International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

COPY RIGHT

ELSEVIER
SSRN

IJIEMR Transactions, online available on 21st Feb 2021. Link
:http://www.ijiemr.org/downloads.php?vol=Volume-10&issue=ISSUE-02

DOI: 10.48047/IJIEMR/V10/I02/18

Title DESIGN A HIGH EFFICIENT SELF REPAIRABLE FAULT TOLERANT SYSTEM
Volume 10, Issue 02, Pages: 88-93

Paper Authors
KPV GOPALAKRISHNA, A RAGHAVARAJU, N VAMSIKRISHNA, N VIJAYSHANKER

USE THIS BARCODE TO ACCESS YOUR ONLINE PAPER

To Secure Your Paper As Per UGC Guidelines We Are Providing A Electronic Bar Code

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

DESIGN A HIGH EFFICIENT SELF REPAIRABLE FAULT TOLERANT SYSTEM

[1]KPV GOPALAKRISHNA, [2]A RAGHAVARAJU, [3]N VAMSIKRISHNA, [4]N VIJAYSHANKER

[1]M.Tech Scholar, Department of ECE, Chebrolu Engineering College, Chebrolu, Guntur Dist, Andhra Pradesh Andhra Pradesh, India

[2,4]Associate professor, Department of ECE, Chebrolu Engineering College, Chebrolu, Guntur Dist, Andhra Pradesh Andhra Pradesh, India

[3]Assistant Professor, Department of ECE, Chebrolu Engineering College, Chebrolu, Guntur Dist, Andhra Pradesh Andhra Pradesh, India

ABSTRACT: Using VLSI more number of transistors can be embedded on a single chip. As the space between transistors or circuits decreasing the system or chip is more susceptible to faults. Fault tolerant systems required to avoid inaccurate results. Multiplexer is a device which selects input signals based on select signal. The existing papers deal with only self checking multiplexer. In this project, design a high efficient self repairable fault tolerant system is implemented. Basically, multipliers are key arithmetic circuits in many of these applications including digital signal processing (DSP). This fault tolerant system firstly performs the pre- processing stage. In pre processing stage bits are generated and next bits are assigned in particular format. By checking repairable logic the operation is performed. Fault tolerant system will detect any types of faults if there or else it will send it trough the output block. Hence this project reduces the errors in effective way.

KEYWORDS: VLSI, Digital signal processing (DSP), Multiplexer, Fault tolerant system, self repairable.

I. INTRODUCTION

As machine learning algorithms are getting more popular, there is an increasing demand for developing hardware accelerators for them. In particular deep neural networks such as Convolutional Neural Networks (CNNs) have multiple traits that make them very attractive for hardware acceleration, such as high structural regularity, high computational complexity, and yet wide applicability and high recognition performance.

FPGAs are one of the most preferred platforms due to their high flexibility and at the same time high parallelism [1]. Hence much effort has been made to create better CNN accelerators on FPGAs. A unique option available to hardware implementations of DNNs (Deep learning Neural Networks) is the flexibility in data width of arithmetic operations [2]. GP-GPUs, for instance, have long provided only two options-either single-precision or double-precision floating point— since integer arithmetic on modern GP-GPUs has zero or negative performance advantage [3]. Recently half-precision was introduced on some select models, but this is a one-time change and not customizable by user. By contrast an ASIC (Application-Specific Integrated Circuit) implementation can choose whatever precision sufficient for the target CNN application. As recent work suggests that 8-bit fixed-point is often enough for inference, even for deep CNNs,

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

there is a good opportunity to increase performance for free by using lower precision without affecting output quality. FPGAs, too, have the flexibility, and using reduced precision means potentially higher throughput on the same FPGA [4].

In practice, however, since most arithmetic operations are implemented using DSP blocks, and DSP blocks, too, support only a limited set of precisions, it is not easy to achieve higher performance through reduced arithmetic precision. For example, the DSP block of Xilinx FPGAs, DSP48E1, can perform a 25x18-bit multiplication only, and there is no way to perform two 8x8-bit multiplications simultaneously on the same DSP block for higher throughput [5]. This paper is about how to turn an ordinary DSP block of an off-the-shelf FPGA device into a 2-way SIMD 40 (SingleInstruction Multiple-Data) MAC (Multiply-and-Accumulate) unit, that can deliver 4 ops/cycle by performing two multiplyand-add operations simultaneously with reduced data width. Though we evaluate our technique for a Xilinx Virtex-7 FPGA only, our method is generic and applicable to other FPGAs with similar hardware DSP blocks. Our technique does not require any change in the FPGA fabric itself. Realizing SIMD add on a DSP unit is trivial and it is already supported. A SIMD multiply using LUTs (Look-Up Tables) is also trivial. The challenge is how to realize SIMD multiply on a DSP block of an FPGA. Using DSP blocks is important, since most DNN implementations on FPGAs rely on DSP blocks for MAC operations, and therefore being able to perform two MACs using one DSP block essentially means free 2X improvement in computation throughput [6].

II. LITERATURE SURVEY

The below figure (3.) shows the structure of existed system. The entire flow chart performs its operation in four stages which are discussed below

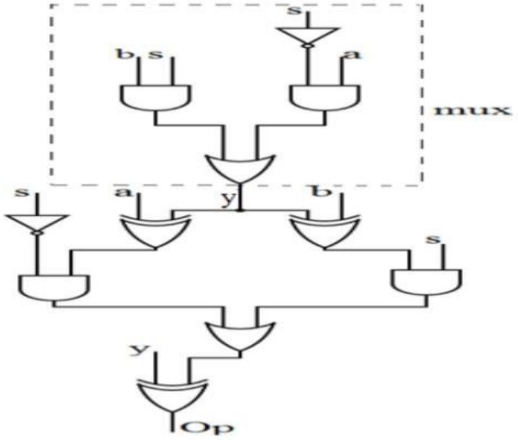

Fig. 1: EXISTED SYSTEM

The technology is scaling down, chip density is increasing so that millions of transistors are embedded on a single die. The yield may decrease due to process variations, deviation in parameters and lithographic effects. This advanced microelectronic technologies more susceptible to faults. The response of a circuit may be invalid because of presence of faults [7-9]. This leads to inaccurate results. Fault secure systems are very much needed to withstand faults. So the self checking and repairing is necessary for correct operation of the circuit. In self checking the fault is detected by circuit itself and in self repairing the circuit can repair itself and produces correct output. The overall circuit performance depends on individual gates of the circuit. Using small number of gates for design can increase the performance in terms of delay, area and power. To get high speed the critical path should be as minimum as possible. Similarly to get low power less number of gates are used at circuit level without compromising the accuracy of the circuit. Multiplexers are

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

used in wide variety of applications like adders, multipliers, communication, digital signal processing etc [10]. Based on the selection signal multiplexer will select the input data and passes it to the output. The presence of fault in a 38 multiplexer causes invalid data at the output. The multiplexer should be fault secure so that it gives valid data at the output even though faults are present in it.

Self checking multiplexer was proposed. This self checking multiplexer designed by using four transmission gates and an inverter as shown in Fig. 1. When CS is low S0N is passed to SN. Similarly when CS is high S1N is passed to SN. Thus it implements the function of multiplexer. In this self checking multiplexer when SN and SN_bar are same then it shows the presence of a fault. By using this structure only fault is detected and can't be repairable. To make the multiplexer self repairing two different structures are proposed. The CS bar signal is the inverted signal of CS. In Fig. 1 the circuit enclosed in square box shows the basic structure of 2:1 multiplexer. Remaining structure which is not included in the square box is used for repairing the above 2:1 multiplexer. The circuit is able to detect all possible single and multiple faults present in the 2:1 multiplexer and repairs the circuit. The circuit gives 100% error recovery. Consider Fig. 1. Assume there is a stuck at '0' fault at y. Since y was stuck at '0', it will give always '0' as the output. However when this value is passed to repairing circuit, it detects the fault and produces correct output. This is shown in Fig. 1. Similarly assume there is a stuck at '1' fault at y. Since y was stuck at '1', it will give always '1' as the output. However when this value is passed to repairing circuit, it detects the fault and produces correct output. This is shown in Fig. 1. So when there is fault in multiplexer block, then output Op gives the inverted value of y. If there is no fault, then y value is passed to the output Op. The above proposed multiplexer 1 uses additional circuitry to repair. In the proposed self repairing multiplexer 2 the building blocks of multiplexer itself are self repairable.

III. PROPOSED ARCHITECTURE

The figure (2) shows the block diagram of proposed system. This fault tolerant system firstly performs the pre-processing stage.

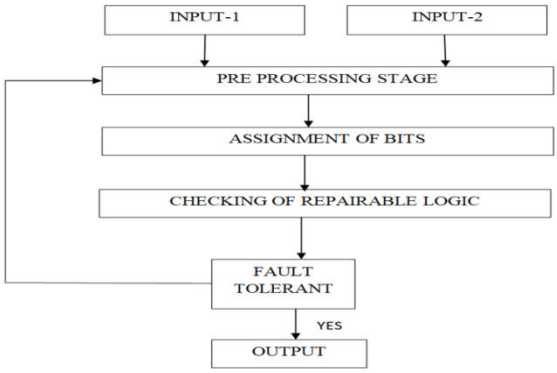

Fig. 2: BLOCK DIAGRAM OF PROPOSED SYSTEM

In pre processing stage bits are generated and next bits are assigned in particular format. By checking repairable logic the operation is performed. Fault tolerant system will detect any types of faults if there or else it will send it trough the output block. Here firstly, the operands are loaded in the multiplier. The arithmetic operations like addition and multiplication operations are performed. The obtained result of this will be saved in the barrel shifter. Here irreducible polynomial function is not used in the system. The main intent of register multiplier is to store the bit representation and give polynomial output a(t). Here parallel load operation is performed in the most significant bit position. In the same way left shift operations are performed in MSB bit. The multiplicand bit is used b(t)

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

value to store the value in register. The parallel load operation is also applied in the multiplicand. The obtained value is stored in the register. The right shift operation is performed in the multiplicand register block. crypto core processor is used to transfer the data in multiplicand register.

The barrel shifter consists of root and load mr and this are taken as input to this block. The multiplier register is generally attached to the finite field arithmetic circuit. In the same way, multiplicand register consists of shift, data_in and load_md bits which are taken as input to the barrel shifter. It will shift the data and as well as load the data in effective way. Result register consists of output and saves the entire arithmetic result. Compared to existed system, the proposed system gives effective results. The result multiplier and multiplicand is saved in the result barrel shifter block. The both a(t) and b(t) values are assigned in the barrel shifter blocks. The obtained values in the barrel shifter block will shift the bits to adder block. This block will perform the addition operation. After performing particular operation, the bits are shifted to the result register. This result register will save the output as product. At last the barrel shifter will perform the parallel operation in effective way.

Partial-Product Multiplication is an alternative method for solving multi-digit multiplication problems. This is a strategy that is based on the distributive (grouping) property of multiplication. The first partial product is created by the LSB of the multiplier, the second partial product is created by the second bit in the multiplier, etc. The final partial products are added with a accurate adder circuit. The accurate adder is decided by architecture/system-level applications. A self configuration technique has been proposed for the scenarios where architecture/systemlevel choice is either unclear or difficult. A carry is propagated through several consecutive bits because of the actual path delay is large. When the actual carry propagation chain is short, there is no need to use approximation configuration, which is intended to cut carry chain shorter.

IV. RESULTS

The figure (3) shows the RTL schematic of proposed system.

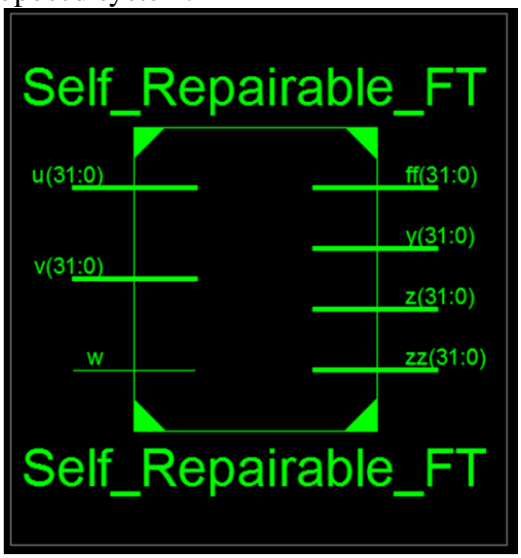

Fig. 3: RTL SCHEMATIC OF PROPOSED SYSTEM

Registertransfer logic deliberation is utilized in equipment portrayal dialects (HDLs) like Verilog and VHDL to make elevated level portrayals of a circuit, from which lower-level portrayals and at last genuine wiring can be determined. Structure at the RTL level is run of the mill practice in present day advanced plan.

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

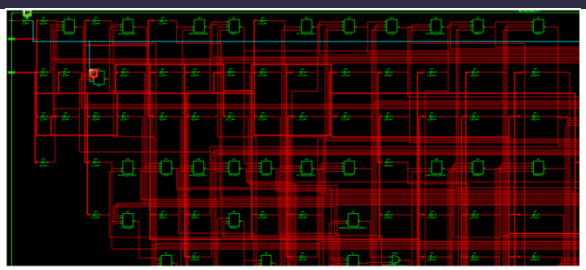

Fig. 4: TECHNOLOGY SCHEMATIC OF PROPOSED SYSTEM

The figure (4) shows the Technology schematic of proposed system. This schematic is generated after the optimization and technology targeting phase of the synthesis process. It shows a representation of the design in terms of logic elements optimized to the target Xilinx device or "technology"; for example, in terms of of LUTs, carry logic, I/O buffers, and other technology-specific components. Viewing this schematic allows you to see a technology-level representation of your HDL optimized for a specific Xilinx architecture, which might help you discover design issues early in the design process

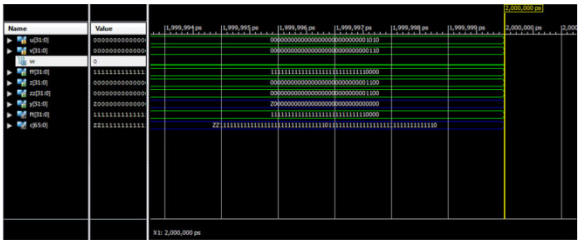

Fig. 5: OUTPUT WAVEFORM OF PROPOSED SYSTEM

The figure (5) shows the output waveform of proposed system. The entire project is implemented using 32 bits.

V. CONCLUSION

As the multiplexer is very vital component in many systems, faults in multiplexer lead to inaccurate results in the systems. Fault tolerant systems must need fault tolerant multiplexer to avoid faults. Hence, in this project, design a high efficient self repairable fault tolerant system was implemented. This fault tolerant system firstly performs the pre-processing stage. In pre processing stage bits are generated and next bits are assigned in particular format. By checking repairable logic the operation is performed. Fault tolerant system will detect any types of faults if there or else it will send it trough the output block. The proposed self repairing multiplexers can be used in fault tolerant systems to get error recoverability. The structure repairs itself so that no external inputs are required to repair. This avoids the area overhead. The circuits are simulated and verified the outputs. The fault tolerance of the proposed structures is verified. Hence this project reduces the errors in effective way.

VI. REFERENCES

[1] Zhiming Zhang, Laurent Njilla, Charles A. Kamhoua, Qiaoyan Yu, “Thwarting Security Threats From Malicious FPGA Tools With Novel FPGA-Oriented Moving Target DefensE”, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Volume: 27, Issue: 3, 2019

[2] Ameer M.S. Abdelhadi, Lesley Shannon, “Revisiting Deep Learning Parallelism: Fine-Grained Inference Engine Utilizing Online Arithmetic”, 2019 International Conference on Field-Programmable Technology (ICFPT), 2019

[3] Ramamohanarao, Qinbin Li, “Exploiting GPUs for Efficient Gradient Boosting Decision Tree Training”, IEEE Transactions on Parallel and Distributed Systems, Volume: 30, Issue: 12, 2019

[4] Dries Vercruyce, Elias Vansteenkiste, Dirk Stroobandt, “How Preserving Circuit Design Hierarchy During FPGA Packing Leads to Better Performance”, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Volume: 37, Issue: 3, 2018

[5] Javier Hormigo, Julio Villalba, "HUB Floating Point for Improving FPGA Implementations of DSP Applications", IEEE Transactions on Circuits and Systems II: Express Briefs, Volume: 64, Issue: 3, 2017

[6] Martin Langhammer, Bogdan Pasca, "Single Precision Natural Logarithm Architecture for Hard Floating-Point and DSP Enabled FPGAs", 2016 IEEE 23nd Symposium on Computer Arithmetic (ARITH), 2016

[7] G. R. Gokhale , S. R. Gokhale, "Design of area and delay efficient Vedic multiplier using Carry Select Adder", 2015 International Conference on Information Processing (ICIP)

[8] Thottempudi Pardhu , N.Alekhya Reddy, "Design of ultra low power multipliers using hybrid adders", 2015 International Conference on Communications and Signal Processing (ICCSP).

[9] Josmin Thomas , R. Pushpangadan , S Jinesh, "Comparative study of performance vedic multiplier on the basis of adders used", 2015 IEEE International WIE Conference on Electrical and Computer Engineering (WIECON-ECE).

[10] S. Narayanamoorthy, H. A. Moghaddam, Z. Liu, T. Park, and N. S. Kim, "Energy- efficient approximate multiplication for digital signal processing and classification applications," IEEE transactions on very large scale integration (VLSI) systems, vol. 23, no. 6, pp. 1180–1184, 2015.