International Journal for Modern Trends in Science and Technology, 8(06): 243-245, 2022

ISSN: 2455-3778 online
DOI: https://doi.org/10.46501/IJMTST0806042
Available online at: http://www.ijmtst.com/vol8issue06.html

Implementation of BCD Floating Point Multiplier

Dr. A V S Swathi [1], G Lakshmi Prabha [2], D Surya Ganeswari [2], A Dilip Sai Raja[2], B Krishna Venkata Gopal[2]

[1]Assistant Professor, Raghu Institute of Technology (A), Visakhapatnam.
[2]Raghu Institute of Technology (A), Visakhapatnam.

To Cite this Article
Dr. A V S Swathi, G Lakshmi Prabha, D Surya Ganeswari, A Dilip Sai Raja and B Krishna Venkata Gopal. Implementation of BCD Floating Point Multiplier. International Journal for Modern Trends in Science and Technology 2022, 8(06), pp. 243-245. https://doi.org/10.46501/IJMTST0806042

Article Info
Received: 12 May 2022; Accepted: 06 June 2022; Published: 10 June 2022.

ABSTRACT

Multipliers play an important part in arithmetic computations, and hence it is necessary to design multipliers that run at high-speed and give precise outputs. Currently, binary multipliers are being used in ALU. But the problem with using binary number system is that they take more time as every binary result has to undergo conversion before displaying it on screen. This conversion is required because digital displaying boards can display only bcd numbers, but not binary. Also most numbers are rounded or truncated during the multiplication process due to its strict representation using only limited number of bits. This is leading to errors which cannot be tolerated in fields like financial, banking and science where the results have to be very accurate . These errors can be rectified by replacing the traditional binary multiplier with bcd multiplier. The advantage in bcd multipliers is that every digit can be exactly represented even with limited bits of 4 or 8. Thus bcd multipliers are highly advantageous while running computation in ALU. BCD multipliers are designed using vedic multipliers and binary to bcd converters. In this thesis, the implementation of various BCD mutlipliers are described. The performance of these various multipliers is compared in terms of delay and speed.

KEYWORDS: ALU, BCD, Floating point numbers

1. INTRODUCTION

The main theme of the thesis is to design single precision binary coded decimal floating point multiplier with less gate count and less delay when compared to existing single precision binary coded decimal floating point multiplier. This single precision Binary Coded Decimal floating point multiplier is designed using Vedic Maths Urdhva Trikbhyam Sutra and parallel prefix adders (ie., Kogge Stone Adder and Brent Kung Adder). Vedic maths is an ancient mathematics system derived from the sanskrit term Vedas, which meaning "knowledge.". Between 1911 and 1918, Indian mathematician Jagadguru Shri Bharathi Krishna Tirthaji founded vedic mathematics.Vedic maths uses formulas which are known as Vedic Sutras and it contains 16 sutras and 13 sub sutras. The vedic maths can surely slove numerical calculations in faster way and in minimal time, where as the normal maths consumes more time and complex in operations. Vedic maths consists of 16 sutras among which only three sutras and two subsutras can be used for multiplication operation. They are Urdhva Tiryakbhyam, Nikilam Navataschcaramam Dashatah, Anurupvena, Ekanvunena Purvena, Antyavordasake'pi. Anurupvena is a sutra that is used to multiply two numbers that are around the same base, such as 40, 40, and so on (multiples of powers of 10).

LITERATURE SURVEY:

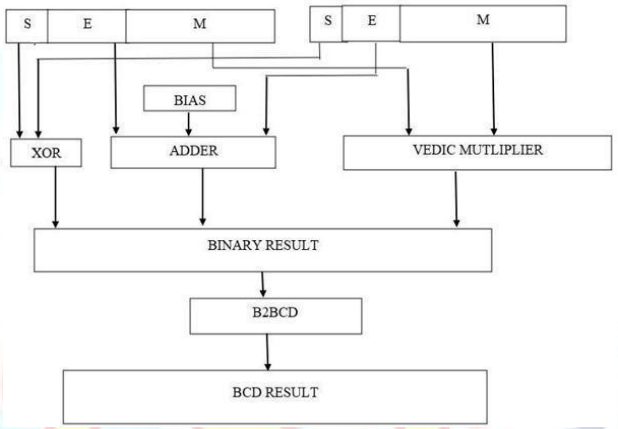
A number is expressed in fixed-point significant digits and scaled using an exponent in a fixed base. The resultant bits of a fixed point operation may be wider than the operands. The resultant bits in the multiplication step will be the sum bit of the operands. The term truncation and rounding would be used to fit the produced bits into the same size as the operand bit size. There is a chance of information loss in this scenario, which could lead to accuracy concerns. The issue is that when large numbers are analyzed, the fixed point is prone to precision loss. Another significant difficulty is that integer fixed point is inconvenient to use in processors because of overflow situations.The decimal point of a floating-point number can be shifted right or left of a fixed number. When compared to fixed-point numbers, floating-point numbers extend the range of representation. Floating point is represented by 3 terms as sign(s) ,exponent(e), mantissa(M). The standard format for floating point is respresented as(−1^s)(b^e)(M).

The IEEE format for floating point arithmetic (IEEE754) was established in the year 1985 by the Institute of Electrical and Electronics Engineering(IEEE). Half, Single, Double, Extended and Quad are the 5 precisions that are formulated by the IEEE standard. The single precision binary floating point is represented by 32-Bits which includes 8-Bit Exponent, 23-Bit Mantissa, Sign bit is 1-Bit.The double precision binary floating point is respresented by 64-Bits which includes 11- Bit exponent, 52-Bit mantissa, 1-Bit sign. Here the sign bit is 1 bit width and if sign bit is 1 then it represents the given number is negative and if sign bit is '0' represents the number is positive. N is 1 bit width which is used to represent whether the given number is BCD or not. Where Exponent is 6 bit wide and Mantissa is 24bit wide.In this thesis single precision BCD floating point is being used. Fig depicts the BCD single precision BCD floating point representation.

EXISTING METHOD:

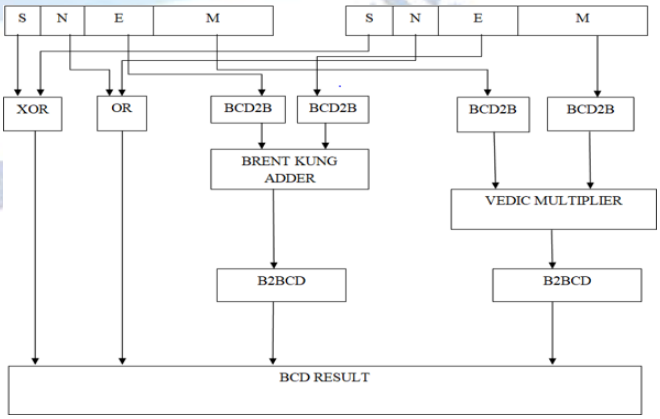
Parallel prefix adders are kind of adder that use prefix operation in order to do efficient addition. These adders are suited for binary addition with wide word. Among parallel prefix adders in this thesis two parallel prefix adders as Kogge Stone, Brent Kung adders are considered. BCD Floating Point Multipliers can be implemented in the following ways Single Precision Binary Floating Point Multiplier with Binary to BCD converter, Single Precision Binary Coded Decimal Floating Point Multiplier (Method 1), Single Precision Binary Coded Decimal Floating Point Multiplier (Method 2).

The 32 bit BCD floating point number is given as input to the Binary floating multiplier with binary to BCD converter. Here Exponent term is 6 bit wide and Mantissa is 24 bit. Sign bit is obtained by XORing of the two sign bits of the two numbers.

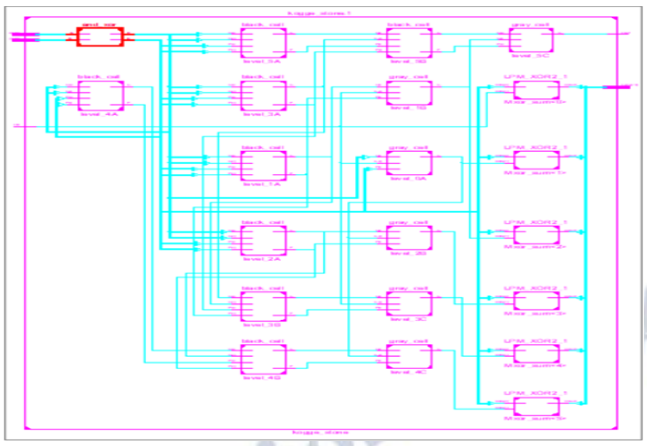


PROPOSED DESIGN:

To further decrease delay and gate count BCDFPM with Brent Kung Adder is used. Using a BCD to Binary converter, the mantissa and exponents are transformed to binary in this BCD Floating Point Multiplier with Brent Kung Adder. The sign bit is computed using XOR, while N is computed using the OR operation. Advantages of BCD Floating Point Multiplier with Brent Kung Adder are no error in result, power dissipation is less, circuits works faster and the disadvantage is circuit size is large.



SIMULATION RESULTS:



S.No		SPBFPM	SPBCDFPM (Method 1)	SPBCDFPM (Method 2)	BCDFPM with Brent Kung Adder
1	Delay	120.611ns	228.238ns	221.79ns	217.19ns
2	No.of slices	1157	1844	1873	1869
3	output	75%	No error	No error	No error

CONCLUSION:

Inthisthesis,SingleprecisionBinaryCodedDecimalFloatingPointMultiplierisimplemented using vedic multiplier Urdhva Trikabhyam sutra and parallel prefix adder(i.e, brentkung adder). The major goal of this thesis is to design single precision Binary Coded Decimalflaoting point multiplier with reduced the delay and gate count when compared to existing singleprecision binary floating point multiplier with binary to BCD converter, single precision BinaryCoded Decimal Floating Point Multiplier (method 1) and Single Precision Binary Coded DecimalFloatingPoint Multiplier (method2).

The delay for the BCD Floating Point Multiplier with Brent Kung Adder is reduced by 4% when compared with the Single Precision Binary Coded Decimal Floating Point Multiplier (method 2). The error % for BCD Floating Point Multiplier with Brent Kung Adder is reduced by 75% when compared with the Single Precision Binary Floating Point Multiplier (SPBFPM). Thus,it can be concluded that if low delay and low gate count is needed than the BCD Floating PointMultiplier with Brent Kung Adder can be used. BCD Floating Point Multiplier with Brent KungAdder has better performance metrics than the Single Precision Binary Coded Decimal FloatingPoint Multiplier (Method 1) and Single Precision Binary Coded Decimal Floating Point Multiplier (Method2) in terms of delay and gate count.

In future work, double precision BCD floating point mutlipliers can be designed. Kogge Stoneadder and Brent Kunga can be replaced by other parallel prefix adders in the architecture for exponentadditionand also thismultipliercan beimplemented inarithmeticdesigns.

Conflict of interest statement

Authors declare that they do not have any conflict of interest.

REFERENCES

[1] V. Ramya, R. Seshasayanan, "Low power single precision BCD floating–point Vedicmultiplier" Microprocessorsand Micro systems72(2019)102930.

[2] R. Anjana, B. Abishna, M.S. Harshitha, E. Abhishek, V. Ravichandra, M.S. Suma,"ImplementationofVedicmultiplierusingKogge-stoneadder",InternationalConferenceon Embedded Systems-(ICES 2014).

[3] R.Gupta, R.Dhar, K.L.Baishnab, Jishan Mehedi, "Design of High performance Vedicmultiplierusingcompressor",IEEE–InternationalConferenceonAdvancesinEngineeringandTechnology(ICAET2014).

[4] Sushma S. Mahakalkar, Sanjay L. Haridas, "Design of high performance IEEE754floatingpointnumberMultiplierUsingVedicmathematics",IEEE,SixthInternational Conference on Computational Intelligence and Communication Networks(2014).

[5] Josmin Thomas, R.Pushpangadan, Jinesh.s, "Comparative study of performance Vedicmultiplier on the basis of adders used ", 2015 IEEE International WIE Conference onElectricaland ComputerEngineering(WIECON-ECE).IEEE,2015.

[6] Havaldar, Soumya, and K. S. Gurumurthy. "Design of Vedic IEEE 754 floating pointmultiplier",2016IEEEInternationalConferenceonRecentTrendsinElectronics,Information&CommunicationTechnology(RTEICT).IEEE,2016.

[7] Jais, Amish, and Prasanna Palsodkar. "Design and implementation of 64 bit multiplierusing vedic algorithm",2016 International Conference on Communication and SignalProcessing(ICCSP).IEEE, 2016.

[8] Bhavesh,Sharma,RuchikaMishra,NileshDidwania,AritraMitra,AmitBaksi,"Comparison of single precision floating point Multiplier using different multiplieralgorithms"