Reverse Converter by Parallel Prefix Adders

Divyashree L.K.[1], Pramod P.[2]

[1]*Student, M.Tech, VLSI Design & Signal Processing, ECE Department, LBS College of Engineering, Kasaragod, India*

[2]*Assistant Professor, ECE Department, LBS College of Engineering, Kasaragod, India*

***Abstract*—**The design of reverse converters based on regular and modular parallel prefix adders is analyzed. The VLSI implementation results show a significant delay reduction and area × time^2 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 nowadays systems. To solve this problem, hybrid parallel-prefix based adder components are presented to design reverse converters which gives trade off between delay and power consumption. A methodology is also described to design reverse converters based on different kinds of prefix adders. This methodology helps the designer to adjust the performance of the reverse converter based on the target application and existing constraints. The proposed structures are synthesized in Xilinx ISE using VHDL.

***Keywords*—**Reverse converter, Parallel Prefix Adder (PPA), HRPX, HMPE

I. INTRODUCTION

In the world of 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 for several applications, including digital signal processing and cryptography. However, its real usage requires forward and reverse converters to be combined in the previous digital systems. The reverse conversion, i.e., residue to binary conversion, is a hard and time-consuming operation.

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) investigate new algorithms and novel arithmetic formulations to achieve simplified conversion formulas and 2) introduce new moduli sets, which can lead to more simple formulations [1]-[2]. 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, we present a complete methodology to knowingly employ parallel-prefix adders in thoroughly 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 considerable growing of power consumption makes the reverse converter not competitive.

Two low power and area hybrid PPA are done to challenge with drawbacks that leads to considerable reduction of the power delay product (PDP) metric and considerable improvements in the area-time2 product (AT^2) when comparing with the original converters without using parallel-prefix adders.

The rest of this paper is organized as follows. Section II discusses related work on the parallel prefix adders. In Section III, new parallel-prefix-based components is explained, Section IV describes reverse converter methodology while Section V gives the results and discussion. Finally, the conclusion is given in section VI.

II. PARALLEL PREFIX STRUCTURE

The Residue number system consists of forward converter, modulo arithmetic units and reverse converter. On comparing with the other parts the reverse converter design is a complex and no modular structure. So more attention is needed in designing the reverse converter thereby preventing the slow operation and compromises the benefits of the RNS. The parallel prefix structure helps to achieve the faster operation in the reverse converter design but causes increased power consumption [3]. In the previous system, the novel specific hybrid parallel prefix adder based components are used to change the previous components there by decrease in the power consumption and makes faster operation.

A. Parallel Prefix Structure

The Parallel prefix structure consists of preprocessing block, prefix carry tree and post processing block. The parallel prefix adder operation begins with preprocessing stage by generating the Generate (Gi) and Propagate (Pi) equation(1) & (3).The prefix carry tree get proceeded with the previous block signal to yield all carry bit signal and these stage contains three logic complex cells such as Black cell, Gray cell and Buffer cell. Black cell compute both the propagate (P(i,j)) and generate (G(i,j)) by using the equation (3) & (4).The Gray cell executes only the generate(G(i,j)) [4].The carry bits generated in the second stage get passed to the post processing block there by generating the sum using the equation (5).The block diagram is shown in the Figure 1.

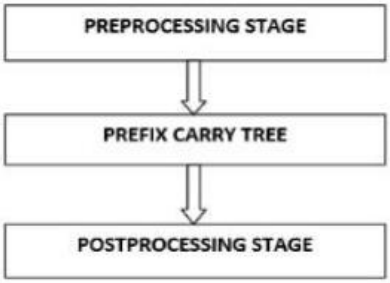

Fig.1 Basic Parallel prefix structure

$G_{m:n}=A_n$ AND B_n (1)
$G_0 = C_{in}$ (2)
$P_{m:n}=A_n$ XOR B_n (3)
$P_0=0$ (4)
$G_{m:n}=G_{n:k}$ OR Pn:k AND $G_{k-1:n}$ (5)
$P_{m:n}=P_{n:k}$ AND $P_{k-1:j}$ (6)
Sn =Pn XOR Cin (7)

The Brent Kung adder prefix structure is employed to achieve the higher speed with reduced power consumption. On comparing with the other parallel prefix adder structure the BK adder is chosen mainly for minimum fan-out and should be higher speed in operation than others [8]-[9]. Fig.2 shows the example BK adder prefix structure which uses the three basic cells in the prefix structure. These structure is elaborated for the proposed design having the modulo addition of (4n+1) for n=5.

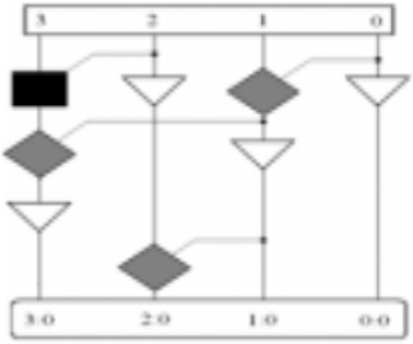

Fig.2:4-bit BK adder prefix structure

B. HRPX Structure :(Hybrid Regular Parallel prefix XOR/OR adder component)

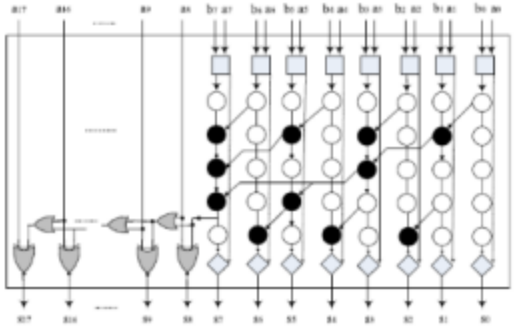
Fig.3 HRPX Structure using BK prefix network

Fig.3 shows HRPX Structure. The regular parallel prefix adder is used to perform the first part of addition and the simplified RCA logic is used perform the second part where the corresponding bits of the operand are fully variable. Full adder can be designed with XOR/OR gates because of the constant operand [5]-[6]-[7]. In these reverse converters design the carry chain is not needed and can be ignored. For most modulo sets (2^n-1) addition is a necessary operation [12]-[13]. The End around Carry (EAC) for (2^n-1) addition is represented with two zero, but for the reverse converter design one zero representation is required. To correct these zero representation problem, a detector circuit was employed in the design but it incorporates additional delay. So, the Binary to excess one converter (BEC) is used to solve the double zero representation issue.

III. NEW PARALLEL-PREFIX-BASED COMPONENTS

The HMPE Structure consists of Regular prefix adder and the Modified Excess One unit. The first two operands are added by the parallel prefix adder and the result is tentatively incremented based on the control signal generated by the prefix structure to assure the single zero representation. The figure 4 shows HMPE Structure.

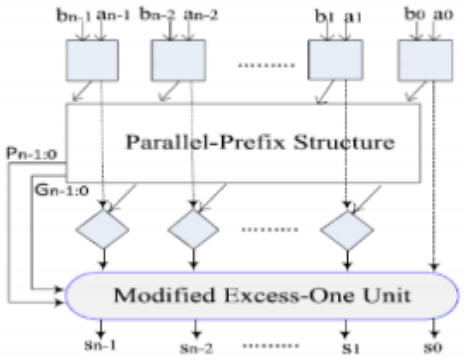

Fig.4: HMPE Structure

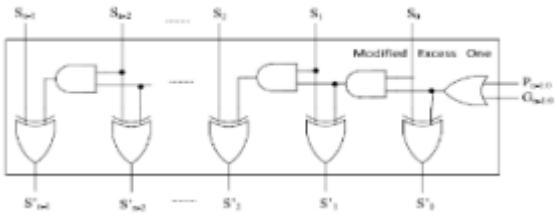

Fig.5: Modified Excess one unit

Figure 5 shows the Modified Excess One unit circuit diagram. The result generated by the prefix structure is conditionally incremented by this unit based on the control signal generated by the parallel prefix adder. The reverse converter design is implemented for (4n+1) modulo addition (n=5) designing the adder and also the multiplier by using the same adder design without using any parallel prefix multiplier structure for designing multiplier. In this design, the adder design is implemented by using the Kogge Stone adder parallel prefix structure. Here the first two operands are added by using the prefix adder pre-processing stage thereby generating the

propagate and generate equation. The first stage processed signal get passed to the next stage called the prefix carry tree, this stage again computes the generate and propagate equation by using the previous output and all the logic cells employed in the Kogge Stone adder network. These processed signals are passed to the post processing block.

IV. REVERSE CONVERTER DESIGN METHODOLOGY

In this section, the methodology of reverse converter design is described. In the following, a method employing distinct components in the architecture of the reverse converter will be presented. Several reverse converters for different moduli sets have been introduced, which can be classified into three classes. The first class consists of converters with a tree of CSAs with EAC followed by a two-operand modulo $2k - 1$ CPA. A second class includes more complex reverse converters, which have several CSAs and CPAs with EACs followed by a final regular subtractor with two operands of different size. The implementation of this subtractor using regular binary-adder results in one operand with some constant bits [14]. The third class covers the reverse converters that have been designed for moduli sets with moduli other than the popular $2n$ and $2n \pm 1$. In the following, we describe a methodology for designing reverse converters in the first and second classes. The suggested method for applying the HMPE and HRPX in the reverse converter is shown in Fig.6.

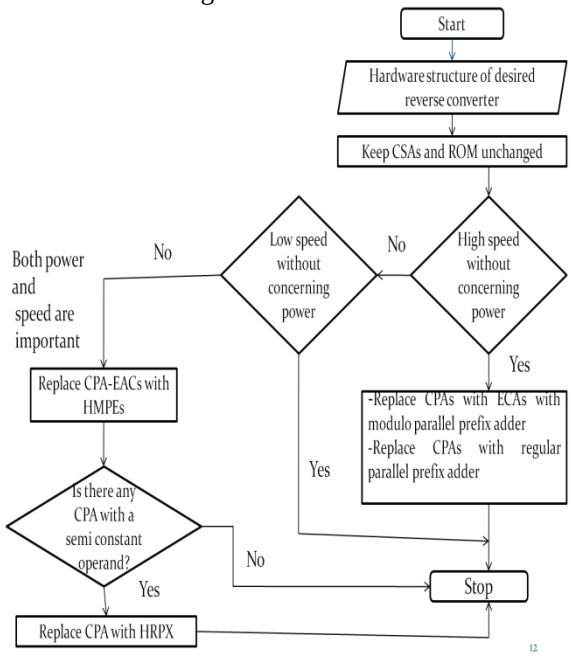

Fig.6: Reverse converter design methodology

If to achieve the low power consumption and hardware cost deprived of speed, no prefix adder is needed. On the other hand, if high speed is the designer goal, the CPAs with EAC and the regular CPAs should be replaced by traditional parallel prefix modulo $2n - 1$ adders and regular parallel-prefix adders, respectively. However, for the VLSI designers, a suitable trade-off between speed, power, and area is often more important. In this case, first, CPAs with the EAC can be replaced by the HMPEs. Then, if the converter contains a regular CPA where one of its operands has a string of constant bits with the value of one, it can be replaced with the HRPX.

V. RESULTS AND DISCUSSIONS

A reverse converter by parallel prefix adder is designed and simulated successfully. The design proposed has been synthesized and simulated using XILINX ISE 14.7 and ModelSim and the corresponding delay calculations have been noted. The simulation process has been carried out for different levels of abstraction. The code has been written in VHDL. By using Brent Kung Adder the delay was reduced. Simulation results are shown in fig 7 and RTL Schematic diagrams are shown in fig 8. Table I shows delay comparison of HRPX and HMPE structure.

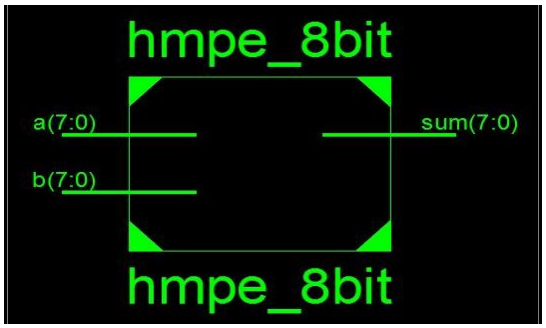

Fig. 7. RTL schematic of HMPE structure.

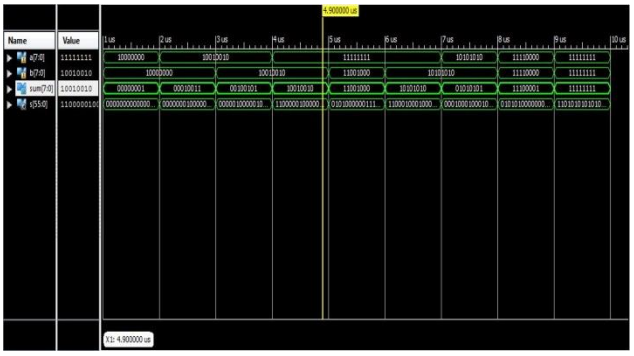

Fig. 8 Simulated output of HMPE Structure

TABLE I. SUMMARY DETAILS

METHOD	DELAY	NUMBER OF 4 INPUT LUTS
HRPX	9.145ns	23
HMPE-BK ADDERS	8.137ns	19

The result shows that HMPE structure has a lower delay (high speed) when compared to that of conventional one.

VI. CONCLUSION

A method of the reverse converter structure to improve their operation and adjust the cost/performance to the application specifications is done. In order to provide the required tradeoffs between performance and cost, new parallel-prefix-based adder components were 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 parallel-prefix adder, and reduce the power consumption compared with the converters that exclusively adopt parallel-prefix adders.

The circuits for all these configurations were designed and specified in the VHDL. Structural or behavioral descriptions can be considered. A structural VHDL description is adopted. The future scope for this work is measuring area and power for each design. Based on the area and power analysis, we will show the best modulo adder design. Finally comparison of existing system and proposed system will be done.

VII. REFERENCES

[1] L. Dadda (1965), "Some Schemes for Parallel Multipliers," Alta Frequenza, vol. 34, pp. 349–356.

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

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

[4] R. Zimmermann, "Efficient VLSI implementation of modulo (2n±1) addition and multiplication," in Proc. 14th IEEE Int. Symp. Comput. Arithmetic, Apr. 1999, pp. 158–167.

[5] R. A. Patel, M. Benaissa, and S. Boussakta, "Fast parallel-prefix architectures for modulo 2n − 1 addition with a single representation of zero," IEEE Trans. Comput., vol. 56, no. 11, pp. 1484–1492, Nov. 2007.

[6] K. Y. Sudheer, B. Rajendra Naik, "Design and Estimation of delay, power and area for Parallel prefix adders," IEEE Trans., vol. 06-08, March.2014.

[7] 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.

[8] 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.

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

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

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

[12] 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.

[13] S. Antao 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.

[14] 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.