

Delay-Insensitive Floating Point Multiply-Add-Subtract Unit

I.A. Sokolov, Y.V. Rogdestvenski¹, Y.G. Diachenko², Y.A. Stepchenkov³, N.V. Morozov⁴,
D.Y. Stepchenkov⁵, D.Y. Diachenko

Institute of Informatics Problems, Federal Research Center "Computer Science and Control" of the
Russian Academy of Sciences (IPI FRC CSC RAS), IPI RAS

{¹YRogdest, ²YDiachenko, ³YStepchenkov, ⁴NMorozov, ⁵DStepchenkov}@ipiran.ru

Abstract — The subject of this paper is a floating point unit implementing fused multiply-add-subtract operation. It belongs to the delay-insensitive self-timed (ST) circuits which do not depend on delays both in cells and on wires. It is fully compliant with IEEE 754 Standard and processes both a sum and difference between product of first two operands and third operand. Each 64-bit input operand contains either one double precision number, or two single precision numbers. Thus presented unit calculates either one operation with double precision numbers, or two simultaneous operations with single precision numbers. Multiplier utilizes modified Booth algorithm. In order to increase its performance, it is divided into two pipeline stages with accelerated forced switching to spacer phase. Booth encoder circuit is integrated into an input FIFO. FIFO is implemented as a register file with an output multiplexer and read/write address counters. Using ternary redundant ST code for multiplying, adding and subtracting provides reduction of unit's complexity. Indication subcircuit considers the constrains imposed by an isochronous area for chosen fabrication technology. For decreasing energy consumption, the fused multiply-add-subtract unit implements one-channel pipeline. The unit is designed for 65-nm CMOS bulk technology using an industrial standard cell library supplemented by self-timed cells. It provides 3 Gflops performance and 2.9-ns latency.

Keywords — delay-insensitive, redundant coding, ternary adder, Wallace tree, isochronous area, FIFO.

I. INTRODUCTION

Hardware implementing two input operands multiplication followed by an add-subtract with third input operand in the same unit can be designed using either intermediate product rounding, or a single round of an overall result. The former version was used in first generations of the digital signal processors. The version with a single rounding the so-called fused multiply-add (FMA) has become de facto standard operation of the modern general purpose processors because it provides higher accuracy than the intermediate rounding.

Currently known implementations of this unit are overwhelmingly synchronous [1]-[3]. Asynchronous solutions, claiming to belong to the class of ST circuits [4]-[5], do not fully indicate the end of switching in all elements of the circuit before transition into the next phase of work. Therefore they cannot be considered as the unit, which proper function does not depend on delays in the cells and wires, i.e. Delay-Insensitive (DI), under all conditions of operation.

Preserving the workability of the DI circuits at midget supply voltages opens up broad prospects for their usage in the portable battery-powered products, as well as for constructing onboard computer complexes, not exacting to the level and stability of power resources. Stable operation under extreme conditions is achieved at the expense of hardware redundancy, additional delay for indication, and presence of a spacer phase in the DI circuits. However, proper design of the DI circuits can substantially reduce this redundancy and additional delay, and in some cases, such as fault-tolerant units [6], obtain the results better than for synchronous analogs.

Earlier authors had already attempted to develop speed-independent FMA unit with gigaflops performance SIFMA [7]-[8] and SIFPC [9]-[10]. However, to achieve the maximum performance, SIFMA had utilized a principle of speculative indication that does not provide its absolute self-check ability, while SIFPC was built as two-channel unit with two-stage pipeline having common inputs and outputs, and adaptive indication not considering real size of an isochronous area [11].

This article presents the results of designing 64-bit DI computing device performing the floating-point multiply-add and multiply-subtract operations in accordance with IEEE 754, i.e. DI Fused Multiply-Add-Subtract (DIFMAS) unit. The complexity and timing characteristics of its implementation are discussed in the sections II and III. The synchronous FMA unit implementation including redundant presentation of the multiplicands [12] was chosen as the prototype of a mathematical computation model because it provides the best performance in the ST circuitry basis. Methodological aspects of designing ST FMA unit have been discussed in detail in [7].

II. DIFMAS FEATURES

As in previous designs, each of three processed operands contains either one double or two single-precision digits. In the latter case it performs two independent operations, FMA and FMS, over two single-precision operand triplets.

A. Block diagram of DIFMAS

The trend in the modern computational tools development consists in ensuring minimum energy consumption at a high enough performance. This is determined by the tendency to use relatively low clock frequency and a large number of the computing nodes on a single VLSI for high-performance computers.

There are two phases in any DI circuit: an active – work, and pause – spacer. This suggests the idea of using two parallel channels, which phases are alternated. Such implementation was presented in [9]-[10]. It had allowed for achieving average 3.15 Gflops performance with synchronous environment and 3.90 Gflops in the absence of unproductive waiting response from a synchronous environment of successful reading result from FMA. However, this resulted in a big hardware costs and relatively high power consumption.

In this regard, a new block diagram DIFMAS was proposed for DI floating-point coprocessor calculating the sum (FMA) and difference (FMS) between a product of first two operands and third operand (fig.1).

In addition to the DIFMAS core, it contains input and output FIFO. They increase the performance of the DIFMAS interfacing with synchronous environment [7] due to the data stream buffering. Both FIFOs are implemented as a multi digit DI shift register on latch triggers with common control circuit, which ensures synchronous write to the FIFO, ST reading from FIFO and FIFO occupancy indicator formation.

Block diagram of the DIFMAS core is represented in fig. 2. Unlike previously developed FMA units [7]-[10], it does not contain any duplicate channels or multipliers. However, compared with previous implementations it has the best ratio "performance / hardware costs" due to innovative multiplier and wider use of redundant ST coding.

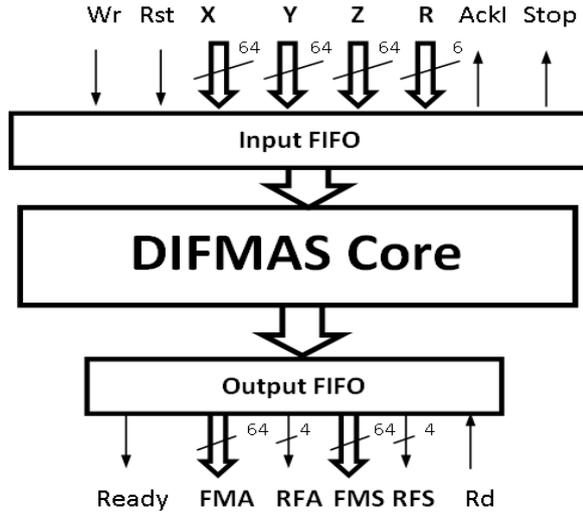


Fig. 1. Block diagram of the DIFMAS

B. DIFMAS multiplier

Multiplier is the most hardware-intensive (55-60% of the total hardware complexity), energy-consuming and time-elapsing unit among all functional blocks in the DIFMAS. This design implements the base case multiplier used in the vast majority of modern computing devices. This is purely combinational circuit, consisting of Booth decoder, Booth encoder (Radix-4) and Wallace tree on base of the redundant binary representation adders [12]-[13].

The ST FMA units described in [7]-[10] include two multipliers for achieving desired performance. They utilize

a dual-rail code as self-timed code for the Booth encoder and redundant (ternary) code for Wallace tree. Both codes have diphas coding discipline consisting of work and

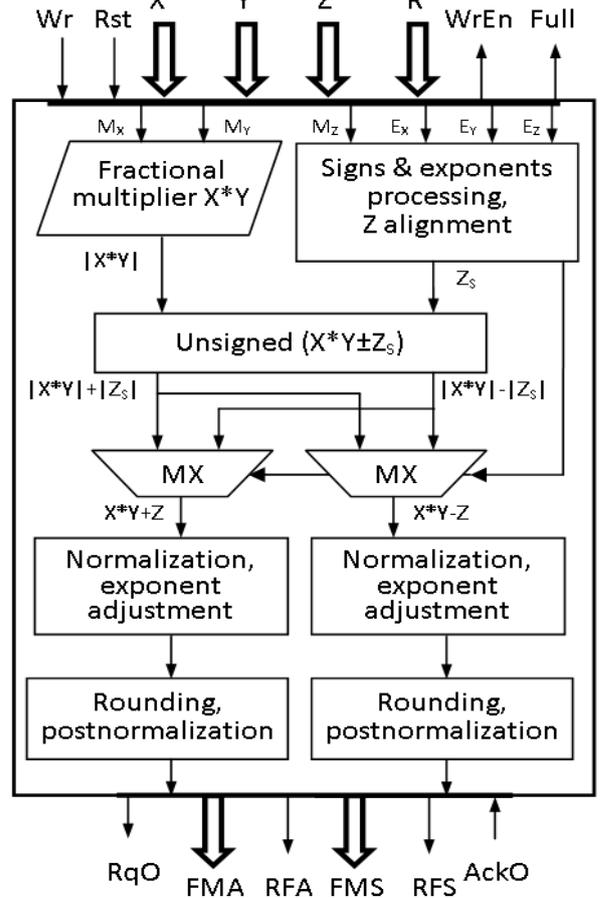


Fig. 2. Block diagram of the DIFMAS's core

spacer phases. Optimization of ST interaction discipline between the stages of the FMA pipeline has ensured calculation latency according to the expression:

$$T_L = N * T_w + T_s.$$

Here T_L is a latent time of the FMA pipeline; T_w is the work phase duration of the corresponding pipeline stage; T_s is spacer duration of the pipeline stage; N is the number of the pipeline stages.

However, the work cycle duration of one pipeline stage remained still equal to the total duration of two phases and this determined the performance of the ST FMA. When implementing a multiplier as one stage of the pipeline in 65-nm CMOS technology, the average duration of the work cycle becomes close to 1.5 ns without considering layout realization. The usage of two parallel multipliers in SIFMA and two parallel channels calculating triples of the operands in SIFPC has allowed for considerable accelerating ST FMAs at the expense of increased hardware complexity.

A simple splitting multiplier onto two pipeline stages allows for reducing each stage cycle duration down to 1 ns without considering layout realization. This is clearly insufficient. A further increase of the number of the pipeline

stages in the multiplier will be extremely expensive and will not lead to a significant reduction of the work cycle duration of the pipeline stages. The main reasons for this are significantly delayed indication of the switch completion of all elements at each pipeline stage due to the high bit depth of the processed operands. Additional complexity is also introduced due to organization of the self-timed registers storing intermediate results.

In order to indicate only one output register of the single pipeline stage of the multiplier, the five-layered "tree" of 3-input hysteresis triggers (H-triggers, [11]) is needed. So total delay of the indication signal formation within two phases for 106 output bits occupies 300-350 ps in 65-nm CMOS process. Indication of the registers introduces additional 100-150 ps. As a result, the total duration of work and spacer phases approaches 1 ns, while the maximum duration of the pipeline stage work cycle should not exceed 800-850 ps without considering layout realization to ensure gigaflops performance range.

The developed release of the DI multiplier solves this problem by means of a number of special techniques:

- Forcing acceleration of the spacer phase of the ST cycle;
- Introducing additional registers for storing intermediate data in second stage of the Wallace tree;
- Refusing total output indicator formation in one-bit adder of the Wallace tree and instead of this using parallel indication of the internal signals with a subsequent timing optimization;
- Using faster ternary adders in the Wallace tree.

Fig. 3 represents one-bit ternary adder circuit used in the Wallace tree of the DIFMAS. This adder is more complex (by 18%) than ones in the preceding ST FMAs [8]. But it has significantly better performance (by 27%) in the DIFMAS pipeline.

Here FS input provides accelerated switching of one-bit adder and entire Wallace tree into spacer state. Three indication outputs (Ind1-Ind3) are combined into a single indication output of the whole Wallace tree by a distributed indication subcircuit taking into account the relations between delays of all indication signals in the different layers and stages of the Wallace tree. An implementation of this approach has allowed for further improving multiplier performance by 12%.

Multiplier developed on base of suggested methods in the DIFMAS is implemented as a two-stage pipeline and provides the total duration of the work and spacer phases at level of 850-900 ps without considering layout realization. This is by 1.7 times less than for an original FMA variant that allows for obtaining DIFMAS of gigaflops range using single multiplier.

The additional hardware costs providing this multiplier performance improvement equal to 12-13%. However, as a result, the multiplier implementation in the DIFMAS has less total complexity by factor of 40-45% versus SIFPC and SIFMA due to the usage of the single multiplier instead of doubled multiplier. This also has caused a propor-

tional decrease of both an energy consumption of the multiply operation, and layout area of the multiplier in the DIFMAS.

The following principles of formation of the dual-rail

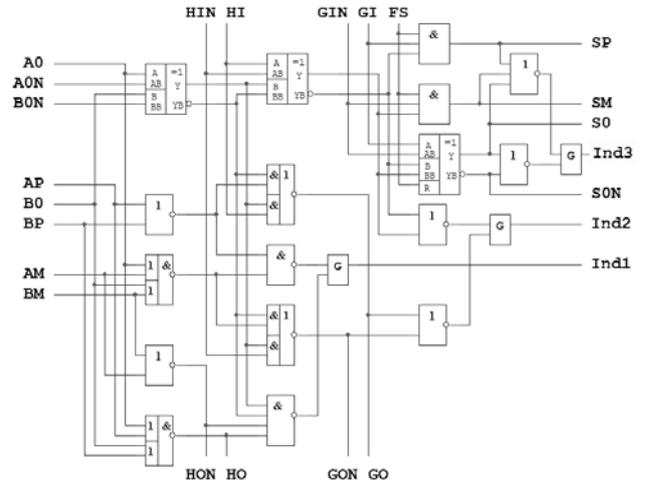


Fig. 3. Ternary one-bit DI adder

inputs of a combinational circuit (pipeline stage) were used in the multiplier:

1. Stage inputs can switch to the work phase (spacer), if this stage has completed switching to the spacer (work phase), and its previous and subsequent stages in the pipeline have allowed for its switching to the work phase (spacer). An input permitting such switch to the opposite phase is generated by the indicators of the current, previous and next pipeline stages.

2. The input permitting switching to the opposite phase of work is similar to the local clock. It is also implemented as "clock tree".

3. Stage inputs are driven by the output register of the preceding pipeline stage. One bit of this register consists of two H-triggers and an indication cell NOR2 or NAND2 depending on a type of the input spacer. Such register stores both dual-rail work state, and spacer state of its inputs.

Output DI FIFO serves as an output register of the DIFMAS. It has dual-rail information inputs, write enable input and bi-phase output providing an interface with synchronous environment.

C. Input and output FIFOs

Input and output FIFOs used in the previous FMA implementations [7]-[10] had one major drawback, namely the relatively high energy consumption. This is due to their circuits based on ST semi dense shift register [11, fig. 11.9]. Data word written to the input register head automatically moves toward an output register head down to the nearest free cell.

On the one hand, this provides the same read data sequence as the written data one without additional hardware costs for a mechanism addressing current output FIFO cell. On the other hand, towards the output head of the FIFO data word forcedly pass through all intermediate FIFO

cells, and causes recharging their parasitic capacitances resulting in additional energy consumption.

Simplicity of the semi-dense register implementation and behavior is accompanied by a substantial increase in the number of its intermediate signals and rather a large complexity, because only each second stage of FIFO stores real data as FIFO is "semi-dense".

So a new FIFO implementation was developed and used in SIFMA. It is founded on a shift register (SR) file with parallel writing and reading. Fig. 4 shows its structure. Input data come to an input head ("IH") that distributes them to the next stage of SR, and W/R_i signals from "Control Unit" allow for writing data "DIn" into regular SR stage. Control Unit detects a state of each stage of the SR and allows for shifting data from the input head of the SR to its output head. It also forms "Stop" signal that notifies data source about fullness of the FIFO, and advises data receiver of data "DOut" readiness. The Control Unit is common for all bits of the FIFO. So its complexity slightly affects total complexity of the FIFO.

Fig. 5 demonstrates an input head of the SR. Such input head implementation improves FIFO interface with synchronous environment.

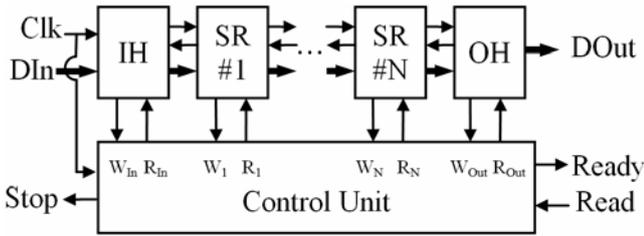


Fig. 4. Flow-chart of the proposed FIFO

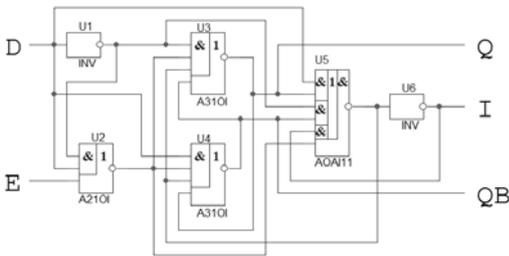


Fig. 5. Input head of the FIFO SR

Table I shows the pin amount and complexity for both variants of one FIFO bit designed for storing 4 operands. The proposed new SR version has by 2.3 times less pins than the semi-dense register. Moreover, proposed variant has by 1.8 time lower complexity.

The input head of the release #1 assumes the usage of classic SI exchange protocol between the FIFO and data source. After writing data into FIFO and forming indication signal reflecting a successful end of the work phase, an input must remain unchanged until an opposite switching indicator signal caused by transition of the write permission signal. However, if the source of the information does not meet the requirements of the SI protocol (for example, implements a synchronous communication proto-

col), FIFO information may be rewritten by an incorrect data. The input head of the release #2 provides SI work mode within both the synchronous, and SI protocol.

Thus, the release #2 of FIFO implementation ensures its reliable operation with significantly lower hardware cost than release #1. This contributes to improve its energy efficiency.

Table 1

Features of two FIFO releases

Release	Pin number	Complexity (CMOS transistors)
Semi-dense FIFO (release #1)		
Input head	11	56
Intermediate cell	13×5	32×5
Output head	12	42
Total	88	258
Proposed FIFO (release #2)		
Input head	5	38
Intermediate cell	11×2	34×2
Output head	11	34
Total	38	140

D. DIFMAS Indication

DIFMAS is the unit processing multiple-bit data and occupying a large area on a VLSI chip. It implements an optimal indication principles on a local level used in the earlier version of FMA (SIFPC) [10]. However, at the level of large functional blocks the DIFMAS indication is implemented as a minimal required one from the viewpoint of the self-timed circuit indication principles. It does not take into account the limitations caused by the isochronous area size [11].

Isochronous area refers to the circuit fragments, whose components operate in "one time" [11, p. 10], and the difference in signal delay in the interconnections after branching does not exceed the minimum switching delay of an arbitrary cell of the standard cell library used for this VLSI implementation.

Fig. 6 shows an example of branching signal A driven by cell U0 for the case of three fan-outs. Signal propagation delay from U0 output to the inputs of U1, U2 and U3 cells (t_{pd1} , t_{pd2} and t_{pd3} respectively) are determined by the chip technology and its layout, namely the lengths of the signal A branches connecting to the elements U1, U2, U3, and their physical implementation (in which rout layers they are released). If the following relations are true simultaneously:

$$\begin{cases} |t_{pd1} - t_{pd2}| < t_{min,pd}, \\ |t_{pd2} - t_{pd3}| < t_{min,pd}, \\ |t_{pd1} - t_{pd3}| < t_{min,pd}, \end{cases} \quad (1)$$

where $t_{min,pd}$ is the minimum switching delay of any element among U1, U2, U3 relative to the corresponding input, then the circuit in fig. 6 is considered to be located entirely in isochronous area. So one may indicate the signal A at the input of any cell from U1-U3. Otherwise, the sig-

nal A should be indicated at the end of the branch with maximum delay.

As the actual delay of the output signal of the $U0$ cell after the branching point A is determined by the specific layout implementation (mutual arrangement of the elements connected to one circuit, and used routing layers), then accurate analysis of inequalities (1) firstly is possible only after

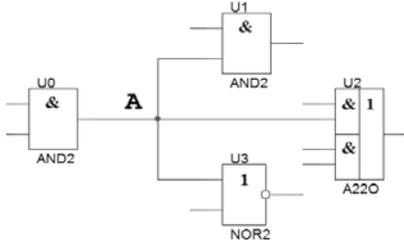


Fig. 6. An example of a signal fan-out for the case of three receivers

layout implementation of the circuit and, secondly, is necessary after each correction of the circuit layout.

Analysis of influence of parasitic capacitances and resistances in the standard 65-nm CMOS technology shows that using second and third metal layers for routing in the worst case leads to parasitic capacitances with linear value of 202 fF/mm. While routing the same trace in fourth and fifth metal layers results in parasitic capacitance of 198 fF/mm. Given that the typical input capacity of the standard library cell does not exceed 1.5 fF, it turns out that the rout interconnections make the main contribution to the signal propagation delay in a chip manufactured on 65-nm CMOS technology.

The simulation results for the circuit shown in fig. 6, taking into account the parasitic parameters extracted from its layout, shows the following:

1. For cell with single output capability, difference between signal propagation delays for traces with lengths differing by 60 μm is about 5 ps. It corresponds to the switch delay of the single inverter for this technology ($t_{\text{min,pd}} = 5$ ps). At that the difference between the delays does not depend on both the type of cell and the complexity of its function.

2. For the cells with high output capability, difference of signal propagation delays on the same traces is even more than for the cells with single output capability (for example, for an inverter with 40 times output capability the delays themselves are reduced, but the difference between them turns out to be the order of 6 ps).

Thus, the size of the isochronous area for CMOS technology with 65-nm design rules does not exceed 60 μm for the cells with single output capability and 50 μm for the cells with great output capability. It should, however, be noted that the size of the isochronous area is associated with a radius of the circle the center of which is located at the point of the trace branching.

The size of the isochronous area is a quite conditional concept. A cell-receiver may be located close enough from

the cell-driver, but the trace to it after the branching point may "stray" on layout. Due to this its length and corresponding parasitic capacitance determining signal delay will be relatively large. It is therefore advisable to speak about isochronous traces, keeping in mind their length (and signal propagation delay over them) after branching points.

An analysis of the net delays in 64-bit divider and square rooter unit designed in 65-nm CMOS process [6] taking into account extracted parasitic capacitances and resistors has shown that a number of the nets, which delay exceeds the minimum cell delay, equals to 7.5%. Almost 30% of them are free of branches. The differences between delays of the branch pieces of about 50-60% of the rest nets do not exceed the minimum cell delay. As a result, only 2% of all nets need in checking they do not cause a break of circuit delay-insensitivity, and re-designing their layout realization if necessary.

For ensuring DI property, the following conditions should be met:

1. Dual-rail signals should be indicated at the input of their receiver, at the end of the longest (delayed) trace.
2. If there are traces beyond a subset of the isochronous traces for a given signal, the signals at the end of all these traces should also be indicated.

These conditions are validated at checking designed circuit on self-timed ability using proper analysis program [14] taking into account the real parasitic parameters extracted from the layout of the analyzed circuit.

III. DIFMAS PARAMETERS

DIFMAS was designed in the standard 65-nm bulk CMOS technology with six metallization layers. DIFMAS parameters are shown in the Table II in comparison with synchronous analogue with closest performance [15]. Timings and energy parameters were obtained on a simulation base without parasitic parameters extracted from a layout for statistically reliable set of the input operand combinations for double and single precision cases.

Performance was determined for typical operating conditions (1.0 V supply, 25°C), because the performance of DI circuits always corresponds to the current conditions. DI circuits do not require taking into account worst case for ensuring workability of the circuit across the guaranteed range of the supply voltage and ambient temperature.

Table 2

DIFMAS Parameters

Name of parameter	Analogue	DIFMAS
Work frequency (GHz)	1.03	1.02
Layout die, mm ²	0.312	0.468
Latency, ns	10.8	2.94
Performance, Gflops	2.06	3.06
Area effectiveness, mm ² /Gflops	0.151	0.153
Range of supply voltage V_{DD}	$V_{DD} \pm 10\%$	$V_{th} \dots V_{BD}$
Detection of the constant malfunctions	-	+

It should be noted that DIFMAS has greater functionality when compared with analogue. For one cycle, it is able to process three double precision operands, or two single-precision operand triplets, calculating at that both an amount and the difference of first two operands production and third operand (this is reflected in the performance parameter). In addition, it has a much wider range of the performance limited only by the threshold voltage of the CMOS transistors (V_{th}) and breakdown voltage of the semiconductor structures (V_{BD}), and terminates at detecting constant malfunctions [11]. Payment for these benefits is greater complexity and, therefore, more power consumption. Energy consumption can be reduced to the desired value by reducing the supply voltage, but this leads to the corresponding decrease in performance. Due to fewer pipeline stages DIFMAS latency is by 3.7 times less than synchronous counterpart.

Therefore, the presented DIFMAS release provides performance at a level of 3.06 Gflops. It reflects a modern trend in designing high-performance computing hardware: usage of more processors with relatively low performance.

IV. CONCLUSION

DIFMAS with a single multiplier designed for CMOS technology with 65-nm design rules demonstrates the high average performance (3.06 Gflops at typical conditions) and latency (less than 3 ns).

The usage of the redundant ST coding, two-stage multiplier, and acceleration of switching multiplier to spacer has provided the development of 64-bit unit, which implements the FMA and FMS operation, corresponds to the modern synchronous analogs on performance and has all advantages of the DI units: pure self-check ability in relation to constant failures, preserving the workability at the midget supply voltage values.

Direction for further research is a study of the possibility of reducing the spacer phase duration for all stages of the DIFMAS pipeline due to development of the localized bitwise system accelerating the transition block cells to spacer state and work phase due to designing faster register for storing intermediate data.

SUPPORT

The study was done with partial support of fundamental researches of Presidium of the RAS, projects #0063-2018-0003 and #0063-2018-004 at the Institute of Informatics Problems, FRC CSC RAS.

REFERENCES

- [1] R.V.K. Pillai, S.Y.A. Shah, A.J. Al-Khalili, and D. Al-Khalili, Low power floating point MAFs – A comparative study // Sixth International Symposium on Signal Processing and its Applications, Kuala Lumpur, 2001, V. 1. P. 284-287.
- [2] P.-M. Seidel, Multiple path IEEE floating-point Fused Multiply-Add // Proc. 46th IEEE International Midwest Symposium on Circuits and Systems, Cairo, Egypt, 2003. P. 1359–1362.
- [3] T. M. Brintjes. Design of a Fused Multiply-Add Floating-Point and Integer Datapath // Master's thesis, University of Twente, Enschede, the Netherlands, 2011. 154 p.
- [4] J.R. Noche, and J.C. Araneta, An asynchronous IEEE floating-point arithmetic unit // Science Diliman, Philippines. 2007. V.19. No. 2. P. 12–22.
- [5] R. Manohar, and B.R. Sheikh, Operand-optimized asynchronous floating-point units and method of use thereof, US patent, № 20130124592. May 2013.
- [6] Stepchenkov Y., Diachenko Y., Zakharov V., Rogdestvenski Y., Morozov N., Stepchenkov D. Quasi-Delay-Insensitive Computing Device: Methodological Aspects and Practical Implementation // PATMOS'2009: Proceedings of the International Workshop on power and timing modeling, optimization and simulation. – Delft, The Netherlands, Springer 2010. P. 276–285.
- [7] Sokolov I.A., Stepchenkov Yu.A., Rozhdestvenskiy Yu.V., Diachenko Yu.G. Samosinhronnoe ustroystvo umnojeniya slojeniya gigaflopsnogo klassa: metodologicheskie aspekty (Speed-Independent Fused Multiply-Add Unit of Gigaflops Rating: Methodological Aspects) // Sb. trudov "Problemy razrabotki perspektivnykh mikro- i nanoelektronnykh sistem". M.: IPPM RAN, 2014. Ch. IV. S. 51-56 (in Russian).
- [8] Stepchenkov Yu.A., Rozhdestvenskiy Yu.V., Diachenko Yu.G., Morozov N.V., Stepchenkov D.Yu., Surkov A.V. Samosinhronnoe ustroystvo umnojeniya-slojeniya gigaflopsnogo klassa: varianty realizatsii (Speed-Independent Fused Multiply-Add Unit of Gigaflops Rating: Implementation Variants) // Sb. trudov "Problemy razrabotki perspektivnykh mikro- i nanoelektronnykh sistem". M.: IPPM RAN, 2014. Ch. IV. S. 57-60 (in Russian).
- [9] Yuri Stepchenkov, Victor Zakharov, Yuri Rogdestvenski, Yuri Diachenko, Nikolai Morozov and Dmitri Stepchenkov. Speed-Independent Fused Multiply Add and Subtract Unit // Proceedings of IEEE East-West Design & Test Symposium (EWDTS'2016), Yerevan, October, 14 - 17, 2016. P. 150-153.
- [10] Stepchenkov Yu.A., Rozhdestvenskiy Yu.V., Diachenko Yu.G., Morozov N.V., Stepchenkov D.Yu., Stepanov B.A., Diachenko D.Y., Rozhdestvenskene A.V. Samosinhronnoe ustroystvo umnojeniya-slojeniya s plavayushey tochkoj (Self-Timed Floating Point Multiply-Add Unit) // Sb. trudov "Problemy razrabotki perspektivnykh mikro- i nanoelektronnykh sistem". M.: IPPM RAN, 2016. Ch 3. S. 149-156 (in Russian).
- [11] Varshvskij V.I. i dr. Avtomatnoe upravlenie asinhronnyimi protsessami v EVM i diskretnykh sistemah (Automatic control of the asynchronous processes in the computers and discrete systems). M.: Nauka, 1986. 400 s. (in Russian).
- [12] H. Makino, Y. Nakase, H. Suzuki, H. Morinaka, H. Shinohara, and K. Mashiko, "An 8.8-ns 54x54-bit multiplier with high speed redundant binary architecture" // IEEE Journal of Solid-State Circuits. 1996. V. 31. No. 6, pp. 773-783.
- [13] Stepchenkov Y.A., Zakharov V.N., Rogdestvenski Y.V., Diachenko Y.G., Morozov N.V., Stepchenkov D.Y. Speed-Independent Floating Point Coprocessor // IEEE East-West Design and Test Symposium, Batumi, Georgia, September 26-29, 2015. P. 111- 114.
- [14] Rozhdestvenskiy Yu.V., Morozov N.V., Rozhdestvenskene A.V. Podsystema sobyitiynogo analiza samosinhronnykh shem ASPEKT (ASPECT – a Subsystem of Event Analysis of Self-Timed Circuits) // Sb. trudov "Problemy razrabotki perspektivnykh mikro- i nanoelektronnykh sistem". M., IPPM RAN, 2010. S. 26-31 (in Russian).
- [15] S. Galal, and M. Horowitz, Energy-Efficient Floating-Point Unit Design // IEEE Transactions on computers. 2011. V. 60. No. 7. P. 913–922.