International Journal of Applied Engineering Research
ISSN 0973-4562 Volume 10, Number 9 (2015) pp. 22535-22544
© Research India Publications
http://www.ripublication.com

Design and Implementation of IEEE 754 Compatible Dual Mode Floating Point Coprocessor

Sriadibhatla Sridevi[1], Alok Jadhav[2], Monika Gupta[3], Snehal Kate[4]
[1] *Associate Professor, School of Electronics Engineering, VIT University, Vellore, India*
[2,3,4] *PG Student, School of Electronics Engineering, VIT University, Vellore, India.*
Email: sridevi@vit.ac.in

Abstract

The floating point coprocessor is one of the essential subsystems of modern processors. It is commonly employed in implementing algorithms like FFT, convolution and consequently a cardinal piece of many signal processing applications. This paper presents an architecture for low power and low area implementation of IEEE-754 compatible dual mode floating point coprocessor (DMFCP) and it's interfacing with the main processor . This coprocessor supports both double precision and single precision arithmetic and performs addition, subtraction, multiplication and division operations. Two parallel single precision approaches are used to implement double precision arithmetic, which facilitates the hardware reuse. To further optimize the area and power costs, addition and subtraction operations as well as multiplication and division units are fused into a single Add-Sub unit and Mul-Div unit respectively. To estimate the area and delay, DMFPC is implemented using Verilog and synthesized using cadence RTL compiler where the design is targeted for 180nm TSMC technology with proper design constraints.

Keywords: Coprocessor, floating point, single precision, double precision, dual mode**.**

Introduction

A coprocessor is used in coordination with the main processor in order to decrease the load on the processor. It supplements the functions of the main processor. Co-processor performs various operations which include floating point arithmetic operations, complicated signal processing and image processing operations, along with FFT and convolution, data encryption, I/O and memory interfacing [1]. By assigning some of the most frequently required larger arithmetic operations to Co-

processor, one can reduce the work burden on the main processor. A coprocessor is an additional computer system, for supplementary operations out of which an additional execution overhead can be avoided.

Now a days, Floating point architectures are becoming dominant than fixed one because of their accuracy and wide range availability. A plenty of research has been made in single precision floating point arithmetic [2]. This paper contains dual mode of operation which provides flexibility of choosing either single precision or double precision floating point arithmetic. The format of single precision and double precision numbers is shown in Table1.

In DSP applications, there are some areas where double precision arithmetic is preferred extensively. Double precision Floating point arithmetic logic unit is backbone of coprocessor which decides the area, power, and frequency. Basically it is an arithmetic unit which performs computerized mathematical operations. It has numerous applications in computerized co-processors, various computational circuits, and so forth [1]. With the progression of VLSI technology, performing complex arithmetic operations with big numbers have become a generous problem, and also it is not possible to implement such system practically. IEEE 754 provides floating point standard, through which calculations with large numbers and complex computation required in many digital applications have become easy and much comfortable with no overhead [3] [4].

Along with the dual mode floating point coprocessor, this paper also contains design of interfacing handshaking protocol of coprocessor with the microprocessor. The design is totally area and speed optimized. The implementation is carried out in Verilog HDL and Synthesized with cadence standard cell library. Physical design is also achieved at the last. In this paper, section II describes the coprocessor and its interfacing. Double precision floating point adder-subtractor, multiplier-divider are described in section III and IV respectively.

Table 1: Format of Single and Double Precision Numbers

Type/Bit Size	**Sign**	**Exponent**	**Mantissa**	**Bias**
Single precision	1 bit	8 bits	23 bits	127 bits
Double Precision	1 bit	11 bits	52 bits	1023

Coprocessor

Generally, processor operates on a high clock frequency, and a complicated DSP application takes more computation time. Normally processor unnecessarily spends its time for computation. Hence, coprocessor works along with the processor to compute such specific complex tasks so that processor won't be overloaded. The interfacing diagram of processor and coprocessor is as shown in Figure 1.

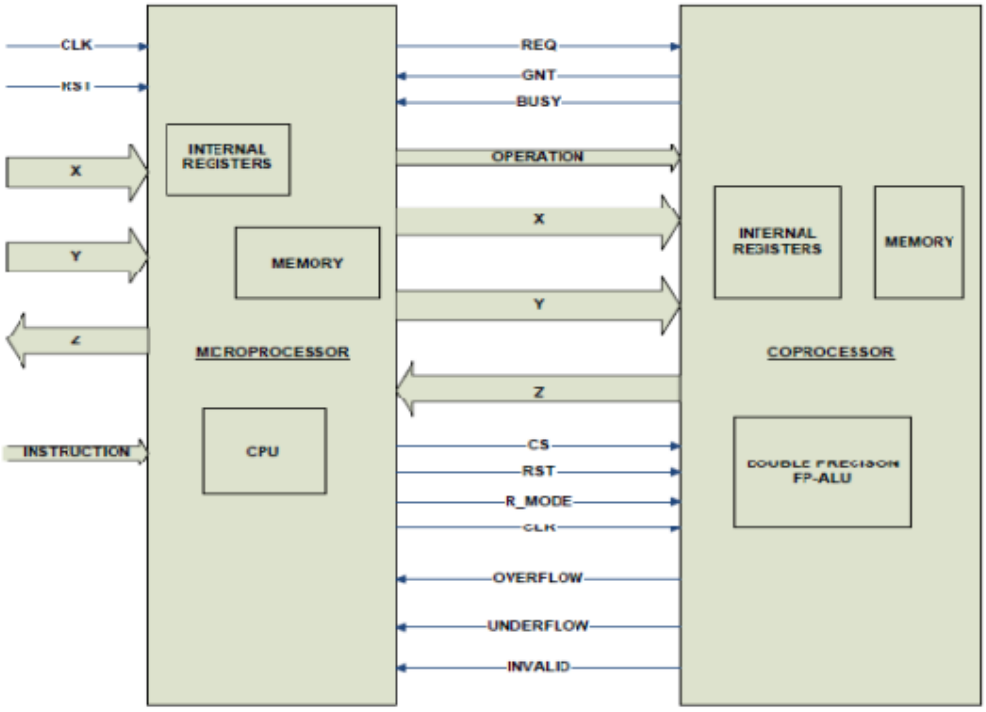

Figure 1: Interfacing of Coprocessor With Main Processor

The instruction is given from user to the processor. After getting instruction, fetching, decoding and execution is carried out sequentially. In this case, for co-processor, instruction is fetched and decoded by processor itself and given to co-processor for execution with some control signals. Identification of the instruction is done by processor itself. The handshaking signals are shown in Figure1. Whenever processor wants to communicate with the co-processor, it will send REQ signal as a request. After getting the request, co-processor acknowledges to the processor by sending GNT or busy signals. The signals indicates the status of co-processor with either busy or request grant signal. CS and RST are used as chip select and reset signals respectively.

Flag status of coprocessor is given by overflow, underflow, invalid signal pins. Operation signal defines the type of operation which is carried out in DMFCP. Apart from these signals, input and output pins are also shown in this figure. The handshaking protocol is designed with comparison to the standard coprocessor designs [5]. The internal registers and memory are separate for both units. Thus by interfacing coprocessor, processor can easily utilize its time to do other operations [6] [7].

Add-Sub Unit

Addition of floating point numbers is a core arithmetic operation in the field of scientific and engineering computations. Since from past many years, considerable

research has been made for improving the architecture of floating point operations. By looking at those results, we focus on unified multi precision architectures with the help of single precision. In literature, many authors have already focused on multi-precision operations of floating point. Hence, this work contains two parallel single precision approaches for operation over the double precision and single precision simultaneously. In research, it has been founded that dual mode single precision based double precision is much faster than existing double precision approach [2]. Figure 2 shows the two parallel single precision adder and subtractor.

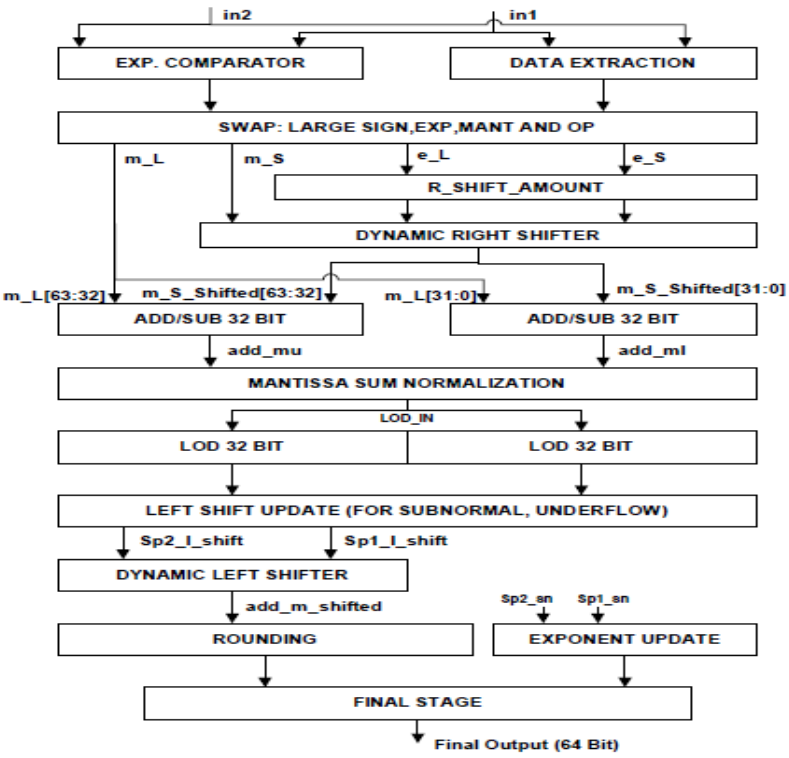

Figure 2: Double precision Floating Point Add-Sub Unit

In case of floating point addition exponents are compared and accordingly addition or subtraction of mantissa is performed. Here, double precision floating point number is divided in to two parallel single precision format. The actual operation is performed just like single precision arithmetic.

Here, main numbers are divided in to two different parts and then data is extracted in the form of sign, exponent, and significand individually. Then, in parallel way, two exponents are compared and right shift amounts are generated for smaller significand. Finally, according to that, addition or subtraction of significand is achieved as per the single precision floating arithmetic. Leading one detector (LOD) is used with normalization unit in order to finalize the result. Sign bit is considered to decide the type of operation. Rounding block is designed with reference to various rounding modes used in arithmetic. At the final stage, all results are combined and perfect double precision number is formed.

In case of co-processor design, both units are totally fused in order to optimize area and performance. The architecture is common for both operations, only the type of instruction decides exact operation to be performed i.e. either add or subtract [8]. In addition to that extra pipelining is also included in the structure in order to optimize the structure.

Mul-Div Unit

Floating point multiplication and division are most versatile operations in DSP applications. The performance is decided by these two operations as they are more complicated and time consuming in case of double precision floating point arithmetic [9]. Here also the architecture is fused for both operations in order to optimize area and speed.

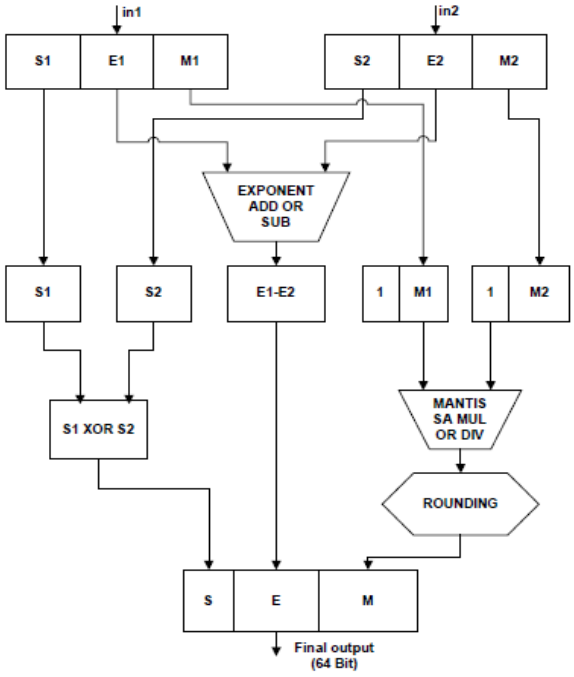

Figure 3: Double Precision floating point Mul-Div unit

In case of double precision floating point multiplication, exponents are added and significands are multiplied, whereas for division, exponents are subtracted and significands are divided [9]. In both the cases, resultant sign bit is Ex-OR of both sign bit. The exact operation is decided by operation control signal of co-processor.

Generally, for multiplication partial products are generated first and then added to each other. Here, in this work, partial products are generated by Radix-8 booth modified bit pair encoding method, which is highly precise and fast.

Table 2: Radix-8 Modified Booth Recoding Table

A3	A2	A1	A0	Operation
0	0	0	0	+0
0	0	0	1	+B
0	0	1	0	+B
0	0	1	1	+2B
0	1	0	0	+2B
0	1	0	1	+3B
0	1	1	0	+3B
0	1	1	1	+4B
1	0	0	0	-4B
1	0	0	1	-3B
1	0	1	0	-3B
1	0	1	1	-2B
1	1	0	0	-2B
1	1	0	1	-B
1	1	1	0	-B
1	1	1	1	-0

The radix 8 booth encoding table is as shown in Table 2. In case of multiplication of A and B, A is divided into overlapping groups of 4 bit numbers, and according to that B is multiplied by factor +1,+2,+3,+4,-4,-3,-2,-1 etc. As we know left shift operation defines the multiplication by 2, hence same concept has been used to generate partial products with minimum hardware. The addition of partial products is done by using Wallace tree with the help of 5:2, 4:2, and 3:2 compressors [9]. The compressors are specifically used in order to reduce the complexity of the design. Hence, overall architecture forms booth recoded Wallace tree multiplier. At the end, result of multiplication is perfectly rounded and normalized and final double precision number is formed [10] [11].

Floating point division comprises subtraction of the exponents and actual binary division of mantissa of given two numbers. The resultant sign bit is calculated by simple Ex-OR operation. Figure 4 shows exact division of 52 bit significands of two floating point numbers. Here, initially one of the numbers is stored in lower nibble of remainder and other one is stored at upper nibble of divisor provided divisor and remainder are taken as twice the size of original number [5].

In the division algorithm, every time divisor is subtracted from remainder and checked for the result. If result is negative then it is restored first and then 0 is appended at right side of quotient else logic 1 is appended at the right side of quotient without restoring the remainder. Whole process repeats for 2N number of iterations for the given significand division. Generally, for 52 bit division, main result of quotient is 26 bits hence rounding doesn't play that much role [12] [13]. At the end, final result is combined in order to form final double precision number. In this work, all the architectures are combined and designed with control signals. Operation signal

coming from processor decides the type of operation that co-processor is supposed to perform. In case of multiply divide structure also the pipelining is done in order to enhance the throughput of the given structure.

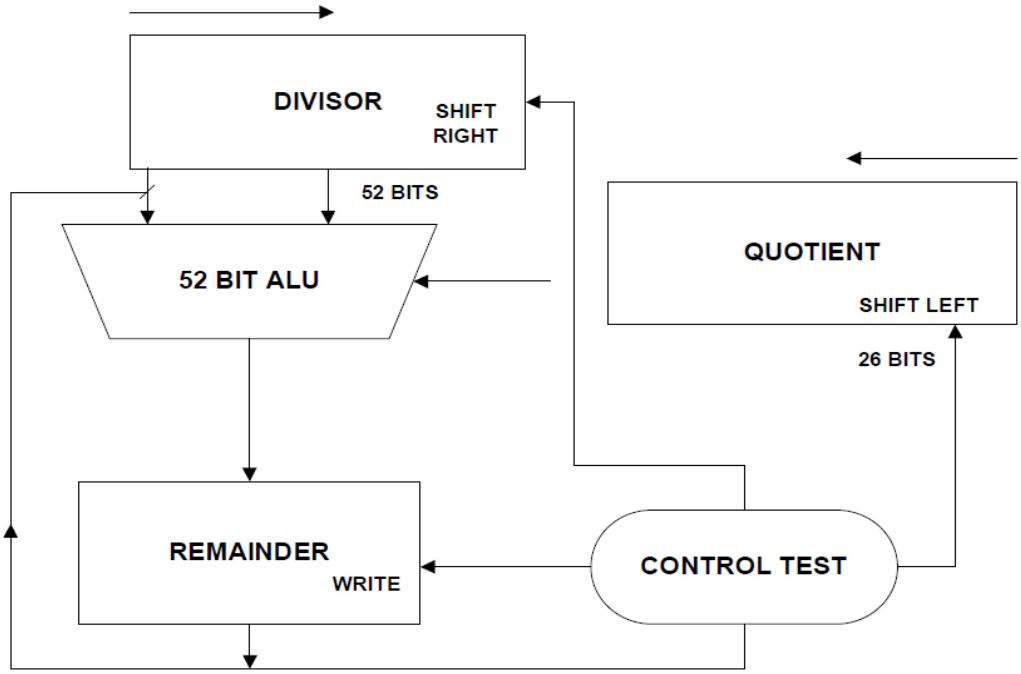

Figure 4: Double Precision Significand Division Unit.

Results and Discussions

The proposed architecture is implemented in Verilog HDL language and synthesized using Cadence RTL Complier where the design targeted for 180nm TSMC technology with proper design constraints. Figure 5 shows the simulation waveforms for DMFCP of coprocessor including relevant operations and protocol signals.

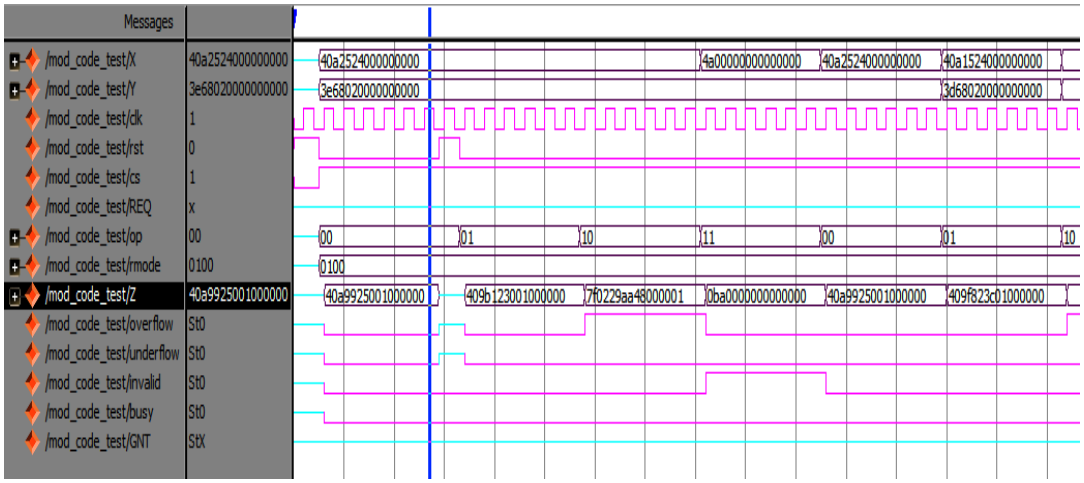

Figure 5: Simulation Results For DMFCP Coprocessor

Table 3: Synthesis Results of Coprocessor Design

Parameter	**Values**
Leakage Power	3200 nW
Dynamic Power	135.227 mW
Total Power	136.101 mW
Area	731429 nm^2
Number of Cells	28609
Timing Slack	12 ps

The synthesis results of the proposed design with TSMC 180 nm technology are compared with that of single precision coprocessor in [1] and are shown in Table 3. The design in [1] supports single precision operations but our proposed design supports both single and double precision arithmetic. The results show that our proposed coprocessor consumes less power.

Physical design is achieved using cadence SOC Encounter. Pipelining is included in the design in order to improve the performance. DFT constraints are also added in to the physical design in order to test the design.

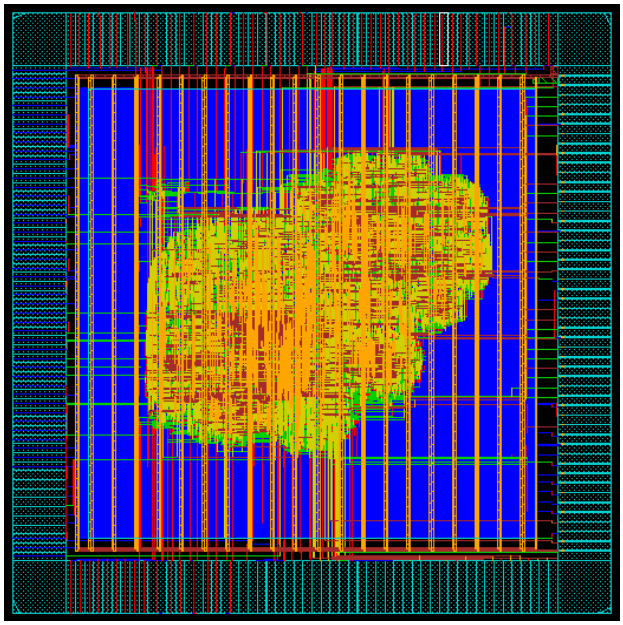

Figure 6: Physical Design View For Coprocessor

The Proposed Architecture provides around 35 % reduced hardware with reference to the previous architectures for double precision operation as structures are two parallel. It also provides 40-60 % less area-delay product compared to existing designs so that high throughput and performance is achieved. As far as our survey is concerned, all the existing designs are in FPGA platform and our design in 180nm TSMC ASIC platform is unique in its kind . So we cannot compare the results directly and hence we implemented the designs proposed in [1] and [6] in 180nm TSMC technology and found that our design provides better area delay product.

Conclusion

The Design and analysis is performed for Double precision floating point based coprocessor. Coprocessor architecture and its interface is simulated and synthesized in TSMC 180 nm CMOS library. We observed that it provides excellent throughput and lesser area as compared to relevant architectures. Henceforth, it is widely used in complicated signal processing and data encryption applications.

References

[1] Manisha Sangwan, A Anita Angeline, “Design and Implementation of Single Precision Pipelined Floating Point Co-Processor”, International Conference on Advanced Electronic Systems (ICAES), 2013.

[2] Manish Kumar Jaiswal, Ray C.C. Cheung, M. Balakrishnan and Kolin Paul, "Unified Architecture for Double / Two-Parallel Single Precision Floating Point Adder", IEEE Transactions, May 2014.

[3] IEEE 754-2008, IEEE Standard for Floating-Point Arithmetic, 2008.

[4] Suresh Srinivasan, Ketan Bhudiya, "Split-path Fused Floating Point Multiply Accumulate (FPMAC)", IEEE 21st Symposium on Computer Arithmetic, 2013.

[5] R Dhanabal, Ushasree G "VLSI Implementation of a High Speed Single Precision Floating Point Unit Using Verilog", Proceedings of 2013 IEEE Conference on Information and Communication Technologies (ICT 2013).

[6] Addanki Purna Ramesh, Ch. Pradeep, "FPGA Based Implementation of Double Precision Floating Point Adder/Subtractor Using VERILOG", International Journal of Emerging Technology and Advanced Engineering, Volume 2, Issue 7, July 2012.

[7] Semih Aslan, Erdal Oruklu and Jafar Saniie, "A High Level Synthesis and Verification Tool for Fixed to Floating Point Conversion", 55th IEEE Internation Midwest Symposium on Circuits and Systems (MWSCAS 2012).

[8] E. E. Swartzlander, Jr. and H. H. Saleh, "Fused floating-point arithmetic for DSP," in Proc. 42nd Asilomar Conf. Signals, Syst., Comput., 2008.

[9] Jagadeshwar Rao M, Sanjay Dubey, "A High Speed and Area Efficient Booth Recoded Wallace Tree Multiplier for fast Arithmetic Circuits", Asia Pacific Conference on Postgraduate Researchin Microelectronics & Electronics (PRIMEASIA), 2012.

[10] A. Baluni, F. Merchant, S. K. Nandy, and S. Balakrishnan, "A fully pipelined modular multiple precision floating point multiplier with vector support," in Electronic System Design (ISED), International Symposium on, 2011.

[11] Muhammad K & Roy K, "Reduced computational redundancy implementation of DSP algorithms using computation sharing vector scaling" IEEE Trans. Very Large Scale Integration (VLSI) Systems, vol.10, no.3, pp.292-300, June 2002.

[12] X. Wang and M. Leeser, "Vfloat: A variable precision fixed- and floating-point library for reconfigurable hardware," ACM Trans. Reconfigurable Technol. Syst., vol. 3, no. 3, pp. Sep. 2010.

[13] K. S. Hemmert and K. D. Underwood, "Fast, efficient floating-point adders and multipliers for fpgas," ACM Trans. Reconfigurable Technol. Syst., vol. 3, Sep. 2010.