A Double Precision Floating Point Multiplier Suitably Designed for FPGAs and ASICs

Himanshu Thapliyal[1], Vishal Verma[2] and Hamid R. Arabnia[3]
[1]CVEST, IIIT Hyderabad, India
[1]Delhi College of Engineering, India
[2]The University of Georgia, Department of Computer Science, USA

***Abstract*— In this paper, a double precision IEEE 754 floating-point multiplier with high speed and low power is presented. The bottleneck of any double precision floating-point multiplier design is the 53x53 multiplication of the mantissas (52 bit mantissa+1 hidden bit). This paper proposes a approach to improve this performance bottleneck by adding a redundant 54th bit initialized to '0' in the mantissas of both multiplicand and multiplier. Now, the 54x54 bit multiplication operation is decomposed and performed by nine parallel 18x18 bit multiplication modules. The 18 bit decomposition is chosen, since nearly 512 dedicated 18x18 bit multipliers are available in the latest FPGAs. Moreover, it will be also helpful in implementing the floating point multiplier as an ASIC since the multiply operation can now be performed easily with small dedicated 18x18 bit multipliers. In the proposed architecture, one redundant 18x18 multiplier is also provided to enforce the feature of self repairability (to recover from the faults in 18x18 bit multiply modules). Reconfigurability at run time is provided for attaining power saving. The multiplier has been designed, optimized and implemented on the Xilinx ViterxE FPGA. Thus, a highly regular, self-repairable floating point parallel multiplier architecture (which can be directly scaled for larger multiplication) is proposed.**

I. INTRODUCTION

High end applications like Image and Digital Signal Processing applications require high floating point calculations throughput. Floating point operations especially multiplication are hard to implement as an ASIC or on FPGAs as their algorithms are quite complex [1]. FPGAs vendors including Xilinx have introduced FPGAs with nearly 512 18x18 bit dedicated multipliers [2]. These architectures can cater the need of high speed integer operations but are not suitable for performing floating point operations especially multiplication. Floating point multiplication is one of the performance bottlenecks in high speed and low power image and digital signal processing applications [3,4,5,6]. The IEEE 754 double-precision format is 64 bits long: 1 bit for the sign, 11 bits for the exponent, and 52 bits for the fraction. Thus, it requires 53x53 bit (52bit mantissa+1hidden bit) integer multiplier to perform the double precision floating point multiplication operation. In the FPGAs, the bottleneck of any double precision floating-point design is the 53x53 bit integer multiplier, since they have only dedicated 18x18 bit multipliers. Recently, there has been significant work on analysis of high-performance floating-point arithmetic on FPGAs [7,8,9,10].

In order to remove the aforesaid problem, the author propose the addition of a redundant 54th bit initialized to '0' to the mantissa of both multiplicand and multiplicand. The 54 bit mantissa can now be multiplied easily with nine parallel 18x18 bit multipliers. The proposed architecture also brings the idea of reconfigurability and self repairability at runtime, thus providing a low power as well as fault recovering architecture. The proposed architecture is especially designed for high performance and low power floating point multiplications in FPGAs and is also beneficial for implementing the multiplier as an ASIC, since the multiply operation can now be performed easily with small dedicated 18x18 bit multipliers.

II. FLOATING POINT MULTIPLIER ARCHITECTURE

The double precision floating point algorithm is divided into three main parts corresponding to the three parts of the double precision format [5]. 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 double precision floating point multiplier. Thus, it can be easily observed from the Figure 1 that 53x53 bit integer multiplier is the main performance bottleneck for high speed and low power operations. In FPGAs, the availability of the dedicated 18x18 multipliers further complicates this problem. This is the driving force that leads to the proposal of novel architecture suitable for performing double precision floating point multiplication operations in FPGAs as well as suitable for implementing as an ASIC.

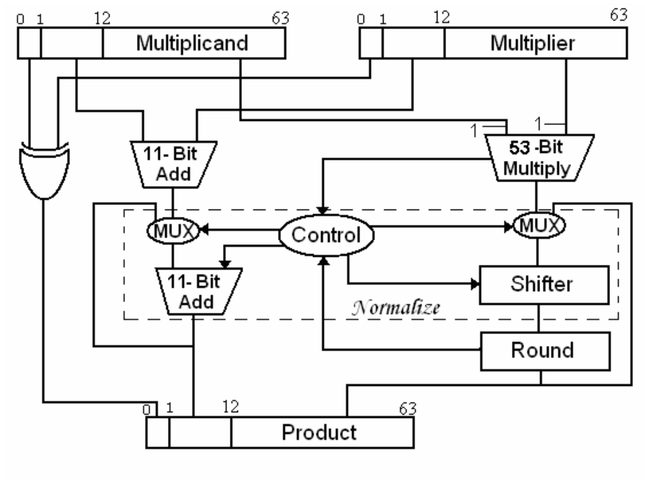

Figure 1. Double Precision Floationg Point Multiplication Architecture

III. PROPOSED DOUBLE PRECISION MULTIPLICATION ARCHITECTURE

In the proposed architecture, the authors propose that a redundant 54th bit initialized to '0' should be concatenated to the 53 bits (52 bit mantissa +1 hidden bit) to have the mantissa multiplication of 54x54 bit (the redundant zero bit would have any effect in the multiplication as it is initialized to '0'). The 54x54 bit multiplication can now be performed by decomposing the 54 bit numbers A and B(multiplicand and multiplier) to 18 bits groups A3,A2,A1 and B3,B2,B1 respectively. Thus the 54x54 bit multiplication operation can now easily be performed by using 9 18x18 bit multipliers available in FPGAs working in parallel. The features of reconfigurabilty and self repairability at run time are also proposed in the architecture for low power and self repairability from faults.

As shown in Figure 2, Checkers at A3, A2 and B3, B2 will check whether the mantissas to be multiplied are of 54 bits, 36 bits or 18 bits and then accordingly will switch on, or switch off; the required 18x18 multiply modules. Thus there is a significant reduction in power consumption if the numbers to be multiplied are less than 54 bits. Self reparability at run time is also provided by providing a redundant 18x18 multiply module to 9 parallel 18x18 bit multipliers.

As shown in Figure 3, a redundant 18x18 bit multiplier is provided to provide the feature of self repairability[11]. The product of the redundant multiplier is distributed to all 18x18 bit multipliers. The 18x18 multiplier to be repaired is specified by the given Aij, Bij and E bits. Then the 18x18 multiplier to be repaired abandons its own output and replaces it by the one from the extra multiplier. It should be noticed that the power supply of the disabled unit (one of the 9 18x18 multiplier) will be turned off through a power enable control to reduce the power dissipation. The test vectors can check whether the specified multiplier is working properly or not.

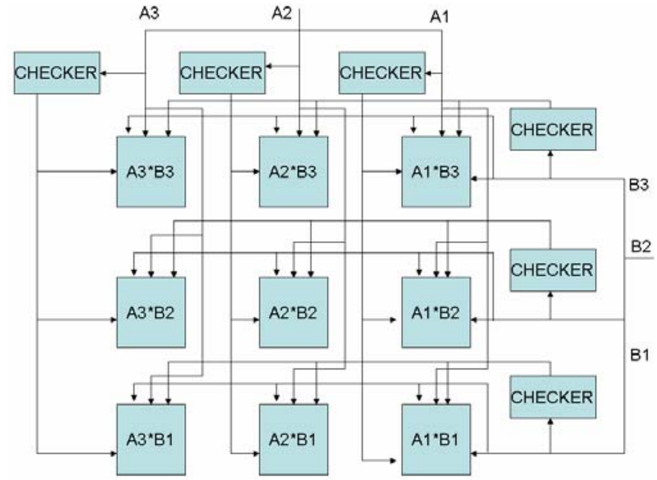

Figure 2. Proposed Multiplier architecture having 54x54 bit multiplication performed by 9 parallel 18x18 bit Multipliers

The advantage of the proposed architecture is that floating point multiplication operation can now be performed easily in FPGA without any resource and performance bottleneck since the latest FPGA has nearly 512 dedicated 18x18 bit multipliers. Thus, the proposed method of decomposition is highly beneficial, since it removes the performance bottleneck faced on implementing floating point operations in FPGAs. Moreover, if there is a fault in any of the 18x18 dedicated multiplier available in FPGA employed in the multiplication design, it can also take care of it. The proposed architecture is also suitable for implementing it as an ASIC since it is designed in highly parallel and low power manner by decomposing the operation to small 18x18 multiply modules. The proposed feature of reconfigurability at run time and self repairability also enhance the performance of the proposed architecture. The large integer multiply operations can also be performed easily by dedicated 18x18 bit multipliers reducing the need of dedicated large size multipliers.

IV. VERIFICATION AND IMPLEMENTATION

In this study, the proposed architecture 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. FPGA synthesis results have shown that the proposed feature of reconfigurability at run time and the control circuitry designed for the introduction of this feature will negligibly increase the area of the proposed floating multiplier architecture. It has been found that for Xilinx VirtexE family, the delay of the proposed floating point architecture is 41.355 with the area (cell usage) of 11095 with the feature of reconfigurability and repairability while the delay is 41.475 ns with the area (cell usage) of 11067 without these additional features. The results are shown in Table 1. Thus the results show that there is a negligible increment in area with the introduction of feature of reconfigurability at run time.

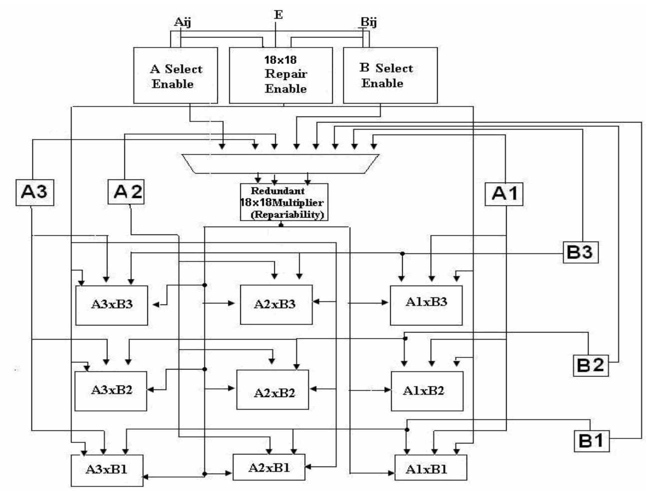

Figure 3. Proposed Feature of Self Repairability

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

Name of Multiplier	Vendor	Device Family & Device	Package	Speed Grade	Cell Use	Estimated Delay(ns)
Proposed Multiplier Without Reconfigurability	Xilinx	VirtexE Xcv300e	Bg432	-8	11067	41.475
Proposed Multiplier With Reconfigurability	Xilinx	VirtexE Xcv300e	Bg432	-8	11095	41.355

V. CONCLUSION

This paper proposes a highly regular self-repairable and reconfigurable double precision floating point multiplication architecture specially designed to make it suitable for FPGA implementation as well as an ASIC. Both the repairing and reconfigurability take the advantage of the partitioning of the circuit into 18x18 bit multiply module, which results in high controllability and observability, inherent in the decomposition approach. The results obtained are quite encouraging and there is a negligible increase in area of the proposed architecture with the proposed features of repairability and reconfigurability. Significant power saving is now possible in the multiplier with the introduction of feature of reconfigurability at run time as appropriate multiply blocks which are not required can now be switched off easily . Self repairability in the multiplier will allow it to recover from logic faults (stuck-at faults) caused by any of 9 18x18 multipliers. The most significant aspect of the proposed architecture is that it will make the FPGAs suitable for performing high speed low power floating point multiplication operations. Since, in latest FPGAs, there are nearly 512 18x18 bit dedicated multipliers available, the proposed approach will be highly beneficial for running a number of double precision floating point multiplications in parallel without creating any performance and resources bottleneck, thus making FPGAs suitable for high performance and low power DSP operations. The proposed architecture can be extended for higher precision. Work on novel exhaustive DFT technique for the proposed architecture is in progress.

REFERENCES

[1] 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.

[2] www.xilinx.com/products/silicon_ solutions/fpgas/virtex/virtex4/

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] Ronald Scrofano, Gokul Govindu, Viktor Pasanna,"A Library of Parameterizable Floating-Point Cores for FPGAs and Their Application to Scientific Computing", ERSA 2005, Las Vegas, Nevada, USA, June 27-30, 2005,pp.137-148

[8] Gokul Govindu, Viktor K. Prasanna, Vikash Daga, Sridhar Gangadharpalli, V. Sridhar, "Efficient Floating-point Based Block LU Decomposition on FPGAs", ERSA 2005, Las Vegas, Nevada, USA, June 21-24, 2004,pp.137-148

[9] Gokul Govindu, Seonil Choi, Viktor K. Prasanna, Vikash Daga, Sridhar Gangadharpalli, V. Sridhar,"A High-Performance and Energy-Efficient Architecture for Floating-Point Based LU Decomposition on FPGAs", IPDPS 2004, Santa Fe, New Mexico, USA,26-30 April 2004.

[10] Gokul Govindu, Ling Zhuo, Seonil Choi, Viktor K. Prasanna, "Analysis of High-Performance Floating-Point Arithmetic on FPGAs", IPDPS 2004,Santa Fe, New Mexico, USA,26-30 April 2004.

[11] Rong Lin and Martin Margala, " Novel Design and Verification of a 16x16-b Self repairable Reconfigurable Inner Product Processor", GLSVLSI'02, April 18-19,2002, Newyork, USA., PP 172-177.