International Journal of Research
Available at https://edupediapublications.org/journals
p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 03 Issue 17
November 2016

Design an reverse converter parallel prefix novel adder

[1]Mr. **P Yuva Kishore Reddy**, M.Tech.,VLSI & ES, E.C.E Dept.,Chintalapudi Engineering College, Ponnur, Guntur Dt

[2]Mr.**A.Trinadha Rao,** Assistant Professor, E.C.E Dept.,ChintalapudiEngineeringCollege,Ponnur, Guntur Dt

[3]Mr.**K. Anka Siva Prasad**, Associate Professor & Vice Principal, E.C.E Dept.,Chintalapudi Engineering College,Ponnur,GunturDt

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 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**

INDEX TERMS: Digital arithmetic, parallel-prefix adder (PPX), parallel distributed arithmetic convolution architecture, reverse converter.

I 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-
2) known adder architectures, such as carry-save adders (CSAs) and ripplecarry
3) 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.

The 3-bit ripple carry adder is shown in Fig.1.The first bit carry is given to second bit full adder and similarly the second bit carry is given to the third bit full adder. The addition operation is performed from least significant bit to most significant bit in ripple carry adder. Configuration logic and routing resources in Field Programmable Gate Array.

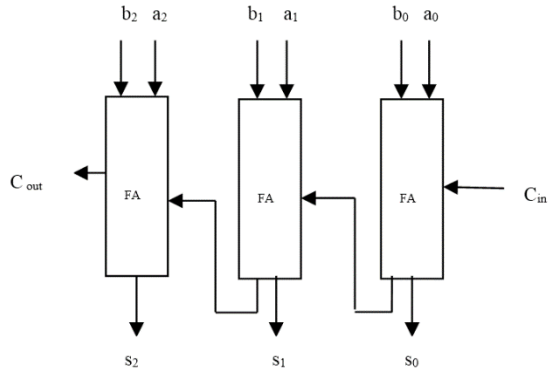

Fig.1: Three Bit Ripple Carry Adder

II LADNER-FISCHER ADDER

The Ladner-Fischer is the parallel prefix adder used to perform the addition operation. It is looking like tree structure to perform the arithmetic operation. Ladner Fischer adder is used for high performance addition operation. The Ladner-Fischer adder consists of black cells and gray cells. Each black cell consists of two AND gates and one OR gate. Multiplexer is combinational circuit which consists of multiple inputs and a single output. Each gray cell consists of only one AND gate.P_i denotes propagate and it consists of only one AND gate given in equation 1. G_i denotes generate and it consists of one AND gate and OR gate given in equation 2.

$$P_i=B_i \text{ AND } B_{i-1} \text{ ---------- (1)}$$

$$G_i=A_i \text{ OR } [B_i \text{ AND } A_{i-1}] \text{ ---- (2)}$$

G_i denotes generate and it consists of one AND gate and OR gate given in equation 3.

$$G_{i-1}=A_{i-2} \text{ OR } [B_{i-2} \text{ AND } A_{i-1}] \text{ --- (3)}$$

IIIPROPOSED LADNER FISCHER ADDER

The proposed Ladner-Fischer adder is flexible to speed up the binary addition and the structure looks like tree structure for the high performance of arithmetic operations.

Research on binary operation elements and motivation gives development of devices. Field programmable gate arrays [FPGA's] are most popular in recent years because theyimprove the speed of microprocessor based applications like mobile DSP and telecommunication. The construction of efficient Ladner-Fischer adder consists of three stages. They are pre-processing stage, carry generation stage, post-processing stage.

Pre-Processing Stage:

In the pre-processing stage, generate and propagate are from each pair of inputs. The propagate gives "XOR" operation of input bits and generates gives "AND" operation of input bits. The propagate (P_i) and generate (G_i) are shown in below equations 4 & 5.

$$P_i=A_i \text{ XOR } B_i \text{ ------ (4)}$$

$$G_i=A_i \text{ AND } B_i \text{ ------ (5)}$$

Carry Generation Stage:

In this stage, carry is generated for each bit and this is called as carry generate (C_g). The carry propagate and carry generate is generated for the further operation but final cell present in the each bit operation gives carry. The last bit carry will help to produce sum of the next bit simultaneously till the last bit. The carry generate and carry propagate are given in below equations 6 & 7.

$$C_p=P_1 \text{ AND } P_0 \text{ ---------- (6)}$$

$$C_g=G_1 \text{ OR } (P_1 \text{ AND } G_0) \text{ ------ (7)}$$

The above carry propagateC_p and carry generation $C_g$in equations 6&7 is black cell and the below shown carry generation in equation 8 is gray cell. The carry propagate is generated for the further operation but final cell present in the each bit operation gives carry. The last bit carry will help to produce sum of the next bit simultaneously till the last bit. This carry is used for the next bit sum operation, the carry generate isgiven in below equations 8.

$$C_g=G_1 \text{ OR } (P_1 \text{ AND } G_0) \text{ ------ (8)}$$

Post-Processing Stage:

It is the final stage of an efficient Ladner-Fischer adder, the carry of a first bit is XORed with the next bit of propagates then the output is given as sum and it is shown in equation 9.

$$S_i=P_i \text{ XOR } C_{i-1} \text{ ------- (9)}$$

It is used for two sixteen bit addition operations and each bit carry is undergoes post-processing stage with propagate, gives the final sum.

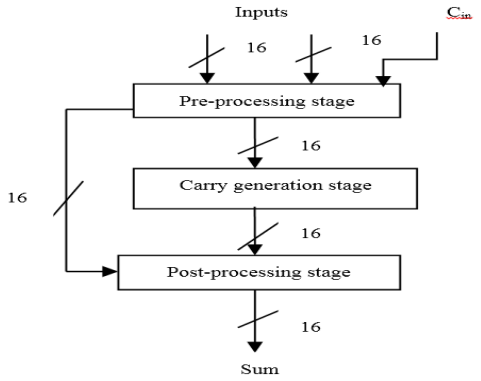

Fig.2: Block Diagram

The first input bits goes under pre-processing stage and it will produce propagate and generate. These propagates and generates undergoes carry generation stage produces carry generates and carry propagates, these undergoes post-processing stage and gives final sum. The step by step process of efficient Ladner-Fischer adder is shown in Fig.2.

The efficient Ladner-Fischer adder structure is looking like tree structure for the high performance of arithmetic operations and it is the fastest adder which focuses on gate level logic. It designs with less number of gates. So, it decreases the delay and memory used in this architecture.

The efficient Ladner-Fischer adder isshown in fig.3 which improves the speed and decrease the area for the operation of 8-bit addition. The input bits A_i and B_i concentrates on generate and propagate by XOR and AND operations. These propagates and generates undergoes the operations of black cell and gray cell and gives the carry C_i. That carry is XORed with the propagate of next bit, that gives sum.

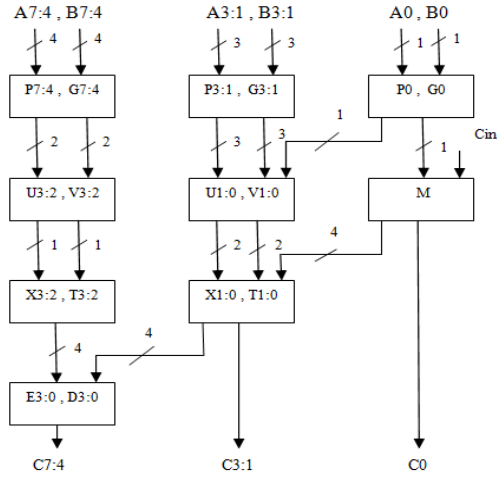

Fig.3: 8-Bit Efficient Ladner-Fischer Adder

The architecture ofEfficient Ladner-Fischer adder gives the less delay and less memory for the operation of 16-bit addition. The properties of the operations are evaluated in parallel and accept the trees to overlap which leads to parallelization.

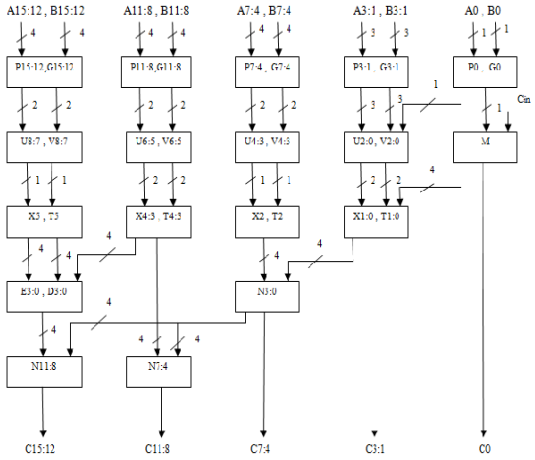

Fig.4: 16-bit Efficient Ladner-Fischer Adder

The architecture of 16-bit Efficient Ladner-Fischer adder is shown in Fig.4. The logical circuit is using multiple adders to find the sum of N-bit numbers. Each addition operation has a carry input (C_{in}) which is the previous bit carry output (C_{out}).

Research on binary addition motivates gives development of devices. Many parallel prefix networks describe the literature of addition operation.

The parallel prefix adders are Brent-kung, Kogge-stone, ladner-Fischer, Sklansky, etc,. The fast and accurate performance of an adder is used in the very large scale integrated circuits design and digital signal processors.

IV SIMULATION RESULTS

The Efficient Ladner-Fischer adder is designed on VHDL (very high speed integration hardware description language). Xilinx project navigator 12.1 is used for synthesis. Simulation results are shown in Fig.5.

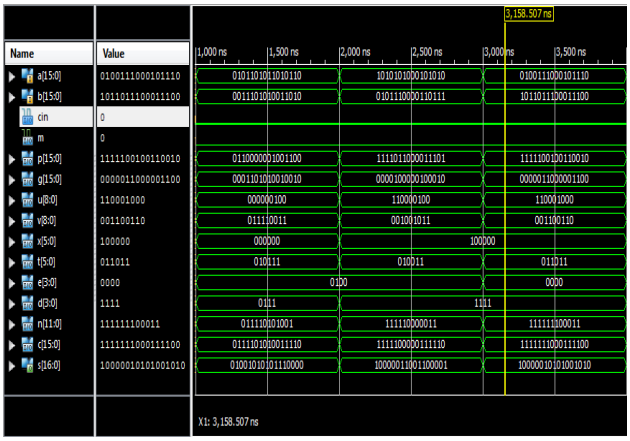

Fig.5: 16-Bit Efficient Ladner-Fischer Adder Simulation Waveform

The design of adders is done on VHDL. The memory and delay performance Efficient Ladner-Fischer adder (ELF) is shown in Table.1

Adder	Delay(ns)	Memory used(kb)
8-bit Efficient Ladner-Fischer adder	11.2	186228
16-bit Efficient Ladner-Fischer adder	12.2	188788

Table.1Delay and memory used in ELF

V CONCLUSION

In this paper, a new approach to design an efficient Ladner-Fischer adder concentrates on gate levels to improve the speed and decreases the area.It is like tree structure and cells in the carry generation stage are decreased to speed up the binary addition. The proposed adder addition operation offers great advantage in reducing delay.

REFERENCES

[1] pakkiraiahchakali, madhukumarpatnala"Design of high speed Ladner - Fischerbased carry select adder" IJSCE march 2013.

[2] David h,k hoe, Chris Martinez and srijyothsnavundavalli"Design and characterization of parallel prefix adders using FPGAs", Pages.168-172, march2011 IEEE.

[3] K.Vitoroulis and A.J. Al-Khalili, "performance of parallel prefix adders implemented with FPGA technology," IEEE Northeast Workshop on circuits and systems, pp.498-501, Aug. 2007.

[4]GiorgosDimitrakopoulos and DimitrisNikolos, "High-Speed Parallel-Prefix VLSI Ling Adders" Feb. 2005.

[5] S.Knowles,"Afamilyofadders," Proc.15th Symp. Comp. Arith.,pp.277-281,June2001.

[6] I. Sutherland, R. SproullandD. Harris, LogicalEffort,SanFrancisco:MorganKaufmann,1999.

[7] R.BrentandH.Kung,"Aregularlayoutforparallel adders,"IEEETrans.Computers,vol.C-31,no.3, pp.260-264,March1982

[8] R.E. Ladner and M.J. Fischer, "Parallel Prefix Computation," J. ACM, vol. 27, no. 4, pages 831-838, Oct. 1980.

[9] P.Kogge and H.Stone,"A Parallel algorithem for the efficient solution of a general class of recurrence relation," IEEE transactions on computers, vol. c-22, no.8, pp.786-793, Aug 1973.

[10]J.Sklansky, "Conditional-sumadditionlogic," IRETrans.ElectronicComputers,vol.EC-9,pp.226-231,June1960.