DESIGN AN AREA AND SPEED OPTIMIZED FPU FOR DSP

Ankita Dixit[#1], Atuliika Shukla[*2], Ravi Mohan Sairam[#3]

[#]*Student, Department of Electronics and Communication, Shri Ram Institute of Technology, Rajeev Gandhi Technical University, Jabalpur, India*
[*]*Asst. Professor, Department of Electronics and Communication, Shri Ram Institute of Technology ,Rajeev Gandhi Technical University, Jabalpur, India*
[#]*Professor (PG Course Incharge), Department of Electronics and Communication, Shri Ram Institute of Technology, Rajeev Gandhi Technical University, Jabalpur, India*
[1]*anku305@gmail.com* , [2]*atulikashukla12@gmail.com* , [3] *ravimohan7677@yahoo.co.in*

Abstract: **FPU is a key element for real time computation of signals and real time data to meet real time scenario of signal processing it is highly required to make computation faster as possible so proposed work is a fast FPU and the chip area is another requirement to design a compact module and less power. To achieve our set goals literature survey done including standard journals like IEEE, IJEAT & Hewlett Packard etc. A Hybrid FPGA has defined course grain modules which use wide data paths, such as, for instance 32 bits wide resources. Selection FPGA affects our design in respect of area and speed so we have chosen a hybrid FPGA for faster and area optimized computation. A FPU module has three sub-modules FM, FA & WB, optimizing these we can optimized overall design we have gone through multiple approaches for Floating multiplication and floating addition we plan to use Mitchell algorithm for multiplication and we will use course grain of Vertex -4 for all logical operations. We have proposal to modify Mitchell for designing FM.**

Keywords: **HDL- Hardware Descriptive Language, ISE- Integrated Synthesis Environment, MA-Mitchell Algorithm, RTL-Register Transfer Level, UCF-User Constrain File.**

I-INTRODUCTION

In this proposed work an optimized design of a 16 bit iterative logarithmic multiplier through efficient coding is done. The design uses an iterative method to implement the logarithmic multiplier so as to increase the speed of multiplication, and reduce the number of logic blocks used to design it.

A floating-point unit (FPU) is a part of a computer system specially designed to carry out operations on floating numbers. Some systems (microcode-based architectures) can also perform various transcendental functions such as exponential or trigonometric calculations, though in most modern processors these are done with software library routines.

As in conventional FPGAs this field programmable wiring consists of a combination of metal wiring tracks and field programmable switches, the most common implementation for such field-programmable switches being a transistor switch controlled by a RAM cell. In general, a Hybrid-FPGA can be thought of as an application specific FPGA where there is some degree of field programmability and alterability. A hybrid FPGA consists of a combination of coarse-grained and fine-grained reconfigurable elements. It provides a high-throughput and cost effective platform for designers to develop applications.

II-OBJECTIVE

To design and implementation of Area and speed optimized course grain Floating Point Unit module on Hybrid FPGA and optimized the area and speed of our design module as compare to existing design modules as of literature survey.

III-LANGUAGE & TOOLS USED

We used Xilinx ISE 13.2 - Web-PACK for our programming. We considered VHDL as our primary language and the target chip considered for HDL implementation is Xilinx Vertex-Pro XC2VP30 FPGA.

IV-PROBLEM FORMULATION

Due to the complexity of floating point number systems, hardware implementations of floating point operations are typically slower than integer operations. Many modern computer programs, such as scientific computation, 3D graphics applications, digital signal processing, and system performance benchmarks have a high frequency of floating point operations. The performance of these applications is often limited by the speed of the floating point hardware. For these reasons, high performances FPUs are now both practical and desirable.

V-METHODOLOGY

As we have proposed to design a FPU unit for that we have to aware about the floating numbers and must keep in mind that computation for floating number is different the computation for non floating numbers, as our FPU design is for 16 bit numbers so we have taken the upper 10 bit out of

16 bit number as integer and remains lower 6 bit of 16 bit number as float. An example is below:-

13.25 => 0000001101.010000
13 . 25

35.6875 => 0000100101.101100
35 . 6875

Proposed FPU has arithmetic and logical operations, arithmetic addition, subtraction, multiplication, & division. Logical operation includes logical AND, OR, XOR, XNOR, NAND, AND, NOR & NOT. There is three blocks need to be design in FPU module as shown in figure below , FA/FS, WB , FM & FD.
FA/ FS => floating addition and subtraction
FM => floating multiplication
FD=> floating division

Module FA/ FS is design using coregent of Hybrid FPGA which is a coarse grain. FM design using our area optimized Mitchell algorithm FD design using our design & proposed technique.

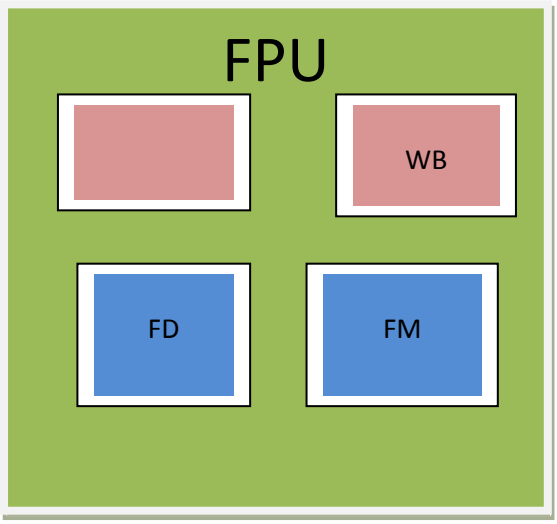

Figure 1: Proposed FPU top architecture

FA/FS: FA is floating addition and FS is floating subtraction, we just need to select a hybrid FPGA and as we have discussed hybrid FPGA has coarse boundary which includes defined coregent of addition and subtraction. We have used them in our programming. FM is Floating multiplication and we have used Mitchell algorithm with necessary modification done by us in it.

VI-EXPECTED RESULTS

With the proposed design it is been expected that the overall area the number of logic elements will get reduce with significant amount. We are expecting area reduction up to 5% than the available work and the speed optimization up to 10% with using fast logarithmic multiplication.

VII. CONCLUSION

Proposed work estimated the number of slices and 4 input LUTs is in FPU decreased by using Mitchell Multiplication Algorithm and Addition Tree. I have come to a conclusion that our design for Floating Point Unit in Hybrid FPGA in which we are using logarithmic Multiplier for multiplication purpose will requires lesser amount of area (gates) & delay (ns) as compared to the reference base papers. These techniques will be able to decrease the area up to a great extent by increasing the Power.

REFERENCES

[1] Sandesh S. Saokar, R. M. Banakar, Saroja Siddamal High Speed Signed Multiplier for Digital Signal Processing Applications, IEEE, 2012

[2] Paladugu Srinivas Teja, design of radix-8 booth multiplier using koggestone adder for high speed arithmetic applications, Emerging Trends in Electrical, Electronics & Instrumentation Engineering: An international Journal (EEIEJ), Vol. 1, No. 1, February 2014

[3] Supriya Sarkar, Sanghita Deb, G .Dilip, A New Architecture for Signed and Unsigned Multiplier by Using Radix-4 Process, International Journal of Emerging Science and Engineering (IJESE) ISSN: 2319–6378, Volume-2, Issue-1, November 2013

[4] D. Naresh, Dr. Giri Babu Kande High Speed Signed multiplier for Digital Signal Processing Applications, IOSR Journal of Electrical and Electronics Engineering (IOSR-JEEE) e-ISSN: 2278-1676, p-ISSN: 2320–3331, Volume 8, Issue 2 (Nov.-Dec.), PP 57-61

[5] Manish Chaudhary, Mandeep Singh Narula, High Speed Modified Booth's Multiplier for Signed and Unsigned Numbers, International Journal of Science and Engineering Investigations vol. 2, issue 12, January 2013

[6] Carlos Eduardo Minchola Guardia, implementation of a fully pipelined bcd multiplier in fpga, 2012 ieee

[7] E. da Costa, J. Monteiro, S. Bampi, A new array architecture for signed multiplication using Gray encoded radix-2m operands, INTEGRATION, the VLSI journal 40 (2007) 118–132, 2006 Elsevier B.V. All rights reserved

[8] D.Srinu1, S.Rambabu2, G.Leenendra Chowdary, Implementation of High Speed Signed Multiplier Using Compressor, International Journal of Advanced Research in Electrical Electronics and Instrumentation Engineering, Vol. 3, Issue 3, March 2014

