AENSI Journals

Advances in Natural and Applied Sciences

ISSN:1995-0772 EISSN: 1998-1090

Journal home page: www.aensiweb.com/ANAS

Design of Multi Output Binary Adder using Modified Parallel Prefix Addition

[1]A. Azhagu Jaisudhan Pazhani, [2]C. Vasanthanayaki

[1]*Assistant Professor, Ramco Institute of Technology (Anna University), Information and Communication Engineering, Rajapalayam, Virudhunagar District, Tamilnadu, India*
[2]*Associate Professor, Government College of Technology (Anna University), Information and Communication Engineering, Coimbatore, Tamilnadu, India*

ARTICLE INFO

Article history:
Received 3 September 2014
Received in revised form 30 October 2014
Accepted 4 November 2014

Keywords:
Modified Prefix Tree, Carry generation logic, Modified flagged inversion cell, propagate signal, Generate Signal.

ABSTRACT

Flagged logic addition is a method for adding two numbers in a fast manner. Due to increasing hardware density low power, area efficient VLSI architecture is necessary. Parallel Prefix addition is a technique which is used in multi operand addition for improving the speed of carry generation. The conventional parallel prefix structures have large area and it consume more power, so the necessary modification is needed in the existing parallel prefix structures. A new technique called modified parallel prefix addition with increment and decrement operation using modified flagged logic has been proposed. The proposed adder perform increment and decrement operations by using flag bits which is generated within modified parallel prefix adder. In this paper, a new architecture for Parallel Prefix addition with modified flagged inversion cell is proposed for multi operand addition. Experimental results show that the proposed 16 bit flagged binary adder has 22.66% average area and 21.04% average power reduction when compared to existing flagged Brent kung, Ladner Fischer and Kogge stone design.

© 2014 AENSI Publisher All rights reserved.

To Cite This Article: A. Azhagu Jaisudhan Pazhani, C. Vasanthanayaki, Design of Multi Output Binary Adder using Modified Parallel Prefix Addition. **Adv. in Nat. Appl. Sci.,** *8(19): 10-16, 2014*

INTRODUCTION

The main functional unit of every processor is its data path. The major element of data-path and addressing units are arithmetic units, such as adders and multipliers. A basic arithmetic component is adder unit and it is used in more complex operations such as addition, division and multiplication. But also simple operations such as increment and decrement the resultant value by one unit. Adder play a key role in every arithmetic units and the efficient adder architecture is necessary in integrated circuit design. The propagation delays of the adder decade the overall arithmetic unit performance. Many adders architectures exist, but area, speed and low power are still challenging. Various different adder architectures for binary addition have been developed and covering a wide range of performance characteristics. Based on the characteristics of conventional parallel prefix structures and its performance are given in the following sections.

Low power and area efficient adder using flagged logic addition architecture is proposed. The performance of the proposed flagged binary adder shows the best overall performance in terms of area, power and delay, when compared with existing flagged prefix adder. The structure of the prefix network specifies the type of the parallel prefix tree. The Prefix network described by Haiku Zhu, Chung-Kuan Cheng and Ronald Graham (2005), has the small depth for the 'N' bit adder. Logarithmic adder structures with a minimum logic depth for reducing the area and power delay product is described by Matthew Ziegler and Mircea Stan (2001).

Minimum logic depth prefix tree with high fan-out for certain computation nodes are described in Sklansky adder (1960). Kogge-Stone adder (1973) has both logic depth and minimum fan out produces the complex prefix tree adder architecture with high layout complexity. Brent-Kung adder (1982) has minimum computation node which makes its area better but it has maximum logic depth which increase the propagation delay when compared with other prefix tree structures. The Han-Carlson adder (1987) has proper balance between the area and delay which is the combined architecture of both Brent-Kung and Kogge-Stone structures. A low power and area efficient Flagged binary adder has been proposed. The proposed flagged binary adder has the least power and area compared with flagged brent kung, flagged ladner fishcher and flagged kogge stone design. In the

Corresponding Author: A. Azhagu Jaisudhan Pazhani, Assistant Professor, Ramco Institute of Technology (Anna University), Information and Communication Engineering, Rajapalayam - 626117, Virudhunagar district, Tamilnadu, India
Tel: +91 9443696072 E-mail: alagujaisudhan@gmail.com

Advances in Natural and Applied Sciences, 8(19) Special 2014, Pages: 10-16

proposed adder group generate signals are used to generate the flag bits and it perform increment and decrement operations. Generated Flag bits are used to produces the corresponding resultant sum bits and its incremented, decremented results which are used in image processing applications.

Prefix Adders:

Parallel Prefix adder architecture is primarily fast when compared other conventional adder architectures. Parallel prefix tree structures have been developed to reduce the propagation delay in adder unit. Proper prefix tree structure improves the performance of integrated circuits and makes the efficient processor design. The conventional parallel prefix tree structures presented in the literature over the year which increases the logic depth, area, power and delay. Parallel Prefix tree adders are family of adders which is derived from the carry look ahead adders. The prefix tree adder is well suited for large input operands and tree network reduce the delay to O ($\log_2 N$) where N represents the input operand size (Alioto and Palumbo, 2006). The difference between a Carry look ahead Adder (CLA) and parallel prefix tree adder is presented in the second stage that make the carry signal fast and makes the binary addition very fast. The block diagram of conventional prefix adder is shown in Fig.1. A parallel Prefix Addition is consists of three step process, namely the pre-processing stage, carry generation and post-processing stage (J. Rebacz, E. Oruklu and J. Saniie, 2009). The pre-processing stage will produce the partial sum and partial carry signals as per equation (1).The partial sum and carry signals are combined to form the final carry signals. Final carry generation is performed in the second stage as per equation (2). The post processing stage produces the final as per equation (3). Multiple blocks are simultaneously executed in the pre processing stage that makes the addition operation very fast. Many parallel prefix adders are exits but Brent-kung, Ladner-Fisher, and Kogge Stone were widely used parallel prefix adders. The major difference between the full adder and parallel prefix adder is that in the full adder, sum and carry signal generation is done in single block but in the prefix adder tree, sum and carry generation are separated. Brent Kung Adder (BKA) has maximum logic depth which increases the latency. BKA occupy less area and it consumes less amount of power (Bostan and Ionescu, 2004). The propagation delay of Brent Kung Adder is equal to $2\log_2 N-2$. The area of Brent Kung Adder is $2N-2-\log_2 N$ where N is the size of input operand and it has large fan-out. Brent Kung parallel prefix tree structure has less computation nodes when compared with other prefix tree structures. The Ladner-Fischer parallel prefix tree adder is used to perform addition operation with minimum fan-out and latency (Koren, 2002). Ladner-Fischer adder is an extended version of Sklansky adder with minimum fan-out. The Kogge Stone Adder (KSA) has proper layout and it is suitable adder in the fabrication of integrated circuits and also it has minimum fan-out (Shubhajit Roy Chowdhury and Aritra Banerjee, 2008). KSA is the fastest adder when compared with other existing adder architecture but it occupies larger area (Koren, 2002). The KSA has the area of ($N \log_2 N$)-N+1 where *N* is the number of input bits (Brent and Kung, 1982).Table.I lists the area and logic depth of N bit existing parallel prefix tree structures.

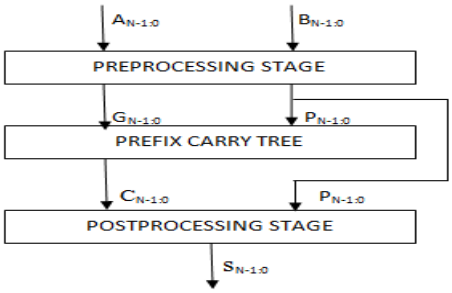

Fig. 1: Conventional prefix adder.

Propagate Signal $P = (A_N \wedge B_N)$ (1)
Generate Signal $G = (A_N B_N)$
Carry $C_{N+1} = (P_N C_N + G_N)$ (2)
Resultant Sum $S_{N-1} = P_N \wedge C_N$ (3)

Table I: Comparative Study on Conventional Prefix Adders.

Adder Type	No. Of Computation Nodes	Fan out
Brent Kung	2N-2-log N	2*log n – 2
Ladner Fischer	N/2 (log N)	log N +1
Kogge-Stone	(N log N) – N+1	log N

Advances in Natural and Applied Sciences, 8(19) Special 2014, Pages: 10-16

Modified Flagged Binary Adder:

The Modified flagged binary adder uses a simple technique for inverting a selected sum bits to find the resultant sum bits with its increment and decrement results such as (A+B+1) , (A+B-1), (A+B) and (A-B). The block diagram of modified flagged binary adder is shown in fig.2. Modified pre processing stage generates the partial sum, carry and added (R_N) signals from the input operand. Modified prefix tree stage produces the final carry from the partial sum, carry and added signal. Middle stage produces the additional output called the flag bits which is used to select the corresponding result sum using flagged inversion cell. This flagged binary addition technique has better performance than incrementer and decrementer in terms of area, power and delay. Additional hardware needed to introduce the flag bits is very less in the proposed flagged inversion cell when compared with existing flagged parallel prefix tree (Vibhuti Dave and Erdal Oruklu, 2010), since flag bits directly depend on the propagate signals.

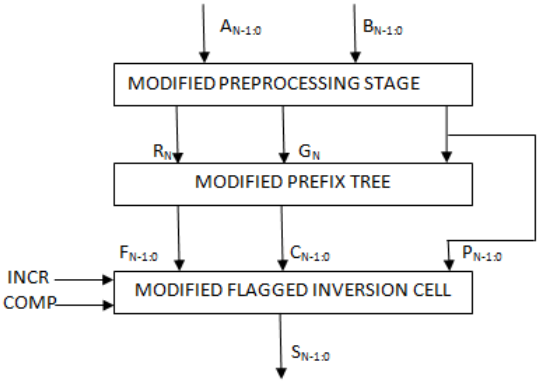

Fig. 2: Modified Flagged Binary Adder.

Modified Preprocessing stage outputs are,

$P_N = (A_N \wedge B_N)$ (4)

$R_N = (A_{N+1} \mid B_{N+1})$ (5)

$G_N = (A_N \& B_N)$ (6)

Modified Prefix Tree:

In general, the major drawback of most of the adders is the carry propagation delay and it is proportional to number of stages in the adder or word length of the input operands. Different carry generation logic has been developed to reduce the overall delay. In the modified prefix adder, efficient ways to speed up the addition is achieved by computing group carry for each pair of input bits simultaneously and it is shown in Fig.5. Group carry signal are calculated for pair of input operands and the resultant carry is identified based on the previous carry signal. Multiple blocks are simultaneously executed in the initial stage that reduces the latency of the proposed adder architecture. Additional hardware modifications are needed to produce the flag bit within the prefix tree architecture. Since the flag bits depend on the group propagate signal. (Vibhuti Dave and Erdal Oruklu, 2010), it is necessary to compute the required group propagate signal. Carry generation logic is incorporated within the prefix adder which divides the input bits into groups with 2 bits per group. Each pair of groups which result carry '1' are identified and carry generation logic is derived using minimization technique. Carry Generation logic diagram is as shown in Fig.3 which has only five logic gates that generate carry for every pair of input bits. This allows the addition of two or more numbers in a very short time compare to conventional prefix tree. Table.II lists the carry output for the pair of input bits. The Boolean functions for the carry outputs can be obtained directly from the Table.II. The equation (7) generates the carry for pair of input data.

Carry for pair of bits $C_{i+2} = A_{i+1} B_{i+1} + A_i B_i [A_{i+1} + B_{i+1}]$ (7)

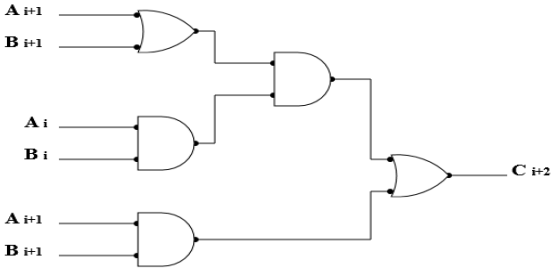

Fig. 3: Carry Generation Logic diagram for pair of bits.

Table II: Carry Generation for Pair of bits.

a_{i+1}	a_i	b_{i+1}	b_i	CARRY
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

Modified Flagged inversion cell:

The modified flagged inversion cell utilize the bit partial sum and carry signal outputs from the modified prefix tree inorder to produce a flag bit. The flag bits are further used to select the appropriate sum bits from the post-processing stage that need to be invert a carry signal to generate a completely new set of results (Vibhuti Dave and Erdal Oruklu, 2010). The new result could be the sum of the two input operands augmented or decremented by unity. The resulting flag bits and carry bits can be used according to the proposed flagged inversion logic as shown in Fig.4 to generate the results as per Table III.

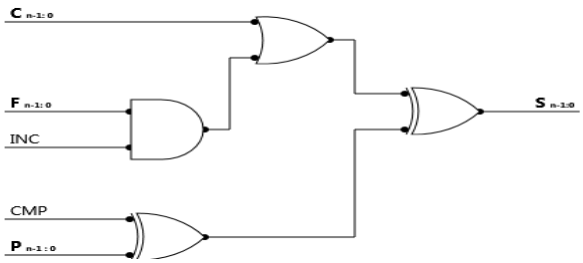

Fig. 4: Modified Flagged Inversion Cell.

Table III: Results of Modified Flagged Binary Adder.

COMP	INCR	RESULT(A+B)	RESULT(A-B)
0	0	A+B	A-B-1
0	1	A+B+1	A-B
1	0	-(A+B+1)	B-A
1	1	-(A+B+2)	B-A-1

.

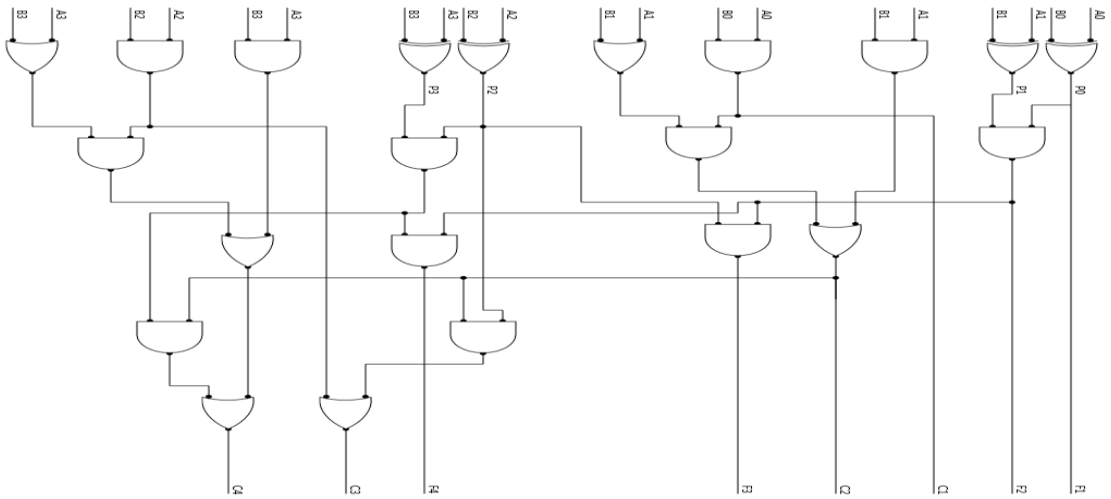

Fig.5: Modified prefix tree (4 bit adder tree).

Advances in Natural and Applied Sciences, 8(19) Special 2014, Pages: 10-16

Example: Addition of two 4 bit input operand using the proposed flagged adder:

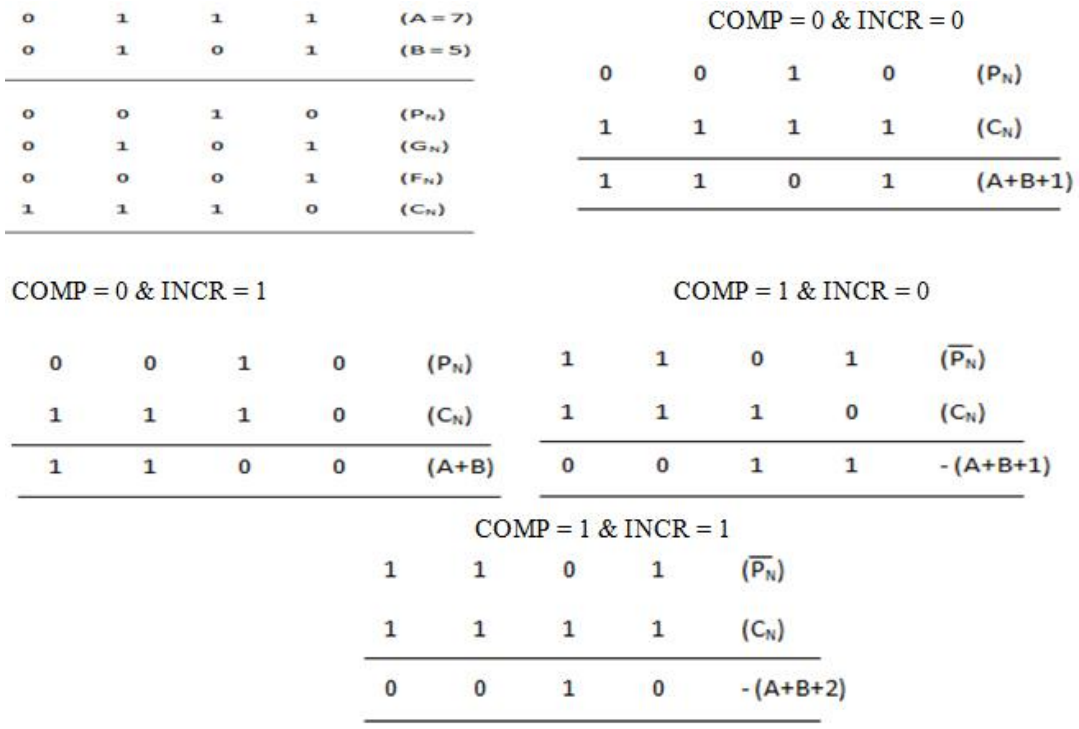

Table IV: Propagate Signal Output of Modified Flagged Inversion Cell.

COMP	PROPAGATE SIGNAL
0	$\overline{P_N}$
1	P_N

Table V: Carry Signal Output of Modified Flagged Inversion Cell.

FLAG BIT	INCR	CARRY
0	0	C_N
0	1	C_N
1	0	C_N
1	1	1

Performance Analysis:

The conventional parallel prefix tree adders and modified flagged prefix adder are designed and implemented in Cadence digital system design tool. Modified flagged prefix adder can able to produce different results according generated flag bits. Conventional and proposed adder was implemented for 8-bit and 16-bit input operand sizes. Theoretical analysis is performed on all adders with regards to gate count. Verilog HDL language is used to develop the proposed adder and it is synthesized in Cadence RTL compiler 9.1 using TSMC 180nm technology. The synthesized net list file and their respective constraints file are imported to Cadence Encounter tool and are used to generate automated layout from standard cells (Cadence tool 2009). The similar design flow is followed for all flagged prefix adder and proposed flagged binary adder. The result was analyzed in terms of area, delay and power with operating voltage is 1.8v and the temperature T=25 degree Celsius. Finally the parasitic capacitance, resistance are extracted and the layout of the proposed adder was generated and it is shown in fig.6. It is clear that the area of the 8 and 16 bit proposed adder is reduced by 14%, 14%, 28% and 8%, 8%, 52% respectively compared to Flagged brent kung, ladner fischer, and kogge stone prefix adder. The total power consumed by proposed 8 and 16 bit flagged binary adder is reduced by 7%, 8%, 22% and 13%, 15%, 28% when compared to brent kung, ladner fischer, and kogge stone prefix adder. Fig.6 shows the layout result of 16-bit modified flagged binary adder.

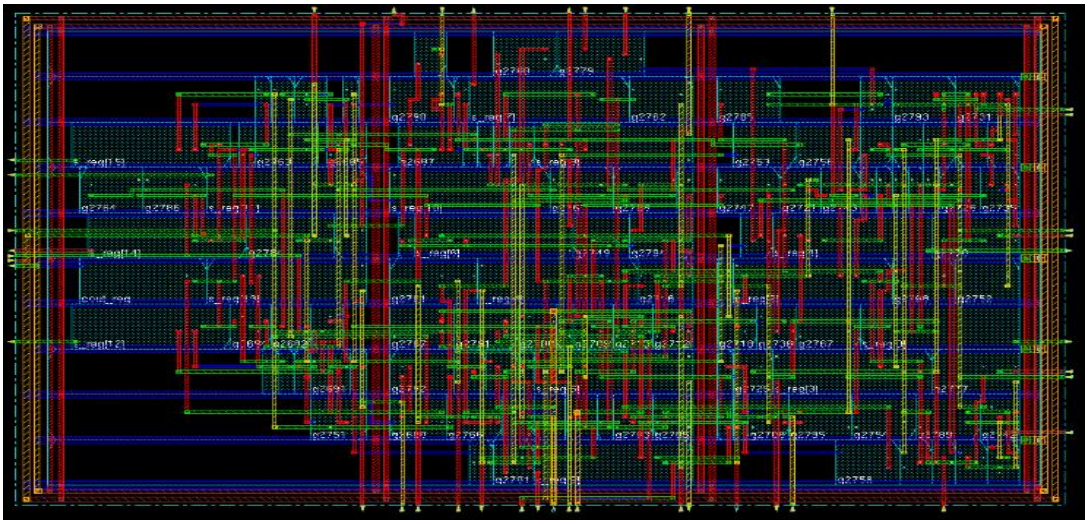

Fig. 6: ASIC layout of proposed 16 bit modified flagged binary adder.

Performance Comparison:

Performance of the proposed flagged binary adder is compared with the Flagged binary adder (Vibhuti Dave and Erdal Oruklu, 2010) in terms area, power and delay. Table.VII list the proposed adder has less area when compared with existing flagged binary adders (Vibhuti, Dave and Erdal Oruklu, 2010).

Table VI: Gate Count Calculation.

Adder Design	4 Bit	8 Bit	16 Bit
Brent Kung	21	49	123
Lander Fischer	21	52	129
Kogge Stone	33	79	179
Flagged Brent Kung	49	114	205
Flagged Ladner Fischer	49	118	208
Flagged Kogge Stone	58	141	256
Proposed Flagged Binary Adder	38	82	186

Table VII: Area Comparisons (mm^2).

Adder Type	4 Bit(mm^2)	8 Bit (mm^2)	16 Bit (mm^2)
Flagged Brent Kung	0.0360	0.1696	0.2803
Flagged Ladner Fischer	0.0365	0.1696	0.2821
Flagged Kogge Stone	0.0485	0.2022	0.5362
Proposed Flagged Binary Adder	0.0285	0.1464	0.2603

Table VIII: Power Comparisons (mw).

Adder Type	4 Bit (mw)	8 Bit (mw)	16 Bit (mw)
Flagged Brent Kung	2.34E-02	5.11E-04	7.06E-04
Flagged Ladner Fischer	2.38E-02	5.18E-04	7.27E-04
Flagged Kogge Stone Fischer	2.81E-02	6.11E-04	9.92E-04
Proposed Flagged Binary Adder	1.97E-02	4.80E-04	6.19E-04

*Total Power = Leakage Power + Dynamic Power

Table IX: Delay Comparison (ps).

Adder Type	4 Bit (ps)	8 Bit (ps)	16 Bit (ps)
Flagged Brent Kung	897	1080	1118
Flagged Ladner Fischer	911	998	1081
Flagged Kogge Stone Fischer	626	789	987
Proposed Flagged Binary Adder	715	914	1010

Comparison results shows that the proposed adder has less area and power efficiency compared to flagged Brent kung, Ladner Fischer and Kogge stone adder. Rise delay and fall delay of proposed flagged binary adder slightly increases for 16 bit adder compared to existing design.

Applications-Image Addition with Constant:

Constant addition with flagged prefix adders can be applied to wide variety of computer arithmetic implementations including floating point operations, decimal arithmetic and image processing. For example

adding a constant offset to all pixels in an image is a very common, fundamental operation to vary the brightness of images. With the proposed flagged constant addition hardware, addition of two images (for super imposed images) I_1 and I_2 with a constant C can be performed simultaneously in a single pass. It is defined as
$I_3(i,j) = I_1(i,j)+ I_2(i,j) + C$ (or)
$I_3(i,j) = I_1(i,j)+ I_2(i,j) - C$

Conclusion:

This paper introduces the idea of producing flag bits to select the corresponding resultant sum and its increment and decrement results, thereby eliminating the need of special adder for third operand addition. Prefix adders have been modified and incorporated with minimal hardware to compute a new set of bits called flag bits. The flag bits can be used to obtain different results. The proposed adder architecture can replace the carry-save adder and multi operand adder. One of the advantages is the availability of different useful results within one circuit. It eliminates the need to have dedicated adder units to perform the operation since the new logic is incorporated within a modified prefix tree adder. The proposed flagged binary adder has a less area and power when compared with existing flagged Brent kung, Ladner Fischer, and Kogge stone adders. The compared results show that the proposed flagged binary adder has 22.66% and 21.04% average area and power reduction for 16 bit adder compared to existing design. Though the proposed adder has less area complexity, it consumes less power. In future, enhanced flagged binary adder will be developed, which is suitable for floating point operations, decimal arithmetic and image processing applications.

REFERENCES

BID–blurred image database. http://www.lps.ufrj.br/profs/eduardo

Cadence, 2009. "Encounter Digital Implementation 9.1 user guide (RTL Compiler*)"*, Version 9.1.

Haikun Zhu, Chung-Kuan Cheng and Ronald Graham, 2005. "Constructing zero deficiency parallel prefix adder of minimum depth," Proceedings of 2005 Conference on Asia South Pacific Design Automation ASP-DAC, 2005. Shanghai, Jan, 2: 883–888.

Koren, I., 2002. "Computer Arithmetic Algorithms Ak Peters Series*"*, Second Edition, Massachusetts: A K Peters Ltd.

Brent, R. and H. Kung, 1982. "A regular layout for Parallel adders," IEEE Transaction on Computers, March, C-31(3): 260-264.

Rebacz, J., E. Oruklu, J. Saniie, 2009. August "Performance Evaluation of multioperand fast decimal adders", in proceedings of the 52nd IEEE international Midwest symposium on circuits and systems.

Alioto, M., G. Palumbo, 2006. "Impact of supply voltage variation on full adder delay", IEEE Transactions on very large scale integration Systems.

Ziegler, M.M., M.R. Stan, 2004. "A Unified Design Space for Regular Parallel Prefix Adders" IEEE Journal of Design, Automation and Test in Europe Conference and Exhibition, 2: 1386–1387.

Matthew Ziegler and Mircea Stan, 2001. "Optimal logarithmic adder structure with a fan-out of two for minimizing area delay product," IEEE International Symposium on Circuits and Systems 2001. Sydney, 2: 657-660.

Burgess, N., 2002. "The flagged Prefix Adder and its Applications in Integer Arithmetic" Journal of VLSI Signal Processing, 31(3): 263-271.

Kogge, P. and H. Stone, 1973. "A parallel algorithm for the efficient solution of a general class of recurrence relations," IEEE Transactions on Computers", 22(8): 786-793.

Ladner, R. and M. Fischer, 1980. "Parallel prefix computation," Journal of ACM. La Jolla, CA, 27(4): 831-838.

Sheikh, HR., Z. Wang, LK. Cormack, AC. Bovik, LIVE image quality Assessmentdatabase. Release 2 http://live.ece.utexas.edu/research/quality/subjective

Vibhuti Dave, Erdal Oruklu, Jafar Saniie, 2010. "Constant Addition With Flagged Binary Adder Architectures", Integration the VLSI journal, 43: 258-267.

Dave, V., E. Oruklu and J. Saniie, 2006. "Analysis, design and synthesis of flagged binary adders with constant addition,'' in: Proceedings of the 49th IEEE International Midwest Symposium on Circuits and Systems, 1: 23–27.

Ionescu, V., I. Bostan, L. Ionescu, 2004,"Systematic Design for Integrated Digital Circuit Structures" IEEE Journal of Semiconductor Conference, 2004, 2: 467–470.

Shubhajit Roy Chowdhury, Aritra Banerjee, Aniruddha Roy and Hiranmay Saha "Design of High Performance Low Power 16 Bit Arithmetic Units Using Kogge-Stone Parallel Prefix Adder Architectures" Proceedings of SPIT-IEEE Colloquium and International Conference, Mumbai, India Vol.2

Sklansky, J., 1960. "Conditional sum addition logic," IRE Transactions on Electronic computers. New York, 9: 226-231.