International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE)
ISSN: 0976-1353 Volume 20 Issue 2 – FEBRUARY 2016.

DESIGN AND IMPLEMENTATION OF LOW POWER FLOATING POINT UNIT USING RECONFIGURABLE DATA PATH: A SURVEY

Lathapriya V [#1] and Sivagurunathan P T [*2]

[#] *II M.E, VLSI design, M.Kumaraswamy College of Engineering, Tamil nadu, INDIA*
[*] *Assistant Professor, Department of ECE, M.Kumaraswamy College of Engineering, Tamil nadu, INDIA*

***Abstract*-Floating point arithmetic is widely used in many areas especially scientific computation and signal processing. The main objective of this paper is to reduce the power consumption and to increase the speed of execution and the implementation of floating point multiplier using sequential processing on the reconfigurable hardwareFloating Point (FP) addition, subtraction and multiplication are widely used in large set of scientific and signal processing computation. In addition, the proposed designs are compliant with IEEE-754 format and handles over flow, under flow, rounding and various exception conditions. The adder/subtractor and multiplier designs can achieve high accuracy with increased throughput. This approach is To provide a high accuracy reconfigurable adders and multipliers for floating point arithmetic. To understand how to represent single and double precision floating point architecture in single architecture using quantum flux circuits for DSP applications.**

Keywords – Floating point unit, Delay ,High Throughput

I. INTRODUCTION

Floating point addition and multiplication are the most frequent floating point operations. Many scientific problems require floating point arithmetic with high level of accuracy in their calculations. A floating point number representation can simultaneously provide a large range of number and a high degree of precision. The IEEE 754 floating point standard is the most common floating point representation used in modern microprocessors.

Efficient use of the chip area and resources of an embedded system poses a great challenge while developing algorithms in embedded platforms for hard real time applications, like digital signal processing, control systems and so on. As a result, a platform of modern microprocessors is often dedicated to hardware for floating point computation. Previously, silicon area constraints have limited the

complexity of the floating point unit or FPU .Advances in integrated circuit fabrication technology have resulted in both smaller feature sizes and areas. It has therefore become possible to implement more sophisticated arithmetic algorithms to achieve higher FPU performance.

The recent advancements in the area of Field Programmable Gate Array (FPGAs) has provided many useful technique and tools for the development of dedicated and reconfigurable hardware employing complex digital circuits at the chip level. Floating point Addition, Multiplication is most widely used operation in DSP/Math processors, Robots, Air Traffic Controller, Digital computers, because of its raising application the main emphasis is on the implementation of floating point multiplier effectively such that it uses less chip area with more clock speed.

II. FORMATS

i. Fixed point Format

A value of a fixed-point data type is essentially an integer that is scaled by a specific factor determined by the type. For example, the value 1.23 can be represented as 1230 in a fixed-point data type with scaling factor of 1/1000, and the value 1230000 can be represented as 1230 with a scaling factor of 1000.

Unlike floating-point data types, the scaling factor is the same for all values of the same type, and does not change during the entire computation.

ii. Floating point format

One of the ways to represent real numbers in binary is the floating point formats.

There are two different formats for the IEEE 754 standard. Binary interchange format and Decimal interchange format. In the multiplication of floating point numbers involves a large dynamic range which is useful in DSP applications.

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 two decimal places can represent the numbers 12345.67, 123.45, 1.23 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 precision.

The IEEE floating point standard defines both single precision and double precision formats. Multiplication is a core operation in many signal processing computations, and as such efficient implementation of floating point multipliers is an important concern. This paper presents the FPGA implementation of double precision floating point multiplier using Quartus tool. The floating point numbers is based on scientific notation[4]. A scientific notation is just another way to represent very large or very small numbers in a compact form such that they can be easily used for computations.

Floating point number consists of three fields:

1. Sign (S): It used to denote the sign of the number i.e. 0 represent positive numbers and 1 represent negative number.
2. Significant or Mantissa (M): Mantissa is part of a floating point number which represents the magnitude of the number.

3. Exponent (E): Exponent is part of the floating point number that represents the number of places that the decimal point (or binary point) is to be moved.

Number system is completely specified by specifying a suitable base β, significant (mantissa) M, and exponent E.

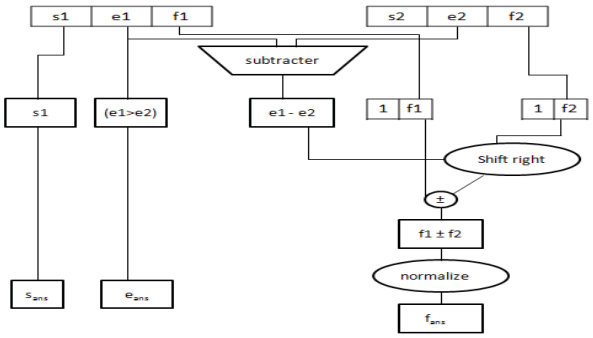

Figure 1 FPU architecture

III.LITERATURE REVIEW

The following section consists of several architecture designs for FPU to undergo the survey to obtain the better solution for low power operation of LSRDP.

A. DESIGN AND IMPLEMENTATION OF SINGLE PRECISION PIPELINED FLOATING POINT CO-PROCESSOR

Manisha Sangwan explained that Floating point numbers are used in various applications such as medical imaging, radar, telecommunications Etc. This paper deals with the comparison of various arithmetic modules and the implementation of optimized floating point ALU. Here pipelined architecture is used in order to increase the performance and the design is achieved to increase the operating frequency by 1.62 times. The logic is designed using Verilog HDL. Synthesis is done on Encounter by Cadence after timing and logic simulation. Carry Look Ahead (CLA) adder save this propagation time by generating and propagating this carry simultaneously in the consecutive blocks. So, for faster operation this carry look ahead adder is used and for division process Goldsmith (GDM) algorithm is used. There are different algorithms available for multiplication such as Booth, modified booth, Wallace, Bough Wooley, Braun multipliers. But issue with multipliers is speed and regular layout so keeping both the parameters in mind modified booth algorithm was chosen. It is a powerful algorithm for signed-number multiplication, which treats both positive and negative numbers uniformly. Ultimately these individual blocks are clubbed to make Floating point based ALU in a pipelined manner to minimize the power and to increase the operating frequency at the same time. These comparative analyses are done on cadence and Xilinx both.

B. EMBEDDED COMPLEX FLOATING POINT HARDWARE ACCELERATOR

Amin Ghasemazar, Mehran Goli, Ali Afzali-Kusha provide a common solution to deal with the large execution delay and power consumption in modern processors is to use a co-processor that is capable of executing specific types of instructions in parallel with the main processor.In this paper, we present a new pipelined Multiple Input Multiple Output (MIMO) ALU that is accelerated to perform complex floating point operations including addition, subtraction, multiplication and division. Our proposed Accelerated complex floating point ALU (AALU) has significantly lower delay (in clock cycles) compared to an earlier work that has an Instruction Set Extension (ISE) based architecture. Hardware is implemented with MIMO (Multiple Input Multiple output) data path in a MIPS processor as well as NIOS II configurable and extensible processor using Instruction Set Extension (ISE). The architecture of AALU to further speed-up the design by utilizing more architecture level techniques such as parallel processing or pipelining.

C.CORRECTLY ROUNDED ARCHITECTURES FOR FLOATING-POINT MULTI-OPERAND ADDITION AND DOT-PRODUCT COMPUTATION

Yao Tao Gao Deyuan Fan Xiaoya presents hardware architectures performing correctly rounded Floating-Point (FP) multi-operand addition and dot-product computation, both of which are widely used in various fields, such as scientific computing, digital signal processing, and 3D graphic applications. A novel realignment method is proposed to solve the catastrophic cancellation and multi-sticky bits. Implementation results show that our architectures not only can produce correctly rounded results, whose errors are less than 0.5 ULP(Unit in the Last Place).Two problems may incur: catastrophic cancellation and multi-sticky bit. Only one rounding operation is performed in both of the proposed FP multi-operand adder and dot-product computation unit. Long-Adder architecture, which is a data parallel architecture and has the same significand adder with long accumulator architecture.Correct Rounding of FADDn and FDPn not only minimizes the maximum error of the result, but also makes the result deterministic, which can improve the portability of software. As the transistors become cheaper and cheaper, the cost of using the hardware method to accelerate complex but frequent operations is no longer unaffordable. Our architectures can reduce the delay of multi-operand addition and dot-product computation compared with the network architecture.

D. A HIGH SPEED FLOATING POINT DOT PRODUCT UNIT

Akash Kumar Gupta implemented in two approaches conventional parallel and fused approach Products are no need to be rounded and only sum has to be rounded. Thus the precision is improved. The presented fused dot product unit decreases the complexity of hardware implementation due to fused technique by allowing shared units compared to conventional parallel approach with discrete units. These all provides fused dot product unit with reduced delay and reduced area and high accuracy as only one rounding operation is performed. The fused primitives are faster, smaller, than the parallel approaches and provide a slightly more accurate result.The Fused dot product unit is more accurate than conventional parallel dot product unit in sense only one rounding is performed over 3 rounding in parallel approach.

E. LOW-POWER LEADING-ZERO COUNTING AND ANTICIPATION LOGIC FOR HIGH-SPEED FLOATING POINT UNITS

GiorgosDimitrakopoulos, a new leading-zero counter (or detector) is presented. New boolean relations for the bits of the leading-zero count are derived that allow their computation to be performed using standard carry-lookahead techniques. The new

circuits can be efficiently implemented either in static or in dynamic logic and require significantly less energy per operation compared to the already known architectures. Efficient technique for handling the error of the leading-zero anticipation logic is also presented.Normalization involves the use of a leading-zero counting (LZC), or detection unit, and a normalization shifter, while in almost all cases, leading-zero anticipation (LZA) logic is also employed to speed up the computation. Our goal is to clarify which part of the prediction circuit that consists of the LZA logic and the LZC unit, is more critical in terms of energy and delay for the performance of the whole circuit. The approach we adopted relies on a hybrid technique where dual-rail dynamic logic is used only in specific parts of the circuit, while the majority of the gates produces single rail outputs. The derivation of this technique is based on a simple observation. A novel technique for handling the possible error of LZA logic was described that imposes the minimum overhead to the normalization shifter without introducing any further limitation.

F. A MULTIPLE-MODE FLOATING-POINT MULTIPLY-ADD FUSED UNIT FOR TRADING ACCURACY WITH POWER CONSUMPTION

Kun-Yi Wu, Chih-Yuan Liang, Kee-Khuan Yu, and Shiann-RongKuang showed wide use of floating-point (FP) multiply and accumulate operations in multimedia and digital signal processing applications, many modern processors adopt FP multiply-add fused unit (MAF) to achieve high performance, improve accuracy and reduce power consumption. FP arithmetic units usually occupy the major portion of a processor's area and power dissipation, a multiple-mode FP multiply-add fused unit which utilizes the iterative multiplication and truncated addition techniques to support seven operating modes with various errors for low power applications. It can execute either one multiply-accumulate operation with three modes, one multiplication operation with two modes or one addition operation with two modes. When compared to the traditional IEEE754 single-precision FP MAF, the proposed unit has 4.5% less area and 23% longer delay to achieve multiple modes which can sacrifice a little (< 1%) accuracy for saving large (> 33%) power consumption. Implementation results exhibited that the proposed MMAF unit can efficiently reduce about 33% and 43% power consumption with 0.130% and 0.909% accuracy lost for MAC operation.

G. UNIFIED ARCHITECTURE FOR DOUBLE/TWO-PARALLEL SINGLE PRECISION FLOATING POINT ADDER

Manish Kumar Jaiswal, Ray C. C. Cheung, M. Balakrishnan, and Kolin Paul stated Floating point (F.P.) addition is a core operation for a wide range of applications. This brief presents an area-efficient, dynamically configurable, multiprecision architecture for F.P. addition. Key components involved in the F.P adder architecture, such as comparator, swap, dynamic shifters,leading one-detector (LOD), mantissa adders/subtractors, and rounding circuit, have been redesigned to efficiently enable resource sharing for both precision operands with minimal multiplexing circuitry. Compared to a standalone DP adder with two SP adders, the proposed unified architecture can reduce the hardware resources by ≈35%, with a minor delay overhead. we have presented an architecture for floating point adder with on-the-fly dual precision support, with both normal and sub-normal support, and exceptional case handling. It supports double precision with dual single precision (DPdSP) adder computation.

H.SPLIT-PATH FUSED FLOATING POINT MULTIPLY ACCUMULATE (FPMAC)

Suresh Srinivasan, KetanBhudiya, RajaramanRamanarayanan, explained about Floating point multiply-accumulate (FPMAC) unit is the backbone of modern processors and is a key circuit determining the frequency, power and area of microprocessors. FPMAC unit is used extensively in contemporary client microprocessors, further proliferated with ISA support for instructions like AVX and SSE and also extensively used in server processors employed for engineering and scientific applications. In this work we have three key innovations to create a novel double precision FPMAC with least ever gate stages in the timing critical path: a) Splitting near and far paths based on the exponent difference (d=Exy-Ez= {-2,-1,0,1} is near path and the rest is far path), b) Early injection of the accumulate add for near path into the Wallace tree for eliminating a 3:2 compressor from near path critical logic, exploiting the small alignment shifts in near path and sparse Wallace tree for 53 bit mantissa multiplication, c) Combined round and accumulate add for eliminating the completion adder from multiplier giving both timing and power benefits. Our design by premise of splitting consumes lesser power for each operation where only the required logic for each case is switching. Splitting the paths also provides tremendous opportunities for clock or power gating the unused portion (nearly 15-20%) of the logic gates purely based on the exponent difference signals. The split path design provides a natural way for gating opportunities and even under normal case may lead to 15-20% of lesser switching gates based on the near or far path operation. This innovation can significantly help the microprocessor designs with fast timing area and power convergence. 1 GHz Leading Zero Anticipator Using Independent Sign-Bit Determination Logic K. T. Lee and K. J. Nowka, Leading Zero Anticipators (LZAs) predict the position of the leading logical one by examining the mantissas of the adder and the addend in parallel with the addition. A Leading Zero Anticipator was fabricated in a 1 GHz PowerPC microprocessor. It initially computes both the positive and negative sum and shift amounts. In parallel, it computes the sign-bit to select one in a final muxing stage. This organization enables the LZA to operate at measured frequencies of up to 1.OGHz unit. This helps to speed the normalization process in normalized floating-point addition or fused multiplication-addition units by generating the proper normalization shift amounts. In conventional designs, either the smaller magnitude operand must be subtracted from the larger or both a leading-zero anticipator and a leading-one anticipator are both implemented and the adder sign-bit selects the normalization shift amount. These solutions require either a magnitude compare or additional delay and load due to the multiplexor dependent on the adder sign-bit. Furthermore, taking the sign-bit from the adder generally requires a long wire which produces extra signal delay and degrades signal integrity. It is, thus, desirable that the sign-bit be determined independently without too much of area and power overhead.

I. DELAY-OPTIMIZED IMPLEMENTATION OF IEEE FLOATING-POINT ADDITION

P. M. Seidel and G. Even, showed that dual path FP-addition algorithm, the common criterion for partitioning the

computation into two paths has been the exponent difference. The exponent difference criterion is defined as follows: The near path is defined for small exponent differences (i.e.,1;0;þ1), and the far path is defined for the remaining cases. The adopted AMD algorithm is the fastest among all considered FP adder algorithms in the logic level analysis The FP-adder implementation corresponding to this SUN patent uses a special path selection condition that simplifies the "near"-path. The "near"-path deals only with effective subtractions and no rounding is required.The delay of our FP-adder algorithm using the Logical Effort Model the delay-optimal implementation of our FP addition algorithm by considering optimized gate sizing and driver insertion. For comparisons we scale the delay estimations from this model to units of FO4 inverter delays (inverter delay with a fanout of four). The algorithm is a two-staged pipeline partitioned into two parallel paths called the R-path and the N-path. A parallel-prefix adder is used to compute the sum and the incremented sum of the significands. The FP-adder design achieves a low latency by combining various optimization techniques such as: A nonstandard separation into two paths, a simple rounding algorithm, unification of rounding cases for addition and subtraction, sign-magnitude computation of a difference based on one's complement subtraction, compound adders, and fast circuits for approximate counting of leading zeros from borrow-save representation.

IV.COMPARISION TABLE

Floating point arithmetic delay

Fixed point adder	21.414
Floating point adder	15.902
Floating point subtractor	12.780
Floating point fused Add Sub (FAS)	16.086
Floating point multiplier	14.583
Floating point Fused Multiply Add	22.14

Floating Point Arithmetic Power(mW)

Fixed point adder	93
Floating point adder	203
Floating point subtractor	203
Floating point fused Add Sub	203
Floating point multiplier	24
Floating point Fused Multiply Add	93

V.CONCLUSION

The floating point calculations consume more power and occupy larger area due to high dynamic range. Using fused floating point techniques the area and power have been reduced. For that reason fused FMA and FAS concept used in FFT butterfly radix-2 calculation**.** Based on the data flow analysis, the proposed fused floating-point adder can be split into three pipeline stages, which increases the throughput. The proposed floating point arithmetic unit can be achieved by adopting the intermediate gating threshold voltage method.

REFERENCES

[1] Jongwook Sohn, Member, IEEE, and Earl E. Swartzlander, Jr (2014) A Fused "Floating-Point Three-Term Adder".IEEE Transactions.

[2] Alexandre F. Tenca (2009) "Multi-Operand Floating-Point Addition" IEEE International Symposium.

[3] Anant G. Kulkarni,Dr. Manoj Jha, Dr. M. F. Qureshi; (2014). Design and Simulation of Eight Point FFT Using VHDL; IJISET - Vol. 1 Issue 3

[4] Anant G. Kulkarni, Dr. M. F. Qureshi, Dr. Manoj Jha,(2013) "Discrete Fourier Transform: Approach To Signal Processing" IEEE.,vol.3.issue 10.

[5] Anant G. Kulkarni & Sudha Nair July-December 2009, "Design and Implementation of Frequency Analyser using VHDL", International Journal of Electronics Engineering , Volume 1, Number 2, pp. 265-268

[6] Bhaskar J. "A VHDL Primer" Pearson Printice Hall Publication Third Edition 2007.

[7] Chih-Yuan Liang, Kee-Khuan Yu, And Shiann-Rong Kuang (2012) "Multiple-Mode Floating-Point Multiply-Add Fused Unit For Trading Accuracy With Power Consumption"IEEE; Page(s): 429 – 435.

[8]Claudio Brunelli, Fabio Garzia, Jari Nurmi. J Real-Time Image Proc (2008). "A coarse-grain reconfigurable architecture for multimedia applications featuring subword computation capabilities".

[9] Daumas, M. Matula, D.W. Jul 1993. "Design of a Fast Validated Dot Product Operation" 11th Symposium on Computer Arithmetic, 1993.Proceedings, pp 62 - 69, 29 Jun-2.

[10]David Elam and Cesar Iovescu. Sep. 2003. "A Block Floating Point Implementation for an N-Point FFT on the TMS320C55x DSP". Application Report SPRA948 page1-11 .