Volume 4, Issue 4, April 2014 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Hardware for Calculation of Natural Exponentiation (e^X) using CORDIC Algorithm

Vipin Tiwari*, Nilay Khare
Department of Computer Science,
MANIT Bhopal, India

Abstract— Calculation of exponentiation is an essential part of many real life applications such as medical, finance and spatial science etc. In this paper, an FPGA based efficient hardware for calculation of exponentiation with the help of COordinate Rotation DIgital Computer (CORDIC) has been proposed. With the use of CORDIC algorithm accuracy up-to five decimal digits on 32 bits input length has been achieved.

Keywords— CORDIC, natural exponentiation, hardware, FPGA, accuracy.

I. INTRODUCTION

Exponential functions have great impact on our daily life as medical science, finance industry, drug industry and other scientific research. Especially medical science and drug industry require very accurate and fast calculation of these functions. In order to solve this problem, a hardware for calculation of exponentiation has been presented in this paper. The proposed hardware is very efficient in term of time and chip area and provides very accurate results.

The CORDIC, first proposed by Jack E. Volder in 1959 [1] is an iterative algorithm to compute all trigonometric, logrithmic and exponential functions using very simple circuitry including adder and shifter only. Due to simple circuitry, power and chip area requirement of proposed hardware are very low. Rest of the paper has been organized as follows. Section 2 introduces CORDIC algorithm, section 3 summarizes some of the previous work done in the same field. Section 4 explains experimental setup used for the implementation. Section 5 summarizes results of experiments and finally section 6 concludes the work presented in this paper.

II. CORDIC ALGORITHM

CORDIC algorithm is an iterative algorithm to convert polar coordinates to Cartesian coordinates using rotations of vectors on coordinate plane. CORDIC works in two modes namely Rotation and Vectoring mode. In rotation mode a given vector is rotated through given angle using a series of micro-rotations each time with small angle. In vectoring mode, the angle between two vectors is calculated by rotating one vector with respect to other. CORDIC algorithm gains its popularity because of its less hardware requirement as it performs all operations with the combination of add or subtract and shift operations. All trigonometric, hyperbolic and logarithmic calculation can be performed using CORDIC along with all arithmetic operations as shown in [2]. CORDIC performs multiplication and division in the form of addition/subtraction and shift [3]. In [4], D.S. Cochran used this algorithm to design HP-35 first hand-held calculator as base algorithm. The generalize form of CORDIC equations [5] are as follows:

$$x_{i+1} = x_i - m\sigma_i y_i \rho^{-S_{m,i}} \quad (1)$$

$$y_{i+1} = \sigma_i x_i \rho^{-S_{m,i}} + y_i \quad (2)$$

$$z_{i+1} = z_i - \sigma_i \alpha_{m,i} \quad (3)$$

Where σ represents the direction of rotation, m represents the type of coordinate system like circular (m=1), linear (m=0) or hyperbolic (m=-1). ρ represents radix of number system, Sm,i stands for non-decreasing integer shift sequence and $\alpha_{m,i}$ is the elementary rotation angle. $\alpha_{m,i}$ directly related to Sm,i with

$$\alpha_{m,i} = \frac{1}{\sqrt{m}} \tan^{-1}\left(\sqrt{m}\rho^{-S_{m,i}}\right) \quad (4)$$

The micro-rotations are not exact but increase the length of vector at each micro-rotation therefore obtained results must be scaled with factor

$$K = \prod_i k_i, \quad (5)$$

$$k_i = \sqrt{1 + m\sigma_i^2 \rho^{-2S_{m,i}}} \quad (6)$$

The CORDIC algorithm for circular coordinate system (m=1) is used for variety of applications involving trigonometric functions. For circular coordinate system CORDIC equations take the form

$$x_{i+1} = x_i - \sigma_i y_i \rho^{-i} \quad (7)$$

$$y_{i+1} = \sigma_i x_i \rho^{-i} + y_i \quad (8)$$

$$z_{i+1} = z_i - \sigma_i \alpha_i \quad (9)$$

Here αi is restricted to be of the form tan(αi) = ρ-i . In this paper, circular coordinate system with binary number system has been used.

III. REVIEWS

Many attempts were made to calculate exponentiation using different techniques for different purposes. Some of them are being summarized here. A hardware based algorithm to calculate Radix-4 modular exponentiation was proposed in [6]. The algorithm works in combination of serial and parallel computation approach using cellular array structure with a bit slice feature which was used in cryptography. In [7], an FPGA based hardware architecture for exponentiation in Galois Fields (2m) using square and multiply method also called binary method. A method for calculation of power root was presented by Taichi Sumo and Yasuyuki Nogami in [8]. These power roots were then used to solve Exponential Inversion problem. Daisuke Suzuki proposed a method to maximize the performance of FPGA resources for modular exponentiation in [9]. The method is useful in public-key cryptosystems. Masaaki, Tsuyoshi and Eiji presented an application of exponentiation in Tate pairing used in cryptography. The work presented in [10] uses exponentiation to generate a unique value of bilinear pairing in the extension fields.

IV. EXPERIMENTAL SETUP

Proposed hardware was simulated using ISIM simulator engine of XILINX Ise 9.2i with Very High Speed Integrated Circuit Hardware Description Language (VHDL) as programming language. Input and output of design were 32 bits long represented in fixed point notation to store fractional numbers. Out of 32 bits least significant 21 bits were used to store fractional part of input/output and remaining 11 bits were used to store the integer part. CORDIC algorithm was implemented in its rotation mode, as in this mode, all intermediate micro-angles are of the form tan-12-i and are stored in ROM. Here, instead of micro-angles, values of natural logarithms of 10, 2, 1.1, 1.01, … , 1.0000001 have been stored in ROM and in other part of ROM values 10, 2, 1.1, 1.01, … , 1.0000001 have been stored. In order to find the value of natural exponentiation of number, First value of natural logarithms stored in ROM are subtracted repeatedly from given number and values corresponding to natural logarithms stored in ROM are multiplied. Equations (7) and (8) can be rewritten as

$$a_{n+1} = a_n - \ln i \quad (10)$$

$$b_{n+1} = b_n * i \quad (11)$$

Equations (10) and (11) are repeated for one value of i till a_{n+1} is positive. Once a_{n+1} becomes negative, next value of i is used. All values of i have been stored in ROM. Both these equations can be executed using same CORDIC architecture.

V. RESULTS

Figure 1 shows the results generated by proposed hardware. All experiments were performed on XC2V250 chip of Virtex2 family of hardware. For experimental purposes inputs were randomly selected as 0.25 and 1.0. Figure 1 shows the values of natural exponentiation for given inputs as 1.2840 and 2.7182 respectively. As in figure, proposed design generates results in 1050 ns with the 20 ns as clock cycle time.

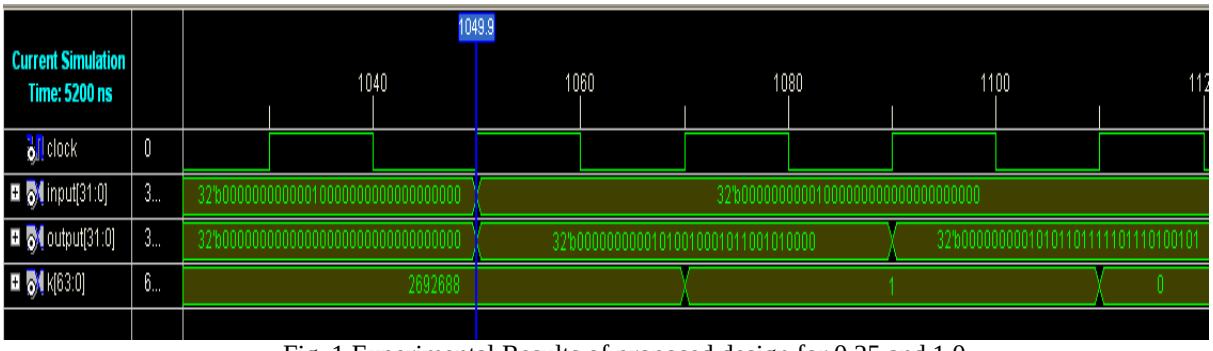

Fig. 1 Experimental Results of proposed design for 0.25 and 1.0

VI. CONCLUSIONS

As targeted, this paper presented an accurate and efficient hardware design to calculate exponentiation. The proposed design can generate accurate results up to 5 decimal places with 32 bits inputs and outputs in 1050 nano seconds (ns) with the clock cycle of 20 ns. The design is applicable in all places which involves the calculation of exponentiation medical science, finance management, network security.

REFERENCES

[1]. JACK E. VOLDER, *The CORDIC Trigonometric Computing Technique,* IRE TRANSACTIONS ON ELECTRONIC COMPUTERS

[2]. Ray Andraka, *A survey of CORDIC algorithms for FPGA based computers,* Copyright 1998 ACM 0-89791-978-5/98/01

[3]. J.E.Meggitt, *Pseudo Division and Pseudo Multiplication Processes,* IBM JOURNAL APRIL 1962

[4]. D. S. Cochran, *Algorithms and accuracy in the HP-35,* Hewlett-Packard Journal, pp. 1–11, June 1972.

[5]. B. Lakshmi and A. S. Dhar, *CORDIC Architectures: A Survey,* Hindawi Publishing Corporation VLSI Design Volume 2010, Article ID 794891

[6]. Naofumi Takagi, *A Radix-4 Modular Multiplication Hardware Algorithm for Modular Exponentiation,* IEEE TRANSACTIONS ON COMPUTERS, VOL 41, NO. 8. AUGUST 1992

[7]. Mario Alberto García Martínez, Guillermo Morales Luna, Francisco Rodríguez Henríquez, *Hardware Implementation of the Binary Method for Exponentiation in GF(2m),* Proceedings of the Fourth Mexican International Conference on Computer Science (ENC'03).

[8]. Taichi Sumo, Yasuyuki Nogami, *The Power Root Calculation for the Exponentiation Inversion Problem,* 3-1-1, Tsushima-naka, Kita, Okayama, Okayama, JAPAN

[9]. Daisuke Suzuki, *How to Maximize the Potential of FPGA Resources for Modular Exponentiation,* Paillier and I. Verbauwhede (Eds.): CHES 2007, LNCS 4727, pp. 272–288, Springer-Verlag Berlin Heidelberg 2007.

[10]. Masaaki Shirase, Tsuyoshi Takagi, and Eiji Okamoto, *Some Efficient Algorithms for the Final Exponentiation of ηT Pairing,* E. Dawson and D.S. Wong (Eds.): ISPEC 2007, LNCS 4464, pp. 254–268, © Springer-Verlag Berlin Heidelberg 2007

