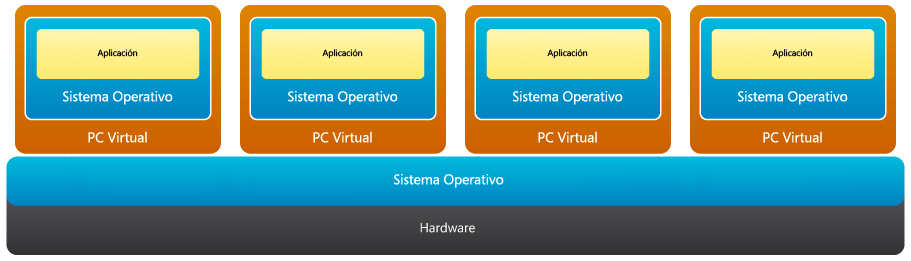
# CAPÍTULO 4) CAMPAÑA DE INYECCIÓN AUTOMATICA

En este capítulo se detallará el proceso de inyección de fallas automatizado y su posterior análisis. La automatización se llevo a cabo con el desarrollo de una aplicación que permitió la inyección sistematizada de fallas, la simulación del circuito y el posterior análisis de los resultados. La elección de desarrollar una aplicación que permitiera automatizar el proceso de inyección fue motivada por un pequeño análisis de los tiempos asociados a cada inyección. El conversor diseñado consta de 63 comparadores (estructuras analógicas donde se realizo la inyección) y cada uno de ellos posee 7 transistores, dando un total de 441 puntos de inyección. Por cada uno de estos puntos se realiza una simulación del tipo transitorio para cada nivel de tensión de entrada. Se definieron un total de 64 niveles de tensión de entrada lo que da como resultado más de 28.000 simulaciones. Por último, este proceso se realiza para ambos tipos de fallas, duplicando la cantidad de simulaciones. Con más de 56.000 simulaciones el tiempo requerido para realizarlas manualmente era considerablemente mayor que el requerido para desarrollar la aplicación que automatizará la etapa de simulación. La mejora del tiempo requerido por cada simulación fue notablemente reducida. Configurar el circuito, simularlo y guardar los resultados, requería aproximadamente 2 minutos de una persona. Para completar el total de las simulaciones una persona tendría que estar 78 días sin descanso inyectando las fallas. Con la aplicación el tiempo se disminuyo a 26 segundos por simulación en una computadora personal. Si bien el tiempo fue mucho mejor, a la aplicación le hubiera llevado 17 días terminar de simular, todavía un tiempo bastante alto. Se recurrió entonces a un entorno virtualizado, como se puede observar en la Figura 1 <FALTA REFERENCIA AUTOMATICA>, que permitía ejecutar 4 instancias de la aplicación. El establecimiento del entorno consumió menos de un dia de trabajo, logrando que las 56.000 inyecciones se realizaran en tan solo 5 días. (El ambiente de desarrollo y simulación se encuentra mas detallado en el APENDICE D)



En la Figura 2 <FALTA REFERENCIA AUTOMATICA y cambiar el numero por 2> se pueden observar las etapas involucradas en la campaña de inyección. El proceso se divide en etapas a realizar por el usuario y por el programa. Comienza por el diseño del circuito (explicado en el capítulo 2), en el cual se tuvieron que tener especiales cuidados a la hora de nombrar los nodos, debido a que el criterio de inyección tiene su base en los nombres de los mismos. Se especificó una convención de nombres para que los criterios fueran simples de definir.

En una segunda etapa, el circuito se exporta del capturador de esquemáticos a un formato estándar de SPICE, esto permite que se pueda utilizar cualquier tipo de capturador de esquemáticos, brindando una mayor libertad al usuario. De la misma manera, el modelo de la falla se especifica utilizando los elementos de las librerías estándares de SPICE.

Por último, el usuario determina un criterio de inyección, que le permite especificar tipo de transistor a inyectar, comparador o comparadores, nodos determinados, etc., e ingresa los datos a la aplicación.

.

Figura ) Interacción usuario-aplicación.

La aplicación, en una primera etapa, lee el archivo descriptivo del circuito, y analiza los nodos involucrados en el mismo. Con esta información compara los nodos con el criterio especificado y comienza a generar archivos circuitales con la falla en cada uno de los nodos determinados. La interfaz del programa para la inyección de la falla se puede ver en la Figura 3<FALTA REFERENCIA AUTOMATICA A LA FIGURA>



Una vez generado todos los archivos, la aplicación esta lista para comenzar a simular. Para esto, un modulo de la aplicación recorre la estructura de directorios donde se almacenaron los archivos y los carga en PSPICE (aplicación SPICE incluida en el paquete Cadence OrCAD). Al terminar cada simulación el archivo de salida es almacenado en un formato estándar conocido por su sigla en inglés CSDF (Common Simulation Data File). La aplicación se encarga de comprimir la información y almacenarla en la base de datos en Internet para su posterior análisis. Esto permitió la posibilidad de un análisis temprano, comparando los resultados de la aplicación contra los simulados manualmente.

La última etapa del proceso, consiste en el modulo de análisis, que se puede apreciar en la Figura 4<FALTA REFERENCIA AUTOMATICA A LA FIGURA>. La aplicación extrae una a una todas las simulaciones de la base de datos y determina si la falla produjo un error, comparando la salida binaria del conversor antes de la inyección (estado de funcionamiento correcto) y todos los demás puntos de la simulación. Esto le permite a la aplicación no solo determinar las fallas que solo producen errores, sino también la duración de los eventos y las variaciones de voltaje absolutas respecto de los niveles de tensión normales. Para cada simulación una fila con los datos es cargada en una tabla de formato estándar CSV (Comma Separated Values), la cual luego es importada por una tabla final de la cual se extraen todos los gráficos que permiten su posterior interpretación.



Figura ) Imagen del programa (pestaña de inyección).

El proceso de inyección consta de 4 pasos, detallados a continuación:

Figura ) Diagrama de proceso de inyección de fallas.

Figura ) Imagen del programa (pestaña de simulación).

El proceso de simulación consta de 4 pasos, detallados a continuación:

Figura ) Diagrama de proceso de simulación y análisis de falla.

Se puede observar en la Figura 5<FALTA REFERENCIA AUTOMATICA>, un ejemplo de una sección de la tabla que devuelve la aplicación



Figura ) Tabla obtenida como resultado del análisis del programa.

La longitud de la tabla se extendió a más de 56.000 filas de datos y junto a la ayuda de un procesador de hojas de cálculo (Microsoft Excel), se realizaron los análisis pertinentes a la campaña de inyección realizada, generando sub-tablas y gráficos de los datos recopilados. A continuación se presentan alguno de éstos análisis.

Cantidad de errores totales discriminando tipos de falla:

|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
| **Tipo de falla** | **Error** | **Éxito** | **Total** |
| *Rampa* | 1991 | 26233 | 28224 |
| *Exponencial* | 206 | 26231 | 28224 |

Tabla ) Tabla de porcentajes de errores según tipo de falla.

Figura ) Porcentaje de errores según tipo de falla.

A partir de estos gráficos de torta, apreciamos que la inyección de la falla tipo rampa genera mayor cantidad de errores de salida en el conversor. Como se mencionó anteriormente, este modelo de falla posee un inicio y fin de perturbación bien definido, a diferencia del modelo exponencial, cuya duración no posee un fin exacto. Sin embargo, en este tiempo, la falla tipo rampa genera una mayor perturbación en el equilibrio de corrientes del nodo afectado, permitiendo así que la falla se prolongue por la lógica decodificadora llegando a los bits de salida.

Cantidad de errores encontrados en función de la tensión de entrada y tipo de falla:



Tabla ) Tabla de fallas según nivel de tensión de entrada.

En la se puede observar una clara diferencia entre los errores producidos entre ambos tipos de fallas. Para los dos casos, con el aumento progresivo de la tensión de entrada se genera un amento de la cantidad de errores observados en la salida del conversor. En el caso tipo exponencial, el aumento generado es lento y de pocas variaciones, a comparación de los errores producidos en el caso tipo rampa que aumentan progresivamente y con variaciones entre cada nivel de tensión.

Figura ) Gráfica de dispersión, cantidad de errores según tensión de entrada (falla rampa).

Figura ) Gráfica de dispersión, cantidad de errores según tensión de entrada (falla exponencial).

Este tipo de resultados se apreció en los análisis realizados de manera manual ( y ), en donde los únicos momentos en que se producían los fallos en el transistor M12 (correspondiente al nodo NDOUT\_P) era cuando la tensión de entrada era superior a la tensión de referencia, lo que nos daba una salida del comparador igual a CERO, o sea, cuando el transistor M12 se encontraba cortado.

Cantidad de errores según nodo de inyección:



Tabla ) Tabla de fallas según nodo de inyección.

De la , se percibe que el nodo NDOUT\_P (transistor M12 de cada comparador) es el nodo sensible del circuito. Este nodo fue el único que generó fallos en la etapa de simulación manual realizada anteriormente.

Figura ) Grafica de barras de errores por nodos de inyección.

El nodo NDNEG\_P (transistor M3) es el menos sensible del circuito, como se puede apreciar, género la menor cantidad de fallas al ser inyectada la perturbación.

Distribución de errores según bit de salida:



Tabla ) Tabla de falla generadas según bit de salida.

A partir de la podemos destacar que todos los errores generados por la inyección tipo exponencial repercutieron tan solo en el bit MSB (ocasionando el total de las fallas). Realizando un pequeño análisis de la estructura interna del conversor se deduce que el efecto causado por la falla exponencial afecta únicamente a este bit debido a que el mismo es una conexión directa de la salida de un comparador, evitando toda la lógica del decodificador que se ve involucrada en la determinación de los restantes bits. Esto nos permite concluir que el decodificador posee un notable efecto de filtrado de los errores producidos por las falla del tipo exponencial. Una posible solución sería intercalar una estructura lógica que no modifique el valor de la salida binaria, como por ejemplo dos negadores en serie, etc. (APENDICE B).

Figura ) Gráfica de distribución de errores según bits de salida.

Distribución de errores según variación de amplitud y duración del evento:

A continuación se presentan dos gráficos de dispersión en los cuales se analiza la duración del evento transitorio y las variaciones de amplitud de éste. Los criterios utilizados para la condición de éxito/error son:

Según su amplitud:

* un CERO lógico, es todo valor de ‘x’ perteneciente al rango: - 0.001V <x< 1.001V. \*
* un UNO lógico, es todo valor de ‘x’ perteneciente al rango: 2.299V <x< 3.301V. \*

\*los rangos poseen una diferencia respecto a los limites definidos de UNO y CERO lógicos, para discriminar variaciones menores a 1mV.

Según su duración:

* un error está considerado como una variación de tensión mantenida por un tiempo mayor a 1ps (para valores menores el simulador demostró tener problemas para converger).

Para el modelo exponencial se analizó el único bit erróneo, y para el modelo rampa el bit LSB que fue el de mayor cantidad de errores(los gráficos de los restantes bits se encuentran en [APENDICE D]). En ambos gráficos se diferenció las inyecciones entre nodos NMOS y PMOS.

A continuación tenemos el análisis de la gráfica para el modelo exponencial:

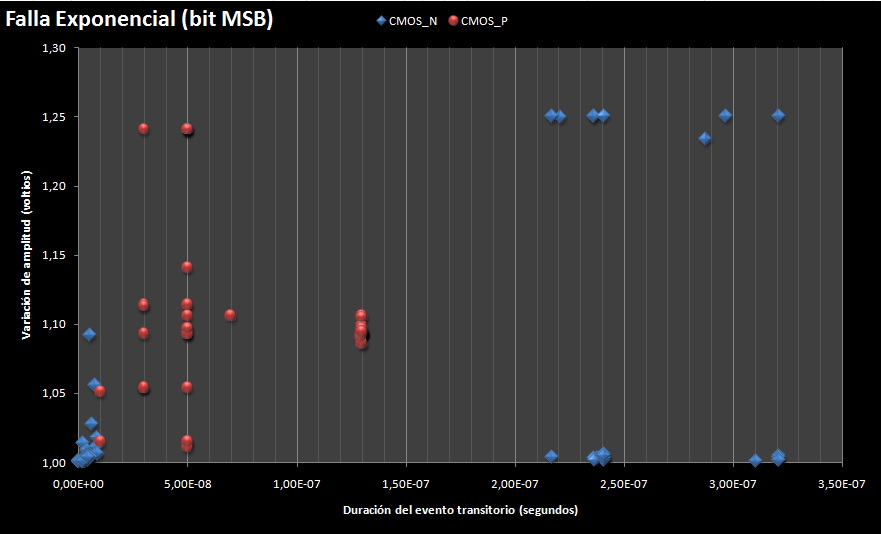


Figura ) Gráfica de dispersión, fallas MSB por inyección exponencial.

La se encuentra dividida en 4 cuadrantes. Se aprecia que la gran mayoría de los errores se encuentran ubicados en el cuadrante inferior izquierdo, lo que indica variaciones de amplitud pequeñas y de rápida recuperación. Observando el cuadrante superior izquierdo se encuentran principalmente las fallas causadas por inyección en nodos PMOS lo que revela que son nodos más sensibles pero se estabilizan rápidamente. En el cuadrante superior derecho encontramos similar cantidad de errores causados por inyección en transistores PMOS y NMOS, pero se percibe que los errores ocasionados a partir de inyecciones en nodos PMOS se encuentran agrupadas en un sector medio, indicando efectos similares de perturbación en todos ellos, a diferencia de los ocasionados por perturbar a los NMOS donde hay una clara variación en la duración de los eventos. Por último, en el sector inferior derecho, se observan únicamente errores formados a partir de inyecciones en nodos NMOS y de tiempos de recuperación más prolongados en comparación con los PMOS.

A continuación tenemos el gráfico de dispersión para las fallas en el bit LSB generadas por la inyección del modelo rampa.

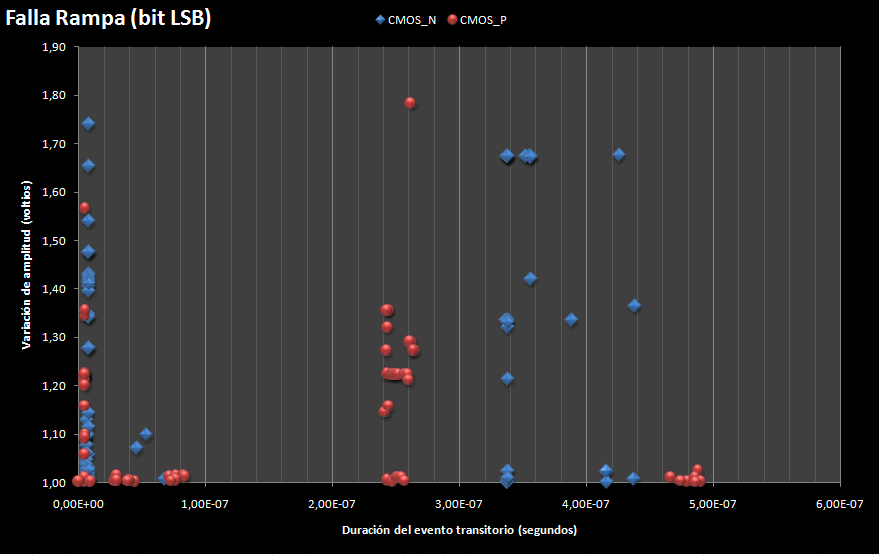


Figura ) Gráfica de dispersión, fallas LSB por inyección rampa.

Al igual que la anterior, la se divide en 4 cuadrantes. Podemos notar que más del 75% de los errores ocurridos en las simulaciones poseen variaciones menores a las 1.15 voltios y una duración de evento máxima de 10uS. El resto de los errores analizados poseen una duración mayor, entre 25 y 50uS, con variaciones bastante dispersas. Los transistores NMOS perturbados se encuentran dispersos en una franja de tiempo entre 33 y 43uS, en cambio, para los PMOS la dispersión es mayor pero se divide en dos franjas muy acotadas, una entre los 24 y 26uS, y la otra entre los 48 y 50uS.

Distribución de errores según tipo de transistor y falla:

En esta sección se presentará un análisis de la sensibilidad del conversor a errores según el tipo de transistor en el cual se inyecta, y también la sensibilidad de éstos al tipo de falla. La tabla a continuación muestra la cantidad de errores en veces y en porcentual:



Tabla ) Tabla de fallas discriminados por tipo de transistor.

Como se puede observar en el grafico, la sensibilidad de los transistores PMOS es considerablemente mayor que los NMOS, mostrando un 88% de errores para este tipo.

Figura ) Gráfica porcentual de errores según tipo de transistor.

Si bien la sensibilidad es mayor para los transistores PMOS, el tipo de falla inyectada nos da otros indicios de las sensibilidades del conversor a los efectos de los ASETs.

Figura ) Gráficas de cantidad de fallas según transistor y modelo inyectado.

Como se observa en la figura 23, la sensibilidad de los transistores NMOS a las fallas es prácticamente la misma. Sin embargo, la diferencia en la cantidad de errores para las inyecciones de los transistores PMOS muestra una fuerte diferencia revelando que los transistores PMOS son gravemente afectados por la inyección tipo rampa.