Reconfigurable High Throughput Function Evaluation for Virtex-2 FPGA

M. Anane, H. Bessalah and N.Anane
Centre de Développement des Technologies Avancées

1. Introduction

A reconfigurable architecture for efficient computation of several elementary functions in double precision floating-point format is presented in this paper**.** These functions, used in scientific computing, must be computed quickly and accurately, which can only be achieved on certain hardware implementations.

The most widely used methods to evaluate elementary functions are based on polynomial approximations [1] and tables based methods [2],[3]. Polynomial approximations are used since they involve easy to compute basic arithmetic operations such as addition, subtraction and multiplication. However, to accurately approximate a function on a rather large interval may require a high degree polynomial, the evaluation of which needs a high number of arithmetic operations leading to a long computation delay and a large memory to store the coefficients.

A solution to avoid these drawbacks is to use tables and piecewise polynomial approximation [4], which consists of splitting the approximation interval in several sub intervals and then using a low degree approximating polynomial in each subinterval and the polynomials coefficients are stored on tables. This allows achieving both a reduction in a memory size and a significant speed-up.

The progress in Field Programmable Gate Arrays (FPGA) provides new options for hardware implementation of elementary functions. The FPGA maintains the advantage of custom functionality, like an ASIC, with the ability of a reconfigurable architecture [5], while avoiding the high development cost.

In this paper, we exploit abilities of FPGA circuits, to design a reconfigurable architecture for double precision computation of elementary functions. The proposed method is suitable for computing the exponential function, and therefore can be applied to the computation of the remaining functions such as reciprocal, square root, logarithm and sine/cosine by reprogramming the FPGA.

2. Main results

Our strategy is to tailor the architecture to the selected FPGA circuit by defining the polynomial degree, regarding the amount of memory available in the circuit. To achieve a precision of 1ulp without exceeding the memory size available in Virtex-2 FPGA circuits [6], third degree polynomials approximation are needed and have been evaluated using Horner's scheme, which allows the reduction of the multiplications number and avoids the computation of x power by rearranging polynomials in Horner form, where third degree polynomial is written as follows:

$$P(x) = ((C3 \times x + C2) \times x + C1) \times x + C0.$$

Our method is based on minimax approximation [7] since it presents the reduced maximum error and the computation of its coefficients is done by MAPLE [8]. The evaluation of our polynomial requires three times computation of the operation (a×x+b), which is executed by one module FMA (Fused Multiplier Adder). The FMA contains three parts: the partial products generation, the reduction of partial products and finally the addition. Our FMAs do not comprise the final addition, which lets theirs outputs in C.S representation. The term b is admitted in the reduction bloc of the partial products as well as the partial products of a×x. The multiplication is carried out according to the modified Booth algorithm [9]. In order to achieve a high throughput rate, the architecture implementing our functions is pipelined using optimized FMAs adapted to hardware resources available in FPGA circuits of Virtex-2 family.

Functions considered in this paper are the reciprocal 1/x, the square root $x^{1/2}$, the exponential, the logarithm and the sine/cosine. They are computed for numbers represented in the IEEE-754 double precision binary floating format [10] as follows:

$$x = (-1)^{s} \times m \times 2^{e}$$

where, s is a one bit sign, e is on 11 bits exponent and m is a 52 bits mantissa normalized to $1 \leq m < 2$ with the Unit in the Last Place (ULP) being 2^{-52}. The mantissa is expressed by m = (1.f) where f is its fractional part.

A pipelined architecture implementing our method is proposed and its execution time and area cost estimations are presented, showing that the architecture has a cycle time of 17.372 ns and an operating frequency over than 57 MHz with a latency of four cycles and a throughput of one result per cycle.

The pipelined architecture presented in this paper was designed by using Xilinx design environment (foundation series 6.3i). All the blocs were described by VHDL codes. The tables were generated by CORE GENERATOR tool.

To guarantee the correct working of our architecture, we have simulated it by using ModelSim XE 5.5b, and synthesized it by FPGA express. Our architecture was implemented on XC2V3000 (- 6) fg676 circuit.

The implementation results are given in the table1:

Table 1: Implementation results.

Accumulated Total area	
Number of Slices	6524 of 14336 (45%)
Number of SRAM blocs	87 of 96 (90%)

We have compared our method to one proposed by Pineiro [11]. Our method is more accurate since it computes in double precision rather than in simple precision. Using a third degree polynomial, the execution time of an elementary function is smaller than of Pineiro using a second order polynomial, where the hardware requirements of his method are lower.

We have also compared our method to the look-up units developed by Mencer and all [12]. This method is based on the first-degree Taylor series expansion of the function to be evaluated, which gives less precision than the minimax polynomial used in our method. In addition, the Mencer's method deals only with the computation precision up to 24 bits, whereas our method goes until 53 bits. This is because the Mencer's method is very expensive in terms of memory and the tables are mapped on CLBs instead of SRAMs. The latency of Mencer's method is four for the 24 bits precision computation, whereas our method has the same latency for a computation precision of 53 bits.

3. Conclusion

A reconfigurable architecture for computing several elementary functions in double precision floating-point format was presented.

Our method has combined the main advantages of linear approximations, the speed and the high accuracy of the minimax approximation. The pipelining implementation has greatly increased the throughput of the architecture.

The implemented architecture is dedicated to the computation of the exponential function and can be adapted to all functions by reprogramming the FPGA. Better performances will be obtained by implementing our method on a Xilinx Virtex-4 FPGA.

References

[1] M. D. Ercegovac and Thomas Lang, *Digital Arithmetic* (Morgan Kaufmann Publishers, 2004).

[2] N. Takagi, IEEE Trans. on Computer, **47**(11), 1216 (1998).

[3] J-M. Muller, "Partially Rounded Small-Order Approximations for Accurate, Hardware-Oriented, Table-Based Methods", 16th IEEE Symposium on Computer Arithmetic (ARITH-16) 2003.

[4] J. A. Pineiro, S. F. Oberman, J. M.Muller and J. D. Bruguera, IEEE Trans. on Computer, **54**, No.3, (2005).

[5] J. Lang, R. Tessier and O. Mencer "Floating Point Unit Generation and Evaluation for FPGAs", 11th IEEE Symposium on field-Programmable Custom Machines (FCMM'03) 2003.

[6] "Virtex™-II 1.5 V Field programmable gate arrays", Advance product specification, DS031-1 (v1.7) October 2001, http://www.xilinx.com/

[7] J. M. Muller, *Elementary functions Algorithms and Implementation* (Birkhauser Boston, 1997).

[8] Waterloo Maple Inc, "Maple V Programming Guide", 1998.

[9] G.W. Bewick, "Fast multiplication, algorithms and implementation". PhD thesis, Stanford University (1994).

[10] American National Standards Institute of Electrical and Electronic Engineer," IEEE standard for binary floating-point arithmetic", ANSI/IEEE Standard, std 754-1985, New York (1985).

[11] J. A Pineiro, J. D Bruguera and J. M. Muller, "FPGA Implementation of Faithful

Polynomial Approximation for Powering Function Computation", Euro micro Symposium on Digital Signal Design. (DSD'2001), Warsaw, Poland

[12] O. Mencer and Wayne Luk, "Parameterized High Throughput Function Evaluation for FPGAs", The Journal of VLSI Signal Processing, Special Issue on Reconfigurable Computing, (Kluwer, 2004).