Vol 05, Article 11506, December 2014
http://ijves.com
International Journal of VLSI and Embedded Systems-IJVES
ISSN: 2249 – 6556

VLSI IMPLEMENTATION OF IEEE754-2008 STANDARD FOR FINANCIAL TRANSACTIONS

T.MURALIKRISHNA, C.SRINIVASA MURTHY
Department of Electronics & Communication Engineering, QIS College, Ongole
muralitakkellapati@gmail.com, srinivasamch@gmail.com

ABSTRACT

Financial transactions are specified in decimal arithmetic. Until the introduction of IEEE 754-2008, specialized software hardware routines were used to perform these transactions but it incurred a penalty on performance. There is a need for accurate analysis of these solutions on representative DFP benchmarks. This Work uses a single precision evaluation of decimal numbers .In this paper we are taking decimal numbers and converting them into to normalization form and then finally to floating point in order to do our calculation like [addition, subtraction, multiplication, division] by using this method there is a scope of increasing accuracy of evaluation of decimal numbers, we also presented the performance analysis that gives the average number of cycles for common DFP operations and the total number of each DFP operation in each benchmark, and highlights the trade-offs between using 64-bit and 128-bit DFP operands for both binary and decimal significant encodings.

Index Terms– Decimal Floating Point(DFP),Field Programmable Gate arrays (FPGA), Floating Point(FP).

1. INTRODUCTION

Traditionally, calculations in the financial world are specified in decimal arithmetic many early computers used decimal arithmetic at the hardware level, but binary computing in hardware soon took over after Von Neumann et al. pointed out the advantages of simplified hardware. Our work is based on examination of the error which is defined as the difference between the answer produced using arbitrary precision decimal arithmetic, and that produced by the IEEE754 binary arithmetic, after financial rounding rules are applied. In 2008, the IEEE Standard 754-1985 was revised to define standards for decimal floating-point arithmetic (IEEE 754-2008, ISO/IEC/IEEE 60559:2011) , yet most general purpose computers still implement binary arithmetic. Decimal calculations are natural to human beings and have been the standard basis for numerical calculations for thousands of years. With the advent of the computer age ,binary number systems have become popular because of the ease with which binary computational mechanisms can be implemented with physical devices However, binary numerical calculations are not adequate for many applications, particularly for transaction processing where correctly rounded decimal calculations are required for financial transactions. In 2008, the IEEE ratified IEEE-754-2008 Standard for Floating-Point Arithmetic which contained a format and specific-ation for performing decimal floating-point (DFP) arithmetic.

Since the ratification of the new IEEE standard, there has been a significant amount of work in the area of DFP hardware .In 2006, IBM introduced DFP execution using millicode support on their z9 Enterprise server.In 2007, IBM released the first DFP hardware accelerator on its Power platform, and in 2008, DFP hardware support was made available on the z10 Mainframe These accelerators were considerably faster than traditional software solutions but were designed to minimize area. Consequently, they were not pipelined. Other pursuits were made in academia such as , but utilized software traps for special cases the hardware did not support .This paper describes the DFP Accelerator on the IBMzEnterprise-196 (z196) processor, which is the first fully IEEE compliant pipelined DFP execution unit available in a commercial product. The DFP Accelerator also executes the fixed-point decimal instructions critical on many commercial trans-action workloads. Financial calculations are specified in decimal, and significant rounding errors may occur due to conversion to from the binary representation. For example, the binary representation of 0.1 lies strictly between two binary floating-point numbers and cannot be exactly represented by either of them so it is approximated1-:100110011001100110011001100110011001100110011001100110011010 X 2-4 in IEEE 754 double precision, which when converted back to decimal gives 0.1000000000000000-5551115123126.Since the ratification of the new IEEE standard, there has been a significant amount of work in the area of DFP hardware. In 2006, IBM introduced DFP execution using millicode support on their z9 Enterprise server.In 2007,IBM released the first DFP hardware accelerator on its Power platform , and in 2008, DFP hardware support was made available on the z10 Mainframe These accelerators were considerably faster than traditional software solutions but were designed to minimize area. Consequently, they were not pipelined. Other pursuits were made in academia such as, but utilized software traps for special cases the hardware did not support.

II. FINANCIAL CALCULATIONS ON FINITE PRECISION MACHINES

The IEEE 754 floating point standard defines the rules for the approximation of real numbers in finite precision machine,as well as for arithmetic operations like addition, subtraction, multiplication, division, etc. Floating-

Vol 05, Article 11506, December 2014
http://ijves.com
International Journal of VLSI and Embedded Systems-IJVES
ISSN: 2249 – 6556

point rep-resentations of a number define a base β and a precision p. IEEE 754-1985 is a binary standard where β ¼ 2 and p ¼ 24 (7.22 decimal digits) for single precision and p ¼ 53 (15.95 decimal digits)for double precision. Length of a word in single precision is 32 bits, 23 bits for the significant, and 8 bits for the exponent and 1 sign bit. In double precision a word is 64 bits long,52 bits for the significant, and 11 bits for the exponent and1 sign bit. In 2008 this standard was updated to IEEE 754-2008, where the radix can be either 2 or 10. IEEE 754-2008 defines the interchange formats, rounding algorithms, operations, and exception handling. Whenever the result of an operation is inexact then by default IEEE standards rounding rule approximates the answer to the nearest representable number. There are two rounding to nearest schemes for this. The standard defines three other rounding modes, called directed roundings.

Payer Payee

Fig. 1. A payer and payee—single sided

Amount is added to the payee, in the payee's currency. Intermediate calculations and representations may be in either double precision binary or exact decimal arithmetic .This financial model, while simplified, is adequate for obtaining insight.

III. SINGLE NODE

An account in a bank can act as a payer or a payee, and the transactions are withdrawals and deposits. The transaction amount Δ could be an arbitrary real value based on some calculations like interest payment. This amount Δ may not be exactly representable if IEEE 754 standard for binary ϵarithmetic is used. In this case, we are interested in ϵ, the error in the capitalization (difference between the exact value, and that calculated using IEEE 754 followed by currency specific rounding methods) at the node X (payer)or Y (payee) as shown in Fig. 1.For computing this in exact decimal arithmetic, we have the relations between the initial and final quantities as shown .Let Xi and Yi represent the initial balance amount at payer node and payee node, respectively.

$$X_f = \rho_c (X_i - \rho_c(\Delta))$$
$$Y_f = \rho_c (Y_i + \rho_c(\Delta))$$

For calculations in double precision, we have an initial step where the decimal account balances are converted from their exact decimal values. Carets are used to refer to binary approximations of decimal quantities'

$$X\hat{}_i = \varepsilon(X_i)$$
$$Y\hat{}_i = \varepsilon(Y_i)$$
$$\Delta\hat{} = \varepsilon(\rho_c(\Delta))$$
$$X_f = \rho_c(\varepsilon(X\hat{}_i - \Delta\hat{}))$$
$$Y_f = \rho_c(\varepsilon(Y\hat{}_i + \Delta\hat{})),$$

The computations are in accordance to IEEE 754 standards and include any IEEE 754 specific rounding as the operands are binary numbers.

Payer Payee

Fig. 2. A payer and payee—single sided

3.1 A TRANSACTION PAIR

A pair of accounts transacting with each other forms a transaction pair. The nodes of the pair may have the same base currency, in which case the transactions are single currency transactions, or they could have different base currencies, where the transactions are multicurrency. Of interest here are the errors in the capitalization of the payer, X or the payee, Y or the total, X+Y . If the base currency of both payer and payee is same (c) as shown in Fig. 2, then there is no currency conversion in the transaction and there are no errors in decimal arithmetic if rounding is not applied

$$X_f = X_i - \rho_c(\Delta),$$
$$Y_f = Y_i + \rho_c(\Delta),$$
$$X_i + Y_i = X_f + Y_f$$

error due to binary computation is derived

$$(X_f+Y_f) \in \triangleq (X_i+Y_i)$$

Vol 05, Article 11506, December 2014
http://ijves.com
International Journal of VLSI and Embedded Systems-IJVES
ISSN: 2249 – 6556

Here, Xf and Yf are derived. Similarly, if the base currency of X and Y is different, then there is a currency conversion step in the transaction, which involves a multiplication by the exchange rate ηXY to convert currency of X into currency of Y , as shown in Fig. 3.

Let x, y be the currencies of X and Y , respectively.

$$X_f = \rho_x(X_i - \rho_x(\Delta)),$$

$$Y_f = \rho_y(Y_i + \rho_y(\eta XY - \rho_x(\Delta))),$$

$$\eta_{XY} X_i + Y_i = \eta_{XY} X_f + Y_f.$$

3.2 INTEREST PAYMENTS AND SPLITS

Periodic interest payments result in a correlated sequence of transaction volumes. A k-way split disbursement of funds is similar to a pair wise transaction, except that the initial amounts are in general non exactly representable (e.g.,Rs. 1.00 split into three pieces). Additional errors beyond the cases discussed above result. If an amount Z is to be split into n parts, then an error between 0 and Mod (Z,n) can occur. Hence, for any given n, the maximum error can be (n -1)

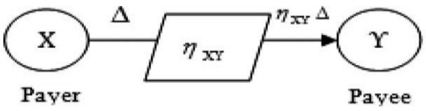


Fig. 3.A payer-payee pair (multicurrency)

3.3 A FINANCIAL NETWORK

A network is formed when accounts across multiple banks or accounts are transacting with each other. Here, each node (account) can have a different base currency so it is basically a multicurrency network.

If there are N accounts in the network and Let Xji and Xjf represent the initial and final balances, respectively, of the jth account, then

$$\sum_{j=1}^{N} XjiXj\$ = \sum_{j=1}^{N} XjfXj\$$$

This is because first, the currency exchange rates are specified only up to six significant figures and second, there are currency specific rounding rules.

4. ANALYSIS OF ERROR PROCESS

We analyze the behavior of the error process using the model of transactions discussed in Section 3. We discuss the properties of the error in representing a single value, followed by properties of error accumulation in basic operations**.**

4.1 ERROR IN REPRESENTING A SINGLE VALUE

First consider a single decimal value in D digits, and its nearest binary representation in B bits. We assume that default round to nearest rounding direction mode is be ingused. We consider that all numbers are normalized to unity(we need only consider this case for insight, if D and B are suitably chosen).

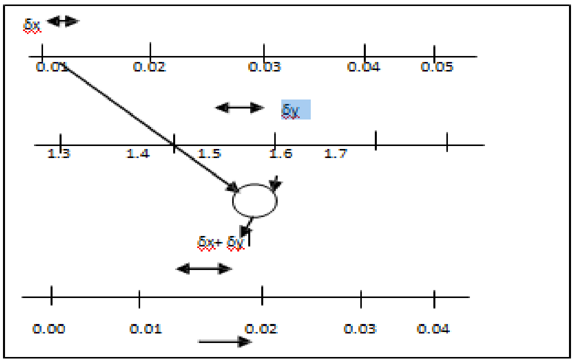


Fig.4. Error in multiplication

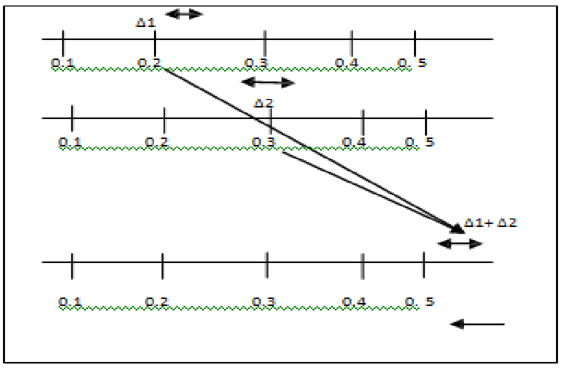


Fig. 5. Error in addition

4.2 ERROR IN A SINGLE ADDITION

If the balance amount x is to be represented in binary format, let δx be the error associated with it (Fig. 6). Let y be the transaction (deposit) amount and δy be the error associated with it.

4.3 Error in a Single Multiplication

The story is different for multiplications .since the product of two exactly representable numbers can easily hit a rounding boundary. Another common decimal operation in financial applicat- ionsis the IEEE 754-2008 quantize operation. This is essentially a multiplication by a power of 10 and the same analysis as above is

Vol 05, Article 11506, December 2014
http://ijves.com
International Journal of VLSI and Embedded Systems-IJVES
ISSN: 2249 – 6556

applicable. With this analysis of errors in number representation and basic operations, we can discuss the impact of finiteprecision in financial calculations.

5. ERROR PROCESS: WORST CASE

we discuss the various transactions below, in order. A novel tabular approach is used to examine the worst case errors, as outlined below. These errors are rare, requiring either very large numbers or accidental matches. This fact will be exploited to obtain high-speed routines using only binary arithmetic

5.1 SINGLE NODE, PAYER/PAYEE: TRANSACTION ERROR MATRIX With capitalization amounts exceeding $10^{13}(2^{45})$, errors are possible in additions, as shown in Section 4. Here, using a tabular approach, we demonstrate a sequence of transaction amounts, such that an error is made in every transaction.

First, we define the capitalization transaction error matrix- (CTEM) T(D,B), where D is the number of decimal digits in the fraction and B is the number of binary bits in the binary approximation, as an nxm matrix, with entries Tij= Error in adding/subtracting Transaction amount Δ_i to from Capital Ci, where n is the number of possible capitalization values and m is the number of possible transaction amounts.

This error is with respect to exact decimal arithmetic. To differentiate between the CTEM for deposits and withdrawals, we represent CTEM as $T_+(D,B)$ and $T_-(D,B)$, respectively.

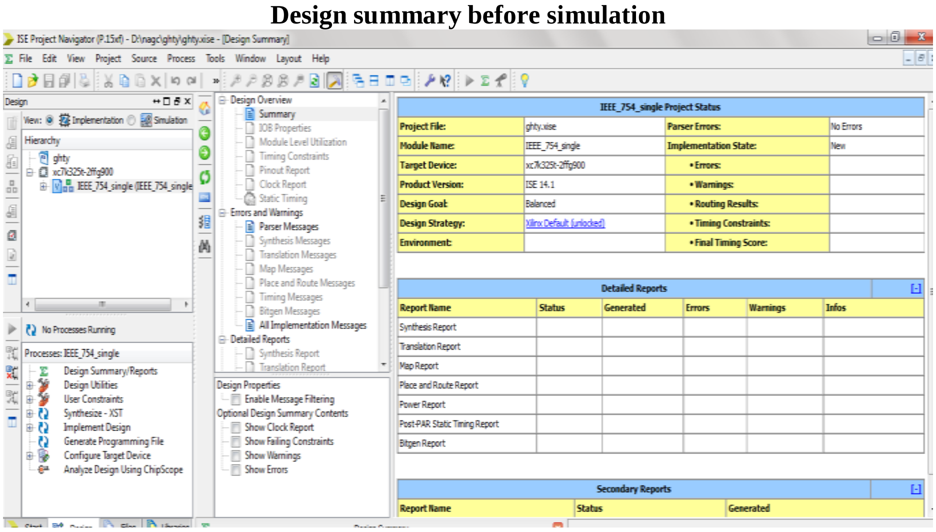
Table1 CTEM T+(1,3)

	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
0.1	-0.1	-0.1	0	0	0	-0.1	-0.1	0	0
0.2	-0.1	-0.1	0	0	-0.1	-0.1	-0.1	0	0
0.3	0	0	0.1	0.1	0.1	0	0	0.1	0.1
0.4	0	0	0.1	0.1	0	0	0	0.1	0.1
0.5	0	-0.1	0.1	0	0	0	-0.1	0.1	0
0.6	-0.1	-0.1	0	0	0	-0.1	-0.1	0	0
0.7	-0.1	-0.1	0	0	-0.1	-0.1	-0.1	0	0
0.8	0	0	0.1	0.1	0.1	0	0	0.1	0.1
0.9	0	0	0.1	0.1	0	0	0	0.1	0.1
1	0	-0.1	0.1	0	0	0	-0.1	0.1	0

Table 1 shows a CTEM $T_+(1,3)$ for deposit transactions(additions).Similarly, shows a multiplication CTEM $T_X(1,3)$ where each entry Tij is the error that results from multiplying the ith capital to jth interest rate.

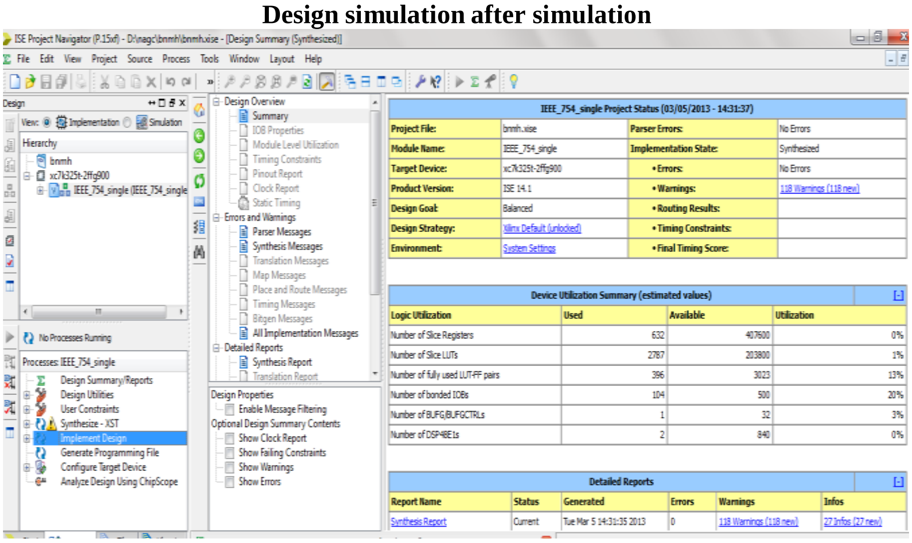
5.RESULTS

Design summary before simulation

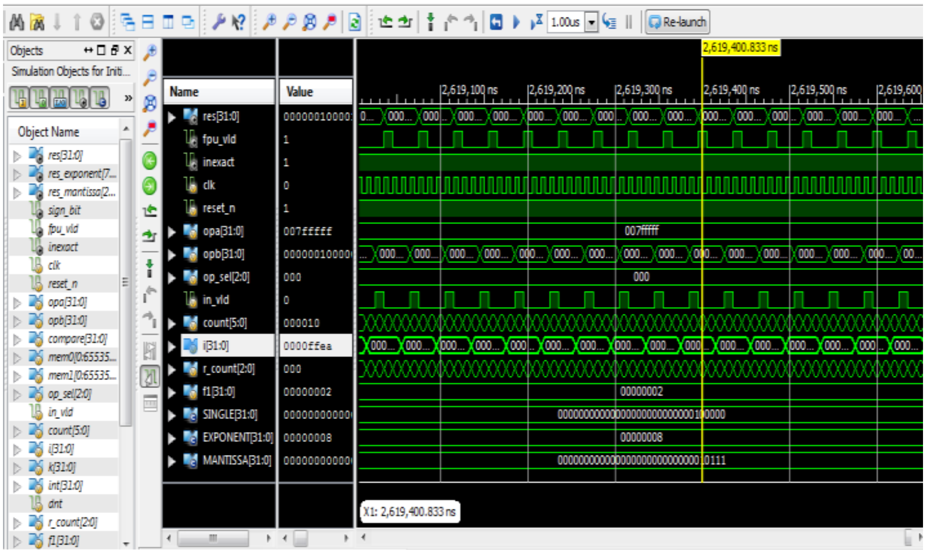


Vol 05, Article 11506, December 2014
http://ijves.com
International Journal of VLSI and Embedded Systems-IJVES
ISSN: 2249 – 6556

Design simulation after simulation



FLOATING POINT ADDITION RESULT



FLOATING POINT SUBTRACTION RESUL

CONCLUSION

If financial software neglect the effects of IEEE-754 finite precision, then the results could be disastrous and huge monetary losses may occur. This paper quantifies these losses using a matrix approach the CTEM.

Vol 05, Article 11506, December 2014
http://ijves.com
International Journal of VLSI and Embedded Systems-IJVES
ISSN: 2249 – 6556

Using the CTEM, we showed that even the use of double precision can be exploited to create arbitrary error sequences, with considerable possibilities of financial arbitrage. Initial evidence testifies to the fact that these sequences can be chosen so as to pass many common statistical tests, and thus avoid detection. Thus, we show that using binary arithmetic directly for financial calculations is inappropriate, as error scan build up without bound, and as we have found statistically undetectable sequences in our experiments.

REFERENCES

1. M. Ganesh Perumal, and G.N. Srinivasa Prasanna, "On Basic Financial Decimal Operations on Binary Machines",2012.
2. IEEE Standard for Floating-Point Arithmetic, IEEE Standard 754- 2008, 2008.
3. W.Kahan, "IEEE Standard 754 for Binary Floati-ng Point,Arithmetic,"Standards Committee of the IEEE CS, 1996.
4. W. Kahan, "Floating-Point Arithmetic Besieged by Business,Decisions,"Proc.IEEE17th,Symp.Computer Arithmetic, 2010.
5. D.M. Gay, Correctly Rounded Binary-Decimal and Decimal-Binary Conversions. Numerical Analysis Manuscript 90-10, Murray Hill, NJ:AT&T Bell Laboratories, 1990
6. OnlineBinary-DecimalConverter," http://www.binaryconvert.com/index.html, 2012.
7. M.J. Anderson, C. Tsen, L.-K.Wang, K. Compton, andM.J Schulte,"Performance Analysis of Decimal Floating-Point Libraries and Itsimpact on Decimal Hardware and Software Solutions," Proc. IEEEInt'l Conf. Computer Design, pp. 465-471, Oct. 2009.
8. A.Fog,"InstructionTables," http://www.agner.org/optimize/instructiontables.pd-f, 2012
9. Decimal Floating-Point (DFP) Benchmarks, http://mesa.ecewisc.edu/display_project.php?projectid=10, 2012.
10. M.Cornea, J. Harrison, C. Anderson, P. Tang, E. Schneider,andE.Gvozdev,"ASoftware,Implementation of the IEEE 754R Decimal Floating-Point Arithmetic,Using the Binary Encoding Forma-t,"IEEE Trans. Computers, vol. 58, no. 2, pp. 148-162, Feb. 2009.
11. J.-M. Muller, N. Brisebarre, F. de Dinechin, C.-P. Jeannerod, V.Lefevre, G. Melquiond, N. Revol, D. Stehle, and S. Torres,Handbook of Floating-Point Arithmetic. Springer, 2010.
12. N.J. Higham, "The Accuracy of Floating Point Summation," SIAM J. Scientific Computing, vol. 14, pp. 783-799, July 1993.

AUTHOR BIOGRAPHY

T. Muralikrishna was born on 30-05-1991 at Ongole, India. He received his, B.Tech in Electronics & Communication Engineering from QISCET Affiliated to JNTU KAKINADA,ONGOLE, AP, India, and Pursuing M.Tech in DECS at Qis College of Engineering, Ongole, affiliated to JNTU, Kakinada, AP, India. Her areas of interest are Vlsi image processing and wireless communication networks

Prof. C. Srinivasa Murthy received his B. Tech in Electronics & Communication Engineering from Dr. Babasaheb Ambedkar Marathwada University, Aurangabad, Maharashtra, India, And M. Tech degree from Bharath University, Chennai, Tamil Nadu,India. He is currently working as an Prof in Qis College of Engineering and technology Ongole, AP, India. And His work experience is 15 years. His areas of interest are Low Power Vlsi design.