24+1

The existing parallel N- point FNT architecture with the unity root 2 consists of one stage of CC and log2N stages of BO. Both architecture involve the same numbers of CC stages and BO stages except their final stages .The proposed CSCC and CSBO stages and BO stages except their final stages. The proposed CSCC and CSBO overcome the disadvantage of carry-propagation addition and don’t require a zero indicator .That are more area delay efficient than the BO and CC respectively. The costs of the final stages of both architectures and approximately equal since every BO is composed of two parallel modulo 2n+1 adder chiefly. Thus the proposed parallel FNT architecture is more novel architecture is very suitable for implementation of the overlap-save and overlap Add techniques which are used reduce a long liner convolution to a series of short cyclic convolution.

4. COMPARISON AND RESULTS

The delay and the area estimations of modulo 2n+1 adder and modulo 2n+1 multiplier in the cyclic convolution are given in Table as a function of the operand size n. “D(n+3)” in Table 1 is defined as shown in Table 4.2

Table-1 Area and delay estimations for arithmetic modulo 2n+1

operator	Area		Delay	
	This project	[3]	This project	[3]
MA	14n	9/2nlogn+n/2+6	8	2logn+3
MM	$8n^2$+n-1	9/2nlogn+$8n^2$+n/2+4	4D(n+3)+1	4D(n+3) $2\log_2 n$+3

“MA” and “MM” represent modulo 2n+1 adder and multiplier respectively.

Table 1 and 2 indicate that for values of Ft ≥28+1 the proposed architecture comprising the CCWA and the BOWA require less delay and area than the previous one. The former results in a 12.6% reduction in area and a 26% reduction in delay respectively compared with the latter in the case Ft is 232+1 and the transform length is 64. Moreover, our algorithm will be more and more advantageous with the growth of modulus width.

Table-2 Area and delay results of cyclic convolution based on FNT

Ft	Area(μm^2)		Delay(ns)	
	This project	[3]	This project	[3]
2^8+1	3.5 x 10^5	3.5 x 10^5	8.9	9.9
2^{16}+1	1.86 x 10^6	2.05 x 0^6	11.6	14.4
2^{32}+1	1.08 x 10^7	1.24 x10^7	15.1	20.4

CONCLUSIONS

The modern programmable structures deliver the possibilities to implement DSP algorithms in dedicated embedded blocks. This makes designing of such algorithm an easy task. However the flexibility of programmable structures enables more advanced implementation methods to be used. In particular, exploitation of parallelism in the algorithm to be implemented may yield very good results.

A novel parallel architecture for the cyclic convolution based on FNT is proposed in the case the principle root of unity is equal to 2 or its integer power. The FNT and the IFNT are accomplished by the CCWA and the BOWA mainly. The pointwise multiplication is performed by the modulo 2^n+1 partial product multiplier. Thus there are very little modulo 2n+1 carry-propagation addition compared to the existing cyclic convolution architecture. A theoretical model was applied to access the efficiency independently of the target technology. VLSI implementations using a 0.13 um standard cell library show the proposed parallel architecture can attain lower area and delay than that of the existing solution when the modulus is no less than 2^8+1.

REFERENCES:

[1]. 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

[2]. 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

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

[4]. 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

[5]. 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

[6]. 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

[7]. 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

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

BIOGRAPHIES:

A.LAXMAN received the M.TECH degree from Srinidhi Institute Of Technology, Ghatkesar, Hyderabad
Currently he is working as an asst. prof. in MAHAVEER Inst. Of Science and Technology, Hyderabad

A.VAMSHIDHAR REDDY received the M.TECH degree from Bandari Srinivas Institute Of Tech.,Chevella, Hyderabad. Currently he is working as an assoc. prof. in RRS College of Eng. And Tech., Hyderabad

L.PRAKASH received the M.TECH degree from TRR College of Eng. And Tech.,Hyderabad
Currently he is working as an assoc. prof. in DVR College of Eng. And Tech., Hyderabad

T.SATYANARAYANA received the M.TECH degree from JBIT,Moinabad, Hyderabad Currently he is working as an assoc. prof. in DVR College of Eng. And Tech, Hyderabad