

Índice

Índice	i
Resumen	1
Abstract.....	2
Resum	3
1 Presentación	5
1.1 Fundamentos y motivación	5
1.2 Objetivos.....	6
1.3 Desarrollo	8
2 Generalidades de los Sistemas Tolerantes a Fallos.....	11
2.1 Generalidades de los Sistemas Tolerantes a Fallos	11
2.1.1 Introducción	11
2.1.2 Atributos de la Confiabilidad	13
2.1.3 Impedimentos de la Confiabilidad	13
2.1.3.1 Averías.....	13
2.1.3.2 Errores.....	14
2.1.3.3 Fallos.....	15
2.1.3.4 Patología de los fallos	16
2.1.4 Medios para alcanzar la Confiabilidad	17
2.1.4.1 Tolerancia a fallos.....	17
2.1.4.2 Eliminación de fallos	17
2.1.4.3 Predicción de fallos.....	18
2.1.4.4 Dependencias entre los medios para alcanzar la Confiabilidad	20
2.1.5 Confiabilidad y Tolerancia a fallos	20
2.1.6 Confiabilidad y Validación	21
2.1.7 Tolerancia a fallos y Validación experimental	21
2.1.8 Validación experimental e Inyección de fallos.....	22
2.2 Técnicas de inyección de fallos.....	25
2.2.1 Introducción	25
2.2.2 Inyección de fallos implementada mediante hardware.....	27
2.2.2.1 Inyección de fallos externa	27
2.2.2.1.1 Inyección de fallos a nivel de pin.....	28
2.2.2.1.2 Inyección de fallos mediante interferencias electromagnéticas	28
2.2.2.2 Inyección de fallos interna	29
2.2.2.2.1 Inyección de iones pesados	29
2.2.2.2.2 Inyección mediante láser.....	29
2.2.2.2.3 Inyección de fallos mediante Scan-Chain	30
2.2.3 Inyección de fallos implementada mediante software.....	30
2.2.4 Inyección de fallos mediante simulación	36
2.2.4.1 Inyección de fallos basada en simulación software	37
2.2.4.1.1 Nivel tecnológico	38

2.2.4.1.2	Nivel de transistor	38
2.2.4.1.3	Nivel lógico.....	39
2.2.4.1.4	Nivel de transferencia entre registros (nivel RT)	39
2.2.4.1.5	Nivel de sistema	40
2.2.4.2	Inyección de fallos basada en emulación	42
2.3	Inyección de fallos híbrida	44
2.4	Resumen y conclusiones.....	47
3	Técnicas de inyección de fallos basadas en VHDL	49
3.1	Introducción	49
3.1.1	Antecedentes previos.....	50
3.2	Técnicas de inyección de fallos basadas en VHDL.....	56
3.2.1	Inyección de fallos mediante órdenes del simulador.....	56
3.2.1.1	Manipulación de señales	56
3.2.1.2	Manipulación de variables	56
3.2.1.3	Modelos de fallos.....	57
3.2.2	Inyección de fallos mediante la modificación del modelo VHDL	57
3.2.2.1	Perturbadores	57
3.2.2.1.1	Perturbador serie simple.....	62
3.2.2.1.2	Perturbador serie complejo	64
3.2.2.1.3	Perturbador bidireccional serie simple.....	65
3.2.2.1.4	Perturbador bidireccional serie complejo.....	67
3.2.2.1.5	Perturbador unidireccional simple de n bits.....	68
3.2.2.1.6	Perturbador unidireccional complejo de n bits.....	70
3.2.2.1.7	Perturbador bidireccional simple de n bits.....	71
3.2.2.1.8	Perturbador bidireccional complejo de n bits.....	73
3.2.2.1.9	Modelos de fallos en perturbadores	74
3.2.2.2	Mutantes	74
3.3	Automatización de las técnicas de inyección en modelos en VHDL	79
3.3.1	Automatización de las órdenes del simulador	79
3.3.2	Automatización de los perturbadores	79
3.3.3	Automatización de los mutantes.....	80
3.3.4	Resumen de los modelos de fallos	81
3.4	Comparación de las técnicas de inyección	82
3.5	Resumen. Conclusiones y líneas abiertas de investigación	84
4	VFIT: La herramienta de inyección de fallos. Modelos de fallos	85
4.1	Introducción	85
4.2	VFIT: VHDL-Based Fault Injection Tool.....	85
4.2.1	Características generales de VFIT.....	85
4.2.2	Fases de un experimento de inyección	86
4.2.3	Diagrama de bloques de VFIT	87
4.3	Modelos de fallos	90
4.3.1	Introducción	90
4.3.2	Mecanismos de fallos	91
4.3.2.1	Fallos permanentes	91
4.3.2.2	Fallos intermitentes.....	92
4.3.2.3	Fallos transitorios.....	92

4.3.3	Influencia de las nuevas tecnologías submicrónicas	93
4.3.3.1	Fallos permanentes	93
4.3.3.2	Fallos intermitentes.....	95
4.3.3.3	Fallos transitorios.....	95
4.3.3.3.1	Radiación de partículas α	96
4.3.3.3.2	Radiación de rayos cósmicos	96
4.3.3.3.3	Otros mecanismos	96
4.3.4	Resumen y conclusiones de los modelos de fallos	97
4.4	Resumen. Conclusiones y líneas abiertas de investigación	98
5	Aplicación de nuevas técnicas de inyección de fallos en modelos VHDL	99
5.1	Introducción	99
5.2	Experimentos de inyección de fallos.....	99
5.3	Resumen. Conclusiones y líneas abiertas de investigación	119
6	Arquitecturas de bus para sistemas distribuidos de tiempo real tolerantes a fallos. Introducción a la arquitectura Time-Triggered	121
6.1	Introducción	121
6.2	Características generales de los sistemas basados en buses.....	121
6.2.1	Buses basados en el tiempo (Time-Triggered Buses)	121
6.2.2	Arquitecturas de buses basados en eventos: CAN y ByteFlight.....	123
6.2.3	Arquitecturas de buses basados en el tiempo	124
6.2.3.1	La arquitectura Time-Triggered (TTA)	124
6.2.3.2	La arquitectura SAFEbus.....	124
6.2.3.3	La arquitectura FlexRay.....	124
6.2.3.4	La arquitectura Time-Triggered CAN (TTCAN).....	125
6.3	La arquitectura Time-Triggered	125
6.3.1	Introducción	125
6.3.2	El protocolo Time-Triggered.....	126
6.3.3	El modelo VHDL del controlador TTP/C: TTP/C-C1 y TTP/C-C2.....	132
6.3.3.1	El modelo VHDL del controlador TTP/C-C1	132
6.3.3.1.1	Unidad de Control del Protocolo (Protocol Control Unit)	133
6.3.3.1.2	Banco de Registros (Register File)	136
6.3.3.1.3	Unidad de Interfaz con el Host (Host Interface Unit)	136
6.3.3.1.4	Interfaz con la ROM (ROM Interface).....	137
6.3.3.1.5	Unidad de Control Temporal (Time Control Unit)	138
6.3.3.1.6	Unidad de CRC	138
6.3.3.1.7	Receptor (Receiver)	138
6.3.3.1.8	Transmisor (Transmitter)	139
6.3.3.1.9	Guardián del Bus (Bus Guardian).....	139
6.3.3.1.10	Bus de registros interno.....	140
6.3.3.2	El modelo VHDL del controlador TTP/C-C2.....	140
6.4	Resumen.....	144
7	Validación de sistemas distribuidos de tiempo real tolerantes a fallos para aplicaciones críticas	145

7.1	Introducción	145
7.2	Validación de sistemas distribuidos de tiempo real tolerantes a fallos.....	145
7.3	El proyecto FIT: Fault Injection in the Time-Triggered Architecture	147
7.3.1	Objetivos del proyecto FIT	147
7.3.2	Técnicas de inyección de fallos del proyecto FIT	148
7.3.2.1	Inyección de fallos física a nivel de pin	148
7.3.2.2	Inyección de fallos mediante iones pesados.....	148
7.3.2.3	Inyección de fallos implementada por software.....	149
7.3.2.4	Inyección de fallos basada en el microcódigo del protocolo.....	149
7.3.2.5	Inyección de fallos basada en VHDL	149
7.3.2.6	Inyección de fallos basada en C-SIM	149
7.3.2.7	Comparación de las técnicas inyección de fallos.....	150
7.4	Validación de la arquitectura Time-Triggered mediante inyección de fallos en VHDL.152	
7.4.1	Inyección de fallos en los modelos VHDL de los controladores TTP/C-C1 y TTP/C-C2 152	
7.4.1.1	Experimentos de ajuste en el TTP/C-C1.....	152
7.4.1.2	Validación del controlador de comunicaciones	156
7.4.1.2.1	Error de diseño en el TTP/C-C1: algoritmo del clique avoidance	156
7.4.1.2.2	Validación del controlador TTP/C-C2	165
7.4.1.3	Comparación y/o combinación de técnicas de inyección.....	167
7.4.1.3.1	Error de diseño: algoritmo de actualización de la señal de vida	167
7.4.1.3.2	Error arbitrario: error ligeramente fuera de las especificaciones	169
7.4.1.3.2.1	Avería de enlace saliente (del inglés, Outgoing link failure).....	171
7.4.1.3.2.2	Avería del babbling idiot	172
7.4.1.3.2.3	Avería de enmascaramiento (del inglés, masquerading failure)	172
7.4.1.3.2.4	Avería ligeramente fuera de las especificaciones (del inglés, Slightly-off- specification failure)	172
7.5	Resumen. Conclusiones y líneas abiertas de investigación	174
8	Conclusiones. Trabajo futuro	177
8.1	Conclusiones	177
8.1.1	Inyección de fallos sobre modelos en VHDL.....	177
8.1.2	Aplicación de la inyección de fallos basada en VHDL	179
8.1.3	Validación de la arquitectura Time-Triggered	180
8.2	Publicaciones	182
8.2.1	Capítulo Libro	182
8.2.2	Revistas	182
8.2.3	Congresos.....	182
8.2.4	Publicaciones con referencia a nuestros trabajos	183
8.3	Trabajo futuro.....	185
	Palabras Clave	187
	Bibliografía.....	191