# Introducción [~5 páginas]

[Descripciones generales]

[Introducción y motivación: hablar del mundo moderno, la importancia de la computación en la actualidad, en el futuro y la importancia de la seguridad de los dispositivos electrónicos.]

## Ordenadores en el mundo moderno

[Donde se pueden encontrar los computadores, que riesgos entrañan y que catástrofes pueden provocar fallos producidos por estos]

## Fallos

[Qué tipos (permanentes y transitorios) de fallos existen, qué pueden causarlos y que consecuencias pueden tener]

## ~~Solución~~

~~[Soluciones a los diferentes fallos]~~

# Estado del Arte[~40 páginas]

[Procesadores: Intel, AMD, ARM]

Tipos de procesadores, hablar de la extensión de la arquitectura ARM, de su juego de instrucciones, de su gran aceptación en el mercado.

Comentar diferentes arquitecturas. Procesadores más básicos vs más complejos.

[Tipos de fallos: Transitorios/Permanentes]

Single Event Upsets (SEU) - Inversión deun bit en un registro o en una memoria.

Single Event Transients (SET) - Inversión temporal en una señal indivuidual. Esto puede provocar alteraciones en señales importantes como "clk" o señales asincronas como un "reset".

Single effect function interrupt (SEFI) - En FPGAs basadas en SRAM, también pueden provocar cambios en el diseño, cambiando su funcionalidad.

Tener en cuenta que la radiacion tambien puede afectar de manera permanente provocando daños que solo se arreglarían cambiando las capas de silicio.[Fuentes de fallos: radiación, picos de energía, transistores quemados...]

[Tolerancia a Fallos: Transitorios, Físicos, Permanentes, Votadores, triplicación de HW, Reconfiguración dinámica]

Tolerancia a los diferentes tipos de fallos. SEUs, SETs, SEFIs, transistores quemados...

[Tolerancia a Fallos en CPUs]

[Buscar información] Hablar de procesadores TF, algún proyecto ya hecho, mandado al espacio o algo...

# Trabajo[~25 páginas]

[CPU, descripción de mi CPU, arquitectura, instrucciones...]

Como he hecho mi CPU, pasos, modulos, ALU, Juego de instrucciones elegido y soportado,

[Votadores, mis votadores...]

En nuestro caso, solo queremos probar los efectos sobre los registros, por lo tanto triplicamos el numero de registros para cada dato, y dejamos un unidad para cada módulo de la lógica del procesador.

Importante explicar que hay que dehabilitar la opción de "equivalent\_register\_removal"

[CPU + Votadores, cometarios sobre la eficiencia de la tolerancia a fallos aplicada...]

Espacio requerido por el procesador, comparar frente al procesador con TF, comparar tiempos de propagación, espacio ocupado en FPGA...

[Posibles errores? Puertas logicas extras necesarias, frente a CPU sin TF... ]

# Resultados[~25 páginas]

[Modulos]

Resultados de implementación, camino critico, tiempo de propagación mas largo...

# Conclusiones[~5-10 páginas]

# Bibliografía[~20-50 referencias]