International Journal of Engineering Innovation & Research
Volume 6, Issue 6, ISSN: 2277 – 5668

Low Power Area Efficient Reverse Converter Design Via Parallel Prefix Adder

Akanksha Gautam[1] and Sandeep Nemade[2]
[1]M.tech scholar, Dept. of EC, TIT Bhopal, akankshagoutam77@gmail.com, India
[2]Asst. Professor, Dept. of EC, TIT Bhopal, nemadesandeeptit@gmail.com, India
*Corresponding author email id: akankshagoutam77@gmail.com

Date of publication (dd/mm/yyyy): 06/12/2017

***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 × 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 nowadays systems. Hence, to solve the high power consumption problem, novel specific hybrid parallel-prefix-based adder components those provide better tradeoff between delay and power consumption. The power, area and delay of the proposed system are analysis using Xilinx 14.2.**

***Keywords* – Digital Arithmetic, Parallel-Prefix Adder, Residue Number System (RNS), Reverse Converter.**

I. INTRODUCTION

These Residue number systems (RNS) and the related arithmetic units are popular in many digital signal processing applications where most computations are restricted to multiplication, addition and subtraction [1]. Application areas, besides general RNS arithmetic, as noted in [2] include:

- Fast number theoretic transforms.
- Discrete Fourier transform.
- Digital filters.

Residue Number Systems (RNS) can represent large numbers with a set of smaller residues according to the assumed moduli set. Subsequently, arithmetic operations, e.g., addition and multiplication, can be performed on each residue independently without any need for carry propagations between them, which leads to the reduction of the carry propagation chain [1]. This facilitates the realization of high-speed and low-power arithmetic units. Therefore, RNS based arithmetic units could be of potential interest for embedded processors, such as those found in mobile devices, for which high speed and low-power consumption are critical. Furthermore, RNS is extremely appropriate for addition and multiplication dominated applications such as digital signal processing [2], Digital filtering [3], communications [4] and cryptography [5], all of which are extremely important in computing today. The main drawback associated to RNS based computation however relates to the overhead introduced by the input and output conversions from binary to RNS and vice versa [6]. With the traditional converters without using parallel prefix adders.

Power dissipation has become one of the major limiting factors in the design of digital ASICs. Low power dissipation will increase the mobility of the ASIC by reducing the system cost, size and weight. DSP blocks are a major source of power dissipation in modern ASICs. The residue number system (RNS) has, for a long time, been proposed as an alternative to the regular two's complement number system (TCS) in DSP applications to reduce the power dissipation. Some research have shown that implementing FIR filters in residue number system (RNS) instead of two's complement number system (TCS) can give a reduction in power dissipation. FIR filters are among the less complex DSP blocks. A general sketch of how RNS computations can be performed is shown in figure 1.

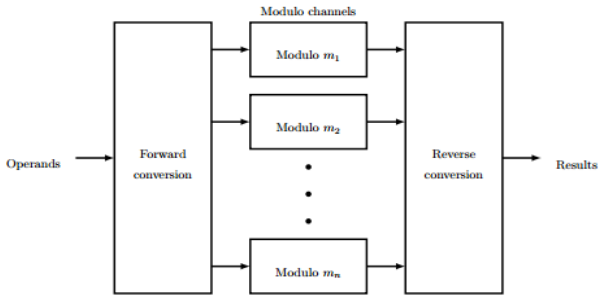

Fig. 1. The basic principle of RNS

II. PARALLEL-PREFIX STRUCTURE

Parallel-prefix structures are found to be common in high performance adders because of the delay is logarithmically proportional to the adder width.
PPA's basically consists of 3 stages.

- Pre computation.
- Prefix stage.
- Final computation.

I.1. Pre Computation

In pre computation stage, propagates and generates are computed for the given inputs using the given equations (1) and (2).

$$Pi = Ai \text{ XOR } Bi \quad (1)$$
$$Gi = Ai \text{ AND } Bi \quad (2)$$

I.2. Prefix Stage

In the prefix stage, group generate/propagate signals are computed at each bit using the given equations.

The black cell (BC) generates the ordered pair and the gray cell (GC) generates only left signal.

I.3. Final Computation

In the final computation, the sum and carryout are the final output.

$$Si = Pi \text{ and } Gi\text{-}1\text{:-}1$$
$$Cout = Gn\text{:-}1$$

Where "-1" is the position of carry-input. The generate/propagate signals can be grouped in different fashion to get the same correct carries. Based on different ways of grouping the generate/propagate signals, different prefix architectures can be created.

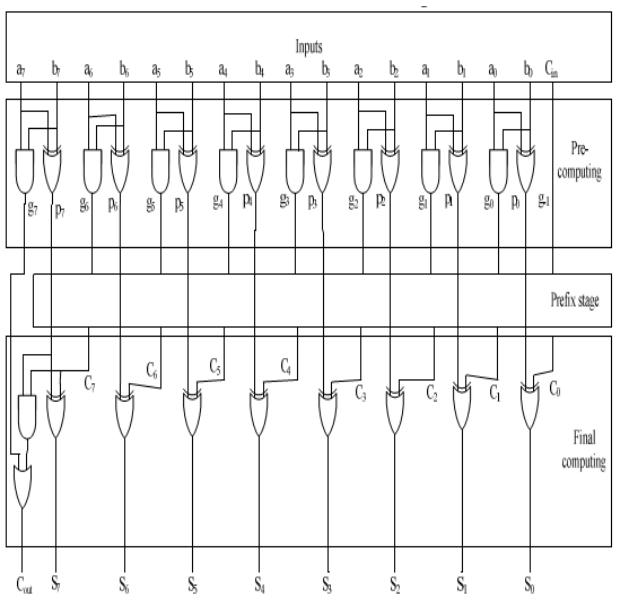

Fig. 2. Parallel-Prefix Structure with carry save notation

III. Proposed Methodology

The main reason for the high power consumption and area overhead of these adders is the recursive effect of generating and propagating signals at each prefix level. However, this method suffers from high fan-out, which can make it usable only for small width operands. However, we could address this problem by eliminating the additional prefix level and using a modified excess-one unit instead. In contrast to the BEC, this modified unit is able to perform a conditional increment based on control signals as shown in Fig. 3, and the resulted hybrid modular parallel-prefix excess-one (HMPE) adder is depicted in Fig. 4. The HMPE consists of two parts:

1) A regular prefix adder.
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.

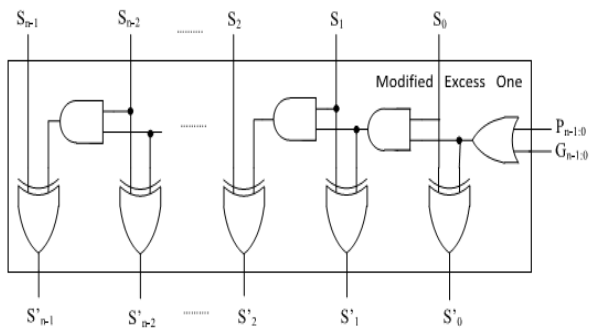

Fig. 3. Modified excess-one unit

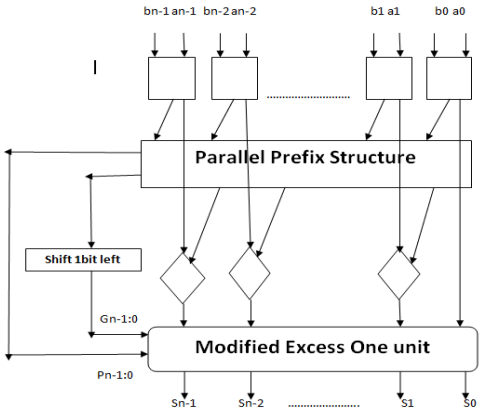

Fig. 4. HMPE structure

IV. Simulation Result

The power, area and delay of the proposed system are analysis using Xilinx 14.2. The simulation results are as follows:

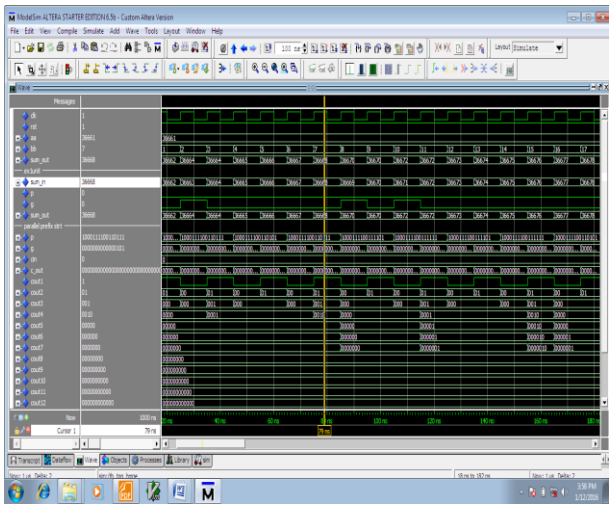

Fig. 5. Simulation output

This figure 5 shows the simulation output of proposed system. That is reversible converter design via parallel prefix adder.

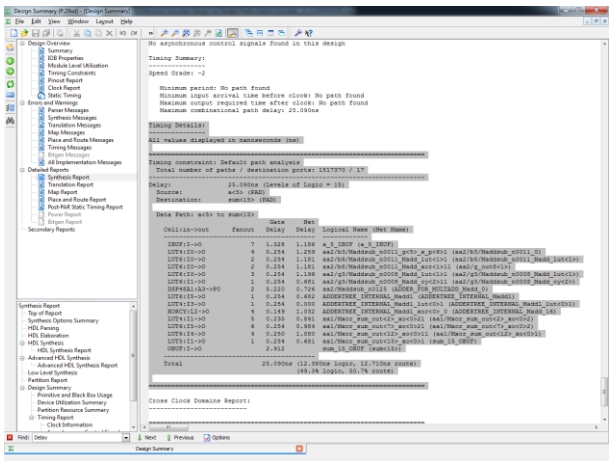

Fig. 6. Delay report

This figure 6 shows the delay report of proposed system. In this fig. delay is 25.090.

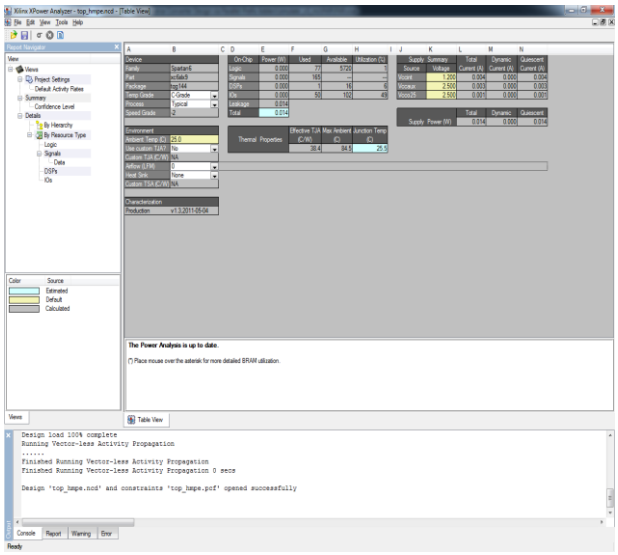

Fig. 7. Power report.

This figure 7 shows the power report of proposed system. In this fig. power is 0.014.

Table 1. Synthesis report

Synthesis Status	Analysis
Total logic elements	1584
Total combinational function	1536
Dedicated logic register	98
Total register	98
Total pins	146
Total virtual pins	0
Total memory bits	96
Embedded Multiplier 9-bit elements	0
Total PLLs	0
Power	0.014
LUTs	77
Number of slice register	13
Number of Flip-Flop	0
Number of AND/OR logic	13

Table 1 shows the synthesis report of proposed work. In this table shows the used are no. of logic register, logic elements, total pins, total PLLs, total combinational function and proposed architecture power is 0.014.

V. Conclusion

This proposed work presents parallel-prefix-based adder components that give better tradeoff in area and delay are thus exhibited to design reverse converters. A methodology is described to design reverse converters depending on various types of prefix adders. 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. He use of modular and regular parallel-prefix adders proposed in this brief in reverse converters highly decrease the delay at the expense of significantly more power and circuit area, whereas the proposed prefix-based adder components allows one to achieve suitable tradeoffs between speed and cost by choosing the right adders for the parts of the circuits that can benefit from them the most. Our proposed work used power is 0.014w.

References

[1] A.A. E. Zarandi, A.S. Molahosseini, M. Hosseinzadeh, S. Sorouri, S. Antao, and Leonel Sousa "Reverse Converter Design via Parallel-Prefix Adders: Novel Components, Methodology, and Implementations" IEEE Tra. on Very Large Slale Int. Sys. Pp. 1-5 2015.

[2] Somayeh Timarchi, Mahmood Fazlali1, and Sorin D. Cotofana, "Unified Addition Structure for Moduli Set {2n-1, 2n, 2n+1} based on a Novel RNS Representation" IEEE pp. 247-252 2010.

[3] Ghassem Jaberipur "On Building General Modular Adders from Standard Binary Arithmetic Components" The CSI Journal on Computer Science and Engineering Vol. 4, No. 2 & 4, Pp. 10-16, 2006.

[4] Saeid Banhanfar and Nadali Zarei "Reverse Converter for the Moduli Set {2n-1, 2n, 2n + 1} base on Grouping Number", IJCSI International Journal of Computer Science Issues, Vol. 10, Issue 6, No 1, November 2013.

[5] Chan Hua Vun, Senior Member, IEEE, Annamalai Benjamin Premkumar, Senior Member, IEEE and Wei Zhang, Member, IEEE, "A New RNS based DA Approach for Inner Product Computation", IEEE Trans. Circuits and Systems — I: Regular Papers, vol. 60, no. 8, August 2013.

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

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