A Special Method for Analysing and Correction of Set Effects in PIC Microcontroller

Sijoy Johnson, N. Kumaresan, Manu Poulose

Abstract: In this paper the fault tolerance behaviour of a PIC micro-controller has been checked. This experiment is based on injection of different transient faults in various points.. The experimental results have been compared in different aspects. Program counter is found to be as the most infected components after simulated results. The failure rate of this program counter is more than 50%. An SET at a node of combinational part may cause a transient pulse at the input of a flip-flop and consequently is latched in the flip-flop and generates a soft-error. When an SET conjoined with a transition at a node along a critical path of the combinational part of a design, a transient delay fault may occur at the input of flip-flop. Thus, studying the behaviour of the SET in these kinds of circuits needs special attention. This paper studies the dynamic behaviour of SET Effects in PIC microcontroller with massive critical paths in the presence of an SET. We also propose novel flip-flop architecture to mitigate the effects of such SETs in combinational circuits. Furthermore, the proposed architecture can tolerant a Single Event Upset (SEU) caused by particle strike on the internal nodes of a flip-flop.

Index Terms: ADC, Single Event Transients (SET),Single event upsets (SEU) ,UART.

I. INTRODUCTION

The importance of fault tolerance at the processor architecture level has been made increasingly important due to rapid advancements in the design and usage of high performance devices [1]. A fault-tolerant system is a system that continues to function correctly in the presence of hardware failures and/or software errors. However, an exhaustive study is essential in order to find the most tenuous components in processor architecture. This analysis can be accomplished by a formal or experimental method to find the most infected component of the microcontroller or microprocessor. Fault injection is one of popular techniques in evaluating the dependability attributes of a system [2]. It can be implemented in three different patterns including physical, software, and simulation.

Manuscript received on July, 2012.

Sijoy Johnson, Embedded System Technology, Anna University of Technology Coimbatore, India University

N Kumaresan, Electronics and Communication Engineering Anna University of Technology Coimbatore,

Manu Poulose , Control and Instrumentation , Anna University of Technology, Coimbatore, India,

Software implemented fault injection consists of reproducing the errors that would have been produced owing to faults presence at software level. It provides a cheaper and more flexible way of injecting fault than most other physical techniques. Simulated fault injection is another way in which a model of a system is simulated in other computer. In this method the logical values are infected during simulation [3]. This article considers the simulation-based fault injection method with the help of PIC SIMULATOR. We have considered an investigation of fault effects and propagation in the PIC 16F877A microcontroller. PICs microcontroller families are made by Microchip Company. Microchip also provides a freeware IDE package called MPLAB, which includes an assembler, linker, software simulator, and debugger.

Utilizing fault tolerance microcontrollers is also one of today's concerning in industrial communities. In [4] SEU's effects in a commercial processor pipeline and cache memories have been examined too. PICs families' microcontrollers are so popular among developer due to their low cost, wide availability, large user base and also extensive collection of application notes. They can be found in most industrial or other application as main controller [5] [6].

The rest of this paper is organized as follows: Section 2 is a brief introduction of PIC 16F877A architecture. Section 3 contains the fault injection process and location. The observation results are indicated in the part 4. And finally Section 5 presents some correction method and conclusion.

II. ARCHITECTURE

The PIC 16F877A is based on Harvard architecture which makes it capable of reading an instruction or reading/writing data from/to the memory simultaneously. It is a mini-RISC microcontroller and made by Microchip Technology. The PIC fetches the stored microcode in its ROM to operate. Although the term was not used at the time, it shares some common features with RISC designs. The PIC 16F877A is an 14 bit accumulator-based microcontroller, supports 36 distinct instructions, two levels stack and also two addressing modes, direct and indirect. Its memory is divided into two distinguished parts which are program and data areas. The program memory is not accessible by users and it consists of instruction commands in binary code. The data memory is a single or collection of bank registers according to the number in place of *x*. For an instance, the PIC 16F877A has just a simple register bank encompass 32×8 registers. The first eight of them are recognized as special registers and the rest as general registers.

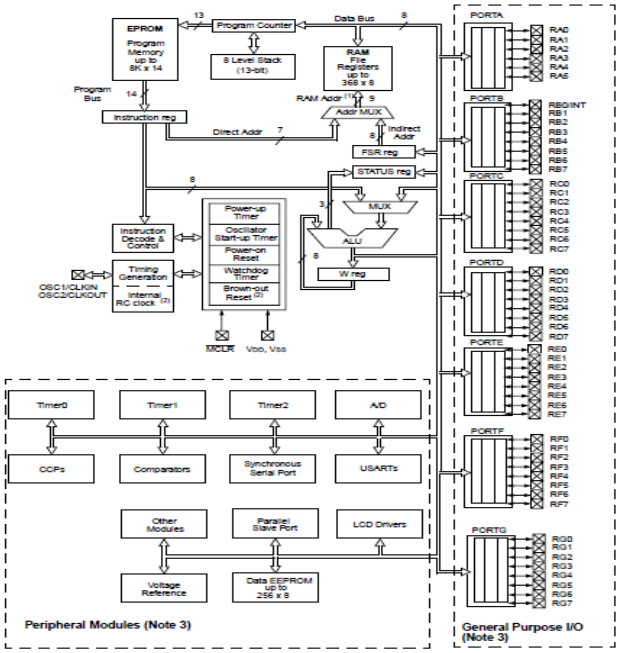

Fig.1. Architecture

III. EXPERIMENTAL SETUP

In order to assess the fault tolerance behaviour, three different programs have been applied as workloads in assembly language. These codes have been assembled and also simulated with MPLAB IDE and PIC Simulator. The three workloads are as followings:

A. Workloads

The microprocessor has been tested using three different applications adc operation, a serial data transmission using the UART, and Timer operations. The ADC is a data intensive application, which spends most of the time making data accusation and storing the results in user memory. In this application, the ALU is the most used component. The Serial Communication program consists in a continuous flow of characters going through the serial port. The UART is the most used component in this application. The timer is set to trigger a periodic interrupt. The interrupt service routine increments a tick counter, and several functions are used to update the system date and time.

TABLE I
FAULT INJECTION WORK LOAD CHARACTERISTICS

	Clock cycle	Total Faults Injected	Average Fault/Mem
ADC Operation	132	3200	320
Serial Communication	68	1280	128
Timer Operation	148	3700	270

Every delay has been quantized using 4-bit resolution. A clock period slightly larger than the critical path delay has been selected, which results in 210 time quanta with this resolution. Faults were injected randomly at any clock cycle and time quantum inside the clock cycle, with an average of 8 SETs per clock cycle in one out of 7 clock cycles. Only some clock cycles at the beginning have been left out.

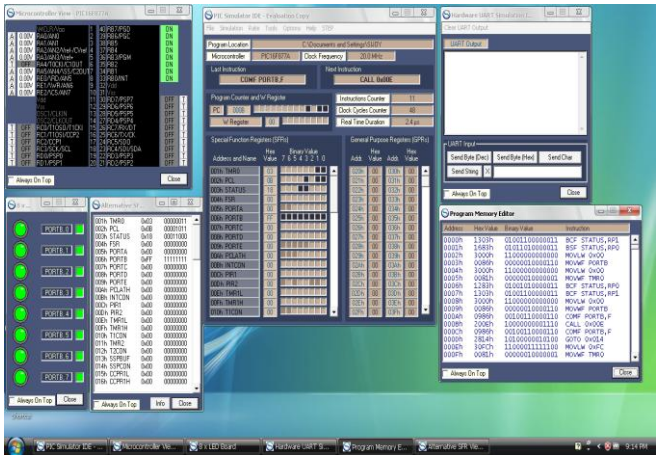

Fig.2. Simulator

According to the generic definition and concepts of fault activity attributes, faults are divided into two classes, propagated and not-propagated. In most of evaluations the propagated ones, real faults, are considered. This experiment also relies on real faults. The propagated faults can be partitioned into two subclasses, latent and active. Active faults relates to some which have been detected. Latent ones pertinent to faults which are present in the system; however, they have not been detected yet.

IV. OBSERVATION AND RESULTS

In the tables shown below, classification of faults is made with respect to three categories. The main three categories are Failure, Recovered and Latent.

TABLE II
FAULT CLASSIFICATION: ADC OPERATION

	Failure	Recovered	Latent
Working Register	18	270	32
Status Register	2	64	54
Option Register	92	32	196
Program Counter	200	100	20
TXSTA Register	0	0	320
RCSTA Register	0	0	320
Port B Register	0	0	320
T1CON Register	0	0	320
T2CON Register	0	0	320
INTCON Register	0	0	320
Total	312 9.75%	466 14.56%	2422 75.68%

Recovered means the error has no functional effect and eventually disappears. Failure means the error propagates to the output port or to several memory positions. Latent means the error does not propagate to the output port, but at least one sequential element (memory or flip-flop) contains an incorrect value at the end of the execution. As expected, most of the faults are Recoverable and Latent due to the masking effects described in the introduction. In all cases, the Program counter is the most critical. The Ports registers are very small, but it produces many failures as it controls the observed outputs.

TABLE III
FAULT CLASSIFICATION: SERIAL COMMUNICATION

	Failure	Recovered	Latent
Working Register	16	97	15
Status Register	7	121	0
Option Register	0	0	128
Program Counter	64	41	23
TXSTA Register	25	103	0
RCSTA Register	0	0	128
Port B Register	0	0	128
T1CON Register	0	0	128
T2CON Register	0	0	128
INTCONRegister	0	0	128
Total	112 8.75%	362 28.28%	806 62.96%

TABLE IV
FAULT CLASSIFICATION: TIMER OPERATION

	Failure	Recovered	Latent
Working Register	18	338	14
Status Register	1	83	286
Option Register	114	48	208
Program Counter	250	120	0
TXSTA Register	0	0	370
RCSTA Register	0	0	370
Port B Register	8	16	346
T1CON Register	0	0	370
T2CON Register	0	0	370
INTCON Register	0	0	370
Total	391 10.56%	996 26.91%	2704 73.08%

V. PROPOSED STRUCTURE

In the sequel, we propose two SET-tolerant flip-flops for transient delay sensitive paths and transient delay insensitive paths of the design. The proposed architectures detect and correct the transient pulse and transient delay fault at the input of the flip-flop. Furthermore, for completeness of the protection the proposed architectures can also protect flip-flops against a possible SEU in the internal nodes of the flip-flops.

A. Delayed sampling protection method

Three-sampling scheme is a conventional approach to detect the erroneous pulse at the input of a flip-flop ([7] and [8]). Figure 3 shows a three-sampling scheme to detect a transient pulse. CLK and D are the clock and data inputs of the flip-flop, respectively. Using three samples a, b, and c, a three-sampling scheme detects and corrects a possible transient pulse on D. To guarantee the correctness of this algorithm, the time interval between each two consecutive samples should be greater than the maximum width of the transient pulse (i.e., $_{max}\Delta \geq \tau$). The first sample is latched at $_{max}\Delta > \tau$ time before the rising edge of the clock. The second sample is latched at the rising edge of the clock. Finally, the third sample is latched at $_{max}\Delta > \tau$ after the rising edge of the clock. In this scheme, b will be selected as the default output. If there is a discrepancy between the first two samples, the third sample will be selected as the output. The first sample is called voter sample, the second sample is called main sample, and the third sample is called arbiter sample. The maximum timing penalty of this method in the presence of a transient pulse is Δ.

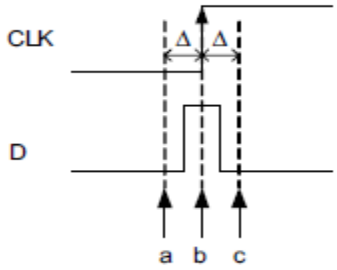

Fig.2. Transient Pulse

An architectural or circuit technique for implementing the proposed sampling methods should consider the following design issues. It should implement delay sampling method to eliminate all possible SET or SEU in the combinational and sequential parts. The proposed techniques and structures introduce a low power, time and area overhead in the normal operation of the circuit. Reusing the present test structures (*e.g.*, scan flip-flops) in a circuit to cope with SET and SEU issues may be a promising technique to propose an optimum (low power, time, and area overhead) SET/SEU tolerant structure.

Using scan latches in parallel with system latches is becoming an efficient way to handle different problems during test and debug of a circuit ([9] and [10]). Sharifi, et al. [9] propose a selective trigger scan architecture made of two parts (system part and test part) to reduce the test data volume and test dynamic power consumption. Kuppuswamy, et al. [10] propose a microprocessor full hold-scan architecture that comprises two distinct circuits: a system flip-flop and a scan portion.

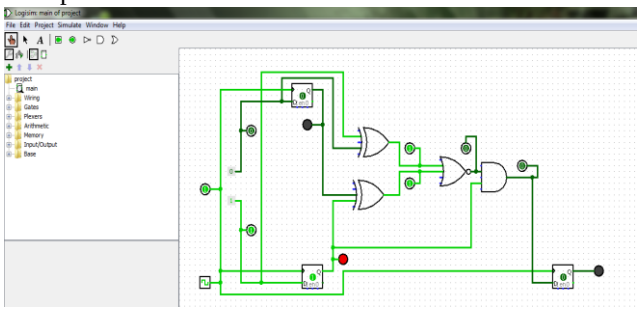

Fig.3. Delayed sample scheme

Using the scan portion of these types of flip-flops, we implement the proposed sampling methods to obtain a soft error tolerant flip-flop. Reusing scan part flip-flop and using clock gating technique, Figure 4 shows our proposed architecture to detect a transient pulse at the input of the flip-flop. The flip flop architecture consists of three parts: system, scan, and

protection portions. Protection portion consists of three gates (an XOR, an AND, and an NOR) and a delay generator. The clocking scheme of the proposed architecture is based on the pulse-flip-flops. Using a delay generator, the proposed architecture samples the first two samples simultaneously. If there is a discrepancy between samples a and b the third sample is latched as the output of the flip-flop.

In the proposed system the delay generator will create a delay which is equal to Δ. The scan portion will get the delayed output which will be compared with system output using XOR gate. If there is a discrepancy between scanned input and system input the data will not be propagated to the further registers. The speciality of this system is that the system will check both input and output as well. By checking both input and output we can create more tolerant system. This will allow the total system to work without SET or SEU errors.

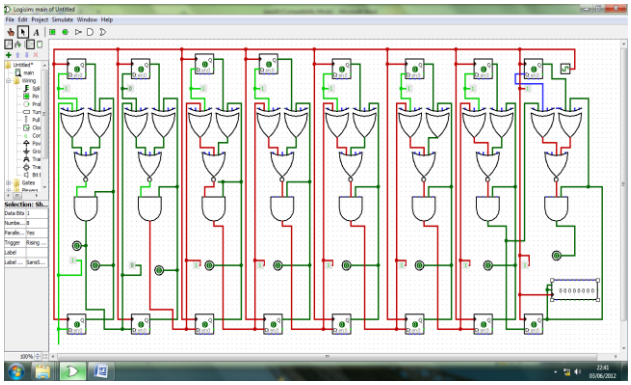

Fig.4. Delayed sample scheme for 8 bit

In addition, sample a, b and c is also latched as the final output if there is a transient delay at the data input. This architecture can also tolerate an SEU at its internal node if $t > 2\times\Delta$ SETh. The condition $t > 2 \times \Delta$ SETh , which guarantees the SET detection, is compassed by considering a minimum-path length constraint during the design process. This minimum path length can be realized by adding buffers to the shortest path during logic synthesis. Therefore, this process introduces a certain amount of power and area overhead.

VI. CONCLUSIONS

In this paper we have proposed a method to evaluate the criticality with respect to SET effects of each gate in a circuit, so that by selectively hardening the most critical cells, the design can satisfy SER requirements with reduced impact on area, timing and power consumption. The method provides information to support space and time redundancy techniques. This paper considers logic circuits with many critical paths; and studies the effect of single event transient (SET) caused by particle strike on the nodes along the critical paths. This paper shows three different erroneous effects of an SET at the input of a flip-flip: The paper also proposes two flip-flop architectures to detect and correct these erroneous effects.

REFERENCES

1. Y. Nakamura, and K. Hiraki. , "Highly Fault-Tolerant FPGA Processor by Degrading Strategy," Pacific Rim International Symposium on Dependable Computing, pp. 75-78, 2002.
2. J. Aidemark, J. Vinter, P. Folkesson, and J. Karlsson, "GOOFI: Generic Object-Oriented Fault Injection Tool," International Conference on Dependable Systems and Networks, pp. 83-88 , 2001.
3. D. Gil, J. Garcia, J. C. Baraza, and P. J. Gil, "A Study of the Effects of Transient Fault Injection Into the VHDL Model of a Fault-Tolerant Microcomputer System," 6th IEEE International On-Line Testing Sysmposium, pp. 73-79, 2000.
4. M. Rebaudengo, M. Sonza Reorda, and M. Violante, "Analysis of SEU Effects in a Pipelined Processor," Proceedings of the IEEE International On-Line Testing Sysmposium, pp. 206-210, 2002.
5. E. Chaboot, J. McCluskey, J. Wu, and Y. Sun, "Microcontroller-Based Artificial Synapse," Proceedings of the IEEE 31st Annual Northeast in Bioengineering Conference, pp. 30-31, 2005.
6. K. Tunlasakun, K. Kirtikara, S. Thepa, V. Monyakul, "A Microcontroller-Based Islanding Detection For Grid Connected Inverter," The 47th Midwest Symposium on Circuits and Systems, pp.267-269, 2004.
7. S. Krishnamohan, and N. R. Mahapatra, "A high efficiency technique for reducing soft errors in static CMOS circuits," Proc. IEEE Inter. Conf. Computer Design (ICCD'04), Oct. 2004, pp.126-131.
8. D. G. Mavis, and P. H. Eaton, "Soft error rate mitigation techniques for modern microcircuits," IEEE Proc. Annual Reliability Physics Sym., 2004, pp. 216-225.
9. S. Sharifi, M. Hosseinabady, and Zainalabedin Navabi, "Reducing Power, Time and Data Volume in SoC Testing Using Selective Trigger Scan Architecture", International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT 2003), 2003, , pp. 352-360.
10. R. Kuppuswamy, P. DesRosier, D. Feltham, R. Sheikh, and P. Thadikaran, "Full hold-scan system in microprocessors: Cost/Benefit analysis," Intel Technology Journal, Vol. 18, No. 1, Feb. 2004; http://developer.intel.com/technology/itj/2004/volume08isssu e01/.
11. J. M. Rabaey, A. Chandrakasan, and B. Nikolic, "Digital Integrated Circuits," Pearson Education, Inc. Upper Saddle River, New Jersey 07458, 2003.

AUTHORS PROFILE

Sijoy Johnson -Currently doing ME in embedded system technology in Anna University of technology Coimbatore .Completed B tech in Electronics and Communication Engineering from Calicut university Kerala

N. Kumaresan - Working as an assistant professor in Anna university of technology coimbatore..

Manu Poulose - Doing ME in Control and Instrumentation at Anna University of Technology ,Coimbatore. Completed B.Tech degree in Electrical and Electronics Engineering from Mahatma Gandhi University , Kerala

.