© Journal of the Technical University - Sofia
Plovdiv branch, Bulgaria
"Fundamental Sciences and Applications"
Vol. 28, 2022

Study of Different Adders Focused on Ultimate Execution Speed in FPGA Based Edge Detection

Dimitre Kromichev

Department of Marketing and International Economic Relations,
University of Plovdiv, 24 Tzar Asen Street,
Plovdiv 4000, Bulgaria

dkromichev@yahoo.com

Abstract. The goal of ultimate execution speed in FPGA based edge detection which uses Gaussian filtering requires defining the values of maximum operating frequency and minimum number of clock cycles taken to secure mathematically accurate result. A critical component in achieving that goal is the simultaneous addition of convolution results calculated by the weighted average function in the process of filtering an input image pixel. The paper studies the capabilities of different addition algorithms to be used in FPGA based edge detection for simultaneous addition of convolution results in terms of guaranteeing minimum number of clock cycles for various Gaussian filter sizes at the maximum operating frequency. On that basis, analyzed is the impact of simultaneous addition of convolution results on the organization of computations in the Gaussian filtering module of FPGA based edge detection. Ten Intel (Altera) FPGA families are used in conducting the explorations.

1. INTRODUCTION

In [1][13] presented is a comparison of carry select adder, carry look ahead adder, carry increment adder, ripple carry adder and Brent Kung adder in a Xilinx platform. In [2][4][12][17] investigated are four types of parallel prefix adder (Kogge Stone, Spanning Tree, Brent Kung, sparse Kogge stone), ripple carry adder, carry look ahead adder and carry skip adder by implementing them in Spartan 6 and Spartan 3E. The conclusion is that ripple carry adder has the least delay of all analyzed adders. Due to the presence of a fast carry-chain, the ripple carry adder designs exhibit better delay performance up to 128 bits. It is argued that carry tree adders have a speed advantage over the ripple carry adder as bit widths approach 256. In [18] presented is a detailed analysis of the fast routing links and special logic needed to perform carry operations in modern FPGAs. In [3][5][11][14][15][16][20] proposed are studies of hardware optimized 8-bit, 16-bit, 32-bit and 64-bit approximate adders aimed at improving the mean squared error. In the proposed 8-bit and 16-bit designs, ripple carry adders are used as sub-adders in all approximate adders. The implementation in Artix 7 of 32-bit addition involving a 8-bit least significant inaccurate sub-adder is reported to reduce the minimum clock period by 7.1% compared to the accurate FPGA adder; for 64-bit addition involving a 8-bit least significant inaccurate sub-adder, the minimum clock period is reported to be reduced by 8.3% compared to the accurate FPGA adder. Compared to the 16-bit accurate adder, speed gain is reported to be 38%, In [6] a new approximate adder design is compared with accurate adders of equal size and a conclusion is arrived at that the proposed approximate adder is capable of higher operating frequencies. For 32-bit addition, the proposed approximate adder is reported to achieve 25% reduction in the number of utilized LUTs compared to the accurate adder with no decrease in performance. In [7] proposed is a new approach to carry increment adder, carry save adder, carry select adder, carry skip adder and ripple carry adder by using conventional and multiplexer based full adder for 32 bit; the adders are studied in terms of area, power distribution, delay and gate count in Spartan 6. In

[10] the implementation of carry increment adder, carry save adder, carry select adder, carry skip adder and ripple carry adder in Spartan 3E shows the highest clock frequencies for carry increment adder and carry select adder. In [8] four 16-bit adders – ripple carry, Brent Kung, Kogge stone and carry skip adder, are designed. It is found that carry skip adder consumes higher power in comparison to the ripple carry adder. In [9] 64-bit ripple carry adder and carry look ahead adder are compared in Spartan 7. It is found that the delay of carry look ahead adder is less than that of ripple carry adder. In [19] on the basis of 4-bit designs it is found that the propagation delay of carry save adder 3.37 ns, carry look ahead adder - 4.005 ns and ripple carry adder - 4.233 ns. In ripple carry adder delay increases linearly with the bit length. Thus it is not efficient when large data widths. In carry look ahead adder, due to the simultaneous computing of carry, delay is reduced.

The objective of this paper is to study the capabilities of different addition algorithms in the simultaneous addition of convolution results in FPGA based edge detection. The focus is on guaranteeing minimum number of clock cycles in computing the weighted average function for various Gaussian filter sizes at the maximum operating frequency $F_{\max}$. The task is to: (1) explore the number of levels of adders which can fit within a clock period of the system clock $Tclk$ at $F_{\max}$, thus defining the number of clock cycles required by the simultaneous addition of convolution results for different Gaussian filter sizes; (2) the capabilities of parallel addition to contribute to the goal of securing a Gaussian filtered pixel every single clock cycle at $F_{\max}$; (2) analyze the impact of simultaneous addition of convolution results on the organization of computations in the Gaussian filtering module of FPGA based edge detection. The explorations are conducted on the basis of ten Intel (Altera) FPGA families. Used tools: Scilab, Intel (Altera) Quartus, TimeQuest Timing Analyzer, ModelSim. The hardware description language is VHDL. Relevant to the analyses and conclusions arrived at are gray scale images.

2. LEVELS OF ADDERS IN TERMS OF MAXIMUM OPERATING FREQUENCY

Maximum operating frequency $F_{\max}$ in FPGA based edge detection which uses Gaussian filtering is defined by $F_{\max}$ of Gaussian filtering module. Because $F_{\max}$ of Gaussian filtering is defined by the maximum operating frequency of hard multiplier $F_{\max}(hardMult)$, the basic requirement for the number of levels of adders in the simultaneous addition of convolution results is

$$\frac{1}{Tclk(adderLevels)} > \frac{1}{Tclk(hardMult)} \tag{1}$$

where

$Tclk(adderLevels)$ is the minimum clock period which guarantees positive slack for a concrete number of levels of two-input adders involved in the simultaneous addition of convolution results

$Tclk(hardMult)$ is the clock period defined by $F_{\max}(hardMult)$.

For all multiplications of Gaussian filter coefficients and image pixels executed within a single clock cycle, the number of addends which must be added simultaneously is a function of Gaussian filter size $z * z$, z is an odd number and $z \geq 3$. Hence the total number of levels of two-input adders required to secure mathematically accurate addition result is defined by

$$\begin{aligned} L &= P \quad \text{for } k = 2^P \\ L &= P+1 \quad \text{for } 2^P < k < 2^{P+1}, \quad P \in N \end{aligned} \tag{2}$$

where

L is the number of consecutive levels of two-input adders

k is the number of addends,.

Therefore, for each concrete algorithm involved in the execution of addition operation the task is to define the largest values for L and k which satisfy (1).

In (2) k is a function of the Gaussian filter size $z * z$. L is directly proportional to the maximum critical path delay of the concrete addition algorithm executed as a purely combinational schematic. Therefore, L is a function of the data width of the addends. The input data width is controlled by the organization of computations in Gaussian filtering. There are two critical input data widths. They are determined by the boundaries of the interval containing all possible input data widths in bits. This interval represents all the values which can be calculated in the Gaussian filtering module in FPGA based edge detection. Because the largest gray scale image pixel value is 2^8-1 and $F_{\max}$ in FPGA based edge detection which uses Gaussian filtering is defined by $F_{\max}(hardMult)$ of 18x18 hard multiplier, it follows that:

- If all convolution results are divided by the normalization factor and then the results are added Simultaneously, the largest possible addend is 8 bits.
- If all convolution results are added simultaneously and then the sum is divided by the normalization factor, the largest possible addend is 25 bits.

Therefore, according to the interval containing all possible input data widths, the organization of computations defines two limitations for L :

- The largest value of L which satisfies (1) is defined by the critical path delay of 8 bit adder.
- The largest value of L which satisfies (1) is defined by the critical path delay of 25 bit adder.

3. EXPLORING DIFFERENT ADDERS FOR SIMULTANEOUS ADDITION OF CONVOLUTION RESULTS

Methodology:

- Ten adders are used in the explorations: ripple carry adder, carry select adder, carry look ahead adder, carry save adder, carry skip adder, carry increment adder, Kogge stone adder, spanning tree adder, Brent Kung adder, sparse Kogge stone adder
- In Intel (Altera) FPGA, adding two integers by using the Lpm_add_sub megafunction or symbol ‘’+’’ is synthesized to ripple carry adder. The functionality of all other adders is implemented in hardware description language
- $F_{\max}(hardMult)$ of 18x18 multiplier is defined for the targeted Intel (Altera) FPGA families. Lpm_mult megafunction is used to implement the hard multiplier in Cyclone II-V and Stratix I-V. In Cyclone family multipliers are implemented with Lpm_mult megafunction using only logic resources. The obtained results are for the highest speed grade in a concrete FPGA family. Experiments are conducted using all values within the inputs' range of the multiplier.
- The number of levels of adders is explored within a single clock cycle
- The number of addends at level # 1 is defined by the Gaussian filter size
- Maximum input data width for $z * z$ number of integer division operations executed immediately after the convolution results are available at the output of hard multipliers is 8 bits
- Maximum input data width for a single integer division operation executed over the sum of all convolution results is 25 bits.

The obtained results are shown in Table 1, Table 2, Table 3 and Table 4.

4. ANALYSIS OF RESULTS

On the basis of satisfying the requirements of (1) and (2), the obtained results for executing simultaneous addition in calculating the weighted average function by using ten different algorithms implemented in the targeted Intel (Altera) FPGA families show that:

- With ripple carry adder, carry select adder and carry save adder simultaneous addition of 8 bit addends can be executed within a single clock cycle for Gaussian filter sizes 3x3, 5x5 and 7x7
- When ripple carry, carry select adder and carry save adders are used for Gaussian filters with $z \geq 9$ and input data width = 8 bits simultaneous addition must be pipelined to satisfy (1)

TABLE 1. Number of levels of adders 8 bits in executing simultaneous addition with different algorithms for input data width of addends = 8 bits under the condition $\frac{1}{Tclk(adderLevels)} > \frac{1}{Tclk(hardMult)}$

FPGA family	Ripple carry adder	Carry select adder	Carry look ahead adder	Carry save adder	Carry skip adder
Cyclone	**6**	**6**	**4**	**6**	**4**
Cyclone II	**6**	**6**	**4**	**6**	**4**
Cyclone III	**6**	**6**	**4**	**6**	**4**
Cyclone IV	**6**	**6**	**4**	**6**	**4**
Cyclone V	**6**	**6**	**4**	**6**	**4**
Stratix	**6**	**6**	**4**	**6**	**4**
Stratix II	**6**	**6**	**4**	**6**	**4**
Stratix III	**6**	**6**	**4**	**6**	**4**
Stratix IV	**6**	**6**	**4**	**6**	**4**
Stratix V	**6**	**6**	**4**	**6**	**4**

TABLE 2. Number of levels of adders 8 bits in executing simultaneous addition with different algorithms for input data width of addends = 8 bits under the condition $\frac{1}{Tclk(adderLevels)} > \frac{1}{Tclk(hardMult)}$

FPGA family	Carry increment adder	Kogge stone adder	Spanning tree adder	Brent Kung adder	Sparse Kogge stone adder
Cyclone	**5**	**5**	**5**	**5**	**5**
Cyclone II	**5**	**5**	**5**	**5**	**5**
Cyclone III	**5**	**5**	**5**	**5**	**5**
Cyclone IV	**5**	**5**	**5**	**5**	**5**
Cyclone V	**5**	**5**	**5**	**5**	**5**
Stratix	**5**	**5**	**5**	**5**	**5**
Stratix II	**5**	**5**	**5**	**5**	**5**
Stratix III	**5**	**5**	**5**	**5**	**5**
Stratix IV	**5**	**5**	**5**	**5**	**5**
Stratix V	**5**	**5**	**5**	**5**	**5**

- With carry increment adder, Kogge stone adder, spanning tree adder, Brent Kung adder and sparse Kogge stone adder simultaneous addition of 8 bit addends can be executed within a single clock cycle for Gaussian filter sizes 3x3 and 5x5
- When carry increment, Kogge stone, spanning tree, Brent Kung and sparse Kogge stone adders are used for Gaussian filters with $z \geq 7$ and input data width = 8 bits simultaneous addition must be pipelined to satisfy (1)
- With carry look ahead adder and carry skip adder simultaneous addition of 8 bit addends can be executed within a single clock cycle only for Gaussian filter size 3x3
- When carry look ahead and carry skip adders are used for Gaussian filters with $z \geq 5$ and input data width = 8 bits simultaneous addition must be pipelined to satisfy (1)
- When the input data width of the addends is 25 bits none of the ten adders can be used to execute sumaltenious addition within a single clock cycle for any Gaussian filter size. In this case, according to (2), for all filter sizes simultaneous addition must be pipelined.

On a comparative basis, the obtained exploration results define two facts focused on the goal of ultimate execution speed in FPGA based edge detection:

TABLE 3. Number of levels of adders 8 bits in executing simultaneous addition with different algorithms for input data width of addends = 25 bits under the condition $\frac{1}{Tclk(adderLevels)} > \frac{1}{Tclk(hardMult)}$

FPGA family	Ripple carry adder	Carry select adder	Carry look ahead adder	Carry save adder	Carry skip adder
Cyclone	**2**	**2**	**1**	**2**	**1**
Cyclone II	**2**	**2**	**1**	**2**	**1**
Cyclone III	**2**	**2**	**1**	**2**	**1**
Cyclone IV	**2**	**2**	**1**	**2**	**1**
Cyclone V	**2**	**2**	**1**	**2**	**1**
Stratix	**2**	**2**	**1**	**2**	**1**
Stratix II	**2**	**2**	**1**	**2**	**1**
Stratix III	**2**	**2**	**1**	**2**	**1**
Stratix IV	**2**	**2**	**1**	**2**	**1**
Stratix V	**2**	**2**	**1**	**2**	**1**

TABLE 4. Number of levels of adders 8 bits in executing simultaneous addition with different algorithms for input data width of addends = 25 bits under the condition $\frac{1}{Tclk(adderLevels)} > \frac{1}{Tclk(hardMult)}$

FPGA family	Carry increment adder	Kogge stone adder	Spanning tree adder	Brent Kung adder	Sparse Kogge stone adder
Cyclone	**1**	**1**	**1**	**1**	**1**
Cyclone II	**1**	**1**	**1**	**1**	**1**
Cyclone III	**1**	**1**	**1**	**1**	**1**
Cyclone IV	**1**	**1**	**1**	**1**	**1**
Cyclone V	**1**	**1**	**1**	**1**	**1**
Stratix	**1**	**1**	**1**	**1**	**1**
Stratix II	**1**	**1**	**1**	**1**	**1**
Stratix III	**1**	**1**	**1**	**1**	**1**
Stratix IV	**1**	**1**	**1**	**1**	**1**
Stratix V	**1**	**1**	**1**	**1**	**1**

- Ripple carry adder presents the best ratio between F_{max} and the minimum number of clock cycles required to calculate mathematically accurate result
- With respect to the organization of computations in Gaussian filtering, it is a must that all convolution results are divided by the normalization factor and then the results are added simultaneously.

.

CONCLUSION

In this paper, presented is a study of ten adders with focus on the goal of ultimate execution speed in FPGA based edge detection which uses Gaussian filtering. The exploration results obtained on the basis of ten Intel (Altera) FPGA families show that ripple carry adder provides the best ratio between maximum operating frequency and minimum number of clock cycles taken to secure mathematically accurate result. The exploration results prove that the optimal organization of computations in Gaussian filtering is: all convolution results are divided by the normalization factor and then the results are added simultaneously.

 Online ISSN 2603-459X

REFERENCES

1. Ananthakrishnan, A. Ajit, A. P. V., K. Haridas, N. M. Nambiar, D. S., FPGA Based Performance Comparison of Different Basic Adder Topologies with Parallel Processing Adder, 2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA), 2019, pp. 87-92
2. Avinash Shrivastava, Shefali Churhe, Hemlata Bhagat, Rajni Wamankar, Design and Estimation of delay, power and area for Parallel prefix adders, International Journal of Engineering Research and Application, Vol. 7, Issue 4, (Part -5), April 2017, pp.01-08
3. A. Dalloo, A. Najafi and A. Garcia-Ortiz, Systematic Design of an Approximate Adder: The Optimized Lower Part Constant-OR Adder, in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 26, no. 8, Aug. 2018, pp. 1595-1599
4. A. Maheswara Reddy, G. V. Vijayalakshmi, Speed Comparison Parallel Prefix Adders with RCA and CSA in FPGA, International Journal of Innovative Research in Electronics and Communications (IJIREC) Volume 2, Issue 5, July 2015, pp. 8-16
5. Balasubramanian, P. Maskell, D.L. Hardware optimized and error reduced approximate adder. Electronics 2019, 8, 1212
6. Balasubramanian, P., Maskell, D., Hardware efficient approximate adder design, Proceedings of the IEEE Region 10 Conference, Jeju, Korea, October 2018, pp. 28–31
7. D. Mohanapriya, N.Saravanakumar, A Comparative Analysis of Different 32-bit Adder Topologies with Multiplexer Based Full Adder, International Journal of Engineering Science and Computing, May 2016, pp. 4850-4854
8. H.V. Ravish Aradhya, Apoorva Raghunandan, Performance Analysis of Advanced Adders Under Changing Technologies, International Journal of Innovations in Engineering and Technology (IJIET), Volume 12, Issue 2, January 2019, pp. 414-419
9. Konda Sai Prakash Reddy Designing Various 64-bit Adders using VHDL in VIVADO, Journal of Innovation in Electronics and Communication Engineering, Vol.9 (2), Jul-Dec 2019, pp. 6-11
10. Komal M. Lineswala, Zalak M. Vyas, Comparative Analysis of Various Adders using VHDL, International Journal of Engineering and Technical Research (IJETR), Volume-3, Issue-4, April 2015, pp. 50-55
11. Lee, J. Seo, H. Kim, Y. Kim, Y Approximate adder design with simplified lower-part approximation, IEICE Electron. Express 2020, 17, 20200218
12. Madhavi K., Kuppam N Chandrasekar, Design and Comparative Analysis of Conventional Adders and Parallel Prefix Adders, International Journal of Engineering Trends and Technology (IJETT), Vol. 9 May 2016, pp. 435-439
13. Nikhita Matti, Rohini Hongal, R B Shettar, Performance Analysis of Different n-bit Adders Using Reversible Logic on FPGA Board Using Chipscope, International Journal of Engineering Sciences & Research Technology, August 2017, pp. 307-311
14. Perri, S. Spagnolo, F. Frustaci, F. Corsonello, P; Efficient approximate adders for FPGA based data paths, Electronics 2020, 9, 1529-1233
15. Prabakaran, B.S. Rehman, S. Hanif, M.A. Ullah, S. Mazaheri, G. Kumar, A. Shafique, M. DeMAS, An efficient design methodology for building approximate adders for FPGA based systems. In Proceedings of the Design, Automation and Test in Europe, Dresden, Germany, March 2018, pp. 19–23
16. Q. Lu, A. M. Gharehbaghi and M. Fujita, Approximate Arithmetic Circuit Design Using a Fast and Scalable Method, 2019 IFIP/IEEE 27th International Conference on Very Large Scale Integration (VLSI-SoC), 2019, pp. 65-70
17. Samraj Daphni, Kasinadar Sundari Vijula Grace Design and Analysis of 32-bit Parallel Prefix Adders for Low Power VLSI Applications Advances in Science, Technology and Engineering Systems Journal Vol. 4, No. 2, 2019, pp. 102-106
18. Senhadji-Navarro, R. Garcia-Vargas, I., Mapping Arbitrary Logic Functions onto Carry Chains in FPGAs, Electronics 2022, pp. 11-27
19. S. P. Balwir, N. S. Panchbuddhe, K. R. Katole, Comparative Analysis of Fast Adder Circuit, International Journal of Advanced Research in Computer and Communication Engineering Vol. 10, Issue 6, June 2021, pp. 102-106
20. Woong Choi, Minseob Shim, Hyelin Seok, Yongtae Kim, DCPA: approximate adder design exploiting dual carry prediction, IEICE Electronics Express, Vol.18, No.23, 2021, pp. 1–4