Bit-parallel algorithms and devices

Vyacheslav Khachumov

Program (software) systems institute RAS, Pereslavl-Zalessky, Russia, vmh@vmh.botik.ru

Abstract. Questions of development and research of new ways parallel and bit-parallel representation of the iterative computing Volder's algorithms, providing accelerated calculation of elementary functions and performance of operations of arithmetics and geometry are considered. Carefully analyzed are the methods of the multisequencing and correction. Modeling results are given.

Key Words. Geometrical processors, Volder's and Puhov's algorithms, rotation, vectoring, parallel calculations, bit-parallel form, group summation, compensation methods, processor unit.

1. INTRODUCTION

Specialized geometrical processors are widely used for solution of such tasks as conversion from the vector form in raster, polygon painting, affine transforms, multiplication, etc. The designing of such devices causes the problem of choice of computing algorithms possessing the effective hardware and software [1].

In this connection the problems are suggested to be solved in Volder's [2] and Meggitt's [3] algorithms. These algorithms can be successfully used for solving trajectory tasks, signal processing, calculations of elementary functions raster-vector scanning, interception of flat polygons and also for carrying out generalized geometric transformations and generation of flat curves. For these purpose it is well enough to have set of such operations: "multiplying-dividing", "vectoring", "rotation" and calculation of elementary functions

The technique of plotting of certain algorithms is fully enough represented in the above-listed works. Volder's and Meggitt's algorithms are known as a mode to possess iteration character. For the first time the task of multisequencing of Volder's algorithms at a level of external cycles was set in operation [4]. In general the parallel calculations may be only achieved at the level of separate formula constituting one iteration procedure. However, in a certain case, it's possible to construct calculating processes with the clipper level of multisequencing, for instance, for the operations "vectoring" and "rotation".

The best effect of acceleration is reached, as is known, in systems with digit-by-digit processing. In this connection calls the big interest the theory of the bit calculations offered by Puhov [5]. The mathematical device was offered to them, permitted to present binary result of calculation of the complex function as the bit-parallel form in which each bit of result expressed is independent through appropriate bits of argument. Thus, similar representations allow to receive at a minimum of calculations in finite sort ready result for each bit of the function.

2. BIT-PARALLEL FORMS

We shall therefore limit ourselves to consideration of some features resulting from the chosen system of base algorithms. Let us consider, for example, current Volder's relationships for the operation "rotation"[2]:

$\varphi_{i+1} = \varphi_i - \varepsilon_i \cdot \mathrm{arctg} 2^{-i}$, $\varepsilon_{i+1} = \mathrm{sign}\, \varphi_{i+1}$, (1)

$X_{i+1} = X_i + \varepsilon_i 2^{-i} \cdot Y_i$, $Y_{i+1} = Y_i - \varepsilon_i 2^{-i} \cdot X_i$, (2)

where: $\varphi_0 = \varphi$, $Y_n = Y$, $X_n = X$, $\varepsilon_0 = +1$, $y_0 = KY'$, $x_0 = KX'$; [Y,X],[Y',X'] - are correspondingly the old and the new coordinates of the vector end rotated through the angle φ; n- is the number of iterations determined by the maximal rating of number representation (being fixed); ε_i, i=0,..,n-1 are Volder's operators, K- is the coefficient of linear distortion, ($K \approx 1.64$).

The values of all operators belong to set {-1,+1}, are calculated in advance and stored in ROM (table 1).

Table 1. Result of calculation of operators

φ	ε_0	ε_1	ε_2	ε_3	ε_4	ε_5	ε_6	ε_7
10	1	-1	-1	1	-1	1	1	-1
20	1	-1	1	-1	-1	-1	1	-1
30	1	-1	1	-1	1	1	-1	1
40	1	-1	1	1	1	-1	-1	-1
50	1	1	-1	-1	-1	1	1	1
60	1	1	-1	1	-1	-1	1	-1
70	1	1	-1	1	1	1	-1	1
80	1	1	-1	1	-1	-1	1	1
90	1	1	1	1	-1	1	-1	-1

Parallel realization of "rotation" operation is of some interest. We have the result of multisequencing of initial relationships for "rotation" (n=8) accordingly:

Table2. Result of representation $X'=X_8$

2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}	2^{-7}	2^{-8}
				Basic		Part:	
B	$\varepsilon_1 A$	$\varepsilon_2 A$	$\varepsilon_3 A$	$\varepsilon_4 A$	$\varepsilon_5 A$	$\varepsilon_6 A$	$\varepsilon_7 A$
			$-\varepsilon_1\varepsilon_2 B$	$-\varepsilon_1\varepsilon_3 B$	$-\varepsilon_1\varepsilon_4 B$	$-\varepsilon_1\varepsilon_5 B$	$-\varepsilon_1\varepsilon_6 B$
					$-\varepsilon_2\varepsilon_3 B$	$-\varepsilon_2\varepsilon_4 B$	$-\varepsilon_2\varepsilon_5 B$
							$-\varepsilon_3\varepsilon_4 B$
						$-\varepsilon_1\varepsilon_2\varepsilon_3 A$	$-\varepsilon_1\varepsilon_2\varepsilon_4 A$
				Compen	sation	part:	
			B	$\varepsilon_1 A$	$\varepsilon_2 A$	$\varepsilon_3 A$	$\varepsilon_4 A$
						$-\varepsilon_1\varepsilon_2 B$	$-\varepsilon_1\varepsilon_3 B$
				B	$\varepsilon_1 A$	$\varepsilon_2 A$	$\varepsilon_3 A$
							$-\varepsilon_1\varepsilon_2 B$
						B	$\varepsilon_1 A$
							B

Table3. Result of representation $Y'=Y_8$

2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}	2^{-7}	2^{-8}
				Basic		Part:	
A	$-\varepsilon_1 B$	$-\varepsilon_2 B$	$-\varepsilon_3 B$	$-\varepsilon_4 B$	$-\varepsilon_5 B$	$-\varepsilon_6 B$	$-\varepsilon_7 B$
			$-\varepsilon_1\varepsilon_2 A$	$-\varepsilon_1\varepsilon_3 A$	$-\varepsilon_1\varepsilon_4 A$	$-\varepsilon_1\varepsilon_5 A$	$-\varepsilon_1\varepsilon_6 A$
					$-\varepsilon_2\varepsilon_3 A$	$-\varepsilon_2\varepsilon_4 A$	$-\varepsilon_2\varepsilon_5 A$
							$-\varepsilon_3\varepsilon_4 A$
						$\varepsilon_1\varepsilon_2\varepsilon_3 B$	$\varepsilon_1\varepsilon_2\varepsilon_4 B$
				Compen	sation	Part:	
			A	$-\varepsilon_1 B$	$-\varepsilon_2 B$	$-\varepsilon_3 B$	$-\varepsilon_4 A$
						$-\varepsilon_1\varepsilon_2 A$	$-\varepsilon_1\varepsilon_3 A$
				A	$-\varepsilon_1 B$	$-\varepsilon_2 B$	$-\varepsilon_3 B$
							$-\varepsilon_1\varepsilon_2 A$
						A	$-\varepsilon_1 B$
							A

where $A=X+\varepsilon_0 Y$, $B=Y-\varepsilon_0 X$.

There are bit-parallel forms for binary representation, where $A=a_1a_2...a_8$, $B=b_1b_2...b_8$, $Y'=y_1y_2...y_8$, $X'=x_1x_2...x_8$ are made from Tables 2,3 and can be written as:

$$
\begin{aligned}
x_1&=b_1, \qquad (3)\\
x_2&=b_2+\varepsilon_1 a_1,\\
x_3&=b_3+\varepsilon_1 a_2+\varepsilon_2 a_1,\\
x_4&=b_4+\varepsilon_1 a_3+\varepsilon_2 a_2+(\varepsilon_3 a_1-\varepsilon_1\varepsilon_2 b_1),\\
x_5&=b_5+\varepsilon_1 a_4+\varepsilon_2 a_3+(\varepsilon_3 a_2-\varepsilon_1\varepsilon_2 b_2)+(\varepsilon_4 a_1-\varepsilon_1\varepsilon_3 b_1),\\
x_6&=b_6+\varepsilon_1 a_5+\varepsilon_2 a_4+(\varepsilon_3 a_3-\varepsilon_1\varepsilon_2 b_3)+(\varepsilon_4 a_2-\varepsilon_1\varepsilon_3 b_2)+((\varepsilon_5 a_1-(\varepsilon_1\varepsilon_4+\varepsilon_2\varepsilon_3)b_1),\\
x_7&=b_7+\varepsilon_1 a_6+\varepsilon_2 a_5+(\varepsilon_3 a_4-\varepsilon_1\varepsilon_2 b_4)+(\varepsilon_4 a_3-\varepsilon_1\varepsilon_3 b_3)+(\varepsilon_5 a_2-(\varepsilon_1\varepsilon_4+\varepsilon_2\varepsilon_3)b_2)+((\varepsilon_6-\varepsilon_1\varepsilon_2\varepsilon_3)a_1-(\varepsilon_1\varepsilon_5+\varepsilon_2\varepsilon_4)b_1),\\
x_8&=b_8+\varepsilon_1 a_7+\varepsilon_2 a_6+(\varepsilon_3 a_5-\varepsilon_1\varepsilon_2 b_5)+(\varepsilon_4 a_4-\varepsilon_1\varepsilon_3 b_4)+(\varepsilon_5 a_3-(\varepsilon_1\varepsilon_4+\varepsilon_2\varepsilon_3)b_3)+((\varepsilon_6-\varepsilon_1\varepsilon_2\varepsilon_3)a_2-(\varepsilon_1\varepsilon_5+\varepsilon_2\varepsilon_4)b_2)+((\varepsilon_7-\varepsilon_1\varepsilon_2\varepsilon_4)a_1-(\varepsilon_1\varepsilon_6+\varepsilon_2\varepsilon_5+\varepsilon_3\varepsilon_4)b_1).
\end{aligned}
$$

$$
\begin{aligned}
y_1&=a_1, \qquad (4)\\
y_2&=a_2-\varepsilon_1 b_1,\\
y_3&=a_3-\varepsilon_1 b_2-\varepsilon_2 b_1,\\
y_4&=a_4-\varepsilon_1 b_3-\varepsilon_2 b_2-(\varepsilon_3 b_1+\varepsilon_1\varepsilon_2 a_1),\\
y_5&=a_5-\varepsilon_1 b_4-\varepsilon_2 b_3-(\varepsilon_3 b_2+\varepsilon_1\varepsilon_2 a_2)-(\varepsilon_4 b_1+\varepsilon_1\varepsilon_3 a_1),\\
y_6&=a_6-\varepsilon_1 b_5-\varepsilon_2 b_4-(\varepsilon_3 b_3+\varepsilon_1\varepsilon_2 a_3)-(\varepsilon_4 b_2+\varepsilon_1\varepsilon_3 a_2)-(\varepsilon_5 b_1+(\varepsilon_1\varepsilon_4+\varepsilon_2\varepsilon_3)a_1),\\
y_7&=a_7-\varepsilon_1 b_6-\varepsilon_2 b_5-(\varepsilon_3 b_4+\varepsilon_1\varepsilon_2 a_4)-(\varepsilon_4 b_3+\varepsilon_1\varepsilon_3 a_3)-(\varepsilon_5 b_2+(\varepsilon_1\varepsilon_4+\varepsilon_2\varepsilon_3)a_2)+((\varepsilon_1\varepsilon_2\varepsilon_3-\varepsilon_6)b_1-(\varepsilon_1\varepsilon_5+\varepsilon_2\varepsilon_4)a_1),\\
y_8&=a_8-\varepsilon_1 b_7-\varepsilon_2 b_6-(\varepsilon_3 b_5+\varepsilon_1\varepsilon_2 a_5)-(\varepsilon_4 b_4+\varepsilon_1\varepsilon_3 a_4)-(\varepsilon_5 b_3+(\varepsilon_1\varepsilon_4+\varepsilon_2\varepsilon_3)a_3)+((\varepsilon_1\varepsilon_2\varepsilon_3-\varepsilon_6)b_2-(\varepsilon_1\varepsilon_5+\varepsilon_2\varepsilon_4)a_2)+((\varepsilon_1\varepsilon_2\varepsilon_4-\varepsilon_7)b_1-(\varepsilon_1\varepsilon_6+\varepsilon_2\varepsilon_5+\varepsilon_3\varepsilon_4)a_1).
\end{aligned}
$$

The value of each element ε_i (i=0,..,7) for "vectoring" can be generating, for example from equations (4), when all $y_i=0$ (vector is on OX axis). On condition that $a_1=0$, $b_1=1$:

$$
\begin{aligned}
\varepsilon_0&=0, \qquad (5)\\
\varepsilon_1&=a_2,\\
\varepsilon_2&=a_3-\varepsilon_1 b_2,\\
\varepsilon_3&=a_4-\varepsilon_1 b_3-\varepsilon_2 b_2-\varepsilon_1\varepsilon_2 a_1,\\
\varepsilon_4&=a_5-\varepsilon_1 b_4-\varepsilon_2 b_3-\varepsilon_3 b_2-\varepsilon_1\varepsilon_2 a_2-\varepsilon_1\varepsilon_3 a_1,\\
\varepsilon_5&=a_6-\varepsilon_1 b_5-\varepsilon_2 b_4-\varepsilon_3 b_3-\varepsilon_1\varepsilon_2 a_3-\varepsilon_4 b_2-\varepsilon_1\varepsilon_3 a_2-(\varepsilon_1\varepsilon_4+\varepsilon_2\varepsilon_3)a_1,\\
\varepsilon_6&=a_7-\varepsilon_1 b_6-\varepsilon_2 b_5-\varepsilon_3 b_4-\varepsilon_1\varepsilon_2 a_4-\varepsilon_4 b_3-\varepsilon_1\varepsilon_3 a_3-\varepsilon_5 b_2-(\varepsilon_1\varepsilon_4+\varepsilon_2\varepsilon_3)a_2+\varepsilon_1\varepsilon_2\varepsilon_3 b_1-(\varepsilon_1\varepsilon_5+\varepsilon_2\varepsilon_4)a_1,\\
\varepsilon_7&=a_8-\varepsilon_1 b_7-\varepsilon_2 b_6-\varepsilon_3 b_5-\varepsilon_1\varepsilon_2 a_5-\varepsilon_4 b_4-\varepsilon_1\varepsilon_3 a_4-\varepsilon_5 b_3-(\varepsilon_1\varepsilon_4+\varepsilon_2\varepsilon_3)a_3+(\varepsilon_1\varepsilon_2\varepsilon_3-\varepsilon_6)b_2-(\varepsilon_1\varepsilon_5+\varepsilon_2\varepsilon_4)a_2+\varepsilon_1\varepsilon_2\varepsilon_4 b_1-(\varepsilon_1\varepsilon_6+\varepsilon_2\varepsilon_5+\varepsilon_3\varepsilon_4)a_1.
\end{aligned}
$$

Thus, the set of operators $\{\varepsilon_0,..,\varepsilon_7\}$ can be calculated in a parallel way and be stored in ROM. Parallel calculation of the expressions (3) obtained is perform by means of algorithms and devices of group summation. The architecture bit-parallel devices is shown on fig 1

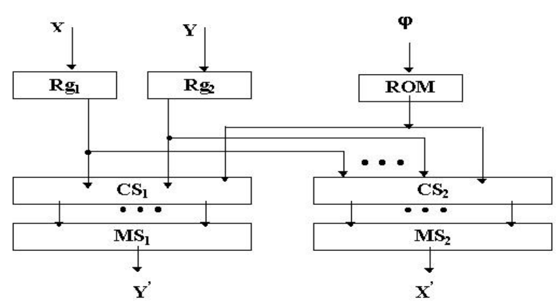

Fig.1

The device contains: sets of registers Rg1, Rg2 for parallel shift and storage of initial operands X, Y; ROM for storage and outputs of the operators appropriate to a given corner of turn on the basis of table1.; combinative schemes CS1 and CS2, forming signs and additional codes of addends according to table 2 and table 3.; multiport summators MS1, MS2 for execution of the operation of group summation.

3. COMPENSATION METHODS

The other feature is connected with obtaining effective compensation methods of linear distortions which are the attributes of Volder's operations of "rotation" and "vector". For the operation of "rotation" the following method of correction of results is suggested. The idea is to use iteration procedure for each transformed coordinate separately and to introduce the correcting angles. It is assumed here that the initial angle of vector position φ' associated with the position of the point [Y,X] is additionally known. Using Volder's procedure "rotation", for example, we turn vector to the angles $\varphi_1=\varphi+d\varphi_1$ and to $\varphi_2=\varphi+d\varphi_2$, where the correcting angles $d\varphi_1$, $d\varphi_2$ are found according to the:

$d\varphi_1=\arccos(\cos(\varphi+\varphi')/K)-(\varphi+\varphi')$,

$d\varphi_2=\arcsin(\sin(\varphi+\varphi')/K)+(\varphi+\varphi')$.

The demand for initial determination of angle of vector position undoubtedly decreases the practical meaning of distortion compensation, however in some cases the method appears quite helpful, for instance, in organizing smooth rotation of a graphical object. Other methods are using multiply table 3,4 elements on coefficient 1/K (compensation part), or duplication of some iterations until $K\approx 2$ (Table 4).

Table 4. Result of simulation of coefficient K

P_i	$P_2=2$ $P_4=0$ $P_6=0$	$p_2=4$ $p_4=0$ $p_6=0$	$p_2=6$ $p_4=0$ $p_6=0$	$p_2=6$ $p_4=2$ $p_6=0$	$p_2=6$ $p_4=4$ $p_6=0$	$P_2=6$ $P_4=6$ $P_6=0$	$p_2=6$ $p_4=6$ $p_6=2$
K	1.7497	1.8590	1.9752	1.9829	1.9907	1.9985	1.9990

P_i -is the number of the Volder's i-iteration duplication.

4. CONSTRUCTION OF A PROCESSOR UNIT

Bit-parallel representations were obtained practically for all main Volder operations. It has allowed to construct structure of uniform processor unit (PU) with a set of large operations.

Two alternate variants PU are offered, one of which bases only on updated Volder algorithms, and another uses in addition Puhov's algorithms. Puhov's algorithms are more effective and compact for operations such as 1/x and $\sqrt{x}$, while Volder's algorithms are good for execution of geometrical operations

Therefore it is expedient to unite in uniform PU the best sides of both directions. Choice of the concrete operation is realized in PU by appropriate customization of its structure. Programming of structure is carried out by commutation of separate components PU.

Shared use of updated Volder's and Puhov's procedures allows to build bit-parallel PU on a uniform methodological basis. PU it is hardware supports the specialized language with which help the user can make structured programming of algorithms of a machine graphics.

5. REFERENCES

1. Khachumov V.M. Geometrical processor with reduced instruction set computing. //Proc. of the 5-th International Conference of Computer Graphics and Visualization (GRAPHICON'95).-St.Peterburg: GRAFO, 1995, vol.2, pp.43.-44.

2.Volder J.E. The CORDIC trigonometric computing technique. //Computer design Development.-1976.- N.-Y.- pp.301-307.

3.Meggitt J.E. Digit by digit methods for polynomials.- IBM J. Res.a. Develop., 1963, vol.7, N 3, pp.237-245.

4.Naseem A. Fisher D.P. The modified cordic algorithm // Proceeding IEEE Intern. Conference on computer design; VLSI in computers, 1985.-pp.144-152.

5.Puhov G.E., Evdokimov V.F., Sinkov M.V. Razrjadno-analogue computing systems. M.: Sov. Radio, 1978.-256 p.

Сведения об авторе

Хачумов Вячеслав Михайлович
Д.т.н, проф.
Зав.лабораторией Института программных систем РАН

152020, Ярославская обл., г. Переславль-Залесский, м. Ботик, ИПС РАН
тел.: (085-35)98030, E-mail: vmh@vmh.botik.ru