17
29

UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES ARAGON

DESCRIPCION Y ESTUDIO DE LOS SISTEMAS FAULT TOLERANT EN DIVERSAS MARCAS Y TIPOS DE COMPUTADORAS

T E S I S

QUE PARA OBTENER EL TITULO DE

INGENIERO EN COMPUTACION

PRESENTAN:

ANTONIO GONZALEZ ALVAREZ
JOSE ALFREDO HERNANDEZ ZAVALETA

DIRECTOR DE TESIS: ING. RICARDO GUTIERREZ OROZCO.

DICIEMBRE DE 1994.

UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

AGRADEZCO A LA UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO Y A LA ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES ARAGON, EL HABERME BRINDADO LA OPORTUNIDAD DE REALIZAR MIS ESTUDIOS A NIVEL LICENCIATURA.

A MIS MAESTROS, QUIENES ME TRANSMITIERON SUS CONOCIMIENTOS EN FORMA DESINTERESADA Y GENEROSA PARA FORMARME COMO PROFESIONISTA.

A MIS COMPAÑEROS, QUIENES CON SU COMPRENSION Y AYUDA, ME APOYARON, MOTIVANDOME A CONCLUIR MIS ESTUDIOS.

INDICE GENERAL

INDICE GENERAL

INDICE DE FIGURAS

INDICE DE FIGURAS

INTRODUCCION

INTRODUCCION

Los sistemas de información son indispensables en el manejo de base de datos, transacciones bancarias, correo electrónico, comunicación vía módem, interconexión de equipos de diferentes proveedores por medio de redes locales y otras formas de comunicación en cualquier disciplina (medicina, ingeniería, militar, gobierno, industria, etc.) que hacen uso de estos recursos.

Todos estos recursos manejan constantemente la información de acuerdo a instrucciones que son dadas por el sistema operativo para obtener los resultados que se ordenaron.

A veces, a través de la experiencia se ha encontrado que algunos equipos o recursos son muy confiables, no manifestando ningún tipo de falla. Estos equipos cuentan generalmente con UCP, Memoria, Unidades de Almacenamiento, pero, ¿qué pasa cuando algunos de estos elementos llega a operar incorrectamente?.

De aquí que surjan otras series de cuestiones: ¿Cuánto tiempo se llevará en corregir al elemento en problemas?, ¿El sistema será establecido inmediatamente sin ninguna pérdida de información si el error se ha presentado en cualquiera de las unidades de almacenamiento?, etc.

De acuerdo a las necesidades que tenga el usuario en sus aplicaciones y de tratar de mantenerlas en ejecución ante la presencia de alguna falla, el sistema requerirá de un cierto nivel de tolerancia a fallas, hasta lograr que no se presente interrupción alguna en el sistema ante la presencia de una falla en alguno de sus elementos.

En este trabajo se han desarrollado los conceptos teóricos de los Sistemas Tolerante a Fallas que pueden ser implantados en cualquier sistema que esté orientado al control de procesos críticos, aplicaciones de alta integridad y disponibilidad de la información .

En el capítulo I, se comienzá describiendo los **Antecedentes de la Evolución de los Sistemas Tolerantes a Fallas,** teniendo en cuenta las técnicas que se han desarrollado ante los cambios tecnológicos y de procesamiento en los Sistemas Tolerantes a Fallas. En este capítulo se manejan conceptos propios del tema, que se definen en los capítulos II y III, ejemplificando cada término para su mejor comprensión.

Al término del capítulo II **(Conceptos Generales)** y del capítulo III **(Conceptos básicos),** se tendrá noción de la importancia del entendimiento de estos conceptos para los temas posteriores.

Todo sistema está diseñado en base a alguna metodología, que en su conjunto se obtendrán los resultados para lo que fue diseñado.

En el Capítulo III, trata sobre las **Técnicas Para El Manejo de Errores,** se describen las estrategias que se implementan en los Sistemas Tolerantes a Fallas para la detección y la recuperación de errores a nivel de código y de módulos. Estas estrategias en su conjunto proporcionan elementos necesarios para mantener la disponibilidad, integridad, confiabilidad del sistema.

Al término de estos tres capítulos se tendrá un mayor conocimiento sobre la teoría básica de los Sistemas Tolerantes a Fallas en cuestión y de sus estrategias para el manejo de errores que ocurran en el sistema.

En los capítulos IV, V y VI, se tratan los temas de **Uniprocesador, Multiprocesador** y **Multicomputadoras**, correspondientemente, en los cuales se hace una descripción del funcionamiento de cada una de sus arquitecturas y las estrategias que son utilizadas para las interrupciones de algunas fallas que se pueden manifestar en cualquiera de sus elementos y mantener la disponibilidad, integridad y confiabilidad en los sistemas de tiempo real.

En el capítulo VII **(Otras Configuraciones Para la Seguridad de la Información)** presentamos algunas de las configuraciones que se han logrado implementar para aquéllos casos de desastre total en una área donde se tiene concentrada la información y tenerla respaldada en una área remota.

Finalmente en el capítulo VIII de **Aplicaciones,** presentamos las diversas áreas donde los Sistemas Tolerante a Fallas pueden ser implementadas de acuerdo a la integridad, disponibilidad y seguridad de las aplicaciones que se estén utilizando.

CAPITULO I

ANTECEDENTES

I. ANTECEDENTES

I.1. GENERACIONES DE LOS SISTEMAS TOLERANTES A FALLAS.

En este capítulo se dividirán a la historia de los sistemas tolerantes a fallas en cuatro generaciones, cada generación se dividió de acuerdo a la tecnología que en ese momento se encontraba a la vanguardia, pasando por los relevadores, bulbo de vacío, transistores y circuitos integrados.

I.2. PRIMERA GENERACION.

Durante los años 40's y principios de los 50's la seguridad en las computadoras fue de gran interés para los primeros inventores y usuarios de la primera generación de computadoras. Los relevadores electromecánicos se utilizaron como dispositivos de conmutación en los años 40's y las válvulas de vacío en los años 50's, los tubos de rayos catódicos, los dispositivos de almacenamiento y algunos otros dispositivos, todos con relativa razón de defectos en los componentes y errores en el diseño; fue necesario el mantenimiento preventivo período por el desgaste mostrado en los tubos de vacío y algunos otros dispositivos.

La necesidad de contar con sistemas operativos que permitieran estar funcionando después de presentarse una falla y saber las causas que llevaban a los sistemas a mantenerse fuera de operación, fueron

desarrollados esquemas de detección a fallas y algunos métodos de recuperación automática, estos esquemas fueron aplicados a generaciones posteriores.

Se tiene poca información de la fecha exacta de la primera computadora digital construída con características a Tolerancia a Fallas se considera que empezó a construírse en Checoslovaquia en 1950 por el Profesor Antonin Suovoda. A esta máquina se le conoció como el SAPO, construída con tambores magnéticos y memoria de tubo de vacío utilizando una representación numérica de punto flotante de 32 bits, con una arquitectura de tres módulos idénticos de UCP y dos módulos de memoria, a esta configuración se le conoció como Triple Modular Redundante (TMR).

La paridad se utilizó para operaciones de lectura y escritura. Solo la instrucción "Restaurar" se utilizó en casos en que la corrección por la operación del "seleccionador" no se logrará, deshabilitando automáticamente la máquina para preservar su estado, para posteriormente llevarla a un estado de recurrencia de la falla.

Todos los errores en la máquina, automáticamente se visualizaban en la consola de los operadores; el SAPO inició su operación en forma regular en 1956 y posteriormente surgen sistemas semejantes tales como el EPUS Y SAGE.

En esta generación se puede afirmar que la detección de fallas fue una característica común en el hardware, teniéndose que mejorar los Sistemas Tolerantes a Fallas en la recuperación automática y la detección, para la próxima generación en los sistemas digitales.

I.3. SEGUNDA GENERACION.

El desarrollo ordenado de los Sistemas Tolerantes a Fallas fue interrumpido por la aparición de los circuitos semiconductores y memorias de ferrita, como componentes de los sistemas digitales. Al incrementarse la fabricación de los nuevos componentes, fue tan grande la demanda que los requerimientos para los sistemas de seguridad que pronto se encontraron sin técnicas para aprovechar los avances de los semiconductores. Respecto a la detección de errores, se tuvo una permanente relación con los dispositivos periféricos menos seguros y los dispositivos de gran almacenamiento.

El bit de paridad fue solo la única técnica de chequeo que se utilizó en la memoria y UCP, aún la paridad no fue tan universalmente utilizada en las máquinas de la segunda generación. El concepto de la construcción de los Sistemas Tolerantes a Fallas desapareció del vocabulario de los diseñadores y los efectos se hicieron ver en muchas computadoras tales como micros y minis (ILLIAC IV y en la CRAY-1).

Este período se caracteriza por contar con la construcción de hardware como asistencia en la detección; el desarrollo de la teoría y programas de diagnóstico como herramienta principal para poder eliminar fallas. Cuando la ventaja de microprogramación se reconoció y el control microprogramado empezó a ser común, los programas de diagnóstico se reemplazaron por los microdiagnósticos en los sistemas de propósito general a fines de los 60's.

Las limitantes de detección a fallas dadas por el hardware, la responsabilidad de mantener una operación segura, fue relegada por "default" al software, que implementó gran parte de la detección y recuperación. Un ejemplo práctico de la aplicación de software con característica de detección y recuperación son las máquinas IBM 9020 de multiprocesamiento y los sistemas MULTICS.

La tendencia en los sistemas de gran escala fue de mantener una computadora dedicada exclusivamente para el diagnóstico y enlace a la computadora central, donde se ejecutarían comandos de recuperación y pruebas para el control de la máquina huésped.

Dos ejemplos de esta configuración; fueron la CDC-STAR 100 que utilizó un diseño de una unidad de control de mantenimiento y la minicomputadora DATA GENERAL NOVA 1200 empleando la computadora AMDAHL 470 V/16 ambas con un procesador de consola para la operación normal y un procesador de mantenimiento para condiciones de fallas donde la máquina principal tiene una lógica dedicada para facilitar estas cuestiones. Una

importante aportación a este amplio concepto en AMDAHL 470 V/16 es el diagnóstico central remoto que puede accesar a la máquina huésped por medio del procesador de la consola local.

El uso de sistemas con característica de detección y recuperación se extendió a áreas de actividad humana en la cual una falla en el sistema colocaba en riesgo la vida humana o que podría causar daños irreversibles, por ejemplo los sistemas de control para la vigilancia de un reactor nuclear o el manejo de aeronaves, también estan sistemas que son usados en otras áreas tales como los sistemas telefónicos de conmutación electromecánica que posteriormente serían de conmutación electrónica.

Los sistemas de conmutación electrónica (SCE) desarrollados por los laboratorios de telefonía Bell se acentuó al paso de la evolución de la telefonía por conmutación electromecánica a la tecnología electrónica, centralizando el control programado de acciones de conmutación. La duplicidad fue usada para todos las partes críticas del sistema, con algún hardware dedicado y software que se utilizó para detectar, localizar e identificar los elementos dañados para posteriormente reemplazarlos manualmente.

El desarrollo de los SCE son los más numerosos y más ampliamente usados en los sistemas digitales en el mundo de hoy. Su mayor alcance de estos sistemas fue de tratar que el sistema no estuviera fuera de servicio por más de dos horas durante 40 años en un ambiente que permitiera el reemplazo

manual de los elementos defectuosos sin la interrupción del sistema. La experiencia empezó a acumularse en cuestión de seguridad surgiendo relativa utilidad de las técnicas de tolerancia a fallas que son utilizadas en varios modelos de SCE.

I.4. TERCERA GENERACION.

Los sistemas OAO (Observatorio Astronómico en Orbita) fueron desarrollados de 1961 a 1965. Su objetivo era diseñar sistemas de satélites espaciales que mantuvieran la seguridad del 95% por año durante el tiempo de órbita.

Se empezaron a desarrollar satélites espaciales que deberían ser lanzados, tales como el SATURNO IV Y SATURNO V por la tripulación del APOLO y SKYLAB, teniendo como característica en su arquitectura, componentes discretos utilizando enmascaramiento redundante a nivel del procesador. Los datos se almacenan en dos copias y los comandos del sistema en cuatro copias, todas estas copias se almacenaban en la memoria principal. Se utiliza la triplicación con un seleccionador para la comparación de los datos generados por los procesadores.

La computadora que se utilizó para apoyar la conducción del SATURNO, tenía una seguridad propuesta del 99% por 250 horas; para su construcción se utilizó una redundancia triple modular para la protección de una serie de

UCP's que constaba de siete módulos. Además el acceso en paralelo a la memoria es en forma duplicada, ocupándose un circuito para verificar la paridad tanto en el acceso a la memoria y en el manejo interno del sistema para poder determinar la posible existencia un error en cualquiera de las memorias que estaban duplicadas. Sí alguna de ellas presentaba alguna diferencia en el bit de paridad, automáticamente se realizaba una recuperación.

Otra característica de Esta generación fueron los esquemas de duplicación y detección; los esquemas de duplicidad se usaron en subsistemas tales como las fuentes de energía; la detección se utilizó en la centralización del control para la recuperación. Se implementó el enmascaramiento de la falla para ocultar las averías en sus componentes y permitiendo su funcionamiento, pero no se realizaba la recuperación del componente dañado.

Durante el transcurso de esta generación surgieron sistemas con mayores características, debido al desarrollo de circuitos integrados, llevando a los satélites espaciales que se mantuvieron a su tiempo de vida por 10 años a la obsolescencia de sus componentes.

La computadora que guió al SATURNO, se desarrolló en el período de 1962 a 1969, su seguridad era alcanzar el 99% por 250 horas de misión y la redundancia triple modular se utilizó para la protección serial de los UCP's, consistiendo de siete módulos.

Las aplicaciones adicionales que requieren mayor tiempo en el sistema sin ser interrumpidos por cualquier falla en el sistema, mejoraron con el surgimiento a mediados de los años 60's, el desarrollo de circuitos integrados, que marca la obsolescencia en algunos componentes redundantes. La computadora que conducía a el SATURNO, llevó de 1 a 10 años el control de las funciones del satélite, que fue el intervalo de tiempo de vida del satélite.

En este período surgen nuevas consideraciones teóricas, tales como: los períodos de una reparación segura, la detección de la falla y la conmutación dentro del sistema, siendo esto una muy clara ventaja de estas técnicas en comparación a la del enmascaramiento de la falla, como se empleaban en los sistemas OAO.

También se mejoraron las técnicas de redundancia, empleando una técnica de redundancia híbrida ("TMR"), en la detección se utilizó el código aritmético de detección de error, empleados en la duplicación y comparación, todas las consideraciones y la anterior característica descritas se concentraron en la computadora JPL STAR, que fue ampliamente probada bajo condiciones de fallas físicas, sirviendo después de varios años, de laboratorio para los sistemas tolerantes a fallas en 1976.

Después se enfocó la investigación a los sistemas distribuidos tolerantes a fallas, la JPL STAR fue utilizada como un procesador de entrada/salida

para los primeros protocolos de los sistemas distribuidos tolerantes a fallas (UDS).

La construcción de esta computadora fue la unión de dos arquitecturas, la JPL STAR y la TARP; tomando de la primera configuración el nivel de integración de los circuitos y la combinación de todo el procesamiento en un solo UCP reemplazable y de la segunda configuración la característica de prueba y reparación del procesador (test and repair processor), que continuamente monitoreaba la parte de la máquina señalando operaciones erróneas, el iniciar programas de restauración y ejecutar la acción de reemplazo interrumpiendo la energía cuando la falla se detectaba en cualquier módulo del sistema, fue significativamente mejorada y además fue diseñado el UCC (Unidad de Control de Configuración). Un segundo mejoramiento en el diseño fue la duplicación del UCP para mejorar la detección de errores.

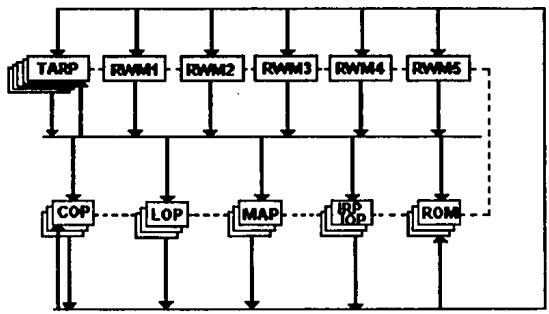

FIGURA I-1. ORGANIZACION DE LA COMPUTADORA STAR.

COP (PROCESADOR DE CONTROL)

LOP (PROCESADOR LOGICO)

MAP (PROCESADOR ARITMETICO PRINCIPAL)

ROM (MEMORIA DE UNICAMENTE LECTURA)

RWM (MEMORIA DE LECTURA Y ESCRITURA)

IOP (PROCESADOR DE ENTRADA Y SALIDA

IRP (PROCESADOR DE INTERRUPCIONES)

TARP (PROCESADOR DE VERIFICACION Y REPARACION)

La complejidad de la unidad UCC fue reducida para utilizarlas como unidades para monitorear el par de UCP's y de la recuperación del sistema.

En esta generación y en las anteriores solo se contribuiyó en la corrección y detección de errores en el código de la memoria y para el "bus" de transferencia.

Los esfuerzos de IBM realizados en la investigación de los sistemas tolerante a fallas no resultó un sistema completo, pero influyó para generar conceptos fundamentales en la teoría de diagnósticos, modelos de seguridad y teoría de la aplicación y chequeo seguro de los circuitos.

Dos de las mayores investigaciones y diseño empezaron a emplearse en sistemas de multiprocesamiento: Ambos sistemas se intenta que sean útiles en computadoras de control de aeropuertos.

I.5. CUARTA GENERACION.

Esta generación se caracteriza por el desarrollo de los circuitos integrados de larga y muy larga escala de integración (LSI y VLSI), reduciendo el tamaño y potencia además de aumentar su eficiencia en nuevas funciones.

En la actualidad se manejan 2 implementaciones para el manejo de fallas, llamados intensivo hardware-tolerante e intensivo software tolerante.

I.5.1. INTENSIVO HARDWARE TOLERANTE

En este tipo de arquitectura tenemos de 2 o hasta 3 UCP's en los cuales se ejecutará la misma instrucción, el mismo dato y en el mismo tiempo, esto es, que en los 3 UCP's se estará corriendo el mismo proceso, tratando que los relojes independientes de cada UCP se encuentren sincronizados, si en una unidad se adelanta una instrucción por medio del seleccionador se detendrá el UCP adelantado para que los otros 2 lo alcancen y tengan la misma instrucción, estos 3 UCP's son considerados como un solo UCP lógico ya que en los tres se está ejecutando el mismo proceso.

El propósito de este tipo de implementación es de que si alguno de estos UCP's fallara, el seleccionador detectará y aislará el componente o módulo responsable, pero al mismo tiempo se continuará con la ejecución del proceso sin ninguna interrupción en el sistema.

I.5.2. INTENSIVO SOFTWARE TOLERANTE

Esta implantación esta compuesta de 2 UCP's, uno llamado **PRIMARIO** y otro llamado **RESPALDO**, en el primario se corren los procesos y en el RESPALDO se hacen copias de los procesos que son activos y de la memoria así como de las transacciones que se hacen en las bases de datos, pero además, el UCP RESPALDO verificará si el UCP primario está corriendo sus procesos correctamente.

Si el RESPALDO le pregunta al primario su estado y no responde entonces el RESPALDO tiene que asumir que el primario se encuentra en funcionamiento incorrecto, esto es, que ocurrió un funcionamiento incorrecto y tomará el control de los procesos que estaban siendo procesados en el primario regresándose a la lectura de los datos anteriores a la verificación, cargarlos en memoria y seguir corriendo los procesos en el punto en donde falló el primario.

En este tipo de implementación, la recuperación de la falla es más tardado que en el anterior, además de que la recuperación en su mayor parte es por software, el RESPALDO regresa a los datos de la última verificación para ser procesados desde ese punto.

En estas implementaciones para el manejo de fallas, se pueden encontrar en los sistemas INTEGRITY y en el CICLON las dos pertenecientes a la

compañía de TANDEM, la primera usa intensivo hardware tolerante y en la segunda intensivo software tolerante.

Además en estas computadoras se utiliza inteligencia artificial, sistemas expertos para el control de la tolerancia a fallas y cuentan con sistemas abiertos para la comunicación con otras computadoras, manejando la mayoría de los protocolos de comunicación existentes, esto, con el fin de contar con los sistemas de alta seguridad llamado desastre total, en la que se pretende tener respaldada la información en un lugar distante del que se encuentra el sistema de cómputo, para que en el momento de que ocurra algún desastre (una guerra, un incendio, etc.), todas sus bases de datos y hasta los procesos se encuentren respaldados en otra computadora remota.

CAPITULO II

CONCEPTOS GENERALES

II. CONCEPTOS GENERALES.

II.1. SISTEMAS TOLERANTES A FALLAS.

A partir de este capítulo se usarán las siglas **STF** para referirse a un Sistema Tolerante a Fallas y **UCP** para referirse a la Unidad Central de Procesamiento.

Definiremos como **componente** a la unidad o módulo que forma parte de una computadora, por ejemplo en hardware tenemos las unidades de discos, controladores de discos, unidad de memoria principal, fuente de poder, unidad central de procesamiento, etc., y en software tenemos los módulos de programas, y paquetes. Se define como **elemento** a la unidad que en conjunto forma a los componentes, es decir, en hardware tenemos, los circuitos integrados, memoria, capacitores resistencias, etc., y en software tenemos, los datos, instrucciones, entradas o salidas lógicas, etc.

Día con día son requeridos sistemas con una alta seguridad en la ejecución de los procesos y en el almacenamiento de la información, por ejemplo tenemos a los sistemas bancarios que usan procesamiento de transacciones en línea ("OLTP" On-Line Transaction Processing), para poder realizar sus movimientos críticos, ya sea en una sala de captura o en cajeros automáticos para mantener así sus bases de datos confiables.

Otro ejemplo práctico de un STF lo tenemos en los satélites artificiales, en éstos se manejan técnicas de redundancia (duplicidad de componentes), tanto en el hardware como en el software, en el momento de que alguno de estos componentes sufra una avería, habrá una conmutación a su componente duplicado que se encuentra, ya sea trabajando en paralelo o actualizando la última información de los procesos y datos, y por lo tanto continúa funcionando.

Sin tolerancia a fallas en los satélites artificiales el hombre tendría que viajar al espacio cada vez que se averiaran, lo cual no sería costeable.

Con base en lo anterior definiremos a un **STF** como aquél que por medio de la implantación de varias técnicas de Hardware y Software, tendrá como objetivo fundamental la integridad de la información y la continuidad de seguir operando al momento de que se genere una falla en alguna de sus partes ya sea de hardware o Software, en el cual se recupera en un lapso de tiempo muy corto sin la necesidad de la intervención inmediata del hombre.

II.2. FUNCIONAMIENTO INCORRECTO, ERRORES Y FALLAS.

II.2.1. FUNCIONAMIENTO INCORRECTO.

Se conoce en los STF como **funcionamiento incorrecto**, cuando un componente de hardware o software, no cumpla con las funciones para las

que fueron diseñados; por ejemplo, cuando un controlador de discos flexibles no graba ni tampoco lee datos, o en el caso de la UCP, teniendo un tiempo procesando, éste envíe salidas erróneas, o que no se puede tener acceso a la unidad de memoria principal para su lectura, etc.

FIGURA II-1. RELACION ENTRE FALLAS, ERRORES Y FUNCIONAMIENTO INCORRECTO.

II.2.2. ERROR :

Es la causa de que se genere un funcionamiento incorrecto y se le conoce como el valor lógico incorrecto de un elemento, esto es, que una compuerta produzca que la salida siempre sea 0 lógico en vez de que la salida fuera 1 lógico.

En el software, por ejemplo, tenemos que en el lenguaje "C" el valor que regresa la función de apertura de archivos, es "0" si el archivo fue abierto y diferente de "0" si no fue abierto, si ocurriera el error, de que el valor retornado fuera "-1" por alguna causa no se podrá abrir el archivo .

Un **error latente** es aquél que está presente en alguna parte del sistema, pero que no producirá un funcionamiento incorrecto hasta que sea activado, por ejemplo, un programa que realice la lectura de determinado archivo, provoque una caída del sistema, mientras no sea leído el archivo, el error se considerará latente hasta que el archivo tenga un acceso de lectura.

II.2.3. FALLA.

Una falla provocará un error y se conoce como falla a una anomalía en el estado físico (hardware) o inicial (software) de algún elemento, causado por factores como: frío, calor, humedad, deterioro, problemas de manufactura, en la especificación del sistema, mal uso, etc.

Una falla ocurrirá, por causas de una temperatura mayor a la especificada en una compuerta lógica, provocando que cambien sus características eléctricas y en consecuencia se produzcan salidas erróneas, la falla es, entonces, el cambio de características eléctricas por causa de la temperatura.

Otro ejemplo es el de un corto circuito en una compuerta, ocasionado por desgaste de la misma, o incorrecta polarización.

Se analizarán 2 ejemplos, uno de hardware y otro de software para relacionar y comprender mejor los conceptos mencionados.

EJEMPLO #1.

Se tiene una unidad de memoria principal con una capacidad de 512 kbyte, con una longitud de palabra de 8 bits, con un direccionamiento de 0 a 511 (RAM de 1X512 bytes) y se encuentra funcionando correctamente. En el instante en que se escribe en la memoria, en la dirección 0111 1111 (256 en decimal), se produce una perturbación electromagnética ocasionada por un rayo y entonces la dirección cambia en el bit mas significativo, esto es a 1111 1111 (512 en binario) y sucede que el byte se graba en la dirección 512, después se vuelve a leer el byte en la dirección en que se tenía que grabar correctamente o sea la 256, leyendo otro valor del dato contenido en el byte.

Del ejemplo anterior podemos deducir lo siguiente:

Interpretación de la falla.- la falla fue la elevación del voltaje en el 8o. bit por causa de la perturbación electromagnética.

Interpretación del error.- el error fue el valor de la dirección en que se grabó en la memoria.

Interpretación del funcionamiento incorrecto.- el funcionamiento incorrecto fue que la memoria al momento de leer la dirección 256, realiza la lectura de otra información y por lo tanto no se obtiene el byte correcto.

Ejemplo #2.

Un programador se equivoca al diseñar un programa que maneja menús, y provocará que el sistema se caiga cuando se escoja el menú encargado de imprimir reportes.

Interpretación de la falla.- la falla es cuando el programador diseñe erróneamente el programa.

Interpretación del error.- el error se producirá en el instante que se seleccione el menú de imprimir los reportes y es un error latente mientras no se escoja el menú.

Interpretación del funcionamiento incorrecto.- el funcionamiento incorrecto es cuando el sistema se caiga.

II.3. PRINCIPALES FACTORES QUE ORIGINAN LAS FALLAS.

Existen diversos factores que originan las fallas en los sistemas digitales que afectan tanto el Software como en el Hardware.

Se dividirán los factores que originan las fallas de la siguiente forma:

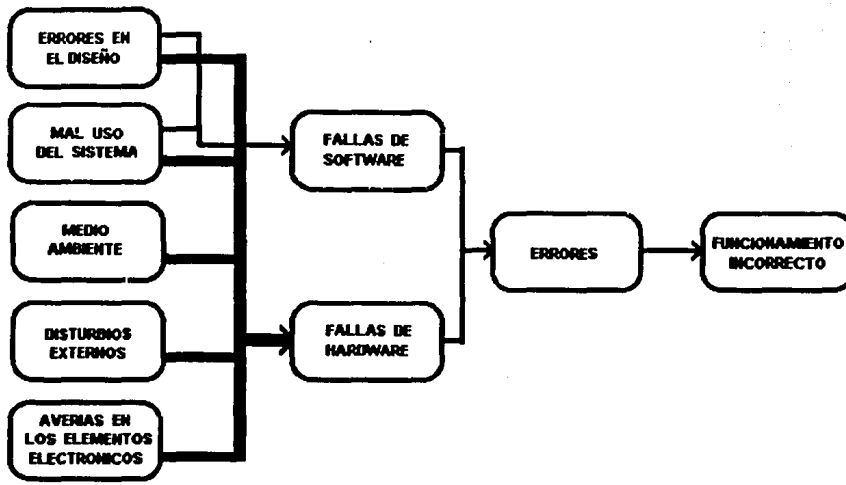

FIGURA II -2. RELACION DE CAUSA Y EFECTO DE FALLAS, ERRORES Y FUNCIONAMIENTOS INCORRECTOS EN UN SISTEMA.

II.3.1. Errores en el diseño.- se consideran dos aspectos, primero en la especificación del sistema y después en la implantación.

II.3.2. Averías en los elementos electrónicos.- provocados por la mala calidad de los elementos electrónicos o simplemente por el desgaste natural.

II.3.3. Medio ambiente.- Son condiciones externas que pueden alterar el funcionamiento correcto del sistema, tales como condiciones extremas de calor, frío, humedad o una atmósfera ionizada.

II.3.4. Disturbios externos.- Son todas las condiciones que no estan controladas por el sistema, que son necesario contar con accesorios para controlar la generación de estos factores, por ejemplo. el Medio Ambiente, Cargas Estáticas, interferencias electromagnéticas, radiaciones, etc., pueden ser controladas por aire acondicionado, eliminación de ruido de motores o aparatos receptores que causen disturbios externos al sistema.

II.3.5. Mal uso del sistema.- ocurre cuando se crean programas de aplicación que por algún error se afecta al sistema o por el manejo incorrecto del sistema.

II.4. CLASIFICACION DE LAS FALLAS.

En el siguiente cuadro sinóptico se puede apreciar la clasificación de las fallas:

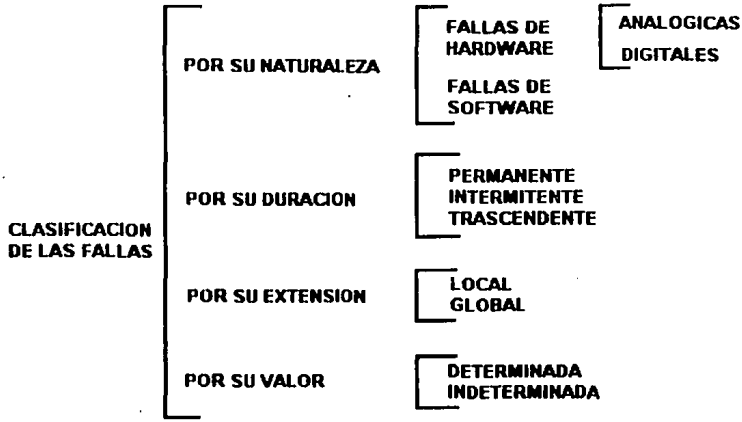

FIGURA II-3. CLASIFICACION DE LAS FALLAS.

II.4.1. POR SU NATURALEZA:

II.4.1.1 Fallas por hardware: Son las ocasionadas por la mala calidad o averías en los elementos electrónicos, por ejemplo, que un circuito integrado se dañe por desgaste o que un capacitor pierda sus propiedades dieléctricas y se convierta en resistencia o conductor y se produzca un corto circuito.

Las fallas de Hardware se subdividen en:

II.4.1.2. Fallas analógicas: Son fallas de circuitos de electrónica analógica, tales como fuentes de energía o convertidores Analógico/Digital y Digital/Analógico, circuitos de interface para sensores y actuadores, etc.

II.4.1.3. Fallas digitales: Estas pueden ocurrir en memorias, microprocesadores, compuertas lógicas, etc.

II.4.1.4. Fallas de Software: son ocasionadas por errores en el diseño de módulos de programas en el sistema operativo, compiladores, traductores y de aplicación.

Supongamos que un sistema operativo no tiene protección en áreas reservadas a la memoria principal, los errores en los programas de aplicación pueden ser ocasionados por ocupar las direcciones que son reservadas a la memoria ocasionando una caída o dañar archivos del sistema operativo o Bases de Datos.

II.4.2. POR SU DURACION:

Este tipo de falla se mide por el tiempo en que el sistema se encuentra en funcionamiento incorrecto y por la recuperación de las operaciones del sistema.

Se dividen en 3 tipos:

II.4.2.1. Falla Permanente: es cuando permanece indefinidamente mientras la acción correctiva no es efectuada.

II.4.2.2. Falla Transitoria: este tipo de falla permanece un período de tiempo, muy pequeño.

II.4.2.3. Falla Intermitente: una falla intermitente aparece y desaparece en una forma aleatoria, en ciertos tiempos o bajo ciertas condiciones, estas fallas son las más difíciles de corregir.

II.4.3. POR SU EXTENSION:

II.4.3.1. Falla Local: si únicamente se afecta un módulo de alguna parte del hardware (por ejemplo los módulos de la memoria principal) o alguna rutina de software.

II.4.3.2. Falla Global: se afecta a la mayoría del hardware, software o ambos. Por ejemplo, si ocurre un error en el contador del programa del microprocesador y la dirección de una localidad de memoria que contenga alguna instrucción no reconocida por el microprocesador, entonces se perderá el control del programa ocasionando una caída del sistema, esta falla producirá que se afecte de manera global al hardware y al software.

II.4.4. POR SU VALOR:

II.4.4.1. Falla Determinada: si el Estado de la falla permanece sin cambiar por todo el tiempo que dura la falla, por ejemplo una falla que siempre de el valor lógico de 1.

II.4.4.2. Falla Indeterminada: si su estado es diferente en un tiempo mayor o menor de T, por ejemplo, un número de fallas de hardware ocasionará que aparezcan cambios de estado entre los valores lógicos de 1 o 0.

II.5. SEGURIDAD

La seguridad requiere de tolerancia a fallas para mejorar el funcionamiento de cualquier sistema que esté diseñado para aplicaciones que requieren de la disponibilidad e integridad de su información..

Un sistema de alta seguridad no requiere únicamente de tolerancia a fallas, también de técnicas de software en las paqueterías usadas para que puedan respaldar la información o las bases de datos que se estén utilizando en situaciones críticas.

La seguridad es cuantificada de dos maneras:

a) DETERMINISTICAMENTE

b) PROBABILISTICAMENTE

II.5.1. DETERMINISTICAMENTE

Para medir la seguridad determinísticamente se deben de tomar el número máximo de componentes con funcionamiento incorrecto que el sistema soporta.

El problema que existe al usar este tipo de medida es que los vendedores de estos sistemas exageran la capacidad de sus productos, ofreciendo servicios que para que éstos funcionen, se tienen que agregar dispositivos especiales o simplemente no especifican cuanta degradación existe en su sistema por causa de chequeo de dispositivos, actualización de bases de datos redundantes o por módulos que se encuentren en funcionamiento incorrecto.

Otros aspectos que se deben de tomar en cuenta en este tipo de medida determinista, son la frecuencia, la probabilidad y el costo para corregir los funcionamientos incorrectos que ocurran.

II.5.2. PROBABILISTICAMENTE

Para medir la seguridad por probabilidad se manejan 2 términos:

II.5.2.1. CONFIABILIDAD.- Es la probabilidad condicional, R(t), de que un sistema sobreviva en el intervalo [to, t], dado que fue operacional en el tiempo to.

R(t) es una función que dependerá de los procesos de falla que afecten al sistema y de los procedimientos que impedirán que se produzca un funcionamiento incorrecto en el sistema cuando ocurra un falla.

En otras palabras, la confiabilidad nos indicará el tiempo en que un sistema se recupera de su funcionamiento incorrecto, si tiende a 0 es semejante a 0 en la función de confiabilidad la recuperación del sistema será casi instantánea, tal vez sin que el usuario final lo perciba, pero tomando en cuenta que dependiendo del tipo de falla o los tipos de falla que ocurran será el tiempo de recuperación del sistema.

En muchos sistemas de tiempo real se requiere de un alto nivel de R(t), por ejemplo, en una planta nuclear si ocurre alguna falla que produzca un funcionamiento incorrecto en el sistema de las barras estabilizadoras en el núcleo del reactor y si se tuviese un valor bajo de R(t), este producirá que las barras se detengan y no entren en sus cavidades para neutralizar a los iones que del material radioactivo se producen, creando un sobrecalentamiento del sistema, tal vez por un instante mientras el sistema se recupera, pero si llegáse a prolongar el tiempo de recuperación el núcleo se podría fundir creando fugas o contaminación de radiaciones.

II.5.2.2. DISPONIBILIDAD.- A(t), es una medida usada para sistemas sujetos a funcionamiento incorrecto y reparación y es definida como la probabilidad que un sistema es operacional en un tiempo t.

La disponibilidad puede ser considerada como un valor de Estado-Estable, ya sea como la probabilidad de que el sistema esté operando en cualquier tiempo aleatorio o como la acumulación de tiempo que el sistema estuvo fuera de servicio (caída del sistema), en un espacio de tiempo especificado, como por ejemplo podíamos acumular el tiempo que estuvo el sistema de cómputo HP-1000 de la ENEP-ARAGON fuera de servicio ya sea por mantenimiento o por caídas del sistema durante un año.

Otro ejemplo es el del sistema de conmutación electrónica del Sistema Telefónico Bell, que demostró tener una disponibilidad de 2 minutos de caída de sistema por año.

II.5.2.3. ALTA DISPONIBILIDAD.- En Este nivel se caracteriza por el uso de protocolos y otras operaciones de protección de bases de datos contra inconsistencias, además de recursos de diagnósticos y recuperación, también pueden seguir operando mientras son reparados, estas características elevarán el valor de disponibilidad a alta disponibilidad.

Se usan también 2 parámetros para medir el valor de Estado-Estable en la disponibilidad que son 2 medias estadísticas, evaluando tiempos de funcionamiento incorrecto y tiempos de reparación, estos son:

MTFI.- MEDIA DEL TIEMPO DEL FUNCIONAMIENTO INCORRECTO, Que es la expectativa del tiempo en el cual el sistema fallará.

MTR.- MEDIA DEL TIEMPO PARA SU REPARACION, es la expectativa de el tiempo para restaurar un sistema con funcionamiento incorrecto a su correcta operación.

II.6. DUPLICACION.

Es la réplica de un componente del STF. Se pueden tener dos o más componentes duplicados con el objeto de que pueda conmutar a otro componente duplicado, en el momento de que el componente principal tenga un funcionamiento incorrecto; por ejemplo, podemos tener tres UCP o dos unidades de almacenamiento (discos duros o memorias), al momento de que alguno falle, el otro disco duro o UCP será conmutado para que continúe con su operación.

II.7. COMPARADOR

Es un dispositivo que comparará las salidas de únicamente dos componentes, por ejemplo cuando se tienen UCP's redundantes, sus salidas será cotejada por el comparador y este activará una señal de error si las salidas son diferentes.

II.8. SELECCIONADOR.

Este tipo de dispositivo se utiliza en las técnicas Triple Modular Redundante. Se tiene tres módulos redundantes, si dos de ellos tienen como salida los datos "a" y "a" respectivamente y el tercero tiene como salida el dato "c". entonces el seleccionador tomará como dato correcto la mayoría de los datos que sean iguales, en este caso el dato "a".

Los comparadores son más confiables que los módulos redundantes que ellos administran, aunque uno solo represente una falla potencial, sin embargo usando los seleccionadores se aumenta la tolerancia a fallas así como su confiabilidad del STF, los seleccionadores son usados en los sistemas Triple Modular Redundante.

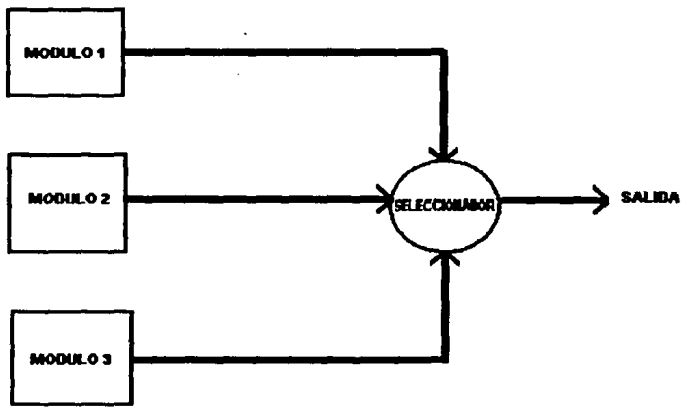

FIGURA II-4. SISTEMA TRIPLE MODULAR REDUNDANTE.

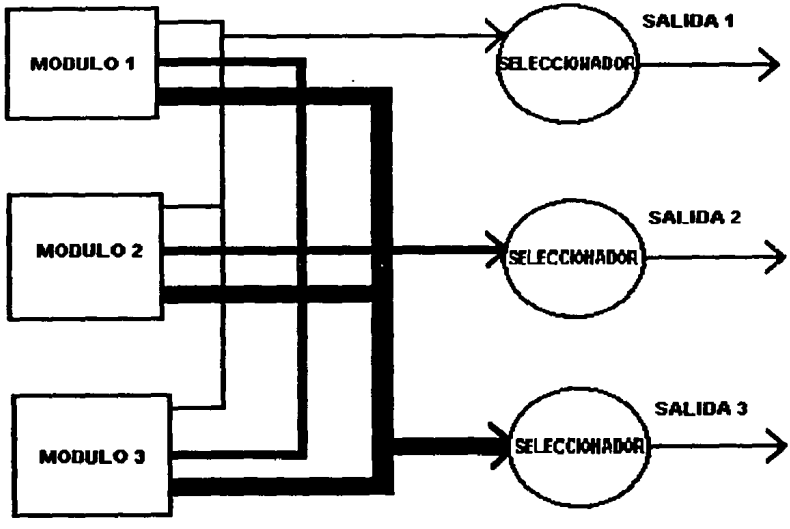

FIGURA II-5. SISTEMA TRIPLE MODULAR REDUNDANTE CON TRES SELECCIONADORES.

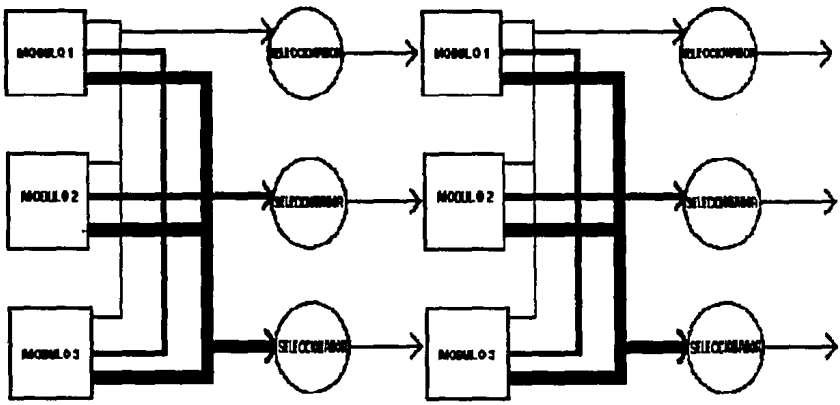

FIGURA II-6. SISTEMA TRIPLE MODULAR REDUNDANTE DUPLICADO.

CAPITULO III

CONCEPTOS BASICOS

III. CONCEPTOS BASICOS

III.1. ESTRATEGIAS PARA MANTENER UN SISTEMA EN EJECUCION NORMAL.

Se han establecido tres estrategias a nivel diseño que tratan de mantener en funcionamiento el sistema, de tal manera, que si las fallas llegaran a presentarse, estas no afecten la continúa operación y la integridad del sistema.

Estas estrategias son:

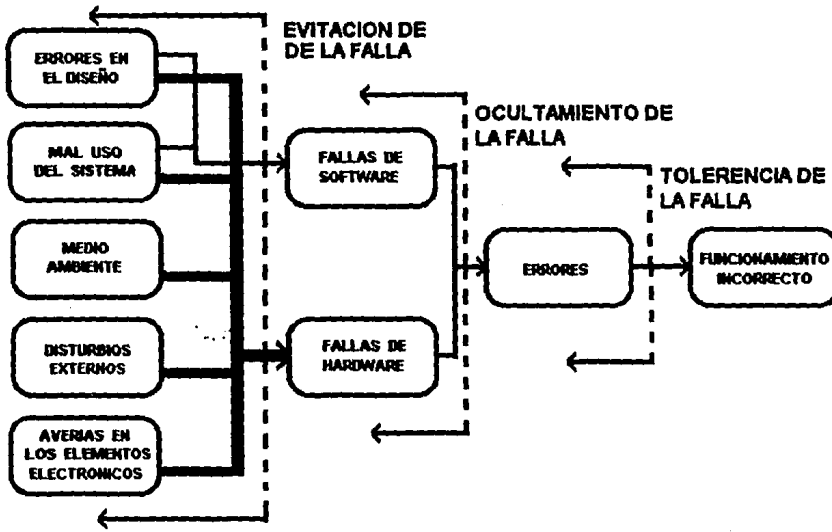

FIGURA III-1. ESTRATEGIAS PARA MANTENER UN SISTEMA EN EJECUCION NORMAL.

- EVITACION DE LA FALLA
- ENMASCARAMIENTO DE LA FALLA
- TOLERANCIA DE LA FALLA

III.1.1. EVITACION DE LA FALLA.

Para contar con un sistema confiable es necesario tomar en cuenta elementos con una alta calidad de fabricación, técnicas de ensamblaje o interconexión de los componentes así como pruebas contínuas de estos ensambles para eliminar las fallas de diseño en el hardware y en el software así como hacer pruebas con simulaciones de fallas para poder detectar errores latentes.

Los aspectos anteriores son usados para prevenir las causas de las fallas, lo que implica que esta estrategia será utilizada únicamente para la revisión de los diseños, protección de los componentes y pruebas del sistema.

III.1.2. ENMASCARAMIENTO DE LA FALLA.

Consideremos un sistema que se encuentra ejecutando un proceso y en un instante dado se presenta una falla en la UCP, entonces la presencia de esta falla dejará al sistema fuera de operación.

Con la implementación de la redundancia tanto en hardware y software, se puede conseguir mayor seguridad en el sistema. En el caso del ejemplo anterior es contar con otros procesadores que estén ejecutando la misma operación al mismo tiempo, que en el momento de la falla. alguno de estos procesadores tome el control de la ejecución del proceso. De tal manera que ante la ocurrencia de una falla ésta sea transparente para el usuario y mantenga al sistema en funcionamiento.

Por lo tanto, el enmascaramiento en cualquier sistema de tiempo real será el ocultar las fallas y permitir al sistema cumplir con las especificaciones de diseño.

El problema de usar la técnica de duplicación de componentes en esta estrategia es la degradación en la ejecución de los procesos, y no advierte de la presencia de una falla en el sistema. Esto es utilizado en aquéllos sistemas con memorias de código de corrección de error y la redundancia en los seleccionadores.

III.1.3. TOLERANCIA A FALLAS.

De las estrategias anteriores que se describieron, estan orientadas a eliminar las fallas en el diseño y ocultar la ocurrencia de la falla (enmascaramiento); en la primera estrategia, el tiempo de recuperación es mayor, ya que se necesita apagar al sistema para darle mantenimiento y no

cuenta con un sistema de detección. En la segunda estrategia, el tiempo de recuperación es menor ya que cuenta con componentes duplicados y usa la redundancia. Esto es, si un componente fallara el componente duplicado será conmutado a otro para que este opere, también en este tipo no se ofrece detección ni tampoco aislamiento, por lo tanto se genera una degradación en el sistema, pero permite cierta continuidad en los procesos. Para darle mantenimiento o corregir las fallas se tiene que apagar al sistema, para que se puedan efectuar estas operaciones en los sistemas que no cuentan con las técnicas de tolerancia a fallas.

En tolerancia a fallas se tiene una mayor continuidad ya que además de usar la duplicación y redundancia, cuenta con otras técnicas que tratan de reducir la degradación en el sistema y además para corregir las fallas, no es necesario apagar al sistema.

Si la tolerancia a fallas se encargará de que los módulos interactuen en forma organizada e íntegra, estas técnicas propias de los STF. se han clasificado de la siguiente manera:

DETECCION DEL ERROR
LOCALIZACION DEL MODULO EN FUNCIONAMIENTO INCORRECTO
CONTENCION DEL MODULO EN FUNCIONAMIENTO INCORRECTO
RECUPERACION DEL SISTEMA

Estas técnicas pueden variar en diferentes partes del sistema y en diferentes tiempos durante su operación.

III.2. TECNICAS DE SISTEMAS TOLERANTES A FALLAS

III.2.1. DETECCION DEL ERROR.

El propósito de esta técnica es la detección y corrección de errores en los códigos que existan en hardware y software que pongan en riesgo el funcionamiento del sistema.

Por los mecanismos de comunicación entre los componentes suceden varios tipos de errores que pueden presentarse en unidades de almacenamiento, en registros, memorias, fuentes de energía, o durante la transmisión de datos que son los mas fáciles de detectar que los errores originados en módulos que pueden modificar la información.

Varios verificadores de código para la detección y corrección de errores son empleados, entre los que se encuentran:

A) A NIVEL DE CODIGO:

BIT DE PARIDAD.
PARIDAD HAMMING.

REDUNDANCIA CÍCLICA.

M-OUT-OF-N.

B) A NIVEL DE MODULO:

AUTOVERIFICACION LOGICA.

DETECCION DE ERRORES EN MODULOS DUPLICADOS

CHEQUEOS DE TIEMPO

III.2.1.1. VERIFICADORES DE CODIGO PARA LA DETECCION DE ERROR Y CORRECCION DE ERRORES.

III.2.1.1.1. A NIVEL DE CODIGO

Estos verificadores de código son usados como medio para reducir la redundancia.

La verificación de flujos de códigos entre el procesador, la memoria y dispositivos de almacenamiento, seria imposible de realizar sin los verificadores de códigos.

Los verificadores de código tales como el bit de paridad, la paridad Hamming entre otros, nos permiten detectar ciertos tipos de errores únicamente.

BIT DE PARIDAD.

El método más utilizado es el del bit de paridad, que detecta errores en los "bus", en la memoria y en los registros. La desventaja es de que no realiza la corrección de errores, en caso de detectarlos.

Actualmente existen componentes VLSI que incluyen generadores de paridad para el bus y elementos de almacenamiento.

PARIDAD HAMMING.

La paridad Hamming es otro mecanismo de detección de error, que es utilizado para detectar errores en la memoria, pero además permite corregirlos.

REDUNDANCIA CICLICA.

El chequeo de redundancia cíclica se encarga en un primer ciclo de la verificación y en otro ciclo de corregir a los errores que pueden presentarse en los canales de comunicación y discos de almacenamiento.

"M-OUT-OF-N"

"m-out-of-n" es utilizado en la detección de errores en memorias programables tales como ROM's, así como también la detección de errores

aritméticos en la unidad lógica aritmética, como por ejemplo la división por cero.

III.2.1.1.2. A NIVEL DE MODULO.

AUTOVERIFICACION LOGICA.

Este verificador de código está diseñado para detectar fallas en circuitos lógicos, es decir, para detectar sus propias condiciones de error internos, por ejemplo cuando una fuente de energía entrega voltaje a los dispositivos que se encuentran conectados a esta fuente, si por alguna razón la fuente no entrega el voltaje requerido, inmediatamente se generará la activación de una señal que indicará un estado de error de la fuente de energía.

Otro ejemplo es en el caso de los ventiladores que si se reduce su velocidad más hayá de lo permitido, el circuito envíará igualmente una señal indicando el error presentado.

DETECCION DE ERRORES EN MODULOS DUPLICADOS

Estos verificadores de código usan circuitos comparadores conectados a las salidas de los módulos redundantes para detectar el componente que se encuentra en funcionamiento incorrecto.

Existen dos métodos para detectar errores en módulos redundantes:

Por comparador.- Este método compara la salida de dos módulos, si en la salida no son iguales, es generada una señal de error indicando que uno de los dos módulos tiene un funcionamiento incorrecto, pero este método no puede identificar cual de ellos está enviando los datos erróneos.

Existen dos configuraciones de comparadores, la primera configuración consiste en un comparador que es un dispositivo externo y la segunda configuración está incluída en los módulos y que consiste de un circuito íntegrado llamado circuito checador. Los módulos operan ejecutando la misma instrucción y al mismo tiempo en diferentes procesadores, todas las líneas de entrada son conectadas a los pines del circuito checador, cada circuito checador de cada módulo recibe las entradas, pero las líneas de salida son manejadas por un solo módulo, pero además las salidas estan dirigidas al circuito checador que será la entrada a un comparador que cotejará los valores traducidos.

Por seleccionador.- operan ejecutando la misma instrucción y al mismo tiempo en diferentes microprocesadores y compara la salida de 2 o mas componentes detectando cuando hay diferencia en la salida de ambos y además nos indica cual componente se encuentra en funcionamiento incorrecto.

VERIFICADORES DE TIEMPO

Si no se cumple cierto evento en un tiempo determinado este verificador mandará una señal de error. Tales eventos pueden ser verificados a través de un dispositivo llamado reloj guardián que medirá cada evento en un tiempo (T) determinado, si se pasa de (T) entonces se envíará una interrupción al sistema y se generará un error. Si el evento se realiza en menos de (T) entonces el reloj guardian reiniciará el reloj para el evento siguiente.

Un evento puede ser una transferencia de datos o la ejecución de una instrucción de un programa.

III.2.2. LOCALIZACION DEL MODULO EN FUNCIONAMIENTO INCORRECTO.

Esta técnica detectará el módulo que se encuentra en funcionamiento incorrecto e implementará el recobro apropiado por medio del ocultamiento del error ocurrido y simulando que el sistema se encuentra en un funcionamiento correcto.

Por ejemplo, si tenemos la transmisión de datos entre una computadora y una impresora, la computadora envía una señal de lista para que la impresora envíe los datos, pero supongamos que en una impresión la

impresora falla y la señal de lista no se activa, entonces el sistema se detendrá hasta que la señal de lista se encuentré activa.

El ocultar el error se refiere a que se simule que la impresora se encuentra trabajando correctamente para que el sistema no se mantenga en espera de la señal de lista.

III.2.3. CONTENCION DEL MODULO EN FUNCIONAMIENTO INCORRECTO.

Consiste en el aislamiento del módulo que se encuentra en funcionamiento incorrecto y además de prevenir que los datos o señales de control incorrectos, afecten el funcionamiento restante del sistema.

El circuito detector y corrector de errores, tales como los seleccionadores, comparadores o verificadores de errores, deben ser capaz de realizar el aislamiento del módulo si el error encontrado no puede ser corregido.

El problema de la técnica de contención es que para los circuitos detectores o correctores no pueden verificarse por sí mismo, en el momento de que éstos fallan.

Un ejemplo, es el funcionamiento del Triple Modular Redundante.

Cada UCP cuenta con su propia memoria, una UCP puede leer datos de las otras memorias pero solamente puede escribir en su propia memoria. Si ocurriese un error en una UCP y se enviará por este motivo a grabar un dato erróneo en su memoria, el seleccionador al momento de comparar los datos de las memorias detectará un error en una de ellas y aislará a la UCP y a la memoria, si no se aislara a la memoria, los otros UCP realizarían la operación de lectura en ella y entonces ocurriría una propagación de errores. Esta Arquitectura se encuentra en la máquina llamada SIFT.

III.2.4. RECUPERACIÓN DEL SISTEMA.

Cuando el sistema es reconfigurado ya sea por hardware o software cuando un error es detectado, debe de existir un tiempo para la recuperación antes de que el sistema sea corregido.

El lapso de tiempo entre la ocurrencia y detección de un error determina la cantidad del daño y el período del tiempo de recuperación.

Un esquema de recuperación para restaurar la correcta operación del sistema es regresar al sistema a un estado previo o punto de recuperación, conocido como punto de inspección.

En este esquema se encuentran 2 máquinas idénticas, una conocida como maestra y otro como esclava, en la maestra se efectuarán todas las tareas y

en la esclava únicamente se inspeccionará el funcionamiento de la maestra, además de realizar una copia de sus datos, estado de la lista de tareas que se estan procesando en la maestra.

Cuando ocurre un funcionamiento incorrecto en la maestra, ésta enviará una señal de error a la esclava, regresándose al último punto de inspección con los datos y estado de lista de tareas que copió de la maestra, posteriormente toma el control reanudando la ejecución de las tareas.

Otro esquema de recuperación en los sistemas de multiprocesadores con memoria compartida, es que, los datos globales y la lista de tareas son almacenados en la memoria.

Cuando un procesador falla en este tipo de sistema, es mas fácil continúar con la tareas después de que ocurrió la falla y además esta arquitectura permite distribuir las cargas de trabajo en las UCP.

III.3. REPARACION DEL SISTEMA

La reparación en los STF pueden ser realizados de dos formas:

A) Por reemplazamiento del módulo que encuentra en funcionamiento incorrecto.

B) Por reconfiguración de la estructura del STF.

III.3.1. REEMPLAZAMIENTO DEL MODULO.

Los módulos en funcionamiento incorrecto pueden ser removidos del sistema físicamente o lógicamente.

Cuando se realiza el reemplazo físicamente, los módulos restantes no reconocerán a este módulo dañado, como si éste nunca hubiése estado conectado al sistema.

Si el reemplazo es lógicamente, el sistema deshabilita la fuente de poder del módulo dañado, forzando sus salidas a un estado inactivo o haciendo que los módulos restantes ignoren al módulo dañado.

El reemplazamiento puede efectuarse en línea o fuera de línea. En algunos sistemas el reemplazamiento se puede realizar estando el sistema en línea (sin necesidad de apagar el sistema), esto es, no es necesario reinicializar el sistema. El sistema operativo se encargará de reconfigurar la estructura del sistema.

En cambio existen otro tipo de sistemas en donde la computadora tiene que estar apagada para poder realizar el reemplazamiento y encender para que se inicialice el sistema y sea reconfigurado.

III.3.2. RECONFIGURACION:

La reconfiguracion tiene la función de cambiar la estructura del sistema cuando se encuentre un módulo en funcionamiento incorrecto.

En un sistema triple modular redundante cuando ocurre un funcionamiento incorrecto en un módulo, el seleccionador reconfigurará el sistema para trabajar con los dos módulos restantes, pero si ocurriera una segunda falla en otro módulo el sistema se reconfigurará para trabajar con un solo módulo y el seleccionador deja de operar.

III.4. REDUNDANCIA

La redundancia son los requerimientos extras que un sistema digital necesita para poder realizar funciones de tolerancia a fallas.

Anteriormente, en los días en que se empezaban a diseñar los STF, la redundancia era entendida como el hardware duplicado, pero no se consideraba al software para el control del hardware adicional, además del tiempo de los procesos para el cambio de un procesador a otro, debería ser mayor, pero hoy en día se consideran 4 tipos de redundancia.

REDUNDANCIA EN EL TIEMPO
REDUNDANCIA EN LA INFORMACION

REDUNDANCIA EN EL HARDWARE

REDUNDANCIA EN EL SOFTWARE

III.4.1. REDUNDANCIA EN EL TIEMPO.

Es la adición de tiempo que se agrega para lograr la detección de errores cuando un error se ha presentado o para correr un proceso redundante.

III.4.2. REDUNDANCIA EN LA INFORMACION.

Es la adición de información extra que se utiliza en los datos (información) para detectar posibles errores en la transmisión de datos y corregirlos.

III.4.3. REDUNDANCIA EN EL HARDWARE.

Es la adición de módulos extras que se requieren para la detección o la tolerancia a fallas.

III.4.4. REDUNDANCIA EN EL SOFTWARE.

Es la adición de código extra para realizar una tarea definida para la tolerancia a fallas y también para la detección de errores.

III.5. SISTEMA DE SEGUIMIENTO DE LUZ.

Para entender mejor los conceptos de redundancia en el tiempo, información, hardware y software, consideraremos la aplicación de un sencillo sistema básico de seguimiento de luz, esto es, no contiene forma alguna de redundancia y posteriormente se aplican los conceptos de los 4 tipos de redundancia para crear un sistema básico tolerante a fallas..

En este sistema básico de seguimiento de luz sin tolerancia a fallas su función es detectar el lugar en donde se emite la mayor cantidad de luz.

Puede ser aplicado en la generación de energía eléctrica por medio de la energía solar. El sensor sigue la posición del sol y orienta a las celdas Solares hacia el astro luminoso para obtener la máxima energía solar.

En la siguiente figura se muestra el esquema de un sistema básico de seguimiento de luz:

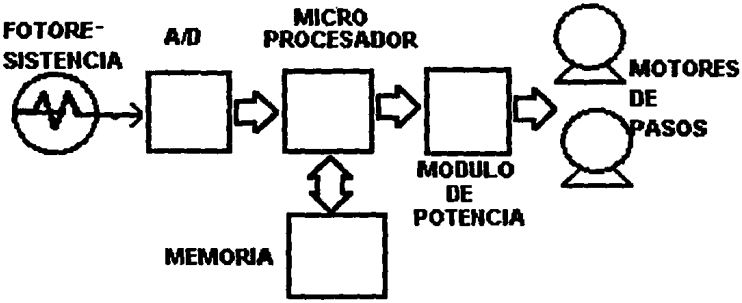

FIGURA III-2. SISTEMA BASICO DE RECEPCION DE LA LUZ SOLAR PARA EL MOVIMIENTO DE LOS MOTORES SIN REDUNDANCIA.

1.- SENSOR (FOTORESISTENCIA)

El sensor se ubica en el panel de las celdas solares y la fotoresistencia será la encargada de mandar un nivel de voltaje al convertidor analógico/digital, según la cantidad de luz detectada.

2.-CONVERTIDOR ANALOGICO/DIGITAL (A/D).

El convertidor analógico/digital cambiará el nivel de voltaje analógico a un código binario, el cual es envíado al procesador.

3.- PROCESADOR

El procesador leerá el código binario y lo comparará con datos anteriormente leídos y enviará un código de 8 bit's a el módulo de potencia.

4.- MODULO DE POTENCIA.

El módulo de potencia es el encargado de mandar la energía necesaria para que los motores se muevan, dependiendo de los datos que envíe el procesador.

5.- ACTUADORES (MOTORES DE PASOS).

Los motores de pasos son los responsables de mover el panel de celdas Solares.

Se trabajan con 2 motores, uno realiza los movimientos laterales y otro movimientos de arriba y abajo.

Como se puede observar este sistema no cuenta con tolerancia a fallas, pero implementaremos unas formas sencillas de redundancia para que el sistema se convierta en un sistema básico de tolerancia a fallas.

III.5.1. IMPLEMENTACION DE REDUNDANCIA EN EL TIEMPO.

Para tener una mayor seguridad de la información procesada se pueden realizar una segunda ejecución de las instrucciones, en el caso del sistema volvería a realizar la función de comparación de datos cada vez que lee un dato de entrada y comparando los dos resultados además de estar seguros que en la comparación los datos no difieren.

Como se comprenderá, se requerirá de mas tiempo para poder volver a procesar la comparación del mismo dato, a este lapso de tiempo extra se le llama tiempo redundante.

Para reducir la redundancia en el tiempo se utilizan microprocesadores mas rápidos o se duplican los procesadores para trabajar en paralelo, esto es, ejecutando las mismas instrucciones al mismo tiempo en cada procesador.

Esto también ayuda a detectar errores de tipo transitorio, si en una primera comparación tenemos un dato y en la repetición de la misma comparación hay diferencias, entonces tendrémos que en nuestra unidad aritmética lógica (UAL) existen fallas de tipo transitorio.

III.5.2. IMPLEMENTACION DE LA REDUNDANCIA EN LA INFORMACION.

Un buen ejemplo para aplicar la redundancia de la información al sistema de seguimiento de luz es la adición del bit de paridad a la palabra o dato que se envía del convertidor analógico/digital al procesador. con ésto se está agregando información extra a los datos de entrada al procesador.

III.5.3 IMPLEMENTACION DE LA REDUNDANCIA EN EL HARDWARE.

Para lograr la redundancia en el hardware en el sistema de seguimiento de luz, consideraremos de nuevo el caso del bit de paridad.

Para implementar el bit de paridad, se tiene que adicionar un circuito generador y verificador de paridad entre el convertidor analógico/digital y el procesador, como podemos observar se está agregando hardware al sistema.

Además de agregar los circuitos de generación y verificación de paridad entre el convertidor analógico/digital, se puede colocar en la memoria, el procesador y el módulo de potencia de los motores de pasos, quedando nuestro sistema de la siguiente manera:

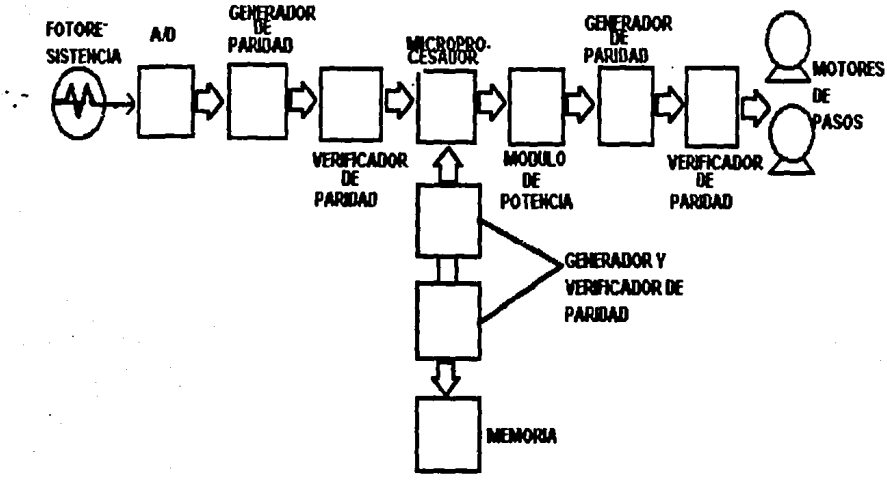

FIGURA III-3. SISTEMA DE RECEPCION DE LA LUZ PARA EL MOVIMIENTO DE LOS MOTORES CON REDUNDANCIA.

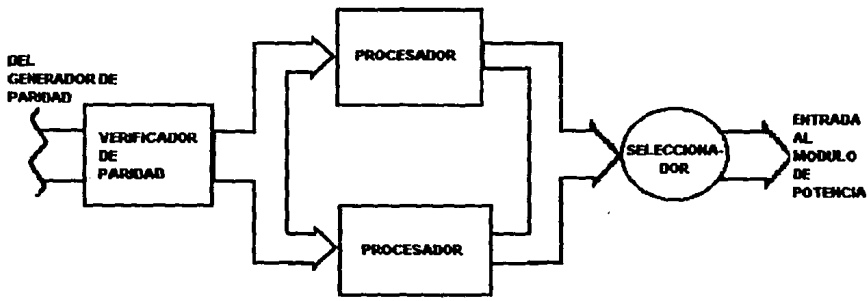

FIGURA III-4. DIAGRAMA A BLOQUES DE LA REDUNDANCIA DEL SISTEMA DE RECEPCION DE LA LUZ PARA EL MOVIMIENTO DE LOS MOTOTES.

III.5.4. IMPLEMENTACION DE LA REDUNDANCIA EN EL SOFTWARE.

En el sistema de seguimiento de luz, se conocen las posibles salidas que el procesador enviará para mover a los motores de pasos.

Para los motores de pasos, se tienen las siguientes palabras de salida de 4 bits, para cada motor.

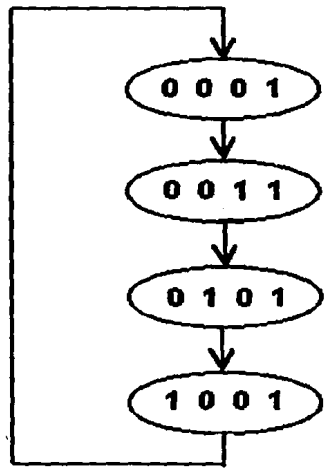

FIGURA III-5. DIAGRAMA DE ESTADOS DEL SISTEMA DE RECEPCION DE LA LUZ PARA EL MOVIMIENTO DE LOS MOTORES.

El diagrama de estados anterior muestra la secuencia de pasos para dar un sentido de giro al motor.

Como se conocen las salidas que el procesador enviará al módulo de potencia para cada motor y además estas salidas mantienen una secuencia, se puede realizar un programa que verifique esta secuencia antes de que sean enviadas al puerto de salida para mover los motores.

A la adición de estas líneas de código se le llama redundancia en el software.

Como se puede observar al adicionar líneas de código se tiene que aumentar el tamaño de la memoria (si se tiene una memoria fija y limitada para que realice las funciones básicas del sensor de seguimiento de luz), esto es, que además debe de haber redundancia en el hardware y redundancia en el tiempo para que sean procesadas las líneas de código adicionadas.

CAPITULO IV

UNIPROCESADOR

IV.1. UNIPROCESADOR

De las tres arquitecturas que estudiaremos la mas simple es la de uniprocesador, que está compuesto de un procesador (lógico), una memoria (lógica) y dispositivos de entrada/salida; como muestra en la siguiente figura.

(Para este capítulo se seleccionó el Sistema Integrity S2 de Tandem en el que se encuentran las características del uniprocesador).

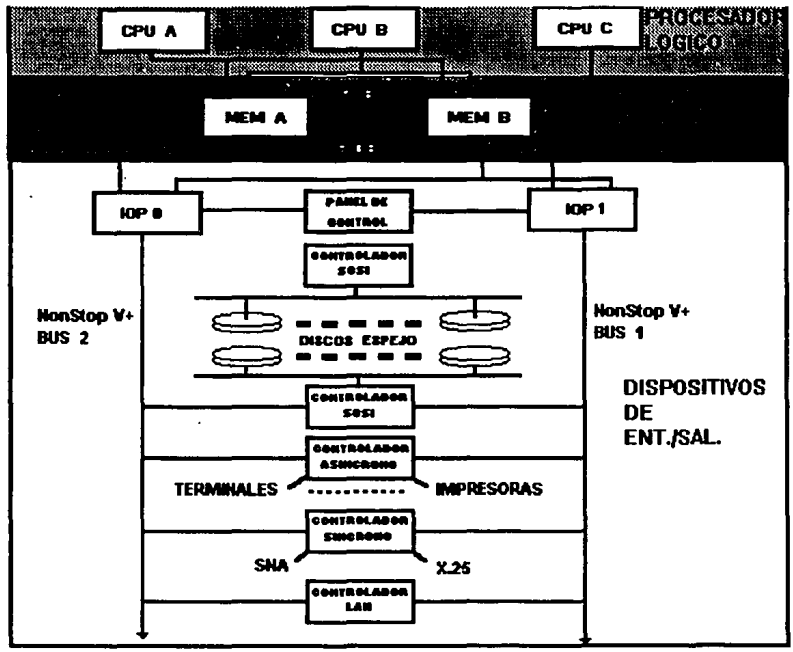

FIGURA IV-1. ARQUITECTURA DEL SISTEMA INTEGRITY S2 REPRESENTADO LA CONFIGURACION DE UN UNIPROCESADOR.

La arquitectura de la Integrity S2 de Tandem está compuesta por los siguientes módulos:

- **UCP'S**
- **MEMORIA**
- **CONTROLADORES DE PUERTOS DE ENTRADA/SALIDA**
- **SUMINISTRO DE ENERGIA**
- **INDUSTRIA ESTANDAR**

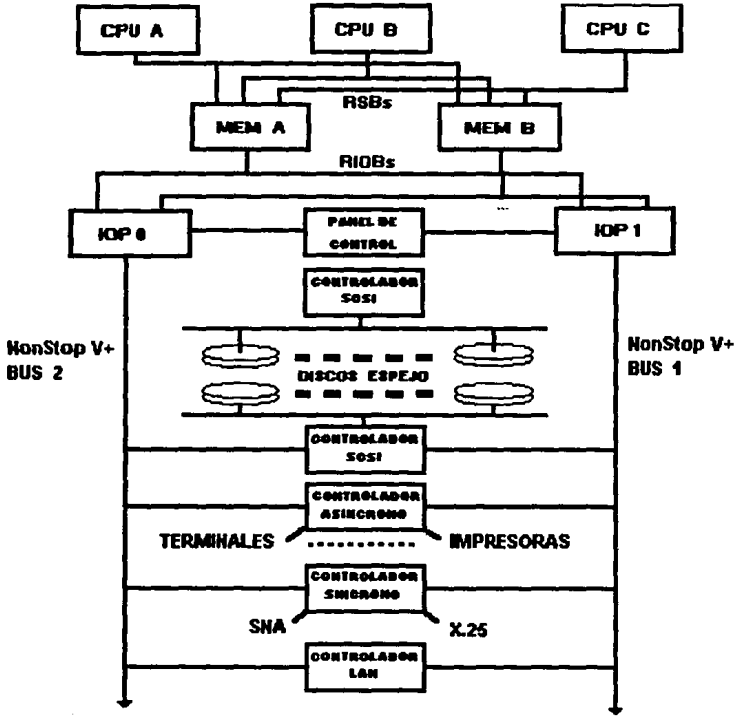

FIGURA IV-2. ARQUITECTURA DEL SISTEMA INTEGRITY S2.

La figura siguiente representa la configuración mínima con la que puede trabajar el sistema Integrity S2 de Tandem.

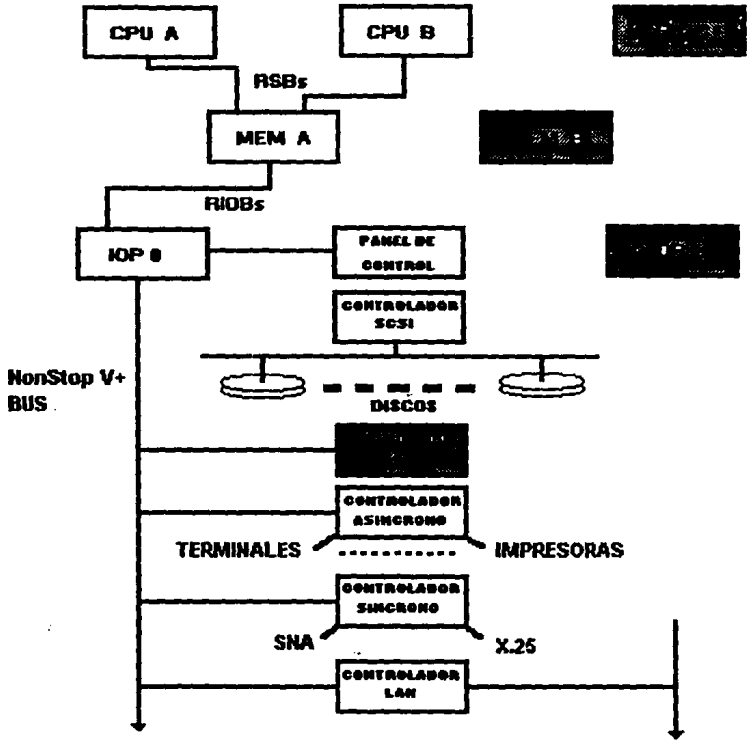

FIGURA IV-3. CONFIGURACION MINIMA DE FUNCIONAMIENTO DEL SISTEMA INTEGRITY S2.

IV.1.1. UCP'S

1.- Está formado por una arquitectura TMR que comprende tres UCP y dos seleccionadores para lograr una seguridad e integridad de los datos procesados.

2.- Los tres UCP trabajan como un simple procesador lógico, esto es, ejecutando la misma instrucción en cada uno de las UCP.

3.- Cada UCP trabaja con su propio reloj y son sincronizados periódicamente por la misma instrucción.

4.- Cada UCP consta de una memoria asíncrona local de 128 kbyte, 64 kbyte para el manejo de datos y 64 kbyte para instrucciones.

IV.1.2. MEMORIA

1.- Esta formado por dos módulos de memoria que trabajan como una sola memoria lógica, esto es que tienen la mismas funciones y datos, Estas memorias trabajan en forma asíncrona.

2.- Los dos módulos de memoria responden a la información proporcionada por las tres UCP y a los dos controladores de puertos de entrada/salida. (PES'S).

3.- Los módulos son designados por software, uno como primario y el otro como respaldo.

4.- Las operaciones de escritura son ejecutadas en ambos módulos.

5.- Las operaciones de lectura se realizan en el primario para la detección de errores y manejo de los bus de datos.

6.- El módulo primario recibe los accesos desde los UCP's y los controladores PES y el módulo de respaldo será forzado a ejecutar las mismas operaciones.

7.- Si la memoria primaria llega a fallar, se desactivará y el módulo de respaldo tomará el lugar como primaria efectuando todas sus operaciones.

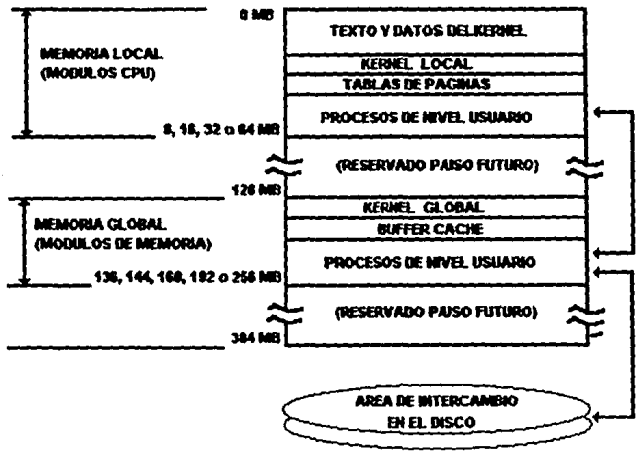

FIGURA IV-4. ADMINISTRACION DE LA MEMORIA.

IV.1.3. CONTROLADORES DE PUERTOS DE ENTRADA/SALIDA (PES).

1.- Consta de dos controladores de entrada/salida, no ejecutando las mismas operaciones.

2.- Cada uno de los controladores es administrado por software, pero uno de ellos tendrá la capacidad de manejar todas las comunicaciones en el caso de que un controlador falle.

IV.1.4. SUMINISTRO DE ENERGIA

1.- Consta de una etapa de energía de corriente alterna.

2.- Cuenta con dos fuentes de poder de corriente directa, que suministra un voltaje de +36 volts de CD a los módulos convertidores de corriente directa (CCD), a los circuitos de carga de baterías y a los módulos de los ventiladores.

Cada fuente de poder maneja dos señales independientes que el software utiliza para detectar el funcionamiento incorrecto de la unidad de distribución de energía, condiciones de calentamiento en el gabinete así como fallas en la fuente de poder.

3.- Cuenta con dos módulos de baterías que suministran +24 volts en los módulos convertidores de corriente directa (CCD) para el caso de que ocurra un falla en el suministro de energía.

4.- Contiene 8 módulos convertidores (CCD), que suministran +5 y +-12 volts. Los cuatro primeros CCD alimentan a las UCP's, memorias, PES y el módulo del panel de control.

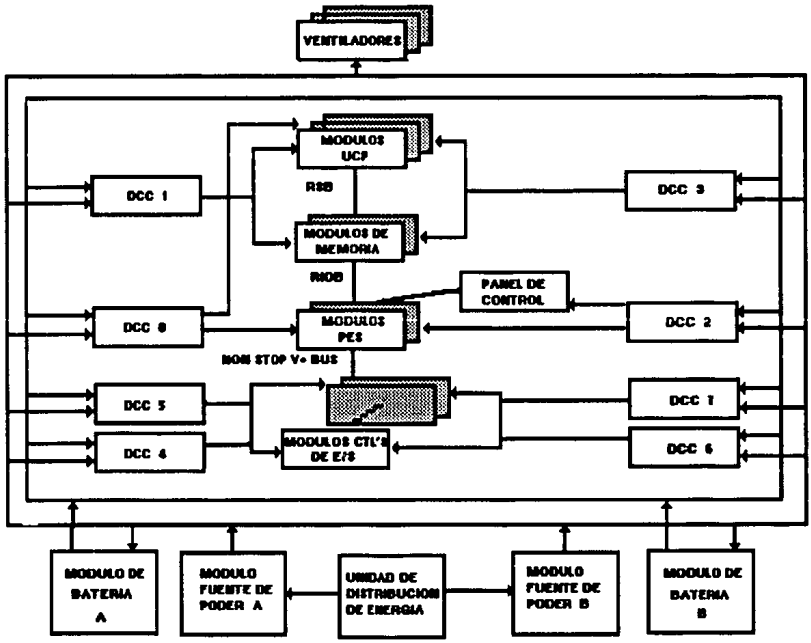

FIGURA IV-5. SISTEMA DE ENERGIA.

IV.1.5. INDUSTRIA ESTANDAR.

1.- Maneja una industria estándar de I/O para una disponibilidad y flexibilidad en diferentes medios ambientes de computadoras.

La industria estándar que trabaja son: SCSI, ASYC, SYNC y ETHERNET LAN.

IV.2. HARDWARE TOLERANTE A FALLAS.

La arquitectura de la Integrity S2 tiene una estructura para que el sistema operativo continúe en un proceso normal sin ninguna interrupción después de que ha ocurrido un funcionamiento incorrecto en alguno de sus componentes.

El sistema operativo es el encargado de colocar fuera de línea a los módulos con funcionamiento incorrecto.

La arquitectura del sistema Integrity S2 tiene una rápida recuperación para el manejo de módulos con funcionamiento incorrecto aislándolos para que no se extiendan errores en el sistema.

Cuenta con dos registros, un registro de estado que almacena los detalles sobre las causas de las fallas y el registro de control, nos indica qué

módulos con funcionamiento incorrecto pueden ser deshabilitados para después poder ser reemplazados y habilitados.

IV.2.1. MANEJO DE MODULOS CON FUNCIONAMIENTO INCORRECTO.

El software usa la información que manda el control de hardware para manejar a los módulos con funcionamiento incorrecto de la siguiente forma:

1.- Aísla la falla.

2.- Coloca en fuera de línea el módulo con funcionamiento incorrecto para que los módulos estantes no sean alterados en su información.

3,- El módulo es probado en forma aislada del sistema para determinar si ocurrió una falla transitoria o una falla permanente.

4.- El módulo es reintegrado en el caso de que sea una falla transitoria, o si sucede una falla permanente el sistema es reconfigurado para que se trabaje sin el módulo con funcionamiento incorrecto.

IV.2.2. DETECCION DE FALLAS.

Para la detección de fallas el sistema Integrity S2 utiliza diversos mecanismos tales como:

- Operaciones de Seleccionadores (Voting) duplicados

- Códigos de detección de error para el almacenamiento o transferencia de datos.

- Verificación sobre el tiempo y secuencia de comunicaciones entre los módulos del hardware (Watchdog).

- Autoverificación de los módulos del hardware.

IV.2.2.1. OPERACIONES DE SELECCIONADORES DUPLICADOS.

Se utilizan unos dispositivos llamados seleccionadores para comparar la información de salida de las UCP.

Los seleccionadores se encuentran ubicados en los módulos de memoria, porque es la interfaz entre las UCP y la memoria, de aquí su importancia, ya que los datos que se almacenan en la memoria deben de ser correctos.
Existen dos seleccionadores en el sistema Integrity S2, los cuales realizarán la función de comparar los datos provenientes de cada UCP, si un dato de

alguna de las UCP fuera diferente, entonces se volvería a procesar la instrucción en la UCP que mandó el dato erróneo y se volvería a comparar el datos de salida de ésta, si el dato es correcto se continuaría con una operación normal, pero si el dato volviera a ser erróneo, la UCP responsable será aislada y se mandará un mensaje a la consola del sistema sobre la situación de los procesadores.

IV.2.2.2. CODIGOS DE DETECCION DE ERROR

Se utilizan dos códigos de detección de error:

- bit de paridad
-checksums

BIT DE PARIDAD

Se utiliza el bit de paridad en los módulos de memoria para garantizar el almacenamiento de los datos así como su acceso a los mismos, además se usa en los subsistemas de E/S para asegurar la transferencia de los datos entre los módulos.

CHECKSUMS

El checksums es usado para verificar el éxito de la transferencia de bloques de datos entre el disco y la memoria.

No solo los datos son verificados, sino también las direcciones para asegurar que los datos sean leídos y escritos en las direcciones correctas.

El bit de paridad, el checksums y los seleccionadores duplicados garantizan la integridad de los datos.

IV.2.2.3. VERIFICACION SOBRE EL TIEMPO Y SECUENCIA DE COMUNICACIONES ENTRE LOS MODULOS DEL HARDWARE .

Este mecanismo es usado por el hardware; a cada solicitud de servicio se asigna una cierta cantidad de tiempo y este mecanismo se encargará de que el servicio solicitado sea realizado en ese espacio de tiempo y conforme a un protocolo.

Por ejemplo, tenemos que la UCP quiere almacenar algunos datos en la memoria global, entonces la memoria tendrá que reconocer el éxito de la escritura en un tiempo determinado, si ésto no ocurriese se genera un error conocido como error de tiempo de salida. (timeout).

IV.2.2.4. AUTOVERIFICACION DE LOS MODULOS DE HARDWARE

Como mencionamos anteriormente la autoverificación trata de que cada módulo es responsable de detectar sus propias condiciones internas de error.

En el sistema Integrity S2 podemos ejemplificar la autoverificación en los siguientes módulos:

- Cuando en el módulo de la fuente de energía tiene una caída de tensión, esta fuente responderá con una señal de estado de error.

- Cuando la velocidad de un ventilador baja, el módulo que controla a los ventiladores enviará una señal de error.

- En un disco, cuando éste no responde a su controlador dentro de un tiempo determinado (watchdog), el controlador responderá con una señal de error.

- En el sistema de la Integrity S2 también existe la autoverificación en los seleccionadores, los cuales cuentan con dos microprocesadores y un comparador para detectar posibles diferencias en los datos de salidas.

IV.3. SOFTWARE TOLERANTE.

En el caso del primer nivel para la evitación de las fallas en el sistema integrity S2, se eliminaron los errores de software en el sistema operativo Non Stop-UX para prevenir posibles errores en el sistema.

IV.3.1. CONFIABILIDAD DEL SISTEMA.

En el sistema Integrity S2 de Tandem, el software es la parte controladora del hardware tolerante, es decir, por medio del software tratará de mantener al sistema en un continuo funcionamiento, para que ésto se logre se tendrá que prevenir y corregir los errores producidos o latentes del software.

Existen 4 característica principales en el sistema Integrity S2 para prevenir y corregir los errores que ocurran en el software, y éstas son:

ROBUSTEZ DEL SISTEMA.
DISCOS ESPEJOS.
SISTEMA AUTOMATICO DE CERRADO Y REINICIO DEL SISTEMA.

IV.3.1.1. ROBUSTEZ DEL SISTEMA.

ELIMINACION DE ERRORES DEL UNIX ESTANDAR.

Una de las mejoras del sistema operativo Non Stop-UX fue la eliminación de muchos errores de software que el sistema operativo UNIX estándar contenía.

ANALISIS Y CORRECCION DE CONDICIONES DE PANICO.

Las condiciones de pánico son errores que ocurren o se encuentran latentes en el software del sistema operativo.

En un sistema operativo UNIX estándar cuando ocurre una condición de pánico, el sistema es detenido.

Para las condiciones de pánico que son mas probables que ocurran, el sistema operativo está provisto de mecanismos de recobro ante estos tipos de errores, para lograr que el sistema continúe en funcionamiento.

RUTINAS DE SUPERVISION PARA VERIFICAR LA CONSISTENCIA DE LA ESTRUCTURA DE DATOS DEL KERNEL.

Existen rutinas de supervisión que se utilizan para lograr el recobro del sistema cuando la integridad del kernel se está en duda.

Cuando el kernel es dañado u ocurre un error de sotfware y el recobro no es posible, el administrador del sistema tiene 2 opciones para realizar el recobro:

a)Correr el sistema en un estado de prueba.

El sistema será probado durante un cierto período de tiempo teniendo como nivel de referencia un umbral de configuración, si el grado de error rebasa este umbral el sistema es detenido, pero si el grado de error no rebasa al umbral, el sistema regresará a un estado normal de operación.

b) detener y reiniciar rápidamente el sistema, para que sea limpiado.

El mecanismo de pánico asegurará que todos los datos dañados serán salvados. El mecanismo de pánico guardara todos los datos dañados en archivos especiales, ésto ayudará a disminuir el tiempo de reboot.

PROTECCION DEL KERNEL POR HARDWARE.

La Integrity S2 está provista de una protección de la memoria por medio del hardware para evitar un daño a las localidades de memoria ocupadas por el kernel y consecuentemente ocurran condiciones de pánico.

IV.3.1.2. DISCOS ESPEJO.

Este sistema cuenta con un dispositivo llamado disco espejo, para asegurar todavía mas a la seguridad del sistema, éste consta de dos discos que pueden ser particionados, el contenido de una partición de un disco será copiado o reflejado sobre la partición del otro disco, el sistema puede ser configurado para que los discos no compartan los módulos controladores

PES o "SCSI", ésto con el fin de que si el acceso a un disco fallara, el acceso al otro disco que contiene la información reflejada no sea afectado.

Las particiones en el sistema Integrity S2 de Tandem, son secciones lógicas, que dividen a un disco.

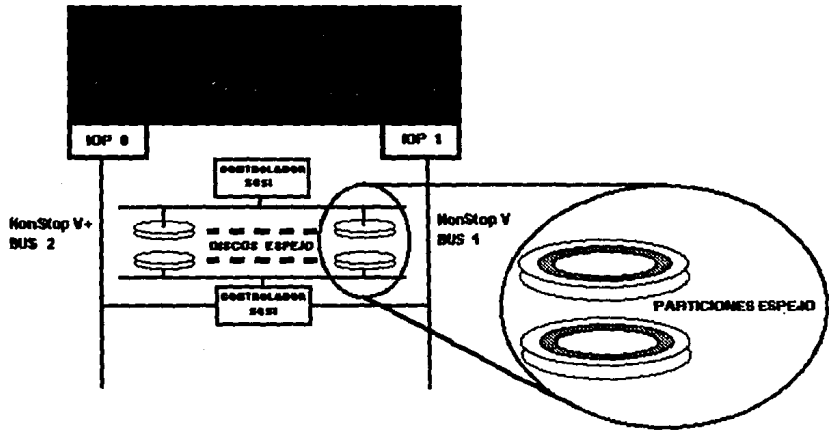

FIGURA IV-6. DISCOS ESPEJO.

Los discos de la Integrity S2, contienen particiones hechas de fábrica que pueden ser reflejadas sobre otras particiones, éstas son:

- la partición que contiene el sistema operativo Non Stop-UX.

- la partición del código de inicio (boot)
- la partición del sistema de archivos
- la partición de datos salvados durante la falla de energía.
- la partición que contiene datos de dispositivos renombrados (alias), etc.

La única partición que no puede ser reflejada sobre otra partición es la que contiene el volúmen encabezador .

Se llamará partición primaria, a la que contendrá los datos originales y se le llamará partición secundaria a la que contendrá el reflejo de los datos de la partición primaria.

Las particiones primaria y secundaria una vez relacionadas como espejo se manejarán como un solo dispositivo, a este tipo de dispositivo se le conoce como dispositivo espejo.

El sistema operativo provee de utilerías para manejar el reflejo entre las particiones y puede ser usado por el administrador del sistema, estando el sistema corriendo.

Las utilerías manejadas para los dispositivos espejo son:

- Examinar la compatibilidad de las particiones para poder realizar el reflejo.

- Crear un dispositivo espejo lógico para 2 particiones y realizar una copia del dispositivo primario al secundario.

- Examinar el estado de todos los dispositivos espejo.

- Terminar y reiniciar la función de reflejo entre las particiones.

- Borrar un dispositivo espejo, así como la asociación entre las particiones.

- Comparar 2 particiones y revisa si son idénticas.

IV.3.1.3. SISTEMA AUTOMATICO DE CERRADO Y REINICIO.

Cuando ocurra una falla de energía ocasionado por la falta del suministro de energía de corriente alterna o que la ausencia de energía sea muy prolongado y rebase el tiempo de respaldo de los módulos de baterías el sistema será cerrado de una forma normal.

También si existe un sobrecalentamiento dentro del gabinete, el sistema será cerrado, evitando posibles fallas del sistema.

IV.4. SISTEMA DE ENERGIA.

El software será el encargado de manejar el funcionamiento incorrecto del sistema de energía de corriente alterna.

El sistema será mantenido con la energía de una batería mientras sucede la ausencia de energía. El software cerrará el sistema de una forma inteligente, salvando el estado del sistéma durante el cierre por medio de un ciclo reanudador y el estado salvado será reanudado cuando el sistema sea reiniciado, utilizando el ciclo reanudador las aplicaciones que se encontraban activas, podrán seguir continúando, después de reiniciar al sistema con pequeñas o ninguna pérdida de datos.

Si se usara el ciclo reinicio, todas las aplicaciones activas serían canceladas durante el cierre del sistema.

El ciclo reanudador o ciclo reinicio son las dos opciones que el administrador del sistema puede usar cuando suceda un funcionamiento incorrecto de suministro de energía de corriente alterna.

El cierre y reiniciación del sistema pueden ser configurados de diferentes formas, según sean las necesidades del usuario.

Como por ejemplo, sí se tienen 2 unidades de almacenamiento masivo, y usamos a una unidad como el reflejo de la otra, se puede configurar al

sistema para que, en el caso de que en alguna de ellas ocurriera un funcionamiento incorrecto en el suministro de energía, la otra unidad puede continúar en operación sin que el sistema sea cerrado, pero si ocurriese el funcionamiento incorrecto de ausencia de energía en la cabina principal (en donde se encuentran los UCP´S, las memorias, etc.) o en las dos unidades de almacenamiento masivo, entonces el sistema procedería a cerrarse.

Si se tuviera una falta de energía, el sistema continuará en operación durante unos cuantos segundos bajo un sistema de baterías, sin que sea iniciado el proceso de cierre, esto es, para el caso de que se presentara un funcionamiento incorrecto del tipo trascendente, o en el caso de que se reanudara el sistema de energía en ese lapso de tiempo, pero si no ocurriese así el sistema empezaría el proceso de cierre del sistema.

Cuando la energía es restablecida, el sistema tiene un lapso de espera aproximadamente de un minuto, asegurando que no halla fluctuaciones de energía, y dando tiempo a que se recarguen las baterías por si sucediera otro funcionamiento incorrecto de energía de corriente alterna.

Solamente cuando la energía es establecida y las baterías han sido recargadas a un nivel de seguridad adecuado, se empezará el proceso de reinicialización del sistema.

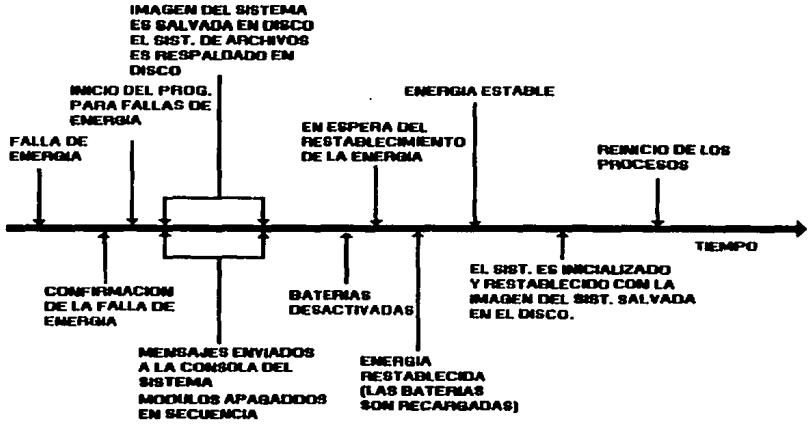

FIGURA IV-7. TIEMPO DE RESTAURACION ANTE LA PRESENCIA DE UN FUNCIONAMIENTO INCORRECTO DE LA ENERGIA.

IV.5. RECOBRO DEL SISTEMA

Existen dos partes fundamentales en donde se realiza el recobro de la falla, en el CPU lógico, la memoria lógica y en los subsistemas de entrada y salida.

IV.5.1. RECOBRO DE LA UCP Y MEMORIA LOGICA

Aquí intervienen las dos capas del sistema (Hardware y Software) en el cual el hardware permitirá continúar los procesos mientras que el software aislará el módulo con funcionamiento incorrecto como el caso del UCP

lógico que en el momento que se encuentre una salida errónea, ésta es ignorada por el software pero también se mantienen ejecutándose los procesos en los dos UCP's restantes. En el caso de la memoria, el hardware le indicará al sistema, cual es la memoria en funcionamiento correcto y la reconocerá como principal para el sistema.

IV.5.2. RECOBRO DEL SUBSISTEMA DE ENTRADA Y SALIDA

El error en el subsistema de entrada y salida se puede presentar en los puertos o en los controladores. Al presentarse un error en alguno de los controladores, los datos que se encontraban en ese instante son perdidos de tal manera que es llevado fuera de servicio por el sistema operativo.

Al perderse un Puerto de entrada y salida, los controladores que se encontraban enlazados al puerto son redireccionados a otro que encuentre en correcto funcionamiento.

IV.6. REÍNTEGRACION

La reintegración puede hacerse de dos formas principalmente, cerrando el sistema y cambiando el módulo en funcionamiento incorrecto y volviendo a levantar al sistema para que se inicialice y se reconozcan todos los módulos.

La otra forma es realizando la reintegración del módulo dañado estando el sistema en línea, ya sea manualmente o por una terminal.

Dependiendo de las condiciones del funcionamiento incorrecto, se intentará la reintegración en el módulo existente o se pediría que el módulo sea reemplazado.

IV.6.1. REINTEGRACION DEL UCP

Antes de que un módulo de UCP se reintegre, éste debe de realizar una auto prueba de funcionamiento. Una vez hecha la verificación de las funciones del UCP, la reintegración es llevada a cabo con una pequeña restauración de los tres UCP's para su sincronización, la nueva UCP tomará los datos desde la memoria global, debido a que los dos UCP's realizaron un proceso de copiado de la memoria global a la local antes de reintegrar la UCP faltante.

CAPITULO V

MULTIPROCESADOR

V. MULTIPROCESADOR

Aplicaciones como el del procesamiento paralelo de datos críticos requieren de computadoras que tengan un alto rendimiento, confiabilidad, disponibilidad y escalables. Para alcanzar este propósito, se han desarrollado arquitecturas tolerante a fallas en Multiprocesadores.

En términos generales un Multiprocesador es una computadora que incluirá varios procesadores que se pueden comunicar y cooperar a diferentes niveles para resolver cualquier problema. La comunicación se puede realizar envíando mensajes de un procesador a otro o compartiendo la memoria.

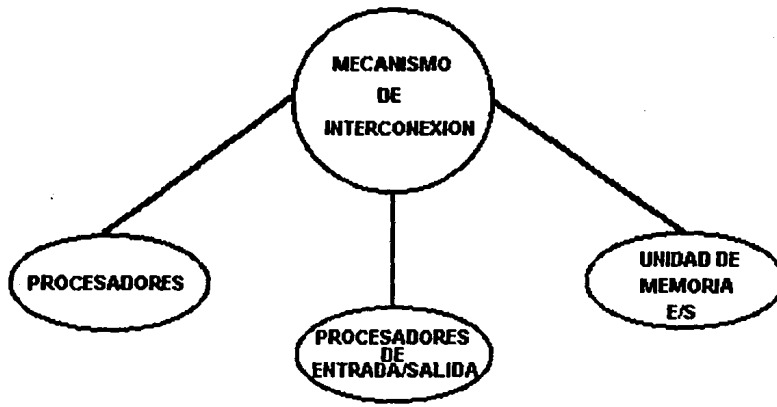

FIGURA V-I. ORGANIZACION DE UN SISTEMA DE MULTIPROCESAMIENTO.

Existen dos Arquitecturas para un Multiprocesador:

Multiprocesador Ligeramente Acoplado

Multiprocesador Estrechamente Acoplado

V.1. MULTIPROCESADOR LIGERAMENTE ACOPLADO

En estos sistemas, cada procesador tiene un conjunto de dispositivos de entrada-salida y una gran memoria local a donde se accesan todos los datos e instrucciones. Los procesos que llegan a ejecutarse en los procesadores, la memoria o los dispositivos de entrada-salida se comunican mediante el intercambio de mensajes a través de un sistema de transferencia de mensajes.

Tandem utiliza este tipo de Arquitectura, siendo su ventaja, el aislar el módulo en funcionamiento incorrecto. Cada procesador falla independientemente sin alterar el funcionamiento de otros procesadores.

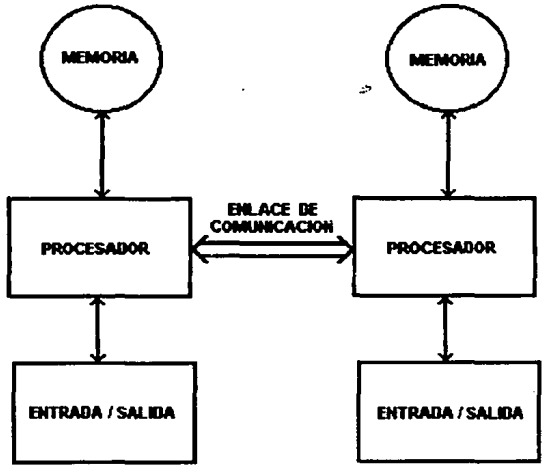

FIGURA V-2. MULTIPROCESAMIENTO LIGERAMENTE ACOPLADO.

V.2. MULTIPROCESADORES ESTRECHAMENTE ACOPLADOS

A través de una memoria principal compartida los procesadores se comunican a una misma velocidad, en cada uno de ellos puede existir una pequeña memoria local o una memoria asíncrona de alta velocidad. La desventaja principal de esta arquitectura es su memoria compartida, esto es, en el momento que falle causará que todos los procesadores queden en funcionamiento incorrecto.

El compartir automáticamente y distribuir la carga de los recursos, es el principal beneficio de esta arquitectura. Todos los procesadores comparten una sola cola de procesos que estan listos para ejecutarse, de tal modo que

todos ellos soporten la gran cantidad de trabajo que se les solicite. Además únicamente una copia de software de cada módulo es necesario para que ocupe la memoria y sea compartida por todos los procesadores.

En una arquitectura de Multiprocesamiento ligeramente acoplado, los procesos deben de ser explícitamente asignados a cada procesador, permitiendo una reducción potencial de la carga. Tal reducción puede ser reparada moviendo el proceso a un diferente procesador; sin embargo, tal movimiento de los procesos entre los procesadores es bastante complejo,

La computadora SEQUOIA es un sistema de Multiprocesamiento Estrechamente Acoplado que trata de aprovechar las ventajas y eliminar sus desventajas de este tipo de Arquitectura a través de la implementación de STF.

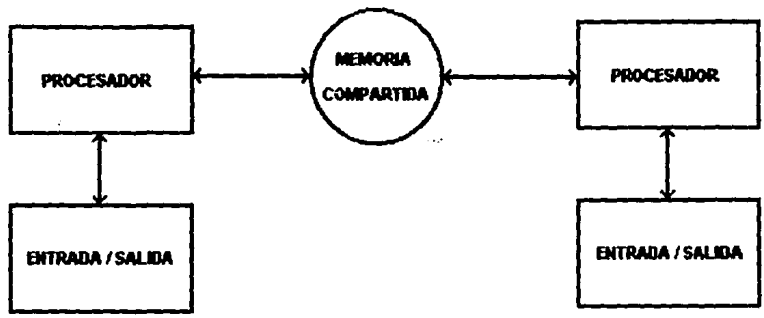

FIGURA V-3. MULTIPROCESAMIENTO ESTRECHAMENTE ACOPLADO.

V.3. ARQUITECTURA EN HARDWARE

La computadora SEQUOIA consiste de tres principales elementos: procesador, memoria, y dispositivos de entrada/salida que están conectados a un bus.

V.3.1. ESTRUCTURA DEL BUS

El sistema de bus consiste de dos bus que operan independientemente, que esta compuesto de tres elementos:

Procesador Local
Memoria Local
Entrada-Salida Local.

Cada Procesador Local puede enlazar a ocho elementos a un bus Global por medio de una interface maestra (IM). Cada memoria local también puede enlazarse al bus Global a través de una interface esclava (IE). La interface maestra monitorea el acceso a los Bus.

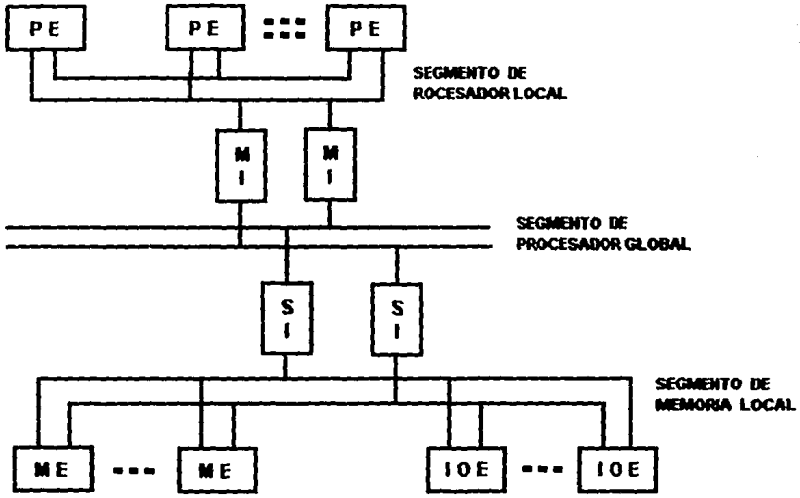

FIGURA V-4. ARQUITECTURA DEL SFT SEQUOIA

V.3.1.1. PROCESADOR LOCAL

Cada procesador local tiene dos microprocesadores Motorola MC68020 que opera a una velocidad de 20 Mhz. Su operación de funcionamiento de los dos microprocesadores es en "lokc step",: es decir, se ejecuta la misma instrucción en el mismo instante que son cotejadas por comparadores que verificarán las operaciones idénticas en cada ciclo de reloj. Cada procesador tiene un reloj local, de tal modo que si hay un funcionamiento incorrecto de uno de ellos, únicamente se deshabilitará el procesador donde reside.

Las actualizaciones escritas en la memoria asíncrona por el MC no son inmediatamente realizadas en la memoria global. En su lugar el sistema operativo debe de preguntar explícitamente al procesador de los bloques de datos encontrados en la memoria principal.

El sistema operativo puede decidir en elegir el refrescar la memoria global con los datos almacenados en la memoria asíncrona o puede haber un desbordamiento. Esto ocurrirá cuando nuevos datos estan referidos a la memoria asíncrona que deben de ser almacenados, pero no hay lugar para éstos, por lo que se decide que los datos o bloques sean movidos de la memoria asíncrona a la memoria global para hacer un espacio para los nuevos datos.

El propósito especial del hardware en los procesadores es la actualización de los bloques de datos en la memoria asíncrona con una sola instrucción. Donde la mitad de esta instrucción es de lectura y la otra de escritura.

V.3.1.2. MEMORIA LOCAL

Cada memoria tiene una capacidad de almacenamiento de 8 a 16 MB de RAM, teniendo un tiempo de acceso de 100 nanosegundos. El sistema operativo asegurará la exclusión mutua al acceso a la memoria compartida

V.3.1.3. ENTRADA Y SALIDA LOCAL

Los puertos de entrada y salida (PES) consiste de dos componentes

ADAPTADOR DE BUS (AB)

ADAPTADOR DE MULTIBUS (AM)

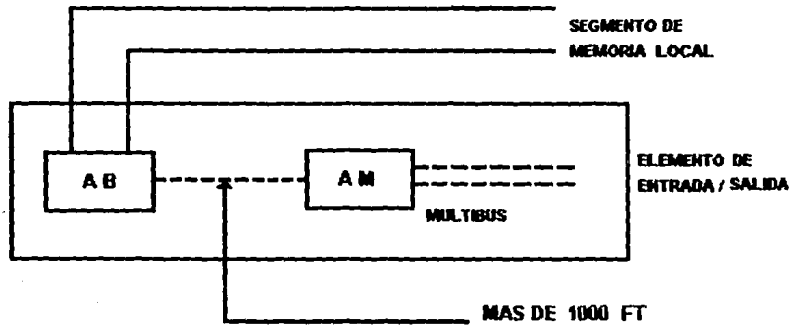

FIGURA V-5. ELEMENTO DE ENTRADA Y SALIDA.

V.3.1.3.1. ADAPTADOR DE BUS

Se enlaza al segmento de la memoria y al AM, que funciona como el bus de los dispositivos de entrada-salida.

V.3.1.3.2. ADAPTADOR DE MULTIBUS

Cada Multibus tiene 2 microprocesadores de autoverificacion, 2 MB de memoria que es utilizado como memoria de paso " buffer " y una PROM para almacenar programas. Cada Multibus soportará hasta 12 controladores para cinta, discos, comunicaciones, etc.

V.4. TOLERANCIA A FALLAS

La detección es la técnica que siempre se considera para elementos críticos que han manifestado una falla cuando estan en operación.

Algunas de las técnicas que son utilizadas son:

Código de detección de error
Comparación de las operaciones duplicadas
Monitoreo de los Protocolos

V.4.1. CODIGO DE DETECCION DE ERROR

Todos los datos son protegidos por códigos de detección de error, tanto los que son almacenados como los transferidos a memoria o al bus.

V.4.2. COMPARACION DE LAS OPERACIONES DUPLICADAS

Debido a que el código de detección de error es mucho mas económico para las unidades de almacenamiento, no así para los microprocesadores; la generación de direcciones lógicas y funciones de administración de la memoria asíncrona se refiere a utilizar la comparación y la duplicación en el hardware. Los comparadores verificarán independientemente todos los datos duplicados, bit por bit, para que posteriormente entre ellos se comparen los resultados obtenidos y se envíe un mensaje al sistema operativo sobre el estado del sistema.

V.4.3. MONITOREO DE PROTOCOLOS

La combinación de los códigos y la comparación de las operaciones duplicadas no detectará todas las fallas en hardware. Por ejemplo, si un procesador hace referencia a una dirección de memoria, pero no hay respuesta, entonces el procesador o la memoria que estan conectados al bus permanecerán deshabilitados esperando una respuesta que nunca llegará. El propósito de esta técnica, es la de detectar las violaciones en la secuencia y la expiración del tiempo entre la intercomunicación de los elementos

Las técnicas de detección informan inmediatamente de la presencia de una falla cuando se presenta, pero la única falla que puede quedar sin

detectarse es del mismo hardware de detección. El sistema operativo tiene la función de verificar que el hardware de detección cumpla con las funciones para las que se diseñó y en caso de no cumplirlas deshabilitarlo para evitar que afecte a otros elementos de hardware.

V.5. ESTRUCTURA DEL SISTEMA OPERATIVO

Las ventajas de un sistema operativo estándar y la combinación de un alto requerimiento de disponibilidad y confiabilidad, SEQUOIA implementa su propio Kernel que ofrece una gran variedad de funcionalidad con el sistema operativo UNIX V.

El kernel de la SEQUOIA ofrece al usuario un nivel de tolerancia a fallas en los procesos y archivos con una rápida recuperación.

V.6. RECOBRO DE LA FALLA

En cualquier momento, el sistema podrá manifestar una falla en cualquiera de los elementos (procesador, memoria, dispositivos de entrada-salida, etc.). Debe de ser posible la recuperación sin ninguna pérdida de información o la duplicación en las operaciones de entrada-salida. Para esto el Kernel de la computadora SEQUOIA garantizará el estado de los procesos en memoria global manteniéndolos en un estado de consistencia.

V.6.1. RECOBRO EN EL PROCESADOR

Supongamos que un procesador falla (y por lo tanto hace su propia desconexión del bus), en el momento cuando no esta envíando segmentos de datos a su asíncrona. Estará una imágen en del proceso que se estaba ejecutando antes de que sucediera la falla en la memoria asíncrona, de tal manera que el sistema operativo necesitará identificar el proceso asignado al procesador dañado para posteriormente asignarlo a una cola de listo y sea ejecutado por otro procesador.

Pero ¿Qué pasa cuando la memoria asíncrona falla cuando está envíando bloques de información a la memoria global?. Cuando ésto sucede, la memoria asíncrona llega a tener un estado de inconsistencia en la memoria global debido a que contiene datos correctos e incorrectos antes y después de que ocurriera la falla. Para evitar este problema, en el sistema SEQUOIA, se ha implementado un sistema de reflejo en dos memorias diferentes.

Cuando la memoria asíncrona de un procesador envía bloques de datos a las memorias lo realiza dos veces, en un primer ciclo de tiempo se envía a la memoria de respaldo y en el siguiente a la memoria designada como primaria, de esta forma se mantiene en algunas de las dos memorias la consistencia de los datos. Cuando sucede la falla en el procesador y se estan envíando una copia de los bloques de datos a la memoria de respaldo sus datos serán inconsistentes y únicamente se mantendrá la consistencia

en la memoria primaria un ciclo antes de que sucediera la falla y es el mismo caso para la memoria primaria.

Cualquiera que sea el caso, una copia inconsistente puede ser procesada nuevamente realizando la lectura de la copia en consistencia para que inmediatamente el proceso sea envíado a una cola de ejecución.

V.7. FUNCIONAMIENTO INCORRECTO DE LA MEMORIA

Todas las operaciones de escritura son reflejadas en los dos módulos de memoria y los de lectura son respaldados en disco. De tal modo, si un módulo falla, todas las páginas se encontrarán en cualquier parte del sistema.

V.8. EVITACION DE PERDIDA EN LOS PUERTOS DE ENTRADA-SALIDA

Cada PES tiene una cola de operaciones pendientes en la memoria principal .La ejecución de una operación de entrada-salida es desarrollada por el procesador envíando al proceso a una cola de ejecución de entrada-salida.

Supongamos que el procesador falla durante la inicialización de las operaciones de entrada-salida. El estado de la memoria es recuperada ya

sea por un estado anterior de la consistencia en la asíncrona o por un nuevo estado. Después de una falla, no siempre es posible para los procesadores indicar cual de las operaciones de entrada-salida han sido enviadas al puerto de entrada/salida. Para evitar la pérdida de las operaciones de entrada/salida, es necesario validar la cola de los procesos listos para ejecutarse y que hayan sido direccionados a los Puertos de entrada/salida. Por lo tanto, después de una falla, cada puerto de entrada/salida es interrogado para determinar cual proceso se ejecutó. Los puertos de entrada/salida mantienen una lista de todas las operaciones en progreso para este propósito.

V.8.1. FALLAS EN LOS DISPOSITIVOS DE ENTRADA/SALIDA

Para evitar las fallas en los dispositivos de almacenamiento secundario se utilizan la duplicación de información en dos diferentes unidades de almacenamiento. Estas unidades de almacenamiento se han designado como discos espejos que estan conectados a diferentes controladores de entrada/salida. El sistema operativo realiza la escritura en ambos discos.

Si una unidad de almacenamiento llega a sufrir una falla, el otro disco toma la carga de trabajo. Si un controlador o puerto de entrada-salida falla, entonces se intenta utilizar una ruta alterna al dispositivo; si el intento falla, el disco espejo es utilizado. Los discos espejos son recuperados en línea.

La comunicación en línea puede ser mas confiable a través de la conmutación a los controladores de diferentes puertos de entrada/salida. Si un controlador falla, el sistema operativo o la aplicación puede conmutar a las líneas de otro controlador.

Una falla permanente de un puerto de entrada-salida es tratado como el funcionamiento incorrecto total de sus controladores; en el caso de una falla transitoria se tratará de reiniciar los PES desde la memoria principal.

V.9. EL PROCESO DE RECUPERACION

Las fallas detectadas en los procesadores es detectado por el registro del selecciondor. Cada procesador cuenta con 128 bytes llamados bloques que se utilizan para especificar el estado del procesador, siendo actualizado cada 100 ms.

Se designa un procesador para seleccionar el estado de los otros procesadores, para determinar periódicamente si alguno de los procesadores estan en funcionamiento correcto. Se utiliza el mismo método para los otros procesadores, que verificarán al procesador seleccionador y esté en funcionamiento correcto. Si cualquiera de los procesadores detecta el funcionamiento incorrecto de alguno de ellos, asumirán la carga de trabajo y colocarán al sistema en un proceso de recuperación del procesador en funcionamiento incorrecto.

Después que el sistema ha fallado y existe un estado potencial de incertidumbre de funcionamiento, únicamente un procesador realizará el proceso de recuperación en un intervalo de tiempo.

Un procesador ejecutará una secuencia de pasos:

- Encontrar el Kernel en la memoria
- Manifestar las anomalías
- Completar el flujo de datos
- Completar las llamadas al sistema

Encontrar el Kernel en la memoria.

El procesador debe ser capaz de encontrar todo el código del kernel para completar el proceso de recuperación. Debe de ser posible aislar al elemento en funcionamiento incorrecto en el sistema.

Manifestar las anomalías.

El procesador intentará encontrar la causa de la falla para ejecutar un pequeño análisis de diagnóstico de quien genera la falla. Después de haber hecho el diagnóstico, el sistema indicará la naturaleza del problema.

Completar el flujo de datos.

El procesador notificará de la falla en el momento que ésta sea detectada. En particular, podrá enviar sus datos a la memoria asíncrona cuando se le notifique de la existencia de la falla.

Completar las llamadas al sistema.

Después que el procesador ejecutó los tres pasos anteriores, el sistema iniciará el proceso de recuperación. Esta fase será ejecutada por el procesador que es designado como el Procesador Ejecutivo, cualquier procesador puede cumplir esta función. La falla del Procesador Ejecutivo es tratado de forma ordinaria que son tratados todos los procesadores y simplemente causará que el proceso de recuperación se restaure desde el principio.

Después de diagnosticar la falla, el procesador ejecutivo ajusta la configuración del hardware, indica cuales módulos estan conectados al sistema y funcionan apropiadamente. La entera recuperación lleva normalmente de uno a dos segundos.

V.10. STF PARA SECCIONES CRITICAS.

Después que la recuperación es finalizada, los procesos continuarán ejecutando a partir de donde surgió la falla, pero puede ser que la configuración del hardware no sea la misma antes de que surgiera la falla, por lo que la correcta reanudación del sistema no debe de depender en ninguna forma de la configuración del hardware.

Estos requerimientos es algunas veces difícil de fijar la dirección física de la memoria o la localización de un registro, debe ser parte del estado de un proceso. Por ejemplo cuando un procesador inicializa el acceso directo a memoria (DMA) para mover los datos de la memoria al puerto de entrada/salida, la localidad física en la cual el dato es movido debe de estar en la memoria asíncrona del procesador cuando es inicializada el DMA. Supongamos que en el momento que el procesador envía datos a su memoria asíncrona para su ejecución se inicia una interrupción por alguna falla. Después de la recuperación, se debe de tratar de continúar la operación en el punto donde se inició la recuperación.

Si la dirección de la memoria fuera utilizada antes de que la falla existiera en la memoria física, tal intento podría resultar en un acceso de datos incorrectos, por lo que el proceso de recuperación no será posible. El sistema operativo continuará tratando de realizar la recuperación, pero siempre se tendrá el mensaje de una falla.

Para evitar este problema, el sistema operativo en cualquier momento debe de estar trabajando con la información del estado del hardware impidiendo el flujo de datos a la memoria asíncrona. Se maneja su pérdida de una forma especial, para evitar que se envíen datos a la memoria asíncrona y ocurra un desbordamiento.

V.11. SINCRONIZACION DE PROCESOS EN LA MEMORIA COMPARTIDA

A menudo, los procesos que son ejecutados en un sistema de multiprocesadores deben de comunicarse y sincronizarse. La ejecución de un proceso puede influir en otro a través de la comunicación. Los procesos que se comunican lo hacen por medio de la sincronización. Un proceso que se ejecuta con una velocidad muy rápida y genera sucesos, deben de ser reconocidos por otro proceso cooperante. El conjunto de restricciones sobre el ordenamiento de estos sucesos constituyen la sincronización requerida por los procesos operantes. El mecanismo de sincronización utilizado retarda la ejecución de un proceso a fin de satifacer tales restricciones.

Para ayudar a los procesos a la sincronización en los accesos a la memoria se hace uso del semáforo. Los semáforos y sus operaciones pueden implementarse en software o hardware. En SEQUOIA, se implementan en el núcleo del sistema operativo, donde se controlan los cambios de estado de un proceso.

Cuando un proceso emite una petición de entrada-salida se bloquea a sí mismo para esperar que concluya esta operación. Algunos procesos deben de activar al proceso bloqueado.

CAPITULO VI

MULTICOMPUTADORAS

VI. MULTICOMPUTADORAS

Cuando un sistema uniprocesador es duplicado completamente, la arquitectura resultante será la de una multicomputadora.

En cada sistema uniprocesador residirá una copia del sistema operativo, además de contar con un canal de comunicación entre los sistemas.

Esta arquitectura es muy versátil para el mantenimiento de los STF, ya que cuando ocurre un funcionamiento incorrecto en cualquier uniprocesador, este puede ser reemplazado completamente.

La Vax ft 3000 es un SFT que es usada como un procesador "stand-alone", y puede ser utilizado para realizar transacciones en línea

Un sistema "stand-alone" consiste en:

- Dos interfaces para el bus interprocesador conectado a cada procesador

- Una unidad de procesamiento de instrucciones

- Memoria principal

- Dos canales de interface de entrada/salida, para permitir la comunicación con los distintos tipos de servicios periféricos.

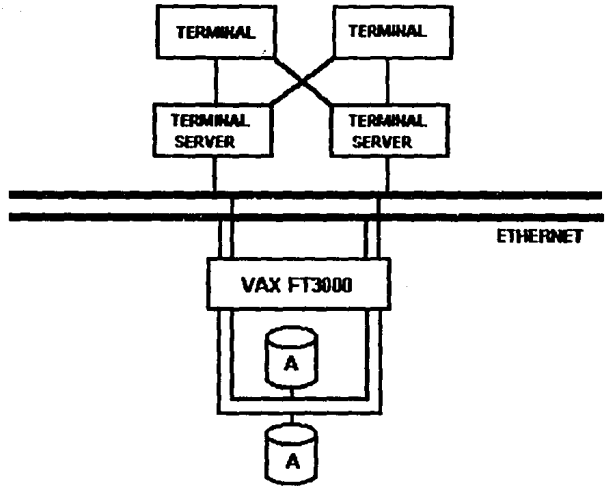

FIGURA VI-1. SISTEMA VAXFT3000 PARA CAPTURA DE DATOS.

Los procesadores deben de soportar procesamiento en paralelo, además de proporcionar tolerancia a fallas.

En un sistema "stand-alone" se utiliza como un procesador de alta disponibilidad.

En un sistema Cluster se puede utilizar como un procesador mas a la red pero con las características de una computadora tolerante a fallas, se usa como un respaldo en los procesos, cuando alguna otra computadora del sistema falla ésta toma control de los procesos y continúa con su operación.

Cuando se habla de transacciones en línea se debe entender como un sistema dedicado a procesos de captura de datos en tiempo real, tales como transacciones en línea (abonos y cargos en transacciones bancarias), sistema de captura crítica, procesos industriales, manufacturación, comunicación de datos, monitoreo de laboratorio, etc.

Para estos sistemas se debe de tener una rápida recuperación en el momento de que ocurra una falla.

Los objetivos de la computadora Vax ft 3000 para lograr un STF así como una alta disponibilidad son:

VI.1. NINGUN PUNTO DE FUNCIONAMIENTO INCORRECTO.

Se refiere a que en un STF una vez ocurrido un error no se debe mostrar ni una degradación de sus recursos por causas de algunos de sus componentes que se encuentren en reparación o aislados del sistema principal.

VI.2. REPARACION EN LINEA.

Esto quiere decir que al sistema se le puede dar mantenimiento sin interrupción de sus operaciones (apagar al sistema).

Como el sistema esta formado por dos zonas, una vez que una zona ha fallado esta se debe de aislar mientras la otra zona se mantiene en funcionamiento, así se podrá cambiar totalmente a la zona que se encuentra en funcionamiento incorrecto completamente sin necesidad de apagar la otra zona.

VI.3. PROTECCION CONTRA ERRORES ACCIDENTALES.

Es la zona que esta en reparación y pueda físicamente ser retirada, mientras que la otra zona continúa ejecutando las aplicaciones correctamente.

VI.4. AUTOVERIFICADORES DE DETECCION DE ERRORES DEL HARDWARE.

Este autodiagnóstico se realiza por el sistema operativo para detectar el funcionamiento incorrecto de los errores que pudieran ocurrir en el hardware detector de fallas.

VI.5. OPERACION TOLERANTE A FALLAS.

El STF de la computadora Vax ft 3000 está compuesta de dos computadoras idénticas, una llamada zona A y otra zona B, en la cual la verificación de funcionamiento es realizada por los controladores de memoria.

Cada zona trabaja en autoverificación y todo el sistema trabaja en "lock step", esto es, en ambas zonas se ejecutan los mismos programas, al mismo tiempo con los mismos datos, en forma sincronizada. Se dice que cada zona trabaja en autoverificación, por que cuando ocurre un funcionamiento incorrecto en alguna de las zonas, ésta se aislará mientras que la zona que se encuentra en funcionamiento correcto continuará trabajando de una forma trasparente para el usuario y los procesos que se encuentran en operación, sin tener una degradación en el sistema.

La principal ventaja de este STF es en el momento de que ocurre un funcionamiento incorrecto, el sistema provocará que el suministro de energía deje de funcionar, desactivando la zona afectada.

La zona en funcionamiento correcto realizará una llamada a un centro de atención indicando que ha ocurrido una falla permanente.

La arquitectura de este STF facilita su reparación sin que haya ningún momento de suspensión de los procesos, la razón de esto es que se trabaja

por zonas, entonces la zona afectada se aísla y se puede reemplazar totalmente sin afectar a la zona en funcionamiento correcto.

VI.6. RECUPERACION

Cuando se ha corregido el funcionamiento incorrecto de alguna de las zonas el contenido de la otra zona (en funcionamiento correcto) es copiado a la otra zona que fue reparada.

El medio que se encarga de realizar todas las actualizaciones del sistema es el "cross link", este dispositivo de realizará todas las operaciones de actualización del sistema, para que ambas zonas, se encuentren con la misma información.

Cuando el "cross link" haya efectuado el copiado de la información y las dos zonas se encuentren iguales, el sistema operativo utilizará el estado de la información en la memoria principal, entonces se establecerá un nuevo estado de inicio, esto es, teniendo la misma información en la memoria principal, el sistema operativo iniciará desde este punto todas sus operaciones y así poder lograr la resincronizacion en ambas zonas, después de esto, el hardware tomará el control de la operación "lock-step" y el sistema será recuperado.

VI.7. ARQUITECTURA TOLERANTE

El STF del sistema Vax ft 3000 esta formado por dos zonas, cada una de ellas formada por una computadora con elementos duplicados.

Las zonas se encuentran enlazadas por un dispositivo de comunicación llamado "cross link". por el cual se envía al sistema operativo el estado de ambas zonas.

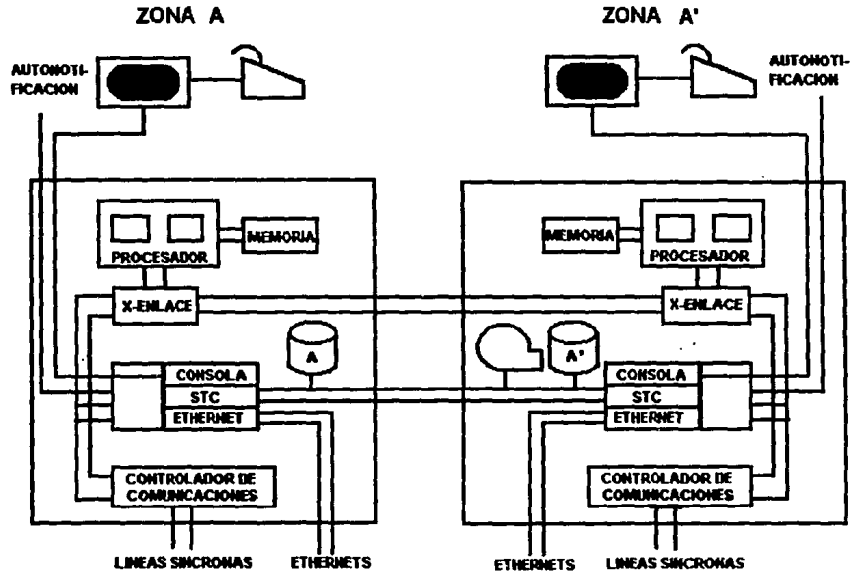

FIGURA VI-2. ARQUITECTURA DEL SISTEMA TOLERANTE A FALLAS DE LA VAXFT3000.

Cada zona consiste de los siguientes elementos:

Procesador lógico

Controlador de Memoria

Cross link

Suministro de energía ininterrumpible

fire-wire

VI.7.1. PROCESADOR LOGICO

El procesador lógico esta constituido por dos procesadores físicos que se encuentran trabajando "lock step" y también realizan una verificación uno del otro.

VI.7.2. CONTROLADOR DE MEMORIA

Es el intermediario entre los procesadores físicos y la memoria. Se encargará de comparar los datos que envían ambos procesadores físicos y generará una señal de error en el momento que se detecte un error en los datos. Los controladores de memoria enviarán una copia a la memoria principal y dos copias de direcciones. En las copias de los bits de direcciones serán envíados el bit de código de detección de error y la señal de control, a su vez la memoria generará los bits de código de detección de

los datos envíados y los comparará con los bits de código de detección de error recibidos.

VI.7.3. CROSS- LINK Y EL FIRE WIRE

Son dispositivos que sirven de interface para el aislamiento y la detección de las fallas. El "cross link" se encargará de actualizar la base de datos en ambas zonas, además de servir de enlace de comunicación para la detección de posibles errores en algunas de las zonas. El "fire wire" es el medio por el cual se verificarán los datos de entrada salida. Cuando se envía una serie de datos al "fire wire" para que sean transmitidos o recibidos, este dispositivo comparará los datos con los códigos de detección de error que fueron transmitidos o recibidos. Cuando el "fire wire" detecta un error, el elemento responsable es aislado, se reintentará la operación, pero si no se logra corregir, se envíarán los datos a una ruta alterna. Aún si persiste el error, el sistema llamará al centro de atención para su verificación.

VI.8. METODOS DE DETECCION

Los dos métodos de detección que son utilizados en STF de la Vax ft 3000 es la duplicación de módulos y los códigos de detección de error.

VI.8.1. DUPLICACION DE LOS MODULOS

Es el principal método utilizado en STF para la detección de errores. Los módulos que se encuentra en el sistema Vax ft 3000 que utilizan el método de la duplicación son:

El procesador
El controlador de memoria
Cross- link
fire wire

Todos estos módulos utilizan la autovericación para la detección de errores en sus datos generados por el hardware.

VI.8.2. CODIGOS DE DETECCION Y CORRECCION DE ERROR

Los códigos de detección y corrección de error son utilizados por los módulos no duplicados, por ejemplo en los elementos de entrada y salida y en bus de datos. En la memoria se generan códigos de detección de error de los datos recibidos que comunican a los módulos. Se tomará en cuenta que las señales de control y de direcciones deben de ser iguales para todas las operaciones que se ejecuten en la memoria, para que la memoria compare estos códigos con los códigos envíados por el generador de paridad.

Los controladores de memoria se encargan de comparar la salida de los procesadores envíando una sola copia de bits de datos a la memoria mientras que las direcciones y las señales de control son envíadas en forma duplicada a la memoria. La memoria verificará estas direcciones para su detección de errores.

VI.9. ELEMENTOS DE ENTRADA Y SALIDA

Para el manejo de los datos de entrada y salida, éste se realiza por medio de paquetes de información, los cuales además de contener los bits de datos mantendrán los códigos de detección de error, así como también contendrá el código verificación del dispositivo de entrada y salida asociado.

Estos paquetes de información cuando son enviados por un módulo duplicado, el "fire wire" comparará las salidas antes de ser envíados a la sección entrada y salida.

En los dispositivos de entrada y salida son almacenados, además de los bits de datos, los códigos de detección de error, para que cuando sean llamados por un módulo, el "fire wire" calcule un código de detección y corrección de error para compararlo con el código de detección almacenado.

De ésta manera con la duplicación de módulos y el código de detección y corrección de error se mantiene una alta integridad en la información.

CAPITULO VII

OTRAS CONFIGURACIONES PARA LA SEGURIDAD DE LA INFORMACION

VII. OTRAS CONFIGURACIONES PARA LA SEGURIDAD DE LA INFORMACION.

VII.1. ALTA DISPONIBILIDAD Y ALTA TECNOLOGIA VAXCLUSTER

En un sistema Vax Cluster son conectadas dos computadoras que trabajan como si fuera un solo sistema, compartiendo la carga de trabajo y la base de datos. También son utilizados en aplicaciones que requieren de una gran potencia de procesamiento y protección contra fallas.

Si una computadora llega a estar fuera de línea por una interrupción por falla, la carga de trabajo es compartida entre los procesadores que permanecen funcionando.

Los sistemas Vaxcluster son escalables en términos de la cantidad de disponibilidad que ellos ofrecen. Todos los recursos disponibles tales como discos, cintas, base de datos, impresoras terminales, etc. Estan disponibles para todas aquellas computadoras que se enlazan al sistema en diferentes configuraciones de comunicación remota.

En este capítulo describimos las características funcionales de cada una de las configuraciones que podemos encontrarnos en la actualidad con los sistemas **Digital**, que a consideración nuestra es la tecnología que está a la vanguardia en este tipo de sistemas.

VII.2. CARACTERISTICAS DE LOS SISTEMAS VAXCLUSTER

Las UCP´s en un sistema VaxCluster pueden compartir las unidades de almacenamiento masivo y la administración de otros recursos. En este ambiente de alta integridad, las UCP´s mantienen su independencia de su propia memoria y del Sistema Operativo. Así cada UCP puede fallar o inicializarse independientemente de otros recursos comunes o de otros procesadores.

La aplicación se ejecutará en uno o mas UCP´s en un sistema de Vaxcluster compartiendo los recursos de una manera coordinada. El software del sistema sincronizará el acceso a los recursos compartidos, previniendo que dos o mas procesos interfieran uno con el otro cuando estan actualizados los datos. Esta coordinación asegurará la integridad de la información durante múltiples actualizaciones concurrentes.

Si un sistema VaxCluster falla, los usuarios pueden registrarse en otro sistema Vax Cluster para crear un nuevo proceso y mantenerlo trabajando.

La característica básica de un sistema Vaxcluster para que aumente la disponibilidad son:

- Compartir trabajos en ejecución por lotes y colas de impresión que son accesadas por cualquier UCP.

- Capacidad de acceso a todos los discos para todas las UCP´s.

- Información de los procesos, el control para todas las aplicaciones y utilerías del sistema.

- Configuración Automática asistido por un procedimiento de comandos que sirven para agregar o eliminar UCP's del sistema o modificar la configuración.

- La capacidad para soportar múltiples discos de arranque, de tal modo que si uno falla, el acceso a los discos satélites es posible.

- La alta disponibilidad puede ser proporcionada por cualquier sistema localizado en un solo sitio y enviarlo a un sistema Vaxcluster de tolerancia a desastre basados en una configuración FDDI.

VII.3. SERVICIOS DE RECUPERACION.

Hay un nivel de falla que no puede ser engañado para atender a un solo sistema. Esto es, cuando un evento catastrófico ocurre que interrumpe completamente la operación. Para solucionar este problema, digital ofrece un rango de servicio de recuperación que hace posible mantener los servicios de aplicación y continúe su operación sin ninguna interrupción en el sistema.

Los servicios de recuperación incluyen:

- Total planeación de recuperación del sistema, un paquete de software que provee una completa guía y soporte en la creación de un plan de recuperación de desastre.

- Servicio de rEstauración, bajo la cual se puede proporcionar un centro de respaldo, junto con el soporte del personal para diagnosticar y recuperar las operaciones.

El servicio de recuperación, el cual es una opción para cubrir los daños causados por accidente. Estos servicios de recuperación varían de una ciudad a otra.

VII.4. CONEXION DE LOS SISTEMAS VAXCLUSTER

Los sistemas Vaxcluster puede conectarse de cuatro formas diferentes y además formar un solo sistema con la unión de Estas conexiones:

DSSI (INTERCONEXION DEL SISTEMA DE ALMACENAMINETO DIGITAL)
NI (INTERCONEXION DE RED)
CI (INTERCONEXION A UNA COMPUTADORA)
FDDI (INTERFACE DE DATOS DISTRIBUIDOS POR FIBRA OPTICA)

VII.4.1. DSSI (INTERCONEXION DEL SISTEMA DE ALMACENAMIENTO DIGITAL)

Consiste en la conexión de dos computadoras Vaxcluster conectadas por el DSSI.

Los sistemas DSSI es un nuevo producto de almacenamiento de datos, diseñados especialmente para mayor almacenamiento funcional. La automática recuperación en el evento de un funcionamiento incorrecto es la característica importante de los DSSI.

VII.4.2. NI (INTERCONEXION DE RED)

Consiste de dos computadoras VaxCluster conectadas en una configuración de una red Ethernet.

VII.4.3. CI (INTERCONEXION A UNA COMPUTADORA)

Consiste de dos o mas computadoras Vaxcluster conectadas en una configuración estrella, una conexión de alta velocidad,

VII.4.4. FDDI (INTERFACE DE DATOS DISTRIBUIDOS POR FIBRA OPTICA).

Consiste de dos o mas computadoras conectadas por una configuración FDDI. Esto permite tener una conexión remota a una gran velocidad en el flujo de los datos.

VII.4.5. LOS SISTEMAS DSSI Y NI

Dos o mas procesadores son necesarios si una instalación desea continuar procesando mientras el otro Esta fuera de línea, donde la mayor preocupación es el período de interrupción de una aplicación.

Los sistemas VaxCluster DSSI tratan de proporcionar un alto nivel de disponibilidad, con un período potencial de transición de unos 20 o 60 segundos dependiendo de la aplicación.

Este nivel de disponibilidad es requerido para un gran número de diferentes aplicaciones y ambientes.

VII.4.5.1. ALTA DISPONIBILIDAD:

En un servidor de base de datos, donde el acceso a la información es crucial, para los sistemas DSSI Vaxcluster combinados con el sistema

operativo VMS proporcionara el acceso compartido y de recuperación a disco por medio de líneas diferentes de bus redundantes.

En un sistema de tiempo compartido hay soporte para varios usuarios, estos pueden compartir los recursos del sistema. Los usuarios pueden registrarse inmediatamente en un segundo sistema, reduciendo el tiempo de caída y teniendo acceso a los mismos datos o proceso antes de que se presentara la interrupción de la falla.

Una amplia cobertura en los sistemas del procesamiento de transacciones, proporciona una capacidad de mantener la ejecución y el acceso a los datos.

UN SERVIDOR DE ARCHIVO PARA VMS.

Si la aplicación se Esta ejecutándo en una Estación de trabajo especifica, el sistema DSSI Vaxcluster puede proporcionar la recuperación automática y la máxima disponibilidad en los datos.

La configuración de los sistemas DSSI Vaxcluster permite a ambas computadoras compartir todos los dispositivos de almacenamiento, proporcionando una conexión directa para el acceso a datos simultáneos.

Un sistema puede ser llevado a estar Fuera de Servicio para realizar el mantenimiento, dirigiéndose los procesos a otro sistema que continúe con

la ejecución de los servicios que se estaban proporcionado al proceso que fue interrumpido por cualquier circunstancia. Cuando solo una aplicación se esta prpcesando, el rendimiento de los dos procesadores del sistema DSSI Vaxcluster se manifestará con una degradación.

Cada DSSI Vaxcluster incluye dos procesadores con elementos de almacenamiento integrados compartidos (EAIC), un bus de almacenamiento DSSI, el sistema operativo VMS y software para el sistema.

En los sistemas NI Vax cluster es utilizado en pequeños sistemas. Este tipo de sistema no requiere de los controladores jerárquicos.

A semejanza de los sistemas CI Vaxcluster ,es utilizado la versión NI que puede recuperar rápidamente y restaurar la integridad de la base de datos en el caso de que suceda una falla en el procesador.

Con los discos espejos, también puede tolerarse la falla de los controladores de disco. A diferencia de los sistemas CI Vax cluster, su interconexión no es redundante.

VII.4.6. LOS SISTEMAS CI Y FDDI

Estos sistemas (CI Y FDDI) proporcionan recuperación automática de los nodos de procesamiento en configuraciones medianas y superiores. El

intervalo de tiempo de la recuperación es unos pocos segundos a unos cuantos minutos dependiendo de los factores, tales como el software del sistema en uso, la programación de la aplicación, etc.

La función básica de los sistemas CI Vaxcluster son:

- Disposición de las aplicaciones, al mismo o al nivel superior de disponibilidad tales como los sistemas DSSI Vaxcluster, con mucho mayor rendimiento y capacidad.

- Enlaces a un amplio rango de las Vax, que proporcionan la capacidad para distribuirlos a una mayor área.

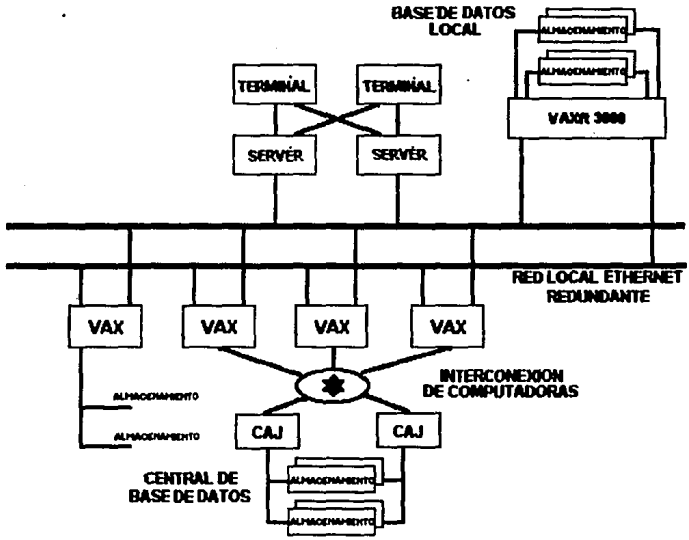

FIGURA VII-1. SISTEMA VAXCLUSTER CONFIGURADO CON UN SISTEMA VAXFT 3000.

VII.5. CONFIGURACION DE LOS SISTEMAS DE ALTA DISPONIBILIDAD

Muchas opciones de configuraciones son hechas posibles por la flexibilidad de la Arquitectura Vax y el amplio rango de disponibilidad que esta proporciona.

Diferentes sistemas y software pueden ser utilizados en una variedad de formas que proporciona exactamente el nivel de disponibilidad que se necesita para la aplicación del usuario

En una configuración muy clara observamos una serie de computadoras Vax ft 3000 tolerante a fallas soportando terminales. Aquí podemos observar que todas las partes del sistema Estan duplicadas (discos de almacenamiento, servidores de terminales, el bus de comunicación Ethernet y los enlaces de comunicación entre ellos).

Tal configuración es ideal para situaciones en la cual las computadoras Vax ft 3000 ejecutará una aplicación dedicada tal como el sistema de despacho de emergencia del 911 (E.U).

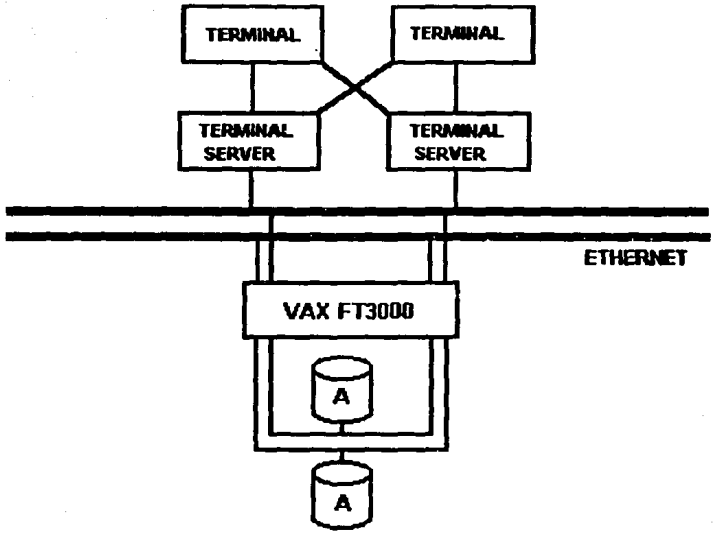

FIGURA VII-2. SISTEMA TOLERANTE A FALLAS DE LA VAX FT3000.

La siguiente configuración es mas compleja y podría ser utilizada para aplicaciones tales como el intercambio de paquetes de información. Aquí los sistemas Vax ft 3000 Estan ejecutando su propia parte de la aplicación, una parte del sistema que se encargará de la captura de los datos, las comunicaciones, funciones de validación. También se encargará de los recursos y administrar la base de datos, a Esta parte se le conoce con el nombre de transacciones en línea (front end).

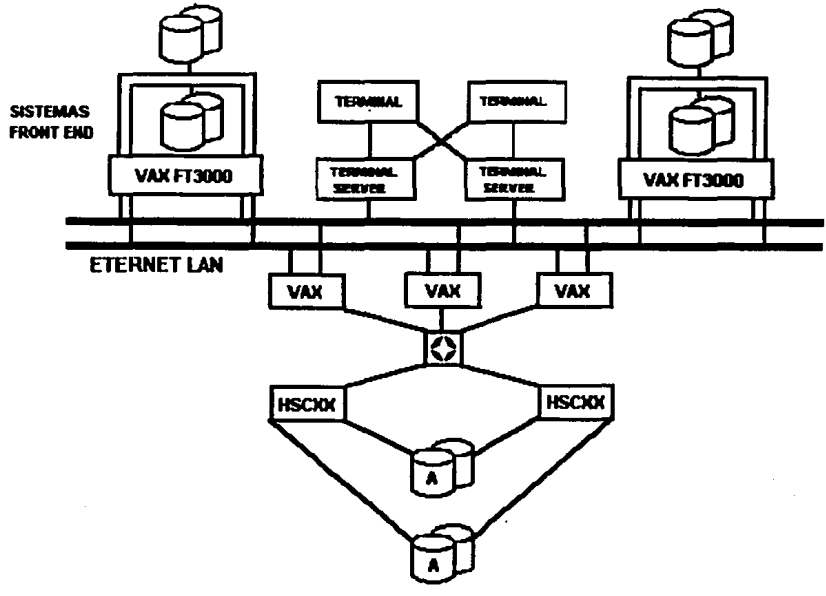

FIGURA VII-3. LOS SISTEMAS VAXFT3000 PARA CAPTURA DE DATOS EN UNA APLICACION DE PROCESAMIENTO DE TRANSACCIONES CENTRALIZADA.

Por ejemplo los sistemas Vax ft 300 quizás manejen datos de entrada en línea que son capturados y ejecutados para peticiones de facturación, mientras los sistemas podrían llevar su administración de la base de datos en la otra mitad de la red.

En la misma configuración, el sistema Vaxcluster incluye tres computadoras, de la cual una es de sustitución. Si Esta computadora se mantiene completamente sin procesamiento de la información (excepto por el sistema

operativo), entonces después de una recuperación, el rendimiento de los sistemas Vaxcluster totalmente será el mismo antes de que sucediera la falla.

La siguiente configuración es ideal para bancos y otros tipos de empresas financieras, donde la falla del procesamiento podrá dar como resultado grandes pérdidas.

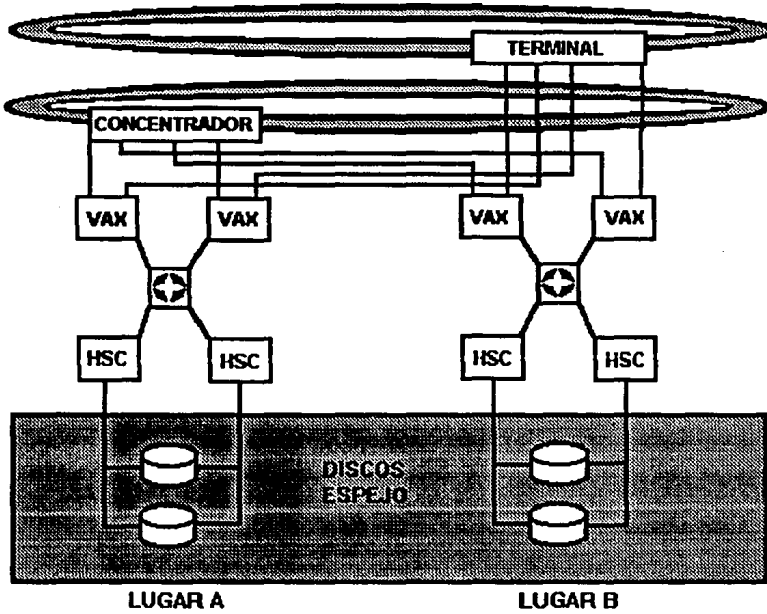

FIGURA VII-4. SISTEMAS VAXCLUSTER CON MULTIPLES ANILLOS FDDI.

La configuración consiste de un sistema Vax cluster distribuido en dos diferentes localizaciones conectadas por múltiples anillos FDDI.

Los múltiples canales dentro y fuera de la configuración CI permite a los controladores de almacenamiento Jerárquico (CAJ) a ser conectadas en redundancia, protegiendo el acceso a los datos en el evento de la ruptura en su conexión, una vez mas hay una duplicación del bus " ETHERNET ".

La configuración siguiente es físicamente la misma a la de los sistemas Stand-lone Vax ft 3000, excepto que encuentra conectado en estrella y con líneas síncronas a un sistema Vax cluster. Este tipo de configuración es utilizado en aplicaciones de cajeros automáticos o aplicaciones similares.

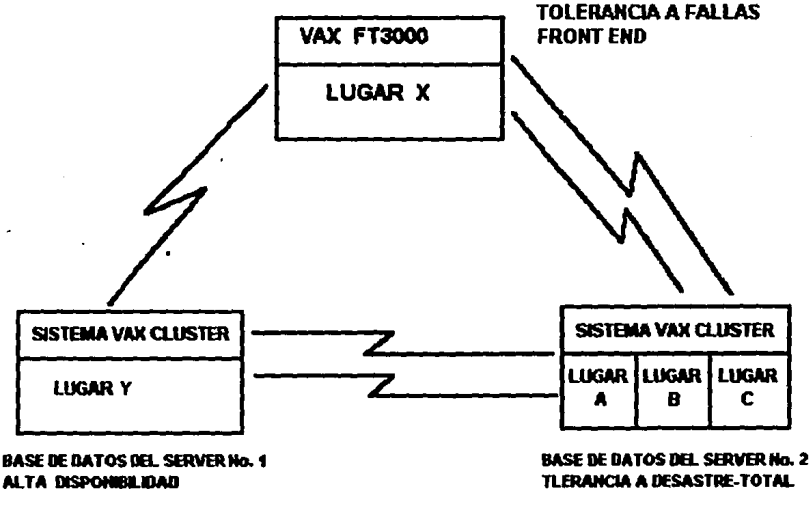

FIGURA VII-5. UN SISTEMA VAXFT3000 ENTRE DOS SISTEMAS VAXCLUSTER.

Con esta configuración, empresas filiales podrían utilizar un sistema Vax ft 3000 y terminales con cargas específicas, mientras la matríz utilice Sistemas VaxCluster que ejecute tareas complementarias.

Por ejemplo, las filiales quizás lleven la administración de formas, la validación de los datos, colas de transacciones, aplicaciones locales, etc. Con el sistema Vax ft 3000 se administran las referencias locales tales como la validación y administración de la Base de Datos. Mientras el sistema VaxCluster quizás se soporte mayor incorporación del procesamiento de transacciones y de la Base de Datos.

En este sentido, el rendimiento, la capacidad de almacenamiento, la disponibilidad están todos distribuídos en toda la organización siendo el costo mas efectivo y mas representativo en cualquier aplicación crítica.

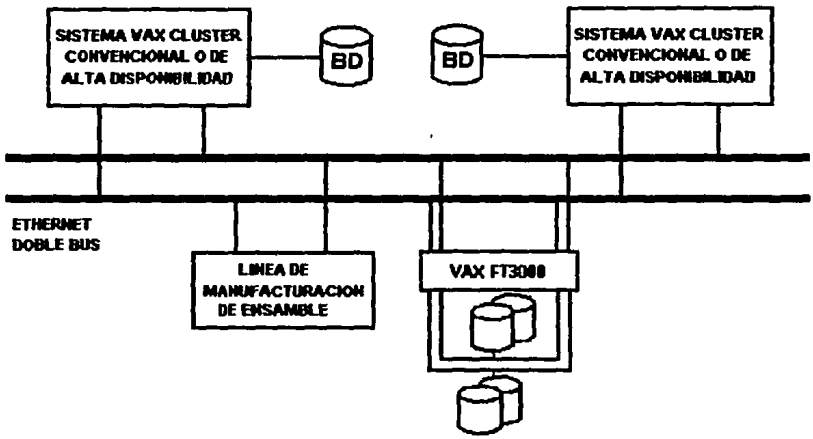

FIGURA VII-6. SISTEMA DISTRIBUIDO.

CAPITULO VIII

APLICACIONES

VIII. APLICACIONES.

Hoy en día, la información que manejan los sistemas de computación se hacen mas prescindibles para el desarrollo de las actividades del ser humano; teniendo un gran impacto en el soporte de aplicaciones de misión crítica, servicios operacionales y aplicaciones de soporte administrativo

VIII.1. SOPORTE DE APLICACIONES DE MISION CRÍTICA.

Se conoce como aplicaciones de misión crítica a la captura de información a la transferencia de fondos, monitoreo de procesos de tiempo real, etc. Esto es, a todas aquellas aplicaciones que son realizadas en línea.

VIII.2. SERVICIOS OPERACIONALES.

En este punto se puede hablar de los servicios de laboratorio e ingeniería, administración de personal, bancaria, etc.

VIII.3. APLICACIONES DE SOPORTE ADMINISTRATIVO.

Se puede hablar de correo electrónico, procesadores de palabras, sistemas de toma de decisiones, etc.

Los tres aspectos anteriores producirán un impacto negativo si su sistema de cómputo no trabaja con un cierto nivel de disponibilidad, además si ocurriese algún tipo de falla en el sistema, se ocasionarian molestias o insatisfacciones del cliente, desventaja ante la competencia, pérdida de clientes, pérdida de oportunidades, altos costos de tiempo perdido, etc.

Los sistemas de cómputo requieren un cierto nivel de disponibilidad dependiendo de su actividad para que pueden realizar su función asegurando la integridad de sus datos y por otro lado la de sus procesos.

VIII.4. AREAS DONDE SE APLICAN LOS STF.

Existen diez áreas en donde se pueden utilizar los STF realizando diferentes actividades.

A continuación describiremos cada una de ellas:

VIII.4.1. FINANCIERA.

BANCOS: Para el manejo de cajeros automáticos, transferencia de fondos y en general todas las aplicaciones de misión crítica.
BOLSA DE VALORES: Para asegurar que no haya pérdidas financieras, pérdidas de oportunidades de compra venta de acciones.

CAJEROS AUTOMATICOS: Mantener las transacciones en línea que estén operando.

VIII.4.2. GOBIERNO

GOBIERNO CENTRAL

SISTEMAS DE DESPACHO DE EMERGENCIA (911 en E.U.A.)

CAPACITACION DE IMPUESTOS

MISCELANEOS

LOTERIA: Registro confiable de jugadores y reporte oportuno de los resultados

VIII.4.3. MILITAR

SISTEMAS DE COMUNICACION SOFISTICADO: Se utiliza a la computadora para transmitir datos codificados y decodificar datos recibidos.

COMUNICACION PARA EL DIRECCIONAMIENTO DE PROYECTILES O NAVES ESPACIALES

SEGURIDAD NACIONAL

METEREOLOGIA: Monitoreo del medio ambiente por medio de sensores conectados a la computadora.

TELEMETRIA: Requiriendo también un cierto grado de disponibilidad e integridad de sus datos.

VIII.4.4. SALUD

HOSPITALES: Se puede utilizar un control de admisión, transferencia y alta de pacientes, servicios de emergencia, monitoreo de pacientes, registros médicos, para la administración de cierta droga a un paciente, monitoreo de sus sistemas vitales, etc.

FARMACIA: Registro de clientes, inventario de medicamentos, punto de venta, manejo de medicamentos controlados, etc.

VIII.4.5. MANUFACTURACION

CONTROL DE PISO DE TALLER

CONTROL DE PROCESOS

MONITOREO DE PROCESOS: Algunos procesos que deben correr sin que ocurra ningun tipo de suspensión del sistema, como en el caso del manejo de las barras de control de un reactor nuclear.

CONTROL DE CALIDAD

MANEJO DEL INVENTARIO

ROBOTICA

VIII.4.6. SISTEMAS DE RADIO Y TELEVISION

MANEJO AUTOMATICO DE OPERACIONES PARA LA TRANSMISION DE LA RADIO Y TELEVISION

EVITAR LAS CAIDAS DE TRANSMISION

MONITOREO DE POSICIONES DE SATELITE

CODIFICACION Y DECODIFICACION DE SEÑALES VIA SATELITE

TRANSMISION DE SEÑALES VIA SATELITE

VIII.4.7. DISTRIBUCION DE MERCADO

TERMINALES DE PUNTO DE VENTA: Almacénes de autoservicio, farmacias, gasolineras, etc.

CONTROL DE INVENTARIO

DETERMINACION DE SISTEMAS DE DEMANDA DE PRODUCCION

CREDITOS: Verificación de créditos.

VIII.4.8. TRANSPORTACION

CONTROL AEREO: Debe tomarse en cuenta que el sistema deberá permanecer en servicio, aún en condiciones de ambiente desagradable y altas condiciones de tráfico aéreo.

RESERVACIONES: En hoteles, líneas aéreas, automóviles, etc.

CONDUCCION DE LIQUIDOS: Se puede utilizar en el control y monitoreo de conductos de gas, petróleo o agua para tener una seguridad y eficiencia.

TRANSPORTES DE MERCANCIA: Se utiliza para el control de rutas desde su planeación hasta su monitoreo, por medio de señales de radio

comunicaciones vía satélite, se puede tener información en tiempo real (en línea), para determinar el balanceo de las cargas, la eficiencia en el transporte, la disponibilidad de los camiones de carga y además de brindar seguridad a sus choferes al momento de determinar la ubicación del camión y determinar con precisión la hora de la llegada del embarque.

VIII.4.9. TELECOMUNICACIONES

REGISTRO DE DATOS POR TELEFONO: Como en el caso de la transmisión de datos por modems y otras señales enviadas por líneas telefónicas.

RED DE SERVICIO 911 EN LOS E.U

TELEFONIA

APLICACIONES DE VALIDACION DE TARJETAS DE CREDITO: Se utiliza como un sistema " front-end", como una captura de información.

CONCLUSIONES

CONCLUSIONES.

Para que un sistema se considere confiable, no únicamente basta con el buen diseño de la arquitectura y el control de calidad de los elementos que componen a una computadora, sino que también es necesario contemplar técnicas de tolerancia a fallas que permitan que el sistema se mantenga en operación ante la presencia de una falla.

Dependiendo de la aplicación de que se trate, se tendrán que contemplar el nivel de tolerancia apropiado, implementado los recursos necesarios para tal fin.

Los diseñadores de los Sistemas Tolerante a Fallas de las computadoras aquí estudiadas, contemplan Sistemas Abiertos que pueden enlazarse a cualquier tipo de red y funcionar como un respaldo de la información de las computadoras restantes, para que en el momento de que ocurra algún evento de desastre, tal como incendio, terremoto, guerra, etc., la información se mantenga íntegra.

Para los Sistemas Tolerante a Fallas la duplicación es la parte esencial, contemplándose cuatro tipos de duplicación, las cuales son:

Duplicación de la información
Duplicación del Hardware
Duplicación del Software

Duplicación en el Tiempo

La duplicación de la información para las tres configuraciones estudiadas, se encontró que estas computadoras manejan las mismas técnicas para la corrección y detección de errores, tales como el bit de paridad, el cheksum y el código hamming. Estas técnicas de detección y corrección de error son utilizadas para verificar que la información transmitida entre dos módulos sea la correcta o de lo contrario sea corregido el error de código encontrado.

En la duplicación de hardware se observó que la diferencia entre las 3 configuraciones resaltaba entre el procesador y la memoria, con respeto a la entrada y salida únicamente se observó que se tenían módulos o bus duplicados.

En la configuración de Uniprocesador se encontró que está compuesto por tres procesadores físicos y dos módulos de memoria, estos módulos son controlados y verificados por un seleccionador y son autoverificables para detectar su correcto funcionamiento, la diferencia que se observó de esta configuración es que sus técnicas de tolerancia a fallas son realizadas mediante el sistema operativo. Mientras que en la configuración de multicomputadoras, se cuentan con dos computadoras idénticas (llamadas zonas) que se encuentran enlazadas por un canal de comunicación mediante la cual se realiza la sincronización de ambas zonas, la ventaja observada de esta configuración es que las técnicas para soportar las fallas

son más sencillas y por lo mismo el software y el hardware utilizados para este fin es menos complejo. Finalmente en la configuración del sistema de multiprocesador, se observó que la manifestación de una falla en la memoria principal, ocasionaría la suspensión del sistema, ya que únicamente esta configuración cuenta con un módulo de memoria que es compartido por todos los procesadores.

La ventaja encontrada para esta configuración es la distribución de la carga de trabajo entre los procesadores en caso de que alguno de éstos fallara; esta es la característica principal de este sistema.

La duplicación del software en la configuración del uniprocesador, aparte de las técnicas de tolerancias a fallas soportadas por el software, se encontró que se hicieron mejoras al kernel del sistema operativo, para soportar las condiciones de pánico que en el unix estándar se presentan; en la configuración de multicomputadoras la duplicación del software se contempla únicamente en la reinicialización de los procesos, y en la configuración de multiprocesadores, el software es el responsable de controlar a todos los recursos incluyendo a los accesos de la memoria compartida y distribución de carga de trabajo de los procesadores .

La redundancia en el tiempo para las configuraciones de Uniprocesador, Multicomputadoras y Multiprocesadores, es realizado por la técnica del reloj guardián, teniendo la función que se ejecute una instrucción en un intervalo de tiempo determinado.

BIBLIOGRAFIA

BIBLIOGRAFIA

LIBROS

Digital Equipment Corporation
Digital's Guide to High-Availability and Fault Tolerance Computing
1990.

Tandem Computers
Introduction to the Integrity S2 System
Segunda edición
Número de parte 52368
Diciembre de 1990.

Barry W. Jhonson
Design and Analisys of Faul Toleran Digital Systems
University of Of Virginia
Wesley Publishing Company.

T. Anderson , P.A. Lee
Fault-Tolerance Principles and Practice
New Jersey
Prentice Hall, 1981.

ARTICULOS DEL IEEE

Algirdas Avizienis
Fault-Tolerance: The Survival Attribute of Digital Systems
Octubre de 1978.

B. Randell, P. A. Lee and P. C. Treleaven,
Reliability Issues in Computing System Desing
University of New Castle Upon Tyne.
Junio de 1978.

Daniel P. Siewiorek,
Fault-Tolerance in Comercial Computers.
Carnegie-Mellon University
Julio de 1990.

Daniel P. Siewiorek,
Architecture of Fault-Tolerant Computers.
Carnegie-Mellon University
Agosto de 1984.

Jean-Claude Laprie,
On Computer System Dependability: Faults, Errors and Failures
Laboratoire D'automatique et D'analyse Des Systemes Du CNRS, Toulouse, France, 1985.

P. A. Bernstein
"Sequioa" A Fault -Tolerant Tightly Coupled Multuprocessor for Transaction Procesing"
Computer, Vol 21, No.2, Feb. 1988, pp 37-45.

Ravishankar K. Iyer and Paola Velardi
Hardware-Related Sofware Erros: Measurement and Analysis
Febrero de 1985.

Victor P. Nelson,
Fault-Tolerant Computing: Fundamental Concepts
Auburn University
Julio de 1990.

FOLLETOS

Digital Equipement Corporation
Guide to Vax Cluster Systems
Enero de 1991.

Digital Equipement Corporation
Digital´s Gude to High-Availability and Fault-Tolerant Computing
1991.

Tandem Ccomputers
Integrity System Family
Fault-Tolerance, High-Performance Unix Systems
Número de clasificación 101303-001
1991.

Tandem Computers.
NonStop-UX Operating System For Integrity Systems, a Highly Reliable Implementation of Unix System V.
Número de clasificación 101302-001
1991.

Tandem Computers
Integrity S2 from Tandem
Reliability, High Performance and Portability.
Número de clasificación 101306
1989.