computing@computingonline.net
www.computingonline.net

ISSN 1727-6209
International Journal of Computing

# SOME ASPECTS OF INCREASING THE PROCESSING POWER OF THE DIGITAL SIGNAL PROCESSORS

**Alexander Palagin, Miroslav Semotiuk, Yaroslav Vizor, Eugeniy Chichirin**

V.M.Glushkov Institute of Cybernetics
of the National Academy of Sciences of Ukraine
E-mail: yaviz@ukr.net

**Abstract:** *An analysis of the non-linear equation solving methods used by the arithmetic units of the computing devices was conducted in order to facilitate creation of the arithmetic units which would be optimal in terms of processing power and hardware overhead. A method and algorithm of a hardware implementation of computing the particular and the reciprocal in the digital signal processors at higher speed and minimal hardware overhead.*

**Keywords:** *digital signal processing, signal processor, architecture, calculus of approximations, computing speed.*

## 1. INTRODUCTION

Computing non-linear functions in the division operations, square roots and and the reciprocal operations used to be quite scarce until recently, amounting to less that 5% of the total number of operations. Thus, they were easily carried out in software mode, having low negative impact of the signal processor speed. With the advent of the new, more complex algorithms [1] employing these operations as the basic ones, a need to ensure a hardware implementation of the said operations arose, for the software methods became the brake hindering the drive to increase the signal processors' (SP) speed. Such algorithms deal with analyzing the signal spread spectrum, computing the complex number modulus or limiting one of the signal's parameters (like amplitude or frequency, for instance) by dividing it by a constant [2], etc. Having analyzed the stated above, the authors came to a conclusion that the search for the new methods of computing the particular and the reciprocal and ways of their implementation has become necessary.

## 2. THE TASK AT HAND

Analysis of the contemporary digital signal processing (DSP) algorithms shows that the DSP's processing speed is most influenced by the addition, subtraction and multiplication operations, which are frequently used. Thus, in order not to make the DSP architecture overly complex and avoid decreasing the DSP processing speed, the division algorithm should be implemented using the same operations to carry the computing out in accordance with advanced methods. Among the numerous methods of speeding up the division operation execution the most interesting ones are the synchronous methods which use the "roundabout" way employing the multiplication operation. Thus, in order to solve the problem at hand, the known synchronous methods of computing the particular with the aid of the multiplication, addition and subtraction operations are to be analyzed.

## 3. ANALYSIS OF THE SYNCHRONOUS METHODS

The division methods proposed by Stefanelli [3] are the first to be reviewed in the article. The main idea of the method boils down to arranging the redundant digits so that the sum of the members belonging to one column would be equal to the current position of the $C \cdot Q$ multiplication product. The research paper [3] describes a matrix device which is quite cumbersome due to the wide range of he acceptable digital values.

The following method is called "Harvard iterative method" and is used when the DSP arithmetic unit is working on the fixed point normalized operands while the current multipliers are using the sign bit as an information one. As a result of replacements the formula (1) is transformed into expression (6), where $x$ is a complement to 1 for $D$ (2) divisor, $N$ – dividend, and $C$ – quotient. This approach is feasible for the DSP arithmetic units supporting 8-bit operands or lower, in all the other

cases it turns out to be less effective than the Newton method.

The Newton iterative method application is also reviewed as a way of computing the quotients in a DSP. This method is employed in the paper [4] for implementing the square root operation, leading to a noticeable increase in the DSP arithmetic unit's processing speed. By using the expression (7) defining the Newton iterative method, the authors have arrived at the iteration formula (11) for computing the real quotient *C*. It has been proven that the Newton method converges at quadratic speed which is much faster than the other methods mentioned above. Besides, it also has a quite high computing accuracy at a comparatively low number of iterations. That's why the Newton method's been chosen as the base for computing the particular and the reciprocal by the DSP arithmetic unit.

## 4. APPLICATION FEATURES

The main thing to keep in mind in order to ensure correct computation of a particular using the formula (8) is the usage of normalization code $H_0$ and the initial reciprocal value $K_0$, stored in read-only memory (ROM). These quotients belong to the set of codes used in the square root computation algorithm [4], thus allowing for lower hardware overhead when designing a arithmetic unit performing the square root and division operations. After the necessary transformations of formula (8), the authors have arrived at expression (15) for the first iteration of the $y_1$ reciprocal value computation. The particular value is computed using the formula (16) only after the iteration process is completed, i.e. when the value $y_{n+1}$ is obtained.

The number of iterations needed to obtain the required result accuracy is known beforehand and depends on the choice of first approximation. When storing the initial conditions $y_0$ in the ROM a note should be taken to keep the difference between the operand bit width and the number of ROM address outputs minimal. Following the recommendations listed in paper [6], a generalized algorithm (Fig.1) for computing the real particular has been created.

The possibility of employing the Newton method for computing the complex particular in the double-channel DSP arithmetic units is described below. The authors have also described a way of using the real particular computation algorithm for implementing the complex number division operations by transforming the complex equation (17) into a system of two real equations (21) in the paper. At the initial stage, the double-channel DSP arithmetic unit equipped with two multiplication – storage units computes the R, I and D values. Then, in accordance to the algorithm (Fig.1), the corresponding channel computes the ReC and ImC.

## 5. CONCLUSION

Employing the proposed method in the DSPs allows to replace the resource-taxing division process with multiplication, having minimal hardware overhead which, in turn, allows to:

- develop an arithmetic unit with minimal hardware overhead;
- refrain from prolonging the processor's instruction cycle;
- obtain a better execution time in comparison to software implementation;
- perform the particular computation with the double-channel DSP arithmetic units;
- perform real-time execution of the core algorithm;

Besides that, the developed arithmetic unit employing the proposed particular computation algorithm can be used as a separate computing core element – generator in the Xilinx CAD system, etc.

Further research in this field shall be directed towards developing the new signal processor architectures capable of widening the spectrum of digital signal processing tasks solved by high-performance hardware-based systems.

## 6. REFERENCES

[1] Gold B., Rader C.M., *Didital Processing of Signals, Ch. 4.* McGraw–Hill, 1969.

[2] A.V. Palagin, M.V. Semotiuk, E.N. Chichirin, K.P. Sosnenko. Acoustic Commander – an integrated operation environment for measurement and calculation of acoustic properties. *Computers, networks and systems,* 2009, No. №8, pp. 3-10. (in Russian).

[3] Stefanelly R. A suggestion for a high-speed parallel binary divider. *IEEE Trans. Comput.,* 1972, v. C-21, No. 1, pp. 42-55.

[4] Y.E. Vizor. Accelerated implementation of non-linear function computing in signal processors and microcontrollers. *Informatisation and management problems,* 2009, No. 3, pp. 20-25. (in Russian).

[5] N.S. Bakhvalov. *Numeric methods.* Moscow: Nauka, 1975. (in Russian).

[6] A.V. Aho, J.E. Hopcraft. *The Design and Analysis of Computer Algorithms.* Addison-Wesley, Reading, Mass., 1974.