IOSR Journal of VLSI and Signal Processing (IOSR-JVSP)
e-ISSN: 2319 – 4200, p-ISSN No. : 2319 – 4197
PP 55-58
www.iosrjournals.org

High Speed Reverse Converter Using Parallel Prefix Adder

Midhuna D[#1], Tamilselvi T[*2]
*[#1]PG Scholar, ME Applied Electronics,[*2] Assistant Professor, ECE Dept., Jerusalem College of Engineering,*
Velachery Main Road, Pallikaranai, Chennai-600 100, Tamil Nadu, India

Abstract: *An unconventional non-weighted number system that has gained a great scientific interest isthe residue number system (RNS), which is capable of parallel, carry-free and high speed arithmetic. It uses residues of a number, in particular modulus for its representation. The design of reverse converter is based on regular and modular adders, which leads significant increasein power consumption and low speed. This is the main reason which prevents the use of Residue Number system in many applications. The parallel prefix based adder components is used to solve the high power consumption problem and provide better trade-off between power consumption and delay.*

***Keywords**: Residue number system, Chinese remainder theorem, parallel prefix adder, reverse converter.*

I. INTRODUCTION

In day to day life Embedded systems have been transformed from simple, single-function control systems to highly complex system. Embedded systems like the personal wireless communication and handheld, portable multimedia and communication devices have created stringent requirements such as performance, power, cost and time- to-market. These battery-powered devices have created a demand for cheap, high performance, and power efficient embedded processors.The residue number system (RNS) plays a significant role in such devices due to low power feature and competitive delay. The residue number system wasused in the implementation of fast arithmetic and fault tolerance in digital systems.The RNS requires forward and reverse conversion. However in reverse conversion the conversion stages are very critical in the evaluation of performance of overall RNS. Compared to other step reverse conversion leads to more delay. Hence the reverse conversion process introduces more overhead in terms of speed and complexityand is more difficult in computation of the process. To improve the performance of the converters, well-known adder architectures were used. To implement carry-propagate adders (CPAs), such as carry-save adders (CSAs) and ripple-carry architectures is used.The ones with carry-look ahead or parallel-prefix architecturesare the fastest and expensive adders.The usage of the parallel-prefix adders to implement converters highly increases the speed and reduces the power consumption problem.

II. BACKGROUND

The residue number system encodes a large number into a group of small numbers which results in significant speed up of the overall data processing. Each large integer can be represented as a set of smaller integers called the residues. Three main steps involved in RNS are forward conversion, arithmetic computation and reverse conversion. The process of encoding the input data into RNS representation is called Forward Conversion.This process can be done by dividing the given conventional number by all the moduli in the moduli set and finding the remainders of the divisions. Reverse Conversion is the process of converting RNS representation into conventional representation. Distinct moduli sets have to be chosen.Hardware components selection is a key to the RNS performance. The use of parallel prefix adder based on different architecture with distinct structure such as Kogge–Stone (KS) and Brent—Kung adder shows a significant increase in performance.

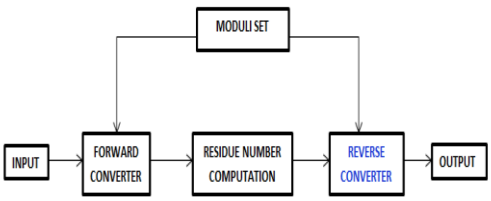


Fig 1: Block Diagram of RNS system

A. Chinese remainder theorem

Consider a pair-wise relatively prime moduli set $\{m_1, m_2,\ldots m_n\}$ and a residue representation $\{r_1,r_2,\ldots r_n\}$ of some number X, i.e.$r_i= |X|m_i$, that number and its residues are related by the equation:
$|X|_M=|\sum_{i=1}^{n} ri|M_i^{-1}|_{mi}\, M_i|_M$ …. (1)

Where M is the product of the m_i's, and $M_i=M/m_i$ .The left-hand side of the modular reduction can be omitted when the valuesinvolved are constrained.
we rewrite X as:

$$X \triangleq \{r1, r2, \ldots, rn\}$$
$$\triangleq \{r1,0,\ldots 0\} + \{0, r2, \ldots,0\} + \{0,0,\ldots, rn\}$$
$$\triangleq X1 + X2 + \cdots + Xn \quad \ldots.\ (2)$$

Hence, X_i's has to be found in this process. A reverse conversion process by itself is the operation of obtaining each X_i which is easier than obtaining X.
Consider now that we want to obtain X_i from $\{0,0\ldots.,r_i,\ldots.,0,0\}$. Except for r_i, the residues ofX_i are zeros. This shows that X_i is a multiple of $m_j$where j≠i . Therefore, X_i can be expressed as:
$X_i \triangleq r1 * \{0,0,\ldots 1,\ldots,0,0\} \triangleq ri * Xi$ …. (3)
Where $\widetilde{X_\iota}$ is found such that $|\widetilde{X_\iota}\,|_{m_i} = 1$. From the above equation the relation between the number r_i and its inverse r_i^{-1} is as follows:
$(r_i \text{ X } r_i^{-1})\text{mod } m_i=1$ …. (4)
We define M_i as M/M_i , where $M = \prod_{i=1}^{k} pi$. Then:
$||M_i^{-1}|_{mi}M_i|_{mi}= 1$ …. (5) (5)
Since all m_i's are relatively prime, the inverses exist:
$\overline{X_i} = |M_i^{-1}|_{mi}M_i$ …. (6)
$X_i = r_iX_i = r_i|M_i^{-1}|_{mi}M_i$ …. (7)
$X=\sum_{i=1}^{n} Xi = \sum_{i=1}^{n} r_i|M_i^{-1}|_{mi}M_i$ …. (8)
Modulo reduction has to be added to both sides of the equationensures that the final value is within the dynamic range.

B. Parallel Prefix Adder

The faster operation in the reverse converter design was achieved with the help of parallel prefix structure.

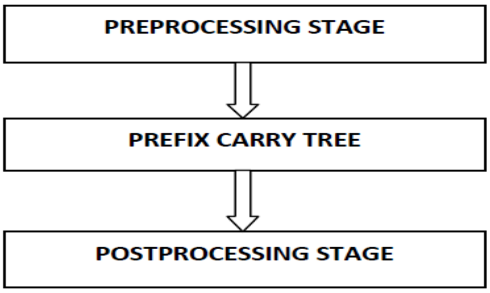


Fig 2: Block diagram of parallel prefix structure

There are three main blocks in parallel prefix structure, they are pre-processing block, prefix carry tree and post processing block. The operation of the adder begins with pre-processing stage by generating the Generate (Gi) and Propagate (Pi) shown in equation [1] & [3]. In prefix carry tree the previous block signal yield all carry bit signal. These stages contain three logic complex cells. They are Black cell, Grey cell and Buffer cell. Both the propagate (P(i,j)) and generate (G(i,j))are computed by the black cell using the equation[3] &[4].The Grey cell executes only the generate(G(i,j)).The post processing block receives the carry bits generated in the second stage which generates the sum and the equation is given[5].The block diagram is shown in the Fig 1
$G_{m:n}=A_n \cdot B_n$ …. (9)
$G_0 =C_{in}$ …. (10)
$P_{m:n}=A_n \oplus B_n$ …. (11)
$P_0 =0$ …. (12)
$G_{m:n}=G_{n:k} + (P_{n:k} \cdot G_{k-1:n})$ …. (13)
$P_{m:n}=P_{n:k} \cdot P_{k-1:j}$ ….. (14)
$Sn =Pn \oplus Cin$ …. (15)

By usingthe Brent Kung adder prefix structurewe can achieve the high speed and reduced power consumption in the system. The BK adder is chosen mainly for minimum fan-out and high speed in operationcompared to other parallel prefix adder structure. The example BK adder prefix structure with three basic cells in the prefix structure is shown in fig 3.

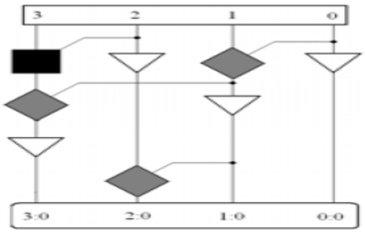


Fig 3: 4-bit BK adder prefix structure

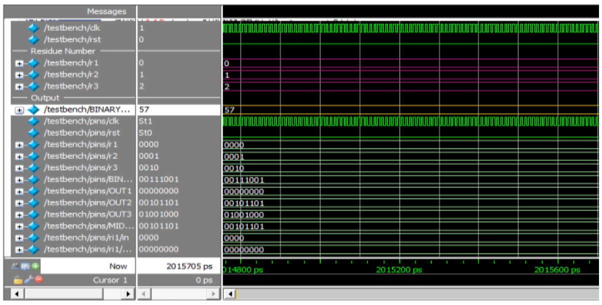
III. DESIGN METHODOLOGY

The methodology of designing a reverse converter is described in this section. The selection of moduli set is the first step involved in designing a reverse converter. Dynamic range, speed, and the hardware realization of RNS are determined by the moduli set selection. The three moduli set (2^n-1, 2^n, 2^n+1) were considered for the design, where n is a natural number. The importance of these moduli set is that they can be efficiently implemented using binary hardware that leads to simple design and offers speed cost benefits. The values of the moduli of the moduli set and the residue numbers must be substituted in CRT conversion algorithm formulas. Here the residue number is the output of the forward converter. The CRT conversion algorithm involves the calculation of recursive moduli inverse. The best way to implement moduli inverse is to save the constant in ROM, which is then multiplied with the residue number and then added using the adders. Arithmetic properties and propositions are used for simplifying the resulting equations. The final equations are realized by adder components like CSA-EAC, CPA-EAC, CPA and PPA. The above said procedure is consolidated as the following algorithm.

Step 1: Set the input
Step 2: Set the moduli set.
Step 3: Calculate the residue number
Step 4: Carryout computation
Step 5: Precomputation of moduli set
Step 6: Calculate recursive moduli inverse
using ROM
Step 7: Calculate the summation using adder
Step 8: Get the output

IV. RESULTS

The circuit can be designed and specified in Verilog. The modulus set (2n-1, 2n, 2^n+1) was chosen. The Proposed system is simulated and verified using ModelSim ALTERA STARTER EDITION 6.4a. The below simulation result shows the output of the system with n=2 for the above moduli set.



```
Messages
/testbench/clk               1
/testbench/rst               0
Residue Number
/testbench/r1                0           0
/testbench/r2                1           1
/testbench/r3                2           2
Output
/testbench/BINARY...         57          57
/testbench/pins/clk          St1
/testbench/pins/rst          St0
/testbench/pins/r1           0000        0000
/testbench/pins/r2           0001        0001
/testbench/pins/r3           0010        0010
/testbench/pins/BIN...       00111001    00111001
/testbench/pins/OUT1         00000000    00000000
/testbench/pins/OUT2         00101101    00101101
/testbench/pins/OUT3         01001000    01001000
/testbench/pins/MID...       00101101    00101101
/testbench/pins/ri1/in       0000        0000
/testbench/pins/ri1/...      00000000    00000000
Now                          2015705 ps  014800 ps   2015200 ps   2015600 ps
Cursor 1                     0 ps
```

Fig 4: Simulation Output

The following Table represents the comparison of adders in terms of delay and frequency which is calculated using the tool Xlinix ISE 14.2 for the target device xc3s400e-5tq144.

Table 1: Comparison of Adders in terms of Delay and Frequency

ADDER	DELAY(ns)	FREQUENCY(MHz)
RCA	12.804	78.102
CLA	7.765	128.785
CSA	7.626	131.129
KS	2.257	443.095
BK	2.021	494.841

V. CONCLUSIONS

In this paper the reverse converter was simulated. The above result shows that the reverse converter simulated using the BK parallel prefix adder network has less delaycompared to other adders. The reverse converter was simulated for the {2^n-1, 2n, 2n+1} moduli set. This shows that the delay is reduced and the efficiency was improved.

REFERENCES

[1]. A. Omondi and B. Premkumar, Residue Number Systems: Theory and Implementations. London, U.K.: Imperial College Press, 2007.

[2]. B. Parhami, Computer Arithmetic: Algorithms and Hardware Designs, 2nd ed., New York, NY, USA: Oxford Univ. Press, 2010.

[3]. J. Chen and J. Hu, “Energy-efficient digital signal processing via voltageover scaling-based residue number system,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 21, no. 7, pp. 1322–1332, Jul. 2013.

[4]. C. H. Vun, A. B. Premkumar, and W. Zhang, “A new RNS based DA approach for inner productcomputation,” IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 60, no. 8, pp. 2139–2152, Aug. 2013.

[5]. S. Antão and L. Sousa, “The CRNS framework and its application to programmable and reconfigurable cryptography,” ACM Trans. Archit. Code Optim., vol. 9, no. 4, p. 33, Jan. 2013.

[6]. A. S. Molahosseini, S. Sorouri, and A. A. E. Zarandi, “Research challenges in next-generation residue number system architectures,” in Proc. IEEE Int. Conf. Comput. Sci. Educ., Jul. 2012, pp. 1658–1661.

[7]. K. Navi, A. S. Molahosseini, and M. Esmaeildoust, “How to teach residue number system to computer scientists and engineers,” IEEE Trans. Educ., vol. 54, no. 1, pp. 156–163, Feb. 2011.

[8]. Y. Wang, X. Song, M. Aboulhamid, and H. Shen, “Adder based residue to binary numbers converters for (2n − 1, 2n, 2n + 1),” IEEE Trans. Signal Process., vol. 50, no. 7, pp. 1772–1779, Jul. 2002.

[9]. B. Cao, C. H. Chang, and T. Srikanthan, “An efficient reverse converter for the 4-moduli set {2n − 1, 2n, 2n + 1, 22n + 1} based on the new Chinese remainder theorem,” IEEE Trans. Circuits Syst. I, Fundam. Theory Appl., vol. 50, no. 10, pp. 1296–1303, Oct. 2003.

[10]. A. S. Molahosseini, K. Navi, C. Dadkhah, O. Kavehei, and S. Timarchi, “Efficient reverse converter designs for the new 4-moduli sets {2n − 1, 2n, 2n + 1, 22n+1 − 1} and {2n − 1, 2n + 1, 22n, 22n + 1} based on new CRTs,” IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 57, no. 4, pp. 823–835, Apr. 2010.

[11]. A. S. Molahosseini and K. Navi, “A reverse converter for the enhanced moduli set {2n − 1, 2n + 1, 22n, 22n+1 − 1} using CRT and MRC,” in Proc. IEEE Comput. Soc. Annu. Symp. VLSI, Jul. 2010, pp. 456–457.

[12]. L. Sousa and S. Antao, “MRC-based RNS reverse converters for the four-moduli sets {2n + 1, 2n − 1, 2n, 22n+1 − 1} and {2n+ 1, 2n − 1, 22n, 22n+1 − 1},” IEEE Trans. Circuits Syst. II, vol. 59, no. 4, pp. 244–248, Apr. 2012.

[13]. L. Sousa and S. Antão, “On the design of RNS reverse converters for the four-moduli set {2n +1, 2n −1, 2n, 2n+1+1},” IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 21, no. 10, pp. 1945–1949, Oct. 2013.

[14]. M. H. Sheu, S. H. Lin, C. Chen, and S. W. Yang, “An efficient VLSI design for a residue to binary converter for general balance moduli (2n − 3, 2n + 1, 2n − 1, 2n + 3),” IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 51, no. 3, pp. 152–155, Mar. 2004.

[15]. R. P. Brent and H. T. Kung, “A regular layout for parallel adders,” IEEE Trans. Comput., vol. 31, no. 3, pp. 260–264, Mar. 1982.

[16]. J. Sklansky, “Conditional sum addition logic,” IRE Trans. Electron. Comput., vol. 9, no. 6, pp. 226–231, Jun. 1960.

[17]. P. M.Kogge and H. S. Stone, “A parallel algorithm for the efficient solution of a general class of recurrence equations,” IEEE Trans. Comput., vol. 22, no. 8, pp. 783–791, Aug. 1973.

[18]. R.Zimmermann, “Binary adder architectures for cell-based VLSI and their synthesis,” Ph.D. dissertation, Integr. Syst. Labor., Dept. Inf. Technol. Electr. Eng., Swiss Federal Inst. Technol., Zurich, Switzerland, 1997.