

Instituto Tecnológico de Costa Rica

Escuela de Ingeniería en Electrónica



Desarrollo de una herramienta de software para el análisis de diagramas de ojo

(Eye Data Structure Analysis Tool – EDSAT)

Informe de Proyecto de Graduación para optar por el título de Ingeniero en Electrónica con el grado académico de Licenciatura

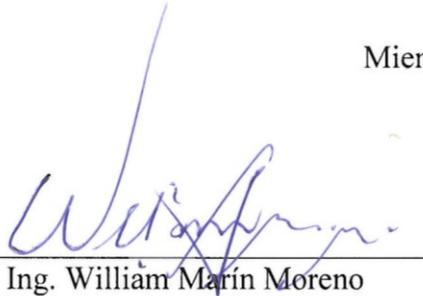
Ignacio Zúñiga Brenes

Cartago, Diciembre de 2009

INSTITUTO TECNOLOGICO DE COSTA RICA
ESCUELA DE INGENIERIA ELECTRONICA
PROYECTO DE GRADUACIÓN
TRIBUNAL EVALUADOR

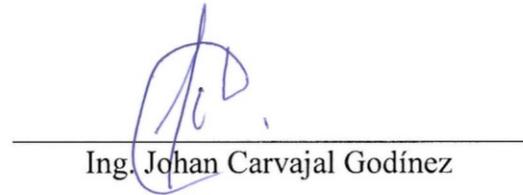
Proyecto de Graduación defendido ante el presente Tribunal Evaluador como requisito para optar por el título de Ingeniero en Electrónica con el grado académico de Licenciatura, del Instituto Tecnológico de Costa Rica.

Miembros del Tribunal



Ing. William Marín Moreno

Profesor lector



Ing. Johan Carvajal Godínez

Profesor lector



Ing. Aníbal Coto Cortés

Profesor asesor

Los miembros de este Tribunal dan fe de que el presente trabajo de graduación ha sido aprobado y cumple con las normas establecidas por la Escuela de Ingeniería Electrónica

Cartago, 17 de Diciembre, 2009

Declaro que el presente Proyecto de Graduación ha sido realizado enteramente por mi persona, utilizando y aplicando literatura referente al tema e introduciendo conocimientos propios.

En los casos en que he utilizado bibliografía, he procedido a indicar las fuentes mediante las respectivas citas bibliográficas.

En consecuencia, asumo la responsabilidad total por el trabajo de graduación realizado y por el contenido del correspondiente informe final.

Cartago, 17 de Diciembre de 2009



Firma del autor

Ignacio Zúñiga Brenes

Céd: 1-1200-0957

Resumen

El presente proyecto ha sido desarrollado con el fin de optimizar el análisis de los datos provenientes de las pruebas que se realizan a los puertos de alta velocidad (QPI/SMI) de los procesadores Intel® de la familia Itanium®. Los datos provenientes de dichas pruebas se presentan en forma de diagramas de ojo y se utilizan para evaluar las capacidades de los canales de transmisión. Anteriormente el análisis podía ejecutarse sólo hasta tener todas las pruebas hechas. La mejora que introduce este proyecto es la capacidad de hacer el análisis en paralelo con el desarrollo de la prueba, teniendo acceso a la información conforme es generada, además presenta indicadores que permiten ubicar los problemas haciendo más sencillo diagnosticarlos y plantear la estrategia para solucionarlos.

Palabras claves: puertos de alta velocidad, software, diagramas de ojo, análisis.

Abstract

The present project was developed to optimize the analysis of the data generated from the high speed ports (QPI/SMI) tests executed on the Itanium® family processors. Data collected from the tests is presented in eye diagrams and is used to evaluate the capacities of the port channels. In order to use the preceding analysis tool it was imperative to get the complete information of the test to analyze which meant a long time before analyzing the data, this project's main contribution to the process is the capability to do the analysis of the data as it is collected and it presents statistic information that helps to locate and solve problems.

Key words: high speed ports, software, eye diagrams, analysis.

INDICE GENERAL

Capítulo 1: Introducción	6
1.1. Problema existente e importancia de su solución	6
1.2. Solución seleccionada	7
Capítulo 2: Meta y Objetivos	10
2.1. Meta.....	10
2.2. Objetivos.....	10
2.2.1. Objetivo General	10
2.2.2. Objetivos específicos	10
Capítulo 3: Marco teórico	12
3.1. Diagramas de ojo o patrones de ojo	13
3.2. Líneas de transmisión diferenciales.....	20
3.3. Proceso de prueba de los productos Intel®	22
Capítulo 4: Descripción de la solución	25
4.1. Generalidades de la solución.....	25
4.2. Lineamientos iniciales del proyecto	26
4.3. Obtención de los datos y creación de la estructura de directorios	29
4.4. Interfaz gráfica para análisis de diagramas de ojo.....	32
4.4.1. Menú de Archivo:	36
4.4.2. Menú Análisis:	40
4.4.3. Menú de Configuración:	41
4.4.4. Propiedades y construcción de la ventana de diagrama de ojo	43
Capítulo 5: Análisis de Resultados	49
Capítulo 6: Conclusiones y recomendaciones	55
6.1. Conclusiones	55
6.2. Recomendaciones	56
Bibliografía y Referencias	57

INDICE DE FIGURAS

Figura 1.	Formación de un Diagrama de Ojo ideal. [5].....	13
Figura 2.	Formación de un Diagrama de Ojo real. [5]	14
Figura 3.	Esquema de propiedades de una señal (Rise time, fall time, overshoot y undershoot). [1]	15
Figura 4.	Propiedades de los diagramas de ojo. [1]	15
Figura 5.	Parámetros a analizar para productos Intel®.....	17
Figura 6.	Punto de entrenamiento y márgenes de variación posibles. [4]	18
Figura 7.	Diagrama de ojo con distorsión por Jitter. [1].....	19
Figura 8.	Esquema de interfaz QPI de Intel®. [3].....	21
Figura 9.	Formato de nombre del archivo de diagrama de ojo individual.	28
Figura 10.	Ejemplo de nombre de archivo de diagrama de ojo.	28
Figura 11.	Estructura de organización de los archivos de Diagrama de ojo.	29
Figura 12.	Diagrama de flujo del proceso de escritura de archivos y reporte.....	31
Figura 13.	Ventana principal del EDSAT.....	32
Figura 14.	Ventana de presentación del diagrama de ojo.....	34
Figura 15.	Ventanas del EDSAT, principal, diagrama y filtros secundarios.	38
Figura 16.	Ejemplo de archivo de sesión.	39
Figura 17.	Segmento de un diagrama de ojo con conteo de errores por punto.	41
Figura 18.	Ejemplo de acción ejecutada al presionar una vez el botón izquierdo...	44
Figura 19.	Ventana de filtrado generada con doble click.	45
Figura 20.	Menú de botón derecho.	46
Figura 21.	Ejemplo de archivo exportado a Microsoft® Excel® por el EDSAT.	47
Figura 22.	Esquema del proceso de prueba y corrección del producto.	52
Figura 23.	Proceso de análisis utilizando la metodología anterior con programas de Perl.....	53
Figura 24.	Proceso de análisis utilizando el EDSAT.	54

INDICE DE TABLAS

Tabla 1.	Descripción del lote analizado para pruebas de eficiencia del EDSAT.....	49
Tabla 2.	Resultados de tiempo de las pruebas realizadas al EDSAT.....	50

Capítulo 1: Introducción

El presente proyecto ha sido desarrollado para el departamento de Ingeniería de Desarrollo del Producto (PDE) de Componentes Intel de Costa Rica, consta de una herramienta de software que permite realizar análisis en tiempo real de los datos provenientes de los equipos de prueba y pertenecientes a las pruebas hechas a la sección de puertos de comunicación de alta velocidad conocidos como QPI (Quick Path Interconnect) o SMI (Scalable Memory Interconnect) de los procesadores de la familia Itanium®.

1.1. Problema existente e importancia de su solución

La nueva generación de interfaces en los chips Intel® utiliza buses unidireccionales con señales diferenciales para obtener velocidades de transmisión de datos hasta 6.4 GTs (Giga Transferencias por segundo). Para medir la calidad de los circuitos transmisores y receptores se utilizan diagramas de ojo en lugar de las tradicionales medidas de VIH/VOH, Setup, Hold timings. Para hacer la validación del diseño y la caracterización es necesario contar con una herramienta de software que permita analizar las características de los ojos.

Esta herramienta debe tener dos modos de operación: conectada al equipo de pruebas para permitir el análisis en línea de los datos durante las sesiones de prueba de ingeniería o desconectada del equipo y trabajar sobre los datos almacenados en los directorios generados luego de dichas sesiones. En ambos modos de operación debe tener la capacidad de realizar los mismos análisis y acceder a los mismos tipos de datos.

Los diagramas de ojo del transmisor (TX) y del receptor (RX) son almacenados en una estructura de memoria llamada EDS por sus siglas en inglés (Eye Data

Structure), las funciones de usuario que analizan e imprimen los archivos de datos desde el equipo de pruebas acceden a la información del EDS.

El problema es la ausencia de una herramienta de software para el análisis de los datos recopilados por el equipo de pruebas que permita acceder tanto trabajando en línea con el equipo de prueba como desde los archivos de datos generados posteriormente a las sesiones de pruebas de ingeniería.

El hecho de que los datos provenientes de las pruebas no puedan ser analizados en tiempo real sino hasta que las pruebas sean concluidas en su totalidad hace el proceso de toma de decisiones muy lento, puesto que tanto las pruebas ejecutadas como el análisis de datos generados pueden demorar varias horas por consiguiente el proceso completo de prueba/análisis puede consumir incluso un día completo de trabajo. Al contar con la disponibilidad de analizar en tiempo real los datos se tiene la capacidad de prever los comportamientos finales y plantear así las posibles causas y soluciones los problemas mediante las tendencias que presenten los datos parciales, reduciendo con esto los tiempos del proceso prueba/análisis.

1.2. Solución seleccionada

El proyecto se enfoca en el desarrollo de la herramienta de software que contiene funciones que permiten realizar diferentes tipos de análisis a los datos recopilados por el equipo de pruebas en las sesiones de prueba de ingeniería y producción. Estas funciones son:

- Recolectar diagramas de ojo y almacenar la información en un disco duro, cuando se trabaja en línea con el equipo de prueba.
- Leer la información desde archivos y llenar la estructura de datos para realizar los análisis cuando se trabaja en modo no en línea con el equipo de prueba.

- Indexar la información de cada una de las pruebas realizadas tomando en cuenta las características de número de identificación visual, número de identificación del núcleo, temperatura, tipo de prueba que se realizó, entre otras.
- Filtrar y desplegar diagramas de ojo únicamente de las líneas seleccionadas según el interés de los análisis que se desee realizar.
- Desplegar información de los diagramas de ojo seleccionados, específicamente su alto y ancho.
- Realizar un análisis estadístico de las características de los diagramas de ojo seleccionados, alto y ancho promedio, así como máximos y mínimos.
- Guardar la información recolectada así como la sesión en sí, de forma que dicha sesión pueda ser recreada en forma automática posteriormente para continuar el análisis de los datos.
- Generar reportes que resuman los datos claves de los análisis.
- Guardar los diagramas de ojo como imágenes para un análisis posterior.

Dado que el equipo de pruebas funciona desde una plataforma Windows®, se hace imperativo que exista una compatibilidad entre la plataforma y la herramienta en lo referente a la forma de manejo de memoria, puertos y demás dispositivos utilizados durante la operación para poder acceder, manipular y procesar los datos que se encuentran dentro del equipo de pruebas. Por este motivo se decidió utilizar para el desarrollo de la herramienta los lenguajes de programación C++ y C#, que además cumplen con las políticas de compatibilidad de la compañía, pues los programas y aplicaciones desarrollados a lo interno, deben ser sujetos de aprovechamiento para otros proyectos por lo que deben ser escritos en lenguajes de uso común en la compañía que permitan su modificación e implementación posterior. Además se

utilizaron herramientas ya desarrolladas como el Shmoodiff que realiza un análisis gráfico de los diagramas de ojo para su interpretación posterior.

Por el hecho de tener que ser ejecutada desde Windows®, la herramienta se diseñó en un ambiente gráfico, con una interfaz que posee diferentes ventanas que permitan ejecutar las opciones que estarán a disposición, desde una consola principal.

Capítulo 2: Meta y Objetivos

2.1. Meta

Comparar los diagramas de ojo de forma simultánea con las pruebas (en línea con el equipo de pruebas) y además poder generar los diagramas desde los archivos de datos que se generan durante las sesiones de prueba de ingeniería.

2.2. Objetivos

2.2.1. Objetivo General

Desarrollar una aplicación de software que permita el acceso a datos desde dos fuentes: el EDS y archivos de datos, aunque no simultáneamente de ambas fuentes; para desplegar y analizar diagramas de ojo a partir de esos datos.

2.2.2. Objetivos específicos

- Escribir funciones de usuario que sean capaces de imprimir en un directorio temporal los datos provenientes de las pruebas (modo en línea con el equipo de pruebas).
- Escribir funciones de usuario que permitan ejecutar una prueba de simulación a partir de los datos en un directorio, llenando el EDS con estos datos, para el caso del análisis en modo fuera de línea con el equipo de prueba.
- Desplegar los diagramas de ojo de manera individual o colectiva, de acuerdo con los criterios escogidos por el usuario.

- Crear menús que permitan la selección de criterios de usuario para escoger las líneas, puertos, unidades, que serán utilizadas para el análisis.
- Generar reportes y almacenar los datos para un análisis posterior.
- Realizar análisis gráfico de los diagramas de ojo utilizando la herramienta Shmoodiff.

Capítulo 3: Marco teórico

A continuación se detallan conceptos sobre comunicaciones digitales, puertos de comunicación de alta velocidad y la forma de caracterizarlos, también algunos detalles de los estándares propios de Intel® y las herramientas desarrolladas en la empresa para implementar estos estándares.

Cuando se debe analizar el comportamiento de enlaces de transmisión digitales es común realizar mediciones de diferentes parámetros para obtener una caracterización del desempeño de estos, entre los más comunes están: BER (Bit Error Rate o Tasa de errores de transmisión) que se refiere al porcentaje de bits recibidos erróneamente sobre el total recibido en un intervalo de tiempo, lo que permite determinar estadísticamente el número de bits que pueden ser transmitidos antes de que se produzca un error. Otro parámetro utilizado es la relación Señal a Ruido (Signal to Noise Ratio, SNR) que permite medir la degradación de la señal transmitida por efecto del ruido de un extremo al otro del canal de transmisión, que entre otras cosas permite determinar si los elementos utilizados poseen las características adecuadas para alcanzar la calidad que se pretende en el enlace.

Sin embargo, a pesar de la existencia de estos parámetros muchas veces la información que proporcionan no es suficiente para visualizar los problemas o determinar su origen; es en estos casos que es necesario analizar propiamente las señales que se transmiten en el enlace para caracterizarlas y asociar los problemas a causas específicas. Al analizar las señales transmitidas se encuentra información tal como: potencia de las señales, formas de onda, desfases (jitter) y niveles de ruido eléctrico, entre otras; todas estas características se pueden reunir en un solo diagrama para su análisis, este diagrama se conoce con el nombre de Diagrama de Ojo o Patrón de ojo.

3.1. Diagramas de ojo o patrones de ojo

Como se explicó anteriormente los Diagramas de Ojo (Eye Diagram, ED), corresponden a una técnica para el análisis de formas de onda digitales, básicamente un ED se forma con la superposición de múltiples datos transmitidos por el enlace, ya sea durante tiempo definido y cantidad de bits variable o para una determinada cantidad de bits con tiempo variable. La figura 1 muestra como se conforma un diagrama de ojo ideal y la figura 2 muestra un diagrama de ojo real.

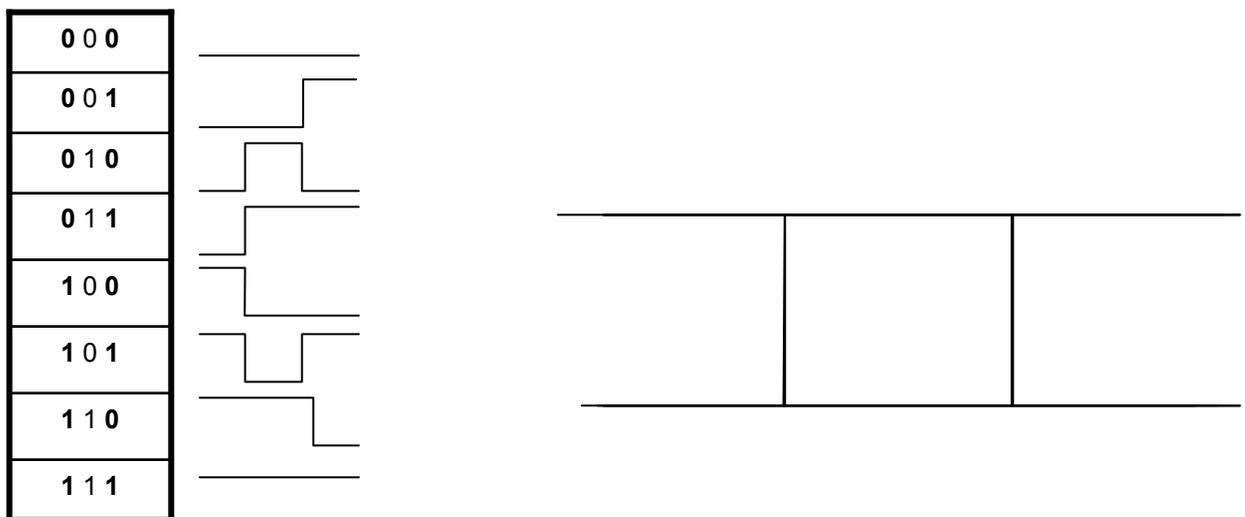


Figura 1. Formación de un Diagrama de Ojo ideal. [5]

En ambas figuras se muestran las secuencias de datos que se envían a través del dispositivo que se está analizando, así entonces al tener una señal digital real (Figura 2), en la que las transiciones (positivas o negativas) no son instantáneas, es posible apreciar como al sobreponer las secuencias de bits transmitidos se consigue observar una forma similar a un ojo, de ahí el nombre de esta técnica.

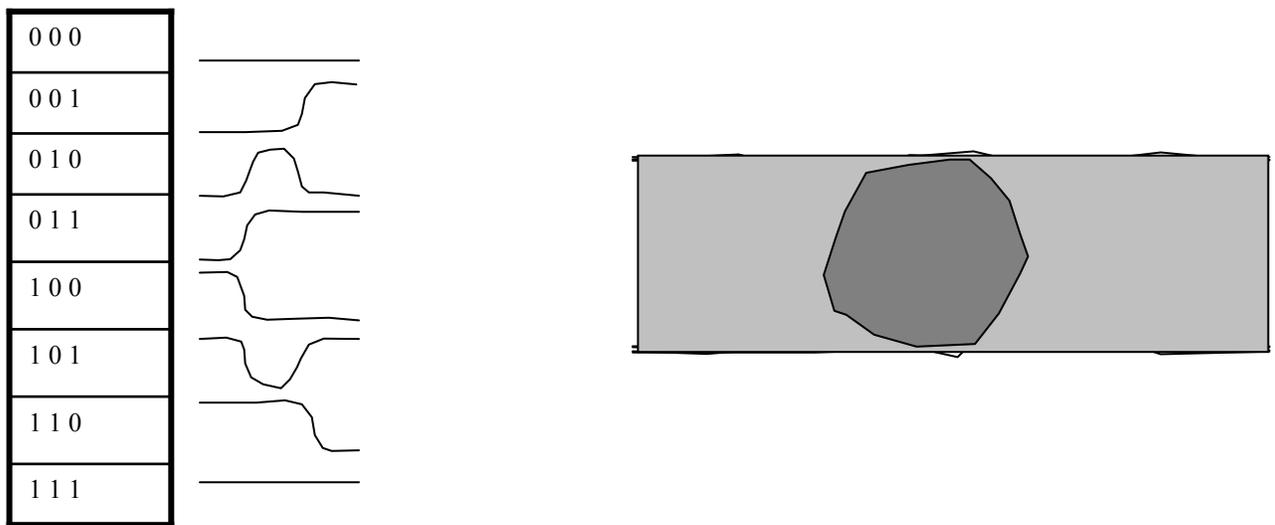


Figura 2. Formación de un Diagrama de Ojo real. [5]

La parte interna del ojo indica la zona de paso y se refiere a la zona en que existe la menor probabilidad de obtener datos erróneos producto del ruido y desfase. Una vez formado el diagrama es posible caracterizar el enlace en análisis, pues este contiene información que permite tomar decisiones para reconfigurarlo y obtener una mayor eficiencia. Esta reconfiguración tiene como objetivo ajustar el puerto para que las lecturas de datos se hagan en un punto lo más cercano al centro del ojo pues esto permite un rango de variación más amplio, lo cual hace al enlace menos susceptible al ruido y desfases siempre y cuando la distorsión no lo saque de la zona central del ojo.

Cuando se debe realizar el análisis de los ojos existen básicamente cuatro parámetros importantes que son: el nivel cero, el nivel uno, el cruce de amplitud y el cruce de tiempo que están asociados con los parámetros eléctricos comunes de tiempo de subida (Rise time), tiempo de bajada (Fall time), sobreimpulso positivo (Overshoot) y sobreimpulso negativo (Undershoot).

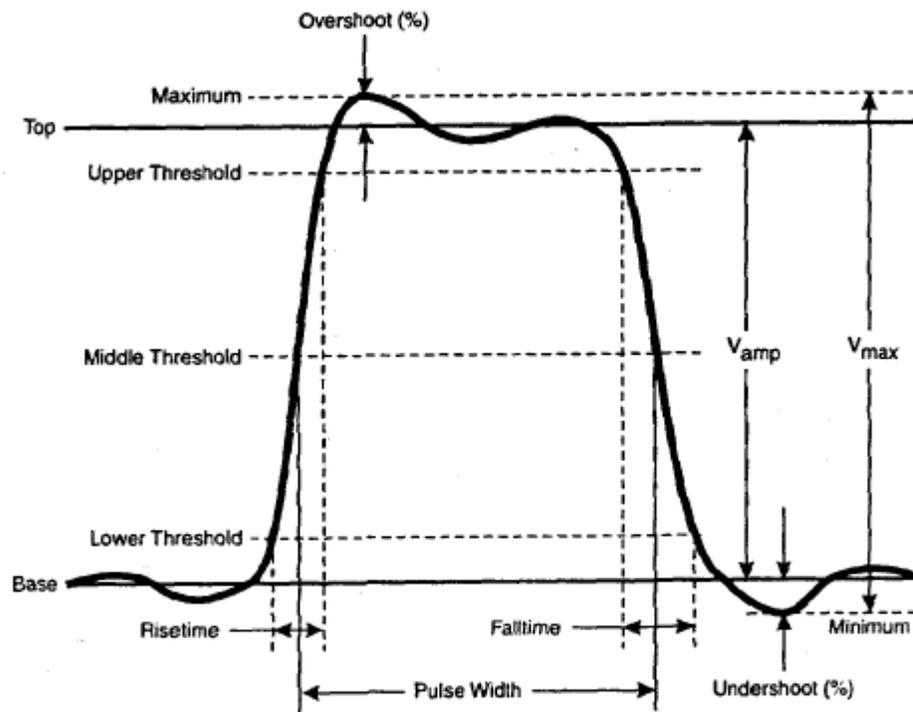


Figura 3. Esquema de propiedades de una señal (Rise time, fall time, overshoot y undershoot). [1]

Según estos parámetros de las señales que se observan en la figura 3 se obtienen los parámetros del ojo: niveles de uno y cero así como el cruce de tiempo y amplitud. En la figura 4 se puede observar estos valores en un diagrama de ojo y a continuación se explican estas propiedades.

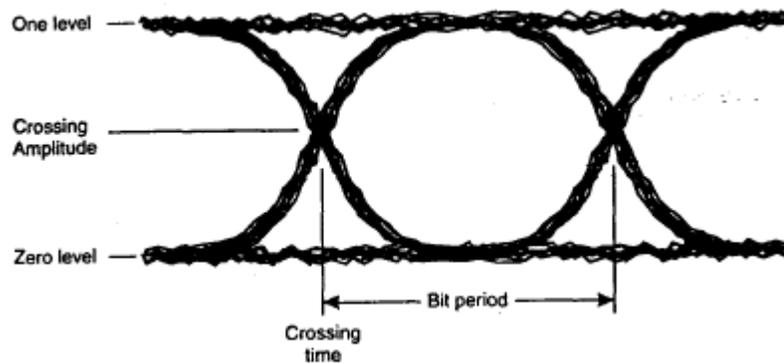


Figura 4. Propiedades de los diagramas de ojo. [1]

- **Nivel de uno:** es una medición promedio del valor de uno lógico de las señales transmitidas, como es entendido no todos los unos transmitidos alcanzan el mismo valor de tensión por lo que existe una banda delgada que puede ser considerada como un uno lógico, esta propiedad se determina entonces mediante un promedio de los valores alcanzados por los datos transmitidos.
- **Nivel de cero:** es el mismo concepto que para el nivel de uno, con la diferencia que este hace referencia al valor del nivel de cero lógico.
- **Cruce del ojo:** describe básicamente un mismo punto con dos valores importantes, tiempo y amplitud; se refiere entonces a los puntos donde empieza y termina el ojo, este cruce es sin más el valor (de tensión y tiempo), en que se produce el cruce de cero a uno y de uno a cero. Según estos cruces que delimitan el ojo se define el periodo del bit, que es el tiempo que tarda un bit en el canal.

En el caso de Intel® se han establecido algunos parámetros propios de análisis que se asemejan a los parámetros estándar de los diagramas de ojo pero que permiten caracterizar los productos para determinar si alcanzan o no las especificaciones de producción; estos son: ancho del ojo, alto del ojo y posición del punto de entrenamiento, como se pueden observar en la figura 5 y se procede a explicarlos.

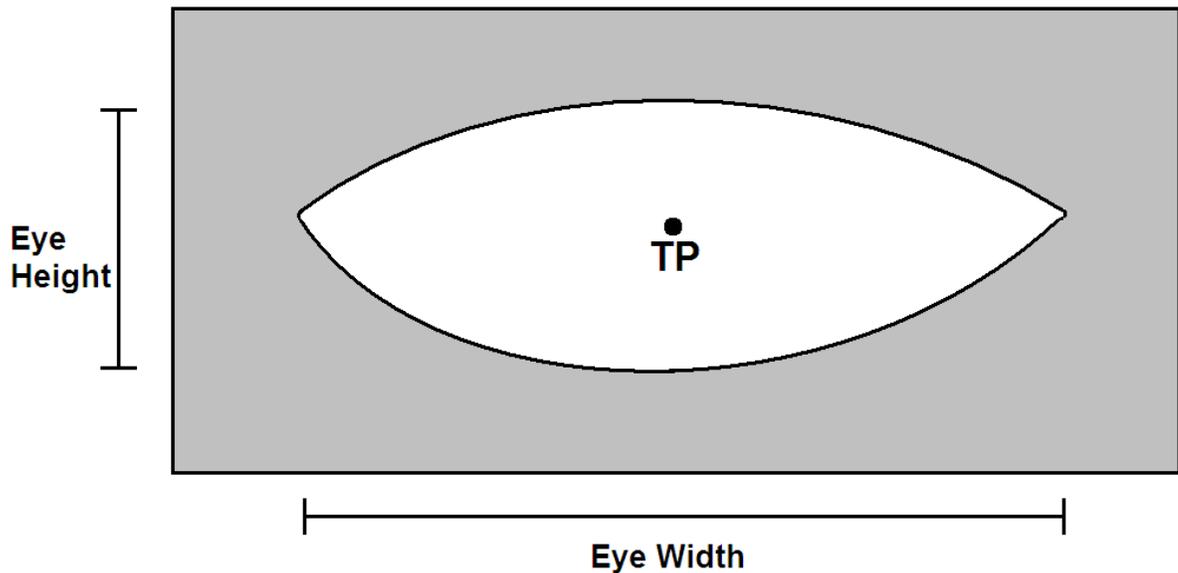


Figura 5. Parámetros a analizar para productos Intel®.

- **Posición del punto de entrenamiento:** es importante pues indica el punto exacto (de tiempo y voltaje) en que la lectura del dato será hecha, de forma que si existe distorsión en la señal la lectura podrá ser aún correcta, siempre y cuando esté dentro de la zona de paso, es por esto que se busca que este punto esté ubicado en el centro del ojo, pues esto maximiza los márgenes de variación tanto en tiempo como en voltaje. Para ejemplificar esto se puede suponer que se transmite la tercera palabra de la figura 2, donde se observa que existe una transición de alta frecuencia y asumiendo que la tensión al transmitir el 1 lógico alcanza solamente dos terceras partes del valor que representa un 1 lógico; si el punto de entrenamiento está por encima de este valor de dos terceras partes, la lectura en ese bit será un 0 y no un 1, más en cambio si el punto de entrenamiento se encuentra en el centro del ojo aún cuando la tensión no sea la correcta el dato se interpretará correctamente, el mismo caso ocurre en la escala del tiempo, si el punto de entrenamiento está corrido es

posible que suceda lectura cuando se está dando la transición entre un bit y el siguiente lo que podría ocasionar una interpretación errónea del dato.

- **Alto del ojo:** la altura del ojo representa el margen de variación que existe en la escala de voltaje, o lo que es lo mismo que tanto puede variar la tensión de una señal sin que se produzca una lectura errónea.
- **Ancho del ojo:** este valor representa que tanto desfase puede existir en la señal sin que se produzca una lectura errónea.

La figura 6 representa gráficamente estos parámetros tomados en cuenta para los productos Intel®.

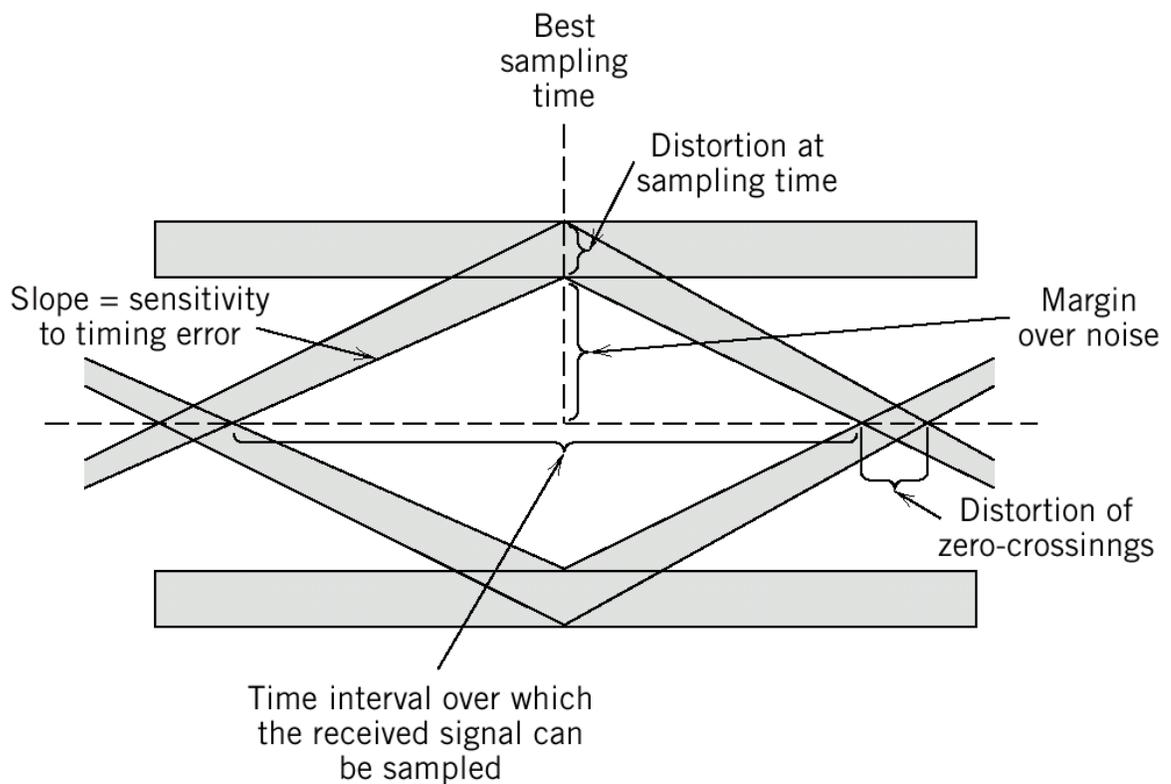


Figura 6. Punto de entrenamiento y márgenes de variación posibles. [4]

Otro valor que debe tomarse en consideración al analizar los ojos es el valor de Jitter o desfase, que corresponde a la desviación de fase con respecto a la posición ideal en el tiempo para una señal digital que se propaga en un canal de transmisión. Es un efecto que causa gran cantidad de problemas en el canal de transmisión, puede llegar incluso a comprometer la calidad y desempeño del enlace, trae consecuencias adversas como aumento en el BER y también es posible que el desplazamiento de las señales imposibilite al receptor a identificar las señales que están siendo transmitidas y lo haga perder el reloj de sincronismo. La figura 7 muestra un diagrama de ojo distorsionado por la acción del Jitter, en el que es posible deducir que el receptor tendrá problemas para hacer la interpretación de los datos.

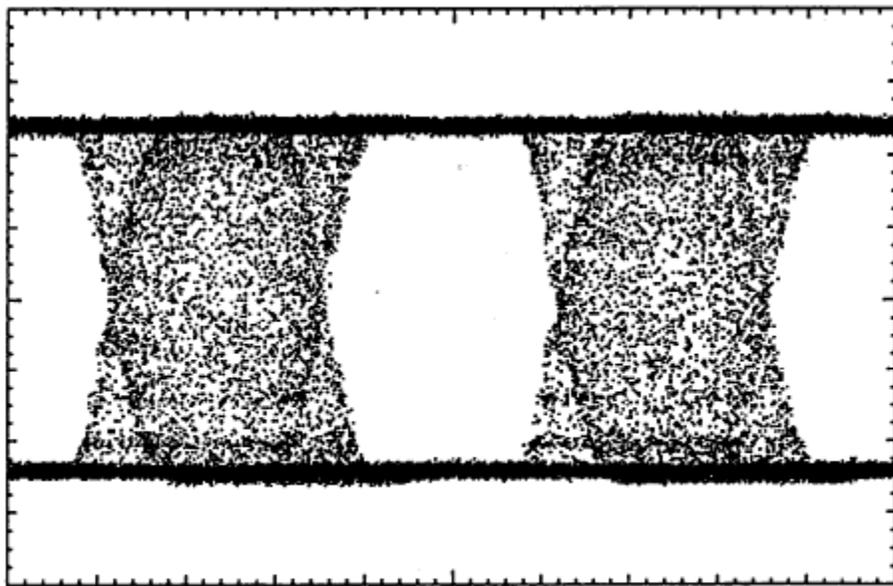


Figura 7. Diagrama de ojo con distorsión por Jitter. [1]

Por otra parte, uno de los mayores problemas que se presentan en la transmisión de señales digitales a alta frecuencia es el ruido eléctrico, es por esto que durante mucho tiempo en la industria se han hecho esfuerzos por atacar este problema, en

cuanto al área de las comunicaciones uno de los modelos más empleados se refiere a las líneas de transmisión diferenciales.

3.2. Líneas de transmisión diferenciales

Una línea de transmisión diferencial se refiere a una línea con dos hilos uno positivo D^+ y otro negativo D^- , de forma que al transmitirse el dato se separa en dos señales y que se reciben utilizando un amplificador diferencial, el cual al efectuar la resta de ambas señales obtiene el valor del dato original transmitido.

$$V_o = D^+ - D^- \quad (1)$$

Con: V_o = tensión de salida

D^+ = tensión de la terminal D^+

D^- = tensión de la terminal D^-

La ecuación 1 define la tensión de salida del amplificador diferencial ubicado en el receptor del enlace. Por ejemplo al recibir un 1 lógico se tendrá una tensión mayor a 0V en D^+ y una de 0V en D^- , por tanto en este caso la tensión de salida será positiva indicando el 1 lógico, caso contrario para un 0 lógico se tendrá una tensión mayor a 0V en D^- y una de 0V en D^+ lo que conllevará una tensión negativa en la salida, indicando el 0 lógico.

Esta técnica se basa en un principio muy sencillo que es: cuando una fuente de ruido introduce una perturbación al enlace lo hace casi por igual sobre D^+ y D^- , pero la perturbación se filtrará al restar ambas señales por lo que su efecto únicamente se verá en el canal de transmisión mas no después del amplificador o al menos su efecto se verá disminuido casi en su totalidad.

Particularmente en la construcción de enlaces de comunicación de alta velocidad, dadas las altas frecuencias de operación, estos sistemas son muy susceptibles al ruido por lo que se pone en práctica esta técnica de los enlaces diferenciales con lo que se consigue disminuir el efecto de este.

En el caso de los procesadores Intel®, la compañía desarrolló una interfaz conocida como QPI (QuickPath Interconnect), del cual se observa un esquema en la figura 8. Es una conexión punto a punto diferencial y unidireccional, teniendo por separado el bus de transmisión y de recepción de datos, utilizando una señal de reloj de hasta 3.2 GHz y haciendo transferencias en ambos semiciclos del reloj (Dual Data Rate), consiguiendo así una velocidad de transmisión de hasta 6.4 GT/s (Giga Transferencias por segundo).

Intel® QuickPath Interconnect A Peer Level Connection

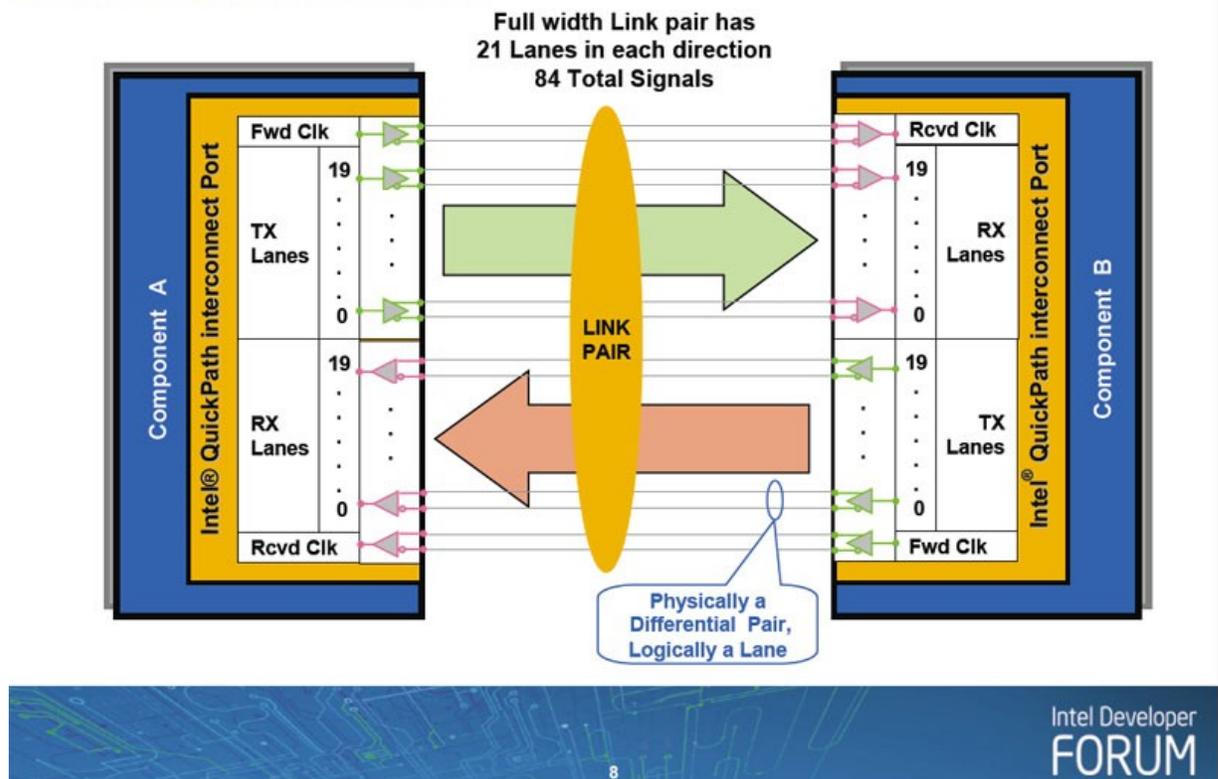


Figura 8. Esquema de interfaz QPI de Intel®. [3]

El proyecto de desarrollo de una Herramienta de Análisis para Diagramas de Ojo está orientado particularmente en acelerar el proceso de toma de decisiones sobre cambios necesarios en los diseños de los procesadores para la consecución de las especificaciones de producción que se ofrecen a los clientes. Se hace necesario

entender el proceso de prueba por el que los productos de Intel® pasan antes de poder ser entregados a los clientes de la compañía.

3.3. Proceso de prueba de los productos Intel®

El ciclo de producción de los productos Intel®, particularmente el de los procesadores comprende un proceso largo que comienza con el diseño del dispositivo, sigue con la validación y culmina con la producción en masa de los procesadores.

La planta ubicada en Costa Rica realiza el proceso de ensamble y prueba, lo que quiere decir que se ubica en los dos últimos pasos del proceso productivo de la empresa, y la herramienta desarrollada es aplicada para la etapa de pruebas de validación, durante la cual las primeras unidades se prueban para determinar si alcanzan o no las especificaciones de producción; cuando no son alcanzadas los datos recolectados en dichas pruebas se analizan para determinar las posibles causas, posteriormente se corrigen los errores de diseño y se repite el proceso hasta alcanzar las especificaciones.

Los resultados de las pruebas que son aplicadas a los diferentes productos son analizadas por PDEs (siglas en inglés de Product Development Engineers); el producto es analizado por secciones por ingenieros expertos en las diferentes áreas, que conocen tanto el diseño de los dispositivos como el proceso de fabricación y que analizan los datos colectados en las pruebas y determinan las posibles causas de las fallas. Entre las secciones en que se divide el producto están: memoria caché, consumo de potencia y análisis térmico, calidad y confiabilidad (Quality and Reliability, Q&R) y puertos de comunicación o dispositivos de entrada y salida de datos (I/O).

El interés del proyecto se centra en éste último bloque de puertos de comunicación, en particular los de alta velocidad o QPI, ya antes mencionados; dentro de este proceso de prueba debe asegurarse que todos los puertos cumplan con las

especificaciones, y para esto deben hacerse pruebas a las etapas de transmisión y de recepción por separado ya que están separadas pues son unidireccionales.

Como se discutió anteriormente para realizar la caracterización de estos dispositivos se utilizan Diagramas de Ojo. Ahora bien la principal limitante en el análisis radica en la gran cantidad de datos que se generan al realizar las pruebas por unidad, tomando el ejemplo del procesador con nombre código Tukwila (TKW), producto para el cual este proyecto fue desarrollado en primera instancia, el cual posee 8 puertos de alta velocidad, seis de los cuales cuentan con 20 líneas y otros dos con 10 líneas tanto para transmisión como para recepción que resulta en 320 diagramas de ojo individuales por unidad, lo que en un lote de 20 unidades significa 6400 diagramas de ojo, si además se realizan las pruebas a diferentes temperaturas los datos se multiplican por el número de pruebas.

El proceso de realización de las pruebas es confidencial, por esto se centrará la atención en la forma de organizar los datos y la forma de analizarlos. Dentro del departamento PDE de Intel® Costa Rica se desarrolló una técnica para almacenar los datos tanto del transmisor como del receptor de los enlaces en un mismo formato a pesar de ser recolectados de forma diferente. Esto se logró con la realización de una estructura de datos denominada EDS (por el inglés de Eye Data Structure), que utilizando el equipo de pruebas que posee la compañía permite imprimir a un archivo de texto los resultados de todas las pruebas hechas a un lote de unidades (caché, I/O, potencia, etc.) para su posterior análisis.

Como parte de la información obtenida de las pruebas de I/O se puede citar los diagramas individuales para cada línea de cada puerto de las diferentes unidades, así como los parámetros antes mencionados de alto y ancho del ojo, y la posición del ojo con respecto al punto de entrenamiento. Para la realización del análisis de los datos obtenidos con las pruebas es necesario determinar si las unidades están o no alcanzando las especificaciones y si no es así de donde provienen los problemas. Específicamente para I/O, los puertos y líneas que los presentan.

Cada diagrama se compone de una cierta cantidad de puntos que indican valores específicos de tiempo y voltaje (la cantidad de puntos depende de la resolución

escogida para la prueba), la prueba determina si la lectura del dato en cada punto fue o no correcta, obteniendo así un estado de paso/falla para cada punto, lo que lleva por consiguiente a la formación del ojo en el diagrama resultante. Debido a que los resultados que se buscan son estadísticos es necesario tomar grandes cantidades de información y analizarla en conjunto para establecer tendencias y formular conclusiones; es por esto que para hacer el análisis de los diagramas obtenidos en las pruebas es necesario hacer una superposición de estos para así lograr identificar las diferencias significativas.

Lograr analizar la cantidad de datos provenientes de las pruebas resulta complejo pues la información que se obtiene en el reporte del equipo de pruebas es un archivo de texto con una lista de diagramas uno después del otro sin tener la capacidad de sobreponerlos. Por esto se realizaron esfuerzos por conseguir herramientas que hicieran posible este análisis, siendo la más significativa una serie de programas de Perl, el primero permitía quebrar el reporte del equipo de prueba en archivos separados con diagramas individuales de cada línea y el segundo tomaba estos archivos y generaba un reporte en formato HTML con una sobreposición en escala de grises de los diagramas seleccionados con una aplicación limitada de filtros.

La principal problemática de utilizar este sistema es el tiempo necesario para conseguir los resultados pues quebrar el reporte del equipo de pruebas tardaba en promedio 12 horas y el análisis hasta 8 horas más, además de la poca flexibilidad para aplicar filtros a los datos pues los parámetros debían ser escritos en línea de comando al ejecutar el programa.

El EDSAT es una solución a estos problemas que brinda flexibilidad para filtrar y presentar los datos, cambiando de línea de comandos a ambiente Windows y que reduce el tiempo de varias horas a unos pocos minutos y que además provee facilidades que antes no existían para manipular los datos.

Capítulo 4: Descripción de la solución

En el presente capítulo se brinda una explicación detallada de ambas partes de la herramienta de software desarrollada para hacer el análisis de los diagramas de ojo producto de las pruebas hechas a los procesadores.

4.1. Generalidades de la solución

La herramienta consta de dos partes principales, las cuales fueron hechas bajo plataformas diferentes de programación. Estas partes son: obtención de los diagramas de ojo desde el equipo de pruebas y creación de la estructura de directorios y la segunda parte se refiere propiamente a la interfaz gráfica de análisis de los diagramas de ojo.

Una de las principales características de la herramienta es que presenta una mayor versatilidad en el análisis que las herramientas anteriores pues permite acceder a los datos de manera más amigable al usuario en un ambiente de ventanas y no desde una consola de mando de DOS como era la interfaz anteriormente disponible.

Como parte de la solución del problema se valoraron diferentes posibilidades de lenguajes de programación y se procedió luego a la escogencia del lenguaje C# para la interfaz gráfica y C++ para la parte de obtención de los datos del equipo de pruebas. Lenguajes como Java, Visual Basic fueron descartados por criterios de compatibilidad y de las prácticas comunes de los grupos de desarrollo de software de la compañía. De la misma forma Perl fue descartado para el desarrollo pues es un lenguaje interpretado y no compilado lo que lo hace más lento en las ejecuciones.

El criterio de compatibilidad fue determinante pues parte de las directivas de la compañía hace referencia a que toda herramienta de software debe ser abierta para su posterior mejora y al consultar con el departamento de desarrollo de software se indicó que el lenguaje utilizado es C#, por otra parte en el desarrollo de las funciones de usuario que obtienen los diagramas de ojo y crean la estructura de directorios desde el equipo de pruebas se utilizó C++ pues la estructura de datos EDS fue

desarrollada en este lenguaje por lo que era necesario utilizar el mismo lenguaje para ser compatible con las funciones existentes.

4.2. Lineamientos iniciales del proyecto

Durante la fase inicial del proyecto se determinó la necesidad de establecer reglas para la utilización de la herramienta a partir de las cuales se haría su desarrollo, obteniendo así un estándar para su utilización y desarrollo.

Como parte de estas reglas se determinó que la herramienta debía funcionar cargando archivos de texto con diagramas de ojo individuales que correspondieran a una línea y puerto específico. Dada la cantidad de archivos resultante de cada prueba también se hizo evidente la necesidad de crear una estructura de directorios que permitiera ordenarlos para facilitar su ubicación individual.

Para identificar los archivos individuales se creó un sistema para nombrar cada archivo, este sistema se compone de 11 campos separados por puntos que entregan información para la identificación única de una línea específica. Estos once campos son:

- **Nombre de la prueba:** indica la prueba a que se somete la unidad en análisis para tener una idea de los resultados a esperar, este apartado indica si el diagrama es de una o dos dimensiones, si la prueba fue realizada al transmisor o al receptor y la velocidad (en Mega Hertz) a la que fue ejecutada.
- **Número de puerto:** en este campo se pone el número correspondiente al puerto al que corresponde el diagrama.
- **Número de línea:** este campo corresponde al número de línea dentro de un puerto del diagrama de ojo.
- **Número de identificación visual:** este campo contiene el número de identificación de la unidad, cabe resaltar que este número es único de cada unidad ensamblada.

- **Temperatura:** cada una de las pruebas que se ejecutan son realizadas a una temperatura específica, algunas son hechas a altas temperaturas para verificar el comportamiento de las unidades ante condiciones de estrés, este campo brinda el detalle de la temperatura a la cual fue ejecutada la prueba que generó ese diagrama particular.
- **Lote de unidades:** al ingresar las unidades al piso de producción ingresan como parte de un lote y no en solitario, este campo provee la información del lote al que pertenece la unidad correspondiente a ese diagrama.
- **Operación:** durante el proceso de pruebas cada unidad es sometida a diferentes pruebas individuales u operaciones, cada una de las cuales tiene una identificación numérica asociada, este campo brinda la información de dicha operación que generó el diagrama. La operación está ligada otras condiciones como temperatura, entonces la misma prueba ejecutada a dos temperaturas tiene entonces dos códigos de operación diferentes para hacer la distinción.
- **Sumario:** este campo indica el sumario al que corresponde el diagrama, normalmente la mayoría de las pruebas corresponden al primer sumario (generalmente sumario 1A), únicamente cuando las unidades fallan en la prueba se les da por así decirlo una segunda oportunidad para descartar problemas de conexión en el equipo de pruebas y otros contratiempos, esta información permite identificar si a la unidad se le realizó la misma prueba más de una vez.
- **Planta:** este campo indica el lugar físico o planta donde se realizó la prueba, por ejemplo si la prueba fue realizada en Costa Rica o en EEUU.
- **Equipo de prueba:** en este campo se detalla el código del equipo de prueba que permite saber en cual equipo específico se realizó la prueba que generó el diagrama.
- **Número de identificación del núcleo:** este campo es un indicador de la ubicación del núcleo de la unidad dentro de una oblea (coordenadas X y Y) así como del lote de obleas al que corresponde ese núcleo, este también es un número único por cada unidad.

Al juntar todos estos campos se obtendría un nombre compuesto según se observa en la figura 9:

Prueba
Puerto
Línea
ID visual
Temperatura
Lote
Operación
Sumario
Planta
Equipo de prueba
ID de núcleo

Figura 9. Formato de nombre del archivo de diagrama de ojo individual.

Todos estos campos presentan un indicador único para cada archivo de cada una de las pruebas que se realizan a las diferentes unidades además de proporcionar información particular que permite filtrar la información para obtener datos provenientes solo de ciertas pruebas o ciertas partes de estas. Un ejemplo de nombre para un archivo de diagrama de ojo se observa en la figura 10:

IO_RX_ODD_EYE_HEIGHT_WIDTH_2400
0
0
2V827060B0004
20
IB109A19
7207
2A
XXX
tester3
F830961B_019_+03_-03

Figura 10. Ejemplo de nombre de archivo de diagrama de ojo.

Donde: XXX es el código de la planta, información confidencial.

En general el nombre del archivo sería de la siguiente manera:

IO_RX_ODD_EYE_HEIGHT_WIDTH_2400.0.0.2V827060B0004.20.7207.2A.XXX.teste3.F830961B_019_+03_-03.txt

Como se mencionó anteriormente aparte de la estructura del nombre de cada archivo es necesario tener una estructura de directorios que permita organizar estos archivos de manera que archivos específicos puedan ser ubicados en una forma más sencilla, pensando en esto se determinó que los diagramas estarían organizados según la figura 11

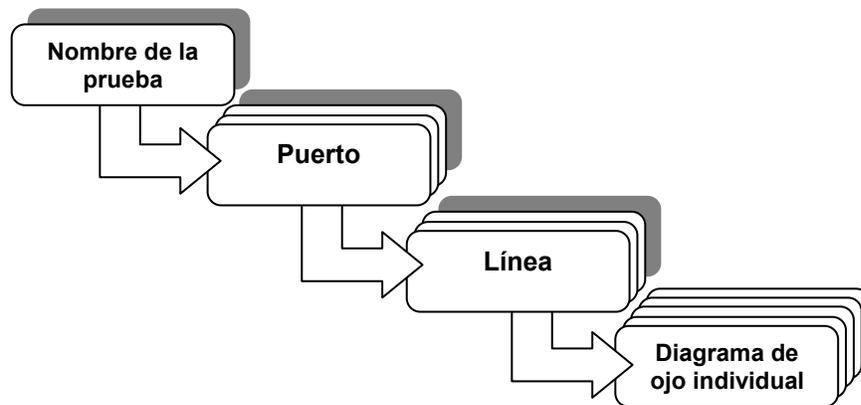


Figura 11. Estructura de organización de los archivos de Diagrama de ojo.

Otro de los aspectos importantes que se especificó al empezar el proyecto fue la información que debían contener los archivos además de propiamente los diagramas de ojo, se determinó que información sobre las características de alto y ancho del ojo debían ser incluidos así como la cantidad de puntos en el eje X y eje Y además de sus unidades respectivas y al final de cada uno una tabla de simbología que indicara a cuantos errores correspondían las letras que forman el diagrama para poder interpretarlo correctamente.

4.3. Obtención de los datos y creación de la estructura de directorios

Dado que los resultados de todas las pruebas se escribían en un solo archivo de reporte y no en varios archivos separados, fue necesario escribir nuevas funciones y

modificar algunas que ya existían en el código de manejo del equipo de prueba de manera que se generaran los archivos individuales cumpliendo con la estructura de directorios necesaria (explicada en la sección anterior) de forma paralela con la escritura del reporte original.

Entre los requerimientos de esta parte estaba el hecho de que este procedimiento debía ejecutarse únicamente cuando la herramienta estaba abierta pues el escribir a dos archivos simultáneamente (el reporte original y el archivo con el diagrama individual) consume más tiempo que solamente escribir un archivo, aún y cuando son unos cuantos milisegundos más este tiempo en el equipo de pruebas es muy valioso y se debe limitar al mínimo, por lo que durante el periodo de producción en el que no existirá un análisis de los datos obtenidos no es necesario crear los archivos individuales sino solo el reporte original. Esto se logró creando una variable global en el equipo de prueba que por defecto es falsa y únicamente cuando la herramienta está abierta se establece verdadera y revisando esta al ejecutar las pruebas se determina si se debe o no ejecutar el procedimiento paralelo.

Al momento de desarrollar esta parte del proyecto se procedió a hacer una función que creara el archivo cumpliendo el formato antes descrito y posteriormente tomara la información del diagrama y la escribiera en el archivo. Luego de esto se identificó la función que escribía el reporte y se determinó que la información correspondiente a cada diagrama era escrita individualmente y simplemente se concatenaba al final del archivo, por lo tanto se modificó de forma que la misma información que iba al reporte se enviaba a la función que escribía el archivo individual.

Para la realización de esta parte de la herramienta fue necesario no solo conocer procedimientos de C++ sino del lenguaje propio del equipo de prueba llamado CorTeX que contiene funciones particulares para aprovechar los recursos disponibles en este y que tiene diferencias significativas con el C++ estándar, también fue necesario conocer y manejar las funciones que permiten obtener y escribir los datos desde el EDS. El esquema de la figura 12 se puede observar un diagrama de flujo del funcionamiento de este segmento.

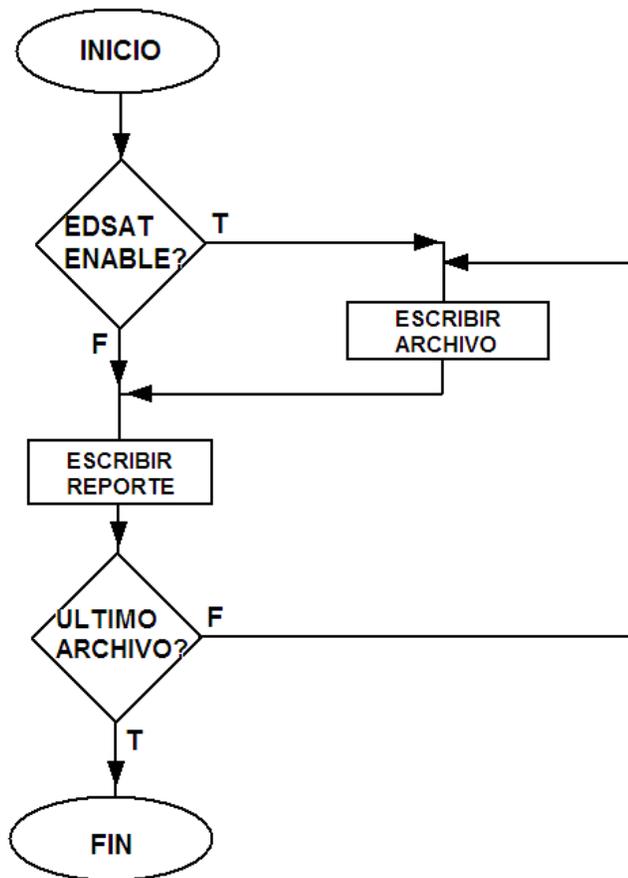


Figura 12. Diagrama de flujo del proceso de escritura de archivos y reporte.

Además del desarrollo de las funciones que permiten crear la estructura de directorios y demás que se explicaron anteriormente se desarrolló una función para hacer posible la simulación de las pruebas a partir de datos provenientes de sesiones de prueba anteriores, de esta forma es posible cargar el EDS con datos que no provienen del equipo de pruebas sino de archivos guardados en el disco. Para validar las nuevas pruebas que se introducen al proceso de los productos es necesario reservar tiempo en el equipo de prueba únicamente para valorar si la prueba se está ejecutando correctamente o tiene algún problema. Esta funcionalidad desarrollada permite entonces hacer la validación sin necesidad de estar conectado al equipo de pruebas, de forma que el desarrollo de nuevas pruebas resulta más ágil pues no se depende de la disponibilidad de tiempo en el equipo de pruebas.

Para esta funcionalidad fue necesario desarrollar una nueva función que leyera los datos escritos en los archivos de texto y los convirtiera a datos que se podían introducir en el EDS utilizando las funciones propias de este. Cada punto del diagrama en el entorno del EDS se identifica como un punto de fallo/paso, al leer de los archivos es necesario identificar el significado de los caracteres para convertirlos a este tipo de puntos del EDS, además cada uno de los nombres de archivo proporciona información importante que es guardada en variables globales del sistema del equipo de pruebas. Cuando se ejecutan pruebas reales estos valores se leen directamente de las unidades pero en el caso de simulación estos datos deben ser escritos manualmente.

4.4. Interfaz gráfica para análisis de diagramas de ojo

En este apartado se explica el funcionamiento de la herramienta en su parte gráfica de interfaz con el usuario que realiza el análisis, se hace mención a las diferentes opciones y facilidades que contiene la herramienta. En la figura 13 se observa la ventana principal que se presenta al ejecutar la herramienta.

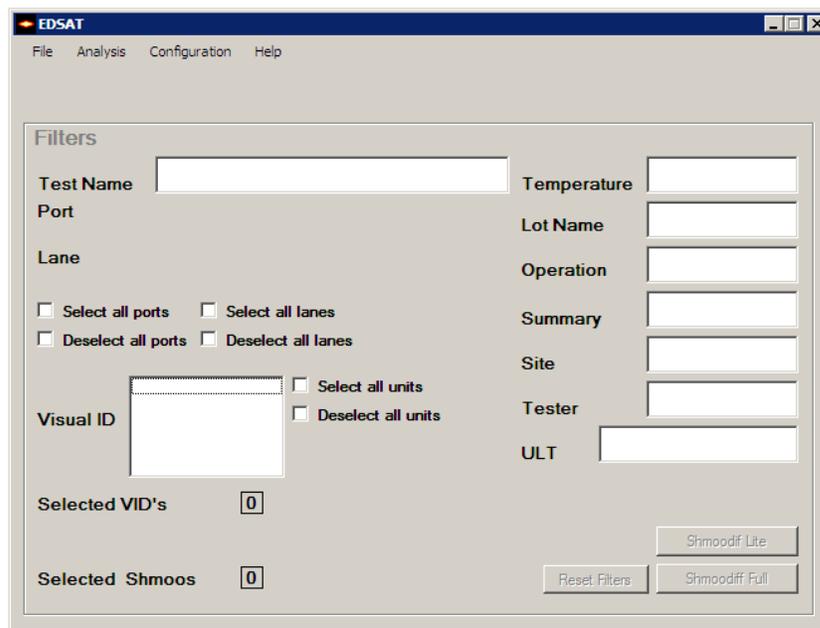


Figura 13. Ventana principal del EDSAT.

Puede observarse entonces en la figura 13 una distribución de campos y algunos indicadores que se refieren a las opciones de filtrado para las que tiene capacidad la herramienta. Para hacer estos filtros posibles se decidió que el nombre de los archivos debía tener el formato antes mencionado, de esta manera el nombre de cada archivo se secciona, obteniendo así la información para llenar las cajas con las opciones disponibles en la información que se carga, también se puede apreciar que existen menús de archivo, análisis, configuración y ayuda, cuyas opciones se explicarán posteriormente.

Estos filtros tienen capacidad de selección múltiple lo que permite hacer combinaciones complejas de criterios para obtener datos específicos que sean de un interés particular, en el caso de los campos de puerto y línea los campos de selección son generados dinámicamente dependiendo de la información disponible, lo que quiere decir que si solo existe información de dos líneas no aparecerán más opciones para seleccionar que esas dos; particularmente esto se hizo pensando en la aplicación del EDSAT para otros productos que no tuvieran la misma cantidad de puertos o líneas que el procesador TKW para el que el EDSAT fue diseñado en primera instancia.

Además de esta ventana existe otro tipo de ventana propia del EDSAT donde se imprimen los diagramas de ojo resultantes al aplicar los filtros, como se observa en la figura 14. En esta ventana se observa el diagrama de ojo con sus escalas particulares en tiempo y voltaje así como las estadísticas propias del diagrama y de los archivos que lo componen, de esta manera se pueden observar los valores mínimos y máximos de alto y ancho de los ojos que componen el diagrama y en el mensaje de ayuda de estas cajas de texto se muestra la ruta completa del archivo que tiene la característica.

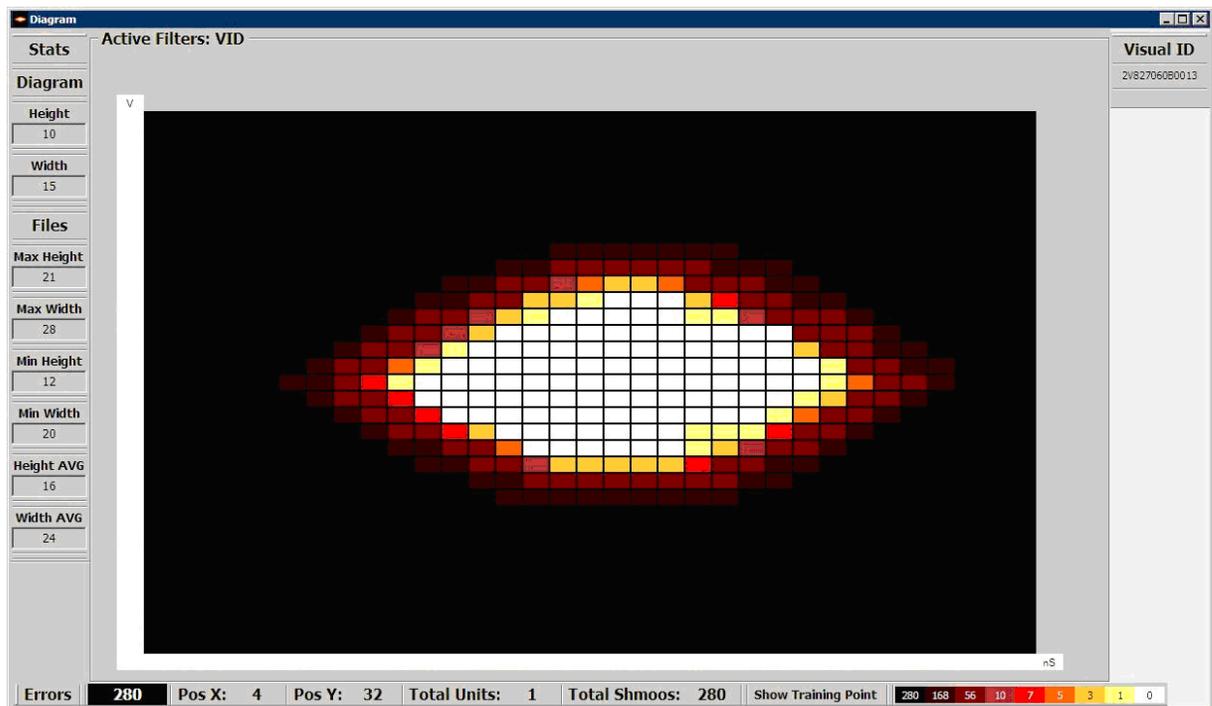


Figura 14. Ventana de presentación del diagrama de ojo.

Nota: la información estadística desplegada en el diagrama de la figura 14 es considerada confidencial por lo que los valores reales fueron sustituidos por la cantidad de espacios existentes en el diagrama.

Por otra parte al lado derecho de la ventana se pueden observar los identificadores de las unidades que fallan en un punto específico, estas se muestran dinámicamente al hacer click sobre un punto en el diagrama. También la información de la ubicación así como la cantidad de fallos que existieron en el punto seleccionado se puede observar en la barra inferior; fuera de esto se tiene una representación de la escala de colores, la cantidad de errores que significa cada color así como la cantidad de archivos que conforman el diagrama y la cantidad de unidades presentes en él, todo esto también en la barra inferior.

Es importante señalar que todas estas barras de información son movibles por el usuario para que éste pueda tener una configuración personalizada de la

herramienta; posteriormente se explicarán las funciones que existen propiamente dentro de la ventana de presentación del diagrama de ojo.

De manera más general lo más importante de conocer en cuanto al EDSAT en sí, es la estructura interna de este, pues su composición es la que hace posibles las funciones disponibles en los menús. La característica más sobresaliente del EDSAT es la manera en que se almacenan los datos provenientes de los archivos y la información que se almacena de estos en la memoria de la aplicación.

Al contrario de lo que podría pensarse el EDSAT no almacena en memoria toda la información que está guardada en los archivos desde que se carga un directorio sino hasta el momento en que se aplican los filtros y se pide generar el diagrama. Esto responde a que a no saturar la memoria de la computadora con información que no está siendo utilizada, aunque haga más lenta la generación del diagrama, de manera que únicamente se cargan los nombres de los archivos y su dirección en disco. Esta medida se tomó pues el EDSAT se ejecuta principalmente desde el equipo de pruebas y es prioritario que este posea disponible el máximo de memoria posible para que las pruebas ejecutadas no produzcan errores por falta de memoria.

Para guardar la información de cada archivo se creó una clase llamada "Eyes" que contiene una propiedad para cada opción de filtrado o lo que es lo mismo para cada parte en que se descompone el nombre de un archivo y una propiedad extra que indica si el objeto calza o no con el criterio de filtrado, de forma que objetos de tipo "Eyes" se puedan agrupar utilizando como criterio cualquiera de estas propiedades.

Es importante resaltar que dado que uno de los requerimientos primordiales del EDSAT es ser capaz de trabajar en línea y no en línea con el equipo de prueba, también existen dos formas de presentación de la aplicación, cuando el EDSAT está en línea todos los análisis se hacen cargando archivos de un directorio previamente definido por lo que se desactivan algunas funciones como abrir directorio y abrir sesión (que serán explicadas posteriormente) y se cargan algunos valores por defecto.

Habiendo comprendido este hecho se puede proceder entonces a la explicación de las diferentes funciones del EDSAT, estas funciones por menús son: para el menú de

archivo: abrir directorio, abrir sesión, salvar directorio, salvar sesión, cerrar, salir; para el menú de análisis: análisis comparativo, en el menú de configuración: esquema de color, modo de despliegue de caracteres y opciones; por último en el menú de ayuda existe un mensaje de información sobre la versión de la aplicación, a continuación se explica detalladamente cada función y su desarrollo.

4.4.1. Menú de Archivo:

Abrir directorio: pide al usuario un directorio en disco donde se encuentran los datos que se quieren analizar, con respecto a esta escogencia establece este como el directorio de trabajo o raíz y además referencia un objeto “escuchador de archivos”, para determinar cada vez que un nuevo archivo es escrito en la raíz o sus subdirectorios, esto permite entonces que la información de los filtros se actualice cuando nuevos datos son incluidos en el directorio de análisis. Esta funcionalidad fue especialmente creada para el modo en línea pues pueden crearse nuevos archivos provenientes de nuevas unidades y que serían puestos a disposición en la aplicación para su análisis.

Guardar directorio: desarrollada especialmente para el modo en línea, permite mover los archivos del directorio raíz a una dirección escogida por el usuario, de manera que se pueda continuar el análisis posteriormente sin necesidad de estar en línea con el equipo de prueba.

Esta medida responde a la forma de trabajar en la compañía pues el equipo de pruebas se puede utilizar únicamente bajo reservación y por un tiempo limitado, de manera que cuando se ejecutan pruebas para analizar los datos puede que el tiempo disponible no sea suficiente por lo que sería necesario reservar más tiempo y por tanto resulta mucho más eficiente copiar los datos y analizarlos posteriormente desde otro lugar para no saturar las horas de uso del equipo de pruebas.

Guardar sesión: se compone de dos partes, primero una que permite copiar los datos del directorio raíz (solo se copia, no se mueven los datos) a un lugar diferente escogido por el usuario, esta parte únicamente se ejecuta si se está trabajando en

modo en línea, para tener un acceso a los datos posteriormente como ya se ha explicado.

La segunda parte de la función permite guardar un archivo de sesión, para recrear posteriormente todas las ventanas de análisis que estén abiertas a la hora de salvar, de forma que si el tiempo en el equipo de prueba se termina, se necesita continuar el análisis y no perder tiempo, puede guardarse la sesión en el estado actual y reiniciar el análisis posteriormente desde el mismo punto donde se dejó.

Este archivo de sesión se compone de varias partes, primeramente se escribe el directorio donde se encuentran los datos, luego listas con las rutas relativas a ese directorio de los archivos que pertenecen a las diferentes ventanas, cada una de estas listas se encuentran separadas por líneas de texto que indican el inicio y fin de ellas. La primera lista de este archivo contiene las referencias a todos los archivos de la ventana principal o sea todos los disponibles en el directorio, luego de esta se escriben las listas de archivos contenidas en cada una de las ventanas de filtros subsecuentes y por último listas de los archivos de las ventanas de diagrama. En la figura 15 se puede apreciar las diferentes ventanas a las que se hace mención.

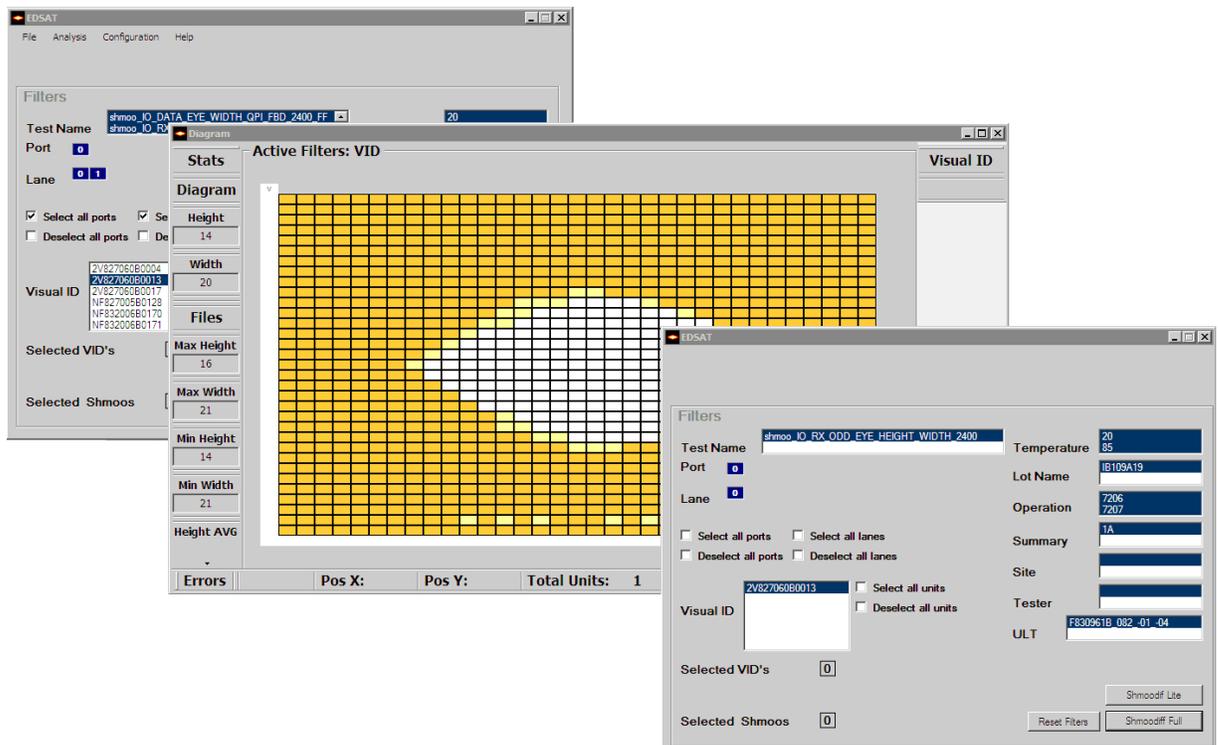


Figura 15. Ventanas del EDSAT, principal, diagrama y filtros secundarios.

En la figura 16 se aprecia el contenido de un archivo de sesión, se pueden apreciar entonces la forma en que se guardan las listas que componen las diferentes ventanas del EDSAT.

```

start_root_dir
C:\Documents and Settings\izunigal\Desktop\New Folder
start_eyes_list
\shmoo_IO_DATA_EYE_2D_QPI_FBD_2400_UF.0.0.2V827060B0004.20.IB109A19.7207.2A.XXX.tester03.F830961B_019_+03_-03.txt
\shmoo_IO_DATA_EYE_2D_QPI_FBD_2400_UF.0.0.NF832006B0171.85.IB109A19.7206.1A.XXX.tester03.F830962A_087_+04_+00.txt
\shmoo_IO_DATA_EYE_2D_QPI_FBD_2400_UF.0.1.NF832006B0170.85.IB109A19.7206.1A.XXX.tester03.F830962A_087_+05_-01.txt
\shmoo_IO_DATA_EYE_2D_QPI_FBD_2400_UF.0.1.NF832006B0171.20.IB109A19.7207.1A.XXX.tester03.F830962A_087_+04_+00.txt
\shmoo_IO_DATA_EYE_WIDTH_QPI_FBD_2400_FF.0.1.NF827005B0128.85.IB109A19.7206.1A.XXX.tester03.F830962B_105_+03_+00.txt
\shmoo_IO_FT_2D_Freq_VIO_SHMOO_B1617.9.10.2V831019B0066.107.B16SHMOO.7203.1A.XXX.tester03.F9109170_829_-04_+00.txt
end_eyes_list
start_diagrams
start_diagram
\shmoo_IO_RX_ODD_EYE_HEIGHT_WIDTH_2400.0.0.2V827060B0013.20.IB109A19.7207.1A.XXX.tester03.F830961B_082_-01_-04.txt
\shmoo_IO_RX_ODD_EYE_HEIGHT_WIDTH_2400.0.0.2V827060B0013.85.IB109A19.7206.1A.XXX.tester03.F830961B_082_-01_-04.txt
end_diagram
start_diagram
\shmoo_IO_RX_ODD_EYE_HEIGHT_WIDTH_2400.0.0.2V827060B0017.85.IB109A19.7206.1A.XXX.tester03.F830961B_082_-01_-04.txt
end_diagram
start_diagram
\shmoo_IO_FT_2D_Freq_VIO_SHMOO_B1617.9.10.2V831019B0066.107.B16SHMOO.7203.1A.XXX.tester03.F9109170_829_-04_+00.txt
end_diagram
start_diagram
\shmoo_IO_FT_2D_Freq_VIO_SHMOO_B1617.9.10.2V842157D0057.107.B1617.9.10.XXX.tester03.F910917A_868_-03_-03.txt
end_diagram
start_diagram
\shmoo_IO_RX_EVEN_EYE_HEIGHT_WIDTH_2400.2.5.2V848089C0003.107.DVDATA02.7251.1A1.XXX.tester04.F925991A_368_+06_+01.txt
\shmoo_IO_RX_EVEN_EYE_HEIGHT_WIDTH_2400.2.5.2V848089C0003.5.DVCOLD01.7251.1A.XXX.tester04.F925991A_368_+06_+01.txt
\shmoo_IO_RX_EVEN_EYE_HEIGHT_WIDTH_2400.2.5.2V848089C0003.5.DVCOLD01.7251.1A1.XXX.tester04.F925991A_368_+06_+01.txt
end_diagram
start_diagram
\shmoo_IO_DATA_EYE_WIDTH_QPI_FBD_2400_FF.0.1.NF827005B0128.85.IB109A19.7206.1A.XXX.tester03.F830962B_105_+03_+00.txt
end_diagram
start_diagram
\shmoo_IO_DATA_EYE_2D_QPI_FBD_2400_UF.0.0.NF832006B0171.85.IB109A19.7206.1A.XXX.tester03.F830962A_087_+04_+00.txt
\shmoo_IO_DATA_EYE_2D_QPI_FBD_2400_UF.0.1.NF832006B0170.85.IB109A19.7206.1A.XXX.tester03.F830962A_087_+05_-01.txt
\shmoo_IO_DATA_EYE_2D_QPI_FBD_2400_UF.0.1.NF832006B0171.20.IB109A19.7207.1A.XXX.tester03.F830962A_087_+04_+00.txt
end_diagram
end_diagrams
start_edsats
end_edsats

```

Figura 16. Ejemplo de archivo de sesión.

Abrir sesión: en esta función se ejecuta el proceso inverso al anterior, es decir, a partir de un archivo de sesión se crean las ventanas y se agrega la información correspondiente a cada una de las ventanas. Básicamente se leen los nombres de los archivos y se crean las listas de objetos tipo “Eyes”, explicados anteriormente y a partir de estos se genera todo el análisis posterior que se desee.

Cerrar: esta función únicamente cierra todas las ventanas generadas y limpia toda la información para volver al estado inicial de la aplicación, o sea sin información cargada; antes de ejecutar esto pregunta al usuario si desea guardar la sesión antes de cerrar el análisis actual.

Salir: ejecuta únicamente una acción, cierra la aplicación y limpia la memoria, si está trabajando en modo en línea borra el directorio donde se guardaron los datos para no saturar la computadora del equipo de pruebas con datos que nadie más utilizará.

4.4.2. Menú Análisis:

Análisis comparativo: esta función dentro del EDSAT provee una funcionalidad especial que permite comparar los datos provenientes de dos o más diagramas independientes haciendo una sobreposición de ambos, el procedimiento es realmente sencillo, en las ventanas de diagrama que se desee comparar se debe seleccionar entre las opciones de menú del botón derecho del ratón la opción de enviar la información a la ventana de análisis comparativo, posteriormente se abre la ventana desde el menú de análisis y el resultado será la generación de una nueva ventana de diagrama con la información sobrepuesta de todos los diagramas seleccionados.

Para lograr esta funcionalidad se realizó un procedimiento mediante el cual cada vez que un diagrama se selecciona para comparar se copia la lista de objetos "Eyes" del diagrama y se concatena con la que existe en la ventana de análisis comparativo.

Ahora bien cuando se desea iniciar un nuevo análisis comparativo con información proveniente de otros diagramas es necesario borrar las listas para empezar de nuevo, para este efecto existen dos botones propios de esta ventana, uno que realiza la limpieza de la lista y el otro que crea el diagrama con los datos actualizados que se le envían.

4.4.3. Menú de Configuración:

Este menú contiene las opciones de presentación de datos de la herramienta, la configuración de las diferentes formas de presentación de los diagramas de ojo, referentes a color e información a desplegar en los diagramas. Con respecto al color existen tres esquemas posibles completamente configurables por el usuario, una escala de tonos azules, una de tonos rojos y una de grises, y la información que cada punto del diagrama contiene también es configurable, puede escogerse entre números, letras o espacios en blanco.

Los números representan la cantidad de diagramas individuales que fallaron en el punto, al observar la figura 17 se pueden apreciar los números en cada punto del diagrama, de la misma forma si se selecciona desplegar letras, aparecerán en cada punto letras que corresponderán a cantidades específicas de fallos, esto pues las letras son la manera estándar en que se presentan los diagramas de ojo en los reportes del equipo de pruebas, por lo que es lo que los ingenieros están acostumbrados a observar e interpretar.

280	280	280	280	277	264	245	218	177	125
280	280	279	277	261	228	196	142	88	43
280	279	274	258	225	188	125	64	27	14
279	271	258	222	183	107	53	23	10	2
269	256	229	182	104	44	20	8	2	
256	234	188	113	48	16	8	1		
239	188	117	51	18	5	1			
211	153	73	24	6	1				
241	189	130	41	11	6				
260	234	183	115	48	14	6			
273	261	236	184	123	49	16	7	2	
279	274	266	234	195	118	58	25	11	4
280	277	276	265	239	204	151	79	41	16
280	280	278	277	271	244	208	170	113	72
280	280	280	279	278	275	258	237	205	158
280	280	280	280	279	278	276	269	257	237

Figura 17. Segmento de un diagrama de ojo con conteo de errores por punto.

Cuando se selecciona no desplegar información en el diagrama únicamente se ven los puntos de colores sin datos, utilizando la degradación de colores para conseguir el efecto que permita identificar la cantidad de errores.

Además de estas opciones también es posible configurar la herramienta para establecer las configuraciones por defecto, siendo posible también cambiar la posición de las barras de estadísticas para que sean ubicadas según el gusto del usuario. Esta opción de valores por defecto crea y lee un archivo de configuración en una dirección estándar en la misma carpeta donde se instala la aplicación; este archivo de configuración es creado la primera vez que se ejecuta la aplicación, si por alguna razón este archivo es borrado la aplicación cargará valores definidos por defecto al instalar la aplicación.

Por último fuera de los menús se encuentran tres botones en la pantalla principal, estos realizan funciones específicas, los botones se habilitan hasta que los datos están cargados en el EDSAT y están listos para ser analizados, el primer botón "Reset Filters" vuelve el estado de todos los filtros al valor por defecto que equivale a seleccionar toda la información disponible para la primera unidad en la lista de números de identificación visual. El segundo botón "Shmoodiff Full" ejecuta directamente el programa original escrito en Perl que genera un reporte en formato HTML y el último botón "Shmoodiff Lite" crea y genera la ventana que contiene el diagrama de ojo como el de la figura 14.

Este último ejecuta la función más importante de los tres botones y funciona de la siguiente manera: primeramente al presionar el botón se revisa la lista de "Eyes" existente que contiene todos los archivos disponibles y se establece cuales de estos objetos coinciden con los criterios de filtrado, posteriormente se crea una nueva ventana de diagrama y se copian los objetos coincidentes como propiedades de esta.

4.4.4. Propiedades y construcción de la ventana de diagrama de ojo

Aunque la ventana principal del EDSAT es la de aplicación de filtros, sin duda la más importante es la ventana de presentación del diagrama de ojo, pues es esta la que contiene de forma gráfica la información que desea analizar.

Al hablar de esta ventana, primeramente es importante mencionar que dado que la información a ser desplegada es diferente siempre, por ejemplo puede que existan diagramas de 24x24 puntos o diagramas de 33x33 puntos, la herramienta debe ser capaz de brindar la misma información en ambas situaciones y además se debe ocupar el espacio de la mejor forma posible. Para este efecto se creó una clase especial de objeto que identifica cada punto del diagrama y se le llamó "Pixels", cada objeto de esta clase es en esencia una caja de texto común pero con propiedades específicas del tipo de datos que representan.

Al momento de crear el diagrama se leen los archivos correspondientes a los criterios de filtrado y se cargan los datos en la memoria de la herramienta, como todos los archivos que componen un diagrama tienen las mismas dimensiones existe un método para saber cuantos de estos archivos tuvieron una falla en un punto específico necesitando solo conocer sus coordenadas (x,y), este número de fallas por punto es una propiedad particular de los objetos "Pixels", cada uno de estos puntos del diagrama se crean por separado y son agregados a una lista, el tamaño y la ubicación en pantalla de cada uno se establece dinámicamente dependiendo del tamaño de la ventana, de forma que no se pierda información cuando esta se hace más grande o más pequeña. Además de los Pixels que contienen información propia de los archivos analizados se agrega a cada diagrama una fila y una columna más con las escalas que se están utilizando.

Para realizar el análisis resulta muy importante que la interacción con los datos sea sencilla y rápida, por esto se agregaron acciones específicas asociadas a los botones del ratón (uno y dos clicks) que facilitan el proceso.

Click único botón izquierdo

Cuando se presiona una única vez el botón izquierdo del ratón sobre uno de los puntos del diagrama la información sobre este punto es actualizada en las barras de información que están a los lados de la ventana; lo primero que sucede es que el punto específico se marca en verde así como sus valores en las escalas para identificar exactamente que punto es, posteriormente los códigos de identificación visual de las unidades que fallaron en el punto seleccionado aparecen listadas en la barra de información de la derecha y en la barra inferior se indica la cantidad de archivos que fallaron en ese punto con su color correspondiente y la posición en X y en Y del punto. Una vez que se selecciona un punto diferente del diagrama el primero vuelve a su estado original y la información de la nueva selección es desplegada. Un ejemplo de esta acción puede observarse en la figura 18.

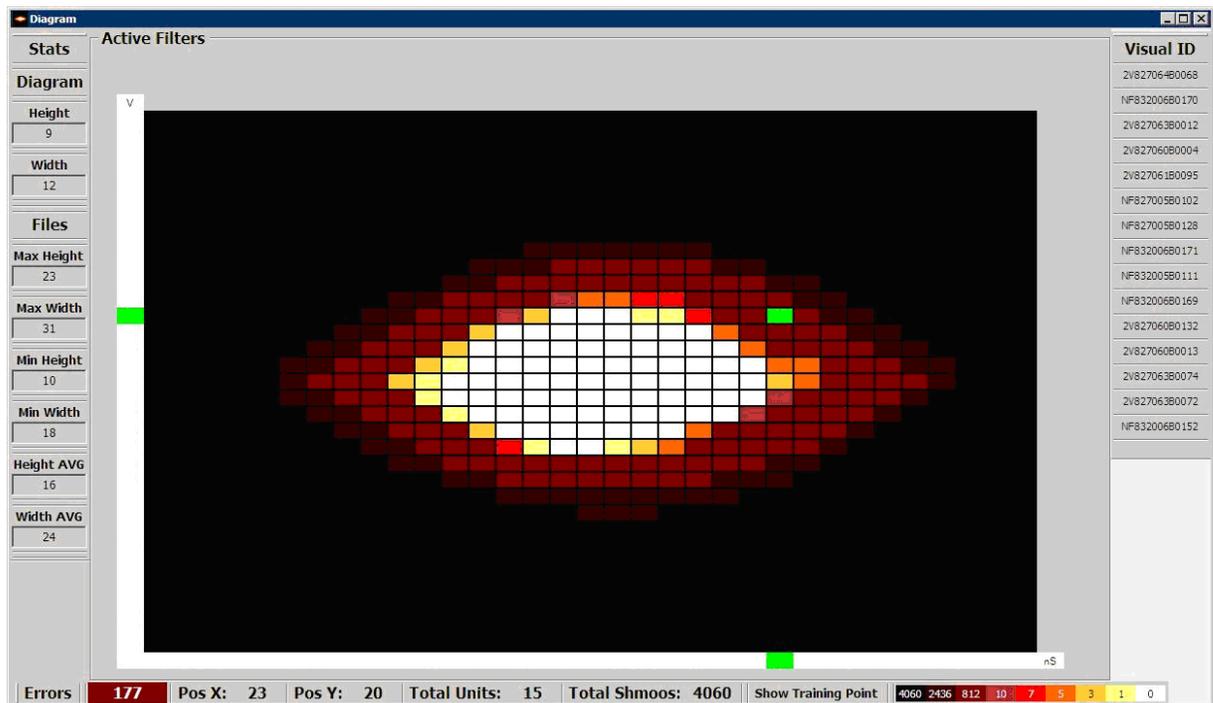


Figura 18. Ejemplo de acción ejecutada al presionar una vez el botón izquierdo.

Nota: la información estadística desplegada en el diagrama de la figura 18 es considerada confidencial por lo que los valores reales fueron sustituidos por la cantidad de espacios existentes en el diagrama.

Doble click

Cuando esta acción se ejecuta sobre cualquiera de los puntos del diagrama una nueva ventana de filtros se genera, lo importante de esta acción es que la información utilizada para llenar las cajas de filtros corresponde a los archivos que pasaron o fallaron en ese punto específico según se escoja. Como se puede observar en la figura 19 esta ventana tiene una diferencia básica con respecto a la principal y es que los menús se deshabilitan para mantener un orden y evitar confundir la información y sacar por ende conclusiones incorrectas. Solamente es posible utilizar la información cargada al crear la ventana para aplicar nuevos criterios de filtrado y así conseguir analizar más específicamente los datos.

EDSAT

Filters

Test Name shmoo_IO_RX_ODD_EYE_HEIGHT_WIDTH_2400

Port 1 5

Lane 3 7

Select all ports Select all lanes

Deselect all ports Deselect all lanes

Visual ID 2V82706380072 NF832006B0171

Select all units Deselect all units

Temperature 20 85

Lot Name IB109A19

Operation 7206 7207

Summary 1A

Site XXX

Tester tester3

ULT F8309600_071_+05_-01 F830962A_087_+04_+00

Selected VID's 0

Selected Shmoos 0

Reset Filters Shmoodiff Lite Shmoodiff Full

Figura 19. Ventana de filtrado generada con doble click.

Click derecho

Al presionar una vez el botón derecho sobre cualquiera de los puntos del diagrama se despliega un menú con las opciones de: acción al doble click, exportar a Microsoft© Excel®, salvar como imagen, generar reporte, desplegar caracteres, cambiar colores y enviar al análisis comparativo, como se puede observar en la figura 20.

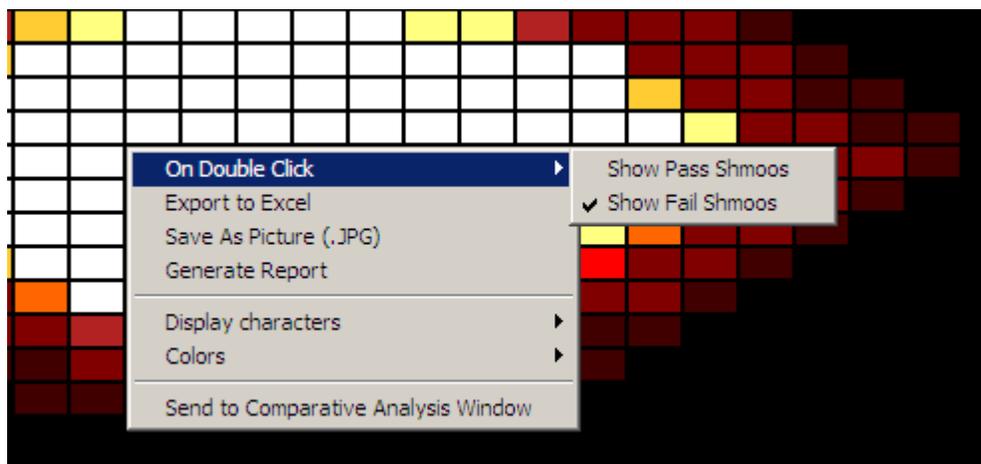


Figura 20. Menú de botón derecho.

La acción al doble click se refiere a la selección de los datos con los que se debe llenar la nueva ventana de filtros, ya sean los archivos que pasaron o que fallaron en el punto. Las opción exportar a Microsoft© Excel® se encarga de exportar la información del diagrama a formato de hoja de cálculo, para este propósito primero se despliega una ventana para pedir la ubicación y el nombre con el que se desea guardar el archivo, posteriormente se crea un objeto de hoja de cálculo en el EDSAT, se rellena con la información del diagrama en formato estándar (de letras) y se le agrega la tabla de simbología, las unidades que corresponden a ese diagrama y las estadísticas con los nombres de los archivos que cuentan con los máximos y mínimos altos y anchos, una vez que se tiene creada esta hoja en el EDSAT se

Aparte de estas opciones para exportar los datos del diagrama en diferentes formatos para su presentación posterior existen dos opciones de interacción directa con el diagrama, estas son las de mostrar información de errores y cambiar colores, como antes se explicó existen tres formas de presentar el diagrama, con letras, con números y sin información escrita en los puntos y además tres esquemas de colores diferentes (azul, rojo y gris); pues bien estas dos opciones se encargan de darle al usuario la posibilidad de cambiar estos parámetros una vez que se generó el diagrama para así no tener que generar nuevamente el diagrama para verlo en otro color o cambiar letras por números por poner un ejemplo.

La última de las opciones de este menú es la de enviar al análisis comparativo, esta opción envía la información del diagrama a la ventana del análisis comparativo que se explicó anteriormente para poder compararla con otro diagrama completamente independiente.

Por último fuera del menú del botón derecho sólo queda una funcionalidad particular que es activada por un botón ubicado en la barra inferior, este botón indica el punto de entrenamiento y lo hace resaltando la fila y la columna del punto de entrenamiento durante dos segundos y luego los regresa a su estado original; esta funcionalidad es muy importante pues permite al ingeniero determinar si el diagrama está corrido en la escala del tiempo o la escala del voltaje para así poder determinar si existen problemas a la hora de realizar la configuración del puerto.

Capítulo 5: Análisis de Resultados

En este capítulo se tratan los resultados obtenidos a partir del uso de la herramienta y comparando el proceso de análisis antes y después de la utilización de la herramienta y los resultados con base en los objetivos del proyecto.

Para comprobar la eficiencia del EDSAT se realizaron pruebas de tiempo utilizando el EDSAT y el programa de Perl existente anteriormente para una única combinación, con el fin de comparar el tiempo necesario para obtener el diagrama correspondiente aplicando filtros específicos utilizando una y otra herramienta. Con datos provenientes de un lote cuyas características se resumen en la tabla 1. Las pruebas fueron realizadas en una computadora con un procesador Intel® Core™ 2 Duo de 2 GHz, 2 GB de memoria RAM y sistema operativo Windows® XP, los resultados de tiempo de estas pruebas se resumen en la tabla 2.

Tabla 1. Descripción del lote analizado para pruebas de eficiencia del EDSAT.

Prueba	Cantidad de unidades	Condiciones de la prueba	Cantidad de archivos individuales procesados
Medición del alto y ancho del ojo para los semiciclos en bajo del receptor del puerto (RX)	16	Temperatura: 20° C, 85° C	4060

Tabla 2. Resultados de tiempo de las pruebas realizadas al EDSAT.

Prueba	Shmoodiff (s)	EDSAT (s)	EDSAT/Shmoodiff (%)
1	424.3	11.1	2.62
2	429.7	17.0	3.96
3	435.6	16.3	3.74
4	488.8	16.3	3.33
5	488.1	12.0	2.46
Promedio	453.3	14.54	3.21

Después de realizar las pruebas se logró determinar que el EDSAT toma en promedio aproximadamente 3.2% del tiempo que toma el Shmoodiff en cuanto al procesamiento y despliegue de los datos, a pesar de que esto ya es una mejora significativa, en esta prueba no se toma en cuenta el mayor logro en ahorro de tiempo que permite el EDSAT que se refiere a la capacidad de alcanzar datos específicos de forma rápida y sencilla.

Este proceso realizado con la herramienta anteriormente disponible, el Shmoodiff, tomaba varias horas pues para analizar los datos se ejecutaba un programa de Perl que tenía como salida una colección de archivos con los diagramas correspondientes a cada una de las combinaciones posibles de filtros, de forma que cuando se identificaba una posible causa de error debía buscarse uno a uno entre los archivos hasta encontrar el correcto y aún peor, si se cometía un error en los parámetros entregados al programa de Perl debía ejecutarse nuevamente todo el proceso que tardaba varias horas. Todo lo contrario al utilizar el EDSAT pues se ejecutan únicamente las combinaciones que se seleccionen generando sólo los diagramas de interés y disminuyendo el tiempo de ejecución a unos cuantos segundos.

Según datos de los análisis anteriores realizados por los ingenieros del departamento PDE de Costa Rica, la ejecución del programa de Perl tomaba alrededor de 8 horas como mínimo a esto se le debía sumar el tiempo de análisis que incluía la búsqueda de los diagramas específicos, alrededor de 8 horas también.

Por el contrario al utilizar el EDSAT todo este proceso es sustituido por uno más simple: cargar un directorio, aplicar filtros a este y generar el diagrama correspondiente, proceso que tarda en promedio 15 segundos en la carga y 15 minutos en el análisis para los datos citados anteriormente. Resultando esto en una mejora de alrededor de 65 veces en tiempo como mínimo pues se está asumiendo el mínimo en los procesos de ejecución de los programas de Perl. Este dato entrega información mucho más representativa en cuanto a la eficiencia del EDSAT con respecto al Shmoodiff pues se toma en cuenta el proceso completo de análisis y no únicamente el tiempo de procesamiento de datos, obteniendo así alrededor del doble de la eficiencia calculada sólo para el tiempo de procesamiento.

Por otra parte existen varias diferencias en el proceso de análisis utilizando una y otra herramienta. La primera diferencia se presenta en la disponibilidad de los datos al terminar la prueba en el equipo de pruebas, cuando se utiliza el EDSAT los archivos individuales son generados y están disponibles conforme se ejecuta la prueba mientras que el primer paso para los programas de Perl está en tomar el reporte del equipo de pruebas y quebrarlo en los archivos individuales.

Esta mejora significativa en el proceso de análisis se ve reflejada en una disminución de los tiempos de respuesta y corrección de los errores de diseño e identificación de fallas de los diseños iniciales de los productos que se desarrollan en la compañía. Así por ejemplo se puede tener un menor tiempo de respuesta entre la liberación de un diseño preliminar y la corrección de sus errores en una segunda versión. Disminuyendo también por consecuencia los tiempos en que los productos pueden ser liberados al mercado en general, todo esto logrado al tener una herramienta de análisis que permite apuntar más específicamente a las posibles causas de errores en un tiempo menor. En la figura 22 se puede observar el impacto que tiene el EDSAT con respecto al tiempo de respuesta en la identificación de fallas, que es la parte más lenta del proceso. Se mantiene sin unidades las especificaciones pues los tiempos de las etapas son información confidencial.

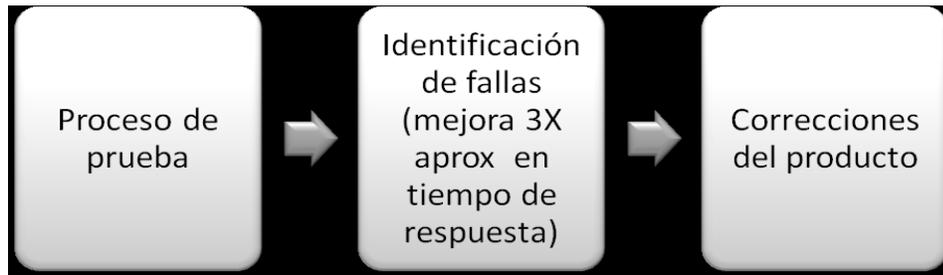


Figura 22. Esquema del proceso de prueba y corrección del producto.

Es importante comprender que la rapidez con que se puedan analizar los datos no necesariamente es la misma con la que se identifican las fallas. Ciertamente el proceso se acelera si los datos son analizados más rápidamente pero la identificación de fallas es siempre un proceso más lento pues conlleva también un análisis de los diseños, comparaciones con comportamientos esperados así como otros estudios más complejos.

La herramienta entonces tiene un gran impacto económico también para la empresa pues el tiempo de ingeniería que se consume en el análisis de datos se ve reducido de la misma forma en alrededor de 65 veces, lo que quiere decir que puede ser aprovechado en otros procesos ya sea de análisis (de las fallas por ejemplo) o desarrollo (de herramientas o soluciones a problemas).

Los objetivos fueron cumplidos a cabalidad, como se explicó anteriormente existen ya funciones dentro del entorno del sistema del equipo de pruebas que permiten tanto crear la estructura de directorios con los datos provenientes de las pruebas como las que permiten la simulación de las pruebas cuando no se está conectado con el equipo de pruebas.

Por otra parte existen funciones dentro del EDSAT que permiten analizar los diagramas de ojo aplicando los criterios de filtrado que el usuario decida establecer y funcionalidades que permiten navegar por los datos de una forma ágil y rápida. Existe además la opción de generar reportes que permiten tener información

condensada para su utilización posterior y por otro lado la opción de guardar los datos de una sesión de análisis para continuar con esta en otro momento.

En las figuras 23 y 24 se pueden apreciar diagramas que ejemplifican los dos procesos y se hace evidente por que la diferencia de tiempos de ejecución entre ambos, el proceso basado en Perl es repetitivo y muy tedioso mientras que el proceso utilizando el EDSAT es simple y de rápida ejecución.

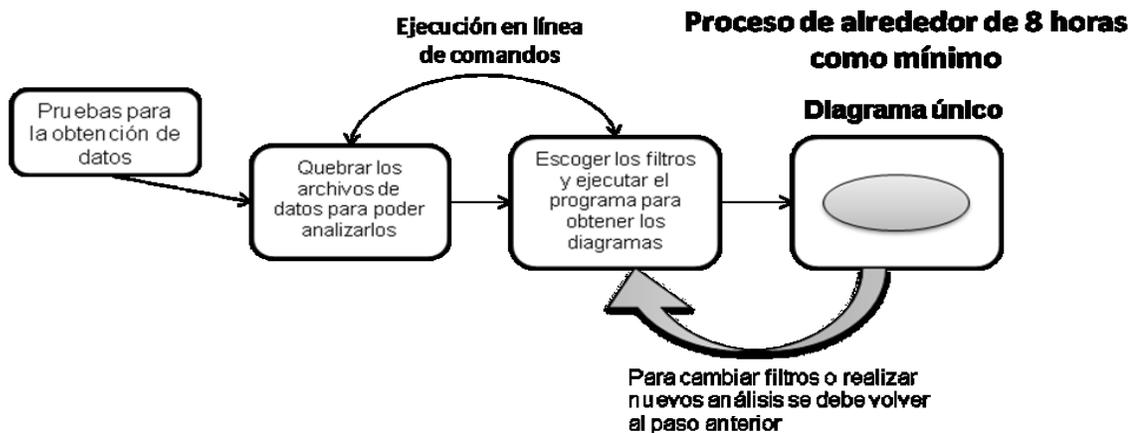


Figura 23. Proceso de análisis utilizando la metodología anterior con programas de Perl.

En resumen el EDSAT puede trabajar en línea y no en línea con el equipo de pruebas, cargar datos provenientes de archivos para simulaciones, aplicar criterios de filtrado y desplegar los diagramas de ojo de forma eficiente, potenciando así las capacidades de análisis para los ingenieros a cargo de los procesos de prueba y validación de productos para las plantas de ensamble y prueba de la corporación Intel, como la ubicada en Costa Rica.

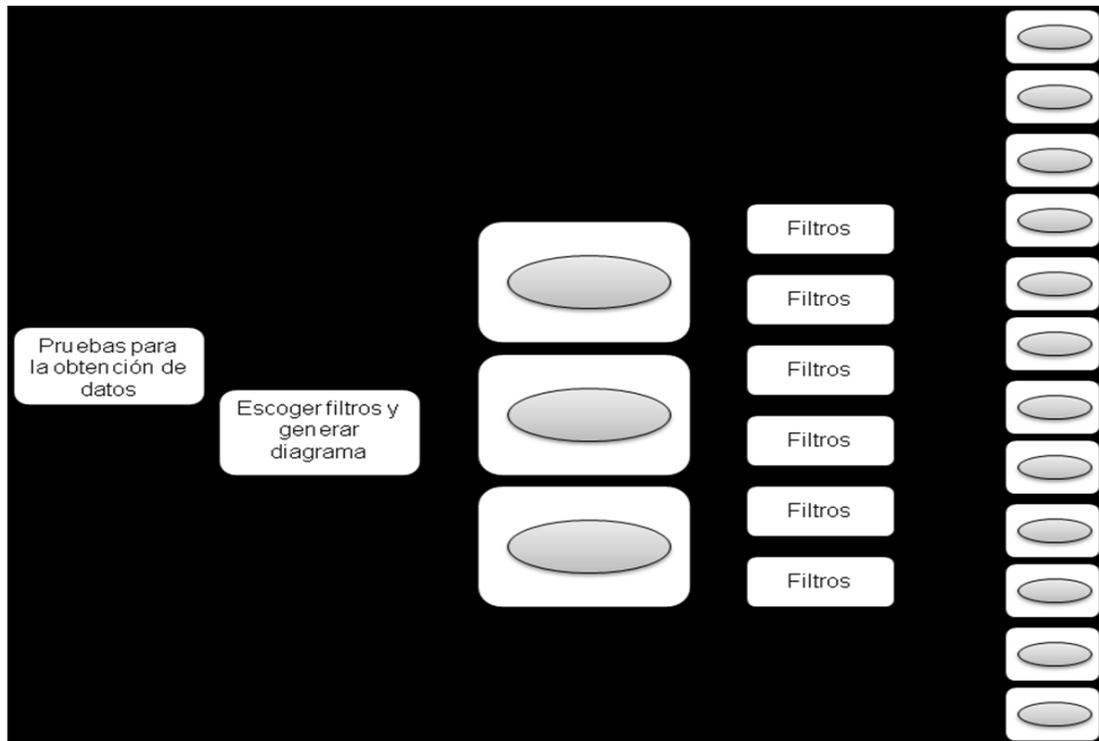


Figura 24. Proceso de análisis utilizando el EDSAT.

Cabe resaltar que el EDSAT está desarrollado de forma que con algunas modificaciones menores es posible aplicar la plataforma a diagramas del área de potencia, memoria caché y cualquier otra área que utilice diagramas basados en texto.

Capítulo 6: Conclusiones y recomendaciones

6.1. Conclusiones

- Mediante las funciones de usuario desarrolladas para el programa del equipo de pruebas es posible imprimir la estructura de directorios que permite el funcionamiento de la herramienta.
- Las funciones de usuario desarrolladas permiten cargar la estructura de memoria del equipo de pruebas con datos previamente almacenados para realizar simulaciones.
- La interfaz de filtros y despliegue de diagramas implementados permiten escoger los criterios que serán aplicados a los datos y presentar únicamente la información que concuerde con estos.
- Utilizando la opción de guardar el entorno de análisis, es posible recrear posteriormente el análisis para continuarlo.
- Los métodos para exportar los diagramas permiten guardar la información en tres formatos diferentes: imagen, texto y hoja de cálculo de Microsoft® Excel®.
- Mediante la opción de hacer un análisis utilizando la herramienta Shmoodiff es posible ejecutar el programa original en Perl y generar un archivo de hipertexto a modo de reporte.
- Con el uso de esta herramienta se logró una mejora del tiempo de análisis de alrededor de 65 veces con respecto al proceso que existía anteriormente, traduciéndose en un ahorro de dinero para la compañía.

6.2. Recomendaciones

- El EDSAT puede utilizarse para analizar otros diagramas que no sean los de ojo sino producto de otras pruebas, tan solo cambiando el formato del nombre de los archivos que contengan los diagramas de estas pruebas. Las pruebas hechas por ejemplo a la memoria caché y en la parte de potencia generan diagramas que deben ser analizados también pero que no son diagramas de ojo, sin embargo si se guardan en archivos individuales y con el formato de nombre que se definió para los diagramas de ojo, es posible realizar los análisis de estos datos utilizando el EDSAT. Para conseguirlo sería necesario hacer únicamente cambios pequeños en las pantallas de presentación pero en cuanto a la manipulación de datos no se necesitarían cambios.

Bibliografía y Referencias

- [1] **Dinamarca Ossa, José Antonio:** Análisis de Diagramas de ojo. Universidad Técnica Federico Santa María, Chile; 2002.
- [2] Diagramas de Ojos, Wikipedia. [en línea], disponible en: http://es.wikipedia.org/wiki/Diagrama_de_ojos. [Consulta: 10 de junio de 2009].
- [3] Imagen Intel Quick Path Interconnect. [en línea], disponible en: <http://www.hardwaresecrets.com/fullimage.php?image=12243>. [Consulta: 13 de junio de 2009].
- [4] **Miró, Verónica M.:** Diagrama de Ojo. [en línea], disponible en: <http://www.eie.fceia.unr.edu.ar/~comunica/cx/DIAGRAMA%20DE%20OJO.ppt>. [Consulta: 10 de junio de 2009].
- [5] Quintana, Ricardo: IBIST for dummies, Intel Corporation internal presentation, Costa Rica; 2008.