



HAL
open science

40 Gop/S/mm² Fixed-Point Operators for Brain Computer Interface in 65nm CMOS

Erwan Libessart, Matthieu Arzel, Cyril Lahuec, Francesco Andriulli

► **To cite this version:**

Erwan Libessart, Matthieu Arzel, Cyril Lahuec, Francesco Andriulli. 40 Gop/S/mm² Fixed-Point Operators for Brain Computer Interface in 65nm CMOS. 2018 IEEE International Symposium on Circuits and Systems (ISCAS), May 2018, Florence, Italy. 10.1109/ISCAS.2018.8351028 . hal-01813164

HAL Id: hal-01813164

<https://hal.science/hal-01813164>

Submitted on 12 Jun 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

40 Gop/s/mm² fixed-point operators for Brain Computer Interface in 65 nm CMOS

Erwan Libessart*, Matthieu Arzel*, Cyril Lahuec* and Francesco Andriulli†

*Electronics Department, IMT Atlantique, France

E-mail: firstname.lastname@imt-atlantique.fr

†Campi Elettromagnetici, Politecnico di Torino, Italia
francesco.andriulli@polito.it

Abstract—The performance of non-invasive Brain-Computer Interface (BCI) depends on the computing performance of the system which solves the inverse problem. So the number of basic operations computed per second determines the BCI's resolution. Architecture with pipelined and parallelized flow is then required, and each operator in this architecture must be optimised to reach the highest possible computing performance. This paper presents the implementation of a fixed-point reciprocal and an inverse square root operators for the STMicroelectronics 65 nm CMOS technology. This paper follows previous works that optimise these operators on FPGA target. Each operator reaches a computing performance of about 40 Gop/s/mm², which improves the literature results by a factor of four. Thus, this works fits well for portable and high performance BCI applications.

I. INTRODUCTION

A Brain Computer Interface (BCI) is a hardware and software system that allows controlling computer or other devices with cerebral activity. The invasive BCI has direct access to the brain signals but is expensive and requires a surgical operation. So, this method is not appropriate to equip many people with this technology. Electroencephalogram (EEG) permits to perform a non-invasive BCI but this method has low resolution if the measured signals are directly used to control a computer. The EEG inverse problem can then be solved in order to determine, from these signals, which dipoles are active in the brain. The larger the number of considered dipoles is, the better the accuracy is, but also the larger the computation time is. For real-time, portable and accurate BCI applications, dedicated circuits have to be designed in a parallelized and pipelined way, so that they provide high computing performance, while minimizing the silicon area. Moreover this requires to process data not with the too complex floating-point format but with the fixed-point format.

Field-Programmable Gate Array (FPGA) devices are great for prototyping proof of concepts and validating algorithm implementations on hardware. Nevertheless, Application-Specific Integrated Circuits (ASICs) reach higher computing frequency and provide a better area integration. Any operator implementation may be optimised in different ways when

This work was supported by Labex CominLabs through SABRE project and by région Bretagne.

compared to the original FPGA implementation. Indeed, ASICs provide access to custom designs, which offer different implementation strategies and foster more adequate operator architectures.

The EEG inverse problem algorithm contains operations such as division or square root [1]. In the case of a pipelined architecture, each operator must be optimised to provide the best possible computing frequency. Reciprocal and inverse square root are intermediate steps to compute division and square root. The implementation of these operators is not straightforward and requires special care due to application's constraints, for both FPGA and ASIC implementations. Several methods to compute reciprocal and inverse square root can be pipelined, for example Look-Up Tables (LUTs) or the CORDIC algorithm [2]. These solutions may have some issues like necessary resources, latency or computing frequency when the size of the input increases. The iterative Newton-Raphson method is favored for floating-point implementations since it takes advantage of the mantissa value being in the interval $[1, 2[$. Indeed, this knowledge on the input allows the initial value to be easily computed and then to roughly double the number of bits of accuracy in each iteration. Newton-Raphson method can be used for both reciprocal and inverse square root [3], [4]. Its implementation usually relies on a memory block used to store coefficients required for the first step of the algorithm. In fixed point format, these operators require special attention, since the input values do not benefit from the mantissa property and thus require a specific computation for the first approximation. Previous work presented Scaling-Less Newton-Raphson (SLNR) architectures [5], [6] designed to maximise the computing frequency on Xilinx FPGAs but which are not optimised for an ASIC implementation.

In this paper, ASIC implementations of high computing performance fixed-point reciprocal and inverse square root operators for the STMicroelectronics 65 nm CMOS are presented. These operators reach the frequency of 1.587 GHz and occupy less than 40 000 μm^2 , which corresponds to a computing performance of 41 Giga operations per second and per mm^2 (Gop/s/mm²). So the proposed architecture is a good answer to

the high-performance BCI constraints. This paper is organized as follows. Section II presents the Newton-Raphson method for both operators and its possible implementations with the impact on the maximum clock frequency on Xilinx FPGA. Different Newton-Raphson solutions for the STMicroelectronics 65 nm CMOS target are described in Section III. Then, a comparison with another 65 nm CMOS reciprocal architecture is done in Section IV. Finally, Section V concludes the paper.

II. RELATED WORK

The Newton-Raphson algorithm can be used to compute the reciprocal or the inverse square root of a number a . This iterative algorithm doubles the number of bits of accuracy during each iteration. The final result x_n is obtained after n iterations of these equations:

$$\text{Reciprocal: } x_{i+1} = x_i(2 - ax_i) \quad (1)$$

$$\text{Inverse square root: } x_{i+1} = \frac{x_i}{2}(3 - ax_i^2) \quad (2)$$

The value x_0 is the first approximation of the desired result and is provided as input with a . Thus, x_0 has to be determined before computing the iteration equation. The usual method involves scaling the input a in a predetermined interval as $[1, 2[$, or $[0.5, 1[$ [3], [7]. Then, x_0 can be calculated by using coefficients stored in a memory block. This yields a good accuracy for the first approximation and then decreases the number of required Newton-Raphson iterations. In previous work, it was showed that this strategy is not the best to maximise the operating frequency on Xilinx FPGAs [5], [6]. These Scaling-Less Newton-Raphson (SLNR) architectures do not need any coefficients, so memory block is not required. This lack of memory block allows increasing the computing frequency because of hardware target features. In fact, on Xilinx FPGAs, the maximum frequency of BRAMs is lower than one of DSP cells, used for the multiplications. Table I shows the difference on Virtex-7 FPGAs, which rely on the 28 nm technology [8]. Therefore coefficients must be avoided

TABLE I
MAXIMAL FREQUENCY COMPARISON BETWEEN DSP CELL AND BRAM
ON VIRTEX-7 FAMILY FPGAS

	Max frequency
DSP48E1	740 MHz
BRAM	601 MHz

to generate the first approximation x_0 . The method proposed in [5], [6] is based on the Leading One Detector (LOD) of a . A simple combinatory circuit can then be used to determined a first approximation that respects the conditions of convergence :

$$\text{Reciprocal: } 0 < a \times x_0 < 2. \quad (3)$$

$$\text{Inverse square root: } 0 < a \times x_0^2 < 3. \quad (4)$$

The accuracy of the first approximation determines the number of required Newton-Raphson iterations. So the products in the conditions of convergence have to be as close as possible to the numerical value 1. Finally, the SLNR architectures ensure this accuracy for the first approximation:

$$\text{Reciprocal: } 0.875 < a \times x_0 < 1.125. \quad (5)$$

$$\text{Inverse square root: } 0.5 < a \times x_0^2 < 1.125. \quad (6)$$

FPGA implementation results for both operators are presented in Table II. The critical path in these architectures is located in the DSP cells for multiplication and gives a 740 MHz computing frequency on a Virtex7-690T. So the SLNR designs fit with high throughput applications on Xilinx FPGAs. This maximum value of 740 MHz is due to the physical structure of the FPGA. Higher computing frequency could be profitable to BCI applications. This explains the desire to adapt SLNR architectures to the ASIC technology.

TABLE II
IMPLEMENTATION RESULTS: 16-BIT RECIPROCAL AND INVERSE SQUARE
ROOT ON VIRTEX-7 690T

Operator	Reciprocal	Inverse square root
LUT	111	306
Flip-Flop	240	262
DSP	6	9
Clock cycles	20	34
Max frequency (MHz)	740	740

III. FIXED-POINT NEWTON-RAPHSON ARCHITECTURES FOR A 65 NM CMOS ASIC

The SLNR architectures introduced in Section II offer the maximum computing frequency on Xilinx FPGAs. Since every hardware target is different, it must be verified that the SLNR architectures yield the highest computing frequency for 65 nm CMOS ASIC. The goal here is to minimise the critical path and the silicon area. A memory-based Newton-Raphson alternative to SLNR can be considered to improve the first approximation and reduce the number of iterations while providing the same accuracy as shown in Fig. 1. Both architectures are proposed for a 16-bit input named a , resented with 8 bits for integer and fractional parts. So, for the memory-based solution, the input must be scaled in order to have a_s in the interval $[1, 2[$. The 10 most significant bits of a_s are used as the address of the RAM. In this way, x_0 is the approximation of the desired value with 10 bits of accuracy. The following Newton-Raphson iteration block ensures the 16-bit accuracy. The output must then be similarly scaled to have the desired value. A synthesis with Encounter RTL Compiler states that the multiplier is the most impactful element on the critical path for the SLNR architecture. The delay can not be reduced, because the multipliers are already between two registers, so that multiplier's propagation delay of 593 ps becomes the comparison reference.

non-pipelined and generic architecture in [9] is not specific to reciprocal or inverse square root computation but is the most effective ASIC implementation in the latest literature. [9] uses a cubic Chebyshev interpolator, which relies on 4 coefficients sets stored in a Lookup-table, as presented in Fig. 4. The Lookup table is composed of 4 memory blocks. The coefficients can be changed in order to compute different functions. The architecture contains 5 multipliers (Multiplier #1 to 3, squarer and cuber). The value x is the input of the architecture and, for the reciprocal computation, must belong to the interval $[1, 2]$. The reciprocal operator synthesis for the IBM 65 nm CMOS technology is done. Table V compares [9] and the memory-based Newton-Raphson reciprocal operator.

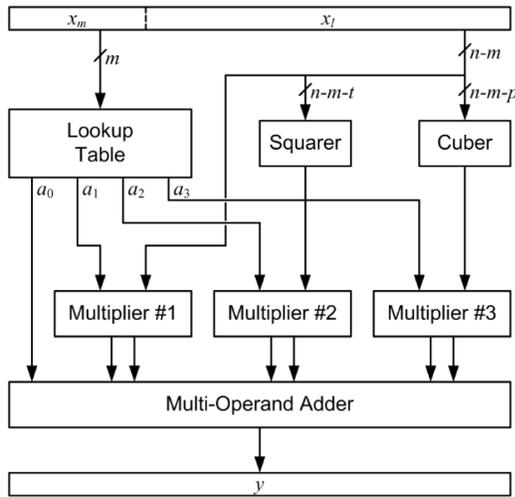


Fig. 4. Cubic Chebyshev interpolator architecture [9]

If [9]’s solution is changed in order to be pipelined, the critical path would be at least equal to the memory delay and then the maximum frequency would be equal to 0.820 GHz. Then, the use of memory blocks has an impact on the occupied area by this architecture. Indeed, the surface required by the memory-based Newton-Raphson solution is twice as small. Finally, the memory-based Newton-Raphson provides 4 times better computing performance than [9]. Table VI presents the results for memory-based Newton-Raphson architectures. Each operator has a similar computing performance and then is a good implementation to save area. So the proposed ASIC implementation improves both the computing frequency and the occupied silicon surface.

V. CONCLUSION

In this paper, the fixed-point SLNR architecture presented in previous work is investigated on the 65 nm CMOS technology and compared with a memory-based Newton-Raphson solution. These solutions have similar computing performance and their use depends on the rest of the algorithm. The computing performance can be increased up to 41 Gop/s/mm², which is a literature improvement by a factor of 4. Thus, this solution

TABLE V
65 NM COMPARISON BETWEEN 16-BIT RECIPROCAL SLNR AND CUBIC CHEBYSHEV ARCHITECTURES

Architecture	Memory-based NR	[9]
Memory	1024 x 16	512 x 20 512 x 12 512 x 12 512 x 10
Memory delay (ns)	0.630	1.22
Multipliers	2	5
Maximum frequency (GHz)	1.587	0.360
Area (μm^2)	38 527	95 700
Gop/s/mm ²	41	9

TABLE VI
STMICROELECTRONICS 65 NM CMOS RESULTS FOR MEMORY-BASED NEWTON-RAPHSON RECIPROCAL AND INVERSE SQUARE ROOT

Operator	Reciprocal	Inverse Square Root
Multipliers	2	3
Maximum frequency (GHz)	1.587	1.587
Area (μm^2)	38 527	39 683
Gop/s/mm ²	41	40

fits the BCI requirements and offers a good trade-off between operating frequency and silicon area.

REFERENCES

- [1] K. J. Huang, W. Y. Shih, J. C. Chang, C. W. Feng, and W. C. Fang, “A pipeline VLSI design of fast singular value decomposition processor for real-time EEG system based on on-line recursive independent component analysis,” in *2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Jul. 2013, pp. 1944–1947.
- [2] R. Andraka, “A survey of cordic algorithms for fpga based computers,” in *Proceedings of the 1998 ACM/SIGDA Sixth International Symposium on Field Programmable Gate Arrays*, ser. FPGA ’98. New York, NY, USA: ACM, 1998, pp. 191–200.
- [3] A. Rodriguez-Garcia, L. Pizano-Escalante, R. Parra-Michel, O. Longoria-Gandara, and J. Cortez, “Fast fixed-point divider based on Newton-Raphson method and piecewise polynomial approximation,” in *2013 International Conference on Reconfigurable Computing and FPGAs (ReConFig)*, Dec. 2013, pp. 1–6.
- [4] M. Allie and R. Lyons, “A root of less evil [digital signal processing],” *IEEE Signal Processing Magazine*, vol. 22, no. 2, pp. 93–96, Mar. 2005.
- [5] E. Libessart, M. Arzel, C. Lahucq, and F. Andriulli, “A Scaling-Less Newton-Raphson Pipelined Implementation for a Fixed-Point Reciprocal Operator,” *IEEE Signal Processing Letters*, vol. 24, no. 6, pp. 789–793, Jun. 2017.
- [6] —, “A scaling-less Newton-Raphson pipelined implementation for a fixed-point inverse square root operator,” in *2017 15th IEEE International New Circuits and Systems Conference (NEWCAS)*, Jun. 2017, pp. 157–160.
- [7] H. C. Neto and M. P. Vestias, “Very low resource table-based FPGA evaluation of elementary functions,” in *2013 International Conference on Reconfigurable Computing and FPGAs (ReConFig)*, Dec. 2013, pp. 1–6.
- [8] Xilinx. (2016) (Accessed: 2017-02-10) Virtex-7 t and xt fpgas data sheet: Dc and ac switching characteristics. [Online]. Available: http://www.xilinx.com/support/documentation/data_sheets/ds183_Virtex_7_Data_Sheet.pdf
- [9] M. Sadeghian, J. E. Stine, and E. G. Walters, “Optimized cubic chebyshev interpolator for elementary function hardware implementations,” in *2014 IEEE International Symposium on Circuits and Systems (ISCAS)*, Jun. 2014, pp. 1536–1539.