ANÁLISIS DE UN COMPUTADOR EMBARCADO PARA SISTEMAS ESPACIALES BASADO EN ARQUITECTURA COLABORATIVA

por

Alberto Martín-Ortega Rico

Co-director: Dr. Sergio López Buedo

Co-directora: Dra. Marta Portela García

Doctorado en Ingeniería Informática y Telecomunicaciones

ESCUELA POLITÉCNICA SUPERIOR

Marzo 2019

Resumen

La industria aeroespacial está en plena ebullición. La última década ha supuesto un cambio en la filosofía de acercamiento al espacio. Donde antes sólo existía inversión pública salvo en muy contadas ocasiones, ahora se abre un mundo nuevo de posibilidades para el capital privado. Para ello, han confluido una serie de factores en el tiempo, que han facilitado esta nueva era. Por un lado, los avances tecnológicos y sociales han abierto un nicho para el denominado "new space". Empresas privadas de todo el mundo emergen en la escena aeroespacial, creando sistemas más baratos y de tiempos de desarrollo más cortos, movidas por un fin comercial más allá del puramente político. Este tipo de desarrollos abre la puerta a nuevas técnicas y tecnologías que permitan mejorar las capacidades y prestaciones de sistemas de procesado, sin mermar la fiabilidad de los mismos. Por otro lado, la creación de una plataforma estándar de pequeños satélites llamada CubeSat, ha facilitado el acceso frecuente y asequible al espacio con oportunidades de lanzamiento disponibles en la mayoría de los vehículos de lanzamiento. El número de lanzamientos por año se ha visto incrementado de forma exponencial. En 2005 se lanzaron un total de 9 CubeSats en todo el mundo. Esta cifra creció hasta los 63 en 2009. En el último año, 2018, se contabilizaron 1027 CubeSats.

Con esta realidad como punto de partida, las misiones espaciales se enfrentan a un implacable aumento de los requisitos en referencia a los computadores de vuelo. Se necesitan capacidades de computación más altas, mientras que el consumo de energía, la masa y el área deben reducirse. Desafortunadamente, los requisitos evolucionan más rápido que la capacidad de los fabricantes para desarrollar mejores procesadores calificados para espacio, por lo que se necesitan técnicas que permitan a los diseñadores usar componentes *COTS*, sin dejar de cumplir con las exigencias de fiabilidad marcadas por el entorno espacial.

Este trabajo de tesis doctoral presenta el endurecimiento colaborativo como una técnica poderosa y eficiente para garantizar la confiabilidad de las tareas críticas de un satélite. Además, los estrictos requisitos de fiabilidad de las misiones espaciales exigen una validación completa en tierra de cualquier diseño que use componentes *COTS* antes de su puesta en órbita. En este trabajo se presenta así mismo una metodología para la

evaluación temprana de la sensibilidad a SEU, basada en la inyección de fallos a través de un sistema de emulación autónomo. La emulación se realiza a nivel de sistema, no a nivel de unidad, para validar las técnicas de endurecimiento colaborativo implementadas en la arquitectura distribuida. Por último, se presenta un completo análisis de los datos obtenidos en vuelo durante los 3 años de misión del computador. Se describen los procesos y las herramientas utilizadas para analizar dichos datos y se expone claramente los errores funcionales encontrados a nivel de unidad, como resultado de la radiación ionizante.

Los resultados experimentales tanto de la validación en tierra como del análisis de los 3 años de misión muestran que, si bien las unidades individuales son vulnerables a los efectos de la radiación, la fiabilidad del sistema en su conjunto no se ve comprometida.

Abstract

The aerospace industry is undergoing a profound transformation. The last decade has supposed a change in the philosophy of approaching space. Where once there was only public investment except on very few occasions, now a new world of possibilities for private capital opens up. For this, a series of factors have come together in time, which have facilitated this new era. On the one hand, technological and social advances have opened a niche for the so-called "new space". Private companies from all over the world emerge in the aerospace scene, creating cheaper systems and shortening development times, moved by a commercial purpose beyond the purely political one. This type of development opens the door to new techniques and technologies that allow improving the capabilities and performance of processing systems, without reducing their reliability. On the other hand, the creation of a standard platform of small satellites called CubeSat, has facilitated frequent and affordable access to space with launch opportunities available in most launch vehicles. The number of launches per year has increased exponentially. In 2005 a total of 9 CubeSats were launched around the world. This figure grew to 63 in 2009. In the last year, 2018, 1027 CubeSats launches were counted.

With this reality as a starting point, space missions are facing a relentless increase in requirements in reference to flight computers. Higher computing capabilities are needed, while energy consumption, mass and area must be reduced. Unfortunately, requirements evolve faster than the ability of manufacturers to develop better qualified processors for space. Therefore, new techniques are needed that allow designers to use COTS components, while still meeting the reliability requirements set by the space environment.

This doctoral thesis work presents the collaborative hardening as a powerful and efficient technique to guarantee the reliability of the critical tasks of a satellite. In addition, the strict reliability requirements of space missions require full validation on the ground of any design using COTS components before they are placed in orbit. In this paper, a new methodology for the early evaluation of the sensitivity to SEU, based on the injection of faults through an autonomous emulation system is presented. The emulation is done at the system level, not at unit level, to validate the collaborative hardening techniques

implemented in the distributed architecture. Finally, a complete analysis of the data obtained in flight during the 3 years of mission is presented. The processes and tools used to analyse these data are described and the functional errors found at unit level are clearly exposed as a result of ionizing radiation.

The experimental results of both ground validation and 3-year mission analysis show that, while individual units are vulnerable to the effects of radiation, the reliability of the system as a whole is not compromised.

A Mara,

quien siempre creyó en mí y tanto me ha enseñado.

Agradecimientos

Esta es la parte sin duda más recompensante de la tesis. Llevo mucho tiempo pensando en a cuanta gente quería agradecer todo su apoyo, y sois muchos/as los que lo habéis hecho. También es verdad que largo ha sido el camino, y definitivamente nunca lo he recorrido solo. Esta tesis doctoral no es únicamente el resultado de la investigación de estos últimos años, pues empezó al terminar la carrera y empezar los *cursos de Doctorado*. En todo este tiempo, tanto familia, como amigos/as y compañeros/as de trabajo me habéis acompañado y enseñado tanto, que sinceramente pienso que esta tesis no es mía, sino nuestra.

En primer lugar, gracias Mara. Gracias por el apoyo incondicional durante todos estos años, gracias por la infinidad de cosas que has hecho por mi para que yo pudiera realizar este sueño. Tantas conversaciones, tanto sufrimientos y alegrías compartidas … sin ti, nada de esto sería posible. Gracias Mamá, gracias Papá. No lo digo tanto por el apoyo recibido durante estos años, que ha sido infinito, ni por los buenos consejos, que también han sido muchos. Lo digo porque consigo lo que consigo por todo lo que habéis dado por mí. El amor, la seguridad, la educación que siempre me habéis dado y seguís dando, han hecho de mi quien soy. Al resto mi familia y en especial a vosotros Edu y Nuria, muchas gracias por estar siempre ahí, escuchando y aguantando mis charlas *suigéneris* de satélites y partículas.

Los inicios siempre son duros, por lo menos para mí, Tú, Celia, impulsaste mi vena investigadora cuando nadie antes lo había hecho. Me hiciste ver las contribuciones que ni yo mismo era capaz de ver, y tiraste de este pesado carro hasta ponerlo en marcha, sin posibilidad de retorno. Gracias por todo ello y, sobre todo, por tu eterna sonrisa que alivia hasta en los momentos más oscuros. Marta, ¡que mente más clara tienes! Cuantas veces me has guiado por la senda correcta, convenciéndome de la forma más amable posible de lo equivocada que era mía. Sin tus consejos esta investigación no habría sido ni la sombra de lo que es ahora, y todo ello siempre acompañado de tu sincera humildad. Quince años hace desde que empecé mi andadura por el mundo de los PLD contigo, Sergio. Me diste mi primer trabajo, me acompañaste en el segundo, y me pusiste en bandeja el tercero. Durante todo este tiempo, siempre has creído en mí, me ha animado,

guiado y ayudado siempre que lo he necesitado, y sin duda sin ti, nunca habría llegado hasta aquí.

Esta investigación se ha nutrido de infinidad de diseños y desarrollos de sistemas espaciales, que han sido creados por vosotros/as, mis compañeros/as y amigos/as del INTA. Sois tantos/as los que me habéis ayudado y enseñado durante estos trece años, que me va a ser imposible nombraros a todos/as. Pero merecéis el esfuerzo de intentarlo. Gracias Santi. Tú eres sin duda el principal responsable de esta tesis. Gracias por dejarme trabajar a tu lado. Gracias por enseñarme prácticamente todo lo que sé de este curioso e infinito mundo que es el espacio. Gracias por aguantar mis inacabables preguntas y por supuesto, por ponerme en la senda de los computadores basados en COTS. Eso sin duda, no es mi mérito, sino el tuyo. José Ramón, cuantas discusiones, cuantas risas, cuantas clases magistrales de electrónica hemos compartido. Ninguna de todas ellas en valde. Gracias por compartir conmigo tanto conocimiento. Gracias Ig, que maravilloso ejemplo de capacidad de trabajo, confianza, aprendizaje constante, y todo adrezado con el buen rollo y la diversión con lo que hacemos. Qué felicidad el haber podido compartir contigo todos estos años de trabajo, y lo que no es trabajo, también. Alix, gracias por tu amor, por tu apoyo incondicional y por ese hombro donde mis penas han tenido cabida durante todo este tiempo. Por ser siempre la primera voluntaria en ayudarme, mil gracias. Nuria, llevamos sólo 4 años compartidos, pero tengo muchas ganas de devolverte esta frase que un buen amigo me dedicó una vez y tanto me gustó: es difícil encontrar alguien que haga tan cómodo el trabajo. ¡Me encanta trabajar contigo! Víctor, cuantos cafés y cervezas hemos compartido con el tema tesis. Siempre me has animado y ayudado, y en este último año y medio, más si cabe. Gracias Vic, intentaré poder compensar al menos un poco de todo tú esfuerzo. Joaquín, Javí, que habría sido de OPTOS de sin vuestros fantásticos Buhitos, que tan buenos resultados han dado y tanto han ayudado a este computador distribuido. Hector, seguramente ya lo sepas, pero eres la persona con más capacidad de inspirar que haya conocido. A mi desde luego lo has hecho en infinidad de ocasiones. Muchas gracias por ello. Gracias Elisa, cuanta informática he aprendido de ti. Tu seriedad, tu rigor, y tu buen hacer me han inspirado en infinidad de ocasiones. Amaya y Fany, gracias por vuestro buen humor y los buenos momentos compartidos.

Por último, no quiero dejar de agradeceros enormemente a vosotros/as mis amigos/as cercanos/as, que durante todo este tiempo habéis permanecido a mi lado, apoyándome y queriéndome. Hugo, Kike, Barrio, David, Mon, Dani, Nacho, Almu, Estheruki, Pantu, De Peter y un largo etcétera … ¡muchísimas gracias!

A todas y todos vosotros, ¡muchas gracias de corazón!

Tabla de Contenido

Lista de Figuras

Lista de Tablas

1 Introducción

Las misiones espaciales han evolucionado de manera incesante durante la última década debido en gran parte a dos fenómenos en auge.

El primero de ellos es el surgir del denominado "new space" ['NewSpace' 2019]. La aparición a nivel global de un movimiento o filosofía donde empresas privadas de todo el mundo emergen en la escena aeroespacial, creando sistemas más baratos y de tiempos de desarrollo corto, movidas por un fin comercial más allá del puramente político o social. Este tipo de desarrollos abre la puerta a nuevas técnicas y tecnologías donde la balanza entre capacidades y fiabilidad se ve claramente desnivelada hacia la primera. La industria aeroespacial clásica es muy conservadora en lo que respecta al uso de nuevos dispositivos y tecnologías, debido principalmente, al factor de riesgo que esto conlleva. Sin embargo, esta nueva era del "new space" abre las puertas al uso de dispositivos comerciales antes vetados en los sistemas espaciales, siempre y cuando se puedan aplicar técnicas de endurecimiento que garanticen la supervivencia y operabilidad de estos. A cambio, el uso de estos componentes, permiten mejorar las capacidades de los sistemas espaciales y acortar el tiempo de desarrollo de estos.

El segundo fenómeno es la creación de una plataforma estándar de pequeños satélites llamada CubeSat ['CubeSat Standard' 1999]. El estándar CubeSat fue creado por la Universidad Politécnica Estatal de California, San Luis Obispo y el Laboratorio de Desarrollo de Sistemas Espaciales de la Universidad de Stanford en 1999 para facilitar el acceso al espacio para estudiantes universitarios. Desde entonces, el estándar ha sido adoptado por cientos de organizaciones en todo el mundo. Los desarrolladores de CubeSat incluyen no sólo universidades e instituciones educativas, sino también empresas privadas y organizaciones gubernamentales. El estándar CubeSat ha facilitado el acceso frecuente y asequible al espacio con oportunidades de lanzamiento disponibles en la mayoría de los vehículos de lanzamiento. La mayoría de lanzadores se han adaptado para poder ser portadores de un gran número de CubeSats, ya que con ello se garantizan un buen número de pasajeros secundarios en cada lanzamiento, mejorando la ratio de beneficio de estos. De hecho, ha habido empresas como Firefly Space Systems, Rocket Lab USA y Virgin Galactic que han desarrollado con ayuda de la NASA, lanzadores

específicos para pequeños satélites tipo CubeSat. Estos lanzadores están pensados para costar menos de 10 M$, que contrasta con los 90 M$ que cuesta el lanzador PROTON-M Ruso, cuyo precio de puesta en una órbita baja (*LEO*), era el más bajo hasta ahora.

Los computadores para aplicaciones aeroespaciales han evolucionado lentamente en comparación con las aplicaciones terrestres, donde los entornos hostiles están restringidos a pocos campos. La radiación ionizante junto con condiciones extremas de temperatura, presión y vibración hacen de la robustez un requisito obligatorio para cada dispositivo electrónico. Los diseños de computadores para aplicaciones aeroespaciales, compuestos por componentes *RadHard* calificados por el fabricante del componente, aumentan notablemente el coste del proyecto. Estos dispositivos han sufrido un proceso de calificación estricto, que debe actualizarse periódicamente, lo que aumenta en gran medida el coste y complica su compra [Halain et al. 2014, Ott, Jin, Chuska and LaRocca 2006, Smith and de la Torre 2006]. Además, para las primeras versiones de la arquitectura del computador, Elegant Bread Board (*EBB*) o Engineering Model (*EM*), los dispositivos comerciales equivalentes son difíciles de obtener, teniendo que recurrir en muchas ocasiones a otros dispositivos comerciales parecidos (pero no iguales), o a tener que las piezas de vuelo a muy alto precio.

Por otro lado, las misiones espaciales actuales, como se ha explicado anteriormente, exigen capacidades de procesamiento más avanzadas. Así mismo, se unen las necesidades de otros tipos de misiones como por ejemplo las de exploración planetaria. Estas misiones requieren de unidades de control con funcionalidades más flexibles, inteligentes y autónomas, sin olvidar la necesidad de miniaturizar dichas unidades con el objeto de poder posarlas de forma exitosa sobre las superficies de los planetas.

En este sentido, son necesarias nuevas propuestas para actualizar y modernizar el diseño y la creación de computadores para las aplicaciones espaciales actuales. Estas nuevas arquitecturas han de permitir la reducción de costes al tiempo que mejoran las capacidades generales del sistema. Moviendo la complejidad desde el fabricante hasta el diseñador computadores, abrirá el mercado a fabricantes no calificados para el espacio. Esto acortará el desarrollo del producto y reducirá el coste. Estos factores han dado un impulso enorme a la creación de sistemas miniaturizados, de altas prestaciones, basados

en componentes *COTS*. Pero de nada sirven estos sistemas si no podemos garantizar su supervivencia en entornos extremos, donde los efectos de la radiación pueden menoscabar las capacidades de los mismo, o incluso llegar a destruirlos.

1.1 Objetivos

Esta tesis doctoral ha tenido como objetivo fundamental el de estudiar la viabilidad y fiabilidad de un computador de vuelo basado en conceptos de endurecimiento colaborativo, donde la fiabilidad viene dada por el endurecimiento colaborativo de los elementos del sistema, y no por la fiabilidad de cada uno de dichos elementos; y donde se consigan beneficios en términos económicos y de capacidad computacional. Para lograr este objetivo principal, se han planteado una serie de objetivos complementarios que han sido abordados para la satisfactoria consecución del estudio. Estos objetivos son:

- Se identifican las principales y más importantes debilidades de los sistemas espaciales basados en COTS. Los problemas planteados anteriormente sobre la utilización de COTS en sistemas espaciales han sido correctamente identificados en la aplicación específica de un computador de vuelo.
- Se han diseñado e implementado algoritmos complejos tolerantes a fallos, capaces de mitigar los puntos únicos de fallo presentados por arquitecturas singulares no colaborativas. Dichos algoritmos han de garantizar la correcta identificación y mitigación de fallos simples, permitiendo al sistema tolerar esos fallos sin perjudicar al desempeño de las tareas críticas del computador.
- Se evalúa de forma teórica la correcta implementación del sistema colaborativo, para un entorno concreto de trabajo del computador en cuestión. Esta modelización del mismo ha permitido realizar tareas de validación en tierra, en las fases tempranas del proyecto.

- Se verifican los datos obtenidos en vuelo, durante la ejecución del computador en un entorno de radiación real, con los obtenidos mediante simulación y o emulación. Esto ha permitido validar el modelo utilizado y el funcionamiento del sistema final.

1.2 Distribución de la Tesis

El documento de esta tesis doctoral está dividido en siete capítulos. Los tres primeros capítulos introducen los conceptos generales relacionados con la situación actual de los computadores de vuelo embarcados para espacio, además de describir las dificultades y características propias de estos y de su entorno. Además, se hace una revisión completa de las técnicas utilizadas para implementar la tolerancia a fallos y se presenta el estado de la técnica con una revisión de las técnicas de inyección existentes hasta el momento en la literatura. Los capítulos cuarto, quinto y sexto describen las aportaciones originales propuestas en esta tesis doctoral. En el cuarto se describe minuciosamente la arquitectura distribuida propuesta para la implementación de un computador de bajo coste y fiable mediante técnicas de endurecimiento colaborativo. Los capítulos quinto y sexto describen los procesos, herramientas y resultados de la validación en tierra y la verificación en vuelo del computador respectivamente. Finalmente, en el capítulo séptimo se exponen las principales contribuciones de esta tesis doctoral, así como las propuestas de líneas futuras de trabajo a seguir.

En el capítulo 2 Entorno Espacial se presentan las características específicas del entorno y las misiones donde han de funcionar los computadores de vuelo. En este capítulo se da una visión general de las condiciones que tienen que soportar, por las cuales se hace necesario la aplicación de técnicas concretas que impriman la fiabilidad necesaria a dichos computadores.

El Estado de la Técnica es una revisión actual de las técnicas de endurecimiento aplicadas a computadores, tanto a nivel unidad como a nivel sistema, con diferentes enfoques:

hardware, software o híbridos. Además, se muestran las técnicas más avanzadas descritas en la literatura para la evaluación de sistemas endurecidos.

El capítulo 4 de Computador basado en Endurecimiento Colaborativo, desgrana los condicionantes previos para el diseño de la arquitectura final, así como la propia arquitectura del computador colaborativo diseñado e implementado para el satélite OPTOS. En este capítulo se presenta la funcionalidad del computador y las técnicas y algoritmos colaborativos implementados para poder realizar las tareas críticas de este de una manera fiable. Por último, se realiza un estudio de radiación del computador en el entorno real del satélite, con el objetivo de conocer a priori la tasa de fallos esperada del mismo durante la misión.

El capítulo 5 de Validación en Tierra, presenta el ecosistema de pruebas creado para poder hacer una validación temprana del computador. La validación se realiza mediante un sistema autónomo de emulación de fallos provocados por radiación, adaptado para ser capaz de realizar dicha emulación a un sistema distribuido. Además, se presentan y justifican los resultados obtenidos durante dicha validación.

El capítulo 6 de Verificación en Vuelo cierra el círculo describiendo los resultados experimentales de los 3 años de misión del computador a bordo del satélite OPTOS. Para ello se explican la filosofía utilizada para la generación de datos en vuelo, que una vez en Tierra son analizados y procesados por dos herramientas diseñadas a tal efecto. Finalmente se realiza un análisis y justificación exhaustivas de los resultados obtenidos.

Para concluir, en el capítulo 7 de Conclusiones se exponen las principales contribuciones de esta tesis doctoral, así como las propuestas de líneas futuras de trabajo a seguir.

2 *Entorno Espacial*

El espacio es un medio hostil para la mayoría de los seres vivos que habitan la Tierra. Pero además, también lo es para los componentes electrónicos que se usan día a día en infinidad de equipos y aparatos de uso cotidiano. Las tecnologías utilizadas en la fabricación de componentes electrónicos comerciales están pensadas para trabajar en las condiciones nominales de la Tierra. Poniendo ejemplos concretos, el rango de temperatura de los componentes comerciales es de 0ºC a +70ºC. Para aquellas aplicaciones algo más extremas, y sobre todo que tengan relación con algún factor de seguridad humana, existen otros rangos extendidos como el rango industrial, de -40ºC a +100ºC, o el específico de automoción, de -40ºC a +125ºC. Por último, para aplicaciones militares, aeronáuticas y espaciales, se utiliza el rango denominado militar, que va desde los -55ºC a los +125ºC [L-std- and Mil-std- 2008]. Prácticamente el 100% de los componentes que se usan en espacio son de grado de temperatura militar. Pero incluso este rango se queda corto para ciertos tipos de misiones o instrumentos. Los telescopios y sensores ópticos utilizados para observar el espacio profundo suelen trabajar a temperaturas cercanas al 0 Kelvin [Laubier et al. 2017]. Por el contrario, las misiones que estudian de cerca las características de nuestro Sol han de trabajar a temperaturas que pueden superar las +200ºC [Jurgens 1998]. En las misiones de exploración planetaria, la instrumentación utilizada para las mediciones atmosféricas, pueden tener recorridos de temperatura de más de 150ºC en pocas horas, provocando un estrés térmico enorme a los encapsulados de los componentes electrónicos [Apestigue et al. 2015]. Cuando a estas condiciones de temperatura extrema le añades la falta de atmósfera, la situación empeora drásticamente. La disipación de calor no es posible por convección, por lo que las transferencias de temperatura se ven mermadas, y se hace necesario la aplicación de técnicas concretas de disipación por conducción, o la aplicación de pinturas especiales para facilitar la radiación térmica.

Aquellos componentes que han de soportar temperaturas más allá del rango militar, los propios diseñadores de instrumentación y sistemas espaciales han de implementar (o en su defecto contratar a empresas específicas) campañas de calificación para demostrar de forma estadística que los componentes serán capaces de sobrevivir en esos entornos de temperatura extremos.

Además, las condiciones de vacío no sólo afectan de forma negativa a la disipación de calor de los componentes electrónicos. Tiene también otros efectos adversos en la electrónica. Por un lado, los materiales de los encapsulados pueden sufrir el efecto denominado como desgasificación, que consisten en la extracción de partículas que se han quedado atrapadas dentro de los componentes durante el proceso de encapsulado y fabricación. Estas partículas quedan con movimiento libre dentro de los sistemas, y tienden a depositarse en los puntos fríos cercanos. Estos puntos suelen ser elementos ópticos como CCD, fotodiodos o células solares, mermando sus capacidades o cambiando sus propiedades. Por otro lado, algunos metales como el estaño, el zinc o el cadmio, sufren un efecto en vacío conocido como "*tin whiskers*" o bigotes de estaño. Son estructuras similares a pelos hechas de un solo grano, o solo unos pocos granos, que a veces brotan de los metales. Los recubrimientos de estaño, zinc y cadmio son especialmente capaces de desarrollar bigotes; pero también se han visto bigotes en oro, plata, plomo y otros metales [Brusse 2002]. En la Fig. 1 se puede observar un claro ejemplo de cómo crecen estos bigotes metálicos sobre los pines de un conector, pudiendo crear cortocircuitos entre ellos.

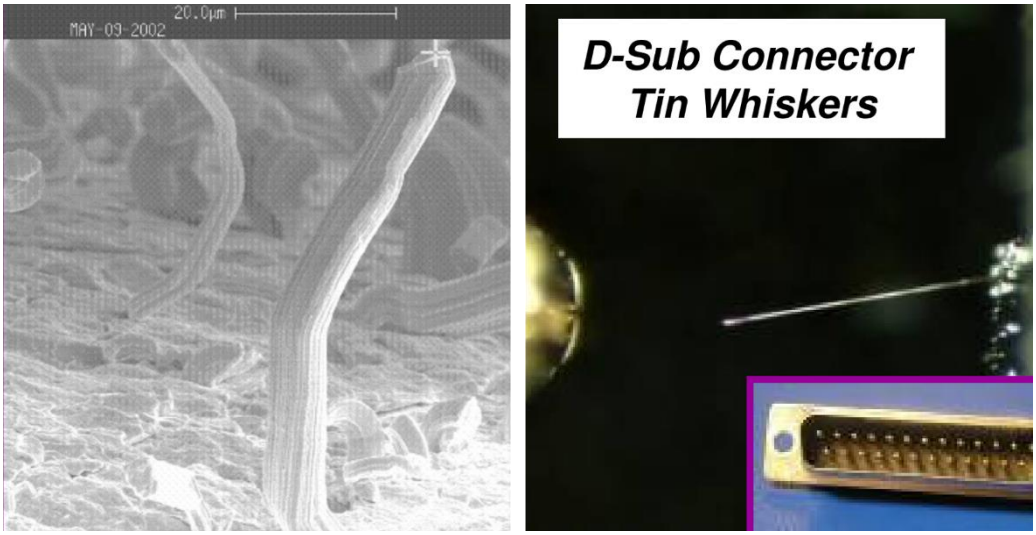


Fig. 1. "Tin whiskers" o bigotes de estaño creciendo en condiciones de vacío.

Las técnicas más utilizadas para mitigar (que no impedir) los efectos del crecimiento de los bigotes metálicos en los sistemas espaciales son dos:

a) La utilización de revestimientos de conformación sobre las partes metálicas de los componentes, capaces de impedir el crecimiento de los bigotes fuera del revestimiento, y en última instancia, protegiendo el resto de partes metálicas del componentes o componentes adyacentes.

b) La utilización de pozos calientes de estaño-plomo (Pb-Sn) para pre-estañar las partes metálicas de los componentes. Esta solución es menos proclive a generar bigotes que otros metales típicos como el oro o el estaño-plata.

Como se puede observar, las condiciones espaciales extremas de temperatura y vacío dificultan enormemente la utilización de infinidad de componentes comunes, además de añadir una gran complejidad a la hora de fabricar, integrar y probar los instrumentos antes de poder ser lanzados al espacio. Sin embargo, ninguno de estos problemas en comparable al que se presenta debido a la radiación ionizante presente fuera de las seguras barreras magnéticas que nos proporciona el núcleo de la Tierra. En la siguiente sección se describen de forma breve los tipos de partículas existentes en los diferentes entornos espaciales. Seguidamente, se explican los efectos que esta radiación produce en los dispositivos electrónicos.

2.1 Entorno de Radiación Espacial

El espacio exterior es un entorno donde hay partículas de muy alta energía que, debido a la radiación ionizante, provocan fallos de funcionamiento en los sistemas electrónicos e incluso pueden producir daños irreversibles en los componentes que integran dichos sistemas. Estas partículas de alta energía se dividen principalmente en tres categorías. La primera corresponde a las partículas atrapadas debido a los efectos de los campos magnéticos planetarios, tales como los anillos (o cinturones) de Van Allen de la Tierra. La segunda corresponde a las partículas procedentes del exterior de nuestro Sistema Solar, contenidas en los denominados rayos cósmicos. La tercera corresponde a la actividad del Sol, que presenta emisiones de radiación periódicas, y se caracteriza por flujos muy elevados de protones e iones pesados, contenidos en el denominado viento solar.

2.1.1 PARTÍCULAS ATRAPADAS DEBIDO AL CAMPO MAGNÉTICO TERRESTRE

Los anillos magnéticos causados por los campos magnéticos planetarios atrapan las partículas con un alto nivel de energía ionizante. Estas partículas proceden de fuentes de irradiación externas, tales como el Sol u otras estrellas más allá del Sistema Solar. En la Tierra, estos anillos magnéticos se denominan cinturones de Van Allen (Van Allen's Belt), son dos (aunque en determinados períodos de alta actividad solar se ha detectado un tercero) y retienen protones con energías de hasta varios cientos de MeV y electrones con energías del orden de MeV. La combinación de sus movimientos en el campo magnético terrestre (rotación a lo largo de las líneas de campo, rebote entre los espejos magnéticos y deriva alrededor de la Tierra) hace que el campo magnético/eléctrico de las partículas sea isotrópico en las naves o satélites que sufren el flujo de estas partículas [R. A. Reed and Ladbury 2000].

En la Fig. 2, se muestra una representación de los cinturones de Van Allen y se pueden observar los tipos de partículas atrapadas en ellos.

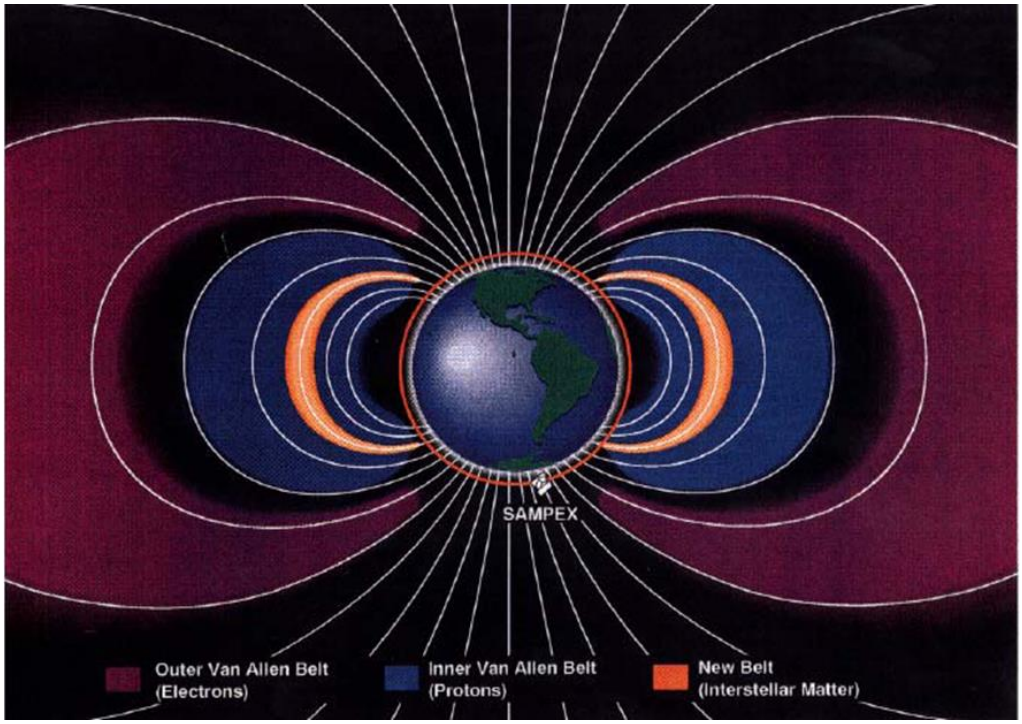


Fig. 2. Representación gráfica de los cinturones de Van Allen.

La energía de los protones atrapados en estos anillos llega hasta cientos de MeV, con unos flujos considerables. A partir de esa energía el flujo de dichos protones decae rápidamente. La mayor población de protones atrapados, con energías mayores que 10 MeV, se sitúa a altitudes por debajo de 20.000 km, mientras que aquellos protones con energías de alrededor de 1 MeV o menos se localizan en altitudes geosíncronas o mayores.

En cuanto a los electrones, hay dos áreas en los anillos de Van Allen donde se encuentran en mayor medida, en la zona interna y en el anillo en sí. En la zona interna las energías que presentan los electrones van hasta 4.5 MeV aproximadamente. El flujo de estos electrones es de alrededor de 106 cm^2s^{-1} para energías superiores a 1MeV. Estos flujos se incrementan gradualmente durante los períodos de mayor actividad solar por un factor de 2 a 3. En la zona exterior (el cinturón en sí) las energías de los electrones son generalmente menores de 10 MeV. Allí el flujo en régimen permanente, para energías superiores a 1 MeV, es básicamente $3 \cdot 10^6 cm^2s^{-1}$. Esta zona es muy dinámica y los flujos pueden variar en varios órdenes de magnitud de un día para otro dependiendo de la actividad solar.

2.1.2 PARTÍCULAS SOLARES

Se cree que hay dos categorías de eventos por partículas solares y que cada una de ellas acelera las partículas de diferente forma. La primera, denominada Llamaradas Solares (por los destellos de luz que acompañan al fenómeno), suceden cuando hay una acumulación excesiva y localizada de energía en el campo magnético del Sol, lo que provoca que se libere una gran cantidad de energía bruscamente. Estas explosiones tienden a ser ricas en electrones y duran horas. La segunda, denominada Expulsión de Masa Coronal (*CME*), por otra parte, es una gran erupción de plasma (un gas compuesto de iones libres y electrones) que lleva una onda de choque saliente conocida como Viento Solar, así como partículas aceleradas. Las CME tienden a ser ricas en protones y duran días [Reames 1999].

Todos los elementos químicos que aparecen de forma natural, desde protones hasta Uranio, están presentes en los Eventos por Partículas Solares.

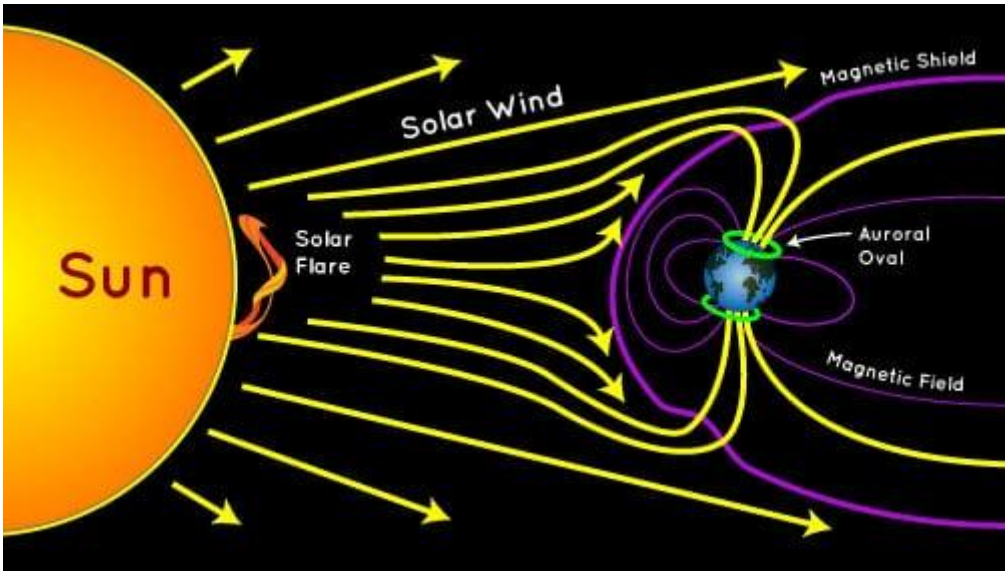


Fig. 3. Ilustración de los diferentes eventos de partículas solares y su entrada en la Tierra.

El Sol es a la vez una fuente y un modulador de las partículas ionizantes en el entorno espacial, que causan daño por irradiación en los materiales y, por tanto, en los sistemas contenidos en las naves espaciales. Si se comprende la actividad cíclica del Sol, se podrá modelar el entorno espacial de radiación donde operan dichas naves. El Sol presenta un ciclo de actividad de 11 años, aproximadamente. Durante este período, hay 7 años de máximo solar, donde los niveles de actividad son altos, y 4 años de mínimo solar, donde los niveles son bajos. En realidad, la transición entre los dos períodos es suave, aunque se considera abrupta por conveniencia para los modelos. Al final de cada ciclo completo de 11 años, la polaridad magnética del sol cambia y comienza otro ciclo de 11 años. Por lo tanto, estrictamente hablando, el ciclo de actividad completa dura aproximadamente 22 años. La polaridad magnética, aparentemente, solo afecta a los flujos de rayos cósmicos galácticos (*GCR*) [Badhwar 1996], y no a los flujos de las partículas atrapadas o a las partículas procedentes de eventos solares. Es por esto que se estudia el problema con un esquema de ciclos de 11 años.

Se sabe que los eventos por partículas solares ocurren con mayor frecuencia e intensidad durante la fase final del máximo solar [Shea and Smart 1995]. Los flujos por electrones atrapados tienden también a ser mayores durante esta fase de declive de actividad [Piet, Bourdarie, Boscher and Friedel 2006]. Los flujos por protones atrapados en orbitas *LEO*

llegan a su máximo durante mínimo solar, pero la localización temporal exacta de su pico máximo depende de su localización espacial [Huston and Pfitzer 1998]. Finalmente, los flujos por GCR también llegan a su máximo durante los mínimos solares y su valor sí depende de la polaridad magnética del Sol.

Para concluir esta sección, la Fig. 4 representa de forma simplificada el espectro de partículas en el entorno espacial cercano a la Tierra [R. Reed 2006].

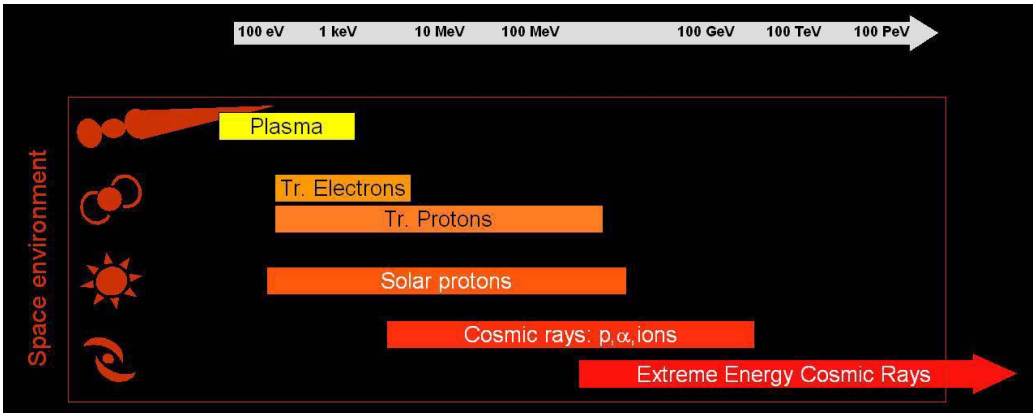


Fig. 4. Diagrama simplificado del espectro de partículas ionizantes en el entorno espacial cercano a la Tierra.

2.1.3 RAYOS CÓSMICOS GALÁCTICOS

Los rayos cósmicos galácticos (*GCR*) son partículas cargadas con muy alta energía que se han generado en el exterior de nuestro Sistema Solar y que se cree que proceden de restos de explosiones de estrellas supernovas. Todos los elementos de la tabla periódica surgidos de forma natural (hasta el Uranio), están presentes en los GCR, aunque hay un salto brusco en los números atómicos mayores que el Hierro (Z=26). Es por ello por lo que generalmente se denomina a las partículas presentes en los GCR como iones pesados. Las energías que presentan pueden ser superiores a 1.000 GeV, aunque aún no se comprenden los mecanismos de aceleración necesarios para alcanzar tales energías. Los flujos son generalmente de unos pocos cm^2s^{-1}, y pueden variar con el ciclo solar [R. Reed 2006].

En la Fig. 5 se puede observar como el viento solar choca contra el medio interestelar, provocando un arco de choque similar al provocado por la magnetosfera de la Tierra contra el viento solar:

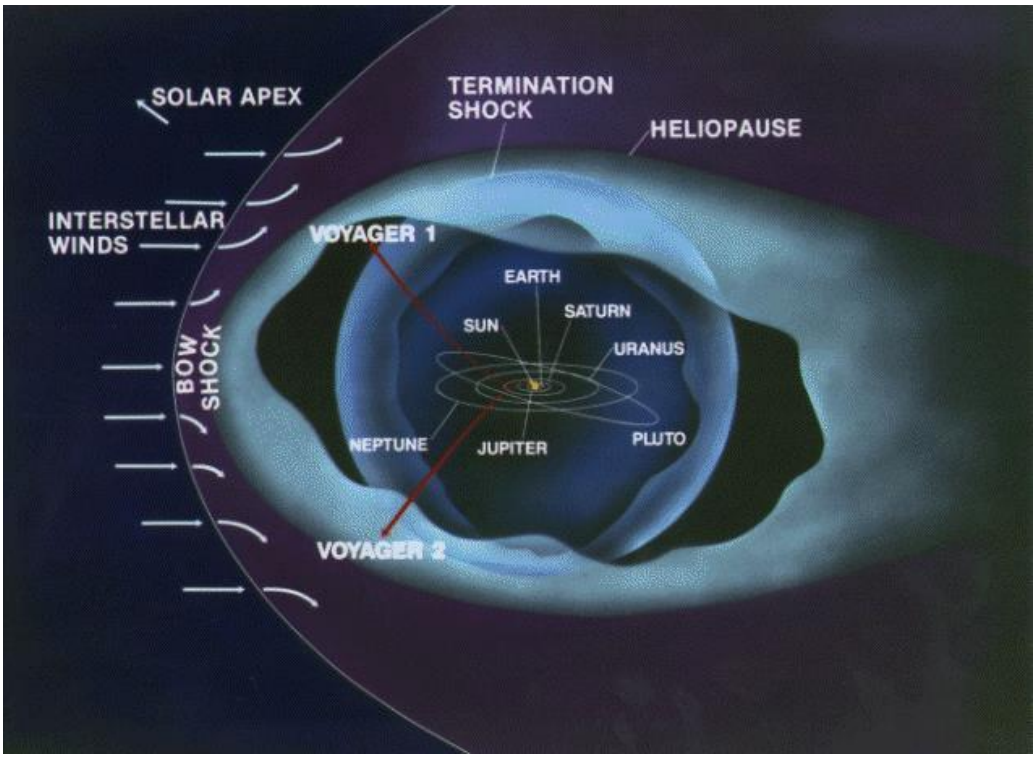


Fig. 5. Ilustración de la Heliosfera con los rayos GCR entrando en el Sistema Solar.

2.2 Efectos de la Radiación en Componentes Electrónicos

Los efectos de la radiación ionizante sobre los dispositivos microelectrónicos se dividen en tres grupos: Efectos de Eventos Simples, Dosis Total de Ionización y Daño por Desplazamiento.

2.2.1 Efectos de Eventos Simples (SEE)

En esta sección se describe la interacción de las partículas ionizantes, procedentes del entorno espacial, con los componentes electrónicos. En esta interacción de iones pesados de muy alta energía con la materia, la energía se transfiere directamente a los electrones. Estos electrones con muy alta energía (también llamados Rayos Gamma)

provocan una cascada de electrones que resulta en un plasma denso de pares electrón-hueco en los átomos del material que recibe la irradiación. Cuando se mide la capacidad de un ion pesado para producir daño en un material semiconductor, se habla de Transferencia Lineal de Energía (*LET*), que es la energía que pierde una partícula ionizante en un volumen de masa sensible, por unidad de distancia. Las unidades que se usan para LET son MeV·cm^2/mg.

A diferencia de los Efectos de Eventos Singulares (*SEE*) provocados por los iones pesados, los SEE provocados por protones ocurren por un fenómeno de recombinación (generación de partículas ionizantes secundarias) producido como resultado de la interacción del semiconductor con el protón incidente. La energía de este protón tiene una influencia significativa en estas recombinaciones y es la razón por la cual se representa su efecto basado en ella.

Los iones pesados, al igual que las partículas alfa y los protones de alta energía, pueden causar SEE estáticos, transitorios y permanentes. Aquí se describe brevemente los eventos más típicos:

SEL: ocurre cuando la estructura parásita PNPN, presente en la tecnología CMOS de pozo simple, se activa debido a una corriente elevada. Este efecto puede ser destructivo o no. Para que ocurra este último caso, la alimentación del dispositivo debe apagarse inmediatamente para desactivar la estructura PNPN antes de que una corriente tan elevada destruya el material.

SET: ocurre cuando el efecto de la partícula ionizante ha ocurrido cerca de la puerta de un transistor MOSFET y se ha generado un pico de corriente tal que resulta en un pulso de tensión suficiente para activar la conmutación de una puerta lógica combinacional. Este efecto es transitorio, pero si la duración de este pulso de corriente es acorde a la frecuencia de operación del circuito, puede propagarse a través de la lógica combinacional del mismo y causar fallos de funcionamiento, bien porque llegue a las salidas del circuito o bien porque alcance elementos de memoria y se almacene un valor erróneo en ellos.

SEU: ocurre cuando se almacena en un elemento de memoria el valor contrario al que debía tener almacenado. Este efecto se conoce como conmutación de bit, bit-flip, y puede deberse a la interacción de una partícula ionizante en el elemento de memoria mismo o a la propagación de un SET. Su efecto se mantiene hasta que se vuelva a escribir el elemento de memoria con un valor correcto.

MBU: ocurre cuando varios elementos de memoria sufren un SEU; es decir, conmutan sus valores almacenados, con lo que hay varios bits erróneos en datos almacenados en memoria del circuito. Este fenómeno es debido al efecto de una sola partícula ionizante, ion o protón. Se ha observado este efecto en celdas de memoria donde muchos nodos, físicamente adyacentes, se ven afectados (relación espacial –spatial relationship). Así mismo, también se ha observado en circuitos secuenciales que operan a una velocidad tan alta que permite capturar los SETs en los elementos de memoria (relación temporal – temporal relationship).

SEFI: ocurre cuando un SEU o un SET afecta a una región crítica de un circuito, como por ejemplo una máquina de estados. Un SEFI es no destructivo, pero puede provocar operaciones incorrectas durante un tiempo prolongado si no se detecta y elimina a tiempo. Un ejemplo de SEFI es un SEU en el circuito de reset que inicializa el circuito mientras está operando normalmente.

2.2.2 DOSIS TOTAL DE IONIZACIÓN (TIDTID)

Por otro lado, los efectos de la Dosis Total de Ionización (*TID*), causan una degradación gradual en los componentes electrónicos, que depende de la energía de ionización absorbida [Dressendorfer n.d.]. La principal causa de degradación por TID se debe al almacenamiento de cargas atrapadas en el óxido. Las capacidades y características del dispositivo podrán ir empeorando a medida que se almacenas más cargas hasta finalmente dejar de funcionar, pudiendo llegar a quemar el dispositivo.

La métrica que se usa para medir el efecto de la TID se denomina Dosis de Ionización, y se define como la energía depositada por unidad de masa del material que comprende el volumen sensible. La unidad que más se utiliza para esta métrica es el "rad", donde 1 rad equivale a 100 erg/g.

2.2.3 DAÑOS POR DESPLAZAMIENTO (DDD)

Por último, los efectos o daños por desplazamiento (***DDD***) son causados por partículas energéticas interactuando con los átomos de las estructuras cristalinas. Los neutrones, electrones, protones e iones pesados pueden desplazar los núcleos de la estructura cristalina creando defectos en la misma. Estos defectos degradan los parámetros de los semiconductores, pudiendo en última instancia dejar de funcionar.

3 *Estado de la Técnica*

En este capítulo se presenta una revisión del estado de la técnica en los temas tratados en este trabajo de Tesis Doctoral. Para ello, se ha divido el capítulo en dos secciones. La primera desgrana las diferentes técnicas aplicadas hasta el momento para el endurecimiento de computadores, con el objetivo de mejorar y asegurar la fiabilidad de estos frente a los efectos de la radiación. La segunda muestra un resumen de las técnicas de evaluación de la fiabilidad descritas en la literatura.

3.1 Técnicas de Endurecimiento y Mitigación

Esta sección describe las técnicas basadas en el diseño a distintos niveles de abstracción que pueden utilizarse para mitigar o detectar los efectos de la radiación en computadores desarrollados con tecnologías sensibles a los mismos (no *RadHard*). De manera que, utilizando tecnologías *RadTol*, inmunes a efectos de *TID* y *SEL*, pero no a *SEU* ni *SET*, se puede mejorar la fiabilidad de los computadores hasta alcanzar niveles adecuados para una aplicación concreta.

Las técnicas existentes en la literatura, aplicadas a procesadores, pueden agruparse en función de si están basadas en modificaciones del hardware, del software o de ambas (técnicas híbridas) [Fayyaz and Vladimirova 2016].

3.1.1 TÉCNICAS DE MITIGACIÓN DE FALLOS BASADAS EN MODIFICACIONES HARDWARE

Las técnicas de mitigación basadas en hardware están orientadas a modificar la arquitectura del procesador, bien añadiendo redundancia o bien añadiendo un bloque hardware adicional para la observación y comprobación del funcionamiento del sistema.

3.1.1.1 Redundancia hardware

Las técnicas de redundancia hardware pueden aplicarse en distintos niveles de detalle: a nivel de elemento lógico, a nivel de bloque o a nivel de circuito. En todos los casos, se comparan las salidas de las distintas réplicas para detectar discrepancias debido a un fallo.

La técnica más popular consiste en replicar tres veces el bloque a endurecer y realizar una votación por mayoría para detectar fallos simples. Esta técnica se conoce como *TMR*. Si en vez de tres réplicas se realizan N, la técnica se denomina *NMR*, donde N debe ser un número impar y, si es mayor que tres, permite detectar errores dobles (5MR) o múltiples (N>5).

Esta técnica requiere disponer de la descripción del circuito para poder modificar su diseño a nivel de elemento o bloque, lo que no es posible en muchos casos. Además, el incremento de recursos lógicos necesarios es significativo y aumenta con el número de réplicas, por lo que generalmente se recurre a un uso selectivo de esta solución [Valderas, Garcia, López and Entrena 2009][Restrepo-Calle, Cuenca-Asensi and Martínez-Álvarez 2015].

Una simplificación de la técnica *NMR*, es la técnica de duplicación con comparación (*DWC*), donde sólo dos réplicas se utilizan para detectar un fallo, pero no mitigarlo. En este caso el procesador podría ser avisado de que se ha producido un error para que pueda realizar la acción necesaria en función de los requisitos de la aplicación. Esta técnica puede aplicarse a nivel de elemento lógico (registros, bloques de control, etc.) o puede aplicarse a nivel del componente. Un ejemplo sería un sistema en el que hay dos procesadores funcionando en *lockstep*. Esta técnica es especialmente aplicable cuando se dispone de un sistema con dos o más núcleos de procesamiento [Hyman, Bhattacharya and Ranganathan 2011].

3.1.1.2 Bloques dedicados al control de flujo

En el caso particular de los procesadores, una técnica habitual es el uso de un bloque adicional que permita detectar fallos en la ejecución de la aplicación. Este bloque puede ser un periférico ya existente en el sistema de computación. La solución más simple es

el uso de un *perro guardián* o ***watchdog*** utilizando la lógica ya incluida en los microprocesadores. El *watchdog* genera una señal de error si el procesador no lo reinicia periódicamente.

Otra solución consiste en añadir un nuevo bloque de aplicación específica que se añade como periférico o se conecta al bus [Tirumurti, Karimi, Maniatakos, Jas and Makris 2010], o a la interfaz de depuración, o la traza disponible en los microprocesadores actuales [Portela-Garcia, Grosso, et al. 2012]. En el caso de los módulos conectados al bus del procesador, son técnicas que modifican la estructura del sistema y que pueden introducir penalizaciones en las prestaciones del mismo. Sin embargo, los módulos conectados a las interfaces de traza o depuración no requieren modificar el sistema puesto que dichas interfaces son accesibles en los procesadores actuales lo que permite su aplicación en circuitos comerciales, cuando no se dispone de una descripción del circuito.

3.1.2 TÉCNICAS DE MITIGACIÓN DE FALLOS BASADAS EN MODIFICACIONES SOFTWARE

Las técnicas de mitigación basadas en software están orientadas a modificar el programa a ejecutar por el procesador a nivel de instrucción o de aplicación. Estas técnicas se basan en añadir redundancia en términos de información (replicando datos), operaciones (replicando instrucciones del programa) o redundancia temporal. Posteriormente se comparan los resultados obtenidos por cada réplica para detectar la presencia de errores [A. Li and Hong 2007] [Vemu and Abraham 2011] [Goeders, Wirthlin, Quinn, James and Bohman 2018].

Las técnicas de mitigación de fallos basadas en modificar el software a ejecutar son muy flexibles y permiten mejorar la fiabilidad de procesadores *COTS*, en los que no se pueden realizar modificaciones de la estructura hardware. Sin embargo, su mayor limitación es que reducen la velocidad y prestaciones del sistema.

3.1.3 TÉCNICAS DE MITIGACIÓN DE FALLOS HÍBRIDAS

Las técnicas de mitigación de fallos que combinan soluciones hardware y software pretenden aprovechar las ventajas de ambos métodos a la vez que contrarrestan las

limitaciones de cada tipo. De esta forma se pretende mejorar la fiabilidad del sistema, añadiendo el menor hardware posible y sin reducir las prestaciones del sistema. Para ello es necesario que las soluciones sean lo menos intrusivas posibles. Mientras que la parte hardware de las soluciones suele dedicarse a detectar fallos en el control de flujo mediante la observación de los buses o interfaces, los bloques software de las técnicas de mitigación están orientadas a detectar fallos en los datos [Parra et al. 2014][Parra Avellaneda 2017].

En el caso de sistemas complejos, como los sistemas distribuidos, existen escasas aportaciones en la literatura hasta el momento. En [Vaskova et al. 2014] se presenta una técnica basada en endurecimiento colaborativo para mitigar el efecto de los fallos en aplicaciones de automoción.

3.2 Técnicas de Evaluación

El proceso de endurecimiento de un circuito o sistema para aumentar su fiabilidad y robustez frente a fallos requiere de técnicas que permitan cuantificar dicho nivel de fiabilidad. Estas técnicas de medida permiten evaluar la eficacia de las técnicas de mitigación aplicadas.

La sensibilidad de un circuito o sistema digital a fallos *SEU* y *SET* depende de dos factores: la tecnología utilizada para la implementación del circuito y la funcionalidad que realiza el uso que se hace de los recursos lógicos. La sensibilidad del sistema se calcula como una combinación de la sensibilidad debida a cada uno de dichos factores.

Para medir el impacto de la tecnología en la sensibilidad de un circuito frente a los fallos de la radiación ionizante se realizan experimentos de irradiación forzada. Estos experimentos consisten en aplicar un haz de partículas cargadas (protones o iones pesados en función de las características del entorno donde operará el sistema finalmente) sobre el circuito a estudiar. La relación entre el número de eventos ocurridos y la fluencia de partículas del haz (número de partículas por unidad de área) se denomina

sección eficaz (σ), y permite caracterizar la sensibilidad de un dispositivo debido a sus características físicas.

Para evaluar cómo la operación del circuito afecta a la fiabilidad del mismo en presencia de fallos, se pueden aplicar técnicas de inyección forzada de fallos durante la etapa de diseño del mismo, puesto que no es un requisito disponer de un prototipo físico. En este caso, existen en el estado de la técnica numerosas aportaciones donde se proponen mecanismos para forzar un fallo y observar su efecto. Con estas técnicas se obtiene la cobertura de fallos (τ) del circuito como la relación que hay entre los fallos que producen un mal funcionamiento del circuito y el número total de fallos *SEU* o *SET* producidos.

La mayor parte de las técnicas de inyección de fallos existentes están orientadas a evaluar componentes, no sistemas. En esta sección se recogen las aportaciones existentes para medir la tolerancia a fallos de sistemas distribuidos.

Existen soluciones teóricas que proponen modelos de los sistemas distribuidos a evaluar y los simulan bajo los efectos de fallos de acuerdo a modelos de fallos seleccionados previamente [Charle, Viti and Tampere 2011] [Wilcox et al. 2015] [Correia, Ferro, Junqueira and Serafini 2012]. Normalmente, los modelos teóricos implican una gran complejidad para poder ofrecer resultados representativos. Esto hace que la mayor parte de las soluciones existentes se basen en métodos experimentales. Las técnicas de inyección de fallos mediante simulación y emulación del circuito o sistema a evaluar permiten estudiar la fiabilidad frente a *SEU* durante etapas tempranas del diseño, lo que facilita dicha labor y reduce el coste total de un diseño robusto. Sin embargo, hay que recalcar que, en cualquier caso, la inyección física, mediante experimentos de irradiación forzada, por ejemplo, es la única forma de cualificar un sistema para una aplicación crítica en seguridad [Quinn, Black, Robinson and Buchner 2013].

En [Berg 2007] [Berg et al. 2009] y [Petrick et al. 2014] se proponen soluciones para evaluar la fiabilidad de componentes de un sistema de forma que a través de componentes fiables se pueden construir sistemas robustos. Sin embargo, algunas de las contribuciones recientes pretenden estudiar la robustez a nivel de sistema, partiendo de la hipótesis de que es posible que un sistema sea tolerante a fallos estando formado por

componentes que no lo son. De esta manera, no sería suficiente con conocer la sensibilidad a fallos de los componentes y habría que realizar el estudio a nivel de sistema.

En la industria de automoción, donde los sistemas distribuidos son muy comunes, existen soluciones para evaluar la funcionalidad del sistema que se basan en el uso de hardware de prototipado (*hardware-in-the loop*) y que han sido aplicadas a la evaluación de los efectos *SEU* y *SEFI* [Wältermann, Schütte and Diekstall 2004] [Praprotnik, Gartner, Zauner and Horauer 2009].

En [Siegle, Vladimirova, Poivey and Emam 2015], los autores proponen un método para analizar la disponibilidad aplicable un sistema de detección de averías distribuido. Dicho método combina técnicas teóricas y de inyección forzada de fallos mediante reconfiguración puesto que el sistema bajo estudio está implementado en una *FPGA* basada en *SRAM* del fabricante Xilinx. En [Fayyaz and Vladimirova 2014], también se utiliza reconfiguración de las *FPGA* que componen un sistema distribuido como mecanismo para analizar la sensibilidad frente a *SEU*.

Todas las soluciones existentes para evaluar la robustez de sistemas distribuidos, ya sean teóricas o experimentales, requieren una mejora de la accesibilidad a los recursos internos del sistema, aumentando la observabilidad y controlabilidad para poder analizar los efectos de los fallos e insertarlos respectivamente. La observabilidad de los efectos de los fallos se identifica como un aspecto clave del proceso de endurecimiento de un sistema. [Liu, Slotine and Barabasi 2013] y [Fatemi, Setoodeh and Haykin 2017] presentan análisis teóricos de la observabilidad en sistemas distribuidos generales. Mientras que en [Fernandes, Santos, Arlindo and Velazco 2007], el estudio presentado se realiza a un alto nivel de abstracción.

En el caso particular de disponer de prototipos o implementaciones finales del sistema, donde los experimentos de irradiación se realizan para incluir los efectos de la tecnología en la sensibilidad frente a fallos, poder aumentar la observabilidad es también critico a la hora de poder extraer información durante dichos experimentos. Sin embargo, existen escasas contribuciones sobre como generalizar soluciones que persigan dicho objetivo [Vaskova, Lopez-Ongil, Portela-Garcia, Garcia-Valderas and Entrena 2013].

4 *Computador basado en Endurecimiento Colaborativo*

Este capítulo presenta la arquitectura del computador distribuido diseñado para el satélite OPTOS. Este computador está basado mayoritariamente en componentes *COTS* de bajo consumo. Con el objetivo de mejorar la fiabilidad del *OBC*, se han implementado una sería de técnicas y algoritmos de endurecimiento colaborativo, que permiten el mantenimiento de las tareas críticas del computador.

El capítulo está dividido en 5 secciones. Se empieza explicando y desarrollando los condicionantes iniciales, los requisitos del computador más allá de su fiabilidad, que sirvieron de marco de trabajo para el diseño y posterior implementación de este. Las siguientes dos secciones, presentan la arquitectura del *OBC* y su funcionalidad respectivamente. La cuarta sección explica detalladamente los procedimientos y algoritmos desarrollados para implementar el endurecimiento del computador. Terminamos el capítulo con el análisis de radiación del computador.

4.1 Antecedentes y Premisas

Para entender la elección de la arquitectura propuesta de un computador de vuelo basado en endurecimiento colaborativo, hay que entender bien todos los condicionantes que propiciaron el diseño final del computador de OPTOS.

OPTOS toma su nombre de la finalización de una década de desarrollo de tecnología inalámbrica difusa en el Instituto Nacional de Técnica Aeroespacial (INTA), en Madrid. Desde finales de los 90 se había desarrollado juntamente con los principales contratistas de satélites europeos y otros centros de investigación, con financiación tanto propia como de la ESA (European Space Agency), una tecnología de comunicaciones ópticas inalámbricas y difusas para Espacio [I. Arruego et al. 2009, Tamayo, Alonso, Jimenez, Arruego and Guerrero 2010]. A esta tecnología se le bautizó con el nombre de OWLS (Optical WireLess for intra-Spacecraft communications). Este desarrollo tecnológico se hizo en parte a través de diferentes programas tecnológicos (ESA Contracts: 16428 & 19545/06/NL/GLC) y de fondos propios del INTA. Así mismo, en el año 2005, el

INTA lanzó un satélite diseñado y fabricado por el propio Instituto, llamado Nanosat-01 [Michelena, Cerdán and Arruego 2009]. Este satélite contaba con diferentes instrumentos basados en la tecnología de comunicaciones ópticas inalámbricas. Los resultados obtenidos en vuelo fueron mejor incluso de lo esperado [Ignacio Arruego, Martinez and Guerrero 2011], pudiendo además de realizar su cometido, servir para identificar los diferentes tipos de partículas energéticas. El siguiente paso natural de cara a elevar la madurez tecnológica de esta tecnología a su máximo nivel TRL9 (Technology Readiness Level 9), era utilizar la tecnología no como experimento en un satélite, sino como parte de la plataforma[1]. De ahí el nombre de OPTOS, cuyo lema inicial fue "OPTOS: *A new light in the Space"*. El INTA quiso diseñar un satélite, cuyas comunicaciones internas fueran íntegramente ópticas e inalámbricas, basadas obviamente en la tecnología de OWLS. El uso de los OWLS como subsistema de comunicaciones a bordo, no sólo servía para elevar el TRL al máximo nivel, sino también para tener un satélite sin un solo cable de datos. Más importante todavía, de cara al computador, este tipo de comunicaciones permitía que todas las unidades pudieran escuchar todos los mensajes que se enviaban a través de ellas, y al mismo tiempo, facilitando la implementación de una arquitectura colaborativa.

Por otro lado, el segundo objetivo de OPTOS fue el de crear una plataforma espacial de rápido acceso y bajo coste, reutilizable como banco de pruebas para el desarrollo de nuevas tecnologías y experimentos científicos en el Espacio. El acceso al espacio proporciona una serie de características que no se pueden encontrar en la Tierra. Principalmente, un entorno de micro gravedad, de vacío y por supuesto, de radiación

[1] Independientemente del tipo de satélite, todos están compuestos por dos partes bien diferenciadas, la plataforma y la carga útil. La plataforma es aquello que proporciona servicio a la carga útil, para que está pueda desarrollar su cometido. La plataforma está compuesta por diferentes subsistemas. Entre ellos, los más comunes son: la estructura, que son las partes mecánicas del satélite que permiten aislar y sujetar el resto de subsistemas y cargas útiles; la unidad de potencia, encargada de proveer y distribuir la energía necesaria; el computador, encargado de comandar y recibir la información; el software embarcado, encargado de ejecutar la operación; las comunicaciones, encargadas de mantener al satélite conectado con las estaciones Terrenas. Por otro lado, la carga útil, es aquello que da sentido al satélite. Por poner un ejemplo, en un satélite de observación de la Tierra para vigilancia de clima, sería el sensor o sensores capaces de obtener la información deseada de la Tierra, como por ejemplo sensores ópticos tipo cámaras, o sensores magnéticos, etc.

con partículas altamente energéticas. Este entorno permite no sólo poder probar nuevas tecnologías para ser utilizadas en futuras misiones, sino también otros muchos experimentos científicos, en el ámbito de la biología, la química y la física entre otras, que nos ayuda a desentrañar los orígenes de la vida y demostrar empíricamente teorías nunca experimentadas. Las problemáticas de acceso al Espacio son básicamente dos: el acceso es caro en términos económicos. Para poder poner en órbita un satélite, hace falta contratar un hueco en un lanzador que coloque al satélite en la situación orbital deseada. Esto, para satélites pequeños[2], puede ser desde 200K€, para los satélites más pequeños, hasta varios millones dependiendo del peso del satélite. A este precio, hay que sumarle el de la fabricación del mismo, sus componentes mecánicos y electrónicos, las pruebas de vibración y termo-vacío, y por supuesto, el personal involucrado. Por otro lado, la alta especialización que supone la construcción de un satélite, capaz de sobrevivir en las condiciones extremas del Espacio, y de montar la infraestructura necesaria en Tierra para establecer comunicaciones diarias con el mismo, no están al alcance de la mayoría de instituciones o universidades. Por todos estos motivos, el INTA decidió diseñar, desarrollar y probar una plataforma que permitiera un fácil acceso al espacio a universidades, empresas y centros tecnológicos españoles.

Por lo tanto, el computador de OPTOS debía tener entre sus principales características, una alta capacidad para su reutilización en múltiples y diferentes misiones. En resumen, las premisas sobre las que se ha construido la arquitectura del computador endurecido, y por tanto han marcado las decisiones de diseño han sido:

a. Utilización de comunicaciones ópticas difusas con un bus de comunicaciones abierto.

b. Maximización de la utilización de componentes COTS con el objetivo de reducir costes y mejorar la capacidad de procesado.

[2] Los satélites se definen principalmente no por su tamaño, sino por su peso. No hay un estándar que defina específicamente cuales son los rangos, pero está comúnmente aceptado los siguientes rangos: fento-satélites, hasta 1 kg; pico-satélites, hasta 10 kg, nano-satélites, hasta 50 kg, micro-satélites, hasta 100 kg; y mini-satélites, hasta 500 kg.

c. Gran limitación de potencia, debido al pequeño tamaño de los paneles solares a utilizar.

d. Alta capacidad de reutilización.

e. Flexibilidad para la adaptación de interfaces de los futuros experimentos e instrumentos.

Junto con estas premisas, el objetivo intrínseco de la misión era utilizar una plataforma estándar que brindara la posibilidad de numerosos y rápidos accesos a lanzadores, pero sin perder de vista la necesidad de que la plataforma debía seguir siendo fiable, y con ello garantizar el éxito de los futuros experimentos en vuelo. Es precisamente ahí, donde el INTA podía aportar un alto valor añadido. La plataforma Espacial resultante, debía aunar las ventajas de reutilización del estándar CubeSat, con una componente de fiabilidad nunca vistas en un satélite de estas características. Concretamente, en lo que al computador de abordo se refiere, la aplicación de técnicas innovadoras de endurecimiento, junto con unos procedimientos de calidad prestablecidos en el ámbito Espacial, debían culminar con un computador de altas prestaciones, barato y robusto.

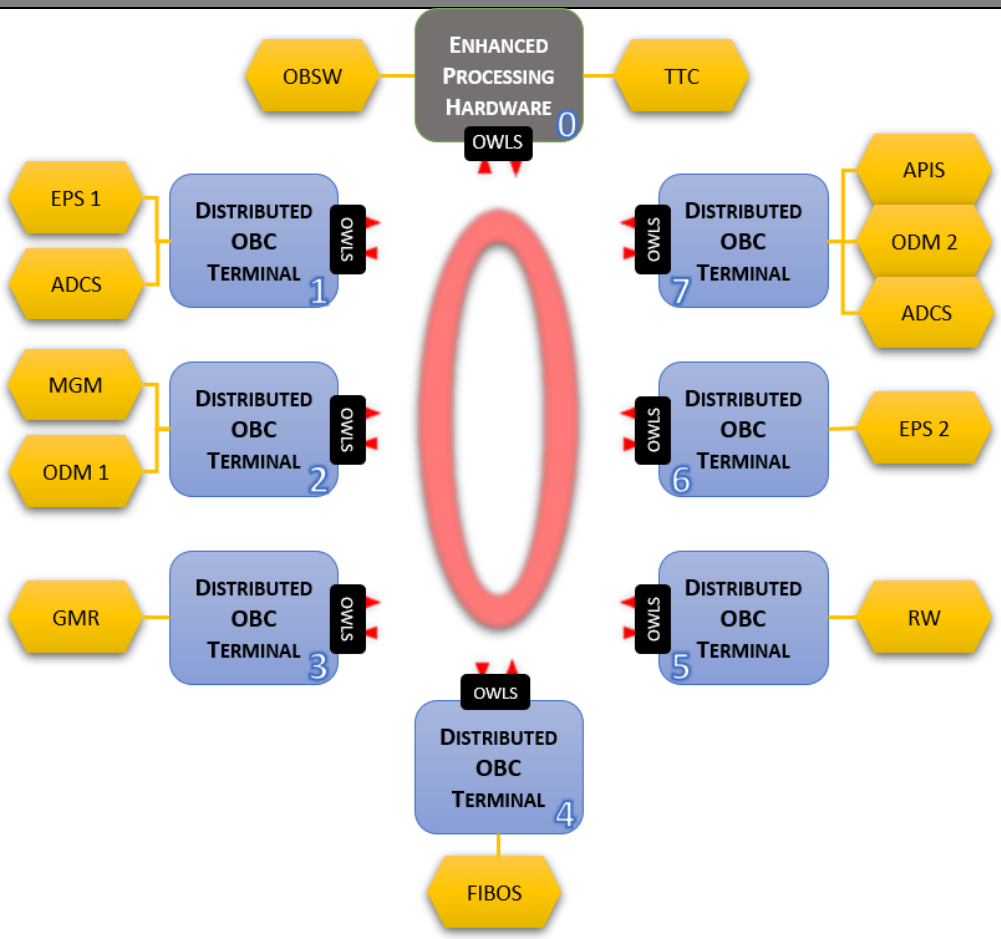
4.2 Arquitectura

La arquitectura del computador propuesta está basada en computador distribuido con endurecimiento colaborativo. Esta arquitectura responde a una solución completa para pequeños satélites, con bajos presupuestos de masa, tamaño, potencia y precio, pero de altas prestaciones. El siguiente cuadro resume las principales capacidades del computador:

DESCRIPTION
El computador distribuido está formado por una unidad de altas prestaciones y siete unidades de bajo consumo. La unidad de altas prestaciones o **EPH** (Enhanced Processing Hardware), está encargada de las comunicaciones con el subsistema de telemetría y telecomandos (TTC), la memoria de almacenamiento masivo, y de

albergar el software de vuelo (OBSW). Las otras siete unidades distribuidas o *DOT* (Distributed OBC Terminal) se encargan de forma **colaborativa** de realizar las **tareas críticas** de: mantenimiento de tiempo real, supervisar el consumo de potencia y propagar las efemérides del satélite.

BLOCK DIAGRAM



PRINCIPALES CARACTERÍSTICAS

- 1 x Optical CAN BUS
- 2 x Serial UART
- Interruptor Reset del Satélite
- Rango de Temperatura -55 a +125 ºC
- Consumo Nominal: 990 mW.
- Radiación:

  TID > 30 krad.
  LET > 63 MeV-cm2/mg
  $MTBF_{Device}$ > 20.83 días
  $MTBF_{System}$ > 3 años

<table>
<tr><td colspan="2">

- Voltajes de entrada: 3.3V y 5.0V

</td></tr>
<tr><td>

EPH

- Arquitectura RISC de 32-bit con CPU @ 48MHz
- Sistemas Operativos Compatibles: uC-Linux, Linux, VxWorks and XMK.
- 64 KByte SRAM interna y 8 Mbit SRAM externa
- 4Gbit FLASH
- Sistema de protección Watchdog externo (HW) e Interno (SW)
- Protocolo de Comunicaciones dedica con subsistema TTC

</td><td>

DOT

- Adquisición de hasta 64 señales analógicas de 10-bit. Hasta 8 en paralelo
- Control y adquisición de hasta 120 señales digitales.
- Control de Temperatura TEC[3].
- 24 KBytes de SRAM (sólo DOT TPA).
- Sistema de protección Watchdog externo (HW) e Interno (SW)
- Endurecimiento colaborativo de:
  - Tiempo Real
  - Supervisión de Consumo
  - Propagación de Efemérides

</td></tr>
<tr><td colspan="2"></td></tr>
</table>

En conjunto, la idea de diseñar un computador distribuido responde a una serie de necesidades y ventajas que proporciona dicha configuración. La idea de poder hacer un computador más robusto mediante técnicas colaborativas a nivel sistema, en contraposición a sistemas no colaborativos, donde la fiabilidad del sistema viene dada por el más débil de sus componentes, proporciona ventajas añadidas que iremos desgranando a continuación.

[3] ThermoElectric Cooler

Para empezar, un sistema distribuido permite la ejecución de múltiples tareas en paralelo, multiplicando la capacidad computacional del sistema de forma muy significativa. Por otro lado, la utilización de múltiples dispositivos suele tener la desventaja de un aumento significativo del consumo en cómputo total. Y es aquí donde entra en juego lo aprendido durante los trabajos previos mostrados en el capítulo anterior. Los componentes *RadHard*, debido principalmente a las tecnologías con las que son fabricados, tienen consumos muy superiores a los aquellos dedicados a un uso comercial y terrestre. Sin embargo, los estudios de radiación practicados [García et al. 2008] a las CPLDs de Xilinx CoolRunner-II®, demuestran que es posible utilizar dichos componentes, cuya tasa de fallo (MTBF, Mean Time Between Failure) está por encima de los 21 días por dispositivo. Sin embargo, estos dispositivos tienen un consumo medio por debajo de los 6,6 mW. Las posibilidades que se presentan al poder utilizar una arquitectura con siete dispositivos de lógica programable (PLD por su nombre en inglés, Programmable Logic Device), son evidentes. Para empezar, estos dispositivos tienen la capacidad de ejecutar múltiples tareas en paralelo, en tiempo real. Esto permite ejecutar a la vez aquellas tareas colaborativas, con el objeto de mantener las tareas críticas de forma fiable, y al mismo tiempo poder dar el soporte necesario al resto de subsistemas y cargas útiles del satélite. Por lo tanto, la arquitectura diseñada para este computador endurecido permite la ejecución de dos funcionalidades en paralelo:

1. Mantener de forma colaborativa las tareas críticas del satélite
    a. Mantenimiento del tiempo real.
    b. Propagación de efemérides.
    c. Supervisión del consumo de las diferentes líneas de potencia del satélite.
2. Dar soporte a cada una de las cargas útiles y subsistemas del satélite:
    a. Recibir los telecomandos desde el EPH y ejecutarlos.

b. Recibir las telemetrías de la C.U.[4]/S.S.[5] y enviarlos al EPH.

c. Controlar y adquirir las señales de entrada/salida para la ejecución de la operación específica de cada una.

Además, y teniendo en cuenta uno de los objetivos principales de la plataforma OPTOS[6], el uso de *PLD* como método de unión entre el *OBC* y las cargas útiles, *PL*asegura la máxima flexibilidad a la hora de implementar nuevos interfaces, garantizando controles rápidos y versátiles de las mismas.

Cada unidad *DOT* de la red ha sido implementada en una *CPLD* CoolRunner-II ™ de Xilinx, el cuál es un dispositivo programable comercial, cuya memoria de configuración se basa en un sistema dual de memorias Flash y SRAM. Esta configuración permite al dispositivo sacar el máximo partido de ambas tecnologías. Por un lado, las memorias FLASH son poco o muy poco susceptibles a *SEU* [García et al. 2008], por lo que esta memoria no se ve dañada por la radiación. Por otro lado, la memoria SRAM tiene accesos más rápidos y un consumo menor. El funcionamiento es el siguiente; cuando la CR-II se enciende, transfiere los datos de la memoria de configuración guardada en FLASH directamente, a una memoria homologa en SRAM. Esto además permite evitar la necesidad de una memoria de configuración externa. Cuando se usa este dispositivo en entornos de radiación, la gran ventaja es que la configuración en FLASH continua intacta, por lo que en cada ciclo de encendido del dispositivo, los posibles errores acumulados en SRAM son borrados y substituidos por la configuración correcta en FLASH. Durante la ejecución, la memoria FLASH no es utilizada, y la SRAM permite consumos muy bajos. En el caso de las *DOT*, el consumo de su *CPLD* ejecutándose a una velocidad de 2 MHz, es inferior a 7 mW.

[4] Carga Útil

[5] Sub-Sistema

[6] Uno de los dos objetivos principales de la plataforma era su reutilización como banco de pruebas científico-tecnológico en el Espacio, garantizando un acceso rápido, de bajo coste y fiable a las universidades, institutos de investigación y empresas españolas.

En esta sección hemos mostrado la arquitectura del computador colaborativo. En la siguiente sección, se describe de forma detallada las principales funcionalidades de cada unidad del OBC.

4.3 Funcionalidad

Como hemos visto el computador distribuido consta de un total de 8 unidades diferentes. Cada una de ellas aporta una parte de funcionalidad específica, dependiendo principalmente de a qué otros subsistemas o cargas útiles de servicio. Además, 7 de ellas implementan de forma colaborativa los algoritmos de endurecimiento que proporcionan fiabilidad a las tareas críticas desempeñadas por el computador. El firmware implementado dentro de las unidades *DOT* se describe en el lenguaje de programación hardware *VHDL*. Las tareas colaborativas son comunes a todas las *DOT* y están implementadas en todas ellas. El siguiente diagrama muestra los componentes más importantes descritos en cada *DOT* (ver Fig. 6)

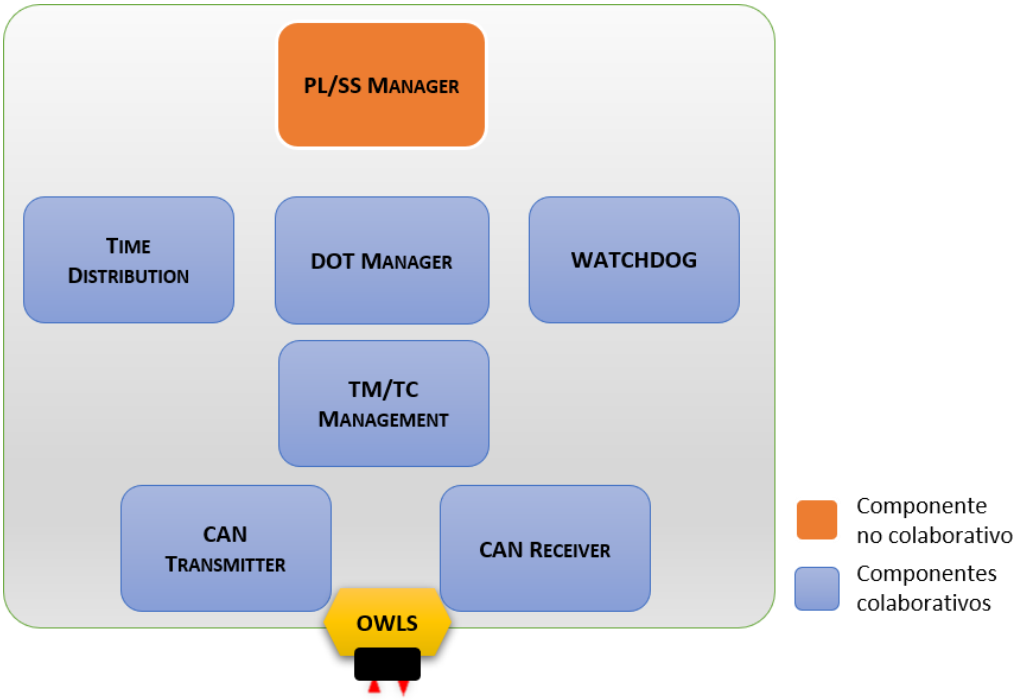


Fig. 6. Diagrama de componentes de las unidades DOT.

Como se puede observar, todos los componentes excepto el encargado de la carga útil o subsistema asociado, participan de las tareas colaborativas. El componente "DOT Manager" gestiona el funcionamiento propio de la DOT. La Fig. 7 muestra el funcionamiento interno del mismo.

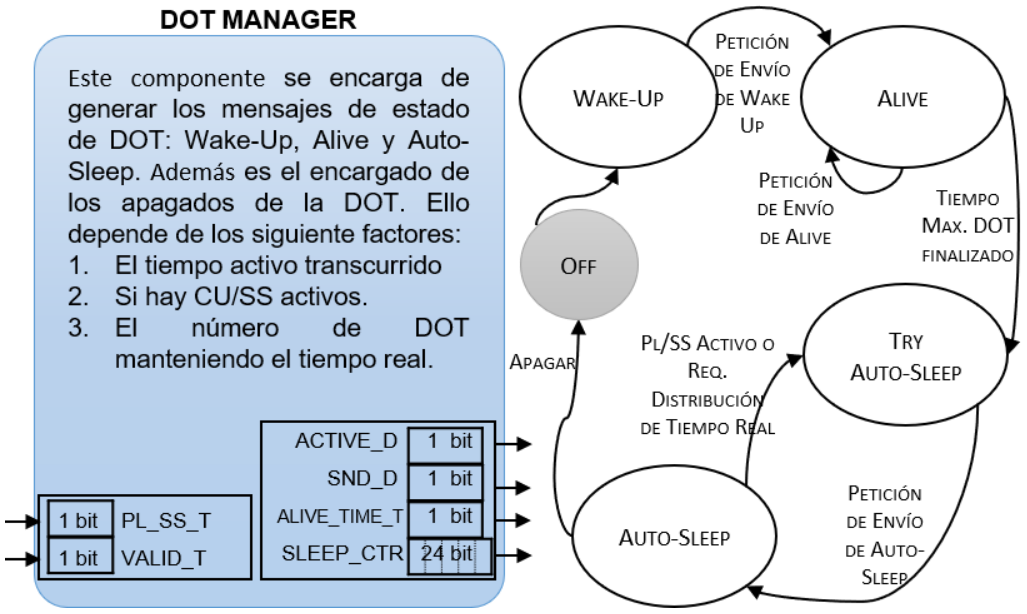


Fig. 7. Diagrama de estados e interfaces del componente colaborativo "DOT Manager".

El componente "Watchdog" es el encargado de identificar los posibles errores funcionales más críticos causados en la DOT debido a *SEU*. La Fig. 8 muestra el funcionamiento interno del mismo.

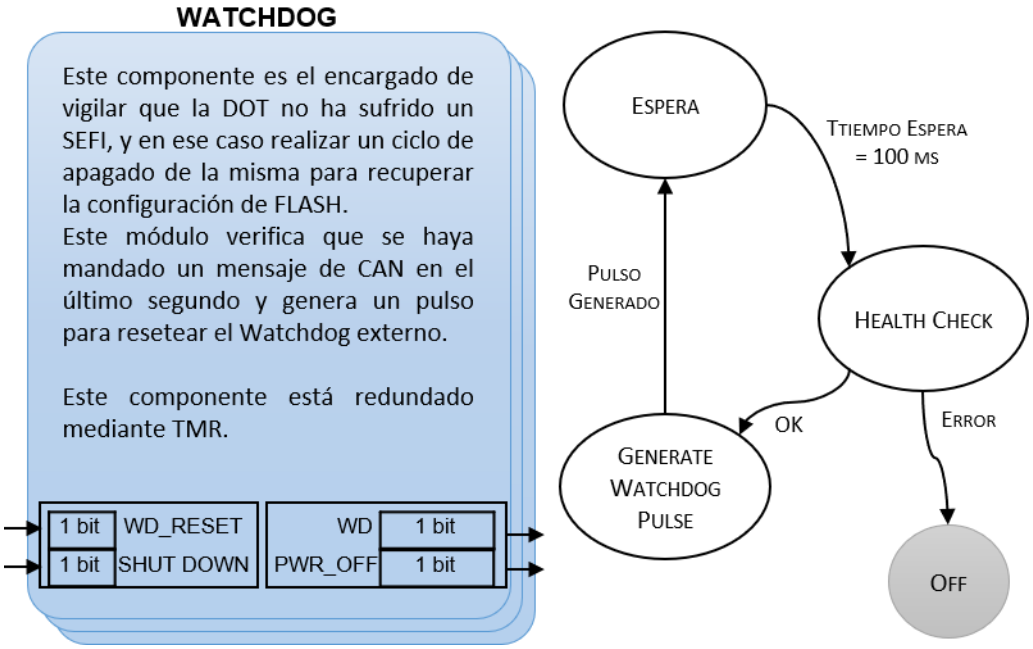


Fig. 8. Diagrama de estados e interfaces del componente colaborativo "Watchdog".

La recepción y envío de mensajes CAN a través del bus está controlada por tres componentes. El "TC/TM Management" hace de interfaz entre el bus y el resto de componentes de la DOT, recibiendo las peticiones de envío internas y distribuyendo los mensajes entrantes a los diferentes componentes. La Fig. 9 muestra el funcionamiento interno del mismo. Por otro lado, los componentes de envío y recepción de mensajes están descritos en las figuras Fig. 10 y Fig. 11 respectivamente.

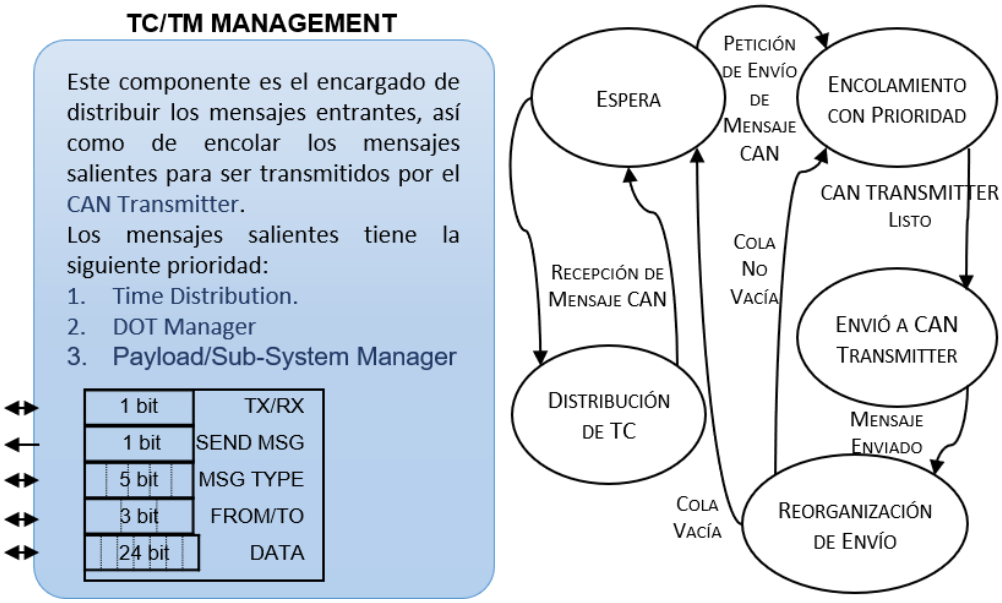


Fig. 9. Diagrama de estados e interfaces del componente colaborativo "TC/TM Management".

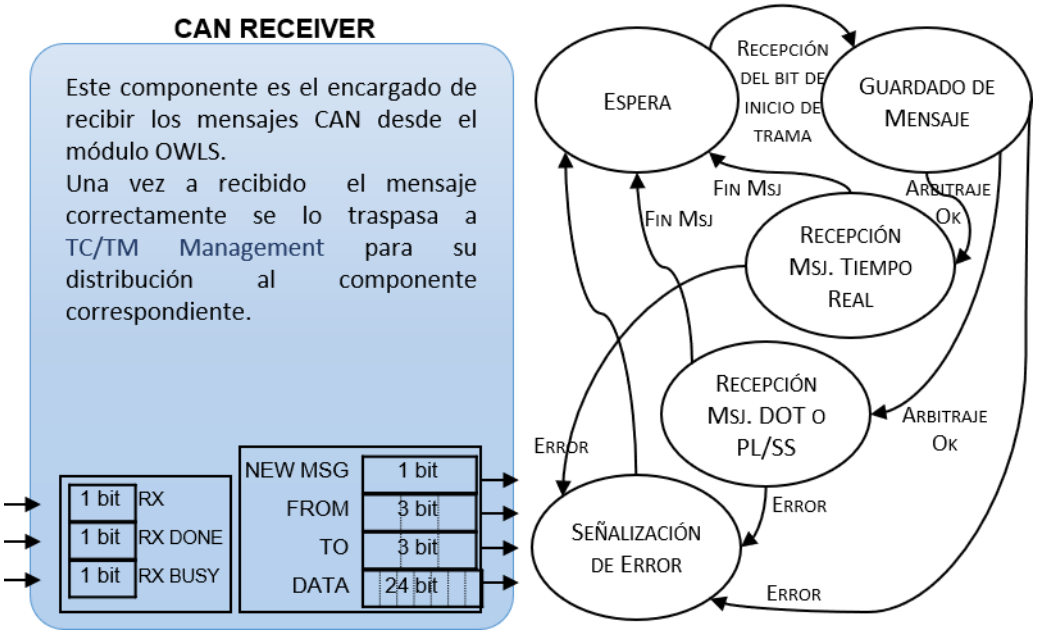


Fig. 10. Diagrama de estados e interfaces del componente colaborativo "CAN Receiver".

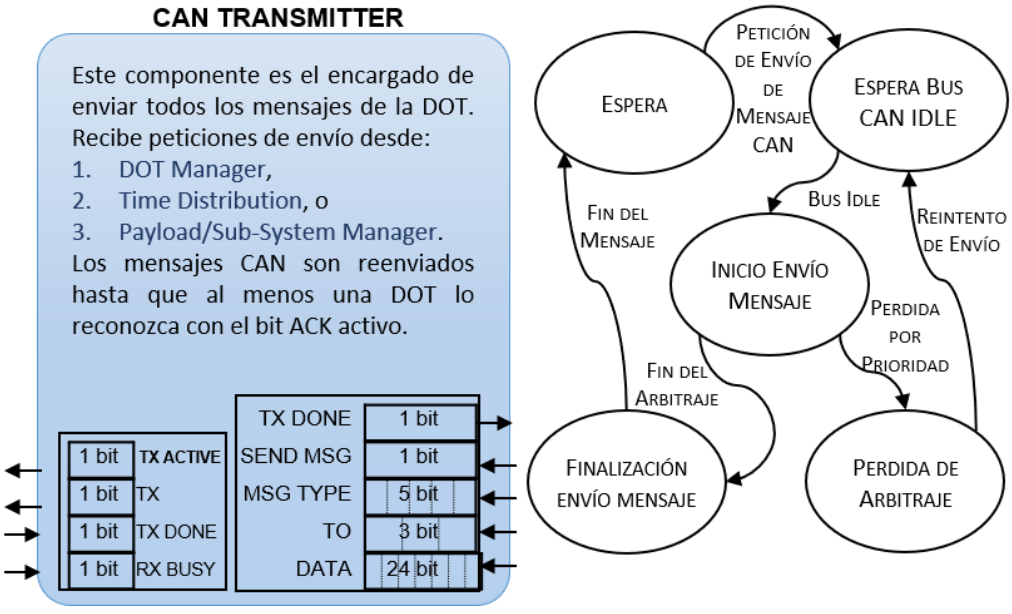


Fig. 11. Diagrama de estados e interfaces del componente colaborativo "CAN Transnutter".

Por último, el componente encargado de gestionar el proceso de mantenimiento en tiempo real (*RTM*) se llama "Time Distribution". Se muestra un esquema de sus

interfaces en la Fig. 12, y una descripción más detallada es descrita en la sección de Procesos Colaborativos.

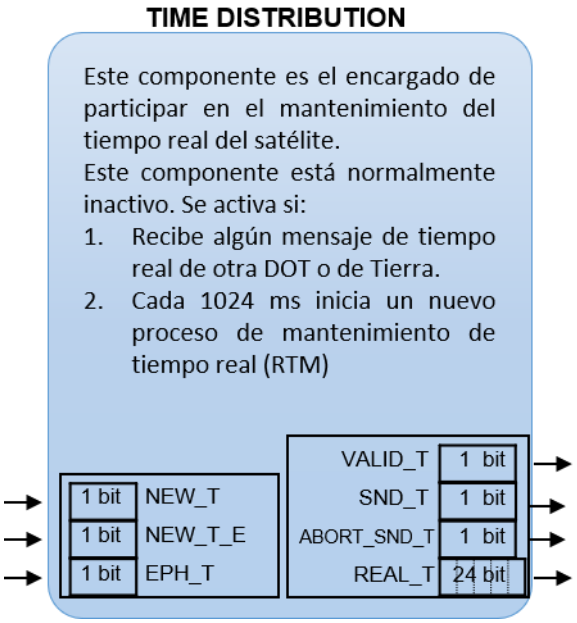


Fig. 12. Diagrama de interfaces del componente colaborativo "Time Distribution".

La descripción del componente "*PL SS MANAGER*", es diferente para cada unidad, y es descrita más adelante en cada una de las subsecciones donde se desgrana la información de cada DOT y sus cargas útiles y/o subsistemas asociados.

Para poder entender mejor el funcionamiento interno del OBC, es importante describir los mensajes con los que cuentan las unidades distribuidas para poder comunicarse entre ellas. En la siguiente sección se muestran los mensajes utilizados por el OBC.

4.3.1 MENSAJES CAN DEL COMPUTADOR

Los mensajes transmitidos desde cualquier *DOT* en el BUS CAN contienen una etiqueta identificativa única que además determina la prioridad del mensaje. Cuando dos o más *DOT* pretenden enviar un mensaje al mismo tiempo, sólo el mensaje de mayor prioridad gana el acceso al bus, mientras que los mensajes de menor prioridad se retransmitirán automáticamente en los siguientes ciclos de bus. El OBC utiliza la versión estándar de

CAN BUS (Parte A), pero con una reducción en el número máximo de bytes a enviar en el campo de datos:

- el identificador de mensaje está formado por 11 bits, con la estructura mostrada en la Tabla 1. Estructura del Identificador de un mensaje CAN. (Tabla 1)
- se limita el número máximo de bytes de datos a tres en lugar de ocho.

El identificador de un mensaje CAN está formado por 11 bits y tiene la siguiente estructura:

Tabla 1. Estructura del Identificador de un mensaje CAN.

IDENTIFICADOR DE UN MENSAJE CAN										
Bit 10 (MSB)	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0 (LSB)
TIPO DE MENSAJE (MT)					*ID de ORIGEN (OI)*			*ID de DESTINO (TI)*		

Tipo de Mensaje (MT)

Este campo del identificador de un mensaje CAN ocupa los bits del 10 al 6, lo que hace un total de 5 bits y su valor determina el tipo de mensaje CAN. Se definen 2 grupos de mensajes destinados únicamente al OBC. Estos son: Distribución de Tiempo Real (TD) y Estado de OBC (ST).

ID de Origen (OI)

Este campo del identificador de un mensaje CAN ocupa los bits del 5 al 3, lo que hace un total de 3 bits y su valor determina la unidad que ha generado el mensaje. Los valores definidos para este campo ordenados de mayor a menor prioridad en el bus pueden verse en la Tabla 2.

Tabla 2. Posibles valores para los campos Origen y Destino del identificador de un mensaje CAN del OBC.

ORIGIN ID	*ORIGIN ID CODE (BINARY)*
DOT 6	**000**
DOT 7 (TPA)	**001**
EPH	**010**
DOT 5	**011**
DOT 1	**100**

DOT 2	**101**
DOT 3	**110**
DOT 4	**111**

ID de Destino (DI)

Este campo del identificador de un mensaje CAN ocupa los bits del 2 al 0, lo que hace un total de 3 bits y su valor determina la unidad que ha generado el mensaje. Aquellos mensajes destinados a ser escuchados por todas las unidades del bus utilizarán como ID de Destino el mismo que el de Origen. Los valores definidos para este campo ordenados de mayor a menor prioridad en el bus pueden verse en la Tabla 2

Como se ha descrito anteriormente, el conjunto de mensajes de CAN definidos para el OBC se divide en dos grupos: distribución de tiempo real (TD) y estado de OBC (ST).

4.3.1.1 Mensajes de Distribución de Tiempo Real

Los mensajes TD han sido concebidos para garantizar el mantenimiento del tiempo real del satélite de una forma distribuida entre todas las *DOT* del OBC y el EPH. Se han creado tres tipos de mensajes necesarios para mantener el tiempo real en el sistema.

- **TD_EPH**: Este mensaje sólo será enviado por la unidad EPH. Con este mensaje la EPH indica al resto de unidades cuál es el tiempo real del satélite recibido desde Tierra. A partir de la recepción de este mensaje el resto de unidades empezarán a mantener el tiempo real del satélite. El campo "TIEMPO REAL" es el tiempo en milisegundos dentro de la hora en la que se encuentra el satélite.
- **TD_BROADCAST**: Este mensaje sólo será enviado por las unidades *DOT* y no por el EPH. Este mensaje será enviado cada 1024 ms. La *DOT* que primeramente alcance el milisegundo 1024 o en caso de empate la de mayor prioridad (por identificador de CAN), será la única que mande el mensaje de TD_BROADCAST.
- **TD_VOTING**: Este mensaje será enviado por cualquier *DOT* que tras recibir y comparar un mensaje de TD_BROADCAST de otra *DOT*, posea un tiempo real distinto al tiempo recibido. Además, toda *DOT* que reciba un mensaje de

TD_VOTING mandará otro mensaje de TD_VOTING con su tiempo real propio[7].

Todos los mensajes de distribución de tiempo real tienen la estructura mostrada en la Tabla 3:

Tabla 3. Estructura de los mensajes de Distribución de Tiempo Real.

IDENTIFICADOR (11 BITS)											***DATOS (24 BITS)***								
Bit 10 (MSB)	9	8	7	6	5	4	3	2	1	Bit 0 (LSB)	Bit 23 (MSB)	22	21	20	…	19	18	17	Bit 0 (LSB)
TIPO MENSAJE (MT)					***ID ORIGEN (OI)***			***ID DESTINO (DI)***			***SIN VALOR***	***RTO***	***TIEMPO REAL***						

El campo *RTO* (Real Time Overflow) especifica si ha habido desbordamiento del tiempo mantenido por las *DOT*. El valor será 0 si no ha ocurrido el desbordamiento, y 1 en caso afirmativo. Debido a que el número de bits utilizados para el tiempo real es de 22, las *DOT* serán capaces de mantener un tiempo relativo a 1 hora, 9 minutos y 54 segundos. Una vez se alcance este máximo, el bit *RTO* se pondrá a 1, y solo será reseteado cuando el EPH mande un mensaje de TD_EPH.

4.3.1.2 Mensajes de Estado del OBC

Este conjunto de mensajes será utilizado por el OBC para conocer el estado particular de cada *DOT*, así como para proporcionar información útil y concreta de los subsistemas y cargas útiles conectados a ellas. Se han definido cuatro mensajes distintos:

- Wake-Up
- Alive
- Auto-Sleep
- Abort Auto-Sleep

[7] El proceso de mantenimiento de tiempo real está descrito en profundidad en la sección Procesos Colaborativos.

Los mensajes de estado del OBC serán generados de forma automática (y no bajo petición) por cada *DOT*, a excepción del mensaje Abort Auto-Sleep, que podrá ser generado por cualquier unidad hacia otra si necesita que el siguiente ciclo se mantenga despierta. Cada terminal (*DOT* y EPH) enviará cada segundo un mensaje con su estado actual (Wake-Up, Alive o Auto-Sleep), por lo tanto, sólo podrá enviar un mensaje por segundo.

4.3.1.2.1 Wake-Up

Este mensaje será enviado por toda unidad (*DOT* y EPH) cada vez que se encienda.

4.3.1.2.2 Alive

Este mensaje será enviado por toda unidad (*DOT* y EPH) siempre y cuando no se acabe de encender, o pretenda apagarse en el inicio del siguiente ciclo. El mensaje de Alive indica al resto de unidades que quien lo manda está encendida, y contiene información útil sobre: si la *DOT* posee el tiempo real y por lo tanto está en condiciones de formar parte del mantenimiento de tiempo real distribuido; información sobre si las cargas útiles o subsistemas asociadas a ellas están encendidas o apagadas; y por último, cuanto tiempo lleva encendida la unidad (en segundos). Toda esta información es útil para el resto del OBC para poder evaluar el estado de salud de las otras unidades, y para mantener las tareas críticas colaborativas.

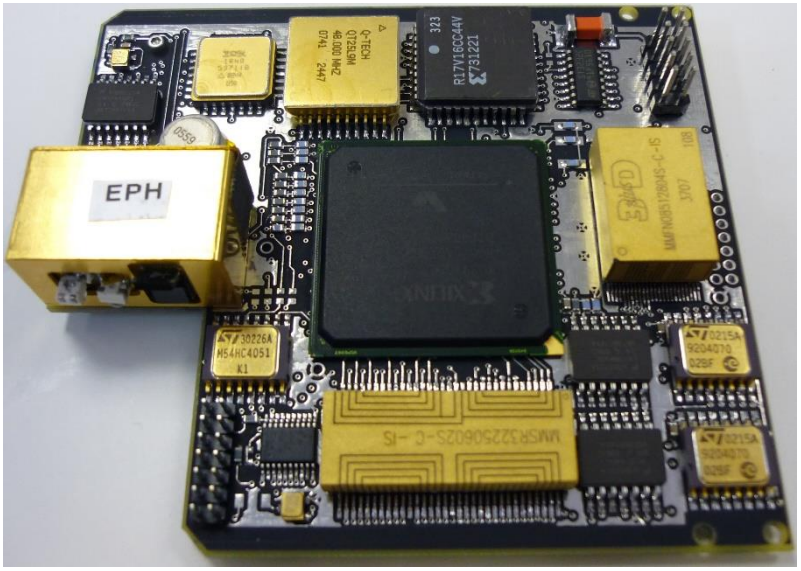
4.3.1.2.3 Auto-Sleep

Este mensaje es utilizado por todas las unidades para señalizar el propósito de apagarse en el siguiente ciclo. Los apagados de las *DOT* han de ser controlados, y por lo tanto, para asegurarse que el sistema no se queda con un número de *DOT* insuficientes para mantener las tareas críticas, sólo una *DOT* podrá apagarse en cada ciclo. El algoritmo de decisión de apagado está explicado con detalle en la sección Algoritmo de Apagado.

Una vez se han explicado los mensajes con los cuales las diferentes unidades del computador distribuido se comunican, se procede a detallar las funcionalidades

concretas de cada unidad. La única unidad que no implementa los algoritmos colaborativos, pero sí se nutre de sus resultados es el *EPH*, descrito a continuación.

4.3.2 UNIDAD 0: ENHANCED PROCESSING HARDWARE



Esta unidad es la encargada de hacer de interfaz con el subsistema de comunicaciones con Tierra (*TTC*), así como albergar el subsistema de software de vuelo (*OBSW*). Debido a la complejidad de ambos subsistemas esta unidad está comandada por una FPGA Virtex-II 1000 (XQ2V1000) de Xilinx [Xilinx Inc. 2007]. Consta de un procesador de 32 bit embebido en la FPGA. Además, corre un sistema operativo que dota al *OBSW* de los recursos necesarios para ejecutar la operación del satélite. Esta unidad es así misma la encargada de almacenar todas las telemetrías generadas por el satélite para su posterior bajada al paso por la estación terrena. Para ello, esta unidad posee una memoria Flash de almacenamiento no-volátil de 4 Gbit. La Fig. 13 muestra el diagrama de bloques del *EPH*:

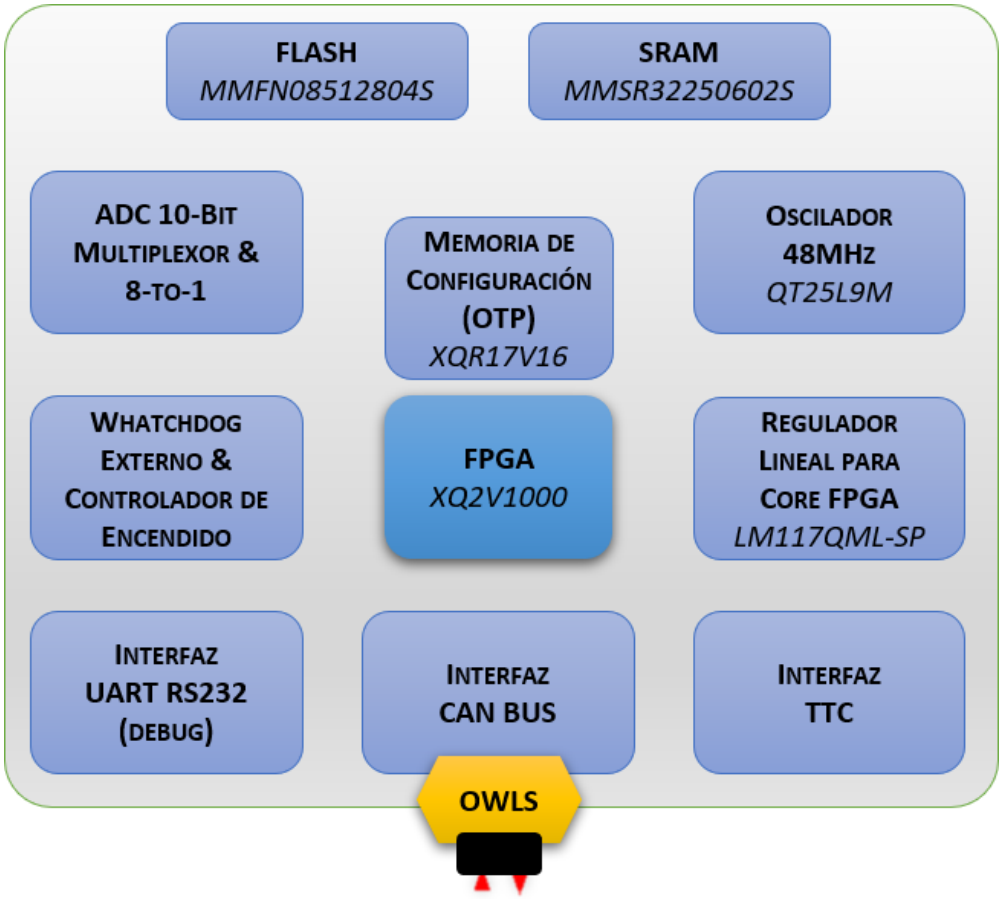


Fig. 13. Diagrama de bloques de la unidad 0: EPH.

El procesador embebido en la FPGA cuenta con una serie IP Cores para controlar los periféricos del procesador. La mayoría de ellos han sido creados ad-hoc para este computador utilizando diferentes técnicas de endurecimiento frente a radiación, con el objetivo de minimizar los errores inducidos por SEU. Sin embargo, el EPH no forma parte en las tareas colaborativos del computador, por lo que el objetivo era maximizar el tiempo operativo del software de vuelo. En la Fig. 14 se puede ver de forma detallada los distintos IP Cores implementados dentro de la FPGA, y los dispositivos externos a los que se conectan. En los siguientes párrafos se describen las funcionalidades de los IP Cores diseñados y desarrollados para el *OBC* de OPTOS.

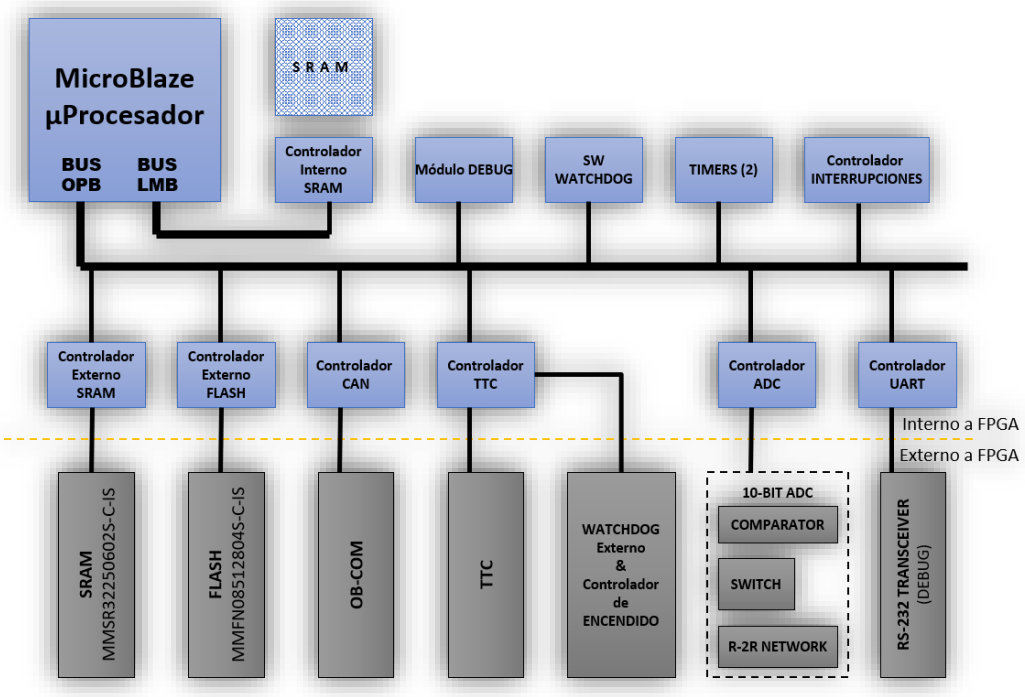


Fig. 14. Diagrama de bloques del procesador del EPH y sus periféricos.

El EPH se comunica con el resto de unidades del OBC a través del "CAN Controller", que implementa el standard del protocolo CAN BUS [GmBH 1998]. La capa física del CAN tiene la singularidad, como se ha explicado antes, de hacerse de forma inalámbrica con señales ópticas difusas. Para ello se utiliza un módulo miniaturizado equipado con transmisores, receptores, y la electrónica de control y procesado necesaria para la implementación de la capa física del CAN [Rivas et al. 2017].

El EPH implementa una interfaz dedicada al subsistema de comunicaciones. Este subsistema está basado en un ASIC para la recepción de telecomandos desde Tierra, con una interfaz serie dedicada. Para la transmisión de telemetrías a Tierra, cuenta con una serie de señales de control (encendido, apagado, recepción y transmisión), así como una señal codificada en Manchester ['Manchester Coding' 2018] de envío directo a través de la etapa analógica del transpondedor. Además, el *TTC* cuenta con tres señales analógicas para informar del estado de salud tanto de la unidad como de las comunicaciones. En el siguiente diagrama se puede ver la interfaz implementada para el control del *TTC*:

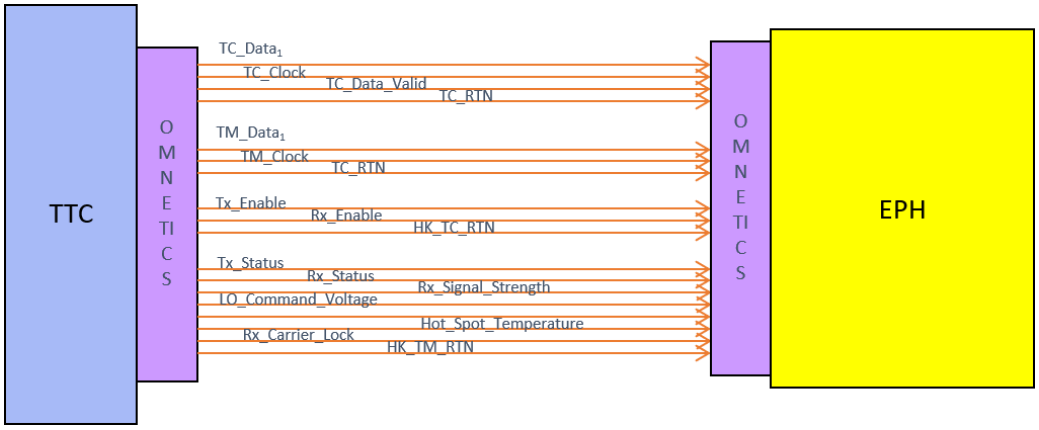


Fig. 15. Interfaz entre TTC y EPH.

Otro de los IP Cores implementados para mejorar la fiabilidad del sistema frente a radiación es el "External Watchdog & Power-up Controller". La primera y más sencilla es la de mantener alerta al sistema externo de watchdog fabricado con componentes Rad-Hard. Lo primero que se ha de proteger del EPH es la posibilidad de quedarse en un estado del cual no pudiera salir, debido a un efecto inesperado provocado por *SEU*. Por ello se crea este módulo externo basado en componentes resistentes a la radiación, con el objetivo de identificar si el procesador estaba parado, y en última instancia, provocar un apagado y un encendido de este transcurrido un tiempo máximo. Por otro lado, este módulo permitía el reinicio del propio EPH, así como el reinicio total del satélite, mediante una línea dedicada al subsistema de potencia (*PDU*). El riesgo de implementar esta funcionalidad dentro de la FPGA, sensible a *SEU*, es alto. Por lo tanto, se ha implementado el módulo utilizando técnicas de redundancia con autocorrección. Este método garantiza que no haya un único punto de fallo. Además, la realimentación de los módulos votadores permite al circuito corregir autónomamente el camino de dato erróneo. En la Fig. 16 podemos ver como el circuito de protección corrige el problema cuando ocurre un cambio de bit en cualquiera de los caminos de datos:

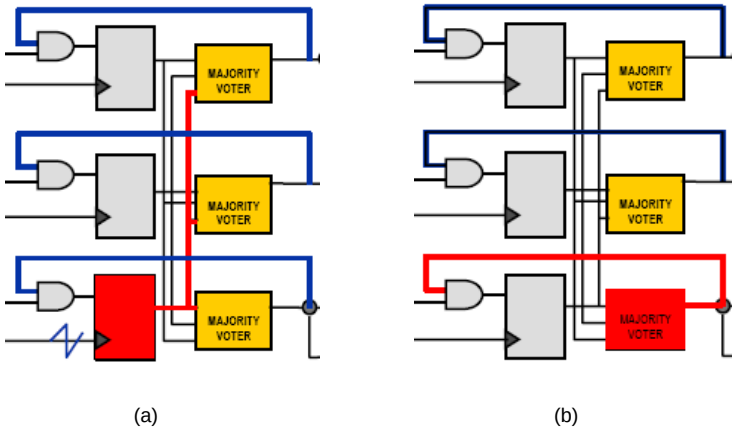


Fig. 16. Ejemplo de circuito de protección frente a SEU con autocorrección.

Con esta técnica de endurecimiento, se controla de forma segura el apagado de la unidad, así como del satélite cuando esto último sea comandado desde Tierra.

4.3.3 UNIDAD 1: DOT EPS1

Esta unidad forma parte del sistema colaborativo de las *DOT*. La unidad se encuentra en la parte inferior del satélite, siendo la tarjeta más cercana al sensor solar situado en el eje +Z, y al sensor solar situado en el eje +Y. Además, también da soporte a una de las dos tarjetas del subsistema de

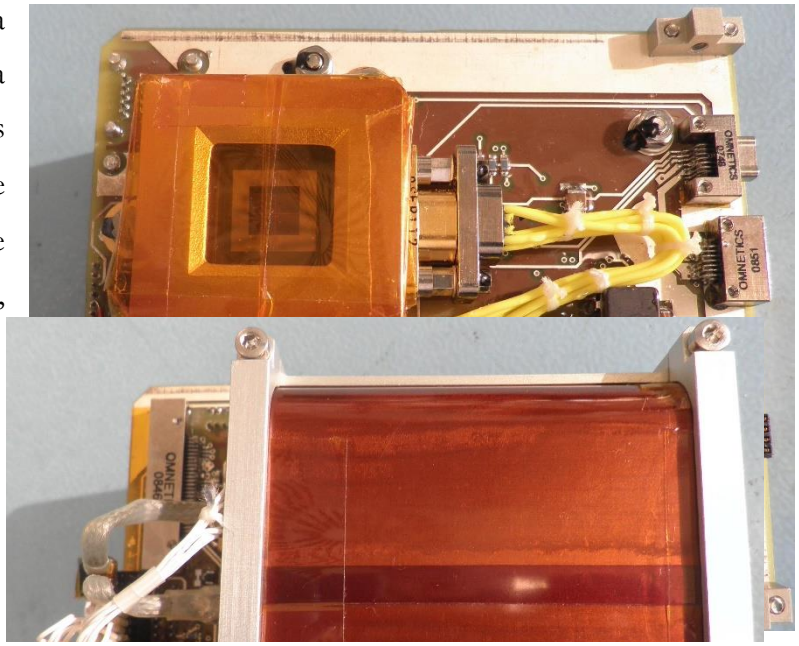


Fig. 18. DOT1, con Sensor Solar +Z y conexioón a paneles solares.

Fig. 17. Batería del satélite controlada por DOT1. Modulos OWLS espía (debug) y de DOT1.

potencia, la tarjeta *EPS1*. Esta tarjeta es la encargada de recibir la corriente de los paneles solares del satélite, y utilizarla tanto para cargar batería como para distribuir la línea primaria (4V) de potencia en el satélite. Por último, está unidad es la encargada del despliegue de antenas y paneles solares. En ella están también situados los conectores de pruebas del satélite, así como la unidad OWLS espía, para poder recibir y evaluar el funcionamiento de las comunicaciones internas, y del sistema colaborativo de las *DOT*.

Las tareas específicas de la DOT1 se implementan en el componente "*PL SS MANAGER*", el cual se encarga de dar soporte a los siguientes dispositivos externos:

- Paneles Solares ±X
- Paneles Solares ±Y
- Sensor Solar -Z
- Sensor Solar +Y
- EPS1

El siguiente diagrama de bloques Fig. 19 muestra las interfaces de la DOT1:

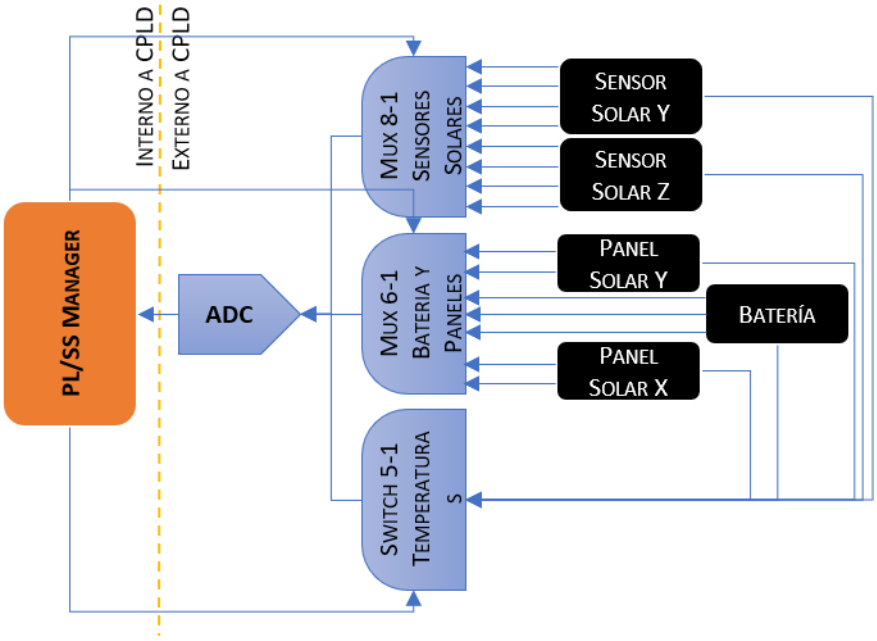


Fig. 19. Interfaz de DOT1 con el susbsistema de potencia.

4.3.4 UNIDAD 2: DOT MGM & ODM1

Esta unidad forma parte del sistema colaborativo de las ***DOT***. La unidad se encuentra en la parte inferior del satélite, justo encima de la tarjeta de ***EPS1*** y debajo de DOT4. La DOT2 es la encargada de la operación de un magnetómetro y de una carga útil de radiación. El magnetómetro o MGM, es el sensor de campo magnético perteneciente al subsistema de ***ADCS***.

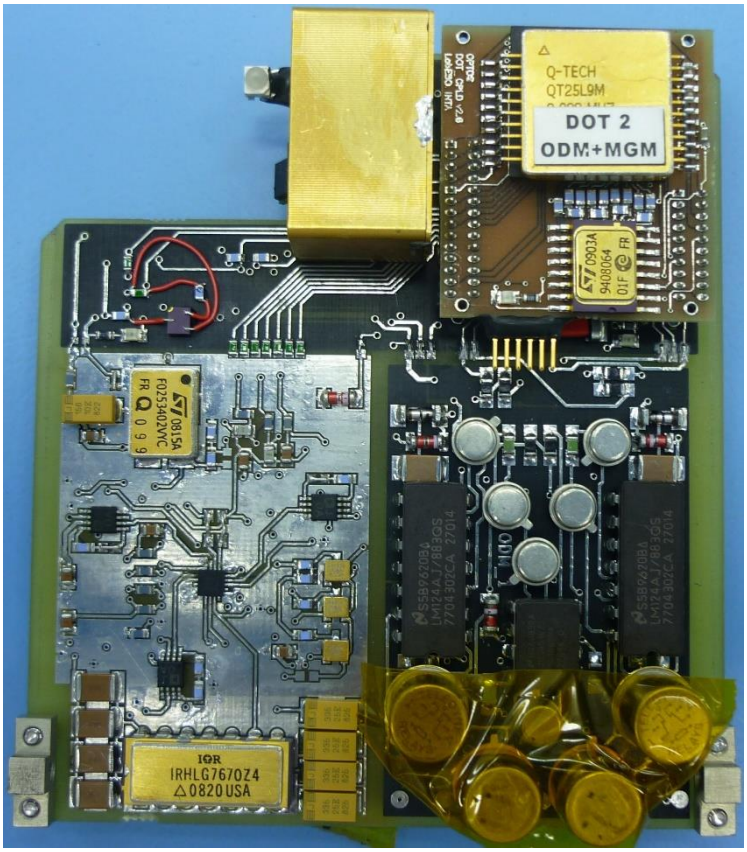


Fig. 20. DOT2 con ODM1 abajo a la derecha y MGM abajo a la izquierda.

Este sensor es capaz de dar medidas precisas de campo magnético en los tres ejes de satélite (X, Y, Z). Al formar parte del subsistema de ADCS, y no ser una carga útil, esta ***DOT*** realiza una medida constante (a una frecuencia de 0,5 Hz) del sensor, y envía a través del bus CAN los datos obtenidos del MGM. Por otro lado, el *OPTOS Dose Monitoring* 1 (***ODM***1) es un sensor basado en tecnología ***RadFET***, capaz de absorber la radiación de las partículas con el fin de monitorizar la dosis total recibida en el propio satélite. De hecho, OPTOS se provee de dos sensores (ODM1 controlado por esta ***DOT***, y ODM2 controlado por la DOT7 TPA) para mejorar la fiabilidad de los datos obtenidos.

El siguiente diagrama de bloques muestra las interfaces de la DOT2 con las cargas útiles:

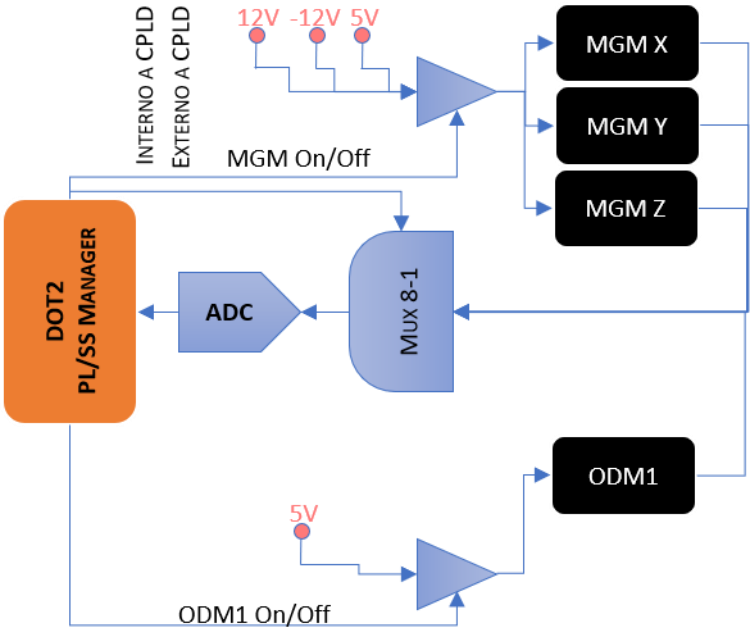


Fig. 21. Interfaz de DOT2 con MGM y ODM1.

4.3.5 UNIDAD 3: DOT GMR

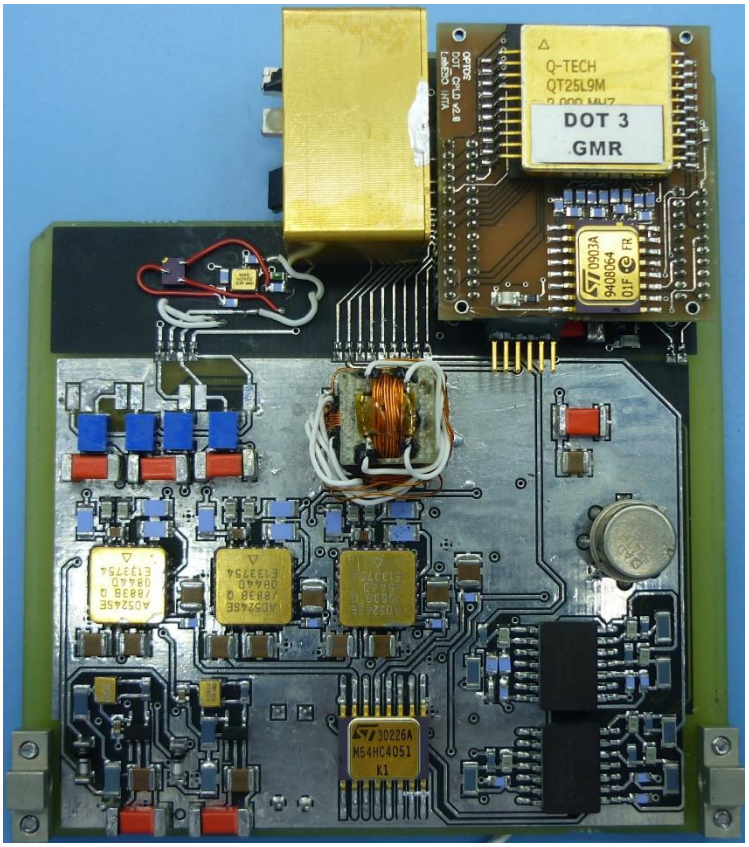


Fig. 22. DOT3 con GMR.

Esta unidad forma parte del sistema colaborativo de las *DOT*. La unidad se encuentra en la parte media del satélite, justo encima de la tarjeta de DOT2 y debajo de DOT4. Esta tarjeta es la encargada de operar una carga útil con un segundo sensor de campo magnético. En este caso, el magnetómetro, es una versión experimental de una Magneto Resistencia Gigante (*GMR*) [Michelena 2013]. El dispositivo *GMR* se basa en el efecto del cambio en la resistencia eléctrica que experimenta una estructura multicapa en presencia de un campo magnético. Esta carga útil, al igual que el magnetómetro MGM, posee tres canales (X, Y, Z) que pueden ser leídos tanto con corriente positiva como negativa- Además, posee otras señales de salud como la temperatura y el voltaje de referencia. En el siguiente diagrama de bloques *(*Fig. 23*)* muestra las interfaces de la DOT3:

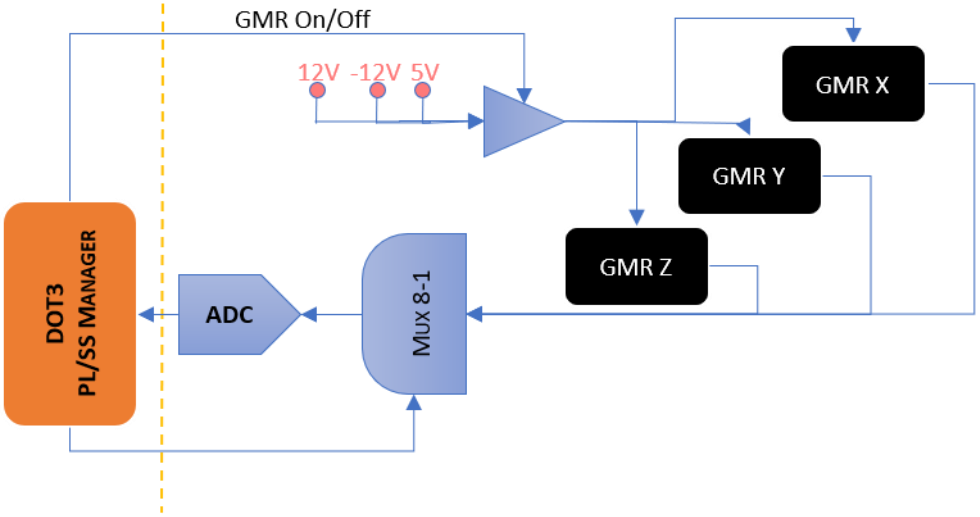


Fig. 23. Interfaz de DOT3 con GMR.

4.3.6 UNIDAD 4: DOT FIBOS

Esta unidad forma parte del sistema colaborativo de las *DOT*. La unidad se encuentra en la parte media del satélite, justo encima de la tarjeta de DOT3 y debajo de DOT5. La DOT4 está encargada de proveer del control y la adquisición necesarios a la carga útil *FIBOS*. FIBOS consiste en un láser con una fibra que está conectada a un

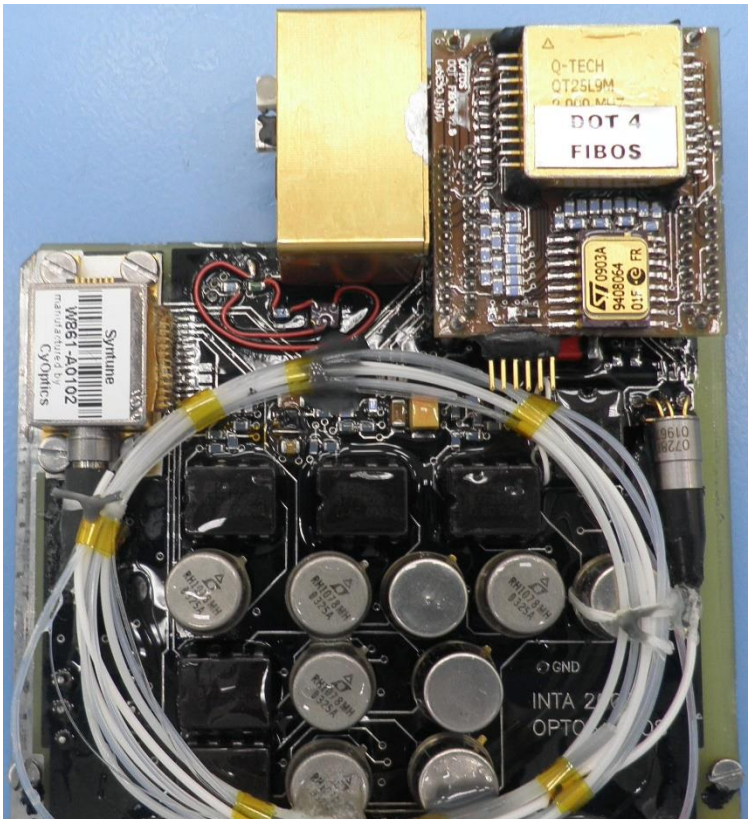


Fig. 24. DOT 5 con FIBOS. Se puede observar el laser en la izquierda y la fibra de Bragg en la parte inferior.

fotodiodo con una fibra de Bragg en el medio [Heredero et al. 2008]. El objetivo de FIBOS es medir la temperatura estudiando las longitudes de onda de un láser que viaja a través de las rejillas de Bragg. La medida obtenida se contrasta con la temperatura detectada de un termistor en la misma ubicación.

Para poder medir en condiciones óptimas, el láser ha de estar totalmente estabilizado en temperatura, y para ello la *DOT* implementa un algoritmo de control mediante la utilización del Peltier. El siguiente diagrama de bloques Fig. 25 muestra las interfaces de la *DOT* con FIBOS:

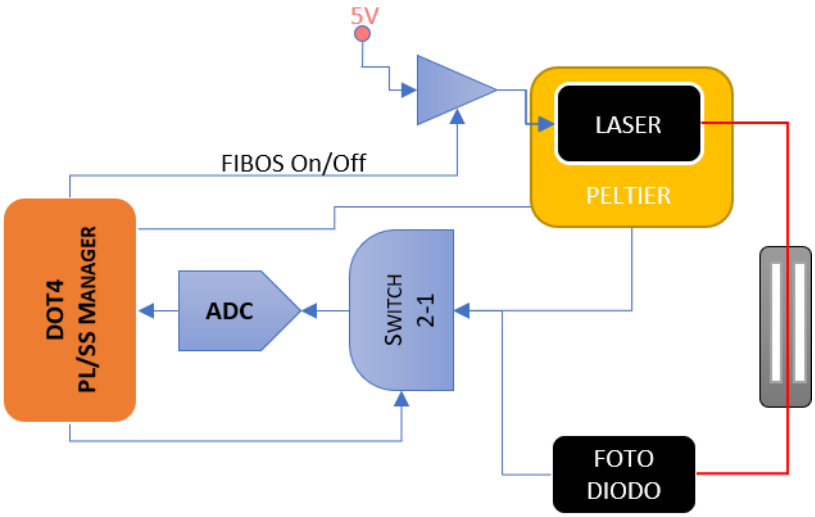


Fig. 25. Interfaz de DOT4 con FIBOS.

4.3.7 UNIDAD 5: DOT RW

Esta unidad forma parte del sistema colaborativo de las *DOT*. La unidad se encuentra en la parte media del satélite, justo encima de la tarjeta de DOT4 y debajo del *EPH*. Esta unidad colaborativa está encargada de controlar la rueda de reacción (*RW*). Este actuador forma parte del subsistema de *ADCS*. Para controlar esta unidad, la *DOT* implementa un protocolo de comunicaciones UART, con un sistema propio de telemetrías y telecomandos. Mediante este protocolo, la DOT5 envía al controlador de la rueda de reacción la velocidad y aceleración deseada. Este dato, es a su vez enviado por el EPH a la DOT5, mediante el software de *ADCS*. Por otro lado, la DOT5 recibe las telemetrías generadas por la propia rueda, y se lo manda al software de *ADCS* mediante mensajes CAN bajo petición.

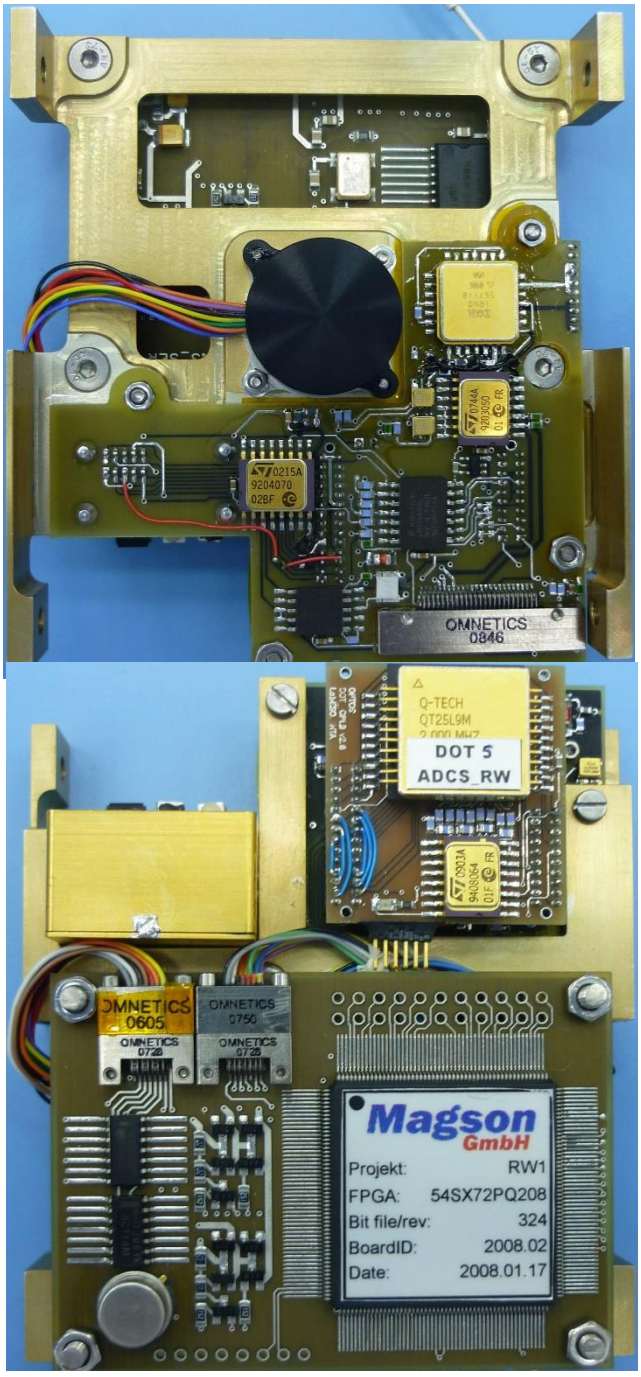


Fig. 26. DOT5 con la rueda de reacción. En la parte inferior se encuentra el controlador, y en la superior la propia rueda.

El siguiente diagrama de bloques Fig. 27 muestra las interfaces de la DOT5:

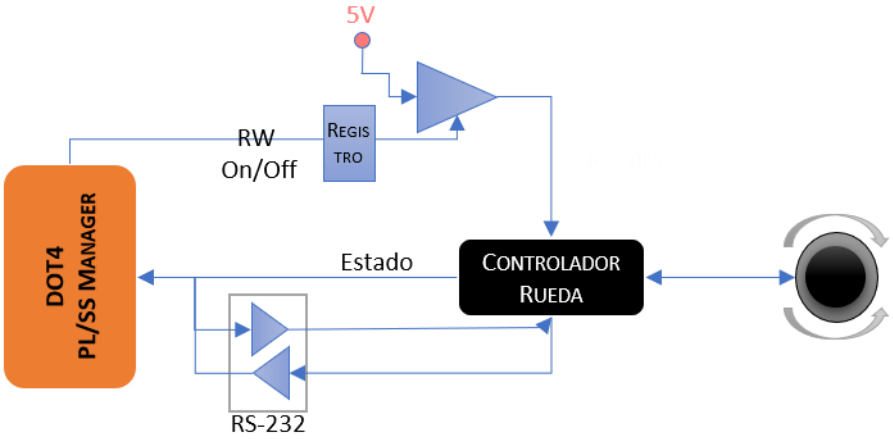


Fig. 27. Interfaz de DOT5 con Rueda de Reacción del subsistema de ADCS.

4.3.8 UNIDAD 6: DOT EPS2

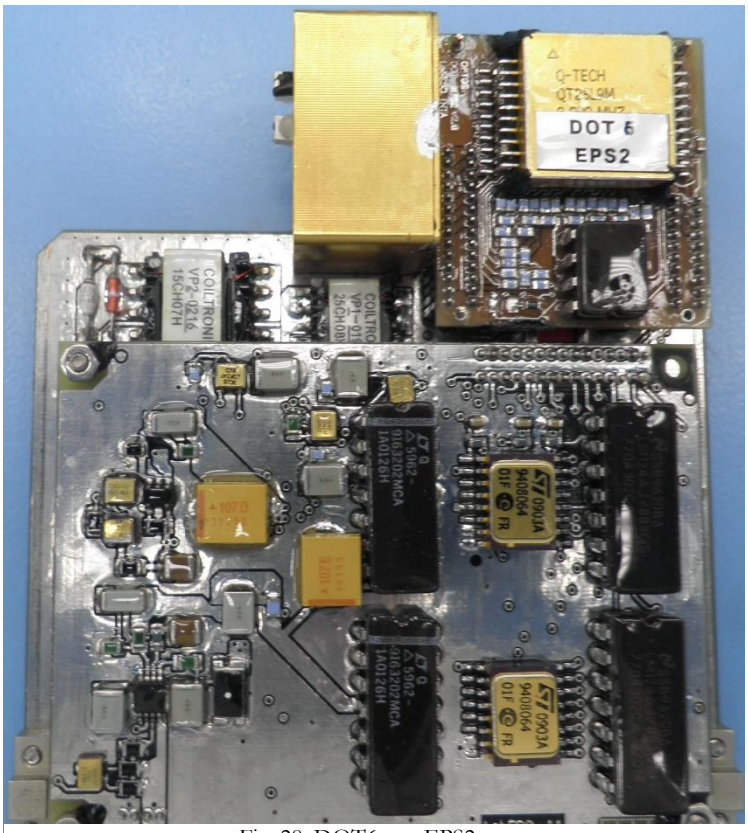


Fig. 28. DOT6 con EPS2.

Esta unidad forma parte del sistema colaborativo de las *DOT*. La unidad se encuentra en la parte alta del satélite, justo encima de la tarjeta de *EPH* y debajo de la *TPA* y el *TTC*. Esta unidad es la encargada de dar soporte a la tarjeta EPS2 del subsistema de potencia del satélite. La *DOT* tiene la capacidad de controlar los

transistores de corte de las alimentaciones secundarias del satélite. Así mismo, se encarga de almacenar los valores de corte de corriente máxima de cada línea, y de leer sus valores periódicamente, con el objetivo de proteger a las cargas útiles de posibles sobreconsumos. EPS2 suministra las siguientes lineas de potencia secundarias al satélite:

- 5,0V y 3,3V para plataforma
- 5,5V y 4,0V para *TTC*.
- 5,0V, 12,0 y -12,0V para carga útil.

El siguiente diagrama de bloques Fig. 29 muestra las interfaces de la DOT6 con EPS2:

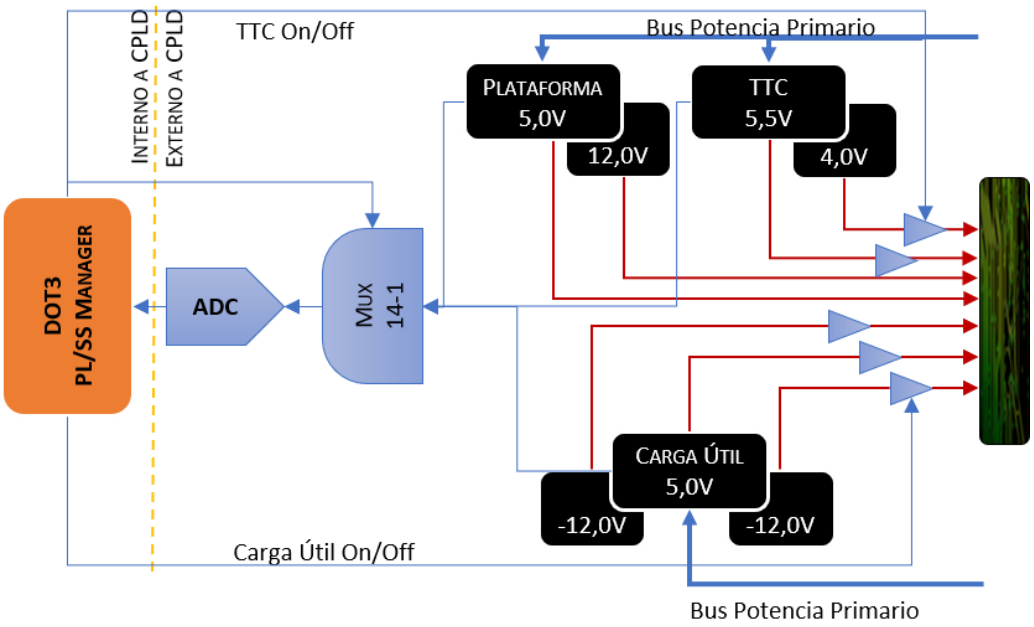


Fig. 29. Interfaz de DOT6 con EPS2.

4.3.9 UNIDAD 7: DOT TPA

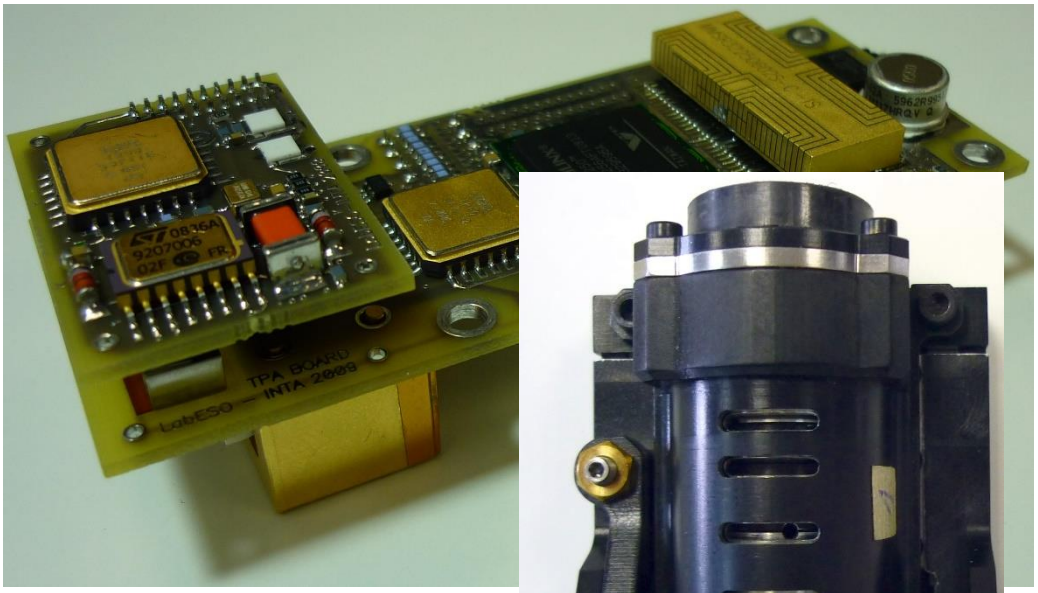


Fig. 30. DOT7 TPA con los conectores para

Esta unidad forma parte del sistema colaborativo de las *DOT*. La unidad se encuentra en la parte alta del satélite, justo encima de la tarjeta de DOT6 y sólo por debajo de la tarjeta del magneto torque Z Además, se encuentra al lado del *TTC*, el cual es una unidad cerrada en una caja metálica para mejorar su aislamiento electromagnético. A esta unidad se le ha dado el nombre de *TPA*, porque da soporte a un gran número de dispositivos colocados en la parte alta del satélite. Para empezar, da soporte al *ODM*2, que es una carga útil igual que el *ODM*1 que hemos visto en la DOT2, con una situación física distinta para mejorar la precisión de dosimetría de partículas en el satélite. Además, también da soporte a *APIS*, una carga útil de observación de la Tierra, con una pequeña óptica basada en un conjunto opto-mecánico atermalizado, para evitar las deformaciones propias de los materiales debido a las variaciones de temperatura. El sensor CMOS de la cámara APIS es un sensor pancromático de 640x480 pixeles, con

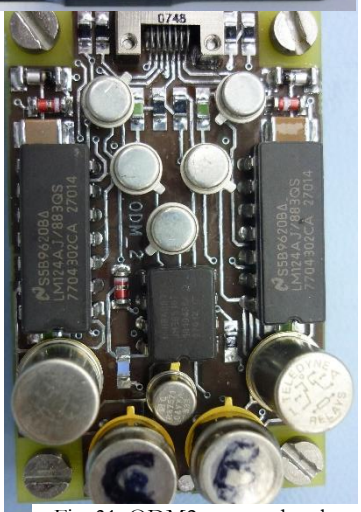


Fig. 31. ODM2 conectado a la DOT7 TPA.

una profundidad de 10-bit. Esto hace, que las imágenes generadas por la carga útil ocupen un total de 3 Mbits. Como la memoria interna de la FPGA de la DOT7 no posee tal cantidad de memoria

En relación con APIS, el satélite cuenta con un obturador en la tarjeta superior del mismo, con salida directa al Espacio. Este obturador es controlado también por TPA. Por último, la TPA da soporte a un sensor y tres actuadores del subsistema de *ADCS*. El sensor de presencia solar es el encargado de avisar al sistema de control del ADCS, que el sensor de imagen de APIS se puede estar viendo expuesto al Sol directo. Esta situación no puede prolongarse en el tiempo, o sino el sensor puede quedar dañado. Los magneto-torques, son las herramientas fundamentales para poder cambiar la actitud del satélite en el Espacio. OPTOS cuenta con 3 de ellos, uno para cada eje del satélite, y todos son controlados por la TPA.

Fig. 32. Cámara APIS con sensor CMOS y óptica atermalizada.

El siguiente diagrama de bloques (Fig. 33*)* muestra las interfaces de la DOT7 con sus dos cargas útiles y los magneto-torques:

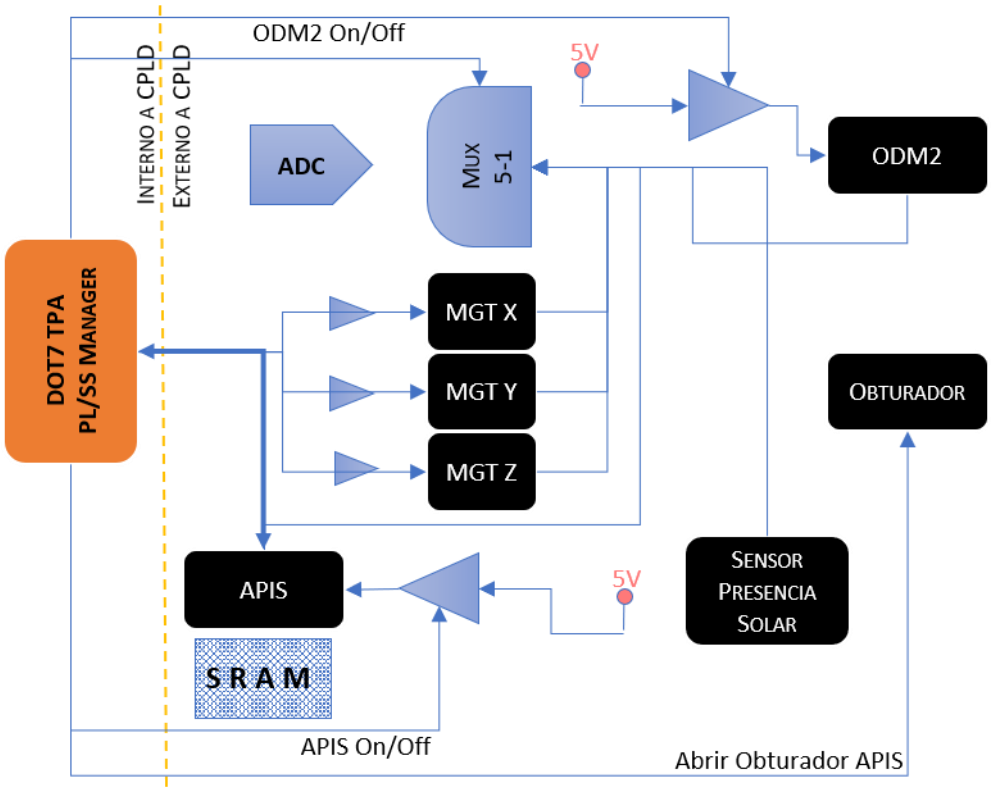


Fig. 33. Interfaz de DOT7 con APIS, ODM2 y subsistema ADCS.

Las tareas específicas de la DOT7 se implementan en el componente "*PL SS MANAGER*", el cual se encarga de dar soporte a los siguientes dispositivos externos:

- ODM2
- APIS
- Obturador del sensor de imagen.
- Maneto-torques X, Y y Z
- Sensor de Presencia Solar

En este apartado, se ha descrito la funcionalidad unitaria de cada *DOT* y del EPH, dentro del conjunto de capacidades del OBC. En el siguiente apartado, se describen las técnicas

colaborativas aplicadas al computador, con el objetivo de endurecer el sistema frente a radiación, y de mantener las tareas críticas del satélite de forma fiable.

4.4 Algoritmos de Endurecimiento

Esta sección describe las técnicas de endurecimiento colaborativo, aplicadas a la arquitectura distribuida del computador de vuelo, con especial énfasis en la implementación de las tareas críticas. Estas técnicas innovadoras de endurecimiento deben entenderse como una filosofía de diseño. De hecho, la solución se presenta como las técnicas y procesos implementados a nivel unidad. El endurecimiento colaborativo es de aplicación a cualquier sistema distribuido con capacidad para cumplir los siguientes requisitos:

- El sistema debe estar compuesto por al menos 4 nodos distintos.
- La inteligencia individual puede ser realizada unitariamente; sin embargo, las tareas críticas deben ser realizadas en colaboración.
- Cada tarea crítica debe ser implementada por al menos 4 nodos diferentes. Sin embargo, no todos los nodos deben implementar todas las tareas críticas.
- Cada nodo debe poder recuperarse de cualquier error después de un período de tiempo predefinido (este tiempo dependerá de la aplicación). Esta recuperación podrá realizarse mediante un ciclo de reinicio, o de cualquier otra forma que se garantice la ausencia de errores provocados por *SEU*.
- Las comunicaciones entre nodos se deben realizar a través de un bus de comunicación no maestro, donde cada unidad debe recibir datos directamente de todas las demás unidades.
- El protocolo de comunicación debe implementar mecanismos de detección de errores para evitar mensajes falsos en el sistema de colaboración.
- Al menos tres nodos diferentes siempre deben estar ejecutándose para cada tarea crítica.

Una tarea crítica es cualquier tarea que pueda afectar a la fiabilidad del sistema. Potencialmente, cualquier tarea que involucre un algoritmo de decisión podría implementarse en colaboración. El punto clave es que cualquier dato de entrada del algoritmo debe entregarse al bus de comunicaciones y, por lo tanto, estar disponible para todos los nodos al mismo tiempo. Las tareas críticas típicas en aplicaciones espaciales son: mantenimiento del tiempo real, tareas de control de determinación de actitud tales como propagaciones de efemérides y determinación del Sol, y aquellas relacionadas con maniobras de control. En aplicaciones terrestres y aéreas, aquellas relacionadas con unidades de medición inercial (*IMU*), actuadores mecánicos y propulsores y, en general, cualquier algoritmo involucrado en decisiones de seguridad humana.

La solución propuesta se basa en una arquitectura colaborativa distribuida. El computador está compuesto por siete Terminales *OBC* Distribuidos (*DOT*) que operan de manera autónoma y una unidad de Hardware de Procesamiento Avanzado (*EPH*). Cada *DOT* controla la carga útil o subsistema al que está conectada, ejecutando los telecomandos recibidos desde *EPH* y devolviendo la telemetría resultante. Todas las *DOT* se conectan a un bus CAN inalámbrico [Rivas Abalo et al. 2017]. Esta tecnología denominada *OWLS* (Comunicaciones Ópticas Inalámbricas Intra-Satélite, por sus siglas en inglés) se ha desarrollado en el INTA en el marco de varios proyectos de la Agencia Espacial Europea (*ESA*) (ESA *GSP* 16428/02/NL/EC y ESA *TRP* 19545/06/NL/GLC). La topología de CAN facilita la colaboración entre las *DOT*, haciendo que todos los mensajes sean accesibles para todas al mismo tiempo y, por lo tanto, permite que las *DOT* compartan tareas comunes con facilidad. La unidad EPH se encarga de comunicarse con Tierra a través del subsistema de *TTC*, y además alberga al subsistema de software de vuelo.

Las técnicas descritas a continuación, han sido diseñadas debido a que las CPLD que comandan cada unidad *DOT*, pueden verse afectadas por SEU, y por lo tanto provocar fallos funcionales en las mismas. Los algoritmos colaborativos se han diseñado en dos capas de seguridad. La primera capa ha sido creada para maximizar la disponibilidad de cada *DOT* y minimizar la probabilidad de ocurrencia de múltiples errores simultáneos. Para ello, se ha desarrollado un conjunto de medidas y protecciones unitarias. Estas

medidas se han aplicado a nivel de *DOT*, para garantizar la fiabilidad del sistema y el éxito de la misión. Las técnicas de endurecimiento unitario han sido aplicadas en tres niveles diferentes, de acuerdo con la criticidad de la situación a manejar. Estas se describen a continuación.

4.4.1 ALGORITMO DE APAGADO

La primera barrera de protección protege a la *DOT* de la acumulación de errores múltiples, provocados por la ocurrencia de múltiples SEU en el tiempo. La memoria de configuración SRAM de la CPLD es sensible a los SEU. Como se describe en [García et al. 2008], la probabilidad de tener un *SEFI* en una *DOT* aumenta drásticamente cuando se acumulan cambios de bit múltiples en la CPLD. Para minimizar la ocurrencia de SEFI, se ha implementado un algoritmo de apagado programado, haciendo que todos los fallos en la memoria SRAM sean reemplazados por los valores correctos, almacenados en la memoria Flash. En este algoritmo, cada *DOT* se apaga en instantes específicos de tiempo (es decir, cada 2s, 4s, 8s, etc.). Una vez que un *DOT* ha alcanzado su propio tiempo de apagado, intentará apagarse. El endurecimiento colaborativo debe garantizar que al menos tres *DOT* estén siempre activados para ejecutar correctamente las tareas críticas. Por lo tanto, se impone una restricción para asegurar que al menos tres *DOT* diferentes estén siempre activas. El algoritmo desarrollado impone las siguientes reglas:

- Cada *DOT* debe almacenar la cantidad de *DOT* activas durante el último segundo.
- Cada *DOT* que solicite un apagado debe transmitir un mensaje de aviso al resto de *DOT*, al menos 1 segundo antes de apagarse. A este mensaje se le llama AutoSleep.
- La transmisión de AutoSleep, que se envía para avisar al resto de *DOT* de un apagado inminente, incluirá información sobre cuánto tiempo lleva en funcionamiento la *DOT*. Por lo tanto, el mensaje de AutoSleep contendrá un contador que se incrementa cada segundo.
- En el caso de que más de una *DOT* quiera apagarse al mismo tiempo, sólo se podrá apagarse aquella *DOT* que lleve despierta más tiempo (con el contador más alto).

- Si se ha activado más de un *DOT* durante el mismo tiempo (el mismo registro de conteo en un momento dado), sólo se apagará aquella que tenga la prioridad más alta (representada por su ID CAN).
- Si hay al menos otras tres *DOT* activas y un *DOT* cumple con todas las reglas anteriormente descritas, esta *DOT* se apagará.

Cuando una *DOT* decide apagarse, y efectivamente se apaga, un circuito hardware externo encenderá la *DOT* después de un período de tiempo preconfigurado. Cuando la *DOT* se despierta, todos los SEU que pudiera haber acumulado durante la activación anterior desaparecen, garantizado la no acumulación de SEU.

4.4.2 SISTEMA DE VIGILANCIA DE COMUNICACIONES

La segunda barrera de protección se centra en la protección de las comunicaciones. Aprovechando las particularidades del protocolo CAN, cada *DOT* ha de asegurarse que no pasa más de un segundo sin obtener respuesta de alguna otra unidad a la escucha en el bus. Todo mensaje CAN está compuesto por 7 campos (ver Fig. 34)[GmBH 1998].

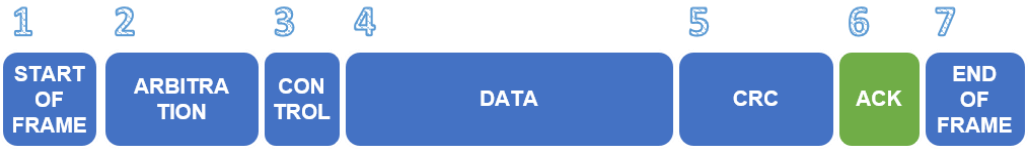


Fig. 34. Formato de mensaje del Bus CAN (Std. A).

El primer campo indica el inicio de una nueva trama de CAN. Seguidamente, el campo de arbitraje es utilizado por las unidades para negociar el control del bus. Solo la *DOT* con el identificador más bajo (prioridad más alta) se hará con el control del bus para la transmisión. Luego, los campos control, datos y *CRC* definen el mensaje que se enviará. La sexta sección es el campo acuse de recibo (*ACK*). Toda *DOT* a la escucha en el bus debe señalar con el bit de acuse de recibo si ha recibido una trama CAN correctamente compuesta, incluso aunque el mensaje no fuera dirigido a ella misma. Las *DOT* utilizan

esta peculiaridad del protocolo para asegurar el correcto funcionamiento de las comunicaciones. Tanto las comunicaciones como el sistema interno de tiempos de cada *DOT* son considerados críticos para llevar a cabo las funciones de endurecimiento colaborativo. Toda *DOT* en el bus debe responder a cada mensaje enviado por los otras *DOT*. En el caso de que una *DOT* no reciba ninguna respuesta durante más de un segundo, esta considerará que bien el módulo de comunicaciones o el de sistema de tiempos están dañados y deben reiniciarse.

La secuencia a seguir por cada *DOT* es la siguiente:

- En el momento de encendido de la *DOT*, se produce un "RESET" para asegurar el estado inicial de todos los registros de esta.
- Un temporizador interno en (TMRu) se establece en 1.5s. Como cada *DOT* debe enviar por lo menos un mensaje Alive o AutoSleep cada segundo, el resto de los *DOT* deben responder en menos de 1.5s.
- Cuando cualquier mensaje de CAN es reconocido correctamente por cualquier *DOT*, el temporizador interno TMRu se restablece en 1.5s.
- Cuando una DOT no recibe ningún acuse de recibo de sus mensajes CAN y el temporizador interno (TMRu) llega a cero, la *DOT* realiza un ciclo de reinicio.

4.4.3 SISTEMA DE VIGILANCIA HARDWARE

La tercera y última barrera de protección unitaria de las *DOT* es un módulo supervisor externo, implementado con dispositivos hardware resistentes a radiación. En caso de que las dos barreras anteriores fallen, y la *DOT* no sea capaz de recuperarse por sí misma de una interrupción funcional, un hardware externo inmune a la radiación está preparado para producir un ciclo de reinicio a la *DOT*.

Cada vez que el hardware externo enciende la *DOT*, se establece un temporizador interno de cuenta regresiva con una cantidad máxima de tiempo que la *DOT* puede estar sin apagarse. Este temporizador solo se restablece mediante un ciclo de apagado-encendido de la *DOT*. Esta situación puede ocurrir por una de estas tres razones:

- El EPH comanda un apagado completo de satélite.
- La propia *DOT* solicita un apagado automático, bien porque el Sistema de Vigilancia de Comunicaciones haya detectado un error, bien porque el Algoritmo de Apagado así lo indique.
- El temporizador interno consume el tiempo máximo de encendido de la *DOT*.

Este sistema asegura que una *DOT* no pueda estar inoperativa más tiempo que el máximo establecido por el temporizador externo.

Las barreras de protección unitaria que se acaban de explicar tenían como objeto maximizar el tiempo operativo de las *DOT* de forma unitaria, a nivel dispositivo. La sección siguiente explica de forma detallada los procesos colaborativos implementados en el computador de OPTOS, para garantizar la fiabilidad de las tareas críticas implementadas por este.

4.4.4 PROCESOS COLABORATIVOS

El mantenimiento del tiempo real es una tarea crítica en todos los sistemas espaciales [Washington 2000]. Esta tarea generalmente es implementada bien por un componente *RadHard* que mantiene específicamente el tiempo real o por un receptor del Sistema Global de Navegación por Satélite (*GNSS*) [Bidikar, Rao, Ganesh and Kumar 2014, Bock, Dach, Yoon and Montenbruck 2009, Lewandowski and Thomas 1991]. Ambas opciones son caras en términos de consumo de energía y precio. Otro de los beneficios del sistema de colaboración propuesto es poder mantener el tiempo Real en vuelo sin el uso de las opciones mencionadas anteriormente.

En el desarrollo del satélite OPTOS, el desafío fue doble. Primero, había que diseñar un computador fiable basado en dispositivos sensibles a SEU. En segundo lugar, estos dispositivos debían de consumir pocos recursos de potencia. La solución resultante a dichos desafíos ha sido la reutilización de una parte de los recursos de cada *DOT*, para

realizar las tareas críticas del satélite de forma redundante por todas las *DOT*. Una de estas tareas críticas es el mantenimiento en tiempo real, que se ha diseñado de manera distribuida dentro de la red descrita anteriormente, evitando puntos únicos de falla gracias al endurecimiento colaborativo.

En esta solución, el tiempo real se mantiene en cada *DOT* despierta mediante un conjunto de contadores internos. Cada *DOT* comparte con el resto su tiempo real (Broadcast) y verifica si difiere de los demás. El tiempo real correcto se comprueba a través de un mecanismo de votación que se detalla a continuación. El tiempo real votado se llama Tiempo Real Global (*GRT*). Este GRT se configura cada segundo mediante un proceso denominado proceso de mantenimiento en tiempo real (*RTM*) a bordo, que se ha implementado localmente en cada *DOT* con una máquina de estados finitos en *VHDL*. Esta implementación de hardware se eligió antes de una software, para maximizar la robustez del sistema y minimizar el tamaño de las *DOT*. El proceso de RTM implementado en cada *DOT* de la red distribuida garantiza un mantenimiento seguro del tiempo real a bordo, requerido por el satélite para operar.

El proceso de RTM involucra a todas las *DOT* despiertas que mantienen su propio tiempo real (*ORT*). Estas *DOT* pueden transmitir dos tipos de mensajes. Primero, su ORT se transmite a los elementos de la red. En segundo lugar, un mensaje de votación se transmite por aquellas *DOT* cuando su ORT no es igual a los tiempos reales recibidos de los otras *DOT*.

Hay un proceso continuo y periódico, en cada *DOT*, por el cual se verifica su propio ORT con el GRT. El GRT es el resultado de una ejecución exitosa de un ciclo de proceso de RTM. Al final de cada ciclo, cada *DOT* tomará una de estas tres acciones: si el ORT es diferente del GRT, el ORT se corrige; si la ORT es igual a la GRT, el ORT se mantiene; si OT aún no tenía una ORT (porque se acaba de despertar), su ORT se establece con el GRT.

El proceso de RTM se basa en las siguientes reglas:

- Cada vez que una DTO se despierta, no tiene ORT. Por lo tanto, debe esperar una ejecución exitosa de un proceso de RTM para adquirir su ORT.
- El tiempo real debe ser enviado periódicamente por toda *DOT* que posea su ORT.
- Cada vez que una *DOT* transmita de forma satisfactoria su ORT, toda *DOT* con ORT propio comparará ambos valores. Aquellas *DOT* que reciban un ORT diferente al suyo, enviarán inmediatamente un mensaje de votación con su ORT.
- Cada *DOT* que recibe un mensaje de votación debe transmitir un mensaje de votación con su propia ORT, independientemente de si es el mismo o diferente al recibido.
- El proceso finaliza cuando todas las *DOT* han enviado sus mensajes de votación, o cuando transcurren 16ms.
- Una vez terminado el proceso de votación, toda *DOT* (incluyendo aquellas sin ORT) hallará cuál es el tiempo real. Cada *DOT* con un tiempo diferente al de la mayoría, pierde su ORT y establece su ORT para que sea igual al GRT.

El diagrama de estados de la Fig. 35, muestra el proceso de mantenimiento de tiempo real implementado en todas las *DOT*.

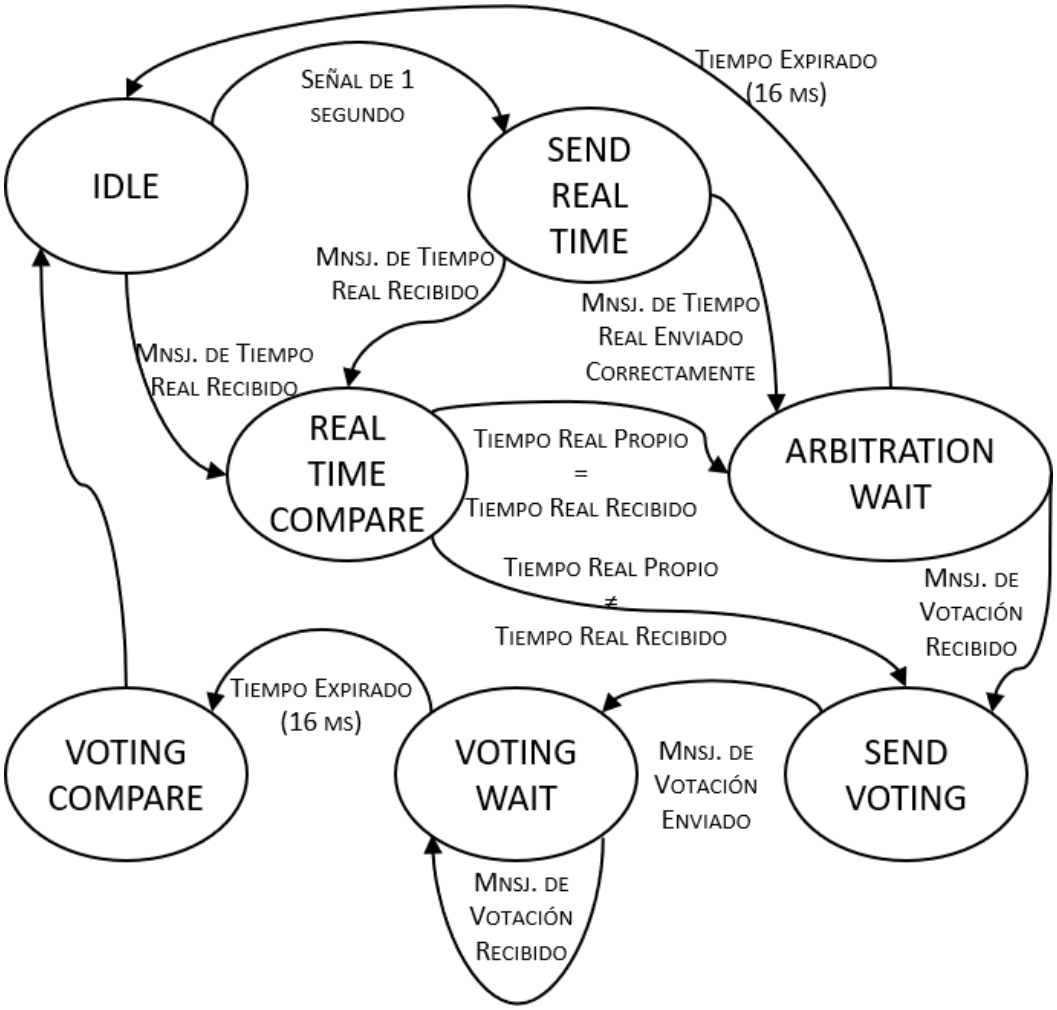


Fig. 35. Diagrama de estados del sistema del proceso de mantenimiento de tiempo real.

El tiempo real que se mantiene dentro de cada *DOT*, tiene una precisión de milisegundos.

La periodicidad del mensaje en tiempo real se ha establecido en 1000 ms. Por lo tanto, cada vez que el tiempo real mantenido por cada *DOT* llegue al siguiente segundo, esta intentará enviar un mensaje CAN hacia el bus CAN. Dos posibilidades surgen en este punto:

- Si hay una *DOT* cuyo oscilador sea más rápido, esta llega antes al siguiente segundo y, por lo tanto, comienza a transmitir el mensaje un poco antes que sus competidoras.

- Todos los *DOT* comienzan a enviar el mensaje en tiempo real al mismo tiempo, sin embargo, solo aquella con el ID de CAN de mayor prioridad gana el arbitraje del bus.

Pase lo que pase, las otras *DOT* que reciben el mensaje en tiempo real de la *DOT* ganadora, compararán su ORT con el recibido. Si no hay diferencia, cada *DOT* esperará 16 ms a recibir un mensaje de votación de cualquier otra *DOT*. Una vez que el temporizador caduque sin más mensajes, este ciclo de proceso de RTM habrá terminado. Sin embargo, si alguna de las otras *DOT* difiere de ORT de la que transmitió el primer mensaje, esta enviará un mensaje de votación con su ORT. Luego, todas las demás *DOT*, incluida la que envió el primer mensaje, deben enviar un mensaje de votación. Nuevamente, cuando el temporizador de 16 ms caduca, el proceso se resuelve y cualquier *DOT* con un ORT diferente de la mayoría (GRT), lo descarta y establece su ORT con el GRT.

En la Fig. 36, el diagrama de tiempos ilustra un ciclo típico de proceso RTM, en el que un DT pierde su ORT debido a un error inducido por SEU.

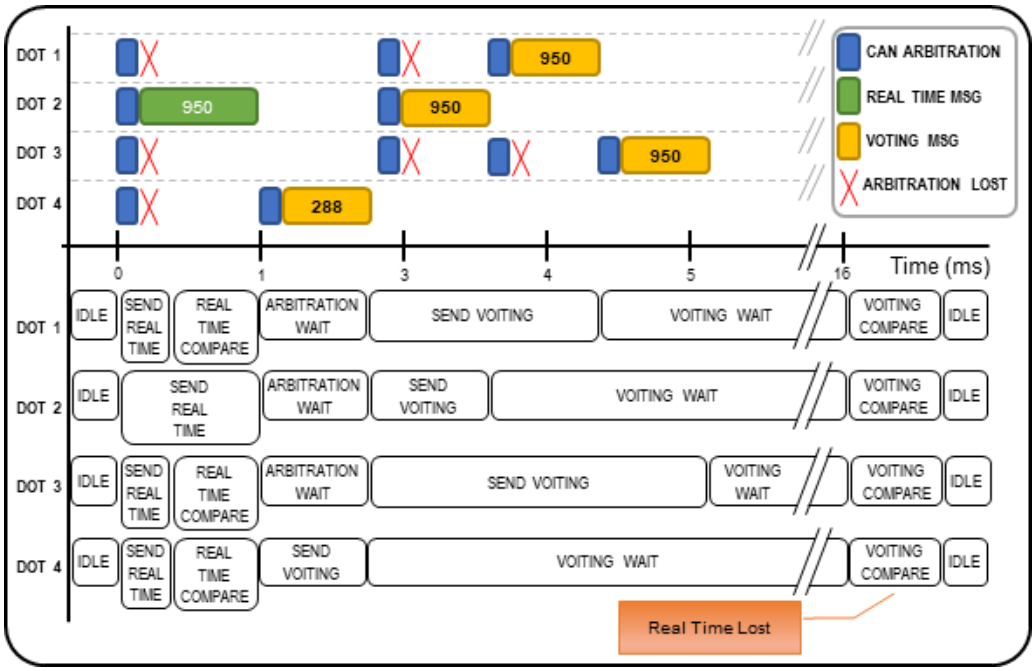


Fig. 36. Ejemplo de proceso de mantenimiento de tiempo real con votación.

Como muestra el ejemplo, todas las *DOT* pretenden enviar un mensaje con su tiempo real al mismo tiempo. Solo el que tiene la prioridad más alta (ID de CAN), que en este caso es la DOT2, logra enviar el mensaje. Luego, todos las demás *DOT* comparan el tiempo real recibido con su ORT. En el ejemplo, la DOT4 tiene un ORT diferente al ORT de la DOT2, por lo tanto, comienza un ciclo de votación enviando un mensaje de votación. En ese momento, todos las demás *DOT* envían un mensaje de votación al bus (incluso DOT2). Nuevamente la DOT2 logra en primer lugar enviar el mensaje de votación. Después la DOT1 y finalmente la DOT3. En este punto, todas las *DOT* están esperando a que llegue un nuevo mensaje de votación de una *DOT* diferente o que expire el temporizador de 16 ms. Cuando finalmente expira el contador, cada *DOT* evalúa internamente si su ORT está en el grupo mayoritario. Si es así, mantiene el ORT; si no, descarta su ORT y establece su ORT con en el GRT calculado.

El éxito del proceso de RTM requiere que cada *DOT* compare su propio tiempo real al mismo tiempo. Para hacerlo, cada *DOT* aprovecha la estructura de mensajes CAN y congela su ORT tan pronto como se recibe el campo de arbitraje de CAN. En ese momento, cada *DOT* sabe si el remitente está transmitiendo un mensaje de tiempo real. Por lo tanto, cada comparación entre tiempos recibidos y propios se realiza con valores congelados almacenados en un registro dedicado dentro de cada *DOT*. El formato del mensaje CAN enviado para el mensaje en tiempo real se muestra en la Fig. 37.

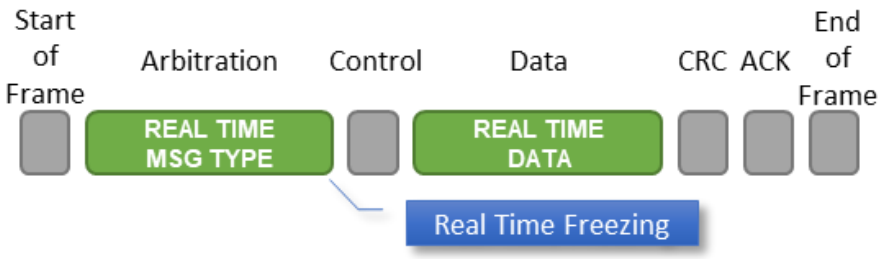


Fig. 37. Punto de congelación del ORT para inter-comparación durante el proceso de RTM.

La congelación del ORT no la hacen solamente las *DOT* recibiendo el mensaje, sino que también lo hace la *DOT* enviando el mensaje. Utiliza los 7 bits restantes (1 del *RTR*, y 6 del campo de control) hasta tener que empezar a enviar los tres bytes del tiempo.

4.5 Análisis de Radiación del Computador

Como parte de las actividades de ingeniería de radiación de la misión OPTOS, se llevó a cabo un análisis completo del entorno de radiación de la misión. Este análisis incluyó la caracterización de los flujos de partículas de alta energía esperados durante la misión. Además, se ha incluido un estudio de simulación detallado sobre la propagación de dichos flujos a través de un modelo 3D completo de la plataforma (con la herramienta FASTRAD [Pourrouquet, Thomas, Peyrard, Ecoffet and Rolland 2010]), donde se introdujeron las estructuras principales del satélite así como los diferentes subsistemas. El objetivo final era proporcionar una estimación realista de los niveles de radiación dentro del satélite, en forma de flujos espectrales de partículas, dosis acumuladas ionizantes y no ionizantes. Este estudio permite predecir los niveles y flujos de radiación en ubicaciones y componentes concretos [Martín-Ortega et al. 2019].

Estos niveles de radiación estimados se han utilizado para la selección de tecnologías de acuerdo con sus tolerancias de radiación y proporcionaron el entorno de entrada para la evaluación de la degradación de elementos sensibles a la radiación al final de la vida útil (*EOL*). Además, los resultados del estudio se han utilizado para la estimación de las tasas promedio de efectos de evento único (*SEE*) para componentes específicos. En particular, se han estimado las tasas de *SEU* en los diferentes dispositivos lógicos programables (*PLD*) de Xilinx, Virtex-II® y CoolRunner-II®, utilizados en el computador. En el caso de la *FPGA* del *EPH*, este dispositivo consta de una memoria de configuración basada en tecnología *SRAM*, además de una memoria Block RAM (también basada en SRAM). Las *CPLD* utilizadas en las *DOT*, constan de una memoria de configuración no-volátil basada en tecnología FLASH, y otra memoria de configuración volátil basada en SRAM. Además, gracias a los estudios previos realizados sobre ambos dispositivos [García et al.

2008, Swift 2004] se ha podido hacer un estudio específico de las tasas de fallo esperadas para cada elemento reprogramable del computador.

En los siguientes párrafos se detallan las técnicas de cálculo y los modelos utilizados para la propagación del entorno de radiación, así como los datos de prueba de irradiación experimental utilizados para la predicción de las tasas de *SEE*.

4.5.1 ENTORNO DE RADIACIÓN DE OPTOS

OPTOS se lanzó el 23 de noviembre de 2013 a una órbita helio-sincrónica con una altitud promedio de 670 km. La misión OPTOS estuvo expuesta a un entorno de radiación severa, principalmente dominada por el flujo de protones y electrones de alta energía, provenientes del anillo interno de Van Allen. Además, existieron perturbaciones ocasionales por eventos de partículas solares, y un flujo constante, aunque menos importante, de partículas muy energéticas por la contribución de Rayos Cósmicos Galácticos (*GCR*).

Para la estimación de los flujos de partículas del cinturón de Van Allen, el modelo ambiental utilizado fue el AP8/AE8 [I. Vette 1991a, 1991b, M. Sawyer and I. Vette 1977], recomendado por la Agencia Espacial Europea (ESA) en sus directrices ECSS [ECSS 2008b, 2008a] como modelo estándar para misiones de orbita cercana. En el caso de eventos solares y GCR, siguiendo las recomendaciones de la ESA, los modelos ambientales aplicados fueron el ESP / PHYSCIC [Xapsos, Stauffer, Jordan, Barth and Mewaldt 2007, Xapsos, Summers, Barth, Stassinopoulos and Burke 1999, 2000], con un nivel de confianza del 95%, y el ISO-15390 ['ISO 15390' 2004].

4.5.2 MODELO 3D DE RADIACIÓN DE OPTOS

Para propagar el entorno de la misión de la sección anterior a ubicaciones específicas dentro de la plataforma OPTOS, se construyó un modelo en 3D de los elementos estructurales del satélite mediante la herramienta FASTRAD y los diseños CAD mecánicos reales de OPTOS. Además, se implementaron modelos simplificados de los circuitos integrados CoolRunner-II® y Virtex-II® del computador. Para ello, se han simulado ambos dispositivos como componentes con encapsulados plásticos, con las dimensiones reales del silicio, y encapsulándolos dentro de una capa de silicona delgada (100 µm de espesor). La Fig. 38 muestra el modelo 3D de OPTOS, tal y como se construyó con la herramienta FASTRAD.

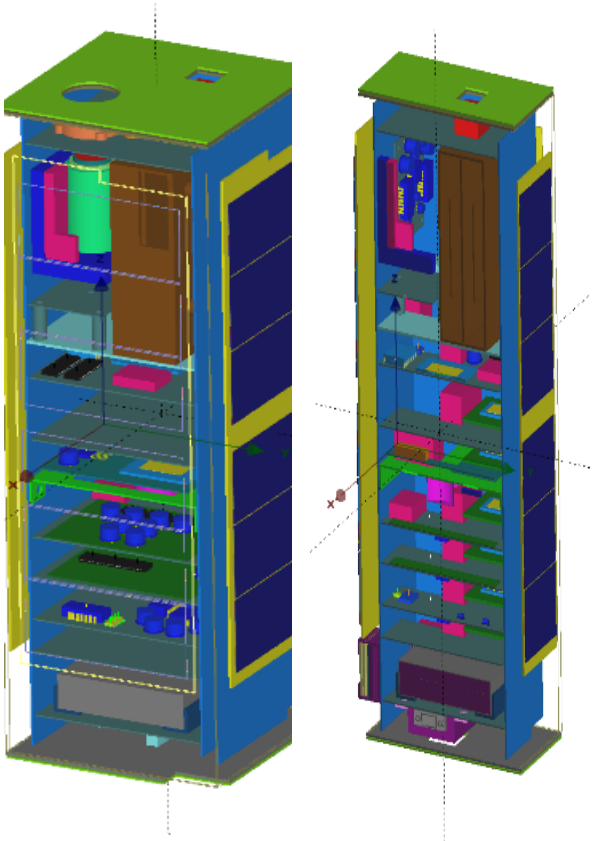


Fig. 38. Modelo 3D de OPTOS realizado con FASTRAD.

El modelo final de OPTOS 3D obtuvo una masa simulada total de 2,5 kg. Una vez que el modelo 3D se verificó geométricamente, se exportó como un archivo en formato GDML [Chytracek, Mccormick, Pokorski and Santin 2006]. Este modelo se introdujo en el software de propagación de partículas, GEANT4 [Agostinelli 2003, Allison et al. 2006]. Por último, se utilizó el módulo Monte-Carlo desarrollado por el *CERN*, capaz de simular el paso y la interacción de la radiación de alta energía en la materia utilizando modelos electromagnéticos y hadrónicos realistas [GEANT4-Collaboration 2010]. En la Fig. 39 se muestra el promedio de flujo espectral anual de protones obtenido para las ubicaciones dentro de la plataforma OPTOS.

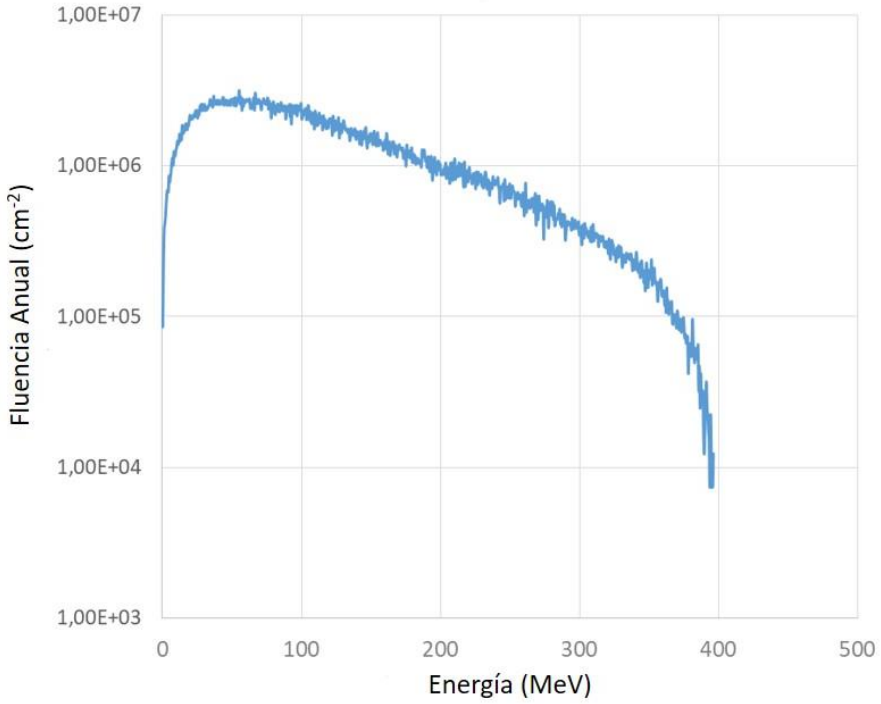


Fig. 39. Espectro promedio de protones por año.

4.5.3 DATOS EXPERIMENTALES DE IRRADIACIONES

En 2007, durante las fases de viabilidad del programa OPTOS, se llevaron a cabo una serie de campañas de irradiación con protones de alta energía [García et al. 2008], con el objetivo de caracterizar la sensibilidad a *SEE* de los dispositivos CoolRunner-II®. Se probaron seis CPLD (*DUT* 1 a 6). Cuatro de los dispositivos se utilizaron para hacer pruebas en modo estático (*DUT* 1 a 3 y 7), con las que se buscaba hallar la sensibilidad de las memorias de configuración de las CPLD (memorias Flash y RAM). Las pruebas realizadas para entender la sensibilidad de la memoria RAM, fueron en una situación de encendido. Por otro lado, las pruebas realizadas para entender la sensibilidad de la FLASH, fueron tanto con los dispositivos encendidos como apagados. Todos los dispositivos se probaron bajo una energía de protones individuales monoenergéticos de 10,75 a 62,91 MeV, y hasta una fluencia máxima de 10^{10} p+/cm^2 o bien 100 SEU, cualquiera que fuese la condición que se obtuviera primero.

Las pruebas estáticas consistían en una aplicación que se conectaba con la unidad en pruebas mientras esta estaba siendo bombardeada con protones. De forma continua, el ordenador situado en una habitación adyacente[8], leía la memoria de configuración del dispositivo. Luego comparaba el resultado con un fichero maestro. Cada bit diferente entre el fichero leído y el maestro, era contabilizado como un *SEU*. Entonces se reiniciaba la memoria de configuración con los valores correctos y se continuaba con las pruebas. Estas acababan cuando la fluencia de protones llegaba a 10^{10} p+/cm^2 o el número de SEU a cien[9].

Durante las pruebas con la memoria FLASH, no se obtuvo ni un solo SEE. Este hecho era determinante, puesto que, si la memoria FLASH hubiera sido sensible a SEE, podría corromperse durante la misión y las CPLD (y por lo tanto las *DOT*) quedaría inservibles. Por otro lado, los resultados obtenidos en las *DUT* 1 y 7 durante las pruebas estáticas de memoria SRAM de configuración se muestran en la Tabla 4. Los resultados de sección eficaz se ajustaron a una función de distribución de Weibull. En la Tabla 4 se muestra los parámetros de ajuste de Weibull para los diferentes *DUT* y SEE:

Tabla 4. Sección eficaz de la memoria de configuración SRAM durante las pruebas estáticas con protones.

DUT	*W*	*S*	E_{Th} *(MeV)*	*Sección Eficaz (cm-2 device^{-1})*
1	*5.4*	*0.58*	*10.8*	$2.13 \cdot 10^{-8}$
7	*5.6*	*0.65*	*10.8*	$2.35 \cdot 10^{-8}$

Las pruebas dinámicas se hicieron sobre las *DUT* 4 y 5. Estas pruebas pretendían hallar la probabilidad de fallo, teniendo no sólo en cuenta los SEU en la memoria de configuración, sino también en los propios registros y recursos lógicos de la CPLD.

[8] Durante las pruebas de radiación con un ciclotrón, inyectando partículas altamente energéticas, todos los equipos de medida han de estar o en otras habitaciones adyacentes o muy protegidos con ladrillos de parafina y plomo.

[9] Este número de SEU es necesario para tener una confianza estadística del 90%.

Durante las pruebas, la CPLD ejecutaba una rutina basada en la utilización de una serie de entradas transformándolas en una salida, que la aplicación en el PC comprobaba su resultado. Cuando el resultado era distinto o había algún problema en las comunicaciones, la aplicación de PC contabilizaba un nuevo SEFI y ejecutaba un ciclo automático de reinicio (limpiando así cualquier rastro de SEU en la CPLD) . Los resultados de las pruebas, en forma de SEE de sección eficaz vs energía incidente de protones, se obtuvieron para cada *DUT* y se ajustaron igualmente a una función de distribución de Weibull. La Tabla 5 muestra los resultados de estas pruebas.

Tabla 5. Sección eficaz de los dispositivos durante las pruebas dinámicas con protones.

DUT	***W***	***S***	***E_{Th} (MeV)***	***Sección Eficaz (cm-2 device^{-1})***
4	4.0	0.46	10.8	$5.70 \cdot 10^{-9}$
5	4.1	0.53	10.8	$6.61 \cdot 10^{-9}$

Para los dispositivos Virtex-II® [Swift 2004], las Tabla 6 y Tabla 7 muestran los resultados de las pruebas de irradiación para SEU y SEFI realizadas por el Xilinx Radiation Test Consortium (*XRTC*) para iones pesados y protones:

Tabla 6. Resultados de SEU de las pruebas de radiación sobre Virtex-II.

Cell	***W***	***S***	***E_{Th} (MeV)***	***Saturation Cross Section (cm-2 bit^{-1})***
Protons				
Config-Bits	*12*	*0.5*	*3.0*	$3.8 \cdot 10^{-14}$
BRAM	*12*	*0.6*	*3.0*	$4.1 \cdot 10^{-14}$
Heavy Ions				
Config-Bits	*33*	*0.8*	*1.0*	$4.37 \cdot 10^{-8}$
BRAM	*17*	*0.9*	*1.0*	$4.19 \cdot 10^{-8}$

Tabla 7. Resultados de SEFI de las pruebas de radiación sobre Virtex-II.

Cell	*W*	*S*	E_{Th} *(MeV)*	*Saturation Cross Section (cm-2 device^{-1})*
Protons				
POR[10]	*12*	*1.0*	*7.0*	$3.74 \cdot 10^{-13}$
SMAP[11]	*12*	*0.5*	*6.5*	$5.72 \cdot 10^{-13}$
JCFG[12]	*12*	*0.5*	*6.0*	$2.86 \cdot 10^{-13}$
Heavy ions				
POR[1]	*22*	*1.2*	*1.5*	$2.50 \cdot 10^{-6}$
SMAP[2]	*17*	*1.0*	*1.5*	$1.72 \cdot 10^{-6}$
JCFG[3]	*17*	*1.0*	*1.5*	$2.51 \cdot 10^{-7}$

4.5.4 ESTIMACIÓN DE TASA DE FALLOS POR RADIACIÓN DEL COMPUTADOR

Finalmente, combinando los modelos de partículas, el espectro de protones atenuado calculado para la estructura del satélite y los datos experimentales de SEE, se ha realizado una estimación de las tasas de ocurrencia de SEE para la misión OPTOS. La Tabla 8 muestra las tasas pronosticadas de SEU (*DUT*1 y 7) y SEFI (*DUT*4 y 5) para el CPLD CoolRunner-II® utilizada en las *DOT* del computador.

Tabla 8. Relación de SEU y SEFI para las CoolRunner-II por año.

Device/Cell	*Proton*	*Ions*	*Total*

[10] Power-On Reset

[11] Select MAP Configuration Port

[12] JTAG Configuration Port

	(device^{-1} year^{-1})		
SEU-SRAM (DUT1)	20 ± 4	---	20 ± 4
SEU-SRAM (DUT7)	21 ± 4	---	21 ± 4
SEFI (DUT4)	5 ± 1	---	5 ± 1
SEFI (DUT5)	6 ± 1	---	6 ± 1

Para las *DUT* 1 y 7, se muestran el número de SEU por año, es decir, el número total de bits que cambiarán de 0 a 1 o de 1 a 0, dentro de la memoria de configuración de las CPLD. El número de SEFI, como se puede observar, es menor que el de SEU. Esto se debe principalmente, a que un SEU en la memoria de configuración, no ha necesariamente de provocar un error en la ejecución. Se calcula que del orden del 90% de los bits de configuración, no afectan negativamente al funcionamiento de la CPLD. Este dato varía dependiendo de la aplicación que se ejecuta. A este dato, hay que añadir los SEFI provocados por impactos directos en los registros y la lógica del dispositivo, que sí están directamente relacionados con la aplicación. Los resultados del análisis indican que si las *DUT* funcionaran durante el 100% del tiempo, se esperarían del orden de un fallo funcional cada 2 meses. Es por esto, que se han implementado las técnicas de endurecimiento explicadas en este capítulo.

El mismo análisis se realizó para los dispositivos Virtex-II (ver Tabla 9). Como se hace notar en la tabla de las CoolRunner-II, no fue posible realizar las pruebas de radiaciones con iones pesados (HI-). Sin embargo, como se puede observar en el caso de la Virtex-II, la contribución de HI-, debido principalmente a la protección de los cinturones de Van Allen, es más de un orden de magnitud menor que la de P+, para la órbita de OPTOS.

Tabla 9. . Relación de SEU para la Virtex-II por año.

Device/Cell	***Proton***	***Ions***	***Total***

		(bit^{-1} day^{-1})	
Config-Bits	$1.4 \cdot 10^{-7}$	$7.6 \cdot 10^{-8}$	$2.2 \cdot 10^{-7}$
BRAM	$1.6 \cdot 10^{-7}$	$9.3 \cdot 10^{-8}$	$2.5 \cdot 10^{-7}$

5 *Validación en Tierra*

En el sector aeroespacial, el desarrollo de sistemas y cargas útiles para uso espacial requiere unos procesos de desarrollo basados en estándares de calidad. Estos procesos ayudan a garantizar la fiabilidad del producto final, maximizando las posibilidades de éxito de las misiones. En Europa, la Cooperación Europea para la Normalización del Espacio (*ECSS*) es una iniciativa establecida desde 1993 para desarrollar un conjunto único y coherente de estándares fáciles de usar, para su uso en todas las actividades espaciales europeas. Estos estándares definen entre otras cosas una serie de posibles filosofías de modelos, donde el producto final o FM, va precedido de otros modelos previos con el fin de ir reduciendo los riesgos en las primeras fases del proyecto, y terminar con un modelo probado, pero no estresado. La Fig. 40 muestra la filosofía más común de desarrollo de modelos en misiones espaciales.

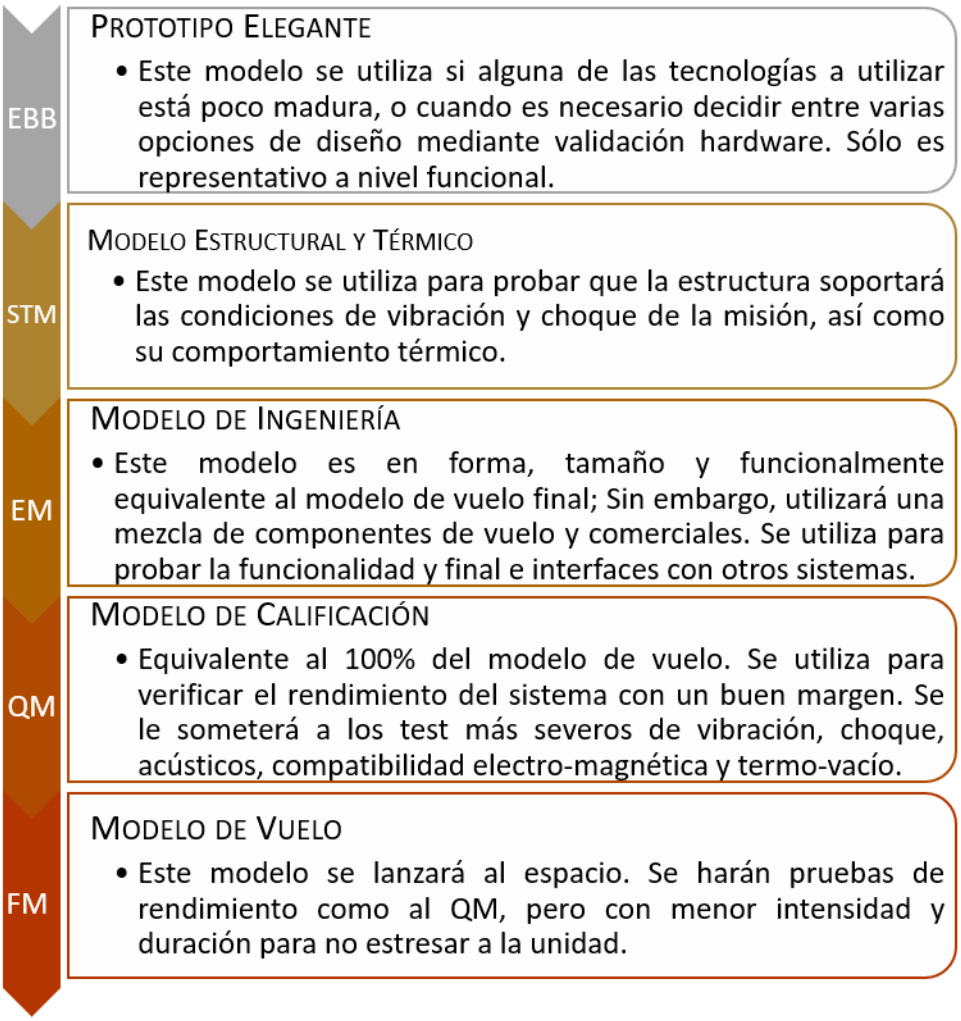


Fig. 40. Modelos de **SS**/**PL** en proyectos espaciales.

Como se puede observar, no existe ninguna prueba ni modelo pensado para ser probado frente a radiación. Normalmente, no hay un proceso de calificación para la sensibilidad a la radiación ionizante del sistema, considerando que, si cada dispositivo es robusto, el sistema compuesto por ellos también debería ser robusto. Solo cuando la tecnología de un dispositivo no es tolerante a *SEU*, se realizan pruebas de calificación previas, con campañas muy costosas, tanto en tiempo como en dinero. Sin embargo, pocos o ninguno de los análisis se realizan en la sensibilidad de la arquitectura global como un sistema completo. Como se ha descrito en el capítulo 2 Entorno Espacial, hay diferentes efectos de la radiación sobre los dispositivos electrónicos. Por un lado, tanto el efecto debido a la acumulación de dosis total (*TID*), como los eventos singulares destructivos (*SEL*), no

pueden ser mitigados por técnicas software. Estos efectos tan sólo pueden ser protegidos bien con la utilización de materiales pantalla (capaces de bloquear o disminuir la energía de las partículas); o bien con la utilización de electrónica capaz de apagar rápidamente los dispositivos cuando se detecta un sobreconsumo; o bien una combinación de ambas. Sin embargo, el efecto de los eventos singulares no destructivos conocido como *SEU*, sí es mitigable con diferentes técnicas de endurecimiento. Cuando se habla de mitigar los efectos de los SEU, lo que se quiere expresar es que el evento de cambio de bit seguirá ocurriendo inevitablemente, pero el efecto que esto tiene en el funcionamiento del circuito será corregido o evitado. Estas técnicas, independientemente de estar orientadas al hardware o al software, se basan en detectar y corregir los efectos de la radiación a nivel dispositivo. El endurecimiento colaborativo expuesto en el capítulo anterior se basa en proteger al sistema como conjunto, permitiendo el desarrollo de tareas críticas de forma fiable.

En este capítulo se presenta el ecosistema desarrollado para poder evaluar la fiabilidad del computador distribuido en su conjunto. Esta herramienta se ha evolucionado sobre el Sistema Autónomo de Emulación [López-Ongil, García-Valderas, Portela-García and Entrena 2007] creado por la Universidad Carlos-III de Madrid. Una breve introducción sobre este sistema se describe en el apartado de Trabajo Previo. La descripción del ecosistema nombrado se presenta en el apartado Ecosistema de Evaluación de Arquitecturas Distribuidas. Por último, los resultados de la validación son presentados en el apartado Resultados Experimentales de la Validación.

5.1 Trabajo Previo

El sistema de emulación autónoma es un sistema creado para simular los efectos de la radiación en aplicaciones concretas implementadas en un *PLD*. El efecto a simular es concretamente el cambio de valor de un registro determinado dentro del circuito. Lo que

comúnmente se llama, un *bit-flip*[13]. Este cambio de valor inesperado puede crear potencialmente un error de funcionamiento en el circuito. Si esto es así, el error puede ser permanente o pasajero. Los principales problemas con los que se encuentran los sistemas de inyección de fallos en PLD son dos: por un lado, el *bit-flip* puede ocurrir en cualquier parte del circuito implementado, pudiendo comprender así una gran cantidad de celdas de memoria (cientos de miles o millones); por otro lado, este hecho puede ocurrir en cualquier momento de la ejecución (cientos de miles o millones de ciclos de reloj para una ejecución concreta). El número de fallos totales es el número de celdas de memoria multiplicado por el número de ciclos de la operación. Para poder dirimir una estadística fiable de la robustez del sistema a emular, hay que probar un número significativo del total. Por ello, el número de inyección de fallos a emular suele ser de cientos o miles de millones.

Para poder hacer una emulación del circuito en un tiempo razonable, el sistema de emulación autónoma implementa toda lo lógica de control dentro la FPGA de pruebas, evitando cualquier tipo de comunicación con el PC que hace de interfaz, hasta haber terminado la emulación. Este sistema, emula el circuito a probar (*CUT*), superponiendo una capa de inyección de fallos y control de resultados. Para ello, el emulador incluye un módulo de aplicación o banco de pruebas, un módulo de inyección de fallos, un módulo de clasificación de errores y un controlador de emulación. El controlador de emulación maneja todo el proceso, inicializando el circuito, aplicando vectores de prueba de entrada, inyectando fallos y permitiendo la comparación entre salidas esperadas y salidas obtenidas.

Con el sistema de emulación autónoma, una campaña de inyección de fallos típica se ejecuta de la siguiente manera. Primero se configura la FPGA. Este paso define el circuito a probar, el banco de pruebas y la lista de fallos. Después el controlador de emulación FPGA repite para cada fallo los siguientes pasos:

[13] Este término se utiliza para describir el cambio de valor 0 a 1 o de 1 a 0, dentro una celda de memoria o registro, provocado por el paso de una partícula altamente energética a través de esta.

1. Establece el circuito en el estado anterior a la inyección del fallo.

2. Inyección de fallo.

3. Emulación del circuito hasta que se clasifique el error o hasta que finalice el banco de pruebas.

Cuando finaliza la campaña de emulación de fallos, el host puede descargar la clasificación de errores obtenidos con la información precisa de qué celda de memoria y en qué momento se produjo el error concreto. Los errores se dividen en tres grupos:

- Latentes: cuando un SEU provoca que un *FF* o más cambien del valor nominal, pero no provocan un error funcional en el dispositivo.

- Silenciosos: cuando un SEU provoca que un *FF* o más cambien del valor nominal pero, antes de acabar la ejecución de ese ciclo de emulación, el error desaparece sin haber provocado ningún error funcional en el dispositivo.

- Funcional: cuando un SEU provoca un error funcional en el dispositivo.

.

5.2 Ecosistema de Evaluación de Arquitecturas Distribuidas

En el apartado anterior se ha descrito el sistema de emulación autónomo, el cual se utiliza como base para crear un ecosistema de evaluación de arquitecturas distribuidas. En este apartado se propone un método original de evaluación para probar la sensibilidad de SEU de los sistemas de hardware distribuidos en las primeras etapas de diseño y al nivel del sistema. Esta solución consiste en crear un ecosistema compuesto por los siguientes módulos interconectados entre sí.

❖ Una placa de desarrollo con:

- Un interfaz UART de comunicaciones con el PC, para iniciar los ciclos de emulación y descargar los resultados.
- Un interfaz CAN para permitir a una aplicación ejecutandose en el PC validar el prototipado del sistema colaborativo.
- Un interfaz JTAG para configurar y programar la FPGA.
- Una FPGA con los siguientes componentes:
    - El **sistema de emulación autónoma** se utiliza para realizar todas las tareas de inyección de fallos, controlar la campaña de inyección y almacenar los errores encontrados. Este módulo está implementado dentro la FPGA de emulación.
    - Un bloque específico (**Módulo-X**) que aumenta la capacidad de observación y control de cada nodo en la red distribuida. El Módulo-X está conectado a la red como un nodo especial para realizar la vigilancia del efecto de fallo, validando las tareas globales, la ejecución correcta y los mensajes a través del enlace de la red. Además, puede utilizarse para enviar mensajes erróneos a través del enlace de comunicación y forzar errores funcionales. Este módulo también está implementado dentro de la FPGA.
    - El **prototipado del computador colaborativo**.

❖ Un ordenador personal con:

- Un interfaz UART de comunicaciones con la placa de prototipado, para comunicación con el sistema autónomo de emulación.
- Un dispositivo USB a CAN para observar las comunicaciones internas del prototipado del sistema colaborativo.
- Un dispositivo USB a JTAG para configurar y programar la FPGA.
- Una aplicación en PC para controlar el inicio y la finalización de la emulación autónoma.

- Una aplicación en PC para validar el funcionamiento del prototipo del computador colaborativo.

Esta solución de evaluación permite la emulación de hardware del computador distribuido, posibilitando la validación de la funcionalidad en Tierra (sin inyección de fallos). El sistema también permite la aplicación de campañas de inyección de fallos, lo que ayuda al diseñador/a a localizar bloques y tareas sensibles, así como a validar la efectividad de los métodos de endurecimiento aplicados.

Esta herramienta mejorada de emulación añade una capa lógica de validación, confiriendo al sistema autónomo la capacidad de evaluar los errores no sólo a nivel circuito, sino también a nivel sistema distribuido.

5.2.1 MÓDULO-X. ANALIZADOR DE ARQUITECTURAS DISTRIBUIDAS

El Módulo-X confiere al ecosistema de emulación de fallos la capacidad de interpretar el funcionamiento del sistema distribuido como tal, y no como un único circuito a probar. Esta nueva capa permite evaluar tanto los fallos unitarios de cada nodo, como los fallos a nivel sistema, permitiendo así realizar campañas de inyección de fallos a arquitecturas distribuidas.

El Módulo-X es responsable de validar cada mensaje de red y señalar cualquier error hacia el sistema de control de emulación. Como módulo no intrusivo, el Módulo-X evalúa la información relacionada con las tareas críticas enviadas por las *DOT* y verifica su corrección contra sus propios valores esperados.

Se le ha llamado Tiempo Real Dorado[14] al tiempo real no corrompido mantenido por Módulo-X, el cual no se ve afectado ni por el sistema de inyección de fallos ni por el proceso de *RTM*. Por lo tanto, el Tiempo Real Dorado es el tiempo de referencia para el sistema. No ha de confundirse el Tiempo Real Dorado con el Tiempo Real Global (*GRT*, explicado en el capítulo 4 Computador basado en Endurecimiento Colaborativo), que

[14] Del término en inglés *Golden*, que se refiere al parámetro de referencia, el inherentemente correcto.

es el resultado de una ejecución exitosa de un proceso de mantenimiento en tiempo real entre *DOT*.

El Módulo-X implementa la funcionalidad descrita a continuación:

- **Comandado**: el módulo tiene la capacidad de configurar el tiempo real como si se hubiera recibido un telecomando de Tierra. Por lo tanto, en cuanto empieza la ejecución de la emulación, este módulo inyecta en el CAN BUS el tiempo real. Una vez que X-Module configura el tiempo real, cada *DOT* activa lo adquirirá automáticamente y comenzará a participar en el proceso de RTM.

- **Vigilancia 1 (sf):** la tarea más importante es identificar cualquier error relacionado con la integridad del tiempo real mantenido por los terminales distribuidos mediante técnicas colaborativas. Para cumplir con esta tarea, el Módulo-X actúa como un supervisor que escucha durante todo el proceso de RTM, y una vez que el proceso se ha resuelto, compara el *GRT* con el Tiempo Real Dorado. Si no hay coincidencia entre ellos, un **error del sistema** (**sf**) se eleva hacia el controlador del emulador.

- **Vigilancia 2 (drtf):** otra medida de la robustez del sistema es identificar cuántos *FF* dentro de cada *DOT* pueden causar un error funcional. Para medir estos valores, el Módulo-X actúa nuevamente como un supervisor que escucha durante el proceso de RTM. Si un *DOT* envía un mensaje en tiempo real que difiere en más de ± 1 ms del Tiempo Real Dorado, se genera un **error en tiempo real de *DOT* (drtf)** hacia el controlador del emulador.

- **Vigilancia 3 (dff):** como es el caso de fallo anterior, un *DOT* puede no tener problemas con su *ORT* y, sin embargo, tener problemas para realizar otras tareas. El Módulo-X conoce cada mensaje CAN que cada *DOT* debe enviar. Si un *DOT* envía un mensaje incorrecto, ya sea en el contenido o en el tiempo, un **error funcional de *DOT* (dff)** se eleva hacia el Controlador del emulador. El X-Module no sólo es capaz de identificar los mensajes erróneos, sino también de identificar la ausencia de mensajes correctos.

Todos estos errores se informan al controlador del emulador, que almacena esta información para cada *FF* y para cada SEU inyectado. Esto permite no sólo saber cómo de robusta es la arquitectura colaborativa, sino también identificar las debilidades en cada terminal distribuido. Si el análisis final así lo recomienda, se deberá endurecer las partes de los sistemas que muestran una mayor sensibilidad a la radiación ionizante. Se pueden usar muchas técnicas simples [Y. Li, Nelson and Wirthlin 2010, Pratt, Caffrey, Graham, Morgan and Wirthlin 2006] para endurecer partes específicas de un diseño digital. Estas técnicas selectivas ha mostrado mejores resultados en términos de endurecimiento del circuito, en comparación con la utilización de recursos con técnicas no selectivas [She and Samudrala 2009].

5.2.2 CASO PRÁCTICO

El computador distribuido de OPTOS ha sido seleccionado como caso de estudio. Este sistema prototipado es un subconjunto de 4 terminales distribuidos diferentes, más el Módulo-X. El conjunto total de *DOT* se ha minimizado buscando el peor escenario posible. Como describe el capítulo 4 Computador basado en Endurecimiento Colaborativo, el computador requiere tener al menos tres *DOT* diferentes con tiempo real, esto significa que cuanto menor sea el número total de *DOT* en el sistema, menor será el número de *DOT* que participe en el proceso de *RTM*. Debido a la reducción del número de unidades, cada una de ellas deberá mantenerse activa durante más tiempo, incrementando así la probabilidad de acumulación de *SEU* y por tanto, de provocar un fallo funcional en la misma.

En la Fig. 41, se muestra un diagrama de bloques del sistema de emulación construido para la campaña de inyección de fallos.

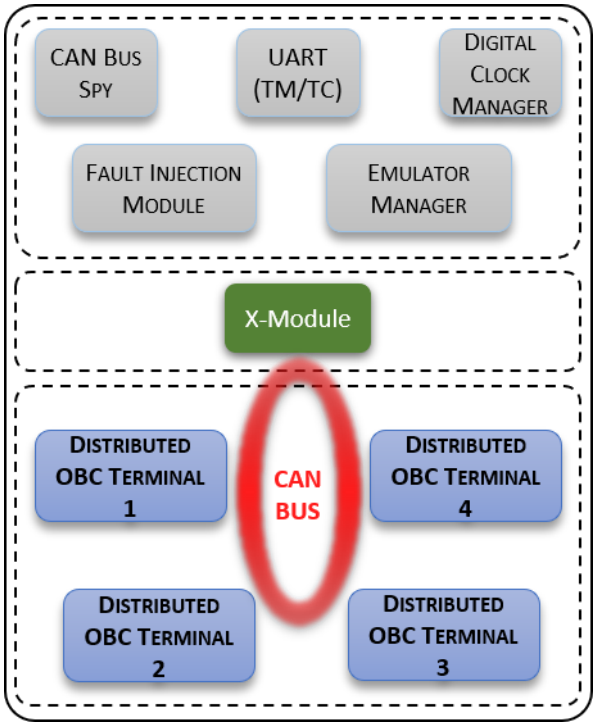


Fig. 41. Sistema de Emulación Autónoma para Computadores Distribuidos.

El ecosistema de emulación está, por lo tanto, compuesto de tres grupos de componentes. Primero, el sistema bajo prueba (*SUT*) que consta de cuatro terminales distribuidos de la arquitectura de procesamiento del satélite OPTOS. En segundo lugar, el **Módulo-X**, que implementa la vigilancia inteligente de errores funcionales junto con un sistema de generación de errores. Tercero, el **sistema de emulación autónoma**, cuyos componentes principales son el módulo de inyección de fallos y el administrador de emulación. Estos son los encargados de aplicar las configuraciones adecuadas en el *SUT*, así como de activar la inyección de fallos y evaluar los efectos de estos, clasificándolos en el diccionario de errores.

5.3 Resultados Experimentales de la Validación

En esta sección se describen las diferentes pruebas realizadas y los resultados obtenidos, durante la campaña de emulación de fallos. En primer lugar, se han validado tanto las capacidades del Módulo-X, como el buen funcionamiento del prototipo de la

arquitectura distribuida del computador de OPTOS. Esta validación se ha realizado mediante simulación. En segundo lugar, se han realizado extensas campañas de inyección de fallos para evaluar la sensibilidad a SEU del sistema distribuido, tanto a nivel unidad como a nivel del sistema.

Para desarrollar la emulación de hardware distribuida, se ha utilizado la placa de desarrollo XUPV5 del Programa Universitario de Xilinx [Xilinx n.d.]. La placa ha sido seleccionada debido al gran conocimiento de esta, por haberla utilizado en emulaciones previas, minimizando así los posibles errores cometidos durante la integración entre los diferentes componentes. La placa XUPV5 seleccionada cumple con casi todos los requisitos descritos en la sección Ecosistema de Evaluación de Arquitecturas Distribuidas, excepto el interfaz CAN. Para poder observar el bus CAN, se han extraído dos E/S dedicadas de la FPGA a través de un conector de propósito general. Se ha utilizado una *PCB* Piggy-Back específica (ver Fig. 42)con un módulo de controlador CAN para conectar el bus con un dispositivo comercial Peak-CAN USB [Systems n.d.].

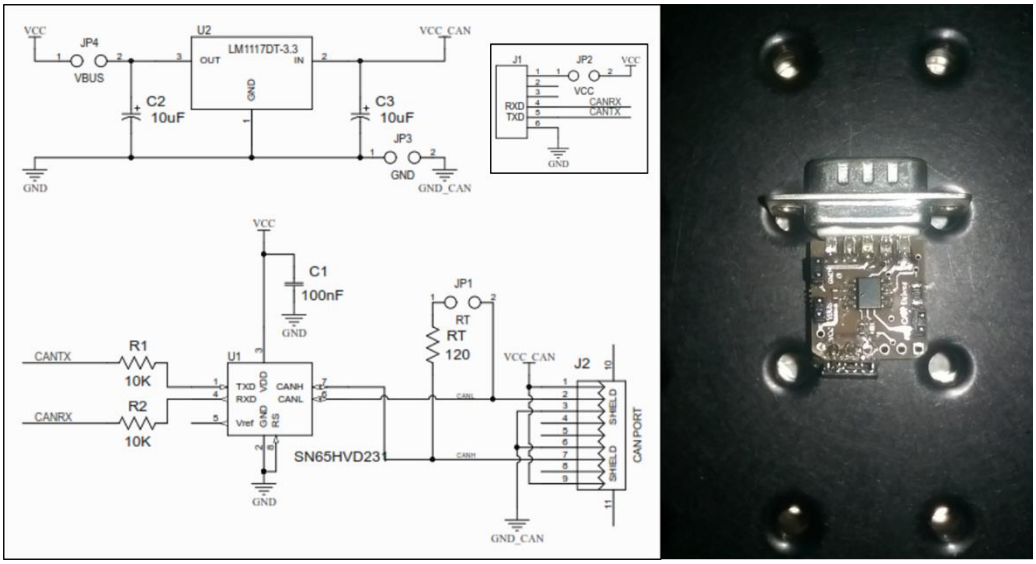


Fig. 42. Tarjeta conversión CMOS a CAN BUS.

5.3.1 VALIDACIÓN FUNCIONAL

La validación funcional del ecosistema completo se realizó mediante dos métodos diferentes. Primero, se realizó una simulación del Módulo-X junto al prototipo de la

arquitectura distribuida. Una vez se obtuvieron los resultados esperados, se procedió a la emulación del ecosistema completo.

5.3.1.1 Simulación

Previo a la emulación sobre el hardware, se simuló una descripción del sistema en un PC mediante la herramienta de simulación ModelSim©. En la simulación, el banco de pruebas se dividió en dos ejecuciones enfocadas a probar los diferentes módulos del sistema.

En la primera prueba, se evaluó el **correcto funcionamiento** del **prototipado de la arquitectura distribuida**. Se ejecutó un ciclo nominal durante el tiempo suficiente para poder observar tres procesos de mantenimiento del tiempo real. El banco de pruebas comprueba en cada caso la validez de las variables de tiempo mantenidas tanto en las *DOT* como en el Módulo-X.

5.3.1.1.1 Validación del prototipado de arquitectura distribuida:

1. T_0: tras la inicialización asíncrona, cada *DOT* debe enviar un mensaje de activación (Wake-Up) a través del CAN BUS con su propio identificador.
2. T_0 + 10 ms: el Módulo-X transmite el Tiempo Real Dorado al bus CAN. Cada *DOT* debe establecer su *ORT* al enviado por el Módulo-X.
3. T_0 + 20ms: X-Module provoca un mensaje de votación falso para forzar el proceso RTM entre todos los *DOT* despertados. El proceso debe tener éxito y confirmar el Tiempo Real Dorado.
4. T_0 + 80 ms: en este punto, el Tiempo Real Dorado se incrementa en un segundo, y todas las *DOT* deben iniciar automáticamente el proceso RTM. El resultado de GRT de este proceso debe ser igual al Tiempo Real Dorado.
5. T_0 + 1024 ms: cada *DOT* debe enviar un mensaje de Alive a través del bus CAN con su propia identificación.
6. T_0 + 1080ms: en este punto, el Tiempo Real Dorado se incrementa en un segundo, y todas las *DOT* deben iniciar automáticamente el proceso RTM. El resultado de GRT de este proceso debe ser igual al Tiempo Real Dorado.

7. T_0 + 2048ms: cada DT debe enviar un mensaje en vivo a través del bus CAN con su propia identificación.
8. T_0 + 2080ms: en este punto, el Tiempo Real Dorado se incrementa en un segundo, y todos los *DOT* deben iniciar automáticamente el proceso RTM. El resultado de GRT de este proceso debe ser igual al Tiempo Real Dorado.
9. T_0 + 2150ms: fin.

En la segunda prueba, se verificó las **capacidades de vigilancia** del **Módulo-X**. El objetivo de la prueba es validar los algoritmos del Módulo-X para la vigilancia del sistema, provocando errores en diferentes *DOT* que deben ser detectados e informados por el Módulo-X. Esto ha sido posible gracias a la alta capacidad de control y observabilidad que proporciona la simulación hardware. Se diseñaron y ejecutaron los siguientes casos de prueba:

- Caso 1. Verificar la capacidad de vigilancia de dff (error funcional de *DOT*). Para ello, se provocó un pulse de reset asíncrono a la DOT1 en medio de la ejecución, lo que provoca la inicialización de este *DOT*.
- Caso 2. Verificar la capacidad de vigilancia drtf (error de *DOT* de tiempo real). En este caso, se cambia el *ORT* de la DOT1 por un tiempo erróneo. El Módulo-X detectará el tiempo erróneo en este *DOT* durante el primer ciclo de proceso RTM.
- Caso 3. Verificar la capacidad de vigilancia sf (fallo del sistema) del Módulo-X Para ello, todas los *DOT* activas se configuran con un ORT erróneo. Esta situación provoca un GRT diferente del Tiempo Real Dorado, en la primera ejecución del ciclo de proceso RTM.

5.3.1.1.2 Validación del Módulo-X

1. T_0: la señal de "reset" del DOT1 se mantiene activa durante los primeros 5 ms. El Módulo-X debe señalizar dos errores funcionales de DOT1 (dff). El primero en T_0, cuando DOT1 no envía el mensaje de Wake-Up. El segundo en T_0 + 5 ms, cuando se recibe un mensaje de Wake-Up desde DOT1.

2. T_0 + 15ms: se introduce un mensaje falso en tiempo real en DOT1, lo que provoca que esta *DOT* adquiera un tiempo real propio incorrecto.
3. T_0 + 20 ms: el Módulo-X envía un mensaje falso de votación en tiempo real para forzar la ejecución del proceso RTM entre todas los *DOT* despiertas. En este caso DOT2, DOT3 y DOT4 tienen el mismo Tiempo Real Propio que el Tiempo Real Dorado, y la DOT1 tiene un Tiempo Real Propio erróneo. Una vez que se completa el proceso de RTM, el GRT debe ser igual al Tiempo Real Dorado, sin embargo, el Módulo-X informará un error de tiempo real de *DOT* (drtf) al Controlador del Emulador.
4. T_0 + 70 ms: en este punto, el DOT1 ha perdió su ORT, porque en el proceso anterior de RTM su tiempo real era incorrecto y, por lo tanto, la DOT1 descartó su ORT. Solo DOT2, DOT3 y DOT4 mantienen el tiempo real en el siguiente ciclo. El banco de pruebas introduce un tiempo real falso en las DOT2 y DOT3. En este momento, DOT2 y DOT3 tienen el mismo tiempo real, pero diferente del Tiempo Real Dorado. Y DOT4 tiene el mismo ORT que el Dorado.
5. T_0 + 80 ms: en este punto, el Tiempo Real Dorado se incrementa en un segundo y DOT4 inicia el proceso de mantenimiento de tiempo real. Como DOT2 y DOT3 tienen un tiempo real diferente, responden con un mensaje de Votación de tiempo real con un tiempo real diferente de DOT4. Dado que una mayoría de *DOT* tiene el mismo tiempo real, el proceso se da por concluido, dando por bueno el tiempo real de las DOT2 y DOT3. Sin embargo, el Módulo-X mantiene el Tiempo Real Dorado, y al comparar el resultado del proceso de RTM con el Tiempo Real Dorado, comprueba que son diferentes y señala un error del sistema (sf) hacia el controlador del emulador.

Como se puede observar, todos los casos de prueba se realizaron con éxito, validando de esta forma las diferentes capacidades implementadas en el Módulo-X

5.3.1.2 Emulación

Durante esta primera prueba de emulación realizada, se aplicó una operación típica del computador dentro del satélite OPTOS. En cuanto a los resultados, sólo se esperaban algunas diferencias leves en el tiempo real de los *DOT*, debido a las pequeñas diferencias

entre los osciladores de cada unidad. Al no inyectarse fallos en esta prueba, no se esperaban otras fuentes de errores.

Se realizó una campaña de emulación de 180 minutos, basada en la recopilación de datos estadísticos suficientes para concluir que el ecosistema estaba funcionando correctamente. Toda la prueba se ejecutó de forma autónoma. Durante la ejecución de la emulación, las siguientes acciones tienen lugar entre las 4 *DOT*:

- Más de diez mil procesos de *RTM* (Mantenimiento del Tiempo Real) entre las *DOT*.
- Cada *DOT* ejecuta al menos mil ciclos de encendido / apagado.
- El sistema de emulación ejecuta 21 x 10^{18} ciclos de reloj.

Los resultados de la emulación no presentan variación con respecto al comportamiento nominal esperado. No se encuentra ninguna de las interrupciones funcionales (sf, drtf y dff), no detectándose FF latentes o silenciosos. El tiempo real fue mantenido por el SUT durante toda la campaña de pruebas.

5.3.2 Resultados de la Campaña de Inyección de Fallos

La complejidad de la campaña de inyección de fallos reside no en la funcionalidad de cada *DOT*, sino en el hecho de probar el sistema distribuido en su totalidad. El computador implementa técnicas de endurecimiento colaborativo a nivel sistema y, por lo tanto, los efectos de los SEU inyectados deben seguirse hasta el final de cada ciclo de colaboración (5000 ms). Esto significa que cada inyección de fallos puede funcionar durante millones de ciclos para asegurarse que el error no se propaga al siguiente ciclo de colaboración. Teniendo esto en cuenta, probar cada FF del prototipo del computador, inyectando fallos en cada posible instante del ciclo completo (5000 ms), llevaría más de 20 meses de ejecución continua. Sin embargo, la confiabilidad de las pruebas estadísticas en *PLD* [Leveugle, Calvez, Maistri and Vanhauwaert 2009] está bien probada, por lo que se plantea hacer una campaña donde se alcance al menos el 10% de cada grupo. Por ello, se ha dividido la funcionalidad colaborativa de cada *DOT* en tres grupos funcionales principales:

- Gestión del Tiempo: este grupo comprende aquellos componentes que implementan el mantenimiento del tiempo real.
- Gestión de Comunicaciones: este grupo comprende aquellos componentes a cargo de las comunicaciones a través del protocolo CAN BUS.
- Gestión de *DOT*: este grupo comprende aquellos componentes a cargo del resto de tareas generales de la *DOT*.

La campaña de inyección de fallos se llevó a cabo durante tres meses consecutivos con un total de 558.816.884 fallos inyectados. En términos del porcentaje de FF probados por cada uno de los grupos de funciones mencionados anteriormente, se muestra en la Tabla 10.

Tabla 10. Porcentajes de FF inyectados con fallo del total.

GESTION TIEMPO	GESTION COMUNICACIONES	GESTION DOT	TOTAL
11%	13%	15%	**14%**

Para analizar los efectos de la inyección de fallos a nivel sistema, la clasificación de errores típica utilizada (silencioso, latente, error funcional) se particulariza. Por un lado, un fallo se clasifica como **error funcional** únicamente cuando la tarea crítica falla a nivel del sistema, es decir, cuando el GRT se ha perdido o su valor es erróneo. Por otro lado, un fallo se clasifica como **latente** cuando algún elemento de memoria en una *DOT* almacena un valor incorrecto hasta el final del ciclo. Este error puede además provocar un error funcional a nivel de *DOT*, pero no a nivel sistema. Se realiza una subclasificación más amplia de este tipo de errores, con el fin de poder diferenciar cuales son las áreas más críticas de las *DOT*. Así, se diferenciará entre errores de tipo drtf y los de dff. En realidad, todos los casos de errores latentes son casos de fallo para la *DOT*, pero a la vez son casos de éxito para el sistema colaborativo, pues significa que las técnicas aplicadas han sido capaces de mitigar el error de una unidad para que no se propague al resto del sistema. Finalmente, un error se clasifica como **silencioso**, cada vez que el fallo inyectado desaparece completamente de la *DOT* sin haber causado ningún error funcional.

La Tabla 11 muestra la clasificación de errores para la campaña de inyección de fallos. Durante toda la campaña, ninguno de los fallos inyectados provocó un error de sistema. Este era el resultado esperado, y por tanto los resultados experimentales validan el método de endurecimiento implementado.

Tabla 11. Porcentajes de errores por tipo, provocados durante la campaña completa de emulación.

Total Fallos Inyectados	Silenciosos (%)	Latentes (%)	Error de Sistema (%)
558.816.884	481.184.231 (86,11%)	77.632.653 (13,89%)	**0**

Una clara mayoría de los fallos (86%) desaparecen durante la ejecución del ciclo sin provocar ningún error funcional en las *DOT*. Estos se errores se clasifican como silenciosos. El porcentaje de inyecciones de fallos silenciosas es muy dependiente del tipo de aplicación y las técnicas de codificación utilizadas. Sin embargo, estos resultados están dentro de los márgenes normales comparado con muchas otras campañas de inyección de fallos realizadas con sistemas similares [Entrena et al. 2012, Portela-Garcia, Lindoso, et al. 2012, Portela-García, López-Ongil, García Valderas and Entrena 2011]. El 14% de los errores se clasifican como latentes. Entre estos errores latentes, solo unos pocos producen alguno de los fallos funcionales anteriormente descritos. En la Fig. 43 se desgranan por tipo de fallo funcional, los porcentajes de errores latentes durante toda la campaña de emulación.

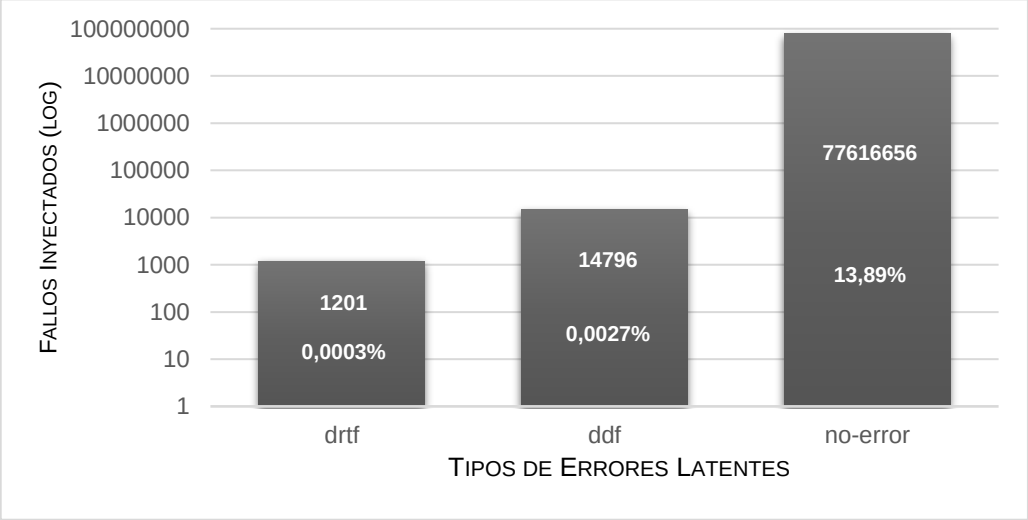


Fig. 43. Distribución de errores latentes por tipo.

Un análisis adicional de los resultados de la campaña de inyección de fallos ha permitido un mejor conocimiento de la sensibilidad a SEU de las *DOT*. Este análisis señala los módulos y FF concretos que son más sensibles a producir errores cuando son alcanzados por partículas energéticas. El uso de este análisis permite endurecer el dispositivo frente a SEU de forma selectiva, reduciendo así la cantidad de recursos a utilizar con el propósito de endurecerlo. Se ha demostrado que las técnicas de redundancia tipo TMR [Abate, Sterpone, Lisboa, Carro and Violante 2009, Sterpone and Battezzati 2010], son fiables frente a SEU no acumulados. Algunos estudios muestran que endurecer una aplicación completa conlleva una sobrecarga de recursos del 300% [Augustin, Gössel and Kraemer 2010], mientras que otros más conservadores aumentan hasta el 400% la sobrecarga para diseños específicos [Bolchini, Miele and Santambrogio 2007, Morgan, McMurtrey, Pratt and Wirthlin 2007]. El diseño actual de *DOT* se implementa con 390 FF de una cantidad total de 512 FF [XILINX 2007] que poseen los dispositivos XC2C512 utilizados en computador de OPTOS. El análisis de FF sensibles arroja que sólo endureciendo el 20% de los FF del dispositivo, se elimina el 95% de los errores producidos (ver Fig. 44). Solamente el 13,89% de los FF (54,17 de 390 FF) produjeron un error latente y, a su vez, sólo endureciendo el 20% de ellos (10,83 de 54,17) conseguimos inmunizar el circuito en el 95% de los casos. Esto implica que, con una baja penalización del 10% (39 de 390 FF), tenemos una gran mejora en la fiabilidad (95%) del circuito.

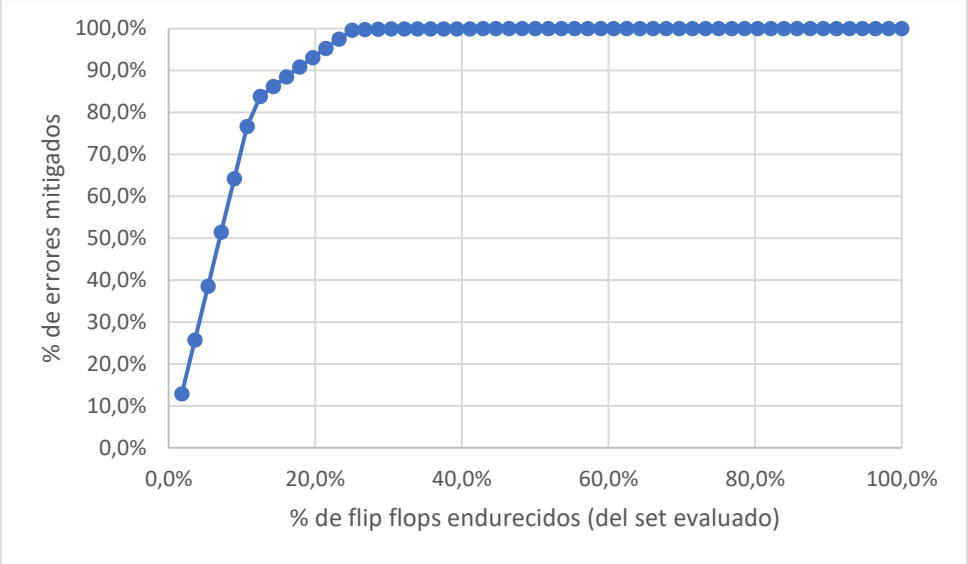


Fig. 44. Comparación entre el porcentaje de errores mitigados frente al de FF endurecidos.

En este capítulo se han descrito las herramientas, procedimientos y conclusiones extraídas del análisis de fiabilidad en Tierra del computador colaborativo. En el siguiente capítulo, se pasa a describir el análisis realizado de la fiabilidad del computador durante los 3 años de misión de OPTOS.

6 *Verificación en Vuelo*

Este capítulo se compone de tres secciones. La primera sección describe cómo el computador colaborativo genera datos, qué tipo de datos y bajo qué circunstancias. La segunda sección describe las herramientas implementadas para analizar los datos recopilados. Finalmente, en la tercera sección se presentan los resultados del análisis.

6.1 Generación de Datos en Vuelo

Para analizar el comportamiento del computador colaborativo con tolerancia a fallos a bordo de un satélite, existen varias consideraciones y dificultades a tener en cuenta. Probablemente la más importante de ellas es la observabilidad del sistema. El sistema que se pretende analizar es a su vez el mismo que generará los datos a estudiar. Por ello, es muy deseable evitar cualquier tipo de procesado en vuelo de los datos a estudiar. Este procesado podría verse a su corrompido por los efectos de la radiación, y crear una ambigüedad entre si el problema fue causado en la ejecución nominal del computador, o a la hora de procesar dicha ejecución. Esto significa que los datos RAW deben enviarse a la Tierra, lo que incrementa considerablemente la cantidad de datos a enviar a Tierra. Este inconveniente se ve incrementado por el hecho de que el *OBC* debe compartir el ancho de banda de las comunicaciones con Tierra con otros 7 subsistemas y 4 cargas útiles. Además, la velocidad del enlace descendente es de 5 Kbps, con solo 3 a 4 contactos con la estación terrestre, con un tiempo de contacto total típico de 20 minutos por día (incluido el enlace ascendente de telecomandos hacia el satélite). Teniendo en cuenta estas limitaciones, se decidió producir un solo tipo de telemetría (es decir, piezas de datos adecuadas para ser almacenadas en la memoria del computador para su posterior descarga a Tierra). En realidad, el OBC produce dos tipos de telemetrías: "OBDH_TM" y "OBDH_DIAG_TM". El contenido de ambas telemetrías es exactamente igual, sin embargo, el momento de generación, así como la duración de estas varían.

El OBC genera de forma nominal un mínimo de 5 mensajes CAN por segundo y un máximo de 16 relacionados con las tareas de endurecimiento colaborativo (léase sección

"Mensajes CAN del Computador" del capítulo 4 Computador basado en Endurecimiento Colaborativo). Esto se debe a que el OBC cuenta con 8 unidades: 7 *DOT* y 1 *EPH*. Cada unidad (incluida el EPH), manda 1 mensaje del tipo "Estado de OBC" por segundo. Además, como parte de la tarea crítica de mantenimiento de tiempo real, se manda un mínimo de 1 mensaje por segundo, y hasta 8 en caso de producirse un ciclo de votación (léase sección 4.4.4 Procesos Colaborativos del capítulo 4 Computador basado en Endurecimiento Colaborativo). Como parte del endurecimiento colaborativo, una regla básica es que un mínimo de 3 *DOT* se mantengan siempre activas a la vez. Por lo tanto, si todas las *DOT* están activas y manteniendo el tiempo real, y se produce un ciclo de votación durante el proceso de *RTM*, el OBC producirá en un segundo un total de 8 mensajes de tipo "Estado de OBC" y 8 mensajes de tipo "Distribución de Tiempo Real" (1 "TD_BROADCAST" y 7 "TD_VOTING"). Por otro lado, si sólo 3 *DOT* están activas y no se produce un ciclo de votación, el OBC producirá en un segundo un total de 4 mensajes de tipo "Estado de OBC" (3 *DOT* y 1 EPH) y 1 mensaje de tipo "Distribución de Tiempo Real" (1 "TD_BROADCAST").

Esta telemetría guarda todos los mensajes generados por el OBC durante el último segundo de cada hora y el primero de la siguiente. Este momento es un momento crítico en el mantenimiento del tiempo real. Durante todo el tiempo, el algoritmo colaborativo de mantenimiento del tiempo real es el encargado de esta tarea, y se realiza de forma distribuida por las 7 *DOT* del computador. Sin embargo, las *DOT* sólo son capaces de almacenar y mantener un total de 69 minutos y 54 segundos (en milisegundos)[15]. Por lo tanto, este tiempo ha de reiniciarse en algún momento previo a esos 69'54", o el tiempo real de las *DOT* se verá desbordado. Es el software de vuelo embarcado en el EPH el encargado de hacerlo. Y el momento elegido es durante el primer segundo de cada hora. Por todo ello, se decidió que los dos minutos más interesantes de cara a ver si las *DOT* habían mantenido el tiempo real y si se producía algún error en el proceso de reinicio del tiempo, eran el último y el primero de cada hora.

[15] Esto se debe a que las DOT mantienen un total de 22 bits. Ya que el tiempo real se mantiene en milisegundos, eso hace un total de 4.194.304 ms. Lo que pasado a horas, minutos y segundos es 1h 9m 54s.

La telemetría nominal del OBC o "OBDH_TM", se guarda automáticamente por el *OBSW* durante dos minutos de cada hora. Teniendo en cuenta que en media el OBC envía 11 mensajes por segundo, este produce un total de 79,2 KBytes (5 bytes por mensaje) por día de información de forma automática. Esto debía suponer, en media, alrededor del 10% del ancho de banda del satélite (768 KBytes por día).

Por otro lado, aunque la órbita de OPTOS (*LEO*) está bastante protegida de las partículas más energéticas por los cinturones de Van Allen y el campo magnético terrestre, la Anomalía del Atlántico Sur (*SAA*) y los polos permiten una mayor irrupción de estas [Stassinopoulos and Raymond 1988], y por tanto, la expectativa de provocar un mayor número de SEU. Para poder aprovechar estas situaciones, se definió la telemetría de tipo "OBDH_DIAG_TM" que, a diferencia de la primera, no se genera de forma automática, sino bajo petición desde Tierra. Esta telemetría se ha utilizado durante las temporadas de verificación del OBC, para activarla al paso del satélite por la SAA y los polos.

Este enfoque ha permitido maximizar la cantidad de datos útiles recuperados para verificar las actividades de colaboración de OBC.

6.2 Herramienta de Análisis

La herramienta de análisis diseñada y desarrollada para estudiar el comportamiento de OPTOS OBC se ha dividido en dos módulos bien diferenciados. El primer módulo, llamado "**TM Decoder**", está destinado a extraer la información útil de los paquetes de datos de telemetría que se transfieren desde OPTOS a la estación de Tierra. Esta herramienta extrae los mensajes CAN, propaga la marca de tiempo de cada mensaje y los almacena en una estructura que es fácil de administrar mediante el segundo módulo.

El segundo módulo se llama "**OBC Simulator**". Este módulo es un software basado en modelos donde cada *DOT* se ha modelado con un conjunto de valores paramétricos. Cada uno de estos modelos puede producir una respuesta determinista no solo en la forma, sino también en el tiempo. Los modelos se alimentan con solo dos entradas:

- **Reloj**: es el tick de cada modelo que simula el oscilador real montado en cada *DOT*. El reloj permite diferencias mínimas para cada *DOT*, ya que cada oscilador tiene pequeñas diferencias en su frecuencia de base, lo que permite la simulación de escenarios más realistas y aleatorios al mismo tiempo.

- **Mensajes CAN**: mensajes generados por las otras unidades que del OBC.

Finalmente, la herramienta de análisis conecta el módulo "TM Decoder" con el módulo "OBC Simulator", alimentándolo con los paquetes de telemetría obtenidos en vuelo. La herramienta de análisis compara la salida de cada modelo de *DOT* con los mensajes reales generados a bordo del satélite. Cuando un mensaje recibido está fuera del comportamiento simulado del modelo, se marca con la consideración de erróneo. Un mensaje erróneo se define por un comportamiento inesperado de la *DOT* debido al efecto de un SEU, o la acumulación de múltiples SEU.

Para validar la herramienta de análisis (entre otras muchas funcionalidades del satélite), se llevó a cabo una prueba de simulación de misión (*MST*) durante la fase de ensamblaje, integración y pruebas (*AIT*) de OPTOS. Durante esta campaña de prueba, la mayoría de los parámetros que afectan la operación del satélite fueron simulados:

- Los períodos de Sol y eclipse se simularon como si se desarrollara una órbita real alrededor de la Tierra. El Sol se simuló con una fuente calibrada del SPASOLAB (Laboratorio de Fotovoltaica Espacial del *INTA*) controlado por una aplicación de determinación orbital. Toda la energía que consumía el satélite provenía de los paneles fotovoltaicos del subsistema de potencia.

- Los contactos con Tierra estaban limitados al paso simulado del satélite por la estación de Tierra, situada en el *INTA*.

- La única conexión de datos con el satélite se realizó a través del subsistema de telemetría y telecomando en RF, por lo que no había ninguna otra conexión eléctrica con este.

El MST se llevó a cabo durante 21 días. Las telemetrías de OBC reunidas se introdujeron en la herramienta de análisis. Sólo 8 mensajes CAN entre más de 332.000 mensajes fueron marcados como erróneos por el Simulador OBC. Después de analizar esos mensajes, todos resultaron ser una falta de comprensión del comportamiento programado de la unidad EPH. El error en el modelo fue corregido. Esta prueba fue útil no solo para validar la herramienta, sino también para ganar más confianza en las técnicas de endurecimiento colaborativo implementadas en la OBC.

6.3 Resultados en Vuelo

Al final de la vida útil el segmento de tierra de OPTOS había recibido un total de 102.439.125 mensajes de CAN, que provenían de las tareas de endurecimiento colaborativo del OBC. Del conjunto completo de mensajes, 280 mensajes han sido categorizados como mensajes erróneos por la herramienta de simulación. La Tabla 12 muestra los números categorizados por *DOT*:

Tabla 12. Tasa de mensajes erróneos totales por unidad.

Unidad	Mensajes Totales	Número de Errores	Tasa de Error
DOT 1	23.991.418	76	3,1678E-06
DOT 2	19.629.342	61	3,10759E-06
DOT 3	10.905.190	22	2,02546E-06
DOT 4	8.724.152	18	2,06324E-06
DOT 5	9.734.432	45	4,62277E-06
DOT 6	14.176.747	28	1,97507E-06
DOT 7	15.277.844	30	1,96363E-06

La relación entre el número de errores y el número de mensajes es muy consistente en DOT 3, 4, 6 y 7. Para DOT 1 y 2, el número de errores aumenta en comparación con el otro *DOT*. La razón por la que esto sucedió es porque DOT 1 y 2 también tienen la máxima prioridad de ID CAN entre todas las unidades. Esto significa que durante la negociación arbitraria del protocolo CAN, cuando dos o más unidades intentan enviar

un mensaje al mismo tiempo de bit, esas unidades de alta prioridad (es decir, DOT 1 y DOT 2) obtendrán acceso al bus. Como se explicó anteriormente, el algoritmo de mantenimiento en tiempo real comienza con cada unidad intentando enviar un mensaje de difusión de tiempo real con su propio tiempo real (*ORT*). Como la mayoría de los mensajes de difusión son enviados por estas unidades, envían más mensajes que las otras, teniendo así más probabilidad de mostrar un fallo funcional provocado por un SEU. Por ello, causan un porcentaje más alto de mensajes erróneos.

La Fig. 45 ilustra muy bien esta situación y la relación entre la cantidad de mensajes enviados y la cantidad de errores detectados.

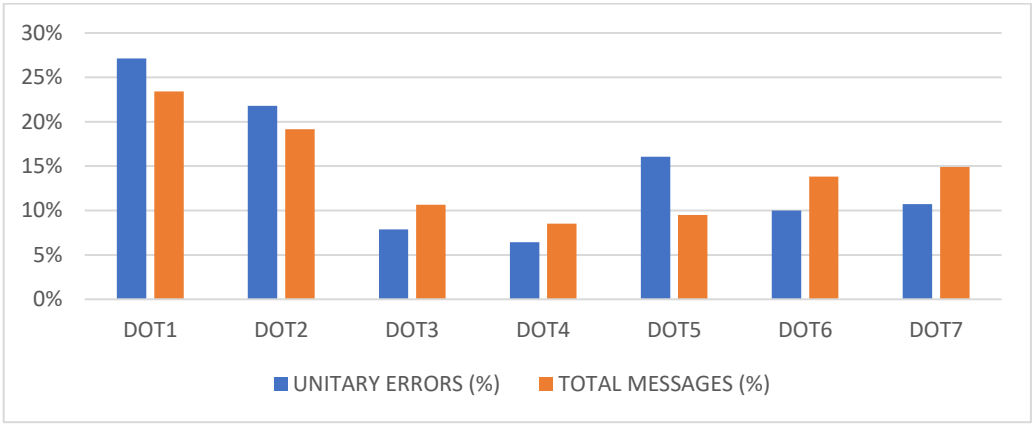


Fig. 45. Porcentaje de mensajes enviados por *DOT*, relacionado con el porcentaje de errores cometidos por *DOT*.

Sin embargo, el algoritmo de mantenimiento en tiempo real no explica por qué la DOT5 también tiene un gran número de errores en comparación con las otras *DOT*. Para encontrar una posible explicación a este problema, hemos estudiado la información de salud del satélite, buscando cualquier cambio significativo en los parámetros medidos de la DOT5. De hecho, solo tres meses después de que OPTOS se inyectara en su órbita, un problema inexplicable comenzó a suceder con el actual mantenimiento del DOT5. Cada *DOT* tiene un consumo de corriente característico que depende de las cargas útiles o subsistemas conectados a ella, así como funcionalidad concreta desempeñada por esta. En realidad, la corriente nominal DOT5 era de 23 mA. Pero tres meses después del lanzamiento, la corriente comenzó a disminuir hasta los 14 mA. Durante la fase de puesta

en servicio[16] del satélite, no se pudo identificar ningún comportamiento inusual del DOT5, sin embargo, el análisis exhaustivo realizado durante los últimos meses y explicado en este documento, muestra que el deterioro en la unidad provocó errores adicionales en toda la misión en esta *DOT* (ver Fig. 46).

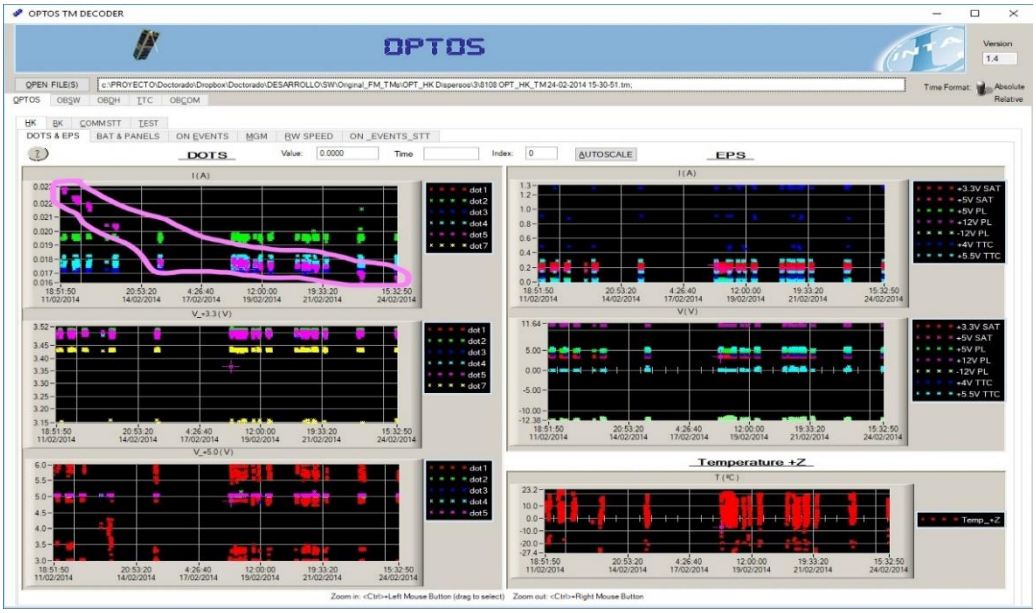


Fig. 46. Análisis de los valores de salud de la DOT5 y el deterioro de sus valores de corriente.

Los mensajes CAN marcados como erróneos se han clasificado de acuerdo con la naturaleza del problema que los causó. Se han definido cinco grupos diferentes:

a) **Error de temporización** *DOT*: este error se señala cuando una *DOT* envía un mensaje de distribución de tiempo real o de votación incorrecto.

b) **Error de gestión de** *DOT*: este error se señala siempre que una *DOT* envía un mensaje de CAN en un intervalo de tiempo inadecuado. El tiempo y el orden de los mensajes que se envían se controlan mediante un módulo VHDL interno que se llama "Gestión de *DOT*".

[16] La puesta en servicio o "commissioning" del satélite, es la fase en la que se hace una verificación funcional completa de extremo a extremo, así como las calibraciones en órbita de los valores de ingeniería.

c) **Error de comunicaciones de** *DOT*: este error comprende aquellos componentes a cargo de las comunicaciones a través del protocolo CAN BUS.

d) **Error genérico de** *DOT*: cualquier otro error encontrado por la herramienta de análisis no categorizado anteriormente se señaliza como error de *DOT*.

e) **Error del sistema**: este error comprende no solo un fallo en una *DOT*, sino también una situación de pérdida de tiempo real en el satélite. Error del sistema significa que las técnicas de endurecimiento colaborativo desarrolladas han fallado en esa situación.

A lo largo de todo el análisis realizado con la telemetría recopilada durante los 3 años de misión de OPTOS, no se encontró ningún Error del Sistema**.** Es decir, **las técnicas de endurecimiento colaborativo evitaron en todos los casos la pérdida de tiempo real**. Este análisis confirma que la fiabilidad del sistema es superior a la suma de la fiabilidad de sus componentes. Se han aplicado con éxito técnicas de endurecimiento en colaboración, para maximizar la fiabilidad y al mismo tiempo reducir el coste, la masa y la potencia, con un OBC fabricado con *COTS*, no específicamente fabricados para ser inmunes a la radiación. Por supuesto, esto nunca podría lograrse si las partes utilizadas fueran muy susceptibles a *SEE* o los efectos de la *TID*.

Con respecto a los errores unitarios encontrados a través del análisis, la distribución real se muestra en la Fig. 6:

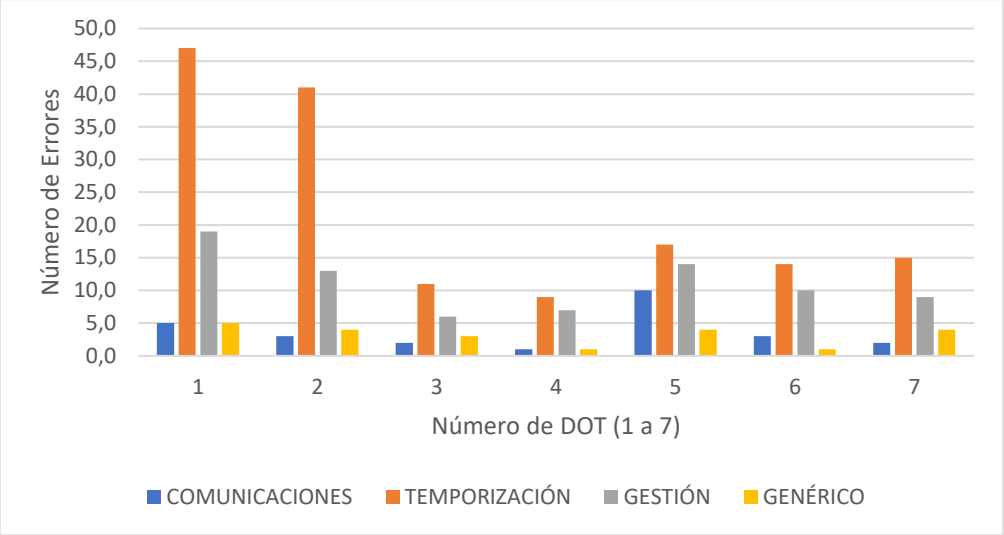


Fig. 47. Distribución de tipos de error por *DOT*.

Como se ve claramente en la gráfica, los errores de **Error de temporización** *DOT* (TIEMPO) son los más predominantes de todos, no solo en las DOT1 y 2 como ya hemos indicado, sino también en el resto de las unidades. Esto se explica por dos razones complementarias:

- El módulo de temporización utiliza una gran cantidad de registros dentro de cada CPLD, y los SEU que golpean esos registros se propagan fácilmente en errores. Este es exactamente el caso opuesto al módulo de comunicación, que se borra en cada ciclo de reloj con nuevos valores que vienen de fuera del CPLD.

- Al mismo tiempo, cada vez que una unidad *DOT* comienza a enviar un mensaje relacionado con el tiempo (ya sea transmisión o votación), si un SEU provoca un error en el módulo de comunicación, la herramienta de análisis entiende que el error proviene del módulo de temporización. No hay manera de diferenciar entre esos errores en tal situación y, por lo tanto, algunos errores de comunicación podrían haberse entendido como errores de tiempo.

Para concluir con el análisis, decidimos incluir la posición de latitud y longitud en cada uno de los mensajes resultantes de errores. Debido al análisis previo de la sección eficaz de ambos dispositivos, los pases del satélite a través de la Anomalía del Atlántico Sur y

por los Polos (especialmente si coinciden con una Llamarada Solar) deberían ser los más favorables para ver los errores producidos por SEU. Así, se creó una herramienta para propagar la posición del satélite desde un archivo TLE proporcionado por *NORAD*. Posteriormente, introdujimos el resultado en una tabla de Google Fusion para poder ver los datos en un archivo de mapa. Como se esperaba, SAA fue el principal foco de los SEU que han afectado al OBC de OPTOS.

Con el objetivo de realizar un análisis diferente, con el fin de encontrar más información sobre los tipos de errores, se decidió incluir dos distribuciones diferentes. En la primera imagen (Fig. 48) cada color de punto representa una unidad diferente (DOT1, 2, etc.), por lo que la distribución es por tipo de unidad. Por otro lado, en la Fig. 49, los puntos han sido categorizados por tipo de error, independientemente de la *DOT* que lo generase.

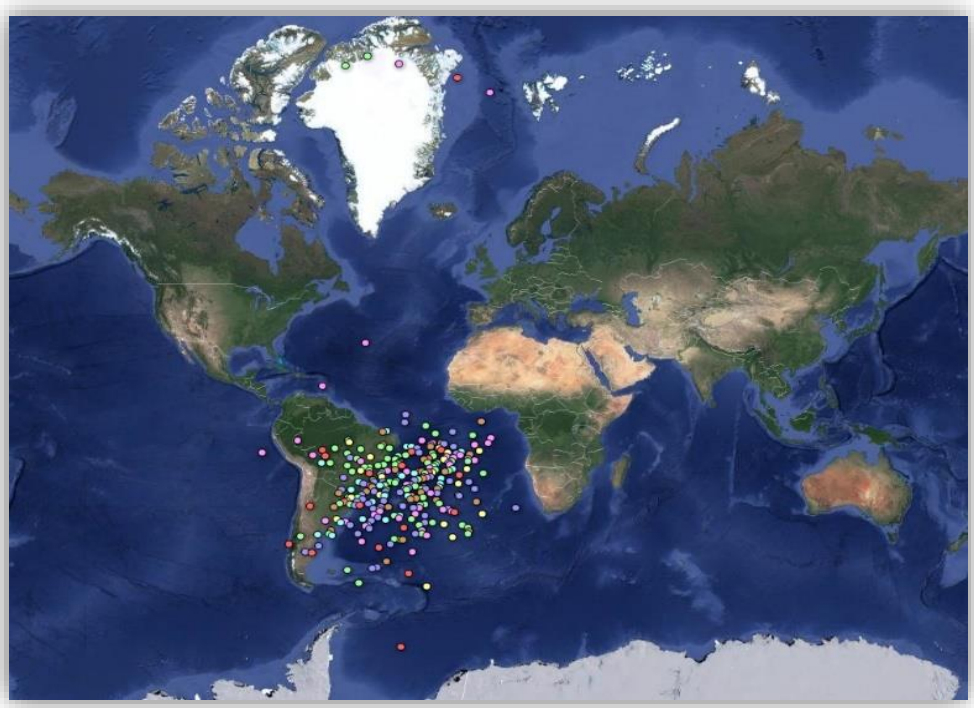


Fig. 48. Mapa del Mundo con una distribución errores por unidad.

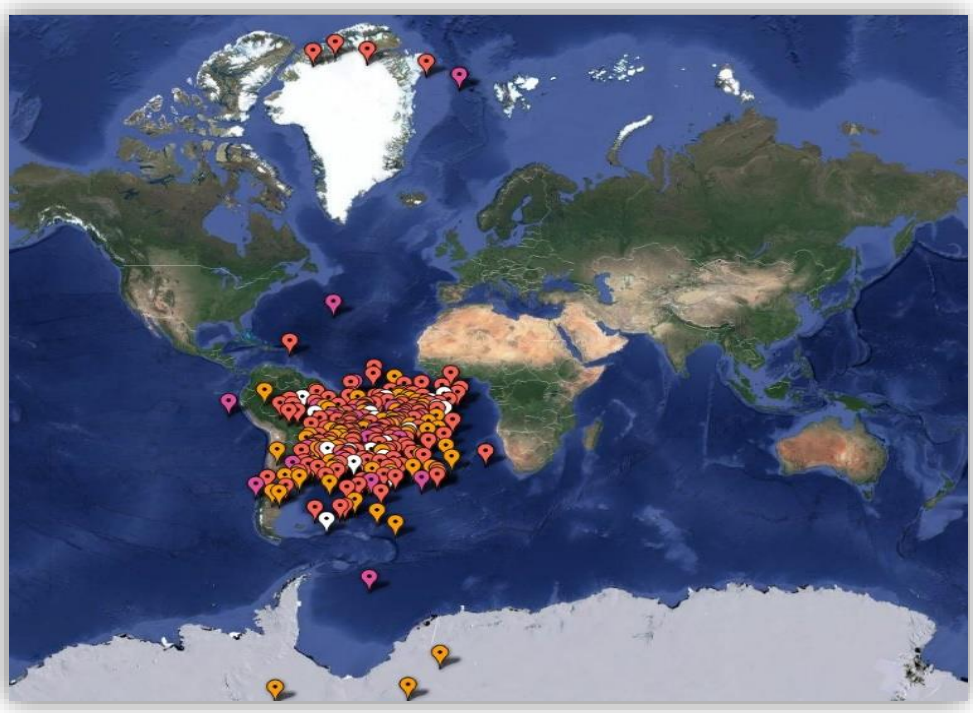


Fig. 49. Mapa del Mundo con una distribución errores por tipo de error.

No se han encontrado diferencias significativas ni por *DOT* ni por tipo de error. La distribución de errores a lo largo de la SAA y los polos es exactamente el comportamiento previsto para un satélite en una órbita LEO como la de OPTOS.

7 Conclusiones

En las casi dos décadas que han transcurrido del presente siglo, hemos comprobado como el número de nano- y pico-satélites que han sido lanzados ha crecido exponencialmente. Sin duda alguna, esta tecnología ha permitido la democratización del espacio, al reducir los costes de lanzamiento hasta unos niveles que los hacen accesibles para universidades y centros de investigación. Más allá del ámbito de la investigación, la posibilidad de lanzar grandes constelaciones de satélites a un precio contenido está dando lugar a nuevas aplicaciones comerciales insospechadas hasta este momento.

Sin embargo, la reducción de los costes de lanzamiento ha puesto el foco en el alto precio de la electrónica específica para el espacio, en especial en el coste de los procesadores empleados para implementar los computadores de vuelo y unidades de control avanzadas. Esta tesis ha abordado este problema, proponiendo una arquitectura distribuida con endurecimiento colaborativo, que es capaz de trabajar con componentes comerciales de propósito general. La principal conclusión la tesis es que, es posible la utilización de componentes *COTS* en el diseño de computadores de vuelo, y en general, en cualquier arquitectura distribuida que cumpla los requisitos especificados. Esto permite mejorar sustancialmente las capacidades de los computadores, manteniendo la fiabilidad del sistema. Para ello, es necesaria la aplicación de las técnicas colaborativas descritas.

Adicionalmente, una importante conclusión es que el bus CAN resulta idóneo para implementar las comunicaciones de este tipo de computadores distribuidos. Implementa un medio físico de difusión, lo que hace que todas las unidades reciban todos los mensajes de forma inmediata, permitiendo así implementar el endurecimiento colaborativo. Esta característica de difusión se complementa además muy bien con las comunicaciones ópticas difusas demostradas en el satélite OPTOS. Además, su protocolo es sencillo, por lo que necesita muy pocos recursos lógicos para ser implementado, y ofrece muy buenas características de protección frente a errores a nivel físico.

A continuación, se presentan las principales contribuciones de los tres principales capítulos de la tesis, el capítulo 4 Computador basado en Endurecimiento Colaborativo, el capítulo 5 Validación en Tierra y el capítulo 6 Verificación en Vuelo.

7.1 Principales contribuciones

El capítulo 4 propone construir un computador embarcado para aplicaciones espaciales empleando mayoritariamente componentes comerciales de bajo coste, usando una arquitectura distribuida con endurecimiento colaborativo. Las principales contribuciones de este capítulo son:

- Diseña una nueva técnica de endurecimiento de computadores, basándose en una serie de algoritmos y procesos colaborativos, creados para ser ejecutados en arquitecturas distribuidas. La técnica propuesta, permite de forma novedosa la utilización de *COTS* para la creación de computadores más baratos y de mejores prestaciones, manteniendo la fiabilidad a nivel sistema.

- Presenta el computador de OPTOS como un caso real de arquitectura distribuida, donde aplica las técnicas de endurecimiento colaborativo diseñadas. Muestra el concepto básico de terminales distribuidos que dan servicio a las diferentes cargas útiles, pero también se encargan de forma colaborativa de realizar las tareas críticas del computador. Además, propone el algoritmo de endurecimiento colaborativo con tres barreras de protección unitarias: algoritmo de apagado, sistema de vigilancia de comunicaciones y sistema de vigilancia hardware.

- Realiza una detallada descripción de las unidades que componen el computador distribuido, constituido por una unidad de altas prestaciones EPH (Enhanced Processing Hardware) y siete unidades de bajo consumo *DOT* (Distributed OBC Terminal). Muestra como las unidades *DOT* pueden construirse con dispositivos CPLD muy sencillos, pero al mismo tiempo, muy efectivos para esta aplicación, ya que la memoria Flash que usan para guardar su configuración es inmune a

radiación, y con un proceso de apagado periódico se consiguen eliminar los errores SEU/SEFI que pudieran haber ocurrido.

- Muestra el proceso para estimar las tasas de SEU y SEFI en las CPLD y FPGA empleadas en el satélite OPTOS, partiendo de la previa caracterización de los componentes frente a radiación, y de un modelo 3D de radiación.

El capítulo 5 propone una metodología para la validación en tierra del computador distribuido con endurecimiento colaborativo, basada en la emulación autónoma de fallos. Las principales contribuciones de este capítulo son:

- El concepto original de un módulo que, unido a una herramienta de emulación autónoma de fallos, tienen la capacidad de validar no sólo unidades concretas, sino arquitecturas distribuidas. El citado Módulo-X, confiere al nuevo ecosistema de validación la inteligencia necesaria para entender de qué manera son mitigados los errores unitarios a nivel sistema, permitiendo la evaluación de arquitecturas distribuidas.

- Se presenta la validación previa a lanzamiento del computador de OPTOS, como caso de uso del ecosistema de emulación de fallos creado. Para ello se prototipa el computador con 4 unidades distribuidas, que representan al sistema bajo pruebas. Completan el ecosistema de validación el Módulo-X y la herramienta de emulación autónoma de fallos. La herramienta de emulación se encarga de la inyección de fallos en el *SUT*. El Módulo-X, se encarga de verificar el funcionamiento del sistema en tiempo real, comprobando la correcta ejecución de las tareas críticas del computador.

- Se presenta un completo análisis de la extensiva campaña de pruebas realizada, donde se demuestra la robustez del sistema frente a errores. Se comprueba como en los casos donde errores latentes se manifiestan en forma de errores funcionales de los terminales distribuidos, o bien en forma de errores en la gestión del tiempo real, estos errores no escalan a errores a nivel de sistema. Es decir, se valida la propuesta de endurecimiento colaborativo para computadores embarcados distribuidos.

- Como parte del análisis realizado, se demuestra que endureciendo un porcentaje pequeño de flip-flops de cada terminal distribuido (10%), se mejora de manera significativa (95% de inmunidad) la fiabilidad de estos. Lo que significa que no es necesario aplicar técnicas TMR de manera global, sino únicamente en los puntos más sensibles.

El capítulo 6, realiza un completo análisis del comportamiento en vuelo del sistema propuesto, validando las hipótesis de diseño y validación. Sus principales contribuciones son:

- Propone e implementa una solución para la monitorización de los errores en el computador embarcado en el satélite OPTOS, basada en almacenar únicamente los mensajes CAN intercambiados por el computador distribuido que se corresponden con el endurecimiento colaborativo. Esta estrategia responde al hecho de que el ancho de banda de las comunicaciones entre satélite y tierra es muy reducido (5 Kb/s) y que el enlace solo está disponible unos 20 minutos por día.

- Diseña y desarrolla una herramienta para estudiar el comportamiento del computador embarcado en el satélite OPTOS. Este se compone de dos módulos bien diferenciados: el decodificador de telemetrías, que extrae la información útil de los paquetes de datos obtenidos en vuelo; y el simulador OBC, que modeliza mediante software el computador embarcado, y compara los comportamientos observados con los que idealmente deberían ocurrir, para así detectar errores de funcionamiento y categorizarlos.

- Realiza un completo análisis de los errores detectados, categorizándolos y geolocalizándolos, y lleva a cabo una discusión de los resultados obtenidos, explicando cuáles son las razones para la mayor o menor prevalencia de cierto tipo de errores en cada unidad.

- Demuestra como la arquitectura de endurecimiento colaborativo es capaz de conseguir que la tasa de fallos del sistema sea menor que la de los componentes que lo constituyen. Así mismo demuestra cómo esta arquitectura es capaz de

evitar que los errores se propaguen en el sistema, permitiendo que el mantenimiento de las tareas críticas operase sin fallo durante los 3 años de vida del satélite OPTOS.

7.2 Trabajo futuro

El planteamiento original de esta tesis plantea la posibilidad de crear arquitecturas distribuidas para computadores de satélites, que permitan mejorar sus capacidades mediante la utilización de *COTS*, manteniendo la fiabilidad de las tareas críticas. La creciente demanda de constelaciones de pequeños satélites en el espacio podría suponer el siguiente paso en la aplicación de las técnicas colaborativas desarrolladas en este trabajo de tesis doctoral. Investigar la forma de aplicar estas técnicas a constelaciones completas de satélites, podría suponer un abaratamiento del coste total de las mismas, permitiendo aún así mantener el correcto funcionamiento de las tareas críticas definidas en cada caso. Posiblemente el reto principal a superar por la investigación sería posiblemente la adaptación de las comunicaciones entre los satélites. En la mayoría de las constelaciones de satélites, no hay una conexión directa entre todos los nodos, por lo que se requiere de relés de comunicaciones para interconectar toda la constelación.

Finalmente, se ha comentado al principio de este capítulo las bondades del bus CAN para implementar las comunicaciones en este tipo de computadores distribuidos. Esto se debe en gran parte debido a la topología del bus, donde todas las unidades reciben todos los mensajes a la vez, sin necesidad de pasar los mensajes de unas unidades a otras (topología en malla). Como línea de investigación futura, se plantea la necesidad de poder adaptar los algoritmos colaborativos, a un tipo de topología más común (en anillo o en árbol), que permita la utilización de protocolos de alta velocidad, como por ejemplo Ethernet. Es importante recordar que por diseño Ethernet es un protocolo de difusión, y aunque en la actualidad se use con conmutadores, el estándar permite trabajar con difusión y se podría plantear la evolución hacia comunicaciones ópticas difusas a 100 Mb/s sobre Ethernet.

7.3 Publicaciones

Publicaciones en revista:

Los dos primeros artículos son revistas indexadas en el *JCR*, y se corresponden con los capítulos 5 y 6. El tercer artículo detalla las comunicaciones ópticas difusas empleadas en el satélite OPTOS, y presenta además el computador distribuido que ha sido analizado en esta tesis (capítulo 4). Es una revista auspiciada por el Consejo de Sociedades Aeroespaciales Europeas (CEAS), la Agencia Europea del Espacio (ESA) y el Centro Aeroespacial Alemán (DLR), y publicada por Springer.

- **A. Martín-Ortega**, M. Portela-García, J. R. de Mingo, S. Rodríguez, J. Rivas, S. López-Buedo, C. López-Ongil, "Early SEU sensitivity assessment for collaborative hardening techniques: A case study of OPTOS processing architecture", *Microelectronics Reliability*, Volume 95, Pages 36-47. April 2019. https://doi.org/10.1016/j.microrel.2019.02.009

- **A. Martín-Ortega**, S. Rodríguez, J. R. de Mingo, S. Ibarmia, J. Rivas, S. López-Buedo, C. López-Ongil, M. Portela-García, "Data Analysis and Results of the Radiation-Tolerant Collaborative Computer On-Board OPTOS CubeSat", *International Journal of Aerospace Engineering*, Volume 2019, Article ID 1425892. https://doi.org/10.1155/2019/1425892

- J. Rivas Abalo, J. Martínez Oter, I. Arruego Rodríguez, **A. Martín-Ortega Rico**, J. R. de Mingo Martín, J. J. Jiménez Martín, B. Martín Vodopivec, S. Rodríguez Bustabad, H. Guerrero Padrón, "OWLS as platform technology in OPTOS satellite", *CEAS Space Journal*, Volume 9, Issue 4, pp 543–554. December 2017. https://doi.org/10.1007/s12567-017-0178-0

En congreso:

Ambos artículos están relacionados con el capítulo 4. El primero, aunque se centra en las comunicaciones ópticas difusas empleadas en el satélite OPTOS, describe además el computador distribuido presentado en dicho capítulo. El segundo aplica a nivel de las técnicas de endurecimiento, pero también a nivel del uso de computadores de capacidades intermedias, y cómo este tipo de computadores pueden resolver muy eficientemente cierto tipo de problemas.

- I. Arruego, J. Rivas, J. Martinez, **A. Martin-Ortega**, V. Apestigue, J.R. de Mingo, J. J. Jimenez, F. J. Alvarez, M. Gonzalez-Guerrero, J.A. Dominguez, "Practical application of the Optical Wireless communication technology (OWLS) in extreme environments", *2015 IEEE International Conference on Wireless for Space and Extreme Environments (WiSEE)*, Orlando, FL, 2015, pp. 1-5. https://doi.org/10.1109/WiSEE.2015.7392981

- **A. Martin-Ortega**, I. Arruego, I. Traseira, N. Andres, J. Rivas, J. Jimenez, A. Gonzalo, S. Rodriguez, J.R. de Mingo, F. Esposito, C. Molfese, F. Cozzolino, B. Saggin, D. Scaccabarozzi, A. Zakharov, G. Dolnikov, A. Lyash, I. Kuznetsov, "MicroMED Processing Unit: Enhanced Computing for Planetary Exploration Payloads", *ExoMars Atmospheric Science and Missions Workshop*, Saariselkä, Finland, 26-30 March 2017.

Publicaciones no directamente relacionadas:

El conocimiento adquirido por el autor durante el desarrollo de su tesis ha sido de utilidad para su contribución a la misión ExoMars 2020, tanto a nivel de la caracterización frente a radiación de componentes comerciales (primer artículo) como de la miniaturización de los sistemas (segundo a cuarto artículos).

- P. Manzano, M. Álvarez, S. Sampedro, M. J. Rivas, I. Traseira, J. Manzano, J. R. Mingo, **A. Martín-Ortega**, N. Andrés, "Heavy Ion Latch-up Test on dsPIC Microcontroller to be used in ExoMars 2020 Mission," *2017 IEEE Radiation*

Effects Data Workshop (REDW), New Orleans, LA, 2017.
doi: 10.1109/NSREC.2017.8115481

- I. Arruego, V. Apéstigue, J. Martínez, J. J. Jiménez, **A. Martín-Ortega**, J. R. de Mingo, M. González-Guerrero, J. Azcue, N. Andrés, F.J. Álvarez, J. Rivas, J. Manzano, F. Serrano, I. Martín, A. Gonzalo, H. Guerrero, S. Espejo, J. Ceballos, A. Ragel, D. Vázquez, F. López, A. J. de Castro, F. Cortés, "InMARS: a comprehensive program for the development of compact atmospheric probes for Mars", *15th International Planetary Probe Workshop (IPPW-2018)*, Boulder, Colorado. June 11-15, 2018.

- V. Apéstigue, I. Arruego, J. Martínez, J.J. Jiménez, J. Rivas, M. González-Guerrero, F.J. Álvarez, J. Azcue, **A. Martín-Ortega**, J.R. de Mingo, A. Gonzalo, B. Martín, N. Andrés, L. Bastide, A. Carretero, I. Martín, M.A. Alcacera, J. Manzano, S. Aparicio, R. López Heredero, M. T. Álvarez, P. Manzano, J. Boland, R. Urqui, J.A. Rodríguez Manfredi, "Mars 2020 Radiation and Dust Sensor Technical Overview", *The Eight Moscow Solar System Symposium*, October 2017.

- V. Apéstigue, I. Arruego, J. Martínez, J.J. Jiménez, J. Rivas, M. González, J. Álvarez, J. Azcue, **A. Martín-Ortega**, J.R. de Mingo, M. T. Álvarez, L. Bastide, A. Carretero, A. Santiago, I. Martín, B. Martín, M.A. Alcacera, J. Manzano, T. Belenger, R. López, D. Escribano, P. Manzano, J. Boland, E. Cordoba, A. Sánchez-Lavega, S. Pérez, A. Sainz López, M. Lemmon, M. Smith, C. E. Newman, J. Gómez Elvira, N. Bridges, P. Conrad, M. de la Torre Juarez, R. Urqui, J.A. Rodríguez Manfredi, "Radiation and Dust Sensor for MARS2020: technical design and development status overview", *European Planetary Science Congress 2015*, Nantes, France, 27 September - 2 October, 2015.

Acrónimos y Terminología

ACK	*Acknowledge*	Acuse de Recibo
ADCS	*Attitude Determination & Control Sub-system*	Subsistema de Determinación y Control de Actitud
AIT	*Assembly, Integration and Test*	Ensamblaje, Integración y Pruebas
APIS	*Athermalized Panchromatic Image Sensor*	Sensor de Imagen Pancromático Atermalizado
CERN	*European Organization for Nuclear Research*	Organización Europea para la Investigación Nuclear
CME	*Coronal Mass Ejection*	Expulsión de Masa Coronal
COTS	*Commercial off the shelf*	Comercial o de Caja
CPLD	*Complex Programmable Logic Device*	Dispositivo Lógico Programable Complejo
CRC	*Code Redundancy Check*	Código de Redundancia Cíclica
CUT	*Circuit Under Test*	Circuito a Probar
DDD	*Displacement Damage Dose*	Dosis de Daños por Desplazamiento
dff	*DOT functional failure*	Error funcional de DOT en el contexto de tipificación de errores del Módulo-X
DOT	*Distributed OBC Terminal*	Unidad Distribuida del Computador
drtf	*DOT real time failure*	Error de tiempo real en DOT en el contexto de tipificación de errores del Módulo-X
DUT	*Device Under Test*	Dispositivo en Pruebas
DWC	*Duplication With Comparison*	Duplicación con Comparación
EBB	*Elegant Bread Board*	Prototipo Elegante
ECSS	*European Cooperation for Space Standardization*	Cooperación Europea para la Normalización del Espacio

EM	*Engineering Model*	Modelo de Ingeniería
ESA	*European Space Agency*	Agencia Espacial Europea
EPH	*Enhanced Processing Hardware*	Unidad de Procesado Avanzado
EPS1	*Electronic Power Subsystem 1*	Subsistema 1 de Distribución de Potencia
EPS2	*Electronic Power Subsystem 2*	Subsistema 2 de Distribución de Potencia
EOL	*End Of Life*	Final de Vida útil
FF	*Flip-Flop*	Registro de memoria
FIBOS	*Fiber Bragg Gratings for Optical Sensing*	Rejillas de Fibra Bragg para la Detección Óptica
FPGA	*Field Programming Gate Array*	Matriz de Puertas Programables por Campo
IMU	*Inertial Measuring Unit*	Unidad de Medida Inercial
INTA	*National Institute of Aerospace Technology*	Instituto Nacional de Técnica Aeroespacial
JCR	*Journal Citation Record*	*Registro de Citaciones en Revista*
GCR	*Galactic Cosmic Rays*	Rayos Cósmicos Galácticos
GMR	*Giant Magneto-Resistance*	Magneto-Resistencia Gigante
GNSS	*Global Navigation Satellite System*	Sistema Global de Navegación por Satélite
GRT	*Global Real Time*	Tiempo Real Global
GSP	*General Studies Programme*	Programa de Estudios Generales
LEO	*Low Earth Orbit*	Orbita Baja Terrestre
LET	*Linear Energy Transfer*	Transferencia Lineal de Energía
lockstep	Los sistemas lockstep son computadores tolerantes a fallos que ejecutan el mismo conjunto de operaciones al mismo tiempo en paralelo.	
MBU	*Multiple Bit Upset*	Trastorno de Múltiples Bits
MST	*Mission Simulation Test*	Prueba de Simulación de Misión
NMR	*N Modular Redundancy*	Redundancia de Módulo de N elemtos

NORAD	*North American Aerospace Defence Command*	Mando Norteamericano de Defensa Aeroespacial
OBC	*On Board Computer*	Computador de Abordo
OBSW	*On Board SoftWare*	Software de Vuelo
ODM	*OPTOS Dose Monitoring*	Monitor de Dosis de OPTOS
ORT	*Own Real Time*	Tiempo Real Propio
OWLS	*Optical Wireless Intra Spacecraft Communications*	Comunicaciones Ópticas Inalámbricas Intra-Satélite
PCB	*Printed Circuit Board*	Tarjeta de Circuito Impreso
PDU	*Power Distribution Unit*	Unidad de Distribución de Potencia
PL	*PayLoad*	Carga Útil
PLD	*Programmable Logic Device*	Dispositivo de Lógica Programable
RadFET	*Radiation-sensitive Field Effect Transistors*	Transistores de Efecto Campo sensibles a Radiación
RadHard	*Radiation Hardened*	Endurecido frente a Radiación. Se dice de aquellos dispositivos que no sufren los efectos de la radiación. Ningún dispositivo es totalmente inmune a la radiación. Todos son inmunes hasta un nivel de energía o de dosis acumulada.
RadTol	*Radiation Tolerant*	Tolerante frente a Radiación. Se dice de aquellos dispositivos o sistemas que no siendo inmunes a la radiación, implementan técnicas HW o SW para mitigar los efectos de la misma.
RTM	*Real Time Maintenance*	Mantenimiento del Tiempo Real
RTR	*Remote Transmission Request*	Petición de Transmisión Remota
RTO	*Real Time Overflow*	Desbordamiento de Tiempo Real
RW	*Reaction Wheel*	Rueda de Reacción
SAA	*South Atlantic Anomaly*	Anomalía del Atlántico Sur
SEE	*Single Event Effect*	Efecto de Evento Singular

SEFI	*Single Event Functional Interrupt*	Interrupción Funcional por Evento Singular
SEL	*Single Event Latch-up*	Evento Destructivo Singular
SET	*Single Event Transient*	Evento Transitorio Singular
SEU	*Single Event Upset*	Trastorno de Evento Singular
sf	*system error*	Error de sistema en el contexto de tipificación de errores del Módulo-X
SRAM	*Static Random-Access Memory*	Memoria Estática de Acceso Aleatorio
SS	*Sub-System*	Subsistema
SUT	*System Under Test*	Sistema a Probar
TID	*Total Ionizing Dose*	Dosis Total Ionizante
TMR	*Triple Modular Redundancy*	Triple Redundancia de Módulo
TPA	*Top Payload Assistance*	Asistente de Carga Útil Superior
TRP	*Technology Research Programme*	Programa de Investigación Tecnológica
TTC	*Telemetry, Tracking and Command*	Sistema de Telemetría y Telecomando
VHDL	*VHSIC (Very High-Speed Integrated Circuit) Hardware Description Language*	Lenguaje de Descripción Hardware para Circuitos Integrados de Alta Velocidad
XRTC	*Xilinx Radiation Test Consortium*	Consorcio de Xilinx para las Pruebas de Radiación

Bibliografía

[Abate, Sterpone, Lisboa, Carro and Violante 2009] Abate, F., Sterpone, L., Lisboa, C. a., Carro, L., Violante, M.: 'New Techniques for Improving the Performance of the Lockstep Architecture for SEEs Mitigation in FPGA Embedded Processors'; IEEE Transactions on Nuclear Science, Vol. 56, No. 4 (2009), pp. 1992–2000. https://doi.org/10.1109/TNS.2009.2013237

[Agostinelli 2003] Agostinelli, S.: 'GEANT4 Collaboration'; Nucl. Instrum. Methods Phys. Res., Sect. A, Vol. 506 (2003), p. 250.

[Allison, Amako, Apostolakis, Araujo, Arce Dubois, Asai, et al. 2006] Allison, J., Amako, K., Apostolakis, J., Araujo, H., Arce Dubois, P., Asai, M., et al.: 'Geant4 developments and applications'; IEEE Transactions on Nuclear Science, Vol. 53, No. 1 (2006), pp. 270–278. https://doi.org/10.1109/TNS.2006.869826

[Apestigue, Arruego, Martínez, Jiménez, Rivas, González, et al. 2015] Apestigue, V., Arruego, I., Martínez, J., Jiménez, J. J., Rivas, J., González, M., et al.: 'Radiation and Dust Sensor For MARS2020: technical design and development status overview'; In EPSC: European Planetary Science Congress (Vol. 10). Nantes (2015). Retrieved from http://adsabs.harvard.edu/abs/2015EPSC...10..813A

[I. Arruego, Guerrero, Rodriguez, Martinez-Oter, Jimenez, Dominguez, et al. 2009] Arruego, I., Guerrero, H., Rodriguez, S., Martinez-Oter, J., Jimenez, J. J., Dominguez, J. A., et al.: 'OWLS: a ten-year history in optical wireless links for intra-satellite communications'; IEEE Journal on Selected Areas in Communications, Vol. 27, No. 9 (2009), pp. 1599–1611. https://doi.org/10.1109/JSAC.2009.091210

[Ignacio Arruego, Martinez and Guerrero 2011] Arruego, I., Martinez, J., Guerrero, H.: 'In-orbit measurement of SET and DD effects on optical wireless links for intra-satellite data transmission'; IEEE Transactions on Nuclear Science, Vol. 58, No. 6 PART 1 (2011), pp. 3067–3075. https://doi.org/10.1109/TNS.2011.2171714

[Augustin, Gössel and Kraemer 2010] Augustin, M., Gössel, M., Kraemer, R.: 'Reducing the area overhead of TMR-systems by protecting specific signals'; In Proceedings of the 2010 IEEE 16th International On-Line Testing Symposium, IOLTS 2010 (2010), pp. 268–273. https://doi.org/10.1109/IOLTS.2010.5560191

[Badhwar 1996] Badhwar, G. D.: 'Galactic Cosmic Radiation Model';, Vol. 17, No. 2 (1996).

[Berg 2007] Berg, M.: 'A New Approach to Single Event Upset Testing of Complex FPGA Designs'; In Single Event Effects Symposium. Long Beach (2007).

[Berg, Buchner, Kim, Friendlich, Perez, Phan, et al. 2009] Berg, M., Buchner, S., Kim, H., Friendlich, M., Perez, C., Phan, A., et al.: 'Enhancing observability of signal composition response and error signatures during dynamic SEE analog to digital device testing'; Proceedings of the European Conference on Radiation and Its Effects on Components and Systems, RADECS, Vol. 57, No. 4 (2009), pp. 313–316. https://doi.org/10.1109/RADECS.2009.5994666

[Bidikar, Rao, Ganesh and Kumar 2014] Bidikar, B., Rao, G., Ganesh, L., Kumar, M.: 'Satellite Clock Error and Orbital Solution Error Estimation for Precise Navigation Applications'; Positioning, Vol. 2014, No. February (2014), pp. 22–26.

https://doi.org/10.4236/pos.2014.51003

[Bock, Dach, Yoon and Montenbruck 2009] Bock, H., Dach, R., Yoon, Y., Montenbruck, O.: 'GPS clock correction estimation for near real-time orbit determination applications'; Aerospace Science and Technology, Vol. 13, No. 7 (2009), pp. 415–422. https://doi.org/10.1016/j.ast.2009.08.003

[Bolchini, Miele and Santambrogio 2007] Bolchini, C., Miele, A., Santambrogio, M. D.: 'TMR and partial dynamic reconfiguration to mitigate SEU faults in FPGAs'; In Proceedings - IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (2007), pp. 87–95. https://doi.org/10.1109/DFT.2007.25

[Brusse 2002] Brusse, J.: 'Tin Whisker Observations on Pure Tin-Plated Ceramic Chip Capacitors'; In Proceedings of the American Electroplaters and Surface Finishers (AESF) SUR/FIN Conference (2002).

[Charle, Viti and Tampere 2011] Charle, W., Viti, F., Tampere, C.: 'Extended Measure for Improved Network Observability Quantification'; In Annual Meeting of Belgian Operational Research Society (ORBEL'25). Belgium (2011), pp. 1–2. Retrieved from https://lirias.kuleuven.be/bitstream/123456789/295391/1/orbel25.pdf

[Chytracek, Mccormick, Pokorski and Santin 2006] Chytracek, R., Mccormick, J., Pokorski, W., Santin, G.: 'Geometry Description Markup Language for Physics Simulation and Analysis Applications'; IEEE Transactions on Nuclear Science, Vol. 53, No. 5 (2006), pp. 2892–2896. https://doi.org/10.1109/TNS.2006.881062

[Correia, Ferro, Junqueira and Serafini 2012] Correia, M., Ferro, D. G., Junqueira, F. P., Serafini, M.: 'Practical hardening of crash-tolerant systems'; Proceedings of the 2012 USENIX Conference on Annual Technical Conference (2012), p. 41. Retrieved from http://dl.acm.org/citation.cfm?id=2342821.2342862

['CubeSat Standard' 1999] (1999). Retrieved from http://www.cubesat.org/about/

[Dressendorfer n.d.] Dressendorfer, P. V: 'Basic mechanisms for the new millennium'; (n.d.).

[ECSS 2008a] ECSS: 'Space engineering: Methods for the calculation of radiation received and its effects, and a policy for design margins' (No. ECSS-E-ST-10-12C); Noordwijk, Netherlands (2008a).

[ECSS 2008b] ECSS: 'Space Engineering: Space environment' (No. ECSS-E-ST-10-04C); Noordwijk, Netherlands (2008b).

[Entrena, Garc??a-Valderas, Fern??ndez-Cardenal, Lindoso, Portela and L??pez-Ongil 2012] Entrena, L., Garc??a-Valderas, M., Fern??ndez-Cardenal, R., Lindoso, A., Portela, M. G., L??pez-Ongil, C.: 'Soft error sensitivity evaluation of microprocessors by multilevel emulation-based fault injection'; IEEE Transactions on Computers, Vol. 61, No. 3 (2012), pp. 313–322. https://doi.org/10.1109/TC.2010.262

[Fatemi, Setoodeh and Haykin 2017] Fatemi, M., Setoodeh, P., Haykin, S.: 'Observability of stochastic complex networks under the supervision of cognitive dynamic systems'; Journal of Complex Networks, Vol. 5, No. 3 (2017), pp. 433–460. https://doi.org/10.1093/comnet/cnw021

[Fayyaz and Vladimirova 2014] Fayyaz, M., Vladimirova, T.: 'Detection of Silent Data Corruption in fault-tolerant distributed systems on board spacecraft'; In Proceedings of the 2014 NASA/ESA Conference on Adaptive Hardware and Systems, AHS 2014

(2014), pp. 202–209. https://doi.org/10.1109/AHS.2014.6880178

[Fayyaz and Vladimirova 2016] Fayyaz, M., Vladimirova, T.: 'Survey and future directions of fault-tolerant distributed computing on board spacecraft'; Advances in Space Research, Vol. 58, No. 11 (2016), pp. 2352–2375. https://doi.org/10.1016/j.asr.2016.08.017

[Fernandes, Santos, Arlindo and Velazco 2007] Fernandes, J. M., Santos, M. B., Arlindo, L., Velazco, R. (TIMA L.: 'Sensitivity to SEUs Evaluation using Probabilistic Testability Analysis at RTL Sensitivity to SEUs Evaluation using Probabilistic Testability Analysis at RTL'; In 8th Latin-American Test Workshop (LATW'07). Cuzco (2007), p. (pp. session-9).

[García, Portela-garcía, López-ongil, Entrena, Martin-ortega, Mingo, et al. 2008] García, M., Portela-garcía, M., López-ongil, C., Entrena, L., Martin-ortega, A., Mingo, J. R. De, et al.: 'The Effects of Proton Irradiation on CoolRunner-II ™ CPLD Technology'; European Conference on Radiation and Its Effects on Components and Systems, , No. 602 (2008), pp. 131–135. https://doi.org/10.1109/RADECS.2008.5944064

[GEANT4-Collaboration 2010] GEANT4-Collaboration: 'Physics Reference Manual'; Manual, Vol. 1 (2010), pp. 1–554.

[GmBH 1998] GmBH, B.: 'Bosch Controller Area Network (CAN) Version 2.0'; Network (1998).

[Goeders, Wirthlin, Quinn, James and Bohman 2018] Goeders, J., Wirthlin, M. J., Quinn, H., James, B., Bohman, M.: 'Microcontroller Compiler-Assisted Software Fault Tolerance'; IEEE Transactions on Nuclear Science, Vol. 66, No. 1 (2018), pp. 223–232. https://doi.org/10.1109/tns.2018.2886094

[Halain, Rochus, Renotte, Auchère, Berghmans, Harra, et al. 2014] Halain, J., Rochus, P., Renotte, E., Auchère, F., Berghmans, D., Harra, L., et al.: 'The Extreme UV Imager of Solar Orbiter – From detailed design to Flight Model'; Proceedings of SPIE, , No. Space Telescopes and Instrumentation 2014: Ultraviolet to Gamma Ray (2014). https://doi.org/10.1117/12.2055207

[Heredero, Frovel, Laguna, Anderson, Garranzo and Belenguer-Dávila 2008] Heredero, R. L., Frovel, M., Laguna, H., Anderson, A., Garranzo, D., Belenguer-Dávila, T.: 'Fiber Bragg gratings for optical sensing (FIBOS) for an aerospace application'; In J. L. Santos, B. Culshaw, J. M. López-Higuera & W. N. MacPherson (Eds.), Fourth European Workshop on Optical Fibre Sensors. Porto: DESTECH PUBLICATIONS, INC, 439 DUKE STREET, LANCASTER, PA 17602-4967 USA (2008), pp. 833–839. https://doi.org/10.1117/12.865347

[Huston and Pfitzer 1998] Huston, S. L., Pfitzer, K. A.: 'A new model for the low altitude trapped proton environment'; IEEE Transactions on Nuclear Science, Vol. 45, No. 6 PART 1 (1998), pp. 2972–2978. https://doi.org/10.1109/23.736554

[Hyman, Bhattacharya and Ranganathan 2011] Hyman, R., Bhattacharya, K., Ranganathan, N.: 'Redundancy mining for soft error detection in multicore processors'; IEEE Transactions on Computers, Vol. 60, No. 8 (2011), pp. 1114–1125. https://doi.org/10.1109/TC.2010.168

[I. Vette 1991a] I. Vette, J.: 'The AE-8 Trapped Electron Model Environment'; NSSDC/WDC-A-R and S 91-24 (1991a).

[I. Vette 1991b] I. Vette, J.: 'The NASA/National Space Science Data Center trapped

radiation environment model program, 1964 - 1991';, Vol. 91–29 (1991b).

['ISO 15390' 2004] (2004). Retrieved from https://www.iso.org/standard/37095.html

[Jurgens 1998] Jurgens, R. F.: 'High-temperature electronics applications in space exploration'; In High-Temperature Electronics (1998). https://doi.org/10.1109/9780470544884.ch21

[L-std- and Mil-std- 2008] L-std-, M. I., Mil-std-, S.: 'Department of Defense Test Method Standard Environmental Engineering Considerations and Laboratory Tests'; Mil-Std-810G (2008). https://doi.org/10.4271/2001-01-1104

[Laubier, Bodin, Pasquier, Fredon, Levacher, Vola, et al. 2017] Laubier, D., Bodin, P., Pasquier, H., Fredon, S., Levacher, P., Vola, P., et al.: 'The PLATO Camera'; In International Conference on Space Optics — ICSO. Corsica, France (2017). https://doi.org/https://doi.org/10.1117/12.2309075

[Leveugle, Calvez, Maistri and Vanhauwaert 2009] Leveugle, R., Calvez, A., Maistri, P., Vanhauwaert, P.: 'Statistical fault injection: Quantified error and confidence'; 2009 Design, Automation & Test in Europe Conference & Exhibition, , No. APRIL 2009 (2009), pp. 502–506. https://doi.org/10.1109/DATE.2009.5090716

[Lewandowski and Thomas 1991] Lewandowski, W., Thomas, C.: 'GPS Time Transfer'; Proceedings of the IEEE, Vol. 79, No. 7 (1991), pp. 991–1000. https://doi.org/10.1109/5.84976

[A. Li and Hong 2007] Li, A., Hong, B.: 'Software implemented transient fault detection in space computer'; Aerospace Science and Technology, Vol. 11, No. 2–3 (2007), pp. 245–252. https://doi.org/10.1016/j.ast.2006.06.006

[Y. Li, Nelson and Wirthlin 2010] Li, Y., Nelson, B., Wirthlin, M.: 'Synchronization techniques for crossing multiple clock domains in FPGA-based TMR circuits'; In IEEE Transactions on Nuclear Science (Vol. 57) (2010), pp. 3506–3514. https://doi.org/10.1109/TNS.2010.2086075

[Liu, Slotine and Barabasi 2013] Liu, Y.-Y., Slotine, J.-J., Barabasi, A.-L.: 'Observability of complex systems'; Proceedings of the National Academy of Sciences, Vol. 110, No. 7 (2013), pp. 2460–2465. https://doi.org/10.1073/pnas.1215508110

[Ĺopez-Ongil, García-Valderas, Portela-García and Entrena 2007] Ĺopez-Ongil, C., García-Valderas, M., Portela-García, M., Entrena, L.: 'Autonomous fault emulation: A new FPGA-based acceleration system for hardness evaluation'; IEEE Transactions on Nuclear Science, Vol. 54, No. 1 (2007), pp. 252–261. https://doi.org/10.1109/TNS.2006.889115

[M. Sawyer and I. Vette 1977] M. Sawyer, D., I. Vette, J.: 'AP8 Trapped Proton Environment for Solar Maximum and Solar Minimum'; (1977).

['Manchester Coding' 2018] (2018). Retrieved from https://en.wikipedia.org/wiki/Manchester_code

[Martín-Ortega, Rodríguez, de Mingo, Ibarmia, Rivas, López-Buedo, et al. 2019] Martín-Ortega, A., Rodríguez, S., de Mingo, J. R., Ibarmia, S., Rivas, J., López-Buedo, S., et al.: 'Data Analysis and Results of the Radiation-Tolerant Collaborative Computer On-Board OPTOS CubeSat'; International Journal of Aerospace Engineering, Vol. 2019 (2019), pp. 1–11. https://doi.org/10.1155/2019/1425892

[Michelena 2013] Michelena, M. D.: 'Commercial Off-The-Shelf GMR Based Sensor on Board Optos Picosatellite'; In Giant Magnetoresistance (GMR) Sensors: From Basis to State-of-the-Art Applications. Berlin, Heidelberg: Springer Berlin Heidelberg (2013), pp. 181–210. https://doi.org/10.1007/978-3-642-37172-1_8

[Michelena, Cerdán and Arruego 2009] Michelena, M. D., Cerdán, M. F., Arruego, I.: 'NANOSAT-01: Three Years of Mission. Magnetic Scientific Results'; Sensor Letters, Vol. 7, No. 3 (2009), pp. 412–415. https://doi.org/10.1166/sl.2009.1069

[Morgan, McMurtrey, Pratt and Wirthlin 2007] Morgan, K. S., McMurtrey, D. L., Pratt, B. H., Wirthlin, M. J.: 'A Comparison of TMR With Alternative Fault-Tolerant Design Techniques for FPGAs'; IEEE Transactions on Nuclear Science, Vol. 54, No. 6 (2007), pp. 2065–2072. https://doi.org/10.1109/TNS.2007.910871

['NewSpace' 2019] (2019). Retrieved from https://en.wikipedia.org/w/index.php?title=NewSpace&oldid=877852115

[Ott, Jin, Chuska and LaRocca 2006] Ott, M., Jin, X., Chuska, R., LaRocca, F.: 'Photonic component qualification and implementation activities at NASA Goddard Space …'; Photonics for Space Environments XI: 14-15 August (2006). https://doi.org/10.1117/12.682259

[Parra Avellaneda 2017] Parra Avellaneda, L. I.: 'Técnicas híbridas de tolerancia a fallos en microprocesadores'; Universidad Carlos III de Madrid (2017). Retrieved from https://e-archivo.uc3m.es/handle/10016/26531

[Parra, Lindoso, Portela, Entrena, Restrepo-Calle, Cuenca-Asensi and Mart??nez-??lvarez 2014] Parra, L., Lindoso, A., Portela, M., Entrena, L., Restrepo-Calle, F., Cuenca-Asensi, S., Mart??nez-??lvarez, A.: 'Efficient mitigation of data and control flow errors in microprocessors'; IEEE Transactions on Nuclear Science, Vol. 61, No. 4 (2014). https://doi.org/10.1109/TNS.2014.2310492

[Petrick, Espinosa, Ripley, Crum, Geist and Flatley 2014] Petrick, D., Espinosa, D., Ripley, R., Crum, G., Geist, A., Flatley, T.: 'Adapting the reconfigurable spacecube processing system for multiple mission applications'; In IEEE Aerospace Conference Proceedings (2014). https://doi.org/10.1109/AERO.2014.6836227

[Piet, Bourdarie, Boscher and Friedel 2006] Piet, A. S., Bourdarie, S., Boscher, D., Friedel, R. H. W.: 'A Model for the Geostationary Electron Environment: POLE, From 30 keV to 5.2 MeV.'; IEEE Transactions on Nuclear Science, Vol. 53, No. 4 (2006), pp. 1844–1850.

[Portela-Garcia, Grosso, Gallardo-Campos, Sonza Reorda, Entrena, Garcia-Valderas and Lopez-Ongil 2012] Portela-Garcia, M., Grosso, M., Gallardo-Campos, M., Sonza Reorda, M., Entrena, L., Garcia-Valderas, M., Lopez-Ongil, C.: 'On the use of embedded debug features for permanent and transient fault resilience in microprocessors'; Microprocessors and Microsystems, Vol. 36, No. 5 (2012), pp. 334–343. https://doi.org/10.1016/j.micpro.2012.02.013

[Portela-Garcia, Lindoso, Entrena, Garcia-Valderas, Lopez-Ongil, Marroni, et al. 2012] Portela-Garcia, M., Lindoso, A., Entrena, L. F., Garcia-Valderas, M., Lopez-Ongil, C., Marroni, N., et al.: 'Evaluating the effectiveness of a software-based technique under SEEs using fpga-based fault injection approach'; Journal of Electronic Testing: Theory and Applications (JETTA), Vol. 28, No. 6 (2012), pp. 777–789. https://doi.org/10.1007/s10836-012-5321-4

[Portela-García, López-Ongil, García Valderas and Entrena 2011] Portela-García, M., López-Ongil, C., García Valderas, M., Entrena, L.: 'Fault injection in modern microprocessors using on-chip debugging infrastructures'; IEEE Transactions on Dependable and Secure Computing, Vol. 8, No. 2 (2011), pp. 308–314. https://doi.org/10.1109/TDSC.2010.50

[Pourrouquet, Thomas, Peyrard, Ecoffet and Rolland 2010] Pourrouquet, P., Thomas, J., Peyrard, P., Ecoffet, R., Rolland, G.: 'FASTRAD 3.2: Radiation Shielding Tool with a New Monte Carlo Module'; In 2011 IEEE Radiation Effects Data Workshop (2010), pp. 1–5. https://doi.org/10.1109/REDW.2010.6062530

[Praprotnik, Gartner, Zauner and Horauer 2009] Praprotnik, O., Gartner, M., Zauner, M., Horauer, M.: 'A test suite for system tests of distributed automotive electronics'; 2nd International Conference on Advances in Circuits, Electronics and Micro-Electronics - CENICS 2009 (2009), pp. 67–70. https://doi.org/10.1109/CENICS.2009.11

[Pratt, Caffrey, Graham, Morgan and Wirthlin 2006] Pratt, B., Caffrey, M., Graham, P., Morgan, K., Wirthlin, M.: 'Improving FPGA design robustness with partial TMR'; In IEEE International Reliability Physics Symposium Proceedings (2006), pp. 226–232. https://doi.org/10.1109/RELPHY.2006.251221

[Quinn, Black, Robinson and Buchner 2013] Quinn, H. M., Black, D. A., Robinson, W. H., Buchner, S. P.: 'Fault simulation and emulation tools to augment radiation-hardness assurance testing'; IEEE Transactions on Nuclear Science, Vol. 60, No. 3 (2013), pp. 2119–2142. https://doi.org/10.1109/TNS.2013.2259503

[Reames 1999] Reames, D. V: 'Particle acceleration at the Sun and in the heliosphere'; Space Science Reviews, Vol. 90, No. 3 (1999), pp. 413–491. https://doi.org/10.1023/A:1005105831781

[R. Reed 2006] Reed, R.: 'Modeling the Space Radiation Environment and Effects on Microelectronic Devices and Circuits'; 2006 NSREC Short Course Notebook Section (2006). Retrieved from http://ci.nii.ac.jp/naid/10026164313/en/

[R. A. Reed and Ladbury 2000] Reed, R. A., Ladbury, R. L.: 'Performance Characterization of Digital Optical Data Transfer Systems for Use in the Space Radiation Environment'; (2000). Retrieved from http://www.archive.org/details/nasa_techdoc_20000093961%5Cnhttp://ia700602.us.archive.org/13/items/nasa_techdoc_20000093961/20000093961.pdf

[Restrepo-Calle, Cuenca-Asensi and Martínez-Álvarez 2015] Restrepo-Calle, F., Cuenca-Asensi, S., Martínez-Álvarez, A.: 'Reducing implicit overheads of soft error mitigation techniques using selective hardening'; In FPGAs and Parallel Architectures for Aerospace Applications: Soft Errors and Fault-Tolerant Design (2015). https://doi.org/10.1007/978-3-319-14352-1_17

[Rivas Abalo, Martínez Oter, Arruego Rodríguez, Martín-Ortega Rico, de Mingo Martín, Jiménez Martín, et al. 2017] Rivas Abalo, J., Martínez Oter, J., Arruego Rodríguez, I., Martín-Ortega Rico, A., de Mingo Martín, J. R., Jiménez Martín, J. J., et al.: 'OWLS as platform technology in OPTOS satellite'; CEAS Space Journal (2017). https://doi.org/10.1007/s12567-017-0178-0

[Rivas, Martinez-Oter, Arruego, Martin-Ortega, De Mingo, Jimenez and Martin 2017] Rivas, J., Martinez-Oter, J., Arruego, I., Martin-Ortega, A., De Mingo, J. R., Jimenez, J. J., Martin, B.: 'Owls as platform technology in OPTOS satellite'; In Proceedings of SPIE

- The International Society for Optical Engineering (Vol. 10562) (2017). https://doi.org/10.1117/12.2296165

[She and Samudrala 2009] She, X., Samudrala, P. K. K.: 'Selective Triple Modular Redundancy for Single Event Upset (SEU) Mitigation'; In Proceedings - 2009 NASA/ESA Conference on Adaptive Hardware and Systems, AHS 2009. Ieee (2009), pp. 344–350. https://doi.org/10.1109/AHS.2009.9

[Shea and Smart 1995] Shea, M. A., Smart, D. F.: 'A comparison of energetic solar proton events during the declining phase of four solar cycles (cycles 19-22)'; Advances in Space Research, Vol. 16, No. 9 (1995), pp. 37–46. https://doi.org/10.1016/0273-1177(95)00312-3

[Siegle, Vladimirova, Poivey and Emam 2015] Siegle, F., Vladimirova, T., Poivey, C., Emam, O.: 'Validation of FDIR strategy for spaceborne SRAM-based FPGAs using proton radiation testing'; In Proceedings of the European Conference on Radiation and its Effects on Components and Systems, RADECS (Vol. 2015–Decem) (2015). https://doi.org/10.1109/RADECS.2015.7365694

[Smith and de la Torre 2006] Smith, G. L., de la Torre, L.: 'Techniques to Enable FPGA Based Reconfigurable Fault Tolerant Space Computing'; 2006 IEEE Aerospace Conference (2006), pp. 1–11. https://doi.org/10.1109/AERO.2006.1655958

[Stassinopoulos and Raymond 1988] Stassinopoulos, E. G., Raymond, J. P.: 'The space radiation environment for electronics'; Proceedings of the IEEE, Vol. 76, No. 11 (1988), pp. 1423–1442. https://doi.org/10.1109/5.90113

[Sterpone and Battezzati 2010] Sterpone, L., Battezzati, N.: 'A new placement algorithm for the mitigation of multiple cell upsets in SRAM-based FPGAs'; Design, Automation & Test in Europe Conference & Exhibition (DATE 2010) (2010), pp. 1231–1236. https://doi.org/10.1109/DATE.2010.5456995

[Swift 2004] Swift, G. M.: 'Virtex-II Static SEU Characterization'; In XILINX SINGLE EVENT EFFECTS (2004).

[Systems n.d.] Systems, P.: 'Peak CAN Bus - USB'; (n.d.). Retrieved 30 April 2016, from http://www.peak-system.com/PCAN-USB.199.0.html

[Tamayo, Alonso, Jimenez, Arruego and Guerrero 2010] Tamayo, R., Alonso, J., Jimenez, J. J., Arruego, I., Guerrero, H.: 'OpticalWireless Links for Intra-Satellite Communications: Reflection Models and Hardware Optimization'; Journal of Aerospace Computing, Information, and Communication, Vol. 7, No. 3 (2010), pp. 118–133. https://doi.org/10.2514/1.30271

[Tirumurti, Karimi, Maniatakos, Jas and Makris 2010] Tirumurti, C., Karimi, N., Maniatakos, M., Jas, A., Makris, Y.: 'Workload-Cognizant Concurrent Error Detection in the Scheduler of a Modern Microprocessor'; IEEE Transactions on Computers, Vol. 60, No. 9 (2010), pp. 1274–1287. https://doi.org/10.1109/tc.2010.265

[Valderas, Garcia, López and Entrena 2009] Valderas, M. G., Garcia, M. P., López, C., Entrena, L.: 'Extensive SEU impact analysis of a PIC microprocessor for selective hardening'; In Proceedings of the European Conference on Radiation and its Effects on Components and Systems, RADECS (2009). https://doi.org/10.1109/RADECS.2009.5994670

[Vaskova, Fabregat, Portela-Garcia, Garcia-Valderas, Lopez-Ongil and Reorda 2014] Vaskova, A., Fabregat, A., Portela-Garcia, M., Garcia-Valderas, M., Lopez-

Ongil, C., Reorda, M. S.: 'Reducing SEU sensitivity in LIN networks: Selective and collaborative hardening techniques'; In LATW 2014 - 15th IEEE Latin-American Test Workshop (2014). https://doi.org/10.1109/LATW.2014.6841924

[Vaskova, Lopez-Ongil, Portela-Garcia, Garcia-Valderas and Entrena 2013] Vaskova, A., Lopez-Ongil, C., Portela-Garcia, M., Garcia-Valderas, M., Entrena, L.: 'SEU sensitivity comparison for different reprogrammable technologies with minority check block'; IEEE Transactions on Nuclear Science, Vol. 60, No. 4 (2013), pp. 2813–2818. https://doi.org/10.1109/TNS.2013.2245343

[Vemu and Abraham 2011] Vemu, R., Abraham, J.: 'CEDA: Control-flow error detection using assertions'; IEEE Transactions on Computers, Vol. 60, No. 9 (2011), pp. 1233–1245. https://doi.org/10.1109/TC.2011.101

[Wältermann, Schütte and Diekstall 2004] Wältermann, D. P., Schütte, D. H., Diekstall, D.-I. K.: 'Hardware-in-the-Loop Testing of Distributed Electronic Systems'; DSpace GmbH (2004), p. 8.

[Washington 2000] Washington, R.: 'On-board real-time state and fault identification for rovers'; Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065), Vol. 2, No. April (2000), pp. 1175–1181. https://doi.org/10.1109/ROBOT.2000.844758

[Wilcox, Woos, Panchekha, Tatlock, Wang, Ernst and Anderson 2015] Wilcox, J. R., Woos, D., Panchekha, P., Tatlock, Z., Wang, X., Ernst, M. D., Anderson, T.: 'Verdi: a framework for implementing and formally verifying distributed systems'; Proceedings of the 36th ACM SIGPLAN Conference on Programming Language Design and Implementation - PLDI 2015 (2015), pp. 357–368. https://doi.org/10.1145/2737924.2737958

[Xapsos, Stauffer, Jordan, Barth and Mewaldt 2007] Xapsos, M. A., Stauffer, C., Jordan, T., Barth, J. L., Mewaldt, R. A.: 'Model for Cumulative Solar Heavy Ion Energy and Linear Energy Transfer Spectra'; IEEE Transactions on Nuclear Science, Vol. 54, No. 6 (2007), pp. 1985–1989. https://doi.org/10.1109/TNS.2007.910850

[Xapsos, Summers, Barth, Stassinopoulos and Burke 1999] Xapsos, M. A., Summers, G. P., Barth, J. L., Stassinopoulos, E. G., Burke, E. A.: 'Probability model for worst case solar proton event fluences'; IEEE Transactions on Nuclear Science, Vol. 46, No. 6 (1999), pp. 1481–1485. https://doi.org/10.1109/23.819111

[Xapsos, Summers, Barth, Stassinopoulos and Burke 2000] Xapsos, M. A., Summers, G. P., Barth, J. L., Stassinopoulos, E. G., Burke, E. A.: 'Probability model for cumulative solar proton event fluences'; IEEE Transactions on Nuclear Science, Vol. 47, No. 3 (2000), pp. 486–490. https://doi.org/10.110923.856469

[Xilinx n.d.] Xilinx: 'Xilinx University Program XUPV5-LX110T'; (n.d.). Retrieved 23 April 2016, from https://www.xilinx.com/univ/xupv5-lx110t.htm

[XILINX 2007] XILINX: 'XC2C512 CoolRunner-II CPLD Data Sheet'; (Vol. 096) (2007). Retrieved from https://www.xilinx.com/support/documentation/data_sheets/ds096.pdf

[Xilinx Inc. 2007] Xilinx Inc.: 'Xilinx DS031 Virtex-II Platform FPGAs: Complete Data Sheet';, Vol. 031 (2007).