A Novel Time-Area-Power Efficient Single Precision Floating Point Multiplier

Himanshu Thapliyal
Centre for VLSI and Embedded System Technologies
International Institute of Information Technology,
Hyderabad, 500019, India
(thapliyalhimanshu@yahoo.com)

M.B Srinivas
Centre for VLSI and Embedded System Technologies
International Institute of Information Technology,
Hyderabad, 500019, India
(srinivas@iiit.net)

Abstract— **In this paper, a single precision IEEE 754 floating-point multiplier with high speed and low power is presented. The bottleneck of any single precision floating-point multiplier design is the 24x24 bit integer multiplier. Urdhava Triyakbhyam algorithm of ancient Indian Vedic Mathematics is utilized to improve its efficiency. In the proposed architecture, the 24x24 bit multiplication operation is fragmented to four parallel 12x12 bit multiplication modules. The 12x12 multiplication modules are implemented using small 4x4 bit multipliers. In the unsigned 24x24 bit multiplier architecture, four redundant 4x4 multiplier are provided to enforce the feature of self repairability (to recover from the faults in each 12x12 multiply modules). Reconfigurability at run time is provided for attaining power saving. The multiplier has been designed, optimized and implemented on an FPGA based system. Thus, a highly regular, self-repairable floating point parallel multiplier architecture (which can be directly scaled for larger multiplication) is proposed.**

I. INTRODUCTION

Image and digital signal processing applications require high floating point calculations throughput. Floating point multiplication is one of the performance bottlenecks in high speed and low power image and digital signal processing applications. The single precision floating point multiplication algorithm is divided into three main parts corresponding to the three parts of the single precision format. The bottleneck of any single precision floating-point design is the 24x24 bit integer multiplier. Conventional 24x24 multiply architectures are implemented in floating point multipliers using array multipliers, redundant binary architectures(Pipeline Stages), modified booth encoding, a binary tree of 4:2 Compressors (wallace tree) and modified carry save array in conjunction with Booth's algorithm[1,2,3,4,5]. There are number of problems associated with tree and array multipliers. Tree multipliers have following problems

- shortest logic delay but irregular layouts with complicated interconnects.
- irregular layouts not only demand more physical design effort, but also introduce significant interconnect delay and make noise a problem due to several types of wiring capacitance.
- the delay of the interconnection is most significant and is not suitable for VLSI implementation.
- significant amount of power consumption as reconfigurability at run time is not provided according to the input data width.

Similarly, array multipliers has also the following drawbacks associated with them

- array multipliers have larger delay and offer regular layout with simpler interconnects.
- interconnects become important in deep submicron design, structures with regular layout and simple interconnects are preferable
- significant amount of power consumption as reconfigurability at run time is not provided according to the input bit width.

In order to remove the aforesaid problem, Urdhava Triyakbhyam algorithm of ancient Indian Vedic Mathematics is utilized. The 24x24 bit mantissa multiplication operation is fragmented to four parallel 12x12 bit multiplication modules. The proposed architecture brings out the idea of reconfiguarability and self repairability at runtime. Four redundant 4x4 multiplier are also provided to enforce the feature of self repairability. Details of the architectures of the proposed floating multiplier are discussed in the next section.

II. MULTIPLIER ARCHITECTURE

The single precision floating point algorithm is divided into three main parts corresponding to the three parts of the single precision format. The first part of the product which is the sign is determined by an exclusive OR function of the

two input signs. The exponent of the product which is the second part is calculated by adding the two input exponents. The third part which is the significand of the product is determined by multiplying the two input significands each with a "1" concatenated to it. Figure 1 shows the architecture of the single precision floating point multiplier.

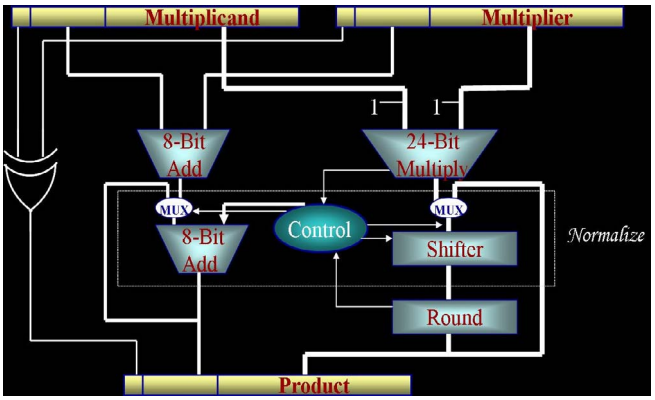

Figure 1. Single Precision Floationg Point Multiplication Architecture

In the proposed, single precision floating point multiplier architecture, it has been tried to provide the feature of reconfigurabilty and self repairability at run time, without compromising on speed improvement. In the proposed architecture, the 24x24 bit mantissa's multiplication operation is fragmented to four parallel 12x12 bit multiplication modules. The 12x12 multiplication modules are implemented using small 4x4 bit multipliers. The whole 24x24 bit multiplication operation is divided into 36 4x4 multiply modules working in parallel. Figure 2 shows the proposed 24x24 bit multiply architecture. In the 24x24 bit multiply architecture reconfigurability at run time is provided with the output of checker working as control signal. If any of (A or B)'s mantissa is of 12 bits only then the Checker will check this and will switch of the multiply blocks which are not required using the control signal. Thus significant power saving can be obtained at the run time. The reconfigurablity has also been extended to individual 12x12 multiply modules as shown in Figure 3.

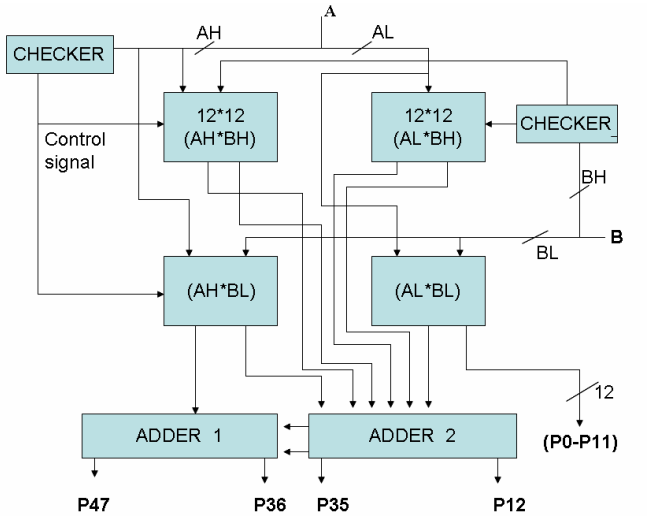

Figure 2. Proposed 24x24 bit Architecture

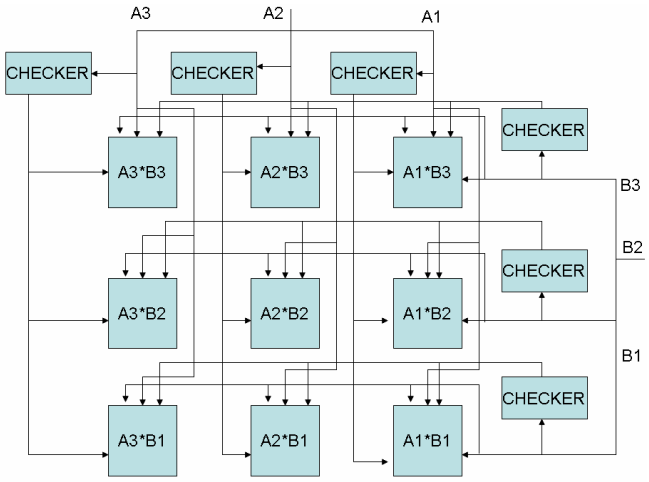

Figure 3. Internal structure of Individual 12x12 multiply module

As shown in Figure 3 , the 12 bit numbers A & B to be multiplied are divided into 4 bits groups A3,A2,A1 and B3,B2,B1 respectively. Checkers at A3,A2 and B3,B2 will check whether the mantissas to be multiplied are of 12 bits, 8 bits or 4 bits then accordingly will switch on, or switch off, the required 4x4 multiply modules. Thus there is a significant reduction in power consumption if the mantissas to be multiplied are less than 12 bits. Self reparability at run time is also provided by providing a redundant 4x4 multiply module to each 12x12 multiply module as shown in Figure 4.

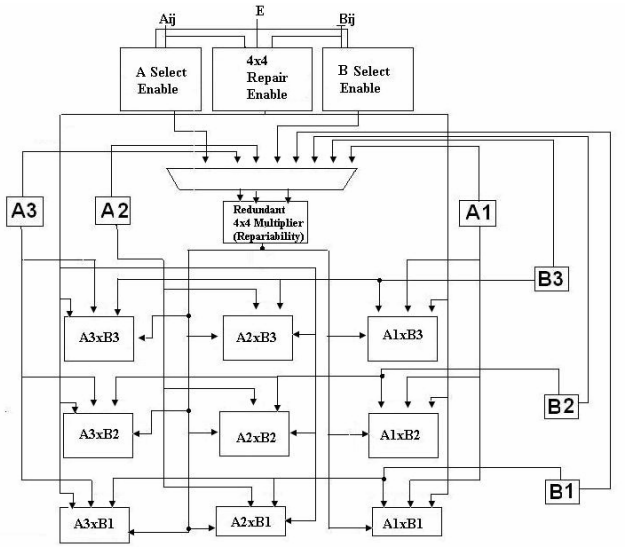

Figure 4. Proposed Feature of Self Repairability

As shown in Figure 4, redundant 4x4 Multiplier is provided to each 12x12 multiply module to provide feature of self repairability. The product of the redundant multiplier is distributed to all 4x4 multiplier. The 4x4 multiplier to be repaired is specified by the given Aij, Bij and E bits. It abandons its own output and replaces it by the one from the redundant multiplier.

III. VERIFICATION AND IMPLEMENTATION

In this study, the algorithm is implemented in Verilog HDL and logic simulation is done in Veriwell Simulator; the synthesis and FPGA implementation is done using Xilinx Webpack 6.1. After gate-level synthesis from high level behavioral and/or structural RTL HDL codes, basic schematics are optimized. The design is optimized for speed and area using Xilinx , Device Family : VirtexE, Device : XCV300e, Package: bg432, Speed grade: -8. The device is made up of multiplexers and LUTs.

IV. RESULT AND DISCUSSION

FPGA synthesis results have shown that the proposed feature of reconfigurability at run time and the control circuitry used, for the introduction of this feature, will marginally increase the delay and area of the multiplier but providing a number of significant advantages. It has been found that for Xilinx, VirtexE family, the delay of the proposed single precision multiplier is 41.203 with the area(cell usage) of 3149 compared to its implementation without the feature of reconfigurability at run time having delay of 41.203 ns with the area(cell usage) of 2967. The results are shown in Table 1. Thus the results show that there is a increment of 9.71% in delay and 6.13% in area with the introduction of feature of reconfigurability at run time which can be considered negligible with the advantages associated with it.

TABLE I. SYNTHESIS RESULTS OF THE PROPOSED FLOATING POINT MULTIPLIER ARCHITECTURE

Name of Multiplier	Vendor	Device Family & Device	Package	Speed Grade	Cell Use	Estimatd Delay (ns)
Proposed Multiplier Without Reconfigurability	Xilinx	VirtexE Xcv300e	Bg432	-8	2967	37.553
Proposed Multiplier With Reconfigurability	Xilinx	VirtexE Xcv300e	Bg432	-8	3149	41.203

V. CONCLUSION

The results obtained are quite encouraging. There is not much increase in area and the delay of the floating point multiplier with the proposed logic. Significant power saving is now possible in multiplier with the introduction of feature of reconfigurability at run time. Self repairability in the multiplier will allow it to recover from logic faults (stuck-at faults) caused by any of 36 4x4 multipliers. The proposed architecture can be extended for higher precision. Work on novel exhaustive DFT technique for proposed multiplier is in progress.

REFERENCES

[1] S. Cui, N. Burgess, M.J. Liebelt and K. Eshraghian , “A GaAs IEEE Floating Point Standard Single Precision Multiplier", Proceedings of the 12th IEEE Symposium on Computer Arithmetic, pp 91-97, Bath, UK, July 19-21 1995.

[2] R. K. Yu and G. B. Zyner, 167 mhz radix-4 floating point multiplier, in Proceedings of the 12th Symposium on Computer Arithmetic (S. Knowles and W. H. McAllister, eds.), (Bath, England), pp. 149-154, 1995.

[3] Mark D. Aagaard and Carl-Johan H. Seger,"The Formal Verification of a Pipelined Double-Precision IEEE Floating-Point Multiplier",Proceedings of the 1995 IEEE/ACM international conference on Computer-aided design,pp. 7 - 10 ,San Jose, California, United States.

[4] Ahmet Akkas, Michael J. Schulte, "A Quadruple Precision and Dual Double Precision Floating-Point Multiplier",. proceedings DSD 2003,pp.76-81,3-5 September 2003, Belek-Antalya, Turkey.

[5] GH. A. Aty, Aziza 1. Hussein, I. S. Ashour and M. Mona,"High-speed, Area-Efficient FPGA-Based -Floating-point Multiplier", Proceedings ICM 2003, pp-274-277,Dec. 9-11 2003, Cairo, Egypt.