Vol 07, Article 11642; Jan-Feb 2016
http://ijves.com
International Journal of VLSI and Embedded Systems-IJVES
ISSN: 2249 – 6556

FLOATING POINT ARCHITECTURE BASED ON FPGA

M.THAMARAI SELVAN[1], N.PASUPATHY[2], K.ASHOK KUMAR[3]
[1]Research scholar, Department of Electronics S.N.R Sons College, Coimbatore, India,
[2]Associate professor, Department of Electronics, Erode Arts and Science College, Erode, India,
[3]Assistant Professor, Department of Electronics S.N.R Sons College, Coimbatore, India,
prof.thamaraiselvanelect@gmail.com , npathy@gmail.com, krjashokcbe@gmail.com

ABSTRACT

This paper presents the floating point architecture which is specifically optimized floating point application. It can be used for implementing control logic and bit oriented operations, with a help of fine grained units. The lookup tables and floating point units are implemented as data path via coarse grained units. In order to compare with existing FPGA devices, it is proposed to use Xilinx-ISE (xc3s400-5fg456) using floating point architecture. Currently, many FPGA based floating point unit's acts as only single precision floating point operation .so application is lacking to analysis area and power consumption. The fine grained units and coarse grained units can be customized to specific applications. For an example, the Existing device for speeding up of floating point application is reduced only in average area, size, position and delay. Improvement of speed is low to design flow offered by FPGA and CAD vendors which are to be performed to estimate static timing analysis. In our proposed system we achieve an improvement of speed and reduction of area and floating point benchmark with Xilinx FPGA device.

Keywords – Architecture, field-programmable gate array (FPGA), Floating Point, Application domains and programming modeling style.

I. INTRODUCTION

Floating point applications are used in implementing in FPGAs. The floating point arithmetic operations which are to be performed to analysis the area, power. In recent technology advance FPGAs used with consideration of area are becoming less and reduced the power consumption of floating point arithmetic. It can more efficient than fixed point of arithmetic to identify the large bit width. In this paper, we present a new area of power performance analysis in the method of Floating point multiplier and also we computing a structured based gate array ASIC and floating point arithmetic method. We are implementing fine and coarse grained units in the FPGA. Both units have specific applications which are used to generate data paths the coarse and fine grained units to specific the control and bit oriented operation. This method is used for evaluating the performance of floating point architecture.

II.FLOATING POINT ARCHITECTURE

The floating point architecture is used for logic block in the transistor or in simple microprocessor.Figure1 shows the basic FPGA Floating point architecture. We can use sequential and combinational logic function.

The logic block includes basic gates, multiplexers, buffers, and look up tables (LUT) and also has typical routing problems. The routing segments are used with wires in varying lengths which can be interconnected in programmable I/O. More number of wires can be used in small fraction of logic block and thereby eliminating poor density and unwanted usage of wires segments and more area.

There are three ways that can be used to implement programmable switching technique STATIC RAM, ANTIFUSE, EPROM.

Most of the FPGA are based on STATIC RAM configuration cells so we can use to configure many times.

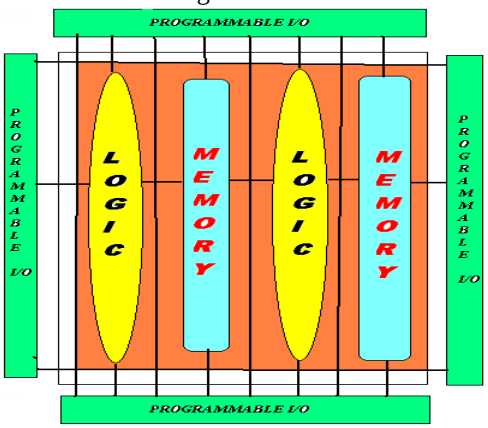

Fig1: FPGA BASIC ARCHITECTURE

Vol 07, Article 11642; Jan-Feb 2016
http://ijves.com
International Journal of VLSI and Embedded Systems-IJVES
ISSN: 2249 – 6556

The main advantage is to quickly implement and test with the standard protocols.
The ANTIFUSE technique can be used in the transistor fabrication and to control the programming current and average resistance. The main advantages are configuration data which remains when the system is powered down and there is no need of any external memory chip to store the configuration data.
The EPROM switching technique is used to turn off floating gate by charging the transistor.
The types of programming modeling are dataflow modeling, structural modeling and behavioral modeling. These types of programming modeling are used, in the various blocks. We analyze the architecture of commercial available FPGA's and depend on density and performance area.

III. IMPLEMENTATION

Figure 2 depicts the top level module that is been implemented in the floating point architecture

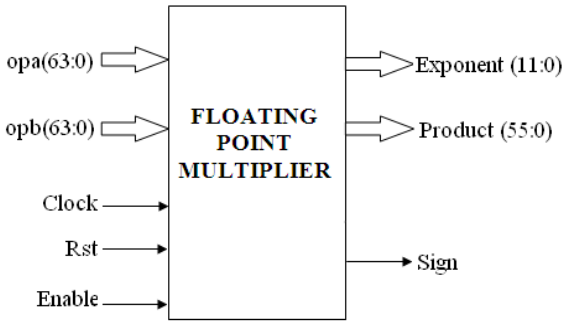

Fig 2: Top Level Module

Figure 3 floating point architecture, we are implemented in the block of input and output buses, control and status bit with a select of multiplexer unit which can be used to perform the floating point multiplier. The addition and subtraction units are used to send into the output of multiplexer. The feedback register are used to control the number of bits to the input and output buses.

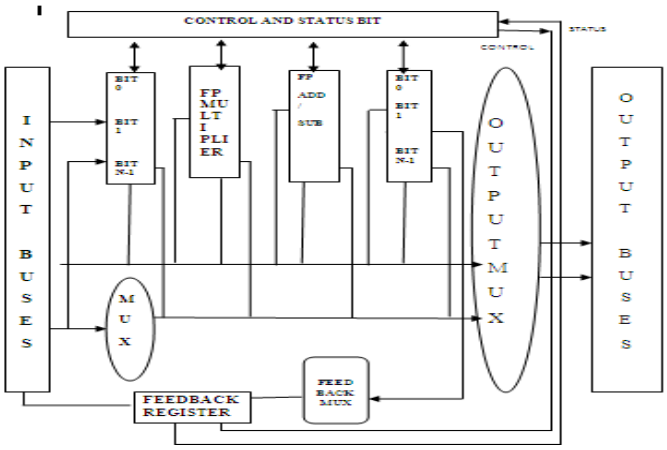

Fig3: FLOATING POINT ARCHITECTURE

IV.RESULT

The floating point multiplier is used to select the target device xc3s400-5fg456 and it is implemented in the floating point architecture. Control and status bit are used to control the register and feedback units. Finally the Input and output buses are placed and routed successfully. Table 1 shows the Utilization summary and Table 2 shows the performance summary of clock report.

Logic Utilization	Used	Available	Utilization
Number of Slice Flip Flops	1,494	7,168	20%
Number of 4 input LUTs	2,397	7,168	33%
Logic Distribution			
Number of occupied Slices	1,604	3,584	44%
Number of Slices containing	1,604	1,604	100%

Vol 07, Article 11642; Jan-Feb 2016
http://ijves.com
International Journal of VLSI and Embedded Systems-IJVES
ISSN: 2249 – 6556

only related logic			
Number of Slices containing unrelated logic	0	1,604	0%
Total Number 4 input LUTs	2,510	7,168	35%
Number used as logic	2,397		
Number of bonded IOBs	200	264	75%
IOB Flip Flops	193		
Number of MULT18X18s	12	16	75%
Number of GCLKs	1	8	12%
Total equivalent gate count for design	80,388		
Additional JTAG gate count for IOBs	9,600		

Table 1: Device Utilization Summary

Clock Net	**Resource**	**Locked**	**Fanout**	**Net Skew(ns)**	**Max Delay(ns)**
clk_BUFGP	BUFGMUX5	No	1086	0.054	0.937

Final Timing Score:	0	**Pinout Data:**	Pinout Report
Routing Results:	All Signals Completely Routed	**Clock Data:**	Clock Report
Timing Constraints:	All Constraints Met		

Table 2 : Performance Summary of clock report

FPGAs provide effective solutions for many coarse grained applications, such as digital signal processing, encryption, scientific data processing, and others. However, commercial FPGAs are fine-grained, and lacks many optimization opportunities. The use of the floating point multiplier reduces the number of adders and number of slices occupied in logic blocks and reduced delay. By selecting proper modules of reduced latency, number of slices and skew delay, significant improvements in speed and area is achieved. Floating point unit is implemented and the floating point multiplier is completely routed.

CONCLUSION

We propose the FPGA of Xilinx xc3s400-5fg456 units to optimize the floating point computation which allows using different configuration of the floating point architecture. We show that the proposed floating point architecture improves the logic utilization and speed for Xilinx FPGA for a variety of application. The Future work includes in developing automated design tools for supporting facilities such as partitioning for partitioning units, and exploring further architectural customizations for a large number of domain-specific applications.

REFERENCES

[1] M. J. Beauchamp, S. Hauck, K. D. Underwood, and K. S. Hemmert, "Architectural Modifications to Enhance the Floating-Point Performance of FPGAs," *IEEE Trans. VLSI Syst.*, 2008.
[2] A. Yan and S. Wilton. Product-term based synthesizable embedded programmable logic cores. *IEEE Trans. on VLSI*, 2006.
[3] C. Ho, P. Leong, W. Luk, S. Wilton, and S. Lopez-Buedo, "Virtual Embedded Blocks: A Methodology for Evaluating Embedded Elements in FPGAs," in *Proc. FCCM*, 2006.
[4] E. Ahmed and J. Rose, "The Effect of LUT and Cluster Size on Deep- Submicron FPGA Performance and Density," *IEEE Trans. VLSI*, 2004.
[5] K. Compton and S. Hauck, "Totem: Custom Reconfigurable Array Generation," in *Proc. FCCM*, 2001.