Challenges of the algorithms optimization and high performance arithmetic coprocessors development for numerical modeling of gas flow and heat transfer in the combustion problem

Sergey Aryashev, Sergey Bobkov, Pavel Zubkovskiy, Eugene Ivasyuk, and Yuri Stepchenkov

Citation: AIP Conference Proceedings **1738**, 220008 (2016); doi: 10.1063/1.4952007
View online: http://dx.doi.org/10.1063/1.4952007
View Table of Contents: http://scitation.aip.org/content/aip/proceeding/aipcp/1738?ver=pdfcov
Published by the AIP Publishing

Challenges of the Algorithms Optimization and High Performance Arithmetic Coprocessors Development for Numerical Modeling of Gas Flow and Heat Transfer in the Combustion Problem

Sergey Aryashev[a], Sergey Bobkov[b], Pavel Zubkovskiy[c], Eugene Ivasyuk[d] and Yuri Stepchenkov[e]

[a]*Ph. D., Head of department, Scientific Research Institute for System Analysis, SRISA, Moscow, Russia*
[b]*Doctor of Science, Head of division, Scientific Research Institute for System Analysis, SRISA, Moscow, Russia*
[c]*Head of section, Scientific Research Institute for System Analysis, SRISA, Moscow, Russia*
[d]*Research fellow, Scientific Research Institute for System Analysis, SRISA, Moscow, Russia*
[e]*Ph. D., Head of department, Institute of Informatics Problems, FRC CSC RAS, Moscow, Russia*

Abstract. Computer simulation of multiscale burning and detonation processes requires an exaflop-scale performance supercomputer. The paper present research from SRISA intended to development high-performance architectures of DSP extensions for burning process simulations. Also a number of solutions for dataflow coprocessor development based on self-timed circuits are proposed.

Keywords: Supercomputer, burning task, statistical analysis, stochastic method, microprocessors architecture extensions, FMA, dataflow coprocessor, self-timed circuits.
PACS: 02.70.Ns, 02.50.Fz

COMPLEX COMBUSTION PROBLEM

Computer simulation of multiscale burning and detonation processes requires an exaflop-scale performance supercomputer. One of the key issues to be required solutions is modeling of new types of fuel combustion in the engines of the new designs. These processes require predictive multiscale modeling, including the course of the reaction at the level of individual molecules and modes of detonation in the combustion chamber. Multi-scale combustion processes does not allow them to direct numerical simulation using existing supercomputers and requires the achievement of exaflop-level performance.

Should be noted that a direct increase in code resolution by increasing the number of nodes throughout the calculated field task is inefficiently. Even with supercomputers, direct reduction of computational cells will lead to a corresponding increase in the time steps numbers, that for long-term prognosis will cause the accumulation of calculation errors, which reduces the benefits of supercomputers.

Exaflop-scale performance is required for multi-scale coupling calculation process with a given accuracy at a pace of their own time each process, the application of the most efficient algorithms for solving different types of problems, ensure the effective exchange of data, not just for the possibility of a direct increase in the number of computational operations.

Authors proposed exaflop-scale class architecture with data exchange system aimed at solving specific problems of combustion, which will allow for a balanced multi-scale modeling of processes taking into account the kinetic gas dynamic and interfacial processes with equal accuracy.

PROFILING RESULTS

At present, computer simulation of combustion and detonation carried out by parallelization of computational programs on multiprocessor and multicore computers and graphics accelerators [1]. By use the statistical analysis (profiling) we can get information about the most commonly used functions in such tasks or functions, the calculation of which takes the longest time.

International Conference of Numerical Analysis and Applied Mathematics 2015 (ICNAAM 2015)
AIP Conf. Proc. 1738, 220008-1–220008-4; doi: 10.1063/1.4952007
Published by AIP Publishing. 978-0-7354-1392-4/$30.00

Table 1 shows the results of the statistical analysis program ETDRK4 method for ODE system with the kinetics of the Warnatz and Maas (burning hydrogen, 9 component, 19 reactions) and gas-dynamic code with the same kinetics, but the solution of the system is carried out by the 4-step Rosenbrock function. The launch was carried out on a computer with x86-processor Intel (R) Core (TM) 2 Duo CPU E6550 2.33GHz. The first column shows the execution of functions as a percentage of the total program time, second column - the name of the function, and the third - the notes.

TABLE 1. Profiling results of the combustion problem.

Runtime	Function name	Comment
20.93%	GetTransport	The transfer coefficients for the gas mixture based on the molar mass, or density components
11.67%	__ieee754_exp	The library function calculating the real exponent (libm-2.17.so)
6.88%	UpdateExrate	The addition of data from an elementary reaction to the intensity of the component formation
6.70%	ElemCofs	The elementary mechanism: the coefficients for the forward and reverse reaction
4.02%	ExplTurbStep	Explicit stage of turbulence
3.53%	GetJacobean	Evaluation of the Jacobian
3.47%	__pow_finite	The library function raising to a power (libm-2.17.so)
2.59%	GetDifComp	Accounting for component diffusion
2.53%	Y2X	Calculation of the density of components in each cell

Code functions analysis and content analysis which most commonly used in the considered implementation of combustion problem show that in most cases carried out the multiplication and addition on matrices and vectors, vector operations division and square root. Function calls of exponent calculation and raising to a power occur in the cycle of the individual elements of the vector, thus using only scalar functions.

The set of functions described above in the particular example of implementation of the combustion problem is typical for many computing tasks, and that the execution of these operations, together with the capabilities of the memory subsystem performance characterizes the microprocessor performance in engineering and scientific calculations.

HIGH PERFORMANCE ARITHMETIC COPROCESSORS

It is possible to allocate family of high-performance microprocessors from Russian developments, on which work is carried out in SRISA. The microprocessor VM6YA [2,3], VM7YA, as well as a promising development VM8YA used different architectural solutions to improve performance for scientific and engineering calculations.

A vector co-processor (CPV) in the processor VM8YA is the result of the development of complex computing (Fig. 1). CPV arithmetic core enables the operation of single and double precision vectors over a width of 128 bits. Peak performance is achieved by using the commands of complex multiplication with addition and subtraction of the third operand ("Fourier Butterfly") and is 10 double-precision operations per cycle or 20 single-precision operations. Computational capabilities of the coprocessor can be used for custom applications, compiled with using standard mathematical libraries.

Important condition for the effective use of a vector coprocessor on the user tasks is the presence of a math library that is optimized for the coprocessor architecture. Some of functions of the widely used BLAS library have been optimized for the architecture coprocessor.

The results of performance tests LinPack profiling shows that the most commonly used functions of the library BLAS (more than 90% of execution time) is a function DGEMM - matrix multiplication, DTRSM - solution of linear differential equations and DGETRF - LU-decomposition, and the DGEMM function is the basis for DTRSM and DGETRF with using block methods. The maximum efficiency is close to 64% in the case where initial data is fully cached by L2, the number of arithmetic instruction is large enough to cover the cost of data transfer and complex data type allows the use of the most productive commands coprocessor.

To achieve high performance during the engineering calculations not only necessary linear algebra math library optimized for the specific architecture, but also the possibility of hardware acceleration evaluation of transcendental functions. Performing these calculations in integer format or a single precision floating point no longer meets the constantly increasing demand for engineering calculations precision.

With limited data bit depth representation in the computer, transcendental (special) functions cannot be calculated exactly, and can only be approximated. One of the most common algorithms for computing special functions is an approximation by polynomial/ rational fraction or by a power series expansion. A main parameter in the design of approximation algorithm is the maximum permissible error in the computation.

FMA unit used as arithmetic core in the vector coprocessor in VM8YA (Fused Multiply-Add) [3]. Internal data bit depth in this unit is 161 bits, which allows it to create a scheme of calculating polynomials with precision up to 0,5ULP. An important feature of the block FMA module is sequence of the steps of internal normalization, and addition with carry propagation. In the ordinary multiply-accumulate modules (FMA, multiply-add fused) is performed addition first, then normalization and rounding, but in described module first performed normalize and then the final addition, combined with rounding. Such an organization processing steps allows more efficient use of internal mantissa result bit width and not to lose the least significant bits in the processing of the worst cases of the argument reduction. Solving the problem of worst case (or catastrophic loss of accuracy) is achieved through the use of formulas for calculating the argument:

$$r = x - k * C1 - k * C2 - k * C3, C = C1 + C2 + C3, k = \text{int\_}rnd(\frac{x}{C}),$$

where x- input operand, *C1, C2, C3* – reduction constants. As the result the reduction error will be less than *1ulp* of double precision format.

To reduce calculates degrees of the polynomial or the amount of "multiply-accumulate" operations it is possible to use parallelism. For example, can lead a polynomial to the form, which involves the simultaneous computation of several parts, Fig. 2c.To implement such an algorithm requires support calculation function of the degree N/2.

The function calculation can be provided by means of rational approximation, (see Fig. 2b), which can reduce the estimated degree polynomial approximating of the numerator and denominator of about half compared with the embodiment shown in Fig. 2a. However, division operation of the numerator and denominator support is required.

The algorithm is shown in Fig. 2c, potentially faster than in Fig. 2b, since the operation FMA, completing the calculation, significantly faster than the DIV operation, and the raising to power operation delay can be masked by major pipeline delay of calculating polynomials. Structural scheme presented in Fig. 2 is fully pipeline.

If «multiplication with accumulation» operation based on FMA module will be chosen as basic operation, the structure diagram of the module implements the whole algorithm becomes more regular, Fig. 2d.

Microprocessors have been implemented as RTL-models on Verilog language. Stochastic method based on own random tests generator are used for coprocessors verification.

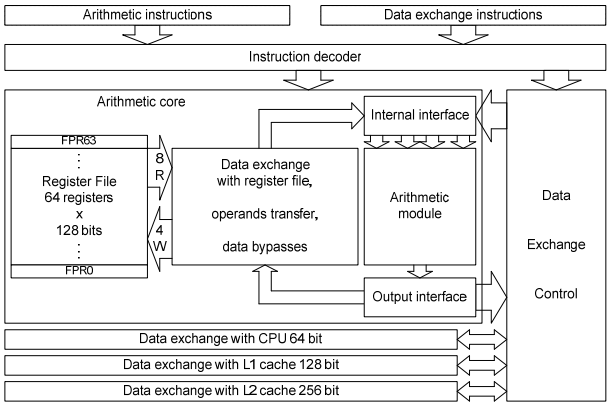

FIGURE 1. Vector coprocessor of the VM8YA processor structure diagram.

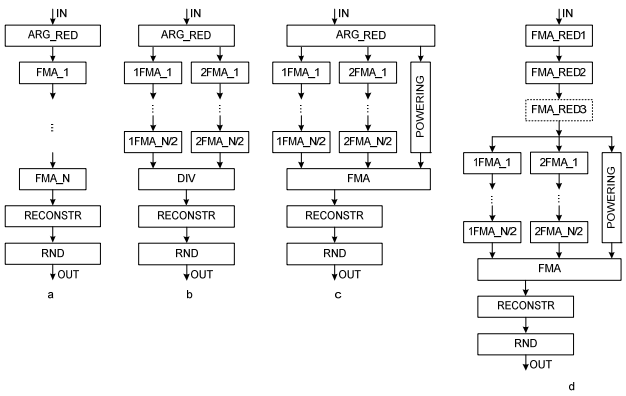

FIGURE 2. Structure diagrams of the elementary functions pipeline modules.

SELF-TIMED CIRCUITS

In super-computer of exaflops range, having hundreds of millions cores, it is necessary to utilize the hardware methods for controlling reliability and validity of calculation results. The usage of self-timed circuitry for implementing the major computational units of the super-computers helps to solve this problem efficiently.

FMA is a standard operation in the modern computers. Most publications cover synchronous FMA units. But during the last years many publications describing asynchronous FMA units implementation have appeared [4]. The latest solutions based on weak transistors do not meet the need to develop jam-resistant and energy effective self-timed (ST) units with proper operation not depending on element's delay, i.e. Speed-Independent (SI) circuits.

When developing SI-adders and SI-multipliers it is reasonable to use the redundant ST-code together with paraphase code. Usage of the redundant ST-coding, together with introducing new subclass of SI-units and realizing SIFMA-unit with minimal pipeline stage number have allowed for the first time to develope the competitive by performance 64-bit FMA unit possessing all advantages of the SI-units: pure self-checking with respect to constant failures, retention workability at very-small supply voltage [5]. These advantages also create conditions for effective implementation of the fail-safe units.

Hardware redundancy and additional delays for indication and spacer phase, which are the intrinsic features of the SI-circuits, are the payment for such advantages. However, a proper designing of SI-circuits allows to reduce this redundancy essentially, and, in some cases, even to achieve better results compared to the synchronous analogs [6, 7].

CONCLUSION

The statistical analysis of multiscale burning and detonation task led to the development architecture of complex dataflow coprocessor for use in exaflop-scale performance supercomputer. Stochastic method based on own random test generator allowed to develop the hardware implementation of coprocessor.

ACKNOWLEDGMENTS

The research is supported by the Russian Foundation for Basic Research (project 13-07-12062).

REFERENCES

1. Rybakin B.P. Vestnik Ufimskogo gosudarstvennogo aviacionnogo tehnicheskogo universiteta, 2012, vol. 16, no. 6, pp. 51. (in Russian).
2. Bobkov S.G., Arjashev S.I., Barskyh M.E., Bychkov K.S., Zubkovskij P.S. Patent RU 2359315, 20.06.2009. (in Russian).
3. Zubkovskij P.S., Ivasjuk E.V., Arjashev S.I. Soprocessor kompleksnyh vychislenij. Problemi Razrabotki Perspektivnih Mikro- i Nanoelektronnih system (MES), 2010. P. 356-359 (in Russian).
4. Noche J.R., Araneta J.C. An asynchronous IEEE floating-point arithmetic unit // Proceedings of Science Diliman. 2007. V.19. №2. P. 12–22.
5. 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, 2009. P. 276–285.
6. Sokolov I., Stepchenkov Y,.Rogdestvenski Y, DiachenkoY. Speed-Independent Fused Multiply-Add Unit of Gigaflops Rating: Methodological Aspects. Problemi Razrabotki Perspektivnih Mikro- i Nanoelektronnih system (MES), 2014. Part IV. P. 51-56 (in Russian).
7. Stepchenkov Y., Rogdestvenski Y., Diachenko Y., Morozov N.,Stepchenkov D., Surkov A. Speed-Independent Fused Multiply-Add Unit of Gigaflops Rating: Implementation Variants. Problemi Razrabotki Perspektivnih Mikro- i Nanoelektronnih system (MES), 2014. Part IV. P. 57-60 (in Russian).