IJIREEICE

ISSN (Online) 2321-2004
ISSN (Print) 2321-5526

International Journal of Innovative Research in
Electrical, Electronics, Instrumentation and Control Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 5, May 2018

Design and Simulation of High Speed Floating Point and Galois Field Multipliers Using Wave-Pipelining

P. V. Bharambe[1], M. N. Thakare[2]

M. Tech. Student (VLSI), Dept. Electronics & Telecomm Engg, B.D. College of Engg, Sevagram, Wardha, India[1]

Head of the Dept, Dept. Electronics & Telecomm Engg, B.D. College of Engg, Sevagram, Wardha, India[2]

Abstract: In this paper, we have presented the design, synthesis and simulation of 32-bit single precision floating point and Galois Field Multiplier using Wave Pipelining. Wave pipelining is a circuit design technique which allows digital synchronous systems to be clocked at rates higher than conventional pipelining techniques. Floating point multiplier is based on IEEE 754 standard which consist of 1-sign bit, 8-exponent bits and 23-mantissa (significand) bits. IEEE 754 standard works for addition, subtraction, multiplication and division, we use multiplication based on this standard. Galois Field multipliers have been widely used in coding theory and cryptography. The Galois Field Theory (GFT) which is used to design this multiplier deals with binary numbers, has the properties of a mathematical "field," and are finite in scope. Many Galois operations such as addition and multiplication are common and match those of regular math. These operations are particularly, handy for checking multiplication results. Finally, Synthesis and simulation of these multiplier designs has been done in Xilinx ISE 13.1 simulator and the coding of the design is done in VHDL language. The delay obtained is 19.418nsec and 4.071nsec for Floating point multiplier and galois field multiplier respectively.

Keywords: Floating Point, IEEE 754, Wave Pipelining, Galois Field Theory, VHDL.

I. INTRODUCTION

Multipliers are key components of many high performance systems such as FIR filters, microprocessors, digital signal processors, etc. A system's performance is generally determined by the performance of the multiplier because the multiplier is generally the slowest element in the system. Since multiplication dominates the execution time of most DSP application so there is need of high speed multiplier. Furthermore, it is generally the most area consuming. Hence, optimizing the speed and area of the multiplier is a major design issue. However, area and speed are usually conflicting constraints so that improving speed results mostly in larger areas. As a result, a whole spectrum of multipliers with different area-speed constraints has been designed with fully parallel Multipliers at one end of the spectrum and fully serial multipliers at the other end. These multipliers have moderate performance in both speed and area.

Binary floating point numbers multiplication is one of the basic functions used in Digital Signal Processing (DSP) application. The IEEE 754 standard provides the format for representation of Binary Floating point numbers in computers. The Binary Floating point numbers are represented in Single and Double formats. The Single consist of 32 bits and the Double consist of 64 bits. The formats are composed of 3 fields; Sign, Exponent and Mantissa. The term floating point is derived from the fact that there is no fixed number of digits before and after the decimal point, that is, the decimal point can float. There are also representations in which the number of digits before and after the decimal point is set, called fixed-point representation. Floating Point Numbers are numbers that can contain a fractional part. For e.g. following numbers are the floating point numbers: 3.0, -111.5, ½, 3E-5 etc. This is rather surprising because floating-point is ubiquitous in computer systems. Almost every language has a floating-point data type; computers from PC's to supercomputers have floating-point accelerators; most compilers will be called upon to compile floating-point algorithms from time to time; and virtually every operating system must respond to floating-point exceptions such as overflow. A number representation (called a numeral system in mathematics) specifies some way of storing a number that may be encoded as a string of digits. In computing, floating point describes a system for numerical representation in which a string of digits (or bits) represents a rational number. The term floating point refers to the fact that the radix point (decimal point, or, more commonly in computers, binary point) can "float"; that is, it can be placed anywhere relative to the significant digits of the number. This position is indicated separately in the internal representation, and floating-point representation can thus be thought of as a computer realization of scientific notation. Over the years, several different floating-point representations have been used in computers; however, for the last ten years the most commonly encountered representation is that defined by the IEEE 754 Standard.

IJIREEICE

ISSN (Online) 2321-2004
ISSN (Print) 2321-5526

International Journal of Innovative Research in
Electrical, Electronics, Instrumentation and Control Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 5, May 2018

Galois Field Theory (GFT) deals with numbers that are binary in nature, have the properties of a mathematical "field," and are finite in scope. Although some Galois computations don't exist in ordinary mathematics, many Galois operations match those of regular math. Addition (Ex-Or) and multiplication are common Galois operations, and logarithms, particularly, are handy for checking multiplication results. For over 40 years, Galois Field multipliers have been used both for coding theory and for cryptography. Both areas are complex, with similar needs, and both deal with fixed symbolic alphabets that neatly fit the extended Galois Field model. Galois field theory is also known as Finite field theory which is generally applied in Elliptic Curve Cryptography, error correction codes, digital signal processing, etc. Hence, specific implementation of hardware based on Galois field arithmetic comes in picture. There are different types of methods provided by Galois for addition, multiplication, etc of polynomial equations. In Galois field multiplication two k-bit inputs A,B are multiplied using modulo logic and gives polynomial P(x) over the finite field F_2^k. Incorrect multiplication will give full leakage of the secret key in cryptosystems. Therefore, it is most important to verify the correct hardware to be implemented of finite field multipliers used in such types of system[3].

Wave-Pipelining is one of the techniques, which is currently being used in VLSI circuit designs. Wave pipelining provides a method for reducing clock cycles and area, power, delay, latency of any synchronous circuit. Nowadays, peoples are using this technique because it gives design, analysis, synthesis as well as implementation across a variety of levels viz. process, route, layout, circuit, logic, timing, and architecture which are the main parameters of VLSI design. The idea of wave-pipelining was originally invented by Cotten, who then changed name to maximum rate pipelining. Wave Pipelining is a circuit Design that allows digital systems to be clocked at higher rates than, that can be achieved with conventional pipelining. Cotten observed that the rate at which logic can propagate through the circuit depends not on the longest path delay but also on the difference between the longest and the shortest path delays. Hence, several computation "waves", i.e., logic signals which are related to different clock cycles, can propagate through the logic simultaneously. Wave-pipelining can also be view as a virtual pipelining, in which each gate acts as a virtual storage element [7].

II. FLOATING POINT MULTIPLIER

The IEEE 754 standard provides the format for representation of Binary Floating point numbers. The Binary Floating point numbers are represented in Single and Double formats. The Single consist of 32 bits and the Double consist of 64 bits. The formats are composed of 3 fields; Sign, Exponent and Mantissa. The Figure 1 shows the structure of Single and Double formats of IEEE 754 standard. In case of Single, the Mantissa is represented in 23 bits and 1 bit is added to the MSB for normalization, Exponent is represented in 8 bits which is biased to 127, actually the Exponent is represented in excess 127 bit format and MSB of Single is reserved for Sign bit. When the sign bit is 1 that means the number is negative and when the sign bit is 0 that means the number is positive. In 64 bits format the Mantissa is represented in 52 bits, the Exponent is represented in 11 bits which is biased to 1023 and the MSB of Double is reserved for sign bit.

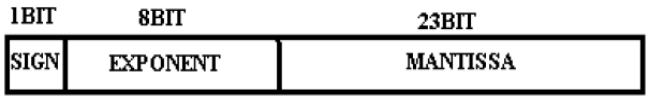

Figure 1: IEEE Format for Single Precision

Multiplication of two floating point numbers represented in IEEE 754 format is done by multiplying the normalized 24 bit mantissa, adding the biased 8 bit exponent and resultant is converted in excess 127 bit format, for the sign calculation the input sign bits are XORed.

The multiplier for the floating point numbers represented in IEEE 754 format can be divided in four different units: Mantissa Calculation Unit Exponent Calculation Unit and Sign Calculation Unit.

Basically the following circuit is used for multiplication of two floating point numbers. There is no definite logic level for representation of decimal point in digital circuit. So it is herculean to store the decimal point into the storing elements like flip flops, registers, memories etc in true form. So we ought to cogitate that how we can store the floating number. So we have a IEEE formats for different ranges like single precision, double precision, quad precision etc. In this paper single precision format is preferred.

IJIREEICE

ISSN (Online) 2321-2004
ISSN (Print) 2321-5526

International Journal of Innovative Research in
Electrical, Electronics, Instrumentation and Control Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 5, May 2018

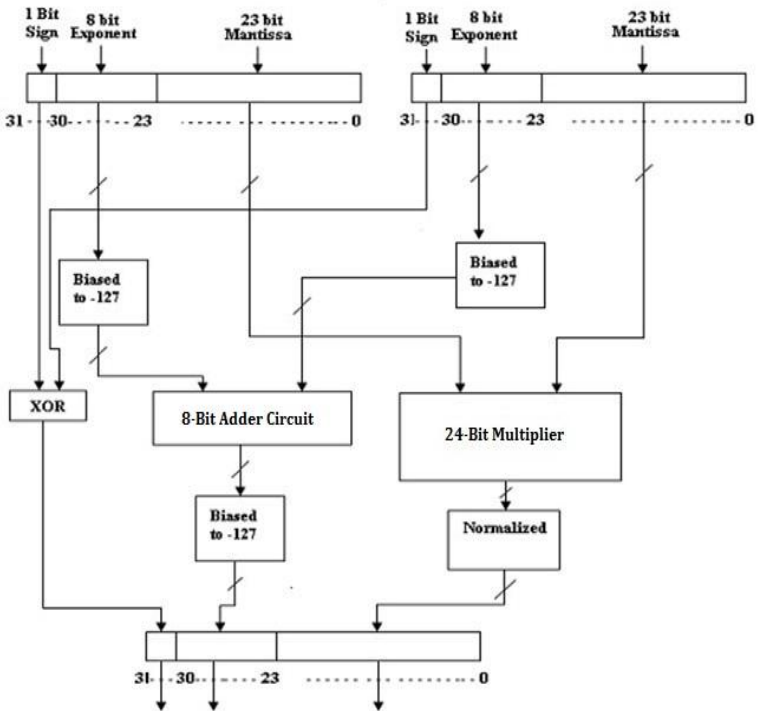

Figure 2: Architecture for 32-bit Floating point Multiplier

The advantage of floating-point representation over fixed-point (and integer) representation is that it can support a much wider range of values. For example, a fixed- point representation that has seven decimal digits, with the decimal point assumed to be positioned after the fifth digit, can represent the numbers 12345.67, 8765.43, 123.00, and so on, whereas a floating-point representation (such as the IEEE 754 decimal32 format) with seven decimal digits could in addition represent 1.234567, 123456.7, 0.00001234567, 1234567000000000, and so on. The floating-point format needs slightly more storage (to encode the position of the radix point), so when stored in the same space, floating-point numbers achieve their greater range at the expense of slightly less precision.

III.GALOIS FIELD MULTIPLIER

Galois field theory is extensively applied in Elliptic Curve Cryptography, error correction codes, digital signal processing, etc. Therefore, dedicated hardware implementations of Galois field arithmetic abound. Multiplication lies at the core of most Galois field computations – where two k-bit inputs A,B are multiplied modulo an irreducible polynomial $P(x)$ over the field $\mathbb{F}2k$. Incorrect (buggy) multiplication can lead to full leakage of the secret key [1] in cryptosystems. Therefore, it is of utmost importance to verify the correctness of hardware implementations of finite field multipliers residing at the core of such systems.

This paper presents Synthesis and Simulation of Galois field multiplier. The need for portable circuits to communicate with high bandwidths pushes the development of high speed and low-power circuits. In this context, efficient Galois Field GF (2m) arithmetic blocks are desired in many fields like error-control coding and cryptosystems. In error control coding, the Galois field GF (2m) arithmetic, mainly the field addition and multiplication, is the basis of Reed-Solomon encoding and decoding. In cryptographic applications, the GF (2m) arithmetic is largely used in elliptic-curve cryptosystems. In these applications, implementation of algebraic blocks greatly influences the system complexity and timing performance [5]. The addition operation in GF (2m) is equivalent to a simple bitwise XOR operation. On the other hand, the multiplication operation requires a larger and a slower hardware. The multiplier design presents a good area which is suitable for elliptic curve crypto processor design. Therefore elliptic curve crypto system can be used in applications that require small area and low consumption power such as smart cards and cellular telephones[1].

IJIREEICE

ISSN (Online) 2321-2004
ISSN (Print) 2321-5526

International Journal of Innovative Research in
Electrical, Electronics, Instrumentation and Control Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 5, May 2018

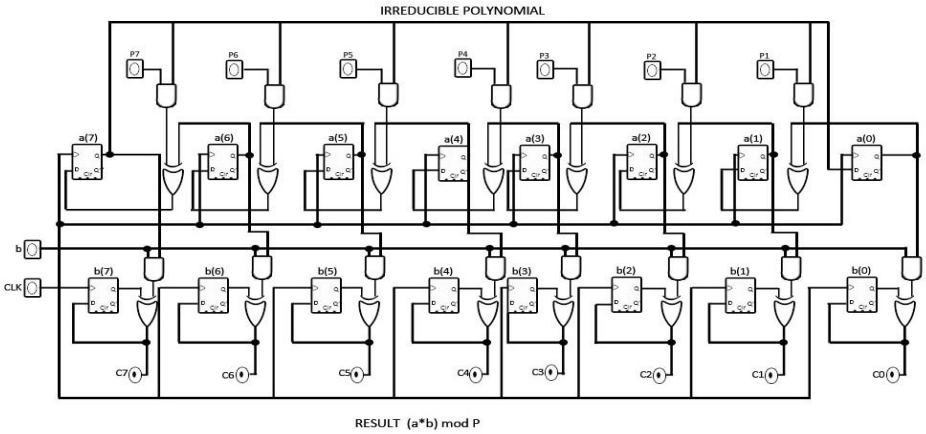

Figure 2: Architecture for 32-bit Floating point Multiplier

IV. EXPERIMENTAL RESULT

A. *Floating Point Multiplier*

RTL VIEW

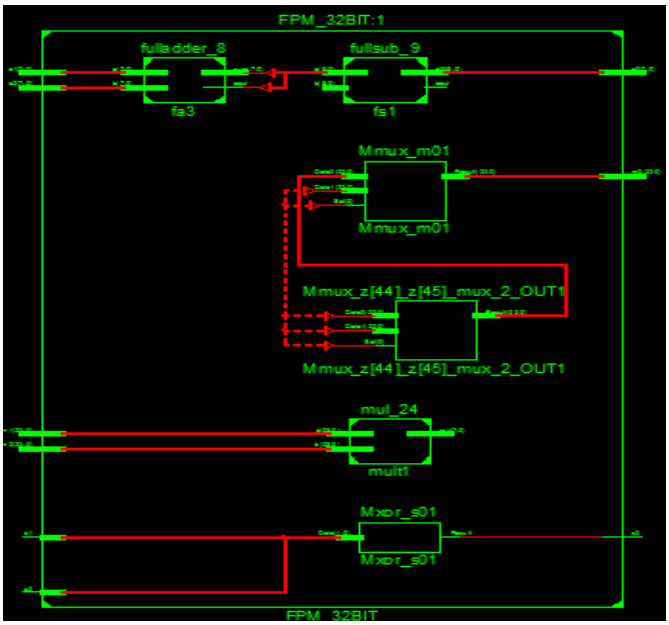

Figure 3: RTL View of 32-bit Floating point Multiplier

SIMULATION RESULT

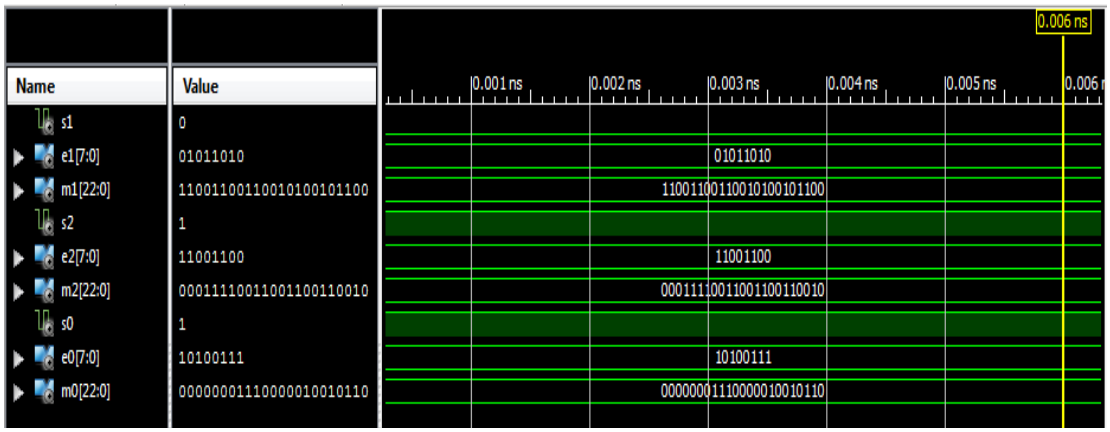

Figure 4: Simulation Result of 32-bit Floating point Multiplier

IJIREEICE

ISSN (Online) 2321-2004
ISSN (Print) 2321-5526

International Journal of Innovative Research in
Electrical, Electronics, Instrumentation and Control Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 5, May 2018

TABLE 1
COMPARISON TABLE (32-BIT FLOATING POINT MULTIPLIER)

	Reference[3]	Proposed Work
No. of Slices	1269	939
No. of LUTs	2270	0
Delay	34.333ns	19.418ns

B. Galois Field Multiplier

RTL VIEW

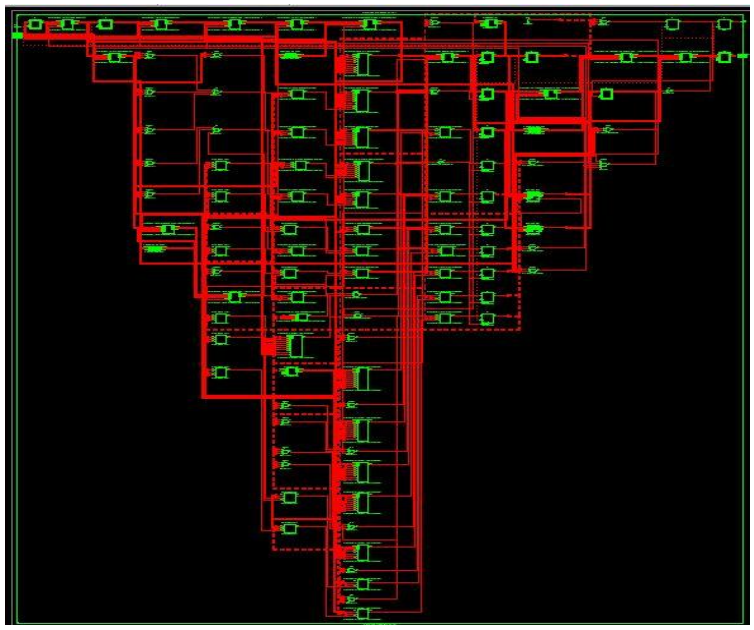

Figure 5: RTL View of 32-bit Galois FieldMultiplier

SIMULATION RESULT

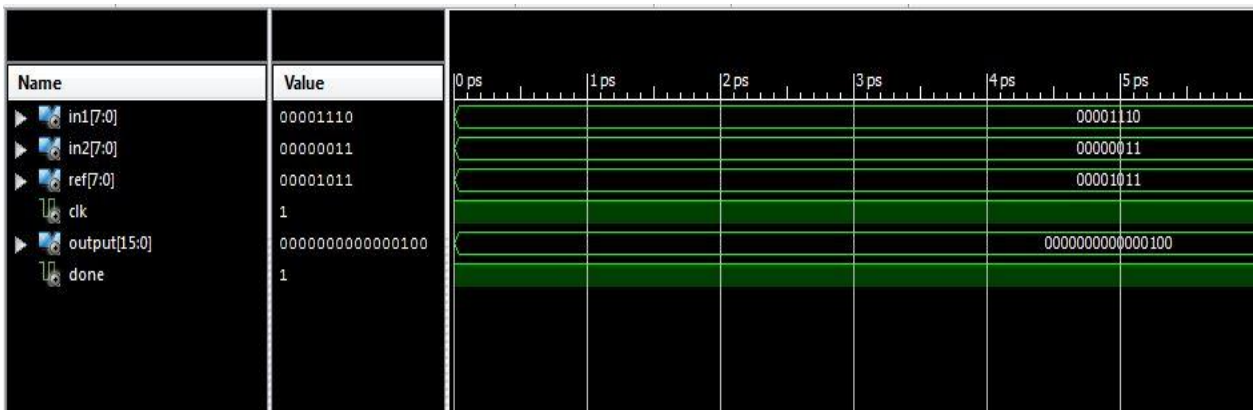

Figure 6: Simulation Result of 32-bit Galois Field Multiplier

V. CONCLUSION AND FUTURE SCOPE

In this paper we designed and simulated Floating Point and Galois Field Multiplier using wave pipelining. The design is area, power and timing efficient using the concept of wave-pipelining. Hence, a single precision floating point multiplier and Galois field multiplier with high speed is the outcome of this work. We used top-down design method in designing these Multipliers and VHDL language is used to describe the system. Wave-Pipelining is achieved with less clock cycles per operation. Through pipelining, the maximum throughput of operation is achieved as per design. The delay obtained for Floating point multiplier and galois field multiplier is 19.418nsec and 4.071ns respectively.

The use of VHDL for modeling is especially appealing since it provides a formal description of the system and allows the use of specific description styles to cover the different abstraction levels. The design can be further implemented for 64-bit i.e. double precision. [12]

IJIREEICE

ISSN (Online) 2321-2004
ISSN (Print) 2321-5526

International Journal of Innovative Research in
Electrical, Electronics, Instrumentation and Control Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 5, May 2018

REFERENCES

[1]Cunxi Yu, Maciej Ciesielski, Efficient Parallel Verification of Galois Field Multipliers, 2017ECE Department, University of Massachusetts, Amherst, 978-1-5090-1558-0/17/$31.00 ©2017 IEEE

[2]M. Anbuselvi, S. Salivahanan, P. Saravanan, Design and analysis of Floating point and Galois fieldmultipliers using Wave-pipelining, 2009 International Conference on Advances in Computing, Control, and Telecommunication Technologies, 978-0-7695-3915-7/09 $26.00 © 2009 IEEE.

[3] JinpengLv, Priyank Kalla, FORMAL VERIFICATION OF GALOIS FIELD MULTIPLIERS USING COMPUTER ALGEBRA TECHNIQUES, 2012 25th International Conference on VLSI Design, 1063-9667/12 $26.00 © 2012 IEEE.

[4] Anna Jain, Baisakhy Dash, Ajit Kumar Panda, Member, IEEE, Muchharla Suresh, Member IEEE, FPGA Design of a Fast 32-bit Floating Point Multiplier Unit, 2012 IEEE.

[5] RamyRaafat, Amira M. Abdel-Majeed, Rodina Samy, A Decimal Fully Parallel and Pipelined Floating Point Multiplier, 978-1-4244-2941-7/08/$25.00 ©2008 IEEE.

[6] Donald A. Joy and Maciej J. Ciesielski, "Clock Period MinimizationWith Wave Pipelining",IEEE Transaction On Computer Aided Designof Integrated Circuits and Systems, vol.12, No.14 April 1993.

[7] Fabian Klass, Maciji Ciesielski, Wayne P. Burleson and Wental Liu,"Wave –Pipelining: A Tutorial and Research Survey",IEEETransactions on Very Large Scale Integration (VLSI)Systems.vol.6,No.3, September 1998.

[8] G.Lakshminarayanan and B.Venkataramanai, "OptimizationTechniques for FPGA-Based Wave-pipelined DSP Blocks", IEEETransactions on Very Large Scale Integration (VLSI) Systems.vol.13,No.7, July 2005.

[9] Jesus Garcia and Michael J. Schulte, A COMBINED 16-BIT BINARY AND DUAL GALOIS FIELD MULTIPLIER, 0-7803-7587-4/02/$17.00 02002 IEEE.

[10] Brian Hickmann, Andrew Krioukov, and Michael Schulte, Mark Erle, A Parallel IEEE P754 Decimal Floating-Point Multiplier, 1-4244-1258-7/07/$25.00 ©2007 IEEE.

[11] Eduardo I. Boemo, Sergio L´opez-Buedo, and Juan M. Meneses, Some Experiments About Wave Pipelining on FPGA's, IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS, VOL. 6, NO. 2, JUNE 1998, 1063–8210/98$10.00 © 1998 IEEE

[12] Wayne P. Burleson, Member, IEEE, Maciej Ciesielski, Senior Member, IEEE, Fabian Klass, Associate Member, IEEE, and Wentai Liu, Senior Member, IEEE, Wave-Pipelining: A Tutorial and Research SurveyIEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS, VOL. 6, NO. 3, SEPTEMBER 1998, 1063–8210/98$10.00© 1998 IEEE.

[12] P. V. Bharambe, Prof. M. N. Thakare, Prof. G. D. Korde, "Review of 32-bit Floating Point and Galois Field Multipliers using Wave-Pipelining", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 5, Issue 1, January 2015, pp 935-938