

Capítulo 1

Introducción

RESUMEN: En este capítulo se realiza una introducción al trabajo realizado durante el proyecto. Se plantea el problema, se enumeran los objetivos del trabajo y se define la estructura de este documento.

1.1. Introducción

Esta monografía es el resultado del estudio e investigación realizados para la asignatura «Trabajo de fin de grado» del Grado en Ingeniería de Computadores que se ha llevado a cabo en el departamento de «Arquitectura de Computadores y Automática» de la Universidad Complutense de Madrid, bajo la dirección del Dr. Jose Miguel Montañana Aliaga.

El trabajo se centra en la implementación de un microprocesador tolerante a fallos transitorios, con un diseño que le permita ser compatible con las instrucciones ARM del repertorio del microprocesador «ARM Cortex M3» [23]. La tolerancia a fallos aplicada ha sido el «modelo de replicado triple de módulos(TMR)» [6].

1.2. Motivación

Hoy en día, el uso de la tecnología y la informática se extienden a nivel mundial, con aplicación a cada vez, un mayor número de campos. La tecnología cada vez está más presente en nuestras vidas, ya no se concibe un hogar o puesto de trabajo sin un ordenador sobre la mesa. El uso de los computadores con carácter personal cada año es más común como podemos ver en la figura 1.1. Las estadísticas, realizadas por el Instituto Nacional de Estadística (INE), muestran que en España más del 95 % de los hogares posee al menos un teléfono móvil, normalmente teléfonos inteligentes, y más

del 70 % posee un ordenador personal, lo que es un indicativo de la necesidad tecnológica que existe en estos tiempos.

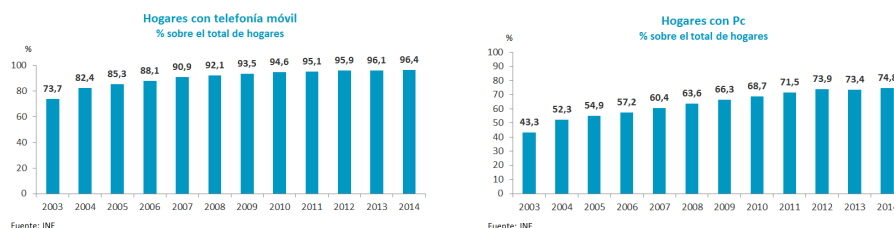


Figura 1.1: Estadísticas de uso en hogares españoles.

Todos los sistemas son susceptibles de sufrir fallos y errores a varios niveles (se explican en la sección 2.3), errores que provocan comportamientos erráticos y no deseados.

En sistemas personales como pueden ser los teléfonos inteligentes y los ordenadores personales, los fallos que sufren, normalmente no provocan consecuencias serias. Pero estos sistemas son solo una pequeña parte de los que existen y que se utilizan día a día en todo el mundo.

En los campos de medicina y transporte también hay multitud de dispositivos diferentes basados en microelectrónica. Sistemas biomédicos que asisten a la vida de una persona, que controlan un marca pasos o asisten a la respiración, si sufren un error pueden provocar consecuencias irremediables. Los sistemas de transporte controlados por sistemas electrónicos pueden sufrir los mismos tipos de fallos y causar catástrofes.

La importancia de asegurar estos dispositivos frente a posibles fallos se deriva de su control sobre sistemas vitales y por ser responsables de la vida de seres humanos. Si uno de estos sistemas falla se pone en riesgo la vida de una o muchas personas.

En el campo aeroespacial, los sistemas pueden no tener incidencia directa sobre pérdidas humanas como consecuencia de sus fallos, pero pueden causar la pérdida del sistema completo con un coste económico muy elevado. El día 7 de octubre de 2008 un avión Airbus A330-303, con origen en Singapur y destino Perth, Australia, sufrió dos descensos rápidos de 650 y 400 pies. Las autoridades australianas llegaron a la conclusión que la «air data inertial reference unit» (ADIRU) podría haber sufrido un fallo provocado por radiaciones cósmicas[[11]].

Las radiaciones cósmicas puede provocar fallos en cualquier sistema electrónico, dañando el mismo permanente o temporalmente, así, los satélites que orbitan alrededor de la tierra y los aviones que se mueven a una gran altura deben ser mucho más robustos que los sistemas que trabajan a nivel del suelo. En la figura 1.2, podemos observar que hay un mayor flujo de radiación cuanto más nos alejamos del ecuador. Esto es causado por tener una

menor protección frente a la mayor radiación y los rayos cósmicos presentes en el espacio.

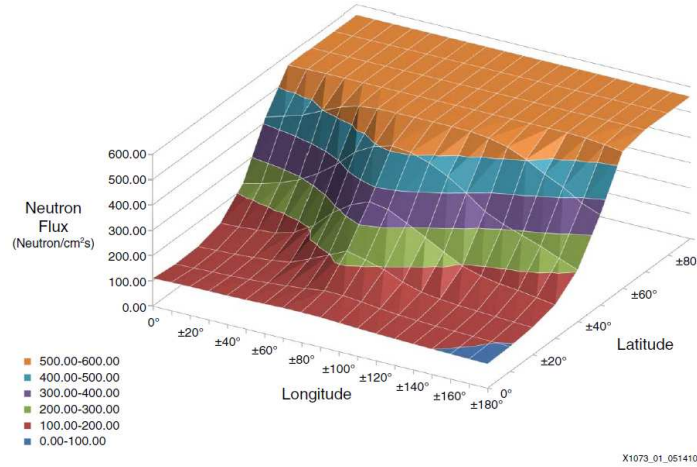


Figura 1.2: Flujo de neutrones a 40.000 pies de altitud[[9]].

Cada vez los sistemas electrónicos contienen transistores de menor tamaño, lo que disminuye su tolerancia a los «ruidos de transmisión» haciéndolos más sensibles a la radiación y a los rayos cósmicos que inducen fallos a los sistemas.

Lo que tienen en común los dispositivos mencionados, y muchos otros presentes en nuestra vida cotidiana, es que son sistemas con componentes micro-electrónicos. Poseen un microprocesador que es el cerebro y responsable de dirigir el sistema ejecutando los programas y como tal sistema electrónico, susceptible de sufrir fallos. Fallo que en este componente puede causar graves consecuencias en el resto del sistema.

Para garantizar el correcto funcionamiento de estos sistemas existen múltiples técnicas de tolerancia a fallos, técnicas que ayudan a detectar los fallos y a recuperar el sistema antes de que causen errores. Aplicando una o varias de estas técnicas se ofrece un sistema robusto y capaz de funcionar ante los fallos inducidos por la radiación u otros agentes externos.

1.3. Planteamiento del problema

Este trabajo tiene en cuenta que el motor principal de muchos sistemas es el microprocesador. El procesador de un sistema es su *cerebro*, más concretamente es el encargado de ejecutar las instrucciones que componen los programas.

Si no se toman medidas de prevención y tolerancia, este *cerebro* puede ver alterado su comportamiento por efectos externos, provocando errores de ejecución del programa. Estos errores a su vez pueden ser causa de un com-

portamiento no deseado alterando los datos, modificando el funcionamiento del propio procesador o de otros componentes del sistema como las memorias o los controladores entrada/salida.

Con este trabajo se pretende ofrecer un medio para evitar estas situaciones concediendo un grado extra de fiabilidad a los sistemas basados en microprocesadores. Para ello se quiere diseñar e implementar un microprocesador sencillo capaz de ejecutar un conjunto reducido de instrucciones (RISC), al que posteriormente se le aplicarán las técnicas de tolerancia a fallos, aumentando así su supervivencia mediante un incremento en su capacidad de detectar e incluso recuperarse de los fallos.

1.4. Objetivos

Por lo comentado en el apartado anterior se ha decidido que este trabajo esté dedicado a diseñar un procesador con un grado de fiabilidad mayor que un procesador convencional.

El proyecto se ha dividido en cuatro tareas dedicadas a la implementación del microprocesador y a la aplicación de la tolerancia a fallos.

1 Primera. Implementación del procesador.

Se ha implementado el procesador segmentado en 5 etapas. Para ello se ha partido de la arquitectura DLX vista en las asignaturas de computadores de nuestro grado.

2 Segunda. Ruta de control

Se ha rediseñado la ruta de control y parte de la ruta de datos. El nuevo juego de instrucciones usado, completamente distinto al que utiliza un DLX convencional, obliga a cambiar la ruta de control. Simulando unos pequeños programas se comprueba que el procesador es capaz de decodificar y ejecutar las nuevas instrucciones.

3 Tercera. Diseño de tolerancia a fallos.

Una vez implementado el procesador completo y comprobado su funcionamiento se diseña y se incorpora la tolerancia a fallos. Para ello se **triplican** los módulos que pueden causar mayor número de fallos y se insertan **votadores** de mayoría.

4 Cuarta. Diseño del sistema de inserción de fallos.

Para finalizar se ha diseñado un sistema externo de inserción de fallos. Este sistema es capaz de alterar los valores de las salidas de los módulos triplicados, para comprobar después como afecta esto al funcionamiento del procesador.

1.5. Estructura del documento

Capítulo 1 *Introducción*:

En el capítulo 1 se realiza la introducción al trabajo propuesto y realizado para el trabajo de fin de grado en el que se basa el presente documento.

Capítulo ?? *Trabajo relacionado*:

En el capítulo ?? se realiza una exposición de la investigación llevada a cabo para realizar el trabajo. Se exponen definiciones y otros datos de interés para el lector.

Capítulo ?? *Procesador*:

En el capítulo ?? se describe el procesador implementado con su estructura y arquitectura.

Capítulo ?? *Proporcionando tolerancia a fallos transitorios*:

En el capítulo ?? se describe cómo se ha proporcionado la tolerancia a fallos y qué técnicas se han utilizado.

Capítulo ?? *Resultados*:

En el capítulo ?? se muestran los resultados obtenidos de las simulaciones realizadas.

Capítulo ?? *Análisis de los resultados*:

En el capítulo ?? se analizan los datos.

Capítulo ?? *Conclusiones*:

En el capítulo ?? se describen las conclusiones tras analizar los resultados.

Bibliografía

*Y así, del mucho leer y del poco dormir,
se le secó el cerebro de manera que vino
a perder el juicio.*

Miguel de Cervantes Saavedra

- [1] V. Arnau Llombart. Manual DLX.
- [2] R. Brinkgreve, W. Swolfs, and E. Engin. *ARM Architecture Reference Manual Thumb-2 Supplement*. 2011.
- [3] C. T. Bustillos. Simulador arm en el ámbito docente. 2012.
- [4] J. Gaisler. A portable and fault-tolerant microprocessor based on the SPARC V8 architecture. *Proceedings of the 2002 International Conference on Dependable Systems and Networks*, pages 409–415, 2002.
- [5] J. C. González Salas. *Filtro adaptativo tolerante a fallos*. PhD thesis, 2014.
- [6] S. Habinc. Functional Triple Modular Redundancy (FTMR). *Design and Assessment Report, Gaisler Research*, pages 1–56, 2002.
- [7] J. L. Hennessy and D. A. Patterson. *Arquitectura de Computadores: Un enfoque cuantitativo*. Mcgraw Hill Editorial, 1993.
- [8] J. L. Hennessy and D. a. Patterson. *Computer Architecture, Fourth Edition: A Quantitative Approach*. Number 0. 2006.
- [9] A. C. Hu and S. Zain. NSEU Mitigation in Avionics Applications. 1073:1–12, 2010.
- [10] O. Ieee-std. LEON3 7-Stage Integer Pipeline. (March), 2010.
- [11] A. O. Investigation. ATSB TRANSPORT SAFETY REPORT Aviation Occurrence Investigation AO-2008-070 Final. (October), 2008.

- [12] Jedec. Measurement and Reporting of Alpha Particle and Terrestrial Cosmic Ray Induced Soft Error in Semiconductor Devices: JESD89A. *JEDEC Solid State Technology Association*, pages 1–85, 2006.
- [13] A. Kadav, M. J. Renzelmann, and M. M. Swift. Fine-grained fault tolerance using device checkpoints. *Proceedings of the eighteenth international conference on Architectural support for programming languages and operating systems - ASPLOS '13*, page 473, 2013.
- [14] H. Kirmann. Fault Tolerant Computing in Industrial Automation. *Lecture notes ABB Corporate ResearchETH*, 2005.
- [15] I. Kuon, R. Tessier, and J. Rose. FPGA Architecture: Survey and Challenges. *Foundations and Trends® in Electronic Design Automation*, 2(2):135–253, 2007.
- [16] A. R. M. Limited. ARM7TDMI-S. (Rev 3), 2000.
- [17] a. R. M. Limited. ARM Architecture Reference Manual. pages 1–1138, 2007.
- [18] W. K. Melis. *Reconstruction of High-energy Neutrino-induced Particle Showers in KM3NeT*. PhD thesis, 2014.
- [19] C. Mobile. Streaming 4K Ultra HD video at home and on the go. pages 0–1.
- [20] J. M. Pascual. *Simulador DLX con repertorio multimedia*. PhD thesis, Universidad Complutense de Madrid, 2011.
- [21] J. Rose, A. E. Gamal, and A. Sangiovanni-Vincentelli. Architecture of Field-Programmable Gate Arrays.
- [22] E. Rotenberg. AR-SMT: a microarchitectural approach to fault tolerance in microprocessors. *Digest of Papers. Twenty-Ninth Annual International Symposium on Fault-Tolerant Computing (Cat. No.99CB36352)*, 1999.
- [23] S. Sadasivan. An introduction to the arm cortex-m3 processor, 2006.
- [24] D. J. Sorin and S. Ozev. Fault Tolerant Microprocessors for Space Missions. *Memory*, pages 1–4.
- [25] U. States. Reduce Cost and Board Space. 374:1–8, 2011.
- [26] I. S. Summary, T. C. Field, M. Long, S. D. Transfer, U. Instruction, and I. S. Examples. ARM Instruction Set. pages 1–60.
- [27] J. M. Torrecillas. RAID - Tolerancia a Fallos.

-
- [28] C. Weaver and T. Austin. A fault tolerant approach to microprocessor design. *Proceedings of the International Conference on Dependable Systems and Networks*, (July):411–420, 2001.