

Novel Approximate Absolute Difference Hardware

Ahmet Can Mert, Hasan Azgin, Ercan Kalali, Ilker Hamzaoglu
Faculty of Engineering and Natural Sciences
Sabanci University
Istanbul, Turkey
{ahmetcanmert, hasanazgin, ercankalali, hamzaoglu}@sabanciuniv.edu

Abstract—Approximate hardware designs have higher performance, smaller area or lower power consumption than exact hardware designs at the expense of lower accuracy. Absolute difference (AD) operation is heavily used in many applications such as motion estimation (ME) for video compression, ME for frame rate conversion, stereo matching for depth estimation. Since most of the applications using AD operation are error tolerant by their nature, approximate hardware designs can be used in these applications. In this paper, novel approximate AD hardware designs are proposed. The proposed approximate AD hardware implementations have higher performance, smaller area and lower power consumption than exact AD hardware implementations at the expense of lower accuracy. They also have less error, smaller area and lower power consumption than the approximate AD hardware implementations which use approximate adders proposed in the literature.

Keywords—Absolute difference, motion estimation, approximate computing, hardware design, low power.

I. INTRODUCTION

Approximate computing is a promising approach for increasing performance, reducing area or decreasing power consumption of exact hardware designs at the expense of lower accuracy [1]-[4]. Approximate computing allows designing faster, lower area and lower power consuming hardware than the exact optimized hardware designs, by trading off speed, area and power consumption with quality. Therefore, it can be used in error tolerant applications.

Absolute difference (AD) operation is heavily used in many applications such as motion estimation (ME) for video compression [5], ME for frame rate conversion [6], stereo matching for depth estimation [7]. Since most of the applications using AD operation are error tolerant by their nature, approximate hardware designs can be used in these applications.

Approximate AD hardware can be designed by using general purpose approximate adders proposed in the literature in exact AD hardware. However, better approximate AD hardware can be designed by using special approximation techniques for AD hardware instead of using general purpose approximate adders.

In this paper, four novel approximate AD hardware designs are proposed. These approximate AD hardware designs use special approximation techniques for AD hardware instead of using general purpose approximate adders proposed in the literature. The proposed approximate AD hardware are compared with two exact baseline AD hardware and ten other approximate AD hardware.

These ten approximate AD hardware are obtained by using five approximate adders proposed in the literature [8]-[10] in the two exact baseline AD hardware. These two exact baseline AD hardware have exact subtractors. Therefore, approximate adders proposed in the literature are used as approximate subtractors by giving 2's complement of one input to the approximate adders instead of the original input.

Two exact baseline AD hardware and all fourteen approximate AD hardware are implemented using Verilog HDL. The Verilog RTL codes are synthesized and mapped to a Xilinx XC6VLX130T FF1156 FPGA with speed grade 3 using Xilinx ISE 14.7. The FPGA implementations are verified with post place and route simulations.

The proposed approximate AD hardware implementations have higher performance, smaller area and lower power consumption than exact AD hardware implementations at the expense of lower accuracy. The proposed approximate AD hardware implementations have less error, smaller area and lower power consumption than the approximate AD hardware implementations which use approximate adders proposed in the literature [8]-[10].

In the hardware implementations of applications using AD operations such as video compression, frame rate conversion and depth estimation, large number of parallel AD hardware such as 512, 1024 are used. In this paper, area and power consumption results are reported for one AD hardware. Area and power consumption reductions achieved by using the approximate AD hardware proposed in this paper would be much larger for the hardware implementations using large number of parallel AD hardware.

The rest of the paper is organized as follows. In Section II, proposed approximate absolute difference hardware are explained. Implementation results are given in Section III. Finally, Section IV presents conclusions.

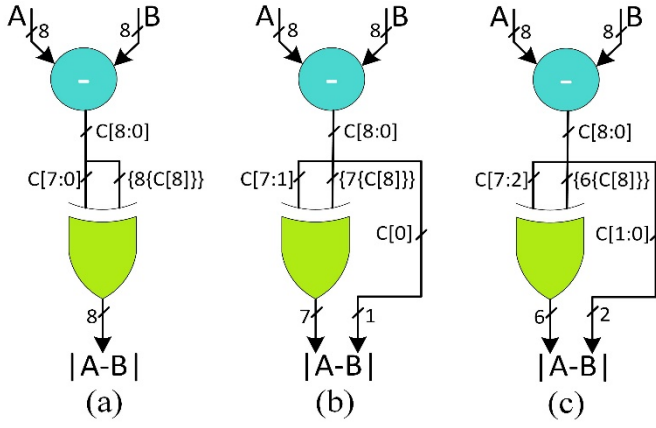


Fig. 1. Proposed approximate absolute difference hardware; (a) proposed_0, (b) proposed_1, (c) proposed_2.

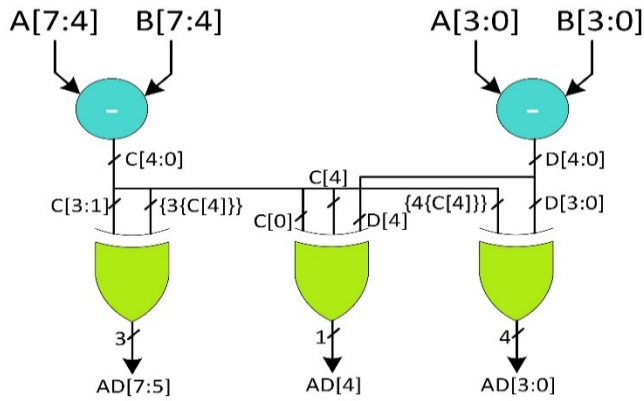


Fig. 2. Proposed approximate absolute difference hardware (proposed_half).

II. PROPOSED APPROXIMATE ABSOLUTE DIFFERENCE HARDWARE

The three proposed approximate AD hardware are shown in Fig. 1. As shown in Fig. 1 (a), proposed_0 hardware consists of a subtractor and XOR gates. First, two 8-bit inputs A and B are subtracted with an exact subtractor hardware. Then, each bit of the subtraction result is XOR'ed with the sign bit of the subtraction result. If $A \geq B$, the sign bit is 0. Therefore, each bit is XOR'ed with 0. In this case, proposed_0 hardware computes the correct absolute difference. If $A < B$, the sign bit is 1. Therefore, each bit is XOR'ed with 1. In this case, the output of proposed_0 hardware is 1 less than the correct absolute difference. Therefore, the maximum error of proposed_0 hardware is 1.

As shown in Fig. 1 (b), in proposed_1 hardware, the most significant 7 bits of subtraction result is XOR'ed with the sign bit. But, the least significant bit of the subtraction result is not XOR'ed with the sign bit. Therefore, proposed_1 hardware has 1 less XOR gate than proposed_0 hardware. However, its maximum error is 2 which is 1 more than the maximum error of proposed_0 hardware.

As shown in Fig. 1 (c), in proposed_2 hardware, the most significant 6 bits of subtraction result is XOR'ed with the sign bit. But, the least significant 2 bits of the subtraction result is

not XOR'ed with the sign bit. Therefore, proposed_2 hardware has 2 less XOR gates than proposed_0 hardware. However, its maximum error is 4 which is 3 more than the maximum error of proposed_0 hardware.

The proposed_half approximate AD hardware is shown in Fig. 2. It uses two 4-bit subtractors instead of one 8-bit subtractor. The results of two 4-bit subtractors are XOR'ed with the sign bit of first 4-bit subtraction result. The middle bit of AD is calculated by XOR'ing sign bits of both 4-bit subtraction results and the least significant bit of first 4-bit subtraction result.

Since using two 4-bit subtractors instead of one 8-bit subtractor significantly reduces the delay of critical path which is carry propagation, proposed_half hardware is faster than proposed_0, proposed_1 and proposed_2 hardware. However, proposed_half hardware has a maximum error of 33 which is larger than the maximum errors of proposed_0, proposed_1 and proposed_2 hardware.

The four approximate AD hardware proposed in this paper are compared with ten other approximate AD hardware. These ten approximate AD hardware are obtained by using five approximate adders proposed in the literature [8]-[10] in the two exact baseline AD hardware shown in Fig. 3. These two exact baseline AD hardware have exact subtractors. Therefore, approximate adders proposed in the literature are used as approximate subtractors by giving 2's complement of one input to the approximate adders instead of the original input.

Ten approximate AD hardware are obtained by replacing exact subtractors in the two exact baseline AD hardware with the following five approximate adders in the literature; Almost Correct Adder I (ACA_I) [8], Almost Correct Adder II (ACA_II) [8], Error Tolerant Adder II (ETA_II) [9], Generic Accuracy Configurable Adder with N, R and P values of 8, 1 and 1, respectively (GEAR_N8_R1_P2) [10] and Generic Accuracy Configurable Adder with N, R and P values of 8, 2 and 4, respectively (GEAR_N8_R2_P4) [10].

Accuracy analysis of the approximate AD hardware proposed in this paper and these ten approximate AD hardware is shown in Table I. For example, B1_ACA_I hardware is obtained by using ACA_I approximate adder in the exact baseline 1 absolute difference hardware. B2_ACA_I hardware is obtained by using ACA_I approximate adder in the exact baseline 2 absolute difference hardware. The eight other approximate AD hardware in Table I are obtained similarly. The proposed_0, proposed_1 and proposed_2 hardware have less accuracy than the ten approximate AD hardware. However, they have much less maximum and average error than the ten approximate AD hardware.

III. IMPLEMENTATION RESULTS

Two exact baseline AD hardware and all fourteen approximate AD hardware are implemented using Verilog HDL. The Verilog RTL codes are verified with RTL simulations. RTL simulation results matched the results of MATLAB implementations of the corresponding approximate AD algorithms.

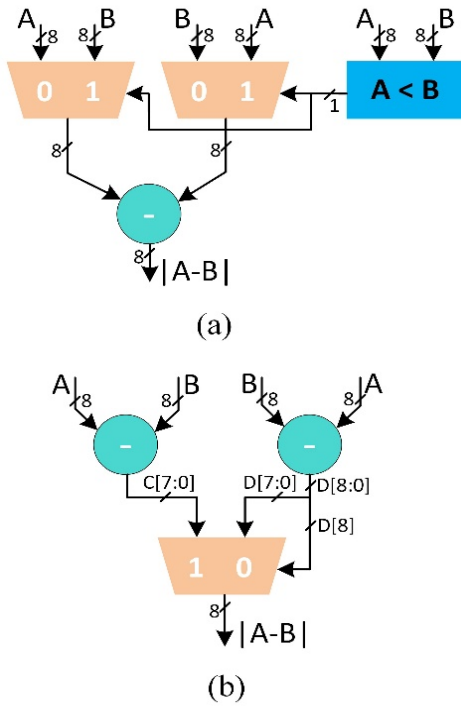


Fig. 3. Exact absolute difference hardware (a) Baseline 1 (b) Baseline 2.

TABLE I. ACCURACY ANALYSIS OF APPROXIMATE ABSOLUTE DIFFERENCE HARDWARE

	Max. Error	Average Error	Accuracy (%)
Proposed_0	1	0.498	50.195
Proposed_1	2	0.496	75.195
Proposed_2	4	0.992	62.695
Proposed_half	33	7.637	39.941
B1_ACA_I	128	2.188	96.679
B2_ACA_I	128	3.418	95.312
B1_ACA_II	64	5.906	84.179
B2_ACA_II	64	7.168	81.250
B1_ETAI	64	5.926	84.179
B2_ETAI	64	7.168	81.250
B1_GeAr_R1_P2	144	10.172	75.488
B2_GeAr_R1_P2	144	14.168	69.922
B1_GeAr_R2_P4	64	1.125	98.242
B2_GeAr_R2_P4	64	1.480	97.656

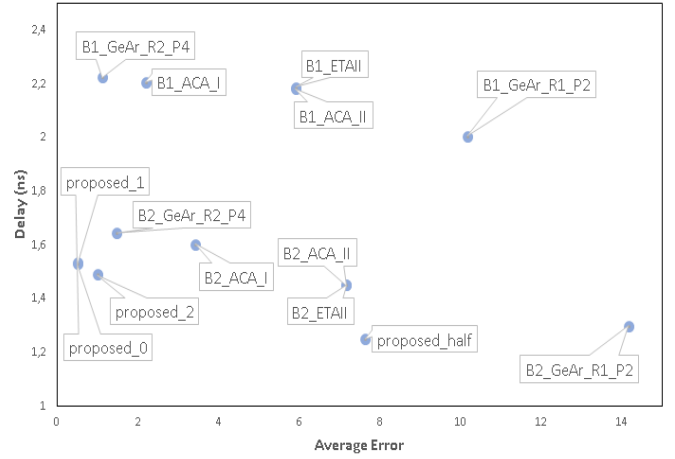


Fig. 4. Average error vs. delay graph.

The Verilog RTL codes are synthesized and mapped to a Xilinx XC6VLX130T FF1156 FPGA with speed grade 3 using Xilinx ISE 14.7. The FPGA implementations are verified with post place and route simulations. Post place and route simulation results matched the results of MATLAB implementations of the corresponding approximate AD algorithms.

Power consumptions of all the FPGA implementations are estimated using Xilinx XPower Analyzer tool. Post place and route timing simulations are performed at 100 MHz and the signal activities of these timing simulations are stored in VCD files. Then, they are used for estimating the power consumptions of the FPGA implementations.

The FPGA implementation results are shown in Table II. All four approximate AD hardware proposed in this paper have higher performance and less area than both exact baseline hardware. Proposed_2 and proposed_half hardware also have lower power consumption than both exact baseline hardware.

The proposed_0, proposed_1 and proposed_2 hardware have less area than the other ten approximate AD hardware. They also have much less maximum and average error than the other ten approximate AD hardware. Proposed_2 and proposed_half hardware also have lower power consumption than the other ten approximate AD hardware.

Average error vs. delay graph for all 14 approximate AD hardware is shown in Fig. 4. Proposed_0, proposed_1 and proposed_2 hardware have the best average error vs. delay performance.

Proposed_0 hardware has the largest area and power consumption among the four approximate AD hardware proposed in this paper. However, it has the smallest maximum and average errors. Proposed_1 hardware has less area than proposed_0. It has same power consumption as proposed_0. It has higher accuracy than proposed_0. It has almost the same average error as proposed_0. But, it has larger maximum error than proposed_0. Therefore, either proposed_0 or proposed_1 hardware can be used in an application depending on its accuracy and hardware requirements.

TABLE II. FPGA IMPLEMENTATION RESULTS OF APPROXIMATE ABSOLUTE DIFFERENCE HARDWARE

	LUT	Slice	Frequency (MHz)	Power (mW)
Exact Baseline 1	20	15	499	4.64
Exact Baseline 2	26	10	599	4.74
Proposed_0	19	10	651	5.26
Proposed_1	17	10	653	5.26
Proposed_2	16	9	671	4.27
Proposed_half	18	7	800	4.52
B1_ACA_I	36	12	453	5.95
B2_ACA_I	34	15	624	5.59
B1_ACA_II	31	13	458	5.54
B2_ACA_II	30	15	689	5.22
B1_ETAI	31	15	457	5.62
B2_ETAI	30	17	688	5.17
B1_GeAr_R1_P2	29	13	499	5.33
B2_GeAr_R1_P2	26	19	771	5.03
B1_GeAr_R2_P4	32	17	449	5.23
B2_GeAr_R2_P4	34	14	608	5.21

Proposed_2 hardware is faster than proposed_0 and proposed_1 hardware. It also has less area and lower power consumption than proposed_0 and proposed_1 hardware. However, it has larger maximum and average error than proposed_0 and proposed_1 hardware. Therefore, it can be used in applications which can tolerate its maximum and average error.

Since using two 4-bit subtractors instead of one 8-bit subtractor significantly reduces the delay of critical path which is carry propagation, proposed_half hardware is the fastest approximate AD hardware. It also has less area than proposed_0, proposed_1, and proposed_2 hardware. However, it has larger maximum and average error than proposed_0, proposed_1, and proposed_2 hardware. Therefore, it can be used in applications which can tolerate its maximum and average error.

In the hardware implementations of applications using AD operations such as video compression, frame rate conversion and depth estimation, large number of parallel AD hardware such as 512, 1024 are used. In this paper, area and power consumption results are reported for one AD hardware. Area and power consumption reductions achieved by using the approximate AD hardware proposed in this paper would be much larger for the hardware implementations using large number of parallel AD hardware.

IV. CONCLUSION

In this paper, four novel approximate AD hardware designs are proposed. These approximate AD hardware designs use special approximation techniques for AD hardware instead of using general purpose approximate adders proposed in the literature. The proposed approximate AD hardware implementations have higher performance, smaller area and lower power consumption than two exact AD hardware implementations at the expense of lower accuracy. The proposed approximate AD hardware implementations have less error, smaller area and lower power consumption than ten approximate AD hardware implementations which use approximate adders proposed in the literature.

None of the four approximate AD hardware proposed in this paper (proposed_0, proposed_1, proposed_2, proposed_half) is better than the other three in terms of all metrics; maximum error, average error, hardware performance, area and power consumption. Therefore, one of them can be used in an application depending on its accuracy and hardware requirements.

ACKNOWLEDGEMENT

This research was supported in part by the Scientific and Technological Research Council of Turkey (TUBITAK) under the contract 118E134.

REFERENCES

- [1] Q. Xu, T. Mytkowicz and N. S. Kim, "Approximate Computing: A Survey," *IEEE Design & Test*, vol. 33, no. 1, pp. 8-22, Feb. 2016.
- [2] M. Shafique, R. Hafiz, S. Rehman, W. El-Harouni, J. Henkel, "Cross-Layer approximate computing: From logic to architectures," *IEEE Design Automation Conference*, Austin, USA, Jun. 2016.
- [3] C. Liu, J. Han, F. Lombardi, "An analytical framework for evaluating the error characteristics of approximate adders," *IEEE Trans. on Computers*, vol. 64, no. 5, pp. 1268-1281, May 2015.
- [4] E. Kalali and I. Hamzaoglu, "Approximate HEVC fractional interpolation filters and their hardware implementations," *IEEE Trans. on Consumer Electronics*, vol. 64, no. 3, pp. 285-291, Aug. 2018.
- [5] O. Tasdizen, A. Akin, H. Kukner, and I. Hamzaoglu, "Dynamically variable step search motion estimation algorithm and a dynamically reconfigurable hardware for its implementation," *IEEE Trans. on Consumer Electronics*, vol. 55, no. 3, pp. 1645-1653, Aug. 2009.
- [6] M. Cetin and I. Hamzaoglu, "An adaptive true motion estimation algorithm for frame rate conversion of high definition video and its hardware implementations," *IEEE Trans. on Consumer Electronics*, vol. 57, no. 2, pp. 923-931, May 2011.
- [7] S. Jin et al., "FPGA Design and Implementation of a Real-Time Stereo Vision System," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 20, no. 1, pp. 15-26, Jan. 2010.
- [8] A. K. Verma, P. Brisk, and P. Ienne, "Variable latency speculative addition: A new paradigm for arithmetic circuit design," *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, Munich, Germany, Apr. 2008.
- [9] N. Zhu, W. L. Goh, and K. S. Yeo, "An enhanced low-power high-speed adder for error-tolerant application," *12th Int. Symposium Integrated Circuit*, Singapore, Singapore, Dec. 2009.
- [10] M. Shafique, W. Ahmad, R. Hafiz, and J. Henkel, "A low latency generic accuracy configurable adder," *ACM/EDAC/IEEE Design Automation Conference*, San Francisco, USA, Jun. 2015.