Decimal Floating Point Arithmetic

Vinod Kapse[1], Puneet Bharadwaj[2] , Y. Arunika Rao[3]
Electronics and Communication, GGITS RGPV University, Jabalpur (M.P.), India

Abstract

The growing popularity of decimal computer arithmetic in scientific , commercial, financial and internet based applications. Binary floating point system is preferred in digital system. Its disadvantage is that it is quite different from manual calculations. Binary floating point introduces unacceptable error. It cannot give the accurate result of decimal values .So decimal floating point is used to get the exact value of decimal number. The IEEE 754 defines standard for Binary Floating- Point Arithmetic (IEEE 754) is the most widely used standard for floating -point computation. As the number of stages increases area and power consumption also increases. In order to reduce area and latency CSA adder with 3-stage pipelining technique is used. Binary to BCD conversion is also done in order to display the overflow and errors. The design is implemented on Xilinx tool and has been synthesis and simulated on the same tool.

Keywords

CSA, Pipelining, Latency, Area, BCD

1. Introduction

Decimal Floating Point arithmetic application and demand in industry is increased now a day. These play a very important role in various electronics equipment like in mobile phone, laptops, tablets etc. Carry Save Adder (CSA) is used with 3-stage pipelining technique enhancing the performance and reducing the area .IEEE-754 defines the standard for single precision floating point standard is also discuss below. BCD is very common in order to display the numeric value.

Floating Point Standard

• Floating point are number in fraction, decimal values, small number like 0.0000000000021, very large number like 5.5×10^{9}
• IEEE 754 floating point standard

– Single precision
– Double precision

Single Precision Floating Point Number

The single precision floating point number representation is shown below

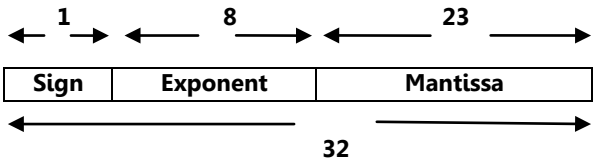


Figure 1: Single -Precision Floating Point Number Representation

Sign: Sign bit determines the sign of the number.

Mantissa: .Mantissa has 23 fraction bits of the significand appear in the memory format but the total precision is 24 bit .

Exponent: Exponent is 8-bit and is bias value is 127.

To achieve a bias of equal to $2^{n-1}-1$ is added to the actual exponent in order to obtain the stored exponent. Exponent is either an 8 bit signed integer from −128 to 127 (2's Complement) or an 8 bit unsigned integer from 0 to 255 .

The range of single and double floating point numbers are shown below:

Table 1 : Ranges of Single and Double Floating Point Number

	Sign	Exponent	Fraction	Bias
Single Precision	1[31]	8[30-23]	23[22-00]	127
Double Precision	1[63]	11[62-52]	52[51-00]	1023

2.Model

Floating Point Arithmetic

Addition is a fundamental arithmetic operation and the design of efficient adder aids as well as in the performance and efficiency of other operation like multiplier and dividers. Binary arithmetic is important for the decimal case since decimal numbers are represented in binary and many concept techniques developed for binary can be applied to some extent as well. One of the most basic elements in addition is the Full Adder(FA) .

Full Adder

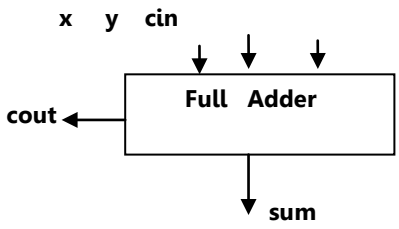


Figure 2- Full Adder

Carry Save Addition- Carry Save Addition is the idea of utilizing addition without carries connected in series. Carry Save Adder is used to compute sum of three or more n-bit binary numbers. Carry Save adder has also three input and two output same as full adder. Figure shows the sum of two 32-bit binary numbers, so 32 full adder are used at first stage.. Let X and Y are two 32-bit numbers and produces partial sum and carry as S and C.

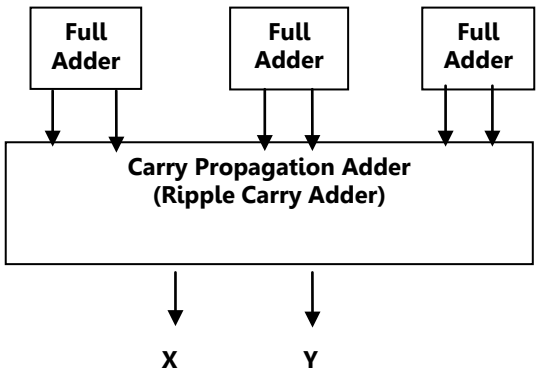


Figure 3 - Computation flow of Carry Save Adder

Exception

There are four exception that may be signaled during multiplication are

1. Invalid
2. Inexact
3. Overflow
4. Underflow

1. **Invalid Operation**- Invalid operation exception is signaled when either operand is a signaling NaN or when zero or infinity are multiplied.

   Addition $(+\infty) + (-\infty)$
   Multiplication $0\times\infty$
   Division $0/0$ or ∞/∞
   Square rooting operand <0
   All are producing Not a Number (NaN) as a result.

2. **Inexact Operation**- Inexact exception is signaled when the rounded result of an operation differs from what would have been computed were both exponent range and precision unbounded.

3. **Overflow Exception-** Overflow Exception is signaled when a result's magnitude exceed's the largest finite number. The detection is accomplished after rounding by examing the computed result as though the exponent range is limited.
4. **Underflow Exception**- Underflow exception is signaled when a result is both tiny and inexact. Tininess is when the result magnitude is tiny and the smallest number exclusive.
5. With a 32-bit word, using 2's complement integer integers from -2^{31} to $2^{31}-1$ can be represented, for a total of 2^{32} different numbers.
   With the floating-point format presented, the following ranges of numbers are possible

• Negative numbers between $-(1-2^{-24})*2^{127}$ and $-0.5*2^{-128}$;

• Positive numbers between $0.5*2^{-128}$ and $(1-2^{-24})*2^{127}$.

• Negative numbers less than $-(1-2^{-24})*2^{127}$, called *negative overflow*;

• Negative numbers greater than $-0.5*2^{-128}$, called *negative underflow*;

• Zero;

• Positive numbers less than $0.5 * 2^{-128}$, called *positive underflow*,

• Positive numbers greater than $(1 - 2^{-24}) * 2^{127}$, called *positive overflow*

3. Previous Work

Kulvir Singh and Dilip Kumar proposed **Modified BoothMultiplier with Carry Select Adder using 3-stage pipelining technique**

This paper presents a high-speed and low area 16×16 bit Modified Booth Multiplier (MBM) by using Carry Select Adder (CSA) and 3-stage pipelining technique. These techniques improve the performance of MBM by reducing the delay time. Simulation results show that the delay is reduced by 56% and the number of SLICES and LUT's are reduced by 4% respectively as compared to high speed MBM. The multiplier circuit is designed using VHDL and simulated using Xilinx ISE Simulator. The power metric of the MBM is evaluated using Cadence tools.

Navdeep Kaur and Rajeev Kumar patial proposed **Implementation of Modified Booth Multiplier using Pipeline Technique on FPGA.**

This paper presents 16×16 bit Radix-4 Modified Booth's Multiplier (MBM) optimized for high speed multiplication by using pipeline Technique. This paper aims at reduction of hardware utilization. This is accomplished by the use of 3:2 compressor adders. An efficient VHDL code has been written, successfully simulated on Modelsim 10.2 simulator and Xilinx 12.4 navigator is used for synthesizing the code. Simulation result shows the clock period of 2.689ns. The selected device to synthesize the code is xc3s500e-4pq208 of Sartan-3E family. The area utilization is shown as 222 numbers of slices and 383 numbers of LUTs.

R. Santhosh Kumar and P. Kalpana Reddy proposed **A New Vlsi Archiecture of Parallel Multiplier-Accumulator Based on Radix-2 Modified Booth Algorith"?**

In a VLSI system, the pipeline architecture of high-speed modified Booth multipliers are used. The proposed multiplier circuits are based on the modified Booth algorithm. The pipeline technique which are the most widely used to accelerate the multiplication speed. To implement the optimally pipelined multipliers, many kinds of experiments have been conducted. The speed of the multipliers is greatly improved by properly deciding the number of pipeline stages and the positions for the pipeline registers to be inserted. The proposed modified Booth multiplier circuits in Verilog HDL and synthesized the gate-level circuits. The resultant multiplier circuits show better performance than others. Since the proposed multipliers operate at GHz ranges, they can be used in the systems requiring very high performance.

4. Proposed Methodology

The block diagram of proposed single precision floating point multiplier. It shows the process of floating point multiplication using IEEE 754 single precision format. Firstly the sign bit of the numbers are exored. Exponents of both the numbers are biased and unbiased in the adder. Both the exponent a and b are add and the bias 127 is subtracted. Mantissa of the number are multiplied in the multiplier and then normalize with left or right shift .If the exponent raises then it go to adder circuit .Before rounding exception are checked. Determine exception flags and special values for overflow and underflow condition, then perform rounding. At the last the result must are found in the final sign, exponent, mantissa block.

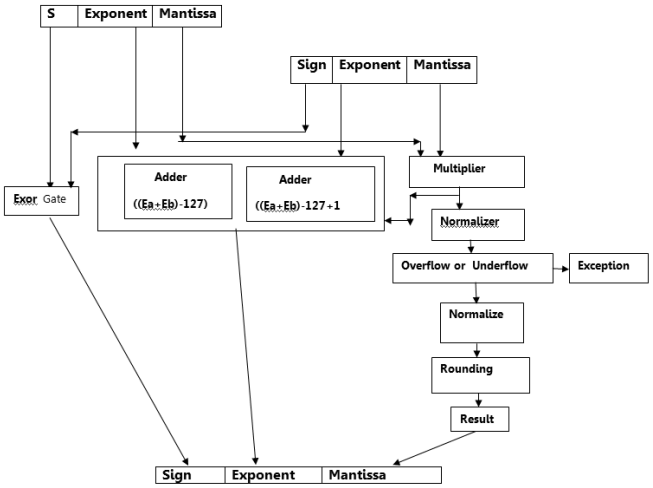


Figure 4 - Block Diagram of Proposed Single Precision Floating Point Multiplier

Compression Block

For compression, three digit decimal number is passed as input to the compressing block where Decimal to BCD converter converts the binary number in BCD encoded number and then DPD number by DPD compression module which finally compresses the BCD number.

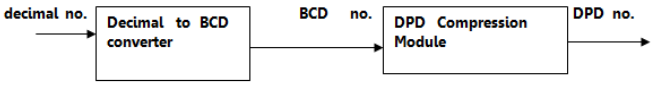


Figure 5- Compression

Expansion Block

For expansion the DPD number is passed as input to the expansion block where first it is Binary to BCD converter

converts the binary number to BCD.The BCD number is then fed to BCD to Decimal Converter for converting BCD to Decimal number.

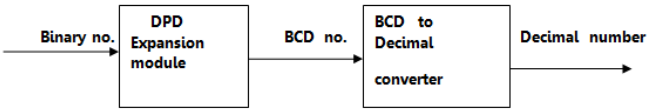
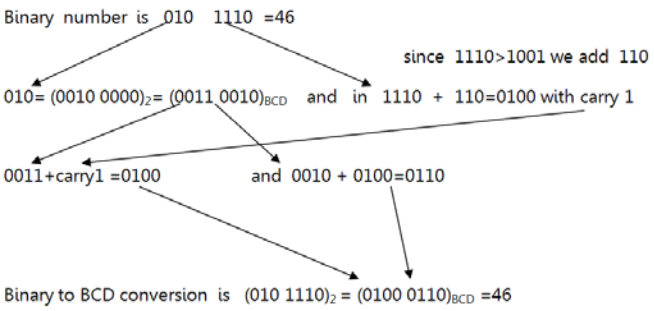


Figure 6 - Binary to BCD converter

BCD is a decimal representation of a number directly coded in binary form . It can be seen that each digit of the decimal number is coded in binary and then concatenated to form the BCD representation of the decimal number. The value of BCD numbers are same as binary number in numbers between 0 to 9.When the number is greater than 9 the value of binary number and BCD number are different. There is certain rule for writing BCD value of binary number greater than 9.

Binary to BCD conversion Calculation is



5. Simulation/Experimental Results

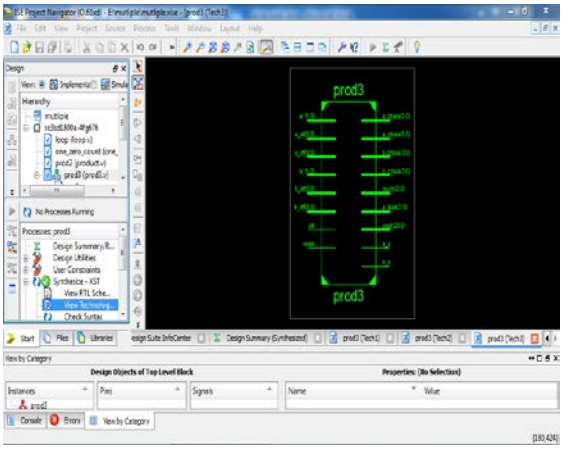


Figure 7 - RTL View

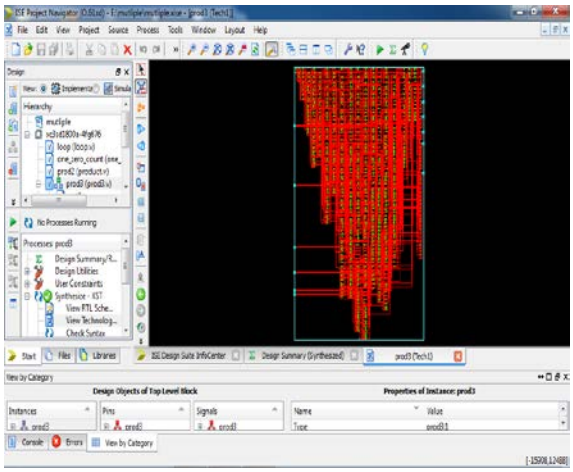


Figure 8- Schematic view

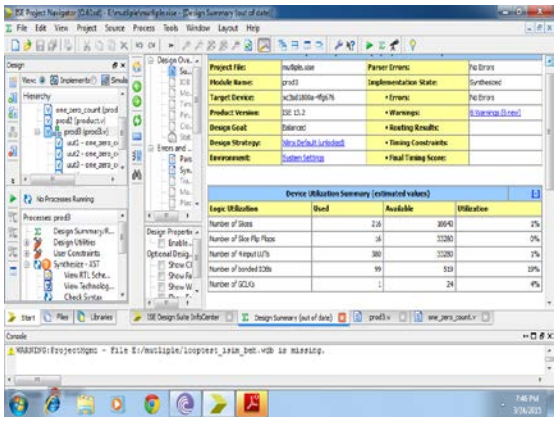


Figure 9- Design Summary

6. Conclusion

The designed arithmetic unit operates on 32-bit operands. It can be designed for 64-bit operands to enhance precision. It can be designed to perform various mathematical operation and arithmetic operation. In the presented work it is concluded that there is a reduction in latency and area.

7. Future Scopes

The designed arithmetic unit operates on 32-bit operands .The future aspect of this thesis work will be implemented the design for purpose of banking, taxation, financial analysis.ALU can be designed for DSP application.. This work has been done using Verilog language. The proposed work is implemented on FPGA.

References

[1] Kulvir Singh and Dilip Kumar "Modified BoothMultiplier with Carry Select Adder using 3-stage pipelining technique." International

Journal of Computer Applications (0975 – 8887) Volume 44– No14, April 2012.

[2] Navdeep Kaur and Rajeev Kumar patial " Implementation of Modified Booth Multiplier using Pipeline Technique on FPGA." International Journal of Computer Applications (0975 – 8887) Volume 68– No.16, April 2013.

[3] R. Santhosh Kumar and P. Kalpana Reddy "A New Vlsi Archiecture of Parallel Multiplier-Accumulator Based on Radix-2 Modified Booth Algorith"?" International Journal of Research ISSN - 2250-1991 Volume : 2, Issue : 9 Sept 2013.

[4] Cornea, Marius ; Harrison, J. ; Anderson, C. ; Tang, P. ; Schneider, E. ; Gvozdev, E. "A Software Implementation of the IEEE 754R Decimal Floating-Point Arithmetic Using the Binary Encoding Format" Computers, IEEE Transactions on Volume: 58 , Issue: 2 DOI: 10.1109/TC.2008.209 Publication Year: 2009 , Page(s): 148 – 162.

[5] Brisebarre, N. ; Louvet, N. ; Martin-Dorel, E. ; Muller, J.-M. ; Panhaleux, A. ; Ercegovac, M.D. " Implementing decimal floating-point arithmetic through binary: Some suggestion " Application-specific Systems Architectures and Processors (ASAP), 2010 21st IEEE International Conference on DOI: 10.1109/ASAP.2010.5540969 Publication Year: 2010 , Page(s): 317 – 320.

[6] Gonzalez-Navarro, S. ; Tsen, S. ; Schulte, M. " A Binary Integer Decimal-based Multiplier for Decimal Floating-Point Arithmetic" Signals, Systems and Computers, 2007. ACSSC 2007. Conference Record of the Forty-First Asilomar Conference on DOI:10.1109/ACSSC.2007.4487228 Publication Year: 2007 , Page(s): 353 - 357

[7] Hongwei Ding ; Pingping Shu ; Xiaojun Wang ; Jun Yang "A Design and Implementation of Decimal Floating-point Multiplication Unit Based on SOPC" Digital Manufacturing and Automation (ICDMA), 2012 Third International Conference on DOI: 10.1109/ICDMA.2012.9 Publication Year: 2012 , Page(s): 36 – 41.

[8] Raafat, R. ; Abdel-Majeed, A.M. ; Samy, R. ; ElDeeb, T. ; Farouk, Y. ; Elkhouly, M. ; Fahmy, A.H. "A decimal fully parallel and pipelined floating point multiplier" Signals, Systems and Computers, 2008 42nd Asilomar Conference on DOI: 10.1109/ACSSC.2008.5074737 Publication Year: 2008 , Page(s): 1800 - 1804

Author's Profile

Professor Dr. Vinod Kapse, Department of Electronics and Communication, Gyan Ganga Institute of Technology and Sciences, Jabalpur ,Madhya Pradesh , India.

Mr. Puneet Bharadwaj, Department of Electronics and Communication, Gyan Ganga Institute of Technology and Sciences, Jabalpur ,Madhya Pradesh , India

Y. Arunika Rao has received his Degree in Electronics & Communication Engineering from Gyan Ganga Institute of Technology and Sciences, Jabalpur ,Madhya Pradesh , India in the year 2008. At present she is persuing M.Tech(Embedded system and VLSI Design) in 2nd year from Department of electronics and communication, Gyan Ganga Institute of Technology and Sciences, Jabalpur ,Madhya Pradesh , India.