Design of 64-Bit Low Power Parallel Prefix VLSI Adder for High Speed Arithmetic Circuits

R.Ramya[1],M.Prema[2],Y.Mohideen Faril Sumaiya[3]
PG Scholar[1,2,3],Bharathiyar Institute Of Engineering For Women,Tamil Nadu.
Assistant Professor,Mrs.P.Sowmiya, Bharathiyar Institute Of Engineering For Women,Tamil Nadu.
rvmramya@gmail.com,premabanu94@gmail.com,sumaiyame.cs@gmail.com,

***Abstract*-- The adding of two dual numbers is the essential and the majority often used mathematics process on microprocessors, digital signal processors and data dispensation submission specific included circuits. Parallel prefix adder is a universal technique for speed up binary calculation. This technique equipment logic functions which determine whether groups of bits will generate or propagate a carry. The future 64-bit adder is designed using four dissimilar types prefix cell operators, even-dot cells, odd-dot cells, even-partially-dot cells and odd-semi-dot cells; it offers robust adder solutions typically used for low power and high-performance propose request needs. The contrast can be made with a variety of input ranges of similar Prefix adders in terms control, number of transistor, number of nodes. Tanner EDA tool was used for simulate the parallel prefix adder designs in the 250nm technology.**

Keywords--Low Power VLSI; Parallel Prefix Adder; dot operators; 64-bit Parallel Prefix Adder;

I. INTRODUCTION

The parallel adder is the most significant component used in arithmetic operation of many processors. With the rising recognition of mobile devices, low power use and high performance included circuits has been the target of recent research. However, the two design criterion are often in conflict and that civilizing one particular aspect of the plan constrain the other. extend from the idea of carry look-ahead computation, a class of parallel carry look-ahead schemes are formed target at high-performance application. A Parallel Prefix Adder (PPA) is equivalent to the CLA adder [2]. The two different in the way their carry creation block is implemented. The parallel prefix carry-look ahead adder was first future some twenty years ago as a means of accelerate *n*-bit addition in VLSI technology. It widely considered as the fastest adder and used for high piece arithmetic circuits in the industries. A three step process is usually involved in the building of a Parallel Prefix Adder. The first step involves the creation of generate and spread signals for the input operand bits. The second step involve the generation of carry signals. In the final step, the sum bits of the adder are generate with the propagate signals of the operand bits and the preceding stage carry bit by a xor gate. The figure 1 shows the three step process is generally involved in the creation of a Parallel Prefix Adder.

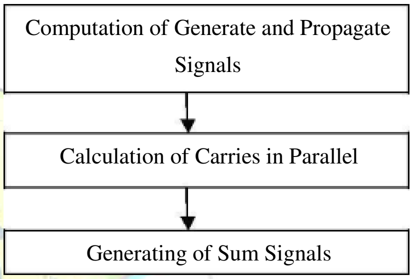

Computation of Generate and Propagate Signals
Calculation of Carries in Parallel
Generating of Sum Signals

Figure 1. Steps in parallel prefix adder

II. EXISTING PARALLEL PREFIX ADDERS

The agreement of the prefix network specify the type of the PPA. The Prefix agreement described by Haiku Zhu, Chung-Kuan Cheng and Ronald Graham, has the negligible depth for a known „n‟ bit adder. Best logarithmic adder structure with a fan-out of two for minimize the area-delay product is existing by Matthew Ziegler and Mircea Stan. The Sklansky adder presents a least depth prefix network at the cost of increased fan-out for certain calculation nodes [8]. The algorithm invented by Kogge-Stone has both best depth and low fan-out but produce massively complex circuit realization and also account for great number of interconnects [9]. Brent-Kung adder has the merit of minimal number of calculation nodes, which yields in compact area but structure has maximum depth which yield slight add to in latency when compared with other structure. The Han-Carlson adder combines Brent-Kung and Kogge-Stone structure to achieve a balance between logic depth and interlock count. Knowles obtainable a class of logarithmic adders with minimum depth by allowing the fan-out to grow [4]. Ladner and Fischer planned a general method to construct a prefix network with slightly higher depth when evaluate with Sklansky topology but achieved some merit by reducing the highest fan-out for computation nodes in the critical path [6]. Sparse tree binary adder proposed by Yan Sun and et al., combines the benefits of prefix adder and carry save adder [3][10]. Integer Linear Programming method to build parallel prefix adders is proposed by Jainhau Liu and et al., The future Parallel Prefix adder for 64-bits has been proposed.

III. PREFIX OPERATORS

In the above equation, „•‟ operator is functional on two pairs of bits and. These bits represent produce and propagate signals used in addition [7]. The output of the operator is a new pair of bits which is again joint using a dot operative „•‟ or semi-dot operator „•‟ with a new pairs of bits. This procedural use of dot operator „•‟ and semi-dot operator „•‟ creates a prefix tree system which ultimately ends in the age group of all carry signals. In the final step, the sum bits of the adder are generated with the propagate signal of the operand bits and the preceding stage carry bit using a xor gate. The semi-dot operator „ • ‟ will be here as last computation node in each editorial of the prefix graph structure, where it is essential to compute only generate term, whose value is the carry generated from that bit to the later bit.

$$(P_i,G_i) \quad (P_{i-1},G_{i-1}) = (P_i.P_{i-1},\ G_i+P_i.G_{i-1})$$

$$(P_i,G_i) \quad (P_{i-1},G_{i-1}) = (G_i+P_i.G_{i-1})$$

These two operators are divided into four types of diverse types prefix operators [5].

IV. PROPOSED 64-BIT PARALLEL PREFIX ADDER

A. 32-Bit Parallel Prefix Adder

Figure 2 shows the design of the future 32-bit parallel prefix adder. The idea is to reduce the huge overlay between the prefix sub-terms being calculated. Hence the associate property of the dot operator is working to retain the number of computation nodes at a smallest amount [1]. The first stage of the calculation is called as pre-giving out. The first stage in the architectures of the 32-bit prefix adder involve the arrangement of generate and spread signals for individual operand bits in active low arrangement. The second stage in the prefix addition is termed as prefix computation. This stage is accountable for creation of group generate and groups spread signals. The stages with odd index use odd-dot and odd-semi-dot cell where as the stage with even index use even-dot and even-semi-dot cells.

B. 64-Bit Parallel Prefix Adder

The Proposed 64-bit parallel prefix adder has ten stages of implementation. CMOS logic family will apply only inverting functions. Thus cascade odd cells and even cells alternatively gives the benefit of removal of two inverters between them, if a dot or a partially-dot computation node in an odd stage receives both of its input edges from any of the even stages and vice-versa. But it is essential to pioneer two inverters in a path, if a dot or a semi-dot totaling node in an even stage receive any of its limits from any of the even stage and vice-versa. From the prefix graph of the future structure shown in Figure 3 , we assume that there are only few ends with a pair of inverters, to make (G, P) as (G, P) or to make $(\overline{G}, \overline{P})$ as (G, P) respectively. The pair of inverters in a path is represented by a ◆ in the prefix graph. By introduce two cells for dot operative and two cells for semi-dot operator, we have eliminated a large number of inverters. Due to inverter removal in paths, the propagation delay in those paths would have reduced. Further we achieve a advantage in power reduction, since these inverters if not eliminated, would have contributed to important amount of power rakishness due to switching.

The output of the odd-semi-dot cells gives the value of the carry signal in that corresponding bit position. The output of the even-partially-dot cell gives the complement value of carry signal in that equivalent bit position.

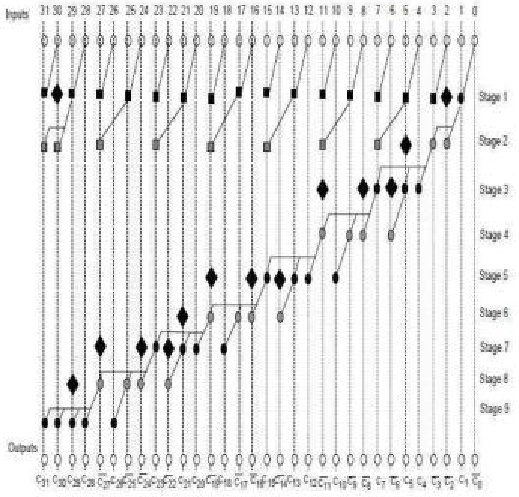


Figure 2. 32-Bit Parallel Prefix Adder

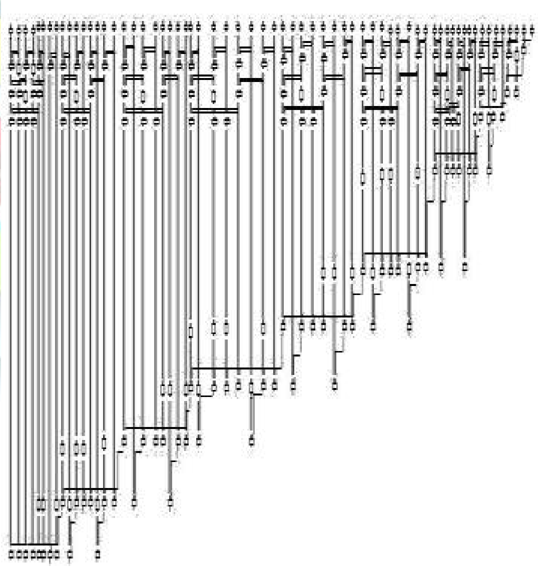


Figure 3. 64-Bit Proposed Parallel Prefix Adder

The final stage in the prefix adding is termed as post-processing. The final stage involves age band of sum bits from the active low spread signals of the individual operand bits and the carry bits generated in true form or balance form. The first stage and last stage are essentially fast because they involve only simple operation on signals local to each bit position. The intermediate stage embodies long distance circulation of carries, so the performance of the adder depends on the transitional stage.

V. SIMULATION RESULTS

C. *Proposed 64-Bit PPA*

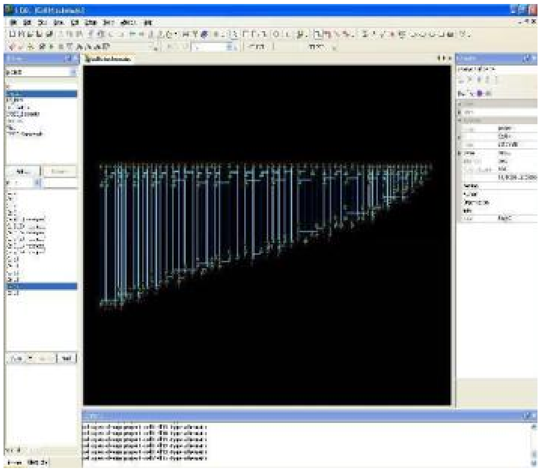


Figure 4. Block diagram for Proposed 64-Bit Parallel Prefix Adder

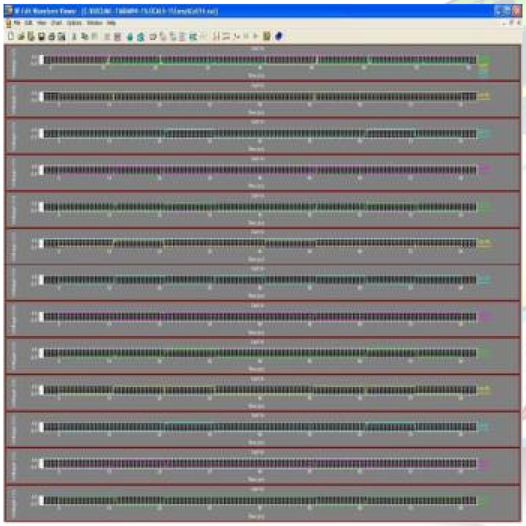
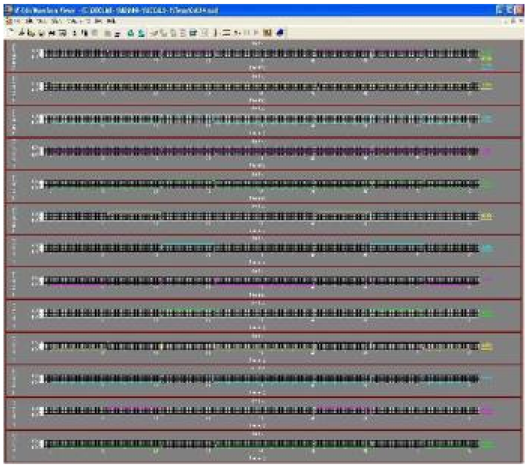


Figure 5. Output Waveform for Proposed 64-Bit Parallel Prefix Adder



D. *Comparison Charts*

1) Power Comparison

The power comparison shows that, when number of input bits increased the power consumed by the adder will also increasing.

Table 1. Power Comparision In Different Voltage Level

Adder	At V=5v (mW)	At V=4v (mW)	At V=3v (mW)	At V=2v (mW)
8-bit	4.45	2.51	.0090	0.0024
16-bit	11.6	5.21	1.91	0.579
32-bit	20.0	8.82	3.03	0.804
64-bit	44.1	21.2	7.84	2.30

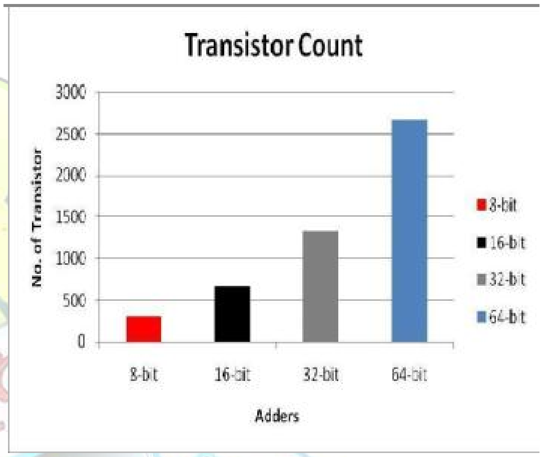


Figure 6. Power Comparison

2) Number of Nodes

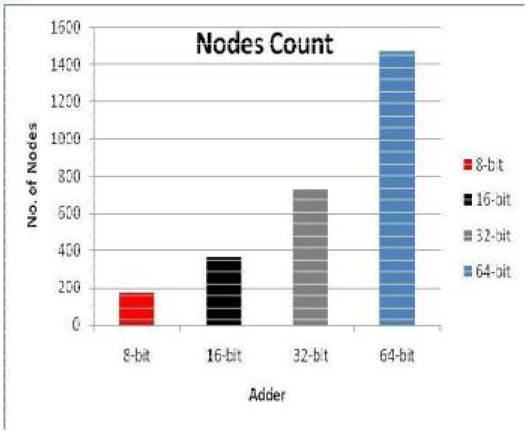


Figure 7. Number of Nodes

3) Number of Transistors

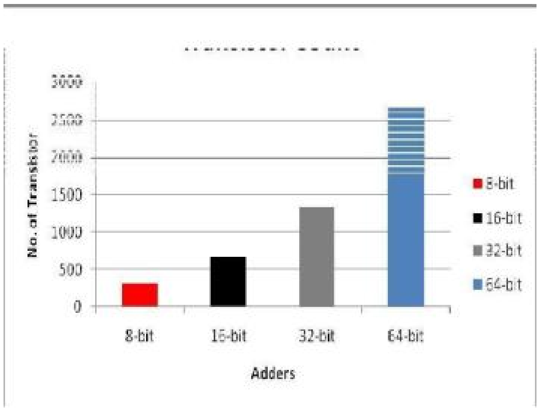


Figure 8. Number of Transistors

The above comparison graph shows the comparison of nodes and transistors in the 64-bit parallel prefix adder design.

VI. CONCLUSION

The parallel prefix formulation of binary calculation is a very suitable way to formally describe an entire relations of parallel binary adders. The proposed low power 64-bit parallel prefix adder were calculated by using the four unlike prefix cell operator and compared with existing parallel prefix adders such as 8-bit PPA, 16-bit PPA, and 32-bit PPA. The performances in terms of Power, Number of nodes, Number of Transistor tradeoffs for various input ranges were analyzed. The power comparison can be complete by varying the supply voltage from 1 to 5 for the whole parallel prefix adder considered. For 64-bit low power Parallel Prefix Adder Seventeen stages were used for generate the sixty three carries outputs and 1339 nodes were formed in the 64-bit low power Parallel Prefix Adder. The Proposed 64-bit low power prefix arrangement makes it more suitable for Arithmetic Logic Units and multiplier units for complex ranges of input data computation. Tanner EDA tool was used for simulate the parallel prefix adder designs in the 250nm technologies.

REFERENCES

[1] P.Ramanathan, P.T.Vanathi, “Novel Power Delay Optimized 32-bit Parallel Prefix Adder for High Speed Computing”, International Journal of Recent Trends in Engineering, Vol 2, No. 6, November 2009.

[2] R. Zimmermann, Binary Adder Architectures for Cell-Based VLSI and their Synthesis, ETH Dissertation 12480, Swiss Federal Institute of Technology, 1997.

[3] David Harris, “A Taxonomy of parallel prefix networks,” Proceedings of the 37th Asilomar Conference on Signals, Systems and Computers Pacific Grove, California, pp.2213-2217, November 2003.

[4] S.Knowles, “A family of adders”, Proceedings of the 15th IEEE Symposium on Computer Arithmetic. Vail, Colarado, pp.277-281, June 2001.

[5] P.Ramanathan, P.T.Vanathi, “A Novel Logarithmic Prefix Adder with Minimized Power Delay Product”, Journal of Scientific & Industrial Research, Vol. 69, January 2010, pp. 17-20.

[6] R. Ladner and M. Fischer, “Parallel prefix computation,” Journal of ACM. La Jolla, CA, vol.27, no.4, pp. 831-838, October 1980.

[7] Andrew Beaumont-Smith and Cheng-Chew Lim, “Parallel Prefix Adder Design”, Department of Electrical and Electronic Engineering, the University of Adelaide, 2001.

[8] J. Sklansky, “Conditional sum addition logic,” IRE Transactions on Electronic computers. New York, vol. EC- 9, pp. 226-231, June 1960.

[9] P.Kogge and H.Stone, “A parallel algorithm for the efficient solution of a general class of recurrence relations,” IEEE Transactions on Computers, vol. C-22, no.8, pp.786-793, August 1973.

[10] R.Brent and H.Kung, “A regular layout for parallel adders,” IEEE Transaction on Computers, vol. C-31, no.3, pp. 260-264, March 1982.

[11] T. Han and D. Carlson, “Fast area efficient VLSI adders,” Proceedings of the Eighth Symposium on Computer Arithmetic. Como, Italy, pp.49-56, September 1987.