# Introduccion y motivacion

**pero nos lo pensamos despues**

# Tema de fallos

**introduccion, radiacion**

## FALLOS TRANSITORIOS

Los fallos transitorios en puertas lógicas, desaparecen después de un tiempo de estabilización, si el error se produce temprano en el ciclo de reloj, la señal se estabilizará al valor correcto esperado y se evita que se propague el error. En caso contrario, no se estabiliza a tiempo y se guarda el valor incorrecto en el biestable que captura esta salida.

Un Fallo transitorio en un biestables, lo definimos en la inversión del valor guardado en ese biestable.

Algunos fallos no se propagan a otros componentes (modulo B al multiplexor). No moletan, no son importantes.

Pero otros fallos , por ejemplo en la salida del modulo A en la figura X, si hay un fallo, este si se propaga a los modulos siguientes, en ese caso en lugar de fallo en el modulo (A y B y mux) diremos que ha habido un error en el modulo

Este error provoca que los valores sean incorrectos en la siguiente etapa si se propaga.

IMPORTANTE NOSOTROS BUSCAMOS SOLUCIONAR TODOS LOS FALLOS AUNQUE NO PRODUZCAN ERROR

Figura X. dibujo de modulo multiplexor....

Para provocar estos errores se forzará a que el biestable guarde un valor incorrecto

## FALLOS PERMANENTES

Cuentas que pasa cuando un elemento (puerta o biestable) se rompe para siempre.

escrbi mas

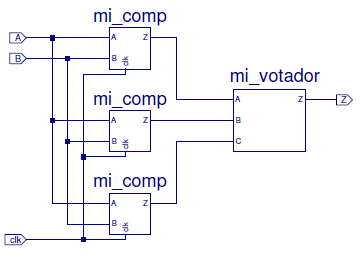
# TOLERANCIA A FALLOS TRANSITORIOS

Intro: nosotros solamente nos centamos en fallos tansitorios en este trabajo

## Como se consigue la tolerancia

Para evitar que los fallos transitorios se conviertan en errores y provoquen circunstancias imprevistas usaremos el método de “Triple modular Redundancy”.

El método aplicado consiste en triplicar el módulo, (o su funcionalidad modificando el circuito interno, pero manteniendo los resultados esperados), y utilizar un sistema votador para detectar, y si es posible enmascarar el fallo evitando que se produzca un error.



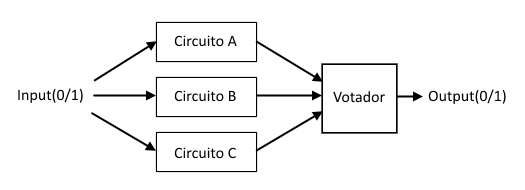


Figura 1. Sistema usando un Votador

El votador usado se conoce como “Majority Gate”. Como podemos ver en la “Figura 1”, recibe tres señales de tres módulos que aplican la misma función al mismo conjunto de valores. Las salidas de estos módulos deberían ser idénticas, salvo que se produzca un fallo en alguna puerta o biestable de los módulos. Este votador facilita la detección y la corrección de estos fallos del sistema.

## Votadores

Un votador es un sistema que recibe N entradas y tiene una salida que coincide con la mayoría de entradas.

Los votadores no están libres de fallos, ya que estos mismos pueden sufrir la radiación y provocarlos. Estos votadores están diseñados para intentar evitar un número limitado de fallos en los módulos repetidos dado por la función donde M es el número de fallos tolerados y N es el número de módulos repetidos, N debe ser un número impar.

La tabla de verdad del votador se puede consultar en Tabla de Verdad 1. sdfsd

Como vemos en la figura Figura 2

Dibujo de vtador 3 entradas de 1 bit

Tala de verdad del votador, diagrama de karnaught e implementación final con puertas lógicas

|  |  |  |  |
| --- | --- | --- | --- |
| A | B | C | F(ABC) |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |

Tabla de Verdad 1. asdasd



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | AB |  |  |  |  |
| C |  | 00 | 01 | 11 | 10 |
|  | 0 | 0 | 0 | 1 | 0 |
|  | 1 | 0 | 1 | 1 | 1 |

Mapa de Karnaught 1. F(ABC) = AB + BC + AC

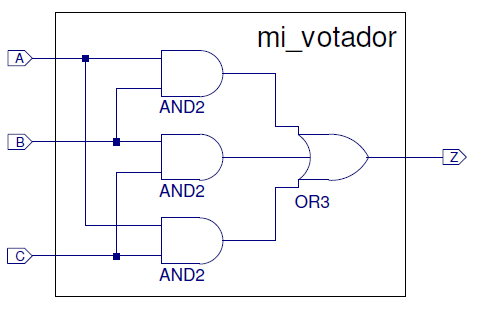
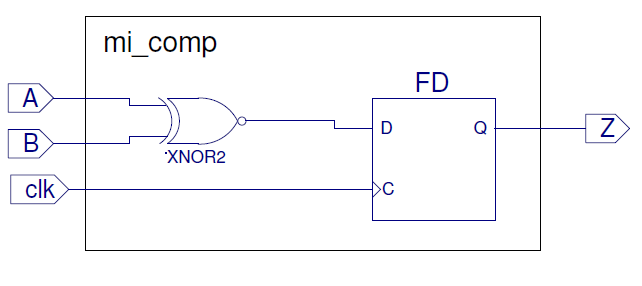
****

Figura 2. Esquema de votador

## Configuración del entorno Xilinx

TAREA



Implementas en vhdl un modulo sencillo con un bistble a la salida

modulo principal contiene 3 copias del modulo anterior conectados al votador

configuracion XILINX, busca como no eliminar elementos repetidos, poreque este diseño eliminaria los modulos repetidos y entonces tambien eliminaria elvotador

TAREA

## injeccion fallos

cada modulo sencillo tiene 3 lineas invertir salida, salida a 0, salida a 1

### MODO POST-INJECION

esto se hace mediante logica combinaional a la salida del biestable, para no introducir otro biestable que cambiaria la temporrizacion en etapas del procesador

### MODO PRE-INJECCION

en paralelo a la generacion de la entrada al biestable tambien funciona la lgica de injecciond efallo,

quizas el modo anterior introduce un retardo para las sieuigente etapa

este quizas iintroduce menos

# Procesador ARM

dices que es un cpu en genral

cuentas algo del ARM, arquitectura efieicenia en energia, y sobre su aceptaion en el mercado moviles, tablets

hemos elegido el cortex m4 porque

pretendemos ser compatibles ocn las instruccions, no pretendemos hacer el cortex m4

## Arquitectura del cortex m4

## nuestra arquitectura

cuantos nivels de segmentacion

dibujo

cuantas puertas sin telorancia fallos

# Resultados

Cuanto hardware con tolerancia a fallos

simulaciones injectando combinaciones de fallos y ver cuales son tolerados esperamos que muchos, y cuantos no tolerados dan error

graficas con barras verticales

podemos probar dos programas distitos si cambia la tolerancia a fallos

# Análisis de resultados,

ahora explicamos los resultados, si son importantes...

# Conclusiones

el trabajo termino bien

hicimos todo

es intersante

tiene futuro en trabajos como inge o docencia

# Bibliografía

(min 20 referencias, y menos de 90)

Use the "Insert Citation" button to add citations to this document.