Power Consumption Reduction with the E-method Algorithm

Romain Michard

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Power Consumption Reduction with the E-method Algorithm

Romain Michard

N° 7323

June 2010

ISSN 0249-6399 ISRN INRIA/RR--7323--FR+ENG

INSTITUT NATIONAL
DE RECHERCHE
EN INFORMATIQUE
ET EN AUTOMATIQUE

INRIA

centre de recherche
GRENOBLE - RHÔNE-ALPES

Power Consumption Reduction with the E-method Algorithm

Romain Michard*

Thème : COM
Équipe-Projet Swing

Rapport de recherche n° 7323 — June 2010 — 10 pages

Abstract: This work is about the reduction of the electrical consumption of the selection function in the polynomial$\{-3, -2, -1, 0, 1, 2, 3\}$ evaluation algorithm known as the E-method and presented by M.D. Ercegovac in the 70s. A choice is possible when selecting the result digits in a redundant number representation, it allows a power consumption reduction. The statistical study of the possible gains has already been made and this work completes that first study by a real consumption evaluation and figures.

Key-words: computer arithmetic, low energy consumption, polynomial evaluation, E-method

* INRIA, INSA Lyon, CITI, F-69621, France - romain.michard@inria.fr

Centre de recherche INRIA Grenoble – Rhône-Alpes
655, avenue de l'Europe, 38334 Montbonnot Saint Ismier
Téléphone : +33 4 76 61 52 00 — Télécopie +33 4 76 61 52 52

Réduction de la consommation énergétique avec l'algorithme de E-méthode

Résumé : Ce travail porte sur l'étude de consommation énergétique liée à la fonction de sélection dans l'algorithme d'approximation de polynômes connu sous le nom de E-méthode proposé par M.D. Ercegovac dans les années 70. La latitude de choix dans la fonction de sélection des chiffres du résultat, en représentation redondante, permet d'envisager de limiter l'activité électrique dans certains cas. L'étude statistique des gains envisageables dans ce cadre a déjà été présentée et ce travail la complète par une véritable étude pratique et chiffrée.

Mots-clés : arithmétique des ordinateurs, basse consommation d'énergie, évaluation de polynôme, E-méthode.

Contents

$$\begin{pmatrix} 1 & -x & 0 & & \cdots & & & 0 \\ 0 & 1 & -x & 0 & & \cdots & & 0 \\ 0 & 0 & 1 & -x & 0 & & \cdots & 0 \\ & & \ddots & \ddots & \ddots & \ddots & & \vdots \\ & & & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & & & & \ddots & \ddots & \ddots & 0 \\ & & & & & & 1 & -x \\ 0 & & & \cdots & & & 0 & 1 \end{pmatrix} \begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ \vdots \\ \vdots \\ y_{n-1} \\ y_n \end{pmatrix} = \begin{pmatrix} p_0 \\ p_1 \\ p_2 \\ \vdots \\ \vdots \\ \vdots \\ p_{n-1} \\ p_n \end{pmatrix} \quad (1)$$

1 Introduction

In many applications it is necessary to evaluate functions (some are complicated ones) using only simple operators as additions and multiplications. Algebraic functions ($1/x, x/y, \sqrt{x}\ldots$) and elementary ones ($\sin(x)$, $\cos(x)$, $\exp(x)$, $\log(x)$, $\arctan(x)\ldots$) could be easily approximated by polynomials [1] by using (for example) the Remes algorithm [2].

The E-method [3, 4] makes it possible to evaluate polynomials with simple shifts and additions only. The result digits are then iteratively determined by the selection function from the former digits and a residual (as the partial remainder in a division). This report extends the work presented in [5]. It performs consumption analyses when the former work was only theoretical.

Section 2 presents the considered algorithms. Section 3 is giving a small scientific background to understand the consumption matter. Section 4 is explaining how this work was done. Section 5 is showing what results could be attained by this method. Finally section 6 is shortly concluding this work.

2 Algorithms

In this section both the original E-method algorithm as explained in [3, 4] and the improvement of [5] concerning a possible power consumption reduction are presented.

2.1 E-method

The *evaluation method* was proposed by M.D. Ercegovac in the 70s [3, 4]. This method makes it possible to solve linear diagonally dominant systems with a simple iteration based on a shift-and-add algorithm. Such targeted systems are presented in the equation (1). The main interest of this method is to allow a polynomial evaluation without any multiplier circuit using a shift-and-add algorithm such as the SRT division one [6, 7]. This allows to think about small operators with a small static energy consumption. Some other algorithms exist for the algebraic/elementary functions evaluation without any multiplier (as the multipartite tables method [8]) but this methods are often dedicated to one unique function.

In this system $\mathcal{A}$ is the matrix, b is the left vector and y is the solution vector. The system size is $n+1$. After its resolution the first y component is the value of the polynomial P evaluated at x. P's degree is n and its coefficients are the components of b. That's to say the solution of $\mathcal{A}y = b$ is $y = [y_0, y_1, \ldots, y_n]^t$ such as

$$y_0 = P(x) = p_n x^n + p_{n-1} x^{n-1} + \cdots + p_0$$

In [3, 4] are the limits of the $P(x)$'s coefficients possible values and of x's ones. These constraints can have a limited impact due to some technical possibilities presented in [9].

Before presenting the E-method iteration some notations should be introduced. It will be useful for the following. The number representation radix is noted β ($\beta = 4$ in this work but the results could be extended to other radices). The residual vector (its size is $n+1$) is w. The different values of a quantity during the time are represented with the square brackets. For example $w[j]$ is the residual vector at the iteration number j. The result digits vector at iteration j is noted $d[j]$. If the radix is 4 those digits can be in the set $\{-2, -1, 0, 1, 2\}$ or in $\{-3, -2, -1, 0, 1, 2, 3\}$.

The E-method algorithm is presented on Figure 1. The residual vector is initialized with the coefficients of the polynomial P (the b vector). The first result vector is the null one.

1 *initialization :*
2 $w[0] \longleftarrow b$
3 $d[0] \longleftarrow 0$
4 *iteration :*
5 **for** j **from** 1 **to** m **do**
6 $w[j] \longleftarrow \beta \times \Big(w[j-1] - \mathcal{A} \times d[j-1] \Big)$
7 $d[j] \longleftarrow S(w[j])$
8 *result :*
9 $y_0[m] = \sum_{i=1}^{m} d_0[i] \beta^{-i}$

Figure 1: E-method algorithm for a polynomial evaluation.

As every shift-and-add algorithm the E-method produces one result digit at each iteration starting from the most significant one. The concatenation of these digits tends towards the mathematical value of the result when the time tends towards infinity. Here each iteration gives a digits vector, $d[j]$. The calculation is cut into stages where the calculation of a line of the system (1) is done by a stage. The $d[j]$'s digits are then propagated over the stages and the final result $P(x)$ is the output of the last stage. The E-method iteration is similar to a division where the "divisor" would be $\mathcal{A}$ (w would be the partial remainder). For each line $i = 0, \ldots, n-1$ of the matrix $\mathcal{A}$ the computation is:

$$w_i[j] = \beta\big(w_i[j-1] - d_i[j-1] + d_{i+1}[j-1]x\big) \tag{2}$$

For $i = n$ the computation is simplified: $w_n[j] = \beta\big(w_n[j-1] - d_n[j-1]\big)$.

In the equation (2) $d_{i+1}[j-1]x$ is only a multiplication of a digit ($d_{i+1}[j-1]$) by a number (x). In practice this " small multiplication" is done by the selection

of the right x's multiple to add among the different possibilities. This is possible because the multiplicand x is constant for the duration of the whole algorithm.

The computation of the new residual digit is only requiring additions/subtractions and multiplications of a number by a digit (and a small one, here it is smaller than 3). A new result digits vector $d[j]$ is produced each iteration. This computation is done using the selection function S defined in a general case by (3) where ρ is the maximal value allowed for the result digits with a redundant notation ($\rho = 3$ in this case). Some details can be found in [3, 4].

$$S(x) = \begin{cases} \text{signe } x \, \lfloor |x| + 1/2 \rfloor \,, & \text{si } |x| \leq \rho \\ \text{signe } x \, \lfloor |x| \rfloor \,, & \textit{sinon}, \end{cases} \tag{3}$$

2.2 A new selection function

In order to decrease the activity of the addition in the computation of $w[j] \longleftarrow \beta \times \Big(w[j-1] - \mathcal{A}d[j-1] \Big)$ a new selection function has been proposed in [5]. It uses a memory to try and make the addition constant when possible. In the values ranges where the selection function can propose two different values for the new digit (iteration j) the last returned digit (iteration $j-1$) is selected if it is possible. This selection is illustrated by Figure 2.

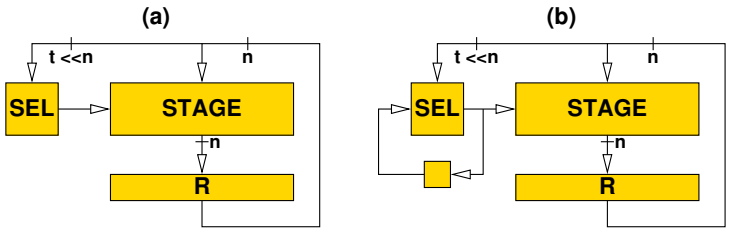

Figure 2: Computation without (a) or with (b) the memory.

The selection function is simple enough to be implemented as a cheaper operator than a table. The original function uses the redundancy to look only at the integer part of the residual ($E(w)$) and at the first bit of the fractional part ($F(w)$). In the new function (with a memory) the redundancy is used to allow a larger choice in the selection in order to try and decrease the electrical activity ($Sel(w[j]) = Sel(w[j-1])$). Then more bits of $F(w)$ have to bee looked at (3 in this case). Figure 3 is illustrating the selection function for both cases.

To check the hardware cost of the memory the new operator has been synthesized on Virtex FPGA circuits for a degree-4, 32-bit precision polynomial and comparisons have been made with the original E-method operator. The results are presented in Table 1. The selection function modification requires a 9% area increase but the critical path is then cut and the global clock can be 4% faster.

3 Power consumption background

For the power consumption purpose it is useful to clarify certain notions. For example it is important to talk about the circuit frequency and to have a clear

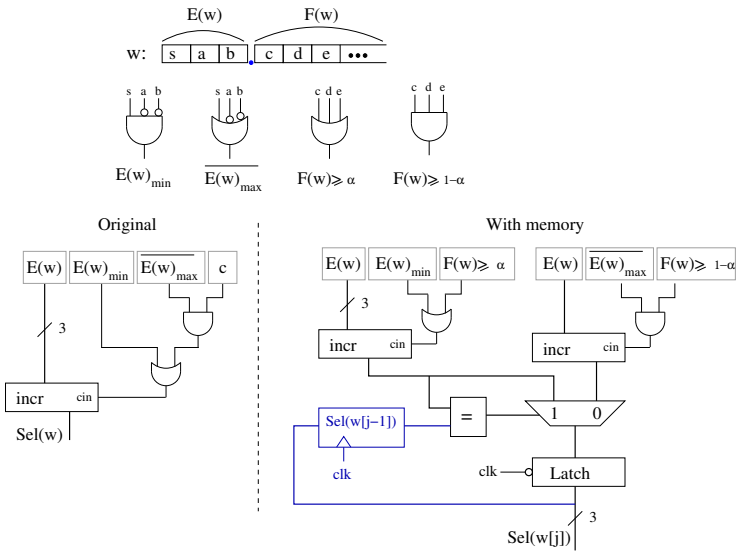

Figure 3: Selection function without (a) or with (b) the memory for $\beta = 4$ and the digits in $\{-3, -2-, 1, 0, 1, 2, 3\}$.

Method	Synthesis effort	Size [# slices]	Period [ns]
Without memory	area	423	19.7
	speed	750	16.4
With memory	area	461	18.9
	speed	812	16.8

Table 1: Synthesis results.

idea of the different terms as power and energy. This section tries to give the necessary background concerning this matter.

3.1 Power/Energy/Batteries

First it is important to clearly differentiate two notions that are often mixed up: the power and the energy. They are linked (as the energy is the product of the power and the time) but definitely not the same. In order to explain the difference it is useful to think about batteries when speaking about the consumption. The power is an instant value to be compared with the power of a battery (in W or in A). The energy is to be compared with the capacity of a battery (in $W \cdot h$, in $A \cdot h$ or in J), it is the quantity of energy that is contained into the battery and that can be consumed before recharging it.

As an example we can compare precisely different operators: an operator requiring twice the power of another one can consume the same amount of energy per operation if it is twice faster. If it is requiring half the power but twice the computation duration it consumes the same amount of energy again.

3.2 Frequency

Then the frequency can be talked about. it is directly linked to the energy consumption. As explained previously an operator running with twice the frequency of a second one can consume the same amount of energy per operation if it is requiring twice the power of the first one. It is easy to understand that to make exactly the same computation, but in more time, is not requiring so much energy.

All this is illustrated on Table 2. This table is intended for showing the behavior of the different values depending on each other. There is no unit and the figures are not relevant.

	Power	**Frequency**	**Duration**	**Energy**
Operator 1	2	2	2	4
Operator 2	1	2	2	2
Operator 3	2	4	1	2
Operator 4	1	1	4	4

Table 2: Illustration with no unit and no real figures.

Those notions are to be carefully handled when dealing with power consumption as it is easy to confuse everything and to obtain non-relevant results.

4 Methodology

Different operators for different algorithms are to be studied in order to obtain a meaningful comparison. First the Horner algorithm is implemented in a studied operator as it is a basic algorithm when speaking of polynomials evaluations. Then operators dedicated to the original E-method [4] have been considered to know how the modification was improving/deteriorating the performances. Finally operators with the modification proposed in [5] have been designed.

The procedure used in this work is presented in [10]. Everything is done exactly the same way. Tools are used in a command-line manner by scripts. That makes the synthesis, simulation (post-place&route one is needed) and power estimation possible.

The CAD tools used are Xilinx ISE 11 (for synthesis and power estimation) and ModelSim 6.6a (for simulation).

5 Results

Figure 4 is showing how different operators are consuming power. The modified E-method one presented in [5] is consuming 42% less than an original E-method operator (which is consuming 16% more than a Horner one) what is a better activity reduction than the statistical study of [5] could have make expectable (30%).

Figure 4 is showing that the 4–2 algorithm consumes less power than the 4–3 one (see [5] for the notations). It seems logical as the more reduced the digit set

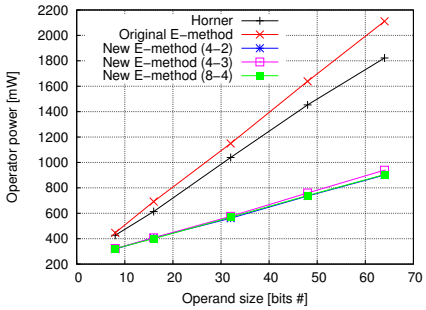

Figure 4: Results for several evaluation algorithms.

is the more frequent the stability of the new quotient digit is (hence the stability of the part computing the addition for the partial remainder). Moreover when the 3 multiple of the "quotient" is needed to be computed as a sum of different terms whereas in the 4–2 version only the 1 and 2 multiples are needed and they are easily obtained by shifts.

The 8–4 version of the algorithm only needs the 4 multiple to be computed more and it only requires another shift. It's very simple to compute and that's explaining there is no difference when using the 8–4 algorithm. Moreover the 8–4 algorithm is computing 3 binary digits at each iteration when the 4–2 and 4–3 ones are computing 2 such digits. Then the 8–4 is requiring less iterations and thus less energy. The counterpart (there must be one !) is that the circuit is larger due to a more complicated selection function.

6 Conclusion

After the theoretical study of [5], these results show that an important power consumption reduction is possible in an E-method operator by adding a small part in the selection function. The (*power consumption*) × (*hardware cost*) tradeof is then a good one and (depending of the main interest of the designer) this improvement could be very interesting.

This algorithm (with the presented improvement) is a good method to evaluate functions as it doesn't require any multiplier which is very expensive for the circuit area.

Acknowledgement

The author would like to thank Arnaud Tisserand for his advises and his help concerning the synthesis script edition.

References

[1] J.-M. Muller, *Elementary Functions: Algorithms and Implementation*. Birkhäuser, Boston, 1997.

[2] E. Remes, "Sur un procédé convergent d'approximations successives pour déterminer les polynômes d'approximation," *C.R. Acad. Sci. Paris*, vol. 198, pp. 2063–2065, 1934.

[3] M. Ercegovac, "A general method for evaluation of functions and computation in a digital computer," Ph.D. dissertation, Dept. of Computer Science, University of Illinois, Urbana-Champaign, 1975.

[4] ——, "A general hardware-oriented method for evaluation of functions and computations in a digital computer," *IEEE Transactions on Computers*, vol. C-26, no. 7, pp. 667–680, 1977.

[5] R. Michard, A. Tisserand, and N. Veyrat-Charvillon, "Étude statistique de l'activité de la fonction de sélection dans l'algorithme de E-méthode," in *5^{ième} journées d'études Faible Tension Faible Consommation (FTFC)*, Paris, May 2005, pp. 61–65.

[6] M. Ercegovac and T. Lang, *Division and Square-Root Algorithms: Digit-Recurrence Algorithms and Implementations*. Kluwer Academic, 1994.

[7] M. Ercegovac, J. Muller, and A. Tisserand, "FPGA implementation of polynomial evaluation algorithm," in *Field Programmable Gate Arrays for Fast Board Development and Reconfigurable Computing*, SPIE, Ed., vol. 2607, Oct. 1995, pp. 177–188.

[8] F. de Dinechin and A. Tisserand, "Some improvements on multipartite tables methods," in *15th International Symposium on Computer Arithmetic ARITH15*, N. Burgess and L. Ciminiera, Eds. Vail, Colorado: IEEE, June 2001, pp. 128–135.

[9] N. Brisebarre and J.-M. Muller, "Functions approximable by e-fractions," in *38th Conference on signals, systems and computers*, Pacific Grove, California, US, Nov. 2004.

[10] R. Michard, "Scripts Use for the Synthesis and the Simulation of VHDL Circuits and Evaluation of the Power Consumption," INRIA, Technical Report RT-0381, Feb. 2010. [Online]. Available: http://hal.archives-ouvertes.fr/inria-00453250/PDF/RT-0381.pdf

Centre de recherche INRIA Grenoble – Rhône-Alpes
655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)

Centre de recherche INRIA Bordeaux – Sud Ouest : Domaine Universitaire - 351, cours de la Libération - 33405 Talence Cedex
Centre de recherche INRIA Lille – Nord Europe : Parc Scientifique de la Haute Borne - 40, avenue Halley - 59650 Villeneuve d'Ascq
Centre de recherche INRIA Nancy – Grand Est : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex
Centre de recherche INRIA Paris – Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex
Centre de recherche INRIA Rennes – Bretagne Atlantique : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex
Centre de recherche INRIA Saclay – Île-de-France : Parc Orsay Université - ZAC des Vignes : 4, rue Jacques Monod - 91893 Orsay Cedex
Centre de recherche INRIA Sophia Antipolis – Méditerranée : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399