Time Redundancy Processor with a Tolerance to Transient Faults Caused by Electromagnetic Waves

Makoto KIMURA, Masayuki ARAI, Satoshi FUKUMOTO, and Kazuhiko IWASAKI
Tokyo Metropolitan University, 6-6, Asahigaoka, Hino, Tokyo 191-0065, Japan
kimura@iel.sd.tmu.ac.jp, {arai, fukumoto, iwasaki}@eei.metro-u.ac.jp

Abstract

With the advancement of semiconductor manufacturing technology into deep-submicron processes, accompanied by higher frequency and lower power consumption, the effects of transient faults on digital circuits are becoming more significant. In this paper, we discuss a time redundancy processor that is designed to tolerate transient faults caused by electromagnetic waves arising from the short-circuit discharge of a capacitor. Firstly, the effects of electromagnetic waves on a sequential circuit implemented on a FPGA were evaluated. A fault model was then established according to the experimental results. Next, a time redundancy processor was designed in order to tolerate against the established fault model. The proposed processor was then implemented on an FPGA and its resistance to transient faults was evaluated.

1. Introduction

With advancements in semiconductor manufacturing technology, it is expected that in near future, VLSI will be designed based on the 65 nm, 45 nm or smaller processes. By reducing the transistor size, noise margins for memory will decline, because the critical charge to flip the stored data will be reduced. Also, crosstalk or electromagnetic noise will have more impact on digital circuits. As a result, the degradation in tolerance of devices against so-called "transient faults" will become a more significant problem.

Transient faults are temporary faults caused by factors such as nuclear radiation or electromagnetic pulse. These temporary faults in hardware are sometimes called "soft errors". In this paper, we use the term soft error for the temporary faults induced by radiation, such as cosmic ray neutrons and alpha particles. More over a temporary faults, including soft errors above, will be referred to as transient faults. Unlike permanent hardware faults, such as 0/1 stuck-at faults, the effects of transient faults can be removed by rebooting of the system or by recovery and retry using backup data. It should be noted that not all transient faults always have a serious effect. Whether a transient fault is activated or not depends on the application. However, transient faults are very difficult to detect in production tests prior to shipping. Especially in the case of mission-critical systems, such as those in the aerospace or automotive fields [1], once a transient fault is activated, the reliability of the operating phase of the system can be significantly reduced.

Currently, approaches to reduce the effect of transient faults have been studied from many perspectives. In the field of circuits, shield protection and application of Error Correcting Code (ECC) for memory are adopted. Furthermore, in the architecture field, spatial and time redundant techniques have been well researched [2]-[5]; the former multiplexes various modules inside the processor, and the latter calculates the same operation several times. While these techniques are mainly focused on the single-bit faults by soft errors, countermeasures are becoming necessary, not only for single-bit faults, but also for the multi-bit faults by general transient faults [6]. However, architecture level tolerance for multi-bit faults has not been well reported.

In this paper, we discuss the use of a time redundancy processor in order to achieve tolerance against transient faults caused by electromagnetic waves. While transient faults induced by electromagnetic waves are basically temporary defects, in many cases the impact and the area of influence is larger than the case of soft errors induced by such as cosmic ray neutron. This is because electromagnetic waves affect the device on the "surface," in contrast to soft errors, which affect the device at a "point." Electromagnetic waves further have larger energy, so that multi-bit faults should be assumed to take place [7].

Time redundancy is a well known approach to achieve high reliability [8]. The fundamental concept of time redundancy is to calculate the identical operation multiple times for different timings. With this, error is detected or masked using a comparator or voter. If the error is transient, it is possible to obtain the correct answer by calculating the same operation again. Time redundancy is suitable for cases where hardware redundancy, such as chip multiplexing, is

not allowed. As another approach to overcome noise in the clock signal line (e.g., crosstalk noise), a multiplexing clock pulse is proposed.

Here, we study the influence of transient faults due to capacitor discharge which affects the processor. Firstly, we will evaluate a model for a transient fault, based on experiments for the effects of electromagnetic waves due to short-circuit discharge of a capacitor. The design and implementation of the time redundancy processor is then described. Finally, we present the results of experiments for transient fault injection with the proposed processor.

2. Transient Fault Modeling

As shown in Fig. 1, the mechanism for generation of transient faults is roughly classified into two cases. In Case 1, the memorizing node of the latch circuit which constitutes a flip-flop (FF) is affected by noise, and the stored data is directly flipped. For the case of soft errors, this type of fault occurs by the attack of particle such as alpha particles or neutrons, and this is called a single event upset (SEU). SEU has become a serious concern, mainly in DRAM or SRAM-based FPGA [9]. Case 2 represents an event where noise or a transient pulse that occurs in the digital circuit reaches a FF. If the arrival of noise to the data signal line is at the same time as the clock pulse, the FF might capture an erroneous value. For the case of a soft error, this is called a single event transient (SET). The higher the operational frequency becomes, the more frequently noise may be captured by the FFs.

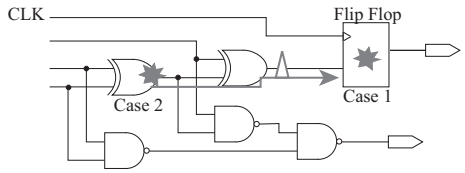


Figure 1. The mechanism of a soft error. (Case 1: Single event upset, Case 2: Single event transient)

Unlike SETs, when noise occurs on the clock signal line instead of the data signal line, the FF may capture the incorrect value at an incorrect timing. Basically, a soft error causes a single-bit error for either Case 1 or Case 2. In contrast, for the case of transient faults, as considered here, we have to deal with multiple-bit errors.

In this study, we first performed experiments to investigate the effects of electromagnetic waves. Short-circuit discharge of a capacitor was employed to generate electromagnetic waves on a sequential circuit implemented in an FPGA. A fault model for the transient fault was then determined based on the experimental results.

Firstly, an experiment was performed for a Case 2 transient fault. Figure 2 shows the arrangement of the circuit under test (CUT). The CUT consists of two registers and an indicator. Each of the two registers is 256-bits. Register 1 alternately outputs all-0 and all-1 to Register 2 during every clock cycle. Register 2 receives the output from Register 1 when the latch signal is high (active). The indicator counts the number of 1s in the outputs of Register 2, and a seven segment LED displays the number of 1s. If Register 2 receives data different from all-0 or all-1, it is determined that an error has occurred. When the indicator detects an error, it disables the Latch signal input. Register 2 then stops receiving the data until the reset signal (RST) is activated. We can identify the type of error, that is, whether the error occurs when receiving an all-0 (“0→1 error”) or when receiving an all-1 (“1→0 error”), by the Sign signal output from Register 1. This signal is controlled by the latch signal input from the indicator, and once an error occurs, it is fixed. This circuit was implemented on an FPGA, and electromagnetic waves from a capacitor discharge were applied to the clock signal (CLK). The experimental conditions are given below. The short-circuit discharge of the capacitor is performed immediately above the FPGA chip (approximately 2 cm above). Figure 3 shows the short-circuit discharge of the capacitor.

- FPGA: Xilinx SPARTAN2 XC2S200-5PQ208C
- Synthesizing tool: Xilinx ISE webpack 7.1
- Capacitor capacity: 6800 uF
- Clock signal: 72 kHz clock signal supplied from a function generator (FG)
- Voltage of capacitor charge: 10 V
- Number of trials (discharge): 200 times

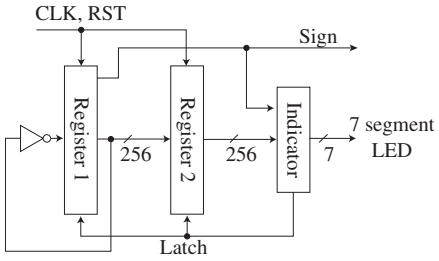


Figure 2. Experimental circuit under test used to capture the effects of transient faults caused by capacitor discharge.

Figure 4 shows a pie-graph representation of the experimental results. The total rate of error occurrence was approximately 75%. It was verified that the electromagnetic waves resulted in a higher rate of errors, and the number

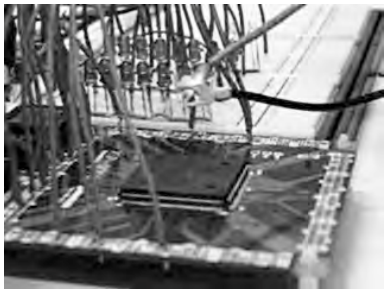


Figure 3. Injecting the transient fault caused by electromagnetic waves from the short-circuit discharge of a capacitor

of 1→0 error occurrences was significantly higher than that of 0→1 error occurrences under these experimental conditions.

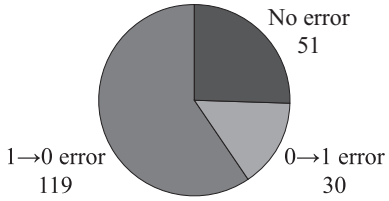
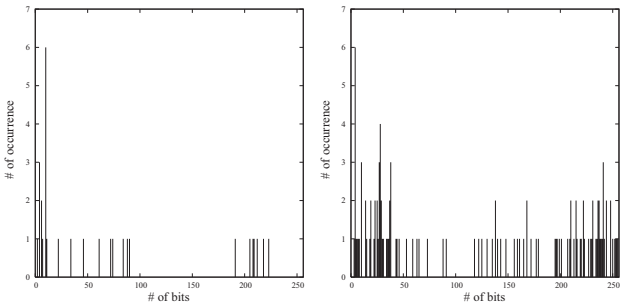


Figure 4. Details for 200 trial experiments of capacitor discharge.

Figure 5 illustrates the distribution of the number of erroneous bits affected by one discharge. The x-axis shows the number of affected bits, and the y-axis shows the frequency of occurrence. The sum of y-axis occurrences from Fig. 5(a) is equal to 30, the number of 0→1 errors. Similarly, the sum of y-axis occurrences from Fig. 5(b) is equal to 119, the number of 1→0 errors. From Figures 5(a) and 5(b), although the distribution of the number of erroneous bits is not completely uniform, we can confirm that the transient faults caused by electromagnetic waves has a larger scale impact than that of soft errors by cosmic-ray neutron and so on, with respect to the number of affected bits.

We then performed an experiment for Case 1 with the same CUT and FPGA, where the capacitor discharges were performed when the clock signal was not supplied. As a result, we verified that the faults like SEU never occurred and the configuration data of the FPGA was not damaged.

Based on the results for Case 1 and Case 2, it is speculated that an instantaneous pulse on the clock signal line, due to discharge of the capacitor, is the major factor for transient faults occurring on the implemented circuit. Therefore, the fault model is defined as follows:



(a) 0→1 error. (b) 1→0 error.

Figure 5. The distribution of erroneous bits caused by one capacitor discharge.

1. The transient fault does not affect the internal signal line in the chip.
2. The transient fault affects only the external clock signal line.
3. The duration of the transient fault is no more than two clock cycles.

3. Time Redundancy Processor

In this section, we describe a time redundancy processor which has tolerance against the fault model given in the previous section. A subset of the instruction set for a Renesas Technology Corp. H8/300 microprocessor was applied to the proposed processor. 35 out of 57 instructions were implemented. The time redundancy processor is described by 2000 lines of VHDL code on the register transfer level.

In our proposed scheme, the time redundancy executes each state of the individual instructions twice. It is possible to detect errors caused by transient faults by comparing the results of the first and second executions.

While hardware redundancy is generally adopted for transient faults, in the present fault model, the same error could occur in the multiplexed registers, and comparison or majority voting would not always be successful in our fault model. Although execution time is doubled in the proposed processor, it is thought that such a malfunction can be avoided.

Figure 6 shows the state transition diagram for the proposed time redundancy processor. The states FET1 to FET3 represent the state of the instruction fetch, and EXE1 to EXE4 represent execution. REC represents the recovery state that is executed when an error is detected. The time

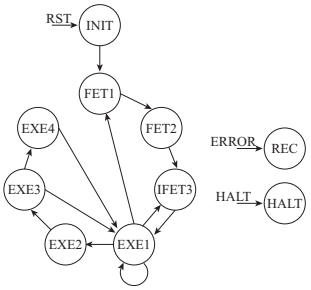
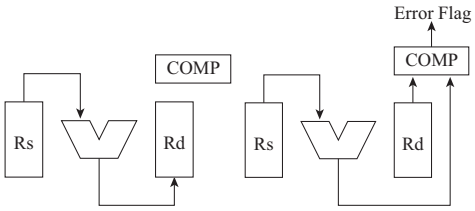


Figure 6. State transition diagram for the time redundancy processor.

redundancy processor executes each state twice and compares the results.

The operation of the time redundancy processor is described as follows. Figure 7 shows an example of operation, where the original value used for execution is not changed. For the initial execution (Fig. 7(a)), the processor executes



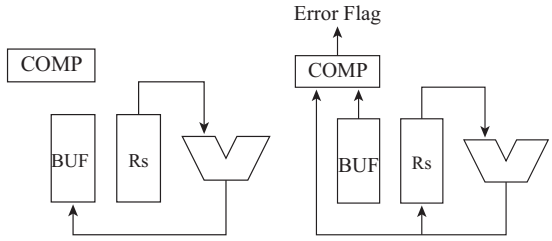
(a) Initial execution (b) Second execution

Figure 7. Example of the processor operation where the original value is unchanged.

the operation using the value in the register Rs, and stores the result to Rd. In the second execution (Fig. 7(b)), the processor compares the result of this execution with the value in Rd.

If the comparator (COMP) observes a disagreement, then it is determined that Rd has an erroneous value according to our fault model, since it is supposed that the transient faults occur in the clock signal line. Because the value in Rs is not changed, the internal state is recovered by a retry from the initial execution step given in Fig. 7(a).

When the processor executes the operation with Rs and stores the result to Rs, it is not able to compare the two results in the way described above. In this case, it compares two results by storing the result of the initial execution to a buffer. Figure 8 shows the examples of this operation using a buffer (BUF). First, the processor executes the operation with the value in Rs, and stores the result to the BUF (Fig. 8(a)). Next, the result of the second execution is compared with the value in BUF (Fig. 8(b)). If a disagreement is observed, then it is determined that the BUF has an erroneous value, for the same reason as that mentioned above. In this case, the internal state is recovered with a retry from the initial step given in Fig. 8(a). If a disagreement is not detected, then the result of this execution, is sent to Rs and the processor moves to the next state. After copying, as shown in Fig 9, the processor compares Rs with BUF, and checks whether the correct value was stored in Rs or not. If a disagreement is detected, the internal state is recovered by copying the value in BUF to Rs. This comparison is simultaneously performed with the first execution of another operation.



(a) Initial execution (b) Second execution

Figure 8. Examples of operation when using a buffer (BUF).

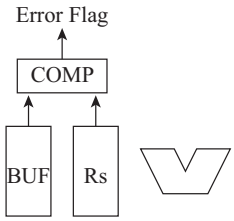


Figure 9. Comparison of Rs with BUF.

When the data is continuously transferred, it is not possible to compare the results in the way described above. In this case, the results are compared by duplicating relational registers. Figure 10 shows examples for this type of operation. Firstly, Rd1 captures the value of Rs1, and then Rs1 captures another data value (Fig. 10(a)). Next, a comparison of the Rd1 and Rs2 values is performed (Fig. 10(b)). If a disagreement is detected, then it is determined that the Rd1 has an erroneous value for the same reason given above. In this case, the values of Rs2 and Rd2 are copied into Rs1

and Rd1 respectively. The state is then recovered by a retry from the initial step given in Fig. 10(a). If a disagreement is not detected, then Rd2 captures the value of Rs2, and Rs2 captures the same value of Rs1. The proposed processor then shifts to the next set of operations. Fig. 11 shows the comparison of Rs1 and Rd1 with Rs2 and Rd2 used to detect errors. If disagreement is detected, then Rs1 and

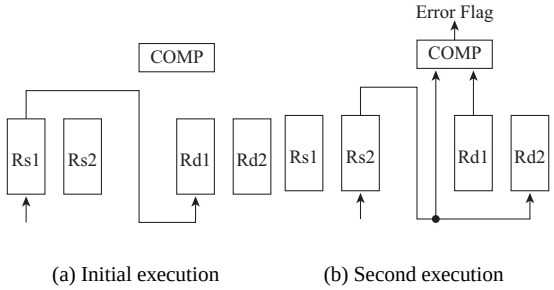


Figure 10. Examples of operation with continuous data transfer.

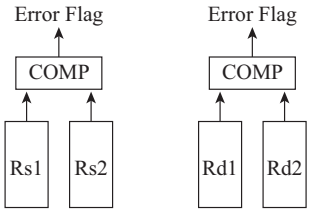


Figure 11. Comparison of Rs1 and Rs2 with Rd1 and Rd2.

Rd1 are copied into Rs2 and Rd2, respectively, in order to recover the state. This comparison is simultaneously performed with the first execution of another operation.

4. Experiment

In this section, we evaluate the tolerance of the implemented proposed processor by injecting transient faults caused by electromagnetic waves. For comparison, a normal processor without any redundancy was also implemented.

4.1. Implementation on an FPGA

Table 1 shows the result of a logic synthesis for the proposed and normal processors. Two processors were synthesized using Xilinx ISE. The chip slice (CS) ratio is the relative ratio of the number of chip slices against the one for the normal processor. The FF ratio and look-up table (LUT) ratio were derived in a similar way.

Table 1. Result of the logic synthesis.

Processor	CS ratio	FF ratio	LUT ratio
Normal	1	1	1
Proposed	1.63	1.58	1.65

Two processors were implemented on a circuit board with a Xilinx Virtex-E FPGA (XCV300E-6PQ240C). Some modules were added to the processors in order to check operations (Fig. 12). DETECTOR checks the occurrence of an error and a failure. In the case of failure or error detection, a corresponding signal is activated. The error signal is activated when an error is detected in the processor. The failure signal is activated when an error is the propagated primary output of the processor.

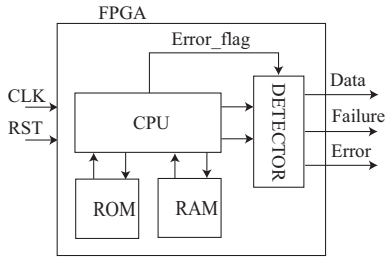


Figure 12. Implementation of the processor.

4.2. Experiments of transient fault injection

Transient faults were injected by the short-circuit discharge of a capacitor, similar to the experiments discussed in section 2, and the capability of error detection and state recovery were evaluated. The experimental conditions were as follows.

- Capacitor capacity: 6800 uF
- Clock supply: 100 kHz clock signal supplied from FG
- Voltage of capacitor charge: 10 V
- Number of trials: 100 times for each processor

The implemented processor repeatedly performed inter-register data transfer instructions. Figure. 13 shows the result for the normal processor. The “Correct” item represents those cases where there was no effect of transient faults on the FFs. The ”Failure” item represents those cases where the effect of transient faults that were captured in the FFs were propagated to primary outputs. Figure 14 shows the

results of the experiment for the time redundancy processor. The "Recovered" item represents those cases where a transient pulse was captured by the FFs, but the processor succeeded in the recovery of the state. The Failure item in this figure represents that where Recovery was failed, and an erroneous value arrived at the primary outputs.

Figures 13 and 14 show that the error occurrence rate of the time redundancy processor was almost the same as that for the normal processor. Nevertheless, the rate of correct operation for the time redundancy processor was 79%, which is much higher than that of the normal processor at 45%. From the results, we can confirm that the time redundancy processor has higher reliability than the normal processor under the influence of electromagnetic waves. However, the ratio of recovery for detected errors is approximately 60%, and our goal, which is to design a processor with complete tolerance for this fault model, has not yet been achieved.

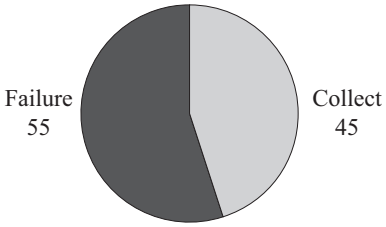


Figure 13. Details of the experimental results for the normal processor.

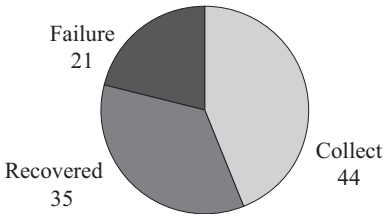


Figure 14. Details of the experimental results for the proposed processor.

5. Conclusion

In this paper, we first examined the influence of transient faults, which originate in the short-circuit discharge of a capacitor, to the sequential circuit on an FPGA. According to this result, we confirmed that electromagnetic waves transiently affect the clock signal and result in errors when the FFs receive data. Therefore, a time redundancy processor was designed, implemented on an FPGA, and the tolerance against transient faults was validated by the injection of transient faults. However, at this time, we were not successful in achieving perfect tolerance to transient faults. Future work will be focused on a more detailed modeling of transient faults, and clarification of the differences between experimental and simulated results, which are required for the design phase. Comparison of our technique with other schemes and evaluation of fault tolerance for other kind of faults will be also studied. In addition, further investigations will concentrate on superior methods for fault injection, with regard to controllability and observability, the consideration of recovery methods with higher reliance, and the reduction of performance and hardware overheads.

References

[1] F. Corno, F. Esposito, M. Sonza Reorda, and S. Tosato, "Evaluating the Effects of Transient Faults on Vehicle Dynamic Performance in Automotive Systems," International Test Conference 2004, pp. 1332-1339, Oct. 2004.

[2] T. M. Austin, "DIVA: A Reliable Substrate for Deep Submicron Microarchitecture Design," International Symposium on Microarchitecture, pp. 196-207, Dec. 1999.

[3] T. N. Vijaykumar, I. Pomeranz, and K. Cheng, "Transient-Fault Recovery Using Simultaneous Multithreading," International Symposium on Computer Architecture, pp. 87-98, May. 2002.

[4] T. Sato, "A Transparent Transient Faults Tolerance Mechanism for Superscalar Processors," IEICE Transactions on Information and Systems, Vol. E86-D, No. 12, Dec. 2003.

[5] S. Mitra, M. Zhang, T. M. Mak, N. Seifert, V. Zia, and K. Sup Kim, "Logic Soft Errors a Major Barrier To Robust Platform Design," International Test Conference 2005, No. 28.3, Nov. 2005.

[6] S. Kundu, "Panel 3: Is The Concern for Soft-Error Overblown?," International Test Conference 2005, p. 3, Nov. 2005.

[7] Y. Koga, T. Matsubara, and Y. Hayashi, "The Electro-Magnetic Pulse and Its Effects," 43th FTC meeting in Japan, Jul, 2000 (in Japanese).

[8] M. Nicolaidis, "Time Redundancy Based Soft-Error Tolerance to Rescue Nanometer Technologies," VLSI Test Symposium 1999, pp. 86-94, Apr. 1999.

[9] A. Messer, P. Bernadat, G. Fu, D. Chen, Z. Dimitrijevic, D. Lie, D. Devi Mannaru, A. Riska, and D. Milojicic, "Susceptibility of Commodity Systems and Software to Memory Soft Errors," IEEE Transactions on Computers, Vol. 53, No. 12, pp. 1557-1568, Dec. 2004.