# Tolerancia a Fallos (Software)

Un método de tolerancia que se puede utilizar, junto a o en vez de, el hardware para la detección y corrección de fallos, es la tolerancia vía software.

Esta tolerancia no modifica el diseño hardware del sistema, así se ahorra área a la hora de imprimir el mismo diseño en silicio o en los recursos al volcarlo en la FPGA. Área que se puede utilizar para ampliar el sistema, o simplemente ahorrar en temas económicos. Además el sistema será más sencillo, no necesitando el hardware añadido para la detección y corrección de los fallos que puedan ocurrir.

Existen diferentes métodos para conseguir esta tolerancia de los que introduciré unos pocos y nombraré otros. [Artech House - Software Fault Tolerance Techniques And Implementation]

Las técnicas para lograr que un sistema sea robusto se categorizan en dos grupos.

1. Construcción: Aquellas que se aplican a la hora de desarrollar e implementar del sistema software.
2. Validación: Aquellas que permiten validar el correcto funcionamiento del sistema una vez construido.

Los diferentes tipos de técnicas son:

1. **Fault Avoidance or Prevention /Evitar o Prevenir Fallos**

Las técnicas de "evitación o prevención de fallos" forman parte de la categoría "Construcción". Son aplicadas en la fase de desarrollo del software para minimizar el número de fallos introducidos o provocados a la hora de implementar el código del sistema software.

Algunos de estos métodos son ampliamente conocidos y utilizados en todo tipo de proyectos. Sirven para desarrollar de forma robusta, ayudando a detectar y arreglar muchos fallos antes de entregar el producto al cliente. Gracias a estas técnicas se preven fallos que podrían producir errores y se toman las medidas necesarias, evitando sorpresas a la hora de instalar el software, y que éste funcione correctamente.

Algunos ejemplos de este tipo de técnicas son:

1. Especificación de requisitos
2. Modelos de programación y diseño estructurados
3. Métodos formales
4. Reutilización de código
5. **Fault Removal / Eliminar Fallos**

Las técnicas de "Eliminación de fallos" están comprendidas entre las técnicas de "Validación". Se somete el sistema software a métodos de verificación y validación, para detectar los errores que pueda tener y poder arreglarlos en la siguiente fase.

Algunas de estas técnicas son:

1. Software de pruebas
2. (Formal Inspection)
3. (Formal Design Proofs)
4. (Fault Removal Summary)
5. **Fault/Failure Forecasting / Pronóstico de Fallos**

Estas técnicas proporcionan un modo de estimar la presencia de fallos(faults), y la aparición y consecuencia de los mismos (failures) y se usan en la etapa de "Validación". Estos medios pueden ayudar a predecir la eficiencia de realizar más pruebas al sistema.

Algunos ejemplos de estas técnicas son:

1. (Reliability Estimation)
2. (Reliability Prediction)
3. (Fault/Failure Forecasting Summary)
4. **Fault Tolerance / Tolerancia a Fallos**

Para reducir los efectos que provoquen los fallos se pueden aplicar técnicas de tolerancia a fallos, estas técnicas se aplican durante el desarrollo en la fase de "Construcción" del software. La tolerancia a fallos proporciona medios para poder actuar frente a la aparición de un fallo, disminuyendo sus efectos.

La tolerancia a fallos no es algo nuevo, ya que existen muchas técnicas y algunas desde hace más de 20 años. Algunas de estas técnicas son:

1. (Single Version Software Environment)
2. (Multiple Version Software Environment)
3. (Multiple Data Representation Environment)
4. (Software Fault Tolerance Summary)

[Ventajas]

[Desventajas]

[Beneficios frente a tolerancia HW]

[Desventajas frente a tolerancia HW]

[Como se puede utilizar esta tolerancia junto con la tolerancia HW]

[]