International Journal of ChemTech Research
CODEN (USA): IJCRGG, ISSN: 0974-4290, ISSN(Online):2455-9555
Vol.11 No.04, pp 121-126, 2018

An Efficient Accuracy Switchable Majority Based Prefix Adder

Sakthivel .B*, Padma .A

CSE, Muthayammal Engineering College, Rasipuram, Namakkal, India

Abstract : Parallel-prefix adders offer a highly efficient solution to the binary addition problem and are well suited for VLSI implementations. Approximate computing allows to improve latency, area, or power consumption for the sake of accuracy. This means that an error in computation may be tolerated as long as it is small enough to maintain a feasible operation of the system.In this work we propose a majority based prefix adder able switching between both exact and inexact mode .the structure more area and delay efficient than existing approximate adders by some modifications in carry calculation stage .proposed structure has been coded in HDL and implemented in vertex FPGA.

Introduction

Numerous logical and designing issues are processed using accurate and deterministic algorithms.However, in many applications involving signal/imageprocessing and multimedia, exact and precise calculations are not always necessary, because these applications areerrortolerant and produce results that are good enough forhuman observation [1]. In these blunder versatile applications, a decrease in circuit multifaceted nature, and hence, region, power and postponement is critical for the operation of a circuit. Thus, inexact processing can be utilized as a part of blunder tolerant applications by lessening exactness, yet at the same time giving significant outcomes speedier or potentially with bring down power utilization.

Inexact processing permits to enhance inertness, region, or power utilization for precision. This implies a blunder in calculation might be endured as long as it is sufficiently little to keep up an attainable operation of the framework. Quite, adders are the foundation of numerous number juggling units. Subsequently, a few deterministic surmised viper procedures, for example, [1]– [3], have been proposed amid the most recent couple of years. Dissimilar to probabilistic adders [4], deterministic approximations by and large take after the approach of part the convey chain in points of enhancing inertness and power utilization [5]. Such part strategies permit playing out the expansion in parallel segments of the contribution with abbreviated convey proliferation. This parallelism can be converted into inertness upgrades, and therefore, diminished vitality utilization

Since the power consumption and speed are critical parameters in the design of digital circuits, the optimizations of these parameters for multipliers become critically important. Veryoften, the optimization of

Sakthivel .B *et al* /International Journal of ChemTech Research, 2018,11(04): 121-126.

DOI : http://dx.doi.org/10.20902/IJCTR.2018.110414

one parameter is performed considering a constraint for the other parameter. Specifically, attaining the desired performance (speed) considering the limited power budget of portable systems is challenging task. In addition, having a given level of reliability may be another obstacle in reaching the system target performance.

Recently, many approximate architectures for arithmetic units which are based on the logic and circuits simplification at different design levels have been proposed.

Designs of approximate circuits have already been proposed using classical logic gates. There have been approximate full adders proposed such as the approximate mirror adder(AMA) designs proposed in [20] and the approximate XOR based adder (AXA1) and approximate XNOR-based adders(AXA2, AXA3) proposed in [21]. Some approximate subtractor(AXSC) designs were proposed in [22].Their design uses a correct adder-subtractor for the most significant bits and uses an approximate adder-subtractor design for the least significant bits. Those designs are not included inour comparison sections due to them being larger than our proposed designs. The smaller number of existing 1-bit designs some what limits the number of existing designs with which we can compare our proposed designs

In [1], author proposes a scheme to generate carry bits with block-carry-in 1 from the carries of a block with block-carry-in 0. This method is then applied to carry-select and parallel-prefix adders to derive a more area-efficient implementation for both the case. In[2],author presented a two novel architectures for designing modulo 2/sup n/+1 adders, based on parallel-prefix carry computation units, the first architecture uses a fast carry increment stage, whereas the second is a fully parallel-prefix solution.I n [3], author introduced a 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.

In [4], author exploited a algebraic properties in the derivation of the proposed fast adder architecture .In [5],author proposed algorithms depart from the traditional approach of modulo 2^n-1 addition by setting the input carry in the first stage of the addition to one, which only ever produces one representation of zero. The architectures not only offer significant speedup in a modulo 2^n-1 addition, but they can also offer a reduction in area and thus provide improvements in the cost functions area times delay^2 and energy times delay .In [6],author introduce a framework that generates formal prefix equations for speculative carry generation .In [7],author presents a novel technique to realize D3L parallel prefix tree adders without significantly compromising speed performance.

In [8],author describes a new advancement in theorem proving based formal verification: a formalization of a parameterized parallel prefix adder developed in the proof assistant Coq.In [9],author presents an efficient quantum-dot cellular automata (QCA) design for the Ladner-Fischer prefix adder then present an efficient QCA design of a hybrid adder that combines the Ladner-Fischer adder with a ripple carry adder. In [10],author proposed a sparse approach enabled by the introduction of the inverted circular idempotency property of the parallel-prefix carry operator and its regularity and area efficiency are further enhanced by the introduction of a new prefix operator. In [11],author present a number of new results on majority logic. Then present a efficient QCA designs for the ripple carry adder (RCA) and various prefix adders.

In [12],author discuss the architecture, design, and testing of the first 16-bit asynchronous wave-pipelined sparse-tree superconductor rapid single flux quantum adder. In [13],author present a new comparator design featuring wide-range and high-speed operation using only conventional digital CMOS cells and prefix logic.In [14],author, parallel prefix operation and carry correction techniques are adopted to eliminate the re-computation of carries. In [15],author, proposes an efficient algorithm to synthesize prefix graph structures that yield adders with the best performance-area trade-off, their approach generates prefix graph structures to optimize an objective function such as size of prefix graph subject to constraints like bit-wise output logic level.

In [16],author present a carry skip adder (CSKA) structure that has a higher speed yet lower energy consumption compared with the conventional one. The speed enhancement is achieved by applying concatenation and incrementation schemes to improve the efficiency of the conventional CSKA (Conv-CSKA) structure.. In [17],author proposed a novel error-tolerant adder (ETA). The ETA is able to ease the strict restriction on accuracy, and at the same time achieve tremendous improvements in both the power consumption and speed performance.

In [18],author proposes a highly defect tolerant Parallel Prefix Adder (PPA) presented design identifies the key elements that can be applied to make the general PPA's defect tolerant: 1) the Generate and Propagate computing hardware is divided into disjoint groups, such that defects in one group will not "contaminate" the computation carried out by the other groups; 2) redundant copies of the results for each group can be derived cost-effectively from the other disjoint groups.

Experimental

Adder and Output Carry

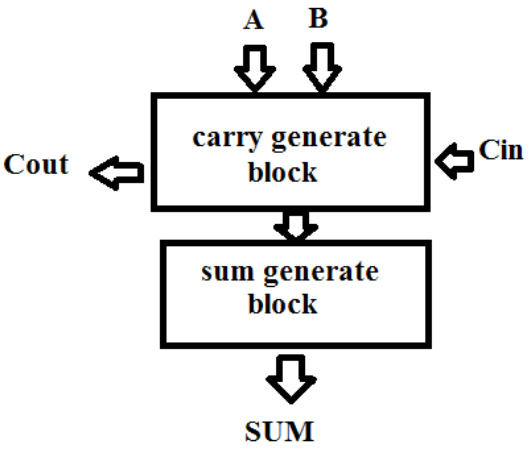

Fig .1 prefix adder

Fig.1 displays the block diagram of an$n$-bit binary adder. The inputs to the adder are given by A = $a_{n-1}a_{n-2}\cdots a_1a_0$, =$b_{n-1}b_{n-2}\cdots b_1b_0$and $C_i$while SUM=$S_{n-1}S_{n-2}\cdots S_1S_0$andthe carry $C_n$are the outputs.

Many methodologies have been proposed for calculating S and C_n effectively. These incorporate the *carry look ahead adder* (CLA) and prefix adders. While CLA provides a way to quick two-operand expansion, fan-out restrictions have prompted the advancement of prefix adders to lessen the convey count to a "prefix" computations. Both CLA and the prefix adders are working on the principle of*generate* and *propagate* of two binary signals $ai$and bi: in particular, $gi = aibi$while $pi = (ai + bi)$. Prefix adders employ an associative operator [2] denoted by ∘and defined as

$(G_i, P_i) \circ (G_j, P_j) = (G_i + (P_iG_j), P_iP_j)$ (1)

The ∘operator is used to express the output carry $C_n$as

$(Cn, 0) = (g_{n-1}, p_{n-1}) \circ (g_{n-2}, p_{n-2}) \circ \ldots \circ (g_1, p_1) \circ (g_0, p_0) \circ (C_0, 0)$ (2)

The majority gate equations of the output carry $C_{i+1}$and the output sum $S_i$are known in (3) and (4); therefore, C_{i+1} influenced by previous carry signals (such as C_i), whereas S_i depends only on the carry signals and the inputs. After generating all the carry signals, the sum signals are calculated in parallel. Hence, the speed of an$n$-bit binary adder primarily depends on the carry generation process.

$C_{i+1} = M(a_i,b_i,c_i)$ (3)
$S_i = M(\overline{C_{i+1}}, M(a_i,b_i,\overline{C_{i+1}}), c_i)$ (4)
The output carry $Cn$of an n-bit adder is usually specified interms of Sas
$C_n = (\text{sum} \geq 2^n) + (\text{sum} = 2^n - 1)\, C_{in}$ (5)

where$S \geq 2^n$and $S = 2^n - 1$ are binary signals. generating the ($S \geq 2^n$) and ($S = 2^n - 1$) signals, calculations of Cnrequires just two gate delays. The definition of the output carry *directly* in terms of majority gates is advantageous to achieve a low delay/circuit complication for majority gate based adders. (5) can be redefined such that the output carry isin terms of majority gates; this is given by

$C_n = M(\text{sum} \geq 2^n, \text{sum} \geq (2^n - 1), C_{in})$ (6)
$\text{sum} \geq 2^n = C_n|_{(Cin=0)}$(7)
$\text{sum} \geq (2^n - 1) = C_n|_{(Cin=1)}$(8)

To calculate the output carry in (6), one majority gate is required after computing the Ssignals. The same output carry in (5)requires two majority gates (therefore incurring in a two majority gate delay).Ssignalis expressed in recursive majority logic form. This is used to efficiently produce the carry as in (6).

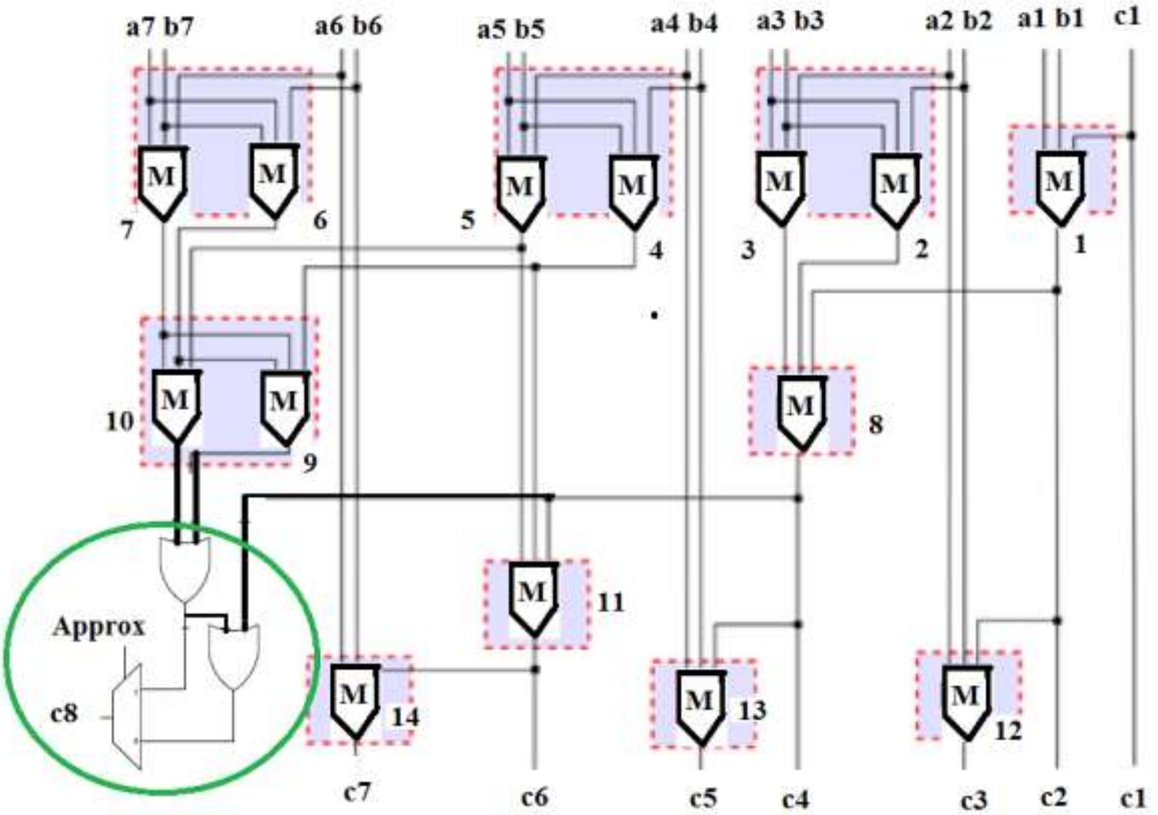

Fig.2 Proposed switchable prefix adder by adding MUX

Let $A = a_{n-1}a_{n-2} \cdots a_1a_0$, $B = b_{n-1}b_{n-2} \cdots b_1b_0$and the (initial) carry$C0$ be inputs to an$n$-bit binary adder. Then, the output carry Cncan be defined in terms of majority gates as$C_n = (a_{n-1}, b_{n-1}, (a_{n-2}, b_{n-2}, \ldots ,(a1, b1, (a0, b0, C0)))$.

In this work we propose to split final carry computation of majority prefix adder into two segments(7).the first segment contains *addition of majority value m8 and m9.*while the other segment m8 .where the left part is called *approximate part* and the right partis called *augmenting part*. If both of these parts are employed to calculate the carry output, the generated *C8* is exact whereas ifonly the approximate part is used, the generated *C8*imprecise.

C8=(m8+m9)+m10; (7)

Fig 2 shows proposed switchable prefix adder by adding MUX in final carry computation stage .M represents majority gates C represents final carry .Based on mode selection the adder will be worked as approximate or exact mode.

Results

The proposed switchable architectures are implemented on a Xilinx Spartan 3e FPGA. Table1 illustrate the implementation results. This leads to the reduction in the period small bit increase in area.

Table 1 performance analysis

S.no	Parameter	Existing	Proposed
1	Slice	8	10
2	LUT	15	18
3	Delay	10.995	10.905

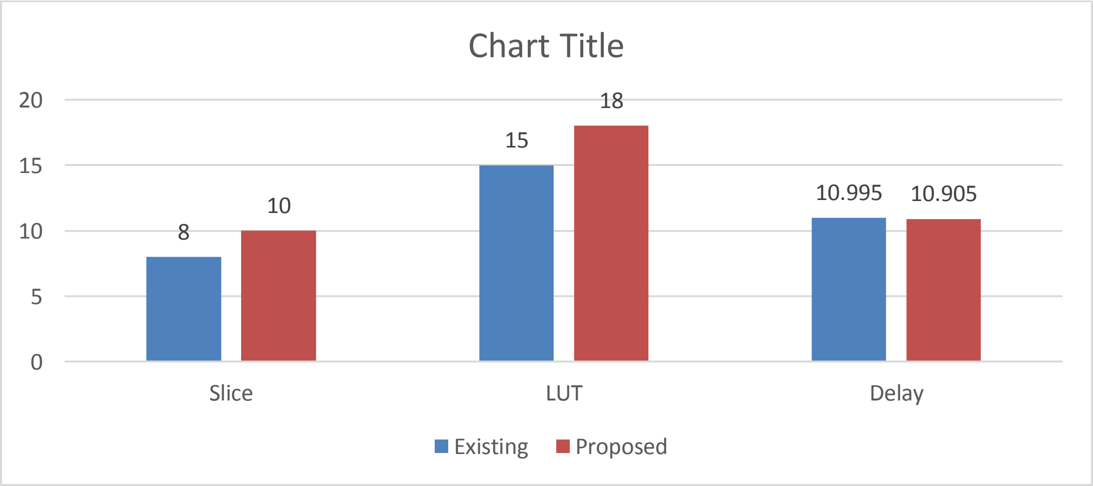

Fig .3 Performance graph

Discussion

In this paper, a low power yet energy-efficient Majority based prefix adder was suggested. The adder has ability of switching between the approximate and exact operating modes making it suitable for both error-tolerant and exact applications. The structure of the proposed adder was based on some modifications to the structure of the majority prefix. To assess the efficacy of the proposed structure, its design parameters were compared to those of some suggested reconfigurable approximate adders.

References

1. Tyagi A, A reduced-area scheme for carry-select adders, IEEE Transactions on Computers, Oct 1993, Volume: 42, Issue: 10.
2. Efstathiou C,Vergos H.T,Nikolos D, Fast parallel-prefix modulo 2/sup n/+1 adders, IEEE Transactions on Computers, Sept. 2004, Volume: 53, Issue: 9.
3. Dimitrakopoulos G and Nikolos D, High-speed parallel-prefix VLSI Ling adders, IEEE Transactions on Computers, Feb. 2005, Volume: 54, Issue: 2.
4. Patel R.A, Benaissa M, BoussaktaS,Fast Modulo 2^{n} - (2^{n - 2} + 1) Addition: A New Class of Adder for RNS, IEEE Transactions on Computers, April 2007, Volume: 56, Issue: 4.
5. Riyaz A. Patel, Mohammed Benaissa, Said Boussakta, Fast Parallel-Prefix Architectures for Modulo 2n-1 Addition with a Single Representation of Zero, IEEE Transactions on Computers, Nov. 2007,Volume: 56, Issue: 11.
6. Youngmoon Choi, Earl E. Swartzlander, Speculative Carry Generation With Prefix Adder, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, March 2008, Volume: 16, Issue: 3.
7. Fabio Frustaci, Designing High-Speed Adders in Power-Constrained Environments, IEEE Transactions on Circuits and Systems II: Express Briefs, Feb. 2009, Volume: 56, Issue: 2.

8. Gang Chen , Formalization of a Parameterized Parallel Adder Within the Coq Theorem Prover, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Jan. 2010, Volume: 29, Issue: 1.
9. Vikramkumar Pudi, K. Sridharan, Efficient Design of a Hybrid Adder in Quantum-Dot Cellular Automata, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Sept. 2011, Volume: 19, Issue: 9.
10. Haridimos T. Vergos, Giorgos Dimitrakopoulos, On Modulo 2^n+1 Adder Design,IEEE Transactions on Computers, Feb. 2012, Volume: 61, Issue: 2.
11. Vikramkumar Pudi, Sridharan K,Low Complexity Design of Ripple Carry and Brent–Kung Adders in QCA, IEEE Transactions on Nanotechnology, Jan. 2012, Volume:11,Issue:1.
12. M. Dorojevets, Ayala C.L ,16-Bit Wave-Pipelined Sparse-Tree RSFQ Adder, IEEE Transactions on Applied Superconductivity, June 2013, Volume: 23, Issue: 3.
13. Saleh Abdel-Hafeez, Ann Gordon-Ross, Behrooz Parhami,Scalable Digital CMOS Comparator Using a Parallel Prefix Tree, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Nov. 2013,Volume: 21, Issue: 11.
14. A Novel Modulo 2n−2k−1 Adder for Residue Number System
15. Subhendu Roy, MihirChoudhury, RuchirPuri, David Z. Pan, Towards Optimal Performance-Area Trade-Off in Adders by Synthesis of Parallel Prefix Structures, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Oct. 2014, Volume: 33, Issue: 10.
16. Milad Bahadori, Mehdi Kamal, Ali Afzali-Kusha,High-Speed and Energy-Efficient Carry Skip Adder Operating Under a Wide Range of Supply Voltage Levels, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Feb. 2016, Volume: 24, Issue:2
17. Ning Zhu, Wang Ling Goh ,Design of Low-Power High-Speed Truncation-Error-Tolerant Adder and Its Application in Digital Signal Processing, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Aug. 2010, Volume: 18, Issue: 8.
18. Soumya Banerjee, Wenjing Rao,A general approach for highly defect tolerant parallel prefix adder design, Design, Automation & Test in Europe Conference & Exhibition (DATE), 2016.
19. Kogge P and Stone H, A parallel algorithm for the efficient solutionof a general class of recurrent equations, IEEE Transactions on Computers, Aug. 1973, Vols. C-22, pp. 786-793
20. Gupta V, Mohapatra D, Park S.P,Raghunathan A, and Roy K, Impact: imprecise adders for low-power approximate computing, In Proceedings of the 17th IEEE/ACM international symposium on Lowpowerelectronics and design. IEEE Press, 2011, pp. 409–414.
21. Yang Z, Jain A, Liang J, Han J, and Lombardi F, Approximatexo r/xnor-based adders for inexact computing, in Nanotechnology(IEEE-NANO), 2013 13th IEEE Conference on. IEEE, 2013, pp. 690–693.
22. Chen L, Han J, Liu W, and Lombardi F, Design of approximate unsigned integer non-restoring divider for inexact computing, in Proceedings of the 25th edition on Great Lakes Symposium on VLSI. ACM,2015, pp. 51–56.

*****