



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
PROGRAMA DE MAESTRÍA Y DOCTORADO EN INGENIERÍA
INGENIERÍA ELÉCTRICA – SISTEMAS ELECTRÓNICOS

DESARROLLO DE UN SISTEMA DE COMANDO Y MANEJO DE INFORMACIÓN TOLERANTE A
FALLAS BASADO EN UN FPGA, BAJO EL ESTÁNDAR CUBESAT

TESIS
QUE PARA OPTAR POR EL GRADO DE:
MAESTRO EN INGENIERÍA

PRESENTA:
ING. CHRISTO ALDAIR LARA TENORIO

TUTOR:
DR. SAÚL DE LA ROSA NIEVES
FACULTAD DE INGENIERÍA, UNAM

CDMX, FEBRERO 2024



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

JURADO ASIGNADO:

Presidente: Dr. Rodríguez Cuevas Jorge

Secretario: Dr. Savage Carmona Jesús

1^{er} Vocal: Dr. De La Rosa Nieves Saúl

2^{do} Vocal: Dr. Pérez Alcázar Pablo Roberto

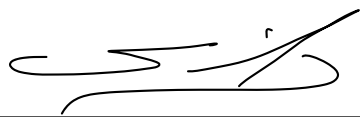
3^{er} Vocal: Dr. Peña Cabrera Juan Mario

LUGAR EN DONDE SE REALIZÓ LA TESIS:

Laboratorio de Instrumentación Electrónica de Sistemas Espaciales (LIESE), Facultad de Ingeniería (FI), Universidad Nacional Autónoma de México (UNAM), México.

TUTOR DE TESIS:

DR. SAÚL DE LA ROSA NIEVES



Firma

AGRADECIMIENTOS

En primer lugar, le agradezco a mi familia: Math, Daniela, Gloria y Martín. Su apoyo, orientación, confianza y amor siempre estarán presentes en mi formación, siendo mi motivación para crecer personal y profesionalmente. Ustedes son los pilares de todo mi esfuerzo y siempre estarán presentes en todo lo que pueda hacer.

A Rebeca, quien diariamente me enseña nuevas perspectivas sobre cómo crecer en múltiples aspectos de mi vida. Tu amor, apoyo, paciencia y compañía incondicional me fortalecen y motivan a esforzarme para lograr las metas que me propongo, las cuales quiero seguir compartiendo contigo.

Al Dr. Saúl de la Rosa Nieves, quien desde el principio me dio la oportunidad de probar todo lo que puedo lograr. Siempre me ha brindado su apoyo, conocimiento y experiencia, lo que me ha servido de guía para crecer, tanto en los proyectos académicos como personales.

A todos mis amigos y colegas del LIESE con los que convivo diariamente y hemos formado una familia, trabajando juntos y apoyándonos mutuamente en los múltiples proyectos del laboratorio.

Al Consejo Nacional de Humanidades, Ciencias y Tecnologías (CONAHCYT), por su apoyo económico durante la realización de este proyecto.

Aldair

ÍNDICE GENERAL

AGRADECIMIENTOS	II
LISTA DE ACRÓNIMOS	X
RESUMEN	XII
I PLANTEAMIENTO DEL PROYECTO	1
1 INTRODUCCIÓN	2
1.1 Presentación del problema	3
1.2 Hipótesis	3
1.3 Objetivo general	3
1.3.1 Objetivos específicos	3
1.4 Alcances	3
1.5 Justificación	4
2 METODOLOGÍA DE DISEÑO	5
II INVESTIGACIÓN PRELIMINAR	7
3 EL ESPACIO Y EL DESARROLLO TECNOLÓGICO	8
3.1 La era espacial y los satélites artificiales	9
3.2 Los satélites pequeños	11
3.2.1 El estándar <i>CubeSat</i>	13
3.3 Calificación de los componentes	14
3.3.1 Nivel de madurez tecnológica (TRL)	17
4 EL MEDIO AMBIENTE DE RADIACIÓN ESPACIAL Y SUS EFECTOS EN LOS COMPONENTES ELECTRÓNICOS	19
4.1 Fuentes de la radiación	20
4.1.1 Ambiente de radiación atrapada	20
4.1.2 Ambiente de radiación en tránsito	21
4.2 Efectos de la radiación en los componentes electrónicos	22
4.2.1 Efectos acumulativos	23
4.2.2 Efectos de evento individual (SEE)	24
5 FPGA EN EL ESPACIO	27
5.1 Antecedentes del FPGA	27
5.1.1 Tecnologías de configuración del FPGA	29
5.2 Fundamentos de la mitigación y la confiabilidad	31
5.2.1 Amenazas de la confiabilidad	32
5.2.2 Atributos de la confiabilidad	33
5.2.3 Medios de la confiabilidad	34
5.3 Estimación del nivel de fiabilidad	34
5.3.1 Modelos combinatorios	37
5.3.2 Modelos de Markov	38
5.4 Tolerancia a fallas en FPGA	39

5.4.1	Redundancia modular triple (TMR)	40
5.4.2	Reconfiguración global	43
5.4.3	Memory scrubbing	43
6	ESTADO DEL ARTE DE SCMI TOLERANTES A FALLAS	45
6.1	Crecimiento y proyección en las misiones realizadas con CubeSat	45
6.2	Tolerancia a fallas en <i>CubeSat</i>	46
6.3	Uso de FPGA en SCMI	46
6.3.1	SCMI comerciales	47
6.3.2	SCMI en desarrollo	49
6.4	Estado tecnológico actual	50
6.4.1	FPGA endurecidos a la radiación	50
6.4.2	Esquemas de mitigación y tolerancia a la radiación	50
6.4.3	Arquitecturas MPSoC en FPGA	51
6.5	Conclusiones	52
III	DISEÑO A NIVEL DE SISTEMA	53
7	DETERMINACIÓN DE LA MISIÓN	54
7.1	Estimación del medio ambiente de radiación	55
8	DESARROLLO DEL CONCEPTO DE DISEÑO	58
8.1	Antecedentes del proyecto	58
8.2	Identificación de las necesidades	59
8.3	Concepto de diseño del sistema	59
9	DISEÑO A NIVEL DE SISTEMA	62
9.1	Planteamiento de los requisitos	62
9.2	Unidad maestra	63
9.2.1	Unidad de procesamiento central (CPU)	63
9.2.2	Bloque de diagnóstico y reconfiguración	63
9.2.3	Periféricos	64
9.2.4	Controlador de memorias	64
9.3	Unidad de supervisión	65
9.3.1	Microcontrolador supervisor	65
9.3.2	Memoria de respaldo y configuración	66
9.4	Banco de memorias	66
9.5	Sistema de control y determinación de posición (ADCS)	67
9.6	Diseño a nivel de sistema	68
IV	DISEÑO DE DETALLE	69
10	DISEÑO DE DETALLE A NIVEL MODULAR	70
10.1	Especificaciones generales	70
10.2	Diseño de detalle a nivel modular	70
10.2.1	Unidad maestra	70
10.2.2	Unidad de supervisión	73
10.2.3	Banco de memorias	73
10.2.4	ADCS	74
10.3	Diseño de detalle del SCMI	74

10.4 Selección de componentes	75
11 IMPLEMENTACIÓN DE LA ARQUITECTURA	76
11.1 Equipo de trabajo	76
11.2 Implementación del SCMI en <i>Vivado Design Suite</i>	77
11.3 Resumen de la implementación de la unidad maestra en <i>Vivado Design Suite</i>	83
12 ESTIMACIÓN DEL NIVEL DE FIABILIDAD	85
12.1 Cálculo de la intensidad de fallas (λ)	88
12.2 Cálculo de la intensidad de reparación (μ)	90
V RESULTADOS Y CONCLUSIONES	92
13 RESULTADOS	93
13.1 Resultados del primer experimento	93
13.2 Variación de la intensidad de recuperación	94
13.2.1 Caso 1 de estudio: $LET_{th} = 3MeVcm^2g^{-1}$	94
13.2.2 Caso 2 de estudio: $LET_{th} = 5.21MeVcm^2g^{-1}$	95
13.2.3 Caso 3 de estudio: $LET_{th} = 9.13MeVcm^2g^{-1}$	95
13.3 Configuración a través de archivos Bitstream parciales	96
13.3.1 Caso 1 de estudio: $LET_{th} = 3MeVcm^2g^{-1}$	96
13.3.2 Caso 2 de estudio: $LET_{th} = 5.21MeVcm^2g^{-1}$	97
13.3.3 Caso 3 de estudio: $LET_{th} = 9.13MeVcm^2g^{-1}$	97
14 CONCLUSIONES	98
15 TRABAJO A FUTURO	99
REFERENCIAS	100
A SCMI COMERCIALES ANALIZADOS	106

ÍNDICE DE FIGURAS

Figura 2.1	Metodología de diseño	5
Figura 2.2	Metodología de diseño detallada	6
Figura 3.1	La Tierra vista desde el espacio	8
Figura 3.2	Animación de un satélite artificial	9
Figura 3.3	Línea del tiempo de algunos sucesos históricos alcanzados durante la era espacial	10
Figura 3.4	Número de satélites lanzados durante 2013 – 2022 y su equivalente en masa total	11
Figura 3.5	Número de satélites pequeños lanzados durante 2013 – 2022, clasificados por su masa, excluyendo las constelaciones <i>Starlink</i> y <i>OneWeb</i>	12
Figura 3.6	<i>CubeSat</i> “ <i>PhoneSat 2.5</i> ”, desarrollado por el <i>Ames Research Center</i> de la NASA	13
Figura 3.7	Estructuras <i>CubeSat</i> y su clasificación en micro y nanosatélites dependiendo de su tamaño	14
Figura 3.8	Jerarquía de los sistemas espaciales	15
Figura 3.9	Incremento de la velocidad de los microprocesadores vs introducción de productos comerciales y productos endurecidos a la radiación	16
Figura 3.10	Nivel de madurez tecnológica (TRL)	17
Figura 3.11	Proceso de evaluación de madurez tecnológica (TMA)	18
Figura 4.1	Representación del campo magnético terrestre protegiendo a la Tierra de la radiación solar	19
Figura 4.2	Representación de los cinturones de Van Allen terrestres	20
Figura 4.3	Representación de un satélite artificial y la Tierra siendo atacados por partículas de alta energía provenientes del Sol	21
Figura 4.4	Representación de un flujo de partículas impactando un área unitaria de la superficie de un material	22
Figura 4.5	Comportamiento de una partícula al penetrar un material	23
Figura 4.6	Camino ionizado generado por el paso de un GCR a través del drenaje de un transistor NMOS	24
Figura 5.1	Arquitectura general de un FPGA	27
Figura 5.2	Estructura básica de un bloque lógico configurable (CLB) en un FPGA	28
Figura 5.3	FPGA con diversos módulos embebidos distribuidos en todo el chip	28
Figura 5.4	Diferentes estructuras de SoC con <i>hardcores</i> embebidos y zonas configurables de FPGA	29
Figura 5.5	Carga paralela de los datos de configuración de un FPGA como maestro	30
Figura 5.6	Niveles de abstracción en donde se pueden aplicar técnicas de mitigación	31
Figura 5.7	Categorización de una falla	32
Figura 5.8	Cadena fundamental de la confiabilidad y sus amenazas	33
Figura 5.9	Intensidad de fallas en función del tiempo (curva de la bañera)	35
Figura 5.10	Cadena de Markov de un sistema con un solo componente	38
Figura 5.11	Cadena de Markov de un sistema con un solo componente con reparación	39
Figura 5.12	Tipos de redundancia aplicada en esquemas de tolerancia a fallas	39
Figura 5.13	Concepto básico de la TMR	40
Figura 5.14	Módulo de Markov de un sistema con TMR	41
Figura 5.15	Módulo de Markov reducido de un sistema con TMR	42
Figura 5.16	Módulo de Markov reducido de un sistema con n componentes con reparación	43
Figura 5.17	Módulo de Markov de un sistema con <i>memory scrubbing</i> probabilístico empleando una memoria con SEC-DED	44
Figura 6.1	Tipos de nanosatélites lanzados por año	45

Figura 6.2	Dispositivo de procesamiento central de los SCMI comerciales, $n = 82$	47
Figura 6.3	Dispositivos reconfigurables empleados en SCMI comerciales, $n = 82$	48
Figura 6.4	Grado de los componentes y TRL de los SCMI comerciales, $n = 82$	48
Figura 7.1	Órbitas aproximadas de los nanosatélites después de su lanzamiento	54
Figura 7.2	Vista preliminar de la misión definida en SPENVIS	55
Figura 7.3	Trayectoria del satélite durante un día y la variación de su altura	56
Figura 7.4	Mapas del flujo total de las partículas cargadas atrapadas	56
Figura 7.5	Espectro promedio total de las partículas cargadas atrapadas	57
Figura 7.6	Espectro LET del flujo de partículas incidentes	57
Figura 8.1	Arquitectura del <i>LIESE-SCMI</i>	58
Figura 8.2	Concepto de diseño del SCMI con el máximo nivel de abstracción	60
Figura 8.3	Concepto de diseño del SCMI con el segundo nivel de abstracción	61
Figura 9.1	Diseño de la unidad maestra con el tercer nivel de abstracción	64
Figura 9.2	Diseño de la unidad de supervisión con el tercer nivel de abstracción	66
Figura 9.3	Diseño a nivel de sistema del SCMI con el tercer nivel de abstracción	68
Figura 10.1	Vista general de la arquitectura del <i>softcore MicroBlaze</i> de <i>Xilinx</i>	71
Figura 10.2	Fiabilidad de un sistema en el tiempo continuo de un sistema sin redundancia, con TMR y TMR con <i>memory scrubbing</i>	71
Figura 10.3	Diagrama de secuencias de las tareas de diagnóstico y reconfiguración	72
Figura 10.4	Diseño de detalle del SCMI	74
Figura 11.1	Organigrama del equipo de trabajo	76
Figura 11.2	Implementación de la unidad maestra	77
Figura 11.3	Subsistema de microcontrolador con <i>MicroBlaze</i>	77
Figura 11.4	Subsistema tolerante a fallas de <i>MicroBlaze</i> con TMR	78
Figura 11.5	Implementación de la CPU	78
Figura 11.6	Arquitectura base de microcontrolador con <i>MicroBlaze</i>	79
Figura 11.7	Arquitectura base de microcontrolador con <i>MicroBlaze</i> y el BRAM	79
Figura 11.8	Implementación del bloque de diagnóstico y reconfiguración	80
Figura 11.9	Implementación del bloque de periféricos	80
Figura 11.10	Implementación de los periféricos para el bus del satélite	81
Figura 11.11	Implementación de los periféricos para la carga útil	81
Figura 11.12	Implementación de los periféricos para el ADCS	82
Figura 11.13	Implementación del controlador de memorias	82
Figura 11.14	Tarjeta de desarrollo <i>Nexys Video</i>	83
Figura 11.15	Reporte de utilización de recursos – <i>post-implementation</i>	83
Figura 11.16	Reporte de consumo energético – <i>post-implementation</i>	83
Figura 11.17	<i>Logic Placement</i> de la unidad maestra en un Artix-7 (XC7A200T-1SBG484C)	84
Figura 12.1	Modelo de Markov para el CPU del SCMI	85
Figura 12.2	Modelo de Markov reducido para el CPU del SCMI	86
Figura 12.3	Relación de los bits de configuración del FPGA	88
Figura 12.4	Resultados del <i>script</i> realizado con <i>Python</i>	88
Figura 12.5	Espectro LET del flujo de partículas incidentes para los tres casos de estudio	89
Figura 13.1	Resultados del análisis de fiabilidad para los tres casos de estudio	93
Figura 13.2	Resultados del análisis de fiabilidad para el caso 1 de estudio, variando la intensidad de recuperación	94

Figura 13.3 Resultados del análisis de fiabilidad para el caso 2 de estudio, variando la intensidad de recuperación	95
Figura 13.4 Resultados del análisis de fiabilidad para el caso 3 de estudio, variando la intensidad de recuperación	95
Figura 13.5 Resultados del análisis de fiabilidad para el caso 1 de estudio, empleando diferentes tamaños del <i>Bitstream</i>	96
Figura 13.6 Resultados del análisis de fiabilidad para el caso 2 de estudio, empleando diferentes tamaños del <i>Bitstream</i>	97
Figura 13.7 Resultados del análisis de fiabilidad para el caso 3 de estudio, empleando diferentes tamaños del <i>Bitstream</i>	97

ÍNDICE DE TABLAS

Tabla 3.1	Clasificación de los satélites artificiales según su masa y sus costos de desarrollo	12
Tabla 3.2	Descripción del grado de los componentes EEE	15
Tabla 4.1	Efectos de evento individual (SEE)	25
Tabla 4.2	Principales SEE en función de la tecnología y familia del componente	26
Tabla 5.1	Características del FPGA en función de su tecnología	30
Tabla 6.1	FPGA y SoC presentes en SCMI en desarrollo	49
Tabla 7.1	Parámetros de la misión	55
Tabla 7.2	Segmentos de la misión	55
Tabla 8.1	Identificación de las necesidades	59
Tabla 8.2	Identificación de las necesidades por unidad del SCMI	60
Tabla 9.1	Determinación de los requisitos a nivel de sistema	62
Tabla 9.2	Determinación de los requisitos de la unidad maestra	63
Tabla 9.3	Determinación de los requisitos de la unidad de supervisión	65
Tabla 9.4	Determinación de los requisitos del banco de memorias	66
Tabla 9.5	Determinación de los requisitos del ADCS	67
Tabla 10.1	Especificaciones generales	70
Tabla 10.2	Selección de componentes	75
Tabla 12.1	Determinación del tiempo entre upsets y la intensidad de fallas	90
Tabla 13.1	Tamaño de los archivos <i>Bitstream</i> de los FPGA de la serie 7 de <i>Xilinx</i>	94
Tabla 13.2	Tamaño de los archivos <i>Bitstream</i> para reconfiguración parcial	96
Tabla A.1	Lista de SCMI comerciales	106

LISTA DE ACRÓNIMOS

ADCS	Attitude determination and control system	JTAG	Join test action group
ASIC	Application-specific integrated circuit	LC	Logic cell
BRAM	Block random-access memory	LCL	Latch-up current limiter
CAN	Controller area network	LEO	Low Earth orbit
CDS	CubeSat design specification	LET	Linear energy transfer
CLB	Configurable logic block	LETth	Threshold Linear energy transfer
COM	Communication system	LIESE	Laboratorio de Instrumentación Electrónica de Sistemas Espaciales
COTS	Commercial off-the-shelf	LMB	Local memory bus
CPLD	Complex programmable logic device	LUT	Lookup tables
CPU	Central process unit	MBU	Multiple-bit upset
DD	Displacement damage	MCU	Multiple-cell upset
DPR	Dynamic and partial reconfiguration	MDM	MicroBlaze debug module
DSP	Digital signal processor	MPSoC	Multiprocessor system on a chip
ECC	Error-correcting code	MRAM	Magneto resistive random-access memory
EDAC	Error detection and correction	NASA	National Aeronautics and Space Administration
EEE	Electrical, electronic and electromechanical	OBC	On-board computer
EPS	Electrical power system	OPAL	Orbiting Picosat Automated Launcher
ESA	European Space Agency	PLD	Programmable logic device
EUA	Estados Unidos de América	QSPI	Quad serial peripheral interface
FAI	Fédération Aéronautique Internationale	RBD	Reliability block diagram
FEC	Forward error correction	RF	Radio frequency
FF	Flip-flop	RHBD	Radiation hardening by design
FI	Facultad de Ingeniería	RHBP	Radiation hardening by process
FPGA	Field-programmable gate array	RTOS	Real-time operating system
FRAM	Ferroelectric random-access memory	SAA	South Atlantic anomaly
GCR	Galactic cosmic rays	SCMI	Sistema de comando y manejo de información
GNC	Guidance, navigation and control	SEC-DED	Single-error correction and double-error detection
GPIO	General purpose input/output	SDR	Software-defined radio
GPU	Graphics processing unit	SEB	Single-event burnout
I2C	Inter-integrated circuit	SEDR	Single-event dielectric rupture
IMU	Inertial measurement unit	SEE	Single-event effects
IOB	Input/output block		
IP	Intellectual property		
ISS	International Space Station		

SEFI	Single-event functional interrupt	TI	Texas Instruments
SEGR	Single-event gate rupture	TID	Total ionizing dose
SEL	Single-event latch-up	TMA	Technology maturity assessment
SEM	Soft error mitigation	TMR	Triple modular redundancy
SEHE	Single-event hard error	TNID	Total non-ionizing dose
SESB	Single-event snap-back	TRL	Technology readiness level
SET	Single-event transient	UART	Universal asynchronous receiver-transmitter
SEU	Single-event upset	UNAM	Universidad Nacional Autónoma de México
SPOF	Single point of failure	UNOOSA	United Nations Office for Outer Space Affairs
SoC	System on chip	URSS	Unión de Repúblicas Socialistas Soviéticas
SPE	Solar particle events	uSD	Secure digital
SPENVIS	Space Environment Information System	USB	Universal serial bus
SPI	Serial peripheral interface	VLSI	Very-large scale integration
SRAM	Static random-access memory		

RESUMEN

Se presenta el desarrollo de un sistema de comando y manejo de información (SCMI) tolerante a fallas basado en un FPGA, bajo el estándar *CubeSat*. El trabajo se desarrolla como una migración del *LIESE-SCMI* hacia una arquitectura más robusta descrita en un FPGA de la familia *Artix 7* de *Xilinx*, tomando como base de procesamiento un módulo IP *softcore MicroBlaze*, en el cual se realizó un análisis de fiabilidad que demostró la necesidad de implementar esquemas de tolerancia a fallas, los cuales consisten en la aplicación de redundancia modular triple (TMR) a nivel de procesador y métodos de *memory scrubbing* a través de un dispositivo supervisor externo con menor probabilidad de falla (MSP430FR5969), el cual reconfigura al FPGA a partir de la presencia de una falla, aumentando su nivel de fiabilidad al disminuir el tiempo de reconfiguración, dirigiéndose hacia un esquema de *scrubbing* prioritario y de reconfiguración parcial dinámica (DPR).



Parte I

PLANTEAMIENTO DEL PROYECTO

1 INTRODUCCIÓN

El espacio exterior siempre ha sido un punto de interés para el ser humano, quien ha aplicado los conocimientos y avances tecnológicos correspondientes a la época para ingeniárselas y construir sistemas que han vuelto realidad una de sus más grandes ambiciones, ir al espacio. A partir del lanzamiento del primer satélite en 1957, la humanidad enfocó su atención en desarrollar sistemas capaces de desempeñar diferentes tareas en órbita, como la exploración espacial, la prestación de servicios de radio, telecomunicaciones y navegación por satélite, además de diversas tareas para fines científicos, privados y militares, impulsando un crecimiento exponencial del impacto generado por el sector espacial en diferentes ámbitos de la sociedad, de modo que hoy en día las tareas que realizan los satélites artificiales son una parte importante en nuestra vida cotidiana.

Actualmente, un sector importante de la comunidad científica, empresas privadas, gobiernos e instituciones educativas se dedican al desarrollo de sistemas espaciales. Sin embargo, la construcción de satélites convencionales de gran tamaño requiere de enormes cantidades de recursos, infraestructura y tiempos de desarrollo, además de la alta complejidad existente en el proceso de lanzamiento. Para resolver estos problemas se comenzaron a construir sistemas para satélites más pequeños, en donde se aprovechan al máximo las dimensiones de pequeñas estructuras y, empleando dispositivos electrónicos modernos, se obtienen sistemas de un tamaño reducido, capaces de desempeñar una gran variedad de aplicaciones espaciales a un bajo costo y en cortos periodos de tiempo, en comparación con satélites de mayor tamaño. Con el pasar de los años, los satélites pequeños se popularizaron hasta la definición de diferentes estándares, clasificándose según su masa en mini, micro, nano y picosatélites, entre los cuales destaca el excelente desempeño de los nanosatélites, por lo que son el punto de interés en este trabajo.

Las ventajas que presentó el diminuto tamaño de los nanosatélites despertaron el interés de instituciones de diferente índole, aumentando su participación en el desarrollo de satélites pequeños. Sin embargo, sus dimensiones restringen la cantidad de elementos que pueden integrarse en su estructura. Por ejemplo, la superficie disponible para la colocación de celdas fotovoltaicas limita la cantidad de energía que se puede suministrar para la transmisión de datos, impactando directamente en la altura con la que el satélite puede mantener comunicación con estaciones terrenas, delimitando su presencia a órbitas bajas terrestres (LEO, por sus siglas en inglés). Debido a esta situación y de la mano con el avance tecnológico actual, los diseñadores han optado por emplear dispositivos electrónicos de muy alta escala de integración (VLSI, por sus siglas en inglés); como microcontroladores, arreglos de compuertas programables en campo (FPGA, por sus siglas en inglés) y los sistemas en un chip (SoC, por sus siglas en inglés). Sin embargo, en este punto es en donde impacta la diferencia entre un sistema convencional que desempeña sus funciones en un entorno terrestre, normalmente accesible para ser supervisado y manipulado por el ser humano; y un sistema espacial, que realiza sus tareas en un entorno hostil y prácticamente inalcanzable, como lo es el espacio, en donde la alta densidad de elementos lógicos en los dispositivos VLSI los hace vulnerables al flujo de partículas cargadas presente en el medio ambiente de radiación espacial. Para combatir esta limitante se aplica el concepto de “tolerancia a fallas”, idea que consiste en la aplicación de técnicas de redundancia como alternativa para aumentar el nivel de fiabilidad de los sistemas espaciales, priorizando el uso mayoritario de componentes comercialmente disponibles (conocidos como componentes COTS), en lugar de componentes de grado espacial de altos costos y limitada disponibilidad.

En este trabajo de tesis se presenta el diseño de un sistema de comando y manejo de información (SCMI) tolerante a fallas bajo el estándar *CubeSat*, empleando un FPGA COTS como unidad de procesamiento central, cuyas características son ideales para implementar técnicas de tolerancia a fallas que permitan aumentar el nivel de fiabilidad del sistema hasta alcanzar el adecuado para resistir el medio ambiente de radiación espacial presente en LEO; además de diseñar un SCMI escalable que cumpla con los requisitos de diseño determinados por una investigación del estado del arte para mantener una alta capacidad de integración con los sistemas utilizados actualmente para nanosatélites. El proyecto toma como antecedentes el diseño del *LIESE-SCMI* [1], [2], migrando hacia una arquitectura con mayores

capacidades de desempeño y el diseño de un esquema de tolerancia a fallas más robusto.

El reporte escrito cuenta con la estructura adecuada para dar seguimiento al proceso de diseño, presentando el planteamiento del proyecto, los resultados de la investigación preliminar y los conceptos necesarios para la comprensión de los temas correspondientes, el proceso de diseño del SCMI justificando todas las decisiones tomadas durante el proceso, el experimento propuesto para validar el desempeño del SCMI presentando los resultados y, por último, un análisis del trabajo realizado, las conclusiones y el trabajo a futuro correspondiente.

1.1 Presentación del problema

El medio ambiente de radiación espacial reduce considerablemente la vida útil de los satélites, induciendo fallas transitorias y una degradación de la funcionalidad de los dispositivos electrónicos. Una forma de resolver esto es utilizar componentes de grado espacial, pero estos elevan drásticamente los costos y presentan una disponibilidad limitada, por lo que es necesario buscar alternativas para aproximar el nivel de fiabilidad al adecuado para desempeñar misiones en LEO, minimizando el impacto en los costos necesarios para el diseño de sistemas espaciales.

1.2 Hipótesis

Las características de un FPGA de grado comercial lo colocan como una buena alternativa para diseñar un SCMI, escalable y de alto desempeño, que presente un nivel de fiabilidad adecuado para desempeñar misiones espaciales en LEO a través de la implementación de técnicas de tolerancia a fallas, manteniendo un bajo costo de desarrollo.

1.3 Objetivo general

Desarrollar un SCMI tolerante a fallas basado en un FPGA para nanosatélites bajo el estándar *CubeSat*, con el nivel de fiabilidad adecuado para desempeñar misiones en LEO.

1.3.1 Objetivos específicos

- Diseñar un SCMI que cumpla con las especificaciones establecidas en la especificación de diseño *CubeSat* (CDS, por sus siglas en inglés) [3] y con los requisitos definidos a partir de una investigación del estado del arte.
- Desarrollar un esquema de tolerancia a fallas descrito en un FPGA, que permita alcanzar el nivel de fiabilidad adecuado para desempeñar misiones en LEO.
- Aprovechar las características de reconfiguración de un FPGA para desarrollar una arquitectura adaptativa que pueda mantener su estado funcional, aún con la presencia de fallas inducidas por el medio ambiente de radiación espacial.
- Diseñar un SCMI escalable de alto desempeño para satisfacer los requisitos de integración de diversas plataformas para nanosatélites.
- Mantener un uso mayoritario de componentes COTS para minimizar los costos de desarrollo.

1.4 Alcances

Este trabajo tiene como alcances el diseño de un SCMI para nanosatélites bajo el estándar *CubeSat*, implementando una arquitectura escalable tolerante a fallas de alto desempeño descrita en un FPGA, complementada con un esquema de recuperación que restaure el estado funcional del SCMI en caso de que se presenten fallas transitorias inducidas por el medio ambiente de radiación espacial, de modo que se alcance un nivel de fiabilidad adecuado para desempeñar misiones en LEO.

1.5 Justificación

Los satélites pequeños han sido un factor importante para el crecimiento del sector espacial, ubicando el desarrollo de nanosatélites como un área de oportunidad adecuada para la participación de diferentes instituciones en proyectos espaciales, fomentando la propuesta de nuevas alternativas para el diseño de nanosatélites empleando componentes COTS de alto desempeño (como microcontroladores, FPGA o SoC) que, en conjunto con técnicas de tolerancia a fallas, pueden ser una opción viable para elevar los niveles de fiabilidad de los sistemas para nanosatélites, minimizando el impacto en los costos necesarios para el diseño y construcción de un SCMI tolerante a fallas.

El proyecto forma parte de las líneas de investigación del Laboratorio de Instrumentación Electrónica de Sistemas Espaciales (LIESE), adjunto al departamento de electrónica de la Facultad de Ingeniería (FI) de la Universidad Nacional Autónoma de México (UNAM), en donde se fomenta el desarrollo a bajo nivel de sistemas para nanosatélites bajo el estándar *CubeSat*. El LIESE cuenta con el diseño del *LIESE-SCMI* en etapa de pruebas, en donde la investigación realizada y la experiencia obtenida sirven como punto de partida para este trabajo, siguiendo las tendencias actuales en el área para migrar hacia un sistema con mayores capacidades de desempeño, escalable y, principalmente, una arquitectura tolerante a fallas más robusta, capaz de tolerar las fallas inducidas por el medio ambiente de radiación espacial presente en LEO.

2 METODOLOGÍA DE DISEÑO

La metodología de diseño es el conjunto de procedimientos y técnicas de rigor científico aplicadas sistemáticamente durante un proceso de investigación para alcanzar un resultado teóricamente válido. En otras palabras, es el soporte conceptual que rige la manera en que se aplican los procedimientos de un proyecto. Definir una buena metodología es fundamental para desarrollar un proyecto cumpliendo con cada uno de los aspectos necesarios para obtener los resultados deseados. En este capítulo se presenta con detalle la metodología de diseño aplicada en el desarrollo de un SCMI tolerante a fallas basado en un FPGA para nanosatélites bajo el estándar *CubeSat*.



La metodología utilizada en este trabajo consiste en un proceso iterativo *Top-Down*¹ de ingeniería desarrollado con base en los objetivos y alcances del proyecto, adaptando el proceso de ingeniería para misiones espaciales de [4] y el proceso para el desarrollo de productos de [5], dividida en cinco etapas, como se presenta en la figura 2.1.

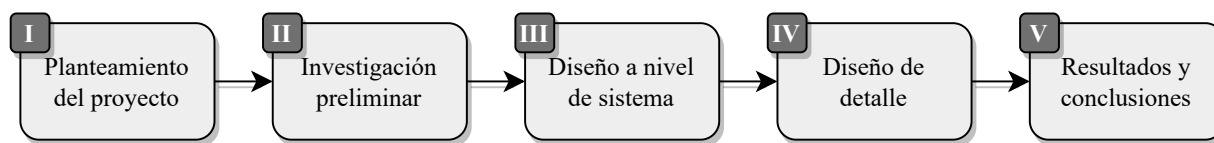


Figura 2.1: Metodología de diseño

I – Planteamiento del proyecto

La etapa **I** consiste en la fase inicial del trabajo, en donde se establecen las bases del proyecto al identificar la problemática principal del proceso de diseño (sección 1.1), se realiza el planteamiento de una hipótesis que puede darle solución al problema (sección 1.2), se determina el objetivo general y los objetivos específicos (sección 1.3), se delimitan los alcances de la tesis (sección 1.4), se justifican las áreas de oportunidad localizadas para el trabajo (sección 1.5), y se determina la metodología de diseño con la que se llevará a cabo el proceso, además de una estimación del tiempo en el que deberá ser desarrollado el trabajo (capítulo 2).

II – Investigación preliminar

Durante la etapa **II** se debe de realizar una investigación de los conceptos técnicos necesarios para el desarrollo del proyecto (capítulos 3 – 5), además de un estudio del estado del arte de los SCMI bajo el estándar *CubeSat* que sirva como punto de partida para determinar los requisitos de diseño del SCMI (capítulo 6).

III – Diseño a nivel de sistema

La etapa **III** inicia con la determinación de la misión (capítulo 7), seguida del inicio de la metodología de diseño *Top-Down*, partiendo desde una perspectiva del sistema empleando el máximo nivel de abstracción y estableciendo las necesidades que debe de satisfacer el SCMI (capítulo 8). Posteriormente, se desarrolla el diseño a nivel de sistema del SCMI siguiendo un proceso iterativo para eliminar gradualmente el nivel de abstracción del sistema hasta obtener el diseño conceptual a nivel modular (capítulo 9).

¹Proceso de diseño que inicia con un alto nivel de abstracción, cuyo nivel de detalle incrementa conforme se avanza en el desarrollo.

IV – Diseño de detalle

La etapa **IV** comienza con la determinación de las especificaciones de diseño, determinadas con base en un análisis profundo de los parámetros que debe de satisfacer el SCMI y la aplicación de los conocimientos en el área para construir el sistema a nivel modular (capítulo 10), para posteriormente implementar la arquitectura propuesta (capítulo 11) y poder estimar el nivel de fiabilidad (capítulo 12).

V – Resultados y conclusiones

La etapa **V** corresponde a la última fase del trabajo, en donde se analizan los resultados de la propuesta presentada en el diseño de detalle (capítulo 13), con los cuales se desarrollan las conclusiones correspondientes del trabajo realizado (capítulo 14) y, finalmente, se describe el trabajo pendiente para tareas futuras (capítulo 15), terminando con la documentación del proyecto.

En la figura 2.2 se presenta la metodología de diseño completa, detallando las tareas correspondientes para cada una de las etapas del proyecto.

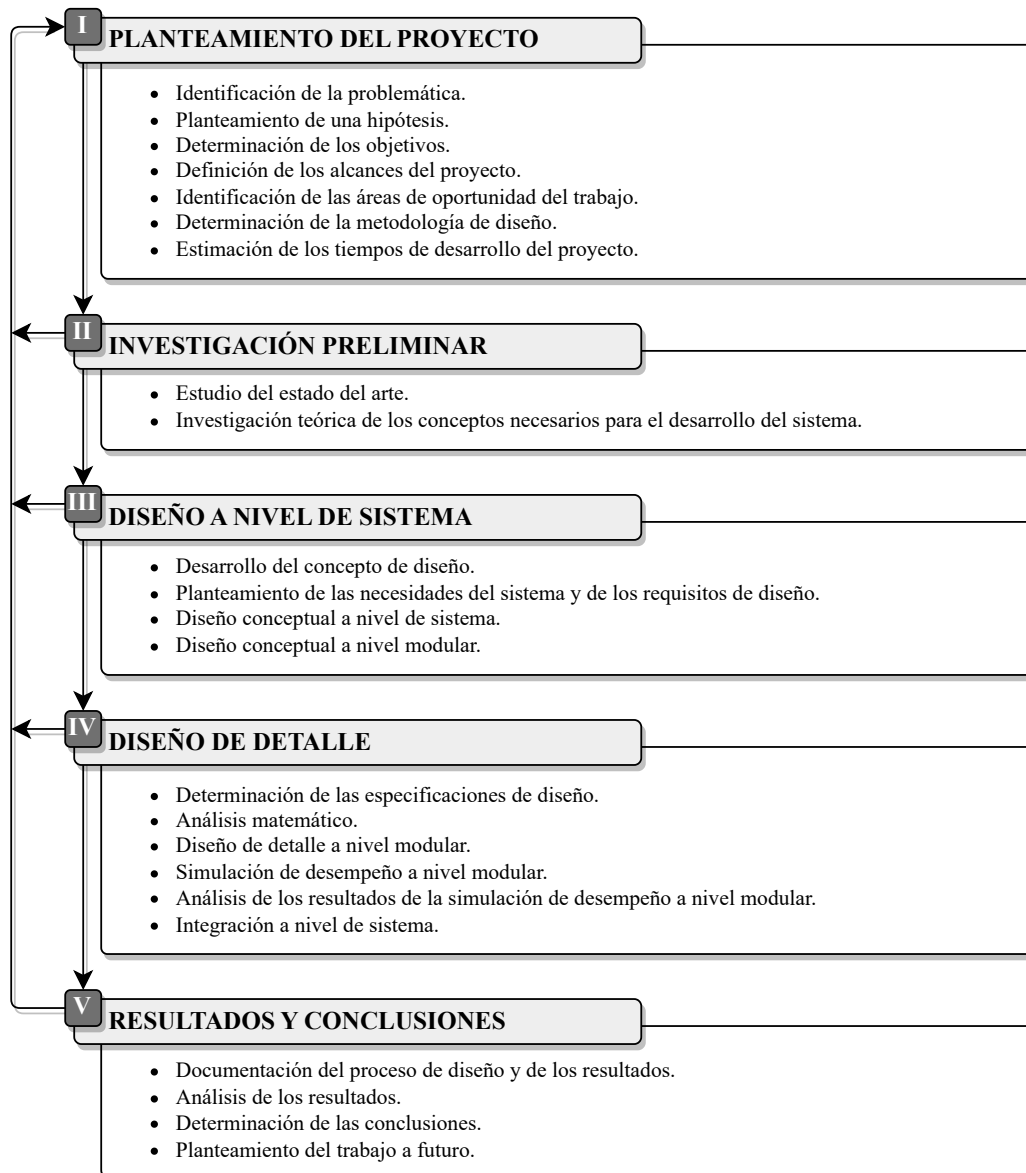


Figura 2.2: Metodología de diseño detallada

Parte II

INVESTIGACIÓN PRELIMINAR

3 EL ESPACIO Y EL DESARROLLO TECNOLÓGICO

No es un secreto el alto interés del ser humano por explorar lo desconocido. Desde los inicios de la astronomía en las civilizaciones antiguas, el invento del telescopio astronómico de Galileo Galilei en 1609, la llegada del ser humano a la Luna en 1969 y hasta el aterrizaje de *Perseverance* en Marte en 2021, la humanidad ha utilizado grandes cantidades de recursos en descubrir lo que se encuentra más allá de los límites de su hogar: la Tierra, y todas las posibilidades que eso representa. En este capítulo se presentan los conceptos básicos que servirán como introducción del trabajo de tesis, comenzando con una breve línea del tiempo de los antecedentes históricos de la era espacial; los fundamentos teóricos de los satélites artificiales y sus aplicaciones en el espacio, así como los factores que propiciaron el éxito de los satélites pequeños; y el desarrollo tecnológico, identificando las características de los dispositivos y sistemas electrónicos para su uso en aplicaciones espaciales.



Se le denomina espacio exterior (o simplemente espacio) a la región externa de la atmósfera de cualquier cuerpo celeste², caracterizada por tener una baja densidad de partículas. La atmósfera terrestre disminuye gradualmente en función de la altura, lo que significa que no tiene un fin abrupto. Sin embargo, la Federación Aeronáutica Internacional (FAI) considera el límite entre la atmósfera terrestre y el espacio en la línea de Kármán³, estimada en 100 km sobre el nivel del mar.

Históricamente, el estudio del espacio ha propiciado el desarrollo de diferentes áreas, tales como el conocimiento del planeta y sus alrededores para fines científicos, aplicaciones militares y la prestación de servicios para la sociedad, las cuales tuvieron un alto crecimiento durante la era espacial. Actualmente, instituciones privadas, académicas y gubernamentales continúan proponiendo soluciones y nuevas tecnologías para mejorar el desempeño de los sistemas espaciales.



Figura 3.1: La Tierra vista desde el espacio [6]

²Entidad u objeto físico presente en el universo observable.

³Altura en donde la densidad de la atmósfera terrestre es tan baja que la velocidad de una aeronave debe ser equiparable a la velocidad orbital para mantener una suficiente sustentación aerodinámica para sostenerse a sí misma.

3.1 La era espacial y los satélites artificiales

Se le conoce como satélite artificial (véase la figura 3.2) a cualquier objeto construido por seres humanos que orbita un cuerpo celeste con un objetivo específico, denominado misión. Estos artefactos son construidos y probados en la Tierra para posteriormente lanzarse al espacio con un misil o nave espacial, que sirven como medio de transporte para colocar al satélite en órbita y que este cumpla su misión. Durante su operación, el satélite rodeará la Tierra estableciendo comunicación cada vez que encuentre línea de visibilidad con una estación terrena, hasta el final de su vida útil en donde el satélite, en función de la altura de su órbita, terminará orbitando el planeta indefinidamente o reingresará a la Tierra para, en algunos casos, desintegrarse en la atmósfera terrestre.



Figura 3.2: Animación de un satélite artificial (*CINDI-CNOFS Satellite*) [6]

Los satélites artificiales han sido fundamentales para el progreso de la humanidad, impulsando el crecimiento de diferentes sectores. El primer satélite artificial de la historia, el *Sputnik I*, fue lanzado por la Unión de Repúblicas Socialistas Soviéticas (URSS) el 4 de octubre de 1957, dando lugar al inicio de la era espacial. A partir de este momento y como parte de la guerra fría (1947 – 1989), inició una competencia entre dos potencias: la URSS y los Estados Unidos de América (EUA), en donde ambas naciones buscaron la superioridad en el área espacial [7]. El enfrentamiento condujo a un desarrollo acelerado de ambos países, alcanzando una serie de sucesos históricos en el ámbito espacial (algunos de estos sucesos se presentan en la línea del tiempo de la figura 3.3). El *Sputnik I* tuvo una masa de 83.6 kg y se posicionó en una órbita elíptica con un apogeo⁴ de 938 km y un perigeo⁵ de 214 km de altura, enviando telemetría a tierra durante los siguientes 22 días hasta desintegrarse al reingresar a la atmósfera terrestre el 4 de enero de 1958 [8]. Desde entonces, diversas naciones identificaron un área de oportunidad en los satélites artificiales, empleando nuevas tecnologías para desarrollar diferentes tipos de misiones, dando lugar a un crecimiento exponencial en el sector espacial e incentivando la recepción de nuevas ideas. Actualmente, el sector espacial se ha fortalecido lo suficiente para la creación de programas espaciales en todo el mundo, colocando a los satélites artificiales como un factor común en la vida cotidiana a través de una gran variedad de misiones, como la prestación de servicios, misiones privadas, aplicaciones militares y experimentos científicos y tecnológicos.

⁴Punto más lejano de una órbita elíptica de un satélite artificial con respecto al centro de la Tierra.

⁵Punto más cercano de una órbita elíptica de un satélite artificial con respecto al centro de la Tierra.

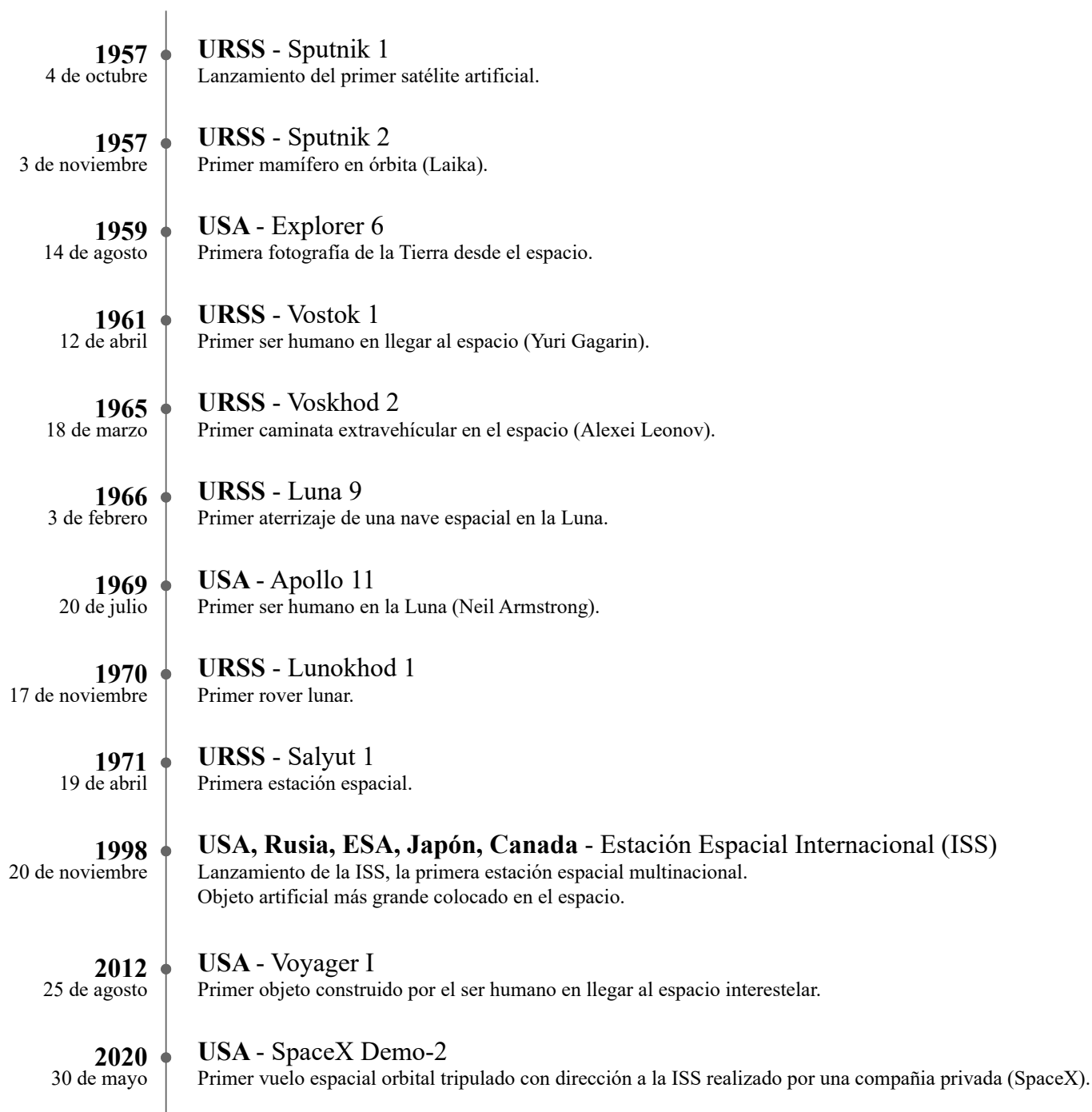


Figura 3.3: Línea del tiempo de algunos sucesos históricos alcanzados durante la era espacial [9]

Desde el lanzamiento del *Sputnik I* hasta julio de 2023, la Oficina de las Naciones Unidas para Asuntos del Espacio Ultraterrestre (UNOOSA, por sus siglas en inglés) ha registrado un total de 14 344 lanzamientos de objetos al espacio, de los cuales 8704 se encuentran operando en órbita [10], estimando hasta la fecha un registro aproximado del 87 % de todos los satélites, sondas, módulos de aterrizaje, naves espaciales tripuladas y elementos de vuelo de estaciones espaciales lanzados a la órbita terrestre o más allá [11], demostrando el enorme crecimiento del sector espacial.

3.2 Los satélites pequeños

Con el paso de los años y a medida que progresó el sector espacial, las misiones aumentaron en complejidad, demandando la construcción de satélites más grandes sustentados por programas espaciales de alto presupuesto. En busca de una alternativa para incrementar la producción de satélites artificiales y siguiendo la tendencia tecnológica moderna, nace la idea de desarrollar satélites más pequeños. Actualmente, la industria utiliza procesos de integración a muy gran escala (VLSI, por sus siglas en inglés)⁶ para construir circuitos más complejos en chips individuales, elevando las capacidades de producción en masa a precios más accesibles. Como en muchas áreas, el sector espacial aprovecha las características de los dispositivos modernos para construir satélites artificiales de alto desempeño en espacios reducidos, disminuyendo la masa, el volumen, los requisitos de potencia y la complejidad del sistema, además de minimizar los recursos y tiempos de desarrollo, factores clave para la participación de instituciones privadas, académicas y gubernamentales de países en vías de desarrollo en proyectos espaciales.

Las ventajas que presentaron los satélites pequeños fueron notables para incrementar su popularidad y frecuencia de lanzamiento, hasta el punto de colocarse dentro de su propia clasificación que, dependiendo de su masa, se dividen en mini, micro, nano, pico y femtosatélite. En la figura 3.4 se presenta el número de satélites lanzados desde el 2013 hasta el 2022, denotando el actual dominio de los satélites pequeños frente a los satélites “convencionales”. Por otro lado, en la figura 3.5 se presenta el número de satélites pequeños lanzados durante el mismo periodo (línea amarilla), agrupados dependiendo de su masa, excluyendo las constelaciones *Starlink*⁷ y *OneWeb*⁸, colocando a los nanosatélites como los satélites pequeños lanzados con mayor frecuencia, por lo que son el área de interés en este trabajo.

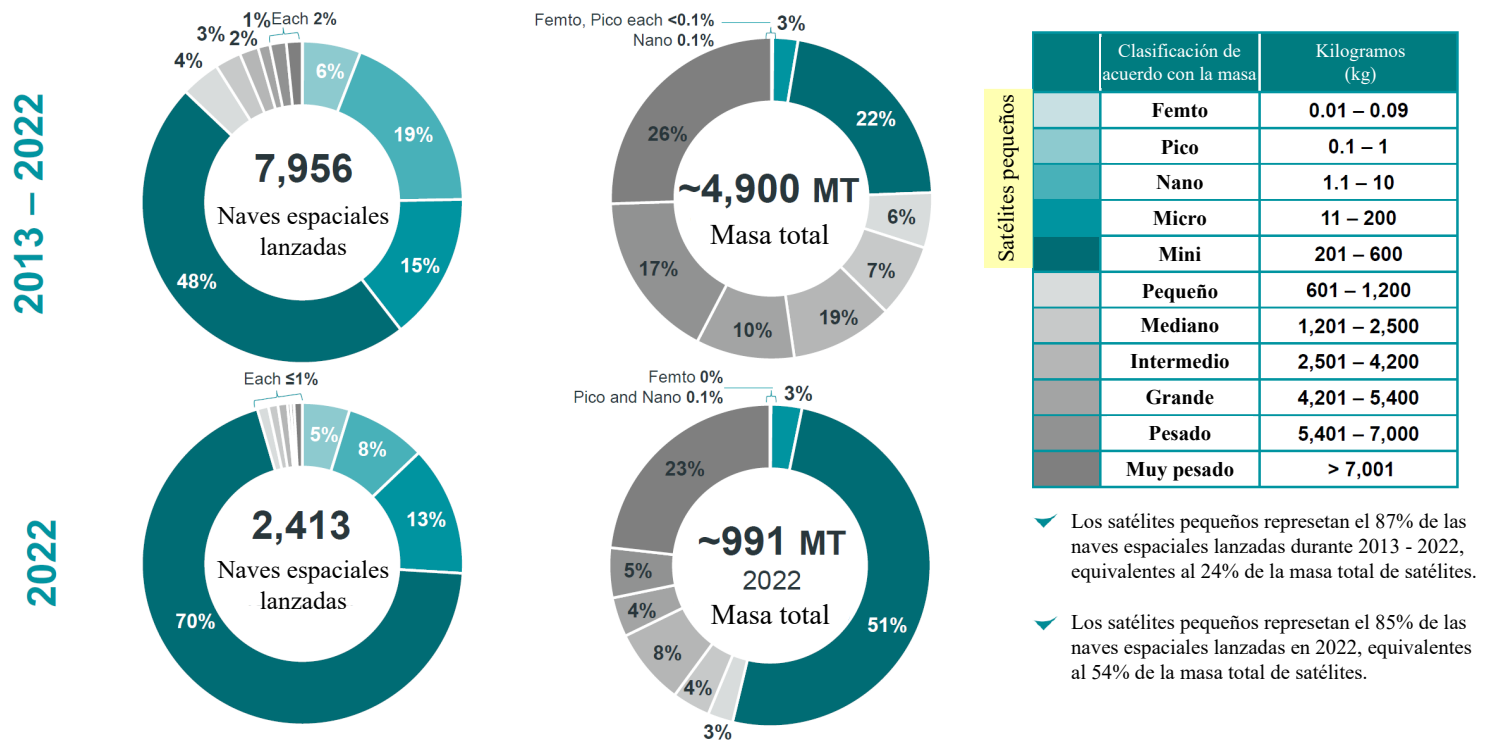


Figura 3.4: Número de satélites lanzados durante 2013 – 2022 y su equivalente en masa total [12]

⁶Proceso de creación de circuitos integrados compuestos por cientos de miles de transistores en un único chip.

⁷Constelación de minisatélites (de aproximadamente 260 kg) que, hasta 2022, contaba con 3570 satélites de comunicaciones en LEO [12].

⁸Constelación de microsátélites (de aproximadamente 150 kg) que, hasta 2022, contaba con 2022 satélites de comunicaciones en LEO [12].

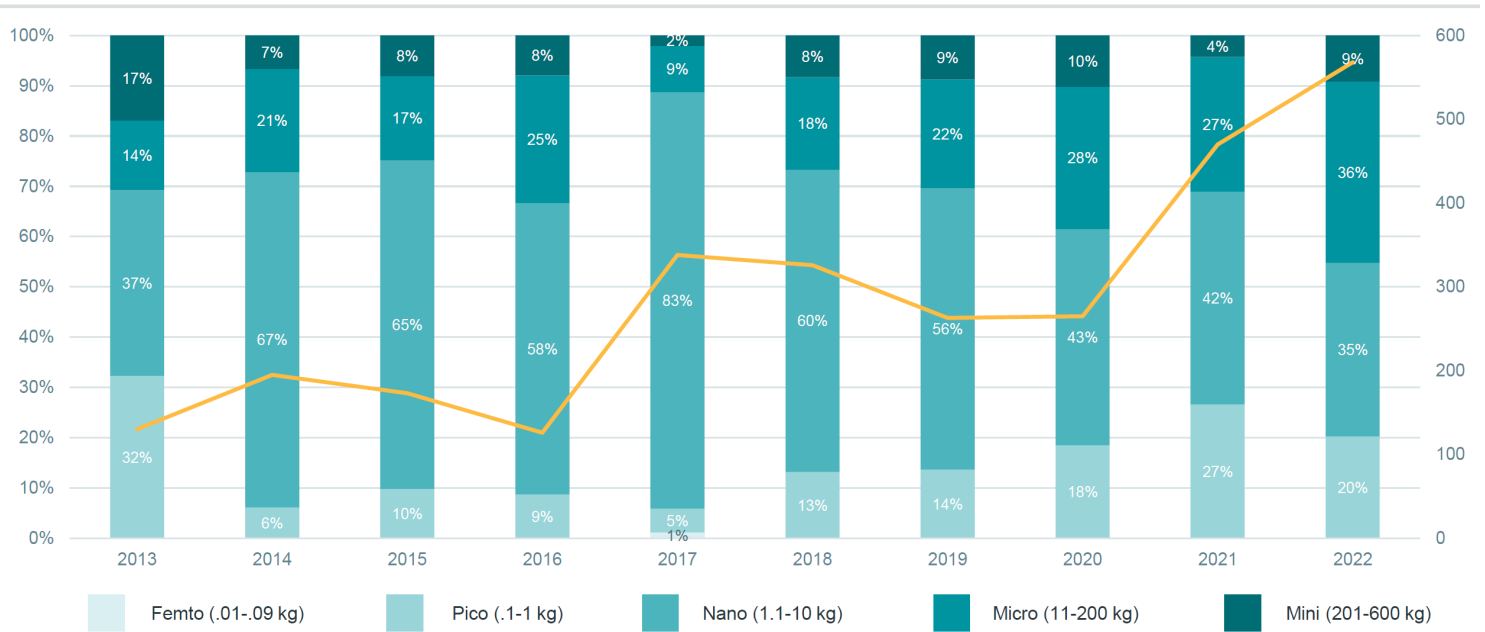


Figura 3.5: Número de satélites pequeños lanzados durante 2013 – 2022, clasificados por su masa, excluyendo las constelaciones *Starlink* y *OneWeb* [12]

El diseño de satélites artificiales presenta un reto de ingeniería, requiriendo la participación de un gran personal capacitado, además del equipo, material e infraestructura con suficiente disponibilidad para operar durante el largo periodo de tiempo que requiere diseñar, construir y probar el sistema. Por otro lado, se debe de considerar el costo de los componentes y el lanzamiento del satélite, dando como resultado el gasto de grandes cantidades de recursos para llevar a cabo un proyecto espacial. A diferencia de lo anterior, el proceso de diseño de los satélites pequeños presenta cambios que impactan directamente en los gastos, siendo este uno de los factores más importantes para el aumento en la frecuencia de lanzamiento. En la tabla 3.1 se presenta la diferencia de los costos necesarios para desarrollar satélites artificiales en función de su masa, demostrando el por qué los satélites pequeños son desarrollados comúnmente por estudiantes o equipos aficionados [4].

Tabla 3.1: Clasificación de los satélites artificiales según su masa y sus costos de desarrollo [13]

Clase	Masa (kg)	Costo (MUSD)	
Satélites grandes	>1000	>134.27	
Satélites medianos	500 – 1000	33.57 – 134.27	
Satélites pequeños	Minisatélite	100 – 500	9.40 – 33.57
	Microsatélite	10 – 100	1.34 – 9.40
	Nanosatélite	1 – 10	0.13 – 1.34
	Picosatélite	<1	<0.13

El creciente interés en los satélites pequeños impulsó la recepción de nuevas propuestas para facilitar el proceso de lanzamiento, buscando nuevas alternativas compatibles con los lanzadores que se utilizaban para colocar en órbita al satélite. Con esta filosofía, se llegó a la definición del *CubeSat*, idea que comenzó principalmente como una herramienta de enseñanza y demostraciones tecnológicas para proveer a los estudiantes un sencillo acceso al espacio, pero recientes proyectos de vuelo seleccionados demostraron que la tecnología ha madurado lo suficiente para que el *CubeSat* también pueda abordar importantes objetivos científicos [14].

3.2.1 El estándar *CubeSat*

El *CubeSat* es un estándar de diseño para nano y microsátélites propuesto en 1999 por los profesores Jordi Puig-Suari, de la Universidad Politécnica Estatal de California (CAL Poly); y Robert Twiggs, de Standford. La idea del *CubeSat* nació con un proyecto de Standford, con el objetivo de colocar en órbita a múltiples picosátélites “*PicoSat*” como parte de un programa de investigación avanzada de defensa, dando como resultado el diseño del microsátélite “Lanzador automático de picosátélites en órbita” (OPAL, por sus siglas en inglés), dentro del cual los estudiantes diseñaron un lanzador rectangular que podía sujetar los *PicoSat* y expulsarlos en el momento deseado [15]. El éxito de la misión demostró la viabilidad del concepto del picosatélite y despertó el interés en el desarrollo de un lanzador que permitiera colocar en órbita a múltiples satélites pequeños contenidos dentro de un lanzador con forma de tubo, simplificando tanto la integración mecánica con el vehículo de lanzamiento como la expulsión del satélite [4]. Después del lanzamiento del OPAL, comenzó un trabajo conjunto con el Cal Poly para diseñar un satélite tan pequeño como el *PicoSat*, teniendo en mente la definición de un estándar de satélites del tamaño de una barra de helado y de un lanzador que pudiera lanzar varios de ellos a la vez [15]. Como resultado, se definió el *CubeSat* como una estructura en forma de cubo con aristas de 10 cm, un volumen de un litro y una masa máxima de 2 kg por unidad [3] (véase la figura 3.6), que puede escalarse apilando múltiples unidades para formar satélites más grandes cuyos nombres están dados por el número de unidades que lo conforman (véase la figura 3.7).

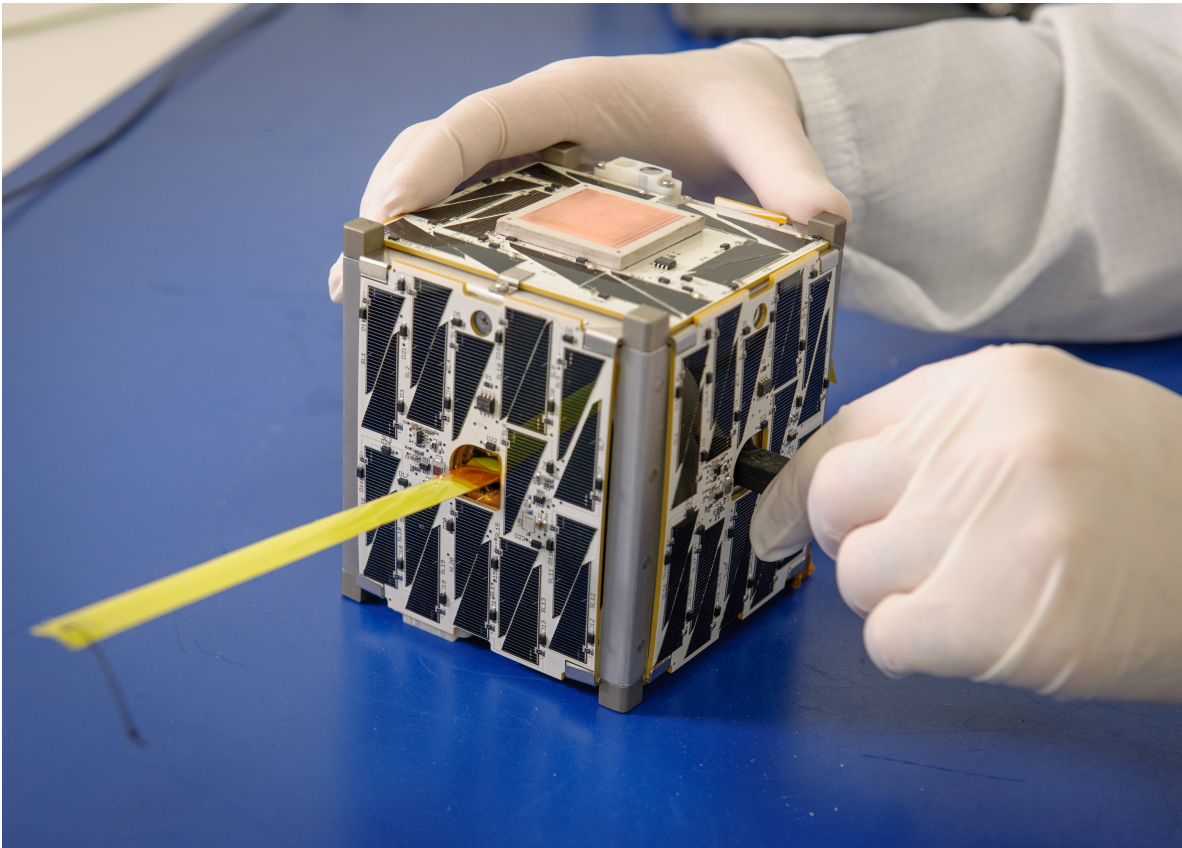


Figura 3.6: *CubeSat* “*PhoneSat 2.5*”, desarrollado por el *Ames Research Center* de la NASA [6]

El éxito del *CubeSat* se debe a varios factores: su forma cúbica, que permite distribuir eficientemente los subsistemas en su interior y colocar paneles solares en sus caras externas; su diminuto tamaño, el cual hizo que los lanzamientos fueran asequibles; y su tipo de contención, permitiendo colocarse dentro de un lanzador cuya estructura disminuyera la posibilidad de dañar el vehículo de lanzamiento [15]. A pesar de comenzar como una idea para introducir a los estudiantes al desarrollo de satélites, el *CubeSat* se ha convertido en un estándar utilizado en todo el mundo para diferentes aplicaciones, incluyendo misiones científicas y militares.

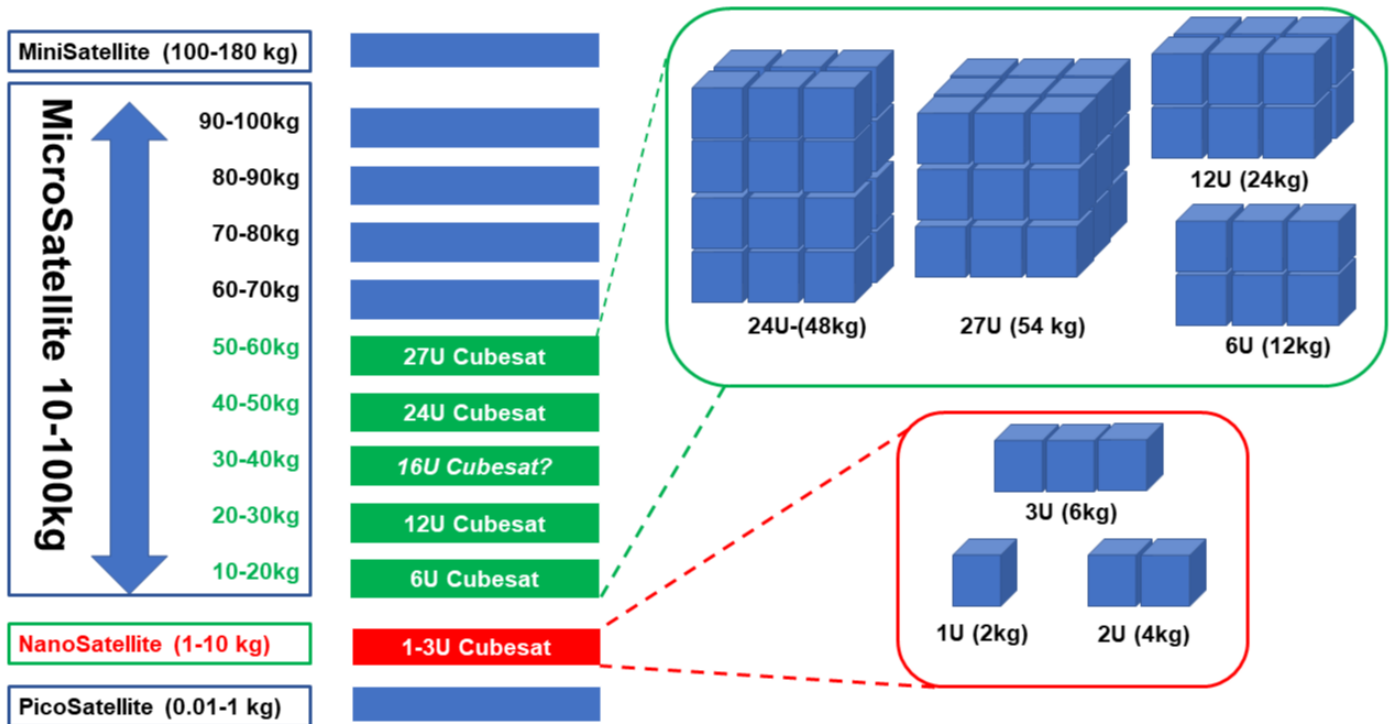


Figura 3.7: Estructuras *CubeSat* y su clasificación en micro y nanosatélites dependiendo de su tamaño [16]

3.3 Calificación de los componentes

Mucho se ha hablado de las ventajas de los satélites pequeños y su impacto en el crecimiento del sector espacial, pero en términos de desempeño ¿cómo puede algo tan pequeño sustituir a sistemas de gran tamaño? Claramente, no se puede comparar la capacidad de un sistema con otro 10, 100 o hasta 1000 veces más grande, por lo que para responder la pregunta es necesario analizar el desarrollo tecnológico de las últimas décadas. Como primer punto ¿de qué depende el buen funcionamiento de un sistema? Siguiendo la jerarquía para sistemas espaciales de la figura 3.8, para que un sistema funcione correctamente es necesario que los elementos que lo conforman también funcionen correctamente. Hablando de los satélites artificiales, en el nivel más bajo se encuentran los componentes eléctricos, electrónicos y electromecánicos (EEE, por sus siglas en inglés) que, según la Agencia Espacial Europea (ESA, por sus siglas en inglés), “su calidad es un factor determinante para su funcionamiento y rendimiento fiables durante la vida útil de la misión, por lo que es esencial garantizar que todos los componentes sean lo suficientemente robustos para soportar el estrés ambiental que enfrenten en su aplicación”.

A finales de la década de los 50, los defectos de diseño y fabricación en los componentes EEE fueron causa de sospecha en la generación de fallas prematuras en los sistemas espaciales. Para entonces, los componentes eran típicamente de grado comercial y calidad militar estándar [18]. A partir de 1962, la Administración Nacional de Aeronáutica y el Espacio (NASA, por sus siglas en inglés) comenzó a desarrollar los primeros documentos para establecer los requisitos que debían de cumplir los microcircuitos para su uso en el espacio. Con el tiempo, esto dio lugar a la estandarización de los procesos de fabricación y calidad para garantizar un rendimiento mínimo adecuado de los componentes EEE, estableciendo en 1972 la clase S para los componentes de grado espacial.

La calificación de los componentes consiste en la aplicación de pruebas de inspección mecánica, eléctrica y de entorno previstas para verificar que los materiales, el diseño, el rendimiento y la fiabilidad a largo plazo de los componentes EEE cumplan con la especificación y la aplicación prevista, además de asegurar que los procesos del fabricante sean consistentes de un lote a otro [19]. Actualmente, se establecen cuatro niveles de calidad (véase la tabla 3.2) para componentes EEE.

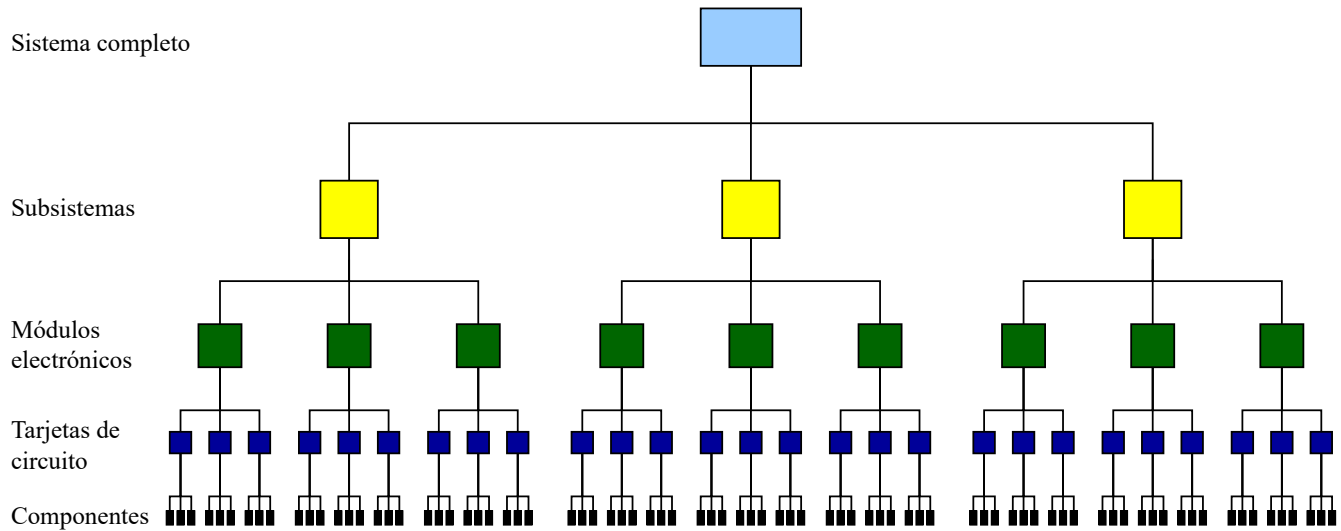


Figura 3.8: Jerarquía de los sistemas espaciales [17]

Tabla 3.2: Descripción del grado de los componentes EEE [19], [20]

Grado	Descripción	Fiabilidad	Costo	Uso típico
1	Componentes calificados con calidad de clase “espacial” o equivalente.	La más alta	Muy alto	Vuelos espaciales.
2	Componentes calificados con calidad de clase “completamente militar” o equivalente.	Muy alta	Alto	Vuelos espaciales o equipo de soporte crítico terrestre.
3	Componentes con calidad de clase “baja militar”, de un proveedor de alta fiabilidad o equivalente. Componentes EEE de grado automotriz.	Media	Moderado	Experimentos en vuelos espaciales, vuelos espaciales no críticos en <i>CubeSat</i> , equipo de soporte crítico terrestre, pruebas para demostraciones y sistemas de soporte terrestres.
4	Componentes con calidad de clase “comercial”, con datos de calificación a discreción del fabricante, sin procesos de monitoreo gubernamentales incorporados durante la fabricación.	Variable	El más bajo	Vuelos espaciales no críticos en <i>CubeSat</i> , equipo no crítico de soporte terrestre, sistemas de soporte terrestre, pruebas para demostraciones y prototipos. Equipo de soporte terrestre crítico limitado.

Para construir un satélite la mejor decisión, en términos de fiabilidad, será emplear componentes de grado 1. Sin embargo, como parte del proceso de ingeniería es importante analizar todas las consecuencias de esta decisión. Para la manufactura de componentes de grado 1, los fabricantes utilizan procesos altamente costosos de endurecimiento a la radiación por diseño⁹ (RHBD, por sus siglas en inglés). Estos componentes son más caros que sus contrapartes comerciales y, debido al esfuerzo y a los costos adicionales asociados con su manufactura, presentan un retraso de varias generaciones (de 5 a 7 años) en términos de la velocidad de procesamiento, potencia y tamaño [21], como se muestra en la figura 3.9. Por otro lado, la disponibilidad del mercado para los componentes de grado espacial es sumamente limitada en comparación con los componentes COTS¹⁰, aumentando aun más los recursos necesarios para

⁹Proceso de manufactura de componentes EEE en donde se aplican técnicas para mitigar el daño, las perturbaciones y la pérdida de datos causados por la radiación espacial.

¹⁰Componentes fácilmente localizables en el mercado, adquiribles al menor precio debido a que únicamente cumplen con las especifica-

el desarrollo de satélites empleando componentes de grado 1.

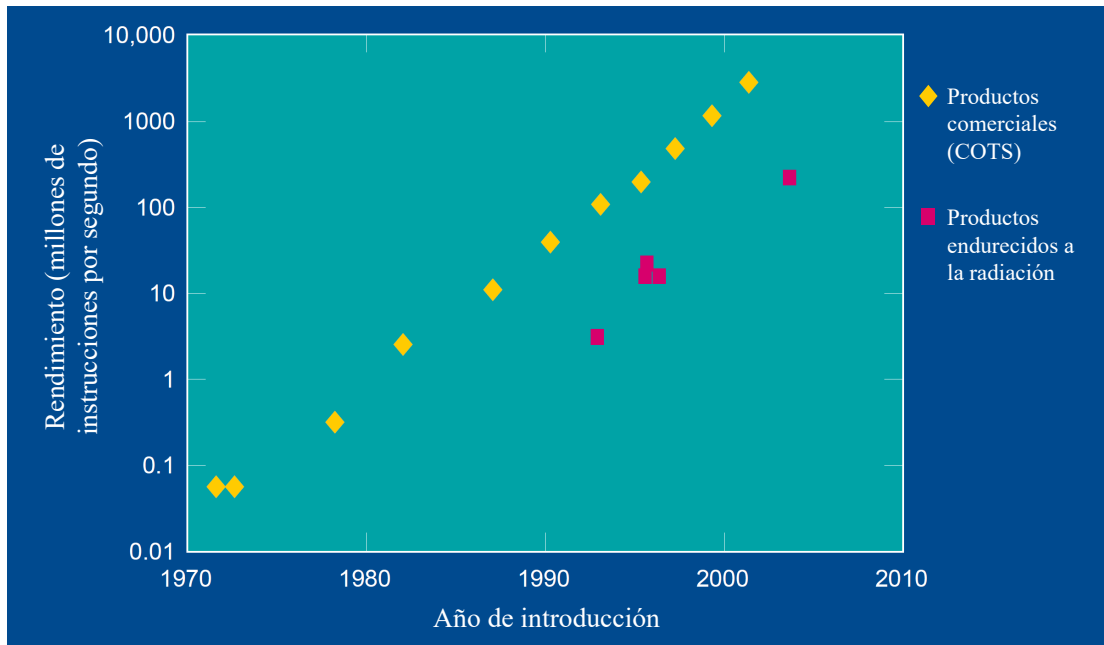


Figura 3.9: Incremento de la velocidad de los microprocesadores vs introducción de productos comerciales y productos endurecidos a la radiación [21]

En conclusión, la selección de componentes para el diseño de satélites pequeños es de vital importancia para aumentar la probabilidad de éxito de la misión. El reto de ingeniería se presenta al aumentar el nivel de fiabilidad del sistema, minimizando el impacto en los costos necesarios para el desarrollo del proyecto. En este punto adquiere relevancia el desarrollo tecnológico, el cual depende en gran medida de los cambios en los semiconductores modernos, en donde se encuentran principalmente: [22]

- El nivel de escala de la tecnología.
 - El incremento de la densidad de elementos lógicos por unidad de área.
 - Los cambios en el suministro de energía y los voltajes lógicos ($< 1V$).
 - La reducción de los márgenes eléctricos dentro de un dispositivo.
 - El incremento en la complejidad del dispositivo.
 - El aumento de la frecuencia de operación ($> GHz$).
- Los cambios en los materiales y estructuras internas.
- El incremento de las entradas/salidas de los encapsulados.

Con estas características, las capacidades actuales de los componentes EEE permiten construir sistemas de grandes capacidades en espacios reducidos, como lo son los satélites pequeños. Además de que, por su naturaleza, los satélites pequeños se usan comúnmente para misiones no críticas y de corta duración en LEO, lo que significa que el satélite estará expuesto a un medio ambiente de radiación “tranquilo”, así que la selección de componentes no está restringida por la necesidad de componentes endurecidos a la radiación. Los componentes modernos típicamente consumen menos potencia, tienen menor masa y volumen, además de que es sencillo trabajar con ellos, manteniendo una baja complejidad que, últimamente, es la enemiga de la fiabilidad. Con estos factores, a pesar de que los satélites pequeños son desarrollados normalmente por estudiantes o equipos aficionados, sus niveles de fiabilidad son comparables con ciones básicas del componente. [19]

sistemas espaciales más grandes gracias a su simplicidad [4]. Sin embargo, el uso de componentes COTS genera otras complicaciones que deberán ser consideradas durante el proceso de diseño para elevar los niveles de fiabilidad. En cuanto al proceso de selección de los componentes, es importante resaltar que un proceso bien dirigido de investigación del mercado puede ayudar a localizar componentes COTS con “herencia de vuelo”, calificación que se les otorga a aquellos componentes que son exactamente iguales a uno que ha sido lanzado y utilizado satisfactoriamente en el espacio con anterioridad, o que se ha sometido a las pruebas ambientales ¹¹ necesarias para demostrar tres aspectos importantes: [23]

- Supervivencia al entorno de lanzamiento y despliegue en el espacio sin pérdida de rendimiento.
- Compatibilidad del material en el espacio sin sufrir degradaciones que puedan ocasionar problemas en los otros subsistemas.
- Alcanzar un rendimiento completo durante el transcurso de la misión.

Sin embargo, hay que considerar que cualquier cambio en el proceso de manufactura o diseño del producto, en su configuración, en sus materiales o en el entorno de lanzamiento, por mínimo que sea, exime al componente de ser calificado con herencia de vuelo.

3.3.1 Nivel de madurez tecnológica (TRL)

Escalando en el nivel de jerarquía de la figura 3.8, es necesario garantizar que todos los sistemas desarrollen sus funciones correctamente. Esto se logra exponiendo previamente a cada uno de los sistemas a una serie de pruebas programadas para validar su diseño y funcionamiento en un entorno simulado o bajo ciertas condiciones, dependiendo del tipo de aplicación. Para los sistemas espaciales, la validación del nivel de desarrollo de un sistema se evalúa a través del nivel de madurez tecnológica (TRL, por sus siglas en inglés). Esta consiste en 9 niveles, cada uno alcanzado al someter al sistema a una serie de pruebas para que vaya progresando en la escala, en donde el TRL 1 es el nivel más bajo y el TRL 9 el más alto, como se observa en la figura 3.10.

Es importante mencionar que la NASA utiliza la escala TRL para evaluar la tecnología de los satélites pequeños, considerando dentro del estado del arte a los sistemas con al menos un TRL 5, indicando que el componente se construyó y operó con elementos de soporte realistas para su validación en un entorno relevante, demostrando su rendimiento general en áreas críticas bajo los criterios de pruebas de desempeño documentadas fundamentadas en predicciones analíticas. La tecnología con TRL igual o menor a 4 es considerada como “en el horizonte”. [16]

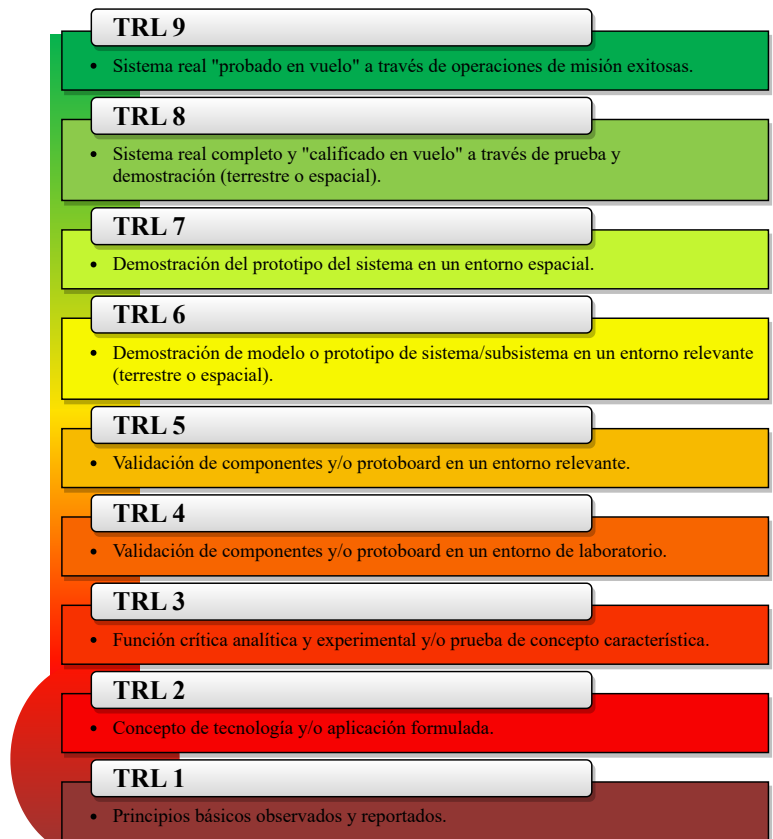


Figura 3.10: Nivel de madurez tecnológica (TRL) [24]

¹¹pruebas

Tomando como base la escala TRL ¿existe una forma de que el diseño de un sistema sea de un nivel mayor desde el inicio del proceso de diseño? En la figura 3.11 se presenta el proceso de evaluación de madurez tecnológica (TMA, por sus siglas en inglés), en donde se observa que si el sistema cuenta con una unidad idéntica en una configuración o arquitectura diferente que ya ha sido probada anteriormente, este automáticamente será catalogado con un TRL 5. En conclusión, un correcto proceso de selección de componentes identificando componentes con herencia de vuelo es un factor relevante para elevar el nivel de fiabilidad del sistema, aun sin evaluar su desempeño previamente.

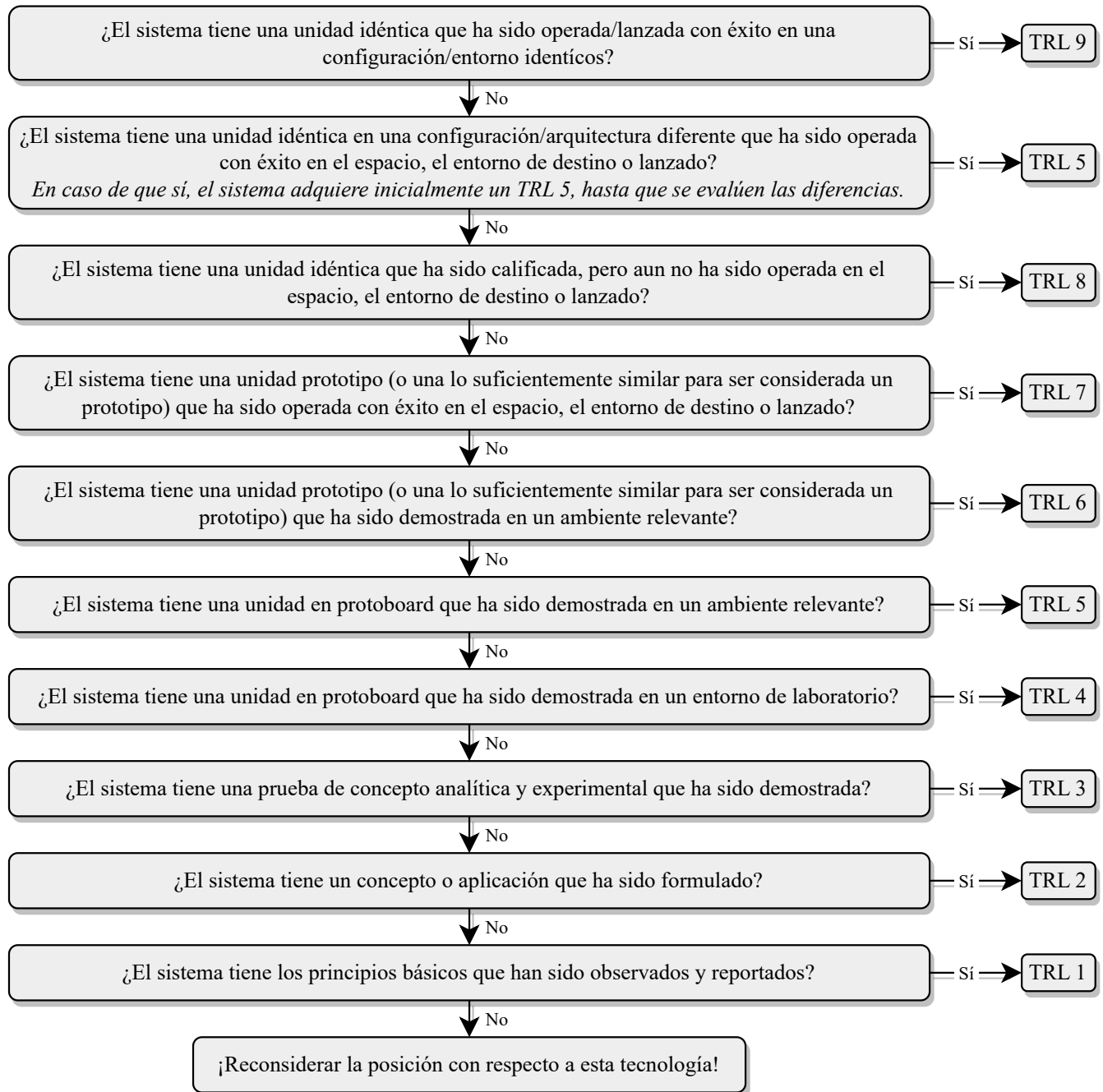


Figura 3.11: Proceso de evaluación de madurez tecnológica (TMA) [16]

4 EL MEDIO AMBIENTE DE RADIACIÓN ESPACIAL Y SUS EFECTOS EN LOS COMPONENTES ELECTRÓNICOS

En todo proceso de ingeniería se debe de considerar el entorno en donde se implementará el sistema. Hablando de los satélites artificiales, estos se colocarán en el espacio para que orbiten la Tierra durante un tiempo determinado. Esto plantea las siguientes preguntas: ¿a qué se enfrentará el satélite en órbita? ¿qué diferencias existen entre el medio ambiente terrestre y el espacial? ¿qué problemáticas hay que considerar para el diseño de un sistema que será expuesto al espacio? En este capítulo se presenta una introducción al medio ambiente de radiación espacial, definiendo sus características y las principales fuentes de radiación que deberá resistir un sistema para poder colocarse en órbita, además de los efectos que genera la radiación en los componentes electrónicos.



A diferencia de la creencia popular, el espacio no es un entorno totalmente vacío. Si bien, es cierto que la densidad de materia en el espacio es mucho más baja que la terrestre, existe un flujo continuo de partículas, cargadas o no, que pueden interactuar con un sistema electrónico a través de un intercambio de energía [25]. A este grupo de partículas se le conoce como radiación, la cual es uno de los aspectos ambientales más críticos que puede conducir a fallas en la tecnología moderna, especialmente en aplicaciones espaciales [26].

La Tierra está rodeada de muchos tipos de radiación y partículas generadas dentro y fuera del sistema solar, las cuales viajan casi libremente en el espacio hasta chocar con la atmósfera terrestre, sirviendo esta como un escudo protector natural que dispersa la energía en múltiples partículas con menor intensidad. En la figura 4.1 se presenta el concepto artístico del impacto de la radiación solar en la Tierra, siendo esta protegida por su campo magnético.

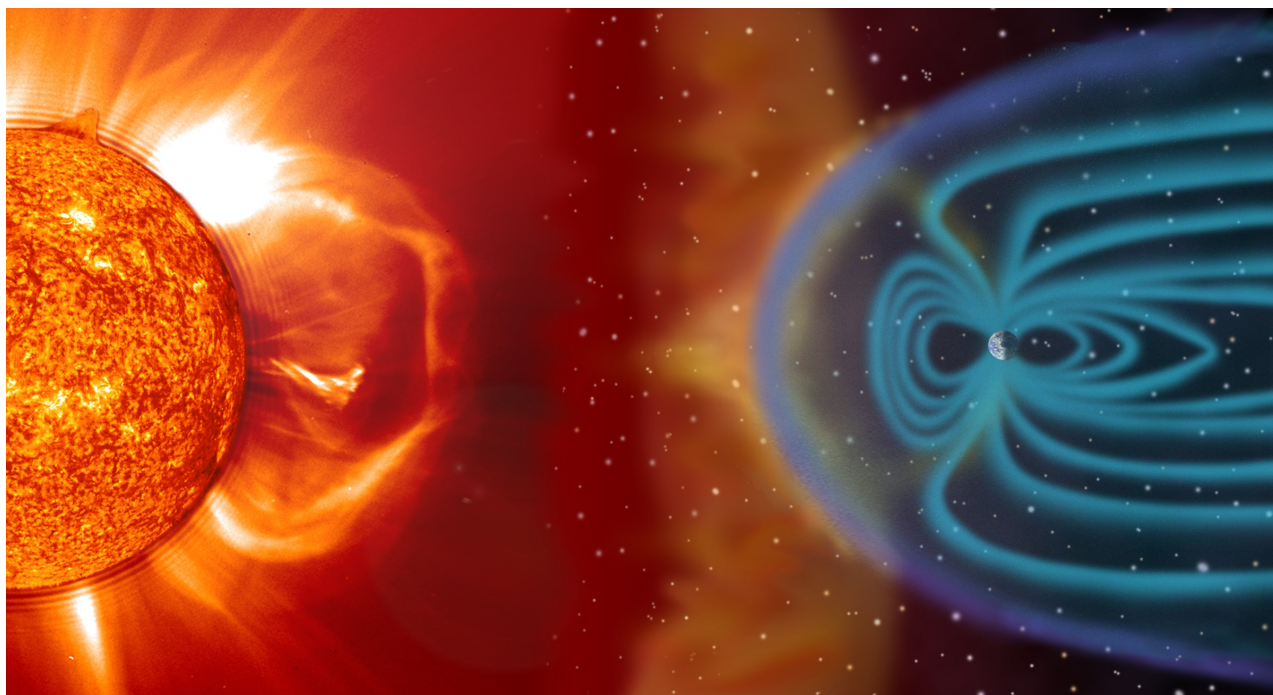


Figura 4.1: Representación del campo magnético terrestre protegiendo a la Tierra de la radiación solar [6]

4.1 Fuentes de la radiación

Es un hecho que el espacio contiene partículas radiactivas, pero ¿de donde provienen estas partículas? De manera general y considerando el diminuto tamaño del sistema solar en comparación con el cosmos, la mayoría de las partículas presentes en el espacio tienen su origen por reacciones nucleares en las estrellas. Estas partículas se caracterizan por tener un alto nivel energético que, como ya se mencionó, disminuye al acercarse a la Tierra e impactarse contra átomos y moléculas que componen a la atmósfera terrestre, transformándose en diferentes partículas que pueden ser atrapadas dentro del campo magnético del planeta. De este modo, el entorno de radiación que representa un riesgo para los satélites artificiales se divide en dos tipos: el medio ambiente de radiación atrapada y el medio ambiente de radiación en tránsito.

4.1.1 Ambiente de radiación atrapada

Como su nombre lo indica, el medio ambiente de radiación atrapada está formado por partículas que fueron confinadas por el campo magnético de la Tierra, viajando con una trayectoria en espiral alrededor de las líneas del campo magnético terrestre durante periodos de tiempo que van desde días hasta años. El movimiento de estas partículas forma una nube de radiación que rodea al planeta conocida como los cinturones de radiación de Van Allen (véase la figura 4.2). Esta zona se divide en dos regiones: el cinturón interno y el externo. El cinturón interno se encuentra desde los 500 hasta los 6000 km de altura sobre la superficie terrestre, con un pico de intensidad localizado aproximadamente a los 3000 km, formado por protones, con una energía que va desde unas cuantas decenas de MeV¹², y electrones, cuya energía se encuentra entre 1 y 10 MeV. Por otro lado, el cinturón externo se extiende desde los 15 000 hasta los 60 000 km de altura sobre la superficie terrestre, presentando un pico de intensidad entre los 12 000 y 22 000 km y compuesto principalmente por electrones de alta energía cuyas concentraciones oscilan en un factor de 1000 [27], [28].

En comparación, el cinturón interno se caracteriza por tener una relativa estabilidad que se ve alterada principalmente por el ciclo solar. Sin embargo, este contiene una región de alta intensidad radiactiva provocada por el desplazamiento entre el centro del campo magnético y el centro geográfico terrestre, generando una zona de campo más débil ubicada en el Océano Atlántico Sur y otra opuesta de campo más fuerte en el norte de Asia. Debido a esto, las partículas cargadas pueden alcanzar altitudes más bajas formando la anomalía del Atlántico Sur (SAA, por sus siglas en inglés) que afecta a los satélites ubicados en LEO, a pesar de que estos orbiten a alturas menores a las del cinturón de radiación interno.

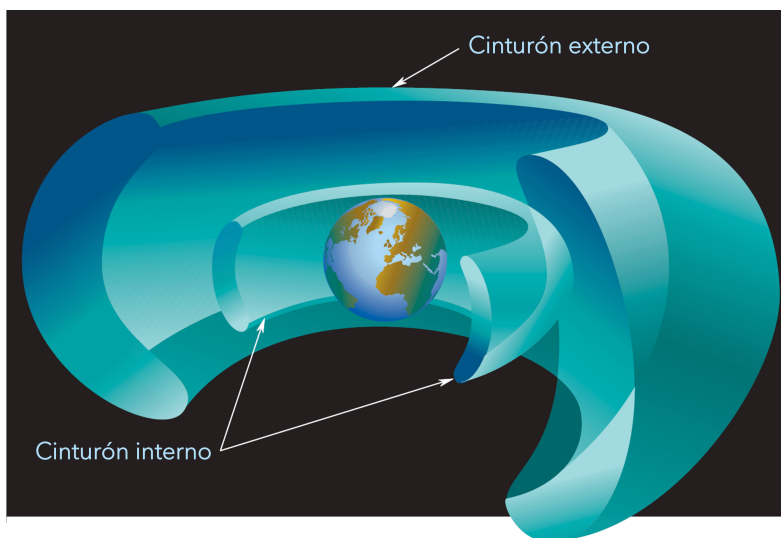


Figura 4.2: Representación de los cinturones de Van Allen terrestres [27]

¹²Unidad que representa la variación de energía necesaria en un electrón para moverse en un campo eléctrico con una diferencia de potencial de 1 V.

4.1.2 Ambiente de radiación en tránsito

A diferencia del ambiente de radiación atrapada, la radiación en tránsito está compuesta por partículas con el suficiente nivel de energía para no quedar atrapadas dentro del campo magnético terrestre, presentando un peligro mayor para los componentes electrónicos. Está compuesta principalmente por protones, electrones y núcleos pesados ionizados de muy alto nivel de energía que va desde 1 MeV/nucleón a más de 10 000 MeV/nucleón [28], generados por la actividad externa al sistema solar en forma de rayos cósmicos galácticos (GCR, por sus siglas en inglés) que acumulan múltiples fuentes de partículas y procesos de aceleración para los cuales normalmente el campo magnético terrestre no es lo suficientemente fuerte para desviarlos. Los GCR son la fuente de radiación de mayor energía en el sistema solar [27], cuyo nivel de intensidad puede ser modulado de forma inversamente proporcional a la actividad solar, el cual emite constantemente gas caliente ionizado, denominado como viento solar, con un rango de energía de entre 1 eV y 10 keV que tiene la capacidad de dificultar la propagación de las partículas de los GCR en el sistema solar durante sus periodos de mayor actividad. Sin embargo, el Sol también representa un riesgo, ya que dependiendo del ciclo solar¹³, este presenta periodos con intensos flujos de partículas de alta energía denominados como eventos de partículas solares (SPE, por sus siglas en inglés), en donde los niveles de energía de las partículas generadas por el Sol se disparan por encima de los MeV y, en algunos casos, pocos GeV, superando incluso a los GCR [28]. Además, el Sol también puede generar erupciones gigantes de gases y plasma en su superficie, produciendo fuertes distorsiones en el campo magnético terrestre, incrementando mucho la absorción de los GCR. En la figura 4.3 se representa a la radiación solar y a los GCR rodeando a la Tierra.

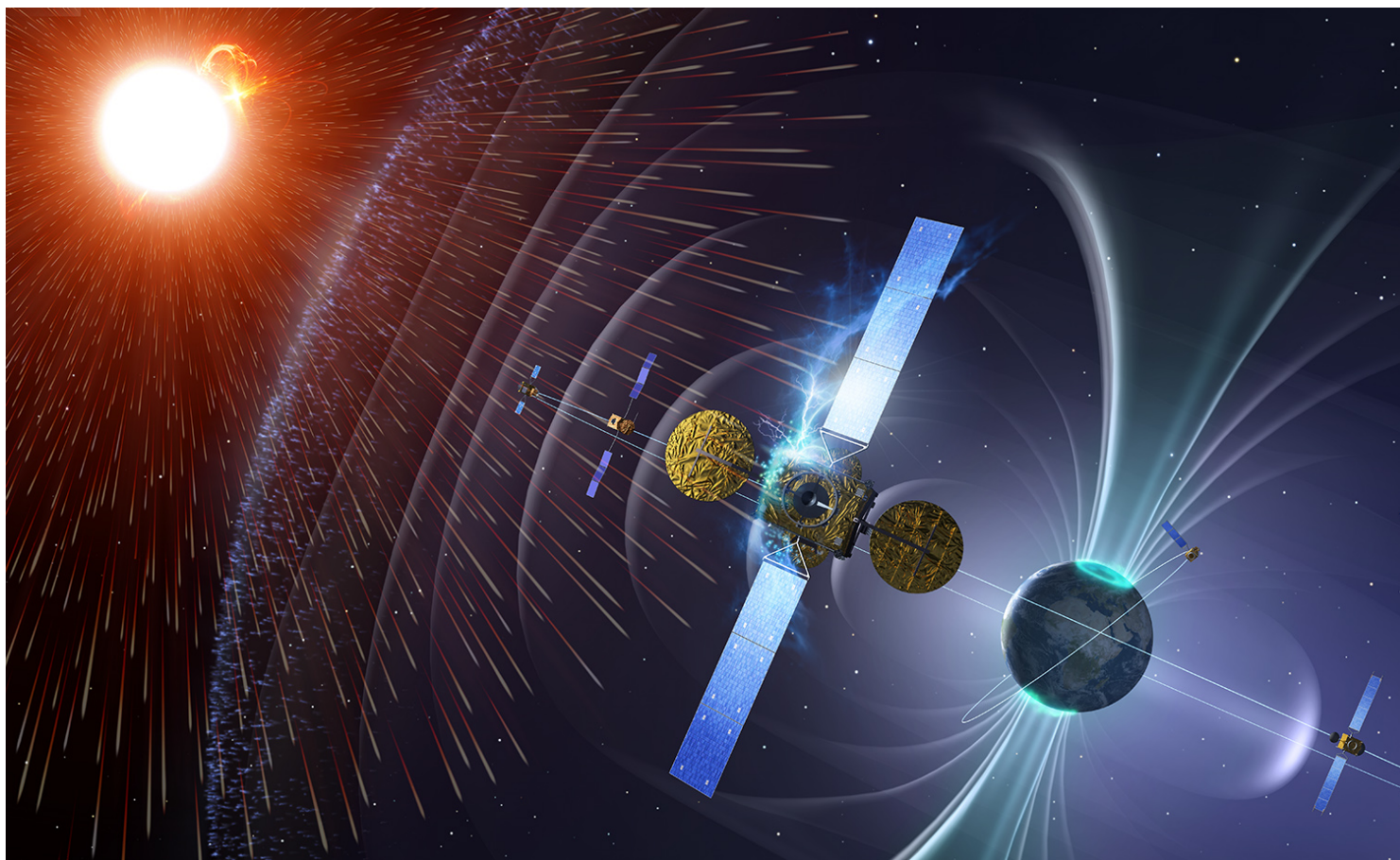


Figura 4.3: Representación de un satélite artificial y la Tierra siendo atacados por partículas de alta energía provenientes del Sol [29]

¹³Periodo en donde el campo magnético del Sol cambia constantemente, regresando a su posición inicial cada 11 años, aproximadamente.

4.2 Efectos de la radiación en los componentes electrónicos

Como ya se mencionó, el medio ambiente de radiación espacial está lleno de partículas con un cierto nivel de carga que interactúan con los dispositivos electrónicos a través de un intercambio de energía. Sin embargo, ahora es importante entender cómo se realiza esta interacción y qué efectos puede producir en los componentes electrónicos.

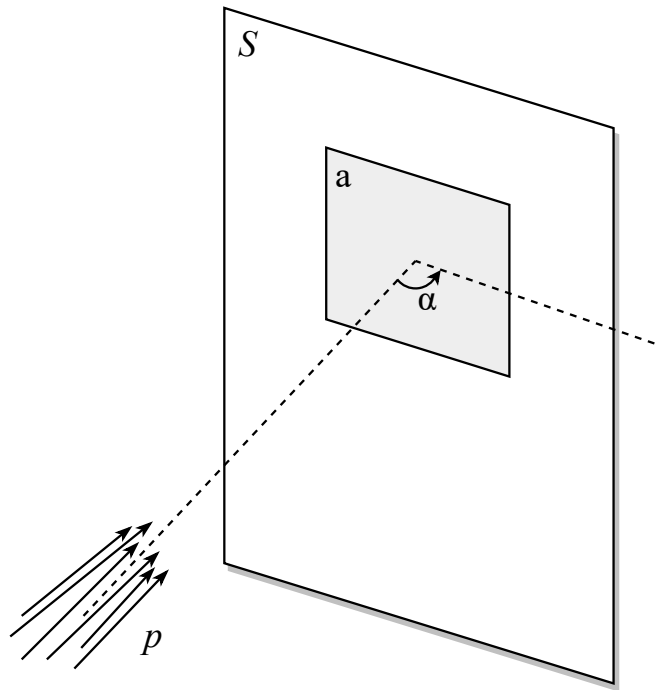


Figura 4.4: Representación de un flujo de partículas impactando un área unitaria de la superficie de un material [25]

Para caracterizar el ambiente de radiación y su interacción con los dispositivos electrónicos es necesario definir tres conceptos: el flujo de partículas (f), la fluencia de partículas (ϕ) y la sección transversal (σ). Empleando la figura 4.4 en donde se tiene una pieza de un material con una superficie S , el flujo de partículas f , definido en (4.1) y medido en [*partículas/cm²s*], representa el número de partículas p que impactan y cruzan una sección de área unitaria a de la superficie S del material en una unidad de tiempo t [25]. Cada una de estas partículas impacta a la superficie S con un ángulo de incidencia α diferente.

$$f = p/a \cdot t \quad (4.1)$$

La fluencia de partículas ϕ , definida en (4.2) y medida en [*partículas/cm²*], describe el número de partículas que cruzan la sección de área unitaria a durante un tiempo t determinado.

$$\phi = \int_0^t f(t)dt \quad (4.2)$$

Por último, la sección transversal σ , medida en [*cm²*] es una medida de área hipotética que representa a la superficie sensible del material, de tal manera que si una partícula cruza esta superficie habrá una interacción que puede provocar un efecto.

Una vez que la partícula impacta la superficie del material o, más específicamente, del dispositivo semiconductor, esta lo atraviesa y viaja a través de él dejando un camino o columna de material ionizado eléctricamente neutral (con

el mismo número de electrones y huecos) determinado por las colisiones de la partícula incidente, también llamada partícula primaria, con los núcleos del material, los cuales desvían a la partícula provocando que esta pierda parte de su energía y se generen partículas secundarias. En la figura 4.5 se presenta un diagrama en donde, por simplicidad, se traza un camino en forma de cilindro que describe la energía generada por las partículas secundarias, cuyo diámetro puede encontrarse entre pocas micras y algunas décimas de micras [30]. Sin embargo, algunas partículas secundarias con la energía (Δ) suficiente, pueden cruzar el material externo al cilindro [25].

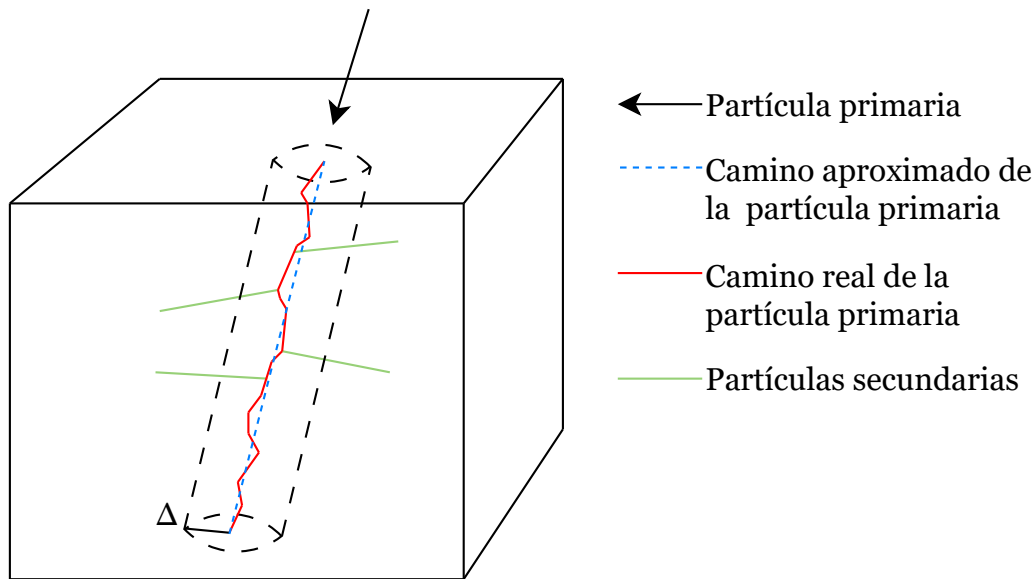


Figura 4.5: Comportamiento de una partícula al penetrar un material [25]

De esta manera, los efectos de la radiación pueden manifestarse a partir del impacto de una partícula primaria o las partículas secundarias generadas por su interacción con los núcleos del material, cuya sensibilidad se determina por su habilidad de “recolectar” la carga generada por las partículas incidentes [30]. Por otro lado, la severidad del daño generado por la radiación varía directamente en función del nivel de energía de la partícula, medido en electronvolt ($1eV = 1.602 \times 10^{-19} J$). En este punto aparecen los conceptos de “dosis de radiación”, que es la habilidad de la radiación de depositar energía por unidad de masa del material, expresada en Grays ($1Gy = 1J/kg$); y la “dosis de radiación absorbida”, que describe la cantidad de radiación absorbida por el material ($1rad = 100erg/g = 10mGy$). Todos los componentes EEE tienen una cierta carga crítica que, si se excede, resultará en uno o varios fenómenos indeseables [30]. De esta forma, todos los componentes electrónicos son susceptibles a los daños por radiación, los cuales se han convertido en uno de los problemas más desafiantes para los sistemas electrónicos modernos [31].

Los efectos de la radiación se presentan de muchas maneras y pueden ser clasificados dependiendo del tiempo necesario para que estos causen una falla o daño en el componente, dividiéndose en los efectos acumulativos y en los efectos de evento individual (SEE, por sus siglas en inglés).

4.2.1 Efectos acumulativos

La exposición de los componentes a la radiación puede generar efectos a largo plazo, degradando progresivamente sus características hasta generar anomalías en su funcionamiento. Estos se dividen en la dosis total de ionización (TID, por sus siglas en inglés) y en los daños por desplazamiento (DD, por sus siglas en inglés).

Comenzando con la TID, sus efectos son inducidos por la transferencia de energía ionizante derivada de la exposición homogénea y continua de un dispositivo a la radiación durante un largo periodo de tiempo, modificando y/o destruyendo los enlaces atómicos del material. Debido a esto, las características de los dispositivos electrónicos pueden sufrir diversas alteraciones, entre las que destacan:

- Cambios en los niveles de voltaje de umbral de los estados lógicos.
- Aumento en el consumo energético del dispositivo.
- Degradación del rendimiento del dispositivo y sus materiales.
- Daños permanentes en los semiconductores.
- Pérdida de funcionalidad en los circuitos electrónicos.
- Aumento de la sensibilidad a los SEE.

Por otro lado, los DD, también conocidos como dosis total no ionizante (TNID, por sus siglas en inglés), son inducidos por la transferencia de energía mecánica a través del impacto de las partículas con los átomos del material y, si es lo suficientemente fuerte para superar la energía de unión, provocar un desplazamiento de su ubicación normal dentro de la estructura cristalina hacia diferentes posiciones, perturbando el orden regular de los materiales semiconductores y generando cambios en la operación de los dispositivos electrónicos [32]. Los efectos producidos por los DD son similares a los de la TID, pero ocurren solo el 0.1 % de las ocasiones [26], además de que afectan a una variedad menor de componentes (como a los dispositivos opto-electrónicos). Por ejemplo, normalmente los circuitos integrados de aplicación específica (ASIC, por sus siglas en inglés) y los FPGA no son afectados por el DD [31].

Los efectos producidos por la TID y los DD pueden mitigarse parcialmente con blindaje alrededor del dispositivo, dispersando así la energía de las partículas que impactan el material.

4.2.2 Efectos de evento individual (SEE)

Los SEE son efectos transitorios resultantes del impacto de una sola partícula (véase la figura 4.6) con el nivel de energía suficiente para generar una respuesta inmediata en los dispositivos electrónicos. Esta respuesta varía dependiendo del volumen sensible atravesado por la partícula, dificultando la predicción del tipo de efecto producido y su ubicación, ya que pueden ocurrir en cualquier momento y lugar durante la misión.

El rango de problemas que generan los SEE en la funcionalidad de los dispositivos electrónicos es muy amplio, cuya gravedad depende de diversos factores, como la vulnerabilidad del componente, las operaciones que se encuentra realizando y la ubicación en donde impacta y atraviesa la partícula, además de la cantidad de energía que esta inyecta en el dispositivo durante su trayectoria. Si la partícula viaja a través de un nodo sensible del dispositivo o circuito, puede generar efectos adversos cuyo nombre define el SEE. En la tabla 4.1 se presentan los principales SEE inducidos en los dispositivos electrónicos, los cuales pueden dividirse en destructivos y no destructivos, dependiendo de la zona y el efecto generado en el componente y la posibilidad de este para regresar a un estado funcional a través de acciones correctivas.

A medida que los componentes electrónicos han disminuido en tamaño pero aumentado en complejidad, han adquirido mayor sensibilidad a la radiación y sus efectos, hablando principalmente de los SEE [28]. Esto se debe a que la carga necesaria para que una partícula genere un SEE se reduce junto al volumen del transistor. Además, el aumento en la densidad de elementos internos y la complejidad de los circuitos embebidos en los dispositivos modernos dificulta en gran medida los procesos necesarios para probar la respuesta de los sistemas a los SEE en todas sus formas posibles [28].

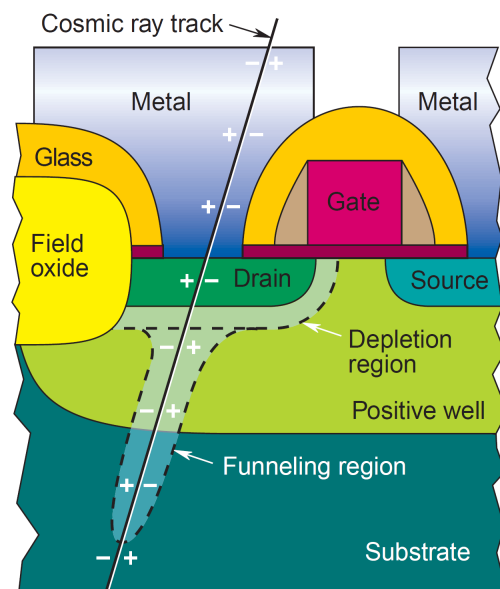


Figura 4.6: Camino ionizado generado por el paso de un GCR a través del drenaje de un transistor NMOS [30]

Tabla 4.1: Efectos de evento individual (SEE) (adaptado de [31], [32])

SEE	Efecto	Descripción	Mitigación
SEE no destructivos			
SET	Transitorio	Pico de voltaje temporal y corriente transitoria que causa un funcionamiento erróneo en el dispositivo. Puede excitar circuitos digitales y analógicos.	Circuitos temporizados (para circuitos digitales) y cargas RC (para circuitos analógicos).
SEU	Upset	Perturbación o transición del estado lógico de un bit (<i>bit-flip</i>) en circuitos de memoria.	Técnicas de detección y corrección de errores (EDAC, por sus siglas en inglés).
MBU	Upset de bits múltiples	Múltiples upsets generados por una misma partícula en una misma palabra de datos.	Códigos de corrección de errores (ECC, por sus siglas en inglés) de más de un bit.
MCU	Upset de celdas múltiples	Múltiples upsets generados por una misma partícula en diferentes celdas de memoria, normalmente adyacentes entre sí.	
SEFI	Interrupción funcional	Perturbación o transición del estado lógico en una sección de estado/control del dispositivo, generando que este se reinicie o pierda la funcionalidad.	Varía dependiendo del error producido. Típicamente se utiliza un reset por software o por ciclo de suministro.
SEE destructivos			
SEL	Latch-up	Flujo de corriente no regulado generado por el disparo de un tiristor parásito.	Interrupción de la alimentación del dispositivo / circuito y limitadores de corriente.
SESB	Snap-back	Disparo de una estructura bipolar parásita (NPN o PNP).	
SEHE	Error grave	Cambio inalterable de estado asociado con un daño permanente o semi-permanente de una celda de memoria.	
SEB	Burnout	Disparo de una estructura bipolar parásita en un transistor de potencia, acompañado de una alta corriente.	Reducción del nivel de tensión de operación.
SEGR / SEDR	Ruptura de puerta / dieléctrico	Ruptura de la puerta de óxido o de cualquier capa del dieléctrico que genera corrientes de fuga en MOSFET de potencia.	El daño inducido al dieléctrico es muy rápido comparado con el tiempo de respuesta de cualquier protección eléctrica, por lo que no se puede proteger del efecto.

La sensibilidad de un dispositivo electrónico a los SEE es expresada por la sección transversal, en función de la transferencia lineal de energía (LET, por sus siglas en inglés) de la partícula, la cual representa la cantidad de

energía $-dE$ transferida al material por unidad de longitud ds , mientras esta viaja a través de él, definida en (4.3) y medida en $MeVcm^3$, siendo directamente proporcional al número de pares de electrones generados por unidad de longitud [33]. De esta manera, para que una partícula produzca un SEE en un dispositivo, esta debe tener una energía que supere el umbral LET (LETth) del material. Si la partícula no sobrepasa el LETth, la cantidad de energía ionizada se acumulará en el material de modo que, a largo plazo, se produzcan los efectos relacionados a la TID.

$$LET_{\Delta} = \frac{-dE_{\Delta}}{ds} \tag{4.3}$$

Parte de la complejidad para medir la sensibilidad de un dispositivo electrónico a los SEE se debe a que, al ser totalmente aleatorios en tiempo y ubicación, pueden generar fallas que no necesariamente tengan un impacto significativo en la operación del componente. En otras palabras, es posible que la misma lógica o la tarea que estaba realizando el sistema cuando se generó el SEE sea capaz de aislar, ignorar o reparar el daño [28]. Por lo tanto, la capacidad de un sistema para soportar las consecuencias de los SEE depende significativamente del conocimiento del diseñador y su habilidad para anticipar lo que le puede ocurrir al sistema durante su tiempo de vida operacional. En la tabla 4.2 se presentan los tipos de tecnología y familias de los componentes electrónicos en donde es común la ocurrencia de SEE.

Tabla 4.2: Principales SEE en función de la tecnología y familia del componente [31]

Tecnología	Familia	Función	SEE no destructivos				SEE destructivos			
			SET	SEU	MBU MCU	SEFI	SEL SESB	SEHE	SEB	SEGR SEDR
MOS de potencia									X	X
CMOS, BiCMOS y SOI	Digital	SRAM		X	X		X	X		
		DRAM		X	X	X	X	X		
		FPGA	X	X	X	X	X	X		
		Flash EEPROM		X		X	X	X		
		μ p / μ controlador	X	X	X	X	X			
	Señal mixta	ADC	X	X		X	X		X	X
		DAC	X	X		X	X		X	X
	Lineal		X			X		X	X	
Bipolar	Digital		X	X					X	X
	Lineal		X	X					X	X

5 FPGA EN EL ESPACIO

Hasta este punto del trabajo es claro el estado actual del sector espacial, el impacto de los nanosatélites en el área, las condiciones del medio ambiente de radiación espacial y los efectos que este puede generar en los dispositivos electrónicos, por lo que es necesario ubicar qué tipo de tecnología se debe de utilizar para satisfacer las necesidades de una misión espacial y qué técnicas existen para el dispositivo seleccionado. En este capítulo se presentan los antecedentes teóricos necesarios para entrar a la etapa de diseño, abordando el concepto del FPGA, su estructura y sus características para entender por qué es un dispositivo con excelentes propiedades para utilizarse en aplicaciones espaciales y los fundamentos de tolerancia a fallas para aumentar el nivel de fiabilidad y, posteriormente, concentrarnos en las principales técnicas aplicadas en FPGA.



5.1 Antecedentes del FPGA

La idea del FPGA surgió a los inicios de la década de los 80 como una convergencia de los dispositivos lógicos programables (PLD, por sus siglas en inglés) y los ASIC al aplicar una aproximación diferente a los dispositivos lógicos programables complejos (CPLD, por sus siglas en inglés) para escalar el tamaño y densidad de los elementos reconfigurables del dispositivo, pero fue hasta 1984 que *Xilinx* desarrolló el concepto y lo presentó al mercado [34]. La arquitectura general de un FPGA (véase la figura 5.1) consiste en una matriz de bloques lógicos configurables (CLB, por sus siglas en inglés) distribuidos a lo largo de todo el dispositivo, conectados a través de una red de interconexiones programable por el usuario y a bloques de entrada/salida (IOB, por sus siglas en inglés), que permiten la interacción del dispositivo con el exterior.

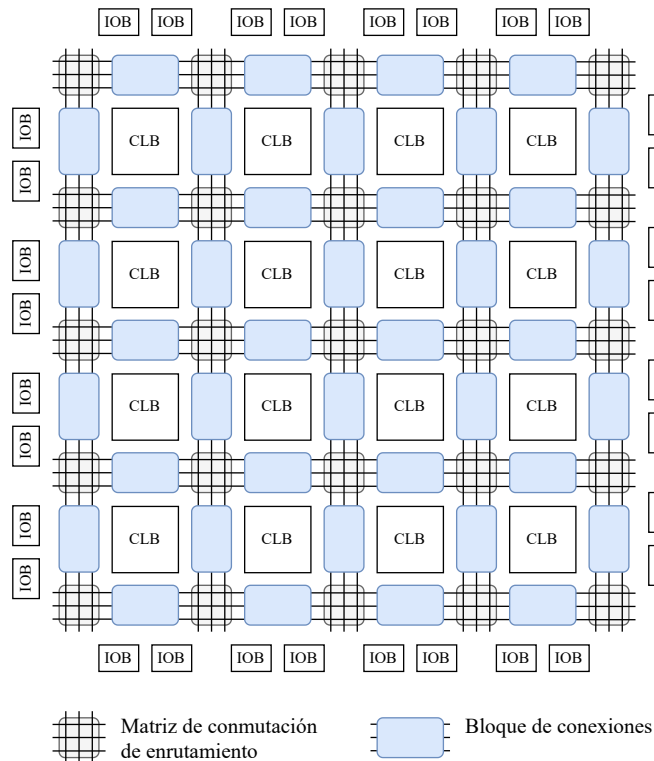


Figura 5.1: Arquitectura general de un FPGA

El principio de operación del FPGA parte de la estructura lógica de su elemento básico, el CLB, denominado como celda lógica (LC, por sus siglas en inglés) por *Xilinx*, que utiliza las tablas de búsqueda (LUT, por sus siglas en inglés) para emular cualquier circuito lógico, en función de la configuración de la tabla de verdad almacenada en la memoria del FPGA. En la figura 5.2 se presenta la estructura básica de un CLB, en donde la LUT determina la función lógica construida por el usuario, cuya salida se selecciona mediante un multiplexor para implementar un circuito combinacional o secuencial a través de un *flip-flop* (FF) de tipo D.

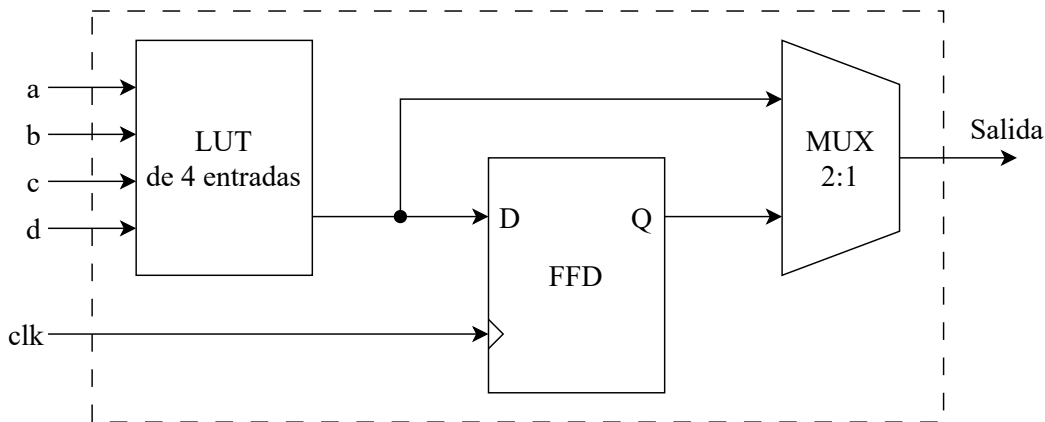


Figura 5.2: Estructura básica de un bloque lógico configurable (CLB) en un FPGA

La estructura del FPGA y sus elementos internos varía en función del año y el fabricante, pero siguen el mismo principio de operación añadiendo algunos elementos adicionales. Por ejemplo, los FPGA modernos integran bloques de alto nivel que agregan mayor funcionalidad al dispositivo, tales como bloques de procesamiento digital de señales (DSP, por sus siglas en inglés), bloques de memoria de acceso aleatorio (BRAM, por sus siglas en inglés), multiplicadores, periféricos de entrada/salida de alta velocidad y procesadores (véase la figura 5.3), que se encuentran embebidos dentro del circuito integrado y pueden ser conectados individualmente o como bloques a través de la matriz de conexiones programables del FPGA, dependiendo de las necesidades del usuario. Con esta idea aparecen los SoC, que mezclan el concepto del FPGA con uno o varios núcleos de procesador embebidos en un único chip, denominados como *hardcores*, los cuales pueden estar integrados afuera (figura 5.4a) o dentro de la zona configurable del FPGA (figuras 5.4b y 5.4c).

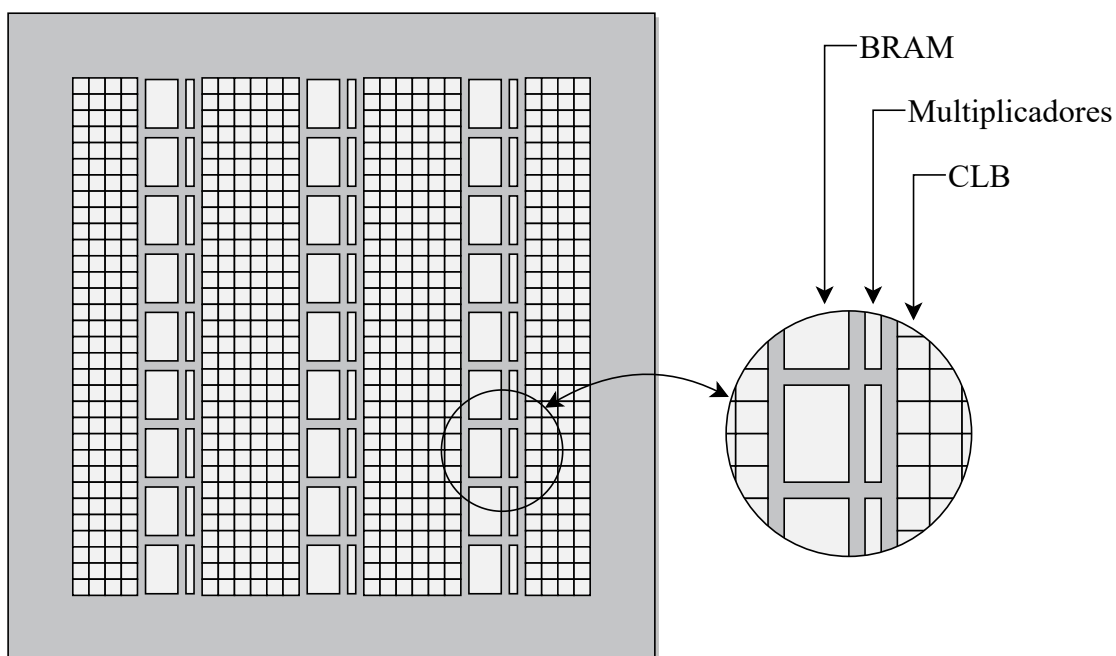


Figura 5.3: FPGA con diversos módulos embebidos distribuidos en todo el chip [34]

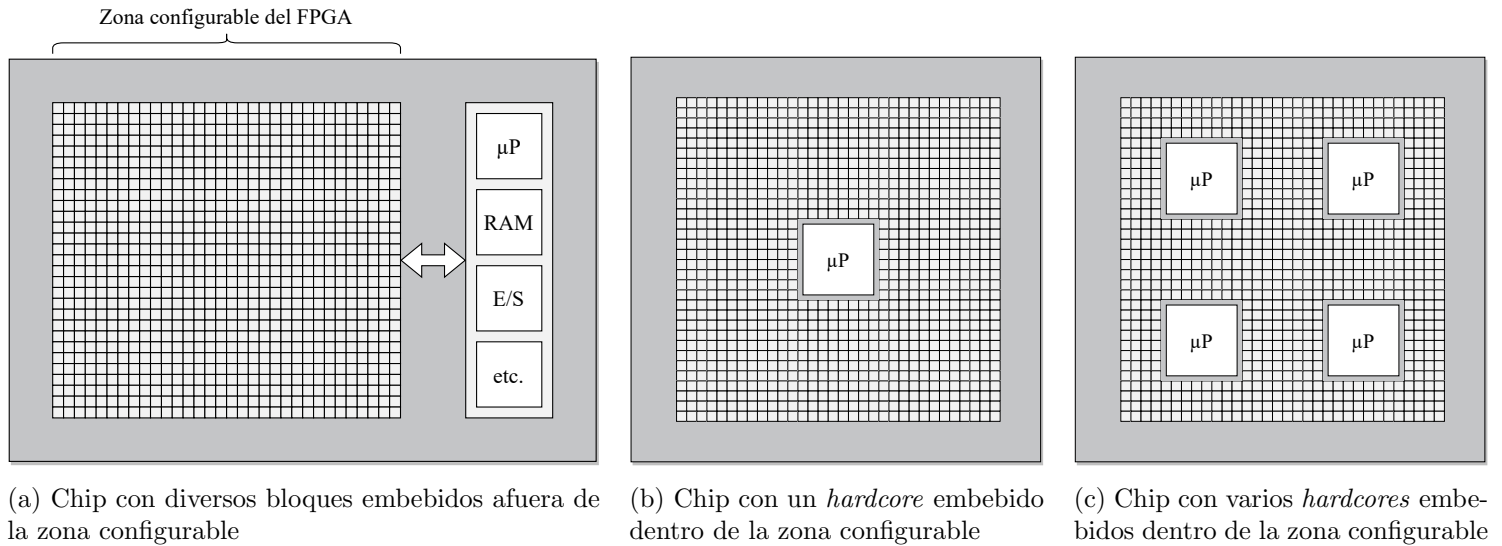


Figura 5.4: Diferentes estructuras de SoC con *hardcores* embebidos y zonas configurables de FPGA [34]

De esta manera, los SoC pueden tener bloques embebidos con diferente propósito que, en combinación con las zonas configurables del FPGA, permiten construir sistemas de muy alto desempeño, capaces de cubrir un amplio rango de aplicaciones. Sin embargo, este comportamiento se puede replicar por un FPGA al configurar un grupo de CLB para construir diferentes circuitos de alto nivel, como memorias, periféricos o procesadores, los cuales también están disponibles por el mismo fabricante o por terceros como módulos ya sintetizados denominados como bloques de propiedad intelectual (IP, por sus siglas en inglés) que el diseñador importa a la herramienta de diseño y puede configurar su funcionamiento, dependiendo de las capacidades del módulo. En el caso de los procesadores, estos se conocen como *softcore*.

5.1.1 Tecnologías de configuración del FPGA

Dependiendo del tipo de tecnología utilizada para almacenar los datos de configuración, los FPGA se clasifican en tres tipos:

FPGA basados en antifuse

Antifuse es una tecnología consistente en dos terminales aisladas entre sí por una resistencia de alto valor, la cual se funde al aplicar pulsos de voltaje y corriente en las zonas determinadas por un archivo de configuración cargado en un programador especial. Este tipo de tecnología permite integrar una alta densidad de elementos dentro de un mismo chip, además de que, por su estructura, presentan una muy alta inmunidad contra los SEE. Sin embargo, una vez “quemados” los fusibles el FPGA pierde la capacidad de reconfiguración al conectarlos permanentemente.

FPGA basados en memoria estática de acceso aleatorio (SRAM)

Como su nombre lo indica, está basada en una memoria estática de acceso aleatorio (SRAM, por sus siglas en inglés) para configurar al FPGA. Sin embargo, debido a que es una tecnología volátil, esta debe programarse siempre que se energice el sistema, por lo que se requiere de una memoria no volátil como parte del circuito de configuración del FPGA, como se observa en la figura 5.5. Entre sus grandes ventajas destacan su muy alta velocidad de reconfiguración y el número ilimitado de veces que se puede realizar, dando amplia versatilidad para reconfigurar al FPGA durante la aplicación, ya tanto para cambiar sus funciones como para corregir errores.

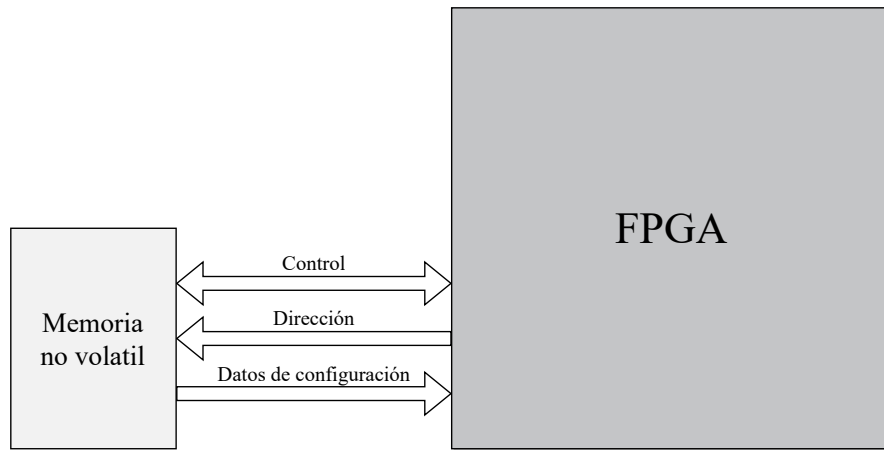


Figura 5.5: Carga paralela de los datos de configuración de un FPGA como maestro [34]

FPGA basados en flash

Los FPGA basados en memoria flash presentan mayor resistencia a los SEE, en comparación con los FPGA basados en SRAM, además de que es una memoria no volátil y no requieren de una memoria externa para configurarse, pero tienen un mayor costo y su capacidad y velocidad de reconfiguración es menor.

En la tabla 5.1 se presenta una comparación de las características principales de las tecnologías empleadas para el FPGA.

Tabla 5.1: Características del FPGA en función de su tecnología [34], [35]

Tipo de tecnología	<i>Antifuse</i>	SRAM	Flash
Estado de la tecnología	Una o más generaciones de retraso	Dentro del estado del arte	Una o más generaciones de retraso
Reconfigurable	No	Sí	Sí
Capacidad de reconfiguración	Nula	Ilimitada	Limitada
Velocidad de reconfiguración	–	Alta	Media (3 veces más baja que la SRAM)
Volatilidad	No	Sí	No
Necesidad de un circuito externo de reconfiguración	No	Sí	No
Buena para prototipos	No	Sí (muy buena)	Sí (razonable)
Uso instantáneo al encender	Sí	No	Sí
Tamaño de la celda de configuración	Muy pequeña	Grande (seis transistores)	Media – pequeña (dos transistores)
Inmunidad contra SEE	Muy alta	Baja	Media – alta
Endurecimiento a la radiación	Sí	No	No realmente
Consumo energético	Bajo	Medio	Medio
Costo	Muy alto	Medio	Alto

5.2 Fundamentos de la mitigación y la confiabilidad

Como se mencionó en el capítulo 4, los sistemas espaciales trabajarán en condiciones ambientales que pueden generar problemas de funcionalidad en los dispositivos electrónicos. Para contrarrestar esto, se utilizan técnicas de mitigación que, comenzando por su definición, consisten en la aplicación de funciones especiales con el objetivo de reducir el riesgo de error, derivado de la ocurrencia de un evento indeseable. Es importante entender que, dependiendo del criterio, las técnicas de mitigación se pueden aplicar en diferentes niveles, como se observa en la figura 5.6.

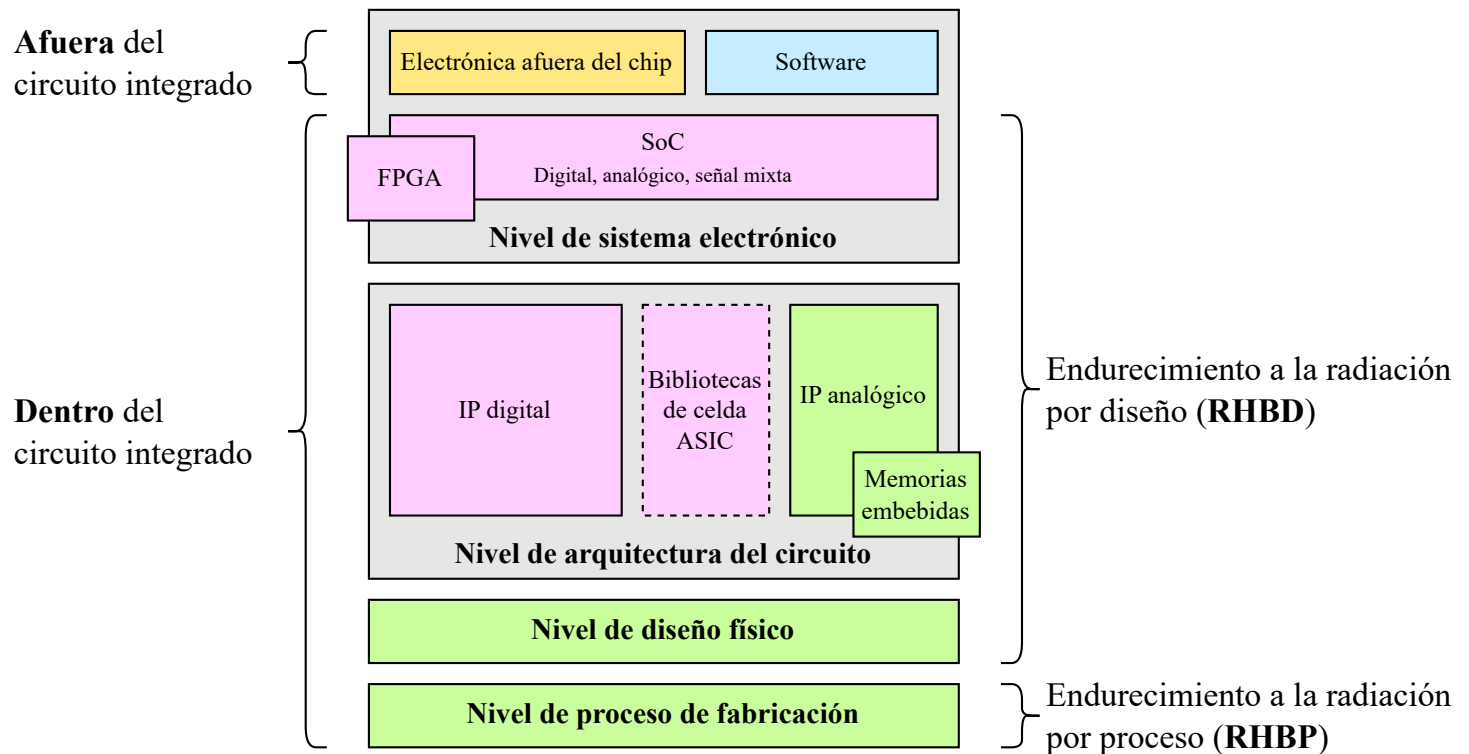


Figura 5.6: Niveles de abstracción en donde se pueden aplicar técnicas de mitigación [31]

Analizando el diagrama desde el nivel más bajo de abstracción, las técnicas de mitigación se pueden aplicar en cuatro niveles, cada uno con los siguientes objetivos:

- **Nivel de proceso de fabricación:** reducir el impacto de la radiación en los circuitos integrados al aplicar técnicas de endurecimiento durante el proceso de fabricación, esquema conocido como endurecimiento a la radiación por proceso (RHBP, por sus siglas en inglés).
- **Nivel de diseño físico:** mitigar los efectos causados por la radiación al optimizar la distribución y el *layout* de los transistores para reducir la sensibilidad del circuito a la radiación.
- **Nivel de arquitectura del circuito:** mitigar los efectos causados por la radiación al aplicar técnicas EDAC, dependiendo de la naturaleza del dispositivo.
- **Nivel de sistema electrónico:** mitigar los efectos causados por la radiación aplicando técnicas de tolerancia a fallas a nivel de componente, unidad o software embebido.

Considerando los alcances del proyecto, nos enfocaremos en las técnicas de mitigación en los niveles de arquitectura del circuito aplicadas en un FPGA. Sin embargo, antes de entrar a detalle en las técnicas de mitigación que se aplicarán en el SCMI, es importante definir el concepto de **confiabilidad** (*dependability*), que consiste en la propiedad de un

sistema para conservar, en un límite de tiempo establecido, los valores de todos los parámetros que caracterizan su capacidad para desempeñar correctamente sus funciones, con base en regímenes y condiciones determinadas.

Para que un sistema se considere confiable, este debe de cumplir con distintos factores, como el grado de calidad de sus componentes y la aplicación de técnicas dirigidas a “proteger” al sistema, de modo que si se presenta algún evento inesperado, disminuya la probabilidad de que el sistema deje de funcionar correctamente. Los esquemas usados para estos propósitos se definen como técnicas de tolerancia a fallas. Antes de entrar a detalle en las técnicas de tolerancia a fallas, es importante definir las tres propiedades fundamentales de la confiabilidad: sus amenazas, sus atributos y sus medios.

5.2.1 Amenazas de la confiabilidad

Las amenazas de la confiabilidad expresan las razones por las que un sistema deja de desempeñar sus funciones [36]. De esta manera, el concepto principal es la **falla**, que es una condición irregular producida por un defecto físico, imperfección, problema de manufactura, error en el diseño o por perturbaciones externas, que ocurre dentro de algún componente de hardware o software. Como se presenta en la figura 5.7, una falla puede categorizarse por cinco atributos: por su causa, ocurriendo dentro o fuera de los componentes, o durante el proceso de diseño de los componentes o del sistema [37]; por su naturaleza, que especifica el tipo de falla; por su duración, en función del tiempo de vida de la falla; el alcance de la falla, a partir de la ubicación de la falla, afectando un módulo de hardware/software o de manera global al sistema; y por su valor, dependiendo de la estabilidad de la falla con el paso del tiempo.

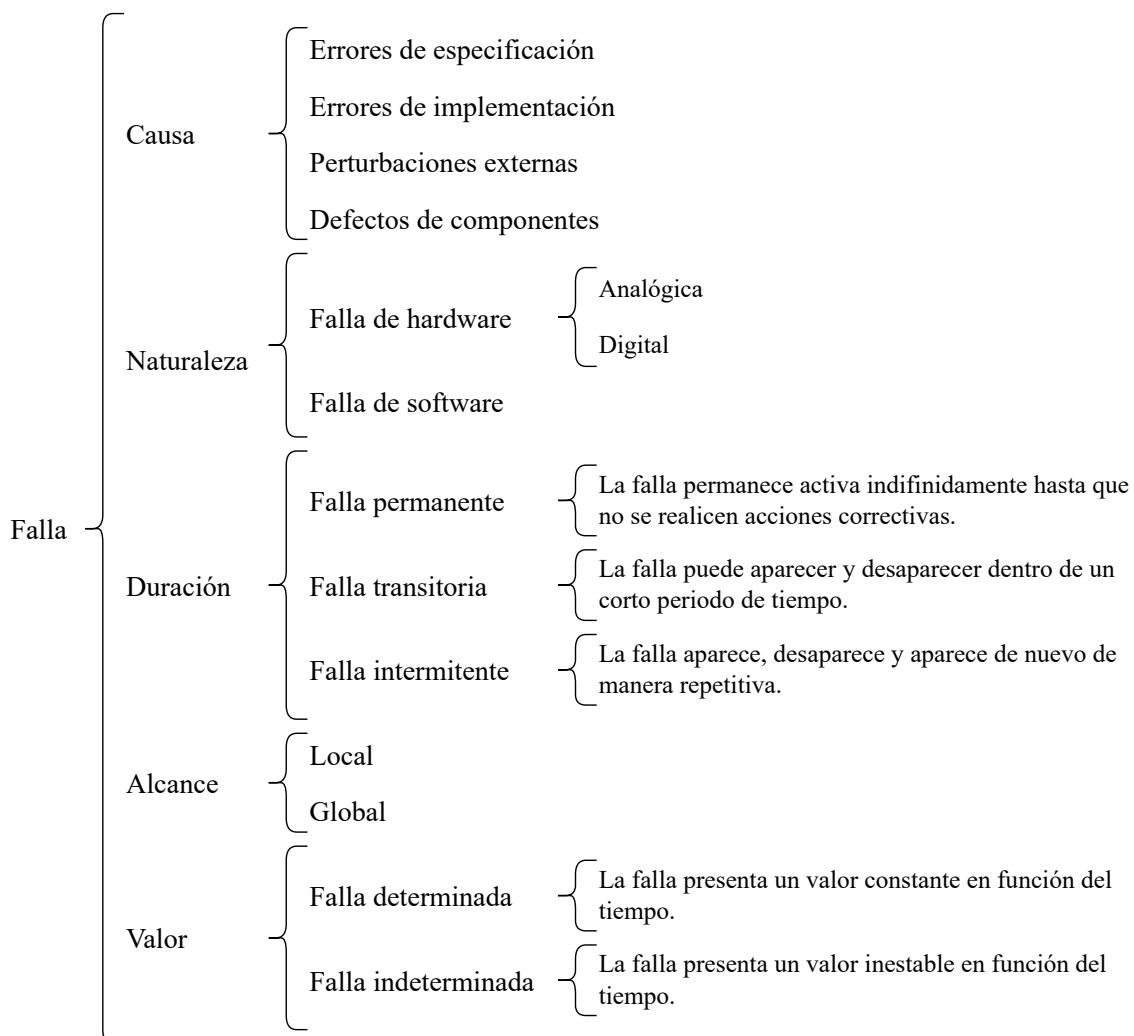


Figura 5.7: Categorización de una falla [37]

Una falla puede existir en un sistema sin que este presente una respuesta incorrecta, lo que se denomina como falla latente, que puede activarse a través de un componente que forma parte del proceso de cálculo del sistema, generando una falla activa (que también puede presentarse directamente desde el exterior del sistema) generando un **error** como consecuencia, definido como una desviación de la precisión o un estado incorrecto resultado de una falla [38]. Un error puede propagarse a través de la interfaz de conexión de un componente que presenta un error con otro componente, dando lugar a otros errores. La propagación de uno o varios errores puede ocasionar un **malfuncionamiento**, que sucede cuando el error hace que el sistema realice sus funciones incorrectamente [39], de modo que este pierda la capacidad de cumplir con los servicios para los que fue diseñado. De esta manera, la presencia de una falla en un elemento puede generar un error en algún componente, que al propagarse puede ocasionar el malfuncionamiento del sistema, que a su vez puede generar una falla externa permanente o transitoria en otro sistema que reciba como entrada los datos del sistema que presentó la falla inicialmente [40]. El proceso anterior se conoce como cadena fundamental de la fiabilidad y sus amenazas, que se presenta en la figura 5.8.

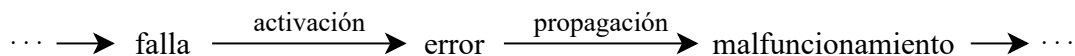


Figura 5.8: Cadena fundamental de la confiabilidad y sus amenazas [40]

5.2.2 Atributos de la confiabilidad

Los atributos de la confiabilidad expresan, de manera cuantitativa y cualitativa, las propiedades esperadas de un sistema. Los tres atributos principales son la fiabilidad, la disponibilidad y la seguridad, pero dependiendo de la aplicación y las métricas establecidas, se pueden considerar también la mantenibilidad, la comprobabilidad y el rendimiento [41].

Comenzando con la **fiabilidad** (*reliability*, $R(t)$), es la probabilidad de que el sistema opere correctamente durante el intervalo de tiempo $[t_0, t]$, dado que el sistema se encontraba funcionando correctamente en el tiempo t_0 . Dicho de otra forma, la fiabilidad es la medida del funcionamiento continuo correcto de un sistema [36]. De manera inversa, la ausencia de fiabilidad (*unreliability*, $F(t)$) es definida como la probabilidad condicional del sistema operando incorrectamente [38]. En 5.1 se presenta la relación entre la fiabilidad y la ausencia de fiabilidad de un sistema.

$$R(t) + F(t) = 1 \quad (5.1)$$

La función de densidad de probabilidad de falla ($f(t)$) puede calcularse en función de la ausencia de confiabilidad del sistema, como se muestra en 5.2.

$$f(t) = \frac{dF(t)}{dt} \quad (5.2)$$

En cuanto a la **disponibilidad** (*availability*, $A(t)$), esta es la probabilidad de que el sistema se encuentre funcionando correctamente en el instante de tiempo t , además de que se encuentre disponible para desempeñar sus funciones determinadas. Dicho de otra forma, es la fracción del tiempo en que el sistema se encuentra en estado de operación [36]. A diferencia de la fiabilidad que considera un intervalo de tiempo, la disponibilidad toma un instante de tiempo, que también representa el tiempo requerido para realizar acciones correctivas en el sistema. A partir de la disponibilidad se obtiene el intervalo de disponibilidad $A(T)$ (5.3), que representa el valor del punto de disponibilidad promediado sobre un intervalo de tiempo T , denotando el tiempo de vida del sistema o el tiempo necesario para completar alguna tarea en particular [36].

$$A(T) = \frac{1}{T} \int_0^T A(t) dt \quad (5.3)$$

5.2.3 Medios de la confiabilidad

Los medios de la confiabilidad son los métodos aplicados durante el proceso de diseño para que un sistema alcance un nivel de confiabilidad determinado. Se clasifican en cuatro tipos, descritos a continuación.

Prevención de fallas

La **prevención de fallas** consiste en las metodologías aplicadas durante el proceso de diseño para evitar la ocurrencia de fallas en un sistema [40], eliminando las causas de fallas a través de rigurosas técnicas de control de calidad durante las etapas de especificación, implementación y fabricación del proceso de diseño, de modo que si el sistema es probado eficientemente, se pueden evitar muchas de sus fallas y componentes defectuosos [36].

Tolerancia a fallas

La **tolerancia a fallas** se refiere a la habilidad de un sistema para funcionar apropiadamente en el caso de que uno o más de sus componentes fallen, por lo que la presencia de fallas no resulta en un daño completo del sistema [38]. En la tolerancia a fallas se aplican cuatro conceptos fundamentales:

- **Enmascaramiento de fallas:** proceso de evitar la propagación de fallas en el sistema a través de un diseño redundante, de modo que las fallas sean transparentes y no tengan repercusiones en el sistema [38].
- **Detección de fallas:** proceso de determinación de la ocurrencia de una falla dentro del sistema.
- **Localización de fallas:** proceso de determinación de la ubicación en donde ocurrió una falla.
- **Contención de fallas:** proceso de aislamiento de una falla para prevenir su propagación y sus efectos en el sistema.

Eliminación de fallas

La **eliminación de fallas** son los métodos aplicados para reducir el número de fallas presentes en el sistema, aplicadas durante el proceso de diseño, a partir de técnicas de verificación, diagnóstico y corrección; y la vida útil, a través de tareas de mantenimiento correctivo y preventivo [36].

Predicción de fallas

La **predicción de fallas** se refiere a las técnicas que permiten estimar la cantidad de fallas presentes en el sistema, las posibles futuras ocurrencias de fallas y sus consecuencias, a partir de evaluaciones cuantitativas y cualitativas que permiten determinar la ocurrencia y activación de fallas durante la vida útil del sistema.

5.3 Estimación del nivel de fiabilidad

Recordando los objetivos del trabajo, la meta es diseñar un SCMI que alcance un nivel de fiabilidad suficiente para desempeñar misiones en el espacio. Con esto en mente y considerando las características del medio ambiente de radiación espacial, se pueden presentar fallas transitorias en duración e indeterminadas en valor, por lo que analizar sus efectos en sistemas digitales termina siendo una tarea extremadamente difícil [37], presentando la problemática: ¿cómo determinar el nivel de fiabilidad del sistema? Para resolver lo anterior, es necesario asumir que las fallas presentan un comportamiento que puede representarse a través de algún modelo matemático, lo que permitirá realizar una estimación de la ocurrencia de fallas para una aplicación dada y, de esta manera, tomar las decisiones correctas durante el proceso de diseño.

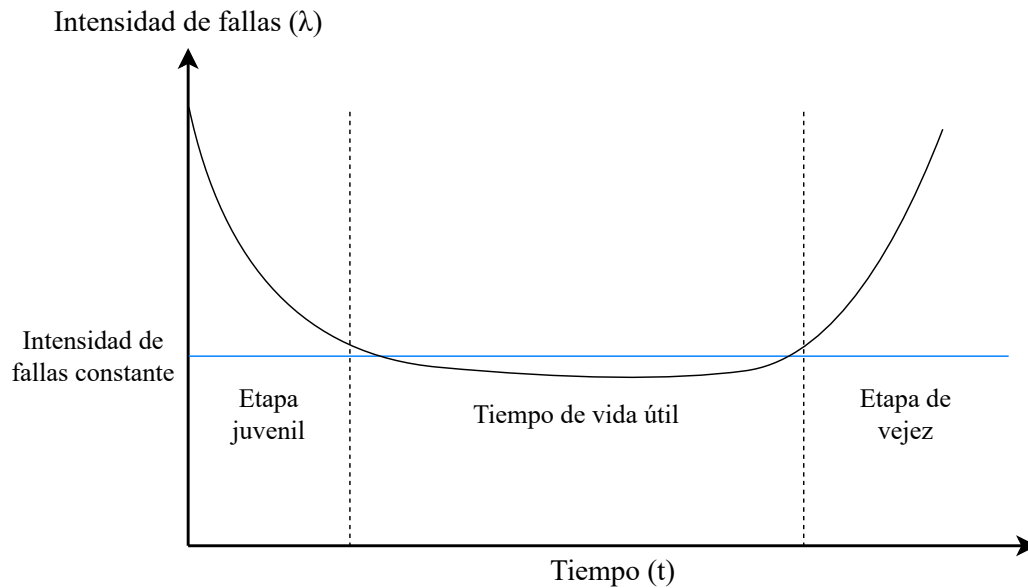


Figura 5.9: Intensidad de fallas en función del tiempo (curva de la bañera) [39]

Previamente al análisis, es importante comenzar con el concepto de intensidad de fallas (λ), que se entiende como el parámetro que indica la vulnerabilidad de un sistema para operar durante un tiempo (t). El cálculo de λ se puede realizar a través de distintas maneras, como lo son procesos prácticos, en donde se mantiene una muestra significativa de algún componente en operación para obtener la estadística en función del tiempo; procesos de aceleración de fallas, sometiendo a los componentes a condiciones extremas para calificar su nivel de calidad, lo que permite identificar sus defectos de origen; o mediante la caracterización de los componentes, empleando las funciones de transferencia que entrega el fabricante para realizar un análisis, considerando las características del ambiente en donde operará el sistema. En el caso particular de este proyecto, los elementos del sistema son componentes electrónicos, para los cuales se ha demostrado que durante su tiempo de vida útil, su función de intensidad de fallas presenta un comportamiento constante [39]. Esto se observa en la figura 5.9, en donde la intensidad de fallas disminuye durante la etapa juvenil del componente, entendiéndose como el periodo de tiempo entre la fabricación del componente hasta el fin de los procesos de control de calidad para validar si cumplen con las especificaciones de diseño, para después mantenerse prácticamente constante durante su largo tiempo de vida útil y, finalmente, aumentar de nuevo después de un tiempo considerable en donde el material semiconductor sufre una degradación natural en sus parámetros debido a su uso prolongado.

En este punto, el interés principal se encuentra en el tiempo de vida útil que, al presentar un comportamiento constante, facilita la obtención de una función matemática que permita determinar la intensidad de fallas y, con esto, la probabilidad de que el sistema no presente fallas durante un tiempo determinado. Entrando en términos de la fiabilidad, se tiene un sistema con N componentes que se encuentran operando al inicio del tiempo de prueba (t_0), de los cuales N_0 representa el número de componentes que continuarán funcionando y N_F el número de componentes que presentaron fallas para un instante de tiempo t , por lo que N representa el universo, como se presenta en 5.4.

$$N = N_0(t) + N_F(t) \quad (5.4)$$

Considerando lo anterior, la fiabilidad ($R(t)$) en un instante de tiempo t se calcula como la relación del número de elementos que continúan operando en el tiempo t , considerando que habían N componentes operando en el tiempo t_0 (5.5).

$$R(t) = \frac{N_0(t)}{N} \quad (5.5)$$

De manera contraria, la ausencia de fiabilidad ($F(t)$) en un instante de tiempo t se calcula como la relación del número de componentes que presentaron fallas para el tiempo t , considerando que habían N componentes operando en el tiempo t_0 (5.6).

$$F(t) = \frac{N_F(t)}{N} \quad (5.6)$$

De esta manera, sustituyendo 5.4 en las ecuaciones 5.5 y 5.6, se obtiene:

$$R(t) = \frac{N_0(t)}{N_0(t) + N_F(t)} \quad (5.7)$$

$$F(t) = \frac{N_F(t)}{N_0(t) + N_F(t)} \quad (5.8)$$

Ahora, hablando en términos de probabilidad en donde la suma de $R(t)$ y $F(t)$ da 1.0 (como se observa en 5.1), se despeja a la fiabilidad, obteniendo la ecuación 5.9.

$$R(t) = 1 - F(t) \quad (5.9)$$

En donde, al sustituir 5.6, se obtiene que la fiabilidad es igual a 1 menos la probabilidad de falla, como se muestra en 5.10.

$$R(t) = 1 - \frac{N_F(t)}{N} \quad (5.10)$$

Con el propósito de obtener la intensidad de fallas, se vincula esta con la fiabilidad, visualizándola en un instante de tiempo pequeño, por lo que se aplica la derivada de la ecuación 5.10, obteniendo a 5.11.

$$\frac{dR(t)}{dt} = \left(-\frac{1}{N}\right) \frac{dN_F(t)}{dt} \quad (5.11)$$

En donde $\frac{dN_F(t)}{dt}$ representa a la **tasa instantánea de fallas**, la cual es un indicador instantáneo de falla, que al definir en términos de la fiabilidad se obtiene la ecuación 5.12.

$$\frac{dN_F(t)}{dt} = \frac{(-N)dR(t)}{dt} \quad (5.12)$$

Si la tasa instantánea de fallas se divide entre el número de elementos operando en ese instante de tiempo $N_0(t)$, se obtiene el cambio en una diferencia de tiempo del número de elementos que presentan fallas, en relación con el universo de elementos en el tiempo t , que corresponde al universo de elementos operativos. Este concepto se denomina como **función de tasa de fallas** ($z(t)$), que es la razón de cambio de las fallas a lo largo del tiempo con respecto al universo (5.13).

$$z(t) = \frac{1}{N_0(t)} \frac{dN_F(t)}{dt} \quad (5.13)$$

Que al definir en términos de la fiabilidad, se obtiene la ecuación 5.14.

$$z(t) = \frac{1}{N_0(t)} \left(-\frac{NdR(t)}{dt} \right) = -\frac{\frac{dR(t)}{dt}}{R(t)} \quad (5.14)$$

Ahora presentando a $z(t)$ como una función diferencial, se obtiene:

$$\frac{dR(t)}{dt} = -z(t)R(t) \quad (5.15)$$

Resolviendo la ecuación diferencial 5.15, se obtiene a 5.16.

$$R(t) = e^{-\int z(t)dt} \quad (5.16)$$

Por último, considerando que la función de tasa de fallas $z(t)$ en los componentes electrónicos es constante durante su tiempo de vida útil (como se muestra en la figura 5.9), se sustituye por λ , obteniendo así a la función de distribución exponencial empleando la intensidad de fallas, con la cual se calcula la fiabilidad de un elemento del sistema para cualquier momento en el tiempo, como se presenta en la **ley de falla exponencial** (5.17), en donde la fiabilidad varía exponencialmente en función del tiempo.

$$R(t) = e^{-\lambda t} \quad (5.17)$$

Debido a su comportamiento constante, la ley de falla exponencial es por mucho la relación más utilizada entre fiabilidad y el tiempo para el análisis de la fiabilidad de los componentes electrónicos [39].

Es importante mencionar que en 5.17 únicamente se obtiene la fiabilidad de un solo elemento del sistema, el cual está compuesto por múltiples elementos y cada uno de ellos tendrá un valor de λ asociado, por lo que se deberá de realizar un análisis de la estructura del sistema que permita aproximar su nivel de fiabilidad. Para esto, se utilizan diferentes técnicas analíticas, destacando los modelos combinatorios y los modelos de Markov.

5.3.1 Modelos combinatorios

El principio de los modelos de análisis combinatorios para aproximar la fiabilidad de un sistema consiste en asumir que las fallas de componentes individuales son mutuamente independientes [36], empleando técnicas probabilísticas para enumerar los diferentes modos en los que el sistema puede mantener su estado funcional. Existen diferentes modelos combinatorios, de los cuales los más utilizados son los diagramas de bloques de fiabilidad (RBD, por sus siglas en inglés) y los árboles de fallas.

Diagramas de bloques de fiabilidad (RBD)

Consiste en una representación abstracta del sistema empleando un diagrama a bloques, en donde los bloques representan a los elementos del sistema y las conexiones entre ellos define su dependencia funcional. El análisis consiste en conectar a los componentes en serie y en paralelo en función de la necesidad del componente para que el sistema opere correctamente.

Árbol de fallas

Los árboles de fallas también emplean una vista abstracta del sistema, en donde la dependencia funcional de los componentes se representa con compuertas booleanas, cuyas entradas serán los componentes del sistema. De este

modo, la falla de un componente se representa con el valor **TRUE** en el nodo correspondiente. El análisis de fiabilidad se realiza mediante las ecuaciones booleanas del sistema.

5.3.2 Modelos de Markov

También conocidos como cadenas de Markov, son una clase especial de los procesos estocásticos [42] que representa gráficamente a un sistema a partir de sus estados en todos los instantes de tiempo, en donde la transición de un estado a otro es determinada solo por el estado presente en un momento dado y no por el estado previo [36], y es caracterizada por las probabilidades de mantenerse en un estado o pasar a otro, en un tiempo determinado.

Los modelos de Markov analizan los eventos en un intervalo de tiempo muy pequeño ($\Delta t \rightarrow 0$). A partir de eso, se aprovecha que la función de distribución exponencial puede expresarse en series de Taylor (5.18) para representar a la fiabilidad mediante una expresión compacta (5.19).

$$R(t) = e^{-\lambda t} = 1 + (-\lambda \Delta t) + \frac{(-\lambda \Delta t)^2}{2!} + \frac{(-\lambda \Delta t)^3}{3!} + \dots \quad (5.18)$$

$$R(t) = e^{-\lambda t} \approx 1 - \lambda \Delta t \quad (5.19)$$

De esta manera, en un sistema con un solo elemento con probabilidad de falla en el tiempo t , la probabilidad de transición entre estados ($P_i(t)$) se expresa con 5.20, en donde λ representa a la intensidad de fallas.

$$P_i(t) = 1 - \lambda \Delta t \quad (5.20)$$

Y al ser un proceso probabilístico, la suma de las probabilidades de que el sistema se encuentre en un estado determinado es igual a uno, como se muestra en 5.21, lo que significa que el sistema siempre se encontrará en solo uno de los n estados en el tiempo t .

$$P_0(t) + P_1(t) + P_2(t) + \dots + P_{n-1}(t) = 1 \quad (5.21)$$

Lo anterior se presenta en la figura 5.10, en donde se tiene un sistema con un solo componente, siendo S_0 el estado funcional inicial y S_1 el estado en el cual el sistema se ubicará después de la ocurrencia de una falla en su componente.

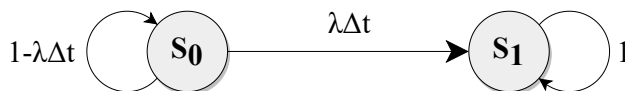


Figura 5.10: Cadena de Markov de un sistema con un solo componente

De la figura 5.10, la probabilidad de que el sistema se encuentre en el estado S_0 en el tiempo t_0 es igual a 1, y una vez que el componente presente una falla, el sistema pasará al estado S_1 , del cual nunca saldrá, lo que se denomina como un sistema sin capacidad de recuperación.

Aplicando las leyes del modelo de Markov y partiendo de que el sistema se encuentra en un estado funcional (estado S_0), se puede obtener la probabilidad de que el sistema continúe funcionando correctamente en el tiempo $t + \Delta t$ (5.22), así como la probabilidad de que el sistema presente una falla y pase al estado de no operación (estado S_1) en el tiempo $t + \Delta t$ (5.23).

$$P_0(t + \Delta t) = (1 - \lambda\Delta t)P_0(t) \quad (5.22)$$

$$P_1(t + \Delta t) = \lambda\Delta tP_0(t) + P_1(t) \quad (5.23)$$

Por otro lado, considerando que el sistema tiene la capacidad de recuperarse del estado con falla S_1 para regresar al estado funcional S_0 (véase la figura 5.11), aparece la intensidad de reparación del componente, representada por μ .

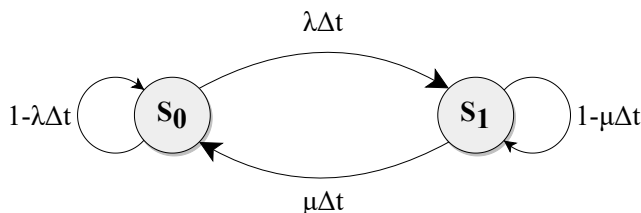


Figura 5.11: Cadena de Markov de un sistema con un solo componente con reparación

El análisis de la fiabilidad a partir de los modelos de Markov tiene mayor utilidad al considerar que un sistema tolerante a fallas va a estar construido por múltiples elementos redundantes que generan un comportamiento dinámico, en donde el sistema se comporta de una manera a partir de la presencia o no de fallas, por lo que utilizar los modelos combinatorios se vuelve una tarea altamente compleja.

5.4 Tolerancia a fallas en FPGA

La idea fundamental de la tolerancia a fallas es la redundancia, que consiste en replicar componentes, módulos o funciones del sistema, de modo que sirvan como soporte para incrementar el nivel de fiabilidad, aun con la presencia de fallas en uno o varios de ellos, o bien, sirvan como elementos de repuesto para continuar desempeñando las tareas que un elemento dañado ya no puede cumplir. Lo anterior se resume en diseñar un sistema con componentes redundantes que no serían necesarios en un ambiente libre de fallas. Si bien, el método más “sencillo” para construir un sistema espacial es utilizar componentes de grado espacial, estos presentan costos extremadamente altos¹⁴ y poca disponibilidad en comparación con los componentes COTS (como se vio en el capítulo 3), por lo que, a pesar de que la aplicación de técnicas de tolerancia a fallas aumenta considerablemente la complejidad durante el proceso de diseño, representa una opción viable para aproximar el nivel de fiabilidad de un sistema construido con componentes endurecidos a la radiación, usando componentes de bajo costo que poseen un nivel de fiabilidad estándar [44].

La redundancia se puede aplicar de diferentes formas, dependiendo del tipo de elementos que son replicados, como se presenta en la figura 5.12.

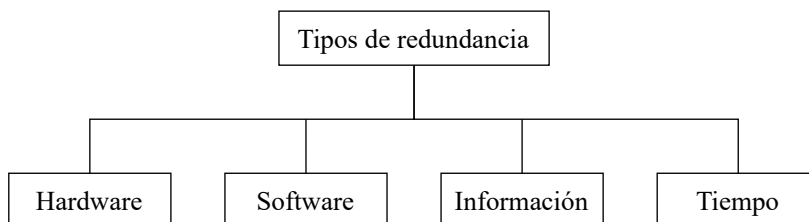


Figura 5.12: Tipos de redundancia aplicada en esquemas de tolerancia a fallas

En este proyecto se busca implementar técnicas de tolerancia a fallas en un SCMI, implementado en un FPGA que, por su naturaleza, presenta características de configuración de su estructura que son favorables para la construcción de

¹⁴Como referencia, el microcontrolador MSP430FR5969 de *Texas Instruments* (TI) de calidad estándar tiene un precio de USD 3.974, mientras que su versión endurecida a la radiación tiene un costo de USD 2,544.696, lo que representa una relación de precio aproximada de 640 [43].

sistemas redundantes. Para esto, existen diversos esquemas de redundancia, pero únicamente se describirán aquellos identificados en el estado del arte (véase más adelante en el capítulo 6) como los utilizados con mayor frecuencia en este tipo de sistemas.

5.4.1 Redundancia modular triple (TMR)

La redundancia modular triple (TMR, por sus siglas en inglés) es un método de redundancia pasiva en hardware en donde se tienen componentes triplicados ejecutando las mismas tareas en paralelo, cuyas salidas son comparadas por un dictaminador de mayoría (véase la figura 5.13), encargado de seleccionar la salida que cumpla con el principio de mayoría. De esta forma, se obtiene una estructura en donde, si un elemento falla, se seleccionará la salida de los dos módulos restantes, enmascarando la falla y previniendo que se propague hacia otros componentes del sistema, permitiendo que este opere normalmente y mantenga su nivel de fiabilidad. Es importante mencionar que una desventaja de la TMR es que, por su estructura, solo permite enmascarar fallas sin identificar el módulo con falla.

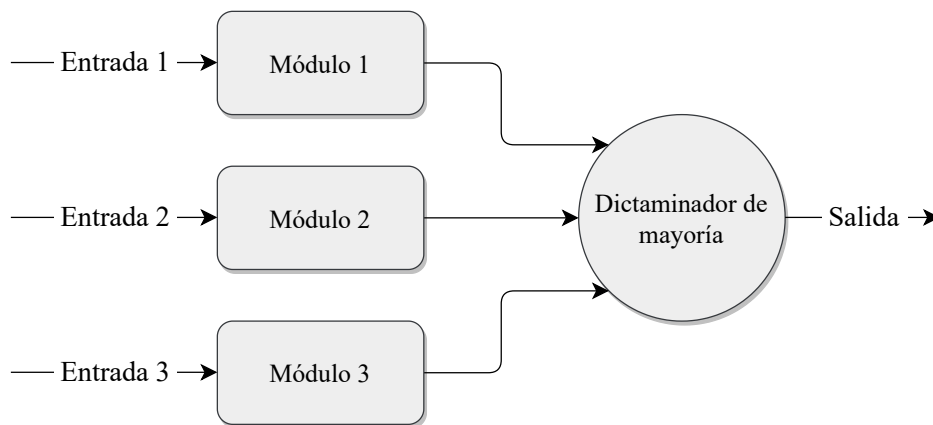


Figura 5.13: Concepto básico de la TMR

La TMR es el modo básico de los sistemas mayoritarios, incrementando el nivel de fiabilidad al aumentar la cantidad de componentes replicados. Sin embargo, a medida que aumenta el nivel de redundancia, se reduce la eficiencia del método [38], además de que aumenta considerablemente los recursos necesarios para su implementación y eleva la complejidad en el diseño del dictaminador de mayoría, aumentando su sensibilidad a fallas.

Para el análisis de fiabilidad de un sistema con TMR se asume que el dictaminador de mayoría es perfecto, por lo que el sistema operará correctamente mientras al menos dos módulos funcionen correctamente. Este comportamiento se representa en 5.24, en donde R_1 , R_2 y R_3 representan a la fiabilidad de cada uno de los módulos.

$$R_{TMR} = R_1R_2R_3 + (1 - R_1)R_2R_3 + R_1(1 - R_2)R_3 + R_1R_2(1 - R_3) \quad (5.24)$$

Posteriormente, al considerar que el nivel de fiabilidad es el mismo para todos los módulos replicados ($R_M = R_1 = R_2 = R_3$), la fiabilidad se reduce a 5.25.

$$R_{TMR} = R_M^3 + 3(1 - R_M)R_M^2 = 3R_M^2 - 2R_M^3 \quad (5.25)$$

Posteriormente, al asumir que la fiabilidad de un sistema con TMR varía en función del tiempo y, de acuerdo con la función de distribución exponencial (5.17), se obtiene 5.26.

$$R_{TMR}(t) = 3e^{-2\lambda t} - 2e^{-3\lambda t} \quad (5.26)$$

Ahora, considerando un sistema real en donde el dictaminador de mayoría no es perfecto, este se encuentra conectado en serie con el sistema, dando como resultado la expresión de fiabilidad presentada en 5.27

$$R_{TMR}(t) = \left(3e^{-2\lambda t} - 2e^{-3\lambda t} \right) R_v \tag{5.27}$$

De esta manera, para que un sistema con TMR presente un nivel de fiabilidad alto, la fiabilidad del dictaminador de mayoría debe de tender a 1, siendo mayor que la fiabilidad de los módulos replicados, como se muestra en 5.28.

$$R_v > 3e^{-2\lambda t} - 2e^{-3\lambda t} \tag{5.28}$$

Es importante aclarar que, por su simplicidad en comparación con los módulos redundantes, el dictaminador de mayoría presenta una probabilidad de fallas mucho menor. Sin embargo, si presenta una falla el esquema completo de TMR también fallará, lo que significa que es un punto común de falla, por lo que en algunas aplicaciones es importante considerar la aplicación de otros esquemas de tolerancia a fallas además de la TMR, como, por ejemplo, la triplicación del dictaminador, entre otros.

Por otro lado, realizando el análisis de la fiabilidad de un sistema con TMR empleando modelos de Markov, se obtiene el esquema de la figura 5.14, en donde el sistema comienza operando en el estado perfecto S_0 con los tres módulos operando correctamente (111). En el momento en el que un módulo falla, el sistema pasa a cualquiera de los estados S_1 (110), S_2 (101) y S_3 (011), manteniendo el correcto funcionamiento del sistema al tener dos módulos (la mayoría) operando correctamente. Posteriormente y considerando que el sistema no tiene capacidad de recuperación para ningún estado, permanecerá operando hasta que se presente una falla en cualquiera de los otros módulos, pasando a los estados S_4 (100), S_5 (001) o S_6 (010), en los cuales el sistema pierde la capacidad de operar correctamente. Por último, el módulo restante puede presentar otra falla, lo que colocará al sistema en el estado S_7 (000), del cual ya no saldrá nunca, a menos de que se apliquen tareas de recuperación.

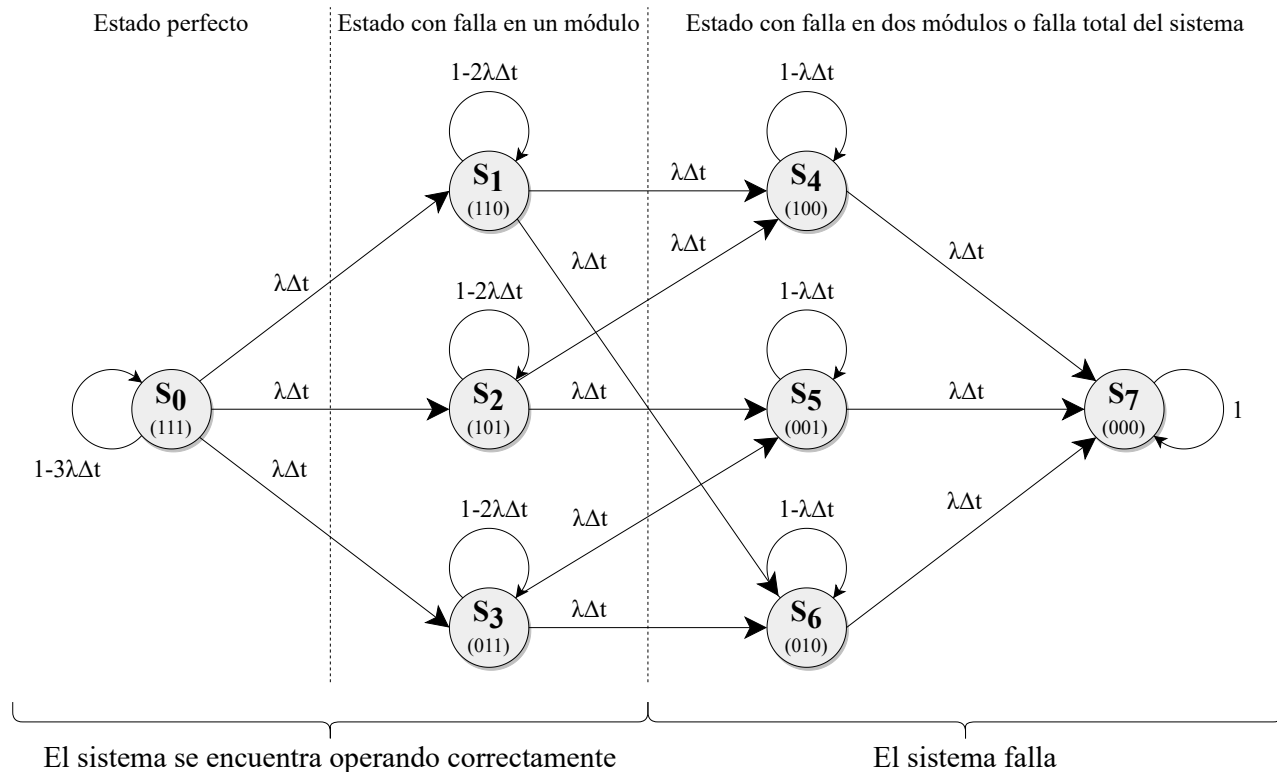


Figura 5.14: Modelo de Markov de un sistema con TMR

La transición entre los estados de operación puede representarse a través de un modelo reducido más sencillo de analizar, en donde únicamente se tienen tres estados: el estado perfecto (3), el estado con falla en un solo módulo (2) y el sistema en estado de falla (F), como se muestra en la figura 5.15.

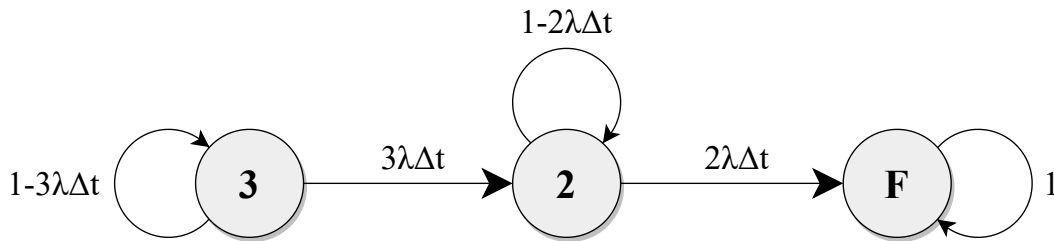


Figura 5.15: Modelo de Markov reducido de un sistema con TMR

De donde, el sistema de ecuaciones resultante para la probabilidad de que el sistema permanezca en cada uno de los estados es:

$$\begin{aligned}
 P_3(t + \Delta t) &= (1 - 3\lambda\Delta t)P_3(t) \\
 P_2(t + \Delta t) &= 3\lambda\Delta tP_3(t) + (1 - 2\lambda\Delta t)P_2(t) \\
 P_F(t + \Delta t) &= 2\lambda\Delta tP_2(t) + P_F(t)
 \end{aligned} \tag{5.29}$$

Arreglando algebraicamente al sistema de ecuaciones 5.29:

$$\begin{aligned}
 \frac{P_3(t + \Delta t) - P_3(t)}{\Delta t} &= -3\lambda P_3(t) \\
 \frac{P_2(t + \Delta t) - P_2(t)}{\Delta t} &= 3\lambda P_3(t) - 2\lambda P_2(t) \\
 \frac{P_F(t + \Delta t) - P_F(t)}{\Delta t} &= 2\lambda P_2(t)
 \end{aligned} \tag{5.30}$$

Considerando que $\Delta t \rightarrow 0$, las ecuaciones diferenciales resultantes están dadas por:

$$\begin{aligned}
 \frac{dP_3(t)}{dt} &= -3\lambda P_3(t) \\
 \frac{dP_2(t)}{dt} &= 3\lambda P_3(t) - 2\lambda P_2(t) \\
 \frac{dP_F(t)}{dt} &= 2\lambda P_2(t)
 \end{aligned} \tag{5.31}$$

Ahora, resolviendo las ecuaciones diferenciales del sistema 5.31, se obtienen:

$$\begin{aligned}
 P_3(t) &= e^{-3\lambda t} \\
 P_2(t) &= 3e^{-2\lambda t} - 3e^{-3\lambda t} \\
 P_F(t) &= 1 - 3e^{-2\lambda t} + 2e^{-3\lambda t}
 \end{aligned} \tag{5.32}$$

Y, por último, obteniendo la fiabilidad $R(t)$ (5.33) y la probabilidad de falla $F(t)$ (5.34), se llega a la misma expresión de fiabilidad con respecto a la solución por método combinatorio (5.26).

$$\begin{aligned} R(t) &= P_3(t) + P_2(t) \\ &= 3e^{-2\lambda t} - 2e^{-3\lambda t} \end{aligned} \quad (5.33)$$

$$\begin{aligned} F(t) &= P_F(t) \\ &= 1 - P_3(t) - P_2(t) \\ &= 1 - 3e^{-2\lambda t} + 2e^{-3\lambda t} \end{aligned} \quad (5.34)$$

5.4.2 Reconfiguración global

Es el modo de recuperación más simple, en donde, sin la necesidad de detectar o ubicar una falla en el sistema, este se reinicia periódicamente de manera global, eliminando la presencia de posibles fallas al sobrescribir los datos de la memoria de configuración para continuar a partir de su estado inicial (antes de la ocurrencia de fallas).

El esquema de un sistema con reconfiguración global se puede analizar a partir de un modelo de Markov con n componentes con reparación, como se presenta en la figura 5.16, en donde (0) representa a los estados funcionales y (1) a los estados de falla del sistema. Sin embargo, este modelo representa que la transición hacia el estado 1 sucede con la falla de un solo elemento, pero el funcionamiento de un sistema es más complejo, requiriendo de un análisis más profundo de los estados de operación.

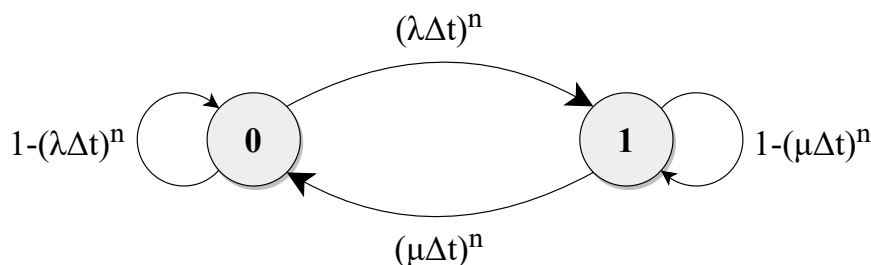


Figura 5.16: Modelo de Markov reducido de un sistema con n componentes con reparación

La reconfiguración global es el método más simple de reconfiguración, realizándose normalmente en sistemas con trabajo intermitente, aprovechando periodos de inactividad para reiniciar el sistema [38].

5.4.3 Memory scrubbing

El *memory scrubbing* consiste en la lectura y comparación periódica de la memoria de configuración del FPGA, de modo que si se detecta la presencia de un SEU, se elimina al sobrescribir la sección dañada de la memoria con el valor original, proceso conocido como reconfiguración parcial. El *memory scrubbing* es el método más efectivo de recuperación de fallas transitorias, y puede ser determinístico, en función del mecanismo dedicado a leer la memoria, leyendo y comparando todas las palabras de memoria; o probabilístico, al leer y verificar una palabra cuando esta sea accedida por el programa en ejecución [45].

El análisis de la fiabilidad del *memory scrubbing* depende de diversos factores, como por ejemplo, si la memoria se encuentra o no protegida por un esquema redundante o, por la cantidad de fallas que se presenten en una misma palabra, de modo que si alcanzan la suficiente cantidad, el EDAC empleado no podrá detectarlas o corregirlas.

Para el caso de que se utilice una memoria sin protección y asumiendo que el control, corrección y la interfaz del circuito son perfectas, la fiabilidad del sistema estará dada por 5.35 [45], en donde w representa el número de bits por palabra y M el número de palabras de la memoria.

$$R(t) = \left(e^{-\lambda wt} \right)^M \quad (5.35)$$

Por otro lado, cuando la memoria se encuentra protegida por un código de detección de error doble y corrección de un solo error (SEC-DED, por sus siglas en inglés), el *memory scrubbing* se puede modelar mediante el diagrama de transición de estados de la figura 5.17, en donde c representa el número de bits verificados por cada palabra y existen tres estados: cuando el sistema se encuentra sin fallas (0), cuando sucede una falla que puede ser corregida por el SEC-DED (1), y cuando se presentan el número de fallas suficientes para que este no pueda detectarlas o corregirlas (2), siendo el estado de no operación y sin capacidad de recuperarse.

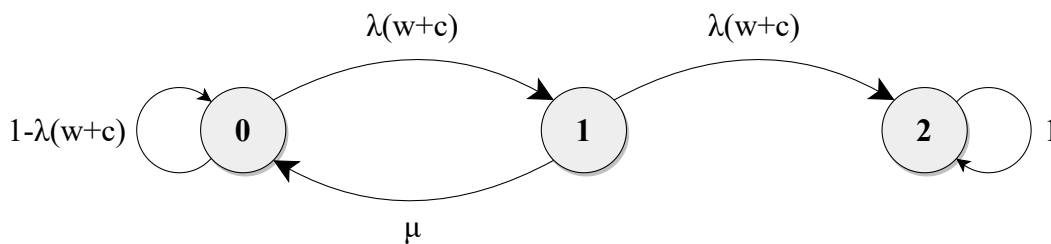


Figura 5.17: Modelo de Markov de un sistema con *memory scrubbing* probabilístico empleando una memoria con SEC-DED [45]

Cuya función de fiabilidad está dada por el producto del nivel de fiabilidad de todas las palabras (5.36), en donde $r(t)$ representa a la fiabilidad de una sola palabra [45].

$$R(t) = r^M(t) \quad (5.36)$$

A diferencia de la reconfiguración global, el *memory scrubbing* consume menor tiempo para la configuración de la memoria, por lo que es más viable su aplicación en sistemas con un flujo de trabajo continuo. Además, se puede combinar con la TMR, dando como resultado un incremento en la fiabilidad, por lo que es extensamente utilizado en misiones críticas [38].

6 ESTADO DEL ARTE DE SCMI TOLERANTES A FALLAS

Hasta este punto del proyecto se tienen las bases teóricas suficientes para comenzar con el proceso de diseño, pero antes de comenzar es importante plantear las preguntas ¿alguien más está desarrollando o ya desarrolló algo similar? y si ese es el caso ¿con qué problemáticas se ha encontrado y cómo las ha resuelto?. Para responder lo anterior se debe de analizar el estado del arte, que consiste en la compilación de los resultados de una investigación, realizada con el objetivo de conocer el nivel de desarrollo actual de una determinada tecnología, así como los problemas presentes en su desarrollo y los métodos que se utilizan para resolverlos. De esta manera, los resultados observados en el estado del arte permiten una definición de objetivos y alcances más certeros, dando un buen punto de partida para el desarrollo de un proyecto. En este capítulo se presentan los resultados del estado del arte, comenzando con las áreas de oportunidad presentes en el desarrollo de SCMI tolerantes a fallas para nanosatélites *CubeSat*, seguido de la justificación para la implementación de técnicas de tolerancia a fallas en *CubeSat* y los resultados del análisis de la presencia de FPGA en SCMI comerciales y en desarrollo. Como siguiente punto, se presenta la tecnología empleada en los SCMI, así como los esquemas de mitigación y tolerancia a fallas aplicados en las arquitecturas de SCMI. Por último, se presentan las conclusiones del estado del arte.



6.1 Crecimiento y proyección en las misiones realizadas con CubeSat

El aumento en el desarrollo de nanosatélites surge de la tendencia en la comunidad científica por diseñar sistemas espaciales más pequeños, impulsados por la disminución de los costos y los tiempos de desarrollo necesarios para construir satélites artificiales. Desde 1998 hasta mayo de 2023 se han lanzado 2286 nanosatélites, de los cuales, más del 92% corresponden al estándar *CubeSat* (2105 en total), estimando más de 2080 lanzamientos de nanosatélites en los próximos seis años [46]. En la figura 6.1 se presenta este comportamiento, además del dominio de los nanosatélites del tipo *CubeSat*, destacando los *CubeSat* 3U y 6U.

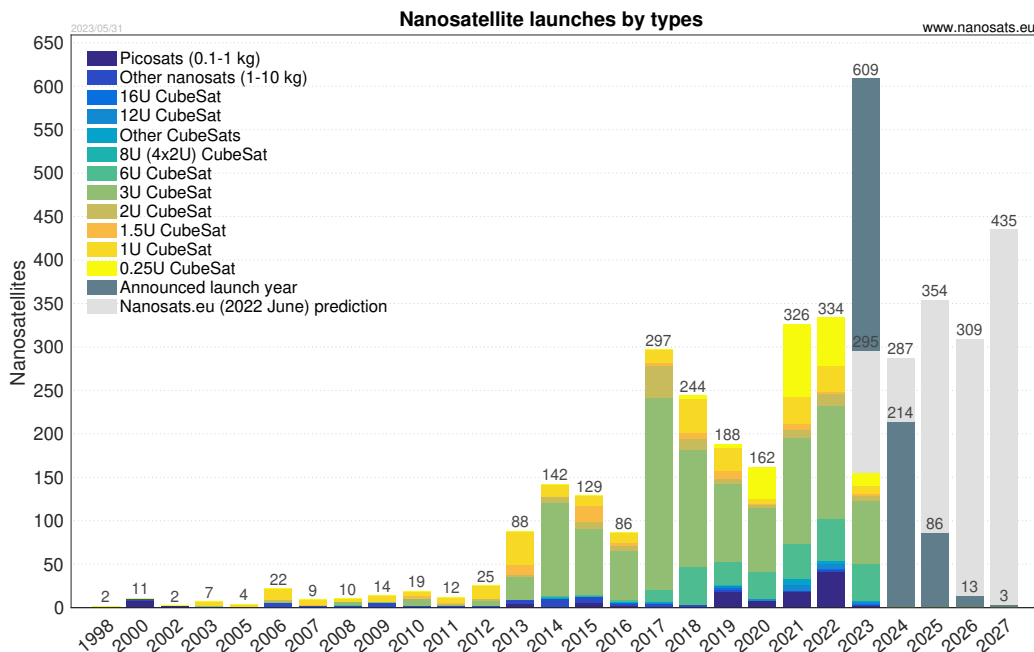


Figura 6.1: Tipos de nanosatélites lanzados por año [46]

6.2 Tolerancia a fallas en *CubeSat*

Como se estudió en el capítulo 5, el concepto de tolerancia a fallas surge de la necesidad de aumentar el nivel de fiabilidad de un sistema, buscando que este desempeñe correctamente sus funciones, aun con la presencia de fallas, durante un tiempo determinado. En aplicaciones espaciales, los sistemas están expuestos al medio ambiente de radiación espacial, en donde el flujo constante de partículas cargadas puede inducir fallas inmediatas en los componentes electrónicos o, como mínimo, reducir drásticamente su tiempo de vida útil. Los tipos de fallas con mayor probabilidad de ocurrencia dependen del tipo de órbita en donde se colocará el satélite. En las orbitas comúnmente empleadas para *CubeSat*, los subsistemas en donde se originan fallas en mayor medida después del lanzamiento son la computadora de a bordo (OBC, por sus siglas en inglés) o SCMI, el sistema de potencia eléctrica (EPS, por sus siglas en inglés) y el sistema de comunicaciones (COM, por sus siglas en inglés), debido a su alta densidad de componentes y arquitecturas similares, empleando múltiples microcontroladores y memorias [47].

Las capacidades tecnológicas actuales permiten construir microprocesadores COTS capaces de satisfacer fácilmente los requisitos de la mayoría de los SCMI para *CubeSat*, distribuyendo así los costos de ingeniería no recurrente en múltiples misiones y permitiendo la reutilización del software, factores importantes para mantener un mercado competitivo [48]. Sin embargo, la exposición de los nanosatélites al medio ambiente de radiación espacial perjudica directamente las capacidades que estos pueden alcanzar, limitando su uso a misiones de corta duración en LEO. Una manera de aumentar la vida útil de los nanosatélites es el uso de componentes EEE de grado espacial, pero el incremento en los costos que esto representa limita su uso en únicamente misiones críticas, en donde el cumplimiento de la misión tiene la mayor prioridad. Para resolver esto, es necesario buscar alternativas que permitan aproximar el nivel de fiabilidad al de los componentes de grado espacial, manteniendo los bajos costos de los componentes COTS. Para lograr esto se aplican técnicas de tolerancia a fallas, aumentando la complejidad durante el proceso de diseño, pero conservando el uso de componentes COTS y las ventajas que esto representa. Actualmente, los desarrolladores toman como prioridad la herencia de vuelo en la selección de componentes COTS, característica que disminuye la probabilidad de falla en el sistema al usar componentes que ya han sido probados con anterioridad, en conjunto con esquemas redundantes que permitan alcanzar los niveles de fiabilidad adecuados para desempeñar misiones en LEO.

6.3 Uso de FPGA en SCMI

El desarrollo de SCMI tolerantes a fallas consiste básicamente en implementar arquitecturas con elementos redundantes, capaces de desempeñar tareas replicadas o mantenerse como unidades de repuesto que puedan cubrir las necesidades del sistema que un módulo funcionando incorrectamente ya no puede. En conclusión, el nivel de complejidad es elevado, por lo que es necesario buscar alternativas que permitan diseñar SCMI que cumplan con los requisitos de fiabilidad.

Con la idea anterior en mente, la tendencia actual de la comunidad es migrar los sistemas para satélites hacia otras tecnologías, entre las cuales se ha identificado a los FPGA como una posible solución, destacando sobre los microcontroladores por su estructura configurable con la capacidad de desempeñar tareas que requieren paralelismo y presentar un mayor nivel de adaptabilidad para obtener un sistema escalable. Además, los FPGA disminuyen la complejidad en la integración de componentes [49], así como la posibilidad de recuperarse de fallas inducidas a través de procesos de reconfiguración, razón por la que ofrecen mayor potencial que los ASIC [50], evitando también los daños de fallas permanentes al usar variantes alternas de su configuración inicial. Sin embargo, la detección de fallas de grado fino no invasivo en FPGA es desafiante [51], siendo un área en proceso de investigación [52], [53].

Para demostrar lo anterior, se realizó una investigación del estado del arte en donde se analizó la presencia de FPGA en SCMI, dividido en dos partes: los SCMI comerciales, disponibles en el mercado de subsistemas espaciales; y los SCMI en desarrollo, identificados en publicaciones presentadas en congresos y eventos dedicados al desarrollo de sistemas espaciales.

6.3.1 SCMI comerciales

El objetivo del análisis de los SCMI comerciales es conocer cuáles son las características de los sistemas probados exitosamente en el entorno espacial, en donde la perspectiva comercial le da prioridad al diseño de SCMI de propósito general que, en términos técnicos, significa que sus sistemas cumplen con los requisitos necesarios para cumplir las misiones típicas para nanosatélites *CubeSat*. De este modo, el conocer las capacidades de los SCMI servirá como punto de partida para obtener las especificaciones de diseño necesarias para desarrollar un SCMI competitivo con el mercado de satélites pequeños.

Para lograr lo anterior, se realizó un estudio profundo de 82 SCMI comerciales (véase el apéndice A), a partir del cual se observó el tipo de tecnología empleada para el dispositivo de procesamiento central, destacando el uso de arquitecturas configurables como FPGA o SoC, como se presenta en la figura 6.2, sobre microcontroladores o unidades de procesamiento gráfico (GPU, por sus siglas en inglés). Es importante resaltar que una cuarta parte de los SCMI estudiados presentan más de un dispositivo programable, pero únicamente utilizan el de mayor desempeño como unidad principal de procesamiento (que es la unidad contabilizada en la figura 6.2), destinando tareas concretas del sistema al otro dispositivo, tales como configuración o supervisión. Otro punto importante es la relación entre el uso de FPGA y SoC, en donde los SoC se emplean con mayor frecuencia, gracias a que en el mercado actual estos dispositivos típicamente tienen embebida una arquitectura multi-núcleo o de sistema multi-procesador en chip (MPSoC, por sus siglas en inglés) y una zona configurable de FPGA, resultando en un dispositivo cuyo uso promete nuevas oportunidades gracias a sus beneficios de desempeño, consumo de potencia, masa y volumen, siendo características relevantes para la construcción de sistemas espaciales con mayores capacidades [50].

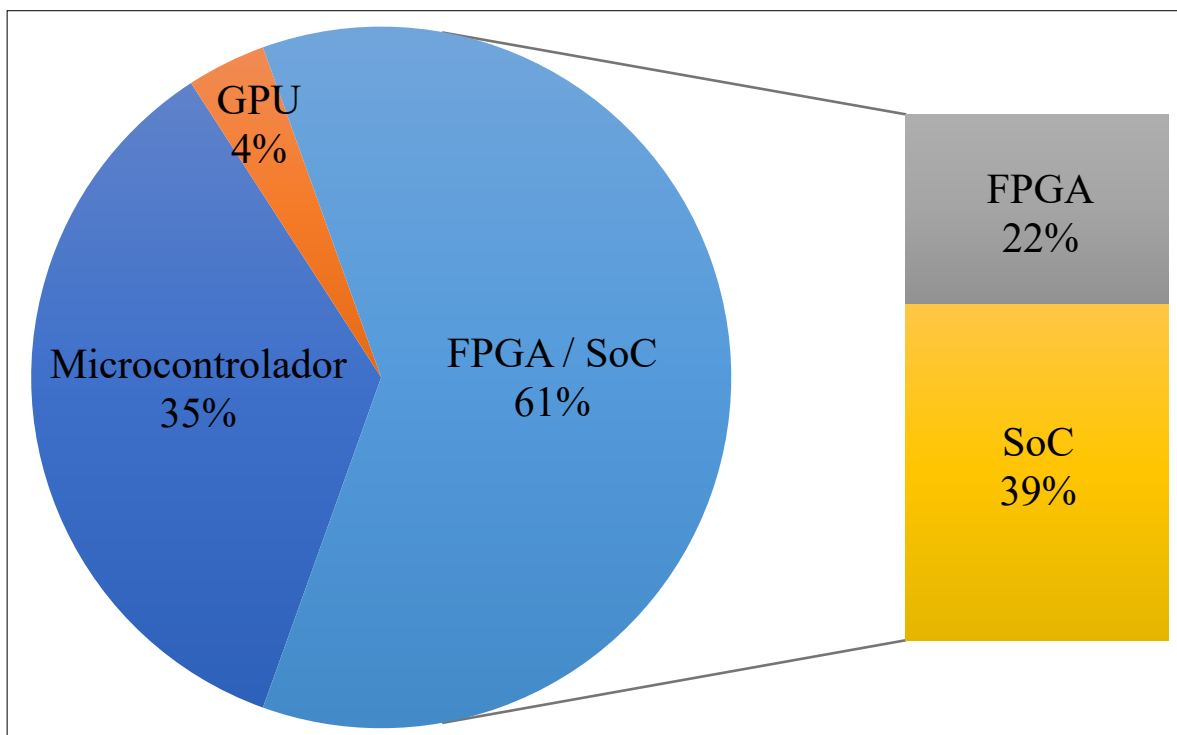


Figura 6.2: Dispositivo de procesamiento central de los SCMI comerciales, $n = 82$

En cuanto a los fabricantes de los dispositivos empleados en SCMI comerciales, en la figura 6.3 se observa claramente el amplio dominio de *Xilinx* y *Microsemi* (adquiridos por *AMD* en 2020 y por *Microchip* en 2018, respectivamente), en donde también se ofrecen arquitecturas MPSoC de *softcore* (como *MicroBlaze* por parte de *Xilinx* y el *LEON3FT* para *Microsemi*), que presentan herramientas de tolerancia a fallas, como la posibilidad de configurarse como núcleos redundantes (ya sea a través de TMR o *lockstep*). Los dispositivos empleados con mayor frecuencia son: por parte de *Xilinx*: las familias *Zynq* (10%), *Zynq UltraScale+* (10%), y *Virtex 5* (5%); y por parte de *Microsemi*: las familias *SmartFusion 2* (9%) y *PolarFire* (5%).

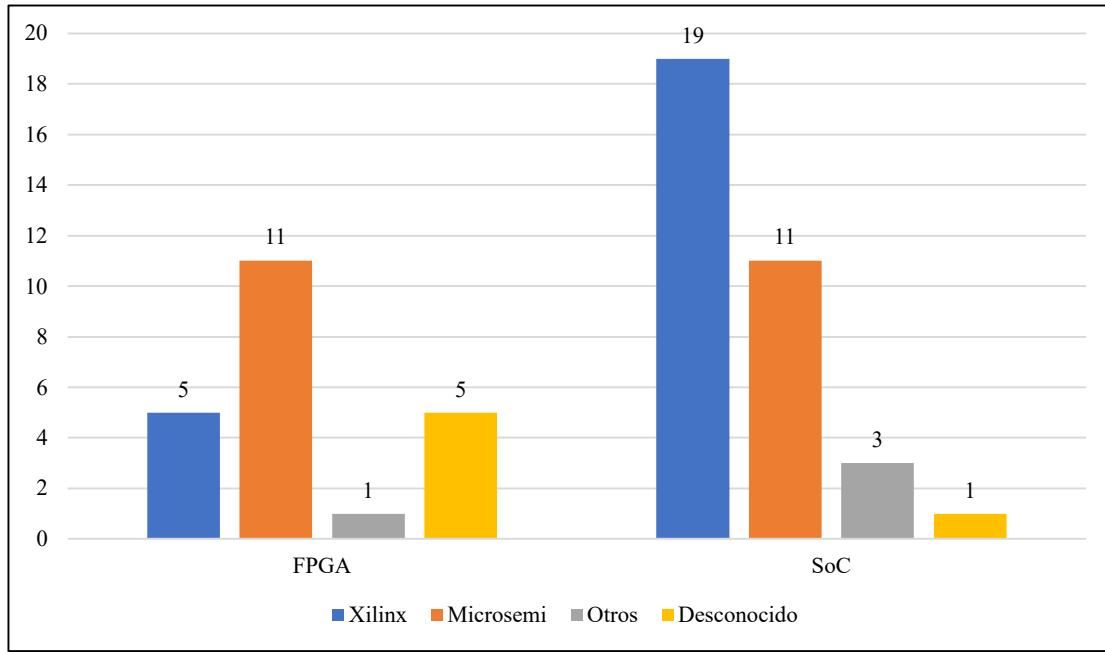
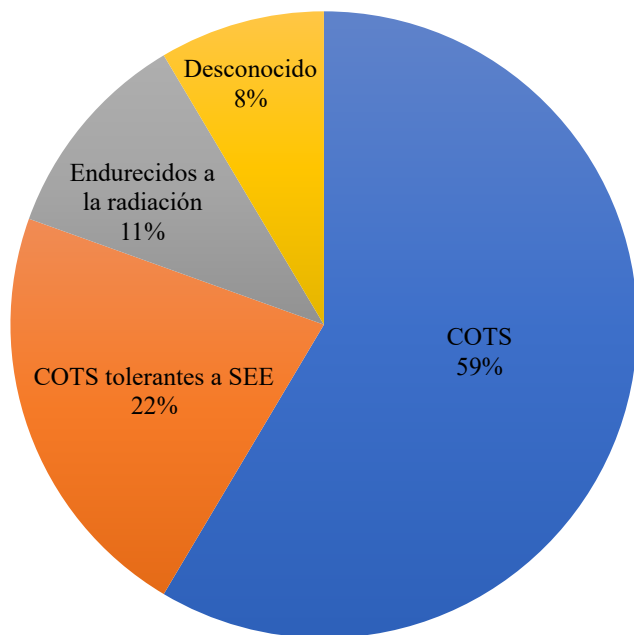
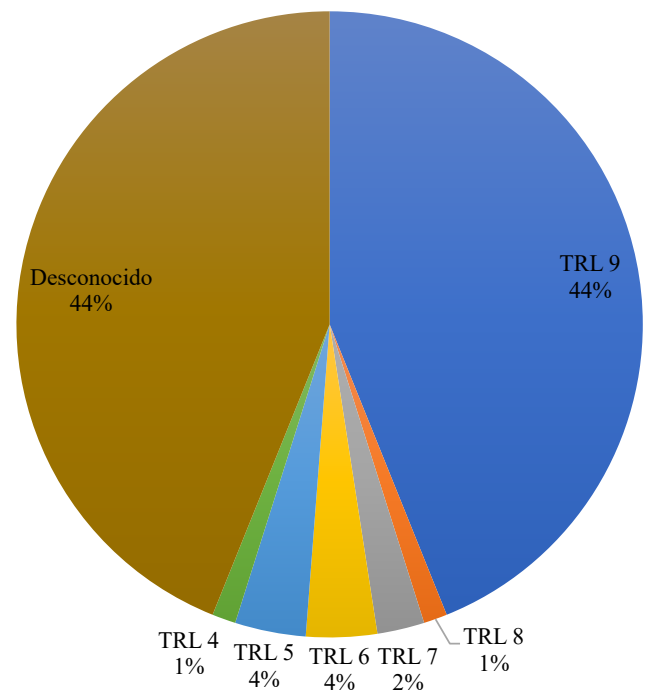


Figura 6.3: Dispositivos reconfigurables empleados en SCMI comerciales, $n = 82$

Como siguiente punto, en la figura 6.4a se observa la tendencia en el uso de componentes COTS para el desarrollo de SCMI comerciales, reflejándose en costos más accesibles en comparación con los componentes endurecidos a la radiación o de grado espacial. Además, un porcentaje relevante de los fabricantes informan que usan componentes COTS tolerantes a SEE, grado alcanzado a través de técnicas de tolerancia a fallas. Por otro lado, quitando a los SCMI que se encuentran en desarrollo o en espera de ser probados en una misión, la mayoría ya cuentan con herencia de vuelo, lo que los califica con un TRL 9, como se presenta en la figura 6.4b.



(a) Grado de los componentes en los SCMI comerciales



(b) TRL de los SCMI comerciales

Figura 6.4: Grado de los componentes y TRL de los SCMI comerciales, $n = 82$

6.3.2 SCMI en desarrollo

En este trabajo se denominan como SCMI en desarrollo a aquellos sistemas identificados en publicaciones académicas o congresos dedicados al desarrollo de sistemas espaciales, promoviendo la investigación científica y brindando alternativas para el desarrollo de SCMI. En la tabla 6.1 se presenta una compilación de diseños de SCMI publicados que, a diferencia de los SCMI comerciales, servirán para analizar las aplicaciones y el tipo de técnicas de tolerancia a fallas aplicadas para alcanzar los niveles de fiabilidad adecuados para desempeñar misiones en LEO.

Tabla 6.1: FPGA y SoC presentes en SCMI en desarrollo

Dispositivo	Aplicación	Técnicas de tolerancia a fallas	Categoría	Satélite	Año	Referencia
<i>Xilinx</i> Kintex KU3P UltraScale+ FPGA	OBC MPSoC tolerante a fallas	Reconfiguración parcial del FPGA. Maestro - supervisor multimaestro. Diagnóstico de bajo nivel. EDAC/ECC en memorias. Redundancia en memorias. Uso de IP.	COTS	CubeSat	2019	[49]
<i>Xilinx</i> Virtex-5 FPGA	Procesamiento de imágenes		COTS	CubeSat	2010	[54]
<i>Microsemi</i> RTG4 FPGA	OBC tolerante a fallas	EDAC en memorias y lógica interna.	QMLV Class-S	SmallSat	2016	[55]
<i>Xilinx</i> Spartan-6 LX45T FPGA	OBC secundaria	Arquitectura lockstep. Redundancia en módulos. Redundancia en tiempo.	COTS	CubeSat	2017	[56]
<i>Xilinx</i> Nexys 4 SoC	OBC tolerante a fallas	TMR. Memory scrubbing.	COTS	CubeSat	2020	[57]
<i>Xilinx</i> Zynq-7000 SoC	Procesamiento de imágenes	DPR. Memory scrubbing.	COTS	¿?	2019	[58]
<i>Xilinx</i> Artix-7 XC7A200T FPGA	Detector de eventos individuales	Reconfiguración parcial del FPGA. TMR. Memory scrubbing. Redundancia en memorias. ECC en memorias.	COTS	CubeSat	2017	[59]
<i>Microsemi</i> SmartFusion2 SoC						
<i>Xilinx</i> XQVR600 FPGA	OBC tolerante a fallas	Reconfiguración parcial del FPGA. TMR en procesadores. EDAC en memorias.	COTS	CubeSat	2013	[60]

De manera similar a los SCMI comerciales, los desarrolladores independientes optan por dispositivos COTS FPGA y SoC de *Xilinx* y *Microsemi*. Lo importante en esta parte es que, en comparación con los sistemas comerciales, estos diseños describen con mayor detalle los tipos de arquitecturas y técnicas de tolerancia a fallas con las que encontraron una opción alterna para aumentar el nivel de fiabilidad de los SCMI. En los SCMI en desarrollo sobresale el uso de módulos IP y esquemas de reconfiguración parcial dinámica (DPR, por sus siglas en inglés) del FPGA, dispositivos supervisores externos, memorias de configuración externas, algunos tipos de redundancia en hardware (como la TMR

o *lockstep*) y la implementación de técnicas EDAC y ECC en memorias, así como el *memory scrubbing*. Entre las aplicaciones se encuentran el procesamiento de datos y una notable tendencia en el desarrollo de SCMI tolerantes a fallas.

6.4 Estado tecnológico actual

El grado de desarrollo actual de la tecnología permite el cumplimiento de un amplio tipo de misiones con aplicaciones científicas y comerciales, empleando nanosatélites *CubeSat* [49]. Lo anterior, sumado con el interés en aumentar la duración de las misiones para *CubeSat*, genera la necesidad de incrementar las capacidades de desempeño del sistema, incluyendo el SCMI. En un principio, los SCMI empleaban microcontroladores de bajo desempeño, pero el planteamiento de misiones que requieren de sistemas más robustos ha llevado a la integración de otros dispositivos, como los FPGA y, eventualmente, la combinación de ambas tecnologías en un único dispositivo (SoC), capaz de cumplir una alta demanda de requisitos con un bajo consumo energético, reduciendo también la complejidad de integración, el volumen y la masa del sistema.

Actualmente, los SCMI desempeñan tareas de cómputo avanzadas empleando SoC y MPSoC comerciales, desarrollados por empresas que emplean altos grados de sofisticación, como el MPSoC *Zynq-UltraScale+* de *Xilinx*, siendo uno de los proveedores con una de las rutas trazadas más avanzada en el desarrollo de tecnología SoC [50]. Sin embargo, estos dispositivos requieren de un proceso de manufactura muy avanzado empleando arquitecturas muy pequeñas y de alta densidad de elementos, lo que los convierte en dispositivos altamente susceptibles a fallas inducidas por el medio ambiente de radiación espacial [49], razón por la cual, algunas empresas se encuentran desarrollando variantes de sus dispositivos endurecidas a la radiación, como *Xilinx* y *Microsemi*.

Recientemente, diferentes tecnologías de procesamiento embebidas endurecidas a la radiación han adquirido disponibilidad en el mercado, lo que ha permitido su uso en gran variedad de misiones espaciales y subsistemas, como el SCMI. Estos dispositivos han sido probados contra la radiación con al menos una resistencia a la TID de $50kRad$. [48]

Después de un análisis de los SCMI presentados anteriormente, se observa el dominio en el mercado de FPGA en aplicaciones espaciales por parte de *Xilinx* y *Microsemi*. Por parte de *Xilinx* se observan el FPGA *Spartan-3E*, el SoC *Zynq-7020* con dos núcleos ARM Cortex-A9 y el MPSoC *Zynq UltraScale+* con cuatro núcleos ARM Cortex-A53, dos núcleos ARM Cortex-R5F y una GPU ARM Mali-400MP2. En cuanto a dispositivos de *Microsemi*, se emplean los FPGA *PolarFire* y *ProASIC3*, y el SoC *SmartFusion2*, con un núcleo ARM Cortex-M3.

6.4.1 FPGA endurecidos a la radiación

En aplicaciones espaciales, los núcleos ARM, así como la estructura de los FPGA embebidos en SoC, han demostrado ser vulnerables a errores de software inducidos por la radiación [61], [62], lo que representa una mayor amenaza a la fiabilidad para los FPGA basados en SRAM, como los de *Xilinx* y *Altera* (adquirida por *Intel* en 2015) [50]. Con la tendencia del uso del FPGA en subsistemas para *CubeSat*, hablando específicamente del SCMI, los fabricantes líderes en el mercado de FPGA de grado espacial (*Xilinx* y *Microsemi*) han desarrollado diferentes familias de FPGA y SoC tolerantes a la radiación. En 2020, algunas de estas familias fueron probadas en ambientes de TID de $100kRad$ [48]. *Xilinx* ha lanzado al mercado múltiples generaciones de FPGA endurecidos a la radiación (como el *Virtex-5QV*) y herramientas de software para diseñar sistemas tolerantes a fallas (por ejemplo, la IP de mitigación de errores suaves (SEM por sus siglas en inglés), de *Xilinx*), que son utilizados en numerosos sistemas espaciales [50]. Sin embargo, es importante reiterar que el costo de los componentes endurecidos a la radiación es extremadamente alto, en comparación con componentes COTS.

6.4.2 Esquemas de mitigación y tolerancia a la radiación

Para evitar el aumento en los costos e incrementar la disponibilidad de componentes, los desarrolladores mantienen procesos de diseño empleando componentes COTS, los cuales no cuentan con el nivel de fiabilidad adecuado para desempeñar misiones en el espacio profundo o en LEO de larga duración. Esto genera la necesidad de buscar

alternativas para aumentar el nivel de fiabilidad de los componentes COTS. Algunos desarrolladores combinan el uso de los componentes COTS con componentes endurecidos a la radiación, en donde los dispositivos comerciales desempeñan tareas demandantes de procesamiento y procesos críticos, supervisados por los elementos endurecidos a la radiación (como temporizadores watchdog) [50]. Por otro lado, se mantienen los procesos de diseño empleando totalmente componentes COTS, implementando técnicas de mitigación y tolerancia a fallas que, dependiendo del ingenio del desarrollador y aprovechando las capacidades tecnológicas actuales, es posible alcanzar niveles adecuados de fiabilidad.

En esta investigación se identificaron aquellas técnicas más empleadas en SCMI, en donde se destaca el uso de códigos EDAC y ECC aplicado en diferentes tipos de memorias dependiendo de la información almacenada en cada una, como la memoria magnética de acceso aleatorio (MRAM, por sus siglas en inglés) para datos críticos y FLASH para almacenamiento de software; circuitos de protección autónoma contra SEL empleando limitadores de corriente de *latch-up* (LCL, por sus siglas en inglés); el *memory scrubbing*, que es un método clásico para proteger a las memorias de configuración en FPGA basados en SRAM, comúnmente apoyándose de una memoria externa con mayor resistencia a la radiación [50]. Por otro lado, en muchos sistemas se integran temporizadores watchdog externos, algunos de ellos más robustos para eliminar SEU y SEFI o escalados a microcontroladores supervisores que ejecutan tareas de diagnóstico y recuperación, aplicando una reconfiguración total o parcialmente el FPGA para aislar o eliminar fallas [49]. Por último, se aplican diferentes tipos de redundancia, como TMR en núcleos de procesador (a través de dispositivos MPSoC) y en memorias (variando el tipo de tecnología).

6.4.3 Arquitecturas MPSoC en FPGA

Los requisitos planteados para el diseño de SCMI han evolucionado las arquitecturas de computadora sencillas de un solo núcleo, a sistemas robustos combinando más de un microcontrolador con un FPGA, lo que ha llevado al uso de dispositivos SoC, que a su vez han evolucionado para implementar sistemas MPSoC, en donde se pueden alcanzar grandes niveles de tolerancia a fallas, pero considerando que puede aumentar considerablemente la complejidad en el proceso de diseño.

El software implementado para aplicar conceptos de tolerancia a fallas en sistemas MPSoC fue identificado como una herramienta prometedora en los primeros días de los microcontroladores [63], pero en ese tiempo era técnicamente inviable e ineficiente, hasta hace pocos años con el estado de desarrollo actual de la tecnología. La tecnología de semiconductores moderna permite superar estas limitantes. Investigaciones recientes ([64], [65]), muestran que las arquitecturas modernas MPSoC pueden, teóricamente, ser explotadas para alcanzar buenos niveles de tolerancia a fallas. Las características y capacidades del FPGA permiten implementar arquitecturas MPSoC, en adición a que la alta disponibilidad de SoC fabricados por diferentes proveedores aumenta la viabilidad para reproducir este tipo de sistemas con un bajo costo.

Entre las arquitecturas MPSoC localizadas en los SCMI analizados, se identificó que el uso de FPGA brinda una amplia gama de posibilidades, pero sin llegar a compararse con las de dispositivos SoC, en donde se tienen uno o varios procesadores embebidos (*hard processor*), operando conjuntamente con un FPGA. Además de esto, algunos fabricantes describen núcleos de procesador adicionales (*soft processor*), optando por utilizar algunos IP comerciales (como el *MicroBlaze* de *Xilinx*) encontrado en [49], [56], [59].

Entre las arquitecturas más interesantes encontradas en el estado del arte y considerando los objetivos del proyecto, se encuentra el trabajo realizado en [49], en donde se desarrolló una arquitectura MPSoC dentro de un FPGA de bajo consumo (*Xilinx Kintex*) empleando cuatro procesadores IP *MicroBlaze* aislados individualmente en diferentes compartimientos. El SCMI cuenta con un microcontrolador supervisor (MSP430FR5969) de TI, como watchdog y ejecutando tareas de debug y diagnóstico de bajo nivel en el MPSoC controlando el puerto correspondiente al estándar del grupo de acción conjunta de prueba (JTAG, por sus siglas en inglés) del FPGA con líneas y entradas y salidas de propósito general (GPIO, por sus siglas en inglés), permitiendo su reinicio por completo o en diferentes partes del MPSoC. Ambos dispositivos comparten un bus de interfaz periférico serial (SPI, por sus siglas en inglés) en modo multimaestro con una memoria de configuración para el FPGA, permitiendo su propia reconfiguración, en caso de una

falla, el supervisor también puede desempeñar esta tarea. Esta arquitectura evita la redundancia en hardware (como TMR), alcanzando la tolerancia a fallas a través de software, implementando un esquema de reconfiguración parcial o total del FPGA empleando variantes de configuración alternas. El software se divide en tres etapas: *coarse-grain lockstep* para el software de vuelo, en donde se detectan y corrigen fallas dentro de un compartimiento manteniendo un estado consistente del software a través de técnicas de corrección de errores hacia delante (FEC, por sus siglas en inglés) en puntos de control; la reparación y recuperación del MPSoC asegurando que se encuentren intactos y disponibles los compartimientos suficientes en todo momento; y, por último, un método de degradación agraciado, en donde, si las etapas anteriores no son suficientes para mantener la estabilidad del sistema, se va a mantener en un estado seguro, degradando sus capacidades de modo que siga cumpliendo sus objetivos. Además de estos procesos, el SCMI emplea IP de proveedores estándar para facilitar la integración de técnicas de tolerancia a fallas, como la SEM, también de *Xilinx*.

En [58] se implementa un esquema de DPR en el FPGA con dos objetivos: soportar la adaptación de funcionalidad e implementar mecanismos de mitigación a fallas en tiempo de ejecución, tal como re-ubicación de módulos o *memory scrubbing* en la memoria de reconfiguración.

En [59] se implementa una arquitectura reconfigurable para la detección de SEU, en donde se ejecutan tres procesadores IP *MicroBlaze* de *Xilinx*, cada uno con un acceso independiente a un bloque TMR de memorias para instrucciones y datos locales. Para obtener la tolerancia a fallas, cada tarea se ejecuta independientemente dos veces en diferentes procesadores con un ligero desfase para evitar la ocurrencia de fallas transitorias. Si se detecta una falla por SEU o SET, todos los procesadores entrarán en una fase de revisión sincronizada comparando sus resultados y, si es necesario, iniciar un núcleo de reparación (realizando lógica interna de reconfiguración por *scrubbing*) para tareas de recuperación. Por otro lado, los bancos de memorias TMR están protegidos a través de *memory scrubbing*, empleando ECC.

6.5 Conclusiones

Se observa un aumento en el uso de nanosatélites *CubeSat*, abarcando un mayor número de misiones y, derivado de esto, un incremento en los requisitos de desempeño y procesamiento para *CubeSat*, reflejándose directamente en el SCMI. Del mismo modo, el desarrollo tecnológico, así como la mayor disponibilidad de componentes más robustos, ha impulsado el cumplimiento de tareas más avanzadas en SCMI para *CubeSat*, lo que ha llevado a integrar diferentes tecnologías para dos objetivos principales: cubrir el desempeño demandado por misiones más complejas y aumentar la vida útil del sistema o, en otras palabras, incrementar los niveles de fiabilidad a través de técnicas de tolerancia a fallas. Para estos propósitos, el FPGA brinda buenas características, abriendo las puertas también al diseño de arquitecturas más robustas, como lo son las MPSoC. Comercialmente, los fabricantes líderes de FPGA brindan soluciones de bajo consumo y alto desempeño en SoC con uno o varios procesadores embebidos o, en su caso, módulos IP para simular o aumentar las capacidades de funcionalidad.

Las aplicaciones principales en donde se implementan FPGA son en tareas de adquisición y procesamiento de datos (computadoras y cargas útiles), en sistemas de comunicación (de radiofrecuencia (RF) y radio definido por software (SDR, por sus siglas en inglés)) y, siendo el tema de interés en este trabajo, SCMI tolerantes a fallas.

Parte III

DISEÑO A NIVEL DE SISTEMA

7 DETERMINACIÓN DE LA MISIÓN

Todo sistema debe de ser desarrollado con base en los requisitos de la misión, de modo que se cumplan las especificaciones necesarias para soportar las condiciones ambientales. En este capítulo, se realiza un análisis de las órbitas en donde se utilizan regularmente los nanosatélites *CubeSat*, sirviendo como etapa preliminar para el desarrollo del concepto de diseño del SCMI.



Recordando los objetivos de este proyecto, se está realizando el diseño de un SCMI de propósito general para nanosatélites *CubeSat*, lo que significa que este debe de satisfacer las especificaciones necesarias para operar correctamente en el ambiente espacial. Para esto, se analizaron las órbitas en las que se colocan regularmente a los nanosatélites (véase la figura 7.1), en donde se observa claramente el uso de LEO¹⁵, destacando a la órbita de la estación espacial internacional (ISS, por sus siglas en inglés) como la más utilizada para los nanosatélites y, por ende, para los *CubeSat*. La razón de esto es que, como se mencionó en el capítulo 3, uno de los factores de éxito del *CubeSat* es su factor de forma, que al estar estandarizado permite usar diferentes plataformas de lanzamiento utilizadas frecuentemente por la ISS para desplegar múltiples *CubeSat* en la órbita deseada.

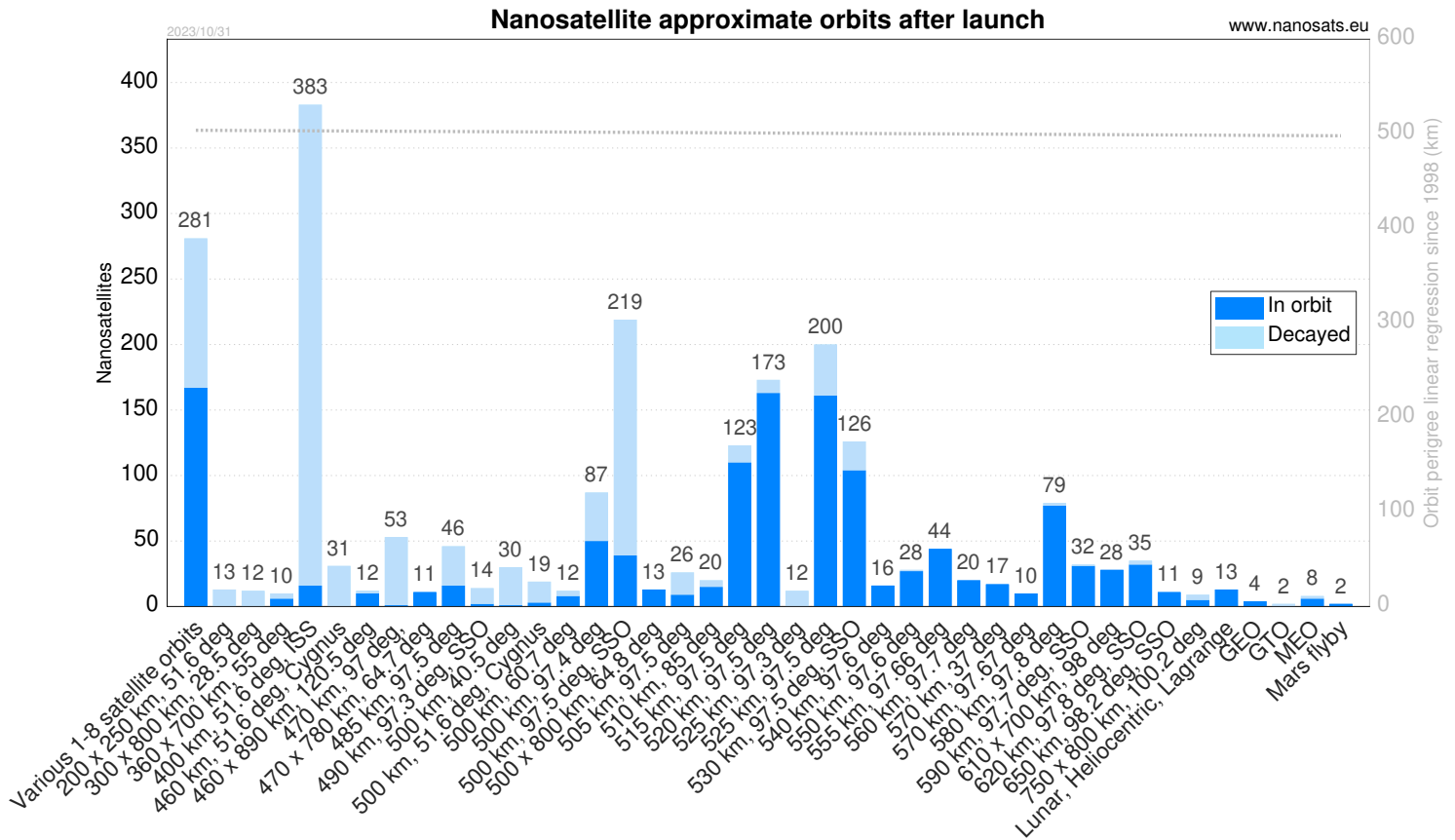


Figura 7.1: Órbitas aproximadas de los nanosatélites después de su lanzamiento [46]

¹⁵Órbita terrestre localizada entre la atmósfera y el cinturón de radiación de Van Allen interno (típicamente ubicada entre los 200 y 2000 km de altura sobre la superficie terrestre).

Por otro lado, el estado de arte de los SCMI comerciales indica que los fabricantes garantizan frecuentemente una operación correcta de sus sistemas para cumplir misiones de hasta 5 años de duración. De esta manera, se definen los parámetros de la misión (véase la tabla 7.1) considerando el tiempo máximo que se ofrece regularmente, esto con el objetivo de analizar las condiciones que permitan diseñar un SCMI que teóricamente sea competitivo con el mercado actual, tomando como referencia a la órbita de la ISS

Tabla 7.1: Parámetros de la misión [66]

Parámetro	Valor			Unidades
	Mínimo	Típico	Máximo	
Duración de la misión			5	años
Altura del perigeo		418		km
Altura del apogeo		419		km
Inclinación orbital		51.64		°
Longitud del nodo ascendente		281.63		°
Argumento del perigeo		324.42		°
Anomalía media		152.28		°

7.1 Estimación del medio ambiente de radiación

Para estimar el medio ambiente de radiación de la órbita seleccionada se utiliza el sistema de información del entorno espacial (SPENVIS, por sus siglas en inglés) de la ESA, que es una interfaz en línea para modelar el entorno espacial y sus efectos; incluyendo rayos cósmicos, los cinturones de radiación natural, las partículas de energía solar, plasmas, gases y las micropartículas de una órbita dada [67].

Comenzando con el análisis, se plantea un escenario hipotético de una misión de 5 años de duración que comenzará al inicio del 2026, definiendo una órbita con los parámetros de la tabla 7.1. Para caracterizar el medio ambiente determinado y debido a los límites de cálculo de SPENVIS la misión se dividirá en 5 segmentos para su análisis (véase la figura 7.2), cuyos parámetros se presentan en la tabla 7.2.

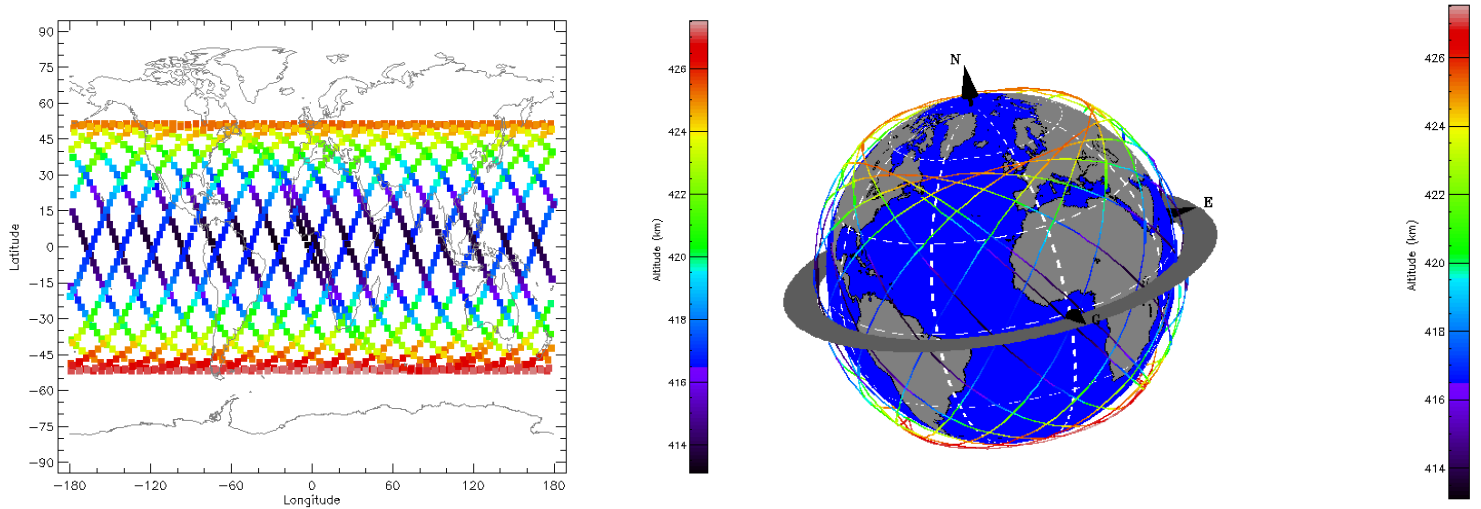
Mission overview	
Orbit around: Earth	
Number of mission segments: 5	
Mission start: 01/01/2025 00:00:00	
Mission end: 31/12/2029 00:00:00	
Mission duration: 1825.00 days (5.00 years)	
Satellite axis: velocity vector	

Figura 7.2: Vista preliminar de la misión definida en SPENVIS [46]

Tabla 7.2: Segmentos de la misión

Segmento	Descripción	Duración	Número de órbitas	Inicio del segmento	Fin del segmento
Segmento 1	1 ^{er} día	1 día	15.52	01/01/2025 00:00	02/01/2025 00:00
Segmento 2	1 ^{er} mes	30 días	465.55	02/01/2025 00:00	01/02/2025 00:00
Segmento 3	20 ^{mo} mes	30 días	465.55	01/08/2026 00:00	31/08/2026 00:00
Segmento 4	40 ^{mo} mes	30 días	465.55	01/04/2028 00:00	01/05/2028 00:00
Segmento 5	60 ^{mo} mes	29 días	450.03	01/12/2029 00:00	30/12/2029 00:00

Con estos parámetros, el satélite completará un promedio de 15.52 órbitas al día, con una altura que oscilará entre los 413 y 427 km sobre la superficie terrestre, como se presenta en la figura 7.3, dando un total de 28324 órbitas en 1825 días.

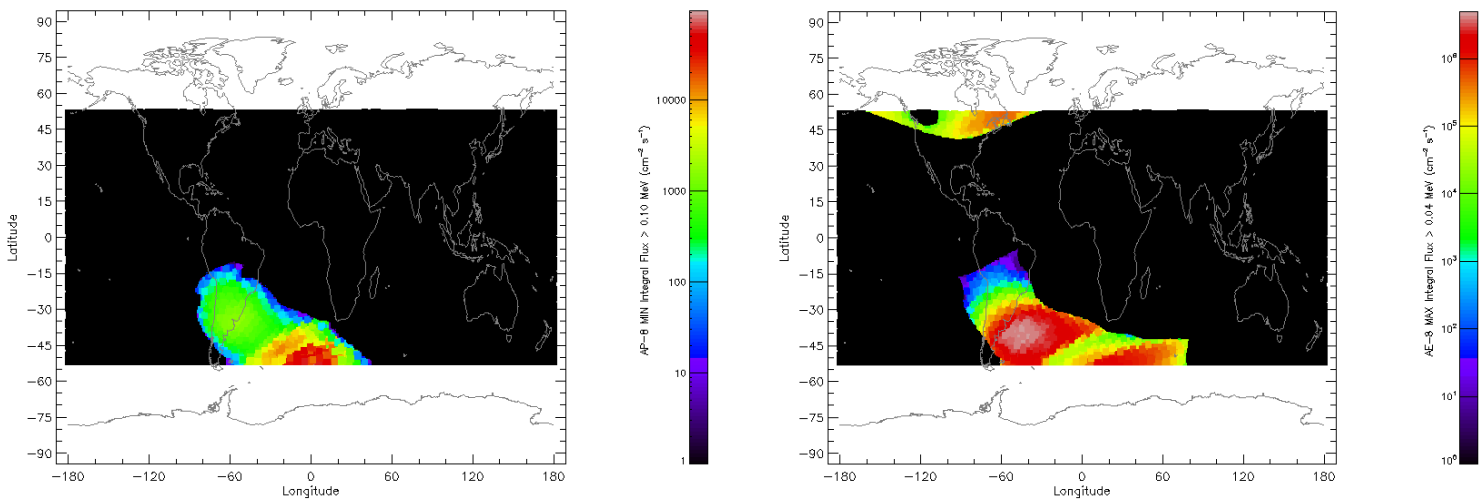


(a) Mapa de la trayectoria del satélite durante un día

(b) Mapa 3D de la trayectoria del satélite en un día

Figura 7.3: Trayectoria del satélite durante un día y la variación de su altura

Durante esta trayectoria, el sistema estará expuesto a la radiación atrapada en la atmósfera terrestre, que consiste en el flujo de protones (figura 7.4a) y de electrones (figura 7.4b) ubicados principalmente en la SAA (debido a que se trata de una órbita muy baja). Estas partículas pueden alcanzar un flujo integral mayor a $10 \times 10^4 \text{ MeV}$ (en el caso de los protones) y $10 \times 10^6 \text{ MeV}$ (en el caso de los electrones), representando una zona de riesgo en donde el SCMI puede presentar fallas inducidas por la radiación.



(a) Flujo de protones con energía $> 0.1 \text{ MeV}$ atrapados

(b) Flujo de electrones con energía $> 0.04 \text{ MeV}$ atrapados

Figura 7.4: Mapas del flujo total de las partículas cargadas atrapadas

A partir de lo anterior, en la figura 7.5 se presenta el espectro promedio de protones y electrones atrapados, los cuales alcanzan energías de hasta ≈ 400 MeV y ≈ 7 MeV, respectivamente.

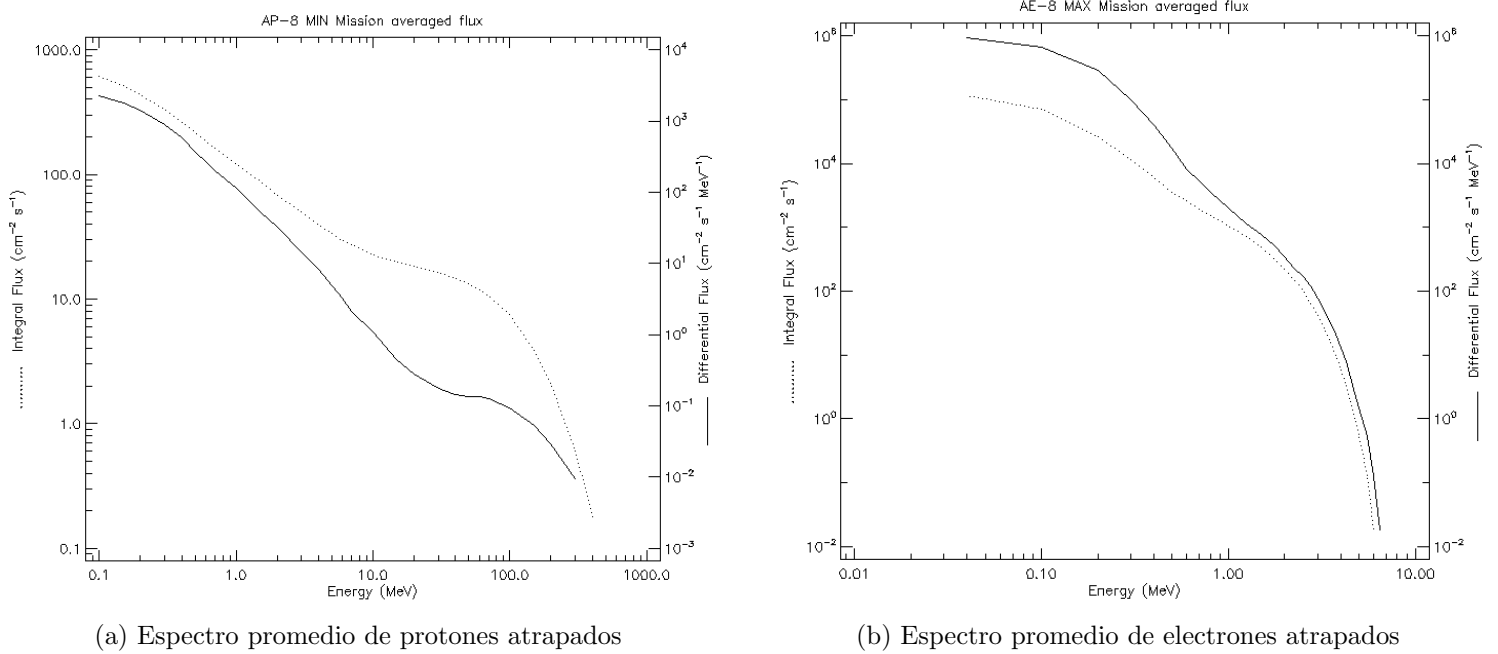


Figura 7.5: Espectro promedio total de las partículas cargadas atrapadas

Por último, en la figura 7.6 se presenta el espectro LET del flujo de partículas incidentes en la nave espacial considerando un blindaje de aluminio de $2mm$.

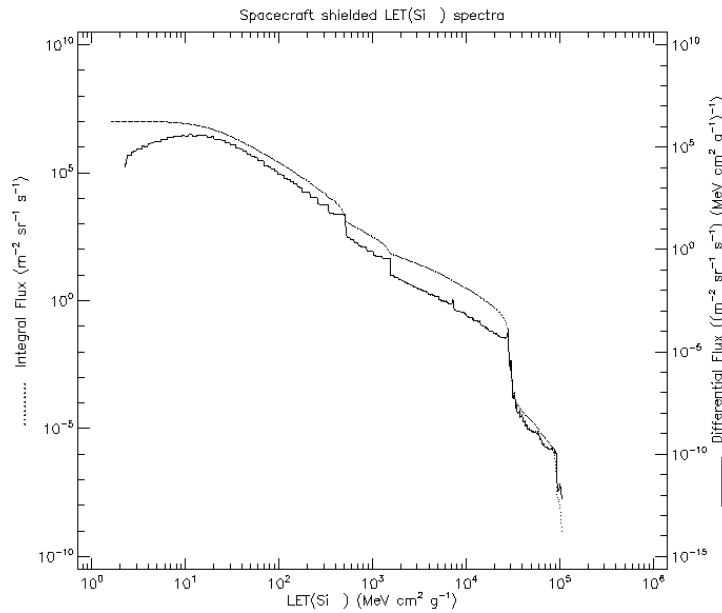


Figura 7.6: Espectro LET del flujo de partículas incidentes

8 DESARROLLO DEL CONCEPTO DE DISEÑO

El diseño es, en esencia, el arte de saber aplicar los conocimientos científicos y técnicos para determinar la mejor solución a una problemática determinada. Durante este proceso, el diseñador debe de aplicar su creatividad, conocimiento y experiencia para identificar un área de oportunidad presente en el problema que pretende resolver, además de las necesidades que se buscan satisfacer con la propuesta. De esta forma, el diseñador tomará las decisiones pertinentes fundamentadas teóricamente para determinar la mejor manera de alcanzar sus objetivos. La primera etapa del proceso de diseño es el desarrollo del concepto, que consiste en el proceso de especular ideas para la resolución del proyecto manteniendo un alto nivel de abstracción, seguida del diseño a nivel de sistema, en donde se resuelven las tareas que realizará el sistema y cada uno de sus bloques, eliminando poco a poco el nivel de abstracción.



8.1 Antecedentes del proyecto

El trabajo de tesis parte del desarrollo del *LIESE-SCMI* [1], [2], (véase la figura 8.1) con el cual se propone una arquitectura maestro – supervisor empleando microcontroladores. En dicho trabajo el microcontrolador maestro se encarga del procesamiento central del SCMI, mientras que el supervisor ejecuta tareas de diagnóstico en el maestro, leyendo periódicamente las etiquetas hash calculadas por el maestro a partir de su código almacenado en flash y sus registros de configuración, para hacer una comparación con una copia de seguridad almacenada en una memoria externa, de modo que, si se detecta alguna perturbación que modifique la operación normal del maestro, el supervisor determine su ubicación y la gravedad de la falla, para posteriormente intervenir y ejecutar las tareas de recuperación pertinentes, que van desde el reinicio del maestro hasta su reprogramación total, con la finalidad de recuperar su último estado funcional. Con este esquema de diagnóstico y recuperación se protege al SCMI de la ocurrencia de SEFI y SEU, que son los efectos más comunes en los sistemas digitales colocados en LEO.

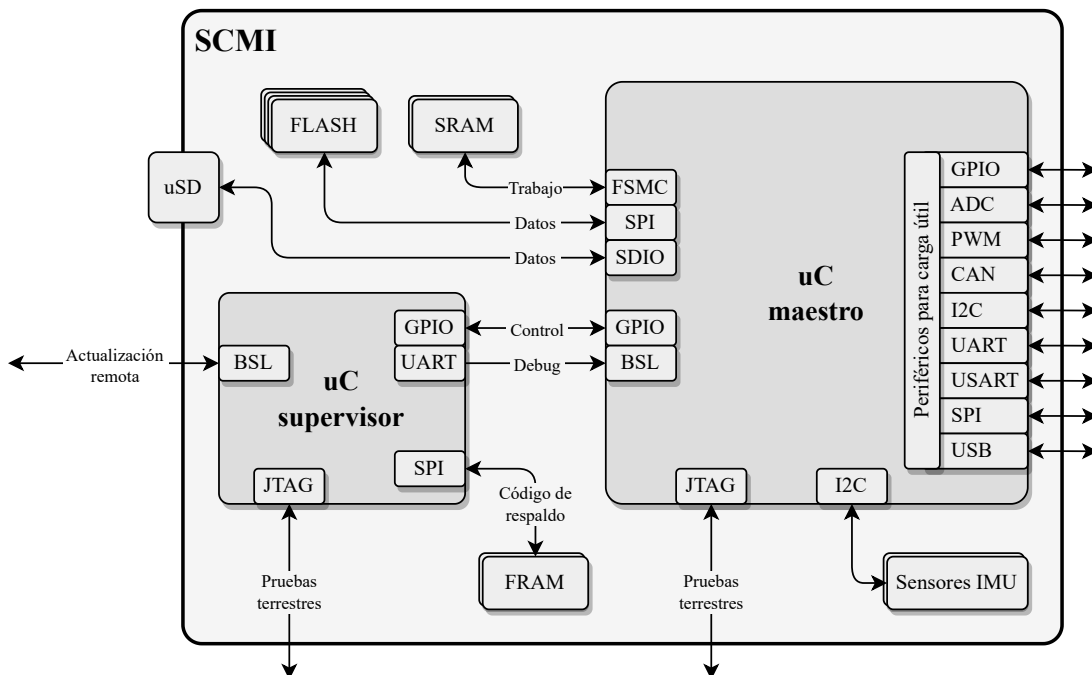


Figura 8.1: Arquitectura del *LIESE-SCMI* [1], [2]

La característica más contundente del *LIESE-SCMI* es que cumple con los requisitos determinados a partir de una investigación del estado del arte, por lo que teóricamente alcanza los niveles de desempeño y de fiabilidad para competir con los SCMI modernos. Sin embargo, el uso de dispositivos con capacidades de hardware bien delimitadas (como los microcontroladores), disminuye las opciones de adaptación del sistema a las fallas inducidas por el medio ambiente de radiación espacial. Para resolver esto, los diseñadores están migrando hacia dispositivos que permitan mayor flexibilidad durante el proceso de diseño para desarrollar arquitecturas más robustas, sin impactar en gran medida en los costos necesarios para la construcción del sistema. Con esto en mente, los objetivos de la tesis parten de la identificación como un área de oportunidad el migrar el diseño del *LIESE-SCMI* hacia una arquitectura con dispositivos configurables (como FPGA o SoC), ya que su estructura es ideal para diseñar un SCMI con los esquemas de tolerancia a fallas más robustos, de modo que cumpla con las características adecuadas para desempeñar misiones en LEO.

8.2 Identificación de las necesidades

Como primer paso en el proceso de diseño, es necesario identificar las necesidades que debe satisfacer el sistema, las cuales están relacionadas directamente con el concepto de diseño y servirán más adelante como base para el planteamiento de los requisitos y posteriormente en la determinación de las especificaciones. En la tabla 8.1 se presentan las necesidades agrupadas por el área de diseño a la que pertenecen, además de que se determina el nivel de importancia que tiene su cumplimiento durante el proceso de diseño, considerando una escala del 1 al 5, en donde el 5 representa la máxima importancia.

Tabla 8.1: Identificación de las necesidades

Código	Necesidad	Importancia
Necesidades generales		
GEN-01	Que el SCMI funcione correctamente en el ambiente de radiación espacial.	5
GEN-02	Que el SCMI sea compatible con una amplia variedad de subsistemas.	4
GEN-03	Que el SCMI pueda cumplir con un amplio rango de aplicaciones.	4
GEN-04	Que el SCMI utilice componentes con alta disponibilidad y accesibilidad.	5
GEN-05	Que el SCMI sea de bajo costo.	5
Necesidades electrónicas		
ELE-01	Que el SCMI tenga poder de procesamiento equiparable al estado del arte.	4
ELE-02	Que el SCMI se pueda actualizar en órbita.	5
ELE-03	Que el SCMI pueda actualizarse/modificarse para mejorar sus características.	5
Necesidades mecánicas		
MEC-01	Que el SCMI pueda integrarse fácilmente en un nanosatélite <i>CubeSat</i> .	4
MEC-02	Que el SCMI pueda integrarse fácilmente con otros subsistemas.	4
Necesidades energéticas		
ENE-01	Que la alimentación del SCMI sea compatible con un nanosatélite <i>CubeSat</i> .	4
ENE-02	Que el SCMI sea de bajo consumo energético.	5

8.3 Concepto de diseño del sistema

Siguiendo con el proceso de diseño y aplicando una metodología *top-down*, el concepto de diseño inicia con el planteamiento de un diagrama a bloques del sistema empleando el máximo nivel de abstracción (véase la figura 8.2), definiendo a las unidades principales del sistema: la **unidad maestra**, que consiste en el conjunto de elementos encargados de las tareas de control y procesamiento del SCMI, además de servir como interfaz de comunicación entre todas las unidades del sistema; la **unidad de supervisión**; encargada de realizar tareas de diagnóstico y recuperación de la unidad maestra, de modo que si esta llega a presentar la ocurrencia de una falla, la unidad de supervisión intervenga en su flujo de operación para diagnosticar su estado y realizar las acciones de recuperación

necesarias para restaurar su operación normal; el **conector del SCMI**, que será la interfaz de interconexión del SCMI con los otros subsistemas y el bus principal del satélite; el **banco de memorias**, que contempla a los dispositivos de almacenamiento de toda la información necesaria para la correcta operación del SCMI; y el **sistema de control y determinación de posición** (ADCS, por sus siglas en inglés), que integra a los sensores necesarios para determinar y controlar la orientación del satélite (sin considerar a los actuadores).

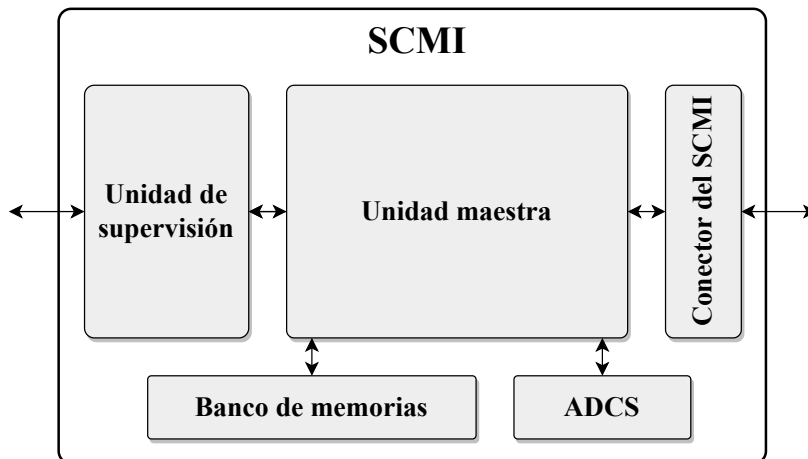


Figura 8.2: Concepto de diseño del SCMI con el máximo nivel de abstracción

Posteriormente a la definición de las unidades principales del SCMI y a partir de la lista de necesidades (tabla 8.1), se determina el nivel de importancia (aplicando la misma escala) para el cumplimiento de las necesidades, pero a nivel de bloque, presentado en la tabla 8.2, con la cual es más claro ubicar qué tipo de necesidades son las de mayor prioridad para cada unidad del SCMI.

Tabla 8.2: Identificación de las necesidades por unidad del SCMI

Código	Unidad maestra	Unidad de supervisión	Conector del SCMI	Banco de memorias	ADCS
Necesidades generales					
GEN-01	5	5	4	4	4
GEN-02	5	1	5	1	1
GEN-03	5	1	5	5	3
GEN-04	5	4	4	4	4
GEN-05	5	3	4	5	5
Necesidades electrónicas					
ELE-01	5	1	1	4	3
ELE-02	5	3	1	1	1
ELE-03	5	3	1	1	1
Necesidades mecánicas					
MEC-01	2	1	5	1	1
MEC-02	4	1	5	1	1
Necesidades energéticas					
ENE-01	5	5	1	5	5
ENE-02	3	5	1	3	3

A partir de lo anterior, el siguiente paso consiste en eliminar el nivel de abstracción en el SCMI con base en las necesidades (tabla 8.2), por lo que es necesario dividir a las unidades principales en módulos con tareas más específicas, las cuales se describen a continuación.

Comenzando con las necesidades que debe de cumplir la unidad maestra, esta debe de implementarse en un dispositivo reconfigurable, flexible y escalable, de modo que se pueda alcanzar un alto nivel de fiabilidad. Con esto en mente y con base en el estado del arte, se propone el uso de un FPGA, aprovechando su estructura para implementar una arquitectura robusta, de alto desempeño y, gracias a su capacidad de reconfiguración permite instrumentar esquemas tolerantes a fallas y escalables, capaz de adaptarse a un amplio rango de aplicaciones y tolerar la ocurrencia de fallas inducidas por el medio ambiente de radiación espacial, evitando en lo posible la degradación de la funcionalidad del SCMI. Para esto, se determina que la unidad maestra debe de dividirse en cuatro módulos fundamentales: la **unidad de procesamiento central** (CPU, por sus siglas en inglés), como el módulo principal del SCMI considerando una arquitectura centralizada al ejecutar y distribuir todos los comandos de control y procesar los necesarios para llevar a cabo la aplicación definida; el **bloque de diagnóstico y reconfiguración**, que recopilará la información sobre el estado funcional de la unidad maestra y mantendrá la comunicación con la unidad de supervisión para ejecutar los procesos de diagnóstico y reconfiguración pertinentes para incrementar la fiabilidad del FPGA; el bloque de **periféricos**, que será la interfaz de comunicación con la unidad del ADCS y con el conector del SCMI; y el **controlador de memorias**, que se encargará de la transferencia de datos con las unidades de almacenamiento.

Por parte de la unidad de supervisión, esta se dividirá en dos elementos: el **microcontrolador supervisor**, que intercambiará información con el bloque de diagnóstico y reconfiguración de la unidad maestra para diagnosticar su estado funcional y, en caso de requerirse, ejecutar los comandos de control para recuperar su operación normal; y la **memoria de respaldo y configuración**, que almacenará el *bitstream* de configuración del FPGA y lo guardará como respaldo para cuando se necesiten ejecutar tareas de reconfiguración y actualización de la unidad maestra.

Con base en lo anterior, el concepto de diseño del SCMI con el segundo nivel de abstracción se presenta en 8.3.

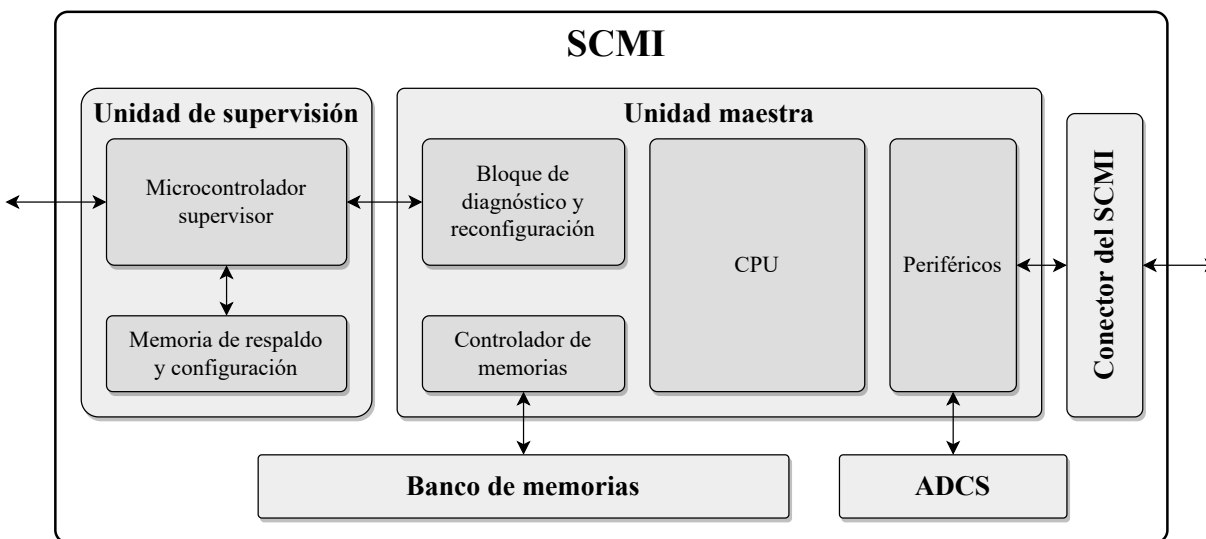


Figura 8.3: Concepto de diseño del SCMI con el segundo nivel de abstracción

9 DISEÑO A NIVEL DE SISTEMA

El diseño a nivel de sistema consiste en la definición de la arquitectura, a partir del concepto de diseño, profundizando con mayor detalle en cada uno de los módulos para determinar su funcionamiento y estructura interna, lo que a su vez llevará a la definición de bloques más pequeños, así como las interfaces con las que van a interactuar entre sí. En otras palabras, se emplea un método de diseño modular que facilite el proceso de diseño agrupando las tareas en pequeños módulos para, posteriormente, realizar un proceso de integración, dando como resultado la estructura completa del sistema.



9.1 Planteamiento de los requisitos

Como primer paso en el diseño a nivel de sistema, es necesario plantear los parámetros que el sistema debe satisfacer o, también llamados, los requisitos a nivel de sistema (tabla 9.1). Todo requisito parte de una necesidad, por lo que es importante relacionar cada requisito con las necesidades que está resolviendo, además de agruparlos en requisitos generales, orientados a cumplir con las necesidades básicas del sistema; y los requisitos funcionales, dirigidos a cubrir las necesidades de operación y desempeño del SCMI, aplicando la misma escala (de 1 a 5) para definir un nivel de prioridad para el cumplimiento de cada uno de los requisitos.

Tabla 9.1: Determinación de los requisitos a nivel de sistema

Código	Requisito	GEN-01	GEN-02	GEN-03	GEN-04	GEN-05	ELE-01	ELE-02	ELE-03	MEC-01	MEC-02	ENE-01	ENE-02	Prioridad
Requisitos generales														
RG-01	Nivel de fiabilidad.	✓	✓	✓										5
RG-02	Reconfigurable.	✓	✓	✓			✓	✓	✓					5
RG-03	Escalable.		✓	✓			✓		✓		✓			5
RG-04	Tolerante a fallas.	✓		✓		✓								5
RG-05	Flexibilidad.		✓	✓					✓		✓			5
RG-06	Compacto.					✓				✓	✓			4
RG-07	Uso de componentes COTS.				✓	✓								5
RG-08	Rango de temperatura.	✓		✓										4
RG-09	Bajo costo.			✓	✓	✓								5
RG-10	Disponibilidad de los componentes.				✓	✓			✓					5
RG-11	Bajo consumo energético.		✓	✓						✓		✓	✓	5
RG-12	Cumplimiento de la CDS.	✓		✓						✓	✓	✓	✓	5
Requisitos funcionales														
RF-01	Nivel de desempeño.			✓			✓							4
RF-02	Procesamiento y envío de datos.		✓	✓			✓							5
RF-03	Control y monitoreo de los subsistemas.		✓	✓			✓							5
RF-04	Validación y distribución de comandos.		✓	✓			✓							5

9.2 Unidad maestra

Iniciando con el proceso de profundización en los módulos del SCMI, la unidad maestra consiste en la arquitectura construida para ser la unidad principal del SCMI, la cual desempeñará todas las tareas de procesamiento, control, gestión de recursos y la distribución de datos y comandos en todos elementos del sistema y todos los subsistemas del satélite. A partir de lo anterior, se determina el nivel de prioridad para el cumplimiento de los requisitos que deben de cumplir cada uno de sus módulos internos (tabla 9.2).

Tabla 9.2: Determinación de los requisitos de la unidad maestra

Código	CPU	Bloque de diagnóstico y reconfiguración	Periféricos	Controlador de memorias
Requisitos generales				
RG-01	5	5	5	5
RG-02	5	5	5	5
RG-03	5	3	5	3
RG-04	5	5	5	5
RG-05	5	3	5	3
RG-06	2	4	4	4
RG-07	5	5	5	5
RG-08	5	5	5	5
RG-09	3	3	3	3
RG-10	5	5	5	5
RG-11	3	5	5	5
RG-12	5	5	5	5
Requisitos funcionales				
RF-01	5	2	3	3
RF-02	5	2	5	5
RF-03	5	1	-	-
RF-04	5	2	-	-

9.2.1 Unidad de procesamiento central (CPU)

La CPU es el elemento central del FPGA empleado como unidad maestra, por lo que es estrictamente necesario que presente un alto nivel de fiabilidad, ya que, si llega a presentar una falla, es altamente probable que esta se propague a otros módulos o unidades del SCMI. Este tipo de elementos se denominan como puntos comunes de falla (SPOF, por sus siglas en inglés) y representan en gran medida al nivel de fiabilidad del sistema, por lo que deben de ser protegidos por técnicas de tolerancia a fallas. Además, la CPU es un elemento mucho más complejo en comparación con los otros módulos, lo que también reduce su nivel de fiabilidad.

Con base en lo anterior y en los resultados del estado del arte, se propone el uso de un *softcore* que cuente con las características para cumplir con los parámetros de desempeño para desarrollar un amplio rango de aplicaciones, protegido por un esquema de TMR a nivel de procesador, con lo cual se podrán enmascarar las fallas que se presenten en un solo procesador.

9.2.2 Bloque de diagnóstico y reconfiguración

El bloque de diagnóstico y reconfiguración es la zona del FPGA encargada de comunicarse con la unidad de supervisión, intercambiando información a partir de una serie de comandos que permitirán determinar el estado operativo de la unidad maestra, de modo que si se detecta la presencia de un error en la memoria de configuración

del FPGA (derivado de la ocurrencia de un SEU) que represente un malfuncionamiento en la operación de alguno de sus módulos necesarios para cumplir con la aplicación, esta servirá como interfaz para realizar acciones correctivas de la unidad maestra. Para estas tareas, se construyen tres bloques: el **bloque de control**, encargado de intercambiar comandos de control con la unidad de supervisión; el **bloque de diagnóstico**, que mandará los datos a la unidad de supervisión para que esta determine el estado operativo de la unidad maestra; y el acceso al puerto JTAG del FPGA, a través del cual se realizará el proceso de configuración correspondiente. De esta manera, además de la TMR aplicada en la CPU, se realizarán tareas de diagnóstico periódicas que podrán llevar a acciones correctivas, dependiendo de la presencia de un error.

9.2.3 Periféricos

Por parte del bloque de periféricos, la unidad maestra debe de tener la capacidad de comunicarse con elementos externos. Para esto, se proponen tres tipos de interfaces de comunicación: las **interfaces para el satélite**, considerando aquellos buses de comunicación que servirán para integrar al SCMI al bus destinado para intercambiar información entre todos los subsistemas del satélite; las **interfaces para la carga útil**, disponiendo de los periféricos necesarios para brindar una alta capacidad de integración al SCMI, de modo que pueda cumplir con una amplia variedad de aplicaciones; y los **periféricos para el ADCS**, con los cuales se implementará una red de comunicación con los sensores del SCMI seleccionados para determinar la orientación del satélite. Además, se utilizará un conector PC/104, que es el conector con mayor presencia en el estado del arte, el cual integrará a las interfaces para el satélite y para la carga útil, en conformidad con el factor de forma PC/104 para *CubeSat* [68].

9.2.4 Controlador de memorias

El controlador de memorias gestionará a los dispositivos de almacenamiento externos a la unidad maestra, por lo que considera a las interfaces de comunicación seleccionadas con base en la selección de las memorias (véase la sección 9.4).

Con base en lo anterior, el diseño de la unidad maestra con el tercer nivel de abstracción se presenta en la figura 9.1.

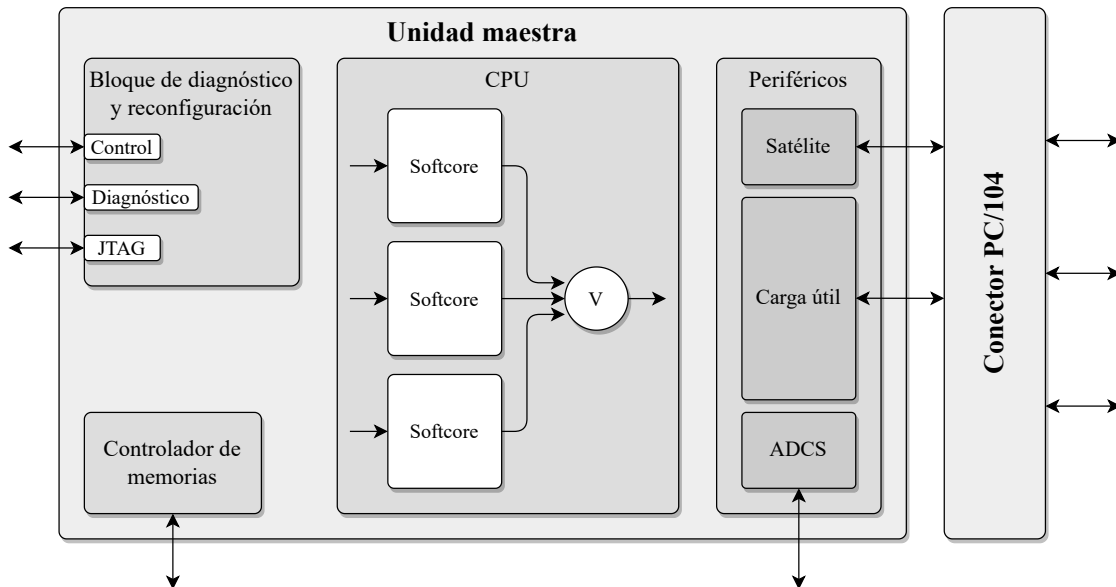


Figura 9.1: Diseño de la unidad maestra con el tercer nivel de abstracción

9.3 Unidad de supervisión

La unidad de supervisión se encargará de mantener una comunicación constante con la unidad maestra, más específicamente con su bloque de diagnóstico y reconfiguración, de modo que sea capaz de detectar el malfuncionamiento en alguno de sus módulos y, a partir de ello, de iniciar a la ejecución de tareas de recuperación, con las cuales se buscará que la unidad maestra recupere su funcionalidad completa después de la ocurrencia de eventos que afecten en algún grado su capacidad de cumplir con la misión.

Con base en lo anterior, el nivel de prioridad para el cumplimiento de los requisitos para la unidad de supervisión se presenta en la tabla 9.3.

Tabla 9.3: Determinación de los requisitos de la unidad de supervisión

Código	Microcontrolador supervisor	Memoria de respaldo y configuración
Requisitos generales		
RG-01	5	5
RG-02	2	2
RG-03	3	3
RG-04	5	5
RG-05	2	2
RG-06	4	3
RG-07	3	3
RG-08	5	5
RG-09	3	3
RG-10	4	4
RG-11	5	5
RG-12	3	3
Requisitos funcionales		
RF-01	2	2
RF-02	2	2
RF-03	3	1
RF-04	4	1

9.3.1 Microcontrolador supervisor

Se propone un microcontrolador supervisor como dispositivo principal de la unidad de supervisión, el cual llevará el control de las tareas de diagnóstico y reconfiguración del SCMI. Esto significa que su uso es mínimo en comparación con las tareas que ejecuta la unidad maestra del SCMI, por lo que su nivel de desempeño no es un requisito de alta prioridad, además de que es necesario ubicar a un dispositivo cuyo consumo energético no impacte en el sistema.

Con base en lo anterior, se identificaron algunas opciones de microcontroladores COTS de bajo desempeño en el estado del arte que, por sus características en el diseño, presentan una resistencia a la radiación elevada, por lo que la propuesta realizada es una opción viable para el microcontrolador supervisor.

Profundizando más en el microcontrolador supervisor, este se divide en cuatro bloques principales: un módulo de **control**, con el cual se mandarían señales de control hacia el FPGA; el **bloque de diagnóstico** y el **bloque de reconfiguración**, que serán las interfaces encargadas del intercambio de datos para las tareas correspondientes; y una interfaz de **debug** para permitir la actualización de su código en órbita, permitiendo también la ejecución de tareas de diagnóstico y recuperación manuales a partir de otro subsistema del satélite y, dependiendo del caso, desde una estación terrena.

9.3.2 Memoria de respaldo y configuración

Debido a que se propone el uso de un FPGA construido con SRAM, es necesario integrar una memoria de configuración que almacene el código de configuración de la unidad maestra. Además, por lo esquemas de tolerancia a fallas propuestos, también se necesita de una memoria que almacene el código de respaldo del FPGA, que pueda ser controlada por el microcontrolador supervisor para ejecutar la reconfiguración global del FPGA. Tomando en cuenta ambas necesidades, se propone el uso de una sola memoria para ambos propósitos, lo que significa que debe de ser una memoria con una resistencia a la radiación cercana a la del microcontrolador supervisor.

De acuerdo con lo anterior, el diseño de la unidad de supervisión con el tercer nivel de abstracción se presenta en la figura 9.2.

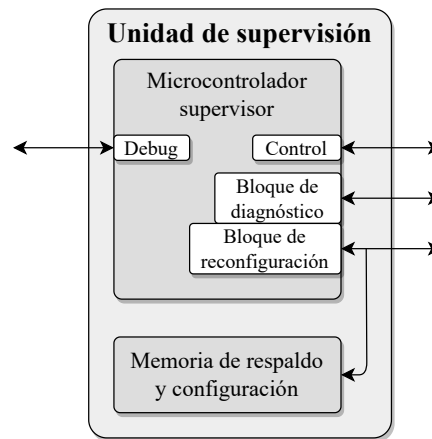


Figura 9.2: Diseño de la unidad de supervisión con el tercer nivel de abstracción

9.4 Banco de memorias

El banco de memorias integra a todos los dispositivos de almacenamiento del SCMI, considerando cuatro tipos de almacenamiento: la **memoria de trabajo**, que utilizará la CPU para realizar el cálculo y procesamiento de los datos; la memoria para el **almacenamiento masivo de datos**, que servirá para almacenar la mayor densidad de los datos obtenidos a partir de la carga útil; la memoria de **almacenamiento crítico de datos**, que se utilizará para respaldar únicamente los datos de alta prioridad, ya sea obtenidos por la carga útil o aquellos necesarios para que se logren los objetivos de la misión; y la memoria para el **sistema operativo en tiempo real (RTOS)**, por sus siglas en inglés), que almacenará el código relacionado al RTOS.

Con base en lo anterior, en la tabla 9.4 se presentan los requisitos para todos los tipos de almacenamiento.

Tabla 9.4: Determinación de los requisitos del banco de memorias

Código	Memoria de trabajo	Memoria de almacenamiento masivo de datos	Memoria de almacenamiento crítico de datos	Memoria para el RTOS
Requisitos generales				
RG-01	3	3	5	4
RG-02	-	-	-	-
RG-03	-	-	-	-
RG-04	3	3	5	5
RG-05	-	-	-	-
RG-06	1	1	3	3
RG-07	5	5	3	4

RG-08	5	5	5	5
RG-09	5	5	3	4
RG-10	5	5	4	5
RG-11	5	4	5	5
RG-12	5	5	5	5
Requisitos funcionales				
RF-01	5	2	1	5
RF-02	5	2	2	5
RF-03	-	-	-	-
RF-04	-	-	-	-

9.5 Sistema de control y determinación de posición (ADCS)

El ADCS de un satélite forma parte del sistema de guía, navegación y control (GNC, por sus siglas en inglés) y está encargado de la determinación de la orientación del satélite a partir de las lecturas de diversos sensores, como seguidores de estrellas, sensores solares, sensores de horizonte, magnetómetros, giroscopios y acelerómetros, entre otros. La integración de algunos sensores del ADCS en un SCMI se presenta comúnmente en el estado del arte, con el objetivo de ofrecer los sensores básicos que requiere un satélite para determinar y, a través de actuadores, controlar la orientación de un satélite, brindando mayores alternativas en un mismo sistema. Considerando lo anterior, el ADCS se divide en dos tipos de sensores: las **unidades de medición inercial** (IMU, por sus siglas en inglés), que son acelerómetros, giroscopios y magnetómetros; y **sensores de temperatura**, que no necesariamente se utilizan para las tareas del ADCS, pero se integran en él por facilidad de conexión, ya que utilizan las mismas interfaces de comunicación.

La determinación del nivel de prioridad de los sensores del ADCS se observa en la tabla 9.5.

Tabla 9.5: Determinación de los requisitos del ADCS

Código	IMU	Sensores de temperatura
Requisitos generales		
RG-01	4	4
RG-02	1	1
RG-03	-	-
RG-04	3	3
RG-05	3	3
RG-06	5	5
RG-07	5	5
RG-08	5	5
RG-09	5	5
RG-10	5	5
RG-11	5	5
RG-12	5	5
Requisitos funcionales		
RF-01	2	2
RF-02	2	2
RF-03	-	-
RF-04	-	-

9.6 Diseño a nivel de sistema

Con base en la descripción de las unidades del SCMI y en los requisitos que estas deben de satisfacer, se presenta en la figura 9.3 el diseño a nivel de sistema del SCMI, integrando a todos sus elementos con el tercer nivel de abstracción.

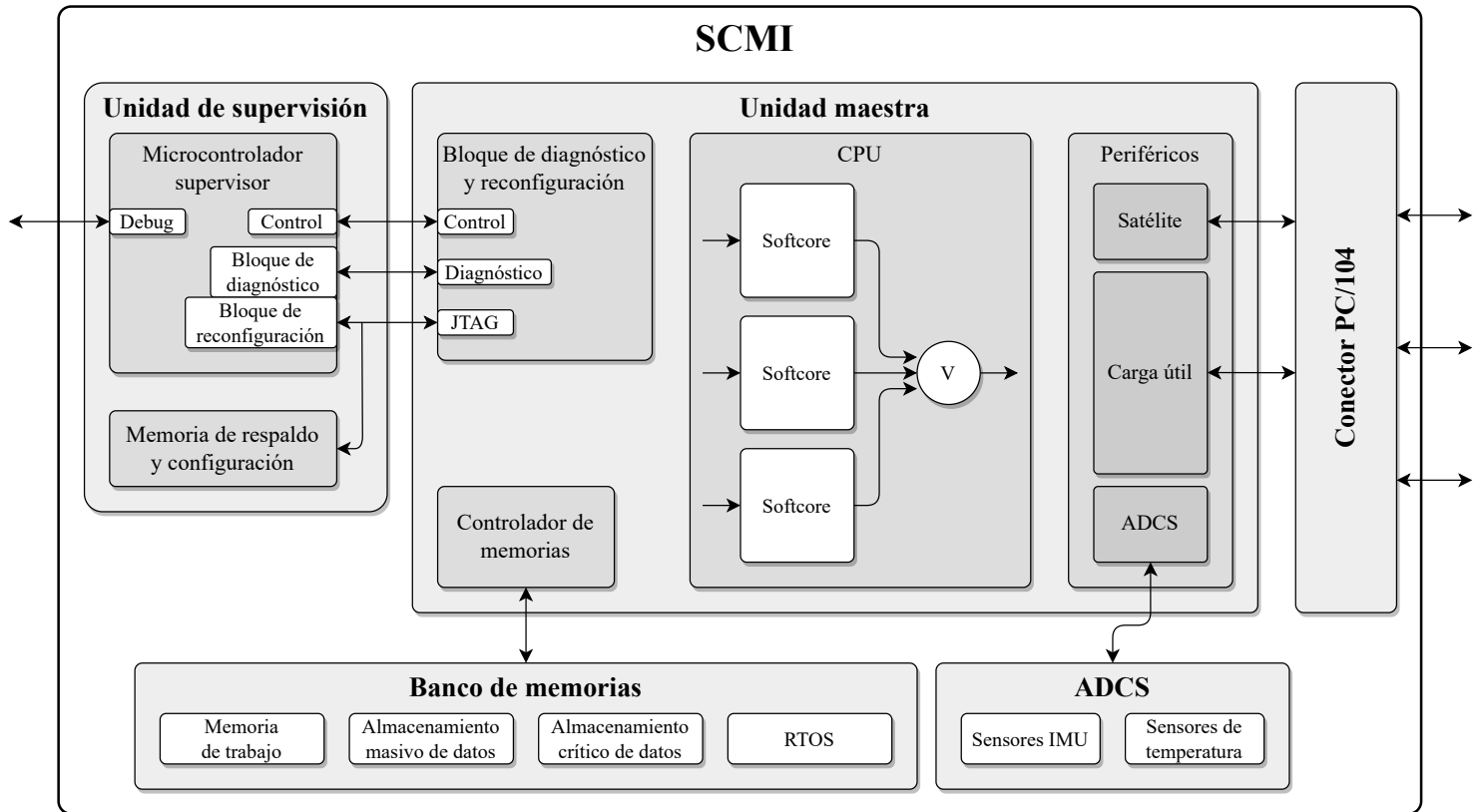


Figura 9.3: Diseño a nivel de sistema del SCMI con el tercer nivel de abstracción

Parte IV

DISEÑO DE DETALLE

10 DISEÑO DE DETALLE A NIVEL MODULAR

En este punto del trabajo se tiene la idea general de los módulos internos del SCMI y las funciones que realizan, pero aun existe la pregunta ¿cómo se cumplirán esos parámetros?. Siguiendo la metodología de diseño, la siguiente etapa consiste en el diseño de detalle, que es el proceso de ingeniería en donde se profundiza completamente en todos los elementos del sistema. Para lograr esto, el primer paso se centra en la determinación de las especificaciones de diseño, proceso con el cual se traducen los requisitos en las métricas necesarias para obtener el “diseño definitivo”. En este capítulo se aplican los conocimientos en el área para determinar las especificaciones que el sistema debe de cumplir, presentando el sistema a nivel modular que resolverá de manera óptima todas las funciones necesarias para obtener un SCMI tolerante a fallas, explicando con el máximo nivel de detalle la estructura interna de los módulos definidos con anterioridad, para presentar al final el diseño de detalle del sistema.



10.1 Especificaciones generales

Comenzando con las especificaciones generales, a partir de la investigación del estado del arte se definen los parámetros más comunes que se deben de cumplir en el diseño (tabla 10.1), respetando la CDS. De este modo, el SCMI será compatible con el estándar *CubeSat*.

Tabla 10.1: Especificaciones generales [66]

Parámetro	Parámetro	Mínimo	Valor Típico	Máximo	Unidades
Tensión de operación	V_{DD}	3	3.3	5	V
Consumo energético	P_D			1500	mW
Temperatura de operación	T_A	-40		85	C
Temperatura de almacenamiento	T_{STG}	-55		105	C
Masa	m			150	g

10.2 Diseño de detalle a nivel modular

En esta parte del trabajo se terminará el proceso de definición de las características, tareas y elementos de todas las unidades del SCMI a nivel modular.

10.2.1 Unidad maestra

Como se mencionó anteriormente, para la unidad maestra se utilizará un FPGA, sintetizando un *softcore* con TMR a nivel de procesador en la CPU que ejecutará las tareas de procesamiento y control del SCMI. Para esto, se selecciona el *softcore MicroBlaze* de *Xilinx* (véase la figura 10.1), el cual se utiliza frecuentemente en los SCMI del estado del arte gracias a sus características, entre las cuales destacan bloques IP para la implementación de TMR, como el dictaminador de mayoría para diferentes bloques internos, un administrador para controlar los bloques redundantes, un comparador de los bloques redundantes, un inyector de fallas para pruebas y la IP SEM para mitigación de errores [69].

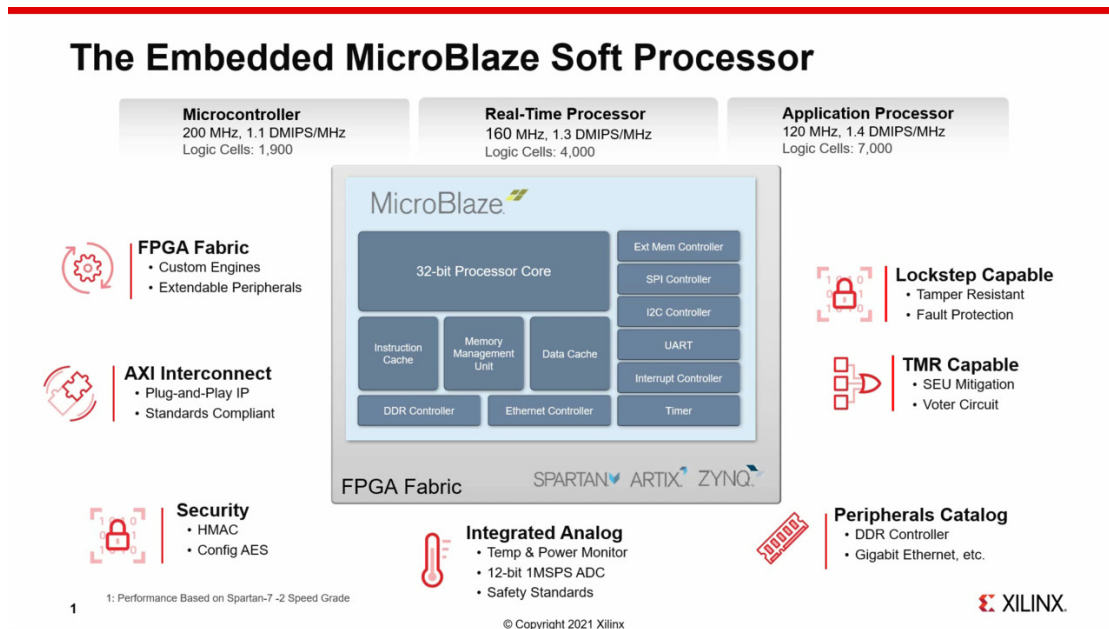


Figura 10.1: Vista general de la arquitectura del *softcore* MicroBlaze de Xilinx [69]

Para usar el *MicroBlaze* como CPU, es necesario utilizar un dispositivo de Xilinx, los cuales son utilizados frecuentemente en sistemas espaciales. Como se presentó en el estado del arte, esto se debe en parte a la gran variedad de dispositivos disponibles en Xilinx, incluyendo desde FPGA hasta SoC con múltiples *hardcores* embebidos, permitiendo la implementación de arquitecturas de alto desempeño, añadiendo a todas las herramientas de diseño de Xilinx, las cuales cuentan con características útiles para el diseño de estos sistemas, como lo es *Vivado Design Suite* y todas sus herramientas de integración de bloques IP, como el mismo *MicroBlaze*.

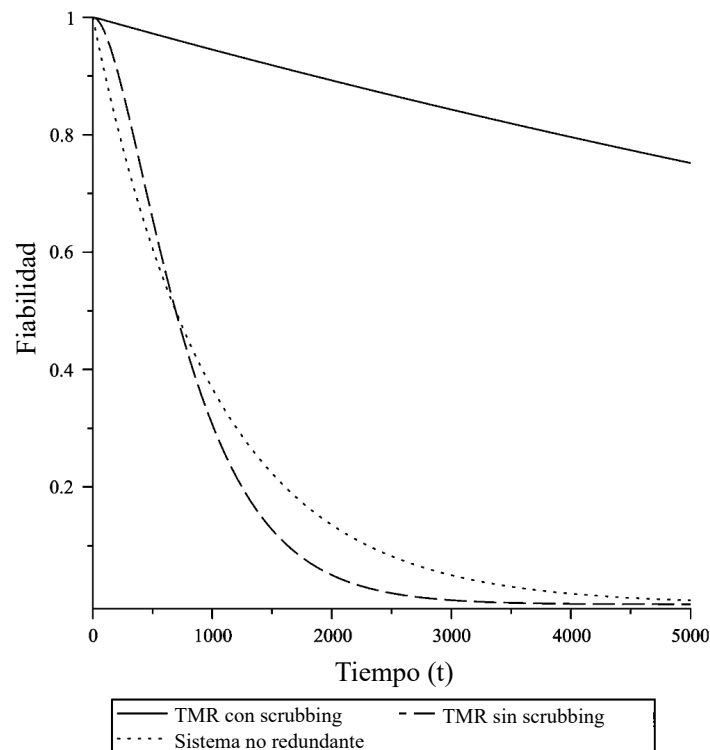


Figura 10.2: Fiabilidad de un sistema en el tiempo continuo de un sistema sin redundancia, con TMR y TMR con *memory scrubbing* [70]

En este proyecto se aplicará TMR en el *MicroBlaze* para incrementar la fiabilidad de la CPU, que es el módulo más importante y complejo del SCMI. Sin embargo, a pesar de que la TMR es efectiva en FPGA basados en SRAM para mitigar SEU, no es suficiente contra la ocurrencia de SEU, por lo que es necesario combinarlo con otras técnicas de mitigación. Para resolver esto, se implementará el *memory scrubbing* para recuperar a un módulo con falla. En la figura 10.2 se presenta el nivel de fiabilidad de un sistema en el tiempo continuo, comparando su comportamiento entre un sistema sin redundancia, un sistema con TMR y un sistema que combina la TMR con el *memory scrubbing*, demostrando un incremento considerable en la fiabilidad al combinar ambas técnicas.

Para implementar el *memory scrubbing* se utilizará el bloque de diagnóstico y reconfiguración, el cual supervisará en paralelo a la CPU mientras esta ejecuta el código de la aplicación (como se muestra en el diagrama de secuencias de la figura 10.3), obteniendo datos de diagnóstico que enviará periódicamente a la unidad de supervisión para que esta ejecute tareas de diagnóstico. De este modo, en el caso de detectarse la presencia de un error en algún módulo triplicado, la unidad de supervisión regresará a la CPU al último estado operativo conocido (definido por un punto de verificación). Posteriormente, se solicitarán de nuevo los datos de diagnóstico para determinar si se eliminó el error, que en caso de persistir, conllevará a la reconfiguración global del FPGA, con el objetivo de regresar al SCMI a su estado perfecto y evitar la acumulación de SEU.

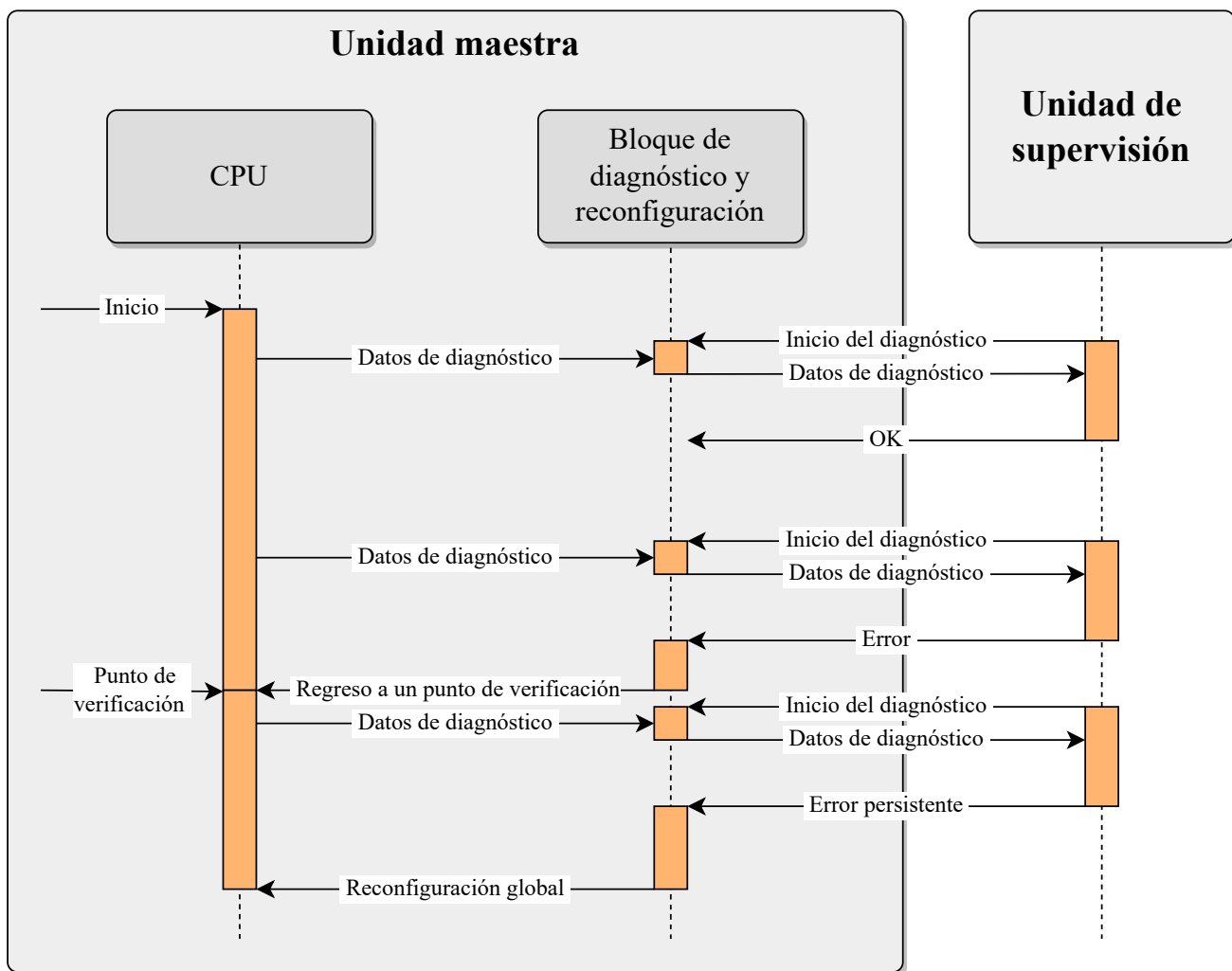


Figura 10.3: Diagrama de secuencias de las tareas de diagnóstico y reconfiguración

De esta manera, se necesita una supervisión constante de la CPU, por lo que se agregará un módulo comparador que recibirá las salidas de los *MicroBlaze* triplicados y entregará los datos al bloque de diagnóstico y reconfiguración, que conectará al microcontrolador supervisor y al FPGA como maestro – esclavo, compartiendo a la memoria de

respaldo y reconfiguración, permitiendo el control y el diagnóstico del FPGA, además de tareas de *debug* a través del puerto JTAG.

Por parte de los periféricos, se seleccionaron aquellos identificados con mayor frecuencia en el estado del arte para cumplir los tres propósitos definidos con anterioridad: la comunicación con el satélite, la comunicación con la carga útil y la comunicación con el ADCS del SCMI.

10.2.2 Unidad de supervisión

La unidad de supervisión se encarga de realizar las tareas de control, diagnóstico y recuperación de la unidad maestra, siguiendo el diagrama de secuencias de la figura 10.3, iniciando el proceso de diagnóstico al solicitar los datos recopilados por el bloque de diagnóstico y reconfiguración del FPGA, con los cuales determinará el estado operativo de la CPU y, a partir de ello, responderá con un “OK” o con el comando para iniciar las tareas correctivas, que van desde el regreso a un punto de verificación del código de la aplicación, hasta la reconfiguración global del FPGA.

El control de las tareas periódicas de diagnóstico y reconfiguración de manera autónoma se realizará en el microcontrolador supervisor, con base en los parámetros de la misión. Además, este se apoyará de una memoria de respaldo y configuración que almacenará el código de oro de la misión, de modo que siempre se cuente con un respaldo de lo que debe de estar ejecutando la unidad maestra.

Otra función importante del microcontrolador supervisor es permitir la actualización remota de la configuración del FPGA o del código de la misión, ya sea para actualizar los objetivos de la misión o para aplicar técnicas de recuperación manuales en el caso de que se pierda la autonomía del SCMI por alguna falla que lo congele en algún punto imposible de detectar por parte del bloque de diagnóstico y reconfiguración. Para estas funciones, se aprovecha que el microcontrolador supervisor ya puede realizar tareas de control y reconfiguración del FPGA, accediendo a su modo *bootloader* a través de un bus transmisor-receptor asíncrono universal (UART, por sus siglas en inglés) para actualizar su código o, usarlo como interfaz para modificar el código de oro de la memoria de respaldo y configuración, para después reconfigurar al FPGA y actualizar su arquitectura.

Para que la unidad de supervisión cumpla con sus funciones es fundamental que se seleccionen dispositivos con menor probabilidad de falla que la unidad maestra, lo que significa que una alternativa es utilizar componentes endurecidos a la radiación. Sin embargo, el aumento del costo de estos componentes es muy grande, por lo que no representa una opción viable considerando los alcances de este proyecto. Para atacar esto, se busca una solución intermedia al seleccionar componentes con herencia de vuelo, de modo que se utilicen componentes caracterizados que hayan sido utilizados previamente en aplicaciones espaciales y obtenido resultados adecuados para su uso en LEO. Para cumplir con estas características se utiliza la tecnología de memoria ferroeléctrica de acceso aleatorio (FRAM, por sus siglas en inglés), que por diseño presenta una mayor tolerancia a la radiación que otro tipo de tecnologías, como la FLASH, presentando un buen desempeño bajo un ambiente de radiación [49].

10.2.3 Banco de memorias

Para el banco de memorias se definieron cuatro tipos de almacenamiento, para los cuales se integrarían los dispositivos de almacenamiento que cumplieran en mejor medida con el tipo de información que guardarán. Sin embargo, la selección de un FPGA de *Xilinx* permite tomar otro tipo de decisiones, ya que estos dispositivos cuentan normalmente con BRAM embebida, la cual se puede distribuir internamente en el FPGA para que la CPU la utilice como memoria de trabajo y para el RTOS. A partir de lo anterior, el banco de memorias únicamente considerará las memorias de almacenamiento masivo y crítico de datos.

Comenzando con el almacenamiento masivo de datos, la principal característica de estas memorias es tener la capacidad suficiente para almacenar los datos adquiridos de la misión, en su mayoría. De esta manera, se definieron unidades redundantes de memoria empleando diferentes tecnologías: memoria FLASH y digital segura (uSD, por sus siglas en inglés).

Por otro lado, la memoria de almacenamiento crítico no necesita ser de gran densidad, ya que su función es almacenar únicamente los datos de mayor prioridad, ya sean aquellos necesarios para cumplir con la misión, o los datos obtenidos de la misión que no se pueden perder. Para cumplir con estos parámetros se utiliza el mismo método de selección que la unidad de supervisión, empleando FRAM que presenten un buen rendimiento en presencia de un ambiente de radiación.

Los bancos de memoria FLASH y FRAM se conectarán a través de la interfaz periférico serial cuádruple (QSPI, por sus siglas en inglés).

10.2.4 ADCS

Por último, el ADCS se dividirá en dos bloques redundantes, cada uno con un giroscopio y un magnetómetro para la determinación de la orientación del satélite; y dos sensores de temperatura local distribuidos en diferentes ubicaciones del SCMI, con el objetivo de monitorear la temperatura del sistema. Para cada bloque de ADCS se utilizará un bus de circuito inter-integrado (I2C, por sus siglas en inglés).

10.3 Diseño de detalle del SCMI

Con base en lo anterior, se integra el diseño de detalle de todas las unidades en un único sistema, dando como resultado el diseño de detalle del SCMI (véase la figura 10.4), en donde se definen también todos los medios de interacción entre los módulos internos.

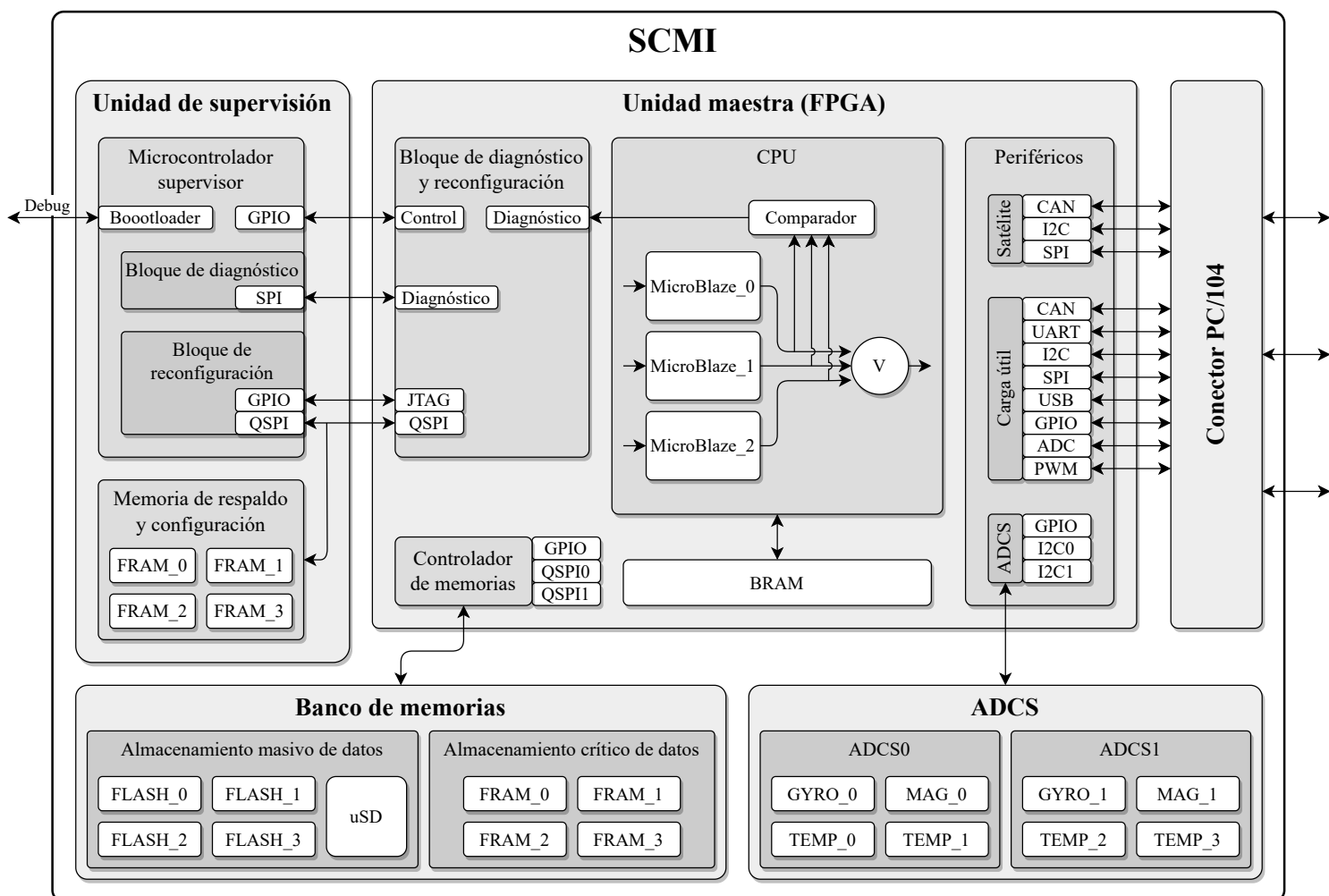


Figura 10.4: Diseño de detalle del SCMI

10.4 Selección de componentes

Considerando las especificaciones del diseño, se realiza la selección de componentes del SCMI, que se presenta en la tabla 10.2.

Tabla 10.2: Selección de componentes

Módulo	Dispositivo	Fabricante	Grado	Herencia de vuelo
Unidad maestra	Artix 7 FPGA	Xilinx	COTS*	Sí
Microcontrolador supervisor	MSP430FR5969	TI	COTS*	Sí
Memoria de respaldo y configuración	CY15B116QSN	Infineon Technologies	Automotriz	Sí
Memoria FLASH	IS25LE01G	ISSI	COTS	-
FRAM	CY15B104Q-SXI	Infineon Technologies	COTS	Sí
Giroscopio	LSM6DSV16XTR	STMicroelectronics	COTS	-
Magnetómetro	MMC34160PJ	MEMSIC	COTS	-
Sensor de temperatura	TMP117AIDRVR	TI	COTS	-

* Dispositivo que cuenta con una versión de grado espacial.

11 IMPLEMENTACIÓN DE LA ARQUITECTURA

Continuando con la metodología de diseño, se han obtenido las especificaciones que se deben de cumplir y, a partir de ellas, se realizó el diseño de detalle del SCMI, el cual se analizó para obtener matemáticamente su nivel de fiabilidad. Posteriormente, llega el momento de construir la arquitectura propuesta para validar su funcionamiento y demostrar de manera práctica su correcto funcionamiento. En este capítulo se presenta la implementación del SCMI construido en *Vivado Design Suite*, respetando la definición de los módulos internos descrita previamente. Además, este proceso permitirá realizar la estimación real de los recursos lógicos necesarios para su implementación y el consumo energético del sistema.

11.1 Equipo de trabajo

Antes de presentar la arquitectura del SCMI es importante mencionar que, debido a la complejidad y a la alta cantidad de tareas técnicas necesarias para el diseño del sistema, fue necesario formar un equipo de trabajo con estudiantes del LIESE que instrumentaron algunos bloques de la unidad maestra del SCMI, siguiendo con la filosofía de diseño a bajo nivel del LIESE para conseguir una alta eficiencia en recursos lógicos y tener un control total sobre la arquitectura, de modo que permita analizar SPOF para aplicar técnicas de tolerancia a fallas que incrementen el nivel de fiabilidad del SCMI. Con esto en mente, el proyecto se está desarrollando respetando el organigrama de la figura 11.1, llevando reuniones semanales para analizar los resultados alcanzados y definir nuevas tareas técnicas.

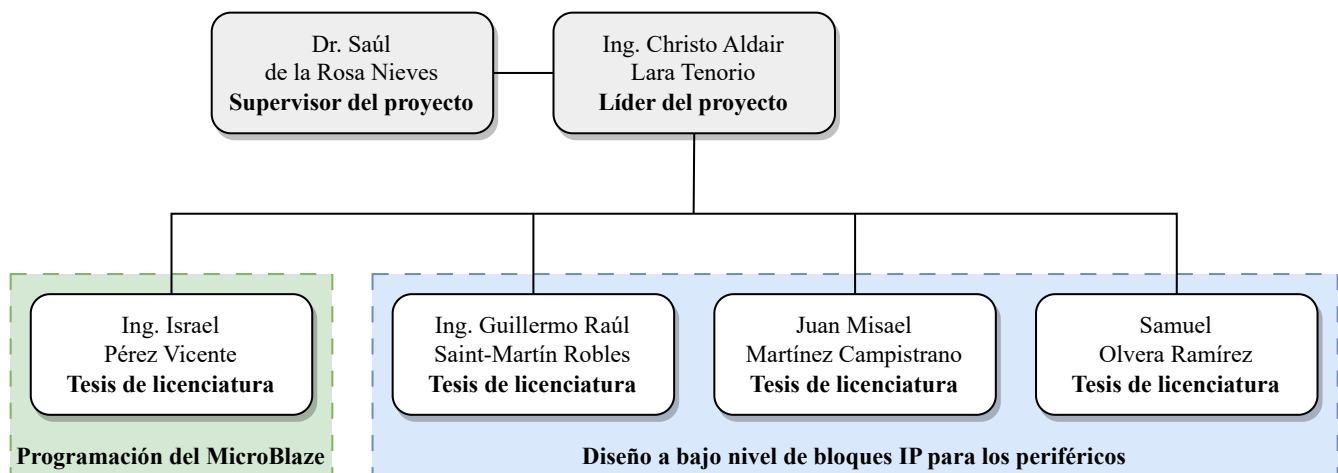


Figura 11.1: Organigrama del equipo de trabajo

Cabe destacar que el trabajo de los miembros del equipo fue lo suficientemente robusto para que desarrollaran sus tesis de licenciatura, de los cuales dos miembros la presentaron exitosamente y actualmente se encuentran cursando su posgrado en ingeniería eléctrica, mientras que los dos restantes se encuentran en proceso de titulación para presentar su trabajo.

Por otro lado, el SCMI continua en desarrollo en el LIESE bajo el mismo esquema de trabajo, con el apoyo de más estudiantes que están llevando a cabo tareas de investigación para implementar las funcionalidades del SCMI que aun no están cubiertas con este trabajo de tesis (capítulo 15).

11.2 Implementación del SCMI en Vivado Design Suite

En esta sección únicamente se presentan las capturas de la arquitectura de la unidad maestra construida en *Vivado Design Suite*, estructurada de la misma manera que el diseño de detalle del SCMI (figura 10.4).

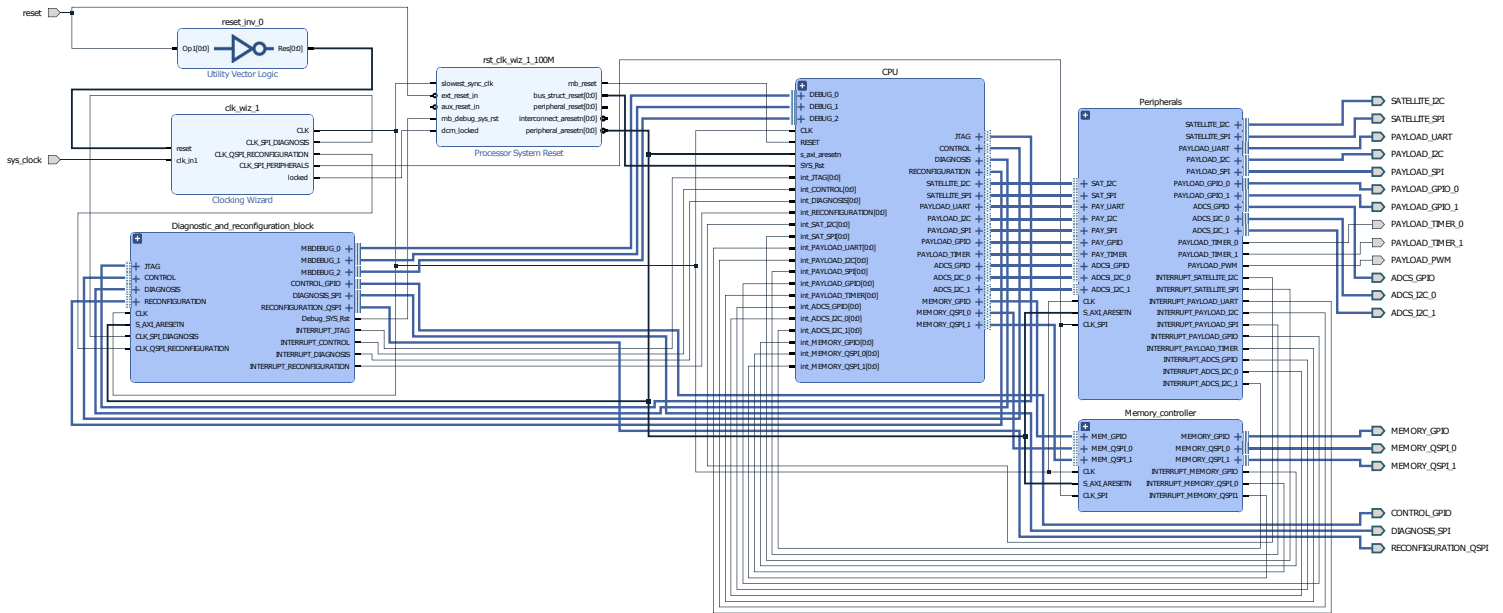


Figura 11.2: Implementación de la unidad maestra

Para la construcción de la CPU se utilizó TMR a nivel de procesador, empleando el *softcore MicroBlaze*, tomando como referencia el subsistema de un solo *MicroBlaze* de la figura 11.3, en donde se tiene el módulo de *debug* del *MicroBlaze* (MDM, por sus siglas en inglés) para la programación del procesador; y a los buses de memoria local (LMB, por sus siglas en inglés) de datos e instrucciones, utilizados como interfaces entre el *MicroBlaze* y los módulos de entrada/salida y los BRAM.

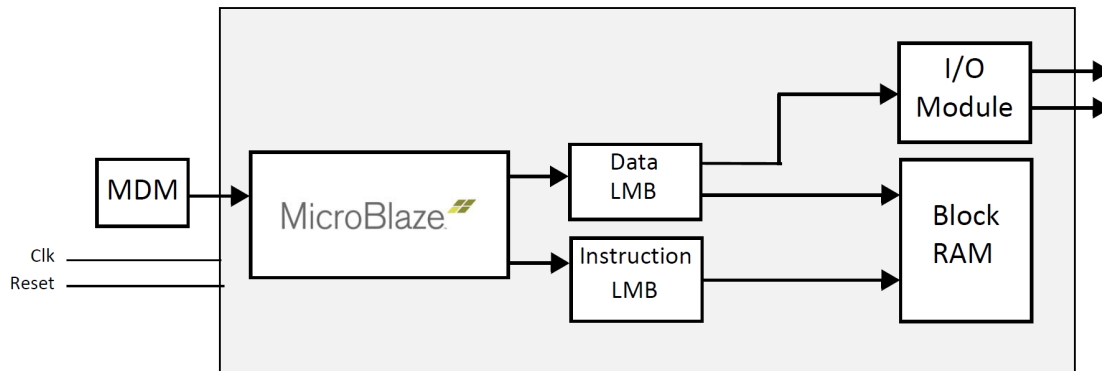


Figura 11.3: Subsistema de microcontrolador con *MicroBlaze* [71]

A partir de esto, la figura 11.4 describe cómo aplicar TMR en el *MicroBlaze*, triplicando el esquema de la figura 11.3 para comparar el acceso a los BRAM de los tres núcleos (tanto para los datos como para las instrucciones) y a los módulos de entrada/salida.

En el esquema de la figura 11.4, los datos entregados por los módulos de entrada/salida son procesados por un dictaminador de mayoría. Sin embargo, anteriormente se mencionó que los periféricos del SCMI se están desarrollando a bajo nivel como bloques IP propios, los cuales serán analizados posteriormente para ubicar SPOF y aplicar tolerancia a fallas, lo que resultará en un consumo menor de recursos al aplicar redundancia en zonas específicas de los periféricos

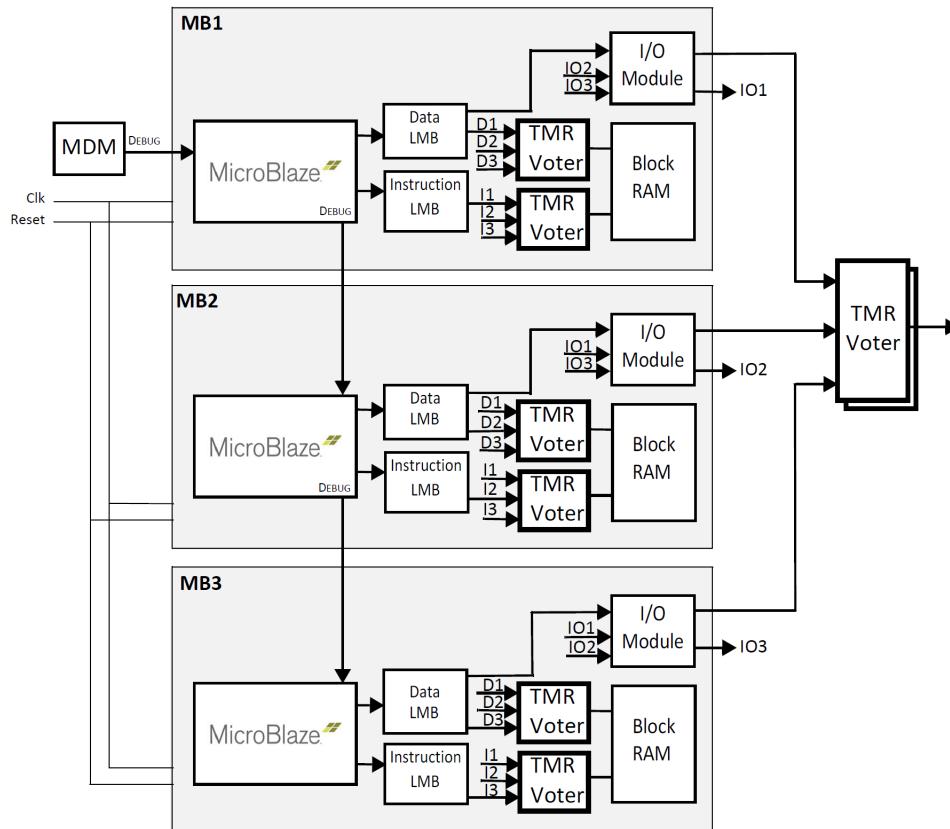


Figura 11.4: Subsistema tolerante a fallas de *MicroBlaze* con TMR [71]

ya no triplicar todo el bloque IP. Con base en esto, únicamente se triplicarán los bloques correspondientes al acceso a los BRAM y no en los módulos de entrada/salida. En la figura 11.5 se presenta a la CPU con la arquitectura de microcontrolador con *MicroBlaze* triplicada (CPU_0, CPU_1 y CPU_2), conectadas al mismo controlador de interrupciones (*microblaze_0_axi_intc*) e ingresando sus salidas a un dictaminador de mayoría (*tmr_voter_0*) para controlar todos los periféricos (*smartconnect_0*).

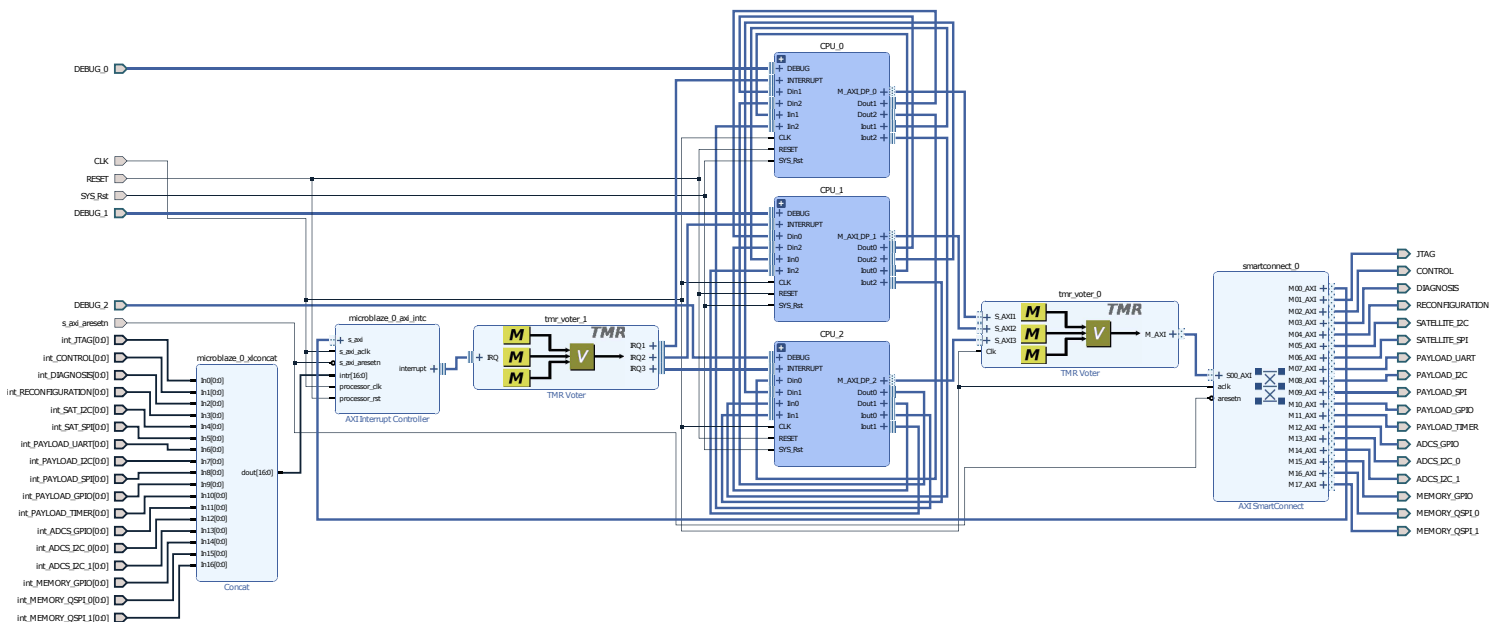


Figura 11.5: Implementación de la CPU

Dentro de los bloques triplicados (CPU_0, CPU_1 y CPU_2) se construyó la arquitectura base de microcontrolador con *MicroBlaze* (figura 11.6), conectando al procesador con el bloque de memoria local (microblaze_0_local_memory).

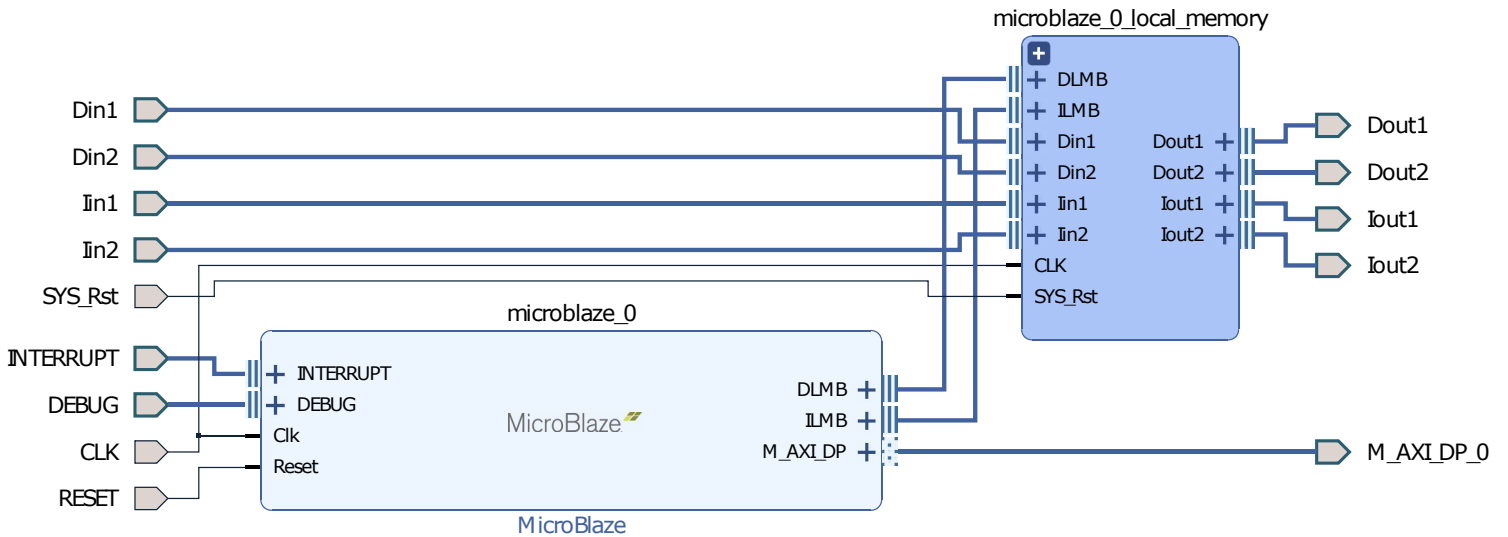


Figura 11.6: Arquitectura base de microcontrolador con *MicroBlaze*

Dentro del bloque de memoria local se tiene el BRAM, triplicando las salidas de los LMB de datos (dlmb) y de instrucciones (ilmb) para que ingresen a los dictaminadores de mayoría (tmr_voter_0 y tmr_voter_1) del bloque actual y con los bloques CPU_1 y CPU_2 para ingresar al BRAM correspondiente (lmb_bram). Lo descrito anteriormente se presenta en la figura 11.7.

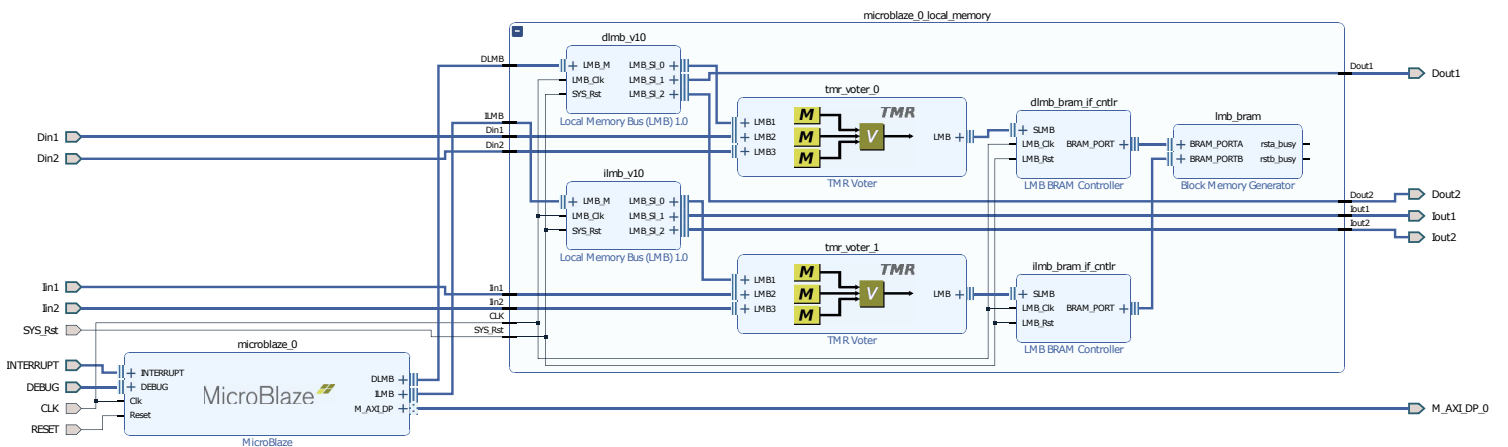


Figura 11.7: Arquitectura base de microcontrolador con *MicroBlaze* y el BRAM

Continuando con el bloque de diagnóstico y reconfiguración, este bloque integra un MDM (mdm_0) encargado de la programación de todos los *MicroBlaze*, un bloque de puertos GPIO (axi_gpio_0) utilizados para el control de la unidad maestra desde la unidad de supervisión, un SPI (axi_quad_spi_0) para el intercambio de datos para diagnóstico con el microcontrolador supervisor y un QSPI (axi_quad_spi_1) destinado al microcontrolador supervisor y a la memoria de respaldo y configuración. Lo descrito anteriormente se muestra en la figura 11.8.

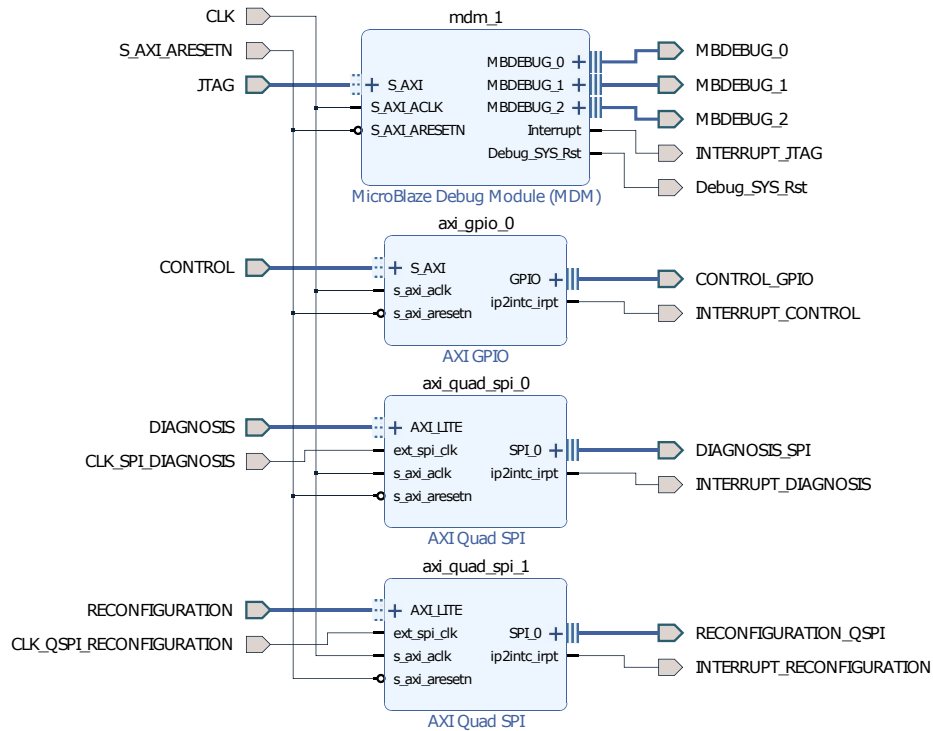


Figura 11.8: Implementación del bloque de diagnóstico y reconfiguración

Posteriormente, se presenta el bloque de periféricos, divididos en el bus del satélite (*Satellite_bus*), la carga útil (*Payload*) y el ADCS (ADCS), como se presenta en la figura 11.9.

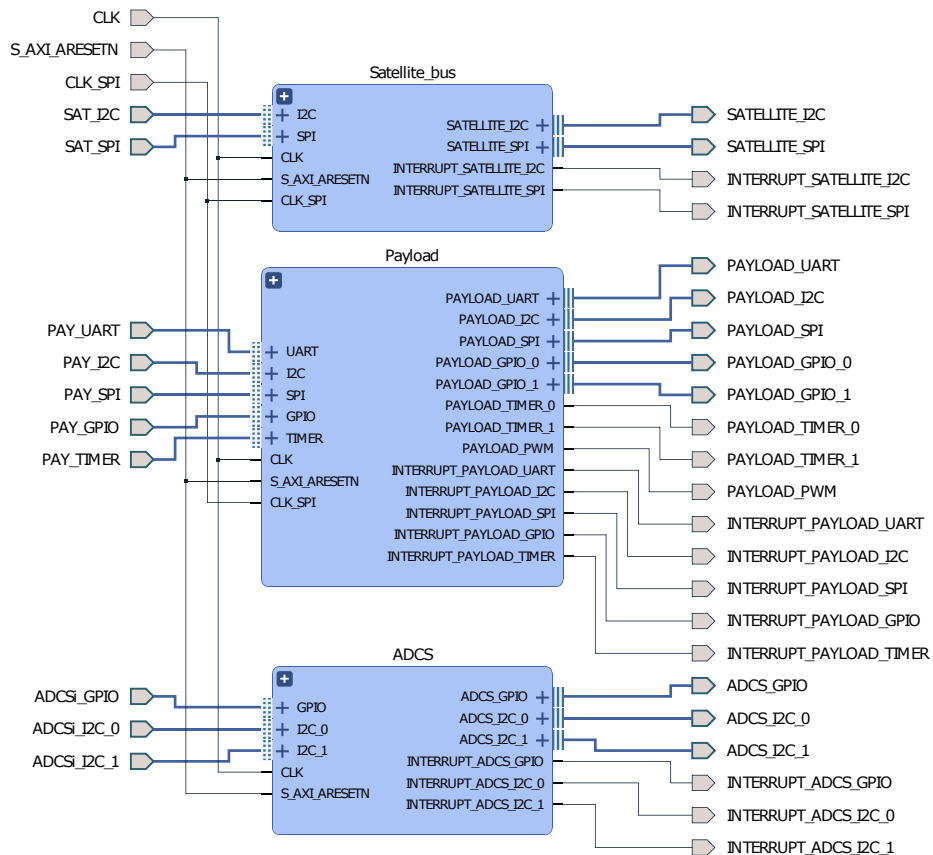


Figura 11.9: Implementación del bloque de periféricos

En las figuras 11.10, 11.11 y 11.12 se presentan los bloques IP gratuitos ofrecidos por *Xilinx*, faltando aquellos que requieren licencia (CAN y USB). Sin embargo, es importante aclarar que los bloques de UART, I2C y SPI a bajo nivel propios se encuentran en la etapa de pruebas y depuración, usando por ahora los de terceros como referencia para la estimación de los recursos lógicos necesarios para implementar a la unidad maestra del SCMI.

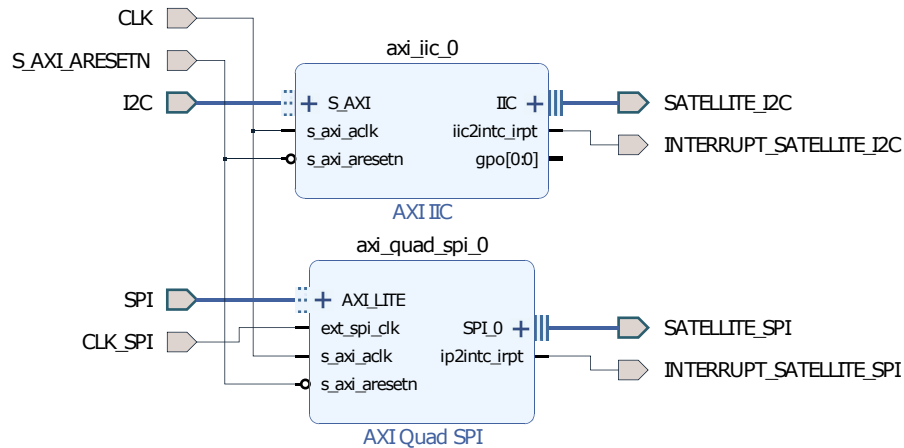


Figura 11.10: Implementación de los periféricos para el bus del satélite

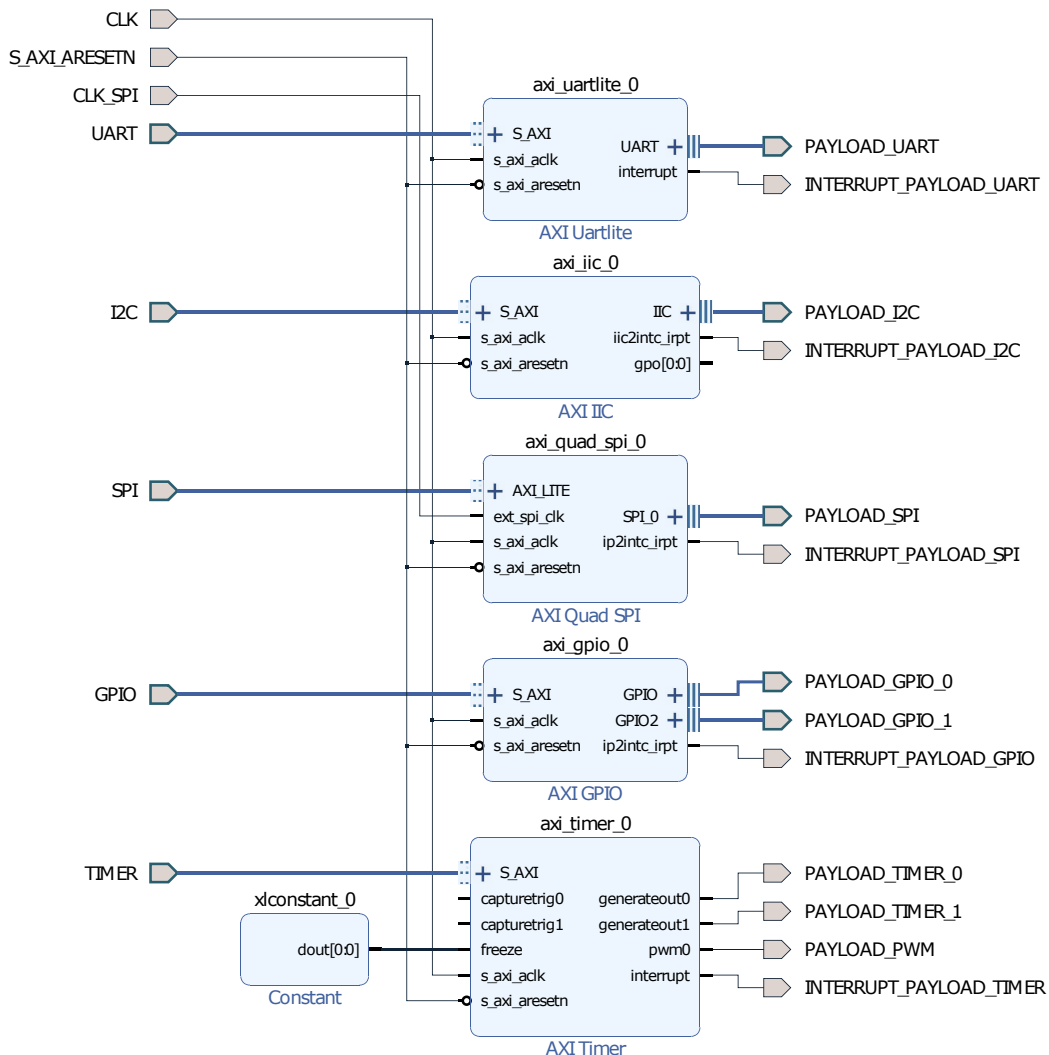


Figura 11.11: Implementación de los periféricos para la carga útil

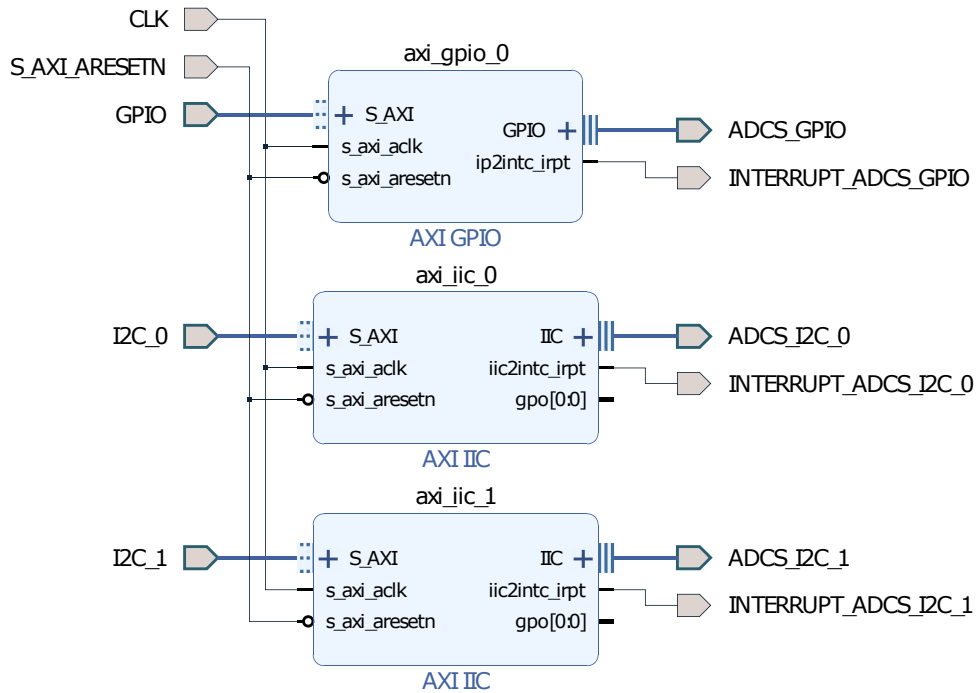


Figura 11.12: Implementación de los periféricos para el ADCS

Por último, se presenta el controlador de memorias en la figura 11.13 con los periféricos necesarios para control (`axi_gpio_0`), las memorias FLASH para el almacenamiento masivo de datos (`axi_quad_spi_0`) y las FRAM para el almacenamiento crítico de datos (`axi_quad_spi_1`).

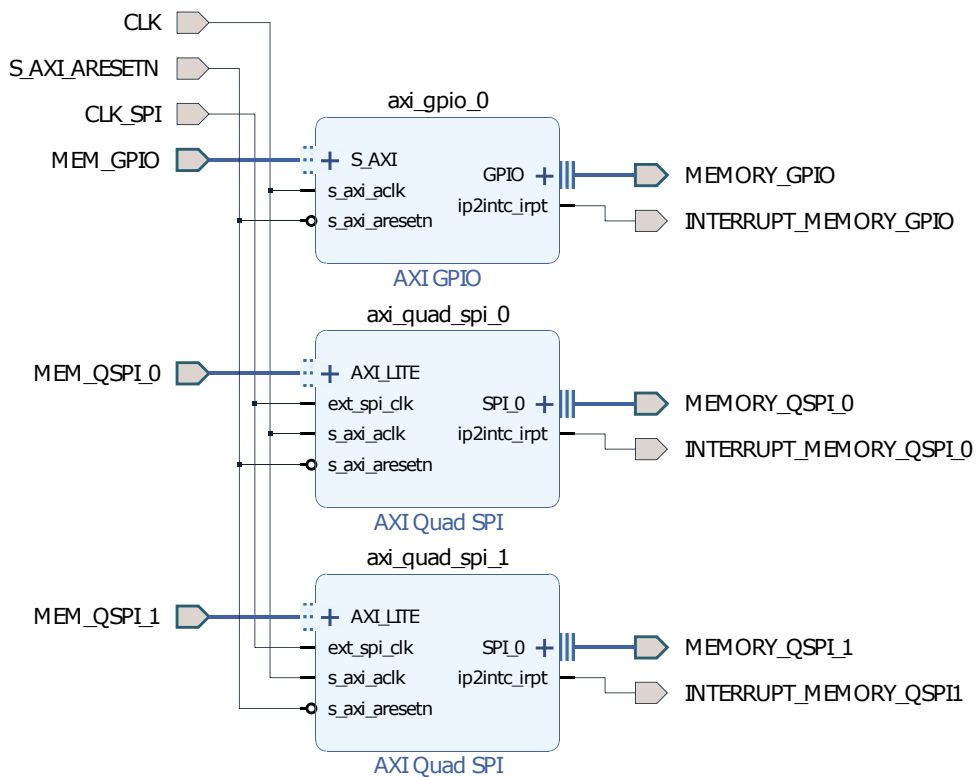


Figura 11.13: Implementación del controlador de memorias

11.3 Resumen de la implementación de la unidad maestra en Vivado Design Suite

Para la implementación de la unidad maestra se utilizó la tarjeta de desarrollo *Nexys Video* (figura 11.14), que utiliza un FPGA *Artix-7 XC7A200T-1SBG484C* de *Xilinx*, el cual es el dispositivo con mayor capacidad de la familia *Artix-7*.

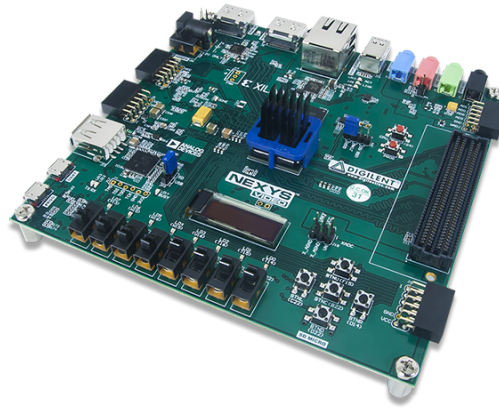
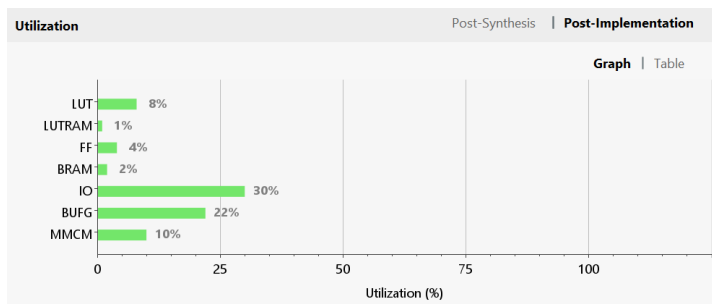


Figura 11.14: Tarjeta de desarrollo *Nexys Video*

En las figuras 11.15 y 11.16 se presentan los reportes de utilización de recursos y la estimación de consumo energético, respectivamente, de *Vivado Design Suite* para la arquitectura implementada.



(a) Gráfica de utilización de recursos

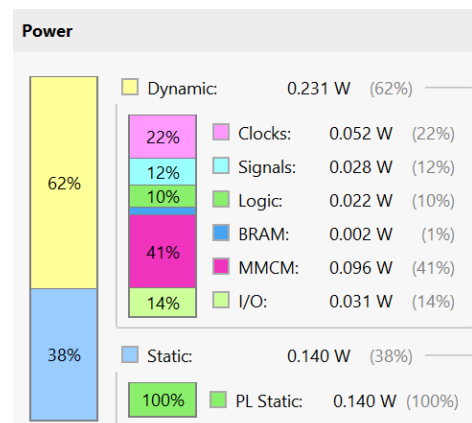
Resource	Utilization	Available	Utilization %
LUT	10732	133800	8.02
LUTRAM	539	46200	1.17
FF	11598	269200	4.31
BRAM	6	365	1.64
IO	85	285	29.82
BUFG	7	32	21.88
MMCM	1	10	10.00

(b) Tabla de utilización de recursos

Figura 11.15: Reporte de utilización de recursos – *post-implementation*

Power	
Total On-Chip Power:	0.371 W
Junction Temperature:	26.2 °C
Thermal Margin:	58.8 °C (17.4 W)
Effective θ_{JA} :	3.3 °C/W
Power supplied to off-chip devices:	0 W
Confidence level:	Low
Implemented Power Report	

(a) Resumen del consumo energético



(b) Consumo energético *On-Chip*

Figura 11.16: Reporte de consumo energético – *post-implementation*

Por último, se presenta el *logic placement* de los elementos sintetizados en *Vivado Design Suite* para la unidad maestra del SCMI. El *logic placement* será útil para el trabajo a futuro, en donde se definirán las zonas del FPGA en donde se implementarán los módulos, así como sus elementos de interfaz para realizar *scrubbing prioritario* y DPR.

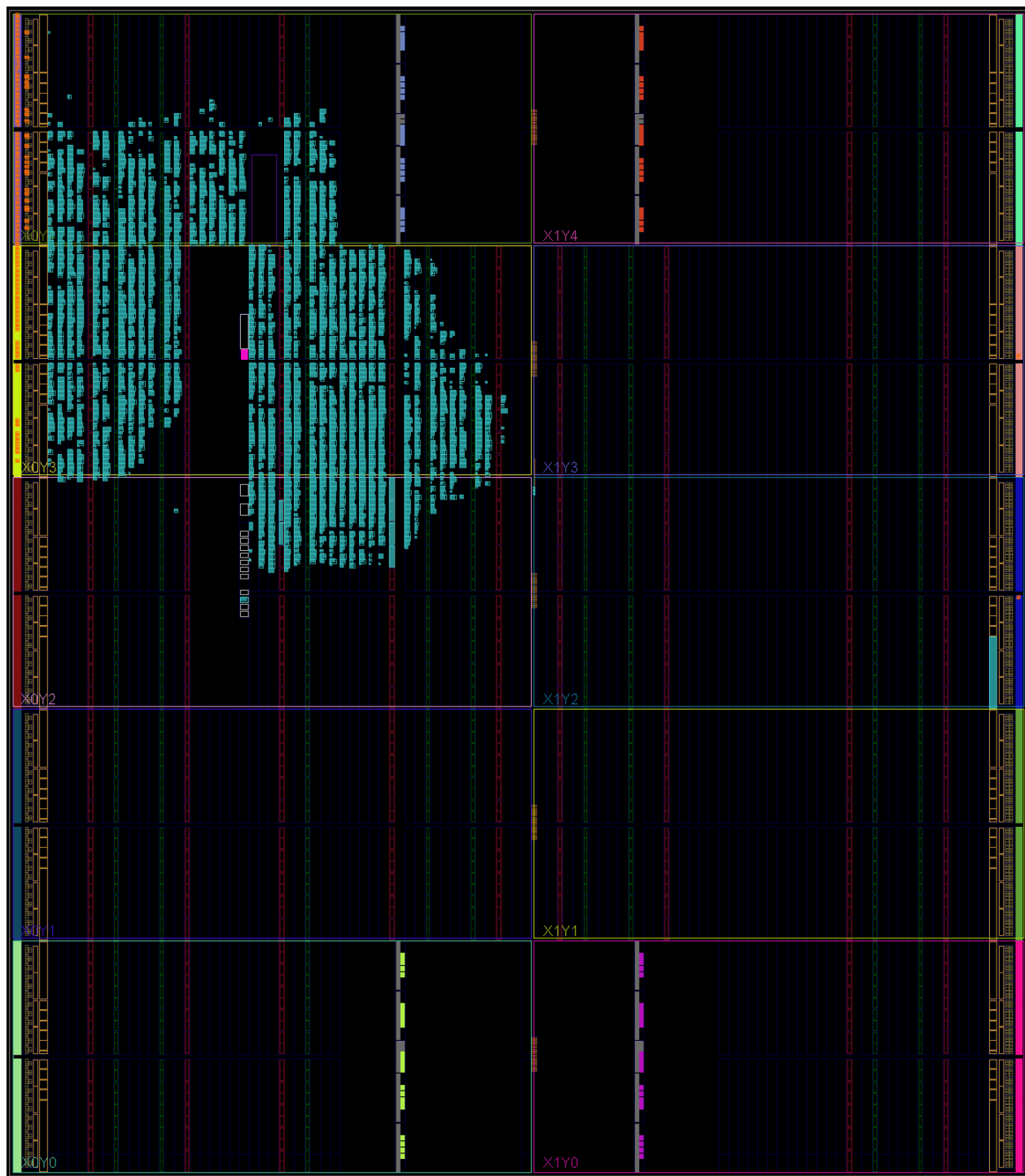


Figura 11.17: *Logic Placement* de la unidad maestra en un Artix-7 (XC7A200T-1SBG484C)

12 ESTIMACIÓN DEL NIVEL DE FIABILIDAD

Una vez que se tiene el diseño de detalle del sistema surge la pregunta ¿el diseño cumple con los parámetros necesarios para operar correctamente en el entorno seleccionado? Para este proyecto, el entorno es el medio ambiente de radiación espacial y el principal parámetro que se debe de cumplir es el nivel de fiabilidad. En este capítulo, se realiza el análisis matemático del esquema de tolerancia a fallas propuesto para la unidad maestra, de modo que se alcance el nivel de fiabilidad adecuado para que el SCMI desempeñe misiones en LEO.



Para iniciar con el análisis de fiabilidad de la unidad maestra del SCMI, se realizó el modelo de Markov (figura 12.1) de la CPU del sistema aplicando TMR y reconfiguración global del FPGA, destacando en color azul los cambios de estado correspondientes a la recuperación del sistema para regresar al estado perfecto S_0 y en rojo los cambios de estado resultados de la ocurrencia de una falla en uno de los módulos triplicados.

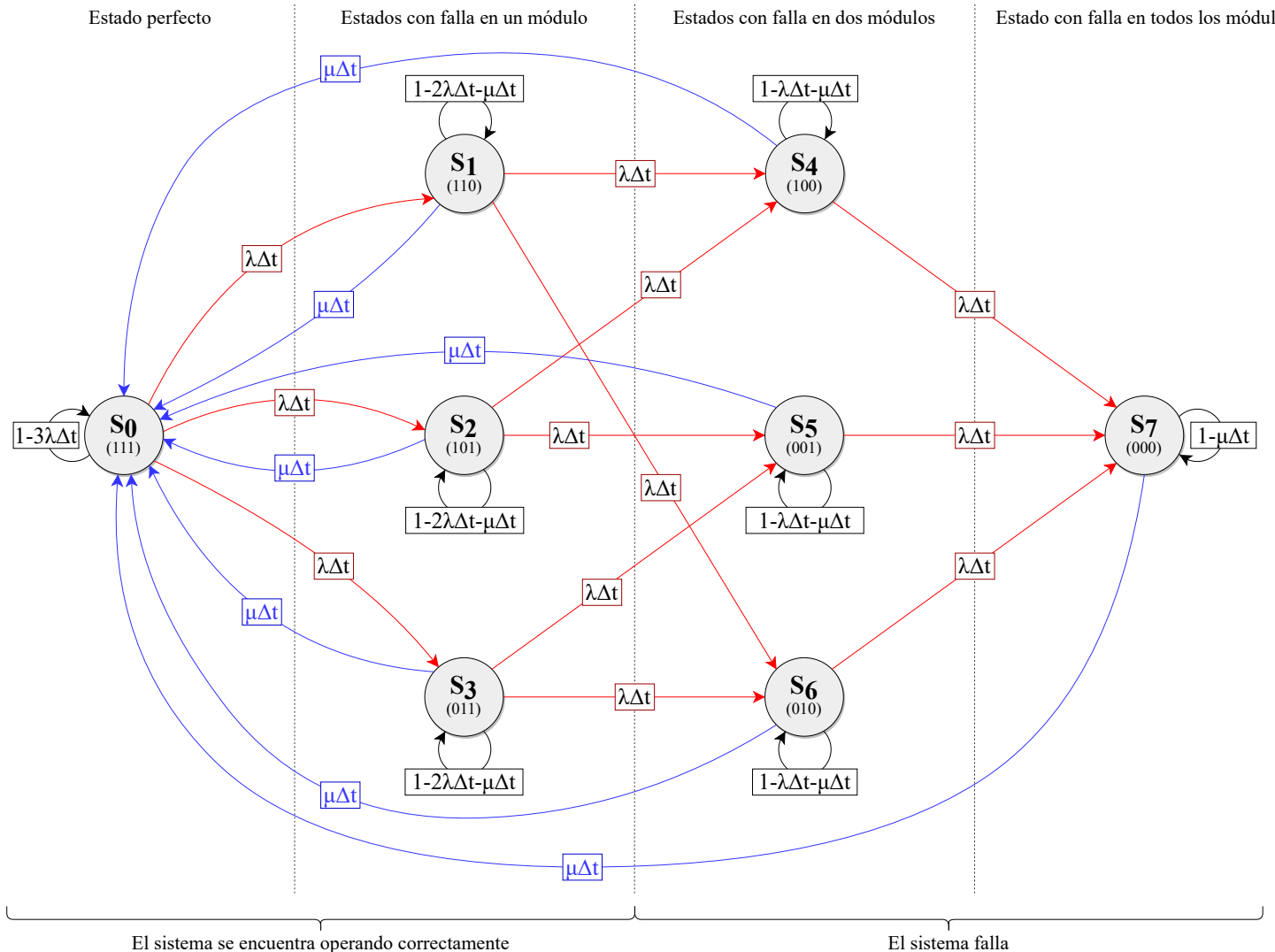


Figura 12.1: Modelo de Markov para el CPU del SCMI

En la figura 12.2 se presenta el modelo de Markov reducido a cuatro estados, en donde el SCMI operará correctamente mientras la CPU se mantenga en los estados S_0 y S_1 y presentará un malfuncionamiento al estar en los estados S_2 y S_3 , siempre y cuando no se hayan realizado tareas correctivas para regresar al estado perfecto S_0 .

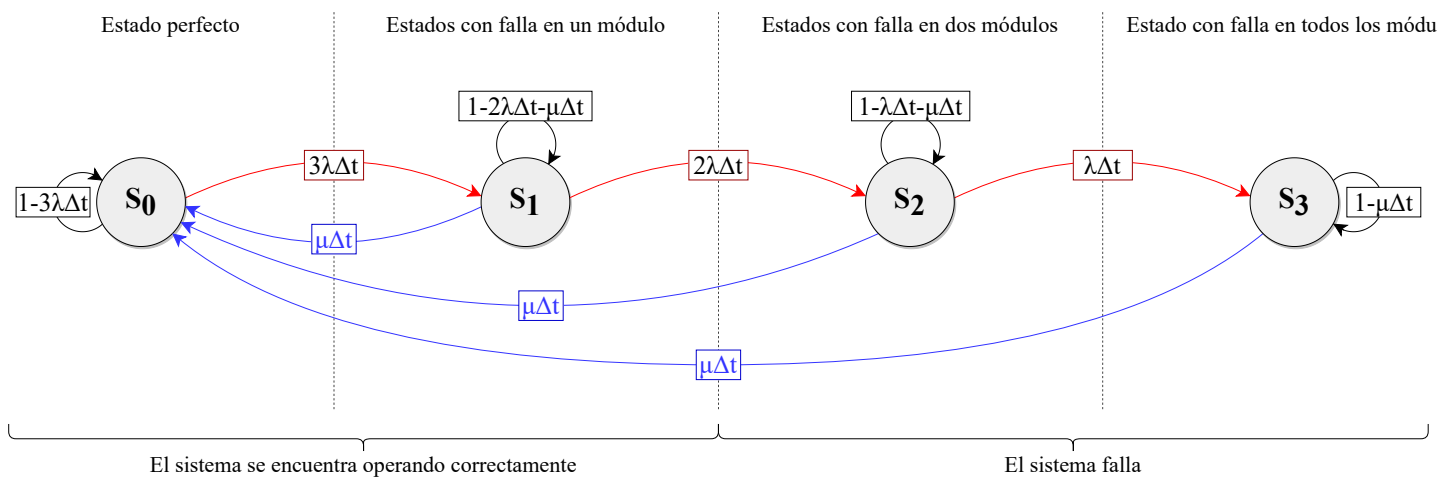


Figura 12.2: Modelo de Markov reducido para el CPU del SCMI

Desde este momento es claro que la reconfiguración global aumenta el nivel de fiabilidad del SCMI al añadir la capacidad de regresar al estado perfecto. Analizando la transición de estados, el SCMI alcanzará el estado S_2 únicamente cuando se presenten dos fallas continuas dentro de un intervalo de tiempo más pequeño que el periodo entre recuperación del sistema y, con mucho menor probabilidad, alcanzando el estado S_3 cuando ocurran tres fallas continuas dentro de este mismo intervalo de tiempo.

Para determinar las probabilidades de que el sistema se encuentre en cada uno de los estados se analiza el modelo de Markov de la figura 12.2, del cual se obtiene el sistema de ecuaciones 12.1.

$$\begin{aligned}
 P_0(t + \Delta t) &= (1 - 3\lambda\Delta t)P_0(t) + \mu\Delta tP_1(t) + \mu\Delta tP_2(t) + \mu\Delta tP_3(t) \\
 P_1(t + \Delta t) &= 3\lambda\Delta tP_0(t) + (1 - 2\lambda\Delta t - \mu\Delta t)P_1(t) \\
 P_2(t + \Delta t) &= 2\lambda\Delta tP_1(t) + (1 - \lambda\Delta t - \mu\Delta t)P_2(t) \\
 P_3(t + \Delta t) &= \lambda\Delta tP_2(t) + (1 - \mu\Delta t)P_3(t)
 \end{aligned}
 \tag{12.1}$$

Al arreglar algebraicamente el sistema 12.1 se llegó a 12.2.

$$\begin{aligned}
 \frac{P_0(t + \Delta t) - P_0(t)}{\Delta t} &= -3\lambda P_0(t) + \mu P_1(t) + \mu P_2(t) + \mu P_3(t) \\
 \frac{P_1(t + \Delta t) - P_1(t)}{\Delta t} &= 3\lambda P_0(t) - 2\lambda P_1(t) - \mu P_1(t) \\
 \frac{P_2(t + \Delta t) - P_2(t)}{\Delta t} &= 2\lambda P_1(t) - \lambda P_2(t) - \mu P_2(t) \\
 \frac{P_3(t + \Delta t) - P_3(t)}{\Delta t} &= \lambda P_2(t) - \mu P_3(t)
 \end{aligned}
 \tag{12.2}$$

Considerando que $\Delta t \rightarrow 0$, las ecuaciones diferenciales resultantes están dadas por 12.3.

$$\begin{aligned}\frac{dP_0(t)}{dt} &= -3\lambda P_0(t) + \mu P_1(t) + \mu P_2(t) + \mu P_3(t) \\ \frac{dP_1(t)}{dt} &= 3\lambda P_0(t) - 2\lambda P_1(t) - \mu P_1(t) \\ \frac{dP_2(t)}{dt} &= 2\lambda P_1(t) - \lambda P_2(t) - \mu P_2(t) \\ \frac{dP_3(t)}{dt} &= \lambda P_2(t) - \mu P_3(t)\end{aligned}\tag{12.3}$$

Para resolver el sistema 12.3, este puede ser expresado de la siguiente forma:

$$\frac{d}{dt}P(t) = A(\lambda, \mu) \times P(t)\tag{12.4}$$

Que se puede expresar con matrices:

$$\frac{d}{dt} \begin{bmatrix} P_0(t) \\ P_1(t) \\ P_2(t) \\ P_3(t) \end{bmatrix} = A(\lambda, \mu) \times \begin{bmatrix} P_0(t) \\ P_1(t) \\ P_2(t) \\ P_3(t) \end{bmatrix}\tag{12.5}$$

Al sustituir $A(\lambda, \mu)$, se obtiene:

$$\frac{d}{dt} \begin{bmatrix} P_0(t) \\ P_1(t) \\ P_2(t) \\ P_3(t) \end{bmatrix} = \begin{bmatrix} -3\lambda & \mu & \mu & \mu \\ 3\lambda & -2\lambda - \mu & 0 & 0 \\ 0 & 2\lambda & -\lambda - \mu & 0 \\ 0 & 0 & \lambda & -\mu \end{bmatrix} \times \begin{bmatrix} P_0(t) \\ P_1(t) \\ P_2(t) \\ P_3(t) \end{bmatrix}\tag{12.6}$$

Cuya solución es:

$$\begin{aligned}P_0(t) &= \frac{\mu + 3\lambda e^{-t(3\lambda+\mu)}}{3\lambda + \mu} \\ P_1(t) &= \frac{3\lambda e^{-t(3\lambda+\mu)}(6\lambda e^{\lambda t} - 3\mu - 6\lambda + 2\mu e^{\lambda t} + \mu e^{t(3\lambda+\mu)})}{6\lambda^2 + 5\lambda\mu + \mu^2} \\ P_2(t) &= \frac{3\lambda e^{-t(3\lambda+\mu)}(9\lambda\mu + 6\lambda^2 + 3\mu^2 - 12\lambda^2 e^{\lambda t} + 6\lambda^2 e^{2\lambda t} - 4\mu^2 e^{\lambda t} + \mu^2 e^{2\lambda t} + 2\lambda\mu e^{t(3\lambda+\mu)} - 16\lambda\mu e^{\lambda t} + 5\lambda\mu e^{2\lambda t})}{6\lambda^3 + 11\lambda^2\mu + 6\lambda\mu^2 + \mu^3} \\ P_3(t) &= -\frac{3\lambda e^{-t(3\lambda+\mu)}(3\lambda\mu + 2\lambda^2 + \mu^2 - 6\lambda^2 e^{\lambda t} + 6\lambda^2 e^{2\lambda t} - 2\mu^2 e^{\lambda t} + \mu^2 e^{2\lambda t} - 2\lambda^2 e^{t(3\lambda+\mu)} - 8\lambda\mu e^{\lambda t} + 5\lambda\mu e^{2\lambda t})}{6\lambda^3 + 11\lambda^2\mu + 6\lambda\mu^2 + \mu^3}\end{aligned}\tag{12.7}$$

12.1 Cálculo de la intensidad de fallas (λ)

Recordando que la intensidad de fallas (λ) es igual al número de fallas en una unidad de tiempo, se necesita conocer el número de fallas que tendrá el sistema en un tiempo determinado. Para esto se calcula el tiempo entre eventos (12.8) [72], en donde σ representa a la sección transversal y ϕ a la fluencia de partículas.

$$\text{tiempo entre eventos} = \frac{1}{4\pi\sigma\phi} \quad (12.8)$$

Para obtener la sección transversal (σ) transversal, es necesario conocer el área real que ocupa la unidad maestra en el FPGA, sin embargo, este dato no se reporta directamente en *Vivado*, por lo que se analizó el *bitstream* del FPGA. Para esto, se parte del concepto de los bits esenciales, definidos como un subconjunto de los bits de configuración del dispositivo asociados con el diseño del circuito [73], de modo que si son perturbados por un SEU puede cambiar el diseño del circuito, lo que significa que permiten determinar la proporción real del FPGA que está en uso por el diseño. En la figura 12.3 se presenta la relación de los bits esenciales, con respecto a los bits de configuración del dispositivo.



Figura 12.3: Relación de los bits de configuración del FPGA [73]

Para determinar la cantidad de bits esenciales en la arquitectura implementada, *Vivado* ofrece múltiples opciones de configuración que entregan archivos *readback* del diseño, de los cuales se generó el *.ebd*, para el cual se programó un *script* en *Python* para contabilizar la cantidad de bits esenciales y la proporción de bits utilizados para el diseño (véase la figura 12.4).

```
Script en Python que contabiliza los 0's y 1's del
archivo SCMI_wrapper.ebd de la implementación en
Vivado Design Suite para determinar la cantidad de
bits esenciales y la proporción de bits utilizados
para la implementación del diseño.
```

SCMI_wrapper.ebd	Bits	Porcentaje
# of bits	61104192	100.00
# of 0's	58862965	96.33
# of 1's	2241227	3.67

Figura 12.4: Resultados del *script* realizado con *Python*

Considerando que se seleccionó un FPGA COTS de la serie 7 de *Xilinx*, estos dispositivos utilizan dos tipos de tecnología para las celdas SRAM [74], la principal diferencia entre ambas tecnologías es el área de la celda: siendo $0.155\mu\text{m}^2$ para la LP¹⁶ y $0.127\mu\text{m}^2$ para la HP¹⁷ [75], se realizó una aproximación considerando el peor de los casos (las celdas SRAM con mayor área). En 12.9 se realiza el cálculo de la sección transversal.

$$\begin{aligned}\sigma &= (\text{numero de bits esenciales utilizados})(A_{SRAM}) \\ \sigma &= (2\ 241\ 227)(0.155\mu\text{m}^2) \\ \sigma &= 347\ 390.185\mu\text{m}^2\end{aligned}\tag{12.9}$$

Posteriormente, se determinó el flujo integral de partículas a partir de la simulación realizada en SPENVIS (figura 7.6), para lo cual se necesita el LETth del dispositivo. Sin embargo, el LETth varía en función del ángulo de incidencia de las partículas cargadas que impactan el material y, debido a que el espacio presenta un flujo de partículas omnidireccional, se consideraron los tres valores de LETth estudiados en [76] para las celdas de SRAM de 28nm , que es el tipo de tecnología empleado en los FPGA de la serie 7 de *Xilinx*. En resumen, en la figura 12.5 se observa el flujo integral correspondiente para los valores del LETth de $3\text{MeV}\cdot\text{cm}^2\cdot\text{g}^{-1}$ (rojo), $5.21\text{MeV}\cdot\text{cm}^2\cdot\text{g}^{-1}$ (azul) y $9.13\text{MeV}\cdot\text{cm}^2\cdot\text{g}^{-1}$ (verde).

$$\begin{aligned}LET_{th1} &= 3\frac{\text{MeV}\text{cm}^2}{\text{g}} = 3000\frac{\text{MeV}\text{cm}^2}{\text{mg}} \approx 10^{3.5} \\ LET_{th2} &= 5.21\frac{\text{MeV}\text{cm}^2}{\text{g}} = 5210\frac{\text{MeV}\text{cm}^2}{\text{mg}} \approx 10^{3.7} \\ LET_{th3} &= 9.13\frac{\text{MeV}\text{cm}^2}{\text{g}} = 9130\frac{\text{MeV}\text{cm}^2}{\text{mg}} \approx 10^{3.9}\end{aligned}\tag{12.10}$$

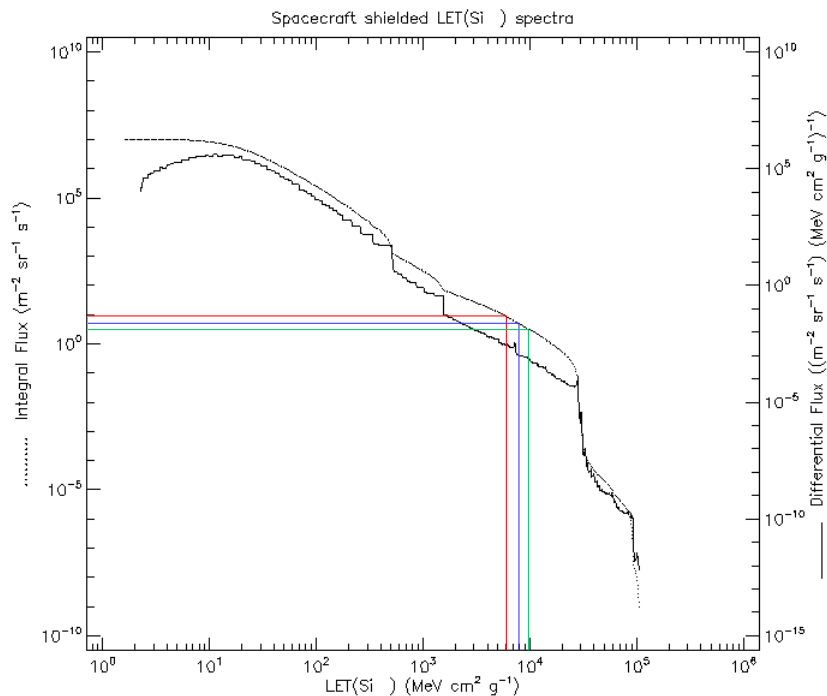


Figura 12.5: Espectro LET del flujo de partículas incidentes para los tres casos de estudio

¹⁶Proceso de optimización para aplicaciones de bajo consumo.

¹⁷Proceso optimizado para aplicaciones de alto desempeño.

$$\begin{aligned}
\phi_1 &\approx 10^1 \frac{\text{iones}}{m^2 \text{srg}} \\
\phi_2 &\approx 10^{0.7} \frac{\text{iones}}{m^2 \text{srg}} \\
\phi_3 &\approx 10^{0.4} \frac{\text{iones}}{m^2 \text{srg}}
\end{aligned} \tag{12.11}$$

Con lo anterior, en la tabla 12.1 se presenta el tiempo entre eventos y la intensidad de fallas para cada escenario (a partir de las ecuaciones 12.8 y 12.12, respectivamente).

$$\lambda = \frac{1}{\text{tiempo entre eventos}} \tag{12.12}$$

Tabla 12.1: Determinación del tiempo entre upsets y la intensidad de fallas

Descripción	Parámetro	Caso 1	Caso 2	Caso 3	Unidades
Umbral LET	LET_{th}	3	5.21	9.13	$\frac{MeVcm^2}{g}$
Fluencia de partículas	ϕ	10^1	$10^{0.7}$	$10^{0.4}$	$\frac{\text{iones}}{m^2 \text{srg}}$
Tiempo entre eventos	-	22.9072	45.7059	91.1953	$\frac{ms}{\text{evento}}$
Intensidad de fallas	λ	0.04365439687	0.02187901343	0.01096547739	$\frac{\text{evento}}{ms}$

En la tabla 12.1 se observa una atenuación en la intensidad de fallas al aumentar el LET_{th}, debido a que la energía de las partículas debe de ser mayor para inducir una falla en la estructura interna del dispositivo, incrementando el tiempo entre eventos.

12.2 Cálculo de la intensidad de reparación (μ)

Para el cálculo de la intensidad de reparación se necesita conocer el tiempo necesario para reconfigurar al sistema y que este regrese a su estado funcional perfecto. Las tareas de reconfiguración se llevarán a cabo a través de un *scrubbing* global, empleando un modo de configuración esclavo serial (el cual es soportado por la serie 7 de *Xilinx* [77]).

Las arquitecturas FPGA y SoC de *Xilinx* tienen una memoria de configuración organizada en *frames*, siendo los segmentos direccionables más pequeños de la memoria de configuración del dispositivo. Analizando los manuales de configuración de *Xilinx*, los FPGA de la serie 7 soportan una frecuencia máxima de reloj para la configuración en modo serial esclavo de 100MHz (100Mb/s). Ahora, al considerar una configuración global del FPGA, se debe de cargar el archivo *bitstream* completo, cuyo tamaño real se puede obtener al generar el archivo *.rbt* con *Vivado*, siendo de 77 845 216 bits. Con esto, se calcula el tiempo de reconfiguración del FPGA (12.13).

$$\begin{aligned}
t_{conf} &= \frac{N_{bits}}{f_{conf}} \\
t_{conf} &= \frac{77\ 845\ 216}{100 \frac{Mb}{s}} \\
t_{conf} &= 778.4521ms
\end{aligned} \tag{12.13}$$

A partir de lo anterior, se obtiene la intensidad de reparación:

$$\begin{aligned}\mu &= \frac{1}{t_{conf}} \\ \mu &= \frac{1}{778,4521ms} \\ \mu &= 0.00128460055 \frac{eventos_{rec}}{ms}\end{aligned}\tag{12.14}$$

Para este caso se observa que el sistema no es fiable en este entorno de radiación, ya que el tiempo necesario para reconfigurar al FPGA es mucho mayor al tiempo entre eventos, lo que significa que la intensidad de falles es mucho mayor a la intensidad de recuperación ($\lambda \gg \mu$).

Parte V

RESULTADOS Y CONCLUSIONES

13 RESULTADOS

Como último paso del trabajo, se debe de analizar el comportamiento del sistema y los esquemas de tolerancia a fallas propuestos para incrementar el nivel de fiabilidad y validar que funcionan. En este capítulo se presentan los resultados de las simulaciones realizadas considerando diferentes escenarios de configuración y recuperación del FPGA.



13.1 Resultados del primer experimento

Como primer experimento se obtuvieron las gráficas en MATLAB para visualizar el comportamiento de la fiabilidad del sistema en función del tiempo, utilizando los parámetros de λ y μ obtenidos en el capítulo 12. Los resultados se presentan en la figura 13.1.

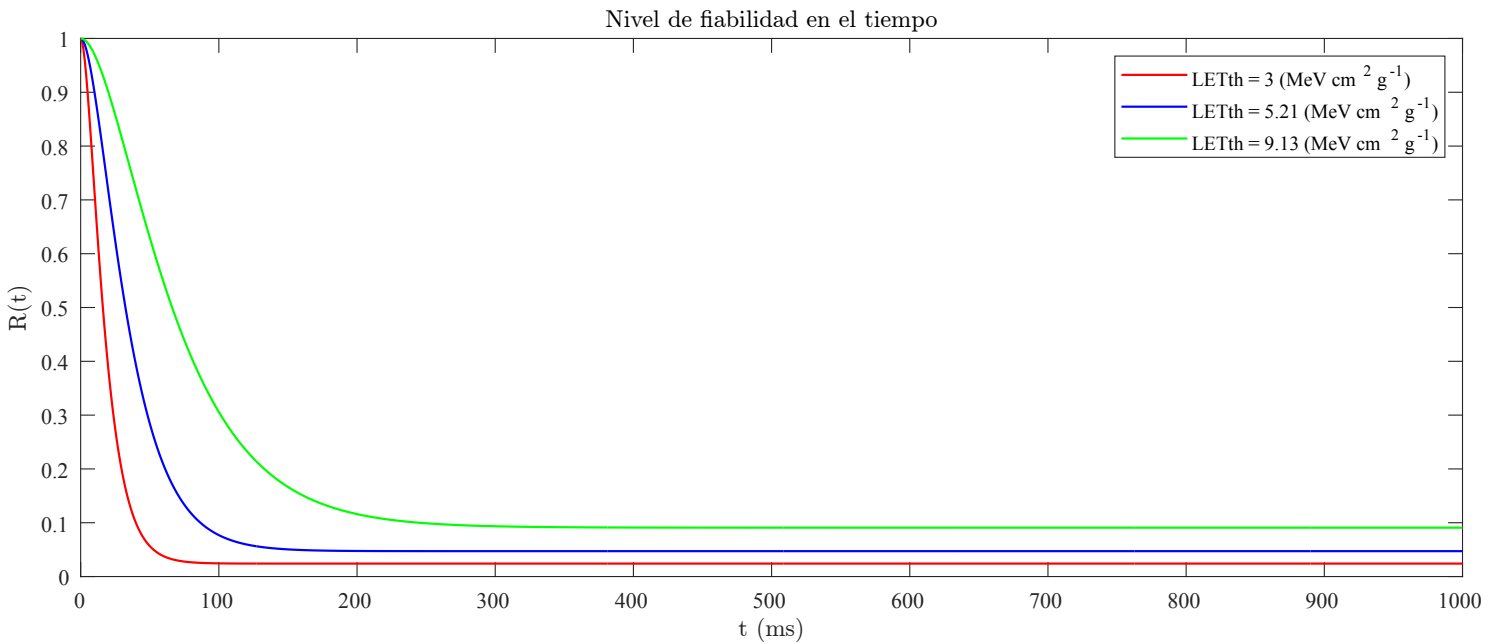


Figura 13.1: Resultados del análisis de fiabilidad para los tres casos de estudio

Como se comentó previamente, el nivel de fiabilidad baja con el paso del tiempo a un nivel no adecuado para garantizar el correcto funcionamiento del sistema. Sin embargo, esta primera gráfica corresponde al peor caso de reconfiguración, configurando de manera global a todo el FPGA que, además, es muy grande para la arquitectura que se desarrolló (en la figura 12.4 se observa que únicamente se está utilizando el 3.67 % de la memoria de configuración del FPGA), por lo que se requiere de un tiempo muy elevado para reconfigurar todo el FPGA.

13.2 Variación de la intensidad de recuperación

Para observar el nivel de fiabilidad del sistema variando la intensidad de recuperación, se tomaron como referencia los tamaños del *bitstream* para cada uno de los FPGA de la serie 7 de *Xilinx*, de modo que disminuya el tiempo de reconfiguración para cada dispositivo, como se presenta en la tabla 13.1.

Tabla 13.1: Tamaño de los archivos *Bitstream* de los FPGA de la serie 7 de *Xilinx* [78]

Dispositivo	Tamaño del <i>Bitstream</i> (<i>b</i>)	Tiempo de configuración (<i>ms</i>)	Intensidad de recuperación ($\frac{\text{eventos}_{rec}}{ms}$)
7A12T	9 934 432	99.3443	0.01006600278
7A15T	17 536 096	175.3609	0.00570252547
7A25T	9 934 432	99.3443	0.01006600278
7A35T	17 536 096	175.3609	0.00570252547
7A50T	17 536 096	175.3609	0.00570252547
7A75T	30 606 304	306.0630	0.00326733
7A100T	30 606 304	306.0630	0.00326733
7A200T	77 845 216	778.4521	0.00128460055

Con base en lo anterior, se realizaron las gráficas de la fiabilidad en función del tiempo para los tres casos de λ y para los cuatro casos de μ .

13.2.1 Caso 1 de estudio: $LET_{th} = 3 MeV cm^2 g^{-1}$

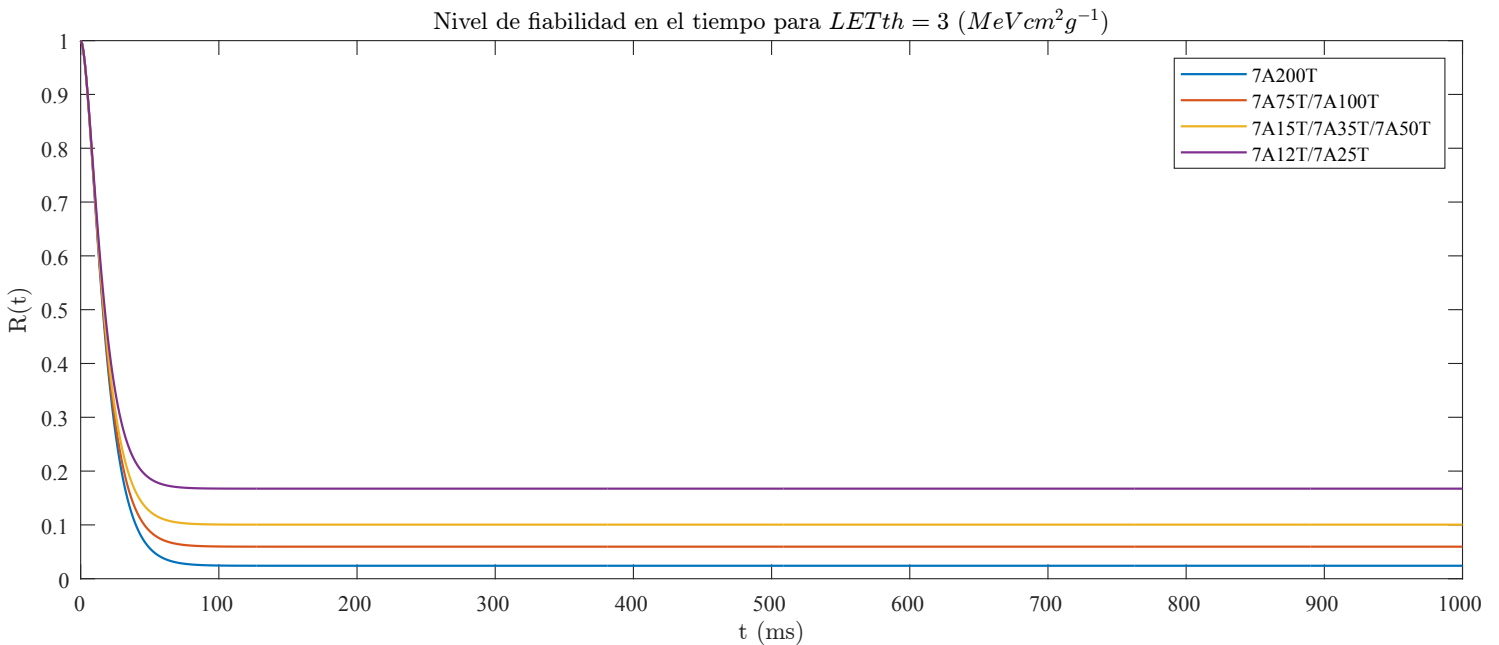


Figura 13.2: Resultados del análisis de fiabilidad para el caso 1 de estudio, variando la intensidad de recuperación

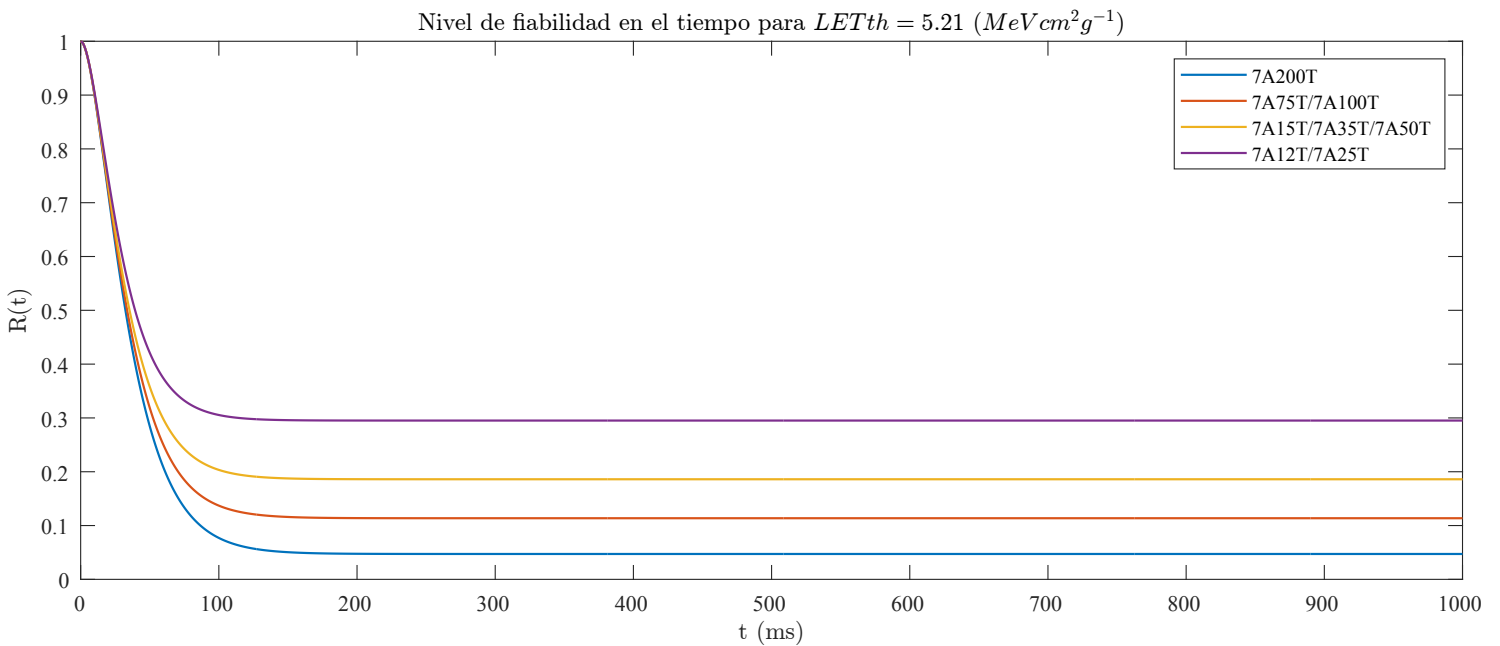
13.2.2 Caso 2 de estudio: $LETth = 5.21 MeV cm^2 g^{-1}$ 

Figura 13.3: Resultados del análisis de fiabilidad para el caso 2 de estudio, variando la intensidad de recuperación

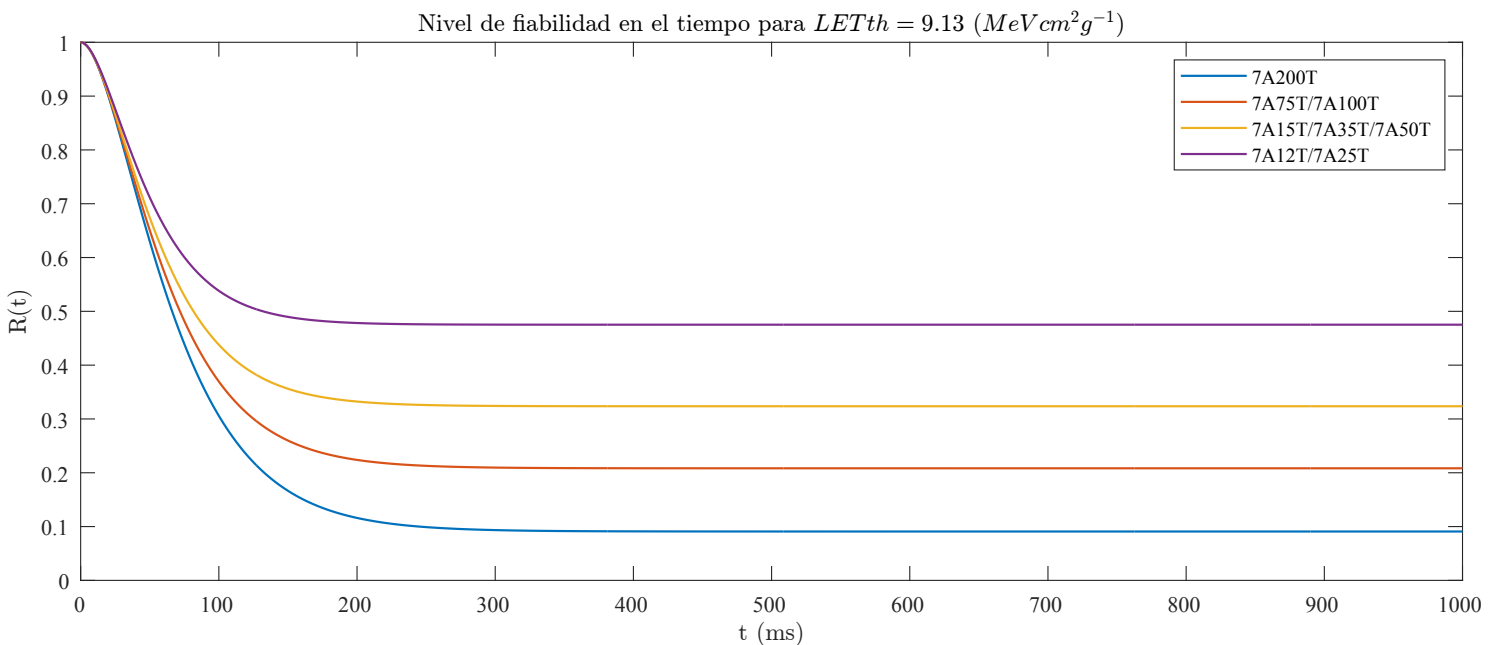
13.2.3 Caso 3 de estudio: $LETth = 9.13 MeV cm^2 g^{-1}$ 

Figura 13.4: Resultados del análisis de fiabilidad para el caso 3 de estudio, variando la intensidad de recuperación

Con los resultados es claro que el nivel de fiabilidad del SCMI cae rápidamente para mantenerse en un nivel muy bajo (alrededor de 0.5 en el mejor de los casos). Esto es debido al uso del esquema más lento de reconfiguración (configuración global) del FPGA, por lo que se realizará el análisis de fiabilidad al emplear reconfiguración parcial del FPGA.

13.3 Configuración a través de archivos *Bitstream* parciales

Para aplicar reconfiguración parcial del FPGA se deben definir archivos *bitstream* parciales, los cuales tienen la misma estructura de un *bitstream* completo pero están limitados por una región de memoria específica definida previamente a partir del diseño de la arquitectura. Para hacer el análisis de fiabilidad empleando archivos *bitstream* parciales se consideró que el tamaño de un *bitstream* parcial es directamente proporcional al tamaño de la región a configurar [79].

Por facilidad, se considerará este análisis para los dispositivos más pequeños de la serie 7 de *Xilinx* (7A12T y 7A25T), los cuales tienen un *bitstream* de 9 934 432 bits, de los cuales la CPU utiliza 2 241 227 bits (figura 12.4) que equivale a aproximadamente el 80% del dispositivo. Para observar el comportamiento de la fiabilidad se propusieron múltiples tamaños de archivos *bitstream* (tabla 13.2).

Tabla 13.2: Tamaño de los archivos *Bitstream* para reconfiguración parcial

Uso de recursos	Tamaño del <i>Bitstream</i> (<i>b</i>)	Tiempo de configuración (<i>ms</i>)	Intensidad de recuperación ($\frac{\text{eventos}_{rec}}{ms}$)
100 %	9 934 432	99.3443	0.01006600278
80 %	7 947 545	79.4754	0.01258250981
60 %	5 960 659	59.6065	0.01677669382
40 %	3 973 772	39.7377	0.02516501962
20 %	1 986 886	19.8688	0.05033016589
10 %	993 443	9.9344	0.1006603318
5 %	496 721	4.9672	0.2013202583

13.3.1 Caso 1 de estudio: $LET_{th} = 3MeVcm^2g^{-1}$

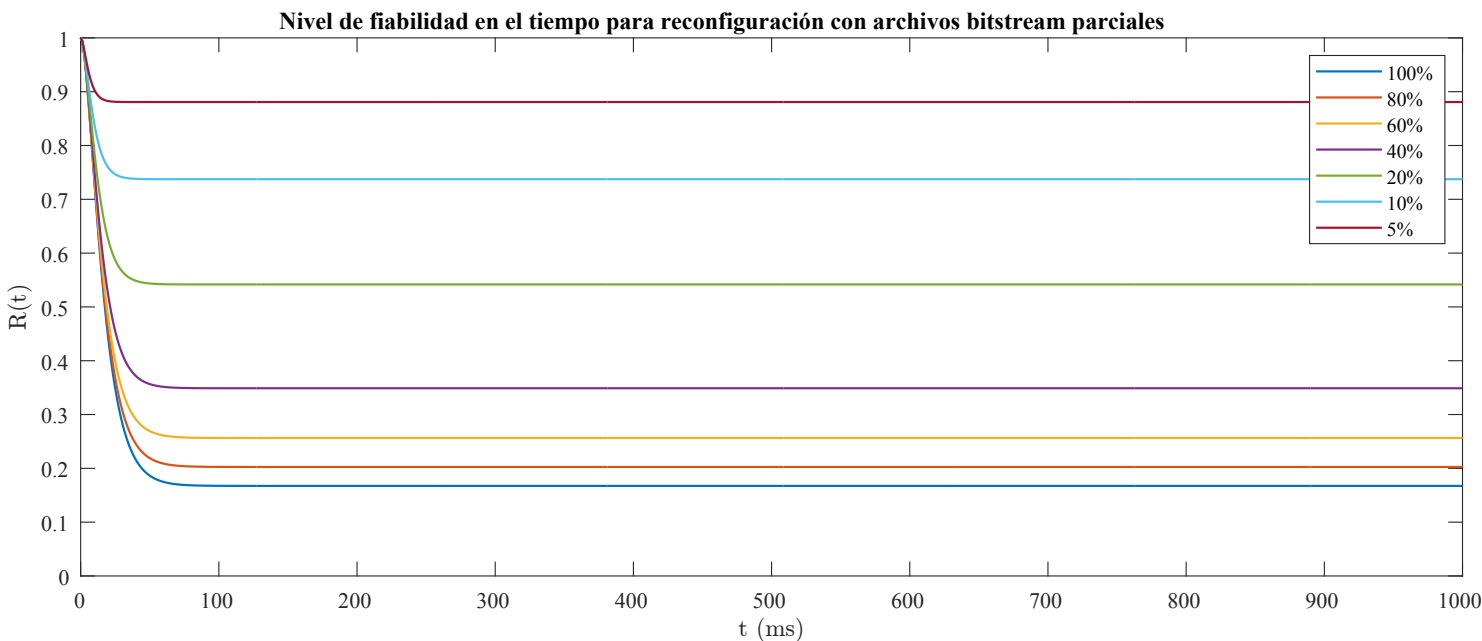


Figura 13.5: Resultados del análisis de fiabilidad para el caso 1 de estudio, empleando diferentes tamaños del *Bitstream*

13.3.2 Caso 2 de estudio: $LET_{th} = 5.21 MeV cm^2 g^{-1}$

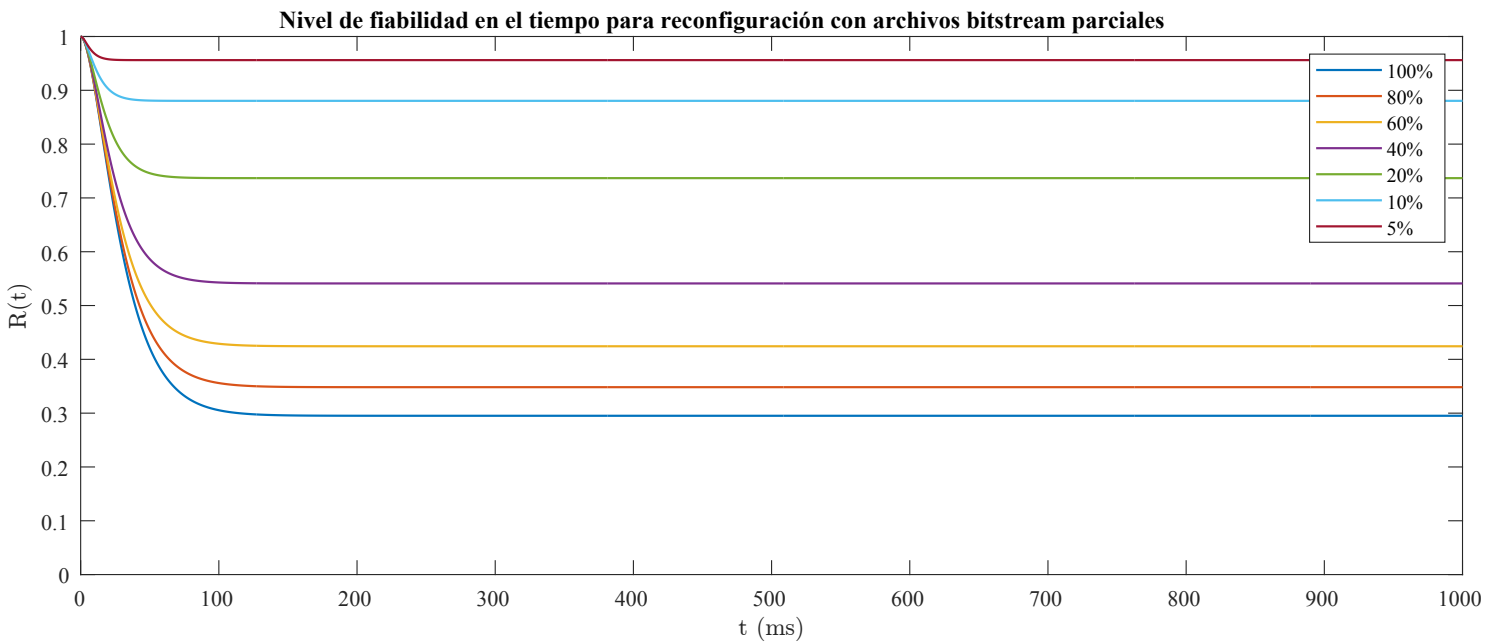


Figura 13.6: Resultados del análisis de fiabilidad para el caso 2 de estudio, empleando diferentes tamaños del *Bitstream*

13.3.3 Caso 3 de estudio: $LET_{th} = 9.13 MeV cm^2 g^{-1}$

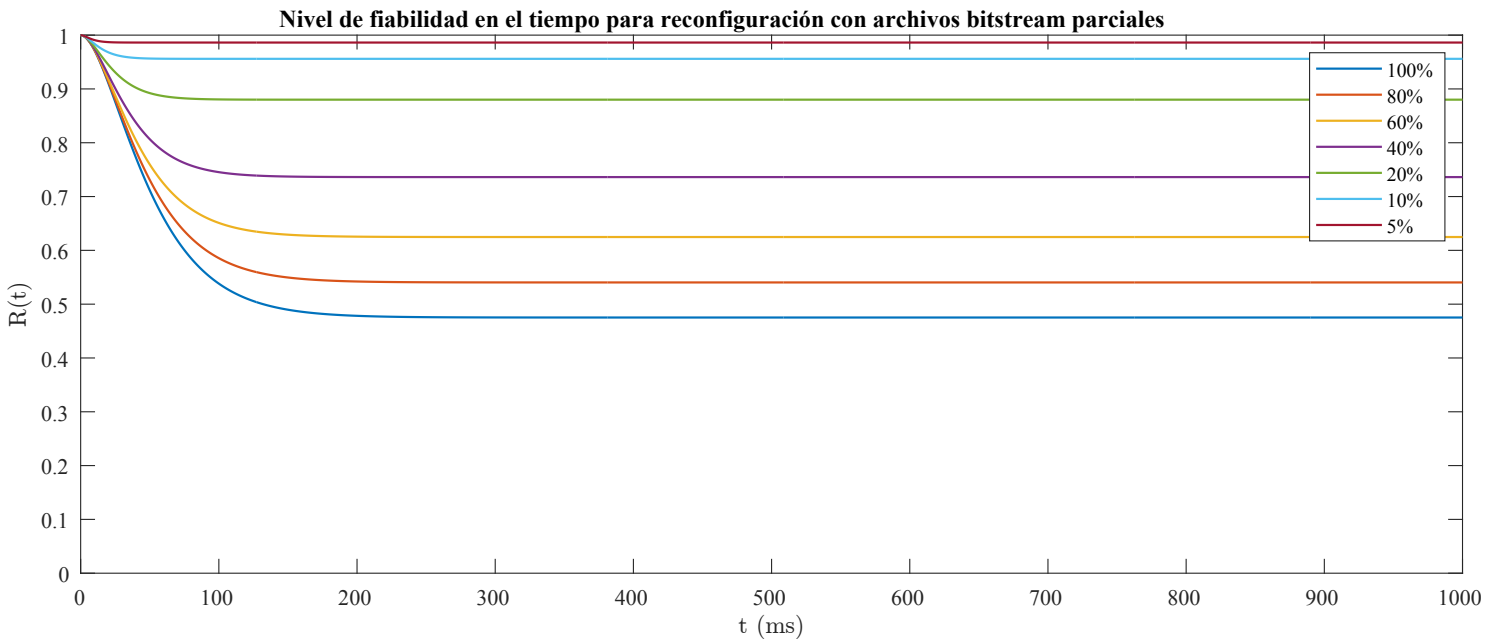


Figura 13.7: Resultados del análisis de fiabilidad para el caso 3 de estudio, empleando diferentes tamaños del *Bitstream*

Con los resultados de las gráficas 13.5, 13.6 y 13.7 se observa un aumento en el nivel de fiabilidad conforme se disminuye el tiempo necesario para reconfigurar el FPGA. De esta manera, la intensidad de recuperación incrementa, siendo mayor a la intensidad de fallas y acercando la fiabilidad del sistema al ideal.

14 CONCLUSIONES

En este trabajo de tesis se desarrolló un SCMI fundamentado en el estado del arte, empleando un FPGA COTS como unidad maestra del sistema, el cual estará expuesto a un ambiente de radiación con alto flujo de partículas cargadas (como se observó en el capítulo 7) que atravesarán el material para impactar el dispositivo y transferirá la energía necesaria para alterar su funcionamiento.

La selección de un FPGA COTS se fundamenta en su bajo costo y alto nivel de desempeño en comparación con componentes de grado espacial. Sin embargo, los componentes COTS presentan una sensibilidad alta a la radiación, degradando rápidamente el nivel de fiabilidad del componente al inducir fallas en su estructura que lo terminarán llevando a un estado de malfuncionamiento. Debido a esto, se observa la necesidad de implementar esquemas de tolerancia a fallas, los cuales son una opción viable para elevar el nivel de fiabilidad y aproximarlo a los niveles adecuados para desempeñar la misión determinada.

Para el diseño del SCMI se propuso una arquitectura con TMR a nivel de procesador, la cual será supervisada por un elemento externo (con un nivel de fiabilidad mucho mayor), con la capacidad de reconfigurar al FPGA para recuperar su estado funcional perfecto y sea capaz de cumplir con la misión durante el tiempo determinado. Como primer análisis del nivel de fiabilidad del sistema se observó la peor situación en donde el tiempo de recuperación es muy alto en comparación con el tiempo entre fallas (el cual es considerablemente bajo al ser un componente COTS), demostrando que el sistema pierde rápidamente su fiabilidad. Sin embargo, existen alternativas de reconfiguración parcial que permiten disminuir el tiempo de reconfiguración, incrementando su capacidad de recuperación y, con ello, mantener un mayor nivel de fiabilidad del sistema. Con estos resultados y la variación de la fiabilidad en función del tiempo de recuperación se demuestra la necesidad de buscar alternativas que permitan seleccionar los puntos comunes de fallas para aplicar técnicas de *scrubbing* prioritario y DPR, impactando en menor medida los tiempos de reconfiguración y, adicionalmente, evitar la interrupción de la operación del FPGA.

En este trabajo de tesis se desarrolló la primera etapa de diseño del *SCMI-FPGA*, proyecto que forma parte de las líneas de investigación del LIESE, FI UNAM, con el cual se dio inicio a una línea de investigación a profundidad de la estructura interna de los FPGA y los métodos para diseñar sistemas tolerantes a fallas, colaborando con múltiples equipos de estudiantes dentro del LIESE debido a la complejidad y densidad de trabajo necesario para implementar técnicas de tolerancia a fallas de grado fino en la arquitectura propuesta, con la finalidad de desarrollar un producto que cumpla con las características establecidas a partir de una investigación del estado del arte y, de este modo, aportar en el desarrollo tecnológico de sistemas espaciales diseñados totalmente en México por estudiantes de la UNAM.

15 TRABAJO A FUTURO

Como se comentó previamente, este trabajo de tesis corresponde a la primera etapa de diseño del proyecto *SCMI-FPGA*, que forma parte de las líneas de investigación del LIESE, FI UNAM. Con esto en mente y a partir de los resultados, se definen las siguientes tareas como parte del trabajo a futuro.

- Integración de los módulos IP tolerantes a fallas diseñados a bajo nivel en el LIESE.
- Análisis de puntos comunes de falla en el SCMI.
- Implementación de técnicas de *scrubbing* prioritario en la unidad maestra del FPGA para incrementar su nivel de fiabilidad.
- Implementación de técnicas de DPR en la unidad maestra del SCMI para incrementar su nivel de fiabilidad.
- Diseñar el esquema de diagnóstico y reconfiguración de la unidad de supervisión del SCMI.
- Diseño y aplicación de técnicas de inyección de fallas para validar los esquemas de tolerancia a fallas propuestos para el SCMI.
- Construcción e integración del SCMI.
- Aplicación de pruebas de entorno espacial para caracterizar el SCMI.
- Integración en una plataforma espacial para validar el funcionamiento del SCMI en órbita.

REFERENCIAS

- [1] C. A. Lara, «Diseño de Un Sistema de Comando y Manejo de Información,» Tesis de licenciatura, Facultad de Ingeniería, Universidad Nacional Autónoma de México (UNAM), Ciudad de México, México, abr. de 2021, 157 págs. dirección: https://tesiunam.dgb.unam.mx/F/2IE1MLHKCANNQV9422LTQ52RMVB88R6CKPRDU9Y1FLNCM3KT23-11690?func=full-set-set&set_number=272194&set_entry=000001.
- [2] C. A. Lara, M. Fragoso, L. M. Juárez et al., «Fault Tolerant Architecture Design of a CubeSat Command and Data Handling System,» en *2023 IEEE 24th Latin American Test Symposium (LATS)*, Veracruz, Mexico: IEEE, 21 de mar. de 2023, págs. 1-6, ISBN: 9798350325973. DOI: [10.1109/LATS58125.2023.10154496](https://doi.org/10.1109/LATS58125.2023.10154496). dirección: <https://ieeexplore.ieee.org/document/10154496/> (visitado 19-07-2023).
- [3] A. Johnstone, «CubeSat Design Specification (CDS) (1U - 12U),» Cal Poly, San Luis Obispo, CA, CP-CDS-R14, jul. de 2020. dirección: <https://www.cubesat.org/>.
- [4] J. R. Wertz, D. F. Everett y J. J. Puschell, *Space Mission Engineering: The New SMAD* (Space Technology Library 28). Torrance: Microcosm Press, ene. de 2011, 1033 págs., ISBN: 978-1-881883-15-9 978-1-881883-16-6.
- [5] K. T. Ulrich, S. D. Eppinger y M. C. Yang, *Product Design and Development*, Seventh edition. New York, NY: McGraw-Hill, 2020, 432 págs., ISBN: 978-1-260-04365-5 978-1-260-56643-7. dirección: <https://www.mheducation.com/highered/product/product-design-development-ulrich-eppinger/M9781260043655.html>.
- [6] NASA. «Image Galleries,» National Aeronautics and Space Administration (NASA). (), dirección: <https://www.nasa.gov/multimedia/imagegallery/index.html> (visitado 03-02-2022).
- [7] M. Garcia. «60 Years Ago, the Space Age Began — NASA.» (oct. de 2017), dirección: <https://www.nasa.gov/feature/60-years-ago-the-space-age-began> (visitado 26-07-2023).
- [8] A. Zak. «First Artificial Satellite Orbits the Earth.» (oct. de 2021), dirección: https://www.russianspaceweb.com/sputnik_mission.html (visitado 26-07-2023).
- [9] A. Zak. «Chronology of Space Exploration.» (jul. de 2023), dirección: <https://www.russianspaceweb.com/chronology.html> (visitado 26-07-2023).
- [10] UNOOSA. «Online Index of Objects Launched into Outer Space.» (2023), dirección: https://www.unoosa.org/oosa/osoindex/search-ng.jsp?lf_id= (visitado 26-07-2023).
- [11] UNOOSA. «United Nations Register of Objects Launched into Outer Space.» (2023), dirección: <https://www.unoosa.org/oosa/en/spaceobjectregister/index.html> (visitado 27-07-2023).
- [12] BryceTech, «Smallsats by the Numbers 2023,» mar. de 2023. dirección: <https://brycetech.com/reports>.
- [13] P. W. Fortescue, G. Swinerd y J. Stark, *Spacecraft Systems Engineering* (Aerospace Series), 4.^a ed. Chichester: Wiley, ago. de 2011, 691 págs., ISBN: 978-0-470-75012-4. dirección: <https://www.wiley.com/en-gb/Spacecraft+Systems+Engineering%2C+4th+Edition-p-9780470750124>.
- [14] Committee on Achieving Science Goals with CubeSats, Space Studies Board, Division on Engineering and Physical Sciences y National Academies of Sciences, Engineering, and Medicine, *Achieving Science with CubeSats: Thinking Inside the Box*. Washington, D.C.: National Academies Press, 6 de oct. de 2016, pág. 23 503, ISBN: 978-0-309-44263-3. DOI: [10.17226/23503](https://doi.org/10.17226/23503). dirección: <https://www.nap.edu/catalog/23503> (visitado 01-08-2023).
- [15] C. Cappelletti, S. Battistini y B. K. Malphrus, eds., *CubeSat Handbook: From Mission Design to Operations*. London, United Kingdom ; San Diego, CA, United States: Academic Press is an imprint of Elsevier, 2021, 469 págs., ISBN: 978-0-12-817884-3. dirección: <https://www.sciencedirect.com/book/9780128178843/cube-sat-handbook>.

- [16] NASA, «State-of-the-Art: Small Spacecraft Technology 2022,» Small Spacecraft Systems Virtual Institute, Ames Research Center, Moffett Field, California, NASA/TP-2022-0018058, ene. de 2023. dirección: <https://www.nasa.gov/smallsat-institute/sst-soa>.
- [17] S. Buchner, «Radiation Hardness Assurance (RHA) for Space Systems,» en *4th International School on the Effects of Radiation on Embedded Systems for Space Applications (SERESSA)*, West Palm Beach, FL, dic. de 2008. dirección: https://nepp.nasa.gov/files/25359/SERESSA08_Buchner.pdf (visitado 28-03-2022).
- [18] M. J. Sampson, «Electrical, Electronic and Electromechanical (EEE) Parts for Spaceflight Applications: A NASA Electronic Parts and Packaging Historical Perspective,» en *NEPP Electronics Technology Workshop*, Goddard Space Flight Center, jun. de 2019. dirección: <https://ntrs.nasa.gov/citations/20190026517> (visitado 28-03-2022).
- [19] NEPP, «Electrical, Electronic, and Electromechanical (EEE) Parts Management and Control Requirements for MSFC Space Flight Hardware,» NASA Electronic Parts and Packaging Program, George C. Marshall Space Flight Center, Alabama, MSFC Technical standard MSFC-STD-3012, feb. de 2012. dirección: <https://nepp.nasa.gov/pages/MSFC-STD-3012.cfm>.
- [20] NASA, «Electrical, Electronic, and Electromechanical (EEE) Parts Assurance Standard,» National Aeronautics and Space Administration (NASA), Washington, DC, Technical Standard NASA-STD-8739.10, 13 de jun. de 2017. dirección: <https://standards.nasa.gov/standard/NASA/NASA-STD-873910>.
- [21] D. C. Mayer y R. C. Lacoé, «Designing Integrated Circuits to Withstand Space Radiation,» *Crosslink: The Aerospace Corporation magazine of advances in aerospace technology*, vol. 4, n.º 2, págs. 30-35, 2003, ISSN: 1527-5264. dirección: <https://www.if.ufrj.br/~mms/lab4/CrosslinkV4N2.pdf>.
- [22] K. A. LaBel, «Challenges for Electronics in the Vision for Space Exploration,» en *Solar and Space Physics for the Vision for Space Exploration (SSPVSE)*, Charlottesville, Virginia, oct. de 2005. dirección: https://radhome.gsfc.nasa.gov/radhome/papers/SSPVSE05_LaBel.pdf.
- [23] T. Space. «Is Space Qualification Product or Process Based?» (2016), dirección: <https://thermal-space.com/space-qualification-product-process/> (visitado 10-08-2023).
- [24] I. Tzinis. «Technology Readiness Level — NASA.» (abr. de 2021), dirección: https://www.nasa.gov/directories/techreadiness/technology/technology_readiness_level (visitado 09-08-2023).
- [25] N. Battezzati, L. Sterpone y M. Violante, *Reconfigurable Field Programmable Gate Arrays for Mission-Critical Applications*. New York, NY: Springer New York, 2011, ISBN: 978-1-4419-7594-2 978-1-4419-7595-9. DOI: 10.1007/978-1-4419-7595-9. dirección: <https://link.springer.com/10.1007/978-1-4419-7595-9> (visitado 26-07-2023).
- [26] R. D. Schrimpf y D. M. Fleetwood, *Radiation Effects And Soft Errors In Integrated Circuits And Electronic Devices* (Selected Topics in Electronics and Systems). WORLD SCIENTIFIC, jul. de 2004, vol. 34, 288 págs., ISBN: 978-981-238-940-4 978-981-4482-15-8. DOI: 10.1142/5607. dirección: <http://www.worldscientific.com/worldscibooks/10.1142/5607> (visitado 09-12-2022).
- [27] J. E. Mazur, «An Overview of the Space Radiation Environment,» *Crosslink: The Aerospace Corporation magazine of advances in aerospace technology*, vol. 4, n.º 2, págs. 10-14, 2003, ISSN: 1527-5264. dirección: <https://www.if.ufrj.br/~mms/lab4/CrosslinkV4N2.pdf>.
- [28] Committee on Space Radiation Effects Testing Infrastructure for the U.S. Space Program, National Materials and Manufacturing Board, Division on Engineering and Physical Sciences y National Academies of Sciences, Engineering, and Medicine, *Testing at the Speed of Light: The State of U.S. Electronic Parts Space Radiation Testing Infrastructure*. Washington, D.C.: National Academies Press, 8 de jun. de 2018, pág. 24993, ISBN: 978-0-309-47079-7. DOI: 10.17226/24993. dirección: <https://www.nap.edu/catalog/24993> (visitado 24-07-2023).
- [29] ESA. «European Space Agency.» (), dirección: <https://www.esa.int/> (visitado 01-09-2023).
- [30] J. Scarpulla y A. Yarbrough, «What Could Go Wrong? The Effects of Ionizing Radiation on Space Electronics,» *Crosslink: The Aerospace Corporation magazine of advances in aerospace technology*, vol. 4, n.º 2, págs. 15-19, 2003, ISSN: 1527-5264. dirección: <https://www.if.ufrj.br/~mms/lab4/CrosslinkV4N2.pdf>.

- [31] ESA, *Space Product Assurance: Techniques for Radiation Effects Mitigation in ASICs and FPGAs Handbook* (European Cooperation for Space Standardization (ECSS)). ESA Requirements and Standards Division, sep. de 2016, 236 págs. dirección: <https://ecss.nl/hbstms/ecss-q-hb-60-02a-techniques-for-radiation-effects-mitigation-in-asics-and-fpgas-handbook-1-september-2016-published/>.
- [32] J. W. Howard Jr. y D. M. Hardage, «Spacecraft Environments Interactions: Space Radiation and Its Effects on Electronic Systems,» Marshall Space Flight Center, MSFC, Alabama, Technical Publication TP-1999-209373, jul. de 1999. dirección: <https://ntrs.nasa.gov/api/citations/19990116210/downloads/19990116210.pdf>.
- [33] K. Iniewski, *Radiation Effects in Semiconductors* (Devices, Circuits, and Systems), 1.^a ed. CRC Press, 2010, 405 págs., ISBN: 978-1-4398-2694-2. DOI: [10.1201/9781315217864](https://doi.org/10.1201/9781315217864). dirección: <https://www.taylorfrancis.com/books/9781439826959> (visitado 09-12-2022).
- [34] C. Maxfield, *The Design Warrior's Guide to FPGAs: Devices, Tools, and Flows*. Boston: Newnes/Elsevier, 2004, 542 págs., ISBN: 978-0-7506-7604-5.
- [35] G. E. N. E. R. A. Technologies. «Space radiation and FPGA for space applications - GENERA Technologies.» (), dirección: <https://www.generatecologias.es/en/space-fpga.html> (visitado 10-09-2023).
- [36] E. Dubrova, *Fault-Tolerant Design*. New York, NY: Springer New York, 2013, ISBN: 978-1-4614-2112-2 978-1-4614-2113-9. DOI: [10.1007/978-1-4614-2113-9](https://doi.org/10.1007/978-1-4614-2113-9). dirección: <https://link.springer.com/10.1007/978-1-4614-2113-9> (visitado 21-07-2023).
- [37] B. W. Johnson, *Design and Analysis of Fault-Tolerant Digital Systems* (Addison-Wesley Series in Electrical and Computer Engineering), Reprinted with corr., [5. Dr.] Reading, Mass. [u.a]: Addison-Wesley, 1995, 584 págs., ISBN: 978-0-201-07570-0.
- [38] M. Yang, G. Hua, Y. Feng y J. Gong, *Fault-Tolerance Techniques for Spacecraft Control Computers*. Singapore: John Wiley & Sons Singapore Pte. Ltd, ene. de 2017, 352 págs., ISBN: 978-1-119-10739-2 978-1-119-10727-9. DOI: [10.1002/9781119107392](https://doi.org/10.1002/9781119107392). dirección: <http://doi.wiley.com/10.1002/9781119107392>.
- [39] B. W. Johnson, *An Introduction to the Design and Analysis of Fault-Tolerant Systems*. Addison-Wesley, feb. de 1996, 108 págs. dirección: https://www.researchgate.net/publication/234812893_An_introduction_to_the_design_and_analysis_of_fault-tolerant_systems.
- [40] A. Avizienis, J.-C. Laprie, B. Randell y C. Landwehr, «Basic concepts and taxonomy of dependable and secure computing,» *IEEE Transactions on Dependable and Secure Computing*, vol. 1, n.º 1, págs. 11-33, ene. de 2004, ISSN: 1545-5971. DOI: [10.1109/TDSC.2004.2](https://doi.org/10.1109/TDSC.2004.2). dirección: <http://ieeexplore.ieee.org/document/1335465/> (visitado 22-09-2023).
- [41] V. Nelson, «Fault-Tolerant Computing: Fundamental Concepts,» *Computer*, vol. 23, n.º 7, págs. 19-25, jul. de 1990, ISSN: 0018-9162. DOI: [10.1109/2.56849](https://doi.org/10.1109/2.56849). dirección: <http://ieeexplore.ieee.org/document/56849/> (visitado 18-09-2023).
- [42] J. R. Norris, *Markov Chains*, 1.^a ed. Cambridge University Press, 28 de feb. de 1997, ISBN: 978-0-521-48181-6 978-0-521-63396-3 978-0-511-81063-3. DOI: [10.1017/CB09780511810633](https://doi.org/10.1017/CB09780511810633). dirección: <https://www.cambridge.org/core/product/identifiier/9780511810633/type/book> (visitado 09-11-2023).
- [43] «Analog — Embedded Processing — Semiconductor Company — TI.Com.» (), dirección: <https://www.ti.com/> (visitado 24-10-2023).
- [44] J. von Neumann, «Probabilistic Logic and the Synthesis of Reliable Organisms from Unreliable Components,» en *Automata Studies*, ép. AM 34, vol. 34, Princeton University Press, 1956. dirección: <https://www.degruyter.com/document/doi/10.1515/9781400882618-003/html>.
- [45] A. Saleh, J. Serrano y J. Patel, «Reliability of Scrubbing Recovery-Techniques for Memory Systems,» *IEEE Transactions on Reliability*, vol. 39, n.º 1, págs. 114-122, abr. de 1990, ISSN: 00189529. DOI: [10.1109/24.52622](https://doi.org/10.1109/24.52622). dirección: <http://ieeexplore.ieee.org/document/52622/> (visitado 11-11-2023).
- [46] E. Kulu. «Nanosats Database — Constellations, Companies, Technologies and More.» (2023), dirección: <https://www.nanosats.eu/> (visitado 04-08-2023).

- [47] M. Langer y J. Bouwmeester, «Reliability of CubeSats - Statistical Data, Developers' Beliefs and the Way Forward,» en *Proceedings of the 30th Annual AIAA/USU Conference on Small Satellites*, jul. de 2016. dirección: <https://digitalcommons.usu.edu/smallsat/2016/TS10AdvTech2/4/>.
- [48] NASA, «State-of-the-Art: Small Spacecraft Technology 2021,» Small Spacecraft Systems Virtual Institute, Ames Research Center, Moffett Field, California, NASA/TP-20210021263, oct. de 2021. dirección: <https://www.nasa.gov/smallsat-institute/sst-soa>.
- [49] C. M. Fuchs, N. M. Murillo, A. Plaat, E. Van Der Kouwe, D. Harsono y T. P. Stefanov, «Fault-Tolerant Nanosatellite Computing on a Budget,» en *2018 18th European Conference on Radiation and Its Effects on Components and Systems (RADECS)*, Goteborg, Sweden: IEEE, sep. de 2018, págs. 1-8, ISBN: 978-1-72810-216-0. DOI: [10.1109/RADECS45761.2018.9328685](https://doi.org/10.1109/RADECS45761.2018.9328685). dirección: <https://ieeexplore.ieee.org/document/9328685/> (visitado 12-12-2022).
- [50] X. Iturbe, D. Keymeulen, P. Yiu et al., «On the Use of System-on-Chip Technology in Next-Generation Instruments Avionics for Space Exploration,» en *VLSI-SoC: Design for Reliability, Security, and Low Power*, Y. Shin, C. Y. Tsui, J. J. Kim, K. Choi y R. Reis, eds., vol. 483, Cham: Springer International Publishing, sep. de 2016, págs. 1-22, ISBN: 978-3-319-46096-3 978-3-319-46097-0. DOI: [10.1007/978-3-319-46097-0_1](https://doi.org/10.1007/978-3-319-46097-0_1). dirección: http://link.springer.com/10.1007/978-3-319-46097-0_1 (visitado 12-12-2022).
- [51] A. Fedi, M. Ottavi, G. Furano et al., «High-Energy Neutrons Characterization of a Safety Critical Computing System,» en *2017 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT)*, Cambridge: IEEE, oct. de 2017, págs. 1-4, ISBN: 978-1-5386-0362-8. DOI: [10.1109/DFT.2017.8244456](https://doi.org/10.1109/DFT.2017.8244456). dirección: <http://ieeexplore.ieee.org/document/8244456/> (visitado 12-12-2022).
- [52] M. Ebrahimi, M. B. R. Parthasarathy, R. Seyyedi y M. B. Tahoori, «Low-Cost Multiple Bit Upset Correction in SRAM-Based FPGA Configuration Frames,» *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 24, n.º 3, págs. 932-943, mar. de 2016, ISSN: 1063-8210, 1557-9999. DOI: [10.1109/TVLSI.2015.2425653](https://doi.org/10.1109/TVLSI.2015.2425653). dirección: <http://ieeexplore.ieee.org/document/7104165/> (visitado 12-12-2022).
- [53] F. Rittner, M. Ristic, R. Glein y A. Heuberger, «Automated Test Procedure to Detect Permanent Faults inside SRAM-based FPGAs,» en *2017 NASA/ESA Conference on Adaptive Hardware and Systems (AHS)*, Pasadena, CA, USA: IEEE, jul. de 2017, págs. 16-23, ISBN: 978-1-5386-3439-4. DOI: [10.1109/AHS.2017.8046354](https://doi.org/10.1109/AHS.2017.8046354). dirección: <http://ieeexplore.ieee.org/document/8046354/> (visitado 12-12-2022).
- [54] D. L. Bekker, T. A. Werne, T. O. Wilson et al., «A CubeSat Design to Validate the Virtex-5 FPGA for Spaceborne Image Processing,» en *2010 IEEE Aerospace Conference*, Big Sky, MT, USA: IEEE, mar. de 2010, págs. 1-9, ISBN: 978-1-4244-3887-7. DOI: [10.1109/AERO.2010.5446700](https://doi.org/10.1109/AERO.2010.5446700). dirección: <http://ieeexplore.ieee.org/document/5446700/> (visitado 12-12-2022).
- [55] R. Merl y P. Graham, «A Low-Cost, Radiation-Hardened Single-Board Computer for Command and Data Handling,» en *2016 IEEE Aerospace Conference*, Big Sky, MT, USA: IEEE, mar. de 2016, págs. 1-8, ISBN: 978-1-4673-7676-1. DOI: [10.1109/AERO.2016.7500849](https://doi.org/10.1109/AERO.2016.7500849). dirección: <http://ieeexplore.ieee.org/document/7500849/> (visitado 12-12-2022).
- [56] A. Hanafi, M. Karim, I. Latachi, T. Rachidi, S. Dahbi y S. Zouggar, «FPGA-based Secondary on-Board Computer System for Low-Earth-Orbit Nano-Satellite,» en *2017 International Conference on Advanced Technologies for Signal and Image Processing (ATSIP)*, Fez, Morocco: IEEE, mayo de 2017, págs. 1-6, ISBN: 978-1-5386-0551-6. DOI: [10.1109/ATSIP.2017.8075514](https://doi.org/10.1109/ATSIP.2017.8075514). dirección: <http://ieeexplore.ieee.org/document/8075514/> (visitado 12-12-2022).
- [57] B. Shashidhara, S. Jadhav e Y. S. Kim, «Reconfigurable Fault Tolerant Processor on a SRAM Based FPGA,» en *2020 IEEE International Conference on Electro Information Technology (EIT)*, Chicago, IL, USA: IEEE, jul. de 2020, págs. 151-154, ISBN: 978-1-72815-317-9. DOI: [10.1109/EIT48999.2020.9208275](https://doi.org/10.1109/EIT48999.2020.9208275). dirección: <https://ieeexplore.ieee.org/document/9208275/> (visitado 12-12-2022).

- [58] A. Rodriguez, L. Santos, R. Sarmiento y E. De La Torre, «Scalable Hardware-Based On-Board Processing for Run-Time Adaptive Lossless Hyperspectral Compression,» *IEEE Access*, vol. 7, págs. 10 644-10 652, jul. de 2019, ISSN: 2169-3536. DOI: [10.1109/ACCESS.2019.2892308](https://doi.org/10.1109/ACCESS.2019.2892308). dirección: <https://ieeexplore.ieee.org/document/8610106/> (visitado 12-12-2022).
- [59] K. Ngo, T. Mohammadat y J. Oberg, «Towards a Single Event Upset Detector Based on COTS FPGA,» en *2017 IEEE Nordic Circuits and Systems Conference (NORCAS): NORCHIP and International Symposium of System-on-Chip (SoC)*, Linkoping: IEEE, oct. de 2017, págs. 1-6, ISBN: 978-1-5386-2844-7. DOI: [10.1109/NORCHIP.2017.8124960](https://doi.org/10.1109/NORCHIP.2017.8124960). dirección: <http://ieeexplore.ieee.org/document/8124960/> (visitado 12-12-2022).
- [60] C. A. Hulme, H. H. Loomis, A. A. Ross y R. Yuan, «Configurable Fault-Tolerant Processor (CFTP) for Spacecraft Onboard Processing,» en *2004 IEEE Aerospace Conference Proceedings (IEEE Cat. No.04TH8720)*, vol. 4, Big Sky, MT, USA: IEEE, mar. de 2004, págs. 2269-2276, ISBN: 978-0-7803-8155-1. DOI: [10.1109/AERO.2004.1368020](https://doi.org/10.1109/AERO.2004.1368020). dirección: <http://ieeexplore.ieee.org/document/1368020/> (visitado 12-12-2022).
- [61] M. Ebrahimi, A. Evans, M. B. Tahoori et al., «Comprehensive Analysis of Sequential and Combinational Soft Errors in an Embedded Processor,» *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 34, n.º 10, págs. 1586-1599, oct. de 2015, ISSN: 0278-0070, 1937-4151. DOI: [10.1109/TCAD.2015.2422845](https://doi.org/10.1109/TCAD.2015.2422845). dirección: <http://ieeexplore.ieee.org/document/7086043/> (visitado 12-12-2022).
- [62] F. L. Kastensmidt, L. Carro y R. Reis, *Fault-Tolerance Techniques for SRAM-based FPGAs* (Frontiers in Electronic Testing), 1.ª ed. Boston, MA: Springer US, 2006, 192 págs., ISBN: 978-0-387-31068-8. DOI: [10.1007/978-0-387-31069-5](https://doi.org/10.1007/978-0-387-31069-5). dirección: <http://link.springer.com/10.1007/978-0-387-31069-5> (visitado 09-12-2022).
- [63] T. Slivinski, C. Broglio, C. Wild et al., «Study of Fault-Tolerant Software Technology,» National Aeronautics and Space Administration (NASA), Hampton, Virginia, USA, Contractor Report NASA-CR-172385, sep. de 1984. dirección: <https://ntrs.nasa.gov/citations/19870002074>.
- [64] M. Liu y B. H. Meyer, «Bounding Error Detection Latency in Safety Critical Systems with Enhanced Execution Fingerprinting,» en *2016 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT)*, Storrs, CT, USA: IEEE, sep. de 2016, págs. 47-52, ISBN: 978-1-5090-3623-3. DOI: [10.1109/DFT.2016.7684068](https://doi.org/10.1109/DFT.2016.7684068). dirección: <http://ieeexplore.ieee.org/document/7684068/> (visitado 12-12-2022).
- [65] E. W. Wachter, V. Fochi, F. Barreto, A. M. Amory y F. G. Moraes, «A Hierarchical and Distributed Fault Tolerant Proposal for NoC-Based MPSoCs,» *IEEE Transactions on Emerging Topics in Computing*, vol. 6, n.º 4, págs. 524-537, oct. de 2018, ISSN: 2168-6750, 2376-4562. DOI: [10.1109/TETC.2016.2593640](https://doi.org/10.1109/TETC.2016.2593640). dirección: <https://ieeexplore.ieee.org/document/7517358/> (visitado 12-12-2022).
- [66] «Heavens-Above.» (), dirección: <https://www.heavens-above.com/> (visitado 19-11-2023).
- [67] «SPENVIS - Space Environment, Effects, and Education System.» (), dirección: <https://www.spennis.oma.be/> (visitado 19-11-2023).
- [68] «PC104 Consortium - PC/104 ConsortiumPC/104 Consortium.» (), dirección: <https://pc104.org/> (visitado 15-11-2023).
- [69] «AMD.» (), dirección: <https://www.amd.com/en.html> (visitado 21-11-2023).
- [70] P. S. Ostler, M. P. Caffrey, D. S. Gibelyou et al., «SRAM FPGA Reliability Analysis for Harsh Radiation Environments,» *IEEE Transactions on Nuclear Science*, vol. 56, n.º 6, págs. 3519-3526, dic. de 2009, ISSN: 0018-9499, 1558-1578. DOI: [10.1109/TNS.2009.2033381](https://doi.org/10.1109/TNS.2009.2033381). dirección: <http://ieeexplore.ieee.org/document/5341388/> (visitado 21-07-2023).
- [71] Xilinx, «MicroBlaze Triple Modular Redundancy (TMR) Subsystem v1.0,» Product Guide PG268, abr. de 2022. dirección: <https://docs.xilinx.com/r/en-US/pg268-tmr>.
- [72] G. Lum, *Electronics in Space Environment*.

- [73] R. Le, «Soft Error Mitigation Using Prioritized Essential Bits,» Xilinx, XAPP538, abr. de 2012. dirección: <https://www.eeweb.com/wp-content/uploads/articles-app-notes-files-soft-error-mitigation-using-prioritized-essential-bits-1339781673.pdf>.
- [74] D. White, «Considerations Surrounding Single Event Effects in FPGAs, ASICs, and Processors,» Xilinx, WP402, mar. de 2012. dirección: https://docs.xilinx.com/v/u/en-US/wp402_SEE_Considerations.
- [75] J. Tonfat, F. L. Kastensmidt, L. Artola et al., «Analyzing the Influence of the Angles of Incidence on SEU and MBU Events Induced by Low LET Heavy Ions in a 28-Nm SRAM-based FPGA,» en *2016 16th European Conference on Radiation and Its Effects on Components and Systems (RADECS)*, Bremen: IEEE, sep. de 2016, págs. 1-6, ISBN: 978-1-5090-4366-8. DOI: [10.1109/RADECS.2016.8093186](https://doi.org/10.1109/RADECS.2016.8093186). dirección: <http://ieeexplore.ieee.org/document/8093186/> (visitado 08-12-2023).
- [76] J. Tonfat, F. Lima Kastensmidt, L. Artola et al., «Analyzing the Influence of the Angles of Incidence and Rotation on MBU Events Induced by Low LET Heavy Ions in a 28-Nm SRAM-Based FPGA,» *IEEE Transactions on Nuclear Science*, vol. 64, n.º 8, págs. 2161-2168, ago. de 2017, ISSN: 0018-9499, 1558-1578. DOI: [10.1109/TNS.2017.2727479](https://doi.org/10.1109/TNS.2017.2727479). dirección: <http://ieeexplore.ieee.org/document/7982708/> (visitado 20-07-2023).
- [77] Xilinx, «Vivado Design Suite User Guide - Partial Reconfiguration,» Xilinx, UG909, dic. de 2018.
- [78] Xilinx, «7 Series FPGAs Configuration,» Xilinx, UG470, dic. de 2023. dirección: https://docs.xilinx.com/v/u/en-US/ug470_7Series_Config.
- [79] Xilinx, «Vivado Design Suite User Guide,» UG909, nov. de 2023. dirección: <https://docs.xilinx.com/viewer/book-attachment/sBScSL3FKcO3VaOVGNQb4w/6bGHmfY3ygosIBsuyij23Q>.

A SCMI COMERCIALES ANALIZADOS

Tabla A.1: Lista de SCMI comerciales













#	Imagen	Fabricante	SCMI	URL
1		AAC CLYDE SPACE	KRYTEN-M3 & KRYTEN-M3 PLUS	https://www.aac-clyde.space/what-we-do/space-products-components/command-data-handling/kryten-m3
2		AAC CLYDE SPACE Xiphos Technologies	Q7S	https://www.aac-clyde.space/what-we-do/space-products-components/command-data-handling/q7s
3		AAC CLYDE SPACE	SIRIUS OBC LEON3FT	https://www.aac-clyde.space/what-we-do/space-products-components/command-data-handling/smallsat-sirius-obc
4		AAC CLYDE SPACE	SIRIUS Quadcore LEON4FT	https://www.aac-clyde.space/what-we-do/space-products-components/command-data-handling/sirius-quadcore
5		AAC CLYDE SPACE	SIRIUS TCM LEON3FT	https://www.aac-clyde.space/what-we-do/space-products-components/command-data-handling/sirius-tcm-leon3ft
6		Alén Space	TRISKEL	https://products.alen.space/es/products/triskel-obc-ttc/
7		Argotec	Fermi	https://www.argotecgroup.com/wp-content/uploads/2022/03/Argotec_FERMI_scheda_prodotto.pdf
8		Argotec	Hack	https://www.argotecgroup.com/wp-content/uploads/2022/03/Argotec_HACK_scheda_prodotto.pdf
9		BAE Systems	RAD510	https://www.baesystems.com/en-media/uploadFile/20210909155547/1573668873270.pdf
10		BAE Systems	RAD5545	https://www.baesystems.com/en-media/uploadFile/20210404061759/1434594567983.pdf
11		BAE Systems	SpaceVPX	https://www.baesystems.com/en-media/uploadFile/20210404055515/1434615908806.pdf
12		Beyond Gravity	Constellation Single Board Computer	https://products.beyondgravity.com/d/SVq1aavsDmtE/library/show/eyJpZCI6MTk0NywidGltZlXNOYw1wIjoiMTY5NjQ2OTg0NiJ9:beyond-gravity:JbyTsARBJR4rQyHZyw-EjL99qKfyXz7YJx2ao2lsKLo

Tabla A.1: Lista de SCMI comerciales (continuación)

#	Imagen	Fabricante	SCMI	URL
13		Beyond Gravity	Lynx	https://products.beyondgravity.com/d/SVq1aavsDmtE/library/show/eyJpZCI6MTQ1MSwidGltZXR0YWIwIjoiMTY5NjQ2OTQ4NSJ9:beyond-gravity:iEx4dMxCCLctJj2EWNnrWnL6oTMZMJ0aHaD9sSUoMA8
14		C3S	CHP-OBC	https://c3s.hu/wp-content/uploads/2022/08/IPC-datasheet-4.0.pdf
15		C3S	On-Board Computer (OBC)	https://c3s.hu/wp-content/uploads/2022/01/C3S_OBC_datasheet.pdf
16		Cubesat Market	ICEPS	https://www.cubesat.market/all-in-one
17		D-Orbit	OBC Core	https://www.dorbit.space/media/0/30.pdf
18		D-Orbit	Simba	https://www.dorbit.space/media/0/32.pdf
19		Emxys	ODALISS DIPP	http://www.emxys.com/dipp
20		Emxys	ODALISS OBC	http://www.emxys.com/obc
21		EnduroSat	ONBOARD COMPUTER (OBC)	https://www.endurosat.com/cubesat-store/cubesat-obc/onboard-computer-obc/
22		EREMS	CPU BOARD	https://www.erems.fr/en/produit/cpu-board-control-and-processing-unit.php
23		EREMS	CPUGEN	https://www.erems.fr/en/produit/cpugen-on-board-computing-module.php
24		GALAXIA Mission Systems	ionVortex Firefly OBC	https://www.galaxiams.com/ionvortex-firefly-obc
25		GALAXIA Mission Systems	ionVortex Raven OBC	https://www.galaxiams.com/ionvortex-raven-obc
26		GAUSS Srl	ABACUS	https://www.gaussteam.com/products/onboard-computer/abacus-2/
27		GAUSS Srl	Hercules	https://www.gaussteam.com/products/onboard-computer/hercules/
28		GOMSpace	NanoMind A3200	https://gomspace.com/shop/subsystems/command-and-data-handling/nanomind-a3200.aspx

Tabla A.1: Lista de SCMI comerciales (continuación)

















#	Imagen	Fabricante	SCMI	URL
29		GOMSpace	NanoMind HP MK3	https://gomspace.com/shop/subsystems/command-and-data-handling/nanomind-hp-mk3-(1).aspx
30		GOMSpace	NanoMind Z7000 (MK2)	https://gomspace.com/shop/subsystems/command-and-data-handling/nanomind-z7000.aspx
31		GUMUSH	N-ART OBCOMMS	https://gumush.com.tr/product/n-art-obcomms/
32		H4R Human for research	OBC	https://www.h4-research.com/h4r_products/obc/
33		Ibeos	Edge-300	https://www.ibeos.com/standard-products
34		Infinity Avionics	PERUN	https://infinityavionics.com/products/perun/
35		Infinity Avionics	VOLKH	https://infinityavionics.com/products/volkh/
36		Innoflight	CFC-300	https://satcatalog.s3.amazonaws.com/components/433/SatCatalog_-_Innoflight_-_CFC-300_-_Datasheet.pdf?lastmod=20210708055823
37		Innoflight	CFC-400XP	https://www.innoflight.com/product/
38		Innoflight	CFC-400XS	https://www.innoflight.com/product/
39		Innoflight	CFC-500P	https://www.innoflight.com/product/
40		Innoflight	CFC-510P	https://www.innoflight.com/product/
41		ISISpace	ISIS On Board Computer	https://www.isispace.nl/product/on-board-computer/
42		KP Labs	Antelope	https://kplabs.space/antelope/
43		KP Labs	Leopard	https://kplabs.space/leopard/
44		KP Labs	Lion	https://kplabs.space/lion/

Tabla A.1: Lista de SCMI comerciales (continuación)















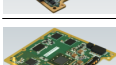


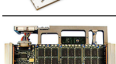
#	Imagen	Fabricante	SCMI	URL
45		NanoAvionics	SatBus 3C2	https://nanoavionics.com/cubesat-components/cubesat-on-board-computer-main-bus-unit-sat-bus-3c2/
46		Nara Space	NST OBC	https://satcatalog.s3.amazonaws.com/components/1340/SatCatalog_-_NARA_SPACE_-_NST_OBC_-_Datasheet.pdf?lastmod=20230103114952
47		Novo Space	SBC002AV	https://www.novo.space/products/sbc002av
48		Novo Space	SBC003AV	https://www.novo.space/products/sbc003av
49		Novo Space	SBC004AV	https://www.novo.space/products/sbc004av
50		OrbAstro	TELOS OBC	https://orbastro.com/subsystems/telos-obc/
51		Pumpkin	Pluggable Processor Module A1	https://www.pumpkinspace.com/store/p144/Pluggable_Processor_Module_A1_%28PPM_A1%29%2C_with_TI%E2%80%99s_MSP430F1612.html
52		Pumpkin	Pluggable Processor Module A2	https://www.pumpkinspace.com/store/p145/Pluggable_Processor_Module_A2_%28PPM_A2%29%2C_with_TI%E2%80%99s_MSP430F1611.html
53		Pumpkin	Pluggable Processor Module A3	https://www.pumpkinspace.com/store/p146/Pluggable_Processor_Module_A3_%28PPM_A3%29%2C_with_TI%E2%80%99s_MSP430F2618.html
54		Pumpkin	Pluggable Processor Module B1	https://www.pumpkinspace.com/store/p137/Pluggable_Processor_Module_B1_%28PPM_B1%29.html
55		Pumpkin	Pluggable Processor Module D1	https://www.pumpkinspace.com/store/p128/Pluggable_Processor_Module_D1_%28PPM_D1%29.html
56		Pumpkin	Pluggable Processor Module D2	https://www.pumpkinspace.com/store/p120/Pluggable_Processor_Module_D2_%28PPM_D2%29.html
57		Pumpkin	Pluggable Processor Module E1	https://www.pumpkinspace.com/store/p129/Pluggable_Processor_Module_E1_%28PPM_E1%29.html
58		SkyLabs	NANOhpc-obc	https://www.skylabs.si/products/nanohpc-obc/
59		SkyLabs	NANOhpm-obc	https://www.skylabs.si/products/nanohpm-obc/
60		SkyLabs	NANOobc-2	https://www.skylabs.si/products/nanoobc-2/
61		S4 Space	Robust On-Board Data Handling Unit	https://www.s4-space.com/files/S4_OBC.pdf
62		SEAKR Engineering	RCC 5	https://www.seakr.com/our-technology/#products

Tabla A.1: Lista de SCMI comerciales (continuación)



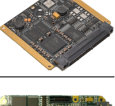
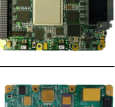
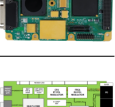
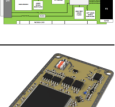
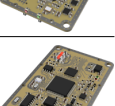






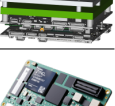
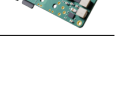





#	Imagen	Fabricante	SCMI	URL
63		SEAKR Engineering	RCC 5 – CPB	https://www.seakr.com/our-technology/#products
64		Space AI	NIO	https://spaceai.com/nio
65		SPACE MICRO	CSP: CubeSat Space Processor	https://www.spacemicro.com/products/digital-systems/CSP%20CUBESAT%20SPACE%20PROCESSOR.pdf
66		SPACE MICRO	PROTON300K	https://satcatalog.s3.amazonaws.com/components/442/SatCatalog_-_Space_Micro_-_Proton300k_-_Datasheet.pdf?lastmod=20210708060236
67		SPACE MICRO	PROTON400K	https://www.spacemicro.com/products/digital-systems/PROTON400K%E2%84%A2%20SINGLE%20BOARD%20COMPUTER%20(SBC).pdf
68		SPACE MICRO	PROTON600K	https://www.spacemicro.com/products/digital-systems/PROTON600K%E2%84%A2%20MULTI-CORE%20COMPUTER.pdf
69		Spacemanic	Deep Thought	https://www.spacemanic.com/deep-thought-onboard-computer/
70		Spacemanic	Eddie	https://www.spacemanic.com/eddie-onboard-computer/
71		SPUTNIX	On-board computer	https://sputnix.ru/en/equipment/cubesat-devices/on-board-computer
72		STM	MicrosatPro	https://www.stm.com.tr/en/our-solutions/command-and-control/microsatpro-2713
73		STM	NanosatPro	https://www.stm.com.tr/en/our-solutions/command-and-control/nanosatpro-2718
74		Unibap	e2155	https://unibap.com/wp-content/uploads/2021/06/1004001-unibap-information-sheet-on-unibap-e2000_e2100-modules.pdf
75		Unibap	e2160	https://unibap.com/wp-content/uploads/2021/06/1004001-unibap-information-sheet-on-unibap-e2000_e2100-modules.pdf
76		Unibap	iX5-100	https://unibap.com/space/spacecloud-products/ix5100/
77		Unibap	iX10-100	https://unibap.com/space/spacecloud-products/ix10100/
78		Xiphos Technologies	Q7	https://xiphos.com/product-details/q7

Tabla A.1: Lista de SCMI comerciales (continuación)

#	Imagen	Fabricante	SCMI	URL
79		Xiphos Technologies	Q8	https://xiphos.com/product-details/q8
80		Xiphos Technologies	Q8J	https://xiphos.com/product-details/q8j
81		Xiphos Technologies	Q8RF	https://xiphos.com/product-details/q8rf
82		Zephyr Computing Systems	Kestrel	https://www.zephyrcomputing.space/data-processing