UNIVERSITY FOR DEVELOPMENT STUDIES

UNIFIED REVERSE CONVERSION ARCHITECTURES FOR SOME POWERS OF TWO MODULI SETS

SIEWOBR HILLARY

Thesis submitted to the Department of Mathematics, Faculty of Mathematical Sciences, University for Development Studies in Partial Fulfillment of the Requirement for the award of Master of Science Degree in Computational Mathematics

2013

UNIVERSITY FOR DEVELOPMENT STUDIES

UNIFIED REVERSE CONVERSION ARCHITECTURES FOR SOME POWERS OF TWO MODULI SETS

BY

SIEWOBR HILLARY (B. Sc. Computer Science)

(UDS/MM/0008/11)

Thesis submitted to the Department of Mathematics, Faculty of Mathematical Sciences, University for Development Studies in Partial Fulfillment of the Requirement for the award of Master of Science Degree in Computational Mathematics

OCTOBER, 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) except where due acknowledgement has been made in the form of citations.

Candidate's Signature:..... Date: 07/11/2013

Name: SIEWOBR HILLARY

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:

Supervisor's Signature:..... Date: 07/11/2013

Name: PROF. KAZEEM A. GBOLAGADE

ABSTRACT

Residue Number System (RNS) has found a wide spread usage in a number of digital signal processing applications such as digital filtering, Discrete Fourier transform , Convolution, Correlation, communication, and cryptography. This is due to the following RNS inherent features: modularity, parallelism, carry free addition, borrow free subtraction, and fault tolerance. The major challenges of RNS architecture lie in moduli set selection and in the reverse conversion (conversion from residue representation to weighted representation). Reverse Conversion (RC) can be achieved either by the traditional Chinese Remainder Theorem (CRT), Mixed Radix Conversion (MRC) or the recently introduced new CRTs (CRT I and II). Several moduli sets have been proposed with algorithms designed for performing RC. In this thesis, two pair of moduli sets : $\{2^{2n+1}-1, 2^n, 2^{n+1}-1\}$ and $\{2^{2n+1}-1, 2^{2n}, 2^{n+1}-1\}$ and $\{2^{n+1}-1, 2^{2n}, 2^{n+1}+1\}$ and $\{2^{(3n+2)/2}-1, 2^{2n}, 2^{(3n+2)/2}+1\}$ are proposed together with unified architecture for efficient reverse conversion. We also propose a unified architecture for efficient reverse conversion in existing moduli sets $\{2^n-1, 2^n+1, 2^n, 2^{2n+1}-1\}$ and $\{2^n-1, 2^n+1, 2^{2n}, 2^{2n+1}-1\}$. Both theoretical and experimental results (from Xilinx ISE 14.3) suggest that the proposed schemes outperform the known related state of the art schemes in terms of area and delay.

ACKNOWLEDGEMENT

My many thanks and God's blessings to my supervisor, Professor Kazeem Alagbe Gbolagade, the Dean-in-Charge of Navrongo Campus, University for Development Studies (UDS) whose mentorship and high demand of quality results has been a great contributor to my personality in research and life as a whole. His simplicity and humility saw us work together both at his home and office since my undergraduate days.

Next, I wish to thank all my Lecturers especially, Dr. Stephen Twum, the Head of Department of Mathematics, Faculty of Mathematical Sciences (FMS), UDS, Navrongo Campus for his professionalism and fatherly care. Mr. David Sankah Laar (Head of Department of Computer Science, FMS, UDS, Navrongo Campus) and the lovely staff of the Department of Computer Science were very understanding and supportive-thanks.

To Rev. Fr. Augustine H. K. Abasi (PhD) and all the members and partners of the Catholic Charismatic Renewal and the Spirit Power family of St. Augustine Chaplaincy, UDS, Navrongo for your love, prayers, encouragement, counsel and support throughout this program and beyond.

Finally, to my family, I say you are only second to God in terms of contribution to my immeasurable success-I am grateful that you gave me the privilege to my right of education and most especially for the love you showed me all my life, may the blessings of God who is Father, Son and Holy Spirit be with you now and forever.

DEDICATION

This work is dedicated to my entire family especially Mr. Jerome Aakakpeleh Siewobr (my dad) and Paulinus Siewobr.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ACRONYMS

ASIC	Application-Specific Integrated Circuit
CPA	Carry Propagate Adder
CRT	Chinese Remainder Theorem
CSA	Carry Save Adder
DCT	Discrete Cosine Transform
DFT	Discrete Fourier Transform
DR	Dynamic Range
DSP	Digital Signal Processing
EAC	End Around Carry
FCT	Fast Cosine Transform
FFT	Fast Fourier Transform
FPGA	Field Programmable Gate Array
GCD	Greatest Common Divisor
LSB	Least Significant Bit
MR	Mixed Radix
MRC	Mixed Radix Conversion
MRD	Mixed Radix Digit
MSB	Most Significant Bit
RC	Reverse Conversion
RNS	Residue Number System
VHDL	VHSIC Hardware Description Language
WNS	Weighted Number System

CHAPTER ONE

INTRODUCTION

1.1 Background of the Study

In the last few decades, researchers have been looking at solving the major challenge in improving computational performance of digital systems i.e., the reduction or elimination of the carry propagation chains in Weighted Number Systems (WNS), e.g., binary number systems. One solution that has been proposed to speed up addition process is by making use of a number system with specialized carry characteristics e.g., Residue Number Systems (RNS) (Gbolagade *et al.* 2010).

RNS is a non-weighted number system that utilizes remainders to represent numbers. It is defined in terms of a set of relatively prime moduli $\{m_i\}_{i=1,k}$ such that the $\gcd(m_i, m_j) = 1$ for $i \neq j$, where gcd means the greatest common divisor of m_i, and m_j, and $M = \prod_{i=1}^{k} m_i$, is the dynamic range. The residues of a decimal number X can be obtained as $x_i = |X|_{m_i}$, thus X can be represented in RNS as $X = (x_1, x_2, x_3, \dots, x_k)$, $0 \leq x_i \leq m_i$. This representation is unique for any integer$X \in [0, M-1]$. $|X|_{m_i}$, is the modulo operation of X with respect to m_i (Gbolagade *et al.* 2011).

RNS is very useful in addition and multiplication dominated digital signal processing applications such as Fast Fourier Transform, digital filtering,

convolution, direct digital frequency synthesis, image processing and cryptography among others. This is due to its inherent features such as carry free addition; borrow free subtraction, digit by digit multiplication without partial product, fault tolerance, and parallelism. The speed of the arithmetic operations relies on the size of the numbers involved; smaller numbers imply faster operations. Since the numbers used in this system are smaller, it is known for faster implementation of arithmetic operations, and hence it is very attractive.

However, RNS has not found a widespread usage in general purpose computing due to the following difficult RNS arithmetic operations: overflow detection (Theodore, 1989 and Theodore, 1990)., magnitude comparison, sign detection, moduli selection, and data conversion (Wang, *et al.* 2002). Out of these numerous RNS challenges, Moduli selection and Data conversion are the two most critical issues (Wang, *et al.* 2002).

RNS architectures are typically composed of three main parts, namely, a forward (binary-to-residue) converter, residue arithmetic units, and a reverse (residue-to-binary) converter (see figure 1.1 below).

Data conversion can be categorized into forward and reverse conversions (performed by the forward and reverse converter respectively). The former involves converting a binary or decimal number into its RNS equivalent while the later conversion involves converting the RNS number into binary or decimal.

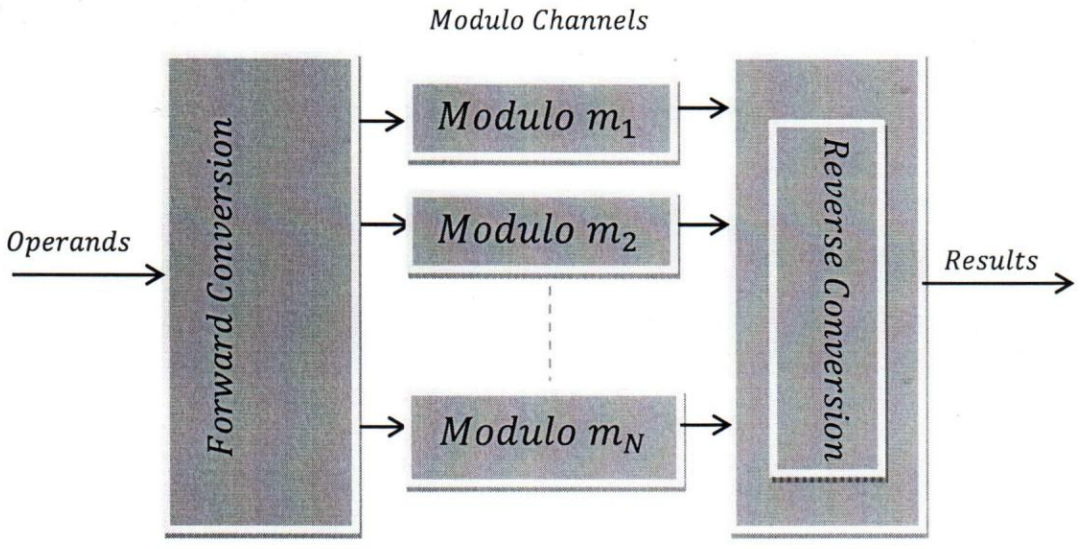

Figure 1.1: RNS Processor

The residue-to-binary converter is the most challenging part of any RNS architecture making reverse conversion relatively, more complex. Therefore for RNS design to be competitive, the conversion process must be very fast and require a low hardware demand so that the conversion overhead may not nullify the fast arithmetic advantage of RNS (Gbolagade, 2010).

1.2 Literature Review

In this section, we review relevant works done on both moduli selection and reverse conversion in RNS. Sub Section 1.2.1 reviews moduli selestion whilst Sub Section 1.2.2 reviews reverse conversion.

1.2.1 Moduli Selection Background

The set of the moduli chosen for RNS affects both the representational efficiency and the complexity of arithmetic algorithms. In general, it is desired to make the speed of the moduli as small as possible, since it is the magnitude of the largest modulus M_{k-1} that dictates the speed of arithmetic operations. It is also often desired to try to make all the moduli comparable in magnitude to the largest one, since with the computation speed is already dictated by M_{k-1} (Parhami, 2000).

However, speed and cost do not just depend on the widths of the residues but also on the moduli chosen. For instance, power-of-2 moduli are more desirable since they simplify arithmetic operations such that modulus 2^{2n} might be better than the smaller modulus $2^n + 1$ (except, perhabs, with table-lookup implementation). Also, moduli of the form $2^a - 1$ where $1 < a \in R$ are desirable and referred to as low-cost moduli (Parhami, 2000).

A special case of RNS with three moduli of the form $(2^n - 1, 2^n, 2^n + 1)$ and popularly known as the traditional moduli set is widely recommended these days because its conversion circuitry offers high speed and simplicity (Parhami, 2010).

The condition that the moduli set must be pair-wise relatively prime must always be taken into consideration. Examples of existing moduli sets include; $\{2^n, 2^n - 1, 2^n + 1\}$ (Piestrak, 1995), $\{2^{n+1} - 1, 2^n, 2^n - 1\}$

(Mohan, 2007), $\{2^n - 1, 2^n + 1, 2^{2n+1} - 1\}$ (Molahosseini *et al.* 2008), and $\{2^n - 1, 2^n, 2^n + 1, 2^{2n+1} - 1\}$ (Molahosseini *et al.* 2010).

1.2.2 Data Conversion Background

The utilization of RNS originated from the Chinese Remainder Theorem (CRT). The CRT is a mathematical idea from Sun Tzu (Master Sun's Arithmetic Manual) in the 4th century AD.

The ancient study of the RNS begins with a verse from Chinese born Sun Tzu's third-century book titled Suan-Ching. It reads;

"We have things of which we do not know the number, If we count them by three, the remainder is 2, If we count them by five, the remainder is 3, If we count them by seven, the remainder is 2, How many things are there?" The answer, 23.

How to get the answer 23 is outlined in Sun Tzu's historic work. He presents a formula for manipulating remainders of an integer after division by 3, 5, and 7. The puzzle essentially asks us to convert the RNS number $(2|3|2)_{RNS(7|5|3)}$ into its decimal equivalent. Sun Tzu formulated a method for manipulating these remainders (also known as residues), into integers. This method is regarded today as the Chinese Remainder Theorem (CRT) and is one of the common rules of converting residues into integers.

1.2.2.1 Chinese Remainder Theorem (CRT)

The CRT is formulated as follows:

For a moduli set $\{m_1, m_2, \ldots, m_n\}$ with the dynamic range $M = \prod_{i=1}^{n} m_i$, the residue number $\{x_1, x_2, \ldots, x_n\}$ can be converted into the decimal number X, according to the CRT as follows (Gbolagade et. al. 2010):

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

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

1.2.2.2 Mixed Radix Conversion

Another method for data conversion is the Mixed Radix Conversion (MRC) which can be represented as (Gbolagade *et al.* 2010):

$$X = a_1 + a_2 m_1 + a_3 m_1 m_2 + \cdots + a_n m_1 m_2 m_3 \ldots m_{n-1} \tag{1.2}$$

where the Mixed Radix Digits (MRDs), $a_i, i = 1, n$ can be computed as follows (Molahosseini *et al.* 2008):

$$a_1 = x_1$$

$$a_2 = |(x_2 - a_1)|m_1^{-1}|m_2|_{m_2}$$

$$a_3 = \left|((x_3 - a_1)|m_1^{-1}|_{m_3} - a_2)|m_2^{-1}|_{m_3}\right|_{m_3}$$

$$a_n = \left|((\ldots(x_n - a_1)|m_1^{-1}|_{m_n} - a_2)|m_2^{-1}|_{m_3} - \cdots - a_{n-1})|m_{n-1}^{-1}|_{m_n}\right|_{m_n} \tag{1.3}$$

Given the MRDs $a_i, 0 \leq a_i < m_i$, any positive number in the interval $[0, \prod_{i=1}^{n} m_i - 1]$ can be uniquely represented.

1.2.2.3 New Chinese Remainder Theorems

As earlier stated, the speed of the arithmetic operations is mainly based on the size of the numbers involved in the operations and therefore smaller numbers imply faster operations. As a result, even the CRT is parallel, it is not desired because of its dependence on the large modulo M operation size. On the other hand, though the MRC requires a smaller operation size, the slow process of finding the MRDs (its serial nature) makes it undesirable too. In order to solve these problems with the traditional CRT and the MRC the New CRTs (CRT-I and CRT-II) were proposed in (Wang *et al.* 2000) to make the computations faster and efficient without any overheads. These are formulated below:

CRT-I:

$$X = x_1 + \ell_1|k_1(x_2 - x_1) + k_2\ell_2(x_3 - x_2) + \cdots + k_{n-1}\ell_2 \ldots \ell_{n-1}(x_n - x_{n-1})|_{\ell_2 \ldots \ell_{n-1}\ell_n} \quad (1.4)$$

CRT-II (for four moduli set):

$$X = Z + \ell_1\ell_2|k_1(Y - Z)|_{\ell_3\ell_4} \quad (1.5)$$

$$Z = x_1 + P_1|k_2(x_2 - x_1)|_{P_2} \quad (1.6)$$

$$Y = x_3 + P_3|k_3(x_4 - x_3)|_{P_4} \quad (1.7)$$

with,

$$|k_1P_1P_2|_{P_3P_4} = 1 \quad (1.8)$$

$$|k_2P_1|_{P_2} = 1 \quad (1.9)$$

$$|k_3 P_3|_{P_4} = 1 \tag{1.10}$$

where P_i and k_i, $i = [1,4]$ are the moduli and multiplicative inverses respectively.

1.3 Related Work and Problem Statement

Wang *et al.* (2002), proposed adder based residue to binary converters for the traditional moduli set i.e. $\{2^n - 1, 2^n, 2^n + 1\}$. Their proposals tuned to be better than (Piestrak, 1995), (Bhardwaj *et al.* 1998) and (Gallaher *et al.* 1997) in terms of both area and delay.

Mohan (2007), proposed the new moduli set $\{2^{n+1} - 1, 2^n, 2^n - 1\}$ together with three RNS-to-binary converters. This moduli set had lower RNS arithmetic speed compared to the traditional moduli set since it eliminates the complex $2^n + 1$ modulo. One of the converters was MRC based whilst the other two were based on the traditional CRT. This moduli set uses moduli of uniform word length (n to n+1 bits). It was derived from a previously investigated four-moduli supper set $\{2^n - 1, 2^n, 2^n + 1, 2^{n+1} - 1,\}$. The current state of the art reverse converter for the moduli set $\{2^{n+1} - 1, 2^n, 2^n - 1\}$ was proposed by Gbolagade *et al.* (2010). The traditional CRT was simplified to obtain two new memory-less residue to binary converters that required mod-$(2^{n+1} - 1)$ operations instead of both mod-$(2^{n+1} - 1)$ and mod-$(2^n - 1)$ required by the converters proposed by Mohan (2007). Whilst the first converter did not cover the entire

dynamic range, the second proposed converter did so. FPGA implementation results indicated that, on average, the proposed limited dynamic range converter achieved about 42% area reduction whilst the proposed full dynamic range converter provided 29.48% area reduction when compared with the related state of the art converter. Both proposed converters also exhibited small speed improvements over the state of the art equivalent converter in Gbolagade *et al.* (2010).

Molahossieni *et al.* (2008) enhanced the modulus $2^{n+1}-1$ in the moduli set $\{2^{n+1}-1, 2^n, 2^n-1\}$ to obtain the moduli set $\{2^{2n+1}-1, 2^n, 2^n-1\}$ with an MRC based converter. A speed efficient CRT based converter which does not cover the entire dynamic range was proposed in Gbolagade *et al.* (2010).

The moduli set $\{2^{2n+1}-1, 2^{2n}, 2^n-1\}$ was proposed in Gbolagade *et al.* (2009) by extending the modulus 2^n in the moduli set $\{2^{2n+1}-1, 2^n, 2^n-1\}$ to the modulus 2^{2n}. In this work the traditional CRT and MRC were simplified to obtain three reverse converters; two fast CRT based converters one of which did not cover the entire dynamic range and an MRC based converter. Experimental results from ASIC 0.13 µm Standard Cell technology implementation showed that on the average the proposed converters were better than the one in Molahosseini *et al.* (2008).

Hariri *et al.* (2007) proposed the moduli set $\{2^{2n}-1, 2^n, 2^{2n}+1\}$, for higher dynamic range three moduli sets. The traditional CRT was

simplified to obtain a low area cost and high speed reverse converter at the same dynamic range when compared to Wang *et al.* (2002).

The dynamic range (DR) provided by most of these moduli sets are either not sufficient for applications which require larger DR and or they do not support applications that require more parallelism. To solve this problem, 4n-bit DR 4-moduli sets such as $\{2^n-1, 2^n, 2^n+1, 2^{n+1}-1\}$ and $\{2^n-1, 2^n, 2^n+1, 2^{n+1}+1\}$ were proposed in Bhardwaj *et al.* (1999) and Vinod and Premkumah (2000) respectively. Recently, higher dynamic range moduli sets $\{2^n-1, 2^n, 2^n+1, 2^{2n+1}-1\}$ and $\{2^n-1, 2^{2n}, 2^n+1, 2^{2n}+1\}$ were proposed in Molahosseini et. al. (2010) with CRT II and CRT I based converters respectively. The moduli set $\{2^n-1, 2^n, 2^n+1, 2^{2n}+1\}$ was obtained by enhancing the power of two modulo in the moduli set $\{2^n-1, 2^{2n}, 2^n+1, 2^{2n}+1\}$ proposed by Cao *et al.* (2003). In Sousa and Antao (2012), an MRC divide and conquer approach was used to present a unified reverse conversion architecture for the moduli sets $\{2^n-1, 2^n, 2^n+1, 2^{2n+1}-1\}$ and $\{2^n-1, 2^{2n}, 2^n+1, 2^{2n+1}-1\}$ which is configurable for either moduli sets. The converter for the $\{2^n-1, 2^n, 2^n+1, 2^{2n+1}-1\}$ moduli set outperformed the one in Molahosseini *et al.* (2010) whilst that of the $\{2^n-1, 2^{2n}, 2^n+1, 2^{2n+1}-1\}$ moduli set outperform the converter in Zhang and Siy, (2008) for the moduli set $\{2^n-1, 2^n+1, 2^{2n}-2, 2^{2n+1}-3\}$. The major disadvantages of these moduli sets include:

i. The presence of very complex multiplicative inverses and moduli that lead to very expensive reverse conversion as reported in Mohan and Premkumar (2007), Antao and Sousa, (2012), Molahosseini *et al.* (2010) and Zhang and Siy, (2008) .

ii. Poor RNS arithmetic speed due to the presence of channel arithmetic units modulo $2^{2n}+1$ e.g. $\{2^n-1, 2^n, 2^{2n}+1, 2^{2n}+1\}$, $\{2^n-1, 2^{2n}, 2^n+1, 2^{2n}+1\}$ and $\{2^{2n}-1, 2^n, 2^{2n}+1\}$

iii. All the converter architectures in these schemes depend heavily on modular adders which impose twice the delay of a regular binary adder at the same area (Sousa *et. al.* 2012).

The ability to solve these problems; proposing new moduli sets with lower RNS arithmetic speed, eliminating complex multiplicative inverses in the conversion process, implementation of converters that do not require the explicit use of modulo operations i.e. require only CSAs and binary CPAs to implement is required to design high speed RNS systems at lower area cost. These interventions are also required in order to efficiently implement some difficult RNS operations, e.g., overflow detection, magnitude comparison, sign detection, division and also to build fault tolerant RNS architectures. Due to the necessities of these interventions, in this thesis, we seek to propose efficient moduli sets with efficient data conversion techniques.

1.4 Research Questions

As a result of the problems stated above, we ask ourselves the following the research questions:

i. Can we propose new moduli sets for efficient reverse conversion?

ii. Can we unify the architecture for reverse conversion in the proposed moduli sets?

iii. Can we write synthesizable VHDL code for the proposed converters as well as the state of the art.

iv. Can we implement the proposed algorithms and the state of the art on Field Programable Gate Arrays (FPGAs)?

v. What will be the effect of the proposed algorithms in terms of hardware resources and conversion speed?

1.5 Objectives

The following objectives have been set to address the research questions above:

i. Propose new and efficient moduli sets for efficient reverse conversion.

ii. Unify the architecture for reverse conversion in the proposed moduli sets.

iii. Implement the proposed algorithms and the state of the art on xilinx FPGA.

iv. Evaluate the performance of the proposed algorithms against the state of the art designs in terms of hardware resources and conversion speed.

1.6 Significance of Study

As stated earlier, the two most critical issues for the residue arithmetic are moduli selection and data conversion. For a successful application of RNS, data conversion must be very fast so that the conversion overhead doesn't nullify the RNS advantages (Gbolagade et. al., 2008). Also, as earlier on stated, efficient moduli selection and data converter designs are required in order to efficiently implement some difficult RNS operations, e.g., overflow detection, magnitude comparison, sign detection, division and also to build fault tolerant RNS architectures.

Proposing newer moduli sets, efficient reverse converters and overflow detection algorithms will therefore bring enhancement in speed and area consumption to DSP intensive computations like digital filtering, convolutions, correlations, DFT, FFT computations, direct digital frequency synthesis, image processing and cryptography. A broad research like the one conducted here will as well contribute towards the realization of RNS based general purpose processors.

CHAPTER TWO

UNIFIED VLSI IMPLEMENTATION OF RESIDUE-TO-BINARY CONVERTERS FOR THE NEW MODULI SETS $\{2^{2n+1}-1, 2^n, 2^{n+1}-1\}$ AND $\{2^{2n+1}-1, 2^{2n}, 2^{n+1}-1\}$

2.1 Introduction and Background

The moduli set $\{2^{n+1}-1, 2^n, 2^n-1\}$ was proposed in Molahosseini *et al.* (2008) by removing the moduos 2^n+1 from the 4-moduli set $\{2^n-1, 2^n, 2^n+1, 2^{n+1}-1\}$. This is due to the fact that performing modulo (2^n+1)-type arithmetic is complex and degrades the entire RNS processor performance in terms of both area and delay (Gbolagade *et al.* 2010). Recently, the modulus $2^{n+1}-1$ in the moduli set $\{2^{n+1}-1, 2^n, 2^n-1\}$ was enhanced to obtain the moduli set $\{2^{2n+1}-1, 2^n, 2^n-1\}$ and its associated MRC based converter was presented. Also, moduli sets $\{2^n-1, 2^n, 2^n+1, 2^{2n}+1\}$ (Cao *et al.* 2003) and $\{2^n-1, 2^n, 2^n+1, 2^{2n+1}-1\}$ (Molahosseini *et al.* 2010) were proposed to support higher dynamic range (DR) and increase parallelism in the RNS arithmetic unit. However, the state of the art converter for the moduli set $\{2^n-1, 2^n, 2^n+1, 2^{2n+1}-1\}$ was proposed in Sousa and Antao (2012) using MRC divide and conquer approach. The major limitation of these moduli sets is the presence of the long delay moduli $2^{2n}+1$ and 2^n+1. In this chapter, we propose two new moduli sets $\{2^{2n+1}-1, 2^n, 2^{n+1}-1\}$ and $\{2^{2n+1}-1, 2^{2n}, 2^{n+1}-1\}$ for efficient data conversion and arithmetic operations in RNS. We further propose a method for unifying

the design of the converters for the proposed moduli sets and show that our schemes outperform the known state of the art related schemes.

The MRC for three moduli sets is given as:

$$X = \ell_1 + \ell_2 m_1 + \ell_3 m_1 m_2 \tag{2.1}$$

where the Mixed Radix Digits (MRDs), $\ell_i, i = 1,3$ can be computed as follows (Gbolagade *et al.* 2010):

$$\ell_1 = x_1,\ \ell_2 = \left|(x_2 - \ell_1)\lambda_{1,2}\right|_{m_2},\ \ell_3 = \left|((x_3 - \ell_1)\lambda_{1,3} - \ell_2)\lambda_{2,3}\right|_{m_3} \tag{2.2}$$

where, $\lambda_{i,j}$ is the multiplicative inverse of m_i with respect to m_j i.e. $\left|\lambda_{i,j} * m_i\right|_{m_j} = 1$.

Given the MRDs $\ell_i, 0 \leq \ell_i < m_i$, any positive number in the interval $[0, \prod_{i=1}^{n} m_i - 1]$ can be uniquely represented.

The remainder of this chapter is organized as follows. In Section 2.2, the new moduli sets and their respective reverse conversion algorithms are proposed. Section 2.3 presents the hardware implementation of the conversion techniques, while Section 2.4 gives a performance comparison with the similar best known state of the art converters and Section 2.5 concludes the chapter.

2.2 Proposed Converters

In this section, we present fast and area efficient reverse converters for the moduli sets $\{2^{2n+1} - 1, 2^n, 2^{n+1} - 1\}$ and $\{2^{2n+1} - 1, 2^{2n}, 2^{n+1} - 1\}$.

The notations presented below are adopted from Sousa *et al.* (2012) and will be very useful in designing the proposed reverse converter:

i. For an n-bit value γ, bits are referred from the Most Significant Bit (MSB) to the Least Significant Bit (LSB) as $\gamma[n-1],\ldots,\gamma[0]$.

ii. $\gamma^l[k]$ refers to an l-bit number such that: $\gamma^l[k] = \gamma[k+l-1]2^{l-1} + \ldots + \gamma[k+1]2 + \gamma[k]$;

iii. $\mathcal{O}$ and $\mathcal{Z}$ each refer to a number whose binary representation is an all-one and all-zero string, respectively;

iv. The symbol $\bowtie$ operates the concatenation of the binary representation of two numbers.

The following Lemmas are important in the design of the proposed converters:

Lemma 1: Modulo 2^s of a number is equivalent to s LSBs of the number.

Lemma 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 $(2^s - 1)$ (Gbolagade *et al.* 2010).

Lemma 3: 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 (Gbolagade *et al.* 2010).

Lemma 4: Modulo a of an integer b of higher length than a (where a is n-bit long) can be expressed as modulo a of the sum of integers gotten by partitioning b into n-bit fields.

Lemma 5: The sum of a and 2^nb is computed as b concatenation a if a is an n-bit number.

Lemma 6: Modulo $(2^s - 1)$ of the sum of two n-bit numbers can be computed using a binary Carry Propagate Adder (CPA) with either a constant carry in of 1 or 0.

2.2.1 Proposed Method for Reverse Conversion

We represent the target moduli sets by $\{2^{2n+1} - 1, 2^{\mho n}, 2^{n+1} - 1\}$, with $\mho = 1,2$. First, we show that the moduli are relatively prime and suitable for RNS. Next, we show that the computation of multiplicative inverses are unity and present a low complexity design which does not require the explicit use of modulo operation.

Theorem 1: The moduli set $\{2^{2n+1} - 1, 2^{\mho n}, 2^{n+1} - 1\}$ includes pair wise relatively prime moduli.

Proof:

From Euclidean theorem, we have:

$$\gcd(2^{2n+1} - 1, 2^{\mho n}) = 1$$

$$\gcd(2^{\mho n}, 2^{n+1} - 1) = \gcd(2^{n+1-\mho} - 1,1) = 1$$

$$\gcd(2^{2n+1} - 1, 2^{n+1} - 1) = \gcd(2^n - 1,1) = 1$$

Thus the elements of the moduli set $\{2^{2n+1} - 1, 2^{\mho n}, 2^{n+1} - 1\}$ are pairwise relatively prime.

Theorem 2: Given the moduli set $\{2^{2n+1}-1, 2^{\upsilon n}, 2^{n+1}-1\}$, where, $m_1 = 2^{2n+1}-1$, $m_2 = 2^{\upsilon n}$, and $m_3 = 2^{n+1}-1$ for every integer $n > 1$, the followings hold true:

$$\lambda_{1,2} = -1 \tag{2.3}$$

$$\lambda_{1,3} = -2 \tag{2.4}$$

$$\lambda_{2,3} = 2^{\upsilon} \tag{2.5}$$

Proof:

Since:

$$|(2^{2n+1}-1) * (-1)|_{2^{\upsilon n}} = |-2^{2n+1}+1|_{2^{\upsilon n}} = \left||-2^{2n+1}|_{2^{\upsilon n}} + |1|_{2^{\upsilon n}}\right|_{2^{\upsilon n}}$$

$$= |1|_{2^{\upsilon n}} = 1$$

Then, $\lambda_{1,2} = -1$ and (2.3) holds true.

Similarly, since:

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

$$= |-1+2|_2 = |1|_{2^{n+1}-1} = 1$$

Then, $\lambda_{1,2} = -2$ and (2.4) holds true.

Similarly, since:

$$|(2^{\upsilon n}) * 2^{\upsilon}|_{2^{n+1}-1} = |2^{\upsilon n+\upsilon}|_{2^{n+1}-1} = |(2^{n+1})^{\upsilon}|_{2^{n+1}-1}$$

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

Then, $\lambda_{1,2} = 2^{\upsilon}$ and (2.5) holds true.

By substituting (2.3), (2.4) and (2.5) into (2.2), the MRDs for the targeted moduli set are computed as follows:

$$\ell_1 = x_1 = x_1^{2n+1}[0] \tag{2.6}$$

$$\ell_2 = |x_1 - x_2|_{2^{\upsilon n}} \tag{2.7}$$

$$= x_1^{\upsilon n}[0] + \overline{x_2^{\upsilon n}[0]} + 1$$

Also,

$$\ell_3 = |(2^{\upsilon+1}x_1 - 2^{\upsilon+1}x_3 - 2^{\upsilon}\ell_2)|_{2^{n+1}-1} \tag{2.8}$$

We can further simplify (2.8) as follows:

$$k_1 = |2^{\upsilon+1}x_1|_{2^{n+1}-1} \tag{2.9}$$

$$= |2^{\upsilon}(x_1^{2n+1}[0] \bowtie \mathcal{Z}[0])|_{2^{n+1}-1}$$

$$= |2^{\upsilon}(x_1^{n+1}[n] + x_1^{n}[0] \bowtie \mathcal{Z}[0])|_{2^{n+1}-1}$$ **(Applying Lemma 4)**

Applying Lemma 3 to (2.9) will give us:

$$k_1 = |k_{11} + k_{12}|_{2^{n+1}-1} \tag{2.10}$$

with,

$$k_{11} = x_1^{n-\upsilon+1}[n] \bowtie x_1^{\upsilon}[2n - \upsilon + 1] \tag{2.11}$$

$$k_{12} = x_1^{n-\upsilon}[0] \bowtie \mathcal{Z}[0] \bowtie x_1^{\upsilon}[n - \upsilon] \tag{2.12}$$

$$k_2 = |-2^{\upsilon+1}x_3|_{2^{n+1}-1} = \overline{x_3^{n-\upsilon}[0]} \bowtie \overline{x_3^{\upsilon+1}[n - \upsilon]}$$ **(Applying Lemma 3)** (2.13)

$$k_3 = |-2^{\upsilon}\ell_2|_{2^{n+1}-1} \tag{2.14}$$

Applying Lemma 4 to (2.11) we have:

$$k_3 = |k_{31} + k_{32}|_{2^{n+1}-1} \tag{2.15}$$

with,

$$k_{31} = \left|2^2 * \left(\overline{\ell_2^{n+1}[0]}\right)\right|_{2^{n+1}-1} \tag{2.16}$$

$$= \overline{\ell_2^{n-\upsilon+1}[0]} \bowtie \overline{\ell_2^{2}[n-1]}$$ **(Applying Lemma 3)**

$$k_{32} = \left|2^{\upsilon} * \left(\mathcal{O}^{\upsilon}[0] \bowtie \overline{\ell_2^{n-\upsilon+1}[\upsilon n - n + \upsilon - 1]}\right)\right|_{2^{n+1}-1} \quad (2.17)$$

$$= \overline{\ell_2^{n-\upsilon+1}[\upsilon n - n + \upsilon - 1]} \bowtie \mathcal{O}^{\upsilon}[0] \textbf{ (Applying Lemma 3)}$$

It is worth noting that k_{31} in (2.15) and all subsequent equations only applies to the case when $\upsilon = 2$ since we cannot apply Lemma 4 cannot be applied to (2.11) when $\upsilon = 1$.

Substituting (2.11), (2.12), (2.13), (2.16) and (2.17) into (2.8) we have;

$$\ell_3 = |k_{11} + k_{12} + k_2 + k_{31} + k_{32}|_{2^{n+1}-1} \quad (2.18)$$

Applying Lemma 6, (2.18) could be further simplified into either (2.19) or (2.20) below:

$$\ell_3 = k_{11} + k_{12} + k_2 + k_{31} + k_{32} + 1 \quad (2.19)$$

$$\ell_3 = k_{11} + k_{12} + k_2 + k_{31} + k_{32} + 0 \quad (2.20)$$

Theorem 3: Given the conditions in theorem 2 above, the binary equivalent X, of an RNS number (x_1, x_2, x_3) can be computed as follows:

$$X = x_1 + 2^{2n+1}\ell_2 - \ell_2 + 2^{(2+\upsilon)n+1}\ell_3 - 2^{\upsilon n}\ell_3 \quad (2.21)$$

Proof:

Substituting $m_1 = 2^{2n+1} - 1$, $m_2 = 2^{\upsilon n}$, $m_3 = 2^{n+1} - 1$ and the MRDs into (2.1) we obtain (2.21).

Equation (2.21) can further be simplified as:

$$X = \tau_0 + \tau_1 + 1 \quad (2.22)$$

with,

$$\tau_0 = x_1 + 2^{2n+1}\ell_2 + 2^{(2+\upsilon)n+1}\ell_3 \quad (2.23)$$

$$= \ell_3^{n+1}[0] \bowtie \ell_2^{\upsilon n}[0] \bowtie x_1^{2n+1}[0] \textbf{ (Applying Lemma 5)}$$

$$\tau_1 = -\ell_2 - 2^{\upsilon n}\ell_3 \quad (2.24)$$

$$= \mathcal{O}^{n}[0] \bowtie \overline{\ell_3^{n+1}[0]} \bowtie \overline{\ell_2^{\mho n}[0]} \textbf{ (Applying Lemma 5)}$$

2.3 Hardware Implementation

The schematic diagram depicted in Figure 2.1 represents the proposed unified architecture for proposed reverse converters for the moduli sets under investigation.

As shown in Figure 2.1, ℓ_2 is computed according to (2.7) using a $\mho$n-bit binary CPA (namely CPA 1), whilst ℓ_3 requires $(\mho + 1)$ CSAs with End-Around Carry (EAC) of length $(n + 1)$-bit (notably CSAs 1 through to CSA $(\mho + 1)$ with EACs) and two binary CPAs of length $(n + 1)$-bit using anticipated computations from (2.19) and (2.20) and a multiplexer to select the correct result. To obtain the final result X, a $(\mho + 3)n + 2$-bit binary CPA 4 is used to compute (2.22).

Table 2.1 presents the values for delay and circuit area required by the proposed converters and the known equivalent state of the art converters in Molahosseini *et al.* (2008), Cao *et al.* (2003) and Sousa and Antao (2012). We assume in this work as in and Antao (2012) that the delay of a CPA with EAC is twice the delay of a binary CPA. Considering $\boldsymbol{\tau}_{FA}$ and Δ_{FA} as the delay and area of a 1-bit Full Adder (FA), respectively. Furthermore, we consider that the area and delay of a half adder (HA) are $\Delta_{HA} = \frac{1}{2}\Delta_{FA}$ and $\boldsymbol{\tau}_{HA} = \frac{1}{2}\boldsymbol{\tau}_{FA}$, respectively.

The values of area and delay for the proposed converters are presented in Table 2.1 as follows: CPA 1 requires $(\mho n)\Delta_{FA}$ and $(\mho n)\boldsymbol{\tau}_{FA}$ to compute

ℓ_2, while ℓ_3 is computed at a delay of $(n + \mho + 3)\tau_{FA}$ and $((\mho + 3)n + 2)\Delta_{FA}$ area (it is worth noting that CSAs 1 and $(\mho + 1)$ are made up of 1 and $\mho$ HAs respectively). It is worth nothing that in the computation of the final result X, Equation (2.22), CPA 4 is made up of $2n + 1$-bit half adders and thus requires an area of $((\mho + 2)n + 1.5)\Delta_{FA}$ and imposes a delay of $((2 + \mho)n + 1.5)\tau_{FA}$

The total delay therefore required for complete reverse conversion using the proposed architecture is one less the sum of delays required for computing (2.7), (2.19) or (2.20) and (2.22) i.e. $(\mho n)\tau_{FA} + (n + \mho + 2)\tau_{FA} + ((\mho + 2)n + 1.5)\tau_{FA} - \tau_{FA} = ((2\mho + 3)n + \mho + 2.5)\tau_{FA}$ since the delay of CSA 1 is hidden in that of CPA 1. Thus the proposed converters will require delays of $(5n + 3.5)\tau_{FA}$ and $(7n + 4.5)\tau_{FA}$ for $\mho = 1$ and $\mho = 2$ respectively. In terms of area, the proposed converters require $(\mho n)\Delta_{FA} + ((\mho + 3)n + 2)\Delta_{FA} + ((\mho + 2)n + 1.5)\Delta_{FA} = ((3\mho + 5)n + 3.5)\Delta_{FA}$ i.e. $(8n + 3.5)\Delta_{FA}$ and $(11n + 3.5)\Delta_{FA}$ respectively for $\mho = 1$ and $\mho = 2$.

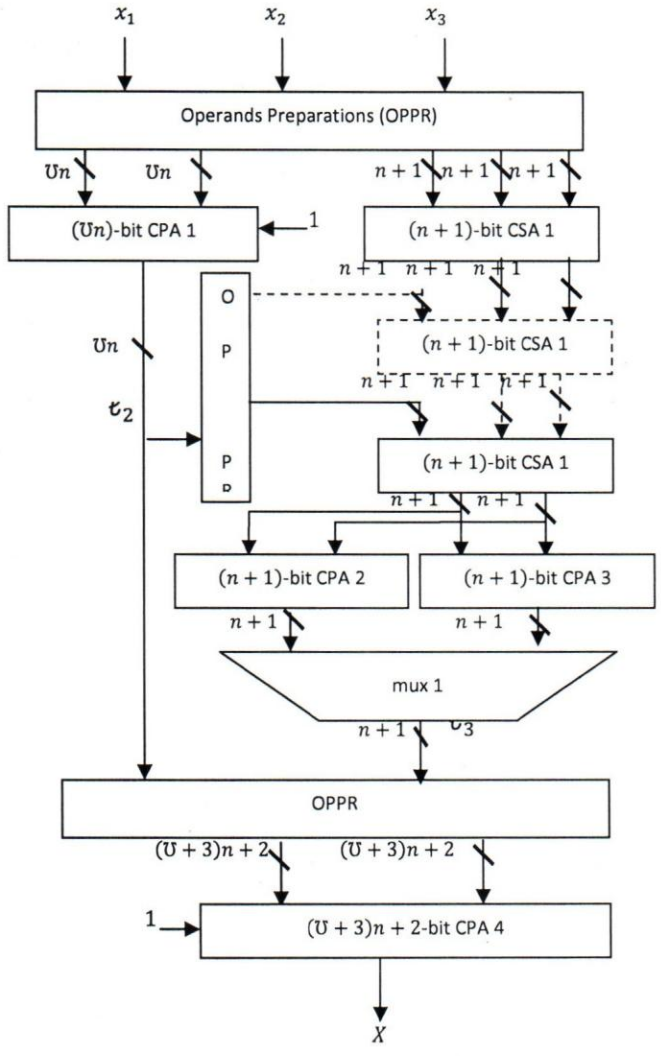

Figure 2. 1: Schematic Diagram of Proposed Unified Converter (Dashed signal lines and adder apply only when $\upsilon = 2$)

Table 2.1: Comparison of the speed of different moduli sets

Moduli Set	Critical Path	Delay
$\{2^{2n+1}-1, 2^n, 2^{n+1}-1\}$	$2^{2n+1}-1$	$2\log_2(n+0.5)+5$
$\{2^{2n+1}-1, 2^n, 2^n-1\}$		
$\{2^{2n+1}-1, 2^{2n}, 2^{n+1}-1\}$		
$\{2^n-1, 2^n, 2^n+1, 2^{2n+1}-1\}$	2^n+1	$2\log_2(n)+6$
$\{2^n-1, 2^n, 2^n+1, 2^{2n}+1\}$	$2^{2n}+1$	$2\log_2(n)+8$

Table 2.2: Area, Delay Comparison

DR	Converter	Area(Δ_{FA})	Delay(τ_{FA})
$4n$	Molahosseini *et al.* (2008)	$9n + 2$	$10n + 3$
	Our Work	$8n + 3.5$	$5n + 3.5$
$5n$	Sousa and Antao (2012)	$13n + 2$	$8n + 1$
	Cao *et al.* (2003)	$11n + 6$	$8n + 3$
	Our Work	$11n + 3.5$	$7n + 4.5$

2.4 Performance Evaluation

To evaluate the performance of the proposed residue to binary converters, we compare our proposals with related state of the art 4n and 5n DR moduli sets presented in Molahosseini *et al.* (2008), Cao *et al.* (2003) and Sousa and Antao (2012).

The speed of the arithmetic unit of RNS systems based on the proposed moduli sets and Molahosseini *et al.* (2008) is dictated by the critical modulos $2^{2n+1} - 1$ whilst those of Sousa and Antao (2012) and Cao *et al.* (2003) are restricted to low-performing critical moduli $2^n + 1$ and $2^{2n} + 1$ respectively. Table 2.1 shows the critical moduli of the proposed moduli sets and the state of the art and their respective unite gate delays adopted from Molahosseini *et al.* (2010). From Table 2.1 it is clear that though Sousa and Antao (2012) and Cao *et al.* (2003) support higher parallelism, the proposed converter for the moduli set $\{2^{2n+1} -$

$1, 2^{2n}, 2^{n+1} - 1\}$ has relatively better RNS arithmetic unit speed. It is also worth noting that the proposed moduli sets have slightly better DR compared to the state of the art. On the part of area and delay requirement, the theoretical analysis presented in Table 2.2 shows that the proposed converters outperform the state of the art converters in Molahosseini *et al.* (2008), Cao *et al.* (2003) and Sousa and Antao (2012) in terms of both area and delay. It is also worth noting that the converters for the proposed moduli sets do not require the use of modular adders (CPAs with EACs for that matter) as opposed to the mods-$2^{2n+1} - 1, 2^{2n} - 1$ and $2^{4n} - 1$ arithmetic required by Molahosseini *et al.* (2008), Cao *et al.* (2003) and Sousa and Antao (2012) respectively.

A well known library of arithmetic units (Zimmermann, 1998), which contains a structural specification of components, namely optimized prefix adders written in synthesizable VHDL code (Sousa and Antao, 2012), was employed to obtain the HDL specification of both the proposed and related state of the art converters. Using the HDL specification of the converters, experimental assessment was carried out using Xilinx ISE 14.3 software to target a Xc4vlx15-10sf363 FPGA. The results obtained after design place and route are given in terms of the number of FPGA slices and input-to-output propagation delays (in nano seconds) for various DR requirements (different values of n) are presented in Table 2.3.

Table 2.3: Converters' Delay τ [ns] and Area [Number of Slices]

DR	Converter	$n = 2$		$n = 4$		$n = 8$		$n = 16$	
		Δ	τ	Δ	τ	Δ	τ	Δ	τ
$4n$	Molahosseini *et al.* (2008)	20	15.4	45	24.8	105	27.3	215	33.8
	Our Work	12	10.5	26	12.7	52	13.4	100	14.9
$5n$	Sousa and Antao (2012)	24	16.2	59	24.2	115	27.0	235	36.8
	Cao *et al.* (2003)	34	18.3	74	20.1	161	23.3	346	23.8

These results suggest that, the proposed converter for the moduli set $\{2^{2n+1}-1, 2^n, 2^{n+1}-1\}$ improves the area and delay in Molahosseini *et al.* (2008) by 50.65% and 49.16% respectively. On the other hand, the converter for the proposed moduli set $\{2^{2n+1}-1, 2^{2n}, 2^{n+1}-1\}$ improves the area and delay in Sousa and Antao (2012) by about 39.03% and 44.53% respectively, whilst the area and delay in Cao *et al.* (2003) are improved by about 53.03% and 32.40% respectively.

2.5 Conclusion

In this chapter, two new moduli sets $\{2^{2n+1}-1, 2^n, 2^{n+1}-1\}$ and $\{2^{2n+1}-1, 2^{2n}, 2^{n+1}-1\}$ together with their associated MRC based reverse converters have been proposed. We presented a unified and configurable architecture for reverse conversion in the proposed moduli sets by presenting a converter for the moduli set $\{2^{2n+1}-1, 2^{\upsilon n}, 2^{n+1}-1\}$ for values of $\upsilon = 1,2$. We demonstrated that the computation of multiplicative inverses could be avoided. The proposed converters and the best known equivalent state of the art converters were implemented on

Xilinx Xc4vlx15-10sf363 FPGA. On average, the converter for the proposed moduli set $\{2^{2n+1}-1, 2^n, 2^{n+1}-1\}$ improves area by about 50.65% and delay by about 49.16% when compared to the known equivalent state of the art converter . Also, the converter for the proposed moduli set $\{2^{2n+1}-1, 2^n, 2^{n+1}-1\}$ improves the area and delay required by the state of the art converter for the moduli set $\{2^n-1, 2^n, 2^n+1, 2^{2n+1}-1\}$ by about 39.03% and 44.53% respectively and also improves the area and delay required by the state of the art converter for the moduli set $\{2^n-1, 2^n, 2^n+1, 2^{2n}+1\}$ by about 53.03% and 32.40% respectively.

CHAPTER THREE

CONFIGURABLE RESIDUE-TO-BINARY CONVERTERS FOR THE MODULI SETS $\{2^{n+1}-1, 2^{2n}, 2^{n+1}+1\}$ AND $\{2^{(3n+2)/2}-1, 2^{2n}, 2^{(3n+2)/2}+1\}$

3.1 Introduction and Background

Due to the lower dynamic range (DR) of the moduli set $\{2^n-1, 2^n, 2^n+1\}$, it was extended to get the moduli set $\{2^{2n}-1, 2^n, 2^{2n}+1\}$ (Hariri *et al.* 2007). However, the major disadvantage of the moduli set $\{2^{2n}-1, 2^n, 2^{2n}+1\}$ is that it requires slow and expensive RC as well as channel arithmetic units modulo $2^{2n}+1$. This type of arithmetic have been well documented to be disadvantageous in previous chapters.

In this chapter, we propose the moduli sets $\{2^{n+1}-1, 2^{2n}, 2^{n+1}+1\}$ and $\{2^{(3n+2)/2}-1, 2^{2n}, 2^{(3n+2)/2}+1\}$ (for even n) which eliminate the major disadvantages of Bhardwaj *et al.* (1999), Hariri *et al.* (2007) and Cao *et al.* (2003) discussed above and have an added advantage of higher DR.

The CRT which is used to derive reverse converters for the proposed moduli sets is presented for three moduli sets as (Gbolagade *et al.* 2010):

$$X = \left|\sum_{i=1}^{3} M_i \left|M_i^{-1} x_i\right|_{m_i}\right|_M \tag{3.1}$$

with,

$$M = \prod_{i=1}^{3} m_i \tag{3.2}$$

$$M_i = \frac{M}{m_i} \tag{3.3}$$

$$\left|M_i M_i^{-1}\right|_{m_i} = 1, \forall\, i = [1,3] \tag{3.4}$$

The remainder of this chapter is organized as follows. In Section 3.2, the new moduli sets and their respective reverse conversion algorithms are proposed. Section 3.3 presents the hardware implementation of the conversion techniques, while Section 3.4 gives a performance comparison with the similar best known state of the art converters and Section 3.5 concludes the chapter.

3.2 Proposed Converters

In this section, we present efficient reverse converters for the moduli sets $\{2^{n+1}-1, 2^{2n}, 2^{n+1}+1\}$ and $\{2^{(3n+2)/2}-1, 2^{2n}, 2^{(3n+2)/2}+1\}$.

The following notations below are adopted from Sousa and Antao (2012) for presenting the proposed method and the corresponding reverse converter:

i. For an n-bit value γ, bits are referred from the Most Significant Bit (MSB) to the Least Significant Bit (LSB) as $\gamma[n-1], \ldots, \gamma[0]$.

ii. $\gamma^l[k]$ refers to an l-bit number such that: $\gamma^l[k] = \gamma[k+l-1]2^{l-1} + \ldots + \gamma[k+1]2 + \gamma[k]$;

iii. $\mathcal{O}$ and $\mathcal{Z}$ each refer to a number whose binary representation is an all-one and all-zero string, respectively;

iv. The symbol $\bowtie$ operates the concatenation of the binary representation of two numbers.

The following Lemmas are important in the design of the proposed converters:

Lemma 1: 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 $(2^s - 1)$ (Gbolagade *et al.* 2010).

Lemma 2: 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 (Gbolagade *et al.* 2010).

3.21 Proposed Method for Reverse Conversion

We represent the target moduli sets by $\{2^{\tau/2} - 1, 2^{2n}, 2^{\tau/2} + 1\}$ with $\tau = 3n + 2, 2n + 2$ ($\tau = 3n + 2$, n must be even). First, we show that the moduli are prime relative and suitable for RNS. Next, we show that the computation of multiplicative inverses are unity and present a low complexity design which requires only carry save adders (CSAs) and binary carry propagate adders (CPAs).

Theorem 1: The moduli set $\{2^{\tau/2} - 1, 2^{2n}, 2^{\tau/2} + 1\}$ includes pair wise relatively prime moduli.

Proof:

From Euclidean theorem, we have:

$$\gcd(2^{2n}, 2^{\tau/2} - 1) = 1$$

$$\gcd(2^{\tau/2} + 1, 2^{\tau/2} - 1) = \gcd(2^{\tau/2} - 1, 2) = 1$$

$$\gcd\left(2^{2n}, 2^{\tau/2}+1\right) = 1$$

Thus the elements of the moduli set $\{2^{\tau/2}-1, 2^{2n}, 2^{\tau/2}+1\}$ are pairwise relatively prime.

Substituting $m_1 = 2^{2n}$, $m_2 = 2^{\tau/2}-1$ and $m_3 = 2^{\tau/2}+1$ into (3.2) and (3.3), we obtain:

$$M_1 = 2^{\tau} - 1 \tag{3.5}$$

$$M_2 = 2^{2n}\left(2^{\tau/2}+1\right) \tag{3.6}$$

$$M_3 = 2^{2n}\left(2^{\tau/2}-1\right) \tag{3.7}$$

Theorem 2: Given the moduli set $\{2^{\tau/2}-1, 2^{2n}, 2^{\tau/2}+1\}$ and holding all conditions as specified above, the followings hold true:

$$M_1^{-1} = -1 \tag{3.8}$$

$$M_2^{-1} = 2^{\tau-2n-1} \tag{3.9}$$

$$M_3^{-1} = -2^{\tau-2n-1} \tag{3.10}$$

Proof:

Since:

$$|(-1) * (2^{\tau}-1)|_{2^{2n}} = 1 \; (|2^{\tau}|_{2^{2n}} = 0)$$

Then (3.8) holds true.

Similarly, since:

$$\left|2^{\tau-2n-1} * 2^{2n}\left(2^{\tau/2}+1\right)\right|_{2^{\tau/2}-1} = |2^{\tau}|_{2^{\tau/2}-1} = 1$$

Then (3.9) holds true.

Similarly, since:

$$\left|-2^{\tau-2n-1} * 2^{2n}\left(2^{\tau/2}-1\right)\right|_{2^{\tau/2}+1} = |2^{\tau}|_{2^{\tau/2}+1} = 1$$

Then (3.10) holds true.

Theorem 3: The binary equivalent X, of an RNS number (x_1, x_2, x_3) can be computed as follows:

$$X = \rho^{\tau}[0] \bowtie x_1^{2n}[0] \tag{3.11}$$

with,

$$\rho = \left|-2^{\tau-2n}x_1 + \left(2^{\tau/2}x_2 + x_2\right)2^{\tau-2n-1} + \left(1 - 2^{\tau/2}\right)2^{\tau-2n-1}x_3\right|_{2^{\tau}-1} \tag{3.12}$$

Proof: By substituting (3.5) through to (3.10) into (3.1) and factorizing 2^{2n} from the right hand side we obtain (3.11).

We can further simplify (3.12) as:

$$\rho = |\lambda_1 + \lambda_2 + \lambda_3|_{2^{\tau}-1} \tag{3.13}$$

with,

$$\lambda_1 = |-2^{\tau-2n}x_1|_{2^{\tau}-1} = \overline{x_1^{2n}[0]} \bowtie \mathcal{O}^{\tau-2n}[0] \tag{3.14}$$

$$\lambda_2 = \left|\left(2^{\frac{\tau}{2}}x_2 + x_2\right)2^{\tau-2n-1}\right|_{2^{\tau}-1} \tag{3.15}$$

$$= |2^{\tau-2n-1}(x_2 \bowtie x_2)|_{2^{\tau}-1}$$

$$= x_2^{2n+1-\tau/2}[0] \bowtie x_2 \bowtie x_2^{\tau-2n-1}[2n + 1 - \tau/2]$$

$$\lambda_3 = \left|\left(1 - 2^{\frac{\tau}{2}}\right)2^{\tau-2n-1}x_3\right|_{2^{\tau}-1} \tag{3.16}$$

$$= \left|2^{\tau-2n-1}x_3 + 2^{3\tau/2-2n-1}x_3\right|_{2^{\tau}-1}$$

$$= |\lambda_{31} + \lambda_{32}|_{2^{\tau}-1}$$

with,

$$\lambda_{31} = \left|2^{\frac{3\tau}{2}-2n-1}x_3\right|_{2^\tau-1} \tag{3.17}$$

$$= \left|2^{\tau-2n}\left(x_3 \bowtie \mathcal{O}^{\frac{\tau}{2}-1}[0]\right)\right|_{2^\tau-1}$$

$$= \overline{x_3^{2n-\frac{\tau}{2}+1}[0]} \bowtie \mathcal{O}^{\frac{\tau}{2}-1}[0] \bowtie \overline{x_3^{\tau-2n}[2n-\frac{\tau}{2}+1]}$$

$$\lambda_{32} = |2^{\tau-2n-1}x_3|_{2^\tau-1} \tag{3.18}$$

$$= \mathcal{Z}^{2n-\tau/2}[0] \bowtie x_3 \bowtie \mathcal{Z}^{\tau-2n-1}[0]$$

Substituting (3.16) into (3.13) we obtain (3.19):

$$\rho = |\lambda_1 + \lambda_2 + \lambda_{31} + \lambda_{32}|_{2^\tau-1} \tag{3.19}$$

3.3 Hardware Implementation

The schematic diagram for the proposed converter is showed in Figure 3.1. We use the method of Mathew *et al.* (2000) to compute ρ according to (3.19). By this method, two τ-bit CSAs and two τ-bit binary CPA's are required. The CPAs work in parallel (CPAs 2 and 3 in Figure 3.1). Whilst one of the CPAs has a constant carry-in of zero, the other has a constant carry-in of one. The correct result is selected by a multiplexer (MUX 1) based on the carry-out of CPA with constant carry-in of zero. This result is concatenated with x_1 at no hardware cost to realize the final binary equivalent of any RNS number.

The values of area and delay are presented in Table 3.1 and obtained as follows: it is worth noting following:

i. CSA 1 is made up of $\left(\frac{3\tau}{2} - 2n - 1\right)$-bit pairs of XNOR/OR gates since λ_1 and λ_{31} contain $(\tau - 2n)$-bit and $\left(\frac{\tau}{2} - 1\right)$-bit 1s all in different bit positions. Thus, this CSA demands an area of $\left(2n + 1 - \frac{\tau}{2}\right)\Delta_{FA}$ and imposes a delay of D_{FA}.

ii. CSA 2 is made up of $\left(\frac{\tau}{2} - 1\right)$-bit AND/OR pairs since λ_{32} contains $\left(\frac{\tau}{2} - 1\right)$-bit 0s. thus demands an area of $\left(\frac{\tau}{2} - 1\right)\Delta_{FA}$ and imposes a delay of D_{FA}.

iii. The CPAs each demand $(\tau)\Delta_{FA}$ and $(\tau)D_{FA}$.

iv. The proposed converters demand an area of $(2\tau + 2n)\Delta_{FA}$ and impose a delay of $(\tau + 2)D_{FA}$ since CPAs 1 and 2 are implemented in parallel.

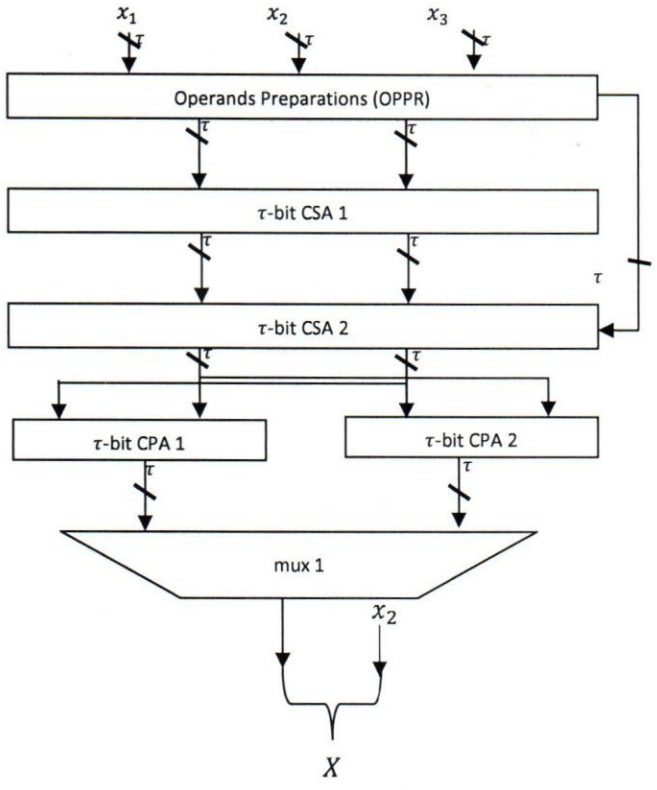

Figure 3.1: Schematic Diagram of Proposed Unified Converter

Table 3.1: Characterization of Each Part of The Proposed Reverse Converter

Component	Area(Δ_{FA})	Delay(D_{FA})
CSA 1	$2n + 1 - \frac{\tau}{2}$	1
CSA 2	$\frac{\tau}{2} - 1$	1
CPA 1	τ	τ
CPA 2	τ	τ
Total	$2\tau + 2n$	$\tau + 2$

3.4 Performance Evaluation

The major limiting factors of the moduli sets in Wang *et al.* (2002) and Hariri *et al.* (2007) compared to the ones proposed in this work is that, at equal DR our proposals present faster RNS arithmetic unit speed as depicted Table 3.2. In Table 3.2, proposed 1 and proposed 2 refer to the moduli sets $\{2^{n+1}-1, 2^{2n}, 2^{n+1}+1\}$ and $\{2^{(3n+2)/2}-1, 2^{2n}, 2^{(3n+2)/2}+1\}$ respectively. Additionally, theoretical analyses of converters for the proposed moduli sets and the state of the art converters in Wang *et al.* (2002) and Hariri *et al.* (2007) are presented in Table 3.3. Converters 1 and 2 in Table 3.3, refer to the converters for the moduli sets $\{2^{n+1}-1, 2^{2n}, 2^{n+1}+1\}$ and $\{2^{(3n+2)/2}-1, 2^{2n}, 2^{(3n+2)/2}+1\}$ respectively. Although faster and more area expensive adders can be used, this analysis considers as in Wang *et al.* (2002) and Hariri *et al.* (2007), that a n-bit CPA with End-Around Carry (EAC) has twice the delay of a normal n-bit CPA, but the same area. The EAC approach is an efficient method to compute modulo 2^n-1 addition, which consists in redirecting the resulting carry-out of an addition into the carry-in (Sousa *et al.* 2012). These results suggest the superiority of our schemes.

Table 3.2: Comparison of The Speed of The Different Moduli Sets

Moduli Set	Critical Path	Delay
Wang *et al.* (2002)	2^n+1	$2\log_2(n)+6$
Hariri *et al.* (2007)	$2^{2n}+1$	$2\log_2(n)+8$
Proposed 1	$2^{n+1}+1$	$2\log_2(n+1)+6$
Proposed 2	$2^{(3n+2)/2}+1$	$2\log_2(1.5n+1)+6$

Table 3.3: Area, Delay Comparison

Converter	DR	Area (Δ_{FA})	Delay (D_{FA})
Wang *et al.* (2002)	$3n$	$4n$	$4n + 2$
Hariri *et al.* (2007)	$5n$	$5n$	$8n + 1$
Converter 1	$4n + 2$	$6n + 2$	$2n + 4$
Converter 2	$5n + 2$	$8n + 4$	$3n + 4$

To fairly evaluate the performance of our schemes compared to the state of the art, a well known library of arithmetic units (Zimmermann, 1998), which contains a structural specification of optimized prefix adders written in synthesizable VHDL code was employed to obtain the HDL specification of both the proposed and related state of the art converters. These HDL specification of the converters were used to perform experimental assessment using Xilinx ISE 14.3 software to target a Spartan 3 FPGA. The results obtained after design place and route presented in terms of the number of FPGA slices and input-to-output propagation delays (in nano seconds) for various DR requirements (different values of n) are presented in Tables 3.4 and 3.5. It is worth noting that for every n in Table 3.4, the corresponding area and delay for Wang *et al.* (2002) and Hariri *et al.* (2007) are derived using the lower or upper bond of $(4n + 2)/3$ and $(4n + 2)/5$ respectively. Similarly, for every n in Table 3.5, the corresponding area and delay for Wang *et al.* (2002) is derived using the lower or upper bond of $(5n + 2)/3$. These

results suggest that at equal DR: The proposed converter for the moduli set $\{2^{n+1}-1, 2^{2n}, 2^{n+1}+1\}$ reduces the area demanded by the related state of the art converter in Wang *et al.* (2002) by about 58.03% and delay by about 63.45% and also reduces the area demanded by the converter in Hariri *et al.* (2007) by about 40.89% and delay by about 59.54% (see Table 3.4). These results further reveal that the proposed converter for the moduli set $\{2^{(3n+2)/2}-1, 2^{2n}, 2^{(3n+2)/2}+1\}$ reduces the area demanded by the related state of the art converter in Wang *et al.* (2002) by about 18.29% and delay by about 60.1%. This proposed converter also reduces the delay imposed by Hariri *et al.* (2007) by about 55.34% at an area cost of about 15.07% (see Table 3.5).

Table 3.4: Converters' Delay τ [ns] and Area [Number of Slices] ($\{2^{n+1}-1, 2^{2n}, 2^{n+1}+1\}$)

Converter	$n=2$		$n=4$		$n=8$		$n=16$	
	Δ	τ	Δ	τ	Δ	τ	Δ	τ
Wang *et al.* (2002)	20	15.4	45	24.8	105	27.3	215	33.8
Hariri *et al.* (2007)	13	11.5	27	13.9	56	14.1	113	15.3
Converter 2	21	14.9	52	18.0	111	22.1	236	22.9

Table 3.5: Converters' Delay τ [ns] and Area [Number of Slices] ($\{2^{(3n+2)/2}-1, 2^{2n}, 2^{(3n+2)/2}+1\}$)

Converter	$n=2$		$n=4$		$n=8$		$n=16$	
	Δ	τ	Δ	τ	Δ	τ	Δ	τ
Wang *et al.* (2002)	20	15.4	45	24.8	105	27.3	215	33.8
Hariri *et al.* (2007)	13	11.5	27	13.9	56	14.1	113	15.3
Proposed	21	14.9	52	18.0	111	22.1	236	22.9

Since the converter in Hariri *et al.* (2007) outperforms the proposed converter for the moduli set $\{2^{(3n+2)/2}-1, 2^{2n}, 2^{(3n+2)/2}+1\}$ in terms of area, we further compare these converters using the area delay square metric (ΔD^2). Results from this metric (shown in Table 3.6) suggest that in terms of overall performance, the proposed converter for the moduli set $\{2^{(3n+2)/2}-1, 2^{2n}, 2^{(3n+2)/2}+1\}$ is 78.64% better than the converter in Hariri *et al.* (2007).

Table 3.6: Converters' $\Delta D^2[10^3$ slices $ns^2]$

Converter	$n = 2$	$n = 4$	$n = 8$	$n = 16$
[4]	10	13	23	24
Converter 2	5	7	10	9

3.5 Conclusion

In this chapter, we proposed efficient residue-to-binary converters for the new moduli sets $\{2^{n+1}-1, 2^{2n}, 2^{n+1}+1\}$ and $\{2^{(3n+2)/2}-1, 2^{2n}, 2^{(3n+2)/2}+1\}$. We presented a unified architecture for reverse conversion in the two moduli sets by presenting a converter for the moduli set $\{2^{\tau/2}-1, 2^{2n}, 2^{\tau/2}+1\}$ with $\tau = 3n+2, 2n+2$. Experiments performed on the proposed converters and the related state of the art using Xilinx ISE 14.3 software to target a Spartan 3 FPGA suggest that the proposed converters outperform the related state of the art schemes.

CHAPTER FOUR

CONFIGURABLE RESIDUE-TO-BINARY CONVERTERS FOR THE MODULI SET $\{2^n-1, 2^{\mu n}, 2^n+1, 2^{2n+1}-1\}$

4.1 Introduction and Background

Recently, larger dynamic range moduli sets $\{2^n-1, 2^n, 2^n+1, 2^{2n+1}-1\}$ and $\{2^n-1, 2^n, 2^n+1, 2^{2n}+1\}$ were proposed in Molahosseini *et al.* (2010) with CRT II and CRT I based converters respectively. In Sousa and Antao (2012), an MRC divide and conquer approach was used to present converters for the moduli sets $\{2^n-1, 2^n, 2^n+1, 2^{2n+1}-1\}$ and $\{2^n-1, 2^{2n}, 2^n+1, 2^{2n+1}-1\}$. The converter for the $\{2^n-1, 2^n, 2^n+1, 2^{2n+1}-1\}$ moduli set outperformed the one in Molahosseini *et al.* (2010) whilst that of the $\{2^n-1, 2^{2n}, 2^n+1, 2^{2n+1}-1\}$ moduli set outperform the converter in Zhang and Siy (2008) for the moduli set $\{2^n-1, 2^n+1, 2^{2n}-1, 2^{2n+1}-3\}$. However, the converters in Sousa and Antao (2012) still exhibit some high area and delay characteristics. Therefore, in this brief we present comparatively faster and lower area cost CRT II based reverse converters for the $\{2^n-1, 2^{\mu n}, 2^n+1, 2^{2n+1}-1\}$ moduli set which is derived by unifying the moduli sets $\{2^n-1, 2^n, 2^n+1, 2^{2n+1}-1\}$ (Molahosseini *et al.* 2010) and $\{2^n-1, 2^{2n}, 2^n+1, 2^{2n+1}-1\}$ (Sousa and Antao, 2012).

The CRT II for four moduli sets is given as (Molahosseini *et al.* 2010):

$$X = Z + P_1P_2|k_1(Y-Z)|_{P_3P_4} \tag{4.1}$$

$$Z = x_1 + P_1|k_2(x_2-x_1)|_{P_2} \tag{4.2}$$

$$Y = x_3 + P_3|k_3(x_4 - x_3)|_{P_4} \quad (4.3)$$

with,

$$|k_1 P_1 P_2|_{P_3 P_4} = 1 \quad (4.4)$$

$$|k_2 P_1|_{P_2} = 1 \quad (4.5)$$

$$|k_3 P_3|_{P_4} = 1 \quad (4.6)$$

where P_i and k_i, $i = [1,4]$ are the moduli and multiplicative inverses respectively.

The remainder of this chapter is organized as follows. In Section 4.2, the new moduli sets and their respective reverse conversion algorithms are proposed. Section 4.3 presents the hardware implementation of the conversion techniques, while Section 4.4 gives a performance comparison with the similar best known state of the art converters and Section 4.5 concludes the chapter.

4.2.0 Proposed Converters

In this section, we present an efficient reverse converter for the moduli set $\{2^n - 1, 2^n + 1, 2^{\mu n}, 2^{2n+1} - 1\}$, with $\mu = 1,2$.

The following notations below are adopted from Sousa and Antao (2012) for presenting the proposed method and the corresponding reverse converter:

i. For an n-bit value γ, bits are referred from the Most Significant Bit (MSB) to the Least Significant Bit (LSB) as $\gamma[n-1], \ldots, \gamma[0]$.

ii. $\gamma^l[k]$ refers to an l-bit number such that: $\gamma^l[k] = \gamma[k + l - 1]2^{l-1} + \ldots + \gamma[k + 1]2 + \gamma[k]$;

iii. $\mathcal{O}$ and $\mathcal{Z}$ each refer to a number whose binary representation is an all-one and all-zero string, respectively;

iv. The symbol $\bowtie$ operates the concatenation of the binary representation of two numbers.

The following Lemmas are important in the design of the proposed converters:

Lemma 1: Modulo 2^s of a number is equivalent to s least significant bits of the number.

Lemma 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 $(2^s - 1)$ (Gbolagade *et al.* 2010).

Lemma 3: 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 (Gbolagade *et al.* 2010).

Lemma 4: Modulo a of an integer b of higher length than a (where a is n-bit long) can be expressed as modulo a of the sum of integers gotten by partitioning b into n-bit fields.

4.2.1 Proposed Method for Reverse Conversion

First, we showed that the computation of multiplicative inverses are unity and present a low complexity design which requires only carry save adders (CSAs) and carry propagate adders (CPAs).

Theorem 1: Given the moduli set $\{2^n - 1, 2^n + 1, 2^{\upsilon n}, 2^{2n+1} - 1\}$, where, $P_1 = 2^{\mu n}$, $\ell_2 = 2^{2n+1} - 1$, $P_3 = 2^n + 1$ and $P_4 = 2^n - 1$ for every integer $n > 1$, the followings hold true:

$$k_1 = 2^{(2-\mu)n} \tag{4.7}$$

$$k_2 = 2^{(2-\mu)n+1} \tag{4.8}$$

$$k_3 = 2^{n-1} \tag{4.9}$$

Proof:

Since:

$$\left|2^{(2-\mu)n} * 2^{\mu n}(2^{2n+1} - 1)\right|_{2^{2n}-1} = |2^{2n}(2^{2n+1} - 1)|_{2^{2n}-1}$$

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

Then (4.7) holds true.

Similarly, since:

$$\left|2^{(2-\mu)n+1} * 2^{\mu n}\right|_{2^{2n+1}-1} = |2^{2n+1}|_{2^{2n+1}-1} = 1$$

Then (4.8) holds true.

The proof for (4.9) was showed in Molahosseini *et al.* (2010).

By substituting (4.7), (4.8) and (4.9) and the given moduli into (4.1) through to (4.3), we have:

$$X = Z + 2^{\mu n}(2^{2n+1} - 1)\rho \tag{4.10}$$

$$Z = x_1 + 2^{\mu n}\omega \tag{4.11}$$

$$= \omega^{2n+1}[0] \bowtie x_1^{\mu n}[0]$$

$$Y = x_3 + (2^n + 1)\lambda \tag{4.12}$$

$$= x_3^{n+1}[0] + \lambda^n\,[0] \bowtie \lambda^n\,[0]$$

with,

$$\rho = \left|2^{(2-\mu)n}(Y - Z)\right|_{2^{2n}-1} \tag{4.13}$$

$$\omega = \left|2^{(2-\mu)n+1}(x_2 - x_1)\right|_{2^{2n+1}-1} \tag{4.14}$$

$$\lambda = |2^{n-1}(x_4 - x_3)|_{2^n-1} \tag{4.15}$$

We can further simplify (4.14), (4.15) and (4.13) as follows:

for (4.14):

$$\omega = \left|2^{(2-\mu)n+1} * \varepsilon\right|_{2^{2n+1}-1} \tag{4.16}$$

$$= \varepsilon^{\mu n}[0] \bowtie \varepsilon^{(2-\mu)n+1}[\mu n]$$

with,

$$\varepsilon = |x_2 - x_1 + m_2|_{2^{2n+1}-1} \tag{4.17}$$

$$= x_2^{2n+1}[0] + \mathcal{O}^{(2-\mu)n+1} \bowtie \overline{x_1^{\mu n}[0]} + \mathcal{O}^{2n+1}[0] + 1$$

for (4.15): Applying Lemma 4 will give us:

$$\lambda = \left|2^{n-1} * \left(x_4^n[0] + \overline{x_3^{n+1}[0]}\right)\right|_{2^n-1} \tag{4.18}$$

$$= |x_4[0] \bowtie x_4^{n-1}[1] + (x_{31}[0] \bowtie x_{31}^{n-1}[1])|_{2^n-1}$$

with,

$$x_{31} = \overline{x_3^n[0] \vee x_3[n]} \tag{4.19}$$

where ∨ means OR operation.

for (4.13): we substitute in (11) and (12) into (13) obtain:

$$\rho = \left|2^{(2-\mu)n} * \alpha\right|_{2^{2n}-1} = \alpha^{\mu n}[0] \bowtie \alpha^{(2-\mu)n}[(2-\mu)n] \tag{4.20}$$

with,

$$\alpha = |k_1 + k_2 + k_3 + k_4|_{2^{2n}-1} \tag{4.21}$$

$$k_1 = |x_3^{n+1}[0]|_{2^{2n}-1} = \mathcal{Z}^{n-1}[0] \bowtie x_3^{n+1}[0] \tag{4.22}$$

$$k_2 = |\lambda^n[0] \bowtie \lambda^n[0]|_{2^{2n}-1} = \lambda^n[0] \bowtie \lambda^n[0] \tag{4.23}$$

$$k_3 = \left|\overline{x_1^{\mu n}[0]}\right|_{2^{2n}-1} = \mathcal{O}^{(2-\mu)n}[0] \bowtie \overline{x_1^{\mu n}[0]} \tag{4.24}$$

$$k_4 = |-2^{\mu n}\omega|_{2^{2n}-1} = |-2^{\mu n}\pi|_{2^{2n}-1} \tag{4.25}$$

$$= \overline{\pi^{\mu n}[0]} \bowtie \overline{\pi^{(2-\mu)n}[n]}$$

$$\pi = |\omega^{2n+1}[0]|_{2^{2n}-1} = \omega^{2n}[0] \vee \omega[2n] \tag{4.26}$$

Theorem 2: Given the conditions in theorem 1 above, the binary equivalent X, of an RNS number (x_1, x_2, x_3, x_4) can be computed as follows:

$$X = \chi^{4n+1}[0] \bowtie x_1^{\mu n}[0] \tag{4.27}$$

with,

$$\chi = \rho^{2n}[0] \bowtie \omega^{2n+1}[0] + \mathcal{O}^{2n+1}[0] \bowtie \overline{\rho^{2n}[0]} + 1 \tag{4.28}$$

Proof:

Substituting (4.11) into (4.10) and simplifying, we obtain (4.27) as follows:

$$X = x_1 + 2^{\mu n}\omega + 2^{\mu n}(2^{2n+1} - 1)\rho \tag{4.29}$$

$$= x_1 + 2^{\mu n}(\omega + 2^{2n+1}\rho - \rho)$$

$$= x_1^{\mu n}[0] + \left(\rho^{2n}[0] \bowtie \omega^{2n+1}[0] + \mathcal{O}^{2n+1}[0] \bowtie \overline{\rho^{2n}[0]} + 1\right) \bowtie \mathcal{Z}^{\mu n}[0]$$

$$= x_1^{\mu n}[0] + \chi^{4n+1}[0] \bowtie \mathcal{Z}^{\mu n}[0]$$

$$= \chi^{4n+1}[0] \bowtie x_1^{\mu n}[0]$$

4.3 Hardware Implementation

The diagram of blocks in Figure 4.1 represents the proposed reverse converter architecture. Considering τ_{FA} and Δ_{FA} as the delay and area of a 1-bit Full Adder (FA), respectively, we consider also, as in Sousa and Antao (2012), that the delay of a Carry-Propagate Adder (CPA) with End-Around Carry (EAC) is twice the delay of a regular CPA at a similar hardware cost and show in Table 4.1 the values for delay and circuit area required by the proposed converters and the equivalent state of the art converters in Sousa and Antao (2012). Furthermore, we consider that the area and delay of a half adder (HA) are $\Delta_{HA} = \frac{1}{2}\Delta_{FA}$ and $\tau_{HA} = \frac{1}{2}\tau_{FA}$,

respectively. The bitwise operations are ignored for area and delay analysis, as they are expected to be negligible regarding the FAs and HAs.

From Figure 4.1, ε is computed according to (4.17) using $2n + 1$-bit CSA 1 with EAC and regular CPA 1. Since (4.17) has $2n + 1$-bit of 1s, CSA 1 is thus reduced into $2n + 1$-bit pairs of XNOR/OR gates. It is worth noting that the carry passed from this CSA to CPA 1 will always contain $(2 - \mu)n + 1$-bit of 1s and thus CPA 1 is made up of $(2 - \mu)n + 1$-bit HAs. Therefore the computation of ε imposes a delay of $(0.5(2 + \mu)n + 1.5)\tau_{FA}$ at the demand of $(0.5(2 + \mu)n + 0.5)\Delta_{FA}$ area. ω is computed by performing $(2 - \mu)n + 1$-bit left cyclic shifting on ε.

We use the method of Mathew *et al.* (2000) to compute λ according to (18). By this method, two n-bit binary CPA's work in parallel (CPAs 2 and 3 in Figure 4.1). Whilst CPA 2 has a constant carry-in of zero, CPA 3 has a constant carry-in of one. The correct result is selected by a multiplexer (MUX 1) based on the carry-out of CPA 2. Therefore, λ requires $(2n)\Delta_{FA}$ and $(n)\tau_{FA}$ to compute. To compute α from (4.21), again the method of Mathew *et al.* (2000) is used. This requires two CSAs (notably CSAs 2 and 3) , two CPAs (CPAs 4 and 5) all of length $2n$-bit, and mux 2. Since k_1 and k_4 in (22) have $(n - 1)$ and $(2 - \mu)n$-bit of 0s and 1s respectively, CSAs 2 and 3 are made up of $(n - 1)$ and $(2 - \mu)n$-bit pairs of AND/OR and XNOR/OR gates respectively. As a result, $\big((\mu + 5)n + 1\big)\Delta_{FA}$ and $(2n + 2)\tau_{FA}$ are required to compute α. A

$(2-\mu)n$-bit left cyclic shifting is performed on α to realize ρ. χ is computed from (4.28) using $(4n+1)$-bit regular CPA 6. It is worth noting that, CPA 6 is made up of $2n+1$-bit HAs since one of its operands has $2n+1$-bit 1s. As a result, the computation of χ demands $(3n+0.5)\Delta_{FA}$ area and imposes a delay of $(3n+0.5)\tau_{FA}$. Finally, the binary equivalent X of an RNS number is computed from (4.27) as χ concatenation x_1 at no hardware cost.

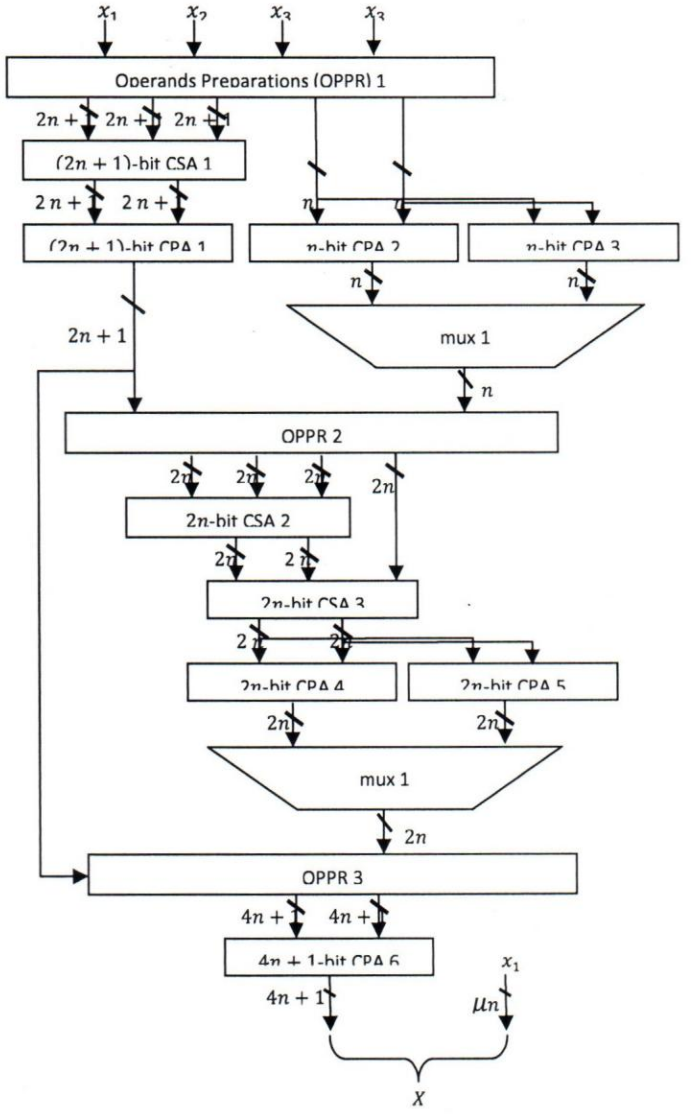

Figure 4.1: Schematic Diagram of Proposed Unified

The values of area and delay in Table 4.1 are obtained as follows: The total area required for the proposed unified converter is the cumulative area required to compute $\varepsilon, \lambda, \alpha$ and χ, which amounts to $\big((11 + 1.5\mu)n + 4\big)\Delta_{FA}$. Therefore, the proposed converter for the moduli set $\{2^n - 1, 2^n + 1, 2^n, 2^{2n+1} - 1\}$ (i.e. when $\mu = 1$) demands an area of $(12.5n + 4)\Delta_{FA}$ whilst the proposed converter for the moduli set $\{2^n - 1, 2^n + 1, 2^{2n}, 2^{2n+1} - 1\}$ (i.e. when $\mu = 2$) demands an area of $(14n + 4)\Delta_{FA}$. On the other hand, the delay of the proposed unified moduli set is $\big((6 + 0.5\mu)n + \mu + 2\big)\tau_{FA}$, which is the cumulative delay required to compute ε, λ and χ since the delay to compute λ is hidden in that required to compute ε. Therefore, the proposed converter for the moduli set $\{2^n - 1, 2^n + 1, 2^n, 2^{2n+1} - 1\}$ imposes a delay of $(6.5n + 3)\tau_{FA}$ whilst the proposed converter for the moduli set $\{2^n - 1, 2^n + 1, 2^{2n}, 2^{2n+1} - 1\}$ imposes a delay of $(7n + 4)\tau_{FA}$.

Table 4.1: Area, Delay Comparison

Moduli Set	Converter	Area (Δ_{FA})	Delay (τ_{FA})
$\{2^n - 1, 2^n + 1, 2^n, 2^{2n+1} - 1\}$	Sousa and Antao (2012)	$13n + 2$	8n+1
	Our Work	$12.5n + 4$	6.5n+3
$\{2^n - 1, 2^n + 1, 2^{2n}, 2^{2n+1} - 1\}$	Sousa and Antao (2012)	$16n + 1$	8n+2
	Our Work	$14n + 4$	7n+4

4.4 Performance Evaluation

To evaluate the performance of the proposed residue to binary converters, we compare our proposals with the related state of the art converters in

Sousa and Antao (2012). From the theoretical analysis presented in Table 4.1, the proposed converters outperform the related state of the art in terms of both area and delay. It is also worth noting that the proposed converters does not require the use of modular adders as opposed to the mod-$2^{2n}-1$ arithmetic required by the converters in Sousa and Antao (2012).

A well known library of arithmetic units (Zimmermann, 1998), which contains a structural specification of components, namely optimized prefix adders written in synthesizable VHDL code (Sousa *et al.* 2012), was employed to obtain the HDL specification of both the proposed and related state of the art converters. Using the HDL specification of these converters, experimental assessment was carried out using Xilinx ISE 14.3 software to target a Spartan 3 FPGA. The results obtained after design place and route are given in terms of the number of FPGA slices and input-to-output propagation delays (in nano seconds) for various DR requirements (different values of n) are presented in Table 4.2. These results suggest that, the proposed converter for the moduli set $\{2^n-1, 2^n+1, 2^n, 2^{2n+1}-1\}$ improves the area demanded by the related state of the art converter in Sousa and Antao (2012) by about 29.09% and delay by about 46.33%. On the other hand, the proposed converter for the moduli set $\{2^n-1, 2^n+1, 2^{2n}, 2^{2n+1}-1\}$ improves the area demanded by the related state of the art converter in Sousa and Antao (2012) by about 24.50% and delay by about 43.45%. It is also worth nothing that this proposed converter outperforms the state of the art converter for the

comparatively n-bit less DR moduli set $\{2^n - 1, 2^n + 1, 2^n, 2^{2n+1} - 1\}$ with about 19.85% and 46.84% in terms of area and delay respectively.

Table 4.2: Converters' Delay τ [ns] and Area [Number of Slices]

Moduli Set	Converter	$n = 2$		$n = 4$		$n = 8$		$n = 16$	
		Δ	τ	Δ	τ	Δ	τ	Δ	τ
$\{2^n - 1, 2^n + 1, 2^n, 2^{2n+1} - 1\}$	Sousa and Antao (2012)	27	23.6	67	33.8	138	41.3	287	58
	Our Work	24	18.4	52	22	100	21.6	192	22.6
$\{2^n - 1, 2^n + 1, 2^{2n}, 2^{2n+1} - 1\}$	Sousa and Antao (2012)	30	22.1	65	35.6	147	36.6	309	53
	Our Work	23	18.3	56	21.8	109	21	228	22.2

4.5 Conclusion

In this chapter, we proposed an efficient residue-to-binary converter for the moduli set $\{2^n - 1, 2^{\mu n}, 2^n + 1, 2^{2n+1} - 1\}$ for $\mu = 1,2$. Experimental results suggest that, the proposed converter for the moduli set $\{2^n - 1, 2^n + 1, 2^n, 2^{2n+1} - 1\}$ improves the area and delay of the related state of the art converter in Sousa and Antao (2012) by about 29.09% and 46.33% respectively, while the proposed converter for the moduli set $\{2^n - 1, 2^n + 1, 2^{2n}, 2^{2n+1} - 1\}$ improves the area and delay of the related state of the art converter in Sousa and Antao (2012) by about 24.50% and 43.45% respectively. These results also suggest that the proposed converter for the moduli set $\{2^n - 1, 2^n + 1, 2^{2n}, 2^{2n+1} - 1\}$ outperforms the state of the art converter for the comparatively n-bit less DR moduli set $\{2^n - 1, 2^n + 1, 2^n, 2^{2n+1} - 1\}$ with about 19.85% and 46.84% in terms of area and delay respectively.

CHAPTER FIVE

CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusions

In this thesis, our focus was first on proposing new and efficient moduli sets, then building efficient residue to binary converters for these and existing moduli sets.

This is an important issue, since, the superiority of Residue Number System (RNS) based processors is mainly based on the combined effect of hardware required by both the moduli selected and reverse converter. This work is also significant since many other RNS difficult computations such as; overflow detection, magnitude comparison, sign detection, etc., largely depend on moduli selection and reverse conversion.

The remainder of this chapter is organized as follows. Section 5.2 summarizes the work and results presented in this thesis. Section 5.3 outlines the major contributions of this thesis, while in Section 5.4 we present some recommendations which will further improve the RNS arithmetic domain when investigated.

5.2 Summary

i. In Chapter two, we presented two new moduli sets $\{2^{2n+1}-1, 2^n, 2^{n+1}-1\}$ and $\{2^{2n+1}-1, 2^{2n}, 2^{n+1}-1\}$ for RNS. Consequently, residue to binary converters for these moduli sets were proposed based on Mixed Radix Conversion (MRC) technique. We unify the architecture for reverse conversion in the

proposed moduli sets by presenting a configurable converter for the moduli set $\{2^{2n+1}-1, 2^{\mho n}, 2^{n+1}-1\}$ for values of $\mho = 1,2$. We showed that, the computation of multiplicative inverses could be avoided and presented low complexity converters that did not require the explicit use of modulo operations as opposed to mods- $2^{2n}-1$, $2^{2n+1}-1$ and $2^{4n}-1$ required by the state of the art. The proposed converters and the state of the art converters were implemented on Xilinx Xc4vlx15-10sf363 FPGA. The implementation results suggest that, on the average, the converter for the proposed moduli set $\{2^{2n+1}-1, 2^{n}, 2^{n+1}-1\}$ improved area and delay by about 50.65% and 49.16%, respectively when compared to the related state of the art converter in Molahosseini *et al.* (2008), while the converter for the proposed moduli set $\{2^{2n+1}-1, 2^{2n}, 2^{n+1}-1\}$ improved area and delay required by the state of the art converter for the moduli set $\{2^{n}-1, 2^{n}, 2^{n}+1, 2^{2n+1}-1\}$ by about 39.03% and 44.53% respectively and also improved the area and delay required by the state of the art converter for the moduli set $\{2^{n}-1, 2^{n}, 2^{n}+1, 2^{2n}+1\}$ by about 53.03% and 32.40% respectively.

ii. In chapter three, we proposed another pair of new moduli sets $\{2^{n+1}-1, 2^{2n}, 2^{n+1}+1\}$ and $\{2^{(3n+2)/2}-1, 2^{2n}, 2^{(3n+2)/2}+1\}$ with unified reverse conversion architecture. We showed that, the

computation of multiplicative inverses could be avoided and presented low complexity converters that did not require the explicit use of modulo operations as opposed to mods-$2^{2n}-1$ and $2^{4n}-1$ required by the state of the art. Experiments were performed on our converters and the state of the art using Xilinx ISE 14.3 software to target a Spartan 3 FPGA board. These experimental results suggest that on the average, at equal DR, the proposed converter for the moduli set $\{2^{n+1}-1, 2^{2n}, 2^{n+1}+1\}$ improved the area demanded by the related state of the art converter in Wang *et al.* (2002) by about 58.03% and delay by about 63.45%. This proposed converter also improved the area demanded by the converter in Hariri *et al.* (2007) by about 40.89% and delay by about 59.54%. These results further showed that the proposed converter for the moduli set $\{2^{(3n+2)/2}-1, 2^{2n}, 2^{(3n+2)/2}+1\}$ on its part, improved the area demanded by the related state of the art converter in Wang *et al.* (2002) by about 18.29% and delay by about 60.1% and delay imposed by Hariri *et al.* (2007) by about 55.34% with 78.64%.

iii. In chapter Four, we proposed efficient residue-to-binary converters for existing moduli sets $\{2^n-1, 2^n, 2^n+1, 2^{2n+1}-1\}$ and $\{2^n-1, 2^{\mu n}, 2^n+1, 2^{2n+1}-1\}$. We unified the architecture for reverse conversion in these moduli sets to realize a confugurable converter for the moduli set $\{2^n-1, 2^{\mu n}, 2^n+1, 2^{2n+1}-1\}$ with

$\mu = 1,2$. The proposed converters did not require the explicit use of modulo operations as opposed to the mod-$2^{2n} - 1$ required by the related state of the art converters. Theoretical analyses of hardware resource and conversion delay suggest that, the proposed scheme outperforms the equivalent state of the art converters. To validate these analyses, both the proposed converters and the state of the art were implemented using Xilinx ISE 14.3 software to target a Spartan 3 FPGA board. Experimental results suggest that compared to the state of the art related converters in Sousa and Antao (2012), the proposed converter improves area and delay by about 29.09% and 46.33% and about 24.50% and 43.45% for $\mu = 1$ and $\mu = 2$ respectively. Additionally, the proposed converter for $\mu = 2$ outperforms the state of the art related converter for the comparatively lower dynamic range moduli set $\{2^n - 1, 2^n, 2^n + 1, 2^{2n+1} - 1\}$ in Sousa and Antao (2012) with about 19.85% and 46.84% area and delay improvements respectively.

5.3 Major Contributions

The major contributions of this study can be summarized by the following:

i. We proposed the moduli set $\{2^{2n+1} - 1, 2^n, 2^{n+1} - 1\}$. We simplified the MRC to obtain a modulo operation free converter

opposed to the mod-$2^{2n+1}-1$ required by the proposal in Mohosseini *et al.* (2008) respectively.

ii. We proposed the moduli set $\{2^{2n+1}-1, 2^{2n}, 2^{n+1}-1\}$ with faster RNS arithmetic speed compared with Sousa and Antao (2012) and Cao *et al.* (2003). We simplified the MRC to obtain a modulo operation free converter opposed to the mods-$2^{2n}-1$, and $2^{4n}-1$ required by the proposals in Sousa and Antao (2012) and Cao *et al.* (2003) respectively.

iii. We proposed the moduli set $\{2^{n+1}-1, 2^{2n}, 2^{n+1}+1\}$ with faster RNS arithmetic speed compared with Wang *et al.* (2012) and Hariri *et al.* (2007) at equal Dynamic Range (DR). We simplified the CRT to obtain a modulo operation free converter opposed to the mods-$2^{2n}-1$, and $2^{4n}-1$ required by the proposals in Wang *et al.* (2012) and Hariri *et al.* (2007) respectively.

iv. We proposed the moduli set $\{2^{(3n+2)/2}-1, 2^{2n}, 2^{(3n+2)/2}+1\}$ with faster RNS arithmetic speed compared with Wang *et al.* (2012) and Hariri *et al.* (2007) at equal DR. We simplified the CRT to obtain a modulo operation free converter opposed to the mods-$2^{2n}-1$, and $2^{4n}-1$ required by the proposals in Wang *et al.* (2012) and Hariri *et al.* (2007) respectively.

v. We proposed an efficient reverse converter for the moduli set $\{2^n-1, 2^n, 2^n+1, 2^{2n+1}-1\}$. We simplified the CRT-II to

obtain a modulo operation free converter opposed to the mod-$2^{2n}-1$ by the proposals in Sousa and Antao (2012).

vi. We proposed an efficient reverse converter for the moduli set $\{2^n-1, 2^{2n}, 2^n+1, 2^{2n+1}-1\}$. We simplified the CRT-II to obtain a modulo operation free converter opposed to the mod-$2^{2n}-1$ by the proposals in Sousa and Antao (2012).

5.4 Recommendations

In this thesis, four new moduli sets and six reverse converters have been proposed. Moduli selection and efficient converter design are a major step in the improvement of RNS based special purpose processors and realization of RNS based general purpose processors. This section presents some recommendations that could further improve RNS usage in both special and general purpose processors:

i. Since special purpose processors are build using the related state of the art schemes in literature, haven proofed in this work that all proposed moduli sets present faster arithmetic channel speed compared to the related state of the art, we recommend that the moduli sets proposed in this work be used in such processors instead of the related state of the art.

ii. We also recommend that since the majority of the algorithms for performing difficult RNS operations such as overflow and sign detection, division, magnitude comparison, etc., are based on

reverse conversion, it will be interesting to re-design existing algorithms for resolving these difficult RNS operations using the efficient reverse converters proposed in this thesis.

iii. Furthermore, we recommend the proposal and investigation of more parallelized moduli sets (beyond the three modulus moduli sets proposed in this work) yet attempting to eliminate the modulo $2^n + 1$ arithmetic (since it is well known for degrading the entire RNS processor in terms of both area and delay).

REFERENCES

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

Bhardwaj, M., Srikanthan, T., and Clarke, C. T., (1999). A reverse converter for the 4 moduli super set $\{2^n - 1, 2^n, 2^n + 1, 2^{n+1} + 1\}$ *IEEE Conference on Computer Arithmetic*.

Cao, B., Chang,C. H., 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. Circuits Syst. I*, Reg. Papers, **50**: 1296–1303.

Cao, B., Chang,C. H., and Srikanthan,T., (2007). A residue-to-binary converter for a new five-moduli set, *IEEE Trans. Circuits Syst. I*, Reg. Papers, **54**: 1041–1049.

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

Gbolagade, K. A., (2012). An Efficient MRC based RNS-to-Binary Converter for the $\{2^{2n+1} - 1, 2^n, 2^{2n} - 1\}$ moduli set, *AIMS*.

Gbolagade, K.A., and Cotofana, S. D., (2008). MRC Technique for RNS to Decimal Conversion for the Moduli Set

{2n + 2, 2n + 1, 2n}, *16th Annual Workshop on Circuits, Systems, and Signal Processing*, 318-321.

Gbolagade, K.A., and Cotofana, S. D., (2009). An O(n) Residue Number System to Mixed Radix Technique, *IEEE International Symposium on Circuits and Systems (ISCAS)*, 521-524.

Gbolagade, K.A., (2010). Effective Reverse Conversion in Residue Number System Processors, *PhD Theses, TU DELFT*, Netherlands.

Gbolagade, K.A., Chaves, R., Sousa, L., and Cotofana, S. D., (2010). 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)*, 2103-2106, Paris.

Gbolagade, K.A., Chaves, R., Sousa, L., and Cotofana, S. D., (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*, pp.26-33.

Gbolagade, K.A., Voicu, G. R.,, and Cotofana, S. D., (2010). An Efficient FPGA Design of Reverse Converter for the Moduli Set{2n + 2, 2n + 1, 2n}, *5th International Summer School on Advanced Computer Architecture and Compilation for Embedded Systems (ACACES)*, 117-120.

Leone,l Sousa and Samuel, Antao, (2012). MRC-Based RNS reverse converters for the Four Moduli sets $\{2^n+1, 2^n-1, 2^n, 2^{2n+1}-1\}$

and $\{2^n + 1, 2^n - 1, 2^{2n}, 2^{2n+1} - 1\}$, *IEEE Transactions on Circuits and Systems*, **9-II (4)**: 244-248.

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.INF. and SYST.*, **E91-D,7**: 2058-2060.

Mathew, J., Radhakrishnan, D., Srikanthan, T., (2000). Fast Residue-to-Binary Converter Architecture, *IEEE*, Las Cruces, NM, USA.

Mohan, P.V.A., and Premkumar, A. B, (2007). RNS-to-binary converters for two four moduli $\{2^n - 1, 2^n, 2^n + 1, 2^{n+1} - 1\}$ and $\{2^n - 1, 2^n, 2^n + 1, 2^{n+1} + 1\}$, *IEEE Trans. Circuits Syst. I, Reg. Papers*, **54**: 1245–1254.

Mohan, P.V.A., (2007). RNS-to-binary converter 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**: 775-779.

Molahosseini, A.S., Dadkhah, C., Navi, K., (2009). A New Five-Moduli Set for Efficient Hardware Implementation of the Reverse Converter,. *IEICE Electronics Express*, **6**: 1006-1012.

Molahosseini, A.S., Navi, K., and M.K. Rafsanjani, M. K., (2008). A New residue to binary converter based on mixed-radix conversion, *3rd International Conference On Information and Communication Technologies: From Theory to Applications* ***(ICTTA)***: 1-6.

Molahosseini, A.S., (2011). Improving the Delay of Residue-to-Binary for a Four-Moduli Set, *Advances in Electrical and Computer Engineering.*

Molahosseini, A.S., 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 $\{2^n - 1, 2^{2n}, 2^n + 1, 2^{2n} + 1\}$Based on New CRTs, *IEEE Transactions on Circuits and Systems I*: Regular Papers, **57**: 823–835.

Parhami, B., (2010). Computer Arithmetic Algorithms and Hardware Designs, Oxford University Press, New York.

Piestrak, S., (1995). A high-speed realization of a residue to binary number system converter, *IEEE Trans. Circuits Syst. II*, **42**.

Rouhifar, M., Hosseinzadeh, M., and Teshnehlab, M., (2011). A new approach to Overflow detection in moduli set $\{2^n, 2^n\text{-}1, 2^{n-1}\text{-}1\}$, *International Journal of Computational Intelligence and Information Security*, **2**: 35-43.

Rouhifar, M., Hosseinzadeh, M., Bahanfar S., and Teshnehlab, M., (2011). Fast Overflow Detection in Moduli Set $\{2^n - 1, 2^n, 2^n + 1\}$, *International Journal of Computer Science Issues*, **(8/3)**: 407-414.

Sousa, L. ; Antão, S. and Chaves, R., (2012). On the Design of RNS Reverse Converters for the Four-Moduli Set $\{2^n - 1, 2^n, 2^n +$

$1, 2^{n+1} + 1\}$, *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*.

Taheri,M. R., Pirhoseinlo, A., Esmaeildoust, M., Esmaeildoust, M., Navi, K., (2012) High Speed Reverse Converter For High Dynamic Range Moduli Set, *International Journal of Advances in Engineering & Technology*, 26-37.

Theodore, L., H., (1989). Residue Addition Overflow Detection Processor, Boing Company, Seatle, Wash, **414276**.

Theodore, L., H., (1990). Method and Apparatus for Pipelined detection of overflow in Residue Arithmetic Multiplication, Boing Company, Seatle, Wash, **472,237**.

Vinod, A. P.,and Premkumar,A. B., (2000). A residue to binary converter for the 4-moduli superset $\{2^n - 1, 2^n, 2^n + 1, 2^{n+1} - 1\}$, *J. Circuits Syst. Comput.*,**10**: 85–99.

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 Transactions on Signal Processing*, **50**: 1772–1779.

Zhang, W., and Siy, P., (2008). An efficient design of residue to binary converter for four moduli set $\{2^n - 1, 2^n + 1, 2^{2n} - 2, 2^{2n+1} - 3\}$ based on new CRT II, *Information Sciences*, **178**: 264–279.

Zimmermann, R., (1998). VHDL Library of Arithmetic Units, in *International Forum on Design Languages - FDL*. Lausanne, **FDL Report**: 1–6.