Design and Implementation of RNS Reverse Converter using Parallel Prefix Adders

M.Kuttimani Rajalingam,
PG Scholar.
Department of VLSI Design
Shanmuganathan Engg
College
Pudukkottai, India

A. Muthumanicckam,
M.E., Asst.prof.
Department of ECE
Shanmuganathan Engg
College
Pudukkottai, India

Mrs.R.Sornalatha,
M.E., Asst.Prof.
Department of ECE
Shanmuganathan Engg
College
Pudukkottai, India

ABSTRACT

The implementation of residue number system reverse converters based on well-known regular and modular parallel prefix adders is analyzed. The VLSI implementation results show a significant delay reduction and area $\times$ time² improvements, all this at the cost of higher power consumption, which is the main reason preventing the use of parallel-prefix adders to achieve high-speed reverse converters in recent systems. Hence, to solve the high power consumption problem, novel specific hybrid parallel-prefix based adder components that provide better trade-off between delay and power consumption are herein presented to design reverse converters. We propose Parallel distributed arithmetic convolution technique in Reverse Converter to increase the system performance

Keywords

Digital arithmetic, parallel-prefix adder (PPX), residue number system (RNS), parallel distributed arithmetic convolution architecture, reverse converter.

1. INTRODUCTION

Now a days with the extensive use of wireless devices, battery-based and portable devices, the residue number system (RNS) can play a significant role due to its low power features and competitive delay. The RNS can provide carry free and fully parallel arithmetic operations [1], [2] for several applications, including digital signal processing and cryptography [3]–[6]. However, its real usage requires forward and reverse converters to be integrated in the existing digital systems. The reverse conversion, i.e., residue to binary conversion, is a hard and time-consuming operation [7]. Hence, the problem of designing high-performance reverse converters has motivated continuous research using two main approaches to improve the performance of the converters: 1) investigating new algorithms and novel arithmetic formulations to achieve simplified conversion formulas and 2) introducing new moduli sets, which can lead to more simple formulations. Thereafter, given the final simplified conversion equations, they are computed using well-known adder architectures, such as carry-save adders (CSAs) and ripple-carry architectures, to implement carry-propagate adders (CPAs) and, more seldomly, fast and expensive adders such as the ones with carry-look ahead or parallel-prefix architectures.

In this brief, for the first time, we present a comprehensive methodology to wisely employ parallel-prefix adders in carefully selected positions in order to design fast reverse converters. The collected experimental results based on area, delay, and power consumption show that, as expected, the usage of the parallel-prefix adders to implement converters highly increases the speed at the expense of additional area and remarkable increase of power consumption. The significant growing of power consumption makes the reverse converter not competitive. Two power-efficient and low-area hybrid parallel-prefix adders are presented in this brief to tackle with these performance limitations, leading to significant reduction of the power delay product (PDP) metric and considerable improvements in the area-time² product (AT²) in comparison with the original converters without using parallel-prefix adders.

2. BACKGROUND

2.1 Residue Number System

The residue number system has been considered a powerful unconventional number system in computer arithmetic due to its attractive features such as carry-limited computations which make them an effective tool to increase speed and reducing power consumption [3], [4]. The forward converter, modulo arithmetic units, and reverse converter are the main parts of the RNS. In contrast to other parts, reverse converter consists of a complex and nonmodular structure. Therefore, more attention should be directed to its design to prevent slow operation and compromise the benefits of the RNS. Both the characteristics of the moduli set and conversion algorithm have significant effects on the reverse converter performance. Hence, distinct moduli sets have been introduced [8]–[14]. In addition to the moduli set, hardware components selection is key to the RNS performance. For instance, parallel-prefix adders are known as unsuitable structures for complex reverse converters because of their high power consumption.

Parallel-prefix adders with its high-speed feature have been used in the RNS modular arithmetic channels. This performance gain is due to parallel carry computation structures, which is based on different algorithms such as [15]–[17]. Each of these structures has distinct characteristics, such as Sklansky (SK), and Kogge–Stone (KS) as they have the maximum and minimum fan-out, respectively, both providing minimal logic depth. Minimum fan-out comes at the expense of more circuit area [18]. Therefore, hardware components selection should be undertaken carefully.

Fig.1: PPX structure.

2.2 Reverse Converter

The regular CPA with end around carry (EAC) [19] is by default a moduli $2^n - 1$ adder with double representation of zero, but in reverse converters a single representation of zero is required. So, a one-detector circuit has to be used to correct the result, which imposes an additional delay. However, there is a binary-to-excess-one converter (BEC) [20], which can be modified to fix the double-representation of zero issue.

In Fig. 3. The HMPE consists of two parts: 1) a regular prefix adder and 2) a modified excess-one unit. First, two operands are added using the prefix adder, and the result is conditionally incremented afterward based on control signals generated by the prefix section so as to assure the single zero representation.

3. PERFORMANCE EVALUATION

3.1 Parallel Distributed Arithmetic Convolution Architecture

Fig.2: Parallel Distributed Arithmetic Convolution architecture

If the existing architecture is working slowly for double representation zero operation, to eliminate the problem we use external unit. Fig.2. Shows the proposed system is used fully series and fully parallel operation. Proposed system components are:

1. MSRA
2. Multiplicand and adder tree

3. Control unit
4. Convolution process
5. Modified excess one unit

3.1 Multiplicand and Adder Tree

Fig.3: Mul-Adder tree

The MSRA is an implemented registers array having each register with 8 bit wide, to be able to store a pixel. The MSRA is designed to perform left/right shifting. Moreover, it allows for the up/down shifting along the vertical direction. The output signals of each registry are connected to the Mul-Adder Tree (Fig.3) to perform the signed to perform the left/right shifting products with the weights of the kernel matrix. The output unit includes an output buffer to store 4 pixels, and a data normalization unit. The normalization unit is required to normalize the size of the convolved pixel to 8 bits. The Control Unit coordinates the data stream within the system architecture. Mainly, it drives the control signals needed to shift the MSRA and to acquire a new row/column.

3.3 Control Unit

The output unit includes an output buffer to store 4 pixels, and a data normalization unit. The normalization unit is required to normalize the size of the convolved pixel to 8 bits. The Control Unit coordinates the data stream within the system architecture. Mainly, it drives the control signals needed to shift the MSRA and to acquire a new row/column.

3.4 Convolution

Convolution is a mathematical way of combining two signals to form a third signal. It is the single most important technique in Digital Signal Processing. Using the strategy of impulse decomposition, systems are described by a signal called the impulse response. Convolution is important because it relates the three signals of interest: the input signal, the output signal, and the impulse response.

Fig.4: Distributed Arithmetic Convolution architecture

The 1-bit design can be easily extended to n-bit input data through a range of approaches, from fully parallel to fully serial. The parallel scheme is shown in Fig. 4. For each row, it consists of n identical instances of the module. Outputs are all added according to the weights of their corresponding bit in the input word which is shown in Fig.2. Clearly, this solution will provide the highest speed at the expense of the highest resources consumption.

4. SIMULATION RESULT

4.1 Simulation

The simulation of reverse converter units are synthesized with the device xc2s50-6, package of tq144 and corresponding Timing waveforms are taken from the Test Bench as shown in fig.6

4.2 Hardware Utilization

The designed reverse converter is synthesized using Xilinx project navigator for device xc2s50-6tq144 the resource utilization.

Table 1. Resource utilization

Device Utilization Summary (Estimated Values)			
Logic Utilization	Used	Available	% Utilization
Number of occupied slices	8	768	1%
Number of 4input LUT's	14	1,536	1%
Number of slices	8	8	100%

containing only related logic			
Number of bonded IOB's	24	92	26%
Total equivalent gate count for design	84	--	--

4.3 RTL Schematic of Top Level Entity

In digital circuit design, Register Transfer Level (RTL) is a design abstraction which models a synchronous digital circuit in terms of the flow of digital signals (data) between hardware registers, and the logical operations performed on those signals. It is used in hardware description languages. The RTL Schematic generated by Xilinx for reverse converter is shown in fig 5

Fig.5: Block diagram of the synthesized reverse converter

Fig.6: timing waveform of reverse conver

4.3 Synthesize Result

Table.2. shows the synthesise result for using Xilinx project navigator for device xc2s50-6tq144

Table 2. Synthesize result

	EXISTING	PROPOSED
POWER(mW)	11.35	9.29
DELAY(ns)	15.50	7.352

Fig.6: performance and power trad-off of different system

5. CONCLUSION

This brief presents a method that can be applied to most of the current reverse converter architectures to enhance their performance and adjust the cost/performance to the application specifications. Furthermore, in order to provide the required trade-offs between performance and cost, new Parallel distributed arithmetic convolution technique components are introduced. These components are specially designed for reverse converters. Implementation results show that the reverse converters based on the suggested components considerably improve the speed when compared with the original converters, which do not use any adder, and reduce the power consumption compared with the converters that exclusively adopt adders.

6. 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 product computation," *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 $(2^n - 1, 2^n, 2^n + 1)$," *IEEE Trans. Signal Process.*, vol. 50, no. 7, pp. 1772–1779, Jul. 2002.
- [9] A. S. Molahosseini and K. Navi, "A reverse converter for the enhanced moduli set $\{2^n - 1, 2^n + 1, 2^{2n}, 2^{2n+1} - 1\}$ using CRT and MRC," in *Proc. IEEE Comput. Soc. Annu. Symp. VLSI*, Jul. 2010, pp. 456–457.
- [10] L. Sousa and S. Antao, "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 Trans. Circuits Syst. II*, vol. 59, no. 4, pp. 244–248, Apr. 2012.
- [11] L. Sousa and S. Antão, "On the design of RNS reverse converters for the four-moduli set $\{2^{n+1}, 2^n - 1, 2^n, 2^{n+1} + 1\}$," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 21, no. 10, pp. 1945–1949, Oct. 2013.
- [12] 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 $(2^n - 3, 2^n + 1, 2^n - 1, 2^n + 3)$," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 51, no. 3, pp. 152–155, Mar. 2004.

- [13] 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.
- [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 $(2^n - 3, 2^n + 1, 2^n - 1, 2^n + 3)$," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 51, no. 3, pp. 152–155, Mar. 2004.
- [15] B. Cao, C. H. Chang, and T. Srikanthan, "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, Fundam. Theory Appl.*, vol. 50, no. 10, pp. 1296–1303, Oct. 2003.
- [16] A. S. Molahosseini, K. Navi, C. Dadkhah, O. Kavehei, and S. Timarchi, "Efficient reverse+ converter+ designs+ for the+new 4-moduli+sets $\{2^n - 1, 2^n, 2^n + 1, 2^{2n} + 1\}$ and $\{2^n - 1, 2^n, 2^{2n}, 2^{2n} + 1\}$ based on new CRTs," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 57, no. 4, pp. 823–835, Apr. 2010.
- [17] J. Sklansky, "Conditional sum addition logic," *IRE Trans. Electron. Comput.*, vol. 9, no. 6, pp. 226–231, Jun. 1960.
- [18] 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.
- [19] 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.
- [20] S. J. Piestrak, "A high speed realization of a residue to binary converter," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 42, no. 10, pp. 661–663, Oct. 1995.
- [21] B. Ramkumar and H. M. Kittur, "Low power and area efficient carry select adder," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 20, no. 2, pp. 371–375, Feb. 2012.
- [22] R. Zimmermann, "Efficient VLSI implementation of modulo $(2^n \pm 1)$ addition and multiplication," in *Proc. 14th IEEE Int. Symp. Comput. Arithmetic*, Apr. 1999, pp. 158–167.
- [23] R. A. Patel, M. Benaissa, and S. Boussakta, "Fast parallel-prefix architectures for modulo $2^n - 1$ addition with a single representation of zero," *IEEE Trans. Comput.*, vol. 56, no. 11, pp. 1484–1492, Nov. 2007.
- [24] H. Kunz and R. Zimmermann, "High-performance adder circuit generators in parameterized structural VHDL," Integr. Syst. Lab., ETH Zürich Univ., Zürich, Switzerland, Tech. Rep