Design and Implementation of Modified Sequential Parallel RNS Forward Converters

K Anitha[1], T S Arulananth[1], R Karthik[1], P Bhaskara Reddy[1]

[1]Department of Electronics and Communication Engineering, MLR Institute of Technology, Hyderabad -43, India.

[1]Orcid: 0000-0002-1685-8971

Abstract

The Residue Number System (RNS) is suitable for DSP architectures because of its ability to perform fast carry-free arithmetic. However, this advantage is over-shadowed by the complexity involved in the conversion of numbers between binary to RNS representations and have prevented the wide-spread use of RNS. Converting a number from a binary representation to its RNS equivalent is known as forward conversion while the inverse operation is called reverse conversion. Even though reverse conversion is generally more complex, forward conversion for arbitrary modulo sets is not simpler. However, forward conversion for arbitrary modulo sets is memory intensive. There are three main approaches for forward conversion. The first approach involves pre-computing all possible values that the conversion requires and storing these values in memory. The second approach involves using efficient arithmetic units called combinational logic along with memory (LUT). In both cases, the memory size requirement increases as the dynamic range increases. The third approach is memory less in that it involves only combinatorial logic in the design. In this thesis we proposed to four different architectures for second approach which uses combinational logic along with memory (LUTs). The first architecture is purely sequential conversion in the second architecture is combination of sequential and parallel. The third architecture is the modified version of second architecture and fourth architecture is purely parallel. Forward converter architecture is designed and implemented to reduce the area, speed. Verilog HDL is used for coding and implementation of different architectures has been done on XILINX VERTEX 5XC5VLX110T-2FF1136 OPEN SPARC board. It is been identified that modified sequential /parallel approach has better performance in speed and area when compared with existing architectures.

Keywords: Residue number system, digital signal processing, modular arithmetic3, moduli set, dynamic range, binary to RNS converter, RNS-based application, parallel processing.

INTRODUCTION

The number yields the remainders 2, 3, and 2 when divided by 3, 5, and 7, respectively? In modern terminology, 2, 3, and 2 are *residues*, and 3, 5, and 7, are *moduli*. Sun Tzu gave a rule, the *Tai-Yen* (*Great Generalization*) for the solution of his puzzle. In 1247, another Chinese mathematician, Qin Jiushao, generalized the Great Generalization into what we now call the *Chinese Remainder Theorem*, a mathematical jewel.

In the 1950s, RNS were rediscovered by computer scientists, who sought to put them to use in the implementation of fast arithmetic and fault tolerant computing. Three properties of RNS make them well suited for these. The first is absence of carry-propagation in addition and multiplication, carry-propagation being the most significant speed-limiting factor in these operations. The second is that because the residue representations carry no weight-information, an error in any digit-position in a given representation does not affect other digit-positions. And the third is that there is no significance-ordering of digits in an RNS representation, which means that faulty digit-positions may be discarded with no effect other than a reduction in dynamic range.

The new interest in RNS [2] was not long-lived, for three main reasons: One, a complete arithmetic unit should be capable of at least addition, multiplication, division, square-root, and comparisons, but implementing the last three in RNS is not easy; two, computer technology became more reliable; and, three, converting from RNS notation to conventional notation, for \human consumption", is difficult. Nevertheless, in recent years there has been renewed interest in RNS. There are several reasons for this new interest, including the following. A great deal of computing now takes place in embedded processors, such as those found in mobile devices, and for these high speed and low-power consumption are critical; the absence of carry-propagation facilitates the realization of high-speed, low-power arithmetic. Also, computer chips are now getting to be so dense that full testing will no longer be possible; so fault-tolerance and the general area of computational integrity have again become more important. Lastly, there has been progress in the implementation of the difficult arithmetic operations. True, that progress has not been of an order that would justify a deluge of letters home; but progress is progress, and the proper attitude should be gratitude for whatever we can get. In any case, RNS is extremely good for many applications-such as digital signal processing, communications engineering, computer security (cryptography), image processing, speech processing, and transforms in which the critical arithmetic operations such as addition and multiplication.

LITERATURE SURVEY

i. Residue Number Systems (RNS) were invented by the third-century Chinese scholar Sun Tzu-a different Sun Tzu from the author of the famous *Art of War*.

ii. A formal design methodology is used to design an optimal architecture for the residue decoding process is obtained. The architecture is modular, consists of simple cells, and is general for any set of moduli.[5].

iii. Amos Omondi explained the basics of forward converter and different architectures in'Residue Number System theory and implementation'.

OBJECTIVES OF THIS WORK

i. The main objective of this work is, designing, simulation and FPGA implementation of RNS based building blocks for applications in the field of DSP .

ii. Since the RNS results in carry free arithmetic operations and supports high-speed concurrent computations, it will be useful to use RNS-based building blocks for DSP applications.

iii. Other objective of this work is improving building blocks by developing new algorithms and improving existing ones.

iv. Studying different moduli sets, analyzing the relation between moduli number and the dynamic range it provides, and evaluating the most efficient ones for different applications with different dynamic range requirements.

v. Verifying the functionality and efficiency of the proposed designs and comparing them against other published ones based on FPGA implementation.

MODULI SET SELECTION

Choosing a proper modulo set is an essential issue for building an efficient RNS with a sufficient dynamic range (DR). The number, form and value of the moduli affect the dynamic range, timing performance and hardware complexity of an RNS-based application [3].

The moduli set in the RNS can be either arbitrary or special. In principal, special moduli sets were suggested in order to simplify the implementation of arithmetic operations. This invariably means that arithmetic on residue digits should not deviate too far from conventional arithmetic, which is just arithmetic modulo a power of two [1]. On the other hand, arithmetic circuits based on arbitrary moduli sets are much more complex and time consuming. These sets are utilized in cases when using special moduli sets imposes some constraints.

The most famous moduli set is {2n – 1, 2n, 2n + 1} [4]. This set has been known as a means of simplifying the calculations necessary to implement the reverse converter (RC). However, this set has modulo (2n + 1) channel that represents the bottleneck of the system. Its arithmetic circuits suffer from the longest delay among all three channels.[6]

In general, arithmetic circuits modulo (2k – 1) are more efficient than those modulo (2k + 1), therefore, it is better to reduce the number of moduli of the form (2k + 1)

Table illustrates the most recently published moduli sets, including the dynamic ranges they provide and possible n values that can be used in these sets.

Table 1: Moduli sets

Number of Moduli	Modulo set	Dynamic range	n odd/even
Three Moduli sets	{2n – 1, 2n , 2n + 1} [4]	3n	any
	{2n–1 – 1, 2n – 1, 2n } [6]	3n – 1	any
	{2n – 1, 2n , 2n+1 – 1} [7]	3n + 1	any
	{2n – 1, 2n , 22n+1 – 1} [8]	4n + 1	any
	{2n – 1, 2n + 1, 22n + 1} [9]	4n	any
	{2n , 22n – 1, 22n + 1} [10]	5n	even
Four moduli sets	{2n – 1, 2n , 2n + 1, 2n+1 – 1} [11]-I	4n + 1	even
	{2n – 1, 2n , 2n + 1, 2n+1 + 1} [11]-II	4n + 1	odd
	{2n/2 – 1, 2n/2 + 1, 2n + 1, 22n+1 – 1} [18]	4n + 1	even
	{2n – 1, 2n , 2n + 1, 22n + 1} [12]	5n	any
	{2n – 1, 2n , 2n + 1, 22n+1 – 1} [13]-I	5n + 1	any
	{2n – 1, 22n , 2n + 1, 22n + 1} [13]-II	6n	any
	any {2n – 1, 2n + 1, 22n – 2, 22n+1 – 3} [15]	6n + 1	any
	any {2n + 1, 2n – 1, 22n , 22n+1 – 1} [19]	6n + 1	any
	{22n+1, 22n + 1, 2n + 1, 2n – 1} [20]	6n + 1	any
Five moduli sets	{2n , 2n/2 – 1, 2n/2 + 1, 2n + 1, 22n–1 – 1} [17]	5n – 1	even
	{2n – 1, 2n , 2n + 1, 2n – 2 (n+1)/2 + 1, 2 n + 2(n+1)/2 + 1} [14]	5n	odd
	{2n – 1, 2n , 2n + 1, 2n–1 – 1, 2n+1 + 1} [16]	5n even	5n even

PROPOSED WORK

The modified sequential parallel RNS Forward converter is proposed in our work. Overall, the proposed architectures facilitate the implementation of RNS based processors by reducing the latency and complexity introduced by the binary stage. This makes it more possible and more practical to build effective RNS based processors.

Modulo channel

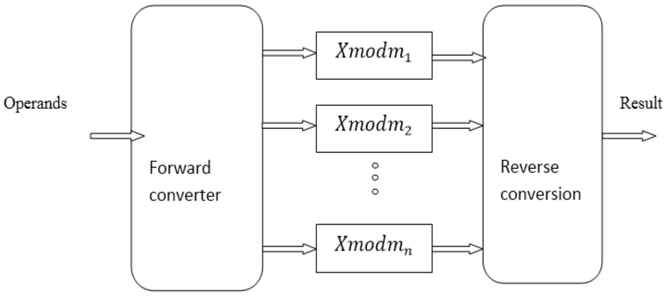

Figure 1. Block diagram of a typical RNS system

- Arithmetic operations based on residue number systems (RNS) can be carried out without intermediate carry digits.
- Converting a number from a decimalor binary representation to its RNS equivalent is known as forward conversion while the inverse operation is called reverse conversion.
- Even though reverse conversion is generally more complex, forward conversion for arbitrary moduli sets is not simpler. For special moduli sets of the 2n type, forward converters require only modular adders and therefore can be easily implemented.
- However, forward conversion for arbitrary moduli sets is memory intensive. There are three main The first approach involves precomputing all possible values that the conversion requires and storing these values in memory.
- The second approach involves using efficient arithmetic units called processing elements (PE) along with some memory. In both cases, the memory size requirement increases as the dynamic range increases.
- The third approach is memory less in that it involves only combinatorial logic in the design. A framework for memory less forward conversion has recently been introduced.

RESULTS AND DISCUSSIONS

From the table2, figure2 and figure 3 we can summarize many parameters of all four algorithms such as LUT's number of flip flops and maximum operating frequencies.

Table 2: Comparison of various Architectures

S.No	Architectures names.	No of slice LUTs.	No of Flip Flops.	Maximum operating Frequency (MHZ)
1	Sequential	56	56	129
2	Sequential and Parallel	42	44	296
3	Modified Sequential and Parallel	22	26	321
4	Parallel	33	67	228

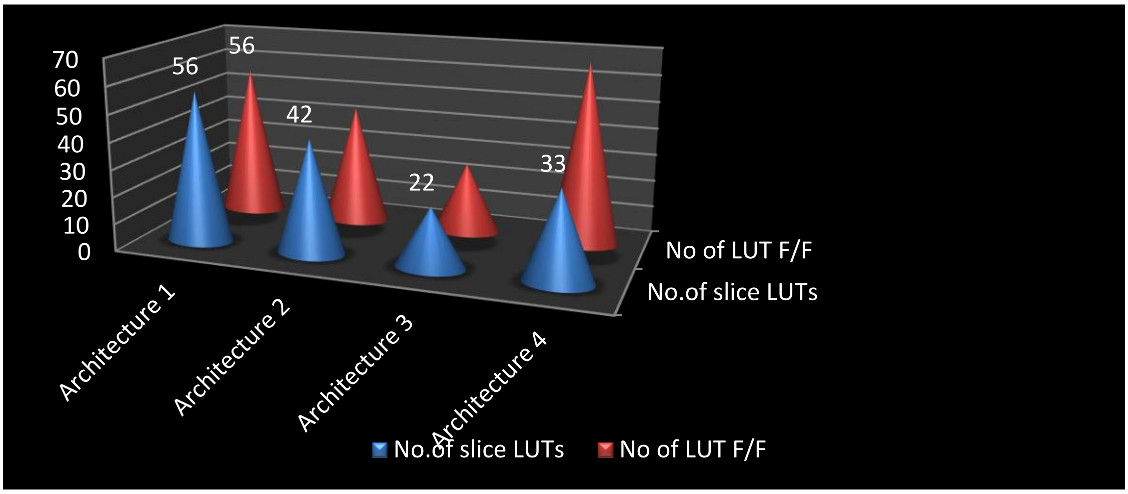

Figure 2: Comparison of various Architectures

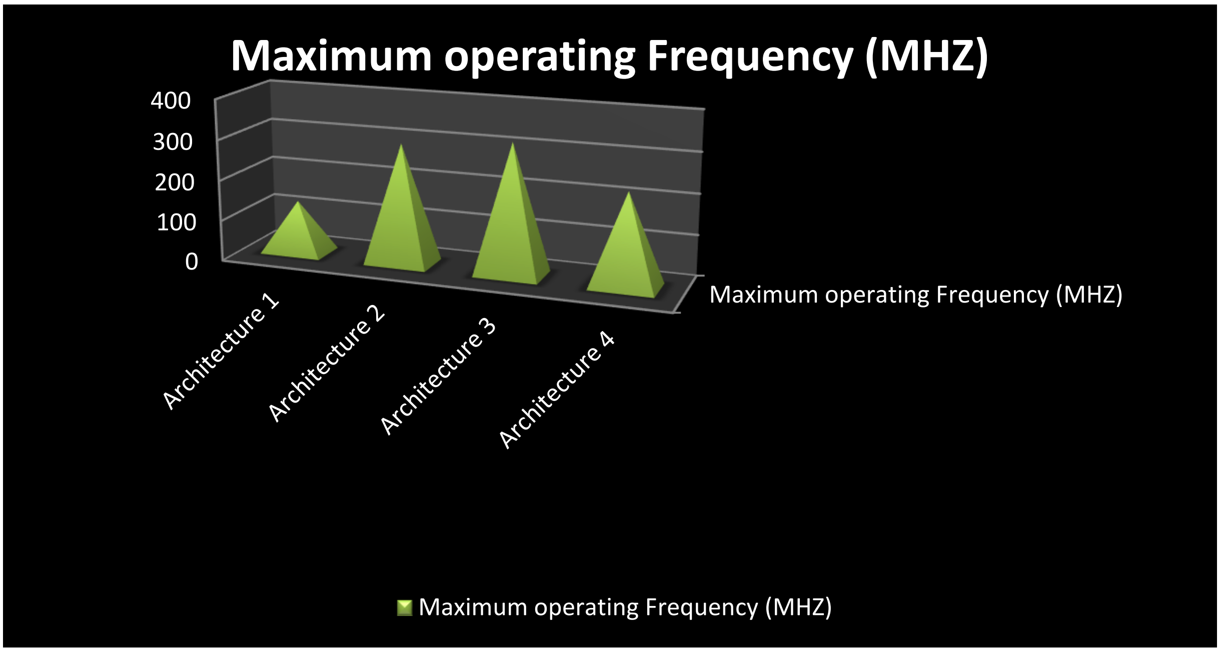

Figure 3: Maximum operating Frequency of various Architectures

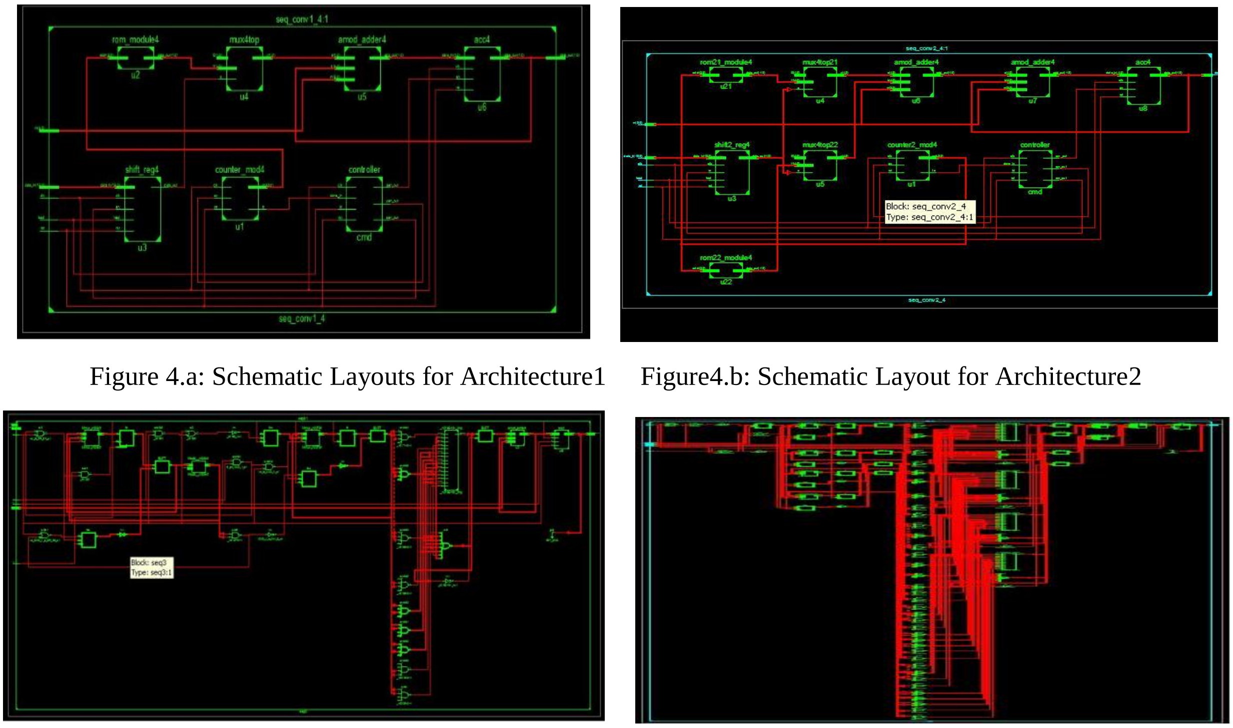

Figure 4.a: Schematic Layouts for Architecture1 Figure4.b: Schematic Layout for Architecture2

Figure 4.c Schematic Layout for Architecture 3 Figure 4.d. Schematic Layout for Architecture4

Figure 4: RTL View of various architectures

Figure 4 (a)-(d) shows the RTL View of all four architectures simulated by Verilog HDL. Architecture four shows that it is less complicated and it's occupying less space. Also this architecture enhances the speed of the system.

CONCLUSION AND FUTURE WORK

This dissertation has investigated all the Arbitrary modulo set forward converter architectures on LUT based for both sequential and parallel techniques with respective to area and speed. The present study confirms previous findings and contributes additional evidence that suggests better architecture among all the forward converters. This current study has shown that in the sequential converter over all device utilization more in this architecture and also the frequency of this architecture is not good. Coming to the sequential/parallel converter the overall device good and frequency has reduced. But in the third architecture modified sequential/parallel converter we have overcome all the performance with device utility and frequency which are better than all the other architecture. Modified Sequential/Parallel Converter (Architecture-3) is the better performance in speed and area when compared to architectures. Modified Sequential/Parallel Converter (Architecture-3) utilizes 50% less recourses when compared to architecture 1 and 2.Modified Sequential/Parallel Converter (Architecture-3) is three times faster than sequential convertor (Architecture-1).In general therefore, it seems that by improving in the converters and modulo-sets, research in the residue number system may take whittle role in future. The issue of forward converter is an intriguing one which could be usefully explored in further research.

REFERENCES

[1] Ahmad Al Badawi, Ahmad M. Baset& QasemS. Abu Al-haija “A novel digital adder design based on Residue number system with special set of moduli {2n+1, 2n, 2n-1}” Department of Computer Engineering, Jordan university of science and technology, Irbid , Jordan. Ijrras4 (1) July 2010.

[2] Amos Omondi school of Electrical and Electronic Engineering South Korea and Benjamin Premkumar school of computer engineering Nanyang technological university Singapore. “Residue number system theory and implementation”2011.

[3] P.V.Ananda Mohan, fellow, IEEE, and A. B. Premkumar, senior member. IEEE Transactions on circuits and systems “RNS-to-Binary Converters for two four-moduli sets” Regular papers, vol. 54, no. 6, June 2007.

[4] Khaledm. Elleithy, “Choosing the system moduli of RNS arithmetic processors”Computer Engineering Department, King Fahd university of petroleum and minerals, Dhahran 31261, Saudi Arabia 2012.

[5] Khaled M. Elleithy" &Magdy A. Bayoumi “Formal Design of RNS Processors” Computer Engineering Department King Fahd hiversify of Petroleum and Minerals Dhahran 31261, Saudi Arabia. China 1991 International Conference on Circuits and Systems, June 1991, Shenzhen, China.

[6] Omar Abdelfattah “Data Conversion in Residue number system” Department of Electrical &Computer Engineering, McgillUniversity Montreal, Canada. January 2011.