Design of Low Power, High Speed Parallel Archietecture for Cyclic Convolution Based on FNT

V.Rajeshwari and A.Anendhar

Abstract: This paper presents a 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 with out addition (BOWA) are proposed to perform the FNT and its inverse(IFNT) except their final stages in the convolution. The point wise multiplication is used in this convolution by modulo 2^m+1 partial product multipliers (MMPM) and output partial products which are inputs to the IFNT. Thus modulo 2^m+1 carry propagation are avoided in the FNT and the IFNT except in their final stages and modulo 2^m+1 multiplier. The execution delay of the parallel architecture is reduced evidently due to decrease of modulo 2^m+1 carry propagation addition. In this paper novel architectures and design of high speed, low power 4-2 compressors capable of operating at ultra low voltages are presented .in the proposed architectures these outputs are , efficiently utlisied to improve the performance of compressors. compared with the existing cyclic convolution architecture , the proposed one has better throughput performance and involves less hardware complexity. synthesis results is carried out by using 180nm soc technology.

Keywords: fermatnumber transform,diminished-1 number systems,code conversion,butterfly operation.

INTRODUCTION*:*

In this paper the cyclic convolution based on FNT is more attractive than compare to other conventional methods. The cyclic convolution based on FFT is widely used in signal processing. Additionally, the dynamic range of the numbers varies widely so that one need to use floating point numbers to avoid scaling and quantization problems. The fermat number transform (FNT) is more attractive than the conventional fast fourier transform (FFT) in the area of cyclic convolution. The transform has been used in several applications such as video processing ,digital filtering and multiplication of large integers important operations in the FNT include the butterfly operation (BO) and the code conversion (CC) which are both composed of modulo 2^n+1 addition mainly. The fast modulo 2^n+1 adder involving carry- propagation addition in the diminished -1 number system is proposed. The carry propagation modulo 2^n+1 additon makes the existing FNT architectures more area and delay. In this paper , a novel archietecture with the root of unity 2 or its integer power is proposed, which is mainly composed of carry save code conversion (cscc) and carry –propagation addition is avoided in the CSCC and the CSBO. Thus the proposed FNT archietecture requires less area and delay than the previous one.

1.1 Important operations

Important operations of the cyclic convolution based of the cyclic convolution based on FNT with the unity root 2 include the CCWA and the BOWA and the MMPM. The CCWA and the BOWA both consist of novel modulo 2^n+1 4-2 compressors mainly which are composed of the 4-2 compressors. The 4-2 compressors, the novel modulo 2^n+1 4-2 compressor, the BOWA are shown in fig 1 . in the figure , X^* denotes the diminished-1 representation of X, i.e., X^* =X-1.

1.2 code conversion with out addition:

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

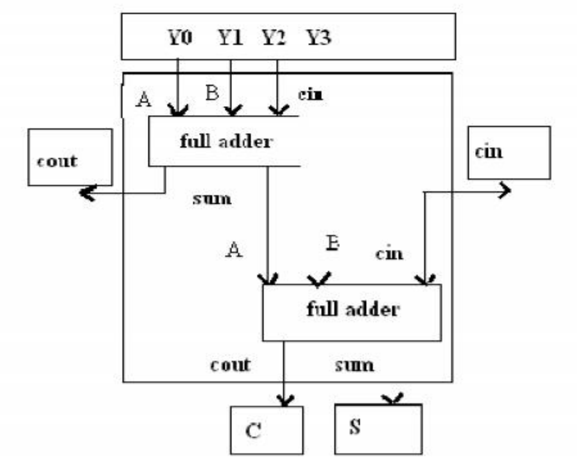


Fig-1 elementary operations of FNT architectures with unity root 2,(a) 4-2 compressors.

V.Rajeshwari is a M.Tech VLSI Aurora scientific technological research acedamy college of engg,Bandlaguda,Hyderabad and A.Anendhar is working as [2]Asst.prof, ECE Department , Aurora scientific technological research acedamy college of engg,Bandlaguda,Hyderabad. Emails: rajeswariv73@yahoo.com, anendhar@gmail.com

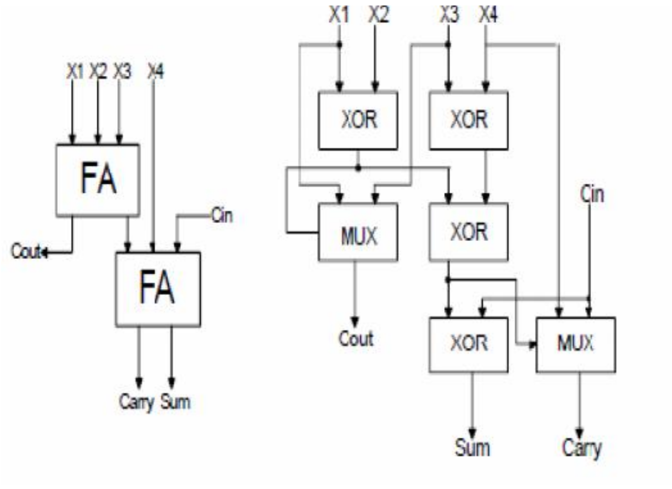


Fig 2 Existing implementation of 4-2 compressor

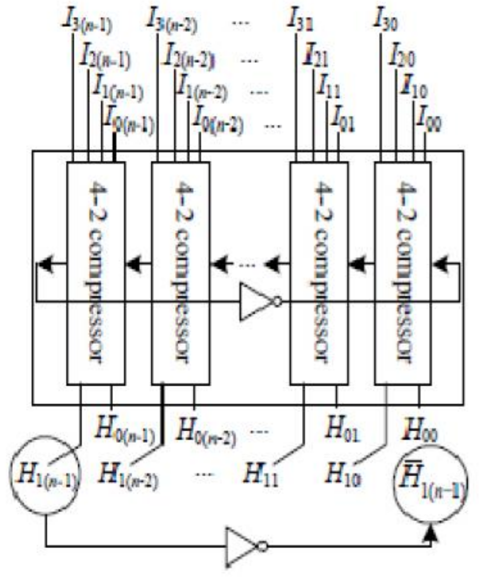

$I_{3(n-1)}$ $I_{3(n-2)}$ ... I_{31} I_{30}
$I_{2(n-1)}$ $I_{2(n-2)}$ ... I_{21} I_{20}
$I_{1(n-1)}$ $I_{1(n-2)}$ ... I_{11} I_{10}
$I_{0(n-1)}$ $I_{0(n-2)}$ ... I_{01} I_{00}
4-2 compressor
4-2 compressor
4-2 compressor
4-2 compressor
$H_{0(n-1)}$ $H_{0(n-2)}$ ... H_{01} H_{00}
$H_{1(n-1)}$ $H_{1(n-2)}$ ... H_{11} H_{10} $\overline{H}_{1(n-1)}$

Fig-3 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

2.BUTTERFLY ARCHITECTURE

2.1 Butterfly operation with out 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 butterfly operation involves four operands. The proposed BOWA involves two modulo 2n+1 4-2 compressors, a multiplier and some inverter.

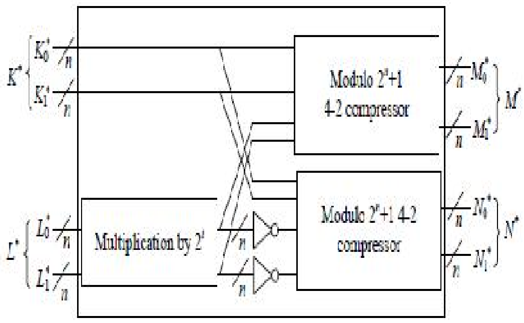


Fig 4 Butterfly operation with out addition

The multiplication by an integer power of 2 in the diminished-1 number system is performed by shifting the low number of n-i bits of the number by i bit positions then inversing and circulating the high order i bits in to the i least significant positions.

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.fig 5 represents the global reconfigurable operator. A simple test of calculation of FFT and IFFT, shows the below figure. This architecture is employed to save the delay and area.

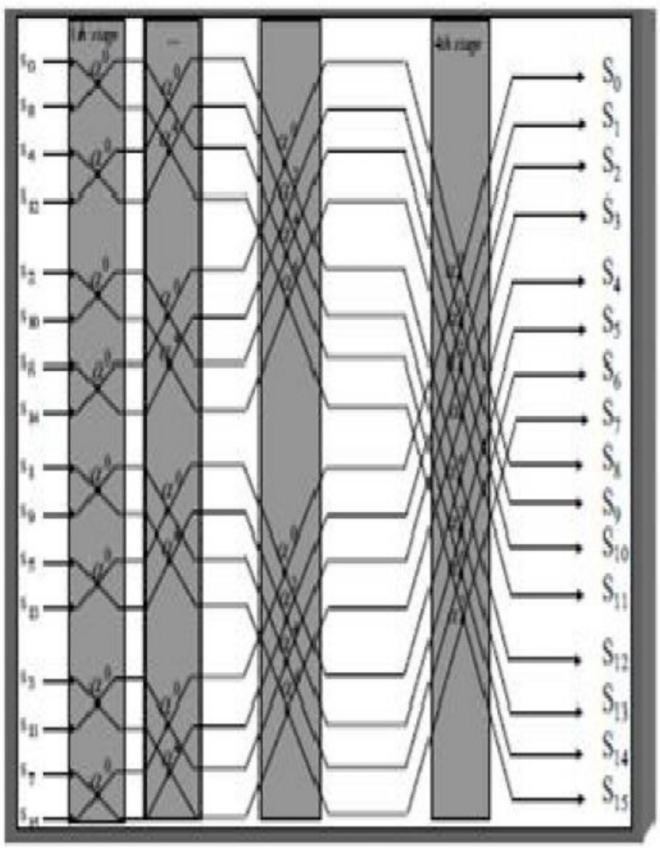


Fig 5 The architecture of FNT operator.

2.3 Modulo 2n+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. 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 and 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.

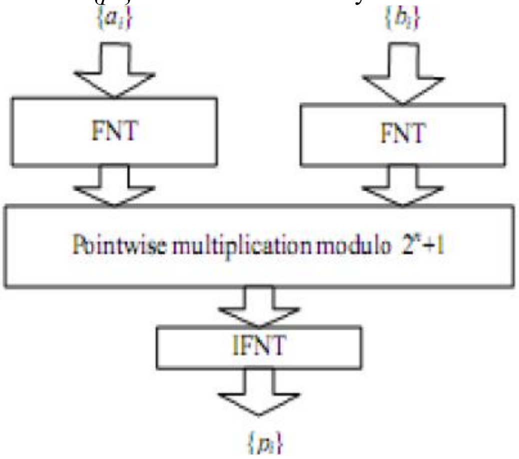


Fig.6 Parallel architecture for the cyclic convolution based on FNT

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 IF*N*T. Illustrative examples of the FNT and the IFNT are shown in Fig. 7 in the case the transform length is 16 and the modulus is 2^8+1. Commentators in Fig.7 are used to adjust the operand order of every stage of FNT and IFNT according to the radix-2 DIT algorithm.

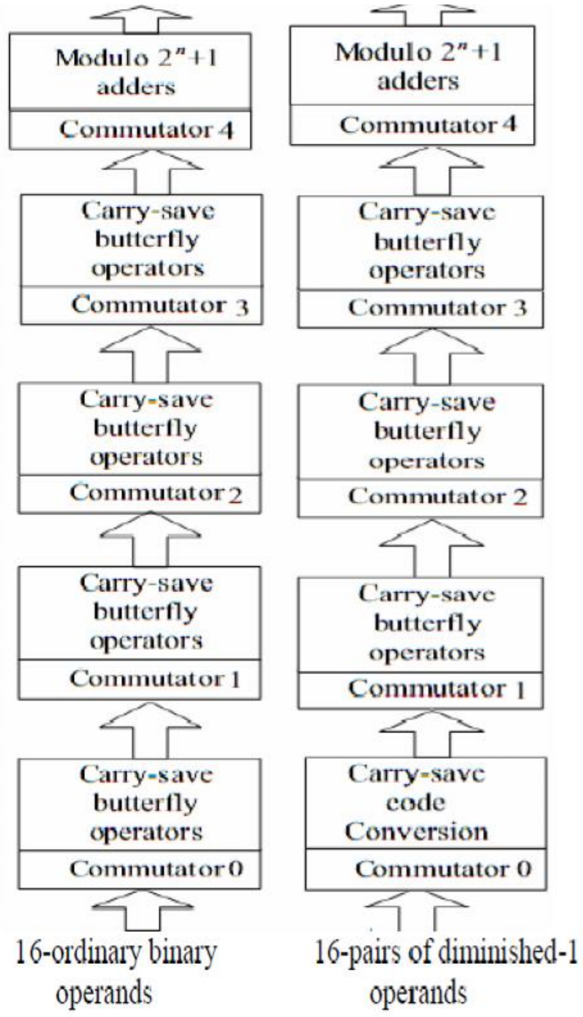

Modulo 2^n+1 adders
Commutator 4
Carry-save butterfly operators
Commutator 3
Carry-save butterfly operators
Commutator 2
Carry-save butterfly operators
Commutator 1
Carry-save butterfly operators
Commutator 0
16-ordinary binary operands
Modulo 2^n+1 adders
Commutator 4
Carry-save butterfly operators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

Fig 7(a) parallel FNT structure
Fig 7(b) parallel IFNT structure

The efficient FNT structure involves log2*N*+ 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 log2*N*-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 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 (Ft=28+1) is the same as that given in the FNT except that all transform coefficients α*N ik* used for the FNT need to be replaced by α*N*-(*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. 5.2(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.

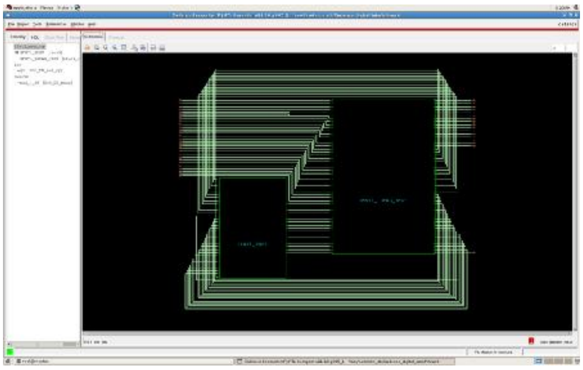
4. COMPARSION AND STIMULATION RESULTS:

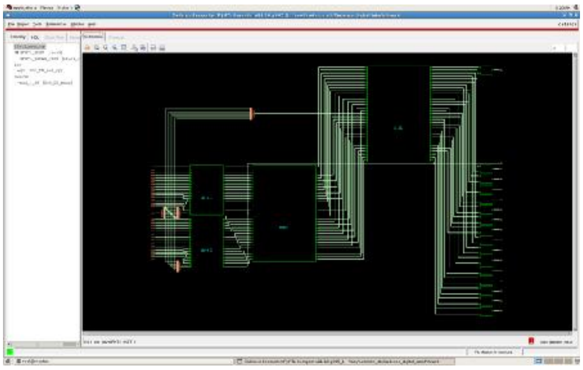
In this section, we compare the proposed parallel architecture for the cyclic convolution against that introduced by Conway. The modulo 2n+1 addition for the diminished-1 number system is the crucial operation which contains a standard n-bit carry propagation computation such as a parallel-prefix adder 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. This model assumes that each two-input gate excluding XOR is equivalent to one elementary gate for both area and delay. An XOR gate counts for two gates for both area and delay. 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.

F_t	Area (μm^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

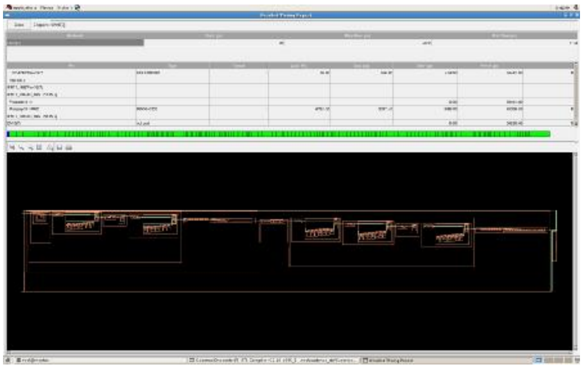
Stimulation results:
Synthesis report in cadence

Ifnt_soc_top

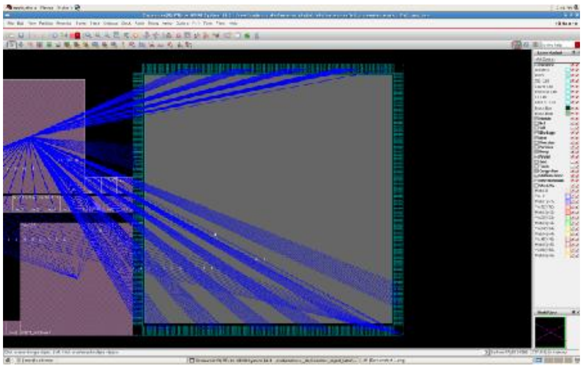




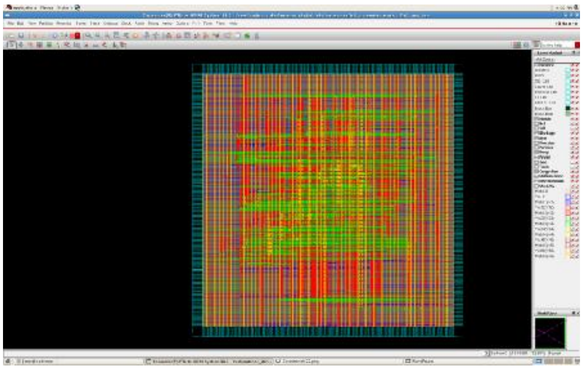
Delay report in rtl compiler:



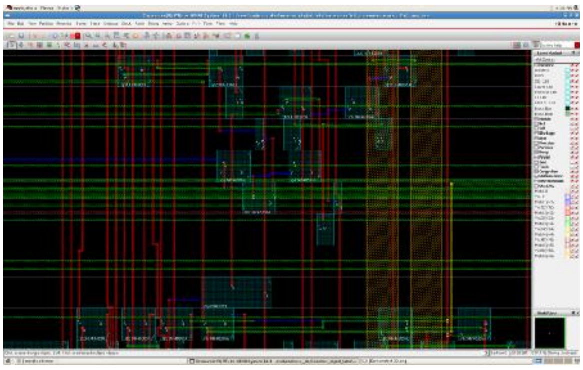
Soc(RTL encounter):



Layout design:

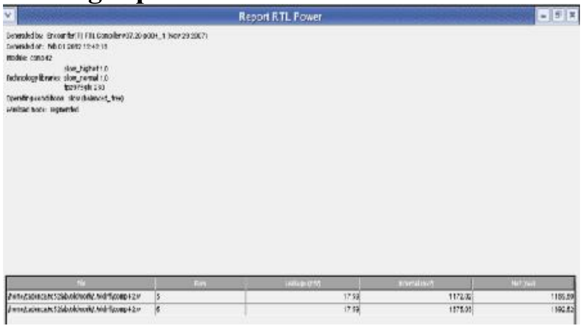


Routing



Power report:

Existing report



Report RTL Power

Generated by: Encounter(R) RTL Compiler v07.20-p004_1 (Nov 29 2007)
Generated on: Feb 01 2012 12:40:15
Module: conv42
Technology libraries: slow_highvt 1.0
slow_normal 1.0
tpz973gwc 230
Operating conditions: slow (balanced_tree)
Wireload mode: segmented

File	Row	Leakage (nW)	Internal (nW)	Net (nW)
/home/cadence/c52lab/old/work/.../comp+2.v	5	17.93	1172.02	1185.89
/home/cadence/c52lab/old/work/.../comp+2.v	6	17.93	1175.01	1190.82

Proposed report

Proposed Architecture RTL Power

5. Conclusion and future scope:

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 $2n+1$ partial product multiplier. Thus there are very little modulo $2n+1$ carry-propagation addition compared to the existing cyclic convolution architecture.

Future Scope

This is the Design of low power,High speed parallel architecture for cyclic convolution based on FNT. Here in this architecture we used modulo adders and modulo multipliers to reduce the round off errors and also to reduce the delay and also that CCWA and BOWA blocks are used for further reduction of the delay. Based on the number of operands as the butterfly stages generally increases there is no other alternative for finding the convolution. The changes can be made either for the blocks like multiplication by 2n or for modulo adder or for modulo multiplier in order to reduce the size. Also that this can be implemented in low power technology by carefully following the parameters and the circuit elements used for the implementation. So as this is the high speed architecture, we further work on for the size or power consumption and others. VLSI implementations using a 0.18 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] L. M. Leibowitz, “A simplified binary arithmetic for the Fermat number transform,” IEEE Trans. Acoustics Speech and Signal Processing, 1976, 24(5):356-359

[8] H. T. Vergos, C. Efstathiou, D. Nikolos, “Diminishedone modulo 2n + 1 adder design”, IEEE Trans. Computers, 2002, 51(12), pp. 1389-1399

[9] 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

[10] C. Efstathiou, H. Vergos, G. Dimitrakopoulos, et al., “Efficient diminished-1 modulo 2n + 1 multipliers”, IEEE Trans. Computers, 2005, 54(4), pp. 491-496

[11] A. Tyagi, “A reduced-area scheme for carry-select adders”, IEEE Trans. Computers, 1993, 42(10), pp. 1163-11.

BOOKS

[1] J. G. Proakis and D. G. Manolakis, Digital signal processing: principles, algorithms, and applications, Prentice Hall, New Jersey, 2007.

[2] Essentials of VLSI Circuits and systems by Kamran Eshraghian, Douglas A. Pucknell, Sholeh Eshraghian.

[3] Modern VLSI Design, System - on - Chip Design Third Edition Wayne Wolf, Pearson Education.

[4] Barbe, D.F. (Ed.) (1982) Very Large Scale Integration – Fundamentals and applications, Springer - Verlag, West Germany / USA.

Name : v.Rajeswari
ADDRESS:
PLOT NO 155A
CHAITNAYA NAGAR COLONY
OPP: B.N.REDDY
KARMANGHAT
HYDERABAD
PIN NO: 500079
EMAIL ID : rajeswariv73@yahoo.com

.