

DISEÑO PARA TESTABILIDAD EN TIEMPO REAL DE FALLAS TIPO “SINGLE EVENT UPSETS” EN PROCESADORES DIGITALES

Pablo A. Ferreyra, Carlos A. Marqués, Javier P. Gaspar y Ricardo T. Ferreyra

Facultad de Matemática Astronomía y Física. Universidad Nacional de Córdoba, Argentina.
e-mail: marques@mail.famaf.unc.edu.ar

ABSTRACT

A Single Event Upset, (“SEU”), is a bit flip caused by the passage of a radiation particle through the silicon area of an electronic device such as a microprocessor or memory. The characterization of a processor vulnerability to “SEU” can be achieved by means of a combination of radiation tests and fault injection techniques. Fault injection techniques can be classified in two main groups: purely software based approaches and combined hardware-software approaches. Due to the long simulation time consumed by purely software based approaches, it is strongly required the development of methods and tools to perform fault injection in real time. To improve the accuracy of the simulation results, and to simplify the required testing systems, the device under test should have a proper set of built in testability issues. In this work we present two novel methods to alter the internal architecture of a simple Central Processing Unit, (“CPU”), to allow SEU-type fault injection in real time.

RESUMEN

Un Evento de Alteración Singular, (“EAS”), es una inversión de un bit causada por el pasaje de una partícula de radiación a través del área de silicio de un dispositivo electrónico tal como un microprocesador o una memoria. La caracterización de la vulnerabilidad de un procesador con relación a los “EAS”, puede ser obtenida por medio de una combinación de ensayos de radiación y técnicas de inyección de fallas. Las técnicas de inyección de fallas pueden ser clasificadas en dos grupos principales: métodos totalmente basadas en “software” y métodos combinados de “hardware” y “software”. Debido a los largos tiempos de simulación consumidos por los métodos totalmente basados en “software”, se torna muy importante el desarrollo de métodos y herramientas para realizar inyección de fallas en tiempo real. Para incrementar la precisión de los resultados de las simulaciones, y para simplificar los sistemas de test requeridos, el dispositivo bajo prueba debe tener un conjunto apropiado de facilidades internas de “testabilidad”. En este trabajo, se presentan dos novedosos métodos para alterar la arquitectura interna de una Unidad Central de Proceso, (“UCP”), con el objeto de permitir inyección de fallas tipo “EAS” en tiempo real.

DISEÑO PARA TESTABILIDAD EN TIEMPO REAL DE FALLAS TIPO “SINGLE EVENT UPSETS” EN PROCESADORES DIGITALES

Pablo A. Ferreyra, Carlos A. Marqués, Javier P. Gaspar y Ricardo T. Ferreyra

Facultad de Matemática Astronomía y Física. Universidad Nacional de Córdoba, Argentina.
e-mail: marques@mail.famaf.unc.edu.ar

RESUMEN

Un Evento de Alteración Singular, (“EAS”), es una inversión de un bit causada por el pasaje de una partícula de radiación a través del área de silicio de un dispositivo electrónico tal como un microprocesador o una memoria. La caracterización de la vulnerabilidad de un procesador con relación a los “EAS”, puede ser obtenida por medio de una combinación de ensayos de radiación y técnicas de inyección de fallas. Las técnicas de inyección de fallas pueden ser clasificadas en dos grupos principales: métodos totalmente basadas en “software” y métodos combinados de “hardware” y “software”. Debido a los largos tiempos de simulación consumidos por los métodos totalmente basados en “software”, se torna muy importante el desarrollo de métodos y herramientas para realizar inyección de fallas en tiempo real. Para incrementar la precisión de los resultados de las simulaciones, y para simplificar los sistemas de test requeridos, el dispositivo bajo prueba debe tener un conjunto apropiado de facilidades internas de “testabilidad”. En este trabajo, se presentan dos novedosos métodos para alterar la arquitectura interna de una Unidad Central de Proceso, (“UCP”), con el objeto de permitir inyección de fallas tipo “EAS” en tiempo real.

1. INTRODUCCIÓN

Los métodos experimentales para obtener las secciones transversales subyacentes de los procesadores y memorias se basan en someter al dispositivo bajo prueba a un flujo de radiación y contar la cantidad de alteraciones producidas, [1]-[2]. La sección transversal obtenida del procesador en estudio, depende muy fuertemente de la aplicación que corre en el mismo. Cada cambio en el “software” obligaría a un nuevo conjunto de ensayos experimentales para obtener una estimación de la confiabilidad del sistema desarrollado, lo que llevaría a

costos y contratiempos inaceptables debido a la gran variabilidad del “software” durante la etapa de desarrollo e incluso durante la fase de mantenimiento y puesta a punto del sistema. Para subsanar este problema, otros autores han introducido el concepto de tasa de errores [3]. Una vez obtenida la tasa de error de una aplicación, particular, la sección transversal de la misma se obtiene multiplicando dicho parámetro por la sección transversal subyacente del procesador adquirida por métodos experimentales. La tasa de errores se consigue por medio de procesos de inyección de fallas [4]-[5]. Existen diversos métodos para inyectar fallas tipo “EAS”, que pueden clasificarse en dos grupos principales: métodos totalmente basadas en “software” y métodos combinados de “hardware” y “software”. Los métodos totalmente basados en “software”, [5]-[7] tienen mayor flexibilidad, menor costo, pero tienen el problema de los muy largos tiempos de simulación requeridos para aplicaciones extensas. Los métodos combinados de “hardware” y “software” subsanan el problema de la velocidad, pero, aparecen dos nuevos problemas. El primero es la complejidad de los sistemas de test requeridos, que pueden llegar a ser mucho más costosos que los propios sistemas bajo prueba, lo que desanima su aplicación. El segundo, es que debido a que los procesadores actuales tienen un conjunto extenso de bits internos “ocultos” no accesibles por el programador, la tasa de error obtenida puede diferir considerablemente de la real.

En este trabajo se bosquejan modificaciones a las arquitecturas tradicionales utilizadas en los procesadores sincrónicos, que permiten resolver los dos problemas anteriores.

2. ACCESO DIRECTO

Las transiciones de estado de los circuitos secuenciales internos de la “UCP” ocurren en los flancos ascendentes o descendentes de la señal de reloj del procesador. La primera solución novedosa propuesta en este trabajo para

simular "EAS" en tiempo real, consiste en permitir a un dispositivo externo, el acceso de cada registro interno de la "UCP", en los intervalos de tiempo localizados en el medio de cualquier par de flancos consecutivos de la señal de reloj del procesador. El dispositivo externo es el sistema de test. La interface entre el sistema de test y el procesador bajo estudio, consiste en un bus de direcciones, un bus de datos, un bus de control y un conjunto de señales de sincronización. Cada registro interno de la "UCP", está "mapeado" en el espacio de direcciones del dispositivo externo. El sistema de test puede apuntar a un registro particular de la "UCP", colocando la dirección del registro en el bus de direcciones de la interface. El bus de control permite seleccionar la operación deseada sobre el registro, (lectura o escritura). Los datos son transferidos desde o hacia el procesador a través del bus de datos de la interface. El dispositivo externo siempre tiene la mayor prioridad para tomar el control de los buses internos, no obstante, para inyectar una falla tipo "EAS" sólo toma el control de los mismos, cuando recibe un pulso de disparo a través del bus de sincronización de la interface. El sistema de test actúa como un controlador externo del bus interno del procesador, con un acceso directo sincronizado a través de las señales de la interface. En efecto, este método puede extenderse fácilmente para generar alteraciones en los bits por medio de accesos directos a memoria, a cualquier dispositivo de almacenamiento masivo de datos conectado al dispositivo bajo prueba. Una vez que se recibe un pulso de disparo de sincronización, los datos de un registro particular son leídos, un bit elegido al azar es modificado, y los datos modificados resultantes son reescritos nuevamente al mismo registro. La operación completa, debe "encajar" en un intervalo de tiempo muy corto, localizado entre dos flancos consecutivos de la señal de reloj del procesador. Este hecho impone severos requerimientos de velocidad al sistema de test. Otro problema principal de este primer método es que la arquitectura inicial del procesador debe sufrir cambios profundos para permitir el acceso a los registros internos desde el exterior.

3. REGISTROS "EAS"

El segundo método es una solución innovadora para resolver ambos problemas anteriores. Se basa en el reemplazo de todos los registros internos del procesador por un tipo especial de registro denominado registro "EAS". Estos registros especiales son una modificación de los registros tipo D standards normalmente utilizados como elementos de almacenamiento de los bits de los registros. Usualmente, los registros tipo D se agrupan en conjuntos de ocho elementos. Un registro "EAS" de 8

elementos de almacenamiento, (8 bits), tiene 5 pines más de entrada que un registro tipo D standard de 8 bits. Tres de estas entradas extras denominadas "selectoras" son necesarias para elegir cual de los ocho bits será alterado, una señal adicional denominada "pulso de alteración" sirve para generar la inversión del bit dentro del registro, si la última entrada adicional, denominada, señal de "habilitación de EAS", está en su estado activo. Las señales "selectoras" de bits provenientes de todos los registros, se conectan en paralelo, y son dirigidas juntas hacia la interface con el sistema de test. Todas las señales de habilitación de "EAS" son controladas por un decodificador 1 entre n cuya función es seleccionar a un solo registro "EAS" que estará activo al recibir el pulso de alteración. De esta manera, el pulso de alteración llega a todos los registros simultáneamente, pero sólo aquel registro que se encuentre seleccionado alterará el bit indicado por las señales selectoras de bits. El resto de los registros permanecerá sin cambios. Las señales de control del decodificador son llevadas hacia la interfaz del sistema de test, formando un bus de direcciones. Al igual que en el método anterior, el sistema de test puede apuntar a un registro particular de la CPU, colocando su dirección en el bus de dirección de la interfaz. Las líneas selectoras pueden considerarse como una extensión del bus de "address" que permite seleccionar el bit que será el blanco del pulso de alteración. Pero en este caso el proceso de alteración es mucho más simple que el necesario en el primer método, porque el sistema de test sólo debe emitir un pulso de alteración. Al igual que en el caso anterior, una señal de sincronización es generada por el dispositivo bajo prueba, para marcar los instantes de tiempo en los cuales las alteraciones pueden tener lugar. El segundo método es una clara mejora respecto del primero cuando se desea alterar los registros o memorias internas del procesador, pero para alterar memorias conectadas en el exterior del procesador, el primer método resulta más directo de aplicar. No obstante, si los registros de almacenamiento de las memorias se diseñan utilizando el concepto de los registros "EAS", resulta fácil adaptar el método para llevar a cabo alteraciones en cualquier dispositivo de almacenamiento masivo externo al procesador.

4. IMPLEMENTACIÓN DEL SISTEMA DE TEST

Ambos métodos son fácilmente implementables en una placa controlable por una interfaz del bus ISA de la PC. Dicha placa está basada en dos FPGA, EPF10K10LC84, de ALTERA, y una memoria RAM donde se aloja el micro código a ejecutar por el sistema bajo estudio. La PC permite "cargar" el circuito bajo prueba dentro de las "FPGA", y el micro código a la memoria RAM a través

de la interfaz con el “bus” “ISA”, (Industrial Standard Architecture). A su vez, permite ejecutar algoritmos de inyección de fallas específicos para determinar la tasa de error de la aplicación. El algoritmo que corre en la PC, tiene acceso a los datos almacenados en todos los registros y posiciones de memoria del sistema bajo prueba, de manera que la determinación del estado de falla del sistema puede llevarse a cabo por inspección de los registros y comparación de sus contenidos con una secuencia de valores patrón, indicativa del correcto funcionamiento del sistema. Debido a que todos los elementos de almacenamiento utilizados en el sistema bajo prueba, tanto en la “FPGA” como en la memoria RAM, son completamente estáticos, la misma PC genera la señal de reloj del sistema bajo prueba. De esta manera, cualquier bit puede ser alterado en cualquier instante de tiempo entre dos flancos consecutivos de la señal de reloj generada por la PC. Dentro de las FPGA se carga el circuito bajo estudio, modificado de acuerdo a alguna de las técnicas descritas en las dos secciones anteriores. El tamaño del circuito se encuentra limitado por la capacidad de las FPGA. El sistema descrito en este trabajo ha sido utilizado para caracterizar una “UCP” elemental con un pequeño programa, pero debido a la posibilidad de utilizar un mayor número de FPGA de mayor capacidad cada una, es posible afirmar que se puede adaptar para el estudio de sistemas más complejos. Es importante recalcar el bajo costo del sistema desarrollado.

5. CONCLUSIONES

Como conclusión final, se presentaron dos métodos de diseño para “testabilidad” para permitir la simulación de fallas tipo “EAS” en tiempo real. Para probar la factibilidad de dichos métodos, se utilizó una placa “esloteable” en el bus ISA de la PC basada en dos FPGA y una memoria RAM estática. El sistema se utilizó para caracterizar una “UCP” elemental, ejecutando un algoritmo muy simple. Aunque aún no se ha llevado a cabo la caracterización de sistemas y algoritmos más complejos, se ha demostrado la factibilidad de alterar la arquitectura básica de un procesador para permitir la realización de sistemas de test de muy bajo costo. Posiblemente, los diseñadores y fabricantes de microprocesadores comerciales, incluyan estos u otros métodos en un futuro próximo, para facilitar la muy compleja tarea del diseño y estimación de confiabilidad de computadoras tolerantes a fallas para el medioambiente espacial.

6. REFERENCIAS

[1] Elder J. H._A “Method for Characterizing a Microprocessor’s Vulnerability to SEU_” *IEEE*

*Transactions on Nuclear Science*_Vol. 35_No. 6, Pages 1678-1681, Dec. 1988.

[2] V. Asenek, C. Underwood, R. Velazco, S. Segui, M. Olfield, R. Velazco, Ph. Cheynet, R. Ecoffet, “SEU induced errors observed in microprocessor systems”, *IEEE Nuclear and Space Radiation Effects Conference (NSREC’98, Newport Beach, (USA)), 20-24 July 1998.*

[3] R. Velazco, S. Rezgui, H. Ziade, “Assessing the soft error rate of digital architectures devoted to operate in radiation environment: a case studied”, *2nd IEEE Latin-American Test Workshop*, , Cancun, México 11 –14 February 20001.

[4] R. Velazco, S. Rezgui, E. Reguer, “THESIC: Una plataforma flexible para la validación funcional de circuitos integrados”, *VII Workshop IBERCHIP*, Montevideo Uruguay, 21 al 23 de marzo del 2001.

[5] P. A. Ferreyra, C. A. Marqués, J. P. Gaspar and R. T. Ferreyra, “A software tool for simulating single event upsets in a digital signal processor”, *2nd IEEE Latin-American Test Workshop*, Cancun, Mexico, 11 –14 February 20001.

[6] F. Vargas, A. Amory, “Circuit Modeling and Fault Injection Approach to Predict SEU Rate and MTTF in complex circuits”, *2nd IEEE Latin American Test Workshop*, Cancun, México, Febreaury 11-14 2001.

[7] R. Velazco, A. Bragagnini, O. Calvo, “Upset-Like fault injection in VHDL descriptions: A method and preliminary results”, *2nd IEEE Latin American Test Workshop*, Cancun, México, Febreaury 11-14 2001.