DMR Based CAN Bus Control System Implemented Into FPGA

Amit Kumar Bhadrawat[1], Prof. Sourabh Sharma[2]
[1]M.TECH. Student, [2]Asst. Prof., (EC) Dept.
[1, 2]Trinity Institute of Technology and Research, Bhopal, M.P., India.

Abstract - Electronics components in many application required maximum level of fault tolerance and high reliability. Application like avionic, railway, deep space mission can serve as an example of these applications. In these applications, electronics components are exhibited to the environment conditions, from among them especially cosmic radiation can have an undesired and destructive effect. In this paper, the design and implementation of DMR based CAN bus control system implemented into FPGA is described. The bus control system uses CAN Aerospace application protocol .the fault tolerant features of the developed system are improved by DMR architecture. Then, experiments With SEU injection into the FPGA configuration memory with both non-DMR and DMR architectures are described, the results presented and evaluated.

Keywords: CAN, FPGA, Fault tolerance, TMR, VHDL.

I. INTRODUCTION

This paper is about fault tolerant CAN bus control system. For this work using FPGA as a platform for implementation. In this research used DMR (double modular redundancy) architecture method to reduce complexity of system. The complexity of digital systems has a significant impact on reliability and diagnostic features of these systems. FPGA-based systems are becoming increasingly popular for space based applications due to their high throughput capabilities and relatively low cost. When faults are detected in any part of the system implemented into FPGA then a possibility to reconfigure it and extend its lifetime exists. SRAM-based FPGAs are susceptible to radiation-induced Single Event Upsets. SEU causes the change in the state of a digital memory element caused by an ionizing particle. As the ionizing particle passes through the device, charge can be transferred from one node to another. This charge transfer can lower the voltage of a memory cell and change its internal state. SEU occurrence in FPGA memory can be seen as a big problem for many digital systems. Therefore, many FT techniques have been proposed and tested for mitigating SEUs in systems implemented into FPGAs

II. SYSTEM MODEL

There has always been hot discussion between what to choose, an FPGA or a conventional hard IP microcontroller. It seems that FPGAs are going to rule in the future because of their flexibility, increasingly better power efficiency and decreasing prices. Often a soft processor is added in the FPGA design to get microcontroller like functionality along-with other concurrent processing. Microcontroller is low-cost, much lower than FPGAs. This is especially true for small applications and large quantities.

FPGAs are concurrent. It can take sequential functionality like adding soft processor core. While the microcontrollers are always sequential. This makes FPGAs better suited for real time applications such as executing DSP algorithms. FPGA are flexible, it can add/subtract the functionality as required. This cannot be done in microcontroller.

With growing interest in the use of SRAM based FPGAs in space and other radiation environments, there is a greater need for efficient and effective fault tolerant design techniques specific to FPGAs. Triple modular redundancy is common fault mitigation technique for FPGAs and has been successfully demonstrated by several organizations. Although TMR has been shown to significantly improve design reliability, it carries a high overhead cost. At a minimum, full TMR of a design requires three times the hardware to implement three identical copies of a given circuit .in addition additional logic is required to implement the majority logic voters. In the worst case, TMR can require up to six times the area of the original circuit .the additional hardware resources required to triplicate the original circuit result in other secondary problems such as increased power and slower timing.

TMR is a static hardware redundancy scheme for masking single faults in a digital circuit. Fig.5. depicts the traditional method of TMR.

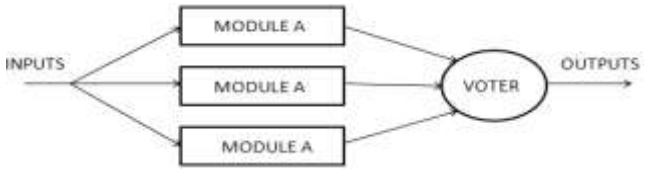

Fig.3.1. Generic flow of TMR

A failure in any one of the three circuit's copies will be masked by the majority voter output. For TMR to work properly in an FPGA there should be no more than one upset in the configuration memory at any given time. More than one upset could defeat completely the majority voters and result in a functional error. Two reasons exist why the TMR based design can be possibly attacked more often.

(1) TMR based designs need a greater area than the non-TMR design.

(2) TMR based design contain voter which is not protected in TMR architecture against faults. As soon as the voter is attacked, the design is completely corrupted and does not work.

III. PREVIOUS WORK

By Karel Szurman1, Jan Kastil, Martin Straka, Zdenek Kotasek, "Fault tolerant CAN bus control system implemented into FPGA" In this paper, the basic ideas of the design and implementation of CAN bus based control system into FPGA platform is described. The bus control system uses CAN Aerospace application protocol. The fault tolerance features of the developed system are improved by TMR architecture. Then, experiments with SEU injection into the FPGA configuration memory with both non-TMR and TMR architectures are described, the results presented and evaluated [1].

IV. PROPOSED METHODOLOGY

To solve the problem identified we plan to develop a model of DMR based CAN bus control system. We have subdivided our project into four different modules.

(i) Can_host (host)

(ii) Can_host2 (host2)

(iii) Bist_system (comparator)

(iv) Frame_tx_imp (multiplexer)

By simulating these modules we will show that data will be able to provide fault tolerant CAN bus control system with reduced complexity and increased fault tolerance of system. Steps by step procedure of designing a DMR based CAN bus control system is mention below.

1. First we subdivided the whole design of control system into sub modules.

2.Then individually design each sub module, write VHDL code for it, check design summary, RTL view and check its outputs either its works properly or not, if some correction needed than go through that make changes and then go to next step.

3. When all the outputs of sub modules come individually, then we go for final simulation of all the output in Xilinx ISE9.2i software. And find the overall design summary, RTL view, and simulated output.

4. When DMR based can bus control system is design and simulate properly, we compares DMR based and TMR based CAN bus control system outputs and check for the fault tolerance feature of the system.

V. SIMULATION/EXPERIMENTAL RESULTS

The operation of DMR based control system is explained using block diagram. The following topics are covered.

1. Block diagram.

2. Operation.

Block Diagram: Fig.1 shows the block diagram of DMR based CAN bus control system. It consists of three main modules, namely can_host, bist_system, and frame_tx_imp three main modules, i.e. Host, Comparator and Multiplexer.

Host: The operation of DMR based CAN host is as follows-

Input is taken from any sensor placed in the car (e.g.: engine) and provided to ADC, at the output of ADC we get a 8 bit digital value which is fed to the CAN host. The CAN host decodes the output obtained from the ADC and forms a particular message using CAN format and transmits it using CAN protocol. In this design we are using repetition of CAN host (i.e. CAN host 2) to increase fault tolerance of the system.

Output of host 1 will give to the input of bist_system module and frame_tx_imp module and output of host 2 give to the input frame_tx_imp module.

Functional Description of Pins used in CAN host

1. *ADC_in [7:0]:* This is digital 8-bit input from ADC.

2. *Clk:* It is a system clock signal.

3. *Inject_fault:* To check the behavior of module using test bench waveform.

4. *Rst:* Reset pin.

5. *Tx_ctrl:* transmission control pin.

6. *Frame_tx:* output signal of CAN host (i.e. data frame).

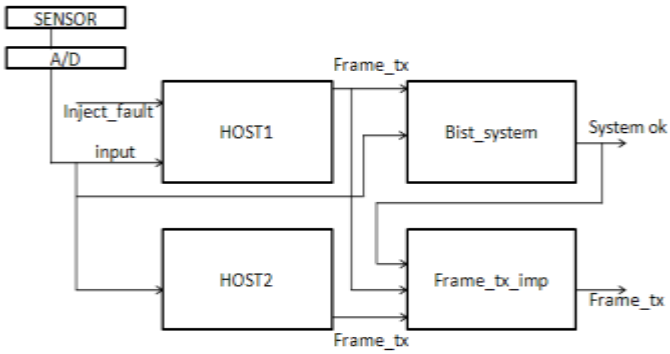

Fig.1 Block diagram of CAN bus control system

Bist_system module: The operation of Bist_system module is as follows-

1. It takes two input signals frame_tx (host output) and 8-bit digital input signal from ADC.

2. Generate CAN host data frame and compare with previous CAN host frame and provide output value i.e. system_ok.

3. If previous CAN host data frame value and generated data frame value are equal then system_ok is 1 otherwise 0 (standard logic).

Functional Description of Pins

1. *Input_data [7:0]:* 8-bit digital input signal from ADC.

2. *Clk*: This is the clock signal.

3. *Generated_output_data:* This is the generated output CAN host 1 (data frame).

4. *Rst:* Reset signal.

5. *Tx_ctrl:* transmission control signal.

6. *System_ok*: this is output signal value in standard logic.

Frame_tx_imp module: The operation of Frame_tx_imp module is as follows-

It takes inputs from both CAN host modules and input from bist_system and work as a multiplexer depending on the input value of system_ok signal.

Functional Description of Pins

1. *System_ok:* This is input signal from bist_system module.

2. *S_frame_tx:* This is input signal from CAN host 1.

3. *S_frame_tx 2:* This is input signal from CAN host 2.

4. *Frame_tx:* Output signal of frame_tx_imp module which contain CAN host data frame.

The results obtained after the simulation process for the DMR based CAN bus control system are presented. Simulations are required to ensure that the design works according to the intended application. Simulation is performed by Xilinx ISE Software Version 9.2i.The results of "control system" obtained are as follows.

Design Summary OF DMR based CAN bus control system

The results are synthesized with Target Device xc3s200-5fg456,Product Version ISE9.2i with Number of Slices used is 411,Number of Slice Flip Flops used is 495,Number of 4 input LUTs used is 739,Number of bonded IOBs is 13 and Number of GCLKs used is 1 with no errors & 12 warnings.

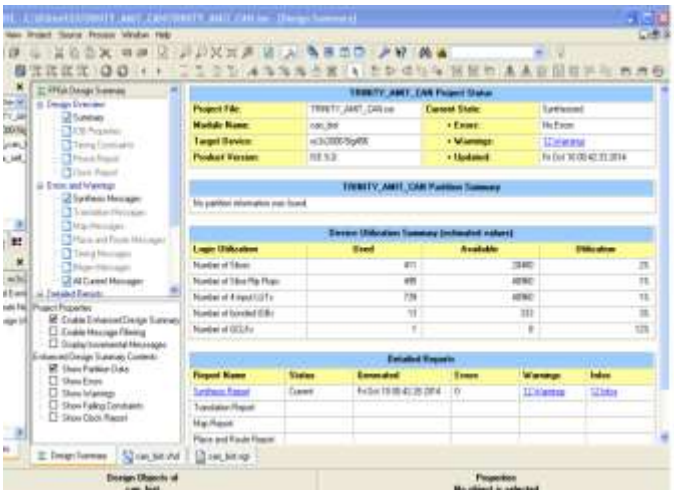

Fig.2 Design summary of DMR based CAN bus control system

The control system circuit is simulated for various selection inputs by creating a test bench waveform source of Xilinx software. The simulation result is shown in table.1

Input signals	Logic values Time At 0ns	Logic values Time At 184.6ns	Logic values Time At 212.5ns	Logic values Time At1988.5ns	Logic values Time At 2011.1ns	Logc vales time At 2065.1ns	Logic values time At 2174.9ns
Adc_in[7:0]	8'h80	8'h81	8'h82	8'h93	8'h94	8'h94	8'h95
Clk	0	1	0	1	0	1	1
Frame_tx	U	U	U	U	U	0	1
tx_ctrl	1	1	1	1	0	0	0
Rst	1	1	0	0	0	0	0
System_ok	U	1	1	1	1	1	0
Inject_fault	0	0	0	0	0	1	1

Table.1.Values of DMR based CAN control system simulation output

Comparison table

DMR is worked on same memory utilization as in case of TMR with less coplexity but fault tolerance feature of control system is increased in DMR based control system.The comparision of different features of TMR and DMR based control system are listed in Table

S.NO.	Logic Utilization	Used		Utilization %		Available
		TMR	DMR	TMR	DMR	
1	Number of Slices	337	411	1	2	20480
2	Number of Slices	354	495	6	1	40960
3	Number of 4 input LUTs	614	739	1	1	40960
4	Number of bonded IOBs	13	13	3	3	333
5	Number of GCLKs	1	1	12	12	8

Table.2 Comparison table

VI. CONCLUSION

In paper, a DMR based CAN bus control system is design by using VHDL.The main sub modules are can_host, bist_system, and frame_tx_imp designed separately. Port Mapping is used to call each sub module. We have increased the fault tolerance of control system i.e. at the cost of small area overhead complexity of system is decreased as compared TMR based control system. It means more fault tolerance can be achieved at the nearly same memory utilization as used for TMR based control system. This can be shown in below table. The design is successfully simulated using Xilinx ISE 9.2i software. The results are stable and reliable and show the correct functionality. Hence, we can improve fault tolerance feature of control system by providing alternative path for output whenever error occurs in system.

VII. FUTURE SCOPES

As in future this design can be design by using different techniques such as implementing control system with partial reconfiguration method, to increase the fault tolerance feature and to provide more robustness in operation of CAN control system.

REFERENCES

[1] Karel Szurman, Jan Kastil, Martin Straka, Zdenek Kotasek, "Fault Tolerant CAN Bus Control System Implemented into FPGA", 978-1-4673-6136-1/13, 2013 IEEE.

[2] Microchip Technology Inc, "MCP2515 - Stand-Alone CAN Controller with SPI Interface," November 2005.

[3] Robert Bosch GmbH, "CAN Specification 2.0," BOSCH, Stuttgart, Technical specification, 1991.

[4] Michael Stock, "CAN Aerospace - Interface specification for airborne CAN applications V 1.7," Stock Flight Systems, 82335 Berg/Farchach.

[5] G. Asadi, S. G. Miremadi, H. R. Zarandi, and A. Ejlali, "Evaluation of fault-tolerant designs implemented on sram-based fpgas," in Proceedings of the 10th IEEE Pacific Rim International Symposiumon Dependable Computing (PRDC'04). Washington, DC, USA: IEEE Computer Society, 2004, pp. 327–332.

[6] J. A. Cheatham, J. M. Emmert, and S. Baumgart, "A survey of fault tolerant methodologies for fpgas," ACM Trans. Des. Autom. Electron. Syst., vol. 11, no. 2, pp. 501–533, 2006.