Self-Timed Floating Point Multiply-Add Unit

Y.A. Stepchenkov, Y.V. Rogdestvenski, Y.G. Diachenko, N.V. Morozov, D.Y. Stepchenkov,
B.A. Stepanov, D.Y. Diachenko, A.V. Rogdestvenskene

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

{YStepchenkov, YRogdest, YDiachenko, NMorozov, DStepchenkov}@ipiran.ru

***Abstract* — The subject of this paper is a Speed-Independent Floating Point Coprocessor (SIFPC) implementing Fused Multiply-Add-Subtract operation. It utilizes mixed dual-rail and redundant self-timed coding, and is compliant with IEEE 754 Standard. SIFPC processes either one operation with double precision numbers, or two simultaneous operations with single precision numbers, and calculates two results: sum and difference between product of first two operands and third operand. SIFPC consists of two identical channels with common input and output. An order of data outputting matches the order of an input data. Each channel implements full data processing path and has two pipeline stages: first is multiplier and exponent calculation, and second is all rest parts. This reduces hardware complexity and accelerates calculations due to reducing number of intermediate registers and cutting number of "bottlenecks" in an indication subcircuit of the unit. An additional speed-up of performance, comparing to a traditional self-timed circuit implementation, is achieved due to utilizing bit-wise and simplified (adaptive) indication. Multiplier utilizes modified Booth algorithm with Wallace tree, self-timed redundant code and ternary adders. First stage of the Wallace tree compresses four dual-rail partial products into two ternary operands. The unit is designed for standard 65-nm CMOS process. It has 1.12 mm2 die size, demonstrates 3.15 Gflops performance and 1.8 ns latency.**

***Keywords* — wallace tree, bit-wise indication, redundant coding, ternary adder.**

I. Introduction

A multiply-accumulate operation long since is a part of an instruction set of the modern digital signal processors (DSP). It supposes an addition (or subtraction) of first input operands production with an accumulator content. A hardware superposition of the multiplying two operands and following addition (subtraction) with third input operand in one unit can be performed with two sequential rounding, which is typical for DSP, or with single rounding. The last case is known as fused multiply-add (FMA) operation. It provides higher calculation accuracy. Thus FMA operation de facto has became a typical operation of the modern central processors.

A bibliography on FMA unit development problem is quietly wide. The most part of all publications describes designing synchronous FMA units [1]-[3]. Recent years demonstrate an increase in number of the publications devoted to asynchronous FMA implementations [4]-[5]. However, the asynchronous solutions would-be the self-timed units are basing usually on "weak" transistors and do not response the requirements for designing noise resistant and power-efficient self-timed units, which correct functioning does not depends on cell's delays (Speed-Independent circuits) at any operating conditions.

SI circuits are free of clock tree and work only "on demand". This ensures decreasing power consumption. Their robustness at ultra-low supply levels opens a wide perspective for obtaining long-playing portable devices with an accumulator supply and for designing on-board computers with limited energy resources. A stable operability in extreme conditions is achieved due to hardware redundancy and the additional delays for SI circuit indication and spacer phase. However, an appropriate design of the SI circuits allows for essential decreasing such redundancy, and for obtaining even better results comparing to the synchronous prototypes in some cases, for example, in fault-tolerant units [6].

Earlier we have already made an attempt to design SIFMA unit in a Gigaflops range [7]-[8]. However, to achieve the speed limit, SIFMA was implemented as self-timed circuit with a speculative indication on a base of the pipeline with large step number. As a result, declared performance was achieved at the expense of waiver of full self-checkability of the SIFMA that actually brought it outside the class SI circuits

This article presents the results of designing 64-bit really SI floating point coprocessor (SIFPC) performing FMA operation in accordance with the IEEE 754 Standard and having performance at the level of 3 Gflops, compared to the previous implementation [7]-[8].

II. SIFPC Features

SIFPC is designed for stream data processing. Each of the three processed 64-bit operands contains either one double or two single-precision IEEE 754 number. In the latter case SIFPC performs two independent FMA operations on two operand triplets of the single precision. An additional feature of SIFPC is the simultaneous calculation of not only the sum but also the difference between product of first two operands and third operand.

A. Block diagram

A trend of the modern computational tools development is an ensuring minimum of 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 per single VLSI for high-performance computers.

Two phases in the work of any SI circuits (namely, active, or work, and pause, or spacer) suggests the idea of using two parallel channel, which phases are alternated. The implementation of similar device SIFMA [8] only multipliers and exponent processing units (MEP) were parallel, as shown in Fig. 1. Addition-subtraction, normalization and rounding steps were performed using common for two channels resources.

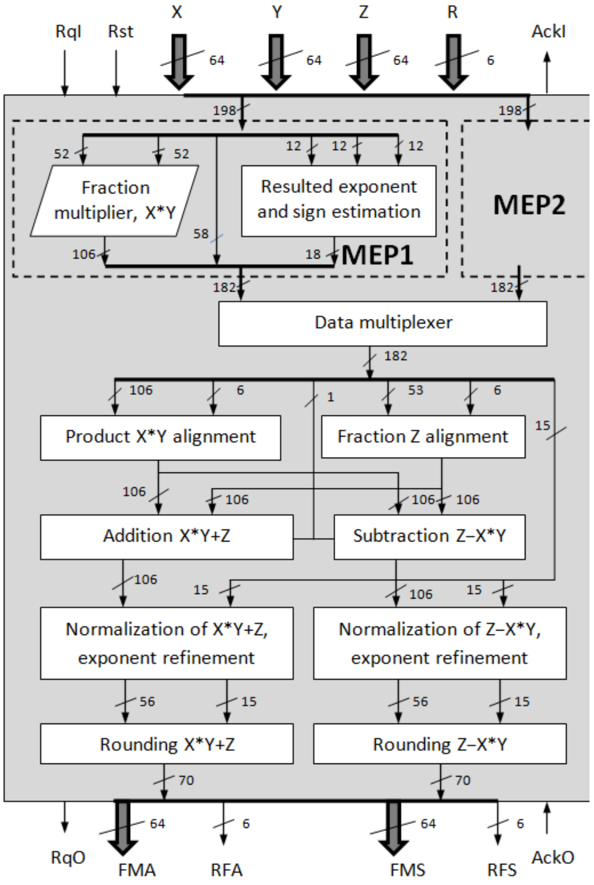


Fig. 1. Block diagram of the SIFMA core

Parallel MEP blocks formed first stage of the SIFMA core pipeline. A common tract of the subsequent calculations was divided into 4 pipeline stages. This enabled to achieve an average performance at the level of 2.82 Gflops during simulating SIFMA in both double and single precision modes. The transition from speculative indication to full one at the same pipeline organization has resulted in deterioration of the performance to the level of 2.31 Gflops. Analysis of the results has showed that performance degradation in the case of full indication was caused by the indication subcircuit. The need to store a large number of signals in the pipeline stage registers, firstly, substantially increased hardware costs of both registers and their indication subcircuit implementation, and secondly, led to an additional delay for forming total indicator of the pipeline stage.

In this regard, a new block diagram of the unit performing FMA operation was proposed. This is SI floating point coprocessor (SIFPC) shown in Fig. 2. Here an entire tract processing input data is duplicated and full indication is utilized, and the number of pipeline stages is minimal. This allowed for reducing hardware cost of the intermediate registers implementation including their indication subcircuits, and for decreasing number of "bottlenecks" in a critical path of the FMA calculation. As a result, the average SIFPC's performance reached 3.15 Gflops when working with a synchronous environment, and 3.9 Gflops in the absence of unproductive waiting a response from a synchronous environment indicating a successful reading of the current result.

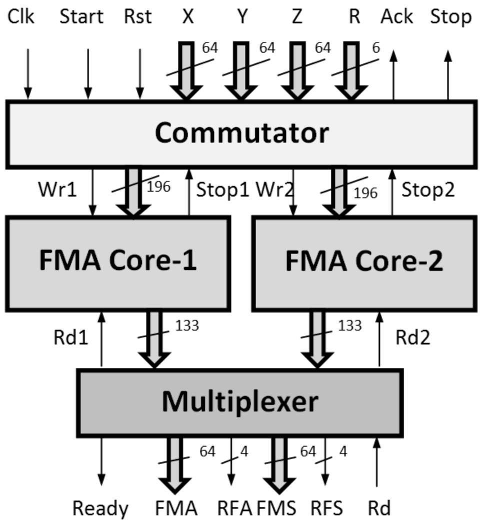


Fig. 2. Block diagram of the SIFPC

Analysis of the minimax curve in "power-area" domain in accordance with the methodology presented in [9], and in relation to the 65-nm technology and assumed performance of the SIFPC, led to determine the main characteristics of the FMA core prototype and choose block diagram (Fig. 3) for its implementation.

Wr and *Ack* signals provide an asynchronous input interface, first one reflects data availability at the SIFPC input, and the latter acknowledges a successful completion of acquiring input data by SIFPC. Similar signals *Ready* and *Rd* provide an asynchronous output interface: signal *Ready* indicates the readyness of the result; signal *Rd* acknowledges the end of reading result by an asynchronous environment.

The core of the FMA contains input and output FIFO. They improve FMA performance when working with synchronous environment [7] due to the buffering of the data stream. Input and output FIFO are implemented as semi-dense self-timed shift registers [10, Fig. 11.9] with 3 words capacity.

The *Clk* input connects to the clock source of the synchronous environment or to the source of the signal indicating the readiness of input operands and operation markers in an asynchronous environment. *Ack* output is used only by asynchronous environment as part of its request-acknowledge interaction with SIFPC.

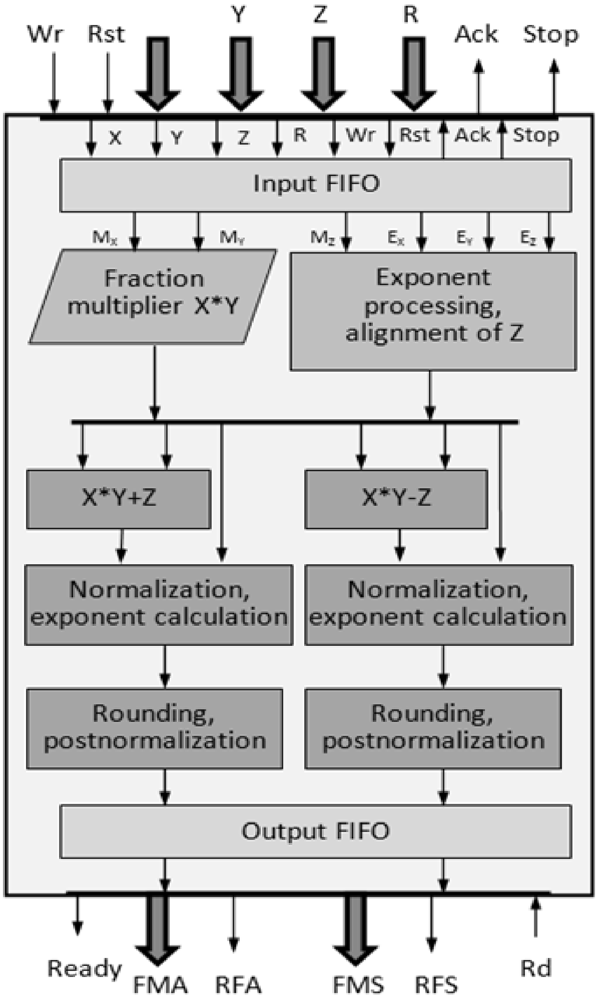

Wr
Rst
Y
Z
R
Ack
Stop
X
Y
Z
R
Wr
Rst
Ack
Stop
Input FIFO
M_X
M_Y
M_Z
E_X
E_Y
E_Z
Fraction multiplier X*Y
Exponent processing, alignment of Z
X*Y+Z
X*Y-Z
Normalization, exponent calculation
Normalization, exponent calculation
Rounding, postnormalization
Rounding, postnormalization
Output FIFO
Ready
FMA
RFA
FMS
RFS
Rd

Fig. 3. Block diagram of the FMA core

Since all functional blocks in SIFPC are SI circuits, result update at the output of the SIFPC happens only after acknowledging the successful result reading by environment by means of signal *Rd* = 1. In the case of asynchronous environment result is updated naturally within the request-response interactions between SIFPC and environment. Synchronous environment can use output *Ready* to write results into its registers, and form signal *Rd* on base of either signal derived from the *Ready* signal or the system clock. However, in the latter case the performance SIFPC will be understated due to unproductive waiting active clock edge after SIFPC output result become ready.

Traditionally the multiplication algorithms are based on modifications of the Booth algorithm and pipelined Wallace tree (WT). However, developers of the modern multi-core processors try to reduce down to a minimum the number of pipeline stages implementing WT. This trend corresponds to the SIFPC: fraction multiplication, result exponent calculation and third operand alignment are implemented in a single pipeline stage.

Normally SI circuits, such as SIFPC, are implemented using the dual-rail data coding. However, the studies have shown [7] that redundant (ternary) self-timed (ST) coding provides the best parameters of the SIFPC.

B. Redundant self-timed coding

Multiplier is the most complicated part of SIFPC. Its prototype is a synchronous circuit implementation [11]. Prototype uses the redundant coding of the operands providing the compression ratios 4:1 at the first stage of the WT and 2:1 in all subsequent stages. Analysis of the possible ST codes has resulted in selecting redundant ST code [12] represented in the Table 1.

Table 1

Redundant ST coding

Coded state	**Ternary code**		
	AP	***AM***	***A0***
+1	1	0	0
0	0	0	1
–1	0	1	0
spacer	0	0	0

Figure 4 demonstrates a circuit of 1-bit ternary adder with full indication of the internal signals and outputs. A dotted oval selects indication subcircuit. The input operands and sum output are represented in redundant ST code (Table 1). Other signals have a dual-rail coding.

Each Adder in the first WT stage is complemented by two circuits converting a difference between the two dual-rail operands into one ternary operand (Fig. 5).

For comparison, Fig. 6 represents 1-bit dual-rail SI adder. It has less implementation complexity, and this affects a total WT complexity: 1-bit ternary WT complexity equals to 2530 CMOS transistors, while dual-rail one has 2260 transistors. But because of the smaller compression ratio in the dual-rail WT its performance turns out to be a worse by 16% than performance of the ternary WT.

Thus, the implementation of the WT with redundant ST coding significantly improves a performance of the multiplier in comparison to classic ST algorithm with dual-rail coding. This is achieved due to reducing the number of compression stages from 7 to 4, while the hardware cost of the ternary WT are larger by 12% than in the dual-rail case.

SIFPC performance largely depends on the implementation of its indication subcircuit.

C. SIFPC indication

SIFPC is a device processing multi-bit data. A classic indication of such device is built on base of forming a common indicator for each functional unit, and leads to a significant slowdown in its work. A "bottleneck" of the multi-bit SI devices is their common indicator that combines all indication signals into one common indicator, and takes part in the request-acknowledge interactions between functional blocks of the SI unit.

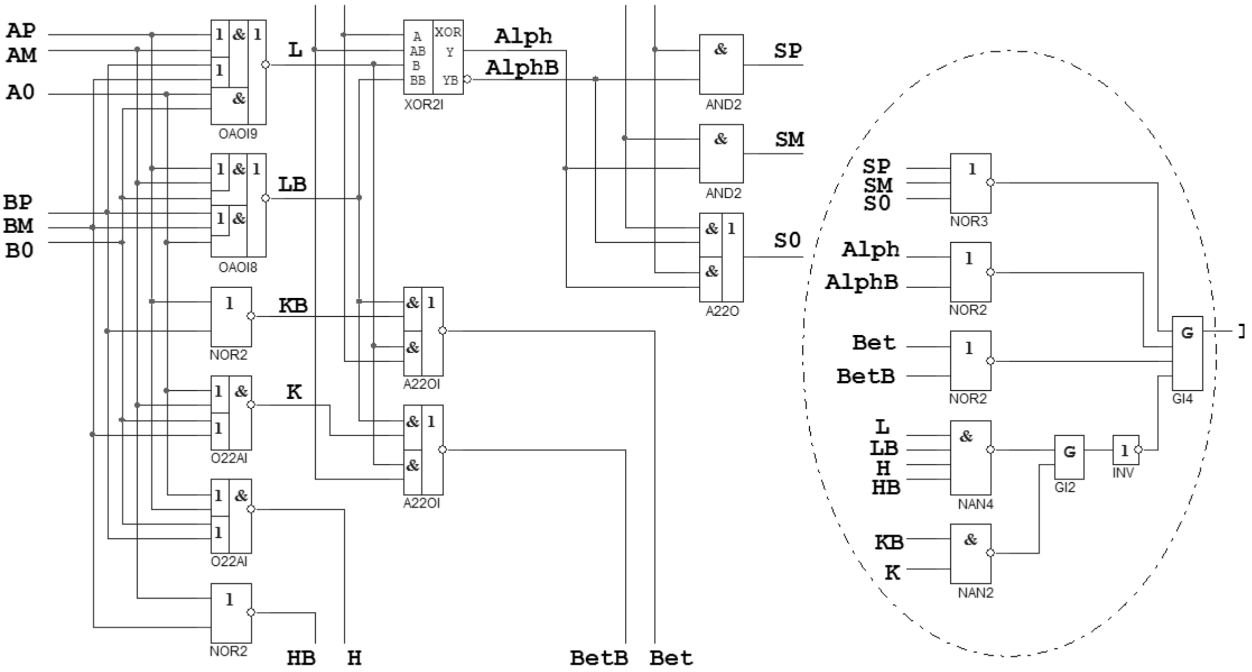


Fig. 4. 1-Bit ternary ST adder

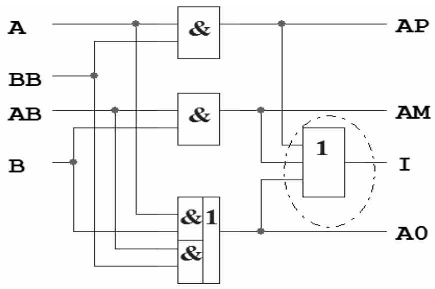


Fig. 5. Converter of the difference between two dual-rail operands to ternary operand

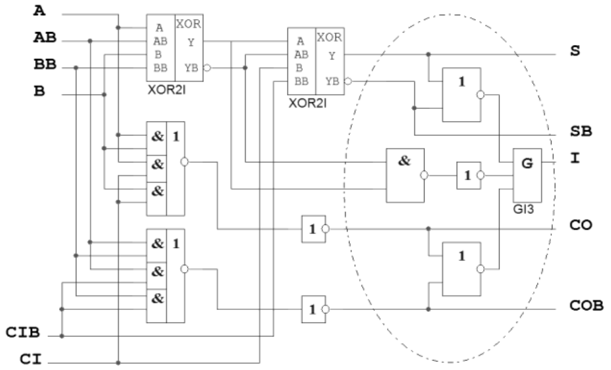


Fig. 6. 1-Bit SI dual-rail adder

From the theoretical viewpoint [10], the circuit is not SI circuit, if at least one element, which was initiated to switch to inverse state during a transition from one phase of the work of the circuit into opposite phase, did not terminate this switch before input state changes. However, an analysis of the practical SI circuits shows that in almost every SI not always absolutely all elements, which were initiated to switch into working phase, take part in the formation of the next state of the information output. This makes a foundation for simplifying multi-bit SI circuits, including SIFPC.

Indication simplification at the level of one bit is based on two properties that characterize the CMOS circuits and SI circuits with properly organized signal discipline:

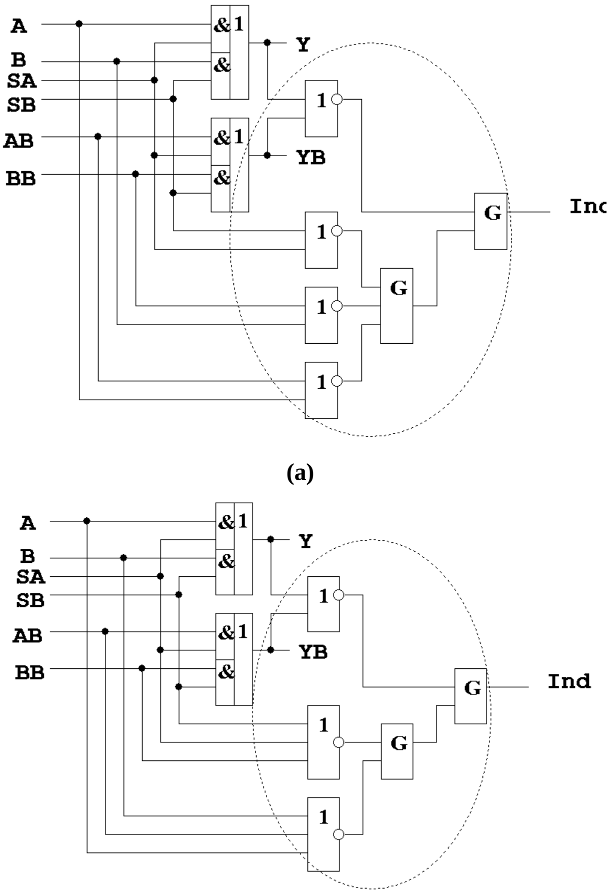
1) CMOS cell ceases its transition if its output state corresponds to the input state.

2) During transition of SI circuit from spacer to working phase each its cell can switch to working state only once.

This allows for using necessary and sufficient, but simplified indication of the working phase and full complete indication of the spacer phase in any SI functional block. Indeed, full indication of the spacer phase ensures that the switching SI circuit into the regular working state will always start from a state in which all the circuit cells are in spacer state. Switch all outputs into the working phase indicates that from the view point of environment circuit transition to a new working state has been completed.

Simplified (adaptive) indication is realized through the establishing indication outputs of first cascade of the indication subcircuit by means of logical functions "OR" (for information signals with zero spacer) or "AND" for information signals with unit spacer. Besides each cell of first cascade of the indication subcircuit should have both components of at least one dual-rail signal among its inputs.

Fig. 7a shows a schematic of a simple CMOS multiplexer 2:1 with dual-rail information inputs and select inputs with zero spacer and indication subcircuit, which is full complete in both phases of the work. Basis of the circuit implementation has a restriction on a number of the consecutively connected p-transistors (no more than three). Fig. 7b demonstrates the circuit of the same multiplexer with a simplified indication. The analysis of these circuits shows that simplified indication subcircuit in 1.1 times easier and by 10-15% faster than case with full indication.



(a)

(b)

Fig. 7. Multiplexer 2:1 with dual-rail signals: (a) full indication; (b) simplified indication

Note that a total indication signal generation for the functional blocks (the pipeline stages) participating in the hand-shake interaction must be made by full indication circuit in both phases of the work of SI unit. In a case of the pipelined implementation of SI unit, one should realize full indication in both phases of the work either for input or for output registers of the pipeline stages.

Simulation of the various digital units has demonstrated that considered adaptive indication method reduces indication subcircuit complexity by factor of 1.5-1.7 (depending on standard cell library) and advances its performance by 5-25%.

Utilizing simplified indication in the working phase results in the formation of a new subclass of SI circuits – namely, the circuits with the adaptive indication (SIAI). This implementation ensures circuit will transit into new working state from the proved spacer state. This provides retention of all practical properties of SI circuits: detection of the constant malfunctions, wide workability range on supply voltage and ambient temperature, etc.

It also should be noted that SIAI circuit ceases to be semi-modular and causes diagnosis regarding the self-timing violation at its analysis by means of available software (for example, [13]). This violation is such only from the viewpoint of the formal theory, because full control of switching all circuit cells into spacer phase before entering new working state ensures that such circuit will remain the self-timed circuit.

The aspects of an indication subcircuit optimization for SI circuits in terms of the hand-shake interaction in the SIFPC pipeline have been discussed in detail in [8].

Proposed method of the indication subcircuit optimized building has reduced hardware expenses of the entire SIFPC circuit by 12% and has improved its performance by 17% in comparison with the version presented in [8].

III. LAYOUT IMPLEMENTATION OF THE SIFPC

SI circuits utilize a redundant information signals coding and indication of the transition processes. Therefore, the complexity of the combinational SI circuits increases in 2.3-2.5 times in comparison with synchronous analogues. The number of the signals in SI circuit is increased in the same proportion.

The ternary encoding of the operands in SI multiplier mitigates this problem. Firstly, the Wallace tree is traditionally implemented on base of carry-save adders. At ternary ST encoding of the summands each such adder in the multiplier has 3 outputs, while at dual-rail ST encoding each adder has 4 outputs. Secondly, carry signals in the ternary SI adders go to an adjacent bit of the same cascade of the Wallace tree and do not occupy any vertical traces.

Analysis of the Booth coder modifications, partial products (PP0...PP26, CS) generator, and the number of signals formed by them as the Wallace tree inputs has shown that the total number of inputs of the Wallace tree (more than 3000) exceeds the number of available traces in the vertical wiring layers for given maximal horizontal size of the designed SIFPC. Therefore Booth coder and partial products generator were integrated into Wallace tree structure, as shown in Fig. 8. Block "Coder Booth & Σ" implements a generator and adder of the partial products indicated in brackets. Block "Σ" adds partial products in brackets.

To obtain two single precision products fairly summarize the partial products PP0-PP12 and adjustment operand CS. Adders of the partial products PP13-PP26 are needed only to multiply double precision operands with 53-bit fraction.

The topology of the multiplier has the die size 404 × 490 μm^2. The die size of entire SIFPC layout (Fig. 9) equals to 982 × 1140 μm^2.

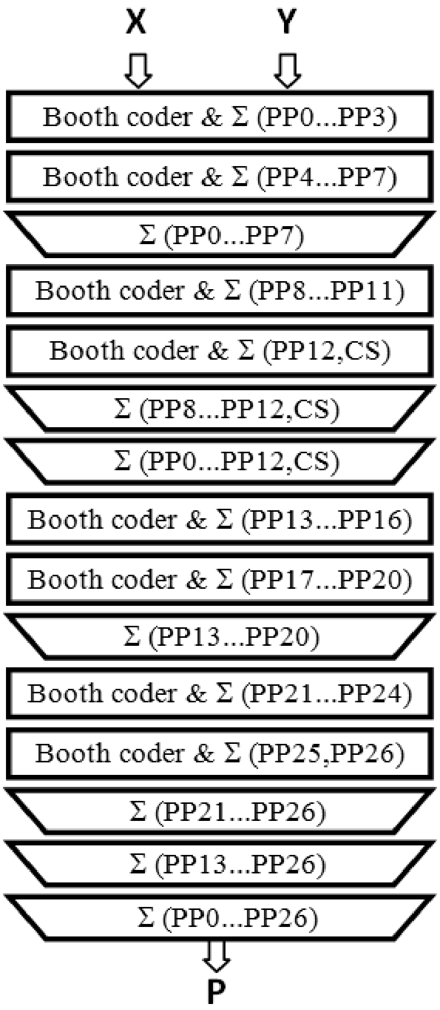

X
Y
Booth coder & Σ (PP0...PP3)
Booth coder & Σ (PP4...PP7)
Σ (PP0...PP7)
Booth coder & Σ (PP8...PP11)
Booth coder & Σ (PP12,CS)
Σ (PP8...PP12,CS)
Σ (PP0...PP12,CS)
Booth coder & Σ (PP13...PP16)
Booth coder & Σ (PP17...PP20)
Σ (PP13...PP20)
Booth coder & Σ (PP21...PP24)
Booth coder & Σ (PP25,PP26)
Σ (PP21...PP26)
Σ (PP13...PP26)
Σ (PP0...PP26)
P

Fig. 8. Product calculation flow

IV. SIFPC PARAMETERS

SIFPC has been designed in an industrial 65 nm CMOS process with six metallization layers. SIFPC parameters in comparison with synchronous analogue of a nearest performance [9] are listed in Table 2. Timing and power parameters were derived from simulating SIFPC unit taking into account the parasitic capacitances and resistors extracted from the layout for statistically reliable set of input operands for double and single precision.

Table 2

SIFPC Parameters

Parameter	Analogue	SIFPC
Input clock, GHz	1.03	1.05
Die size, mm^2	0.312	1.12
Latency, ns	10.8	1.84
Performance, Gflops	2.06	3.15
Die size efficiency, mm^2/Gflops	0.151	0.321
Workability range on supply V_{VDD}	V_{VDD}±10%	V_{th}...V_{BD}
Constant faults detection	–	+

Performance was determined for typical operating conditions (1.0 V supply voltage, 25^{0}C ambient temperature), because the performance of the SI circuits always corresponds to the current operating conditions, and they do not require to consider the worst case.

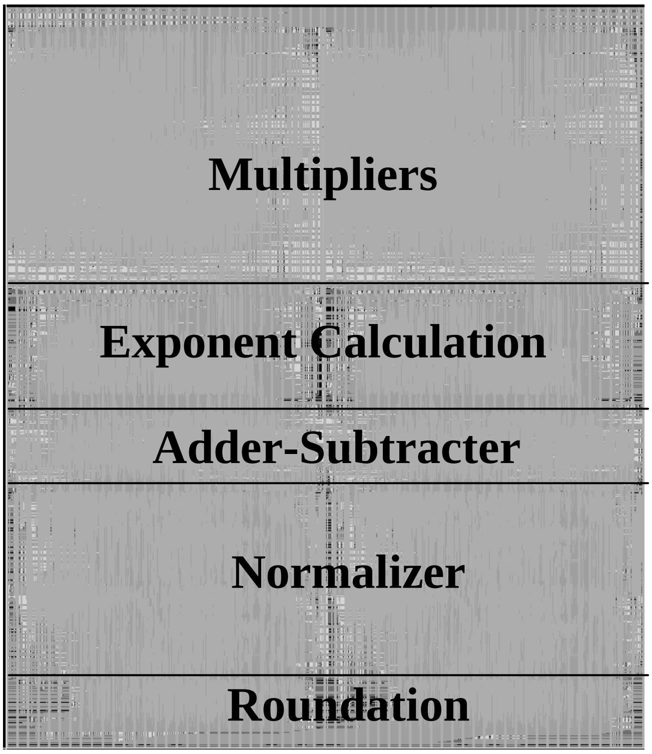


Fig. 9. SIFPC layout

Note that SIFPC has greater functionality when compared with analogue. In one cycle, it is able to process a triple of double precision operands or two triples of single-precision operands, calculating not only the sum but also the difference between first two operands product and third operand. In addition, it has a much wider workability range. In supply voltage it is bounded only by a threshold voltage (V_{th}) of the CMOS transistors from below and by a breakdown voltage of semiconductor active structures (V_{BD}) from above. And it stops if any constant fault happens [10]. Payment for these benefits is greater complexity and, therefore, more power consumption. The later can be decreased to the desired value by reducing the supply voltage accompanied by corresponding decrease in performance. Due to fewer pipeline stages SIFPC latency is in 5.9 times less than latency of the synchronous counterpart.

Therefore, the presented SIFPC provides performance at level of 3.15 Gflops due to parallelization of the operations. It utilizes a modern trend in building high-performance computing tools: using more processors with relatively low performance, instead of raising system frequency of each processor.

This also helps to solve the problem of improving reliability of the modern computing systems, for example, by duplicating SIFPC self-checked in relation to constant faults for building fault-tolerant computers.

V. Future researches

To reduce the hardware costs and power consumption of the SIFPC, we are going to study an implementation of its multiplier as a two-stage pipeline calculating two halves of the Wallace tree for most and least significant bits separately and consistently. This should remain the same performance of the SI-FMAS while using one computing channel instead of two identical channels due to co-processing dual-rail calculation discipline in one channel at two pipeline stages.

Additional improvements can be achieved due to using ternary ST coding temporary results not only in multiplier, but also at the following stages of the data processing, and by utilizing speculative estimating result normalization.

VI. Conclusion

Applying SI circuitry for modern computers implementation helps to effectively use hardware methods for monitoring safety and validity of the calculations.

SIFPC with two concurrent FMA blocks designed on 65 nm CMOS process demonstrates the high average performance (3.15 Gflops at typical conditions) and low latency (less than 2 ns).

Usage of the redundant ST coding, simplified indication and the minimum number of the pipeline stages ensure the development of competitive on performance 64-bit coprocessor implementing FMA operation and having all the advantages of the SI circuits: full self-checkability in relation to the constant failures, preserving the workability when midget supply voltages.

Support

A study was done with partial support of fundamental research program of the ONIT RAS for 2016 year (draft 1.4) in the Institute of Informatics problems of the FRC CSC RAS.

References

[1] Pillai R.V.K., Shah S.Y.A., Al-Khalili A.J., Al-Khalili D. Low power floating point MAFs – A comparative study // Proc. Sixth International Symposium on Signal Processing and its Applications, Kuala Lumpur, 2001, V. 1. P. 284-287.

[2] Seidel P.-M. 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] Bruintjes T. M. Design of a Fused Multiply-Add Floating-Point and Integer Datapath. Master's thesis, University of Twente, Enschede, the Netherlands, 2011. 154 p.

[4] Noche J.R., Araneta J.C. An asynchronous IEEE floating-point arithmetic unit / Science Diliman, Philippines. 2007. V.19. No. 2. P. 12-22.

[5] Manohar R., Sheikh B.R. Operand-optimized asynchronous floating-point units and method of use therefor, US patent, № 20130124592. May 2013.

[6] Stepchenkov Y., Diachenko Y., Zakharov V., Rogdestvenski Y., Morozov N., Stepchenkov D. Self-Timed Computing Device for High-Reliable Applications // Proc. International Workshop on power and timing modeling, optimization and simulation (PATMOS'2009), Delft, Netherlands, 2009. P. 276-285.

[7] Sokolov I.A., Stepchenkov Y.A., Rozhdestvenskii Y.V., Diachenko Y.G. Speed-Independent Fused Multiply-Add Unit of Gigaflops Rating: Methodological Aspects // Problems of Perspective Micro- and Nanoelectronic Systems Development - 2014. Proceedings / edited by A. Stempkovsky, Moscow, IPPM RAS, 2014. Part IV. P. 51-56 (in Russian).

[8] Stepchenkov Y.A., Rozhdestvenskij Y.V., Diachenko Y.G., Morozov N.V., Stepchenkov D.Y., Surkov A.V. Speed-Independent Fused Multiply-Add Unit of Gigaflops Rating: Implementation Variants // Problems of Perspective Micro- and Nanoelectronic Systems Development - 2014. Proceedings / edited by A. Stempkovsky, Moscow, IPPM RAS, 2014. Part IV. P. 57-60 (in Russian).

[9] Galal S., Horowitz M., Energy-Efficient Floating-Point Unit Design // IEEE Transactions on computers. 2011. V. 60. No. 7. P. 913-922.

[10] Varshavskij V.I. et al. Avtomatnoe upravlenie asinhronnymi processami v JeVM i diskretnyh sistemah [Automatic control of the asynchronous processes in computers and discrete systems]. Moscow, Russia, 1986, 400 p. (in Russian).

[11] Makino H., Nakase Y., Suzuki H., Morinaka H., Shinohara H., Mashiko K.. 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.

[12] Stepchenkov Y.A., Zakharov V.N., Rogdestvenski Y.V., Diachenko Y.G., Morozov N.V., Stepchenkov D.Y. Speed-Independent Floating Point Coprocessor // Proc. IEEE Eeast-West Design and Test Symposium, Batumi, Georgia, September 26-29, 2015. P. 111- 114.

[13] Rozhdestvenskii Y.V., Morozov N.V., Rozhdestvenskene A.V. ASPECT – a Subsystem of Event Analysis of Self-Timed Circuits // Problems of Perspective Micro- and Nanoelectronic Systems Development - 2010. Proceedings / edited by A. Stempkovsky, Moscow, IPPM RAS, 2010. P. 26-31 (in Russian).