

Plataforma de Simulación y Análisis para el Estudio de los Efectos de la Radiación sobre Circuitos de Comunicaciones

O. Ruano¹, J.A. Maestro¹, P. Reyes¹ y P. Reviriego²

¹Universidad Antonio de Nebrija, Calle Pirineos, 55, Madrid - oruan@nebrija.es

²Universidad Carlos III de Madrid, Avenida de la Universidad 30, Leganés

Resumen

Un factor crítico a tener en cuenta en los circuitos electrónicos que se embarcan en sistemas espaciales, es la fiabilidad. En un ambiente hostil como presenta el espacio, los efectos de la radiación causan múltiples fallos conocidos como SEE (Single Event Effects) entre los cuales caben destacar los Single Event Upsets (SEU) y los Single Event Transients (SET). Para prevenir estos fallos, la práctica más extendida es el uso de la técnica Triple Modular Redundancy (TMR), pero además de no ser 100% efectiva, es costosa en área y potencia. En este artículo se presenta una plataforma de simulación para Circuitos de Procesado de Señal y Comunicaciones cuya misión es inyectar fallos emulando el comportamiento de una fuente de radiación. La plataforma está basada en una herramienta conocida como SST que desarrolló inicialmente la ESA. Como ejemplos de uso se contraponen dos implementaciones de filtros de promediado, obteniendo como resultado qué estructura es más robusta a los SEUs.

1. Introducción

La industria aeroespacial, debido a su relativa juventud, es un sector en continuo crecimiento. Esto se pone de manifiesto con la adopción de una estrategia espacial elaborada conjuntamente por la ESA y la Comisión Europea con miras a fortalecer el sector [1]. Al amparo de programas de cooperación europea, grandes empresas del sector como Alcatel Espacio, participan en programas de satélites de comunicación, suministrando Equipos y Subsistemas de TTC, Sistemas de Tratamiento Digital de Datos y Procesado de Señal para Satélites.

Cuando estos circuitos de comunicación se someten a condiciones de trabajo extremas, su fiabilidad se ve claramente afectada [2]. Entre las principales fuentes de error que se encuentran se pueden destacar los cinturones de Van Allen, los rayos cósmicos, y las partículas de energía solar. Resulta especialmente interesante la Anomalía del Atlántico Sur producto de los cinturones magnéticos de Van Allen mencionados anteriormente. Como podemos ver en la Figura 1, la zona destacada afecta a órbitas bajas comúnmente transitadas por satélites.

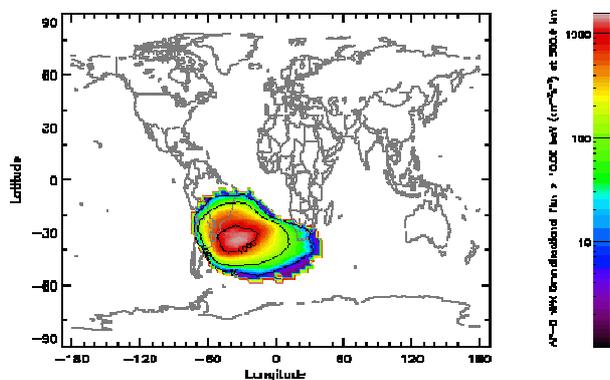


Figura 1

Este trabajo ha sido subvencionado por el Ministerio de Educación y Ciencia mediante el proyecto ESP-2006-04163 (parcialmente financiado con fondos FEDER).

De esta manera, los sistemas que se embarcan en satélites o naves espaciales presentan a menudo comportamientos erráticos debido al exceso de radiación, pudiendo causar entre otros, modificaciones en las células de memoria, efecto denominado “Single Event Upset” (SEU). Existen actualmente una serie de repositorios mantenidos por importantes agencias donde poder reflejar las incidencias ocurridas en el transcurso de las misiones y sus causas: Payload Flight Anomaly Database (PFAD), NASA GSFC’s Spacecraft Orbital Anomaly Report (SOAR), Air Force’s Orbital Data Acquisition Program (ODAP) y Space Systems Engineering Database (SSED) [3]. Los principales efectos de la radiación sobre los componentes se dividen en:

- Efecto de Dosis de Ionización Total (TID): efecto que resulta en una degradación de los parámetros funcionales del circuito como consecuencia de una exposición permanente a radiación.
- Single Event Latchup (SEL): condición que provoca la pérdida de funcionalidad del dispositivo debido a la aparición de corrientes parásitas elevadas.
- Single Event Upset (SEU): corresponde a un cambio de estado en un elemento de memoria, inducido por el impacto de una partícula energética como un protón.
- Single Event Transient (SET): efecto en el que una corriente transitoria o pico de voltaje es inducido en un circuito. Si este *glitch* es capturado por un elemento de almacenamiento da lugar a un SEU.

Las compañías expertas en microelectrónica que operan dentro de la industria aeroespacial suministran soluciones endurecidas (“radiation-hardening-by-process RHBP”) [4] que presentan una mayor resistencia frente a la radiación, gracias al uso de

materiales como el óxido de silicio (SiO) o el zafiro (SOS) que actúan como escudo. Este tipo especial de proceso industrial supone un coste excesivamente caro y además las prestaciones que aporta frente a sus homólogos comerciales son pobres en términos de potencia, velocidad y tamaño. Por ello, emerge la necesidad de buscar nuevas técnicas dentro de los procesos estándares, a la hora de obtener sistemas tolerantes a fallos, que permitan paliar los efectos de la radiación abaratando costes. Uno de los enfoques que más se está utilizando a la hora de producir dichos sistemas, es el uso indiscriminado de redundancia, como ocurre con la técnica TMR (Triple Modular Redundancy, Figura 2), y los códigos CRC, Hamming [5]. Estas aproximaciones, conllevan un incremento en área y potencia que en ocasiones son inaceptables.

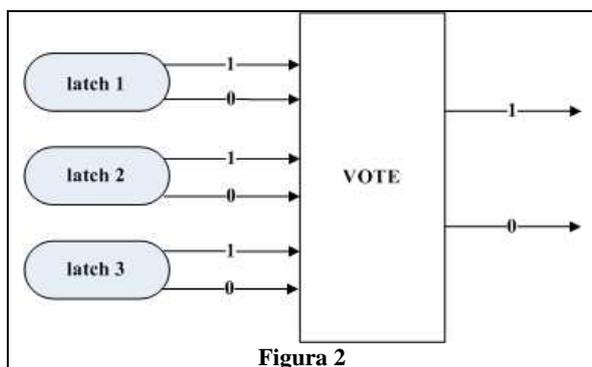


Figura 2

En esta línea de búsqueda de nuevas alternativas capaces de ofrecer un buen nivel de protección a costes bajos, se ha desarrollado una plataforma para la simulación de efectos de la radiación en entornos de diseño comerciales (Modelsim), donde poder llevar a cabo un estudio de tolerancia a fallos en sistemas digitales. Su propósito es ayudar a los diseñadores a predecir potenciales puntos débiles del diseño, evitando el uso masivo de técnicas generales de redundancia como TMR. Asimismo, puede utilizarse para evaluar la efectividad de técnicas de tolerancia a fallos.

El paper está organizado como sigue: la sección 2, resume el trabajo llevado a cabo hasta el momento dentro del área. La sección 3 explica la plataforma de inyección de fallos que se ha desarrollado; por último, como ejemplo, en la sección 4 se muestran los resultados experimentales obtenidos por la plataforma sobre dos implementaciones de filtros de promediado, que permiten al diseñador comparar la fiabilidad de ambas implementaciones en la fase de diseño.

2. Trabajos Relacionados

Dentro del campo de la simulación, se han llevado a cabo una serie de plataformas con el objetivo de evaluar los efectos de la radiación sobre circuitos electrónicos. Estos sistemas se usan para evaluar dichos circuitos durante el proceso de diseño o bien una vez que han sido manufacturados. Dichas plataformas siguen dos procedimientos distintos:

- Test de radiación real sobre circuitos.
- Test en entornos de simulación.

El modo de análisis con radiación real recibe el nombre de *Hardware Fault Injection without Contact*. Esta técnica se caracteriza porque el circuito es expuesto a radiación real. Los resultados obtenidos muestran un análisis muy preciso de los puntos débiles del diseño, a cambio de un coste muy elevado, que únicamente puede ser acometido por las grandes Agencias Espaciales. Para ello se emplea un acelerador de partículas que simula rayos cósmicos sobre el circuito. Una alternativa más barata que ha ganado aceptación en la comunidad científica ha sido la aplicación de pulsos láser que simulan los efectos de partículas energéticas [6].

La segunda propuesta consiste en hacer tests sobre un entorno de simulación que abre las puertas a una gran mayoría de grupos de investigación debido a la disminución de gastos frente a la primera aproximación. Esta metodología puede llevarse a cabo empleando un Hardware o un entorno de Software. El primer modo se conoce como *Hardware Fault Injection with Contact* y la gran mayoría de los trabajos que se apoyan en él, usan las nuevas familias de FPGAs que proporcionan características explotables para la inserción de fallos como la “partial reconfiguration” y el “readback” y herramientas del tipo JBits que trabajan con el bitstream de los dispositivos. Algunos trabajos previos en esta línea son:

- La ESA System Division, junto con la Universidad de Sevilla, ha desarrollado el sistema FT-UNSHADES [7], compuesto por una placa que alberga dos FPGA's Xilinx.
- El centro de investigación LAAS-CNRS, situado en Toulouse, ha implementado el sistema Messaline4 [8], que lleva a cabo la inyección alterando las propiedades eléctricas a través de los pines que sirven de interfaz.
- La Chalmers University of Technology, en Suecia, ha creado el sistema FIST5 [9]. Este sistema hardware, puede emplear ambos métodos de inyección de fallos indistintamente.
- La Technical University of Wien trabaja sobre una arquitectura distribuida tolerante a fallos conocida como MARS6 [10], (Maintainable Real-Time System).

La segunda aproximación hace uso de un entorno de simulación software para emular los efectos de la radiación. En este caso, el diseño se representa a través de un lenguaje de descripción hardware (HDL) como puede ser VHDL o Verilog, y los errores se inyectan a través de un testbench diseñado para este propósito vía software. En esta área se encuentran los siguientes trabajos:

- La South Florida University, junto con Honeywell Space System Inc, han desarrollado un método de redundancia para la protección de circuitos frente a SEUs [11].
- La Universidad de Texas en Austin ha creado Ferrari7 [12] (Fault and Error Automatic Real-Time Injection).

- La Agencia Espacial Europea (ESA) ha desarrollado una primera versión de una herramienta llamada SST [13]. Trabaja sobre Modelsim 5.8 y permite la inserción de SEUs y SETs de forma flexible.

La plataforma propuesta en el presente artículo está enmarcada dentro del ámbito del mecanismo de simulación basada en software.

3. Visión General de la Plataforma de Simulación

Se ha desarrollado una plataforma para la simulación de efectos de la radiación en entornos de diseño comerciales (Modelsim), donde poder llevar a cabo un estudio de tolerancia a fallos en sistemas digitales. Su propósito es ayudar a los diseñadores a predecir potenciales puntos débiles del diseño, evitando el uso masivo de técnicas generales de redundancia, como TMR, ineficientes tanto en área como en potencia. Asimismo, puede utilizarse para evaluar la efectividad de técnicas de tolerancia a fallos.

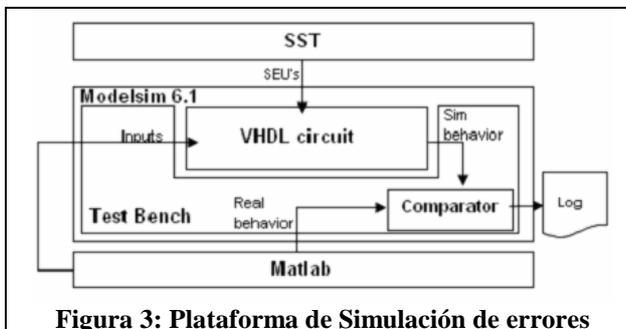


Figura 3: Plataforma de Simulación de errores

El flujo de diseño empieza con la descripción de un circuito en forma de código VHDL, un test bench y un tren de estímulos como entrada. A través del módulo SST se inyectan upsets en la simulación VHDL y se ven cuáles son los efectos sobre el comportamiento. En el último paso, se compara la simulación libre de errores obtenida desde Matlab (golden data), con la salida de Modelsim. Esta comparación muestra si los SEEs inyectados (SEUs o SETs) producen algún efecto negativo. En caso afirmativo, se pueden adoptar las siguientes alternativas para solventar el problema:

- Aplicar TMR únicamente sobre los puntos débiles del sistema.
- Seleccionar otra implementación más tolerante a SEEs. Esto usualmente conlleva un incremento de coste, que es indeseable la mayor parte de las veces.
- Diseñar técnicas de protección alternativas específicas para cada circuito que sean más efectivas y más baratas que TMR [14].

La plataforma la componen: una herramienta llamada SST usada para la inserción de errores sobre un simulador de circuitos; Matlab que se encarga tanto de implementar las funciones de test benches (a través de *link for modelsim*) como de proporcionar a la SST un flujo de errores que siga una determinada distribución de probabilidad, y por último un simulador comercial de

circuitos llamado Modelsim versión 6.1, sobre el que corre la simulación.

A continuación se explicarán los tres módulos en profundidad:

SEUs Simulation Tool (SST): La versión inicial de la SST fue desarrollada en 2004 en el Centro de Tecnología e Investigación Espacial europeo (ESTEC) y consiste en un conjunto de scripts Perl y Tcl usados para la preparación del entorno y ser capaces de producir upsets (bit flips) sobre el circuito. Perl es el lenguaje de programación principal y Tcl/Tk se usa tanto para interactuar con el simulador (Modelsim) como para el desarrollo de una interfaz gráfica (GUI). Esta primera versión utiliza Modelsim 5.8 y permite trabajar con independencia del diseño y siempre de un modo no intrusivo sobre cualquier registro o señal interna del DUT, mientras la simulación está en marcha. Actualmente nuestro grupo de investigación ha realizado modificaciones importantes sobre la SST que mejoran y extienden el software original del siguiente modo:

- La SST ha sido actualizada a la última versión del simulador de Mentor Graphics (Modelsim 6.1).
- Se ha dotado de nuevas funcionalidades a la SST, consideradas necesarias en cualquier proceso de simulación, de forma que la herramienta se presente como único punto de acceso para el diseñador a la hora de configurar sus tests. En estos test además de la inyección de fallos se contemplan otras posibilidades como checkpoints, restores y comparaciones de distintas simulaciones.

Modelsim 6.1 SE Design Environment (Mentor Graphics). Este entorno estándar, ampliamente usado en la industria, permite especificaciones de circuitos usando varios Lenguajes de Descripción Hardware, como VHDL. Este módulo es responsable de cargar el diseño bajo prueba, y ejecutar su simulación durante la etapa de diseño. La descripción del entorno se divide en dos:

- El diseño propiamente dicho, que contiene la funcionalidad del circuito.
- El Test Bench, que genera diferentes escenarios a los que someter el circuito (asentados en los valores de entrada proporcionados por Matlab), capturaré las salidas del circuito, y las comparará con los resultados esperados también suministrados por Matlab.

Matlab (MathWorks). Este módulo, prevé el comportamiento teóricamente correcto de los sistemas, que será comparado con la salida actual de Modelsim. La diferencia entre ambos comportamientos indicará la presencia de un SEU. Igualmente se ha añadido la posibilidad de inyectar campañas de SEUs conforme a una determinada distribución de probabilidades como puede ser Poisson.

Como resumen, el flujo de trabajo de la plataforma puede ser el siguiente (ver Figura 4):

1. Cargar el circuito (VHDL) habiéndose comprobado el comportamiento previamente.
2. Aplicar un tren de estímulos al circuito insertando SEEs de forma aleatoria, manualmente en un punto determinado que nos interese, o bien lanzar una campaña según cierta distribución de probabilidades.
3. Recoger el comportamiento del circuito una vez llevada a cabo la operación.
4. Reproducir el comportamiento correcto (golden) del circuito a través de Matlab.
5. Comparar la salida de 3 y 4. Si hay diferencias, recogerlas en un fichero de discrepancias para posteriormente poder analizar los fallos.
6. Repetir desde 2 todas las campañas de fallos que se deseen realizar.
7. Si el circuito presenta vulnerabilidades, estudiar las causas y efectos de las mismas para adoptar medidas de corrección como puedan ser:
 - Aplicar TMR (más costoso).
 - Usar una implementación del circuito más tolerante entre las existentes (normalmente más costosa).
 - Diseñar técnicas de protección “ad hoc” (normalmente más eficientes que otras alternativas).

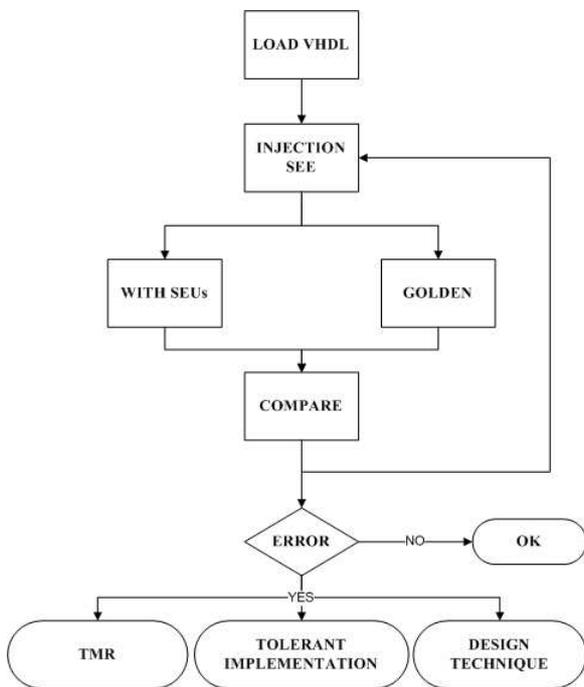


Figura 4

Las principales ventajas de la plataforma en contrapunto con trabajos basados en simulación hardware, son:

- Posibilidad de testear los circuitos en una etapa temprana del diseño sin necesidad de ningún prototipo. Es especialmente importante para el caso de ASICs donde los ciclos de diseño son largos y los prototipos caros.

- Posibilidad de inserción de upsets de manera precisa tanto en tiempo como en espacio sin grandes esfuerzos.
- Posibilidad para introducir SETs (Single Event Transients), que inviertan un bit de un cable en un nodo combinacional, lo cual en hardware resulta extremadamente difícil.

Respecto a trabajos anteriores en el campo de la simulación software presentamos las siguientes mejoras:

- Soporta el sistema Modelsim 6.1 de Mentor Graphics ampliamente usado por los diseñadores.
- Permite la fácil inserción de diferentes estímulos y modelos estadísticos de fallos dentro del circuito, permitiendo la valoración de los mismos.

4. Trabajos Experimentales

Los trabajos experimentales están orientados a valorar el comportamiento de de dos estructuras diferentes de filtros de promediado (*moving average filters*) [15] en presencia de SEUs. Un filtro de promediado implementa la siguiente ecuación:

$$y[n] = \frac{1}{N} \sum_{i=0}^{N-1} x[n-i]$$

Donde normalmente N es una potencia de 2 que permite realizar la división como una operación de desplazamiento. Las estructuras que vamos analizar, tradicional y recursiva, se muestran en la Figura 5:

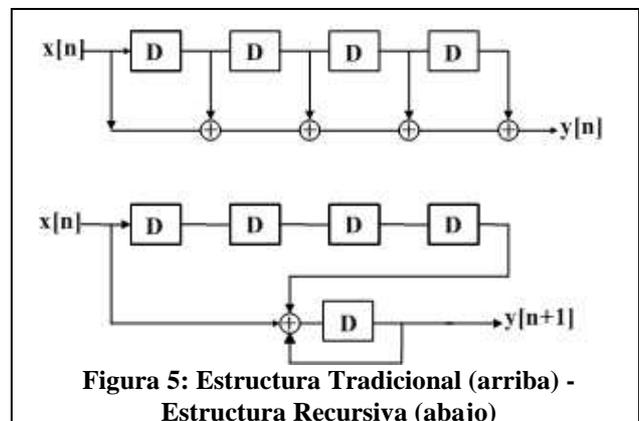


Figura 5: Estructura Tradicional (arriba) - Estructura Recursiva (abajo)

La figura de arriba presenta un esquema tradicional. A simple vista podemos predecir que la presencia de un SEU en cualquiera de sus etapas, producirá un efecto transitorio o temporal. La razón de esto radica en que con independencia de la etapa en la que se manifieste el SEU, su efecto sólo dura hasta que el contenido del flip-flop sale de la línea de retardo. La información de cada flip-flop se desplaza hacia la derecha una posición cada ciclo de reloj, lo cual hace que el efecto de un SEU esté presente como máximo una serie de ciclos iguales al número de etapas del filtro (el caso peor).

Por otro lado, la figura de abajo representa una estructura recursiva. Esta es mucho más eficiente en términos de área (menos sumadores), pero por el contrario los

posibles SEUs tienen un efecto permanente. Se puede observar que aunque la línea de retardo también se mueve a la derecha, la salida del filtro se acumula en un registro dotando al efecto generado por el SEU de un carácter permanente hasta que el sistema sufre un reset.

El primer experimento consistirá en excitar ambos filtros (tradicional y recursivo) con una señal de entrada común (Figura 6 – arriba). Si no hay SEUs presentes, la salida esperada (comportamiento) debería ser la misma en ambos casos. Esto se puede ver en la Figura 6 abajo: ambos comportamientos coinciden.

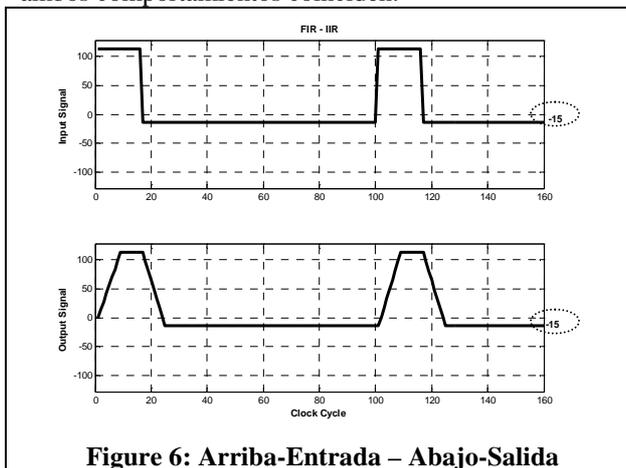


Figure 6: Arriba-Entrada – Abajo-Salida

El segundo experimento deberá aplicar la misma señal de entrada, pero esta vez insertando un SEU en el circuito (en el registro 1) en el instante 1000 ns (ciclo 10). A fin de demostrar la importancia de la posición del SEU, esta prueba será repetida dos veces:

- Primero, el SEU incide en el bit menos significativo (LSB) del registro 1.
- Segundo, el SEU afecta al bit más significativo (MSB) del registro 1.

Ambos resultados se muestran para la estructura tradicional (Finite Impulse Response, FIR o no recursiva) en la Figura 7.

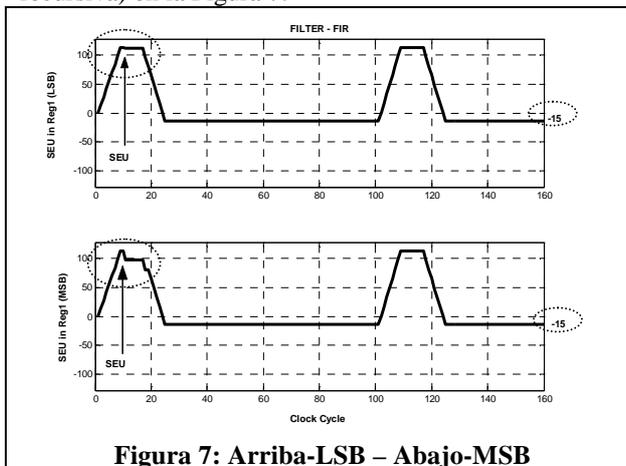


Figura 7: Arriba-LSB – Abajo-MSB

La primera conclusión es que el efecto del SEU en el LSB es insignificante: el comportamiento no se diferencia del correcto, ya que la magnitud del error es muy pequeña (Figura 7 arriba).

Sin embargo, para el caso del MSB (Figura 7 abajo), una perturbación leve se puede observar cuando el SEU impacta (alrededor del ciclo 10), e instantes después el buen funcionamiento se recupera automáticamente. De esta manera, la salida en el último ciclo de reloj alcanza el nivel correcto de -15.

Ahora la misma prueba se repite para la estructura recursiva (Infinite Impulse Response, IIR). Ambos resultados son mostrados en la Figura 8.



Figure 8: Arriba-LSB – Abajo-MSB

Otra vez, la primera conclusión es que el efecto del SEU en el LSB es insignificante (Figura 8 arriba). Sin embargo, cuando el SEU afecta el MSB, se pone de manifiesto un comportamiento errático importante. El filtro expone un comportamiento incorrecto y no se recupera en ningún momento. Esto se puede ver porque la salida en el último ciclo de reloj (Figura 8 abajo) tiene un valor de +1, cuando la salida correcta (Figura 6 abajo) tiene un valor de -15. Otro efecto ocurre durante los ciclos 110 a 117. El acumulador se desborda por el efecto del SEU y por eso el filtro muestra un comportamiento tan errático.

El último experimento ilustra el efecto que produce un SEU cuando afecta al acumulador del filtro IIR. El acumulador se muestra sensible a esta clase de riesgos, y un error allí tiene consecuencias drásticas, ya que la salida del filtro se corresponde directamente con los bits superiores del acumulador. Los resultados se ven en la Figura 9. De nuevo, un SEU en el LSB conlleva un menor impacto (Figura 9 arriba). Sin embargo, un SEU en el MSB produce un resultado notoriamente errático (Figura 9 abajo), como era de esperar.

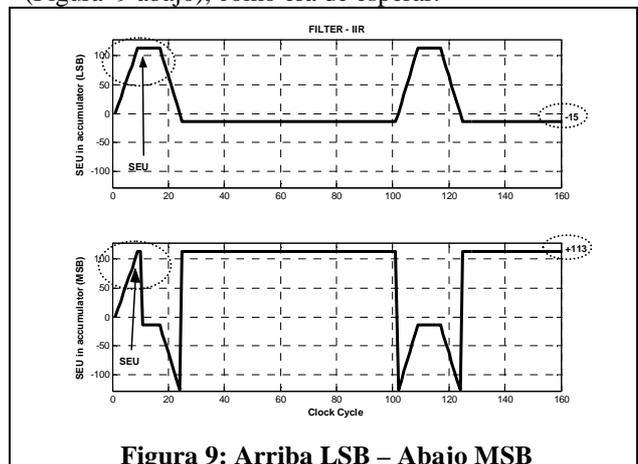


Figure 9: Arriba LSB – Abajo MSB

5. Conclusiones y Trabajo Futuro

Las ventajas que los entornos de simulación pueden ofrecer han sido analizadas usando una nueva plataforma de simulación. A través de la integración en los flujos de diseño comerciales, esta propuesta aumenta la capacidad de probar tecnologías tolerantes a fallos sin necesidad de introducir una elevada sobrecarga de tiempos y recursos. Gracias a la simulación basada en software, la plataforma se muestra como una herramienta factible para la mayoría de grupos de investigación que no cuentan con la posibilidad de acceder a hardware o equipos costosos como puedan ser las instalaciones de un acelerador de partículas, un láser o las nuevas familias de FPGAs. Prevedemos que la metodología puede mejorar el flujo de diseño en las siguientes áreas: hallazgo anticipado de errores, tiempos de desarrollo menores y capacidad de evaluación rápida para nuevas técnicas de protección. Como ejemplo de uso de la plataforma, las propiedades de tolerancia a fallos inherentes a la estructura de filtros de promediado FIR e IIR han sido probadas, determinando que la estructura tradicional FIR (aunque más costosa), es más resistente a errores que la recurrente IIR.

Como líneas de trabajo futuras, se asumen dos principalmente:

- Mejora de la plataforma de simulación, dotándola de más eficiencia tanto en rendimiento de tiempos como en nuevas funcionalidades.
- Uso de la plataforma para el estudio de otros dispositivos como pueda ser el controlador genérico 8051, además del diseño de nuevas técnicas de protección para estos circuitos.

6. Referencias

[1] 'La Industria Aeroespacial', <http://www.aero.upm.es/es/departamentos/economia/investiga/informe>.
[2] H. C. Koons et al., 'The Impact of the Space Environment on Space Systems', Aerospace Report no. TR-99(1670)-1, 20 July 1999.

[3] R. Lacoce, D. Mayer, J. Osborn and S. Brown, 'New Strategies for Radiation Hard Electronics', 2001 MRQW, December 11, 2001.
[4] P. Dodd and L. Massengill, 'Basic Mechanisms and Modeling of Single Event Upsets in Digital Microelectronics', IEEE Transactions on Nuclear Sciences, vol. 50, no. 3, June 2003.
[5] P. K. Samudrala, J. Ramos and S. Katkooi, 'Selective Triple Modular Redundancy for SEU Mitigation in FPGAs', Military and Aerospace Applications of Programmable Logic and Devices (MAPLD), September 2003.
[6] J.A. Agapito et al, 'Radiation Tests On Commercial Instrumentation Amplifiers, Analog Switches & Dacs', 7th workshop on Electronics for LHC Experiments, 2001.
[7] M. Aguirre et al., 'An FPGA based hardware emulator for the insertion and analysis of Single Event Upsets in VLSI Designs', Procs. of RADECS'04.
[8] J. Arlat et al., 'Fault Injection for Dependability Validation: A Methodology and Some Applications', IEEE Transactions on Software Engineering, vol 16 no 2, February 1990.
[9] O. Gunnetlo, J. Karlsson, and J. Tonn, 'Evaluation of Error Detection Schemes Using Fault Injection by Heavy-ion Radiation', Proc. 19th Ann. Int'l Symp. Fault-Tolerant Computing, IEEE CS Press, Los Alamitos, Calif., 1989.
[10] J. Karlsson, J. Arlat, and G. Leber, 'Application of Three Physical Fault Injection Techniques to the Experimental Assessment of the MARS Architecture', Proc. Fifth Ann. IEEE Int'l Working Conf. Dependable Computing for Critical Applications, IEEE CS Press, Los Alamitos, Calif., 1995.
[11] N. Rollins, M. Wirthlin, P. Graham and M. Caffrey, 'Evaluating TMR Techniques in the Presence of Single Event Upsets', 2003 MAPLD International Conference, September 2003.
[12] G. A. Kanawati, N. A. Kanawati, and J. A. Abraham, 'FERRARI: A Tool for the Validation of System Dependability Properties', Proc. 22nd Ann. Int'l Symp. Fault-Tolerant Computing, IEEE CS Press, Los Alamitos, Calif., 1992.
[13] D. Gonzalez-Gutierrez, 'Single Even Upset Simulation Tool Functional Description', ESA Report TEC-EDM/DCC-SST2, July 2004.
[14] P. Reyes, P. Reviriego, O. Ruano and J.A. Maestro 'Efficient Structures for the Implementation of Moving Average Filters in the Presence of SEUs using System Knowledge' Radiation Effects on Components and Systems Workshop, RADECS 2006.
[15] A. V. Oppenheim and R. W. Schaffer, 'Discrete Time Signal Processing', Prentice Hall, 1999.