IMPLEMENTATION OF LOW AREA AND HIGH SPEED PARALLEL ARCHITECTURE FOR CYCLIC CONVOLUTION BASED ON FNT IN VLSI DESIGN

Maheswara Rao B.[1] **Banoth Krishna**[2]

[1]Dept of ECE., Khammam institute of Technology, & Science. Khammam.
[2]Dept of ECE, KITE Women's College of Professional Engineering Sciences, Shabad, Hyderabad.
Email: [1]maheswar414@gmail.com, [2]banothkrishna@gmail.com

Abstract

Cyclic convolution is also known as circular convolution. It is simpler to *compute* and produces less output samples compared to linear convolution. Given $x(n)$ and $h(n)$, the length - N cyclic convolution can be expressed as where $(n-K)$ mod N returns the remainder of the integer division of $n-k$ by N. There are many architectures for calculating cyclinc 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 FN is that they are recursive. This paper presents a high speed parallel 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) are proposed to perform the FNT and its inverse (IFNT) except their final stages in the convolution. 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. The execution delay of the parallel architecture is reduced evidently to the decrease of modulo $2^n + 1$ carry-propagation addition.

General Terms

This paper studying parallel architecture design for various applications in VLSI Design, for high speed data transformation, using Fermat number transform, different designing method to provide high speed data transmission.

Keywords: VLSI, DSP, FPGA Implementation using Verilog and VHDL coding method. Tools are Xilinx new version

I. 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 of 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 ($Ft = 2^{2t} + 1$, t^{th} Fermat; t is an integer) the NTT. Turns into the Fermat Number Transform (FNT).

$$y(n) = \sum_{n=0}^{N-1} x(k)\, h((n-k) \bmod N)$$

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 code conversion (CC) stage which converts the normal binary numbers into their diminished-1 representation is compulsory.Other arithmetic operations described originally by Leibowitz includes 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 accomplished the point wise multiplication so that the final carry-propagation addition of two partial product in the multiplier is avoided.

The FNT of a sequence of length N

$$X(K) = \sum_{t=0}^{N-1} x(i)^{ik}, k = 0, 1 \ldots N-1$$

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

The inverse FNT is given by

$$x_t = \frac{1}{N} \sum_{k=0}^{N-1} x_k a_N^{-ik} \bmod F_t \,(i = 0, 1 \ldots N-1)$$

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, $\alpha = 2$, $Ft = 2^{2t} + 1$ and $N = 2.2^t$ where t is an integer.

II. 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 $2n + 1\ 4 - 2$ compressors mainly which are composed of the 4-2compressor introduced by Nagamatsu . The $4 - 2$ compressor, the novel modulo $2n + 1\ 4 - 2$ compressor and the BOWA are denotes the diminished-1 representation of X, $ix \wedge n = x - 1$.

A. 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. 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:

$$A = 2^n A_H + A_L$$

$$B = 2^n B_H + B_L$$

And

$$M_0 = (2^n - 1) - A_H = \bar{A}\, H$$

$$M_1 = (2^n - 1) - B_H = BH$$

$$M_2 = (2^n - 1) - B_L = B_L$$

If the subsequent operation of CC is modulo $2^n + 1$ addition, assign A_L, M_0, B_L and 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:

$$I_0 = I_{0\,(n-1)}\, I_{0\,(n-2)} \cdots\cdots I_{01}\, I_{00}$$

$$I_1 = I_{1\,(n-1)}\, I_{1\,(n-2)} \cdots\cdots I_{11}\, I_{10}$$

$$I_2 = I_{2\,(n-1)}\, I_{2\,(n-2)} \cdots\cdots I_{21} I_{20}$$

$$I_3 = I_{3\,(n-1)}\, I_{3\,(n-2)} \cdots\cdots I_{31}\, I_{30}$$

We obtain the sum vector H_0^* and carry vector. H_1^* In the diminished-1 number system

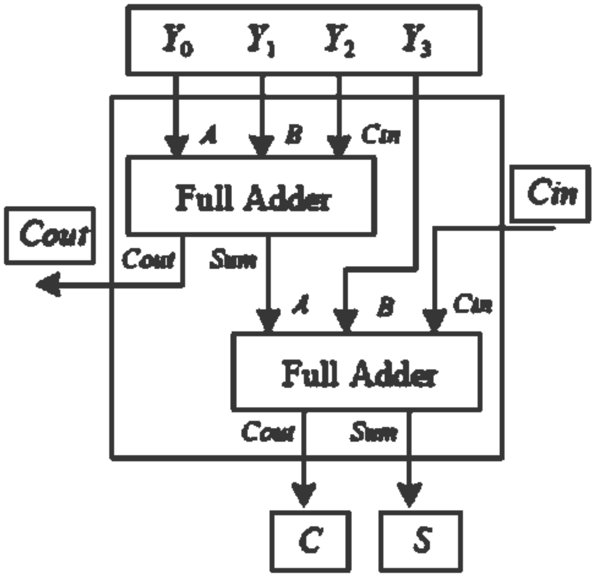


Fig. 1. Compressor

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. That is to say $H_0^* = H_{0\,(n-1)}\ H_{0\,(n-2)} \ldots\ldots H_{01}H_{00}$

$$H_1^* = H_{1\,(n-2)} \ldots\ldots H_{11}\ H_{10}\ H_{1\,(n-1)}$$

The result of modulo $2^n + 1$ addition of A^* and B^* is equal to the result of modulo $2^n + 1$ addition of and H_1^* in this way, A and B are converted into their equivalent diminished-1 representations H_0^* and H_1^* If the subsequent operation is modulo $2^n + 1$ subtraction, we assign A_1, M_0, M_2 and B_H to I_0, I_1, I_2, I_3 respectively. 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 previous BO.

B. *Butterfly operation without addition*

After the CCWA, we obtain the results of modulo $2n + 1$ addition and subtraction in the diminished-1Representation. 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 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 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.

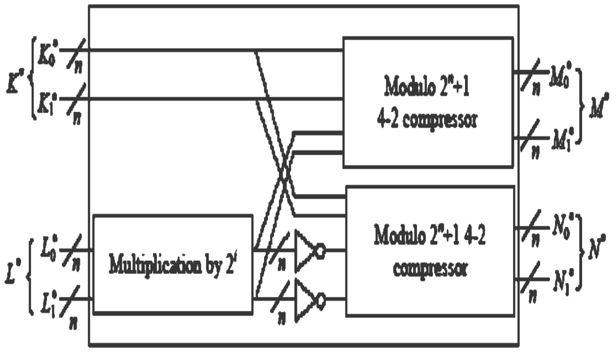


Fig. 2. Butterfly operation without addition

C. *Modulo $2^n + 1$ partial product multiplier*

For the modulo $2^n + 1$ multiplier proposed by Efstathiou, there are $n + 3$ partial products that are derived by simple AND and NAND gates [11]. An FA-based Dadda tree that reduces the $n + 3$ partial

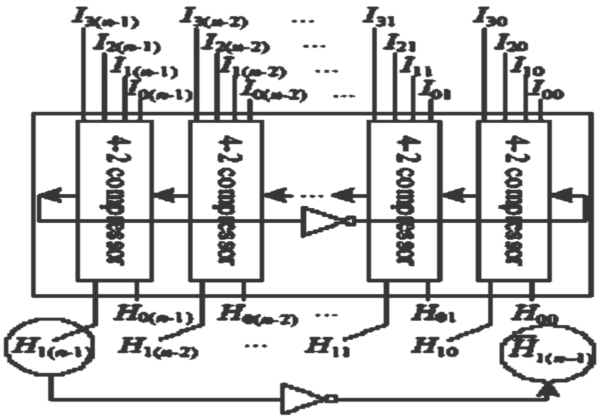


Fig. 3. Modulo $2^n + 1$ 4-2compressor

products into two summands is followed. Then a modulo $2^n + 1$ 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 $2^n + 1$ 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 save the delay and the area.

III. 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. 2.Includes the FNTs, the point wise multiplication and the IFNT mainly. FNTs of two input sequences $\{ ai \}$ and $\{ bi \}$

produce two sequences $\{ Ai \}$ and $\{ Bi \}$ $(i = 1, 2, \ldots N - 1)$. Sequences $\{ Ai \}$ and $\{ Bi \}$. Squences $\{ Ai \}$ 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.

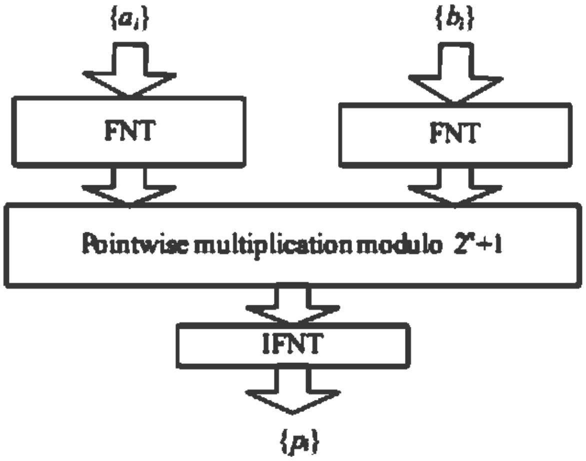


Fig. 4. Pointwise multiplication module

PARALLEL ARCHITECTURE FOR THE CYCLIC CONVOLUTION BASED ON FNT

IFNT. Illustrative examples of the FNT and the IFNT are shown in Fig. 3 in the case the transform length is 16 and the modulus is $2^8 + 1$. Commutators in Fig. 3 are used to adjust the operand order of every stage of FNT and IFNT according to the radix-2 DIT algorithm.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 $\log_2 N - 1$ stages 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 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.

IV. COMPARISON AND RESULTS

In this section, we compare the proposed parallel architecture for the cyclic convolution against that introduced by Conway [3]. The modulo $2^n + 1$ addition for the diminished-1 number system is the crucial operation which contains a standard n-bit carry-propagation computation such as a parallel-prefix adder with a carry-logic block and a zero indicator of the diminished-1 operand to determine whether to

Perform subsequent operations. It produces the longest execution delay and requires large area in the previous solution. The proposed CCWA and BOWA overcome the disadvantage of the carry-propagation adder and don't require a zero indicator. Thus our architecture is faster and more efficient than the existing one.

Thus, a full adder has an area of seven gates and a delay of four gates. This model does not involve the cost of buffering and routing, but achieve a reasonable accuracy for the purpose of comparison. The delay and the area estimations of modulo 2^n+1 adder and modulo 2^n+1 multiplier in the cyclic convolution are given in table

To obtain more accurate results, we describe the proposed parallel cyclic convolution in verilog for $F_1 = 2^8+1, 2^{16}+1, 2^{32}+1$. The validated Verilog code is synthesized using a 0.13-μm CMOS standard cells library in the worst operating condition by the Synopsis Design Compiler. The units of area and delay are μm^2 and *ns* respectively. Each design was recursively optimized for speed until the EDA software can't provide a faster design. The results for the fastest derived implementation are listed.

Table 1 Indicate that for values of F_t=> 2^8,1, 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 F_t is $2^{32}+1$ and the transform length is 64. Moreover, our algorithm will be more and more advantageous with the growth of modulus width.

Structures for FNT and IFNT $(Ft = 2^8+1)$

Table 1. Area and delay results of cyclic convolution based on FNT

F_t	Area $(\mu\, m^2)$		Delay (ns)	
	This paper	[3]	This paper	[3]
2^8+1	3.5×10^5	3.9×10^5	8.9	9.9
$2^{16}+1$	1.86×10^6	2.05×10^6	11.6	14.4
$2^{32}+1$	1.08×10^7	1.24×10^7	15.1	20.4

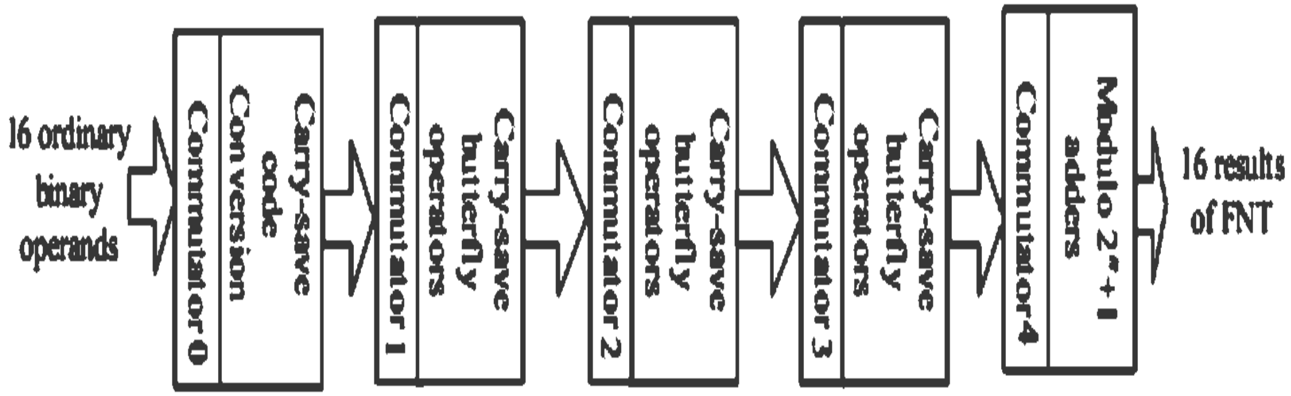

16 ordinary binary operands
Commutator 0
Carry-save code conversion
Commutator 1
Carry-save butterfly operators
Commutator 2
Carry-save butterfly operators
Commutator 3
Carry-save butterfly operators
Commutator 4
Modulo 2^n+1 adders
16 results of FNT

Fig. 5. (a) Parallel FNT structure

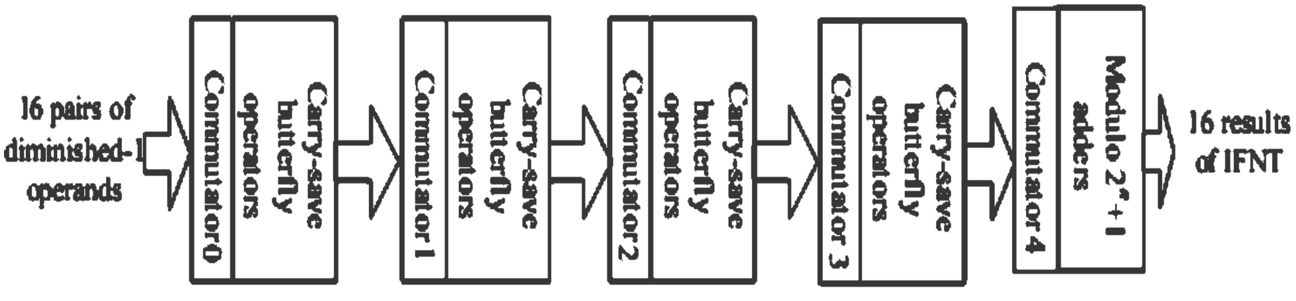

16 pairs of diminished-1 operands
Commutator 0
Carry-save butterfly operators
Commutator 1
Carry-save butterfly operators
Commutator 2
Carry-save butterfly operators
Commutator 3
Carry-save butterfly operators
Commutator 4
Modulo 2^n+1 adders
16 results of IFNT

Fig. 5. (b) Parallel IFNT structure

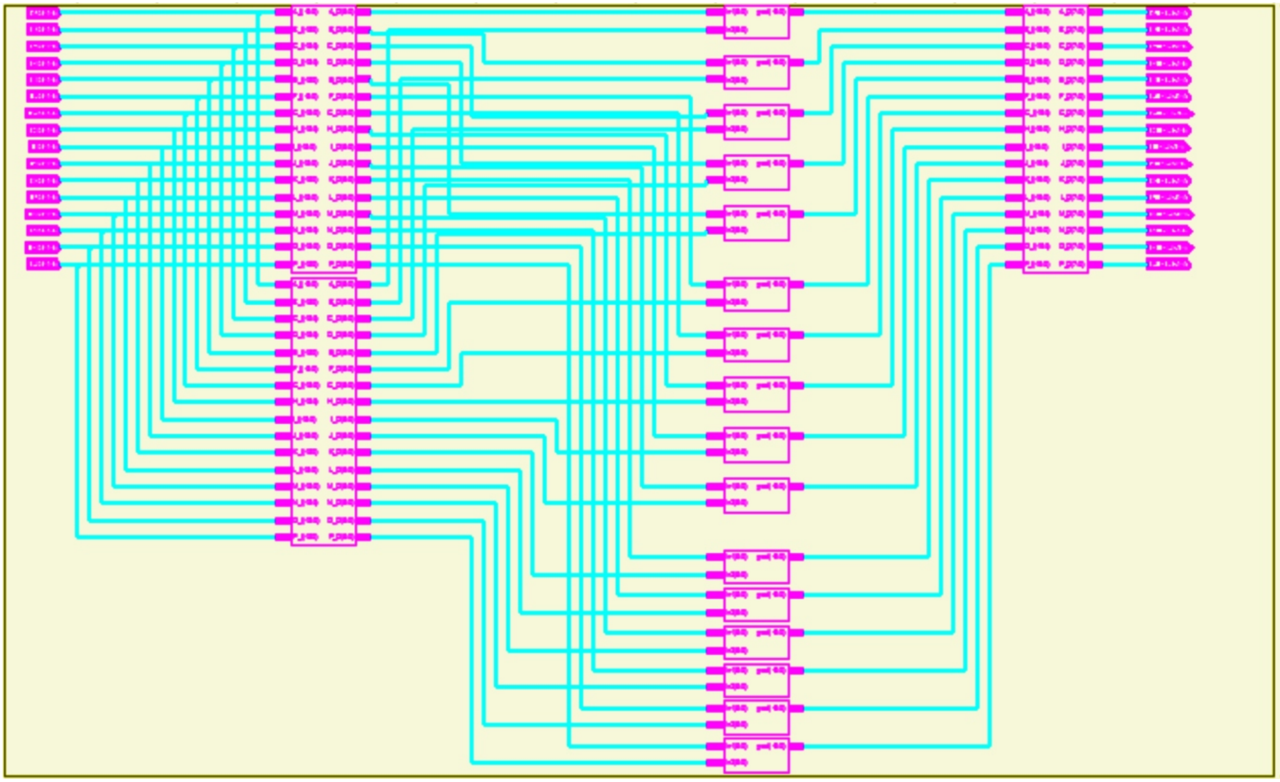


Fig. 6. FNT parallel Architecture

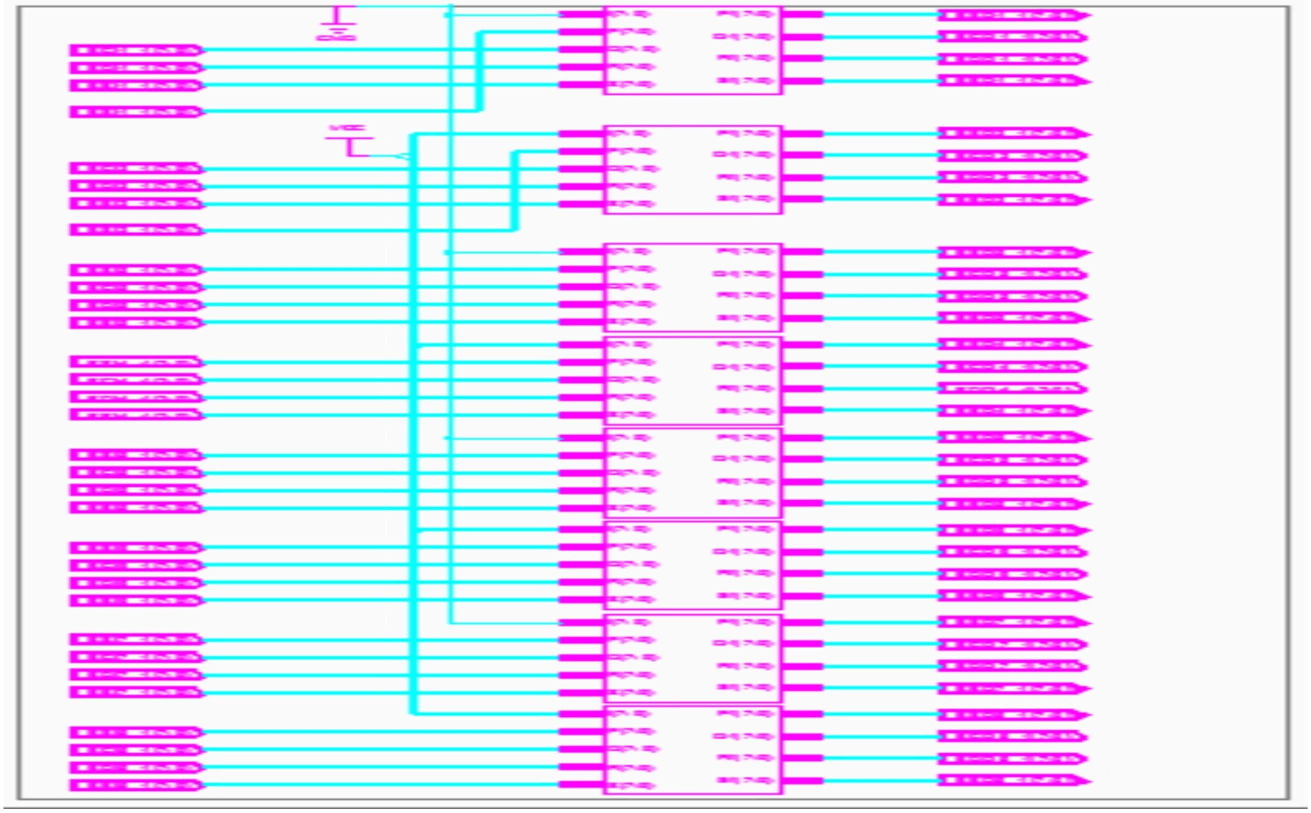


Fig. 7. RTL Schematic of Butterfly Operation

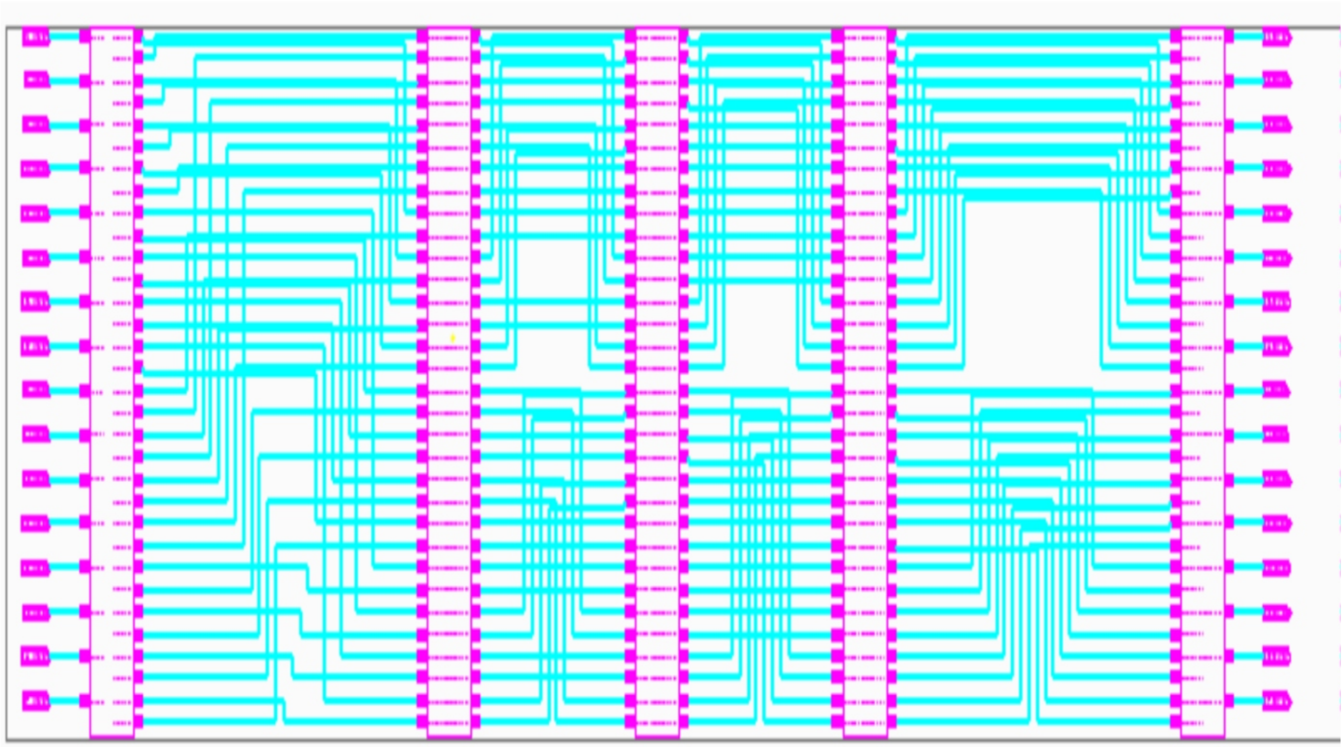
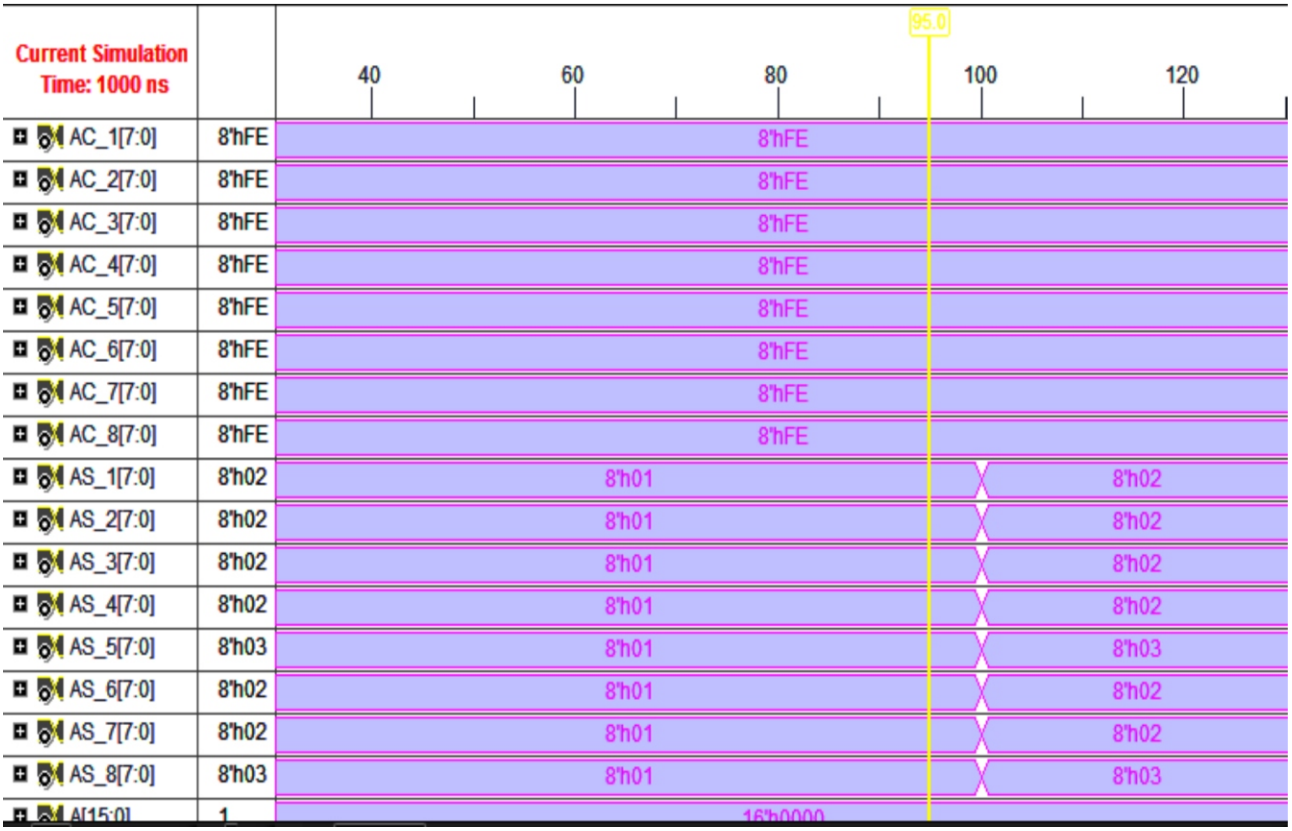
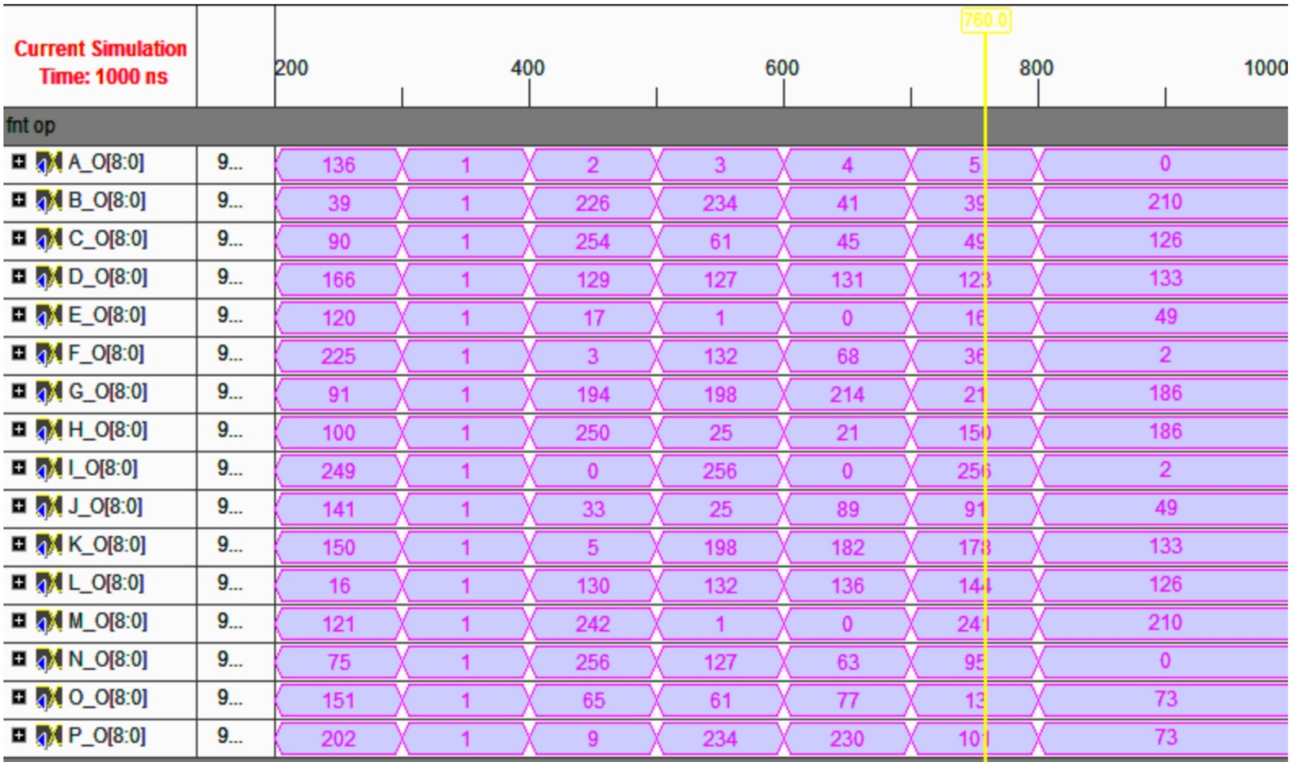


Fig. 8. FNT parallel Architecture over all Schematic



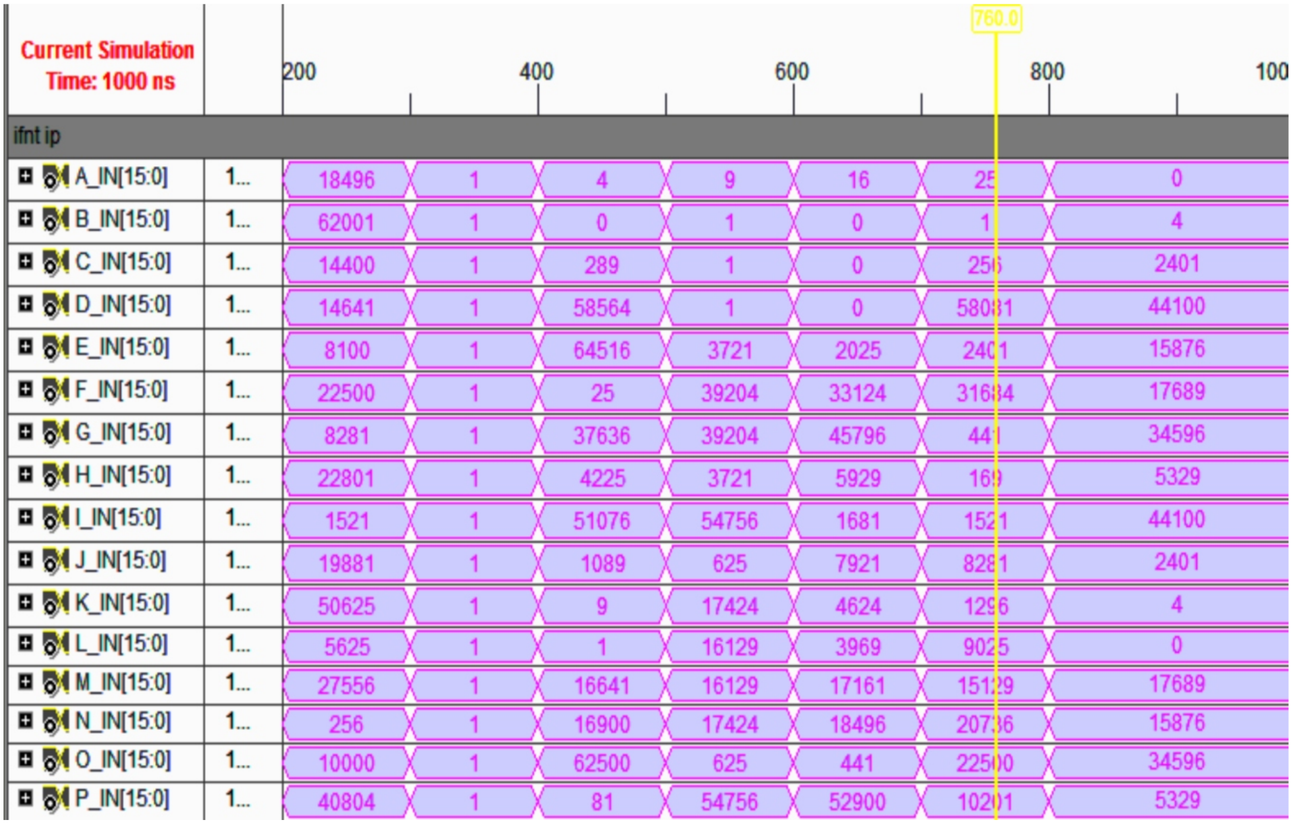
```
Current Simulation
Time: 1000 ns                          95.0
                      40        60        80        100        120
AC_1[7:0]   8'hFE                          8'hFE
AC_2[7:0]   8'hFE                          8'hFE
AC_3[7:0]   8'hFE                          8'hFE
AC_4[7:0]   8'hFE                          8'hFE
AC_5[7:0]   8'hFE                          8'hFE
AC_6[7:0]   8'hFE                          8'hFE
AC_7[7:0]   8'hFE                          8'hFE
AC_8[7:0]   8'hFE                          8'hFE
AS_1[7:0]   8'h02                 8'h01                    8'h02
AS_2[7:0]   8'h02                 8'h01                    8'h02
AS_3[7:0]   8'h02                 8'h01                    8'h02
AS_4[7:0]   8'h02                 8'h01                    8'h02
AS_5[7:0]   8'h03                 8'h01                    8'h03
AS_6[7:0]   8'h02                 8'h01                    8'h02
AS_7[7:0]   8'h02                 8'h01                    8'h02
AS_8[7:0]   8'h03                 8'h01                    8'h03
A[15:0]     1                              16'h0000
```

Fig. 9. Code conversion simulation result



A_O[8:0]	9...	136	1	2	3	4	5	0
B_O[8:0]	9...	39	1	226	234	41	39	210
C_O[8:0]	9...	90	1	254	61	45	49	126
D_O[8:0]	9...	166	1	129	127	131	123	133
E_O[8:0]	9...	120	1	17	1	0	16	49
F_O[8:0]	9...	225	1	3	132	68	36	2
G_O[8:0]	9...	91	1	194	198	214	21	186
H_O[8:0]	9...	100	1	250	25	21	150	186
I_O[8:0]	9...	249	1	0	256	0	256	2
J_O[8:0]	9...	141	1	33	25	89	91	49
K_O[8:0]	9...	150	1	5	198	182	178	133
L_O[8:0]	9...	16	1	130	132	136	144	126
M_O[8:0]	9...	121	1	242	1	0	241	210
N_O[8:0]	9...	75	1	256	127	63	95	0
O_O[8:0]	9...	151	1	65	61	77	13	73
P_O[8:0]	9...	202	1	9	234	230	101	73

Fig. 10. FNT Top module



A_IN[15:0]	1...	18496	1	4	9	16	25	0
B_IN[15:0]	1...	62001	1	0	1	0	1	4
C_IN[15:0]	1...	14400	1	289	1	0	256	2401
D_IN[15:0]	1...	14641	1	58564	1	0	58081	44100
E_IN[15:0]	1...	8100	1	64516	3721	2025	2401	15876
F_IN[15:0]	1...	22500	1	25	39204	33124	31684	17689
G_IN[15:0]	1...	8281	1	37636	39204	45796	441	34596
H_IN[15:0]	1...	22801	1	4225	3721	5929	169	5329
I_IN[15:0]	1...	1521	1	51076	54756	1681	1521	44100
J_IN[15:0]	1...	19881	1	1089	625	7921	8281	2401
K_IN[15:0]	1...	50625	1	9	17424	4624	1296	4
L_IN[15:0]	1...	5625	1	1	16129	3969	9025	0
M_IN[15:0]	1...	27556	1	16641	16129	17161	15129	17689
N_IN[15:0]	1...	256	1	16900	17424	18496	20736	15876
O_IN[15:0]	1...	10000	1	62500	625	441	22500	34596
P_IN[15:0]	1...	40804	1	81	54756	52900	10201	5329

Fig. 11. IFNT Top module

VI CONCLUSIONS

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 point wise multiplication is performed by the modulo $2^n + 1$ partial product multiplier. Thus there are very little modulo $2^n + 1$ 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.*

[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

Mr.Bhukya Maheswara Rao He has completed his B.Tech in ECE from Sri Raja Rajeswari Engineering College-Karepally, Khammam (JNTUH) in 2006 and M.Tech (VLSI-SD)-From Hyderabad Institute of Technology & Management (JNTUH) in 2010. His interested area's are Digital Design, Low Power VLSI Design, Analog VLSI Desgin, FPGA Technology.

Mr.Banoth Krishna He has completed his B.Tech in ECE from Dr.Paul Raj Engineering College-Bhdrachalam (JNTUH) in 2005 and, M.Tech (VLSI-SD)-From SITAMS-Chittoor (JNTUH)in 2008.His interested area's are Digital Design, Low Power VLSI Design, Analog VLSI Desgin, FPGA Technology.