INTERNATIONAL JOURNAL OF PURE AND APPLIED RESEARCH IN ENGINEERING AND TECHNOLOGY

A PATH FOR HORIZING YOUR INNOVATIVE WORK

DESIGN AND IMPLEMENTATION OF HIGH SPEED VLSI ADDER USING LING EQUATIONS

PRADNYA JADHAV[1], POONAM CHAUDHARI[2], NIVEDITA JOSHI[3], MINAL S. GHUTE[4]

1. Asstt. Prof. Deptt. of Electronics Engg., YCCE, Nagpur.
2. Asstt. Prof. Deptt. of Electronics Engg., YCCE, Nagpur.
3. Asstt. Prof. Deptt. of Electronics & Telecomm. Engg., YCCE, Nagpur.
4. Asstt. Prof. Deptt. of Electronics & Telecomm. Engg., YCCE, Nagpur.

Accepted Date: 27/02/2014 ; Published Date: 01/05/2014

Abstract: Parallel-prefix adders offer a highly efficient solution to the binary addition problem and are well-suited for VLSI implementations. In this paper, a novel framework is introduced, which allows the design of parallel-prefix Ling adders. The proposed approach saves one-logic level of implementation compared to the parallel-prefix structures proposed for the traditional definition of carry look ahead equations and reduces the fan out requirements of the design. Experimental results reveal that the proposed adders achieve delay reductions of up to 14 percent when compared to the fastest parallel-prefix architectures presented for the traditional definition of carry equations.

Keywords: Adders, Parallel-prefix carry computation, Computer arithmetic, VLSI adder, Ling adder.

Corresponding Author: MS. PRADNYA JADHAV

Access Online On:

www.ijpret.com

How to Cite This Article:

Pradnya Jadhav, IJPRET, 2014; Volume 2 (9): 526-531

PAPER-QR CODE

-

INTRODUCTION

BINARY addition is one of the primitive operations in computer arithmetic. VLSI integer adders are critical elements in general purpose and digital-signal processing processors since they are employed in the design of Arithmetic-Logic Units, in floating-point arithmetic data paths and in address generation units. They are also employed in encryption and hashing function implementation. A large variety of algorithms and implementations have been proposed for binary addition. When high operation speed is required, tree structures, like parallel-prefix adders, are used. Parallel-prefix adders are suitable for VLSI implementation since they rely on the use of simple cells and maintain regular connections between them. The prefix structures allow several tradeoffs among the number of cells used, the number of required logic levels, and the cells' fan-out. A recent comparison of the most efficient adder architectures has been presented in. Several variants of the carry-look ahead equations, like Ling carries, have been presented that simplify carry computation and can lead to faster structures. Adders form an almost indispensable component of every contemporary integrated circuit. To cope with varying requirements of time and area efficiency, several adder architectures have appeared ranging from the smallest ripple-carry adders with the linear to the operand length delay up to the Carry Look-Ahead (CLA), conditional-sum and parallel-prefix adders which provide the fastest possible implementations at the expense of the largest circuit sizes. Between these two categories lie the carry-skip and carry-select architectures, which give a good alternative, since they combine relatively small area and substantially reduced delays. All these architectures can be thought as alternative ways of solving the problem of computing a carry signal at each bit position of the result. Ling [8] on the other hand, proposed instead of having a single signal at each bit position for encoding the carry, to allow this encoding to be spread in two signals, relaxing the carry computation unit of some of its complexity.

II. PERVIOUS WORK

The structure of the prefix network specifies the type of the PPA. The Prefix network described by Haiku Zhu, Chung-Kuan Cheng and Ronald Graham , has the minimal depth for a given 'n' bit adder. Optimal logarithmic adder structures with a fan-out of two for minimizing the area-delay product is presented by Matthew Ziegler and Mircea Stan. The Sklansky adder presents a minimum depth prefix network at the cost of increased fan-out for certain computation nodes. The algorithm invented by Kogge-Stone has both optimal depth and low fan-out but produces massively complex circuit realizations and also account for large number of interconnects. Brent-Kung adder has the merit of minimal number of computation nodes, which yields in reduced area but structure has maximum depth which yields slight increase in latency when

-

compared with other structures. The Han-Carlson adder combines Brent-Kung and Kogge-Stone structures to achieve a balance between logic depth and interconnect count. Knowles presented class logarithmic adders with minimum depth by allowing the fan-out to grow. Ladner and Fischer proposed a general method to construct a prefix network with slightly higher depth.

when compared with Sklansky topology but achieved some merit by reducing the maximum fan-out for computation nodes in the critical path. Related work on PPA literature such as Ling adder , achieve improved performance gains by changing the equation of the dot operator ‘ • ’.

III. IMPLEMENTATION

A .Parallel-Prefix Formulation Of Ling Addition:

This is a systematic methodology that allows the parallel-prefix computation of Ling carries. In order to describe the proposed approach, at first an 8-bit adder will be used as an example. The Ling carries at the fourth and the fifth bit position are equal to,

$$H4 = g4 + g3 + p3.g2.g1 + p3.p2.p1.go$$

$$H5 = g5 + g4.g3 + p4.p3.g2 + p4.p3.p2.g1 + p4.p3.p2.p1.go$$

Rewriting using basic definition,

$$H4 = g4 + g3 + p3.p2.(g2 + g1) + p3.p2.p1.p0.g0$$

$$H5 = g5 + g4 + p4.g3 + p4.p3.g2 + p4.p3.p2.g1 + p4.p3.p2.p1.g0$$

Assuming that,

$$G_i^* = g_i + g_{i-1} \qquad and \qquad p_i^* = p_i \cdot p_{i-1}$$

Rewriting equations,

$$H4 = G_4^* + p_3^*.G_2^* + p_3^* \cdot p_1^i.G_0^*$$

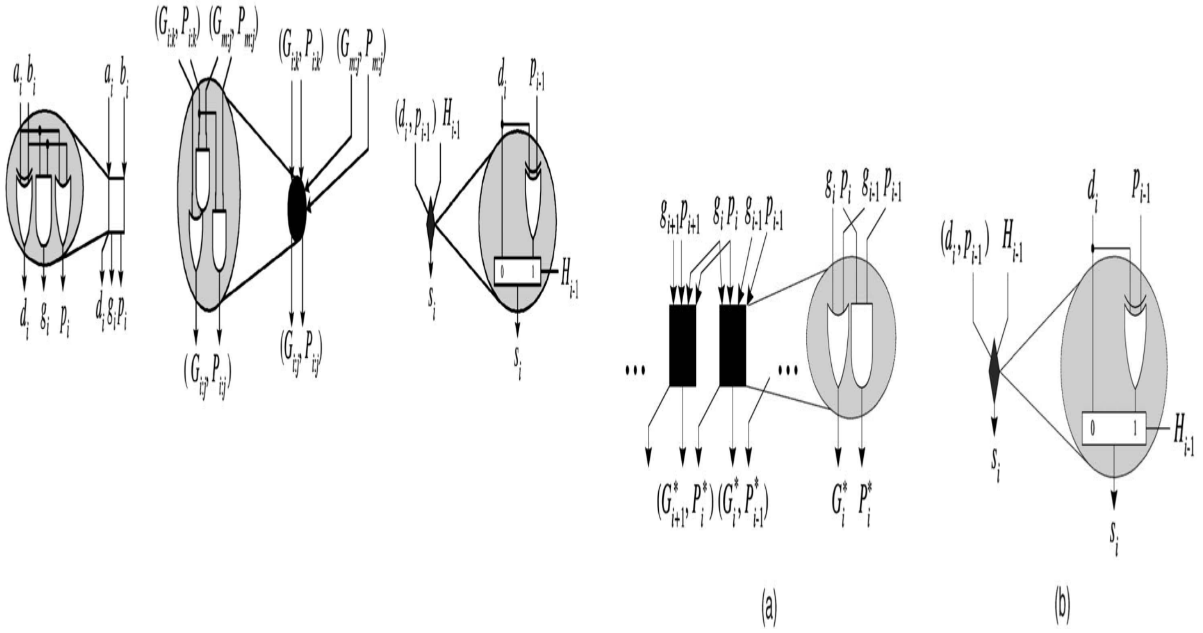
$$H5 = G_5^* + p_4^*.G_3^* + p_4^* \cdot p_2^i.G_1^*$$

Therefore, by using the intermediate generate and propagate pairs and by treating separately the Ling carries of the even and the odd-indexed bit positions, each carry Hi, in the case of an 8-bit adder, can be derived using the operator ‘o’.

-

The generation of intermediate generates and propagates pairs and the new cell used for the computation of sum bit in the case of a Ling adder,

The logic level implementations of the basic cells used in parallel prefix carry computation is given bellow,


$a_i\ b_i$
$a_i\ b_i$
$d_i\ g_i\ p_i$
$d_i\ g_i\ p_i$
$(G_{i:k}, P_{i:k})\ (G_{m:j}, P_{m:j})$
$(G_{i:k}, P_{i:k})\ (G_{m:j}, P_{m:j})$
$(G_{i:j}, P_{i:j})$
$(G_{i:j}, P_{i:j})$
$(d_i, p_{i-1})\ H_{i-1}$
$d_i\ p_{i-1}$
0
1
H_{i-1}
s_i
s_i
$g_{i+1}\ p_{i+1}\ g_i\ p_i\ g_{i-1}\ p_{i-1}$
$g_i\ p_i\ g_{i-1}\ p_{i-1}$
...
...
$(G^*_{i+1}, P^*_i)\ (G^*_i, P^*_{i-1})$
$G^*_i\ P^*_i$
$(d_i, p_{i-1})\ H_{i-1}$
$d_i\ p_{i-1}$
0
1
H_{i-1}
s_i
s_i
(a)
(b)

B. Hybrid Parallel-Prefix/Carry-Select Ling Adders:

The goal for high-speed adder architectures with reduced area and wiring has led to the design of hybrid parallel-prefix/carry-select adders. Fig. 3 shows a hybrid 32-bit adder which employs a Kogge-Stone parallel-prefix structure for the generation of the carries c4k,

k =1; 2; . . . ; n=4, and 4-bit carry select blocks. The carry select block computes two sets of sum bits, i.e s_i^0, s_i^1 and the final sums are selected via a multiplexer according to the value of c4k. The goal of such hybrid structures is to overlap the time required for the computation of the carries at the boundaries of the carry select blocks with the time needed to derive the sum bits.

-

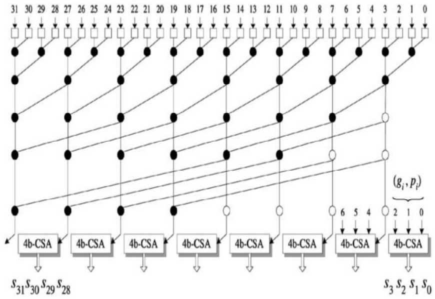


Fig 3: A 32 –bit Hybrid Parallel-Prefix/Carry-Select Ling Adders

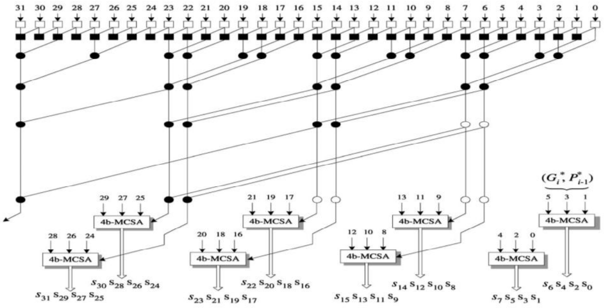
The design of hybrid parallel-prefix/carry-select Ling adders requires some minor modifications to the carry-select block. This is required since,

1. The proposed prefix structures generate the Ling pseudo carries Hi instead of the real carries ci and, thus, a sum bit cannot be directly selected according to the value of Hi.

2. The carries and the sum bits of the even and odd bit positions are generated separately.

3. The carry-select blocks take as inputs the pairs

The equivalent 32-bit hybrid Ling adder is shown in Fig. 4. The Ling carries are computed on the corresponding even and odd bit positions and used to select the final sum bits that have been concurrently produced by the 4-bit Modified Carry-Select Adders(MCSA).



CONCLUSION

A systematic methodology for designing parallel-prefix Ling adders has been introduced in this paper.

-

After simulation of two structures the timing analysis is performed on Xilinx 13.4 ISE suite. The maximum combinational delay for Ling adder is 15.384ns where as for PPCs traditional adder is 21.869ns.from the results achieved it is clear that proposed adder reduces delay by 6.685ns.

The proposed adders preserve all the benefits of the traditional parallel-prefix carry computation units, while, at the same time, offering reduced delay and fan-out requirements. Hence, high-speed datapaths of modern microprocessors can truly benefit from the adoption of the proposed adder architecture.

VI. **REFERENCES**

1. P.M. Kogge and H.S. Stone, "A Parallel Algorithm for the Efficient Solution of a General Class of Recurrence Equations," IEEE Trans. Computers, vol. 22, no. 8, pp. 786-792, Aug. 1973.

2. R.E. Ladner and M.J. Fisher, "Parallel Prefix Computation," J. ACM, vol. 27, no. 4, pp. 831-838, Oct. 1980.

3. R.P. Brent and H.T. Kung, "A Regular Layout for Parallel Adders," IEEETrans. Computers, vol. 31, no. 3, pp. 260-264, Mar. 1982.

4. T. Han and D. Carlson, "Fast Area-Efficient VLSI Adders," Proc. Symp. Computer Arithmetic, pp. 49-56, May 1987.

5. S. Knowles, "A Family of Adders," Proc. 14th Symp. Computer Arithmetic, pp. 30-34, Apr. 1999. Reprinted in ARITH-15, pp. 277-281.

6. A. Beaumont-Smith and C.C. Lim, "Parallel-Prefix Adder Design," Proc.15th Symp. Computer Arithmetic, pp. 218-225, June 2001.

7. V.G. Oklobdzija et al., "Energy-Delay Estimation Technique for High-Performance Microprocessor VLSI Adders," Proc. 16th Symp. Computer Arithmetic, pp. 15-22, June 2003.

8. H. Ling, "High-Speed Binary Adder," IBM J. R&D, vol. 25, pp. 156-166,May 1981.

9. Georgas Dimitrakopoulos and Dimities Nikolas' High-Speed Parallel-Prefix VLSI Ling Adders" IEEE transaction on computer, VOL. 54, NO. 2, FEBRUARY 2005.