

TK
S102
B11

www.udsspace.uds.edu.gh

UNIVERSITY FOR DEVELOPMENT STUDIES

FOE - 000006

UNIVERSITY FOR DEVELOPMENT STUDIES

**EFFICIENT RESIDUE TO BINARY CONVERTERS
FOR SOME POWERS OF TWO MODULI SETS**

EDEM KWEDZO BANKAS



2013

UNIVERSITY FOR DEVELOPMENT STUDIES

EFFICIENT RESIDUE TO BINARY CONVERTERS FOR SOME POWERS OF TWO MODULI SETS

BY

EDEM KWEDZO BANKAS [B.Ed. (Maths), M.Ed (Computer Edu. & Tech)]

(UDS/DMS/0002/10)

THESIS SUBMITTED TO THE DEPARTMENT OF MATHEMATICS,
FACULTY OF MATHEMATICAL SCIENCES, UNIVERSITY FOR
DEVELOPMENT STUDIES, IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE AWARD OF DOCTOR OF PHILOSOPHY
DEGREE IN COMPUTATIONAL MATHEMATICS



SEPTEMBER, 2013

DECLARATION

Student

I hereby declare that this thesis is the result of my own original work and that no part of it has been presented for another degree in this University or elsewhere




Mr. Edem Kwedzo Bankas

7th November, 2013

Supervisor

I hereby declare that the preparation and presentation of the thesis was supervised in accordance with the guidelines on supervision of thesis laid down by the University for Development Studies.



Prof. Kazeem Alagbe Gbolagade

7th November, 2013



DEDICATION

This thesis is dedicated to God Almighty and all members of my family.



ACKNOWLEDGEMENT

I most especially would like to express my profound gratitude to my supervisor, Prof. dr. Kazeem Alagbe Gbolagade, who is a mentor par excellence. He has motivated me to be a better researcher. He always encouraged me and taught me to always do the right thing and to be my self. I have learnt a lot from his philosophy of life that patience, hardwork, and dedication to duty helps in sumounting any situation we are confronted with, both in life and science. I am very grateful to him for mentoring me as his academic son. In addition, he encouraged me to be more critical in analysing issues and has taught me the art of quality technical writing which has immeasurably contributed to the successful completion of this thesis. I very much appreciate his insightful experience and editorial assistance.

I very much appreciate the valuable feedback and comments I received from Dr. Ibrahim Yakubu Seini after reading through the thesis.

I am ever thankful to Ernest Beyuo for the assistance and hospitality he offered to me when I first arrived in Navrongo. Also, I wish to say a big thank you to Raymond Nero for all his efforts.



I am very grateful to all the staff and Lecturers of the Department of Computer Science, University for Development Studies (UDS), for their numerous support. Particularly, I would like to thank Mr. David Sankala, Head of Computer Science Department, UDS, for his wonderful encouragements. Additionally, I wish to express my heart felt appreciation to Dr. Stephen Twum, the Head of Mathematics Department, UDS, for his continual support and encouragements. I would like to thank all other staff and lecturers of UDS, too numerous to mention for their abundant support.

I would like to specially say a big thank you to the following people: Mr. Emmanuel Nartey, Frank Gbadago, Clemence Fomevor, Emmanuel Kutorglo, Emmanuel Botchway, Michael Ekoe, Peter Kwame Werekoh, Marcel Boamah, Baba Seidu, Francis Atsugah, and Rev. Fr. Samuel Ezar Bulu.

I am very much grateful to my parents Mrs. Grace Appiah-Bankas, late Mr. Joseph Strong-Bow Kwadzo Bankas, and all the members of my family, particularly my siblings for their wonderful support and prayers in more difficult times.

Finally, I am very thankful to Joyce Klutse for her outstanding friendship, support, prayers, and love.





ABSTRACT

In this thesis, a number of efficient Residue Number System (RNS) to binary converters have been proposed. The traditional Chinese Remainder Theorem (CRT), New CRT, and the Mixed Radix Conversion (MRC) schemes are employed. First, the traditional CRT is modified in order to obtain an efficient reverse converter for the popular length three moduli set $\{2^n, 2^n - 1, 2^n + 1\}$. Second, the modified scheme for the moduli set $\{2^n, 2^n - 1, 2^n + 1\}$ is hybridized with the MRC in order to obtain an efficient converter for the moduli set $\{2^n, 2^n - 1, 2^n + 1, 2^{2n+1} - 1\}$. Additionally, an efficient RNS to binary converter is presented for the moduli set $\{2^n, 2^{2n+1} - 1, 2^{2n+2} - 1\}$ in order to cater for applications requiring larger dynamic range. The following novel moduli sets have also been proposed $\{2^{2n} - 1, 2^n, 2^{2n+1} - 1\}$, $\{2^{2n}, 2^{2n+1} - 1, 2^{2n} - 1\}$, $\{2^{2n+1} - 1, 2^{2n+1}, 2^{2n} - 1\}$, and $\{2^{2n}, 2^{2n} + 1, 2^{2n} - 1\}$. These moduli sets exhibit interesting features that make them applicable in building efficient RNS to binary converters. The associated RNS to binary converters for the proposed moduli sets have also been presented. All the proposed converters in this thesis are purely adder based and memoryless. The performance of the proposed converters are evaluated both theoretically and experimentally. The obtained results indicate substantial

improvement over the best known equivalent state of the art converters in terms of hardware resources and conversion time.





TABLE OF CONTENTS

Declaration	ii
Dedication	iii
Acknowledgement	iv
Abstract	vi
Table of Contents	viii
List of Tables	xiii
List of Figures	xv
Acronyms	xvii
1 INTRODUCTION	1
1.1 Background Studies on Reverse Conversion	4
1.2 Literature Survey and Problem Statement	9
1.3 Major Contributions and Thesis Overview	14



2	A MODIFIED CRT SCHEME FOR THE MODULI SET $\{2^N, 2^N - 1, 2^N + 1\}$	20
2.1	Background	21
2.2	Proposed Reverse Conversion Algorithm	22
2.3	Hardware Realization	30
2.4	Performance Analysis	31
2.4.1	Theoretical Analysis	31
2.4.2	FPGA Experimentation	33
2.5	Conclusion	34
3	A UNIFIED CRT AND MRC SCHEME FOR THE $\{2^N, 2^N + 1, 2^N - 1, 2^{2N+1} - 1\}$ MODULI SET	37
3.1	Background	39
3.2	Proposed Algorithm	40
3.3	Hardware Realization	45
3.4	Performance Analysis	48
3.4.1	Theoretical Analysis	48
3.4.2	FPGA Experimentation	49
3.5	Conclusion	50
4	A NEW $5N$ BIT DR MODULI SET $\{2^{2N} - 1, 2^N, 2^{2N+1} - 1\}$	54
4.1	Background	55



4.2	New $\{2^{2n} - 1, 2^n, 2^{2n+1} - 1\}$ Moduli Set and its Reverse Converter . .	56
4.3	Hardware Realization	62
4.4	Performance Evaluation	64
4.4.1	Theoretical Analysis	64
4.4.2	Experimental Analysis	65
4.5	Conclusion	66
5	MRC ADDER BASED CONVERTER FOR	
	$\{2^N, 2^{2N+1} - 1, 2^{2N+2} - 1\}$ MODULI SET	69
5.1	Background	71
5.2	Proposed Reverse Conversion Algorithm	73
5.3	Hardware Realization	79
5.4	Performance Comparison	80
5.4.1	Theoretical Evaluation	80
5.4.2	Experimental Analysis	82
5.5	Conclusion	84
6	EFFICIENT CONVERTER FOR A NEW	
	$\{2^{2N+1} - 1, 2^{2N}, 2^{2N} - 1\}$ MODULI SET	87
6.1	Background	89
6.2	Proposed New Moduli Set and Reverse Conversion Algorithm	90
6.3	Hardware Realization	96
6.4	Performance Comparison	97



6.4.1	Theoretical Analysis	97
6.4.2	Experimental Analysis	99
6.5	Conclusion	101
7	A NOVEL MODULI SET $\{2^{2N+1} - 1, 2^{2N+1}, 2^{2N} - 1\}$	104
7.1	Background	105
7.2	$\{2^{2n+1} - 1, 2^{2n+1}, 2^{2n} - 1\}$ Moduli set	107
7.3	New CRT I Based Converter	108
7.4	Hardware Implementation	114
7.5	Performance Analysis	117
7.5.1	Theoretical Evaluation	117
7.5.2	Experimental Evaluation	118
7.6	Conclusion	120
8	A $6N$ BIT DYNAMIC RANGE MODULI SET CONTAINING $(2^K + 1)$ MODULUS	124
8.1	Background	126
8.2	New Moduli Set with Reverse Converter	127
8.3	Hardware Realization	132
8.4	Performance Analysis	133
8.4.1	Theoretical Evaluation	133
8.4.2	Experimental Evaluation	135
8.5	Conclusion	138

9 SUMMARY AND CONCLUSIONS	140
9.1 Summary	141
9.2 Major Contributions	146
9.3 Future Research Directions	148
References	150
Appendix	163
A List of Publications	163



List of Tables

2.1	Hardware Requirement of the proposed converter	32
2.2	Hardware Complexity and Delay Comparison	33
2.3	Experimental Delay and Area Comparison for $\{2^n, 2^n - 1, 2^n + 1\}$. .	36
3.1	Area and Delay Comparison for $\{2^n, 2^n + 1, 2^n - 1, 2^{2n+1} - 1\}$	49
3.2	Experimental Delay and Area Comparison for $\{2^n, 2^n + 1, 2^n - 1, 2^{2n+1} - 1\}$	51
3.3	Area-Time square ($\Delta\tau^2$) Comparison	51
4.1	Area-Delay Comparison	65
4.2	Experimental Delay and Area Comparison for $\{2^{2n} - 1, 2^n, 2^{2n+1} - 1\}$	66
5.1	Theoretical Delay and Area Comparison	83
5.2	Experimental Delay and Area Comparison for $\{2^n, 2^{2n+1} - 1, 2^{2n+2} - 1\}$	83
6.1	Theoretical Delay and Area Comparison	99
6.2	Experimental Delay and Area Comparison	102
7.1	Theoretical Area and Delay Comparison	118
7.2	Experimental Delay and Area Comparison for $\{2^{2n+1} - 1, 2^{2n+1}, 2^{2n} - 1\}$	119



7.3	Area-Time square ($\Delta\tau^2$) Comparison	119
8.1	Theoretical Area and Delay Comparison	136
8.2	Experimental Delay and Area Comparison for $\{2^{2n}, 2^{2n} + 1, 2^{2n} - 1\}$.	137





List of Figures

2.1	Block diagram for the Proposed Converter	31
2.2	The Area Comparison of Converters	34
2.3	The Delay Comparison of Converters	35
3.1	Proposed hardware Structure	47
3.2	The Area Comparison of Converters	50
3.3	The Delay Comparison of Converters	52
3.4	The Area Time square metric Comparison of Converters	53
4.1	Block diagram for the Proposed Reverse Converter	63
4.2	The Area Comparison of Converters	67
4.3	The Delay Comparison of Converters	68
5.1	Block diagram for the Proposed Converter	81
5.2	The Delay Comparison of Converters	84
5.3	The Area Comparison of Converters	85
6.1	Block diagram for the Proposed Converter	98

6.2	The Area Comparison of Converters	100
6.3	The Delay Comparison of Converters	101
7.1	Block diagram for the Proposed Converter	116
7.2	The Area Comparison of Converters	120
7.3	The Delay Comparison of Converters	121
7.4	The Area Time square metric Comparison of Converters	122
8.1	Proposed Cost Efficient Reverse Converter	134
8.2	Proposed Speed Efficient Reverse Converter	135
8.3	The Delay Comparison of Converters	137
8.4	The Area Comparison of Converters	138





ACRONYMS

RNS	Residue Number System
GCD	Greatest Common Divisor
CRT	Chinese Remainder Theorem
MRC	Mixed Radix Conversion
MRD	Mixed Radix Digits
DSP	Digital Signal Processing
ROM	Read Only Memory
CSA	Carry Save Adder
CPA	Carry Propagate Adder
EAC	End Around Carry
DCT	Discrete Cosine Transform
FPGA	Field Programmable Gate Array
VHDL	VHSIC Hardware Description Language
DR	Dynamic Range



CHAPTER 1

INTRODUCTION

Over the past decades, there has been a continuing research interest in improving the speed, reducing the area cost and power consumption of digital systems. One of the challenges in improving the performance of digital systems from the computational point of view is the carry propagation problem, which is characteristic of conventional number systems such as binary and decimal number systems. This is due to the fact that carry propagation limits the performance of arithmetic operations. This carry propagation problem is the major contributor to the internal delay of processors built with the conventional number systems (Szabo and Tanaka, 1967), (Parhami, 2000), (Mohan, 2002), (Omondi and Premkumar, 2007).

In literature, there are two broad categories of proposed solution to the carry propagation problem. First, the use of fast methods to compute the carries, e.g., carry look ahead adders, carry skip, prefix calculation and the like. Second, the use of an alternative number system with special carry characteristics, e.g., Redundant (Signed) Digit Number Systems and Residue Number System (RNS) (Szabo and

Tanaka, 1967), (Parhami, 2000), (Mohan, 2002), (Omondi and Premkumar, 2007). The use of fast methods to compute the carries require additional cost. Thus, the growing recognition and choice of unconventional number systems in the design of cost effective processors. RNS is an example of unconventional number system (unweighted number system) with interesting carry characteristics.

RNS is a non weighted carry free number system defined by a set of relatively prime integers $\{m_1, m_2, m_3, \dots, m_n\}$ called the moduli, such that $\gcd(m_i, m_j) = 1$ for $i \neq j$, where \gcd means the greatest common divisor of m_i and m_j . The Dynamic Range (DR) of the system is $[0, M)$, where $M = \prod_{i=1}^n m_i$, such that any integer $X \in [0, M)$ has a unique RNS representation given by an ordered set of residues $X = (x_1, x_2, x_3, \dots, x_n)$, $|X|_{m_i} = x_i$, $i = 1, 2, \dots, n$. For the sake of simplicity, we denote in this thesis $|X|_{m_i}$ to mean $X \bmod m_i$.

An operation \otimes over an RNS is defined as $(c_1, c_2, c_3, \dots, c_n) = (a_1, a_2, a_3, \dots, a_n) \otimes (b_1, b_2, b_3, \dots, b_n)$, where $c_i = |a_i \otimes b_i|_{m_i}$. Here, the computation of each c_i depends on only a_i, b_i and m_i . The resulting c_i s are all computed in parallel in their respective arithmetical units m_i often referred to as channels. In principle, there exist no carry propagation between the channels during the process of computation, and therefore provides the support for carry free addition, borrow free subtraction, and digit by digit multiplication without partial products. It can be seen that, there is no ordering significance between the digits in RNS. The output of any operation such as addition, subtraction, or multiplication depends absolutely on the corresponding digit of the



operands involved. This phenomenon results in high speed arithmetic unit realization in RNS based processors.

RNS has found application in many areas where addition and multiplication dominate the computational process, such as Digital Signal Processing (DSP) e.g, Digital Filtering, (Nussbaumer, 1976), (Jenkins and Leon, 1977), (Jenkins, 1978), (Conway and Nelson, 2004), (Toivonen and Heikkila, 2006), (Pontarelli et al., 2010), (Chalivendra et al., 2011). Correlation (Psaltis and Casasent, 1979), (Chen et al., 1990), Convolution (Beckmann and Musicus, 1993), Discrete Cosine Transform, (Fernandez et al., 2000),

It is interesting to note that, despite all the features of RNS and its potential to achieving high speed computation in digital systems, it is yet to find a widespread utilization in designing general purpose processors. The following are the bottlenecks which are being addressed essentially by research: Magnitude comparison, Signed detection, Overflow detection, Moduli selection, Scaling, Division, Residue to binary (Reverse) Conversion, Binary to residue (forward) conversion and other Complex arithmetic operations.

RNS as a field of study has attracted lots of research attention because, if sufficient solution to many of the above mentioned bottlenecks are provided, then the dream of building general purpose processors based on RNS would become a reality. It must be noted that, most of the challenges of RNS rely heavily on effective Reverse Conversion. Reverse Conversion is the computational process of transforming a residue number representation to its equivalent weighted number. It is known to be



the most complex and critical component of the RNS processor and therefore require substantially fast converter designs. It is for this reason that this thesis explores the VLSI Characterization of effective Reverse Conversion based on either the Chinese Remainder Theorem (CRT), New CRT or the Mixed Radix Conversion (MRC).

The rest of this chapter is structured as follows: Section 1.1 provides the background information on RNS Reverse Conversion algorithms. In Section 1.2, related Literature survey is presented. Section 1.3, concludes the discussion in this chapter with the thesis overview and how the entire research work is organized and presented.

1.1 Background Studies on Reverse Conversion

Research into Reverse Conversion can be traced to the puzzle posed by the Chinese scholar Sun Tzu of the first century (Mohan, 2002), (Omondi and Premkumar, 2007) as : What is the value of a number that has the remainders 2, 3, and 2 when divided by the numbers 7, 5, and 3 respectively? In fact, the residues are 2, 3, and 2 while the moduli are 7, 5 and 3. According to literature, this puzzle happens to be the first ever documentation on number representation using multiple remainders, (Gbolagade, 2010). The answer to this puzzle, 23 is demonstrated in Sun Tzu's historic work (Mohan, 2002), (Omondi and Premkumar, 2007). In actual fact, the puzzle seeks to convert the RNS number $(2|3|2)_{RNS(7|5|3)}$ into its equivalent weighted number. Sun Tzu formulated an algorithm for transforming the residue numbers into its equivalent weighted number in his book (Mohan, 2002). In 1247, another Chinese Mathematician



Qin Jiushao, generalized Sun Tzu's algorithm into what is now appropriately called the Chinese Remainder Theorem (CRT), (Omondi and Premkumar, 2007). Later in 1734, Euler presented a formal proof for CRT (Mohan, 2002).

In the 19th century, Carl Friedrich Gauss set forth the CRT and the theory of RNS in his celebrated *Disquisitiones Arithmetical*, (Soderstrand et al., 1986). The 20th century saw the first ever RNS hardware built by Lehmer, Svoboda, and Valach in Czechoslovakia, by using the vacuum tube technology at the time, (Mohan, 2002), (Gbolagade, 2010). Between the 1950's and 1960's, various research work on the subject of modular arithmetic and on residue number systems were carried out at the Harvard Computation Laboratory, (Aiken and Semon, 1959), Lockheed Missiles and space Company (Huang et al., 1985), (Tanaka, 1962), Westinghouse, (Slotnick, 1962). In those years, the main interest was focused on the design and implementation of On Board Processors for space, avionic and military applications characterized by high speed and very high reliability. Szabo and Tanaka (1967) as well as Watson and Hastings (1967) contributed by documenting the description of residue arithmetic, its basic principles, shortcomings together with a comprehensive metadata of knowledge and research efforts by some researchers in their books.

However, between 1967 and 1977, there was an impediment on RNS research output. This was because, RNS researchers and processor designers were to some extent not convince of ever realizing the dream of building general purpose processors (Gbolagade, 2010). Perhaps due to the challenges that confronts RNS implementation. Then, Jenkins and Leon (1977) rejuvenated the interest of researchers by their efforts



in generating wonderful results and that gave another hope to the research community. Consequently, many fascinating outputs were generated by other researchers between 1977 and 1985. Many of these results were compiled by Soderstrand et al. (1986). The trend was continued by Mohan (2002), when he brought together RNS research articles up to 2001. From 2002 to date, interesting results have been reported by some authorities in the field. Among them are: Al-radadi and Siy (2002), Yang et al. (2005), Mohan (2007), Molahosseini and Navi (2007), Gbolagade and Cotofana (2008a), Gbolagade and Cotofana (2008c), Gbolagade and Cotofana (2008e), Lin et al. (2008), Gbolagade et al. (2009), Gbolagade (2010), Chalivendra et al. (2011), Bankas and Gbolagade (2012).

As mentioned earlier, RNS architecture consists of three main components, namely, a binary-to-residue converter, residue arithmetic units, and a residue-to-binary converter. It is an undeniable fact that, the residue-to-binary converter is the most challenging and complex part of any RNS architecture. In this regard, extensive research work on designing efficient reverse converter is paramount to create the enabling environment for the realization of the ultimate dream of building RNS based processors. Most of the research work on residue-to-binary converters are based on either the CRT, New CRT or MRC, (Molahosseini and Navi, 2007), (Lin et al., 2008), (Gbolagade and Cotofana, 2008d), (Gbolagade and Cotofana, 2008b), (Gbolagade and Cotofana, 2009a), (Gbolagade and Cotofana, 2009a), (Gbolagade and Cotofana, 2009b), (Gbolagade, 2009b), (Gbolagade et al., 2011), (Gbolagade, 2010).



Generally, given a moduli set $\{m_i\}_{i=1,k}$, the residues (x_1, x_2, \dots, x_k) can be converted into its corresponding weighted number X , using the CRT as follows (Szabo and Tanaka, 1967):

$$X = \left| \sum_{i=1}^k M_i \left| M_i^{-1} \right|_{m_i} x_i \right|_M, \quad (1.1)$$

where $M = \prod_{i=1}^k m_i$, $M_i = \frac{M}{m_i}$ and M_i^{-1} is the multiplicative inverse of M_i with respect to m_i .

The main bottleneck regarding the use of CRT has to do with the computation of modulo M operation which can be time consuming and expensive in terms of area and power consumption for large M .

An alternative method, the MRC, relatively does not involve the large modulo- M computation present in CRT. However, the problem with the MRC is that it is naturally a sequential process. Suppose we have a residue number (x_1, x_2, \dots, x_k) with the corresponding defined moduli set $\{m_i\}_{i=1,k}$, its decimal equivalent can be computed as (Szabo and Tanaka, 1967):

$$X = a_1 + a_2 m_1 + a_3 m_1 m_2 + \dots + a_n m_1 m_2 m_3 \dots m_{k-1}, \quad (1.2)$$



where the Mixed Radix Digits (MRDs) $a_{i=1,k}$ are given as follows:

$$\begin{aligned}
 a_1 &= x_1 \\
 a_2 &= \left| (x_2 - a_1) |m_1^{-1}|_{m_2} \right|_{m_2} \\
 a_3 &= \left| \left((x_3 - a_1) |m_1^{-1}|_{m_3} - a_2 \right) |m_2^{-1}|_{m_3} \right|_{m_3} \\
 &\vdots \\
 &\vdots \\
 &\vdots \\
 a_k &= \left| \left(\left((x_k - a_1) |m_1^{-1}|_{m_k} - a_2 \right) |m_2^{-1}|_{m_k} \right. \right. \\
 &\quad \left. \left. - \dots - a_{k-1} \right) |m_{k-1}^{-1}|_{m_k} \right|_{m_k}
 \end{aligned} \tag{1.3}$$

For the MRDs $a_i, 0 \leq a_i < m_i$, any positive number in the interval $\left[0, \prod_{i=1}^k m_i - 1\right]$ can be uniquely represented. It can be observed from Equation (1.3) that, the calculation of $a_{i=1,k}$ is done using only arithmetic mod m_i as against the computation of large mod M in CRT. This is therefore a major advantage for MRC.

Further, Wang (1998) proposed the New CRTs I, II, and III. For the purpose of this thesis, we concentrate on New CRT I. Given a moduli set $\{m_1, m_2, m_3, m_4\}$, its equivalent weighted number X can be converted from its residue representation (x_1, x_2, x_3, x_4) as follows:



$$X = x_1 + m_1 |k_1(x_2 - x_1) + k_2 m_2(x_3 - x_2) + k_3 m_2 m_3(x_4 - x_3)|_{m_2 m_3 m_4}, \quad (1.4)$$

where

$$|k_1 m_1|_{m_2 m_3 m_4} = 1, \quad (1.5)$$

$$|k_2 m_1 m_2|_{m_3 m_4} = 1, \quad (1.6)$$

$$|k_3 m_1 m_2 m_3|_{m_4} = 1. \quad (1.7)$$

1.2 Literature Survey and Problem Statement

As mentioned earlier, moduli set selection is a major issue of importance since the complexity and the speed of the RNS processor depend on the selected moduli set, (Gbolagade, 2009a), (Gbolagade et al., 2009), (Bankas and Gbolagade, 2012). It has been well established that, powers-of-two related moduli sets simplify the required arithmetic operations and generates efficient hardware implementations of the RNS architecture. Moreover, the utilization of powers-of-two moduli sets result in the realization of ROM-less reverse converters. Extensively, several structures have been proposed to perform the reverse conversion for different moduli sets.

Many efficient reverse conversion algorithms, which are adder-based and memoryless have been proposed for the moduli set $\{2^n, 2^n - 1, 2^n + 1\}$. This moduli set has





gained popularity and plays an important role in digital signal processing. Some of the conversion methods used in literature are based on the use of special formulae, (Gallagher et al., 1997), others use either the traditional CRT, (Andraos and Ahmad, 1988), (Piestrak, 1995), (Conway and Nelson, 2003), (Wang et al., 2003), or New CRT (Wang et al., 2002). It has been observed that, Andraos and Ahmad (1988) employed compact form of multiplicative inverse to simplify the CRT. Based on similar principles, Piestrak (1995) presented an adder based reverse converter, which is an improvement on Andraos and Ahmad (1988). Another area-delay efficient converter based on New CRT was proposed by Wang et al. (2002) where three different converters using either $2n$ bits or n bit adders were presented. The $2n$ bit adder based converter is faster and requires about half the hardware required by those presented in Andraos and Ahmad (1988) and Piestrak (1995). Further, Bi et al. (2004) proposed new theorems which further reduces the modulo operation required by the New CRT. The modulo part of the residue to binary converter for $\{2^n, 2^n - 1, 2^n + 1\}$ based on the proposed theorems was also presented. The FPGA implementation results indicate that, the proposed converter is twice faster and requires 50% less hardware and power for the modulo operation than the converters in Wang et al. (2002) (Gbolagade, 2010). Given that, the speed and the hardware complexity of the resulting reverse converters proposed for the moduli set $\{2^n + 1, 2^n, 2^n - 1\}$ can be further reduced because of its number theoretic properties,

- Can we obtain a more efficient reverse converter for the moduli set $\{2^n, 2^n - 1, 2^n + 1\}$?
- In terms of speed and area cost for the reverse converter, which is more pronounced?

The Dynamic Range (DR) of the Moduli set $\{2^n, 2^n - 1, 2^n + 1\}$ is $3n$ bits. In recent years, due to the increasing demand for some applications that require larger DR, one option that became available was to extend and enhance the moduli set $\{2^n, 2^n - 1, 2^n + 1\}$ to a length 4 moduli set by $(2^{2n+1} - 1)$, i.e., $\{2^n - 1, 2^n, 2^n + 1, 2^{2n+1} - 1\}$ which has a $5n$ bit DR, presented by Molahosseini et al. (2010). The proposed reverse converter was based on the New CRT II. In fact, the moduli set $\{2^n - 1, 2^n, 2^n + 1, 2^{2n+1} - 1\}$ was presented as an alternative to $\{2^n - 1, 2^n, 2^n + 1, 2^{n-1} - 1, 2^{n+1} - 1\}$ in Cao et al. (2007), and it was shown to have demonstrated better performance in terms of conversion delay and hardware requirements. Given that, the speed and hardware complexity of the resulting residue to binary converter can be further improved,

- Can we obtain an efficient reverse converter for $\{2^n - 1, 2^n, 2^n + 1, 2^{2n+1} - 1\}$ moduli set by hybridizing a modified CRT with MRC?

It is reported that, the moduli set $\{2^{2n} + 1, 2^n, 2^{2n} - 1\}$ was investigated by Hariri et al. (2008). It was demonstrated that the proposed residue to binary converter had better area and time complexity when compared with Cao et al. (2003) and



Hiasat (2003). In the same year, Molahosseini et al. (2008) presented a reverse converter for the general three moduli set $\{2^\alpha, 2^\beta - 1, 2^\beta + 1\}$ where $\alpha < \beta$ and their values are set in order to provide the desired DR. As a result, using their general moduli set, they proposed $\{2^{n+k}, 2^{2n} - 1, 2^{2n} + 1\}$ and $\{2^{2n-1}, 2^{2n+1} - 1, 2^{2n+1} + 1\}$ as alternatives to the moduli sets $\{2^{2n} + 1, 2^n, 2^{2n} - 1\}$, Hariri et al. (2008) and $\{2^n - 1, 2^n + 1, 2^{2n} - 2, 2^{2n+1} - 3\}$, Zhang and Siy (2008) respectively. This is due to the imbalance problem in the moduli sets in Hariri et al. (2008) and Zhang and Siy (2008) which is caused by the large bit-width differences between the various moduli and also, for the fact that, their proposed reverse converters achieved higher performance.

It is interesting to note that, the presence of $(2^n + 1)$ type modulus in a moduli set degrades the overall speed, thus increasing the processing time. The reason is that, multiplication by powers of 2 with respect to $(2^n + 1)$ is not as simple as circular left shift in $(2^n - 1)$ type modulus. To address this issue, many moduli sets such as $\{2^{2n+1} - 1, 2^{2n}, 2^n - 1\}$ have been presented (Gbolagade et al., 2009).

- Can we obtain an enhanced moduli set from $\{2^{2n+1} - 1, 2^{2n}, 2^n - 1\}$, and then design its efficient residue to binary converter?

Given that applications requiring larger DR are of practical interest, we present solution the following question:

- Can we obtain different length 3 moduli sets with $6n$ bit DR together with their efficient associated reverse conversion algorithms?



Recently, the moduli set $\{2^{2n+2} - 1, 2^{2n+1} - 1, 2^n\}$, Modiri et al. (2012) was proposed with its corresponding residue to binary converter. The proposal was based on the MRC, which requires a high conversion time. We resolve the following issue:

- Can we achieve a residue to binary converter with a more efficient conversion time and area cost when compared to the proposed converter presented in Modiri et al. (2012)?

Moreover, the requirement of moduli sets with larger DR by many DSP applications led to the extension of the length of moduli sets to four, five, or generally higher length moduli sets as one of the options. Examples are: $\{2^n - 1, 2^n, 2^n + 1, 2^{n+1} + 1\}$ (Bhardwaj et al., 1999), $\{2^n - 1, 2^n, 2^n + 1, 2^{n+1} + 1\}$ (Vinod and Premkumar, 2000), $\{2, 2^n - 1, 2^n + 2^{n-1} - 1, 2^{n+1} + 2^n - 1\}$ (Al-radadi and Siy, 2002), $\{2^n - 3, 2^n + 1, 2^n - 1, 2^n + 3\}$ and $\{2^n, 2^n + 1, 2^n - 1, 2^{n+1} + 1\}$ (Sheu et al., 2004), $\{2^n, 2^n + 1, 2^n - 1, 2^{n+1} - 1\}$ and $\{2^n, 2^n + 1, 2^n - 1, 2^{n+1} + 1\}$ (Mohan, 2007), $\{2^n - 1, 2^n + 1, 2^{2n} - 2, 2^{2n+1} - 3\}$, (Zhang and Siy, 2008), $\{2^n - 1, 2^n, 2^n + 1, 2^{2n+1} - 1\}$ and $\{2^n - 1, 2^n + 1, 2^{2n}, 2^{2n} + 1\}$, (Molahosseini et al., 2010), $\{2^k, 2^n - 1, 2^n + 1, 2^{n+1} - 1\}$ (Chalivendra et al., 2011), $\{2^k, 2^n - 1, 2^{n-1} - 1, 2^{n+1} - 1\}$ (Wesolowski et al., 2012), $\{2^n - 1, 2^n, 2^{n+1} - 1, 2^{n+1} + 2^n - 1\}$ (Quan et al., 2012). As stated earlier, the selection of moduli sets with their associated conversion methods greatly have an influence on the conversion time and hardware resources.



The issue then is, for larger DR moduli sets which can lead to efficient internal RNS arithmetic circuits, as well as high performance reverse converter, we provide solutions to the following pertinent issues:

- Can we obtain $6n$ bit DR moduli sets which are free from $(2^n + 1)$ modulus with their corresponding efficient reverse converter which can compete satisfactorily with similar existing larger DR moduli sets by vertically extending some powers of two modulus?
- Can we also obtain a $6n$ bit DR moduli set which may contain $(2^n + 1)$ modulus with its corresponding efficient reverse converter but can compete satisfactorily with similar existing larger DR moduli sets by vertically extending some powers of two modulus?

1.3 Major Contributions and Thesis Overview

In this thesis, we present several novel residue to binary converters based on either the CRT, New CRT, or MRC. Additionally, a number of moduli sets are proposed with their associated reverse converters. The major contributions and how they are organized in this thesis is as follows:

1. Chapter 2 presents a modified CRT Technique for the popular moduli set $\{2^n, 2^n + 1, 2^n - 1\}$. The proposed converter is memoryless and adder based. We performed both theoretical and FPGA experimentation of our scheme and



the state of the art. From the theoretical assessment, we deduced that, our converter is advantageous in terms of speed while requiring more hardware resources. The synthesis results obtained demonstrated vividly that, on the average, and contrary to the theoretical result, our proposed reverse converter is significantly better than the state of the art in terms of both speed and area resources.

2. In Chapter 3, the converter presented in Chapter 1 is hybridized with the MRC to obtain an effective converter for the moduli set $\{2^n, 2^n - 1, 2^n + 1, 2^{2n+1} - 1\}$. The proposed scheme is purely adder based and memoryless and is implemented in a two level design. The first level consist of deriving the equivalent weighted number of the residues (x_1, x_2, x_3) using the modified CRT algorithm for the popular moduli set $\{2^n, 2^n - 1, 2^n + 1\}$ presented in Chapter 1. Next, the resulting equivalent weighted number from the first level is hybridized with the fourth residue x_4 using the MRC with respect to the composite moduli set $\{2^{3n} - 2^n, 2^{2n+1} - 1\}$. We further simplified the resulting architecture to obtain an adder based and momoryless reverse converter that utilizes only Carry Save Adders (CSAs) and Carry Propagate Adders (CPAs). We compared our scheme with existing state of the art converter proposed by Molahosseini et al. (2010) both theoretically and experimentally.
3. In Chapter 4, we propose a new $5n$ bit DR moduli set $\{2^{2n} - 1, 2^n, 2^{2n+1} - 1\}$ derived from the four-moduli set $\{2^n, 2^n - 1, 2^n + 1, 2^{2n+1} - 1\}$ presented by



Molahosseini et al. (2010) by combining $2^n - 1$ and $2^n + 1$ into $2^{2n} - 1$. Next, we propose an efficient MRC based RNS-to-binary converter for the new moduli set. When compared to the converter proposed by Molahosseini et al. (2010), theoretically speaking, our converter outperforms the state of the art in terms of both hardware requirements and delay. Also, we performed an FPGA experimentation of our converter and that of $\{2^n, 2^n - 1, 2^n + 1, 2^{2n+1} - 1\}$ by Molahosseini et al. (2010) by describing the converters in VHDL using Xilinx.

4. Chapter 5 presents an efficient RNS to binary converter for the moduli set $\{2^n, 2^{2n+1} - 1, 2^{2n+2} - 1\}$ which contains low-cost moduli and has a larger dynamic range compared to other existing $(5n)$ bit DR state of the art moduli sets. The proposed reverse converter for the moduli set $\{2^n, 2^{2n+1} - 1, 2^{2n+2} - 1\}$ is based on MRC. Additionally, we further simplified the resulting architecture in order to obtain a reverse converter that utilizes only 1 level of CSA together with three CPAs, i.e., CPA1, CPA2 or CPA3, and CPA4. The proposed converter is purely adder based and memoryless. The performance of the proposed converter is evaluated both theoretically and experimentally by FPGA implementation.
5. In Chapter 6, we propose a new moduli set $\{2^{2n}, 2^{2n+1} - 1, 2^{2n} - 1\}$ which is an enhancement of the moduli set $\{2^n, 2^{2n+1} - 1, 2^{2n} - 1\}$ and its associated efficient RNS to binary converter. The proposed reverse converter is based on MRC. The divide and conquer approach was used to implement the MRC, where the moduli set is grouped into two. The first phase combines the first and the



second residue with respect to the subset $\{2^{2n}, 2^{2n+1} - 1\}$. In the second phase, the result of the first phase is combined with the third residue with respect to the composite moduli set $\{(2^{2n})(2^{2n+1} - 1), 2^{2n} - 1\}$. Additionally, we further simplified the resulting architecture in order to obtain a reverse converter that utilizes only 2 levels of CSAs together with three CPAs. Theoretical analysis reveals that our proposal has a delay of $(10n+4)t_{FA} + 2t_{MUX}$ with an area cost of $(12n + 2)FAs$ and $(2n)HAs$. Subsequently, we expressed the required hardware resources in terms of HA. Then for experimental comparison, we described our converter and those by Gbolagade et al. (2009) and Molahosseini et al. (2010) in VHDL and subsequently implemented them on an FPGA.

6. Chapter 7 we introduces a new moduli set $\{2^{2n+1} - 1, 2^{2n+1}, 2^{2n} - 1\}$ which also contains only low-cost moduli and has a large dynamic range. A novel and effective reverse converter is proposed based on the New CRT. Additionally, we further simplified the resulting architecture in order to obtain a reverse converter that utilizes 4 levels of CSAs, namely: CSA1, CSA2, CSA3, and CSA4 together with two CPAs, i.e., CPA1 or CPA2 and CPA3 or CPA4. The proposed converter is purely adder based and memoryless. Both theoretical and experimental assessment of our proposal against best state of the art were carried out. Theoretically speaking, our proposal has a delay of $(8n + 6)t_{FA} + 2t_{MUX}$ with an area cost of $(12n + 4)FA$ and $(6n - 1)HA$. We compared the proposed reverse converter with state of the art converters



$\{2^n - 1, 2^n, 2^n + 1, 2^{2n+1} - 1\}$ and $\{2^n - 1, 2^n + 1, 2^{2n}, 2^{2n+1} - 1\}$ proposed by Molahosseini et al. (2010) and Molahosseini and Navi (2010) respectively. The result therefore shows that our scheme is faster than the equivalent state of the art converter $\{2^n - 1, 2^n + 1, 2^{2n}, 2^{2n+1} - 1\}$, but requires some more area resources. Further, for the experimental comparison, we described our proposed reverse converter and the considered state of the art in VHDL and performed the implementation on an FPGA using a wide range of values on n . The synthesis results confirmed that of the theoretical assessment.

7. In Chapter 8, we propose a novel moduli set $\{2^{2n}, 2^{2n} + 1, 2^{2n} - 1\}$ with its corresponding reverse converter using the CRT. The moduli set is a $6n$ bit DR and therefore appropriate for applications requiring specifically $6n$ DR. We simplified the CRT to obtain an effective algorithm. Further, we reduced the resulting architecture in order to obtain a reverse converter that utilizes only two CSAs and a CPA. We performed both theoretical and experimental evaluation of our proposal. The theoretical analysis shows clearly the advantages of our moduli set and its associated reverse converter. This is confirmed by the experimental results. We described our scheme and those presented by Cao et al. (2003) and Molahosseini and Navi (2010) in VHDL and carried out the implementation on an FPGA using a wide range of values on n .



8. Chapter 9 presents the concluding remarks of this thesis. It summarizes the whole work of the thesis, highlights the major contributions and then provides future research directions.





CHAPTER 2

A MODIFIED CRT SCHEME FOR THE MODULI SET $\{2^N, 2^N - 1, 2^N + 1\}$

As mentioned in the first Chapter, there exist several moduli sets that have been investigated in literature, e.g., $\{2^n, 2^n - 1, 2^n + 1\}$ (Bi et al., 2004), (Andraos and Ahmad, 1988), (Piestrak, 1995), (Wang et al., 2002), $\{2^n, 2^n - 1, 2^{n+1} - 1\}$ (Mohan, 2007), (Lin et al., 2008), (Gbolagade et al., 2010b), $\{2^n, 2^n - 1, 2^{n-1} - 1\}$ (Hosseinzadeh et al., 2009). The moduli set $\{2^n, 2^n + 1, 2^n - 1\}$ is the most popular length three moduli set (Wang et al., 2003). It is interesting to note that several conversion methods have been proposed for the moduli set $\{2^n, 2^n + 1, 2^n - 1\}$ (Bi et al., 2004), (Andraos and Ahmad, 1988), (Piestrak, 1995), (Wang et al., 2002). Some of these proposed conversion methods use special formulae, others use either the traditional CRT or the New CRT (Wang et al., 2002). Andraos and Ahmad (1988) employed compact form of the relevant multiplicative inverses to simplify the CRT. Based on similar principles, Piestrak (1995) presented an adder based

reverse converter, which is an improvement on (Andraos and Ahmad, 1988). Another area-delay efficient reverse converter based on New CRT was later presented by (Wang et al., 2002). All the converters presented by Bi et al. (2004), Andraos and Ahmad (1988), Piestrak (1995), and Wang et al. (2002) are for the moduli set $\{2^n, 2^n + 1, 2^n - 1\}$. The speed, area, and the hardware complexity of the resulting reverse converters proposed for the moduli set $\{2^n, 2^n + 1, 2^n - 1\}$ can be further reduced since the moduli set supports several interesting number theoretic properties. In this chapter, we propose a novel high speed RNS reverse converter for the moduli set $\{2^n, 2^n + 1, 2^n - 1\}$. First, the proposed reverse converter is based on the simplification of the traditional CRT. We simplify further the resulting architecture in order to achieve a reverse converter that utilizes only Carry Save Adders (CSAs) and Carry Propagate Adders (CPAs).

The rest of the Chapter is organized as follows. Section 2.1 provides a brief background information. In Section 2.2, the proposed algorithm is presented. Section 2.3 describes the hardware realization of the proposed algorithm and Section 2.4 gives a performance comparison with the state of the art reverse converters. Finally, the Chapter is concluded in Section 2.5.

2.1 Background

Efficient high speed and low complexity data converters are required to convert numbers from binary-to-residue (B-R) or residue-to-binary (R-B) representations. A



high speed B-R converter is required at the front end of the system as well as a high speed R-B converter at the back end. Many of the difficult operations such as scaling, magnitude comparison, division, and sign detection invariably need conversion from RNS to binary representations (Cardarilli et al., 1998), (Dhurkadas, 1990), (Mohan, 2007) and therefore a major issue of concern.

Given a moduli set $\{m_i\}_{i=1,k}$ with DR, $M = \prod_{i=1}^k m_i$, the residues (x_1, x_2, \dots, x_k) can be converted into the corresponding decimal number X , using the well known CRT, as follows, (Szabo and Tanaka, 1967), (Mohan, 2002):

$$X = \left| \sum_{i=1}^k m_i |M_i^{-1}|_{m_i} x_i \right|_M \quad (2.1)$$

where, $M = \prod_{i=1}^k m_i$, $M_i = \frac{M}{m_i}$ and M_i^{-1} is the multiplicative inverse of M_i with respect to (w.r.t) m_i .

2.2 Proposed Reverse Conversion Algorithm

The proposed algorithm is formulated using the following theorems:

Theorem 2.1. *Given the moduli set $\{m_1, m_2, m_3\}$ with $m_1 = 2^n$, $m_2 = 2^n + 1$, and $m_3 = 2^n - 1$, the following hold true:*



$$|(m_2 m_3)^{-1}|_{m_1} = -1, \quad (2.2)$$

$$|(m_1 m_3)^{-1}|_{m_2} = 2^{n-1} + 1 = \frac{m_1}{2} + 1, \quad (2.3)$$

$$|(m_1 m_2)^{-1}|_{m_3} = 2^{n-1} = \frac{m_1}{2} \quad (2.4)$$

Proof. :

For Equation (2.2), we have $|-1 \times (m_2 m_3)|_{m_1} = |-1 \times (2^{2n} - 1)|_{2^n} = 1$,

similarly, in Equation (2.3) we have

$$\left| \frac{m_1}{2} + 1 \times (m_1 m_3) \right|_{m_2} = |(2^{n-1} + 1) \times (2^{2n} - 2^n)|_{2^{n+1}} = 1,$$

Again, for Equation (2.4) we have $\left| \frac{m_1}{2} \times (m_1 m_2) \right|_{m_3} = |2^{n-1} \times (2^{2n} + 2^n)|_{2^{n-1}} = 1$.

□

The following important relations are used in the proof of the subsequent theorem:

Given the moduli set $\{m_1, m_2, m_3\}$ with $m_1 = 2^n$, $m_2 = 2^n + 1$, $m_3 = 2^n - 1$, the following hold true:

$$m_3 = m_1 - 1, \quad (2.5)$$

$$m_2 = m_1 + 1, \quad (2.6)$$

$$m_2 = m_3 + 2, \quad (2.7)$$

$$m_1 m_2 = m_1 m_3 + 2m_1 \quad (2.8)$$



Theorem 2.2. *The decimal equivalent of the RNS number (x_1, x_2, x_3) with respect to the moduli set $\{m_1 = 2^n, m_2 = 2^n + 1, m_3 = 2^n - 1\}$ is computed as follows:*

$$X = m_1 \left\lfloor \frac{X}{m_1} \right\rfloor + x_1 \quad (2.9)$$

where, $\left\lfloor \frac{X}{m_1} \right\rfloor = |u_1 + u_2 + u_3|_{m_2 m_3}$, $u_1 = (\frac{1}{m_1} - m_2 + 1)x_1$, $u_2 = (\frac{m_1 m_2}{2} - 1)x_2$ and $u_3 = \frac{m_1 m_2}{2} x_3$

Proof. :

Using the traditional CRT in Equation (2.1) for $n = 3$, we have:

$$X = \left| M_1 |M_1^{-1}|_{m_1} x_1 + M_2 |M_2^{-1}|_{m_2} x_2 + M_3 |M_3^{-1}|_{m_3} x_3 \right|_M. \quad (2.10)$$

Substituting Equations (2.2), (2.3), (2.4) into Equation (2.10) and applying the appropriate expressions of M_1, M_2 , and M_3 we obtain:

$$X = \left| -m_2 m_3 x_1 + \frac{m_1 m_1 m_3 x_2}{2} + m_1 m_3 x_2 + \frac{(m_1 m_3 + 2m_1)(m_1 x_3)}{2} \right|_M \quad (2.11)$$

We proceed by substituting Equations (2.5), (2.6), and (2.7) into Equation (2.11) and simplifying to obtain:



$$X = \left| m_1 m_2 \left(-x_1 + \frac{m_3 x_2}{2} + x_2 + \frac{m_3 x_3}{2} \right) + m_1 m_2 x_3 - \frac{m_1 m_3 x_2}{2} - \frac{m_1 m_3 x_3}{2} - 2m_1 x_2 - m_1 x_3 + m_2 x_1 \right|_M \quad (2.12)$$

$$X = \left| m_1 m_2 \left[\frac{m_3}{2} (x_2 + x_3) + (x_2 - x_1) \right] + m_1 m_2 x_3 + m_2 x_1 - m_1 (x_3 + 2x_2) - \frac{(m_1 m_3)(x_2 + x_3)}{2} \right|_M \quad (2.13)$$

Substituting $m_1 m_3$ in Equation 2.8 into Equation 2.13 and re-arranging we obtain:

$$X = \left| m_1 m_2 \left[\frac{m_3}{2} (x_2 + x_3) + \frac{x_2 + x_3}{2} - x_1 \right] + m_1 (x_2 + x_3) - m_1 (x_3 + 2x_2) + m_2 x_1 \right|_M, \quad (2.14)$$

further simplification gives:

$$X = \left| m_1 m_2 \left[\frac{m_3}{2} (x_2 + x_3) + \frac{x_2 + x_3}{2} - x_1 \right] + m_2 x_1 - m_1 x_2 \right|_{m_1 m_2 m_3} \quad (2.15)$$



Replacing the expression $m_2x_1 - m_1x_2$ in Equation (2.15) by $m_1(x_1 - x_2) + x_1$ we obtain:

$$X = \left\lfloor m_1m_2 \left[\frac{m_3}{2}(x_2 + x_3) + \frac{x_2 + x_3}{2} - x_1 \right] + m_1(x_1 - x_2) + x_1 \right\rfloor_{m_1m_2m_3} \quad (2.16)$$

Dividing both sides of Equation (2.16) by m_1 and taking the floor, we have:

$$\left\lfloor \frac{X}{m_1} \right\rfloor = \left\lfloor m_2 \left[\frac{m_3}{2}(x_2 + x_3) + \frac{x_2 + x_3}{2} - x_1 \right] + x_1 - x_2 + \frac{x_1}{m_1} \right\rfloor_{m_2m_3} \quad (2.17)$$

Further simplification gives:

$$\left\lfloor \frac{X}{m_1} \right\rfloor = \left\lfloor \left(\frac{1}{m_1} - m_2 + 1 \right)x_1 + \left(\frac{m_1m_2}{2} - 1 \right)x_2 + \frac{m_1m_2}{2}x_3 \right\rfloor_{m_2m_3} \quad (2.18)$$

Let $u_1 = \left(\frac{1}{m_1} - m_2 + 1 \right)x_1$, $u_2 = \left(\frac{m_1m_2}{2} - 1 \right)x_2$ and $u_3 = \frac{m_1m_2}{2}x_3$

$$\left\lfloor \frac{X}{m_1} \right\rfloor = \left\lfloor u_1 + u_2 + u_3 \right\rfloor_{m_2m_3} \quad (2.19)$$

Finally, we obtain:

$$X = m_1 \left\lfloor \frac{X}{m_1} \right\rfloor + x_1 \quad (2.20)$$

□

In order to reduce the hardware complexity, we use the following properties as in Gbolagade et al. (2010b) to simplify Equation (2.20):



Property 1: The multiplication of a residue number by 2^k in modulo $(2^p - 1)$ is computed by k bit circular left shifting

Property 2: A negative number in modulo $(2^p - 1)$ is calculated by subtracting the number in question from $(2^p - 1)$. In binary representation, the ones complement of the number gives the result.

Let the residues (x_1, x_2, x_3) have binary representation as follows:

$$x_1 = (\underbrace{x_{1,n-1}x_{1,n-2}\dots x_{1,1}x_{1,0}}_n) \quad (2.21)$$

$$x_2 = (\underbrace{x_{2,n}x_{2,n-1}\dots x_{2,1}x_{2,0}}_{n+1}) \quad (2.22)$$

$$x_3 = (\underbrace{x_{3,n-1}x_{3,n-2}\dots x_{3,1}x_{3,0}}_n) \quad (2.23)$$

Evaluating u_1

$u_1 = \left| \left(\frac{1}{2^n} - (2^n + 1) + 1 \right) x_1 \right|_{2^{2n-1}}$. We evaluate the four parts of u_1 separately using property 1 and property 2 where applicable:

$$u_{11} = \left| \frac{1}{2^n} x_1 \right|_{2^{2n-1}} = \overbrace{00\dots 00}^n \cdot \underbrace{x_{1,n-1}\dots x_{1,0}}_n \quad (2.24)$$

$$u_{12} = \left| -2^n x_1 \right|_{2^{2n-1}} = \overbrace{\bar{x}_{1,n-1}\bar{x}_{1,n-2}\dots\bar{x}_{1,1}\bar{x}_{1,0}}^n \underbrace{11\dots 11}_n \quad (2.25)$$

$$u_{13} = \left| x_1 \right|_{2^{2n-1}} = \overbrace{11\dots 11}^n \underbrace{\bar{x}_{1,n-1}\bar{x}_{1,n-2}\dots\bar{x}_{1,1}\bar{x}_{1,0}}_n \quad (2.26)$$



$$u_{14} = |x_1|_{2^{2n-1}} = \overbrace{00 \dots 00}^n \underbrace{x_{1,n-1} x_{1,n-2} \dots x_{1,1} x_{1,0}}_n \quad (2.27)$$

Therefore by adding Equations (2.24) through to (2.27), we have the value of u_1 . However, u_{11} could be ignored because it is approximately zero as the floor function makes it negligible. Considering u_{12} and u_{14} , the n leftmost bit of u_{14} are zeros, and the n rightmost bit of u_{12} are ones. By adding u_{12} and u_{14} we obtain u'_1 , where \vee denotes OR:

$$u'_1 = \overbrace{\bar{x}_{1,n-1} \bar{x}_{1,n-2} \dots \bar{x}_{1,1} \bar{x}_{1,0}}^n \underbrace{1 \vee x_{1,n-1} 1 \vee x_{1,n-2} \dots 1 \vee x_{1,0}}_n \quad (2.28)$$

Evaluating u_2

$$u_2 = \left| \left(\frac{2^{2n} + 2^n}{2} - 1 \right) x_2 \right|_{2^{2n-1}} = |2^{2n-1} x_2 + 2^{n-1} x_2 - x_2|_{2^{2n-1}}$$

We evaluate the three parts of u_2 separately using property 1 and property 2 where applicable as follows:

$$u_{21} = |2^{2n-1} x_2|_{2^{2n-1}} = x_{2,0} \overbrace{00 \dots 00}^{n-1} \underbrace{x_{2,n} x_{2,n-1} \dots x_{2,1}}_n \quad (2.29)$$

$$u_{22} = |2^{n-1} x_2|_{2^{2n-1}} = \overbrace{x_{2,n} x_{2,n-1} \dots x_{2,1} x_{2,0}}^{n+1} \underbrace{00 \dots 00}_{n-1} \quad (2.30)$$



$$u_{23} = -x_2 = \overbrace{11\dots11}^{n-1} \underbrace{\bar{x}_{2,n}\bar{x}_{2,n-1}\dots\bar{x}_{2,1}\bar{x}_{2,0}}_{n+1} \quad (2.31)$$

Again, by considering u_{21} and u_{22} , we manipulate them using the $n - 1$ zero bits in each to obtain u_2'

$$u_2' = x_{2,0} \vee x_{2,n} \underbrace{x_{2,n-1}\dots x_{2,1}}_{n-1} x_{2,0} \vee x_{2,n} \underbrace{x_{2,n-1}\dots x_{2,1} x_{2,0}}_{n-1} \quad (2.32)$$

Evaluating u_3

$$u_3 = \left\lfloor \left(\frac{2^{2n} + 2^n}{2} \right) x_3 \right\rfloor_{2^{2n-1}} = \left\lfloor (2^{2n-1} x_3 + 2^{n-1} x_3) \right\rfloor_{2^{2n-1}}$$

Again we evaluate the two parts of u_3 separately as follows:

$$u_{31} = \left\lfloor 2^{2n-1} x_3 \right\rfloor_{2^{2n-1}} = x_{3,0} \overbrace{00\dots00}^n \underbrace{x_{3,n-1}x_{3,n-2}\dots x_{3,1}}_{n-1} \quad (2.33)$$

$$u_{32} = \left\lfloor 2^{n-1} x_3 \right\rfloor_{2^{2n-1}} = 0 \overbrace{x_{3,n-1}\dots x_{3,1} x_{3,0}}^n \underbrace{00\dots00}_{n-1} \quad (2.34)$$

From the above, we manipulate u_{31} and u_{32} to obtain:

$$u_3' = \overbrace{x_{3,0}x_{3,n-1}\dots x_{3,1}x_{3,0}}^{n+1} \underbrace{x_{3,n-1}x_{3,n-2}\dots x_{3,1}}_{n-1} \quad (2.35)$$

Now it is convenient to express the sum of all the binary strings given by (27) - (37).

Let $\left\lfloor \frac{X}{m_1} \right\rfloor = \alpha$.



$$\alpha = \left| u'_1 + u_{13} + u'_3 + u'_2 + u_{23} \right|_{2^{2n}-1} \quad (2.36)$$

2.3 Hardware Realization

The hardware structure of the proposed reverse converter is based on Equations (2.20) and (2.36). Calculation of the operands of α requires $3n$ inverters. Since u_{13} and u_{23} both have n bits of "1's", two full adders FAs in the Carry Save Adder (CSA) with End around Carry (EAC) are reduced to two half adders (HAs). The five $2n$ bit numbers are summed up modulo $(2^{2n} - 1)$. This requires three levels of $2n$ CSAs with EAC, followed by a $2n$ bit Carry Propagate Adder (CPA) with a carry-in of 1. It is important to note that, n and $n - 1$ most significant bits of two operands that always have inputs equal to 1. This in anticipation will result in a final one's complement adder that generates an end-around carry with the one's complement adder reduced to a normal CPA with a constant carry-in of 1. Since x_1 is an n -bit number, the computation of Equation (2.20) requires no additional hardware as the desired result is obtained by concatenating x_1 with α . Figure (2.1) below depicts the structure of the proposed hardware.



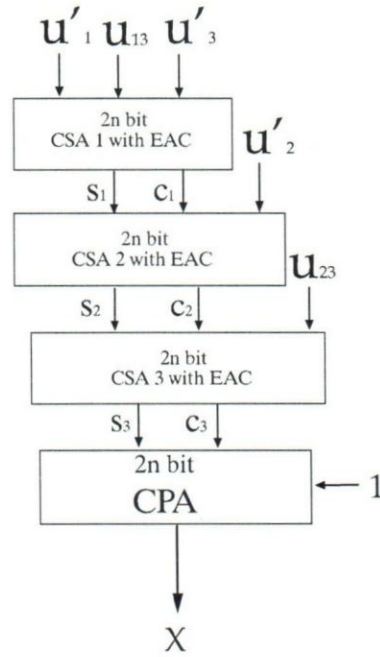


Figure 2.1: Block diagram for the Proposed Converter

2.4 Performance Analysis

In this section, we evaluate the performance of the proposed RNS to binary high speed converter with the best state of the art equivalent converter proposed in Wang et al. (2002) both theoretically and FPGA experimentation.

2.4.1 Theoretical Analysis

As outlined in the previous section, we sum up five $2n$ bit numbers u'_1 , u_{13} , u'_3 , u'_2 , and u_{23} modulo $(2^{2n} - 1)$. For this reason, three CSAs with EAC which include $8n$ FAs is sufficient. However, considering the operands, some of these FAs can be reduced

Table 2.1: Hardware Requirement of the proposed converter

Components	Full Adder	Half Adder
CSA1	n	n
CSA2	$2n - 2$	2
CSA3	$n + 1$	$n - 1$
CPA	$2n$	0
Total	$6n - 1$	$2n + 1$

to HAs. The CSA with EAC has an inverter level for all operands, so the inversions in u'_1, u_{13} and u_{23} are executed in both CSA1 with EAC and CSA2 with EAC. From Equations (2.25), (2.28), and (2.35), n of the FAs in CSA1 is reduced to n HAs. This means that CSA1 with EAC consist of n FA and n HAs. Also, since Equation (2.32) has 2 bits of XNOR/OR, $2n$ FAs are reduced to $(2n - 2)$ FAs in CSA2 with EAC. Then in CSA3 with EAC, we have $(n - 1)$ bit of 1's, reducing $2n$ FAs to $n + 1$ FAs. The $2n$ bit modulo adder has an approximate complexity of $2n$ FA and an approximate delay of $(2n)t_{FA}$, where t_{FA} is the delay of one FA. Table 2.1 presents the characteristics of each part of the proposed reverse converter in terms of FA and HA where the complexity of XNOR/OR are considered as HA for simplicity sake.

The hardware architecture of the proposed reverse converter consists of $(6n - 1)$ FAs and $(2n + 1)$ HAs. The delay of a CSA is the same as that of a FA, so the proposed converter has a total delay of $(3t_{FA} + t_{CPA(2n)} + t_{inv})$. Table 2.2 compares the performance of our proposal with the state of the art reverse converter in Wang et al. (2002). The data for converters presented by Ibrahim and Saloum (1988),



Table 2.2: Hardware Complexity and Delay Comparison

Converter	FA	AND/OR	XOR/XNOR	CLAs-2n	Delay
Ibrahim and Saloum (1988)	$6n$	-	$n + 1$	2	$2t_{CPA(n)} + 2t_{CPA(2n)} + 2t_{XOR}$
Andraos and Ahmad (1988)	$6n$	$4n - 2$	$2n$	1	$3t_{CPA(2n)} + 2t_{XOR} + \lceil \log(2n) \rceil t_{AND}$
Bhardwaj et al. (1998)	$6n$	$n + 3$	$n + 1$	1	$2t_{FA} + t_{inv} + t_{CPA(2n)} + t_{MUX}$
Piestrak (1995)	$4n$	$2n - 1$	$2n$	1	$2t_{FA} + t_{inv} + 2t_{CPA(2n)} + 3t_{MUX}$
Wang et al. (2002)	$2n$	-	1	1	$t_{inv} + t_{MUX} + t_{FA} + 2t_{CPA(2n)}$
proposed	$6n - 1$	-	$2n + 1$	-	$3t_{FA} + t_{CPA(2n)} + t_{inv}$

Andraos and Ahmad (1988), Bhardwaj et al. (1998), Piestrak (1995), Wang et al. (2002) given in Table 2.2 are obtained from Wang et al. (2002). In summary, it is evident from the table that the proposed converter is faster than the state of the art converters while requiring higher area cost.

2.4.2 FPGA Experimentation

To achieve accurate estimations for hardware resources and delay requirements for our proposal and the state of the art, we described them in VHDL, and implemented using FPGA. The target device is Xilinx Spartan 6 xc6slx45t-3fpg484, with Xilinx ISE 14.3. The delay is assessed in nanoseconds while the area is evaluated by the number of occupied slices. The converters were implemented with a wide range of values of n . Table 2.3 compares the area and delay of our scheme and that presented by Wang et al. (2002). As expected, the delay of our proposal is better. Contrary to the theoretical result, the experimental evaluation reveals that our scheme utilizes relatively lower hardware resources. The performance of our proposal against the state of the art in terms of area and delay, is depicted in Figures 2.2 and 2.3 respectively. From Table 2.3, it is indicative that, on the average, our proposed converter is 9.16%



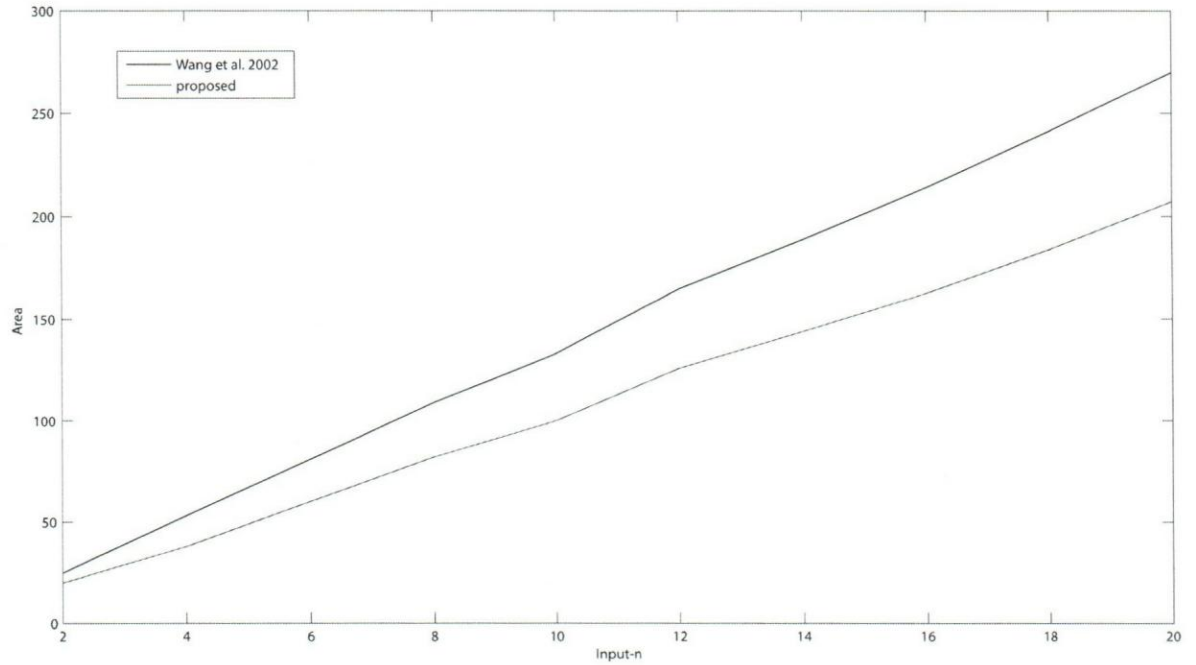


Figure 2.2: The Area Comparison of Converters

faster than the state of the art, while in terms of hardware resources, our scheme saves about 24.05% area.

2.5 Conclusion

In this chapter, a novel high speed and low cost reverse converter for the moduli set $\{2^n, 2^n + 1, 2^n - 1\}$ has been proposed based on the CRT. The resulting architecture was further reduced to obtain a R-B converter that utilizes only CSAs and CPA with a carry-in of 1. The proposed converter is memoryless and adder based and



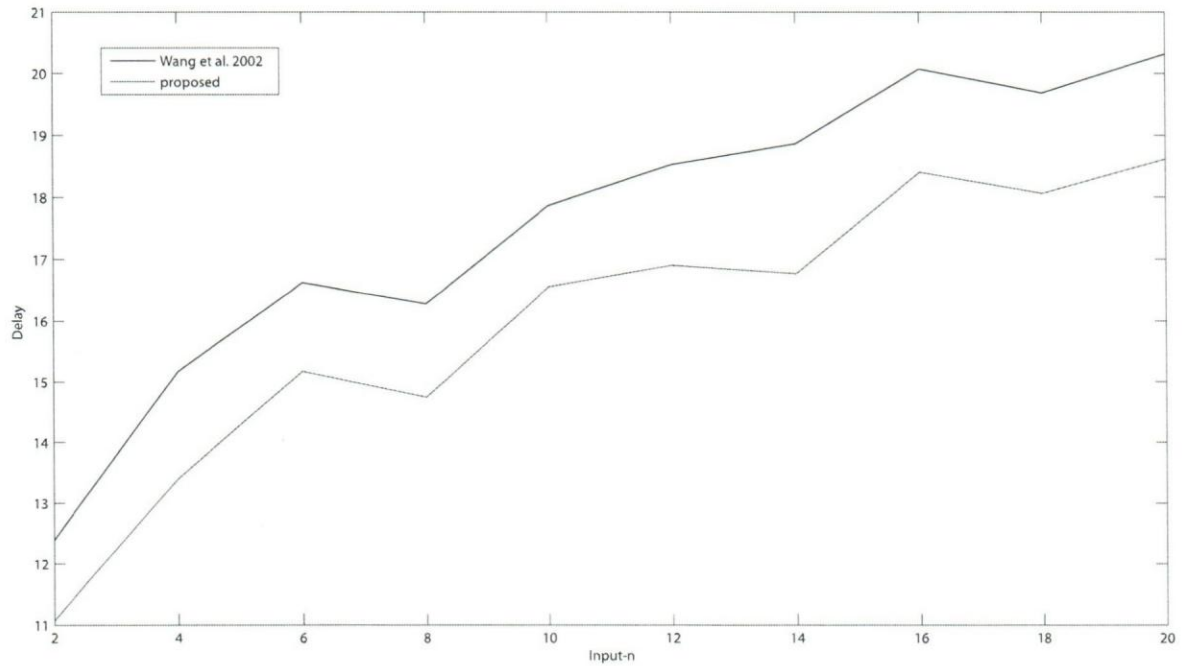


Figure 2.3: The Delay Comparison of Converters

is suitable for VLSI realization. Next, we performed both theoretical and FPGA experimentation to evaluate our scheme with existing state of the art. The theoretical assessment shows that our converter is advantageous in terms of speed at the expense of slightly higher area cost. Finally, we implemented our scheme and the state of the art on Xilinx Spartan 6 FPGA. The experimental results indicate that, contrary to the theoretical result and on the average, our proposed converter is 9.16% faster than the state of the art, while in terms of hardware resources, our scheme saves about 24.05% area.



Table 2.3: Experimental Delay and Area Comparison for $\{2^n, 2^n - 1, 2^n + 1\}$

n	Wang et al. (2002)		Proposed Converter	
	Delay	Area	Delay	Area
2	12.390	25	11.075	20
4	15.167	53	13.407	38
6	16.625	81	15.168	60
8	16.279	109	14.742	82
10	17.860	133	16.555	100
12	18.528	165	16.903	126
14	18.866	189	16.766	144
16	20.076	214	18.402	163
18	19.681	241	18.055	184
20	20.315	270	18.616	207

In order to enhance parallelism, in the next chapter, we hybridize the modified CRT technique presented in this chapter with the MRC to achieve an efficient reverse converter for the moduli set $\{2^n, 2^n + 1, 2^n - 1, 2^{2n+1} - 1\}$.





CHAPTER 3

A UNIFIED CRT AND MRC SCHEME FOR THE $\{2^N, 2^N + 1, 2^N - 1, 2^{2N+1} - 1\}$ MODULI SET

In recent years, due to the increasing demand for some applications that require larger dynamic range and increased parallelism, length 4 moduli sets such as $\{2^n - 1, 2^n, 2^n + 1, 2^{n+1} \pm 1\}$ (Mohan, 2007), $\{2^n - 1, 2^n, 2^n + 1, 2^{n-1} - 1\}$ (Cao et al., 2005), $\{2^n - 1, 2^n, 2^n + 1, 2^{2n} + 1\}$ (Cao et al., 2003), and $\{2^n - 1, 2^n, 2^n + 1, 2^{2n+1} - 1\}$ (Molahosseini et al., 2010) generally referred to in literature as length 4-moduli supersets of $\{2^n, 2^n + 1, 2^n - 1\}$ have been investigated with their respective reverse conversion algorithms proposed. Some of these conversion algorithms use the Chinese Remainder Theorem (CRT), a combination of CRT and Mixed Radix Conversion (MRC) algorithms (Molahosseini et al., 2009), a combination of New CRT and MRC (Molahosseini et al., 2009). Molahosseini et al. (2010) introduced the two new 4-moduli sets $\{2^n - 1, 2^n, 2^n + 1, 2^{2n+1} - 1\}$ and $\{2^n - 1, 2^n + 1, 2^{2n}, 2^{2n} + 1\}$. Their proposed reverse converter for $\{2^n - 1, 2^n, 2^n + 1, 2^{2n+1} - 1\}$ moduli set was obtained

from New CRT II with better performance and hardware requirement when compared with other equivalent $5n$ bit dynamic range state of the art reverse converters.

Given that the hardware complexity and speed of the resulting reverse converter for the 4-moduli set $\{2^n, 2^n + 1, 2^n - 1, 2^{2n+1} - 1\}$ proposed by Molahosseini et al. (2010) can be further improved, we propose in this Chapter, a novel hybridized reverse converter for the moduli set $\{2^n, 2^n + 1, 2^n - 1, 2^{2n+1} - 1\}$. First, the proposed algorithm is based on a hybridization of the modified CRT algorithm for the moduli set $\{2^n, 2^n + 1, 2^n - 1\}$ presented in Chapter 2 of this thesis and the MRC method, resulting in a two level design. In the first level, the equivalent weighted number of the residues (x_1, x_2, x_3) is obtained by using the modified CRT scheme for the popular moduli set $\{2^n, 2^n + 1, 2^n - 1\}$. Next, the resulting weighted number equivalent from the first level is hybridized with the fourth residue x_4 using MRC with respect to the composite moduli set $\{2^{3n} - 2^n, 2^{2n+1} - 1\}$.

The rest of the chapter is structured as follows. Section 3.1 provides a brief background information. In Section 3.2, the proposed algorithm is formulated. Section 3.3 describes the hardware implementation of the proposed algorithm and Section 3.4 evaluates the performance of the proposed scheme. Finally, the chapter is concluded in Section 3.5.



3.1 Background

Given a moduli set $\{m_i\}_{i=1,k}$, the residues (x_1, x_2, \dots, x_k) can be converted into the corresponding decimal number X in the following ways: First, by the use of the well known CRT, which is given as (Szabo and Tanaka, 1967):

$$X = \left| \sum_{i=1}^k M_i \left| M_i^{-1} \right|_{m_i} x_i \right|_M \quad (3.1)$$

where, $M = \prod_{i=1}^k m_i$, $M_i = \frac{M}{m_i}$ and M_i^{-1} is the multiplicative inverse of M_i with respect to m_i .

Second, the MRC, can also be used. Suppose we have a residue number representation (x_1, x_2, \dots, x_k) with respect to the moduli set $\{m_i\}_{i=1,k}$ and Mixed Radix Digits (MRDs), $\{a_i\}_{i=1,k}$, the decimal equivalent of the residues can be computed as follows (Szabo and Tanaka, 1967):

$$X = a_1 + a_2 m_1 + a_3 m_1 m_2 + \dots + a_n m_1 m_2 \dots m_{k-1}, \quad (3.2)$$



where, the MRDs $a_{i=1,k}$ are given as follows:

$$\begin{aligned}
 a_1 &= x_1 \\
 a_2 &= \left| (x_2 - a_1) \left| m_1^{-1} \right|_{m_2} \right|_{m_2} \\
 a_3 &= \left| \left((x_3 - a_1) \left| m_1^{-1} \right|_{m_3} - a_2 \right) \left| m_2^{-1} \right|_{m_3} \right|_{m_3} \\
 &\vdots \\
 &\vdots \\
 &\vdots \\
 a_k &= \left| \left(\left((x_k - a_1) \left| m_1^{-1} \right|_{m_k} - a_2 \right) \left| m_2^{-1} \right|_{m_k} \right. \right. \\
 &\quad \left. \left. - \dots - a_{k-1} \right) \left| m_{k-1}^{-1} \right|_{m_k} \right|_{m_k}
 \end{aligned} \tag{3.3}$$

3.2 Proposed Algorithm

Given $\{2^n, 2^n + 1, 2^n - 1, 2^{2n+1} - 1\}$ as the 4-moduli set with corresponding residues (x_1, x_2, x_3, x_4) , the proposed algorithm which consists of two levels is formulated using the following theorems:

Theorem 3.1. *Given the moduli set $\{m_1, m_2, m_3\}$ with $m_1 = 2^n$, $m_2 = 2^n + 1$, and $m_3 = 2^n - 1$, the decimal equivalent of the residue numbers (x_1, x_2, x_3) is computed as:*

$$A = m_1 \left\lfloor \frac{A}{m_1} \right\rfloor + x_1 \tag{3.4}$$



where, $\left\lfloor \frac{A}{m_1} \right\rfloor = |u_1 + u_2 + u_3|_{m_2 m_3}$, $u_1 = (\frac{1}{m_1} - m_2 + 1)x_1$, $u_2 = (\frac{m_1 m_2}{2} - 1)x_2$ and $u_3 = \frac{m_1 m_2}{2}x_3$

$\left\lfloor \frac{A}{m_1} \right\rfloor$ can be represented as:

$$\alpha = \left\lfloor \frac{A}{m_1} \right\rfloor = |u'_1 + u_{13} + u'_3 + u'_2 + u_{23}|_{2^{2n-1}}, \quad (3.5)$$

$$u'_1 = \overbrace{\bar{x}_{1,n-1} \dots \bar{x}_{1,1} \bar{x}_{1,0}}^n \underbrace{1 \vee x_{1,n-1} \dots 1 \vee x_{1,0}}_n \quad (3.6)$$

$$u_{13} = |x_1|_{2^{2n-1}} = \overbrace{11 \dots 11}^n \underbrace{\bar{x}_{1,n-1} \bar{x}_{1,n-2} \dots \bar{x}_{1,0}}_n \quad (3.7)$$

$$u'_3 = \overbrace{x_{3,0} x_{3,n-1} \dots x_{3,0}}^{n+1} \underbrace{x_{3,n-1} x_{3,n-2} \dots x_{3,1}}_{n-1} \quad (3.8)$$

$$u'_2 = x_{2,0} \vee x_{2,n} \underbrace{\bar{x}_{2,n-1} \dots \bar{x}_{2,1}}_{n-1} x_{2,0} \vee x_{2,n} \underbrace{x_{2,n-1} \dots x_{2,0}}_{n-1} \quad (3.9)$$

$$u_{23} = -x_2 = \overbrace{11 \dots 11}^{n-1} \underbrace{\bar{x}_{2,n} \bar{x}_{2,n-1} \dots \bar{x}_{2,1} \bar{x}_{2,0}}_{n+1} \quad (3.10)$$

Proof. : This theorem has been proved in Chapter 2, Section 2.2. □



Theorem 3.2. Given $\{m_1 = 2^n, m_2 = 2^n + 1, m_3 = 2^n - 1, m_4 = 2^{2n+1} - 1\}$ as the superset, Equation (3.11) holds true:

$$k = |(2^n(2^{2n} - 1))^{-1}|_{2^{2n+1}-1} = 2^{2n+1} - 2^{n+2} - 1 \quad (3.11)$$

Proof. : If it can demonstrated that $|(2^n(2^{2n} - 1)) \times (2^{2n+1} - 2^{n+2} - 1)|_{2^{2n+1}-1} = 1$, then $(2^{2n+1} - 2^{n+2} - 1)$ is the multiplicative inverse of $(2^n(2^{2n} - 1))$ with respect to $2^{2n+1} - 1$:

Letting $b = |(2^n(2^{2n} - 1)) \times (2^{2n+1} - 2^{n+2} - 1)|_{2^{2n+1}-1}$, we have:

$$\begin{aligned} b &= |(2^{5n+1} - (2^{2n+2})(2^{2n} - 1) - 3 \times 2^{3n} + 2^n)|_{2^{2n+1}-1} \\ &= \left| \frac{1}{2} \times 2^n + 1 - \frac{3}{2} \times 2^n + 2^n \right|_{2^{2n+1}-1} \\ &= |2^{n-1}(1 - 3) + 2^n + 1|_{2^{2n+1}-1} \\ &= 1. \end{aligned} \quad (3.12)$$

□

Theorem 3.3. Given $\{2^n, 2^n + 1, 2^n - 1, 2^{2n+1} - 1\}$ as the superset, the decimal equivalent X of the RNS number (x_1, x_2, x_3, x_4) can be computed as:

$$X = A + a_4(2^{3n} - 2^n) \quad (3.13)$$

where $a_4 = |(x_4 - A)k|_{2^{2n+1}-1}$, A and k are given by Equations (3.4) and (3.11) respectively.



Proof. : Given the moduli set $\{m_1 = 2^n, m_2 = 2^n + 1, m_3 = 2^n - 1, m_4 = 2^{2n+1} - 1\}$ with residues (x_1, x_2, x_3, x_4) , using the MRC, its decimal equivalent is computed using Equation (3.2) for $\{a_i\}_{i=1,4}$. Now, by considering the moduli set $\{2^n, 2^n + 1, 2^n - 1\}$ with residues (x_1, x_2, x_3) , its decimal equivalent is computed by using Equation (3.4). Therefore, $A = a_1 + a_2 m_1 + a_3 m_1 m_2$. Next, we consider the composite set $\{2^{3n} - 2^n, 2^{2n+1} - 1\}$ with residues (A, x_4) . For a two moduli set $\{m_A, m_4\}$, (3.2) becomes

$$X = A + a_4 m_1 m_2 m_3 = A + a_4 (2^{3n} - 2^n) \quad (3.14)$$

□

In order to reduce the hardware complexity, we use the following properties to simplify Equation (3.14): (Gbolagade et al., 2009).

Property 1: The multiplication of a residue number by 2^k in modulo $(2^p - 1)$ is computed by k bit circular left shifting

Property 2: A negative number in modulo $(2^p - 1)$ is calculated by subtracting the number in question from $(2^p - 1)$. In binary representation, the ones complement of the number gives the result.

Let the residues (x_1, x_2, x_3) have binary representation as follows:

$$x_1 = \underbrace{(x_{1,n-1} x_{1,n-2} \dots x_{1,1} x_{1,0})}_n \quad (3.15)$$



$$x_2 = \underbrace{(x_{2,n}x_{2,n-1}\dots x_{2,1}x_{2,0})}_{n+1} \quad (3.16)$$

$$x_3 = \underbrace{(x_{3,n-1}x_{3,n-2}\dots x_{3,1}x_{3,0})}_n \quad (3.17)$$

$$x_4 = \underbrace{(x_{4,2n}x_{4,2n-1}\dots x_{4,1}x_{4,0})}_{2n+1} \quad (3.18)$$

Note that:

$$a_4 = |(x_4 - A)(2^{2n+1} - 2^{n+2} - 1)|_{2^{2n+1}-1} \quad (3.19)$$

$$= |(x_4 - A)(-2^{n+2})|_{2^{2n+1}-1} \quad (3.20)$$

$$= |-2^{n+2}x_4 + 2^{n+2}A|_{2^{2n+1}-1} \quad (3.21)$$

$$= |a_{41} + a_{42}|_{2^{2n+1}-1} \quad (3.22)$$

The parameters in Equation (3.22) can then be simplified as follows:

$$\begin{aligned} a_{41} &= |-2^{n+2}x_4|_{2^{2n+1}-1} \\ &= \underbrace{\bar{x}_{4,n-2}\dots\bar{x}_{4,0}}_{n-1} \underbrace{\bar{x}_{4,2n}\dots\bar{x}_{4,n-1}}_{n+2} \end{aligned} \quad (3.23)$$

For a_{42} , it is interesting to note that A is a $3n$ bit number and is represented as

$\overbrace{A_{3n-1}A_{3n-2}\dots A_0}^{3n}$. Therefore,

$$a_{42} = |2^{n+2}(2^{2n+1}A_{3n-1}\dots A_{2n+1} + A_{2n}A_{2n-1}\dots A_0)|_{2^{2n+1}-1} = |a'_{42} + a''_{42}|_{2^{2n+1}-1}.$$

$$a'_{42} = \underbrace{A_{3n-1}\dots A_{2n+1}}_{n-1} \underbrace{00\dots 00}_{n+2} \quad (3.24)$$



$$a''_{42} = \overbrace{A_{n-2} \dots A_0}^{n-1} \underbrace{A_{2n} \dots A_{n-1}}_{n+2} \quad (3.25)$$

Now, let us rewrite Equation (3.22) as:

$$a_4 = |a_{41} + a'_{42} + a''_{42}|_{2^{2n+1}-1} \quad (3.26)$$

Thus, in order to obtain X using Equation (3.14), a_4 which is a $2n + 1$ bit number can be represented as $a_{4,2n}a_{4,2n-1} \dots a_{4,0}$. So

$$X = A + 2^{3n}a_4 - 2^n a_4, \quad (3.27)$$

where,

$$2^{3n}a_4 = \overbrace{a_{4,2n}a_{4,2n-1} \dots a_{4,0}}^{2n+1} \underbrace{00 \dots 00}_{3n} \quad (3.28)$$

and

$$2^n a_4 = \overbrace{00 \dots 00}^{2n} \underbrace{a_{4,2n}a_{4,2n-1} \dots a_{4,0}}_{2n+1} \overbrace{00 \dots 00}^n \quad (3.29)$$

$$-2^n a_4 = \overbrace{11 \dots 11}^{2n} \underbrace{\bar{a}_{4,2n}\bar{a}_{4,2n-1} \dots \bar{a}_{4,0}}_{2n+1} \overbrace{11 \dots 11}^n \quad (3.30)$$

3.3 Hardware Realization

The hardware implementation of the proposed reverse converter for the moduli set $\{2^n, 2^n + 1, 2^n - 1, 2^{2n+1} - 1\}$ is based on Equations (3.4), (3.5), (3.26), and (3.27).





The hardware architecture consists of two levels, the first level is based on a modified CRT and utilizes Equations (3.4) and (3.5). Implementation of Equation (3.5) requires a five operand modulo $2^{2n} - 1$ adder, where $2n$ bit numbers u'_1, u_{13}, u'_3, u'_2 , and u_{23} are added with three levels of $2n$ bit Carry Save Adder (CSA) with End Around Carry (EAC) followed by a $2n$ bit Carry Propagate Adder (CPA) with a carry in of 1. It must be noted that, two operands in this level have n and $n - 1$ most significant bit input equal to 1. This will result in the final one's complement adder always generating an end around carry. This phenomenon demonstrates that, the one's complement adder can be reduced to a normal CPA with a constant carry-in equal to 1. This therefore makes the delay $t_{CPA}(2n)$. Note also that, the computation of Equation (3.4) requires no additional hardware since the desired result is obtained by concatenating n bit number x_1 with α .

In the second level, Equation (3.26) which consists of three $(2n + 1)$ bit numbers a_{41}, a'_{42} , and a''_{42} are added with a $(2n + 1)$ bit CSA with EAC followed by a $(2n + 1)$ bit CPA. The addition of the operands in Equation (3.26) modulo $(2^{2n+1} - 1)$ can be accelerated with anticipated computation. Thus, we compute $s_4 + c_4$ for both $c_{in} = 0$ and $c_{in} = 1$ and the right result is selected with a multiplexer. This process is concluded with the computation of Equation (3.27) which requires $(5n + 1)$ bit subtractor implemented by regular CPA with a constant carry in of 1. Since A is a $3n$ bit number, it is concatenated with (3.28) and the final result is obtained with the $(5n + 1)$ bits CPA.

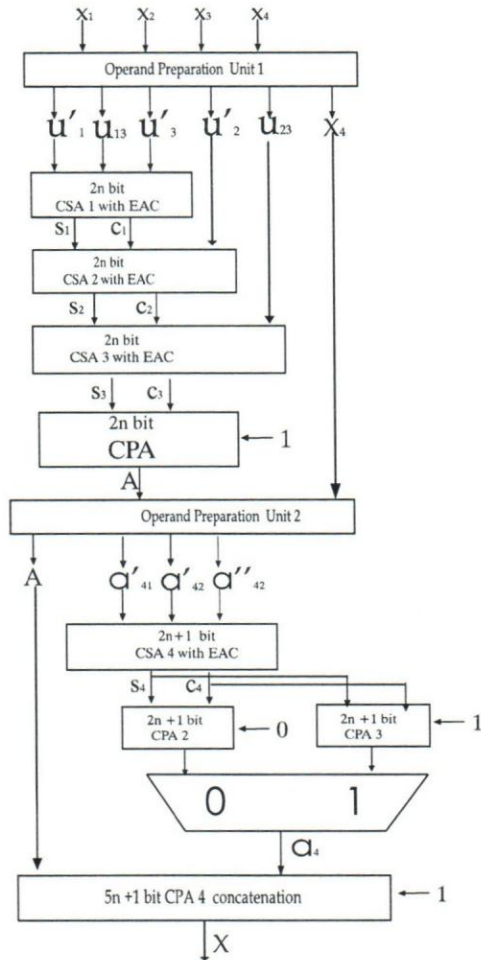


Figure 3.1: Proposed hardware Structure

3.4 Performance Analysis

In order to evaluate the performance of our scheme, we compared our proposal with the best known state of the art equivalent reverse converter. The comparison was carried out both theoretically and experimentally.

3.4.1 Theoretical Analysis

The performance of the proposed reverse converter is evaluated theoretically in terms of area cost and conversion delay. We compare our proposal with equivalent state of the art reverse converters presented by Molahosseini et al. (2010) and Molahosseini et al. (2009). The hardware utilization of our proposal is computed in terms of Full Adders (FAs) and Half Adders (HAs). It is clear from Table 3.1 that our proposal is faster than the existing state of the art but requires slightly more hardware resources in comparison with that presented in Molahosseini et al. (2010). In the case of the scheme presented by Molahosseini et al. (2009), our proposal is more efficient in terms of both delay and area.

The total delay of our reverse converter is the sum of all the delay of the two levels mentioned above. For the first level, the delay is $(2n + 3)t_{FA}$, while it is $(7n + 3)t_{FA} + t_{MUX}$ in the second level. Therefore, the total delay of the proposed converter is $(9n + 6)t_{FA} + t_{MUX}$ which is faster than that presented in Molahosseini et al. (2010). However, in terms of area, our scheme utilizes $(11n + 2)$ FA and $(5n + 1)$ HA, while



Table 3.1: Area and Delay Comparison for $\{2^n, 2^n + 1, 2^n - 1, 2^{2n+1} - 1\}$

Converter	Molahosseini et al. (2010)	Molahosseini et al. (2009)	Proposed
	$\{2^n, 2^n + 1, 2^n - 1, 2^{2n+1} - 1\}$	$\{2^n, 2^{n/2} - 1, 2^{n/2} + 1, 2^n + 1, 2^{2n-1} - 1\}$	$\{2^n, 2^n + 1, 2^n - 1, 2^{2n+1} - 1\}$
FA	$8n + 2$	$10n + 5$	$11n + 2$
HA	$5n$	$7n - 5$	$5n$
Area Cost in HA (Δ)	$21n + 4$	$27n + 5$	$27n + 5$
Delay (τ)	$(12n + 5)t_{FA}$	$(13n + 1)t_{FA}$	$(9n + 6)t_{FA} + t_{MUX}$

the state of the art utilizes $(8n + 2)$ FA and $(5n)$ HA. In Table 3.1, our proposed converter outperforms the state of the art equivalent reverse converter for the $5n$ bit dynamic range moduli set presented by Molahosseini et al. (2010) in terms of delay at the expense of slightly more hardware resources.

3.4.2 FPGA Experimentation

Additionally, we carried out an experimental assessment by implementing our proposed scheme and the one presented by Molahosseini et al. (2010) on Xilinx Spartan 6 xc6slx45t-3fgg484 FPGA device using Xilinx ISE 14.3. The synthesis results are given in terms of the number of slices and the conversion time in nano seconds. We used a wide range of values of n for the implementation. The FPGA implementation result is presented in Table 3.2. The synthesis results indicate that, on the average, our proposed converter is capable of performing the reverse conversion 9.80% faster, with extra hardware cost of 4.14% when compared with the state of the art presented by Molahosseini et al. (2010). The performance of our proposal against the state of the art converters in terms of area and delay, is depicted in Figures 3.2 and 3.3 respectively. In order to obtain an adequate comparison, the Area-Time



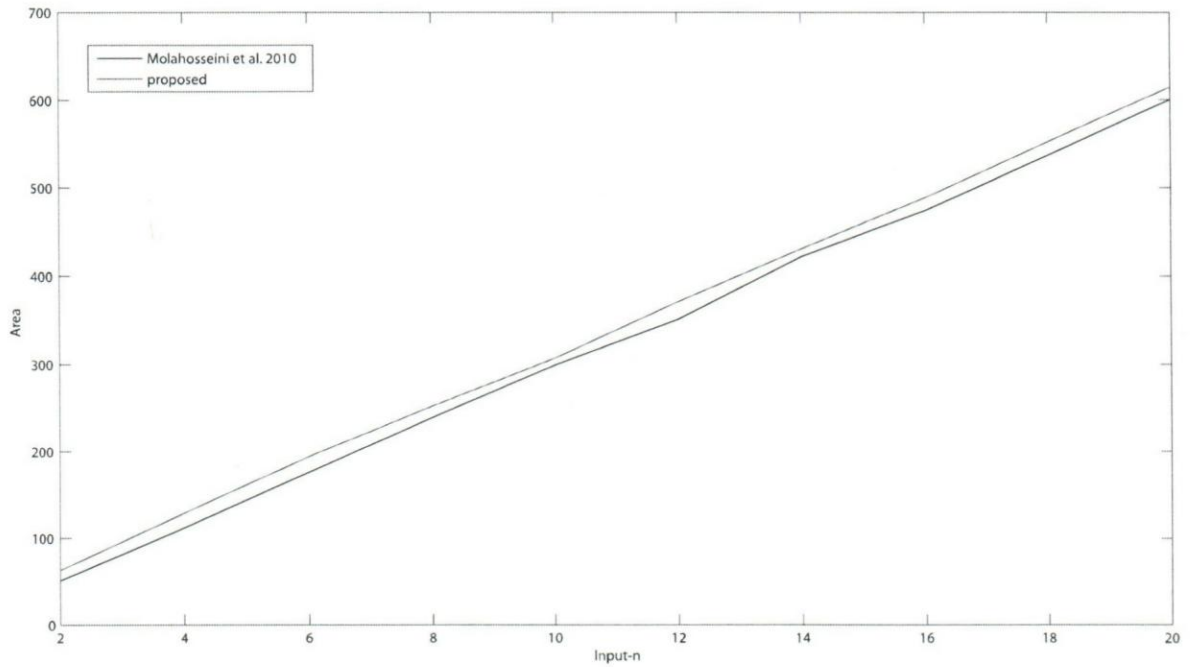


Figure 3.2: The Area Comparison of Converters

square efficiency metric was used, and its result illustrated in Table 3.3. The metric suggests that our proposed reverse converter is 19.34% efficient than the one proposed by Molahosseini et al. (2010). The Area-Time square metric comparison is depicted in Figure 3.4.

3.5 Conclusion

In this chapter, we proposed a new efficient reverse converter for the moduli set $\{2^n, 2^n + 1, 2^n - 1, 2^{2n+1} - 1\}$ based on a modified CRT and MRC. Additionally, we simplified further the resulting architecture in order to obtain a reverse converter that



Table 3.2: Experimental Delay and Area Comparison for $\{2^n, 2^n + 1, 2^n - 1, 2^{2n+1} - 1\}$

	Molahosseini et al. (2010)		Proposed Converter	
n	Delay	Area	Delay	Area
2	22.980	51	23.535	63
4	35.459	112	30.001	129
6	35.115	176	33.859	194
8	38.190	239	34.745	252
10	40.730	299	37.107	307
12	43.399	351	40.373	371
14	44.343	422	39.750	431
16	46.909	474	42.201	489
18	48.196	537	43.726	552
20	52.797	600	41.913	614

Table 3.3: Area-Time square ($\Delta\tau^2$) Comparison

	Molahosseini et al. (2010)	Proposed Converter
n	($\Delta\tau^2$)	($\Delta\tau^2$)
2	26932.1004	34895.4622
4	140822.1563	116107.7401
6	217019.1276	222407.7849
8	348575.7879	304218.1863
10	496020.9371	422717.3408
12	661099.0936	604722.2569
14	829779.2959	681006.9375
16	1043015.3290	870872.0321
18	1247372.8210	1055403.6180
20	1672513.9250	1078613.5350



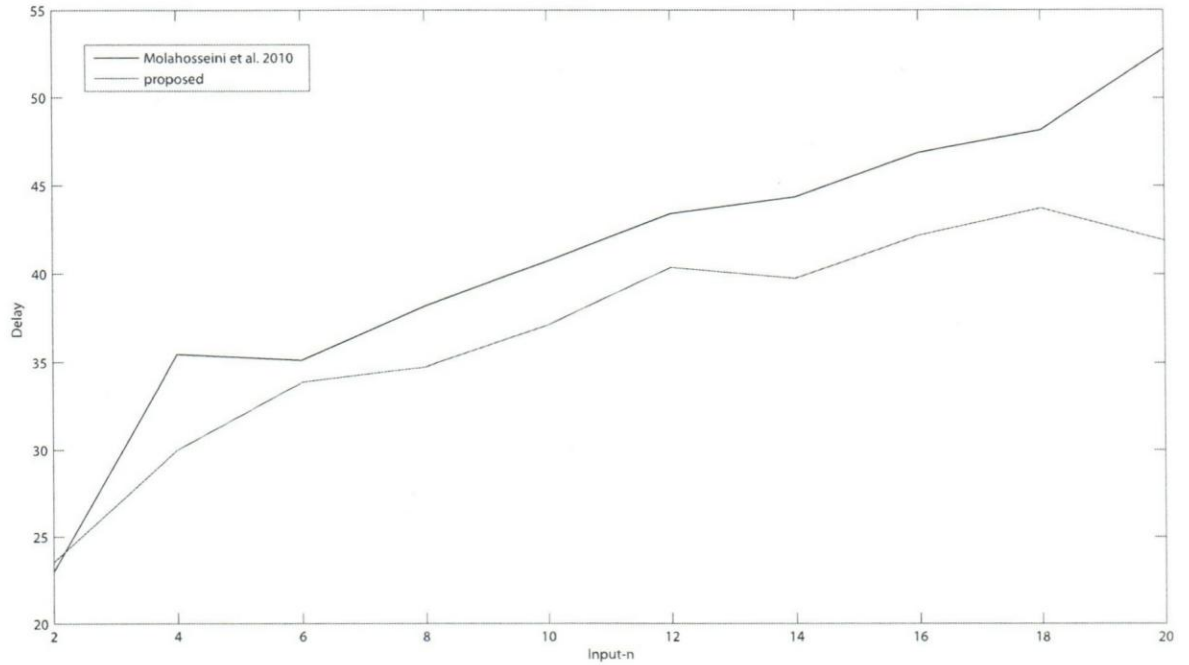


Figure 3.3: The Delay Comparison of Converters

utilizes only CSAs, a multiplexer and CPAs. The proposed converter is purely adder based and memoryless. Our proposal has a delay of $(9n + 6)t_{FA} + t_{MUX}$ with an area cost of $(27n + 5)HA$. The performance of the proposed reverse converter is evaluated both theoretically and experimentally by FPGA implementation. The theoretical analysis indicates that, the proposed reverse converter outperforms existing state of the art converters in terms of delay at the expense of slightly more hardware resources. For the experimental comparison, we described the proposed RNS to binary converter and the one presented by Molahosseini et al. (2010) in VHDL and carried out the implementation on an FPGA using a wide range of values of n . The synthesis results indicate that, on the average, our proposed converter is capable of performing the



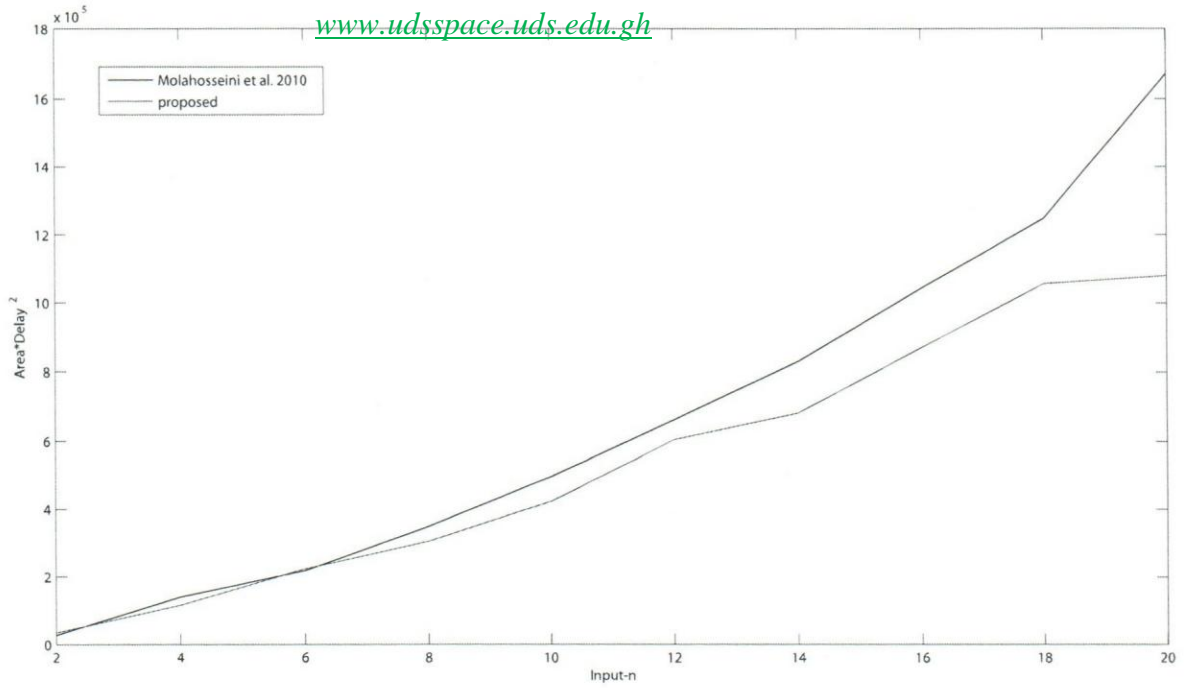


Figure 3.4: The Area Time square metric Comparison of Converters

reverse conversion 9.80% faster, with extra hardware cost of 4.14% when compared with the state of the art. In order to obtain an adequate comparison, the Area-Time square efficiency metric was used. The metric suggests that our proposed reverse converter is 19.34% efficient than the one proposed by Molahosseini et al. (2010).

In the next chapter, we derive a new length 3-moduli set from the moduli set $\{2^n, 2^n + 1, 2^n - 1, 2^{2n+1} - 1\}$ which has been considered in this chapter and then obtain its corresponding efficient residue to binary converter.



CHAPTER 4

A NEW $5N$ BIT DR MODULI SET

$$\{2^{2N} - 1, 2^N, 2^{2N+1} - 1\}$$

Given the moduli set $\{2^n - 1, 2^n, 2^n + 1, 2^{2n+1} - 1\}$ presented by Molahosseini et al. (2010), we combine the moduli $(2^n + 1)$ and $(2^n - 1)$ to form one modulus in order to obtain the 3-moduli set $\{2^{2n} - 1, 2^n, 2^{2n+1} - 1\}$. The advantages of this moduli set rest on the fact that it is free from $2^n + 1$ type modulus (Efstathiou et al., 1994). Further, based on the Mixed Radix Conversion (MRC), we present an efficient RNS to binary converter.

The rest of the chapter is organized as follows. Section 4.1 provides a brief background information on reverse conversion. In Section 4.2, we demonstrate the validity of the moduli set for RNS, and then formulate the corresponding reverse conversion algorithm. The hardware implementation of the proposed algorithm is described in Section 4.3, and Section 4.4 evaluates the performance of the proposed scheme. Finally, the chapter is concluded in Section 4.5.

4.1 Background

As mentioned previously, the main methods for reverse conversion are based on the Chinese Remainder Theorem (CRT), New CRT and Mixed Radix Conversion (MRC) techniques. In this chapter, we utilize the MRC. Given a moduli set $\{m_i\}_{i=1,3}$, the residues (x_1, x_2, x_3) can be converted into the corresponding decimal number X using the MRC as follows (Szabo and Tanaka, 1967):

$$X = a_1 + a_2 m_1 + a_3 m_1 m_2 \quad (4.1)$$

where a set of digits $\{a_1, a_2, a_3\}$, which are the Mixed Radix Digits (MRDs) are given as follows:

$$\begin{aligned} a_1 &= x_1 \\ a_2 &= \left| (x_2 - a_1) \left| m_1^{-1} \right|_{m_2} \right|_{m_2} \\ a_3 &= \left| \left((x_3 - a_1) \left| m_1^{-1} \right|_{m_3} - a_2 \right) \left| m_2^{-1} \right|_{m_3} \right|_{m_3} \end{aligned} \quad (4.2)$$



4.2 New $\{2^{2n} - 1, 2^n, 2^{2n+1} - 1\}$ Moduli Set and its Reverse Converter

We present a two-level residue-to-binary conversion algorithm for the moduli set $\{2^{2n} - 1, 2^n, 2^{2n+1} - 1\}$. The first level combines the first and the second residues with respect to the subset $\{2^n, 2^{2n} - 1\}$. The second level combines the third residue with the result of the first level with regard to the composite moduli set $\{2^n(2^{2n} - 1), 2^{2n+1} - 1\}$. First, we wish to show that the numbers $2^{2n} - 1$, 2^n , and $2^{2n+1} - 1$ are relatively prime.

Theorem 4.1. *The moduli set $\{2^{2n} - 1, 2^n, 2^{2n+1} - 1\}$ contains pairwise relatively prime moduli.*

Proof. :

From Euclidean theorem, we have:

$$\gcd(a, b) = \gcd(b, |a|_b),$$

Therefore,

$$\begin{aligned} \gcd(2^{2n} - 1, 2^n) &= \gcd(2^n, |2^{2n} - 1|_{2^n}) \\ &= \gcd(2^{2n}, -1) = 1 \end{aligned}$$

Also,

$$\begin{aligned} \gcd(2^{2n+1} - 1, 2^{2n} - 1) &= \gcd(2^{2n} - 1, |2^{2n+1} - 1|_{2^{2n} - 1}) \\ &= \gcd(2^{2n} - 1, 1) \\ &= 1 \end{aligned}$$



Again

$$\begin{aligned} \gcd(2^{2n+1} - 1, 2^n) &= \gcd(2^n, |2^{2n+1} - 1|_{2^n}) \\ &= \gcd(2^n, 1) \\ &= 1 \end{aligned}$$

Thus, the numbers $(2^{2n} - 1)$, (2^n) , and $2^{2n+1} - 1$ are relatively prime since all the greatest common divisors are equal to 1. \square

Theorem 4.2. *For the moduli set $\{2^{2n} - 1, 2^n, 2^{2n+1} - 1\}$, the following holds true:*

$$|(2^n)^{-1}|_{2^{2n}-1} = 2^n, \quad (4.3)$$

$$|(2^n(2^{2n} - 1))^{-1}|_{2^{2n+1}-1} = 2^{2n+1} - 2^{n+2} - 1, \quad (4.4)$$

Proof. :

If it can be demonstrated that $|(2^n) \times (2^n)|_{2^{2n}-1} = 1$, then 2^n is the multiplicative inverse of 2^n with respect to $2^{2n} - 1$. $|(2^n) \times (2^n)|_{2^{2n}-1}$ is given by:

$|2^{2n}|_{2^{2n}-1} = 1$, thus Equation (4.3) holds true.

In the same way if $|(2^n(2^{2n} - 1)) \times (2^{2n+1} - 2^{n+2} - 1)|_{2^{2n+1}-1} = 1$, then $(2^n(2^{2n} - 1))$ is the multiplicative inverse of $(2^{2n+1} - 2^{n+2} - 1)$ with respect to $(2^{2n+1} - 1)$.

$|(2^n(2^{2n} - 1)) \times (2^{2n+1} - 2^{n+2} - 1)|_{2^{2n+1}-1}$ is given by:

$|(2^n(2^{2n} - 1))(2^{2n+1} - 2^{n+2} - 1)|_{2^{2n+1}-1} = |1|_{2^{2n+1}-1} = 1$, thus Equation (4.4) holds true. \square



In order to reduce the hardware complexity, we use the following properties (Gbolagade et al., 2009):

Property 1: Modulo $(2^s - 1)$ multiplication of a residue number by 2^t , where s and t are positive integers, is equivalent to t bit circular left shifting.

Property 2: Modulo $(2^s - 1)$ of a negative number is equivalent to the one's complement of the number, which is obtained by subtracting the number from $(2^s - 1)$.

For the considered moduli set $\{2^n, 2^{2n} - 1, 2^{2n+1} - 1\}$ with the corresponding residues (x_1, x_2, x_3) , let the binary representations of the residues be:

$$x_1 = (x_{1,n-1}x_{1,n-2}\dots x_{1,1}x_{1,0}) \quad (4.5)$$

$$x_2 = (x_{2,2n-1}x_{2,2n-2}\dots x_{2,1}x_{2,0}) \quad (4.6)$$

$$x_3 = (x_{3,2n}x_{3,2n-1}\dots x_{3,1}x_{3,0}) \quad (4.7)$$

Now, let us consider the moduli set $\{2^n, 2^{2n} - 1\}$ and $Z_1 = (x_1, x_2)$. Using the MRC algorithm, given by Equation (4.1), for the two moduli set $\{2^n, 2^{2n} - 1\}$ and making use of Equation (4.3), we have:

$$\begin{aligned} z_1 &= a_1 + a_2 m_1 \\ &= x_1 + 2^n \left| (2^n)^{-1} \right|_{2^{2n}-1} (x_2 - x_1) \Big|_{2^{2n}-1} \\ &= x_1 + 2^n |2^n (x_2 - x_1)|_{2^{2n}-1} \end{aligned} \quad (4.8)$$

Let

$$z_2 = x_2 - x_1 \quad (4.9)$$



Note that x_2 is a $2n$ bit number while x_1 is an n bit number. In order to add x_2 and \bar{x}_1 , x_1 must be converted to a $2n$ bit number, which is given by:

$$\begin{aligned} x_1 &= \underbrace{(00 \cdots 0)}_n \underbrace{(x_{1,n-1} x_{1,n-2} \cdots x_{1,0})}_n \\ \bar{x}_1 &= \underbrace{(11 \cdots 1)}_n \underbrace{(\bar{x}_{1,n-1} \bar{x}_{1,n-2} \cdots \bar{x}_{1,0})}_n \end{aligned} \quad (4.10)$$

Using property 1, z_1 can be simplified. Suppose that:

$$z_1 = x_1 + 2^n z_3, \quad (4.11)$$

where,

$$\begin{aligned} z_3 &= \left\lfloor 2^n z_2 \right\rfloor_{2^{2n-1}} \\ &= \left\lfloor 2^n \underbrace{z_{2,2n-1} \cdots z_{2,n+1}}_n \underbrace{z_{2,n} z_{2,n-1} \cdots z_{2,1} z_{2,0}}_n \right\rfloor_{2^{2n-1}} \\ &= \underbrace{z_{2,n-1} \cdots z_{2,1} z_{2,0}}_n \underbrace{z_{2,2n-1} \cdots z_{2,n+1} z_{2,n}}_n \end{aligned} \quad (4.12)$$

Note that z_1 in Equation (4.11) can be achieved by a shift and concatenation operation. Thus, z_1 is a $3n$ bit number and can be represented as:

$$z_1 = \underbrace{z_{2,n-1} \cdots z_{2,0}}_n \underbrace{z_{2,2n-1} \cdots z_{2,n}}_n \underbrace{x_{1,n-1} \cdots x_{1,0}}_n. \quad (4.13)$$

Next, consider the composite moduli set $\{2^n(2^{2n} - 1), 2^{2n+1} - 1\}$ and let $X = (z_1, x_3)$. Using the MRC algorithm, given by Equation (4.1), for the two moduli



set $\{2^n(2^{2n} - 1), 2^{2n+1} - 1\}$, X can be computed as:

$$X = z_1 + 2^n(2^{2n} - 1)z_4, \quad (4.14)$$

where,

$$z_4 = |z_5(x_3 - z_1)|_{2^{2n+1}-1}, \quad (4.15)$$

From Equation (4.4), z_5 is given by:

$$\begin{aligned} z_5 &= |(2^n(2^{2n} - 1))^{-1}|_{2^{2n+1}-1} \\ &= 2^{2n+1} - 2^{n+2} - 1. \end{aligned} \quad (4.16)$$

Using Equation (4.16) and (4.11), Equation (4.15) can be written as:

$$\begin{aligned} z_4 &= |(2^{2n+1} - 2^{n+2} - 1)(x_3 - z_1)|_{2^{2n+1}-1} \\ &= |-2^{n+2}(x_3 - z_1)|_{2^{2n+1}-1} \\ &= |-2^{n+2}(x_3 - x_1 - 2^n z_3)|_{2^{2n+1}-1} \\ &= |-2^{n+2}x_3 + 2^{n+2}x_1 + 2(2^{2n+1})z_3|_{2^{2n+1}-1} \\ &= |-2^{n+2}x_3 + 2^{n+2}x_1 + 2z_3|_{2^{2n+1}-1}. \end{aligned} \quad (4.17)$$

Let

$$z_4 = |u_1 + u_2 + u_3|_{2^{2n+1}-1}, \quad (4.18)$$

where,

$$u_1 = -2^{n+2}x_3, u_2 = 2^{n+2}x_1, u_3 = 2z_3.$$



u_1 , u_2 , and u_3 can be simplified using properties 1 and 2 as follows:

$$\begin{aligned}
 u_1 &= \left| -2^{n+2} \underbrace{(x_{3,2n} \cdots x_{3,1} x_{3,0})}_{2n+1} \right|_{2^{2n+1}-1} \\
 &= \left| -2^{n+2} \underbrace{(x_{3,2n} \cdots x_{3,n-1})}_{n+2} \underbrace{(x_{3,n-2} \cdots x_{3,0})}_{n-1} \right|_{2^{2n+1}-1} \\
 &= \underbrace{\bar{x}_{3,n-2} \cdots \bar{x}_{3,0}}_{n-1} \underbrace{\bar{x}_{3,2n} \cdots \bar{x}_{3,n-1}}_{n+2}, \tag{4.19}
 \end{aligned}$$

$$\begin{aligned}
 u_2 &= \left| 2^{n+2} \underbrace{(x_{1,n-1} \cdots x_{1,1} x_{1,0})}_n \right|_{2^{2n+1}-1} \\
 &= \left| 2(2^{n+1}) \underbrace{(x_{1,n-1} \cdots x_{1,0})}_n \right|_{2^{2n+1}-1} \\
 &= \left| 2 \underbrace{(x_{1,n-1} \cdots x_{1,0})}_n \underbrace{(00 \cdots 0)}_{n+1} \right|_{2^{2n+1}-1} \\
 &= \underbrace{(x_{1,n-2} \cdots x_{1,0})}_{n-1} \underbrace{(00 \cdots 0)}_{n+1} \underbrace{x_{1,n-1}}_1, \tag{4.20}
 \end{aligned}$$

$$\begin{aligned}
 u_3 &= \left| 2 \underbrace{(z_{2,n-1} \cdots z_{2,0})}_n \underbrace{(z_{2,2n-1} \cdots z_{2,n})}_n \right|_{2^{2n+1}-1} \\
 &= \underbrace{(z_{2,n-1} \cdots z_{2,0})}_n \underbrace{(z_{2,2n-1} \cdots z_{2,n})}_n \underbrace{0}_1. \tag{4.21}
 \end{aligned}$$

Now, let Equation (4.14) be represented as

$$X = B + C, \tag{4.22}$$



where,

$$\begin{aligned}
 B &= z_1 + 2^{3n} z_4 \\
 &= \underbrace{z_{1,3n-1} \cdots z_{1,0}}_{3n} + \underbrace{z_{4,2n} \cdots z_{4,0}}_{2n+1} \underbrace{00 \cdots 0}_{3n} \\
 &= \underbrace{z_{4,2n} \cdots z_{4,0}}_{2n+1} \underbrace{z_{1,3n-1} \cdots z_{1,0}}_{3n}, \\
 C &= -2^n z_4
 \end{aligned} \tag{4.23}$$

$$\begin{aligned}
 &= -(\underbrace{00 \cdots 0}_{2n} \underbrace{z_{4,2n} \cdots z_{4,0}}_{2n+1} \underbrace{00 \cdots 0}_n) \\
 &= \underbrace{11 \cdots 1}_{2n} \underbrace{\bar{z}_{4,2n} \cdots \bar{z}_{4,0}}_{2n+1} \underbrace{11 \cdots 1}_n.
 \end{aligned} \tag{4.24}$$

4.3 Hardware Realization

The hardware implementation of the proposed reverse converter is based on Equations (4.9), (4.18), and (4.22). The proposed RNS-to-binary converter is depicted by Figure 4.1. In the figure, x_2 and \bar{x}_1 are added using a Carry Propagate Adder (CPA). Since there are n bits of '1's, n bits Half Adders (HAs) and n bits Full Adders (FAs) can be utilized in order to reduce the area cost. The variables u_1 , u_2 , and u_3 are added by a Carry Save Adder (CSA) with an End-Around Carry (EAC) producing the values s_1 and c_1 . These values must be added modulo $(2^{2n+1} - 1)$ to obtain z_4 , i.e., a CPA with EAC. The CSAs contain $(n + 2)$ bits of '0's. Thus, the area cost can also be reduced by employing $(n + 2)$ bits HAs and $(n - 2)$ bits FAs. The final result is obtained by adding B and C, given by Equation (4.22), using a CPA with



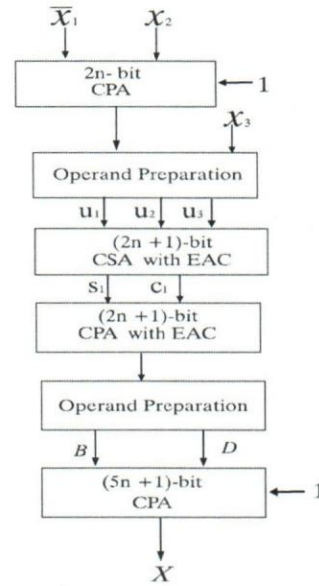


Figure 4.1: Block diagram for the Proposed Reverse Converter

"1" carry in. It should be noted that B is a $(5n+1)$ bit number and for C to be added to B , it must also be converted to a $(5n+1)$ bit number. In order to do this, 1's are appended to the result of complementations, as given in Equation (4.24). Since there are $3n$ bits of '1's, the area cost can be further reduced by using $3n$ bits HAs and $(2n+1)$ bits FAs. In Figure 4.1, there are three CPAs with the following delays: $2nt_{FA}$, $(4n+2)t_{FA}$, and $(5n+1)t_{FA}$ bits. This implies that the proposed design requires $(5n+2)$ HAs and $(6n+1)$ area cost with a conversion time of $(11n+4)$.



4.4 Performance Evaluation

In order to obtain realistic results, we performed both theoretical and experimental assessment of our scheme in comparison to other state of the art.

4.4.1 Theoretical Analysis

The performance of the proposed converter is evaluated in terms of area cost and conversion delay. We note that for reverse converters with different moduli sets to be fairly compared, the moduli sets should provide the same dynamic range and also rely on the same arithmetic unit speed (Molahosseini et al., 2008). The converters presented by Molahosseini et al. (2008), Gbolagade et al. (2010a), and our proposed converter are for moduli sets that rely on similar arithmetic unit speed. While the converters presented by Molahosseini et al. (2008) and Gbolagade et al. (2010a) are for $4n$ -bits dynamic range moduli sets, the proposed converter is for $5n$ -bits dynamic range moduli set. Thus, comparing the performance of the proposed converter with the ones in Molahosseini et al. (2008) and Gbolagade et al. (2010a) is not very appropriate. The performance of the converters presented by Molahosseini et al. (2008) and Gbolagade et al. (2010a) have been included in Table 4.1 to show that, our proposed converter with larger dynamic range, can compete favourably with the existing similar 3-moduli set converters.

Since the proposed converter and the one for the moduli set $\{2^n, 2^n-1, 2^n+1, 2^{2n+1}-1\}$ presented by Molahosseini et al. (2010) enjoy the same dynamic range, we compare



Table 4.1: Area-Delay Comparison

Converters	Molahosseini et al. (2008)	Gbolagade et al. (2010a)	Molahosseini et al. (2010)	Proposed Converter
Dynamic Range	$4n$ -bits	$4n$ -bits	$5n$ -bits	$5n$ -bits
FA	$9n + 2$	$9n + 2$	$8n + 2$	$7n + 2$
HA	$2n + 2$	$5n + 4$	$5n$	$4n + 1$
Delay	$(10n + 5)t_{FA}$	$(7n + 7)t_{FA}$	$(12n + 5)t_{FA}$	$(11n + 4)t_{FA}$

their performances in terms of area and delay. From Table 4.1, it can be seen that the proposed converter outperforms the one proposed by Molahosseini et al. (2010), from which it was derived, in terms of both area cost and conversion time.

It should be noted that apart from RNS to binary conversion speed and cost advantages, efficient internal RNS arithmetic circuits and efficient binary-to-RNS converter can easily be developed for the considered moduli set.

4.4.2 Experimental Analysis

We carried out an experimental assessment of our scheme and the state of the art converter proposed by Molahosseini et al. (2010) for the moduli set $\{2^n, 2^n - 1, 2^n + 1, 2^{2n+1} - 1\}$. We described both converters in VHDL and implemented them on Xilinx Spantan 6 xc6slx 45t-3fpg484 FPGA, with Xilinx ISE 14.3. The implementation was carried out for a wide range of values of n in order to obtain a relationship between the converters. The obtained results is presented in Table 4.2. The result serves as a confirmatory test of the theoretical analysis. On the average, our proposed converter is capable of performing 18.56% faster than the one presented for $\{2^n, 2^n - 1, 2^n + 1, 2^{2n+1} - 1\}$ by Molahosseini et al. (2010). Also, interms of area, our scheme achieves a 17.36% reduction in area resources. Clearly, our proposed converter outperforms



Table 4.2: Experimental Delay and Area Comparison for $\{2^{2n} - 1, 2^n, 2^{2n+1} - 1\}$

n	Molahosseini et al. (2010)		Proposed Converter	
	Delay	Area	Delay	Area
2	22.980	51	18.373	47
4	35.459	112	25.999	103
6	35.115	176	31.110	147
8	38.190	239	32.737	201
10	40.730	299	35.909	243
12	43.399	351	33.697	291
14	44.343	422	33.714	341
16	46.909	474	37.442	386
18	48.196	537	43.037	452
20	52.797	600	40.335	484

the state of the art. The performance of our proposal against the state of the art converters in terms of area and delay, is depicted in Figures 6.3 and 4.3 respectively.

4.5 Conclusion

This chapter proposed a new $5n$ bit DR moduli set $\{2^{2n} - 1, 2^n, 2^{2n+1} - 1\}$ derived from the four-moduli set $\{2^n, 2^n - 1, 2^n + 1, 2^{2n+1} - 1\}$ presented by Molahosseini et al. (2010) by combining $(2^n - 1)$ and $(2^n + 1)$ into $(2^{2n} - 1)$. Next, we propose an efficient MRC based RNS-to-binary converter for the new moduli set. When compared to the converter proposed by Molahosseini et al. (2010), theoretically speaking, our converter outperforms the state of the art in terms of both hardware requirements and delay. Also, we performed an FPGA experimentation of our converter and that of $\{2^n, 2^n - 1, 2^n + 1, 2^{2n+1} - 1\}$ by Molahosseini et al. (2010) by describing the converters in VHDL using Xilinx Spartan 6 on ISE 14.3. The experimental results



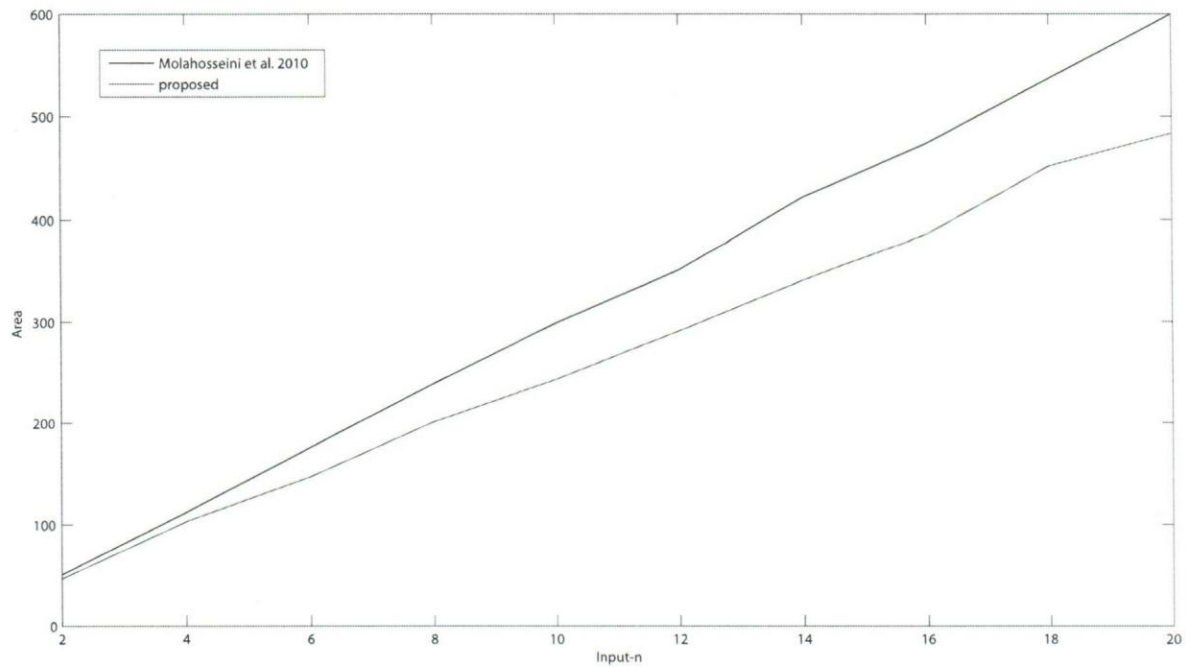


Figure 4.2: The Area Comparison of Converters

suggests that, averagely, our proposed architecture is 18.56% faster and saves about 17.36% hardware resources.

In the next chapter, we improve on the reverse converter for a recently proposed $5n$ bit DR moduli set.



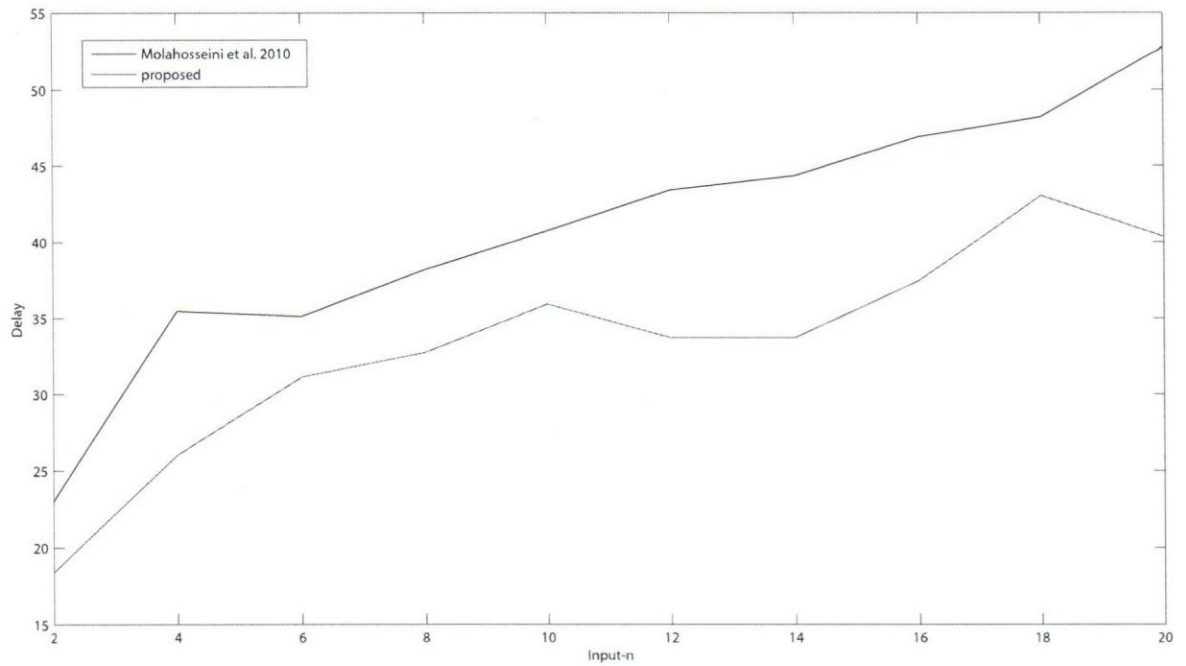


Figure 4.3: The Delay Comparison of Converters





CHAPTER 5

MRC ADDER BASED CONVERTER FOR $\{2^N, 2^{2N+1} - 1, 2^{2N+2} - 1\}$ MODULI SET

At length, several interesting moduli sets, such as $\{2^n, 2^n - 1, 2^n + 1\}$ (Bankas and Gbolagade, 2012), (Wang et al., 2002), $\{2^n, 2^n - 1, 2^{n-1} - 1\}$ (Hosseinzadeh et al., 2009), (Hiasat and Abdel-AtyZohdy, 1998), $\{2^{n+1} + 1, 2^{n+1} - 1, 2^n\}$ (Molahosseini and Navi, 2007), $\{2^n, 2^{n+1} - 1, 2^n - 1\}$ (Mohan, 2007) have been proposed in order to reduce RNS processor's hardware complexity. Based on these moduli sets, fast conversion algorithms, which give room for a widespread utilization in special purpose processors, have been proposed (Molahosseini et al., 2010). However, for applications requiring larger dynamic range (DR), moduli sets such as those proposed by Bankas and Gbolagade (2012), Wang et al. (2002), Hosseinzadeh et al. (2009), Hiasat and Abdel-AtyZohdy (1998) is insufficient for high speed performance computational systems. Hence, large dynamic range moduli sets such as $\{2^n - 1, 2^n, 2^n + 1, 2^{2n} + 1\}$ (Cao et al., 2003), $\{2^n, 2^{2n} - 1, 2^{2n} + 1\}$

(Hariri et al., 2008), $\{2^n - 1, 2^n + 1, 2^{2n} - 2, 2^{2n+1} - 3\}$ (Zhang and Siy, 2008), $\{2^{2n} - 1, 2^{2n+1}, 2^n - 1\}$ (Navi and Esmaeildoust, 2010), and many others have been introduced.

Due to the fact that arithmetic operations with respect to $2^n + 1$ modulus is not as simple as the end around carry based $2^n - 1$ modulus and this degrades the RNS architecture performance, the moduli set $\{2^{n+1} - 1, 2^n, 2^n - 1\}$ Mohan (2007) was proposed by removing the modulus $2^n + 1$ from the moduli set $\{2^{n+1} - 1, 2^n + 1, 2^n, 2^n - 1\}$. Subsequently, the moduli sets $\{2^{2n+1} - 1, 2^n, 2^n - 1\}$ and $\{2^{2n+1} - 1, 2^{2n}, 2^n - 1\}$, with their associated converters, were proposed by Molahosseini et al. (2008) and Gbolagade et al. (2009), respectively. Recently, the moduli set $\{2^{2n+2} - 1, 2^{2n+1} - 1, 2^n\}$ with $5n$ bits DR has been proposed with its accompanying reverse converter based on the Mixed Radix Conversion (MRC) method (Modiri et al., 2012). Given that the moduli set $\{2^{2n+2} - 1, 2^{2n+1} - 1, 2^n\}$ with its accompanying reverse converter is of practical interest, we present in this chapter an efficient MRC based reverse converter for the moduli set $\{2^{2n+2} - 1, 2^{2n+1} - 1, 2^n\}$, by parallelizing and optimizing the MRC algorithm.

The rest of the chapter is structured as follows: Section 5.1 provides a brief background on reverse conversion methods. In Section 5.2, the proposed reverse converter is presented. Section 5.3 describes the hardware implementation of the proposed converter. Also the performance of the proposed converter is evaluated in Section 5.4. The chapter is then concluded in Section 5.5.



5.1 Background

Reverse conversion is the process of translating from residue representation back to conventional notations such as decimal or binary. The main methods for reverse conversion are based on the Chinese Remainder Theorem (CRT), New CRT and MRC techniques. Given a moduli set $\{m_i\}_{i=1,k}$, the residues (x_1, x_2, \dots, x_k) can be converted into the corresponding decimal number X using MRC as follows (Szabo and Tanaka, 1967):

Given a 3-moduli set $\{m_1, m_2, m_3\}$, the number X can be obtained from its residue representation (x_1, x_2, x_3) as follows:

$$X = a_1 + a_2 m_1 + a_3 m_1 m_2 \quad (5.1)$$

where a set of digits $\{a_1, a_2, a_3\}$, which are the Mixed Radix Digits (MRDs) are given as follows:

$$\begin{aligned} a_1 &= x_1 \\ a_2 &= \left| (x_2 - a_1) \left| m_1^{-1} \right|_{m_2} \right|_{m_2} \\ a_3 &= \left| \left((x_3 - a_1) \left| m_1^{-1} \right|_{m_3} - a_2 \right) \left| m_2^{-1} \right|_{m_3} \right|_{m_3} \end{aligned} \quad (5.2)$$

Property 1: The multiplication of a residue number by 2^k in modulo $(2^p - 1)$ is computed by a k bit circular left shift.



Property 2: A negative number in modulo $(2^p - 1)$ is calculated by subtracting the number in question from $(2^p - 1)$. In binary representation, the one's complement of the number gives the result.

For the moduli set $\{2^{2n+2} - 1, 2^{2n+1} - 1, 2^n\}$ under consideration, the residues (x_1, x_2, x_3) have the binary representation as follows:

$$x_1 = (x_{1,2n+1}x_{1,2n}\dots x_{1,1}x_{1,0}), \quad (5.3)$$

$$x_2 = (x_{2,2n}x_{2,2n-1}\dots x_{2,1}x_{2,0}), \quad (5.4)$$

$$x_3 = (x_{3,n-1}x_{3,n-2}\dots x_{3,1}x_{3,0}). \quad (5.5)$$

For the moduli set $\{2^{2n+2} - 1, 2^{2n+1} - 1, 2^n\}$, based on the MRC represented by Equation (5.1), the following relation has been presented by Modiri et al. (2012):

$$X = x_1 + (2^{2n+2} - 1)(V_2 + (2^{2n+1} - 1)V_3) \quad (5.6)$$



where,

$$\begin{aligned}
 V_2 &= V_{21} + V_{22} \\
 V_{21} &= \underbrace{x_{2,2n}x_{2,2n-1}\dots x_{2,0}}_{2n+1} \\
 V_{22} &= \underbrace{\bar{x}_{1,2n}\dots\bar{x}_{1,0}}_{2n+1} + \underbrace{11\dots 11}_{2n} \bar{x}_{1,2n+1} \\
 V_3 &= V_{31} + V_{32} + V_{33} \\
 V_{31} &= \underbrace{x_{3,n-1}\dots x_{3,0}}_n \\
 V_{32} &= \underbrace{\bar{x}_{1,n-1}\dots\bar{x}_{1,0}}_n \\
 V_{33} &= \underbrace{V_{2,n-1}\dots V_{2,0}}_n.
 \end{aligned} \tag{5.7}$$

Given that the moduli set in question is desirable, we propose a more efficient reverse converter by optimizing the MRC.

5.2 Proposed Reverse Conversion Algorithm

Given the RNS number representation (x_1, x_2, x_3) for the moduli set

$\{2^n, 2^{2n+2} - 1, 2^{2n+1} - 1\}$, we propose an algorithm that computes its decimal equivalent based on MRC.

Theorem 5.1. *Given the $\{2^n, 2^{2n+2} - 1, 2^{2n+1} - 1\}$ moduli set, the following hold true:*



$$k_1 = |(2^n)^{-1}|_{2^{2n+2}-1} = 2^{n+2}, \quad (5.8)$$

$$k_2 = |((2^n)(2^{2n+2} - 1))^{-1}|_{(2^{2n+1}-1)} = 2^{n+1}. \quad (5.9)$$

Proof. :

If it can be demonstrated that $|(2^n) \times (2^{n+2})|_{2^{2n+2}-1} = 1$, then 2^{n+2} is the multiplicative inverse of 2^n with respect to $2^{2n+2} - 1$. $|(2^n) \times (2^{n+2})|_{2^{2n+2}-1} = |2^{2n+2}|_{2^{2n+2}-1} = 1$.

Similarly,

$$|(2^n)(2^{n+1})(2^{2n+2})|_{(2^{2n+1}-1)} = |(2^{2n+1})(2^{2n+2} - 1)|_{(2^{2n+1}-1)} = |(2^{2n+2} - 1)|_{(2^{2n+1}-1)} = 1.$$

Hence, Equations (5.8) and (5.9) holds true. \square

Theorem 5.2. *The decimal equivalent of the RNS number (x_1, x_2, x_3) with respect to the moduli set $\{m_1, m_2, m_3\}$ in the form $\{2^n, 2^{2n+2} - 1, 2^{2n+1} - 1\}$, can be computed as follows:*

$$X = r_1 + 2^{3n+2} \cdot r_2 - 2^n \cdot r_2, \quad (5.10)$$



where,

$$r_1 = x_1 + 2^n A, \quad (5.11)$$

$$A = A_1 + A_2,$$

$$\begin{aligned} A_1 &= \underbrace{x_{2,n-1} \dots x_{2,0}}_n \underbrace{x_{2,2n+1} \dots x_{2,n}}_{n+2} \\ A_2 &= \underbrace{\bar{x}_{1,n-1} \dots \bar{x}_{1,0}}_n \underbrace{11 \dots 11}_{n+2} \\ r_2 &= |r_2^i + r_2^{iii} + r_2^{iv}|_{2^{2n+1}-1} \\ r_2^i &= \underbrace{x_{3,n-1} \dots x_{3,0}}_n \underbrace{x_{3,2n} \dots x_{3,n}}_{n+1} \\ r_2^{iii} &= \underbrace{\bar{r}_{1,3n} \dots \bar{r}_{1,2n+1}}_n \underbrace{11 \dots 11 \bar{r}_{1,3n+1}}_{n+1}, \\ r_2^{iv} &= \underbrace{\bar{r}_{1,n-1} \dots \bar{r}_{1,0}}_n \underbrace{\bar{r}_{1,2n} \dots \bar{r}_{1,n}}_{n+1} \end{aligned} \quad (5.12)$$

Proof. :

Let us first consider the moduli set $\{2^n, 2^{2n+2} - 1\}$ and $r_1 = (x_1, x_2)$. Substituting Equation (5.8) into Equation (5.1) for a length 2 moduli set, we obtain:

$$r_1 = x_1 + 2^n |(2^{n+2})(x_2 - x_1)|_{2^{2n+2}-1}, \quad (5.13)$$

Rewriting Equation (5.13), we achieve:

$$r_1 = x_1 + 2^n \cdot A, \quad (5.14)$$



where,

$$\begin{aligned} A &= |2^{n+2}(x_2 - x_1)|_{2^{2n+2}-1} \\ &= |A_1 + A_2|_{2^{2n+2}-1} \end{aligned} \quad (5.15)$$

For the sake of completeness, consider the moduli set $\{2^n, 2^{2n+2} - 1, 2^{2n+1} - 1\}$ with the corresponding residues (x_1, x_2, x_3) having binary representations respectively as:

$$\begin{aligned} x_1 &= (x_{1,n-1}x_{1,n-2}\dots x_{1,1}x_{1,0}), \\ x_2 &= (x_{2,2n+1}x_{2,2n}\dots x_{2,1}x_{2,0}), \\ x_3 &= (x_{3,2n}x_{3,2n-1}\dots x_{3,1}x_{3,0}). \end{aligned} \quad (5.16)$$

Now, we simplify further Equation (5.15) by applying properties 1 and 2 to reduce the hardware complexity as:

$$\begin{aligned} A_1 &= |2^{n+2}(x_2)|_{2^{2n+2}-1} \\ &= \underbrace{x_{2,n-1}\dots x_{2,0}}_n \underbrace{x_{2,2n+1}\dots x_{2,n}}_{n+2} \end{aligned} \quad (5.17)$$



$$\begin{aligned} A_2 &= -|2^{n+2}(x_1)|_{2^{2n+2}-1} \\ &= \underbrace{\bar{x}_{1,n-1} \dots \bar{x}_{1,0}}_n \underbrace{11 \dots 11}_{n+2} \end{aligned} \quad (5.18)$$

Next, we consider the composite moduli set $\{2^n(2^{2n+2}-1), 2^{2n+1}-1\}$ and let $X = (r_1, x_3)$. By utilizing the MRC algorithm given by Equation (5.1) for a length 2 moduli set, we obtain:

$$\begin{aligned} X &= r_1 + (2^n)(2^{2n+2}-1) |k_2(x_3 - r_1)|_{2^{2n+1}-1}, \\ &= r_1 + (2^n)(2^{2n+2}-1) |2^{n+1}(x_3 - r_1)|_{2^{2n+1}-1} \end{aligned} \quad (5.19)$$

Note that, r_1 is as defined in Equation (5.11). Let

$$\begin{aligned} r_2 &= |2^{n+1}(x_3 - r_1)|_{2^{2n+1}-1} = |2^{n+1}x_3 - 2^{n+1}r_1|_{2^{2n+1}-1}, \\ &= |r_2^i + r_2^{ii}|_{2^{2n+1}-1} \end{aligned} \quad (5.20)$$

Simplifying further to reduce the hardware complexity by applying properties 1 and 2, we obtain:



$$r_2^i = |2^{n+1}x_3|_{2^{2n+1}-1} \quad (5.21)$$

$$= \underbrace{x_{3,n-1} \dots x_{3,0}}_n \underbrace{x_{3,2n} \dots x_{3,n}}_{n+1} \quad (5.22)$$

Also,

$$r_2^{ii} = |-2^{n+1}r_1|_{2^{2n+1}-1}, \quad (5.23)$$

$$= \left| -2^{n+1} \left(\underbrace{r_{1,3n+1} \dots r_{1,2n+1}}_{n+1} + \underbrace{r_{1,2n} \dots r_{1,0}}_{2n+1} \right) \right|_{2^{2n+1}-1} \quad (5.24)$$

$$= |r_2^{iii} + r_2^{iv}|_{2^{2n+1}-1} \quad (5.25)$$

By splitting Equation (5.25), we have:

$$\begin{aligned} r_2^{iii} &= \left| -2^{n+1} \left(\underbrace{00 \dots 00}_n \underbrace{r_{1,3n+1} r_{1,3n} \dots r_{1,2n+1}}_{n+1} \right) \right|_{2^{2n+1}-1} \\ &= \underbrace{\bar{r}_{1,3n} \dots \bar{r}_{1,2n+1}}_n \underbrace{11 \dots 11 \bar{r}_{1,3n+1}}_{n+1}, \end{aligned} \quad (5.26)$$

$$\begin{aligned} r_2^{iv} &= |-2^{n+1}(r_{1,2n} \dots r_{1,0})|_{2^{2n+1}-1} \\ &= \underbrace{\bar{r}_{1,n-1} \dots \bar{r}_{1,0}}_n \underbrace{\bar{r}_{1,2n} \dots \bar{r}_{1,n}}_{n+1} \end{aligned} \quad (5.27)$$



r_2 can be represented as:

$$r_2 = \left| r_2^i + r_2^{iii} + r_2^{iv} \right|_{2^{2n+1}-1}. \quad (5.28)$$

Hence Equation (5.10) holds true. \square

5.3 Hardware Realization

The hardware architecture of our proposed reverse converter for the moduli set $\{2^n, 2^{2n+1}-1, 2^{2n+2}-1\}$ is based on Theorem 5.2. The implementation of the main equation i.e., Equation (5.10) requires all the relations in Equations (5.11) through to (5.12). Figure 5.1 illustrates the block diagram of the proposed converter. The simplified versions of Equations (5.10) and (5.14), reduces the hardware complexities of the converter. First of all, the operands (5.17) and (5.18) are added modulo $(2^{2n+2}-1)$. It must be noted that the addition is done using Carry Propagate Adder (CPA) with End Around Carry (EAC).

Also, the three operands r_2^i , r_2^{iii} , and r_2^{iv} represented in Equations (5.22), (5.26), and (5.27) are added modulo $2^{2n+1}-1$ using a Carry Save Adder (CSA) followed by a $2^{2n+1}-1$ bit CPA with EAC. Some of the Full Adders (FAs) are reduced to Half Adder (HA), since some inputs of the CSA have constant values of one. It can be observed that Equation (5.18) has $(n+2)$ bits of 1's and Equation (5.26) also has (n) bits of 1's. This means that in principle, $(2n+2)$ of the FAs in the CPA with EAC



can be reduced to $(n + 2)$ HA's, while $(2n + 1)$ of FA in the CSA 1 can be reduced to (n) HA. This therefore indicates that, the proposed reverse converter requires $(9n + 5)$ FAs and $(2n + 2)$ HAs. To achieve a speed efficient converter, we utilize a multiplexer with inputs Equation (5.22), (5.26), and (5.27), where the select line is connected to the carry out of CPA 2. Finally, since r_1 is a $3n + 2$ bit number and r_2 a $2n + 1$ bit number, Equation (5.10) is achieved with a $5n + 3$ bit CPA with EAC. Hence the proposed reverse converter has $(11n + 9)t_{FA} + t_{MUX}$ delay.

5.4 Performance Comparison

In order to properly evaluate the performance of our proposed converter, both theoretical analysis and experimental implementation were performed in terms of conversion time and area cost.

5.4.1 Theoretical Evaluation

From the theoretical point of view, the hardware utilization of our proposal is derived in terms of FAs and HAs. We compare our proposal with equivalent best known state of the art reverse converters presented by Gbolagade et al. (2009) for the moduli set $\{2^{2n+1} - 1, 2^{2n}, 2^n - 1\}$ and Modiri et al. (2012) with same moduli set under consideration. For the classes of $5n$ bit DR state of moduli sets, the moduli set under investigation is also a $5n$ bit DR but of a larger DR compared



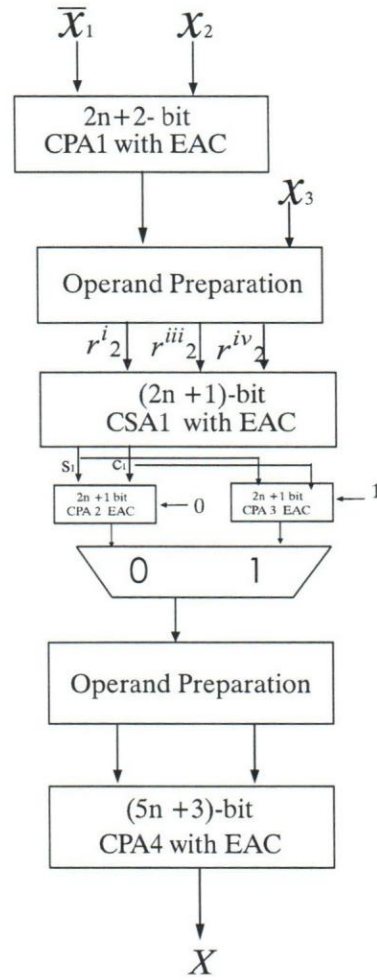


Figure 5.1: Block diagram for the Proposed Converter

to some others such as $\{2^n, 2^n + 1, 2^n - 1, 2^{2n+1} - 1\}$, $\{2^n - 1, 2^n, 2^n + 1, 2^{2n} + 1\}$, $\{2^{2n}, 2^{2n+1} - 1, 2^n - 1\}$, $\{2^n, 2^{2n} - 1, 2^{2n} + 1\}$. This therefore is an indication that our moduli set offers more advantage in terms of DR.

The proposed reverse converter requires a delay of $(11n + 9)t_{FA} + t_{MUX}$ while the best known reverse converter for the same moduli set in Modiri et al. (2012) has $(14n + 8)t_{FA}$ delay. In terms of area cost, our proposal utilizes $(9n + 5)$ FA and $(2n + 2)$ HA, while the state of the art requires $(8n + 3)$ FA and $(6n + 3)$ HA. To simplify the area comparison we assume that one FA has an area about twice as large than the HA and expressed the area cost for all the considered designs in terms of HAs.

5.4.2 Experimental Analysis

Additionally, we carried out experimental comparison by describing our proposed converter and the one presented by Modiri et al. (2012) in VHDL, and then implemented them on xilinx Spartan 6 xc6slx45t-3fgg484 FPGA, with Xilinx ISE 14.3. In order to properly evaluate the relationship between our converter and the state of the art under investigation, a wide range of values of n were implemented. The performance of the converters were evaluated in terms of area expressed by the number of occupied slice LUTs and the delay in nano seconds (ns). The results are illustrated in Table 5.2 for various values of n .

As expected, the experimental results confirmed the theoretical assessment. The results indicate that on the average, the proposed converter reduces the area by 35.98% when compared with that presented by Modiri et al. (2012). Also, in terms of delay, our proposal is about 6.76% speed up of conversion time.



Table 5.1: Theoretical Delay and Area Comparison

Converter	Gbolagade et al. (2009)	Modiri et al. (2012)	Proposed
DR	$5n$	$5n$	$5n$
FA	$15n + 2$	$8n + 3$	$9n + 5$
HA	$4n + 3$	$6n + 3$	$2n + 2$
Area Cost in HA (Δ)	$34n + 7$	$22n + 9$	$20n + 12$
Delay (τ)	$(13n + 7)t_{FA}$	$(14n + 8)t_{FA}$	$(11n + 9)t_{FA} + t_{MUX}$

Table 5.2: Experimental Delay and Area Comparison for $\{2^n, 2^{2n+1} - 1, 2^{2n+2} - 1\}$

	Gbolagade et al. (2009)		Modiri et al. (2012)		Proposed Converter	
n	Delay	Area	Delay	Area	Delay	Area
2	20.891	72	10.442	51	10.756	36
4	26.823	156	18.525	103	14.806	69
6	30.043	241	19.186	153	18.304	97
8	31.666	352	22.042	202	18.955	132
10	34.596	409	21.405	250	22.684	158
12	40.696	484	23.777	298	22.056	187
14	37.016	573	29.903	347	21.732	220
16	41.712	654	24.885	397	24.803	254
18	42.025	737	25.747	447	25.710	283
20	42.570	841	25.632	498	26.757	322



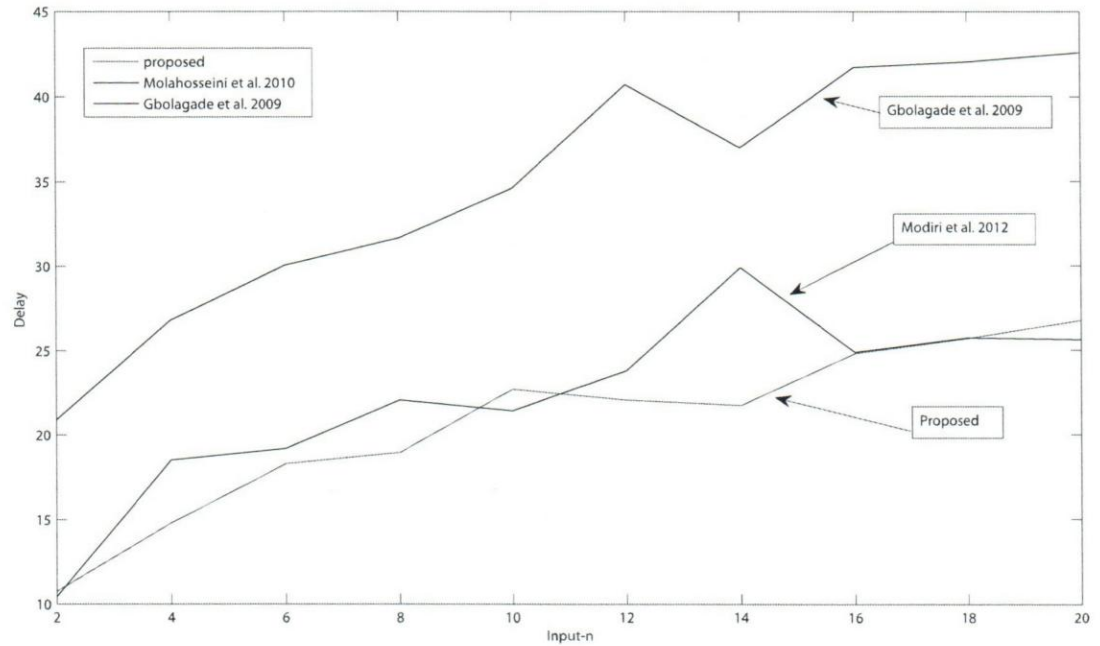


Figure 5.2: The Delay Comparison of Converters

5.5 Conclusion

In this chapter, we proposed an efficient RNS to binary converter for the moduli set $\{2^n, 2^{2n+1} - 1, 2^{2n+2} - 1\}$ which contains low-cost moduli and has a larger dynamic range compared to other existing $(5n)$ bit DR state of the art . The proposed reverse

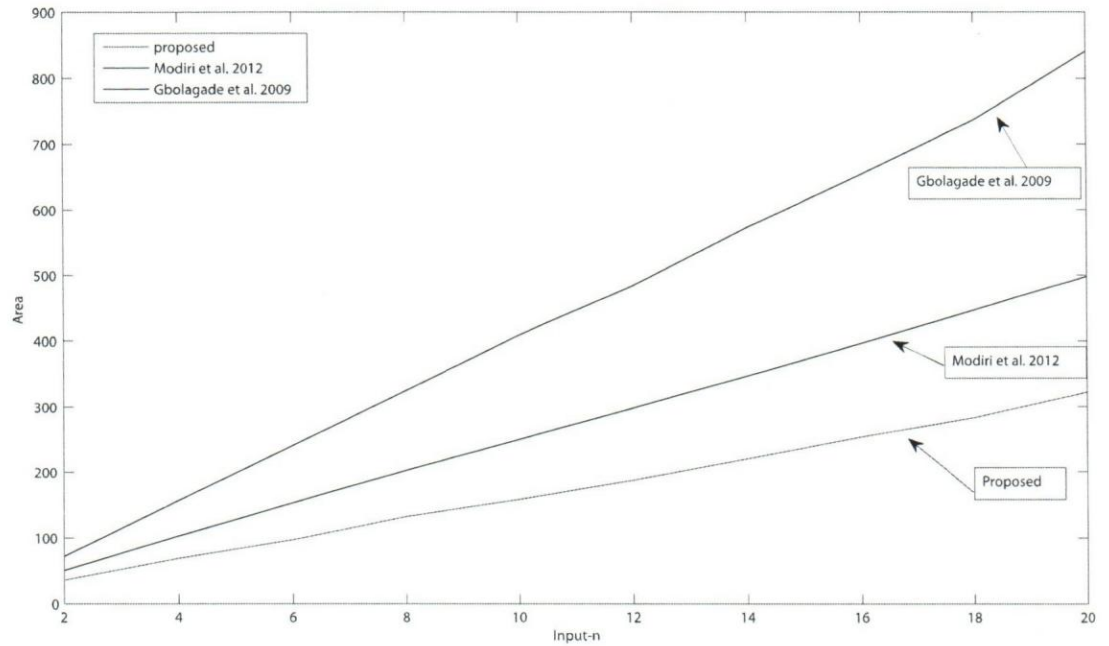


Figure 5.3: The Area Comparison of Converters

converter for the moduli set $\{2^n, 2^{2n+1} - 1, 2^{2n+2} - 1\}$ is based on MRC. Additionally, we further simplified the resulting architecture in order to obtain a reverse converter that utilizes only 1 level of CSA together with three CPAs i.e., CPA1, CPA2 or CPA3, and CPA4. The proposed converter is purely adder based and memoryless. The performance of the proposed converter is evaluated both theoretically and experimentally by FPGA implementation. Averagely, we found out that, our proposal



outperforms the state of the art with approximately 35.98% and 6.76% in terms of area cost and conversion time respectively.

Given that, the moduli set $\{2^{2n} - 1, 2^n, 2^{2n+1} - 1\}$ presented in Chapter 4 provides insufficient dynamic range for some applications, we present in the next chapter a moduli set which is an enhanced form of $\{2^{2n} - 1, 2^n, 2^{2n+1} - 1\}$ with its associated reverse converter.





CHAPTER 6

EFFICIENT CONVERTER FOR A NEW $\{2^{2N+1} - 1, 2^{2N}, 2^{2N} - 1\}$ MODULI SET

Moduli selection and their respective data conversion algorithms are the two most important issues for a successful RNS realization. Different moduli sets have been presented with their associated residue to binary conversion schemes. The moduli set $\{2^n, 2^n - 1, 2^n + 1\}$ has attracted a lot of attention, primarily because of its interesting number theoretic properties. This moduli set has a dynamic range of $(3n)$ bit and the disadvantage that, multiplication by powers of 2 with respect to the modulus $(2^n + 1)$ is not as simple as left circular shift in a $(2^n - 1)$ modulus (Gbolagade et al., 2009). For this reason, moduli sets that are free of $(2^n + 1)$ modulus, e.g., $\{2^k, 2^k - 1, 2^{k-1} - 1\}$ have been proposed with their associated converters in Hiasat and Abdel-AtyZohdy (1998), Hosseinzadeh et al. (2009). Also, $\{2^{n+1} - 1, 2^n, 2^n - 1\}$ was also presented by Mohan (2007). The need for moduli sets with larger Dynamic Range (DR) than $\{2^k, 2^k - 1, 2^{k-1} - 1\}$ and $\{2^{n+1}, 2^n, 2^n - 1\}$ led to the consideration of moduli sets

such as $\{2^{2n+1} - 1, 2^n, 2^n - 1\}$ (Molahosseini et al., 2008), and $\{2^{2n+1} - 1, 2^{2n}, 2^n\}$ (Gbolagade et al., 2009) with their associated reverse conversion algorithms.

Further, the continues demand for exploring the parallelism property of RNS together with achieving larger DR enticed lots of researchers to investigate length 4, 5, e.t.c. moduli sets. For example, the following moduli sets with their respective reverse conversion algorithms have been suggested in literature $\{2^n - 1, 2^n, 2^n + 1, 2^{n+1} - 1\}$ (Vinod and Premkumar, 2000), (Cao et al., 2005), (Mohan, 2007), $\{2^n - 1, 2^n, 2^n + 1, 2^{n+1} + 1\}$ (Mohan, 2007), $\{2^n - 1, 2^n, 2^n + 1, 2^{n-1} - 1\}$ (Cao et al., 2005), $\{2^n - 3, 2^n + 1, 2^n - 1, 2^n + 3\}$ (Sheu et al., 2004).

The need for $(2^n + 1)$ free moduli sets led to the suggestion of $\{2^{n+1} - 1, 2^n, 2^n - 1\}$ in Mohan (2007) by removing the modulus $2^n + 1$ from the moduli set

$\{2^{n+1} - 1, 2^n + 1, 2^n, 2^n - 1\}$. Subsequently, the moduli sets $\{2^{2n+1} - 1, 2^n, 2^n - 1\}$ and $\{2^{2n+1} - 1, 2^{2n}, 2^n - 1\}$, with their associated converters were proposed by Molahosseini et al. (2008) and Gbolagade et al. (2009) respectively. Infact, $\{2^{2n+1} - 1, 2^{2n}, 2^n - 1\}$ was proposed as an enhancement of $\{2^{2n+1} - 1, 2^n, 2^n - 1\}$ to achieve a larger DR. In the same way, $\{2^{2n+1} - 1, 2^n, 2^n - 1\}$ was also suggested as an improvement over $\{2^{n+1} - 1, 2^n, 2^n - 1\}$. Given that the moduli set $\{2^{2n+1} - 1, 2^{2n}, 2^n - 1\}$ can be further enhanced to obtain a larger DR with its accompanying reverse converter, we present in this chapter an efficient reverse converter for a new moduli set $\{2^{2n+1} - 1, 2^{2n}, 2^{2n} - 1\}$, by parallelizing and optimizing the MRC algorithm.



The rest of the chapter is structured as follows. Section 6.1 provides a brief background on reverse conversion methods. In Section 6.2, the proposed new moduli set with its associated reverse converter is presented. Section 6.3 describes the hardware implementation of the proposed converter with a performance analysis which compares our proposal with state of the art converters . The chapter is then concluded in Section 6.4.

6.1 Background

As mentioned earlier, the main methods for reverse conversion are based on the Chinese Remainder Theorem (CRT), New CRT and Mixed Radix Conversion (MRC) techniques. In this Chapter, we utilizes the MRC. Given a moduli set $\{m_i\}_{i=1,3}$, the residues (x_1, x_2, x_3) can be converted into the corresponding decimal number X using the MRC as follows (Szabo and Tanaka, 1967):

$$X = a_1 + a_2m_1 + a_3m_1m_2 \quad (6.1)$$



where a set of digits $\{a_1, a_2, a_3\}$, which are the Mixed Radix Digits (MRDs) are given as follows:

$$\begin{aligned} a_1 &= x_1 \\ a_2 &= \left| (x_2 - a_1) |m_1^{-1}|_{m_2} \right|_{m_2} \\ a_3 &= \left| \left((x_3 - a_1) |m_1^{-1}|_{m_3} - a_2 \right) |m_2^{-1}|_{m_3} \right|_{m_3} \end{aligned} \tag{6.2}$$

6.2 Proposed New Moduli Set and Reverse Conversion Algorithm

For a given RNS moduli set to be valid, it is required that all the elements in the moduli set are relatively prime. Thus, in order to prove that the proposed set can be utilized for the construction of valid RNS architecture, we have to demonstrate that the moduli $2^{2n+1} - 1$, 2^{2n} , and $2^{2n} - 1$ are pair-wise relatively prime.

Theorem 6.1. *The moduli $2^{2n+1} - 1$, 2^{2n} , and $2^{2n} - 1$ are pair-wise relatively prime numbers.*

Proof. :

It has already been shown that the pairs $(2^{2n+1} - 1, 2^{2n})$ and $(2^{2n+1} - 1, 2^{2n} - 1)$ are co-prime by Gbolagade et al. (2009) and Bankas et al. (2013) respectively. Hence, we need to demonstrate that $(2^{2n}, 2^{2n} - 1)$ are relatively prime. From the



Euclidean theorem, we have $\gcd(a, b) = \gcd(b, |a|_b)$, therefore, $\gcd(2^{2n}, 2^{2n} - 1) = \gcd(2^{2n} - 1, |2^{2n}|_{2^{2n}-1}) = 1$. Hence, it can be concluded that the moduli set $\{2^{2n+1} - 1, 2^{2n}, 2^{2n} - 1\}$ contains relatively prime moduli and it is a valid RNS moduli set. \square

Theorem 6.2. *Given that $\{2^{2n+1} - 1, 2^{2n}, 2^{2n} - 1\}$ is a valid RNS moduli set, the following relation holds true:*

$$|(2^{2n})^{-1}|_{2^{2n+1}-1} = 2, \quad (6.3)$$

$$|((2^{2n})(2^{2n+1} - 1))^{-1}|_{2^{2n}-1} = 1, \quad (6.4)$$

Proof. :

If it can be demonstrated that $|(2^{2n}) \times 2|_{2^{2n+1}-1} = 1$, then 2 is the multiplicative inverse of 2^{2n} with respect to $2^{2n+1} - 1$.

Thus, $|(2^{2n}) \times 2|_{2^{2n+1}-1} = |(2^{2n+1})|_{2^{2n+1}-1} = 1$. In similar manner,

$$|(2^{2n}) \times (2^{2n+1} - 1) \times 1|_{2^{2n}-1} = 1 \quad \square$$

Theorem 6.3. *Considering $\{2^{2n}, 2^{2n+1} - 1\}$ as a moduli set, the decimal equivalent of the RNS number (x_1, x_2) with respect to the moduli set $\{2^{2n}, 2^{2n+1} - 1\}$ can be obtained as follows:*

$$r_1 = x_1 + 2^{2n} |(2x_2 - 2x_1)|_{2^{2n+1}-1} \quad (6.5)$$



Proof. :

Sustituting Equations (6.3) into (6.1) for a length 2 moduli set, we obtain:

$$r_1 = x_1 + 2^{2n} \left| (x_2 - x_1) \right|_{(2^{2n}-1)} \left|_{2^{2n+1}-1} \right|_{2^{2n+1}-1} \quad (6.6)$$

$$r_1 = x_1 + 2^{2n} \left| (2x_2 - 2x_1) \right|_{2^{2n+1}-1} \quad (6.7)$$

□

Theorem 6.4. For $\{(2^{2n})(2^{2n+1} - 1), 2^{2n} - 1\}$ moduli set containing a composite modulus $(2^{2n})(2^{2n+1}-1)$, the decimal equivalent of the RNS number (r_1, x_3) with respect to the moduli set $\{(2^{2n})(2^{2n+1} - 1), 2^{2n} - 1\}$ can be computed as follows:

$$X = r_1 + (2^{2n})(2^{2n+1} - 1) \left| (x_3 - r_1) \right|_{2^{2n}-1} \quad (6.8)$$

Proof. Again we substitute Equations (6.4) and (6.7) into (6.1) for a length 2 moduli set to obtain:

$$X = r_1 + (2^{2n})(2^{2n+1} - 1) \left| (x_3 - r_1) \times 1 \right|_{2^{2n}-1} \quad (6.9)$$

$$X = r_1 + (2^{2n})(2^{2n+1} - 1) \left| (x_3 - r_1) \right|_{2^{2n}-1} \quad (6.10)$$



□

The hardware complexity required for the implementation of Equation (6.10) can be further reduced by using the following properties from Bankas et al. (2013):

Property 1: The multiplication of a residue number by 2^k in modulo $(2^p - 1)$ is computed by a k bit circular left shift.

Property 2: A negative number in modulo $(2^p - 1)$ is calculated by subtracting the number in question from $(2^p - 1)$. In binary representation, the one's complement of the number gives the result.

Let the residues (x_1, x_2, x_3) with respect to the moduli set $\{2^{2n}, 2^{2n+1} - 1, 2^{2n} - 1\}$ have binary representations as follows:

$$x_1 = (x_{1,2n-1}x_{1,2n-2}\dots x_{1,1}x_{1,0}), \quad (6.11)$$

$$x_2 = (x_{2,2n}x_{2,2n-1}\dots x_{2,1}x_{2,0}), \quad (6.12)$$

$$x_3 = (x_{3,2n-1}x_{3,2n-2}\dots x_{3,1}x_{3,0}). \quad (6.13)$$

From Equation (6.10), r_1 which is given by (6.5) can be directly rewritten as:

$$r_1 = x_1 + (2^{2n})v \quad (6.14)$$

$$v = |(2x_2 - 2x_1)|_{2^{2n+1}-1} = |u_1 + u_2|_{2^{2n+1}-1} \quad (6.15)$$



In Equation (6.15), u_1 and u_2 are represented as follows:

$$u_1 = |2x_2|_{2^{2n+1}-1} = \underbrace{x_{2,2n-1} \dots x_{2,0} x_{2,2n}}_{2n+1} \quad (6.16)$$

$$u_2 = |-2x_1|_{2^{2n+1}-1} = \underbrace{\bar{x}_{1,2n-1} \dots \bar{x}_{1,0} 1}_{2n+1} \quad (6.17)$$

Now, by rewriting Equation (6.10) directly, we obtain:

$$X = r_1 + (2^{2n})(2^{2n+1} - 1)w \quad (6.18)$$

$$w = |(x_3 - r_1)|_{2^{2n}-1} = |u_3 + u_4|_{2^{2n}-1} \quad (6.19)$$

Again in Equation (6.19), u_3 and u_4 can be represented as:

$$u_3 = |x_3|_{2^{2n}-1} = \underbrace{x_{3,2n-1} \dots x_{3,0}}_{2n} \quad (6.20)$$

$$u_4 = |-r_1|_{2^{2n}-1} = y_1 + y_2 + y_3 \quad (6.21)$$

$$y_1 = \underbrace{\bar{r}_{1,4n-1} \dots \bar{r}_{1,3n-1} \dots \bar{r}_{1,2n-1}}_{2n} \quad (6.22)$$



$$y_2 = \underbrace{\bar{r}_{1,2n} \dots \bar{r}_{1,n} \dots \bar{r}_{1,0}}_{2n} \quad (6.23)$$

$$y_3 = \underbrace{11 \dots 11 \bar{r}_{1,4n}}_{2n} \quad (6.24)$$

Therefore Equation (6.19) can be represented as:

$$w = |u_3 + y_1 + y_2 + y_3|_{2^{2n-1}}. \quad (6.25)$$

For simplicity sake, let us rewrite Equation (6.18) as:

$$X = r_1 + 2^{4n+1}w - 2^{2n}w = M + N. \quad (6.26)$$

where,

$$\begin{aligned} M &= r_1 + 2^{4n+1}w \\ &= \underbrace{w_{2n-1}w_{2n-2} \dots w_0}_{2n} \underbrace{r_{1,4n}r_{1,4n-1} \dots r_{1,0}}_{4n+1} \end{aligned} \quad (6.27)$$

$$\begin{aligned} N &= -2^{2n}w \\ &= \underbrace{11 \dots 11}_{2n+1} \underbrace{\bar{w}_{2n-1}\bar{w}_{2n-2} \dots \bar{w}_0}_{2n} \underbrace{11 \dots 11}_{2n} \end{aligned} \quad (6.28)$$



6.3 Hardware Realization

The hardware implementation of the proposed reverse converter for the moduli set $\{2^{2n}, 2^{2n+1} - 1, 2^{2n} - 1\}$ is based on Equations (6.7) and (6.18). Figure 6.1 illustrates the block diagram of the proposed converter. u_1 and u_2 are added modulo $2^{2n+1} - 1$ in order to obtain v . We speed up the addition process by using anticipated computation, where we compute $u_1 + u_2$ for both $cin = 0$ and $cin = 1$ and the right result is selected with a MUX. The computation of Equation (6.14) is obtained just by a shift and a concatenation operation with no cost on computational hardware. The four operands u_3, y_1, y_2 and y_3 are reduced to two numbers, sum and carry represented by, s_2 and c_2 by using 2 cascaded $2n$ bits CSA with end around carries (EACs). Then, the $2n$ bit modulo adder (CPA2) does the modulo addition of s_2 and c_2 to generate one number w , i.e., $w = |s_2 + c_2|_{2^{2n}-1} = |u_3 + y_1 + y_2 + y_3|_{2^{2n}-1}$. To speed up this addition, we utilize anticipated computation by computing $s_2 + c_2$ for both $cin = 0$ and $cin = 1$. This is followed by the selection of the right result with a MUX. In order to perform the addition represented by Equation (6.26), the two operands M and N must be expanded to $(6n + 1)$ -bit numbers, since M is a $(6n + 1)$ -bit number. The final result is then simplified by CPA5 with a constant carry-in of 1.

We now compute the complexity and delay of the residue to binary converter. It must be noted that some of the operands in Equation (6.18) results in the reduction of Full adders (FAs) to Half Adders (HAs), since some inputs of the CSA and CPA have constant values of one. It can be observed that Equation (6.17) has only one



bit of 1's and Equation (6.24) has $(2n - 1)$ bits of 1's. This means that in principle, $(14n + 2)$ FA is reduced to $(12n + 2)$ FA and $(2n)$ HA. This therefore indicates that, the proposed reverse converter requires $(12n + 2)$ FAs and $(2n)$ HAs. Also, the total delay required by the reverse converter is $(10n + 4)t_{FA} + 2t_{MUX}$.

6.4 Performance Comparison

The reverse converter presented in this chapter is for a novel moduli set $\{2^{2n+1} - 1, 2^{2n}, 2^{2n} - 1\}$ with a DR of $6n$ bit. Hence, to verify its performance against existing state of the art converters, we compared with a converter with similar DR $\{2^n - 1, 2^n + 1, 2^{2n}, 2^{2n} + 1\}$ Molahosseini et al. (2010). Also, to show that our proposal can complete favourably with some existing state of the art with $5n$ bit DR, we include the converter $\{2^{2n+1} - 1, 2^{2n}, 2^n - 1\}$ Gbolagade et al. (2009), for the fact that the moduli set under consideration is an enhanced form of $\{2^{2n+1} - 1, 2^{2n}, 2^n - 1\}$. We compare their performances in terms of area and delay both theoretically and experimentally.

6.4.1 Theoretical Analysis

The proposed reverse converter requires a delay of $(10n + 4)t_{FA} + 2t_{MUX}$, while that of $\{2^n - 1, 2^n + 1, 2^{2n}, 2^{2n} + 1\}$ is $(8n + 3)t_{FA}$, while $\{2^{2n+1} - 1, 2^{2n}, 2^n - 1\}$ exhibits a delay of $(13n + 7)t_{FA}$. In terms of area, our proposed reverse converter utilizes



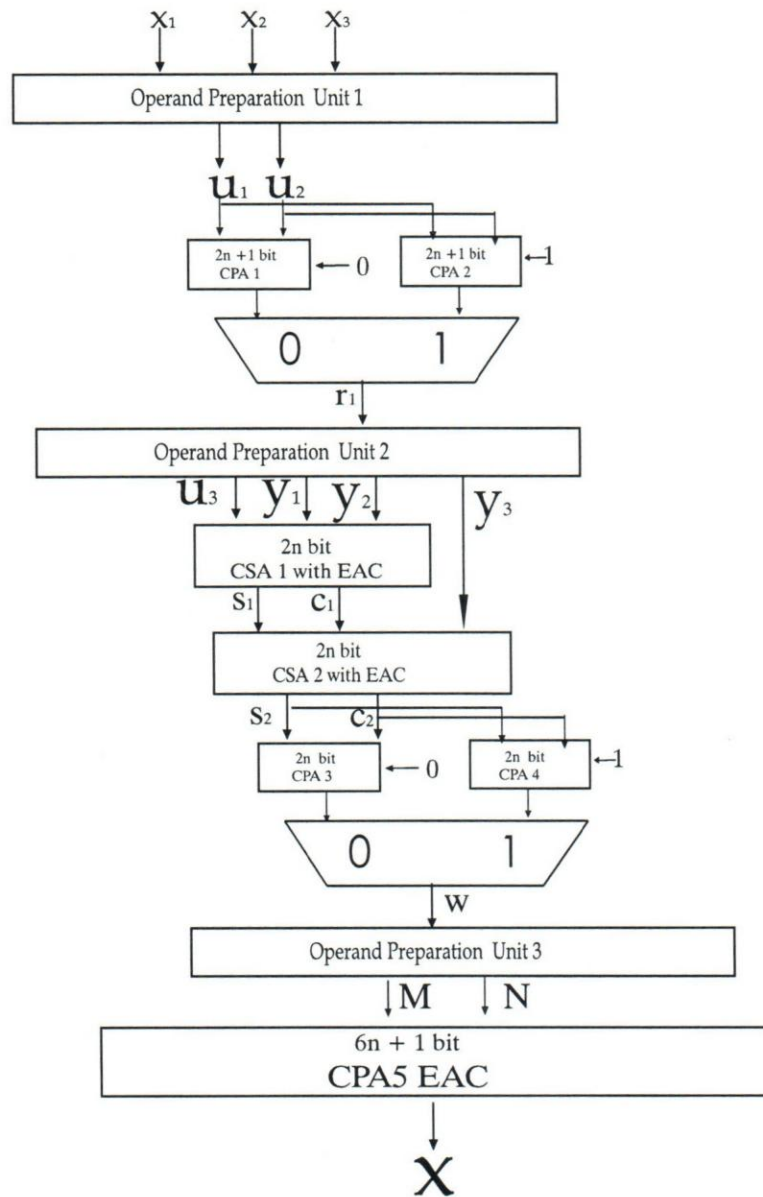


Figure 6.1: Block diagram for the Proposed Converter

Table 6.1: Theoretical Delay and Area Comparison

Converter	Gbolagade et al. (2009)	Molahosseini et al. (2010)	Proposed
DR	$5n$	$6n$	$6n$
FA	$15n + 2$	$10n + 6$	$12n + 2$
HA	$4n + 3$	$6n - 6$	$2n$
Area Cost in HA (Δ)	$34n + 7$	$26n + 6$	$26n + 4$
Delay (τ)	$(13n + 7)t_{FA}$	$(8n + 3)t_{FA}$	$(10n + 4)t_{FA} + 2t_{MUX}$

$(12n + 2)$ FA and $(2n)$ HA. For $\{2^n - 1, 2^n + 1, 2^{2n}, 2^{2n} + 1\}$, it requires $(10n + 6)$ FA and $(6n - 6)$ HA area resources, while $\{2^{2n+1} - 1, 2^{2n}, 2^n - 1\}$ utilizes $(15n + 2)$ FA and $(4n + 3)$ HA. Clearly, it can be seen that our proposal outperforms the two state of the art converters in terms of only area while for conversion time, the converter $\{2^n - 1, 2^n + 1, 2^{2n}, 2^{2n} + 1\}$ seems to be better. Further, to simplify the area comparison we assume that one FA has an area about twice the HA and expressed the area cost for all the considered designs in terms of HAs. The result of the comparison is summarized in Table 6.1.

6.4.2 Experimental Analysis

For a more accurate comparison, we described the converters in VHSIC Hardware Description Language (VHDL) and simulated with ISE v.14.3. Using the VHDL descriptions of our proposed converter and state of the art converters, implementation using xc6slx45t-3fpg484 as target FPGA technology device was successfully achieved. The area is evaluated by the number of occupied slices LUTs. Table 6.2 compares the delay and area of the converters for a range of values of n . The experimental results



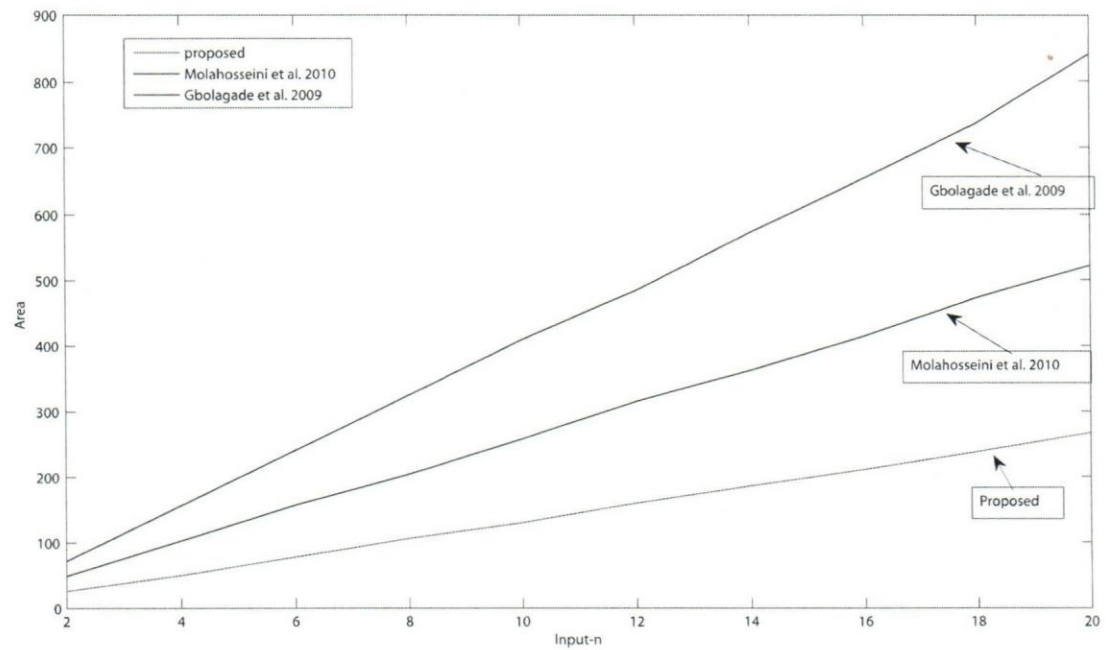


Figure 6.2: The Area Comparison of Converters

contradicts the theoretical results. Rather, it indicates clearly that our converter outperforms the state of the art. The delay and area of our proposal are the least compared to existing state of the art. In order to ease the comparison, Figures 6.2 and 6.3 are generated to show the practical conversion time and hardware resources of our design against the state of the art. The synthesis results indicate that, on the average our converter is 32.75% faster than the converter proposed by Molahosseini et al. (2010) and exhibits a 49.21% reduction in hardware resources.

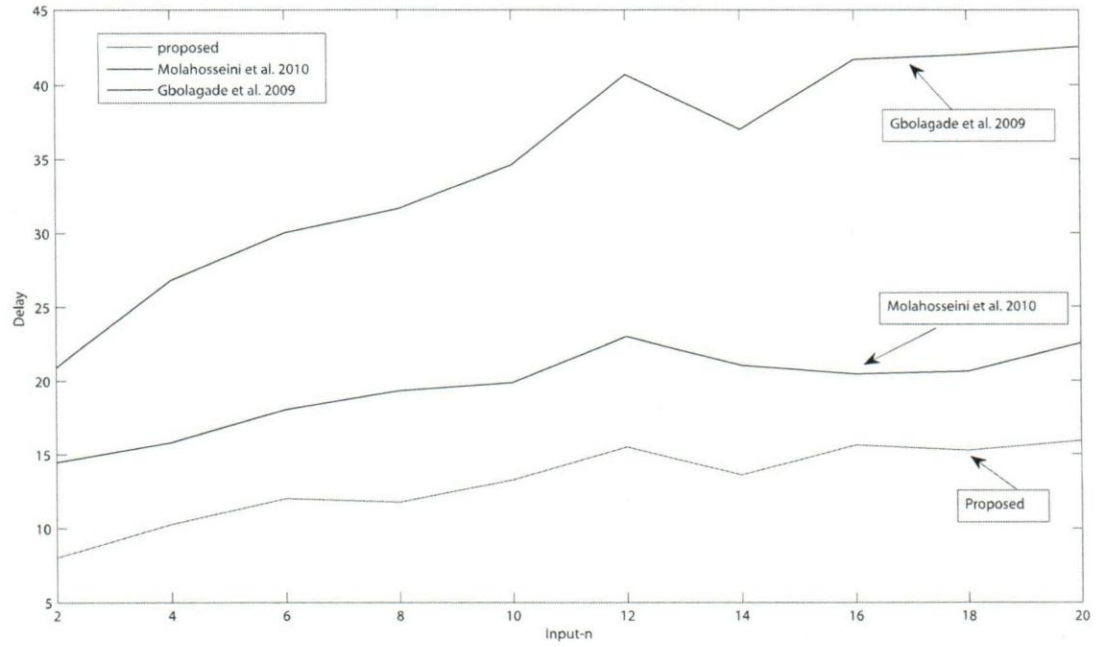


Figure 6.3: The Delay Comparison of Converters

6.5 Conclusion

In this chapter, we proposed a new moduli set $\{2^{2n}, 2^{2n+1} - 1, 2^{2n} - 1\}$ which is an enhancement of the moduli set $\{2^n, 2^{2n+1} - 1, 2^{2n} - 1\}$ with its associated efficient RNS to binary converter. The proposed reverse converter is based on MRC. The divide and conquer approach was used to implement the MRC, where the moduli set is grouped into two. The first phase combines the first and the second residue with respect to the subset $\{2^{2n}, 2^{2n+1} - 1\}$. In the second phase, the result of the first phase is combined with the third residue with respect to the composite moduli

Table 6.2: Experimental Delay and Area Comparison

n	Gbolagade et al. (2009)		Molahosseini et al. (2010)		Proposed	
	Delay	Area	Delay	Area	Delay	Area
2	20.891	72	14.479	49	8.039	26
4	26.823	156	15.815	103	10.286	50
6	30.043	241	18.043	157	11.989	78
8	31.666	352	19.324	204	11.757	106
10	34.596	409	19.883	258	13.253	130
12	40.696	484	23.003	315	15.497	160
14	37.016	573	21.043	363	13.601	186
16	41.712	654	20.458	415	15.692	211
18	42.025	737	20.621	473	15.261	238
20	42.570	841	22.569	522	15.924	267

set $\{(2^{2n})(2^{2n+1} - 1), 2^{2n} - 1\}$. Additionally, we further simplified the resulting architecture in order to obtain a reverse converter that utilizes only 2 levels of CSAs together with three CPAs. Theoretically analysis reveals that our proposal has a delay of $(10n+4)t_{FA}+2t_{MUX}$ with an area cost of $(12n+2)FAs$ and $(2n)HAs$. Subsequently, we expressed the required hardware resources in terms of HA. For the experimental assessment, we described our converter and those presented by Gbolagade et al. (2009) and Molahosseini et al. (2010) in VHDL and subsequently implemented them on an FPGA. A wide range of values of n were investigated through the implementation. The synthesis results clearly indicate that, on the average our reverse converter is capable of performing 32.75% faster than the one proposed by Molahosseini et al. (2010). Also, in terms of hardware resources, our proposal exhibits a 49.21% reduction when compared with the same state of the art. These results show that the proposed new moduli set with its associated reverse converter achieves better performance in terms of area and delay than existing state of the art converters.



Given that moduli sets with larger DR are of practical interest, we propose in the chapter that follows, a new moduli set $\{2^{2n+1} - 1, 2^{2n+1}, 2^{2n} - 1\}$ with its associated efficient residue to binary converter.





CHAPTER 7

A NOVEL MODULI SET $\{2^{2N+1} - 1, 2^{2N+1}, 2^{2N} - 1\}$

As illustrated in earlier chapters, several interesting powers of two moduli sets, such as $\{2^n, 2^n - 1, 2^n + 1\}$ (Bankas and Gbolagade, 2012), (Wang et al., 2002), $\{2^n, 2^n - 1, 2^{n-1} - 1\}$ (Hosseinzadeh et al., 2009), (Hiasat and Abdel-AtyZohdy, 1998), $\{2^{n+1} + 1, 2^{n+1} - 1, 2^n\}$, (Molahosseini and Navi, 2007), $\{2^n, 2^{n+1} - 1, 2^n - 1\}$ (Mohan, 2007) have been investigated. This is because they are more applicable to Digital Signal Processing Applications. Based on these moduli sets, efficient reverse conversion algorithms, which give room for a widespread utilization of RNS in special purpose processors, have been proposed (Molahosseini et al., 2010).

Interestingly, due to the fact that arithmetic operations with respect to $2^n + 1$ modulus is not as simple as the end around carry based $2^n - 1$ modulus and this degrades the RNS architecture performance, the moduli set $\{2^{n+1} - 1, 2^n, 2^n - 1\}$ presented by Mohan (2007) was proposed by removing the modulus $2^n + 1$ from the moduli set $\{2^{n+1} - 1, 2^n + 1, 2^n, 2^n - 1\}$. Subsequently, the moduli sets $\{2^{2n+1} - 1, 2^n, 2^n - 1\}$ and $\{2^{2n+1} - 1, 2^{2n}, 2^n - 1\}$, with their associated converters, were proposed by



Molahosseini et al. (2008) and Gbolagade et al. (2009), respectively. Given that applications requiring larger Dynamic Range (DR) than the ones offered by the moduli set $\{2^{2n+1} - 1, 2^{2n}, 2^n - 1\}$ are of practical interest, we propose in this chapter the moduli set $\{2^{2n+1} - 1, 2^{2n+1}, 2^{2n} - 1\}$ as an alternative candidate to the one in Gbolagade et al. (2009). Apart from having a larger DR, the suggested moduli set is more balanced when compared with the moduli set $\{2^{2n+1} - 1, 2^{2n}, 2^n - 1\}$. After demonstrating that this moduli set always results in legitimate RNS, we propose two new novel reverse converters based on the New Chinese Remainder Theorem (CRT) and Mixed Radix Conversion (MRC).

The rest of the chapter is structured as follows. Section 7.1 provides a brief background on reverse conversion methods. In Section 7.2, the novel moduli set is introduced, and the corresponding algorithms are presented in Section 7.3. Section 7.4 describes the hardware implementation of the proposed converter. The performance of our proposal is evaluated and compared with the state of the art in Section 7.5. The chapter is concluded in Section 7.6.

7.1 Background

Given a moduli set $\{m_i\}_{i=1,k}$, the residues (x_1, x_2, \dots, x_k) can be converted into the corresponding decimal (base 10) number X , according to the MRC as follows, (Szabo and Tanaka, 1967):

$$X = a_1 + a_2 m_1 + a_3 m_1 m_2 \quad (7.1)$$

where a set of digits $\{a_1, a_2, a_3\}$, which are the Mixed Radix Digits (MRDs) are given as follows:

$$\begin{aligned} a_1 &= x_1 \\ a_2 &= \left| (x_2 - a_1) |m_1^{-1}|_{m_2} \right|_{m_2} \\ a_3 &= \left| \left((x_3 - a_1) |m_1^{-1}|_{m_3} - a_2 \right) |m_2^{-1}|_{m_3} \right|_{m_3} \end{aligned} \quad (7.2)$$

Similarly, the New CRT I algorithm proposed by Wang (1998) can also be used to convert RNS to its decimal equivalent. Given a 3-moduli set $\{m_1, m_2, m_3\}$, the number X can be converted from its residue representation (x_1, x_2, x_3) as follows:

$$X = x_1 + m_1 |k_1(x_2 - x_1) + k_2 m_2(x_3 - x_2)|_{m_2 m_3}, \quad (7.3)$$

where,

$$|k_1 m_1|_{m_2 m_3} = 1, \quad (7.4)$$

$$|k_2 m_1 m_2|_{m_3} = 1. \quad (7.5)$$



It has been demonstrated by Bi et al. (2004) that, given any integer B , m_1 , and m_2 ,

$$|B|_{m_1 m_2} = m_1 \left\lfloor \frac{B}{m_1} \right\rfloor_{m_2} + |B|_{m_1}. \quad (7.6)$$

This property is used in the proof of one of our theorems.

7.2 $\{2^{2n+1} - 1, 2^{2n+1}, 2^{2n} - 1\}$ Moduli set

For a given RNS moduli set to be valid, it is required that all the elements in the moduli set are co-prime. Thus in order to prove that the proposed set can be utilized for the construction of valid RNS architecture, we have to demonstrate that the moduli $2^{2n+1} - 1$, 2^{2n+1} , and $2^{2n} - 1$ are pair-wise relatively prime.

Theorem 7.1. *The moduli $2^{2n+1} - 1$, 2^{2n+1} , and $2^{2n} - 1$ are pair-wise relatively prime numbers.*

Proof. :

From the Euclidean theorem, we have $\gcd(a, b) = \gcd(b, |a|_b)$, therefore, $\gcd(2^{2n+1} - 1, 2^{2n+1}) = \gcd(2^{2n+1}, |2^{2n+1} - 1|_{2^{2n+1}}) = 1$.

Similarly, $\gcd(2^{2n+1}, 2^{2n} - 1) = \gcd(2^{2n} - 1, |2^{2n+1}|_{2^{2n} - 1}) = 1$. Again, $\gcd(2^{2n+1} - 1, 2^{2n} - 1) = \gcd(2^{2n} - 1, |2^{2n+1} - 1|_{2^{2n} - 1}) = 1$. Thus, from these results, it can be concluded that the moduli set $\{2^{2n+1} - 1, 2^{2n+1}, 2^{2n} - 1\}$ contains relatively prime moduli and it is a valid RNS moduli set. \square



7.3 New CRT I Based Converter

Given the RNS number representation (x_1, x_2, x_3) for the moduli set

$\{2^{2n+1} - 1, 2^{2n+1}, 2^{2n} - 1\}$, we propose an algorithm that computes its decimal equivalent based on New CRT (Wang, 1998).

Theorem 7.2. *Given the $\{2^{2n+1} - 1, 2^{2n+1}, 2^{2n} - 1\}$ moduli set, the following hold true:*

$$(2^{2n+1} - 1)^{-1} \big|_{(2^{2n+1})(2^{2n}-1)} = 2^{2n+1} - 1, \quad (7.7)$$

$$((2^{2n+1} - 1)(2^{2n+1}))^{-1} \big|_{(2^{2n}-1)} = 2^{2n-1}. \quad (7.8)$$

Proof. :

If it can be demonstrated that

$$(2^{2n+1} - 1) \times (2^{2n+1} - 1) \big|_{(2^{2n+1})(2^{2n}-1)} = 1, \quad (7.9)$$

then $2^{2n+1} - 1$ is the multiplicative inverse of $2^{2n+1} - 1$ with respect to $(2^{2n+1})(2^{2n} - 1)$.

$$\begin{aligned} (2^{2n+1} - 1) \times (2^{2n+1} - 1) \big|_{(2^{2n+1})(2^{2n}-1)} &= |2^{4n+2} - 2^{2n+2} + 1|_{2^{4n+1}-2^{2n+1}} \\ &= |2^{2n+2} - 2^{2n+2} + 1|_{2^{4n+1}-2^{2n+1}} \\ &= 0 + 1 = 1. \end{aligned} \quad (7.10)$$



Similarly,

$$\begin{aligned} |(2^{2n+1} - 1) \times (2^{2n+1}) \times (2^{2n-1})|_{(2^{2n-1})} &= |(2^{2n+1} - 1) \times (2^{4n})|_{(2^{2n-1})} \\ &= |(2^{2n+1} - 1)|_{(2^{2n-1})} = 1. \end{aligned} \quad (7.11)$$

Thus Equation (7.7) and (7.8) holds true. \square

Theorem 7.3. *The decimal equivalent of the RNS number (x_1, x_2, x_3) with respect to the moduli set $\{m_1, m_2, m_3\}$ in the form $\{2^{2n+1} - 1, 2^{2n+1}, 2^{2n} - 1\}$, can be computed as follows:*

$$X = x_1 + (2^{2n+1} - 1)(2^{2n+1})w + (2^{2n+1} - 1)|x_1 - x_2|_{2^{2n+1}}, \quad (7.12)$$

where, $w = |[x_2 - x_1 + 2^{2n-1}x_3 - 2^{2n-1}x_2]|_{2^{2n-1}}$.

Proof. :

Substituting Equation (7.7) and (7.8) into Equation (7.3), we obtain:

$$X = x_1 + (2^{2n+1} - 1) |(2^{2n+1} - 1)(x_2 - x_1) + (2^{2n-1})(2^{2n+1})(x_3 - x_2)|_{(2^{2n+1})(2^{2n-1})}. \quad (7.13)$$

Simplifying further,

$$X = x_1 + (2^{2n+1} - 1) |(2^{2n+1} - 1)(x_2 - x_1) + (2^{4n})(x_3 - x_2)|_{(2^{2n+1})(2^{2n-1})}. \quad (7.14)$$



Rewriting (7.14), we have:

$$X = x_1 + (2^{2n+1} - 1) |T|_{(2^{2n+1})(2^{2n-1})}, \quad (7.15)$$

where,

$$T = (2^{2n+1} - 1)(x_2 - x_1) + (2^{4n})(x_3 - x_2). \quad (7.16)$$

Applying Equation (7.6) to simplify (7.15), we obtain:

$$X = x_1 + (2^{2n+1} - 1)(2^{2n+1})w + (2^{2n+1} - 1)C, \quad (7.17)$$

where,

$$\begin{aligned} w &= \left| \left[\frac{T}{2^{2n+1}} \right] \right|_{2^{2n-1}} \\ &= \left| [x_2 - x_1 + 2^{2n-1}x_3 - 2^{2n-1}x_2] \right|_{2^{2n-1}}, \end{aligned} \quad (7.18)$$

and

$$C = |x_1 - x_2|_{2^{2n+1}}. \quad (7.19)$$



Further simplification gives,

$$X = x_1 + 2^{4n+2}w - 2^{2n+1}w + 2^{2n+1}C - C, \quad (7.20)$$

where all the parameters remain as defined. \square

We can further reduce the hardware complexity by making use of the following properties to simplify Equation (7.20), Bankas and Gbolagade (2012).

Property 1: The multiplication of a residue number by 2^k in modulo $(2^p - 1)$ is computed by a k bit circular left shift.

Property 2: A negative number in modulo $(2^p - 1)$ is calculated by subtracting the number in question from $(2^p - 1)$. In binary representation, the one's complement of the number gives the result.

Let the residues (x_1, x_2, x_3) have the binary representation as follows:

$$x_1 = (x_{1,2n}x_{1,2n-1}\dots x_{1,1}x_{1,0}), \quad (7.21)$$

$$x_2 = (x_{2,2n}x_{2,2n-1}\dots x_{2,1}x_{2,0}), \quad (7.22)$$

$$x_3 = (x_{3,2n-1}x_{3,2n-2}\dots x_{3,1}x_{3,0}). \quad (7.23)$$



In Equation (7.18), the various parameters are simplified using Property 1 and 2 as follows:

$$\begin{aligned}
 v_1 &= |x_2|_{2^{2n-1}} \\
 &= |(x_{2,2n}x_{2,2n-1}\dots x_{2,0})|_{2^{2n-1}} \\
 &= \left(\underbrace{x_{2,2n-1}x_{2,2n-2}\dots x_{2,0} \wedge x_{2,2n}}_{2n} \right) \quad (7.24)
 \end{aligned}$$

$$|-x_1|_{2^{2n-1}} = |v_2' + v_2''|_{2^{2n-1}} \quad (7.25)$$

$$v_2' = \left(\underbrace{11\dots 11\bar{x}_{1,2n}}_{2n} \right) \quad (7.26)$$

$$v_2'' = \left(\underbrace{\bar{x}_{1,2n-1}\bar{x}_{1,2n-2}\dots\bar{x}_{1,0}}_{2n} \right) \quad (7.27)$$

$$\begin{aligned}
 v_3 &= |2^{2n-1}x_3|_{2^{2n-1}} \\
 &= |2^{2n-1}(x_{3,2n-1}x_{3,2n-2}\dots x_{3,0})|_{2^{2n-1}} \\
 &= \left(\underbrace{x_{3,0}x_{3,2n-1}\dots x_{3,1}}_{2n} \right) \quad (7.28)
 \end{aligned}$$



$$|-2^{2n-1}x_2|_{2^{2n-1}} = |v'_4 + v''_4|_{2^{2n-1}} \quad (7.29)$$

$$v'_4 = \left(\underbrace{\bar{x}_{2,2n}11\dots11}_{2n} \right) \quad (7.30)$$

$$v''_4 = \left(\underbrace{\bar{x}_{2,0}\bar{x}_{2,2n-1}\dots\bar{x}_{2,1}}_{2n} \right) \quad (7.31)$$

Now, let us rewrite Equation (7.18) as:

$$\begin{aligned} w &= \left| \left[\frac{T}{2^{2n+1}} \right] \right|_{2^{2n-1}} \\ &= \left| \left[v_1 + v''_2 + v_3 + v''_4 + v'_4 + v'_2 \right] \right|_{2^{2n-1}}, \end{aligned} \quad (7.32)$$

where, $T, v_1, v''_2, v_3, v''_4, v'_4$, and v'_2 are represented by Equations (7.16), (7.24), (7.27), (7.28), (7.31), (7.30) and (7.26), respectively. Similarly, $C = x_{1,2n}x_{1,2n-1}\dots x_{1,1}x_{1,0} + \bar{x}_{2,2n}\bar{x}_{2,2n-1}\dots\bar{x}_{2,1}\bar{x}_{2,0}$.

Finally, Equation (7.20) may be simplified as follows:

$$X = M + N \quad (7.33)$$



where,

$$M = \underbrace{w_{2n-1}w_{2n-2}\dots w_0}_{2n} \underbrace{C_{2n}C_{2n-1}\dots C_0}_{2n+1} \underbrace{x_{1,2n}x_{1,2n-1}\dots x_{1,0}}_{2n+1} \quad (7.34)$$

$$N = \underbrace{11\dots 11}_{2n+1} \underbrace{\bar{w}_{2n-1}\bar{w}_{2n-2}\dots \bar{w}_0}_{2n} \underbrace{\bar{C}_{2n}\bar{C}_{2n-1}\dots \bar{C}_0}_{2n+1} \quad (7.35)$$

7.4 Hardware Implementation

The hardware structure of the proposed reverse converter for the Moduli set $\{2^{2n+1} - 1, 2^{2n+1}, 2^{2n} - 1\}$ relies on Equation (7.20) and (7.32). The realization of Equation (7.20) is as a result of the reduction of modulo $(2^{2n+1})(2^{2n} - 1)$ operation in Equation (7.15) into two modulo operations in parallel. One is based on 2^{2n+1} and the other is based on $2^{2n} - 1$. Also, the operand T is reduced to $\lfloor \frac{T}{2^{2n+1}} \rfloor$ without any additional hardware required. It must be noted that, $|x_1 - x_2|_{2^{2n+1}}$ is a truncation and a $(2n + 1)$ Least Significant Bits (LSB) of T , while the floor of T , i.e., $\lfloor \frac{T}{2^{2n+1}} \rfloor|_{2^{2n-1}}$ is the remaining part after the truncation. Also, since T is a $(4n + 1)$ bit number, $\lfloor \frac{T}{2^{2n+1}} \rfloor|_{2^{2n-1}}$ is the $2n$ Most Significant Bit (MSB) of T .

The parameters $v_1, v_2'', v_3, v_4'', v_4'$, and v_2' in Equation (7.32) are added by 4 cascaded $2n$ bit Carry Save Adders (CSAs) with end-around carries (EACs) resulting in the values s_4 and c_4 which are further reduced to one number w by modulo $2^{2n} - 1$ carry





propagate adder in $(4n)t_{FA}$ delay. The floor of the parameters in (7.32) must be added modulo $2^{2n} - 1$. Again, to speed up the computation process in (7.32), we anticipate two cases. For case 1, if $x_2 - x_1 + 2^{2n-1}x_3 - 2^{2n-1}x_2 < 0$, then $2^{2n+1} \lfloor \frac{T}{2^{2n+1}} \rfloor = 0$, reducing Equation (7.20) drastically in terms of computational load. Case 2 involves all the parameters in the addition modulo $2^{2n} - 1$. We utilize anticipated computation to speed up the parallel addition of x_1 and x_2 with CPA1 with EAC. We compute $x_1 + x_2$ for both $cin = 0$ and $cin = 1$ and then select the right result with a MUX with a conversion time of $(2n+1)t_{FA}$. Simultaneously, the other branch which involves the addition of 4 cascaded $2n$ bit CSAs is also subjected to anticipated computation which results in $(2n+4)t_{FA}$ delay. The final result is obtained with a delay of $(6n+2)t_{FA}$. Hence, the proposed converter has a total delay of $(8n+6)t_{FA} + 2t_{MUX}$.

It is interesting to note that, some of the operands in Equation (7.32) result in the reduction of FAs to HAs. Since both Equation (7.26) and (7.30) have $(2n-1)$ bits of 1's, $(2n)$ bits of the Full Adders (FAs) in the CSA with EAC are reduced to $(2n-1)$ HAs in CSA1 and CSA3 operands. Also, the implementation of Equation (7.33) for the final result utilizes $(2n+1)HA$ and $(4n+1)FA$. This means that the proposed converter uses $(12n+4)FA$ and $(6n-1)HA$ hardware resources. The overall structure of the proposed converter is illustrated in Figure 7.1. We note that the operand preparation units do not require any additional hardware and do not induce any extra delay as they just concatenate and place the bits in the right position by proper wire routing.

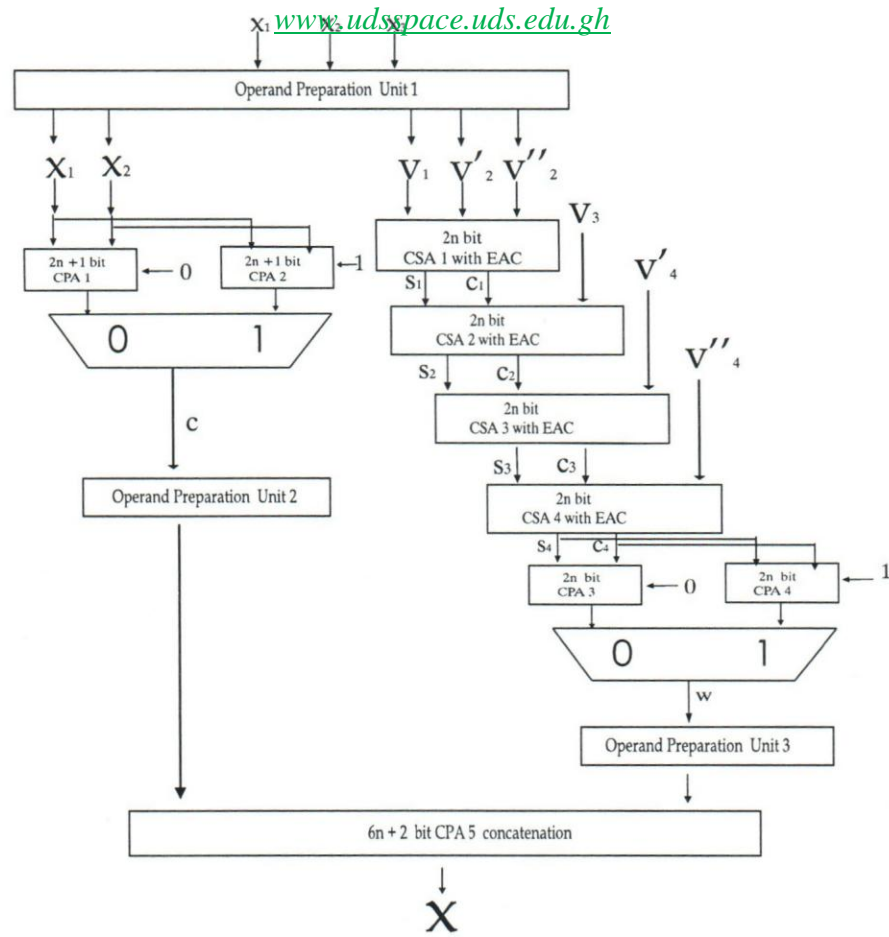


Figure 7.1: Block diagram for the Proposed Converter



7.5 Performance Analysis

The performance of the proposed reverse converter is evaluated theoretically in terms of conversion time and area cost. The hardware utilization of our proposal is computed in terms of Full Adders (FAs) and Half Adders (HAs). We compared our proposal with equivalent best known state of the art reverse converters presented by Molahosseini et al. (2010) and Molahosseini and Navi (2010). Molahosseini et al. (2010) presented two four-moduli sets $\{2^n - 1, 2^n, 2^n + 1, 2^{2n+1} - 1\}$ and $\{2^n - 1, 2^n + 1, 2^{2n}, 2^{2n} + 1\}$ with their corresponding reverse converters.

We note that for reverse converters with different moduli sets to be fairly compared, the moduli sets should provide the same dynamic range. That notwithstanding, we compared our proposal with similar dynamic range moduli set and other existing lesser dynamic range moduli sets. We therefore carried out both theoretical and experimental evaluation of our proposal with existing best state of the art.

7.5.1 Theoretical Evaluation

The performance of the proposed converter is evaluated in terms of conversion delay and area cost. Our scheme requires a delay of $(8n + 6)t_{FA} + 2t_{MUX}$ while the best known $6n$ bit DR presented by Molahosseini and Navi (2010) has $(12n + 6)t_{FA}$ delay. In terms of area, our proposal utilizes $(12n + 4)FA$ and $(6n - 1)HA$, while the best known state of the art requires $(12n + 2)FA$ s and $(4n + 1)HA$ s. To simplify the area comparison we assumed that one FA has an area about twice as large than the

Table 7.1: Theoretical Area and Delay Comparison

Converters	Molahosseini et al. (2010)	Molahosseini and Navi (2010)	Proposed Converter
	$\{2^n - 1, 2^n, 2^n + 1, 2^{2n+1} - 1\}$	$\{2^n - 1, 2^n + 1, 2^{2n}, 2^{2n+1} - 1\}$	Proposed Converter
DR	$5n$	$6n$	$6n$
FA	$8n + 2$	$12n + 2$	$12n + 4$
HA	$5n$	$4n + 1$	$6n - 1$
Area Cost in HA (Δ)	$21n + 4$	$28n + 5$	$30n + 7$
Delay (τ)	$(12n + 5)t_{FA}$	$(12n + 6)t_{FA}$	$(8n + 6)t_{FA} + 2t_{MUX}$

HA and expressed the area cost for all the considered designs in terms of HAs. The result of this comparison is presented in Table 7.1. In Table 7.1, it can be seen that our converter is faster while requiring some more hardware resources.

7.5.2 Experimental Evaluation

For the experimental assessment, we described the converters in VHDL and implemented them on a Xilinx Spartan 6 FPGA with the target device xc6slx45t-3fpg484 to first, verify the correctness of the designs and then synthesized the designs. A performance evaluation is carried out in terms of area and delay for each converter. The results are given in terms of the number of slices and the input-to-output gate delays in nanoseconds. Table 7.2 presents a summary of the various dynamic range requirements. To confirm the theoretical assessment, the experimental results indicate that, averagely, our proposed scheme is 37.34% faster than the converter presented by Molahosseini and Navi (2010) for $\{2^n - 1, 2^n + 1, 2^{2n}, 2^{2n+1} - 1\}$. While in terms of area, our proposal requires about 13.34% area resources. The performance of our proposal against the state of the art converters in terms of area and delay, is depicted in Figures 7.2 and 7.3 respectively. In order to obtain an adequate comparison,



Table 7.2: Experimental Delay and Area Comparison for $\{2^{2n+1} - 1, 2^{2n+1}, 2^{2n} - 1\}$

n	Molahosseini et al. (2010)		Molahosseini and Navi (2010)		Proposed Converter	
	$\{2^n - 1, 2^n, 2^n + 1, 2^{2n+1} - 1\}$ (MOL-I)		$\{2^n - 1, 2^n + 1, 2^{2n}, 2^{2n+1} - 1\}$ (MOL-II)		$\{2^{2n+1} - 1, 2^{2n+1}, 2^{2n} - 1\}$	
	Delay	Area	Delay	Area	Delay	Area
2	22.980	51	26.650	72	20.679	84
4	35.459	112	37.011	141	25.261	169
6	35.115	176	43.637	209	28.059	243
8	38.190	239	42.755	275	29.956	328
10	40.730	299	51.522	351	31.535	408
12	43.399	351	51.892	426	32.868	494
14	44.343	422	56.011	498	32.945	572
16	46.909	474	57.778	563	34.451	651
18	48.196	537	61.110	635	34.766	727
20	52.797	600	59.663	713	35.286	823

Table 7.3: Area-Time square ($\Delta\tau^2$) Comparison

n	$\{2^n - 1, 2^n + 1, 2^{2n}, 2^{2n+1} - 1\}$	Proposed Converter
	$(\Delta\tau^2)$	$(\Delta\tau^2)$
2	51136.0200	35920.1674
4	193143.7911	107841.9624
6	397975.2437	191315.7179
8	502697.2569	294334.715
10	931735.2859	405738.1398
12	1147124.1370	533670.8795
14	1562341.5960	620833.3703
16	1879461.3710	772653.2821
18	2371364.3840	878706.5476
20	2538047.2550	1024718.778

the Area-Time square efficiency metric was used. The Area-Time square metric, illustrated in Table 7.3 and depicted in Figure 7.4 suggests that our proposed reverse converter is 57.96% efficient than the one proposed by Molahosseini and Navi (2010) for $\{2^n - 1, 2^n + 1, 2^{2n}, 2^{2n+1} - 1\}$.



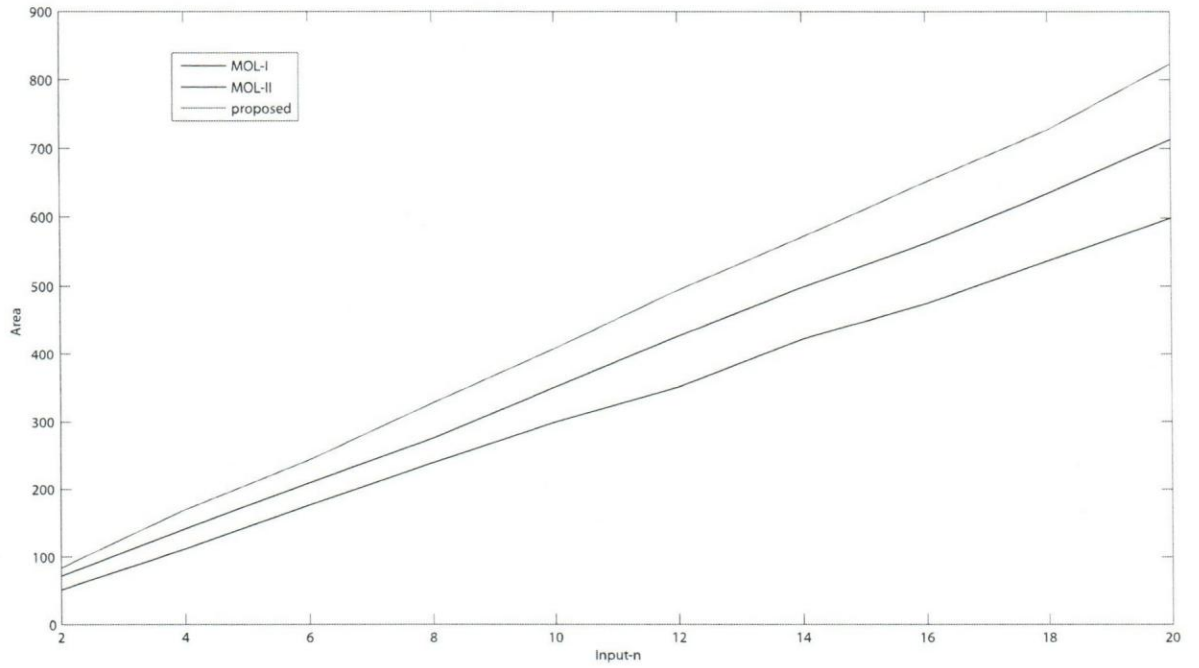


Figure 7.2: The Area Comparison of Converters

7.6 Conclusion

In this chapter, we introduced a new moduli set $\{2^{2n+1} - 1, 2^{2n+1}, 2^{2n} - 1\}$ which contains only low-cost moduli and has a large dynamic range. A novel and effective reverse converter is proposed based on the New CRT. Additionally, we further simplified the resulting architecture in order to obtain a reverse converter that utilizes 4 levels of CSAs namely, CSA1, CSA2, CSA3, and CSA4 together with two CPAs i.e., CPA1 or CPA2 and CPA3 or CPA4. The proposed converter is purely adder based and memoryless. Both theoretical and experimental assessment of our



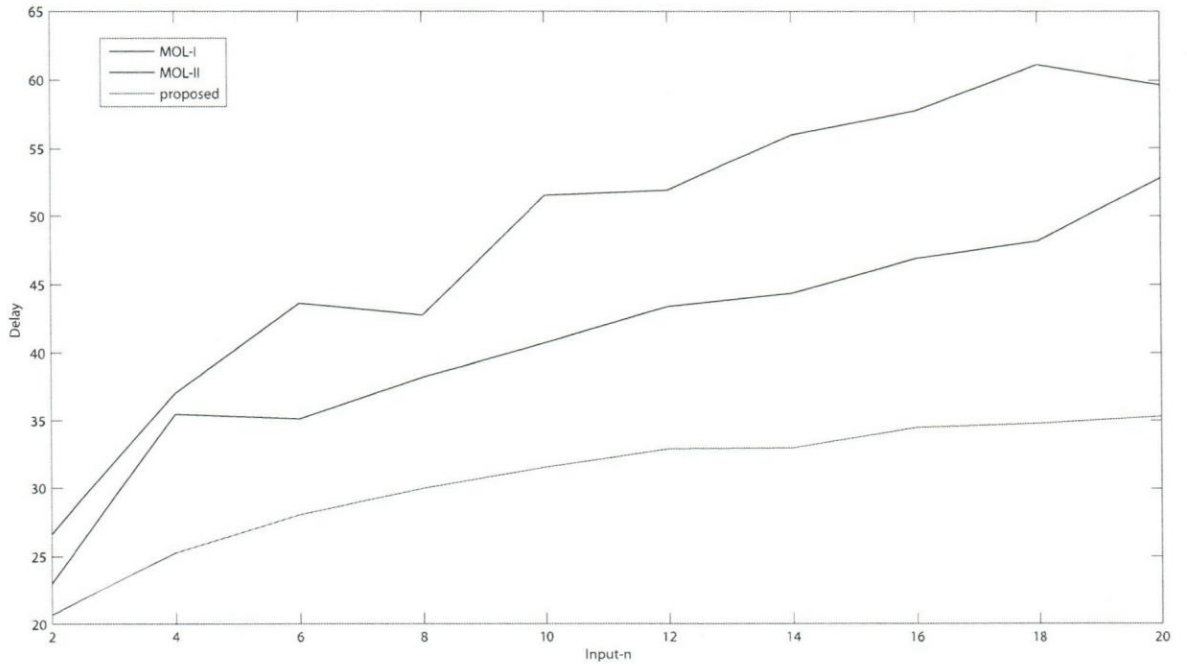


Figure 7.3: The Delay Comparison of Converters

proposal against best state of the art were carried out. Theoretically speaking, our proposal has a delay of $(8n + 6)t_{FA} + 2t_{MUX}$ with an area cost of $(12n + 4)FAs$ and $(6n - 1)HAs$. We compared the proposed reverse converter with state of the art converters $\{2^n - 1, 2^n, 2^n + 1, 2^{2n+1} - 1\}$ and $\{2^n - 1, 2^n + 1, 2^{2n}, 2^{2n+1} - 1\}$ proposed by Molahosseini et al. (2010) and Molahosseini and Navi (2010) respectively. The result therefore shows that our scheme is faster than the equivalent state of the art converter $\{2^n - 1, 2^n + 1, 2^{2n}, 2^{2n+1} - 1\}$, but requires some more area resources.



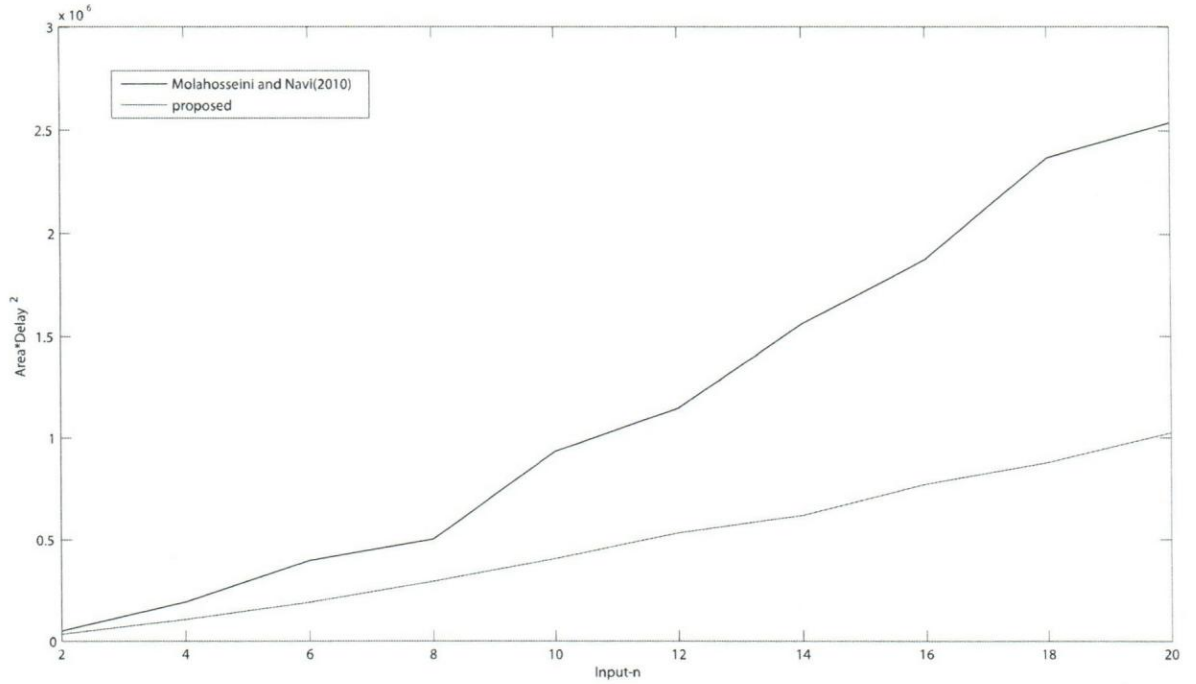


Figure 7.4: The Area Time square metric Comparison of Converters

Further, for the experimental comparison, we described our proposed reverse converter and the considered state of the art in VHDL and performed the implementation on an FPGA using a wide range of values on n . The synthesis results confirmed the theoretical analysis. In terms of speed, our converter is 37.34% faster than the converter presented by Molahosseini and Navi (2010) for $\{2^n - 1, 2^n + 1, 2^{2n}, 2^{2n+1} - 1\}$. While in terms of area, our proposal requires about 13.34% more area resources. In order to ascertain the efficiency of our scheme, we used the Area-Time Square ($\Delta\tau^2$) metric which is presented in Table (7.3). The ($\Delta\tau^2$)

metric suggests that our converter is 57.96% more efficient than the one presented by Molahosseini and Navi (2010) for $\{2^n - 1, 2^n + 1, 2^{2n}, 2^{2n+1} - 1\}$.

In the chapter that follows, we propose a new $6n$ bit DR moduli set $\{2^{2n}, 2^{2n} + 1, 2^{2n} - 1\}$ by enhancing the modulus 2^n to 2^{2n} in the $\{2^n, 2^{2n} + 1, 2^{2n} - 1\}$ moduli set. After demonstrating that this moduli set is valid for RNS, we formulate its corresponding efficient reverse converter.





CHAPTER 8

A $6N$ BIT DYNAMIC RANGE MODULI SET CONTAINING $(2^K + 1)$ MODULUS

The basis for RNS Implementation is a moduli set which consists of a set of relatively prime numbers. Many different moduli sets have been investigated with several reverse conversion structures with $3n$ bit DR such as $\{2^n, 2^n - 1, 2^n + 1\}$ (Bankas and Gbolagade, 2012), (Wang et al., 2002), $\{2^{n+1}, 2^n, 2^n - 1\}$ (Mohan, 2007), $\{2^{n+1} + 1, 2^{n+1} - 1, 2^n\}$ (Molahosseini and Navi, 2007), $\{2^n, 2^{n+1} - 1, 2^n - 1\}$ (Lin et al., 2008) have been proposed in order to reduce RNS processor's hardware complexity to give room for a widespread utilization of RNS in special purpose processors (Molahosseini et al., 2010). Also, other $4n$ bit Dynamic Range (DR) moduli sets such as $\{2^n, 2^n + 1, 2^n - 1, 2^{n+1} + 1\}$ (Mohan, 2007), $\{2^n, 2^n + 1, 2^n - 1, 2^{n+1} - 1\}$ (Mohan, 2007), (Vinod and Premkumar, 2000), $\{2^n - 1, 2^n, 2^n + 1, 2^{n-1} - 1\}$ (Cao et al., 2005), and $\{2^n - 3, 2^n - 1, 2^n + 1, 2^n + 3\}$ (Sheu et al., 2004), have been



presented to increase the parallelism property in RNS arithmetic unit. Additionally, other researchers utilized 4 and 3 moduli sets respectively to provide larger DR than $4n$ bit. Examples of $5n$ bit DR moduli sets proposed are $\{2^n - 1, 2^n, 2^n + 1, 2^{2n} + 1\}$ (Cao et al., 2003) and $\{2^{2n+1} - 1, 2^{2n}, 2^n - 1\}$ (Gbolagade et al., 2009), $\{2^n, 2^{2n} - 1, 2^{2n} + 1\}$ (Hariri et al., 2008). Subsequently, other $5n$ bit DR moduli sets such as $\{2^n, 2^n - 1, 2^n + 1, 2^n - 2^{(n+1)/2} + 1, 2^n + 2^{(n+1)/2} + 1\}$ (Hiasat, 2003), and $\{2^n, 2^n - 1, 2^n + 1, 2^{n+1} - 1, 2^{n-1} - 1\}$ (Cao et al., 2007) have also been presented with their associated reverse converter algorithms.

Research on $6n$ bit DR moduli sets has become very competitive and useful. Recently, Molahosseini et al. (2010) suggested the $5n$ bit DR moduli set

$\{2^n - 1, 2^n, 2^n + 1, 2^{2n+1} - 1\}$ with its accompanying reverse converter based on the New CRT II. Later, Molahosseini and Navi (2010) presented another moduli set $\{2^n, 2^n + 1, 2^{2n}, 2^{2n+1} - 1\}$ which is derived from $\{2^n - 1, 2^n, 2^n + 1, 2^{2n+1} - 1\}$ by enhancing the modulus 2^n to 2^{2n} . From the above, it can be realised that researchers have aimed at achieving larger DR moduli sets which can lead to efficient internal RNS arithmetic circuits as well as high performance residue to binary converters.

We propose the new moduli set $\{2^{2n}, 2^{2n} + 1, 2^{2n} - 1\}$ by enhancing the modulus 2^n to 2^{2n} in $\{2^n, 2^{2n} + 1, 2^{2n} - 1\}$, (Hariri et al., 2008). After demonstrating that the proposed moduli set results in legitimate RNS, we present a novel and efficient reverse converter based on the Chinese Remainder Theorem (CRT).

The rest of the chapter is structured as follows. Section 7.1 provides a brief background information on reverse conversion. In Section 7.2, the novel moduli set

is introduced, and the corresponding reverse conversion algorithm is presented. The hardware implementation of the proposed algorithm is described in Section 7.3, and Section 7.4 evaluates the performance of the proposed scheme. Finally, the chapter is concluded in Section 7.5.

8.1 Background

The main methods for reverse conversion are based on the Chinese Remainder Theorem (CRT), New CRT and MRC techniques. This chapter utilizes the CRT. Given a moduli set $\{m_i\}_{i=1,3}$, the residues (x_1, x_2, x_3) can be converted into the corresponding decimal number X using the CRT as follows (Szabo and Tanaka, 1967):

$$X = \left| \sum_{i=1}^k m_i \left| M_i^{-1} \right|_{m_i} x_i \right|_M \quad (8.1)$$

where $M = \prod_{i=1}^k m_i$, $M_i = \frac{M}{m_i}$ and M_i^{-1} is the multiplicative inverse of M_i with respect to (w.r.t) m_i .

The complexity of Equation (8.1) is significantly reduced by using the proposed moduli set $\{2^{2n}, 2^{2n} + 1, 2^{2n} - 1\}$. As a result, we present an algorithm that computes the equivalent decimal number of the RNS number (x_1, x_2, \dots, x_n) . First of all, we demonstrate that the computation of the respective required multiplicative inverses can be eliminated. This makes our design a memoryless converter.



8.2 New Moduli Set with Reverse Converter

For a given RNS moduli set to be legitimate, it is required that all the elements in the set to be co-prime. Thus, in order to prove that the proposed set can be utilized for the construction of valid RNS architecture, we have to demonstrate that the moduli 2^{2n} , $2^{2n} + 1$, and $2^{2n} - 1$ are pair-wise relatively prime.

Theorem 8.1. *The moduli 2^{2n} , $2^{2n} + 1$, and $2^{2n} - 1$ are pair-wise relatively prime numbers.*

Proof. :

Since $2^{2n} + 1$, and $2^{2n} - 1$ have been already shown to be relatively prime by Hariri et al. (2008), we only need to demonstrate that 2^{2n} is coprime to $2^{2n} + 1$, and $2^{2n} - 1$.

From the Euclidean theorem, we have $\gcd(a, b) = \gcd(b, |a|_b)$. Therefore,

$\gcd(2^{2n}, 2^{2n} + 1) = \gcd(2^{2n} + 1, |2^{2n}|_{2^{2n}+1}) = \gcd(2^{2n}, -1) = 1$. Similarly, $\gcd(2^{2n}, 2^{2n} - 1) = \gcd(2^{2n} - 1, |2^{2n}|_{2^{2n}-1}) = \gcd(2^{2n}, 1) = 1$. The numbers 2^{2n} , $2^{2n} + 1$, and $2^{2n} - 1$ are pairwise relatively prime because all the greatest common divisors are 1. \square

Theorem 8.2. *Given that the moduli set $\{m_1, m_2, m_3\}$ with $m_1 = 2^{2n}$, $m_2 = 2^{2n} + 1$, and $m_3 = 2^{2n} - 1$ the following holds true:*

$$|(m_2 m_3)^{-1}|_{m_1} = -1 \quad (8.2)$$



$$|(m_1 m_3)^{-1}|_{m_2} = -2^{2n-1} \quad (8.3)$$

$$|(m_1 m_2)^{-1}|_{m_3} = 2^{2n-1} \quad (8.4)$$

Proof. :

If it can demonstrated that $|(2^n + 1) \times (2^n - 1) \times -1|_{2^{2n}} = 1$, then -1 is the multiplicative inverse of $(2^n + 1) \times (2^n - 1)$ with respect to 2^{2n} : $|(2^{4n-1}) \times -1|_{2^{2n}} = 1$.

Similarly, $|(2^{2n}) \times (2^{2n} - 1) \times (-2^{2n-1})|_{2^{2n+1}} = |(2) \cdot (2^{4n-1})|_{2^{2n+1}} = 1$. Again,

$$|(2^{2n}) \times (2^{2n} + 1) \times (2^{2n-1})|_{2^{2n-1}} = |(2) \cdot (2^{4n-1})|_{2^{2n-1}} = 1.$$

□

The relation below is utilized in the subsequent theorem: Given the moduli set $\{m_1, m_2, m_3\}$ with $m_1 = 2^{2n}$, $m_2 = 2^{2n} + 1$, and $m_3 = 2^{2n} - 1$, the relation below holds true:

$$m_2 m_3 = m_1^2 - 1 \quad (8.5)$$

Theorem 8.3. *Based on the moduli set $\{2^{2n}, 2^{2n} + 1, 2^{2n} - 1\}$, the RNS number (x_1, x_2, x_3) can be converted into its equivalent weighted number X by:*

$$X = m_1 \left\lfloor \frac{X}{m_1} \right\rfloor + x_1 \quad (8.6)$$



where,

$$\left\lfloor \frac{X}{m_1} \right\rfloor = \left\lfloor -m_1x_1 - 2^{2n-1}m_3x_2 + 2^{2n-1}m_2x_3 \right\rfloor_{m_2m_3} \quad (8.7)$$

Proof. :

From Equation (8.1), for $k = 3$, we obtain:

$$X = \left\lfloor M_1 \left\lfloor M_1^{-1} \right\rfloor_{m_1} x_1 + M_2 \left\lfloor M_2^{-1} \right\rfloor_{m_2} x_2 + M_3 \left\lfloor M_3^{-1} \right\rfloor_{m_3} x_3 \right\rfloor_{m_1m_2m_3} \quad (8.8)$$

Using Equations (8.2), (8.3), and (8.4) in the above equation, we have:

$$X = \left\lfloor -m_2m_3x_1 + m_1m_3(-2^{2n-1})x_2 + m_1m_2(2^{2n-1})x_3 \right\rfloor_{m_1m_2m_3} \quad (8.9)$$

Substituting Equation (8.5) into (8.9) we have:

$$X = \left\lfloor (m_1^2 - 1)(-x_1) + m_1m_3(-2^{2n-1})x_2 + m_1m_2(2^{2n-1})x_3 \right\rfloor_{m_1m_2m_3} \quad (8.10)$$

Dividing both sides of Equation (8.10) by m_1 and computing the floor value of both sides, we obtain:

$$\left\lfloor \frac{X}{m_1} \right\rfloor = \left\lfloor -m_1x_1 - 2^{2n-1}m_3x_2 + 2^{2n-1}m_2x_3 \right\rfloor_{m_2m_3} \quad (8.11)$$



Equation (8.11) can therefore be directly rewritten as:

$$\left\lfloor \frac{X}{2^{2n}} \right\rfloor = \left| -2^{2n}x_1 - (2^{2n-1})(2^{2n} - 1)x_2 + 2^{2n-1}(2^{2n} + 1)x_3 \right|_{m_2m_3} \quad (8.12)$$

Following the basic integer division definition in RNS, we finally have:

$$X = m_1 \left\lfloor \frac{X}{m_1} \right\rfloor + x_1 \quad (8.13)$$

$$= 2^{2n} \left\lfloor \frac{X}{2^{2n}} \right\rfloor + x_1 \quad (8.14)$$

□

In order to reduce the hardware complexity, we use the following properties to simplify Equation (8.12):

Property 1: The multiplication of a residue number by 2^k in modulo $(2^p - 1)$ is computed by k bit circular left shifting

Property 2: A negative number in modulo $(2^p - 1)$ is calculated by subtracting the number in question from $(2^p - 1)$. In binary representation, the ones complement of the number gives the result.

Let the residues (x_1, x_2, x_3) have binary representation as follows:



$$x_1 = \underbrace{(x_{1,2n-1}x_{1,2n-2}\dots x_{1,1}x_{1,0})}_{2n} \quad (8.15)$$

$$x_2 = \underbrace{(x_{2,2n}x_{2,2n-1}\dots x_{2,1}x_{2,0})}_{2n+1} \quad (8.16)$$

$$x_3 = \underbrace{(x_{3,2n-1}x_{3,2n-2}\dots x_{3,1}x_{3,0})}_{2n} \quad (8.17)$$

Equation (8.12) can be directly rewritten as:

$$\left\lfloor \frac{X}{2^{2n}} \right\rfloor = |u_1 + u'_2 + u''_2 + u'''_3|_{2^{4n-1}} \quad (8.18)$$

where,

$$u_1 = |-2^{2n}x_1|_{2^{4n-1}} = \underbrace{\bar{x}_{1,2n-1}\bar{x}_{1,2n-2}\dots\bar{x}_{1,0}}_{2n} \underbrace{11\dots 11}_{2n} \quad (8.19)$$

$$u_2 = |-2^{4n-1}x_2 + 2^{2n-1}x_2|_{2^{4n-1}} \quad (8.20)$$

$$u'_2 = |-2^{4n-1}x_2|_{2^{4n-1}} = \bar{x}_{2,0} \underbrace{11\dots 11}_{2n-1} \underbrace{\bar{x}_{2,2n}\bar{x}_{2,2n-1}\dots\bar{x}_{2,1}}_{2n} \quad (8.21)$$

$$u''_2 = |2^{2n-1}x_2|_{2^{4n-1}} = \underbrace{x_{2,2n}x_{2,2n-1}\dots x_{2,0}}_{2n+1} \underbrace{00\dots 00}_{2n-1} \quad (8.22)$$

$$u_3 = |2^{4n-1}x_3 + 2^{2n-1}x_3|_{2^{4n-1}} \quad (8.23)$$



$$u'_3 = |2^{4n-1}x_3|_{2^{4n}-1} = x_{3,0} \underbrace{00\dots 00}_{2n} \underbrace{x_{3,2n-1}\dots x_{3,1}}_{2n-1} \quad (8.24)$$

$$u''_3 = |2^{2n-1}x_3|_{2^{4n}-1} = 0 \underbrace{x_{3,2n-1}\dots x_{3,0}}_{2n} \underbrace{00\dots 00}_{2n-1} \quad (8.25)$$

By considering Equation (8.24) and (8.25), it is clear that they can be manipulated to obtain u'''_3 represented as:

$$u'''_3 = x_{3,0} \underbrace{x_{3,2n-1}\dots x_{3,0}}_{2n} \underbrace{x_{3,2n-1}\dots x_{3,1}}_{2n-1} \quad (8.26)$$

8.3 Hardware Realization

The hardware structure of the proposed reverse converter for the moduli set $\{2^{2n}, 2^{2n} + 1, 2^{2n} - 1\}$ is based on Equation (8.18) and (8.14). Figure 8.1 and 8.2 represents the block diagram of the proposed Cost Efficient (CE) and Speed Efficient (SE) converter respectively. In Equation (8.14), the parameters u_1, u'_2, u''_2 , and u'''_3 are added using two $4n$ bit Carry Save Adders (CSA) with end around carry to produce s_2 and c_2 . Next, these must be added modulo $2^{4n} - 1$ in order to obtain $\left\lfloor \frac{X}{m_1} \right\rfloor$ to achieve a CE converter. The final result, computed from Equation (8.14) is therefore obtained simply by a shift and a concatenation operation, which requires no additional hardware resources. To achieve a SE converter, we speed up the addition process by utilizing anticipated computation. By this, we compute $s_2 + c_2$ for both



$cin = 0$ and $cin = 1$ and then the right result is selected with a MUX. Again, the implementation of Equation (8.14) to obtain the final result requires no additional hardware.

It is interesting to note that, some of the Full Adders (FAs) are reduced to Half Adders (HAs) because some of the inputs of the CSA have constant values of 0's and 1's. It can be observed that our CE proposal utilizes $10nFA$ and $2nHA$ and a conversion time of $(8n + 2)t_{FA}$, while the SE converter requires $10nFA$, $2nHA$ and t_{MUX} , with a delay of $(4n + 2)t_{FA}$.

8.4 Performance Analysis

The performance of the proposed reverse converter is evaluated in terms of hardware cost and conversion time. In order to properly evaluate the performance of our proposal against the state of the art, both theoretical and experimental analysis were performed.

8.4.1 Theoretical Evaluation

We compared our converter with the state of the art converters presented by Cao et al. (2003) and Molahosseini and Navi (2010) with $5n$ and $6n$ bit Dynamic Range



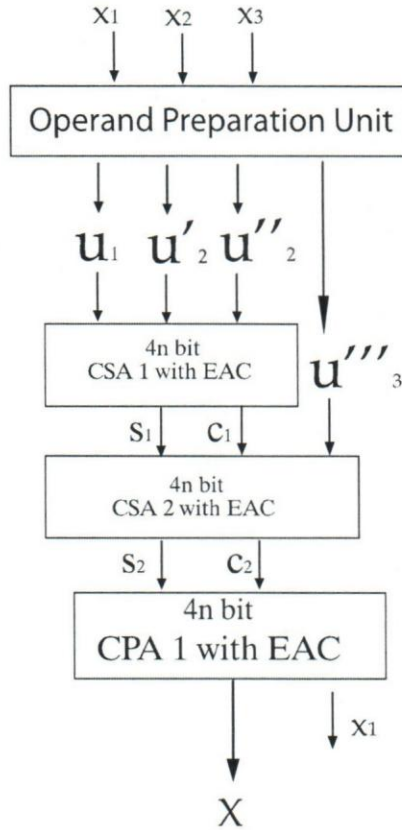


Figure 8.1: Proposed Cost Efficient Reverse Converter

(DR) respectively. The hardware utilization is derived in terms of FA and HA. The inclusion of Cao et al. (2003) in the comparison is to demonstrate that, our converter can compete favourably with other existing state of the art. The theoretical analysis is presented in Table 8.1. From the table, it is seen clearly that our proposal outperforms the state of the art in terms of area and delay. For our CE converter, the delay is $(8n+2)t_{FA}$ while that of Molahosseini and Navi (2010) exhibits a delay of $(12n+6)t_{FA}$. To further simplify the area comparison, we assumed that one FA is twice as large

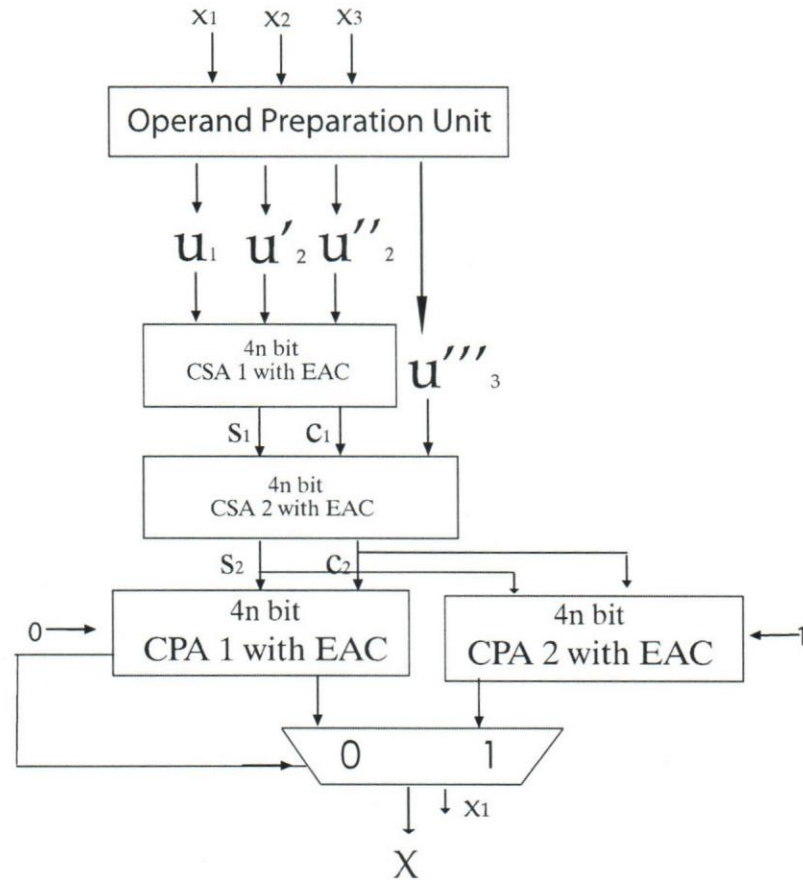


Figure 8.2: Proposed Speed Efficient Reverse Converter

as an HA, and then expressed the area cost for all the considered designs in terms of HA. It is therefore evident that our converter utilizes lesser area resources.

8.4.2 Experimental Evaluation

For the experimental assessment, the converters were described in VHDL and then implemented on Spartan 6 xc6slx45t-3fpg484 FPGA, with Xilinx ISE 14.3 for various dynamic range requirements. The performance is evaluated in terms of area measured

Table 8.1: Theoretical Area and Delay Comparison

Converter	Cao et al. (2003)	Molahosseini and Navi (2010)	Proposed
DR	$5n$	$6n$	$6n$
FA	$11n + 4$	$12n + 2$	$10n$
HA	$9n - 4$	$4n + 1$	$2n$
Area Cost in HA (Δ)	$31n + 4$	$28n + 5$	$22n$
Delay (τ)	$(8n + 3)t_{FA}$	$(12n + 6)t_{FA}$	$(8n + 2)t_{FA}$

according the number of slices and delay corresponding to the critical path in nanoseconds. Table 8.2 shows the synthesized results for the various values of n . To confirm the theoretical results, the experimental results clearly shows the superiority of our converter over the state of the arts. In comparison with the reverse converter presented by Molahosseini and Navi (2010), the generated values strongly suggest that, on the average, our proposal is capable of performing 52.35% faster than the converter proposed by Molahosseini and Navi (2010). Also, in terms of area cost our converter exhibits a 43.94% reduction with respect to state of the art. Figures 8.3 and 8.4 presents the performance of our proposal against the state of the art in terms of area and delay



Table 8.2: Experimental Delay and Area Comparison for $\{2^{2n}, 2^{2n} + 1, 2^{2n} - 1\}$

n	Cao et al. (2003)		Molahosseini and Navi (2010)		Proposed	
	Delay	Area	Delay	Area	Delay	Area
2	18.202	54	26.650	72	15.541	36
4	22.020	107	37.011	141	17.189	71
6	26.074	157	43.637	209	20.870	113
8	24.507	210	42.755	275	22.909	152
10	26.983	270	51.522	351	24.396	197
12	26.647	327	51.892	426	24.508	238
14	27.640	386	56.011	498	25.749	286
16	28.392	440	57.778	563	26.286	324
18	29.724	489	61.110	635	27.422	358
20	29.971	546	59.663	713	27.656	402

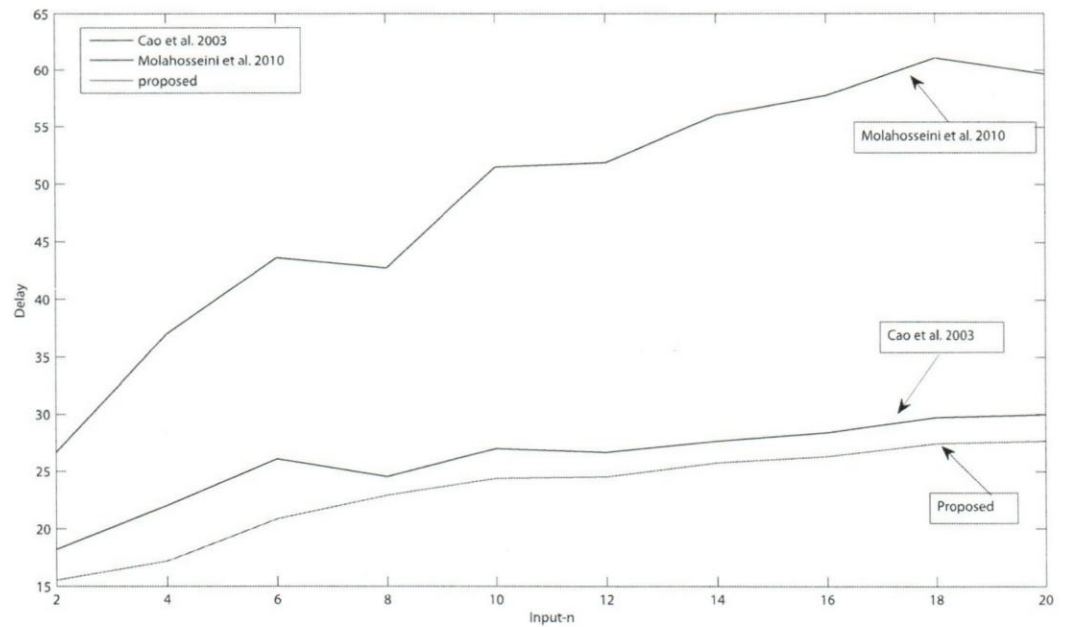


Figure 8.3: The Delay Comparison of Converters



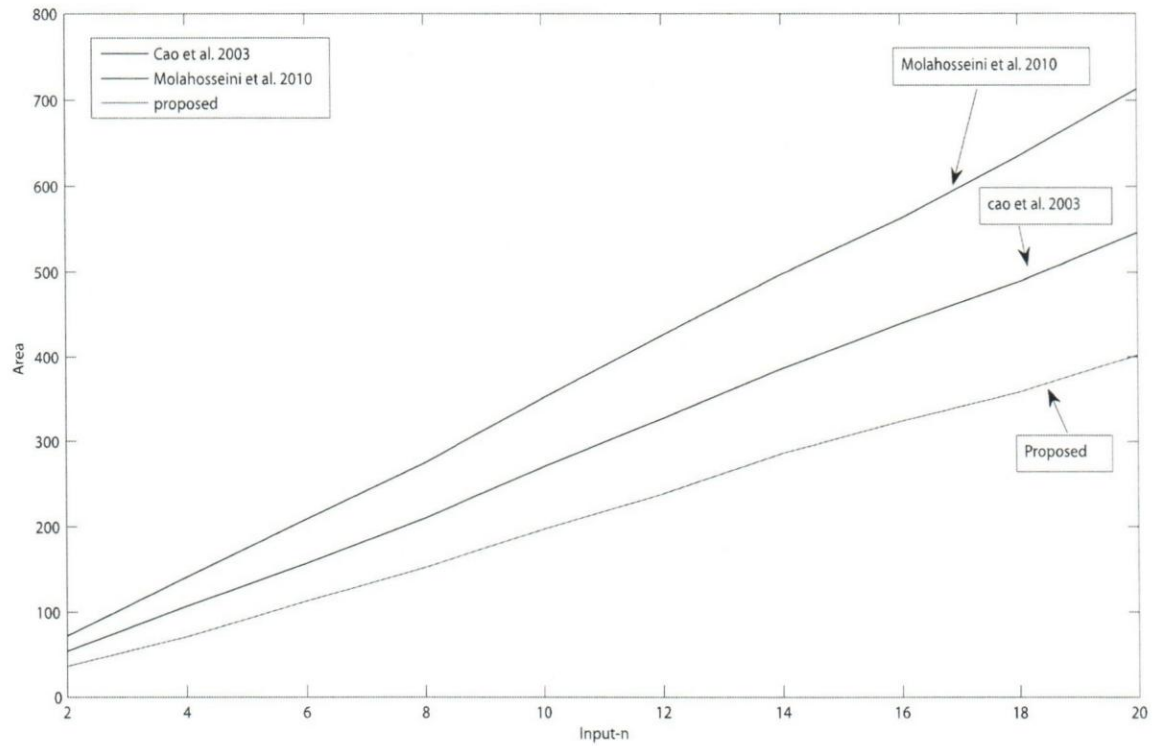


Figure 8.4: The Area Comparison of Converters

8.5 Conclusion

In this chapter, we proposed a novel moduli set $\{2^{2n}, 2^{2n} + 1, 2^{2n} - 1\}$ with its associated reverse converter using the CRT. The moduli set is a $6n$ bit DR and therefore appropriate for applications requiring specifically $6n$ DR. We simplified the CRT to obtain an effective algorithm. Further, we reduced the resulting architecture in order to obtain a reverse converter that utilizes only two CSAs and a CPA. We performed both theoretical and experimental evaluation of our proposal. The theoretical analysis showed clearly the advantages of our moduli set and its associated



reverse converter. This was confirmed by the experimental results. We described our scheme and those presented by Cao et al. (2003) and Molahosseini and Navi (2010) in VHDL and carried out the implementation on an FPGA using a wide range of values on n . The synthesis results is summarized in Table 8.2. The results indicate that on the average, our scheme is 52.35% faster than the converter proposed by Molahosseini and Navi (2010) in terms of speed, while it exhibits a 43.94% reduction in area cost. Clearly, the results show that, our proposal outperforms the best known state of the art.

In the this chapter, we conclude the presentation of our contributions. In the chapter that follows, we present the overall conclusion of our research investigations. Additionally, we give a summary of our major contributions and then recommend some future research directions which are worth investigating.





CHAPTER 9

SUMMARY AND CONCLUSIONS

This thesis contributes to the ambitious long term vision of Residue Number System (RNS) general processor realization. We focused on designing efficient residue to binary converters for some powers of two moduli sets. We utilized either the Chinese Remainder Theorem (CRT), New CRTs, or the Mixed Radix Conversion. Residue to Binary conversion is a very important issue in RNS based processor realization because, many of the challenges of RNS rely heavily on efficient reverse converters. In as much as efficiency continues to be achieved with respect to reverse conversion, then the more closer we get to in finding solution to the many bottlenecks facing RNS wide spread usage. We proposed efficient residue to binary processors that result in high speed and cost effective hardware resources.

The rest of this chapter is organized as follows: Section 9.1 presents a summary of the work and results generated in this thesis. Section 9.2, highlight the main novel contributions of this thesis. Future directions and some open issues worthy of further investigation are presented in Section 9.3

9.1 Summary

In this thesis, several contributions have been presented in terms of moduli sets proposition with their associated reverse converters and some improvement on existing state of the art converters. We used either the CRT, New CRT or MRC appropriately. The summary of our work is as follows:

1. In Chapter 2, we presented a modified CRT based reverse converter for the popular moduli set $\{2^n, 2^n - 1, 2^n + 1\}$. The proposed converter is memoryless and adder based. We performed both theoretical and FPGA experimentation of our scheme and the state of the art. From the theoretical assessment, we deduced that our converter is advantageous in terms of speed while requiring more area resources. Then, we implemented our scheme with the state of the art on Xilinx Spartan 6 FPGA. The results demonstrated vividly that, on the average, and contrary to the theoretical result, our proposed reverse converter significantly outperforms the state of the art in terms of both speed and hardware resources. On the average, our proposed converter is 9.16% faster than the state of the art, while in terms of hardware resources, our scheme saves about 24.05% area when compared with the one presented by Wang et al. (2002).
2. In Chapter 3, we hybridized the converter presented in Chapter 1 with the MRC to obtain an effective converter for the moduli set $\{2^n, 2^n - 1, 2^n + 1, 2^{2n+1} - 1\}$. The proposed scheme is purely adder based and memoryless and was implemented in a two level design. The first level consist of deriving the equivalent



weighted number of the residue (x_1, x_2, x_3) using the modified CRT algorithm for the popular moduli set $\{2^n, 2^n - 1, 2^n + 1\}$ presented in Chapter 1. Next, the resulting equivalent weighted number from the first level was hybridized with the fourth residue x_4 using the MRC with respect to the composite moduli set $\{2^{3n} - 2^n, 2^{2n+1} - 1\}$. We further simplified the resulting architecture to obtain an adder based and memoryless reverse converter that utilizes only Carry Save Adders (CSAs) and Carry Propagate Adders (CPAs). We compared our scheme with existing state of the art converter proposed by Molahosseini et al. (2010) both theoretically and experimentally. The theoretical assessment indicates that our converter is better than the state of the art in terms of both area and delay. The FPGA experimentation results contradicts the theoretical assessment with respect to only the area resources requirement. Our proposed converter achieves on the average 9.80% speed up, while requiring about 4.14% area resources. For a more accurate comparison, we further used the Area Time square metric, which shows that our proposed scheme outperforms the state of the art by 19.34%.

3. In Chapter 4, we proposed a new $5n$ bit DR moduli set $\{2^{2n} - 1, 2^n, 2^{2n+1} - 1\}$ derived from the four-moduli set $\{2^n, 2^n - 1, 2^n + 1, 2^{2n+1} - 1\}$ presented by Molahosseini et al. (2010) by combining $(2^n - 1)$ and $(2^n + 1)$ into $(2^{2n} - 1)$. Next, we proposed an efficient MRC based RNS-to-binary converter for the new moduli set. When compared to the converter proposed by Molahosseini



et al. (2010), theoretically speaking, our converter outperforms the state of the art in terms of both hardware requirements and delay. Also, we performed an FPGA experimentation of our converter and that of $\{2^n, 2^n - 1, 2^n + 1, 2^{2n+1} - 1\}$ by Molahosseini et al. (2010) by describing the converters in VHDL using Xilinx Spartan 6 on ISE 14.3. The experimental results suggest that, on average, our proposed architecture is 18.56% faster and saves about 17.36% hardware resources.

4. In Chapter 5, we proposed an efficient RNS to binary converter for the moduli set $\{2^n, 2^{2n+1} - 1, 2^{2n+2} - 1\}$ which contains low-cost moduli and has a larger dynamic range compared to other existing $(5n)$ bit DR state of the art. The proposed reverse converter for the moduli set $\{2^n, 2^{2n+1} - 1, 2^{2n+2} - 1\}$ is based on MRC. Additionally, we further simplified the resulting architecture in order to obtain a reverse converter that utilizes only 1 level of CSA together with two CPAs i.e., CPA1, CPA2 or CPA3, and CPA4. The proposed converter is purely adder based and memoryless. The performance of the proposed converter is evaluated both theoretically and experimentally by FPGA implementation. On average, we found out that, our proposal outperforms the state of the art with approximately 35.98% and 6.76% in terms of area cost and conversion time respectively.

5. In Chapter 6, we proposed a new moduli set $\{2^{2n}, 2^{2n+1} - 1, 2^{2n} - 1\}$ which is an enhancement of the moduli set $\{2^n, 2^{2n+1} - 1, 2^{2n} - 1\}$ and its associated





efficient RNS to binary converter. The proposed reverse converter is based on MRC. The divide and conquer approach was used to implement the MRC, where the moduli set is grouped into two. The first phase combined the first and the second residue with respect to the subset $\{2^{2n}, 2^{2n+1} - 1\}$. In the second phase, the result of the first phase is combined with the third residue with respect to the composite moduli set $\{(2^{2n})(2^{2n+1} - 1), 2^{2n} - 1\}$. Additionally, we further simplified the resulting architecture in order to obtain a reverse converter that utilizes only 2 levels of CSAs together with three CPAs. Theoretical analysis reveals that our proposal has a delay of $(10n+4)t_{FA} + 2t_{MUX}$ with an area cost of $(12n+2)FAs$ and $(2n)HAs$. Subsequently, we expressed the required hardware resources in terms of HA and computed the $\Delta\tau^2$ metric. Then for experimental comparison, we described our converter and those by Gbolagade et al. (2009) and Molahosseini et al. (2010) in VHDL and subsequently implemented them on an FPGA. A wide range of values of n were investigated through the implementation. The synthesis results generated clearly indicates that on the average our reverse converter is capable of performing 32.75% faster than the one proposed by Molahosseini et al. (2010). Also, in terms of area resources, our proposal exhibits 49.21% reduction when compared with same state of the art.

6. In Chapter 7, we introduced a new moduli set $\{2^{2n+1} - 1, 2^{2n+1}, 2^{2n} - 1\}$ which contains only low-cost moduli and has a large dynamic range. A novel and

effective reverse converter is proposed based on the New CRT. Additionally, we further simplified the resulting architecture in order to obtain a reverse converter that utilizes 4 levels of CSAs namely, CSA1, CSA2, CSA3, and CSA4 together with two CPAs i.e., CPA1 and CPA2. The proposed converter is purely adder based and memoryless. Both theoretical and experimental assessment of our proposal against the best state of the art were carried out. Theoretically speaking, our proposal has a delay of $(8n + 6)t_{FA} + 2t_{MUX}$ with an area cost of $(12n + 4)FA$ and $(6n - 1)HA$. We compared the proposed reverse converter with the state of the art converters $\{2^n - 1, 2^n, 2^n + 1, 2^{2n+1} - 1\}$ and $\{2^n - 1, 2^n + 1, 2^{2n}, 2^{2n+1} - 1\}$ proposed by Molahosseini et al. (2010) and Molahosseini and Navi (2010) respectively. The result therefore shows that our scheme is faster than the equivalent state of the art converter $\{2^n - 1, 2^n + 1, 2^{2n}, 2^{2n+1} - 1\}$, but requires some more area resources. Further, for the experimental comparison, we described our proposed reverse converter and the considered state of the art in VHDL and performed the implementation on an FPGA using a wide range of values on n . The synthesis results confirmed that of the theoretical assessment. In terms of speed, our converter is 37.34% faster than the converter presented by Molahosseini and Navi (2010) for $\{2^n - 1, 2^n + 1, 2^{2n}, 2^{2n+1} - 1\}$. While in terms of area, our proposal requires about 13.34% more area resources. In order to ascertain the efficiency of our scheme, we used the Area-Time Square ($\Delta\tau^2$) metric. The ($\Delta\tau^2$) metric

suggests that our converter is 57.96% more efficient than the one presented by Molahosseini and Navi (2010) for $\{2^n - 1, 2^n + 1, 2^{2n}, 2^{2n+1} - 1\}$.

7. In Chapter 8, we proposed a novel moduli set $\{2^{2n}, 2^{2n} + 1, 2^{2n} - 1\}$ with its corresponding reverse converter using the CRT. The moduli set is a $6n$ bit DR and therefore appropriate for applications requiring specifically $6n$ DR. We simplified the CRT to obtain an effective algorithm. Further, we reduced the resulting architecture in order to obtain a reverse converter that utilizes only two CSAs and a CPA. We performed both theoretical and experimental evaluation of our proposal. The theoretical analysis shows clearly the advantages of our moduli set and its associated reverse converter. This is confirmed by the experimental results. We described our scheme and those presented by Cao et al. (2003) and Molahosseini and Navi (2010) in VHDL and carried out the implementation on an FPGA using a wide range of values on n . The results indicate that on the average, our scheme is 52.35% faster than the converter proposed by Molahosseini and Navi (2010) in terms of speed, while it exhibits a 43.94% reduction in area cost. Clearly, the results shows that, our proposal outperforms the best known state of the art.

9.2 Major Contributions

In this thesis, we made a number of contributions summarized as follows:



1. We modified the traditional CRT for the popular moduli set $\{2^n, 2^n - 1, 2^n + 1\}$.
The proposed scheme significantly improved upon the existing state of the art converter in terms of area cost and conversion time
2. We hybridized the converter presented in Chapter 1 with the MRC to obtain an effective converter for the moduli set $\{2^n, 2^n - 1, 2^n + 1, 2^{2n+1} - 1\}$.
3. We proposed a new $5n$ bit DR moduli set $\{2^{2n} - 1, 2^n, 2^{2n+1} - 1\}$ derived from the four-moduli set $\{2^n, 2^n - 1, 2^n + 1, 2^{2n+1} - 1\}$ presented by Molahosseini et al. (2010) with its associated reverse converter.
4. We proposed an efficient RNS to binary converter for the moduli set $\{2^n, 2^{2n+1} - 1, 2^{2n+2} - 1\}$ which contains low-cost moduli and has a larger dynamic range compared to other existing $(5n)$ bit DR state of the art.
5. We proposed a new moduli set $\{2^{2n}, 2^{2n+1} - 1, 2^{2n} - 1\}$ which is an enhancement of the moduli set $\{2^n, 2^{2n+1} - 1, 2^{2n} - 1\}$ and its associated efficient RNS to binary converter.
6. A new moduli set $\{2^{2n+1} - 1, 2^{2n+1}, 2^{2n} - 1\}$ is proposed. The said moduli set contains only low-cost moduli and has a large dynamic range. A novel and effective reverse converter is proposed based on the New CRT.
7. We proposed a novel moduli set $\{2^{2n}, 2^{2n} + 1, 2^{2n} - 1\}$ with its corresponding reverse converter using the CRT.



9.3 Future Research Directions

This thesis focused on proposing efficient residue to binary converters based on either the CRT, New CRT and MRC. This happens to be one of the major steps to addressing the bottlenecks that hinder RNS realization in designing general purpose processors. Having provided some solution to the main challenges of RNS general purpose processor realization, we present in this section some future research directions that would further move some steps closer to the ultimate dream of building RNS based general purpose processors and then improve RNS wide spread usage in specific purpose processors.

1. Based on the efficient residue to binary converters proposed in this thesis, it will be interesting to re-design or completely design new algorithms for resolving some of the difficult RNS operations such as: signed detection, magnitude comparison, overflow detection, division and scaling as they heavily rely on reverse conversion.
2. The application of our proposed moduli sets and their respective reverse converters in image encryption will be an interesting line of research that may generate fascinating results.
3. Given that larger dynamic range moduli sets are of practical importance, it will be interesting to investigate which is more efficient with respect to either length 3, length 4 or higher length moduli sets with similar dynamic range.



4. The use of Signed Digit (SD) Number System in connection with RNS has been suggested as a method of eliminating completely the remaining carry propagation in RNS arithmetic. It will be interesting to exhaust the possibilities of combining RNS and SD in designing efficient converters.
5. Forward Conversion is another interesting area that requires extensive research attention.
6. Using RNS to reduce power consumption in digital systems is another research area of great importance that requires extensive research.
7. RNS has been used to resolve arithmetic problems which arise in finding solution to systems of equations with integer coefficients. This will be an interesting area to apply RNS.





References



References

Aiken, H. and Semon, W. (1959). Advanced digital computer logic. *Tech. Rep. WADC TR*, pages 49–472. 5

Al-radadi, E. and Siy, P. (2002). Four-moduli set $\{2, 2^n - 1, 2^n + 2^{n-1} - 1, 2^{n+1} + 2^n - 1\}$ simplifies the residue to binary converters based on crt ii. *PERGAMON Computers and Mathematics with Applications*, 44:1581–1587. 6, 13

Andraos, S. and Ahmad, F. (1988). A new efficient memoryless residue to binary converter. *IEEE Trans. on Circuits and Systems-II*, 35(11):1441–1444. 10, 20, 21, 33

Bankas, E. and Gbolagade, K. (2012). A speed efficient rns to binary converter for the moduli set $\{2^n, 2^n + 1, 2^n - 1\}$. *Journal of Computing*, 4, 5:83–88. 6, 9, 69, 104, 111, 124

Bankas, E., Gbolagade, K., and Cotofana, S. (2013). An effective new crt based converter for a novel moduli set $\{2^{2n+1} - 1, 2^{2n+1}, 2^{2n} - 1\}$. *24th IEEE International*

Conference on Application-specific Systems, Architectures and processors (ASAP 2013). Washington, USA, pages 142–146. 90, 93

Beckmann, P. and Musicus, B. (1993). Fast fault-tolerant digital convolution using a polynomial residue number system. *IEEE Transactions on Signal Processing*, 41(7):2300–2313. 3

Bhardwaj, M., Premkumar, A., and Srikanthan, T. (1998). Breaking the $2n$ -bit carry propagation barrier in residue to binary conversion for the $\{2^n + 1, 2^n, 2^n - 1\}$ moduli set. *IEEE Trans. on Circuits and Syst. II*, 45:998–1002. 33

Bhardwaj, M., Srikanthan, T., and Clarke, C. (1999). A reverse converter for the 4-moduli superset $\{2^n - 1, 2^n, 2^n + 1, 2^{n+1} + 1\}$. *IEEE Symp. Computer Arithmetic*, pages 168–175. 13

Bi, S., Wang, W., and Al-Khalili, A. (2004). Modulo deflation in $\{2^n + 1, 2^n, 2^n - 1\}$. *Proc. IEEE International Symposium on Circuits and Systems (ISCAS'04)*, 2:429–432. 10, 20, 21, 107

Cao, B., Chang, C., and Srikanthan, T. (2003). An efficient reverse converter for the 4-moduli set $\{2^n - 1, 2^n, 2^n + 1, 2^{2n} + 1\}$ based on the new chinese remainder theorem. *IEEE Trans. on Circuits and Systems-I: Fundamental Theory and Applications*, 50(10):1296–1303. 11, 18, 37, 69, 125, 133, 134, 136, 137, 139, 146



- Cao, B., Chang, C., and Srikanthan, T. (2007). A residue-to-binary converter for a new five-moduli set. *IEEE Trans. on Circuits and Systems-I: Regular Papers*, 54(5):1041–1049. 11, 125
- Cao, B., Srikanthan, T., and Chang, C. (2005). Efficient reverse converters for two four-moduli sets $\{2^n - 1, 2^n, 2^n + 1, 2^{n+1} - 1\}$ and $\{2^n - 1, 2^n, 2^n + 1, 2^{n-1} - 1\}$. *IEE Proc.-Comput. Digit. Tech.*, 152(5):687–696. 37, 88, 124
- Cardarilli, G., Er, M., and Lojacona, R. (1998). Rns-to-binary conversion for efficient vlsi implementation. *IEEE Transactions on Circuits and Systems II*, 45(6):667–669. 22
- Chalivendra, G., Hanumaiah, V., and Vrudhula, S. (2011). A new balanced 4-modulul set $\{2^k, 2^n - 1, 2^n + 1, 2^{n+1} - 1\}$ and its reverse converter design for efficient fir filter implementation. *GLSVLSI'11*, pages 139–144. 3, 6, 13
- Chen, J., Jenkins, W., Hein, I., and O'Brien Jr, W. (1990). Design and implementation of high speed residue number system correlator for ultrasonic time domain blood flow measurement. *IEEE*, (2):2893–2896. 3
- Conway, R. and Nelson, J. (2003). New crt-based rns converter using restricted moduli set. *IEEE Trans. on Computers*, 52(5):572–578. 10
- Conway, R. and Nelson, J. (2004). Improved rns fir filter architectures. *IEEE Trans. on Circuits and Systems-II: Express briefs*, 51(1):26–28. 3



Dhurkadas, A. (1990). An efficient residue to binary converter design. *IEEE Transactionson Circuits and Systems.*, 37:849–850. 22

Efstathiou, C., Nikolos, D., and Kalamatianos (1994). Area-time efficient modulo $2^k - 1$ adder design. *IEEE Transactions on Circuits and Systems II.*, 41:463–466.

54

Fernandez, P., Garcia, A., Ramirez, J., and lloris, A. (2000). Fast rns-based 2d-
dct computation on field -programmable devices. *Proceedings of the IEEE Signal
Processing Systems Workshop, LA, USA*, (2):365–373. 3

Gallaher, D., Petry, F., and Srinivasan, P. (1997). The digital parallel method for fast
rns to weighted number system conversion for specific moduli $\{2^k - 1, 2^k, 2^k + 1\}$.
IEEE Trans. on Circuits and Systems-II., 44(1):53–57. 10

Gbolagade, K. (2010). *Effective Reverse Conversion in Residue Number System
Processors*. PhD Thesis, Delft University of Technology, Delft, The Netherlands,
ISBN 978-90-72298-10-2. 4, 5, 6, 10

Gbolagade, K., Chaves, R., Sousa, L., and Cotofana, S. (2009). Residue-to-binary
converters for the $\{2^{2n+1} - 1, 2^{2n}, 2^n - 1\}$ moduli set. *2nd IEEE International
Conference on Adaptive Science and Technology, Accra Ghana*, pages 26–33. 6,
9, 12, 17, 43, 58, 70, 80, 83, 87, 88, 90, 97, 99, 102, 105, 125, 144

Gbolagade, K., Chaves, R., Sousa, L., and Cotofana, S. (2010a). An improved rns
reverse converter for the $\{2^{2n+1} - 1, 2^n, 2^n - 1\}$ moduli set . *IEEE International*



Symposium on Circuits and Systems (ISCAS 2010), Paris, France, pages 2103–2106. 64, 65

Gbolagade, K. and Cotofana, S. (2008a). An efficient rns to binary converter using the moduli set $\{2n + 1, 2n, 2n - 1\}$. *XXIII Conference on Design of Circuits and Integrated Systems (DCIS 2008), Grenoble, France*. 6

Gbolagade, K. and Cotofana, S. (2008b). Generalized matrix method for efficient residue to decimal conversion. *Proceedings of 10th IEEE Asia Pacific Conference on Circuits and Systems (APCCAS 2008), Macao, China*, pages 1414– 1417. 6

Gbolagade, K. and Cotofana, S. (2008c). Mrc technique for rns to decimal conversion for the moduli set $\{2n + 2, 2n + 1, 2n\}$. *Proceedings of 16th Annual Workshop on Circuits, Systems, and Signal Processing, Veldhoven, The Netherlands*, pages 318–321. 6

Gbolagade, K. and Cotofana, S. (2008d). Residue number system operands to decimal conversion for 3- moduli sets. *Proceedings of 51st IEEE Midwest Symposium on Circuits and Systems, Knoxville, USA*, pages 791– 794. 6

Gbolagade, K. and Cotofana, S. (2008e). A residue to binary converter for the $\{2n + 2, 2n + 1, 2n\}$ moduli set. *Proceedings of 42nd Asilomar Conference on Signals, Systems, and Computers, California, USA.*, pages 1785– 1789. 6



Gbolagade, K. and Cotofana, S. (2009a). An $O(n)$ residue number system to mixed radix conversion. *Proceedings of the 2009 IEEE International Symposium on Circuits and Systems (ISCAS 2009), Taiwan, China.*, pages 521– 524. 6

Gbolagade, K. and Cotofana, S. (2009b). A reverse converter for the new 4- moduli sets $\{2n + 3, 2n + 2, 2n + 1, 2n\}$. *Proceedings of the 2009 IEEE International Conference on Electronics, Circuits and Systems (ICECS09), Hammamet, Tunisia.*, pages 113– 116. 6

Gbolagade, K., Voicu, G., and Cotofana, S. (2010b). Memoryless rns to binary converters for $\{2^{n+1} - 1, 2^n, 2^n - 1\}$ moduli set. *21st IEEE International Conference on Application Specific Systems, Architectures, and Processors (ASAP 2010), Rennes, France.*, pages 301–304. 20, 26

Gbolagade, K., Voicu, G., and Cotofana, S. (2011). An efficient fpga design of residue-to binary converter for the moduli sets $\{2n + 1, 2n, 2n - 1\}$. *IEEE Transactions on Very large Scale Integration (VLSI) Systems.*, 19(8):1500–1503. 6

Gbolagade, K. A. (2009a). Moduli selection guidelines for efficient residue-to-decimal conversion. *Far East Journal of Electronics and Communication. Pushpa Publishing House*, 3(1):53–67. 9

Gbolagade, K. A. (2009b). A shorter algorithm for efficient residue to decimal conversion. *Advances in Computer Science and Engineering. Pushpa Publishing House*, 3(2):147–156. 6



- Hariri, A., Navi, K., and Rastegar, R. (2008). A new high dynamic range moduli set with efficient reverse converter. *Computers and Mathematics with Applications*, 55:660–668. 11, 12, 70, 125, 127
- Hiasat, A. (2003). Vlsi implementation of new arithmetic residue to binary decoders. *IEEE Trans. on Very large Scale Integration (VLSI) Systems*, 13(1):153–158. 12, 125
- Hiasat, A. and Abdel-AtyZohdy, H. (1998). Residue-to-binary arithmetic converters for the moduli set $\{2^k, 2^k - 1, 2^{k-1} - 1\}$. *IEEE Transactions on Circuits and Systems-II. Analog and Digital Signal Processing*, 45(2):204–209. 69, 87, 104
- Hosseinzadeh, M., Molahosseini, A., and Navi, K. (2009). A parallel implementation of the reverse converter for the moduli set $\{2^n, 2^n - 1, 2^{n-1} - 1\}$. *World Academy of Sciences, Engineering and Technology*, 55:494–498. 20, 69, 87, 104
- Huang, C., Gelders, A., and Peterson, D. (1985). Residue processing. *Technical report, Lockheed Missiles and Space Company, Inc.* 5
- Ibrahim, K. and Saloum, S. (1988). An efficient residue to binary converter design. *IEEE Trans. on Circuits and Systems.*, 35(7):1156–1158. 32, 33
- Jenkins, W. (1978). Techniques for residue-to-analog conversion for residue encoded digital filters. *IEEE Trans. on Circuits and Systems*, CAS-25:555–562. 3



- Jenkins, W. and Leon, B. (1977). The use of residue number system in the design of finite impulse response digital filters. *IEEE Trans. on Circuits and Systems*, 24:191–200. 3, 5
- Lin, S., Sheu, M., and Wang, C. (2008). Efficient vlsi design of residue-to-binary converter for the moduli set $\{2^n, 2^{n+1} - 1, 2^n - 1\}$. *IEICE Trans. Info and Systems*, E91-D(7):2058–2060. 6, 20, 124
- Modiri, S., Movaghar, A., and Barati, A. (2012). An efficient reverse converter for the new modulus set $\{2^{2n+2} - 1, 2^{2n+1} - 1, 2^n\}$. *International Journal of Advanced Research in Computer Science and Software Engineering*, 2(8):447–452. 13, 70, 72, 80, 82, 83
- Mohan, A. (2002). *Residue Number Systems. Algorithms and Architectures*. Kluwer Academic Publishers. 1, 2, 4, 5, 6, 22
- Mohan, A. (2007). Rns to-binary converters for a new three moduli set $\{2^{n+1} - 1, 2^n, 2^n - 1\}$. *IEEE Trans. on Circuits and Systems-II: Express briefs*, 54(9):775–779. 6, 13, 20, 22, 37, 69, 70, 87, 88, 104, 124
- Molahosseini, A., Dadkhah, C., and Navi, K. (2009). A new five-moduli set for efficient hardware implementation of reverse converter. *IEICE Electronic Express*, 6:1006–1012. 37, 48, 49



Molahosseini, A. and Navi, K. (2007). New arithmetic residue to binary converters.

International Journal of Computer Sciences and Engineering Systems., 1(4):295–

299. 6, 69, 104, 124

Molahosseini, A. and Navi, K. (2010). A reverse converter for the enhanced moduli

set $\{2^n - 1, 2^n + 1, 2^{2n}, 2^{2n+1} - 1\}$ using crt and mrc. *IEEE Annual Symposium on*

VLSI., pages 456–457. 18, 117, 118, 119, 121, 122, 123, 125, 133, 134, 136, 137,

139, 145, 146

Molahosseini, A., Navi, K., Dadkhah, C., Kavehei, O., and Timarchi, S.

(2010). Efficient reverse converter designs for the new 4-moduli sets

$\{2^n - 1, 2^n, 2^n + 1, 2^{2n+1} - 1\}$ and $\{2^n - 1, 2^n + 1, 2^{2n}, 2^{2n} + 1\}$ based on new crts.

IEEE Transaction on Circuits and Systems-I: Regular Papers, 57(4):823–835. 11,

13, 15, 16, 17, 18, 37, 38, 48, 49, 50, 51, 52, 53, 54, 64, 65, 66, 69, 97, 99, 100, 102,

104, 117, 118, 119, 121, 124, 125, 142, 143, 144, 145, 147

Molahosseini, A., Navi, K., Hashemipour, O., and Jalali, A. (2008). An efficient

architecture for designing reverse converters based on a general three-moduli set.

Journal of Systems Architecture, 54:929–934. 12, 64, 65, 70, 88, 105

Navi, K. and Esmaeildoust, M. (2010). A general converter architecture with low

complexity and high performance. *IEICE Transactions on Information systems.*,

E94-D(2):264–273. 70



Nussbaumer, H. (1976). Digital filtering using cotransforms in finite fields. *Electronic Letters*, 12(5):113–114. 3

Omondi, A. and Premkumar, B. (2007). *Residue Number Systems. Theory and Implementation*. Imperial College Press. 1, 2, 4, 5

Parhami, B. (2000). *Computer Arithmetic*. Oxford University Press, New York. 1, 2

Piestrak, S. (1995). A high-speed realization of a residue to binary number system converter. *IEEE Trans. on Circuits and Systems-II: Analog and Digital Signal Processing*, 42(10):661–663. 10, 20, 21, 33

Pontarelli, S., Cardarilli, G., Re, M., and Salsano, A. (2010). Optimized implementation of rns fir filters based on fpgas. *J Sign process Syst.*, 12. 3

Psaltis, D. and Casasent, D. (1979). Optical residue arithmetic: a correlation approach. *Optical Society of America*, 18(2):163–171. 3

Quan, S., Pan, W., Xie, Y., and Hao, Y. (2012). Rns-to-binary converter for new four-moduli set $\{2^n - 1, 2^n, 2^{n+1} - 1, 2^{n+1} + 2^n - 1\}$. *Chinese Journal of Electronics*, 21(3):430–434. 13

Sheu, M., Lin, S., Chen, C., and Yang, S. (2004). An efficient vlsi design for a residue-to-binary converter for general balance moduli $\{2^n - 3, 2^n + 1, 2^n - 1, 2^n + 3\}$ and $\{2^n, 2^n + 1, 2^n - 1, 2^{n+1} + 1\}$. *IEEE Trans. of Circuits and Systems-II: Express briefs*, 51(3):152–155. 13, 88, 124



- Slotnick, D. (1962). Modular arithmetic computing techniques. *Tech. Rep. Westinghouse Electric Corporation, Air Arm Division.*, (ASDTDR-63-280). 5
- Soderstrand, M., Jenkin, G., Jullien, G., and Taylor, F. (1986). *Residue Number Systems Arithmetic: Modern Applications in Digital Signal Processing*. IEEE press, Piscataway, NY, USA. 5, 6
- Szabo, N. and Tanaka, R. (1967). *Residue Arithmetic and its Application to Computer Technology*. MC-Graw-Hill, New York. 1, 5, 7, 22, 39, 55, 71, 89, 105, 126
- Tanaka, R. (1962). Modular arithmetic techniques. *Tech. Rep. ASTDR, Lockheed Missiles and Space Co.*, (2-38-62-1A). 5
- Toivonen, T. and Heikkila, J. (2006). Video filtering with fermat number theoretic transforms based on residue number systems. *IEEE Trans. on Circuits and Systems for Video Tech.*, 16(1):92–101. 3
- Vinod, A. and Premkumar, A. (2000). A memoryless reverse to binary converter for the 4-moduli superset $\{2^n - 1, 2^n, 2^n + 1, 2^{n+1} + 1\}$. *Journal of Circuits, Syst. and Computers*, 10:85–99. 13, 88, 124
- Wang, W., Swamy, M., Ahmad, M., and Wang, Y. (2003). A study of the residue-to-binary converters for the three moduli sets. *IEEE Trans. on Circuits and Systems-I: Fundamental Theory and Applications*, 50(2):235–243. 10, 20
- Wang, Y. (1998). New chinese remainder theorem. in *Proc. Asilomar Conference, USA*, pages 165–171. 8, 106, 108



- Wang, Y., Song, X., Aboulhamid, M., and Shen, H. (2002). Adder based residue to binary number converters for $\{2^n - 1, 2^n, 2^n + 1\}$. *IEEE Trans. on Signal Processing*, 50(7):1772–1779. 10, 20, 21, 31, 32, 33, 36, 69, 104, 124, 141
- Watson, R. and Hastings, C. (1967). *Residue Arithmetic and Reliable Computer Design*. Washington DC, Spartan. 5
- Wesolowski, M., Patronik, P., Berezowski, K., and Biernat, J. (2012). Design of a novel flexible 4-moduli rns and reverse converter. *ISSC. NUI Maynooth*. 13
- Yang, J., Chang, C., and Wang, C. (2005). An iterative modular multiplication algorithm in rns. *ELSEVIER Applied Mathematics and Computation*, 44(171):637–645. 6
- Zhang, W. and Siy, P. (2008). An efficient design of residue to binary converter for four moduli set $\{2^n - 1, 2^n + 1, 2^{2n} - 2, 2^{2n+1} - 3\}$. *International Elsevier Journal of Information Sciences*, 178(1):264–279. 12, 13, 70





Appendix A

List of Publications

1. Bankas E., Gbolagade, K. (2012) A Speed Efficient RNS to Binary Converter for the Moduli Set $\{2^n, 2^n + 1, 2^n - 1\}$. *Journal of Computing*. 4(5). 83-88.
2. Bankas E., Gbolagade, K., and Cotozana, S. (2013) An Effective New CRT based Converter for a Novel Moduli set $\{2^{2n+1} - 1, 2^{2n+1}, 2^{2n} - 1\}$. *24th IEEE International Conference on Application-specific Systems, Architectures and Processors (ASAP2013)*. Washington, USA. 142-146.
3. Bankas E., Gbolagade, K. (2013) A New Efficient RNS Reverse Converter for the 4-Moduli Set $\{2^n, 2^n + 1, 2^n - 1, 2^{2n+1} - 1\}$. *International Journal of Computer and System Sciences*. World Academy of Science, Engineering and Technology (Accepted).
4. Bankas E., Gbolagade, K. (2013) Residue to Binary Converter for a Balanced Moduli set $\{2^{2n+1} - 1, 2^{2n}, 2^{2n} - 1\}$ *International Conference on Awareness*

*Science and Technology (iCAST2013). The University of Aizu, Wakamatsu
City, Japan (Accepted).*



[iCAST 2013] Paper Decision and Camera Ready Submission

Conference Management Toolkit <cmt@microsoft.com> on behalf of
AwareMedia Chair <awaremedia13pc@gmail.com>

Sat 8/24/2013 1:16 PM

To: Bankas, Edem <eb433007@ohio.edu>;

Dear Edem Bankas, Paper#: 59 Title: A Residue to Binary Converter for a Balanced Moduli set $\{2^{2n+1}-1, 2^{2n}, 2^{2n}-1\}$

happy to inform you that your above paper has been accepted by the main session or special session of iCAST 2013 for presentation at the conference and inclusion in the proceedings.

login to the CMT system (<https://cmt.research.microsoft.com/ICUM2013>) with your e-mail account, you will find the reports from the PC members at the "Reviews" link of the "Paper Decision" column in the Author role.

note that you will prepare your camera-ready version by carefully revising your paper taking into account the comments by the PC members.

camera-ready version of your paper must be submitted to the iCAST camera-ready paper submission site by September 1.

you need to prepare two files: (1) IEEE COPYRIGHT AND CONSENT FORM by IEEE Electronic Copyright Site (2) Camera-ready paper (PDF) by IEEE PDFeXpress before submitting to our CMT system.

you can visit the iCAST camera-ready paper submission site and find the guide for the submission in the following step.

login to the "<https://cmt.research.microsoft.com/ICUM2013>" with your e-mail account

click the link the "Author" Role --> "Manage Submissions" --> "Edit" of the "Camera-Ready | Presentations" field.

The link will be "[https://cmt.research.microsoft.com/ICUM2013/Protected/Author/CameraReadySubmission.aspx?ID=\\$PaperID\\$](https://cmt.research.microsoft.com/ICUM2013/Protected/Author/CameraReadySubmission.aspx?ID=$PaperID$)".

3. Submit your two files (Copyright and Camera-ready paper) to follow the guide in the page.

Please note that any delay may prevent the inclusion of your paper in the proceedings. Please follow exactly the instructions from IEEE in order to prepare your final version.

The iCAST 2013 conference site has provided all necessary information for conference registration and hotel accommodation (<http://web-ext.u-aizu.ac.jp/conference/conf2013/icast13/venue.html>).



Our conference will be held at University of Aizu, Aizu-Wakamatsu City, Fukushima, Japan on 11/2-4, 2013.

www.udsspace.uds.edu.gh

The conference requires that at least one author register for the conference, attend, and present the paper.

You will find detailed registration information at the conference website (<http://web-ext.u-aizu.ac.jp/conference/conf2013/>).

Again, congratulations on your accepted paper!

Looking forward to seeing you at the iCAST 2013 in November!

Best regards,

013 PC Chairs

UNIVERSITY FOR DEVELOPMENT STUDIES



Journal Paper Review - A New Efficient RNS Reverse Converter for the 4-Moduli set $\{2^n, 2^n + 1, 2^n - 1, 2^{2n+1} - 1\}$ Review

Waset - Info <info@waset.org>

Tue 9/24/2013 9:55 AM

To Bankas, Edem <eb433007@ohio.edu>; edemkb@gmail.com <edemkb@gmail.com>;

1 attachment

7826938312edem_kwedzo_bankas.pdf;

m Kwedzo Bankas,

iewing your submitted paper, you are kindly requested to revise your paper as per the following review remarks.

Name

ames should be written according to information below;

ames: Times New Roman, 11 pt., Centered, Title Case (Capitalize Each Word)

INT: Do NOT write your institution address below your name.

e the following instructions at: <http://www.waset.org/downloads/title.jpg>

on

write your institution address below AUTHORS' name.

part should be written at the bottom of the first page on the left Times New Roman, 8 pt. as:

and financial support acknowledgments can be written here.

urname is with the National Institute of Standards and Technology, Boulder, CO 80305 USA (corresponding author to provide phone: 505-555-5555; fax: 505-555-5555; e-mail: author@boulder.nist.gov).

e the following instructions at: <http://www.waset.org/downloads/affiliation.jpg>

rences should be cited

nces should be cited in the text.

e the following instructions at: <http://www.waset.org/downloads/references.jpg>

ptions

PTIONS: Captions should be written Times New Roman, 8 pt., centered

BLE: Upper case, Number: Tables are numbered with Roman numerals

Capitalize each word: Units for Magnetic Properties THEN CHOOSE Font-Effects-Small Caps

COMMON MISTAKES: Table 1, Table 2.1 etc..

Please see the following instructions at: <http://www.waset.org/downloads/table.jpg>

Table in Sentence

Table numbers should be given with Roman integers in the text:

CORRECT: Table I exemplifies some of co-processing residues encountered at

WRONG: Table 1 exemplifies some of co-processing residues encountered at

Abbreviation in the Sentence

Use the abbreviation Fig. even at the beginning of a sentence.



WRONG: Figure 2.1 represents... CORRECT: Fig. 2 represents....

www.udsspace.uds.edu.gh

Ref. 10, 16, 17, and 19 are not cited in the paper.

Best regards,

A. Ariston
Editor-in-Science
International Science Council
Tel: +971559099620
www.waset.org

UNIVERSITY FOR DEVELOPMENT STUDIES

