The 8th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications
24-26 September 2015, Warsaw, Poland

Effectiveness of Matrix and Pipeline FPGA-Based Arithmetic Components of Safety-Related Systems

Julia Drozd [1], Oleksandr Drozd [1], Svetlana Antoshchuk [1], Alex Kushnerov [2], Valery Nikul [1]
[1] Odessa National Polytechnic University, Odessa, Ukraine
drozd@ukr.net, svetlana_onpu@mail.ru, leraniku@yandex.ru
[2] Department of Electrical Engineering, Tel-Aviv University, Israel
kushnero@post.tau.ac.il

Abstract—**The paper is devoted to design of the digital components for safety-related instrumentation and control systems using the modern CAD tools. Traditionally, the digital components are built with matrix parallelism that reduces fault tolerance of circuits and safety of systems in their checkability. Circuits with bitwise pipeline data processing have advantage in checkability, but are considered as less efficient. Matrix and pipeline methods for multiplying the numbers are compared with respect to the ratio of throughput to complexity. To estimate the relative effectiveness of these methods the resource approach is applied. The obtained estimations are verified experimentally using the multipliers realized on FPGA. Qualitatively, the expected and experimental estimations are consistent, and the reasons of their quantitative deviations are analyzed[1].**

Keywords—safety-related system; arithmetical digital component; resource approach; matrix and pipeline parallelism; FPGA project; complexity; throughput

I. Introduction

Modern tools for the FPGA design provide a designer with wide variety of options, for example CAD Altera Quartus II [1].

Presently, the digital components for critical applications are redesigned using the FPGAs, to ensure the functional safety of high-risk objects, for example nuclear power units [2–4].

The digital components in such a design have the traditional structure and are composed of simultaneous units with the matrix parallelism for processing the numbers in parallel codes. It is peculiar for the FPGAs to support the matrix parallelism extensively. However, the safety-related systems differ significantly from other computer systems by diversifying the operating mode, which is divided into a normal and emergency.

Analysis of the digital components with the matrix parallelism demonstrates low and different testability of the circuit in the normal and emergency modes. This causes the problem of hidden faults, which can be accumulated in the digital component during a long operation in the normal mode.

The hidden faults appear in the most critical emergency mode and reduce the fault tolerance of components and functional safety of the entire safety-related system [5].

To solve the problem of hidden faults the data processing in serial codes with a bitwise pipelining was proposed in [6].

It enhances and equalizes the testability in the normal and emergency modes. However, this method contradicts the tradition of building the high-performance digital units, which is based on the matrix of parallelism. To estimate the relative effectiveness of the matrix and bit-pipelined methods we apply the resource approach. The obtained estimations are verified experimentally using the FPGA multipliers. Qualitatively, the expected and experimental estimations are consistent, and the reasons of their quantitative deviations are analyzed.

How the use of matrix parallelism justified for improving the performance of units in the safety-related systems? This question can be answered using the resource approach [6–8]. It is based on analysis of development of the computer world and the world around (WA), which in turn, is considered as a process of coping with problems i.e. challenges due to the WA.

Solving the problem is composed of performing certain amount of work in limited time period, i.e. achieving some throughput, obtaining reliable results, and investing resources, which provide the above throughput and reliability. Resources comprise all the necessary for problem solving: models, methods and means. The models are human understandings of the WA and its components. The methods are description of resource transformation or estimation. The means comprise materials and tools.

All the resources are elements of the WA and therefore they develop from simple one to real by structuring under the WA features. The most evident features in the computer world are parallelism and fuzziness. Simple

[1] The authors would like to thank to the IEEE R8 Voluntary Contribution Fund for travel support that allowed presentation of this work at the Conference.

resources are sequential and exact. They appear in accordance with the initial understandings and abilities of a human. Real resources, characterizing the WA, are parallel and approximate. In the resource development occurring under the influence of parallelism and fuzziness, the resource approach identifies a number of levels: replication, diversification and autonomy. In the hierarchy they occupy the low, L, the middle, M and the high, H levels respectively. Each preceding level serves for the next one. In all the levels, from L to H, the goal is surviving. It is provided by different methods: increasing throughput, safety (reliability of results) and independence (self-sufficiency), respectively [7].

To reveal preferred directions for improving the methods of resource development, the resource approach proposes to arrange the levels of parallelism in ascending order and refer the matrix and pipelined computing to the levels L and M respectively. Note that the method of prepared results is referred to the H level [6, 9].

The multiple effect methods proposed in [8] improve simultaneously a number of important factors, without loss in all the rest, by increasing the level of resource development. These methods can be shown on example of the personal computer development. During last 20 years they have passed a way of improving the general parameters of clock rate (throughput) and memory size simultaneously increasing their from KHz up to GHz and from Mb up to Tb, accordingly [10]. This improvement is result of development of the hardware supporting the approximate data processing from co-processors of non-obligatory delivery (Intel 8087/287/387) to several floating-point pipelines in the CPU and many thousand floating-point pipelines in graphic processor with execution of the parallel calculations on technology CUDA [11].

This paper is aimed to estimate the effectiveness of arithmetic digital units designed for the safety-related systems with the modern CADs using the resource approach and analysis of experimental data. Peculiarities of the FPGA design in the CAD Altera Quartus II are considered in Section 2 from the point of view of the resource approach. The theoretical increase in the level of resource development in the FPGA projects was verified experimentally. Comparative analysis of the experimental and expected results is presented in Section 3.

II. Peculiarities of the FPGA Digital Design

Digital design using the CAD ALTERA Quartus II employs the parallelism of all the levels of resource development. FPGA chips allow prototyping the vast majority of projects. Being programmed for specific project, they comprise patterns of results in the Look-Up Tables (LUTs), canned networks of fast signal propagation for adder units and IP-cores.

The LUT-oriented architecture of circuits realized in the FPGA projects allows pipelining the calculations. This is because the FPGA logic element (LE) comprises an operational element (a function generator in the form of LUT) and a programmable register PR (flip-flop configured for D, T, JK, or SR operation), which can be connected serially representing a pipeline section. FPGA chip contains matrix of LE, matrix of ready iterative array multipliers, built-in circuits of fast carry propagation for accelerated addition of the numbers represented in parallel codes in the corresponding units with the matrix parallelism [12].

What role plays each kind of parallelism in effective design?

Mainly due to preparing the results, the FPGA projects feature a number of remarkable characteristics. Let us list them: computing performance and trustworthiness of results, design operativeness and manufacturability, versatility and flexibility of obtained circuits, as well as the most important advantage – combination of attainable levels, which evidences on their consistency [13]. All this characterizes preparing the results in respect to the FPGA design as a method of multiple effect.

Matrix parallelism is the fundamental in the FPGA design. Moreover, the concept of digital design with a matrix spatial parallelism for data processing in parallel codes is in fact imposed to the developer.

The built-in iterative array multipliers, libraries of various units with the matrix parallelism, scalable structures for adding numbers with fast carry propagation in the processing of parallel codes are proposed. Under these conditions, the pipeline parallelism is implemented at the macro level in the form of a pipeline system, sections of which are single-cycle concurrent matrix type units that operate with parallel code numbers.

How the low level L of matrix parallelism is expressed in the effectiveness of the FPGA projects?

An illustrative example is the execution of both codeword addition as the basic arithmetic operation and multiplication as the key operation for processing of approximate data in the floating-point formats [14, 15]. The latter follows from the fact that multiplication is naturally used in the number representation in the floating-point formats. The prevalence of these formats can be explained by structuring the resources under the features of approximated WA [16, 17].

The matrix parallelism limits the possibilities of parallel computing by dependence on two factors. One is incomplete data when the operands to execute the operation are not available yet. The other factor is unevaluated conditions according to which the algorithm is branched. So, the matrix parallelism can be effectively applied only when the above limiting dependencies are absent. On the contrary, it is widely used with multiple data dependencies, namely when the numbers are represented in parallel codes [18, 19]. All their bits are available simultaneously to execute an operation. This however, does not provide simultaneous processing since calculating the most significant bit usually requires the least significant bit to be known.

The elements of single-cycle circuit with the matrix parallelism are characterized by the utilization coefficient $K_U = T_E / T_T$ or the downtime coefficient $K_D = 1 - K_U$, where T_E is the delay of single element, and T_T is the clock duration.

Arithmetic n-bit adder of parallel codes is composed of n connected serially (by carry) full adders and operates sequentially, i.e. adds the numbers bit-by-bit [20]. The full adders are used one after another with the utilization factor $K_U = 1 / n$, i.e. a little part of the operation time.

The fastest version of the iterative array multiplier executes the addition for $2n - 2$ delays of the full adder. That is $2n - 2$ full adders are connected in series [21] and each of nearly n^2 full adders of the multiplier has the utilization factor $K_U = 1 / (2n - 2)$.

For example, if $n = 32$ then in the case of addition $K_U = 3.1\%$ and $K_D = 96.9\%$. In the case of multiplication $K_U = 1.6\%$ and $K_D = 98.4\%$ for 10^3 of the full adders.

With transition to 64-bit platform K_U deteriorates twice and becomes less than 1%.

In this case, the downtime is accompanied by increase in the energy consumption up to 30% [22].

To increase the level of parallelism it is advisable to use the bit-pipelined algorithms and minimize the matrix parallelism of pipelined sections, bringing it to single operational element, which processes the numbers bit-by-bit. Such a parallelization in serial codes can be implemented in the form of a multi-threaded processing shown in Fig. 1 for case of k pipelines with LUT-oriented architecture.

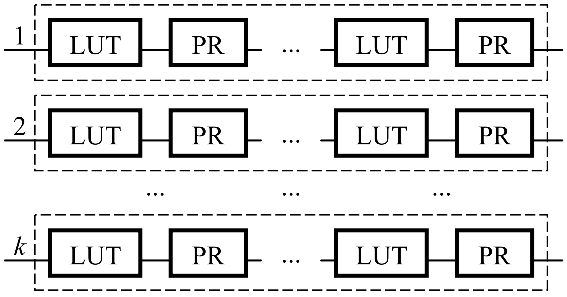

Figure 1. Multi-threaded system of k bit-by-bit pipelines with LUT-oriented architecture.

In this situation the matrix parallelism is used at macro level of the system and is represented in the best form, as a set of independent, simultaneously operating pipelines, i.e. there is no dependency on data.

The duty cycle of the elements in a bitwise pipeline is defined by the residual spatial matrix parallelism i.e. by the presence of two units in the pipeline section: the operational element and register, as well as by their serial connection.

The pipeline cycle duration $T_T = T_E + T_R$ allows us to define the utilization coefficient as $K_U = T_E / (T_E + T_R)$ or in the form of

$$K_U = K_{ER} / (1 + K_{ER}), \tag{1}$$

where $K_{ER} = T_E / T_R$, and T_R is the register delay.

The downtime coefficient is calculated as $K_D = 1 / (1 + K_{ER})$. It is evident that if $K_{ER} = 1$, i.e. $T_E = T_R$, then $K_U = K_D = 50\%$. In case $K_{ER} > 1$, i.e. $T_E > T_R$, $K_U > 50\%$, and $K_D < 50\%$.

The coefficient K_{ER} is used in [6] to compare the iterative array and bit-pipelined multipliers with respect to the time required to execute the operation, according to

$$k_T = 2n\,(1 + K_{ER}) / (2n - 2 + K_{ER}), \tag{2}$$

where $k_T = T_P / T_M$, T_P and T_M are the time needed to execute the operation in the iterative array and bit-pipelined multipliers.

The latter ratio can be written as

$$k_T = z + a, \tag{3}$$

where $a = (2z - z^2 + 2n) / (2n - 2 + z)$.

Note that $a = 1$ and $k_T = 3$ for $z = 2$ and $0 < a < 1$ for $z > 2$.

The complexity of the iterative array and bit-pipelined multipliers is estimated by a quadratic and linear dependence of equipment cost on the operand size n.

Namely, $H_M = 2(n^2 - n)\,q$ and $H_P = 6n\,q$ (including the complexity $n\,q$ of the control register, which can be used for a number of pipeline multipliers), where q is a full adder complexity and equated it flip-flop complexity.

The expected superiority of the pipeline approach over the matrix one is also estimated by their complexity H_M and H_P as $k_H = H_M / H_P$ and by their effectiveness as $k_E = k_H / k_T$ (for $k_T = 3$). For different operand size n these coefficients are given Table I.

TABLE I. EXPECTED ESTIMATIONS TO COMPARE THE ITERATIVE ARRAY AND BIT-PIPELINED MULTIPLIERS

n	16	24	32	40
k_H	5.0	7.7	10.3	13.0
k_E	1.7	2.6	3.4	4.3

Thus, increasing the level of parallelism from the matrix to the pipelined one, may not only to solve the problem of hidden faults, but also increase the throughput with simplification of the circuitry, which refers it to the multiple effect methods. How this confirmed in practice in case of the FPGA design?

III. COMPARISON OF THE ITERATIVE ARRAY AND BIT-PIPELINED FPGA MULTIPLIERS

A comparative analysis was carried out for the library iterative array multiplier and the bit-pipelined multiplier whose circuit is given in [23].

These units were designed in the CAD Altera Quartus II v.13 and implemented for the operand size n = 16, 24, 32 and 40 in EP2C35F672C6 Altera Cyclone II FPGA chip. The experimental results, namely the timing parameters and the complexity of the obtained circuit are given in Table II.

TABLE II. EXPERIMENTAL RESULTS

n	16	24	32	40
T_P	175.9	290.9	413.2	539.2
H_P	78	123	158	200
T_M	32.5	43.5	54.2	64.5
H_M	344	714	1216	1850
k_T	5.41	6.69	7.62	8.36
k_H	4.41	5.80	7.70	9.25
k_E	0.81	0.87	0.99	1.11

Diagrams of the relative k_E, expected, W, and experimental, E effectiveness are shown in Fig. 2 for the same multiplier operand size n.

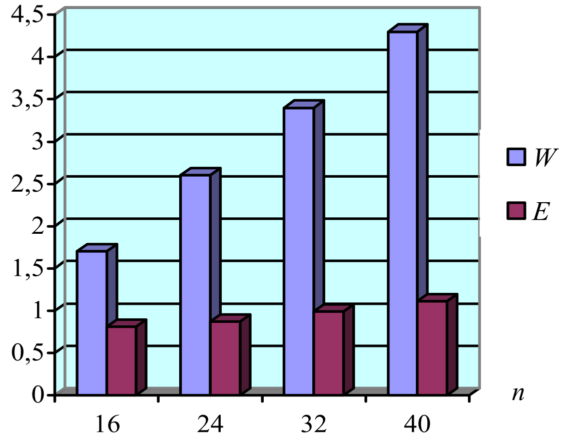

Figure 2. Diagrams of the expected and experimental relative effectiveness W and E for the pipeline and matrix methods.

As evident from Fig. 2, the relative effectiveness k_E of the pipeline methods is directly proportional to n for both cases W and E. Moreover, $E > 1$ for $n > 32$ and the pipeline methods becomes more efficient than the matrix one. Thus, the expected effect is confirmed qualitatively. However, the experimental estimation many times loses to the expected one. The loss values of time T, complexity H and effectiveness E are summarized in Table III.

TABLE III. MISMATCH BETWEEN THE EXPECTED AND THE EXPERIMENTAL RESULTS

n	16	24	32	40
T	1.8	2.3	2.5	2.8
H	1.1	1.3	1.3	1.4
E	2.0	3.0	3.3	3.9

Due to the mismatch between the expected and experimental estimations of the time required to the iterative array multiplier to execute the operation the loss value of time T is dominant. For the considered operand size n, the designed multiplier has appeared to be faster in 1.8 – 2.8 times.

This effect can be explained by that the FPGA design of the iterative array multiplier is based on the method of result preparing, which pertains to the level H of the resource development. The method provides a fast propagation of the carry and sum in the iterative array multiplier owing to the method of conditional carry and canned circuits of carry propagation between the adjacent LE.

To make the experimental estimations close to expected ones it is expedient to use the method of result preparing not only for improving the design with the level L of matrix parallelism, but also for improving the effectiveness of advanced pipelining, related to the level M.

IV. CONCLUSIONS

The experiments have confirmed the viability of increasing the level of parallelism from the matrix to the pipelined one in the design of arithmetic units for the safety-related systems.

The obtained estimations demonstrate that as compared to the matrix method the relative effectiveness of the pipeline method increases with increase in the size of data processed.

The experimental estimation has exceeded the unity thereby showing that the pipelining is advantageous in practice, when the effectiveness is defined as the ratio of throughput to complexity. The increase and equalizing the testability of digital components in the normal and emergency modes are of high importance for the safety-related systems since they allow solving the problem of hidden faults.

All this characterizes the multithreaded bit-pipelined processing as a multiple effect method, which makes the obtained circuit better simultaneously by the throughput, complexity and testability without loss in other factors.

The fact that the experimental estimations are less than the expected ones evidences on a significant role of the result preparing method in efficient design of the circuits for processing the numbers in parallel codes with the matrix parallelism. In the experiments with the iterative array multiplier it is expressed in faster calculations.

To improve the agreement between the experimental and expected estimations further research is required. It can be accomplished by better adaptation of the pipeline approach to the peculiarities of the existing FPGA architectures.

In the design of safe digital components a significant increase in the level of parallelism can be obtained by advanced bit-pipelined processing, which in turn, requires advanced CAD tools with massive result preparing.

REFERENCES

[1] *Netlist Optimizations and Physical Synthesis*, Qii52007-2.0. Quartus II Handbook, vol. 2, Altera Corporation, 2004.

[2] E. S. Bachmach, A. D. Herasimenko, V. A. Golovir, V. S. Kharchenko, Yu. V. Rozen, A. A. Siora, V. V. Sklyar, V. I. Tokarev, *FPGA-based NPP I&C Systems: Development and Safety Assessment*, V. S. Kharchenko, V. V. Sklyar, (eds.), RPC Radiy, National Aerospace University "KhAI", SSTC on Nuclear and Radiation Safety, 2008, 188 p.

[3] A. Andrashov, V. Kharchenko, A. Siora, V. Sklyar, A. Volkoviy, "Certification of FPGA-based safety Instrumentation and Control platform in accordance with IEC 61508," in *Proceedings of the First International Workshop on Critical Infrastructure Safety and*

Security (CrISS-DESSERT'2011), Kirovograd–Kharkov, Ukraine, 11–13 May 2011, vol. 1, pp. 148-152.

[4] E. Bakhmach, V. Kharchenko, A. Siora, V. Sklyar, V. Tokarev, "Design and qualification of I&C systems on the basis of FPGA technologies," in *Proceedings of the 7th International Topical Meeting on Nuclear Plant Instrumentation, Control, and Human-Machine Interface Technologies (NPIC&HMIT 2010)*, Las Vegas, Nevada, 7-11 November 2010, pp. 916-924.

[5] M. Drozd, A. Drozd, "Safety-related instrumentation and control systems and a problem of the hidden faults," in *Proceedings of the 10th International Conference on Digital Technologies*, Zhilina, Slovak Republic, 9-14 July 2014, pp. 137-140.

[6] V. S. Kharchenko (eds.), *Green IT-Engineering, vol. 1, Principles, Models, Components*, National Aerospace University "KhAI", Kharkiv, Ukraine, 2014, 594 p.

[7] J. Drozd, A. Drozd, D. Maevsky, L. Shapa, "The levels of target resources development in computer systems," in *Proceedings of the IEEE East-West Design & Test Symposium*, Kiev, Ukraine, 26-28 September 2014, pp. 185-189.

[8] J. Drozd, A. Drozd, S. Antoshchuk, V. Kharchenko, "Natural development of the resources in design and testing of the computer systems and their components," in *Proceedings of the 7th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*, Berlin, Germany, 12-14 September 2013, pp. 233-237.

[9] A. V. Drozd, M. V. Lobachev, J. V. Drozd, *Dedicated Architectures of Computers, Learning aid*, Odessa: Science and Technique, 2004, 120 p.

[10] M. Guk, *Processors Intel: from 8086 to Pentium II*, Saint Petersburg: Piter, 1997, 224 p.

[11] *NVIDIA CUDA Compute Unified Device Architecture, Programming Guide / Version 1.0*, NVIDIA Corporation, 2007, 113 p.

[12] *Cyclone FPGA Family Data Sheet*, Altera Corporation, 2003, http://www.altera.com.

[13] *Design Optimization for Altera Devices. Qii52005-2.0. Quartus II Handbook*, vol. 2, Altera Corporation, 2004.

[14] W. Kahan, "IEEE standard 754 for binary floating-point arithmetic," *Lecture Notes on the Status of IEEE 754*, Elect. Eng. & Computer Science University of California, Berkeley, May 1996.

[15] D. Goldberg, "What every computer scientist should know about floating-point arithmetic," *ACM Computer Surveys*, vol. 23, no 1, 1991, pp. 5-18.

[16] M. Guk, *Hardware of IBM PC: Encyclopaedia*, 2nd ed., Saint Petersburg: Piter, 2003, 928 p.

[17] G. Kondratenko, Y. Kondratenko, D. Romanov, "Fuzzy models for capacitative vehicle routing problems in uncertainty," in *Proceedings of the International DAAAM Symposium*, 2006, pp. 205-206.

[18] S. Fernbach, (eds.), *SuperComputer. Hardware and Software organization*, Moscow: Padio and Communication, 1991, 320 p.

[19] A. Tanenbaum, *Structured Computer Organization*, 4th ed., Upper Saddle River, NJ: Prentice Hall, 1999, 698 p.

[20] J. Koren, *Computer Arithmetic Algorithms*, Prentice Hall, Englewood Cliffs, New Jersey, 1993, 224 p.

[21] A. O. Melnyk, *Architecture of Computer*, Volyn Region Printing House, Lutsk, Ukraine, 2008, 470 p.

[22] A. P. Chandracasan, R. Sheng, S. Brodersen, "Low-power CMOS digital design," *IEEE Journal of Solid-State Circuits*, vol. 27, issue 4, 1992, pp. 473-484.

[23] A. V. Drozd, E. L. Polin, V. N. Oginsky, E. A. Godis, J. V. Drozd, "Device for multiplication of binary numbers," Patent of USSR, SU 1587498, G 06 F 7/52, Bulletin no. 31, 1990.