Efficient Concurrent Error Detection and Correction of Soft Errors in NTT-based Convolutions

Anne O'Donnell*, Chris J. Bleakley*, Pedro Reviriego** , Juan Antonio Maestro**

*UCD School of Computer Science & Informatics, University College Dublin
email : anneodonnell@eircom.net

** Universidad Antonio de Nebrija, C/ Pirineos, 55 E-28040 Madrid, Spain
email: {previrie, jmaestro}@nebrija.es

Abstract— A system for soft error detection and correction is proposed for digital Integrated Circuit (IC) implementation of convolution. The convolution is implemented in a Residue Number System using Fermat Number Theoretic Transforms. The flexibility afforded by the Modified Overlap Technique in allowing transforms of differing lengths in a convolution makes it possible to easily detect and correct soft errors by means of a Single Redundant Channel and pattern matching technique. The proposed system gives area reductions in the majority of cases examined, when compared with Triple Modular Redundancy. In the case of large (e.g. 28 and 32 bit) word lengths, the proposed system provides area reductions of up to 30%.

***Keywords*—NTT, RNS, Soft Errors, Convolution**

I. INTRODUCTION

THE trend towards small area, high speed and low power in computer applications has increased the rate of occurrence of radiation induced soft errors. Soft errors cause a malfunction of a circuit that typically results in a temporary change in the logic value of a flip-flop or logic gate. Radiation effects have been an issue in space applications for decades and are now becoming a concern for many ground level applications. The increase in soft errors is due to shrinking transistor size and lower power voltages that make transistors more vulnerable to radiation effects. It is well documented that radiation induced soft errors are a major source of malfunction in modern computers [1]. For future circuits, it is estimated that transistor size and critical charge will continue to decrease with process scaling. The problem of radiation induced soft errors will intensify as a result. A number of techniques have been proposed to address the issue of soft errors in electronic circuits, ranging from modification of the manufacturing process to the design of transistors at the physical level to system oriented techniques that introduce redundancy to detect and correct soft errors when they occur [2]. The most common example of the redundancy based techniques is Triple Modular Redundancy (TMR) where the elements in the system are tripled and majority voting is used to correct errors. Specific redundancy based techniques to protect commonly used algorithms have also been proposed, for example, in the area of signal processing [3].

Convolution is one of the most commonly used signal processing algorithms. It is widely used to implement digital filters where a very long input signal is convolved with a short length fixed sequence. Convolution in its direct form is a very computationally expensive operation. For this reason, transform based convolutions are often used in practical applications. Typically, convolution is realised by calculating the Fast Fourier Transform (FFT) of the sequences to be convolved, multiplying the outputs element-by-element and calculating the Inverse FFT. Number Theoretic Transforms (NTT) also have the Cyclic Convolution Property (CCP) and can be used for efficient implementation of convolution [4]. The main advantage of NTTs over

FFTs is that they do not introduce round offs errors and so provide exact convolution results, which is essential for some applications.

Surprisingly, fault tolerant techniques for transform-based convolution have received little attention in the literature. Clearly, Triple Modular Redundancy (TMR) can be applied. More advanced approaches, based on the use of Residue Number Systems (RNS), have been presented in [5,6,7]. These methods require at least two extra, redundant RNS (RRNS) moduli in order to detect and correct a single fault. The difficulty with these approaches is that the redundant moduli significantly increase area.

In this work, we propose a scheme for detection and correction of soft errors in NTT-based convolutions. The scheme uses a Single Redundant Channel (SRC) to compute the convolution in parallel with the basic channel and a pattern matching technique. The Single Redundant Channel is used to detect errors in the convolution output. Inspection of the pattern of these errors allows the system to determine in which channel the soft error occurred. The system then performs correction by selecting the output of the channel in which no error occurred as the convolution output. The advantage of this approach is that only one redundant channel is needed, rather than two as in TMR and RRNS. This provides a significant area saving relative to these approaches. In this work, we compare the area of the proposed solution to that of conventional TMR applied to the lowest area NTT based convolution. In theory, the proposed scheme can detect and correct all soft errors in the system. In practice, in some case, some errors may not be corrected due to circuit level timing issues. These issues are discussed and solutions proposed.

The paper is organised as follows. Section II provides background information on NTT and RRNS. The proposed fault tolerant convolution approach is presented in section III. A complexity analysis is provided in Section IV comparing the area cost of the proposed approach with TMR. Finally, the conclusions from this work are presented in Section V.

II. BACKGROUND

Convolution is a very computationally expensive operation and in its direct form involves n^2 multiplications. As a means to reduce this computational complexity, the Cyclic Convolution Property (CCP) of the Fast Fourier Transform (FFT) has led to convolutions being performed by means of transforms. A convolution can be realised by getting the transform of each sequence to be convolved, multiplying the outputs element by element and then getting the inverse transform.

Number Theoretic Transforms (NTT) which replace the complex domain with a finite field or ring are defined by the relationships:

$$X(k) = \sum_{i=0}^{N-1} x(i)\omega^{ik} \qquad k = 0,1,...,N-1$$

$$x(i) = N^{-1}\sum_{i=0}^{N-1} X(k)\omega^{-ik} \qquad i = 0,1,...,N-1$$

where N is the transform length, ω is the Nth primitive root of unity in Z_M and all operations are carried out modulo M. For real inputs, NTTs have the advantage that the kernels (ω) are integers as opposed to the complex roots of unity required for the FFT. Thus, use of NTTs results in exact arithmetic which eliminates any round off error

A major problem restricting the use of NTTs is the strict relationship between the modulus M, the transform length N and the kernel, ω. Another consideration is that certain moduli and transform length combinations are more suited than others to area efficient NTT implementations because multiplication by the kernel can be implemented as a bit shift and addition.

Fermat Number Theoretic Transforms (FNTT) are used in this work. Fermat numbers are of the form $2^{2^t}+1$ where t is an integer [4]. The advantage of using Fermat numbers as moduli in NTTs is that composite transform lengths of the form 2^n are available combined with optimum kernels of the form 2^k, which allow multiplication using shifts. Use is also made herein of Generalised Fermat Number Transforms (GFNT) of the form $M = 2^{2^k S}+1$, as described by Toivonen and Heikkila [8]. Table 1 provides a full list of the moduli, with the related transform lengths and kernels used in this work.

Transform based convolutions use the well-known Overlap Add or Overlap Save techniques to convolve a short, fixed coefficient sequence with a long data sequence. Typically, a single transform is repeatedly applied to the incoming data. In [9], Conway showed that lower area NTT based convolutions can be

Table 1: Moduli M, transform lengths L and kernels ω used in the proposed system

M	L	ω	M	L	ω
2^8+1	32	$2^3(2^4-1)$	$2^{24}+1$	32	$2^7(2^{12}-1)$
2^8+1	16	2	$2^{24}+1$	16	2^3
2^8+1	8	2^2	$2^{24}+1$	8	2^6
$2^{12}+1$	16	$2^4(2^6-1)$	$2^{28}+1$	16	$2^3(2^{14}-1)$
$2^{12}+1$	8	2^3	$2^{28}+1$	8	2^7
$2^{16}+1$	64	$2^0(2^8-1)$	$2^{28}+1$	4	2^{14}
$2^{16}+1$	32	2	$2^{32}+1$	128	$2^0(2^{16}-1)$
$2^{16}+1$	16	2^2	$2^{32}+1$	64	2
$2^{16}+1$	8	2^4	$2^{32}+1$	32	2^2
$2^{20}+1$	16	$2^2(2^{10}-1)$	$2^{32}+1$	16	2^4
$2^{20}+1$	8	2^5	$2^{32}+1$	8	2^8

realised using a Modified Overlap Technique (MOT) wherein a number of parallel moduli are used. Convolution is performed independently in each modulus. The moduli are selected such that they form a Residue Number System. The overall convolution outputs are obtained by combining the parallel moduli results using the Chinese Remainder Theorem (CRT). Since this approach allows for transform of differing lengths it relieves the word length-transform length problem and so reduces the area of the system.

Fault tolerance in RNS has previously been implemented using a technique known as Redundant Residue Number System (RRNS). It is defined as an RNS with N moduli for the basic non-redundant system and r additional redundant moduli for error detection [10]. r redundant moduli are required to detect and correct $[r/2]$ errors, where $[x]$ is the largest integer such that $[x] \leq x$, [10]. All $N+r$ moduli must be relatively prime and each redundant modulus must be greater than the individual non-redundant moduli. It is this latter condition which causes the provision of redundancy to significantly increase area in traditional RRNS. This is not required in the proposed technique.

III. PROPOSED METHOD

The error detection and correction technique proposed herein is based on the idea of adding a Single Redundant Channel (SRC) which together with pattern matching allows us to detect and correct errors. Both the basic and redundant channel computes the convolution in parallel as illustrated in Figure 1. If the outputs of the two channels are different, an error is detected by identification of the channel in error by analysis of the error pattern. Once the channel in error is known, correction can be performed by selecting the convolution output from the other channel.

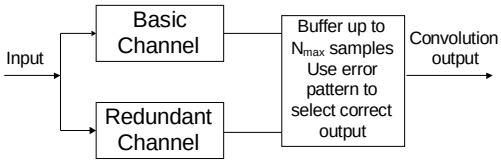

Figure 1: Block Diagram of the proposed approach.

The key to the system is the identification of the channel in error by pattern matching. We note that the NTT has the same structure as the FFT as illustrated in Figure 2. Now consider a single channel of the convolution system. If a soft error occurs in one of the nodes of the forward NTT, the error will propagate to at least one of the outputs of the NTT and so to all of the outputs of the inverse NTT and from there to all N_i convolution outputs, where N_i is the length of the transform. Similarly, a soft error in the multiplier will give rise to errors in all N_i convolution outputs. If the soft error occurs in the inverse NTT, a more complex error pattern will emerge at the convolution output. Consider, for example, a soft error in an 8-point NTT occurring at the first input to stage 2. Errors will result at outputs $X(0)$, $X(2)$, $X(4)$ and $X(6)$ as shown in Figure 2. In general, for an NTT of length

$N = 2^n$ where n is an integer, a single error at the input to a stage g gives rise to r errors with a separation s at the output of the NTT where $r=N/2^g$ and $s=2^g$.

Consider the mismatch pattern arising from a soft error at the input to stage g of the inverse transform, where $g=0$ is the first stage and $g=n-1$ is the final stage. Due to the structure of the inverse transform, the number of possible mismatch patterns is 2^g, the number of mismatches in each pattern is 2^{n-g} and the separation of mismatches is 2^g samples. Table 2 illustrates the case where $N=16$.

In the proposed approach all channels must have different transform lengths N_i. This is made possible by the use of the Modified Overlap Technique [9]. Also, in this work, the most area efficient implementation for a given wordlength and filter length is found. In some cases this involves multiple moduli implemented in an RNS. The system is designed so that each modulus supports a different transform length. Therefore, in most cases it is possible to distinguish the channel in error based on the number of errors and their separation.

Table 2: Mismatch pattern for N=16, n=4.

Stage (g)	Possible patterns	Number of mismatches per pattern	Separation of mismatches
0	1	16	1
1	2	8	2
2	4	4	4
3	8	2	8

Soft errors in the forward NTT and multiplier lead to a burst of N_i errors with zero separation in the convolution output. Soft errors at the input to stage g in the inverse NTT of the ith channel lead to $N_i/2^g$ errors in the convolution output at a separation of 2^g. Since N_i is unique and a power of 2, the channel in error can be identified in all cases by multiplying the number of errors and their separation, except for those occurring in the final stage of the INTT. Thus, the final stages of the INTT are protected by applying DMR in one of the channels (redundant or non redundant).

In real circuits, the arrival times of the errors at the output of the system, due to a single soft error event, will not be exactly the same. In some cases, this may mean that, the error pattern differs from that predicted by this analysis. The frequency of this occurrence depends on the ratio of the duration of the soft error and the timing skew between the soft error location and the outputs of the circuit. The frequency of the occurrence of this problem may be reduced by careful design of the circuit. Reducing the timing skew between outputs would reduce the problem. Registering the outputs of each butterfly would eliminate the problem. Alternatively, the pattern recognition may be designed so as to be robust to the problem.

Most soft errors are single bit errors[1]. It is assumed in this analysis that only one soft error occurs during the computation of a convolution transform block. This is a reasonable assumption given the frequency of soft errors. The same assumption is also implicit in TMR.

A delay line is used at the output of both channels such that the error pattern can be used to identify the channel in error and the correct output be selected from the other channel.

The length of the delay line when all N_i are a power of two is $N_{max}/2+1$ where N_{max} is the largest transform length. In all cases, a delay line of length N_{max} would be sufficient.

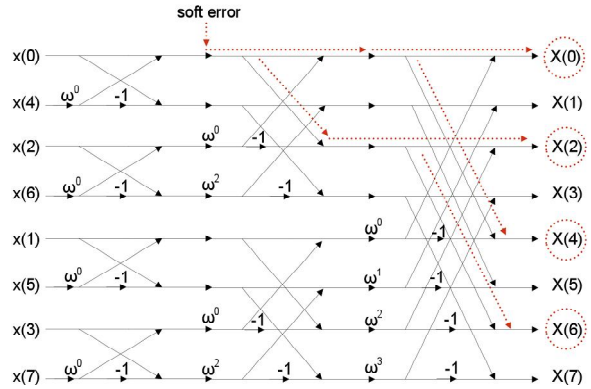

Figure 2: 8-point radix-2 NTT showing path of error at the input to stage 2.

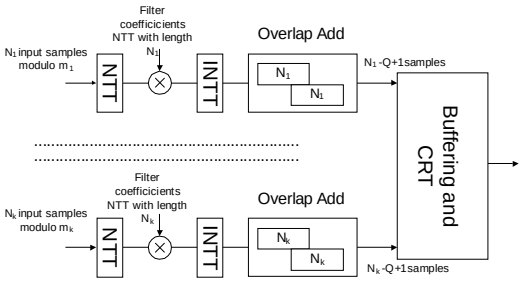

Figure 3: Block Diagram of a channel.

The overall scheme is illustrated in Figure 1. The structure of a channel is similar to the one proposed in [9] and it is shown in Figure 3 for the general case in which multiple moduli are used in the channel. In that case an additional buffering stage followed by a reconstruction stage using the Chinese Remainder Theorem (CRT) [9] are needed after the overlap add stage to compute the convolution. When only one non-redundant modulus is used in the channel, the CRT logic is not needed.

IV. COMPLEXITY ANALYSIS

A Matlab computer program is used to search for the lowest area TMR and SRC NTT-based convolution systems over a range of word lengths and filter lengths. The conditions to be met in this search are that the moduli used cover the required wordlength, n; that the transform lengths be greater than the filter length Q; that all channels used in a particular implementation have different transform lengths. The program also searches for RNS combinations of moduli which are more area efficient than a single modulus for a given wordlength and filter length. In an RNS implementation the moduli used must be relatively

prime.

The moduli (M), transform lengths (L) and corresponding kernels (ω) used in the search are given in Table 1. It can be seen that all NTT kernels used are either of the form 2^k , requiring two additions per butterfly, or $\pm 2^k$ ($2^{M/2} \pm 1$), which require three additions per butterfly [8]. In the latter case, even powers of the kernel are powers of 2 and only one stage in an NTT requires odd powers of ω, involving three additions per butterfly.

The transform length and wordlength in a particular channel has implications for the complexity of the twiddle factor ω as can be seen from Table 1. The Matlab program gives the combinations which produce the minimum area per unit output given the above conditions.

The areas of the implementations are based on the areas of the forward NTTs, multiplication stages and the inverse NTTs. It is assumed that the filter coefficients are pre-computed. The number of adds in the NTT is calculated as the product of the number of adds per butterfly, the number of stages and the number of butterflies per stage.

For example, to implement an NTT with modulus $2^{12}+1$ and transform length 16, the kernel, $2^4(2^6 - 1)$ is needed. There are 4 stages and 8 butterflies per stage but one stage has 3 additions per butterfly and three stages have 2 additions per butterfly. This gives $3\times1\times8 + 2\times3\times8 = 72$ adds for the NTT and a total of 144 adds for the convolution together with 16 multiplications.

The Gate Equivalent (GE) area for addition modulo 2^n+1 is $9n/2\log_2 n+n/2+6$ [11] and for multiplication is $8n^2 + n - 2$ + the area of one addition [12]. The area is calculated on the basis of GEs per unit output. Following [9], the area per unit output for a channel is calculated as the total area for the channel / (N-Q+1), where N is the transform length associated with the particular modulus used in that channel and Q is the filter length.

Added to this is the cost of the extra Q-1 additions for every N-Q+1 outputs needed for the overlap add method. The cost of duplicating the final stage of the INTT need only be added to one of channels

The cost of the voting system for TMR and the cost of the logic needed to distinguish the error patterns in SRC are not included in this analysis. However, it is unlikely that there would be sufficient differences to affect the area comparisons. The area

of conversion to and from RNS is not considered since in our system, in instances where RNS is found to be the most efficient implementation, it is used for both TMR and SRC and therefore does not influence the percentage savings The area estimation results are shown in Table 3. The percentage column gives the percentage saving for Single Redundant Channel compared to TMR. The word length, transform length couples give the most area efficient combination of moduli and transform length for the particular word length and filter length. In the case of TMR, two couples indicate that an RNS with two moduli is the most efficient implementation. In the case of SRC, the moduli and transform lengths are given in two columns, one for the basic channel and one for the redundant channel

Consider a word length of 20 bits and a sequence of length 27. In this case, the most efficient implementation was found to be the two moduli combination of modulus $2^8 + 1$ with transform length 32, (8,32) and modulus $2^{16} + 1$ with

transform length 64, (16,64). This combination is used for TMR and as the basic, non-redundant channel in SRC. The Chinese Remainder Theorem (CRT) is needed in both cases to reconstitute the results. Redundancy is provided in the case of SRC with modulus $2^{32} + 1$ and transform length 128, giving three different transform lengths for the purpose of pattern matching. The final stage of the INTT is duplicated in the redundant channel. Comparison of the basic and redundant outputs will determine whether an error has occurred or not.

In this example, if an error is detected in the CRT reconstituted result from the first two moduli (basic channel), we select the third (redundant channel). If the error occurs in the third modulus we use the CRT reconstituted result from the first two

As can be seen from Table 3, significant savings can be made in many cases, particularly for large word lengths. In the case of filter lengths 27 and 31 with word lengths 12 and 16, the available moduli and transform lengths for the proposed system give a greater area than that of TMR. This is due to the extremely efficient system, modulus $2^{16} + 1$ and transform length 64, used in the TMR implementation in these cases.

V. CONCLUSIONS

An efficient soft error tolerant NTT based convolution system is proposed. The structure of the NTT is used in the design of the system that is capable of detecting and correcting up to 100% of

Table 3: Area comparisons between TMR and the proposed SRC.
Q=Filter length; n = wordlength as in 2^n+1 ; (n,T/L) = wordlength, transform length pairs; %=Percentage saving=((TMR-SRC)/TMR) *100

Q	n	TMR		SRC			%
		(n,T/L)	Area(GE)	(n,T/L)		Area(GE)	
				Basic channel	Redund-ant. channel		
19	12	(16,64)	26,609	(16,64)	(16,32)	22,194	17
	16	(16,64)	26,609	(16,64)	(16,32)	22,194	17
	20	(8,32)(16,64)	40,326	(8,32)(16,64)	(32,128)	37,980	6
	24	(8,32)(16,64)	40,326	(8,32)(16,64)	(32,128)	37,980	6
	28	(32,64)	71,017	(32,64)	(32,128)	49,686	30
	32	(32,64)	71,017	(32,64)	(32,128)	49,686	30
23	12	(16,64)	29,229	(16,64)	(16,32)	28,327	3
	16	(16,64)	29,229	(16,64)	(16,32)	28,327	3
	20	(8,32)(16,64)	48,575	(8,32)(16,64)	(32,128)	41,867	14
	24	(8,32)(16,64)	48,575	(8,32)(16,64)	(32,128)	41,867	14
	28	(32,128)	73,255	(32,64)	(32,128)	53,166	27
	32	(32,128)	73,255	(32,64)	(32,128)	53,166	27
27	12	(16,64)	32,401	(16,64)	(32,128)	36,969	-14
	16	(16,64)	32,401	(16,64)	(32,128)	36,969	-14
	20	(8,32)(16,64)	64,879	(8,32)(16,64)	(32,128)	48,738	25
	24	(8,32)(16,64)	64,879	(8,32)(16,64)	(32,128)	48,738	25
	28	(32,128)	76,215	(32,64)	(32,128)	57,256	25
	32	(32,128)	76,215	(32,64)	(32,128)	57,256	25
31	12	(16,64)	36,319	(16,64)	(32,128)	39,432	-9
	16	(16,64)	36,319	(16,64)	(32,128)	39,432	-9
	20	(32,128)	79,417	(32,64)	(32,128)	62,158	22
	24	(32,128)	79,417	(32,64)	(32,128)	62,158	22
	28	(32,128)	79,417	(32,64)	(32,128)	62,158	22
	32	(32,128)	79,417	(32,64)	(32,128)	62,158	22

isolated soft errors. When compared to TMR, the proposed system gives significant area saving in most cases examined and up to 30% in the case of large bit widths.

The authors' future work includes application of a modified version of the scheme to FFT based convolution systems. This can be done by using two FFTs of different lengths for the basic and redundant channel and a pattern matching for error correction that is similar to the one presented in this paper for NTTs.

REFERENCES

[1] R. Baumman "Soft errors in advanced computer systems" IEEE Design and Test of Computers, vol .22, issue 3, May - June 2005, pp. 258 – 266.

[2] M. Nicolaidis "Design for soft error mitigation" IEEE Transactions on Device and Materials Reliability vol. 5, issue 3, Sept. 2005, pp. 405 – 418.

[3] A. Reddy and P Banarjee "Algorithm-based fault detection for signal processing applications", IEEE Transactions on Computers, vol. 39, issue 10, Oct. 1990, pp 1304 – 1308.

[4] C.A. Agarwal and C.S. Burrus "Fast Convolution Using Fermat Number Transforms with Applications to Digital Filtering", IEEE Transactions on Acoustics, Speech and Signal Processing, vol. 22, issue 2, April 1974, pp.87 – 97.

[5] P.E. Beckmann and B. R. Musicus "Fast fault-tolerant digital convolution using a polynomial residue number system" IEEE Transactions on Signal Processing vol. 41, issue 7, July 1993, pp. 2300 – 2313.

[6] S. Sundaram and C.N. Hadjicostis "Fault-Tolerant Convolution Via Chinese Remainder Codes Constructed From Non-Coprime Moduli" IEEE Transactions on Signal Processing, vol. 56, issue 9, Sept. 2008, pp. 244 – 4254.

[7] A.B. O'Donnell and C.J. Bleakley "Area efficient fault tolerant convolution using RRNS with NTTS and WSCA", Electronics Letters vol. 44, issue 10, May 2008, pp. 648 – 649.

[8] T. Toivonen and J. Heikkila "Video filtering with Fermat number theoretic transforms using residue number system" IEEE Transactions on Circuits and Systems for Video Technology, vol. 16, issue 1, Jan 2006, pp. 92 – 101.

[9] R. Conway "Modified Overlap Technique Using Fermat and Mersenne Transforms", IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 53, issue 8, Aug. 2006, pp. 632 - 636.

[10] H.E Etzel and W.K Jenkins "Redundant Residue Number Systems for Error Detection and Correction in Digital Filters", IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. ASSP-28, issue 5, Oct 1980, pp 538-545.

[11] H.T. Vergos and C. Efstathiou and D. Nikolos "Diminished-one modulo 2^n+1 adder design", IEEE Transactions on Computers, vol. 51, issue 12, Dec. 2002, pp.1389 – 1399.

[12] C. Efstathiou, H.T. Vergos, G. Dimitrakopoulos and D. Nikolos "Efficient diminished-1 modulo $2^n + 1$ multipliers" IEEE Transactions on Computers, vol. 54, issue 4, April 2005, pp.491 – 496.