Fast SEU Fault Injection in the SoC-Memory

Vladislav Kh. Khanov, Sergey A. Chekmarev
Siberian State Aerospace University named after Academician M.F. Reshetnev, SibSAU
Krasnoyarsk, Russia

Abstract – **In this paper the SoC-microprocessor testing problem using SEU fault injection in the SoC-memory is considered. It is substantiated the minimal time delay of fault injection is a necessity for some tests. We overview the current approach to SEU fault injection ensuring the minimal time delay. It is offered the fault injection method in the SoC-microprocessors using the on-chip fault injector. Possibilities of the proposed method are suggested. By way of illustration the injection in LEON3 SoC-microprocessor is regarded. The general injection algorithm is described. The outcome of experiment supported the injection minimal time delay are given.**

Index Terms – **Fault tolerance of integrated circuit, fast fault injection, system-on-chip (SoC), on-chip fault injector.**

I. INTRODUCTION

FAULT Tolerance (FT) is an important requirement for digital equipment working in hard conditions, such as in space. The providing fault tolerance of entire onboard system includes fault tolerance of Electronic Component Base (ECB), onboard system architecture, and software.

The fault injection methods are used for the fault tolerance testing of complete onboard system or ECB including microprocessors [1, 2]. These methods emulate the space ionizing effect. FI testing methods for the microprocessor architectural treatment and software provide SEU sensitivity check. SEU is visualized in register-oriented memory architecture [3]. This type is the most common but it is reversible which entails no instant fault of onboard system. Also it is parried well for example by Error Detection and Correction (EDAC) based on Error Corrections Code (ECC) [4].

In recent times the SoC-microprocessors are widely used for onboard equipment for example microprocessor LEON2/LEON3 [5], so fault tolerance task is important today. In this paper FI-concept is considered for purposes of SoC-microprocessors.

It is necessary to define the emulate fault model for FI-campaign determining the real scenario of space ionizing impact on microprocessor work. So the important question is time rate and minimal time interval between two adjacent events FI. In a number of papers the fast injection methods [6] and real time FI [7, 8] are considered. The term «real time FI» is used by authors [8] similarly to term «real time debugging», however has little in common with real time theory. Based on global approach with real time debugging using resource access via the testing port and On Chip Debugging (OCD) [9], it means that program flow and FI are not dependent because injection is without the microprocessor stopping. At that proposed design does not consistent with the low invasiveness principle during the FI in internal microprocessor memory that limits practicality.

II. STATEMENT

The necessity of the fast FI with the minimal time delay between two injections is considered. There is provided the fast FI method for SoC-microprocessors submitted in a form of microprocessor IP-block. For FI the OCD-method is used. Solution should be minimally invasive. Injection is in internal microprocessor memory (registers, cache) and external short term memory. The experimental task is to try the developed approaches in the context of FI in LEON3 SoC-microprocessor memory. It is needed to estimate the time parameters for injection in internal and external memory, and prospects of approach application.

III. SEU FAULT INJECTION WITH MINIMAL DELAY

Most modern microprocessors such as SoC-microprocessors have OCD-design what allow to get access to internal resource. Feature a property such as reading and modification of internal registers, cache, and external memory provides a useful mechanism for FI.

OCD-fault injection in internal memory consists of: 1) breakpoint installation 2) breakpoint suspends program execution allowing to read the random memory cell and change it in incorrect value, for example SEU; 3) execution recovery of suspending program. Further if the memory is accessed by background program through reading of incorrect memory cell then EDAC-mechanism should fix a fault. The abnormal program termination will be executed in case of EDAC-mechanism is implemented incorrectly or does not match the fault ratio. Statistics on the number of introduced fault and fixed (or unfixed) error allow for the conclusion about microprocessor memory protective value.

FI in external short memory with microprocessor stopping is similar; however injection is by the storage controller. Also there is opportunity to structure the injection without microprocessor stopping in external memory at time when memory is not accessed by microprocessor.

FI executing with single scenario and single campaign consists of several time phases (Fig. 1), where *t1* – injection setup time, *t2* – single injection runtime is what memory cell inverts at *t1*, *t3* – inter-injection time.

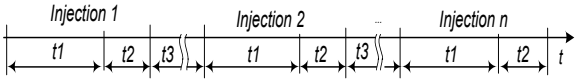

Fig. 1. Injection process.

Time length *t3* determines the injection rate and depends on scenario requirements of executable campaign; *t1* is determined by FI system capabilities. IF injection scenario involves that *t3=0*, then injection intervals are determined only by *t1*.

Here we have several examples when injection intervals should be minimal. Prime example is injection for a multi bit upset (MBU) imitation. MBU make simultaneous inverting of several adjacent memory cells. Fig. 2 presents double fault options in array of memory cells depending on heavy ion impacts on memory. In case A the injection must be made in two adjacent cells of one memory word during one writing operation. In cases *B* and *C* double fault may be made only for two operations: one bit inverting in one memory word first and then doing that in

other memory word. For presenting this process by single operation it is necessary that *t3* be minimal.

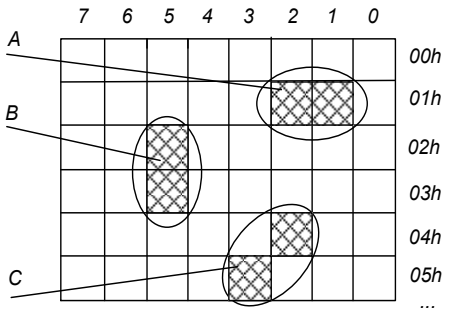

Fig. 2. Double fault options in memory.

FI in memory during exchanging between microprocessor under-test and another device via synchronous interface is another example. In this case with stopping microprocessor *t2+ t3* must be less than timing cycle. Another example is injection in external memory without microprocessor stopping. In this case injection infrastructure waits for external memory freeing after that injection is by external memory controller. If injection time *t2* be great then processor is idle waiting for injection termination.

As can be seen from the above, these examples illustrate that timing minimization in injection has high priority. Existing approaches to FI based on OCD, have overall property: it is either used external control computer [10] or in the latter case special external injection device [6, 7, 8] which connected to control computer, in other words external hardware-in-the-loop environment is fault source. This injection system development prevent from injection with minimal time and high rate, because in both cases there is time required to interaction between injector and OCD by external debug interface, leading to injection time increment.

IV. ON CHIP INJECTOR

Unlike the rest of FI methods using OCD the offered method includes FI-infrastructure is in SoC-microprocessor as specialize IP-block (Fig.3). OCD-injector independently injects and gathers statistics. It is not necessary the optional equipment except fault injector in the capacity of IP-block; FI based on standard SoC-microprocessor hardware. External control computer has support role: it used only for statistics collection of FI-campaign results.

Both the internal and the external memory are used for FI by fault-injector. For injection in internal memory the processor stopping is used, but there is minimal predictable fixed delay which defined by on-chip-bus behavior and agreed with processor cycle.

Injection in external memory is only in real time without processor stopping while external memory is not accessed by processor. Processor stopping also may be used for injection in external memory.

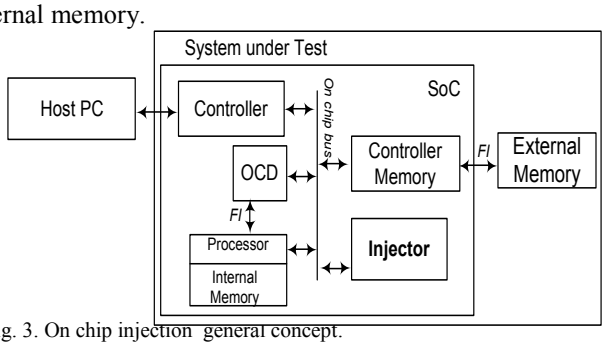

Fig. 3. On chip injection general concept.

OCD, memory controller and fault injector are on on-chip-bus (OCB). The condition must be executed for this method. It provides small connectivity between FI module and processor core. Interaction is provided only by OCB. At the end of the test process injector excluded from microprocessor without changing. So this method allows using small FI-block as will be seen later, and possibility to insert and then except FI-block after FI-experiments without opportunistic intervention in microprocessor hardware.

V. EXAMPLE OF FAULT INJECTION SYSTEM

System under test (Fig. 4) consists of SoC-processor implemented in FPGA and external memory. The external computer interfaced with system under test. SoC-processor includes processor core, fault injector, external memory controller, and external interface controller. Processor and fault injector are entirely independent. External memory controller and external interface controller provide exchange with external short memory and control computer accordingly.

Functionally fault injector includes three blocks: FI controller, fault generator, and result storage. FI controller coordinates work of fault injector components; using the information obtained from generator controller introduces faults into required space of internal or external memory; it gets information from ECC-memory status register about error detection and saved this information.

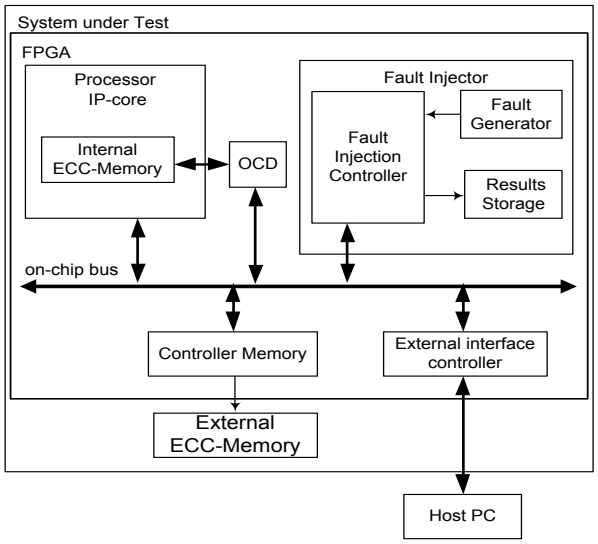

Fig. 4. FI system structure.

New fault is formed by generator using fault list for internal or external memory injection. Fault list may be generated individually by pseudorandom number generator or preloaded from control computer. The next options are defined for each injection: error type: single error (single-bit error) or double-bit error (error in two adjacent bits); error memory location; error location in 32 bit memory word; injection time in case of memory faults accidental covering.

Result storage saves experiment results. In fact it is memory unit. In simplest case the result storage contains only number of detected and fixed (or unfixed) errors. For deep analysis it is necessary to save the error memory location in processor memory. In this case the result storage is table (several tables) whose contents are given to control computer after FI-campaign.

The submitted structure allows injecting both in internal or external memory with processor stopping and in external short

memory without processor stopping. In the latter case the processor work and fault injection are parallely and independently. Injector monitors the access to external memory by inbuilt tool of SoC-processor. Then at the right time it injects the fault by external memory controller. In this case each injection should be done during the minimal time because the external memory closed for the time for processor. Proposed solution complies with the requirements because FI time delay is defined by only OCB behavior where fast data transfer transaction is provided.

Injection in internal memory without processor stopping is impossible because the intervention in processor core is required what fails to satisfy little invasiveness.

Despite FI-campaign scenario variety the big differences are defined by only FI mode: with processor stopping and without processor stopping; and fault covering method: uniform random or predefined.

Uniform random fault covering is close to natural emergence process of SEU by space radiation. Predefined testing is for shared checkout of hardware and software fault tolerance. Fault options are predefined. Faults are if program reached the specific events.

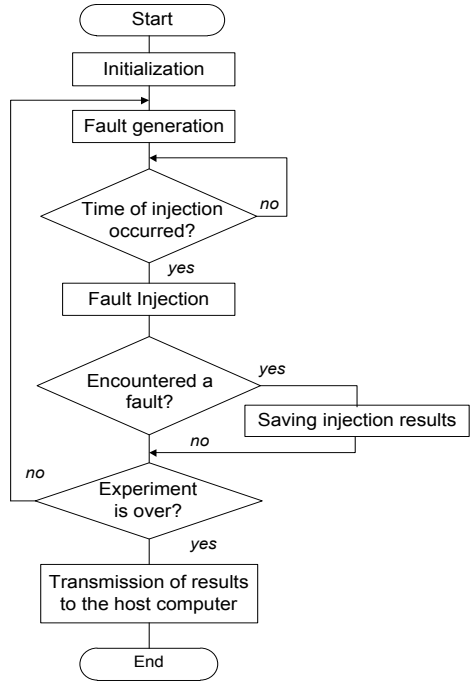

Fig.5 Algorithm of random fault covering with processor stopping.

Injection algorithm includes the following stages: initialization, generation and fault injection, memoization and logging in external computer. This sequence is saved for all developed injection methods but the content of separate stages changes.

For example during initialization depending on method the predefined fault list may be required. Processor turns in debug mode if processor stopping is predefined in testing process. In predefined mode the watchpoints are sets.

Fault generation is prepared injection. It involves the parameter definition of coming fault. It is read from preloaded fault list or it is generated automatically.

Injection for random fault covering includes prime read of memory cell what is not necessary in predefined fault covering because datum value is known before injection.

The further stages of methods are the same: memoization and logging in external computer.

In the design of FI-campaign you must consider that not every memory injected fault will be detected. Memory cell with injected fault may be rewrite by processor before the fault tolerant mechanism activation. Injection caused the fault is effective. The effective fault number depends on testing software opportunity to memory reading during injection. After experiment confirmed processor insensitivity to faults the fault injector is eliminated.

V. EXPERIMENTAL RESULTS

Proposed methods were approved in the context of SoC-processor LEON3 implemented in flash-FPGA of Microsemi A3PE3000L [11]. On-chip-bus is AMBA 2.0 [12]. The description language of processor is VHDL. It has open fault tolerant version. The using of fault tolerant code Hsiao (39, 32) [13] provides the fault tolerance of internal and external memory. It allows fixing the single error and detecting double-bit error.

Conducted research confirm the following fault injector options: operation with processor stopping and without processor stopping ; operation of automatic fault parameter generation; operation of set-up parameter reading from the predefined fault list; fault injection for all kinds of microprocessor memory; predefined mode option[14, 15].

The resource usage of FPGA logical units by on-chip fault injector is below 2%. The on-chip memory space is proportional to the planned FI number saved in fault list; and it is for 100 faults – 1,7%, for 500 faults – 2,6%, for 2000 faults – 8,0%.

The injection time phases (injection preparation and realization) ties to the processor cycle time

$$t = \frac{N_0 \pm \Delta_N}{F},$$

where N_0 – average cycle number, Δ_N – maximal deviation from the average cycle number, F – microprocessor system frequency.

The Table I gives the injection delays for microprocessor system frequency 25MHz.

TABLE I
INJECTION DELAYS

Delay type		Fault injector operations	
		with processor stopping	without processor stopping
Injection setup time *t1*	µs	0,6-1	0,12
	CPU cycles	20±5	3
Injection timing *t2*	µs	1,88-2,44	1,32-1,88
	CPU cycles	54±7	40±7

Thus, injection by means of on-chip injector is very fast. Delay float value caused by operation aspects of on-chip-bus AMBA 2.0 controller .

Suggested approach realization of on-chip injection is executed using the powerful SoC-processor LEON3. The fault mechanism verification of internal and external memory based on code Hsiao is done. Afterwards the injector was excluded from SoC. These results are the high practical relevance of the proposed method.

VI. SUMMARY AND CONCLUSION

Fast fault injection is important for some FI-campaign scenario, for example, it is multibit fault injection and injection during the synchronous data exchange between SoC-processor and external devices.

We have suggested the on-chip injection method for the fast fault injection. It is suppose that fault injection infrastructure is in SoC like an IP-block. In that case delays of injection preparation and realization defined only by SoC-processor behavior and processor cycle time.

Proposed method is high functionality. It allows injecting both the internal and the external memory with uniform random or predefined covering.

The method is little invasiveness and it has the high prospects for the fault tolerant testing of SoC-microprocessors.

REFERENCES

[1] Arlat, J. Comparison of Physical and Software-Implemented Fault Injection Techniques / J. Arlat, Y. Crouzet, J. Karlsson, P. Folkesson, E. Fuchs, G. H. Leber. // IEEE Transactions on Computers. – 2003. – no. 52. – Pp. 1115-1133.

[2] Ziade, H. A Survey on Fault Injection Techniques / H. Ziade, R. Ayoubi, R. Velazco // The International Arab Journal of Information Technology. – 2004. –№2. – vol. 1. – Pp.171-186.

[3] Elks, C. R. Development of a Fault Injection-Based Dependability Assessment Methodology for Digital I&C Systems // C. R. Elks, N. J. George, M. A. Reynolds, M. Miklo, C. Berger, S. Bingham, M. Sekhar, B. W. Johnson // NUREG/CR-7151, United States Nuclear Regulatory Commission. – vol. 1. – 2012 – 201p.

[4] R. Blahut. Theory and Practice of Error Control Codes. M.: Mir, 1986. – 576p. (In Russ.)

[5] LEON3 Processor // Aeroflex Gaisler − 2014. – Available at: http://www.gaisler.com/index.php/products/processors/leon3.

[6] Portela-Garcia, M. Fault Injection Approach for Measuring SEU Sensitivity in Complex Processors / M. Portela-Garcia, C. Lopez-Ongil, M. Garcia-Valderas, L. Entrena. A Rapid // 13th IEEE International On-Line Testing Symposium – Heraklion, Crete, Greece, 2007 – Pp.101-106.

[7] André V. Real Time Fault Injection Using Enhanced OCD – A Performance Analysis /André V. Fidalgo, Gustavo R. Alves, José M. Ferreira // Defect and Fault Tolerance in VLSI Systems, 2006. DFT'06. 21st IEEE International Symposium – 2006. – Pp. 254-264.

[8] Fidalgo, A. Real-time fault injection using enhanced on-chip debug infrastructures / A. Fidalgo, M. Gerigota, G. Alves, J. Ferreira // Microprocessors and Microsystems – 2011. – № 25. – Pp. 441-452.

[9] On Chip Debug // ASSET InterTech. − 2015. – Available at: http://www.asset-intertech.com/Technologies/On-Chip-Debug.

[10] Gaisler, J. LEON3-FT-RTAX SEU test results / J. Gaisler, // Gaisler Research – 2005 – 8p.

[11] ProASIC3L FPGAs // Microsemi − 2015. – Available at: http://www.microsemi.com/products/fpga-soc/fpga/proasic3l.

[12] AMBA Open Specification, ARM // Available at: http://www.arm.com/products/system-ip/amba/amba-open-specifications.php.

[13] Hsiao M. A class of optimal minimum odd-weight column SEC-DED codes / M. Y. Hsiao // IBM J. Res. Develop. – 1970. – vol. 14 – no. 4 – Pp. 395–401.

[14] Chekmarev S.A**.** Fault injection via on-chip debugging in the internal memory of systems-on-chip processor/ S.A. Chekmarev, V.Kh. Khanov // TIAA 2015, IOP Conf. Series: Materials Science and Engineering. – Vol. 94, 2015 – 6p.

[15] Chekmarev S.A.Modification of Fault Injection Method via On-Chip Debugging for Processor Cores of Systems-On-Chip. / S.A. Chekmarev, V.Kh Khanov, O.A. Antamoshkin // 2015 International Siberian Conference on Control and Communications (SibCon). – Russia, Omsk, 2015 – 4p.

Khanov Vladislav, Ph.D., assistant professor of information technology security, Siberian State Aerospace University named after academician M.F. Reshetnev. Research interests: space electronic instrument-making. Author and co-author of more than 80 scientific papers and study guides.

Chekmarev Sergey, an engineer, Siberian State Aerospace University. Named after academician M.F. Reshetnev. Research interests: space electronic instrument-making. Author and co-author of more than 15 scientific papers.