High-performance approximate half and full adder cells using NAND logic gate

Hamaro Waris¹, Chenghua Wang¹, and Weiqiang Liu[1a)

Abstract In this letter, NAND-based approximate half adder (NHAx) and full adder (NFAx) cells are proposed for low power approximate adders. NHAx and NFAx architectures are built using NAND logic gate which has a minimal normalized gate delay among all the CMOS-based integrated circuit digital logic family; therefore, an improvement of 29% in the critical path delay is achieved. For the performance evaluation, 8-bit ripple carry adder (RCA) is then built using proposed cells. RCA-NFAx shows a good power-efficiency trade-off when both power delay product and error metric NMED are considered with reference to the previous approximate 1-bit FA based RCA designs.

Keywords: CMOS, NAND, RCA, NMED, PDP

1. Introduction

The strict power constraints and high performance require new design strategies to cope up with computing efficiency [1, 2]. One such emerging and promising design methodology is named as approximate or inexact computing [3]. A lot of applications such as computer vision, data mining, image/video processing and deep learning neural networks, etc., reveal inherent error resilience [4, 5]. Therefore, approximate computing paradigm produce less accurate but acceptable quality results by relaxing the stringent error requirement [6, 7].

Arithmetic units are the most power hungry component and are an integral part all of the computing hardware designs [8, 9, 10]. Therefore, approximation in hardware focus to reduce power consumption of these arithmetic units [11]. Adder is the basic building block used in these arithmetic units. In adders, the addition process is governed by the carry generation and propagation mechanism [12, 13, 14]. Some well-known adders designed using approximation techniques are named as speculative [15, 16], segmented [17, 18] and carry select adders [19, 20]. However, this design approach is not suitable for energy-efficient circuits as this process requires multiple overlapping sub-adders [21]. Generally, for the low power approximate adder designs, the most appropriate design approach is to replace exact FAs with the approximate FAs [22, 23].

In the technical literature, approximate FAs using different logic styles have been proposed, e.g., Approximate Mirror Adders (AMAs) [24] use standard static CMOS techniques; pass transistor logic is used in XOR/XNOR based FAs [25]; TGAx based FAs [26] use transmission gate. All approximate FAs as compared to the exact design provide less area and power consumption at the cost of relaxing accuracy. However, most of the designs have a degraded performance (high delay) due to the internal structure and node capacitance. For example in AXA3 [25] delay is higher, due to the threshold voltage drop of the pass transistors. Similarly in [24], AMA5 cell lacks the output drive strength; therefore, extra buffers must be added to design it. On contrary to that, NHAx and NFAx cells show a good speed-accuracy trade-off, achieving high-performance (lower delay) while maintaining error distance (ED) as 1. Normalized CMOS logic gate delays used in this work are as follows; NAND (0.5), OR/AND (0.7), XOR/XNOR (1.0) [27].

2. NHAx cell

HA takes two 1-bit inputs and outputs corresponding Sum (S) and Carry (C). The Boolean expressions for NHAx are given by Eq. (1) and Eq. (2), respectively. Fig. 1 shows S computation is carried out using NAND gate resulting in one incorrect result, while C has accurate output.

\[ NHAx_{sum} = \overline{ab} \]  
\[ NHAx_{carry} = \overline{ab} \]  

Fig. 1. Logic diagram of NHAX cell.

The path delay for S is reduced to 0.5 as compared to 0.7 unit delay of previous best approximate so-called approximate half adder (AHA) [28], i.e., an improvement of 28% in delay.

3. NFAx cell

FA takes three 1-bit inputs and outputs corresponding S and C. NFAx expressions for (S,C) are given by Eq. (3) and Eq. (4), respectively. Fig. 2 shows the gate level design of NFAx.
\[ NFAx_{sum} = (\overline{a}b)c \quad (3) \]
\[ NFAx_{carry} = (\overline{a}b)\overline{c} \quad (4) \]

NFAX handle approximation in such a way that error difference between exact output and approximate output is always kept as one. Since C has more binary weight with respect to S, error in C output will adversely affect the result. Therefore, approximation in C is considered only for the cases where S is also approximated. This results in one error in the C computation and in total three inaccurate results out of eight as shown in Table I.

**4. Performance evaluation**

For the fair analysis, both transistor and logic level approximate 1-bit FA cells are taken into account for comparison. AFA [28] is selected among the approximate logic level designs as to the author’s best of knowledge this is the most recent proposed logic level 1-bit approximate design. As discussed earlier, many transistor level designs based on different logic styles are found in technical literature; therefore, the best designs from each logic style and corresponding gate level realizations are considered for reference.

**4.1 Critical path delay**

Table II shows the analysis of critical path delay reduction achieved by the NFAX cell. By using normalized gate delay, for NFAX, the critical path delay is 1.0 as compared to 1.4 unit delay for AFA (logic level design) and AMA4 (transistor level best design), respectively, i.e., an improvement of 29% in the critical path delay.

**4.2 Hardware resource analysis**

The proposed NFAX cell along with other approximate 1-bit FA cells are realized at gate level using Verilog hardware description language (HDL). These gate level realizations are used to build an approximate RCAs as shown in Table III. 8-bit RCAs are designed by replacing least significant part of RCAs with the approximate FA cells respectively. Synopsys Design Compiler is used to synthesize the said designs at 45 nm technology node. All the basic logic gates are mapped using standard CMOS library and supply voltage is kept at 1.25 V. Optimization primitive is kept off and pure combinational logic is analyzed for the calculation of delay, area, power and PDP. NMED [30] is used as a metric for error analysis, 8-bit RCAs are simulated in MATLAB against all input combinations.

| Inputs | Exact-FA | NFAX | ED |
|--------|----------|------|----|
| a      | b        | c    | C  |
| 0      | 0        | 0    | 0  |
| 0      | 1        | 0    | 1  |
| 0      | 0        | 1    | 1  |
| 0      | 1        | 1    | 0  |
| 1      | 0        | 0    | 0  |
| 1      | 1        | 1    | 1  |
| 1      | 0        | 1    | 0  |
| 1      | 1        | 1    | 1  |
| 1      | 1        | 1    | 1  |

| Designs   | Area (um²) | Delay (ns) | Power (uW) | PDP (fJ) | NMED (10⁻¹³) |
|-----------|------------|------------|------------|----------|---------------|
| RCA-NFAX  | 29.16      | 0.33       | 5.98       | 1.97     | 0.12          |
| RCA-AFA   | 36.06      | 0.37       | 7.81       | 2.88     | 0.10          |
| RCA-AMA4  | 33.57      | 0.37       | 6.63       | 2.45     | 0.12          |
| RCA-AXA1  | 42.86      | 0.41       | 10.23      | 4.19     | 0.13          |
| RCA-TGA2  | 36.06      | 0.39       | 7.81       | 3.04     | 0.11          |
| RCA-InAX2 | 41.18      | 0.43       | 9.13       | 3.92     | 0.09          |

Table III shows RCA-NFAX achieved high performance (reduced delay). In addition to that, RCA-NFAX has the lowest PDP as compared to all other approximate FA designs. RCA-InAX2 has the smallest NMED among all the designs as they generate error only in the S part.

**4.3 NMED vs PDP analysis**

To comprehend the accuracy-power trade-off, all the designs are further compared with reference to PDP and NMED as shown in Fig. 3. RCA-AFA and RCA-TGA2 have a nominal NMED at the cost of high power consumption. RCA-AXA1 and RCA-InAX2 have a large PDP as compared to all other designs. On the other hand, RCA-AMA4 and RCA-NFAX have similar NMED but smaller PDP of RCA-NFAX makes it a favorable choice for power-efficient designs.
5. Conclusion

This letter presents NAND-based approximate half adder (NHAx) and full adder (NFAx) cells for area constrained designs. The proposed cells are designed using NAND logic gate, as a result reducing the critical path delay for the 1-bit FA cell. An improvement of 29% in the critical path delay is achieved with reference to previous best 1-bit approximate FA cell. The comparative performance analysis is done by designing 8-bit RCA using proposed NHAx and NFAx cells. The result shows RCA-NFAx has a lower PDP, i.e., an improvement of 32% and 20% in PDP, as compared to RCA-AFA (logic level) and RCA-AMA4 (transistor level) FA cells, respectively. RCA-NFAx outperforms all other designs in terms of PDP and proves to be the most efficient building block for low-power high-performance approximate adders.

Acknowledgments

This work is supported by grants from the National Natural Science Foundation of China (61871216), the Six Talent Peaks Project in Jiangsu Province (2018-XYDXX-009) and the Fundamental Research Funds for the Central Universities China (No. NE2019102).

References

[1] J. Han and M. Orshansky: “Approximate computing: An emerging paradigm for energy-efficient design,” IEEE ETS (2013) 1 (DOI: 10.1109/ETS.2013.6569370).
[2] R. Nair: “Big data needs approximate computing: Technical perspective,” Commun. ACM 58 (2014) 104 (DOI: 10.1145/2688072).
[3] S. Venkataramani, et al.: “Computing approximately, and efficiently,” DATE (2015) 748 (DOI: 10.1109/DATE.2015.1113).
[4] V. K. Chippa, et al.: “Analysis and characterization of inherent application resilience for approximate computing,” DAC (2013) 113 (DOI: 10.1145/2463209.2488873).
[5] J. Schlachter, et al.: “Design and applications of approximate circuits by gate-level pruning,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst. 25 (2017) 1694 (DOI: 10.1109/TVLSI.2017.2657799).
[6] Q. Xia, et al.: “Approximate computing: A survey,” IEEE Des. Test. 33 (2016) 8 (DOI: 10.1109/MDAT.2015.2505723).
[7] S. Venkataramani, et al.: “Approximate computing,” IEEE VLSID (2016) 3 (DOI: 10.1109/VLSID.2016.128).
[8] A. Sampson, et al.: “Approximate storage in solid-state memories,” IEEE Micro (2013) 1.
[9] W. J. Dally, et al.: “Efficient embedded computing,” Computer 41 (2008) 27 (DOI: 10.1109/MC.2008.224).
[10] S. Mazahir, et al.: “Adaptive approximate computing in arithmetic datapaths,” IEEE Des. Test 35 (2018) 65 (DOI: 10.1109/MDAT.2017.2772874).
[11] R. Venkatesan, et al.: “MACACO: Modeling and analysis of circuits for approximate computing,” ICCAD (2011) 667 (DOI: 10.1109/ICCAD.2011.6105401).
[12] A. B. Kahng and S. Kang: “Accuracy-configurable adder for approximate arithmetic designs,” DAC (2012) 820 (DOI: 10.1145/2228360.2228509).
[13] K. Du, et al.: “High performance reliable variable latency carry select addition,” DATE (2012) 1257 (DOI: 10.1109/DATE.2012.6176685).
[14] Y. Kim, et al.: “An energy efficient approximate adder with carry skip for error resilient neuromorphic VLSI systems,” ICCAD (2013) 130 (DOI: 10.1109/ICCAD.2013.6691108).
[15] I.-C. Lin, et al.: “High-performance low-power carry speculative addition with variable latency,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst. 23 (2015) 1591 (DOI: 10.1109/TVLSI.2014.2355217).
[16] A. K. Verma, et al.: “Variable latency speculative addition: A new paradigm for arithmetic circuit design,” DATE (2008) 1250 (DOI: 10.1109/DATE.2008.4484850).
[17] D. Mohapatra, et al.: “Scalable effort hardware design: Exploiting algorithmic resilience for energy efficiency,” DATE (2011) 1.
[18] X. Yang, et al.: “Approximate adder with hybrid prediction and error compensation technique,” ISVLSI (2016) 373 (DOI: 10.1109/ISVLSI.2016.16).
[19] J. Hu, et al.: “A new approximate adder with low relative error and correct sign calculation,” DATE (2015) 1449.
[20] L. Li and H. Zhou: “On error modeling and analysis of approximate adders,” ICCAD (2014) 511 (DOI: 10.1109/ICCAD.2014.7001399).
[21] H. Jiang, et al.: “A review, classification, and comparative evaluation of approximate arithmetic circuits,” ACM J. Emerg. Technol. Comput. Syst. 13 (2017) 1 (DOI: 10.1145/3094124).
[22] S. Angizi, et al.: “Composite spintronic accuracy configurable adder for low power digital signal processing,” ISQED (2017) 391 (DOI: 10.1109/ISQED.2017.7918347).
[23] H. Cai, et al.: “Approximate computing: A survey,” ACM Comput. Surv. 52 (2019) 6 (DOI: 10.1145/3285067.3285101).
[24] V. Gupta, et al.: “Low-power digital signal processing using approximate adders,” IEEE Trans. Comput.-Aided Design Integr. Circuits Syst. 32 (2013) 124 (DOI: 10.1109/TCAD.2012.2217962).
[25] Z. Yang, et al.: “Approximate XOR/XNOR based adders for inexact computing,” IEEE-NANO (2013) 690 (DOI: 10.1109/NANO.2013.6720793).
[26] Z. Yang, et al.: “Transmission gate-based approximate adders for inexact computing,” NANOARCH (2015) 145 (DOI: 10.1109/NANOARCH.2015.7180603).
[27] W. Liu, et al.: “Design of approximate radix-4 booth multipliers for error tolerant computing,” IEEE Trans. Comput. 66 (2017) 1435 (DOI: 10.1109/TC.2017.2672976).
[28] S. Venkatachalam and S.-B. Ko: “Design of power and area efficient approximate multipliers,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst. 25 (2017) 1782 (DOI: 10.1109/TVLSI.2016.2643639).
[29] H. A. F. Almurib, et al.: “Inexact designs for approximate low power addition by cell replacement,” DATE (2016) 660 (DOI: 10.1109/DATE.2016.6176685).
[30] J. Liang, et al.: “New metrics for the reliability of approximate and probabilistic adders,” IEEE Trans. Comput. 62 (2013) 1760 (DOI: 10.1109/TC.2012.146).