BIT EFFICIENT RESIDUE NUMBER SYSTEM BASED LOW POWER RECONFIGURABLE DSP PROCESSOR

M. Zaheer Ahamed[1], S. Nagaraja Rao[2]

[1]*Asst. Prof., Dept of ECE, Brindavan Institute of Tech &Science, Kurnool, A.P., India*
[2]*Vice Principal & Prof. of ECE, Dr. KVSR Institute of Technology, Kurnool, A.P., India*

Abstract

In modern era of advanced computing and high end multimedia there is a need for low power reliable and portable electronic systems. Even though many techniques have been proposed to reduce power over the years, out of all, the best possible solution, so far has been to over scale the power supply. When a system is over scaled the power reduces drastically but this also increases the complexity of the design. The other alternate is to replace the conventional binary number systems with Residue Number System. The residue number system is a non-weighted number system which speeds up arithmetic operations by dividing them into smaller parallel operations and provides carry-free addition, multiplication and borrow-free substraction operations. In this paper we propose a technique to reduce power based on bit efficient Residue Number System with a proper choice of prime moduli . Here we consider a Digital Signal Processor that can be reconfigured as a case study. The technique proposed can reduce power upto 34 percent thn a conventional binary number system based DSP Processor

Keywords: *Voltage Over scaling, Prime Moduli, Critical Path and Non Weighted number systems*

----------***----------

1. INTRODUCTION

In recent times, Residue Number System (RNS) are being popular to implement a variety of specialized high-performance Digital Signal Processing (DSP) systems for its carry-free nature. Weighted number systems such as the binary number system, decimal number system etc has a carry chain [1]. It is often limiting the performance of arithmetic operations [2, 3]. In RNS, several residue digits represent a number. So, arithmetic operations like additions, subtractions and multiplications of higher bit numbers can be decomposed and performed in set of parallel sub-operation. As a result carry propagation, which is a genuine problem in weighted number systems, will be minimized in residue systems. RNS is extremely efficient for many applications such as digital signal processing [4,5,6] communications engineering, computer security (cryptography) [6] etc. Generally, number of bits required in residue number system is greater than that of weighted number systems because RNS gives the number of residues same as the cardinality of the moduli set, increasing the number of bit required to express it in RNS. A number system is said to have higher bit efficiency if the bit required to represent a particular dynamic range is lower. There are many important parameters that determine the efficiency of RNS and bit efficiency is one of them. The bit efficiency depends on the choice of the moduli set [7]. There are several techniques [7,8,9] for moduli set generation reported in the literature {2n ,2n +1,2n -1} , -1 {2 , 2 -1,2 -1} n n n and 2 {2 1, 2 1,2 -1} n n n + + . For these schemes no algorithm is given to generate a moduli set; they are generated heuristically by finding a suitable n. The contributions of paper are following:

1. proposed an algorithm to generate any moduli set with finite cardinality in a given dynamic range.
2. bit efficiency of the proposed scheme is better than all other scheme given in the literature.
3. theoretical analysis and proof of the proposed scheme to show that the proposed solution
gives better results than the existing scheme [9].
4. Applicability of this scheme in a reconfigurable DSP Processor

2. OVERVIEW OF RESIDUE NUMBER SYSTEM

An RNS is defined by a set of relatively prime moduli. If P denotes the Moduli set, then

$$P = \{p_1, p_2, p_3, \ldots p_L\},\ \text{GCD}(p_i, p_j) = 1, \text{ for } i \mathrel{!}= j$$

Any integer in the residue class $Z_{m,}$ where

$M = p_1 * p_2 * p_3 * \ldots p_L$ Has a unique L- tuple representation given by

$X \xrightarrow{RNS} (X_{1,} X_{2,} \ldots, X_L)$ Where $X_i = X \bmod p$; i^{th} Residue of X

While RNS adds, subtracts, and multiplies efficiently, division is not a closed operation.

If X, Y and Z have RNS representation given by

$X \xrightarrow{RNS} (X_{1,} X_{2,} \ldots, X_L)$; $Y \xrightarrow{RNS} {}_{2,} \ldots, Y_L)$; $Z \quad (Z_{1,} Z_{2,} \ldots, Z_L)$

Then denoting o to represent + or *, the RNS version of the Z=XoY satisfies

$$Z \xrightarrow{RNS} (Z_1, Z_2, \ldots, Z_L) = ((X_1 o\, Y1) \bmod p_1, \ldots, (X_L o\, Y_L) \bmod p_L)$$

If Z belongs to Z_M.

The ith RNS digit, namely Z_i is defined in terms of $(X_i o\, Y_i)$ mod p_i only. That is, no carry information need to be communicated between residue digits. Since there is no requirement to manage carry information from RNS digit to digit, the overhead of manipulating carry information can be avoided. The net result is very high speed concurrent operations, and it is speed alone that makes RNS attractive.

3. CHINESE REMAINDER THEOREM

This theorem is an important cornerstone in the modern theory of Residue Number System. The procedure for converting remainders or residues, into integers is given by Chinese Remainder Theorem (CRT).

$$X = \left(\sum_{i=1}^{L} s_i \left(X_i s_i^{-1} \right) \bmod p_i \right) \bmod M$$

Where $s_i = M/ p_i$ s_i^{-1} are called the multiplicative inverse of s mod p_i so that

$(s_i^{-1} s_i) \bmod p_i = 1$.

4. BIT EFFICIENCY IMPROVEMENT OF RNS

The bits required to implement all the blocks of RNS number are depends on moduli set. Let N be the number of bit then 2^N is called dynamic range. Now $(r_0, r_1, r_2, \ldots, r_{t-1})$ denotes the t-moduli set, where $r_0, r_1, r_2, \ldots, r_{t-1}$ all are relatively prime and product of these t numbers should be greater or equal to $2^N - 1$. Total bits required is calculated as $\lceil \log_2 r_0 \rceil + \lceil \log_2 r_1 \rceil + \lceil \log_2 r_2 \rceil + \ldots + \lceil \log_2 r_{t-1} \rceil$ Bit width of the different arithmetic block (like, adder, multiplier) of residue systems depend on the number $\lceil \log_2 r_0 \rceil + \lceil \log_2 r_1 \rceil + \lceil \log_2 r_2 \rceil + \ldots + \lceil \log_2 r_{t-1} \rceil$. Lower the value of this term, more optimized design of RNS in terms of bit width in achieved. Choice can be made over the various moduli set (like, three-moduli, four- moduli) and also the number within the set. In this section we describe an algorithm to generate any number of moduli set for a given precision.

Module find_moduli (N,n,SM)
//Input: N (no. of Bit), n (no. of moduli set)
//Output: SM (Efficient moduli set)

Step 1: $x = \left\lfloor \sqrt[n]{2^N - 1} \right\rfloor$

Step 2: if x is even then $2n = x$

else $2n = x + 1$

Step 3: When n = 3

if $((2n)(2n+1)(2n-1) \geq (2^N - 1))$ then $S_M = \{2n, 2n+1, 2n-1\}$

else n will be incremented till $((2n)(2n+1)(2n-1) \geq (2^N - 1))$ condition will be satisfied.

Step 4: if n = 4 then

Let $k = \left\lceil 2^N - 1 \Big/ \{(2n)(2n+1)(2n-1)\} \right\rceil$ Find the smallest number $k_1 \geq k$, where k_1 is relatively prime to $2n$, $2n+1$ and $2n-1$.

$S_M = \{2n, 2n+1, 2n-1, k1\}$

5. ARCHITECTURE OF RECONFIGURABLE RNS PROCESSOR

The general architecture of a reconfigurable RNS processor is shown in Figure 3. Given a moduli set hardware complexity depends on the functionalities of the RNS. Because of the space issue, a simplified structure is shown using only three arithmetic operators. It contains

1. 2 Binary to RNS converters
2. 7 MUXs
3. Adder
4. Subtractor
5. Multiplier
6. RNS to Binary converter [26][27]

Binary numbers are passed to the processor as inputs which first are converted to the RNS number. Here the selection of moduli is very much important because the proper selection of moduli optimizes the bit efficiency, area of the processor & time to process the particular function. After the conversion of the binary number to its corresponding residue representation, the arithmetic operations can be performed. As any binary number produces a set of RNS numbers depending upon the number of moduli used, m copies of arithmetic units (adder, subtractor, multiplier etc) are required to perform some arithmetic operation of a number when it is converted to RNS, where m is the number of moduli used in that scheme. As the residues can be independently operated, parallel arithmetic operations can be performed on the residue set. Figure 4 shows the control & data flows between the various paths. The Programmable Controller can program the RRNS Processor directly or Programmable Memory is used to store the bit stream. Programmable Controller is governed by the General

purpose CPU. In general, all modular arithmetic operations like Binary to RNS conversion or RNS addition, multiplication are implemented in chip by using two different methods [25]. One is the table look-ups, implemented by PLA. Second one is the Hybrid Methods, which is the combination of the legacy hardware, like full adders, with a table look-up, which can be used to convert the output of the legacy hardware to the correct residue format. Use of PLA gives a faster hardware than the Hybrid method, but the later takes less area in the chip than the former. PLA can be a good choice of modern reconfigurable RNS processor because of their regular dense structure & easy interconnection. The area & the speed of the reconfigurable RNS processor depends of the number of moduli in the moduli set, the method used for generation as well as the scheduling algorithm used.

6. DESIGN PROCEDURE

In the reconfigurable architecture, there is no fixed path between the device units, but the path can be changed depending upon the requirements. MUX are used before the inputs of the device units that act as the switch determining a specific path with respect to some particular select condition. For an example, suppose there are x number of adders, y number of subtractor & z number of multiplier in the processor. Also we are considering that the chip is accepting k number of inputs. So in general, for all the arithmetic unit having 2 inputs, the MUX in front of the inputs must be having of (x inputs coming from the outputs of adders + y inputs coming from the outputs of subtractors + z inputs coming from the outputs of multipliers + k external inputs). So the MUX must have _log2(x + y + z + k)_ select lines. In our example (Figure 3), for simplicity, we have taken x = y = z = 1, k = 2. So we can use 5 x 1 MUXes having 3 select lines before all the arithmetic devices in general. As the output coming from all the units are fed to the inputs of all the unit devices in general, it is possible to have any combination of the arithmetic operations computed by the processor. In our work, the aim is to design a RNS processor which can be reconfigured dynamically to compute some pre determined functions. For this, the unit operations need to be analysed & sequenced in terms of the inputs & arithmetic operations. As the arithmetic operations are depicted in terms of the select condition on the MUX, the inputs as well as the select conditions need to be stored using a LUT. The bit sequences are stored in the LUT block wise, each block has some particular address. When the address is given for some function, these inputs & select conditions are passed to the input MUXes

7. CONCLUSION

In this paper we proposed an algorithm to generate any moduli set of finite cardinality for a given dynamic range and given the proof of correctness for this proposed algorithm. We have also shown that bit efficiency of the proposed scheme is better than all other schemes given in the literature. In future we will be working on how these parameters, bit efficiency, h/w complexity and time can be optimized for a reconfigurable RNS processor. Another moduli set can be proposed which is better than our proposed scheme considering the three parameters mentioned above, using which we can get the optimized values of the same.

REFERENCES

[1]. John P. Hayes, “Computer Architecture and Organization”, Mcgraw-Hill, 2004.
[2]. Kai Hwang, “Computer Arithmetic: Principles, Architecture and Design”, John Wiley & Sons, 1979.
[3]. Chao-Lin Chiang and Lennart Johnsson, “Residue Arithmetic and VLSI”, Technical Report. California Institute of Technology, 2002.
[4]. Taylor F., "A Single Modulus ALU for Signal Processing", IEEE Transactions on Acoustics, Speech, Signal Processing, vol. 33, pp. 1302-1315, 1985.
[5]. Freking W.L. and Parhi K.K., "Low-power FIR digital filters using residue arithmetic", 31st Asil. Conference on Signals, System & Computer, Pacific Grove, CA, USA, pp. 739-43, 1997.
[6]. Yen Sung-Ming, Kim Seungjoo, Lim Seongan, Moon Sang-Jae, “RSA Speed-up with Chinese Remainder Theorem Immune against Hardware Fault Cryptanalysis”, IEEE transactions On Computers, vol. 52, issue 4, pp. 461 - 472 , April 2003.
[7]. Amos Omondy, Benjamin Premkumar, “Residue Number System theory and implementation”, Imperial College Press, 2007.
[8]. Zhongde Wang, G.A. Jullien and W.C. Miller, “An Efficient 3-Modulus Residue to Binary Converter”, IEEE transactions on Circuits and Systems, vol. 3, pp. 1305-1308, 1996.
[9]. Wei Wang, M.N.S. Swamy, and M.O. Ahmad, “Moduli Selection in RNS for Efficient VLSI Implementation”, International Symposium on Circuits and Systems, vol.4, pp. IV-512- IV-515, 25- 28 May, 2003.
[10]. LEE Ki Ja, “Interval Arithmetic Operations in Residue Number System”, IEICE transactions on information and systems, pp.1361-1371, 2002.
[11]. C Ding, D Pei, and A Salomaa, “CHINESE REMAINDER THEOREM: Applications in Computing, Coding, Cryptography”,
Word Scientific, 1996.

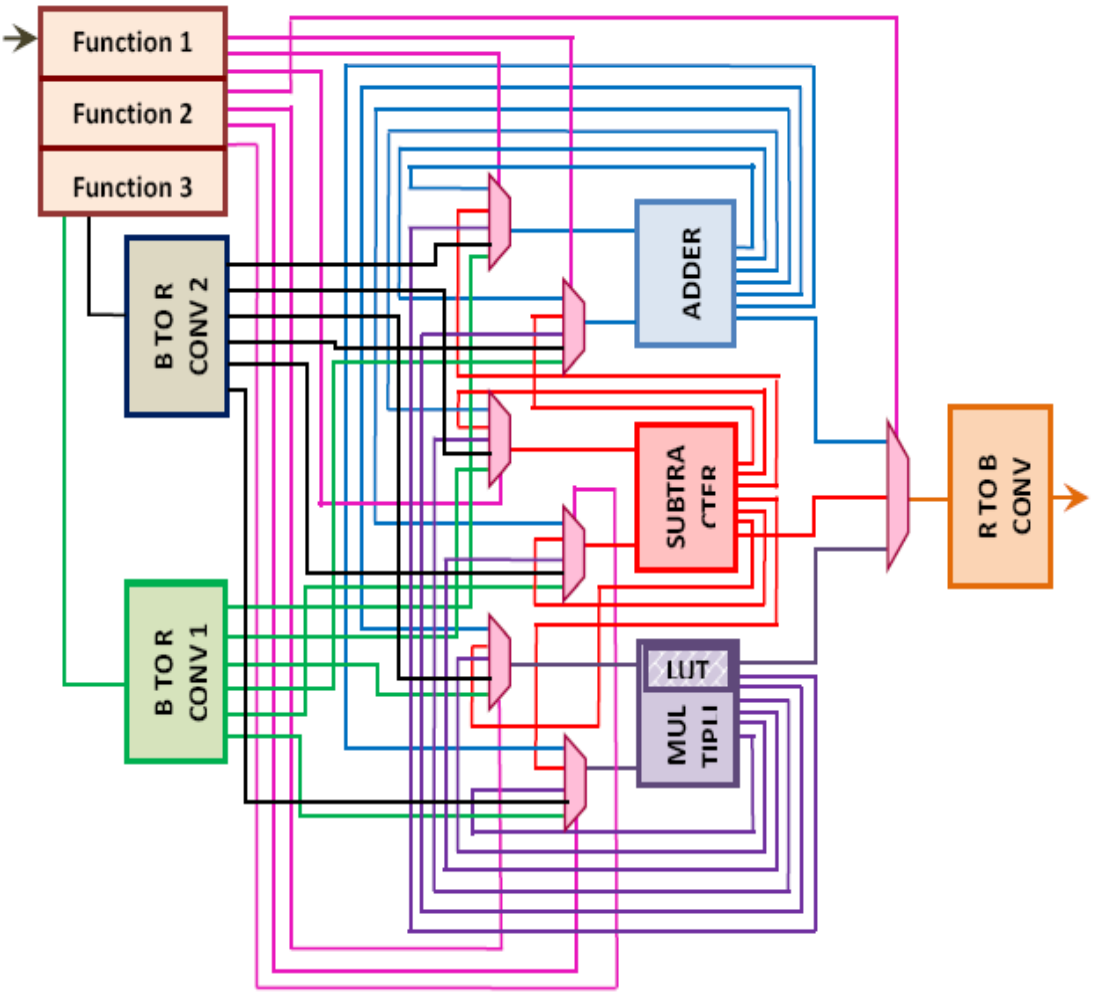

Fig 1 : Simplified diagram of the proposed Reconfigurable RNS Processor

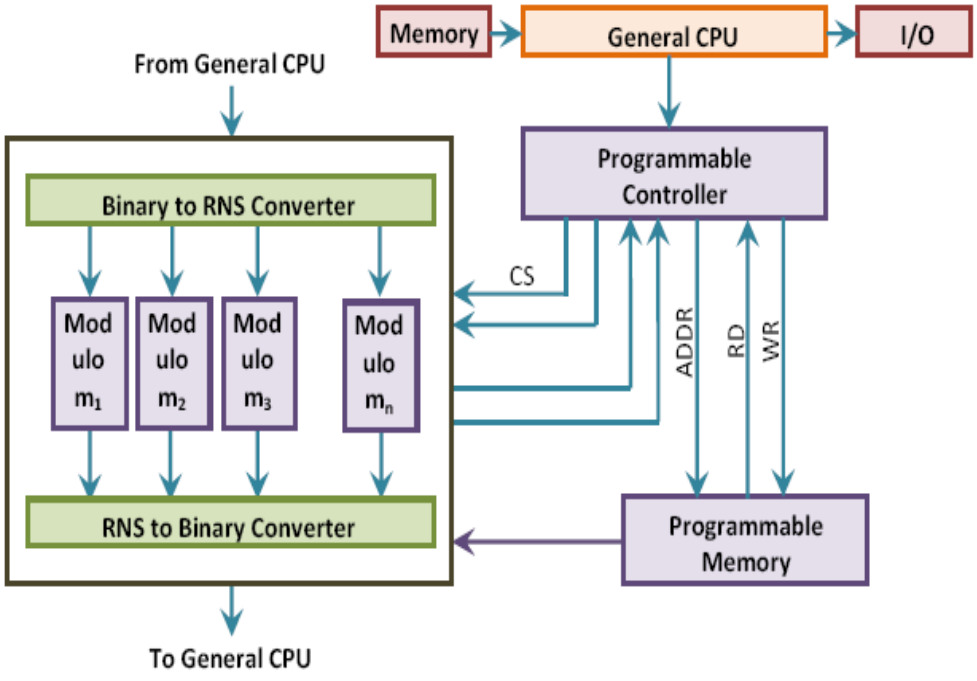

Fig 2 : Control & Data flow in the proposed Reconfigurable RNS Processor

BIOGRAPHIES

Mr. M. Zaheer Ahamed has persued his B.Tech from ALFA college of Engg, & Tech. Allagadda and M.Tech from VIF college of Engg. & Technology, Hyderabad. Presently he is working as Asst. Prof in Dept of ECE of Brindavan Institute of Technology & Sciences, Kurno ol. His research areas of interest are Low power VLSi, VLSI for signal and image processing, Fault detection and testing of digital circuits. he has 3 International Journal Publications & 4 International Conference papers to his credit. He is a Life member of ISTE

Dr.S.Nagaraja Rao is a Doctorate in Electronics and Communications Engineering with 23 Years of experience from which 7 years in Industry and 16 years in Academics. He obtained his graduation in Electronics and Communication Engineering in 1990 from S.V.University, Tirupathi. He completed his M.Tech Degree in DSCE from J.N.T.University, Hyderabad in 1998. He received his Ph.D Degree from JNTUA, Anantapuramu, Andhra Pradesh in 2011. He is an able administrator and has experience as Professor of ECE,HOD, Dean (R& D) in various Engineering Colleges before he assumed the office of **Vice Principal** at Dr.KVSRIT. He is a Life Member of ISTE (MISTE) and Member of Instrument society of India(MISOI). He published 15 research contributions in various International Journals, International conferences and National conferences.