


RESUMEN de la tesis de Alejandro Hinojosa Corona presentada como requisito parcial para la obtención del grado de MAESTRO EN CIENCIAS DE LA COMPUTACION. Ensenada, Baja California, México, Enero de 1988.

SISTEMA DE CAPTURA Y DETECCION DE EVENTOS SISMICOS PARA LA RED SISMOLOGICA RESNOR, BASADO EN UNA COMPUTADORA PERSONAL IBM-PC.

Resumen aprobado:



M. en C. Carlos R. Duarte Muñoz
Director de Tesis

El problema a resolver con la presente tesis consistió en transportar el sistema de captura y detección de eventos sísmicos de la Red Sismológica del Norte de Baja California (RESNOR) que gira alrededor de una minicomputadora PRIME 350, a una microcomputadora personal IBM PC.

Se describe el desarrollo del nuevo sistema de captura y detección de datos sísmicos, capaz de detectar en tiempo real aquellas señales que, provenientes de la red telemétrica digital, se identifiquen como eventos sísmicos para almacenarlos en forma apropiada y efectuar su posterior procesamiento.

Se explican las razones por las cuales se decidió desarrollar un sistema equivalente al que opera en la actualidad. Se describen el equipo digital, programación de apoyo y la integración de éstas dos herramientas para lograr el objetivo.

Se definen las especificaciones del problema (entrada y salida), se plantea el diseño y finalmente su implementación con sus respectivos resultados.

CENTRO DE INVESTIGACION CIENTIFICA Y DE

EDUCACION SUPERIOR DE ENSENADA

DEPARTAMENTO DE CENTRO DE COMPUTO ELECTRONICO

SISTEMA DE CAPTURA Y DETECCION DE EVENTOS SISMICOS

PARA LA RED SISMOLOGICA RESNOR,

BASADO EN UNA COMPUTADORA

PERSONAL IBM-PC

TESIS

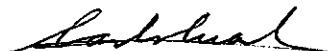
que para cubrir parcialmente los requisitos necesarios para

obtener el grado de MAESTRO EN CIENCIAS presenta:

ALEJANDRO HINOJOSA CORONA

Ensenada, B.C.N., Enero de 1988.

TESIS APROBADA PARA SU DEFENSA POR:



M.C. Carlos Roberto Duarte Muñoz, Director del Comité



M.C. Fernando Favela Vara, Miembro del Comité



M.C. Hugo Homero Hidalgo Silva, Miembro del Comité



M.C. Francisco Suárez Vidal, Miembro del Comité



Ing. Salvador Castañeda Avila, Jefe de Centro de Cómputo Electrónico



M.C. Cuauhtémoc Nava Button, Director Académico

Tesis presentada en Febrero 3, 1988.

DEDICATORIA

A mi esposa Martha y a mis padres Rosalba y Jose Antonio.

AGRADECIMIENTOS

Al Centro de Investigación Científica y Educación Superior de Ensenada por haberme formado profesionalmente con su programa de postgrado y con la experiencia adquirida en la participación de proyectos de investigación de la División de Ciencias de la Tierra.

Al centro de cómputo del CICESE por facilitar su equipo para el desarrollo de la presente tesis.

A mi comité de tesis y al director del centro de cómputo por su firme apoyo: M. en C. Francisco Suarez V., M. en C. Carlos Duarte M., M. en C. Fernando Favela V., M. en C. Hugo Hidalgo e Ing. Salvador Castañeda.

CONTENIDO

1	INTRODUCCION	1
2	TARJETA COPROCESADOR PC68K	6
2.1	Introducción	6
2.2	Circuitos Integrados principales que componen PC68K	7
2.2.1	VT2130 Memoria de Acceso Aleatorio de Doble Puerto (MDP)	8
2.2.2	Puerto Serie MK68564	10
2.2.3	MC68681 Puertos Serie	12
2.2.4	MC6821 Puertos Paralelos	14
2.2.5	DS1216 Reloj en Tiempo Real	15
2.2.6	Interfaz con la computadora IBM - PC (PCI)	18
3	SISTEMA OPERATIVO OS-9/68000	19
3.1	EL NUCLEO	21
3.1.1	Administración de Memoria	21
3.1.2	Llamadas al Sistema	30
3.1.3	Administración del Procesador y Ejecución de Procesos	34
3.2	Sistema de Entrada/Salida E/S	44
3.2.1	Administradores de Archivos	46
3.2.2	Módulos Manejadores de Dispositivos de E/S	48
4	OS-9 y PC68K	58
5	DISEÑO DEL SISTEMA DE CAPTURA	64
5.1	Especificaciones	64
5.1.1	Objetivo	64
5.1.2	Entrada	64
5.1.3	Salida	66
5.2	Algoritmo de Detección	67
5.3	Procesos Principales	70
5.3.1	Módulo Adquisición y Reformatización	71
5.3.2	Módulo de Estampado de Marcas de Tiempo	74
5.3.3	Módulo de Detección-Grabación	76
5.4	Estructuras de datos de Apoyo	87
5.4.1	ESTADO de Sistema	87
5.4.2	Colas de DATOS	88
5.4.3	Colas de PROMEDIOS CORTOS	92
5.4.4	Tabla de PROMEDIOS LARGOS	94
5.4.5	FUNCION DE DETECCION	95
5.4.6	Cola de RESULTADOS	97

CONTINUACION DE CONTENIDO

6 IMPLEMENTACION	100
6.1 Módulo de DATOS Compartido	100
6.2 Manjedores de Dispositivos	103
6.3 Manejador MA68681. Adquisición-Reformatización	106
6.4 Manejador RE68681. Estampado de Marcas de Tiempo	108
6.5 Módulo de Detección-Grabación	109
6.6 Concurrencia al Módulo de DATOS	111
7 RESULTADOS	113
8 CONCLUSIONES y RECOMENDACIONES	116
9 LITERATURA CITADA	118

LISTA DE FIGURAS

Figura	página
1. Asignación de las señales en el conector de 40 patas que se encuentra en la parte posterior de la tarjeta de expansión para los 6 puertos serie de los 3 dispositivos DUART MC68681.	12
2. Asignación de las señales del adaptador de interface paralelo (PIA) MC6821, al conector de 40 patas que se encuentra en la parte posterior de la tarjeta PC68K.	14
3. Registros del reloj en tiempo real DS1216, valor de sus rangos en Binario Codificado a Decimal (BCD) y orden de acceso: del bit 0 del registro 0 al bit 7 del registro 7.	17
4. Formato de Módulo Básico de Memoria	24
5. Mapa de Memoria Típico para OS-9/68000	27
6. Mapa de memoria inicial y contenido de registros de un nuevo proceso.	38
7. Formato del Encabezado de un Módulo Manejador de Dispositivo.	50
8. Campos Estandar Adicionales al Encabezado de Módulos Descriptores de Dispositivo.	54
9. Definiciones Universales para Descriptores de Trayectorias.	56
10. Parámetros del algoritmo de detección para una estación. Se muestra la ventana corrida de promedios cortos y la de resultados de la comparación de pcorto/plargo vs. umbral.	80
11. Evento sísmico reportado en 4 de las 5 estaciones de la gráfica. En la parte superior de la figura se muestra la ventana corrida de resultados y el valor acumulativo (resumen) de la ventana.	83
12. Organización de las varias estructuras de datos que se encuentran en el módulo de DATOS que es accesado en forma concurrente por los tres procesos: detección/grabación, adquisición/reformatización y estampado de marcas de tiempo.	102

LISTA DE TABLAS

Tabla	página
1. Distribución de las señales de los puertos serie en los conectores de 9 patas (DB9) que se encuentran en la parte posterior de la tarjeta PC68K. Corresponden a los dispositivos MK68564.	10
2. Mapa de memoria de los registros de los puertos serie MK68564 en tarjeta PC68K.	11
3. Mapa de los registros de los tres dispositivos DUART MC68681 (puertos serie) en memoria de PC68K.	13
4. Relación entre el número del vector de excepción del 68000, mnemónico de la llamada a OS-9 asociada y la función del vector.	45

SISTEMA DE CAPTURA Y DETECCION DE EVENTOS SISMICOS PARA LA RED SISMOLOGICA
RESNOR, BASADO EN UNA COMPUTADORA PERSONAL IBM-PC.

INTRODUCCION

RESNOR (Red Sismológica del Norte de Baja California) es una red de telemetría digital desarrollada por la División de Ciencias de la Tierra del CICESE (Centro de Investigación Científica y Educación Superior de Ensenada) que se emplea para capturar datos sísmicos en la región Norte de Baja California (Valerdi J. 1978) (Medina M. et al. 1979) (Medina et. al 1984).

El sistema monitorea actividad sísmica a través de una red de estaciones telemétricas distribuidas en la región norte de Baja California entre los paralelos 31 y 32-45 N y los meridianos 114-30 y 117 W.

Actualmente, (invierno de 1987) la red está constituida por 8 estaciones en operación que transmiten información sísmica continuamente hacia las instalaciones del CICESE, en la ciudad de Ensenada B. C. a través de un sistema de comunicación por radio desarrollado específicamente para tal fin. Los datos que se reciben en el CICESE se alimentan a una minicomputadora digital modelo PRIME 350, que se encarga de decidir si contienen información de interés y almacenarlos para su posterior procesamiento.

Desde el punto de vista de operación, el sistema puede dividirse en 3

subsistemas principales: estaciones sísmicas, sistema de comunicaciones y sistema de captura y procesamiento de datos. El subsistema de estudio en la presente tesis es el tercero.

El objetivo del sistema de captura y procesamiento de datos es detectar en tiempo real aquellas señales que, provenientes de la red telemétrica digital, se identifiquen como eventos sísmicos para almacenarlos en forma apropiada y efectuar su procesamiento.

El sistema de captura y detección de eventos sísmicos de la red RESNOR actualmente está en operación y tiene capacidad para 25 estaciones. Este sistema se empezó a desarrollar en 1980 y se ha madurado a través del tiempo. Esto ha permitido adquirir experiencia en sistemas de captura en tiempo real y en los sistemas PRIME. El sistema opera en forma automática brindando reportes casi instantáneos de la información sísmica que llega al CICESE en tiempo real. (Hinojosa A. et al 1982).

Desde 1980 se ha venido acumulando un gran banco de información sísmológica de la región Norte de Baja California. El desarrollo de la red en su conjunto (los 3 subsistemas) y los datos generados por ella han producido un gran número de reportes técnicos, publicaciones científicas, presentaciones en congresos y tesis de maestría. Existen más de 30 trabajos que se han generado alrededor de RESNOR. Las razones por las cuales se decidió trasladar el sistema de captura actual a uno que estuviera implementado alrededor de un computadora personal son las siguientes:

- Menor costo de recuperación en caso de descompostura del sistema o pérdida total.
- Poder utilizar el equipo PRIME 350 para propósitos generales al igual que los otros equipos PRIME con que cuenta el CICESE (750, 400, 2350).
- Poder operar en un ambiente sin tantas restricciones de humedad y temperatura, como en el que opera el sistema actual.
- Poder contar con un sistema de captura portátil que permitiera instalar redes de telemetría con mayor flexibilidad en regiones de interés.

A través de la historia del sistema de captura actual, se han desarrollado varios programas de utilerías que permiten entre otras cosas: monitorear el estado del sistema (Hinojosa A. et al 1982), cálculo de magnitudes de los eventos registrados (Vidal A. et al 1987), despliegue gráfico e interactivo para la depuración y manejo de la información capturada (Duarte C. et al 1981), generación de boletines mensuales, procesado automático de la información (Hinojosa A. et al 1983), etc.

La presente tesis se limita a la adquisición, detección y almacenamiento apropiado de la información sísmica transmitida por las estaciones de RESNOR. Esto no quiere decir que no se vayan a desarrollar las utilerías, sino que están fuera del contexto del presente trabajo y se implementarán en el nuevo ambiente en un futuro inmediato.

Como se estableció en el proyecto de tesis, el primer paso en la solución del problema fue buscar en el mercado alguna tarjeta que tuviera ciertas características; entre éstas estaban:

- Que pudiera operar en el ambiente de la computadora personal IBM PC/XT/AT a través de su ducto.
- Existencia de varios puertos serie. Entre 8 y 16.
- Cierta capacidad de procesamiento autónomo con ayuda de un coprocesador.
- 64 Kbytes de memoria de acceso dinámico (RAM) o más de ser posible.

Se encontró una tarjeta que cumplía con los requisitos especificados. Esta es la tarjeta coprocesador PC68K fabricada por la empresa Emerald Computer Inc. de Portland, Oregon, E.U.A. En la sección de descripción de equipo, se desglosan las características de esta tarjeta. La decisión de comprar la electrónica implicó que el trabajo de esta tesis haya sido enfocado a la programación de la electrónica seleccionada para la aplicación.

Esta tarjeta viene acompañada de un sistema operativo, el OS-9 desarrollado por la empresa Microware Systems Corp. de Des Moines, Iowa, E.U.A. Este sistema operativo está orientado para aplicaciones en tiempo real. Es parecido al sistema UNIX y está implementado para la familia de microprocesadores 68000 de Motorola. Es un sistema operativo de multitareas/multiusuarios. Los aspectos concernientes a nuestra aplicación están descritos en la sección de sistema operativo OS-9.

La idea del diseño del sistema de captura, es que la tarjeta coprocesador realice las tareas de adquisición, detección y almacenamiento de la información detectada como evento sísmico y por otro lado la computadora IBM-PC se encarga de la depuración, edición y manejo de la información

sísmica capturada. Las dos computadoras, PC68K e IBM-PC, trabajan en forma independiente pero, a su vez, comparten recursos como disco duro, impresora, etc.

El presente trabajo está organizado de la siguiente manera. Primero se describe la mecánica (hardware) del equipo PC68K, después los aspectos del sistema operativo OS-9 concernientes a la aplicación a desarrollar. Una vez descritas las herramientas, se plantea el diseño de los procesos y estructuras de datos de apoyo. Finalmente la implementación del diseño y resultados obtenidos.

2 TARJETA COPROCESADOR PC68K

2.1 Introducción

La tarjeta coprocesador PC68K provee un mecanismo por el cual los procesadores de Motorola 68000 ó 68010 pueden ser utilizados en el ambiente de la microcomputadora IBM AT/XT/PC, permitiendo así que el usuario del 68000 comparta los recursos de la IBM (periféricos y programas).

La tarjeta PC68K junto con otra tarjeta de expansión, puede soportar hasta 9 usuarios (por medio de 8 puertos serie más uno en el monitor de la PC). Tiene un puerto paralelo que está alambrado para ser compatible con el estándar Centronics. Además contiene 2 Mbytes de memoria dinámica, un reloj en tiempo real y 2K bytes de memoria dinámica, estos dos últimos respaldados por baterías (no volátiles). También tiene 1K bytes de memoria dinámica de doble acceso, por la cual se comunican las dos microcomputadoras (IBM-PC y PC68K).

El coprocesador PC68K requiere de programación de soporte que permite a éste operar dentro de la microcomputadora huésped IBM-PC y compartir algunos de sus recursos. El fabricante (Emerald Computers Inc.) provee estos programas, uno de los cuales es la interfaz entre la PC y PC68K abreviada de aquí en adelante como PCI. Este programa es un archivo tipo .COM que al ser ejecutado es cargado en memoria y permanece residente. La función de este programa es permitir al usuario el cambio de control de IBM a PC68K y viceversa, dependiendo de la solicitud del usuario.

Dentro de la otra programación de soporte que se adquirió están: un compilador de lenguaje C, un ensamblador, un depurador y el sistema operativo OS-9 que contiene varias utilerías (editores, etc.) El sistema operativo OS-9 es un sistema de multiusuario, multitareas parecido a UNIX.

En el capítulo 3 se describen los aspectos más relevantes del sistema OS-9 que nos permitirán desarrollar nuestra aplicación: "Sistema de captura y detección de la red sísmológica RESNOR"

2.2 Circuitos Integrados principales que componen PC68K

En el apéndice de bibliografía viene las referencias a los manuales técnicos de los fabricantes de los circuitos integrados, por si' el lector tiene interés en investigar más sobre ellos. En los próximos párrafos sólo se hace mención a los aspectos importantes sobre el diseño de la tarjeta PC68K que le permiten coexistir con la IBM-PC y también se hace referencia a los puertos serie, paralelo y reloj en tiempo real. Este material se tradujo de partes del manual "68000 CoProcessor Board. Installation and Technical Reference" que acompaña a la tarjeta y es proporcionado por el fabricante (Emerald Computers, Inc.).

2.2.1 VT2130 Memoria de Acceso Aleatorio de Doble Puerto (MDP)

PC68K utiliza un dispositivo VT2130 que es una memoria de acceso aleatorio (RAM) con doble puerto. Este dispositivo es direccionado en forma concurrente por la computadora huésped IBM-PC o PC68K. Este dispositivo es por medio del cual ambos procesadores intercambian información. También provee el mecanismo de interrupción con el cual cualquier procesador puede interrumpir al otro. La computadora huésped IBM-PC es interrumpida vía línea IRQ 2, 3 o 4. Esta es seleccionable por medio de selectores en la tarjeta de PC68K. Se configuró para usar la línea IRQ2. La PC68K es interrumpida por la interrupción de nivel 4. Cualquier dispositivo o programa que se comunique entre los dos sistemas utiliza la MDP.

Como cualquier localidad de la MDP puede ser direccionada por cualquiera de los dos sistemas, es posible que ambos procesadores pudieran acceder la misma localidad de memoria al mismo tiempo. Si esto ocurre, PC68K tiene más alta prioridad ignorando así a la IBM.

La IBM direcciona la memoria MDP directamente. Las 1024 localidades (400 hex) son direccionadas de MDP:0000 a MDP:03FF. El segmento MDP es configurable por el usuario por medio de selectores en la tarjeta PC68K. Se asignó el segmento E000 (hex) a MDP. PC68K direcciona la MDP como un dispositivo de E/S utilizando sólo direcciones nones. PC68K direcciona la MDP de EC0001 a EC07FF. Para calcular la dirección de PC68K usar la

siguiente fórmula hexadecimal:

$$EC0001 + (\text{desplazamiento de MDP} * 2) = \text{Dirección de PC68K}$$

Los siguientes ejemplos son derivados de esta fórmula:

Dirección IBM-PC	Dirección PC68K
MDP:0000	EC0001
MDP:0001	EC0003
MDP:0100	EC0201
MDP:03FF	EC07FF

Un procesador puede generar una interrupción para el otro escribiendo a un byte en particular dentro de MDP. Específicamente, la PC68K escribe un byte en MDP:3FF para interrumpir a la PC, y la PC escribe en la dirección MDP:3FE para generar una interrupción a la PC68K. El procesador interrumpido lee el byte escrito para limpiar la interrupción. El dato escrito en estos bytes puede usarse como información para el procesador interrumpido para decodificar que tipo de servicio se está solicitando.

Existe una localidad en la MDP que no forma parte del segmento de MDP, pero, es utilizado como puerto de control por la PC68K. Esta dirección es llamada puerto de control (MDP:800), y es usada para reinicializar y causar una interrupción no enmascarable (NMI) a la PC68K. Modificar el bit en 1 de MDP:800 causará la siguiente acción:

MDP:800 bit 0 causa una NMI a PC68k

MDP:800 bit 1 causa una reinicialización de PC68K (reset)

2.2.2 Puerto Serie MK68564

PC68K usa el dispositivo MK68564 SIO (entrada/salida serie) de doble canal. Este es un dispositivo periférico multifunción capaz de manejar protocolos síncronos y asíncronos.

Los dos canales (A y B) son accedidos a través de 2 conectores de 9 patas (DB9) en la parte posterior de la tarjeta. La distribución de las señales en las patas de los conectores se muestra en la tabla 1.

Tabla 1. Distribución de las señales de los puertos serie en los conectores de 9 patas (DB9) que se encuentran en la parte posterior de la tarjeta PC68K. Corresponden a los dispositivos MK68564.

Pata	Señal	Dirección de PC68K
1	Transmite reloj	Salida
2	Transmite datos	Salida
3	Recibe Datos	Entrada
4	RTS	Salida
5	CTS	Entrada
6	DTR	Salida
7	Tierra	Común
8	CD	Entrada
9	Recibe reloj	Entrada

El mapeo de los registros del MK68564 en el espacio de memoria del 68000 se muestra en la tabla 2. El MK68564 utiliza interrupción de nivel 3. Canal A del dispositivo usa los vectores de excepción 68 a 71 y el canal B usa los vectores 74 a 67.

Tabla 2. Mapa de memoria de los registros de los puertos serie MK68564 en tarjeta PC68K.

Dirección	Registro
E80001	Canal A. Registro de comandos.
E80003	Canal A. Registro de control de modo
E80005	Canal A. Registro de control de interrupción
E80007	Canal A. Registro 1 de palabra sync
E80009	Canal A. Registro 2 de palabra sync
E8000B	Canal A. Registro de control de receptor
E8000D	Canal A. Registro de control de transmisor
E8000F	Canal A. Registro 0 de estado
E80011	Canal A. Registro 1 de estado
E80013	Canal A. Registro de datos
E80015	Canal A. Registro de constante de tiempo.
E80017	Canal A. Registro de control generador de tasa de señalización(Baud rate)
E80019	Canal A. Registro de vector de interrupción.
E80021	Canal B. Registro de comandos.
E80023	Canal B. Registro de control de modo
E80025	Canal B. Registro de control de interrupción
E80027	Canal B. Registro 1 de palabra sync
E80029	Canal B. Registro 2 de palabra sync
E8002B	Canal B. Registro de control de receptor
E8002D	Canal B. Registro de control de transmisor
E8002F	Canal B. Registro 0 de estado
E80031	Canal B. Registro 1 de estado
E80033	Canal B. Registro de datos
E80035	Canal B. Registro de constante de tiempo.
E80037	Canal B. Registro de control generador de tasa de señalización(Baud rate)
E80039	Canal B. Registro de vector de interrupción.

2.2.3 MC68681 Puertos Serie

La tarjeta de expansión MMIO de la PC68K tiene 3 dispositivos MC68681. El MC68681 es un DUART (Dual Universal Asynchronous Receiver/Transmitter) que es un transmisor/receptor asincrono dual (2 canales). Cada DUART tiene 2 canales de comunicación asincrona para sumar un total de 6 canales.

Todos los 6 canales son accedidos por medio de un conector de 40 patas en la parte posterior de la tarjeta de expansión. La asignación de las patas se muestra en la figura 1.

DUART #	Canal	Señal	Patas		Señal	Canal	DUART #
3	A	Tx	1	21	IPO	A	3
3	A	Rx	2	22	Tierra	A	3
3	A	OPO	3	23	Tierra	A	3
3	B	Tx	4	24	IPO	B	3
3	B	Rx	5	25	Tierra	B	3
3	B	OPO	6	26	Tierra	B	3
2	A	Tx	7	27	IPO	A	2
2	A	Rx	8	28	Tierra	A	2
2	A	OPO	9	29	Tierra	A	2
2	B	Tx	10	30	IPO	B	2
2	B	Rx	11	31	Tierra	B	2
2	B	OPO	12	32	Tierra	B	2
1	A	Tx	13	33	IPO	A	1
1	A	Rx	14	34	Tierra	A	1
1	A	OPO	15	35	Tierra	A	1
1	B	Tx	16	36	IPO	B	1
1	B	Rx	17	37	Tierra	B	1
1	B	OPO	18	38	Tierra	B	1
		Tierra	19	39	Tierra		
		Tierra	20	40	Tierra		

Figura 1. Asignación de las señales en el conector de 40 patas que se encuentra en la parte posterior de la tarjeta de expansión para los 6 puertos serie de los 3 dispositivos DUART MC68681.

Estos 6 puertos, más los dos de la tarjeta principal de PC68K, nos da un total de 8 puertos serie. A través de estos puertos entrará la información de las estaciones de RESNOR al sistema. El mapeo de los registros de los tres dispositivos MC68681 en la memoria de la PC68K, se muestra en la tabla 3.

Tabla 3. Mapa de los registros de los tres dispositivos
DUART MC68681 (puertos serie) en memoria de PC68K.

Dirección	Lectura	Escritura
Dx0001	Registro de Modo A	Registro de Modo A
Dx0003	Registro de estado A	Registro selector de reloj A
Dx0005	NO ACCESAR	Registro de commando A
Dx0007	Dato de entrada A	Dato de salida A
Dx0009	Reg. cambio de entrada	Registro control auxiliar
Dx000B	Reg. estado interrup.	Reg. máscara interrupción
Dx000D	Contador Modo : MSB	Contador/temporizador alto
Dx000F	Contador Modo : LSB	Contador/temporizador bajo
Dx0011	Registro de Modo B	Registro de Modo B
Dx0013	Registro de estado B	Registro selector de reloj B
Dx0015	NO ACCESAR	Registro de commando B
Dx0017	Dato de entrada B	Dato de salida B
Dx0019	Reg. vector interrup.	Reg. vector de interrupción
Dx001B	Puerto entrada	Reg. config. puerto salida
Dx001D	Comando inicio contador	Registro puerto salida (set)
Dx001F	Comando alto contador	Registro puerto salida (reset)

donde x = 4 para DUART 1
 8 para DUART 2
 C para DUART 3

La intención de este mapa es indicar las localidades de memoria de los registros de los DUARTS dentro de la PC68K. Para mayor información sobre la descripción de cada uno de los registros del DUART consultar la referencia que se menciona en la bibliografía sobre MC68681.

El MC68681 usa el nivel de interrupción número 5. Los 6 canales usan el vector de excepción 29 (auto-vector).

2.2.4 MC6821 Puertos Paralelos

PC68K utiliza un dispositivo MC6821 que es un adaptador de interfaz paralelo (PIA). Este dispositivo nos permite conectar la PC68K a periféricos a través de dos puertos bidireccionales de 8 bits cada uno y 4 líneas de control. Este dispositivo es accesado a través de un conector de 40 patas de las cuales sólo 36 se utilizan (marcado como P2 en la tarjeta PC68K).

Señal	Patas		Señal
CA2	1	19	TIERRA = GND
PA0	2	20	TIERRA
PA1	3	21	TIERRA
PA2	4	22	TIERRA
PA3	5	23	TIERRA
PA4	6	24	TIERRA
PA5	7	25	TIERRA
PA6	8	26	TIERRA
PA7	9	27	TIERRA
CA1	10	28	TIERRA
PB1	11	29	TIERRA
PB2	12	30	TIERRA
PB3	13	31	PB6
PB4	14	32	PB7
PB5	15	33	TIERRA
TIERRA	16	34	CB1
TIERRA	17	35	CB2
N/C	18	36	PB0
(Ignorar las	xx	xx	4 últimas patas)
	xx	xx	

Figura 2. Asignación de las señales del adaptador de interfaz paralelo (PIA) MC6821, al conector de 40 patas que se encuentra en la parte posterior de la tarjeta PC68K.

La asignación de las señales del MC6821 a las patas del conector se muestra en la figura 2. En esta figura se utilizó la nomenclatura de MC6821. Esta asignación de patas facilita la conexión a impresoras con conectores de 36 patas tipo Centronics.

El mapa de memoria para el MC6821 es el siguiente:

E40001	Canal A. Datos/ Registro de dirección de datos
E40003	Canal A. Registro de control
E40005	Canal B. Datos/ Registro de dirección de datos
E40007	Canal B. Registro de control

El MC6821 usa el nivel de interrupción número 2 y es auto-vector. Las líneas de interrupción IRQA y IRQB están alambradas en una sola.'

2.2.5 DS1216 Reloj en Tiempo Real

El reloj en tiempo real convierte una memoria estándar RAM CMOS de 2K x 8 ó 8K x 8 en una memoria no volátil.

La memoria CMOS no volátil es direccionada como puerto de E/S con direccionamiento impar:

2K x 8 direcciones de CMOS son de E00001 a E01001

8K x 8 direcciones de CMOS son de E00001 a E04001

El acceso a DS1216 es a través de la dirección E00001. Se establece una comunicación con el DS1216 por medio del reconocimiento de un patrón de 64

bits en cadena que debe ser igualada por medio de 64 ciclos consecutivos de escritura en la localidad E00001 con la información apropiada.

El patrón de reconocimiento en hexadecimal es C5, 3A, A3, 5C, C5, 3A, A3, 5C. Con la secuencia correcta de los 64 bits, el DS1216 es habilitado y la transferencia de datos de o hacia los registros que mantienen el tiempo puede efectuarse. En los próximos 64 ciclos el DS1216 transmitirá o recibirá datos por la localidad E00001.

La información del DS1216 está contenida en 8 registros de 8 bits cada uno que son accedidos secuencialmente bit por bit. Esta información de los registros está en formato Binario Codificado a Decimal (BCD). La lectura o escritura de los registros siempre se lleva a cabo recorriendo los 8 registros, iniciando con el bit 0 del registro 0 y terminando el bit 7 del registro 7. En la figura 3 se muestran los registros del DS1216, su rango de valores en BCD y forma de acceso.

El bit 7 (aa) del registro 3, define el modo de operación de formato 12 o 24 horas. Cuando es 1, modo de 12 horas es seleccionado. En el modo de 12 horas, el bit 5 del mismo registro (3) nos indica si es am o pm. Cuando es 1, es pm. Cuando el modo de operación es de 24 horas, el bit 5 es el segundo bit de las decenas de horas.

Bits 4 y 5 del registro de días (4) son utilizados para controlar la reinicialización y el oscilador. Bit 4 controla la reinicialización (RESET). Cuando es 1, la entrada de la pata de reinicialización del

DS1216 es ignorada. Cuando es 0, una entrada en bajo en la pata de reinicialización, provocará que el DS1216 aborte la transferencia de datos sin cambiar el contenido de sus registros. El bit 5 controla al oscilador. Puesto en 1 el oscilador se parará, y en 0 arrancará al oscilador.

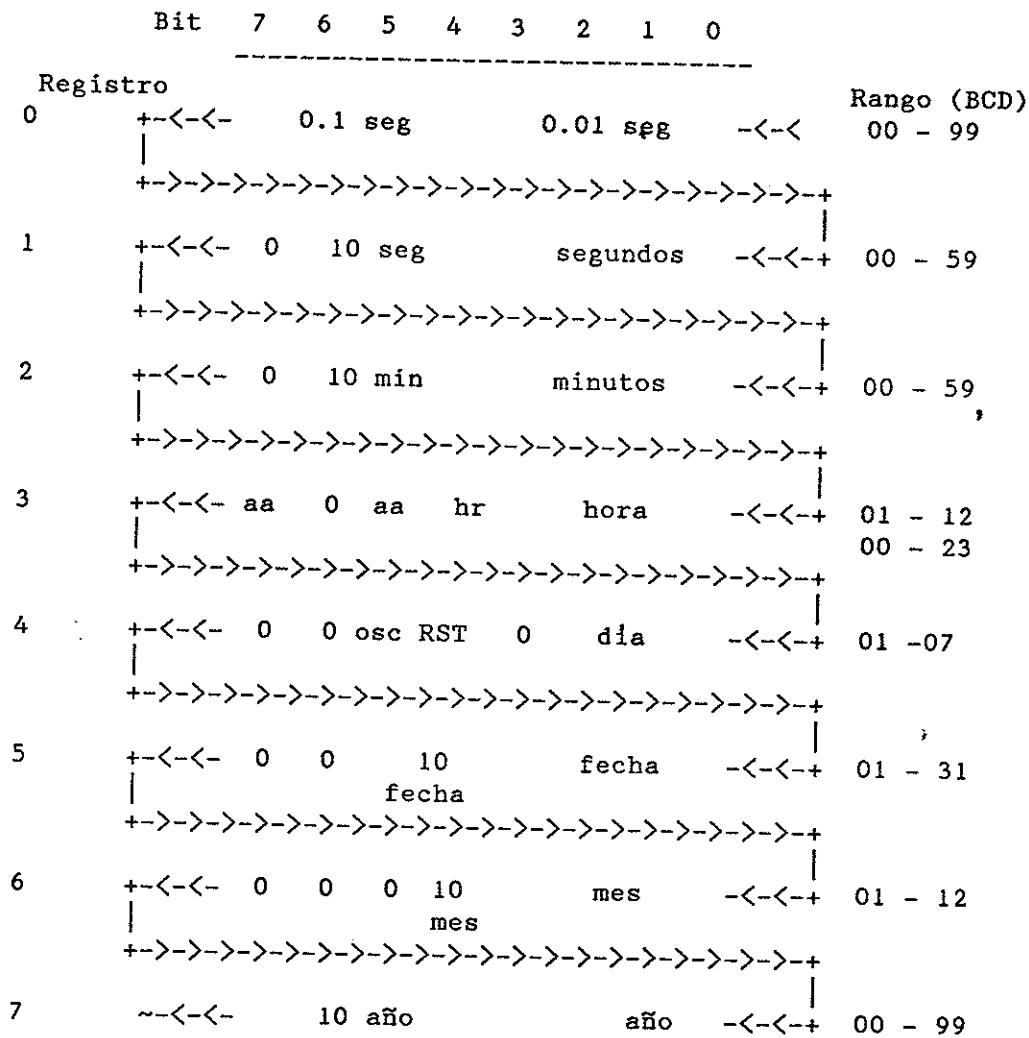


Figura 3. Registros del reloj en tiempo real DS1216, valor de sus rangos en Binario Codificado a Decimal (BCD) y orden de acceso: del bit 0 del registro 0 al bit 7 del registro 7.

2.2.6 Interfaz con la computadora IBM - PC (PCI)

Toda la comunicación entre la PC68K y la huésped IBM que utilicen MDP, se lleva a cabo a través de la PCI en la computadora IBM-PC. PCI es un programa en PC-DOS (o MS-DOS) llamado PC68K.COM. Cuando un programa que está corriendo en la PC68K solicita E/S de disco (por ejemplo), una solicitud es generada a través de la MDP y pasada a la PCI para su procesamiento.

Debido a la pugna por el acceso a la MDP mencionada anteriormente, se adaptó el concepto de maestro/esclavo en el diseño del programa PCI. PC68K asume siempre el papel de maestro al pasar información a través de la MDP. La PC68K generará una interrupción a la PC al escribir un byte identificador de tareas en la localidad de interrupción MDP:3FF. La PC al ser interrumpida, hará la lectura de MDP:3FF limpiando la interrupción impuesta obteniendo la identificación de tarea. Con esta identificación, la PC determinará que acción tomar por la solicitud. Posteriormente le da servicio a la solicitud y después escribe un byte identificador de tareas en la MDP:3FE interrumpiendo así a la PC68K. Esta última interrupción es normalmente utilizada como reconocimiento (ACK) a la PC68K que su solicitud ha sido atendida y que la MDP puede ser liberada para una próxima solicitud. Esto nos asegura que sólo un procesador intentará acceder la MDP en un momento dado.

Los bytes identificadores de tareas del 0 al 7F están reservados para el uso del sistema operativo. Valores del 81 al FF están disponibles para aplicaciones de usuarios.

3 SISTEMA OPERATIVO OS-9/68000

El poder y flexibilidad del microprocesador de 32 bits 68000 de Motorola, son realizados por el estructurado y avanzado sistema operativo OS-9/68000, que incorpora capacidades de multi-usario y multi-tareas. El sistema operativo OS-9/68000 desarrollado por Microware Systems Corporation es modular, que puede ser configurado según la máquina 68000 sobre la cual se vaya a montar. Desde pequeños sistemas basados en memoria de lectura exclusiva hasta sistemas grandes de multiusuario. Es apropiado para una gran variedad de aplicaciones en computadoras 68000, especialmente para aplicaciones en tiempo real. En OS-9 existen cuatro niveles de modularidad. Estos se describen a continuación.

Nivel 1 - Módulos del NUCLEO, RELOJ e INIT

El NUCLEO provee servicios básicos del sistema. Estos consisten en administración de entrada/salida, multi-tareas, memoria y el enlace a otros módulos del sistema.

El módulo del RELOJ es un manejador por programa para la electrónica específica del reloj en tiempo real. INIT es una tabla de inicialización usada por el NUCLEO durante el encendido del sistema. Esta especifica tamaños iniciales de tablas, nombres de dispositivos, etc.

Nivel 2 - Administradores de archivos (RBF,SCF,SBF,PIPEMAN,NFM)

Los Administradores de archivos realizan el procesado de solicitudes de E/S (entrada/salida) para grupos similares de dispositivos de E/S. El administrador de archivos de bloques aleatorios RBF (Random Block File manager) procesa todas las funciones de dispositivos de tipo disco. El administrador de archivos de caracteres secuenciales SCF (Sequential Character File manager) procesa todos los dispositivos de almacenamiento no masivo. Estos operan básicamente caracter por caracter (terminales, impresoras). Este tipo de administradores de archivos, son los que utilizaremos para los datos de RESNOR por la naturaleza de estos.

El administrador de bloques secuenciales SBF (Sequential Block File manager) se encarga de dispositivos de cinta magnética. PIPEMAN administra comunicación entre procesos, utilizando áreas de memoria de almacenamiento temporal para la transferencia de datos. El administrador de archivos de red NFM (Network File Manager) procesa solicitudes de datos sobre la red de OS-9.

Nivel 3 - Manejadores de dispositivos (Device Drivers)

Los Manejadores de dispositivos realizan las funciones físicas básicas de E/S para controladores específicos de E/S. Son dependientes de la electrónica. Los manejadores de los puertos serie, serán los encargados de proporcionarnos las muestras que llegan de las estaciones RESNOR. Se describen al final de este capítulo al conjuntar OS-9 y PC68K.

Nivel 4 - Descriptor de dispositivos.

Los Módulos descriptores de dispositivos son pequeñas tablas que asocian puertos específicos de E/S con su nombre lógico, manejador de dispositivo (device driver) y su administrador de archivos (file manager). Estos módulos también contienen la dirección física del puerto y los valores de inicialización. Con el uso de descriptores de dispositivos, solo una copia de cada manejador de dispositivos es necesaria para todos los dispositivos que pertenezcan al mismo grupo, independientemente de cuántos estén colgados del sistema.

Un componente importante no mostrado es el Shell, que es un intérprete de mandos. Técnicamente es un programa y no forma parte del sistema operativo por sí mismo.

3.1 EL NUCLEO

3.1.1 Administración de Memoria.

3.1.1.1 Funciones Básicas del NUCLEO

El NUCLEO de OS-9 es un administrador del sistema, supervisa y controla los recursos de la máquina. Es un módulo compacto escrito en ensamblador del 68000. Es independiente de posición en memoria. Las funciones básicas del NUCLEO son las siguientes:

- 1.-Procesado de solicitudes de servicio (llamadas al sistema).
- 2.-Administración de memoria.
- 3.-Inicialización del sistema.
- 4.-Administración del procesador 68000.
- 5.-Administración de Entrada/Salida.
- 6.-Procesado de excepciones e interrupciones.

Cuando se hace una llamada al sistema, ocurre una trampa (trap) de usuario al NUCLEO. El NUCLEO determina que tipo de llamada al sistema quiere realizar el usuario. OS-9 tiene dos tipos generales de llamadas al sistema:

-Llamadas que realizan entrada/salida

-Llamadas que efectúan funciones del sistema

Las llamadas a funciones del sistema, son efectuadas directamente por el NUCLEO. Las Llamadas que solicitan entrada/salida, son transferidas a otras partes del OS-9 y no son efectuadas por el NUCLEO.

3.1.1.2 Funciones de Administración de Memoria del Núcleo.

OS-9 maneja la asignación física de memoria a programas y los contenidos lógicos de memoria por medio del uso de módulos de memoria.

Los módulos de memoria son la estructura de datos básica de OS-9, que permite un ambiente de programación modular. Para que cualquier objeto

(un programa, una tabla, etc.) sea cargado a memoria, debe estar en el formato estándar de OS-9 de módulos de memoria. Esto permite a OS-9 mantener un directorio que contiene el nombre, la dirección y otra información relacionada con cada módulo que se encuentra en memoria.

El sistema operativo sigue la pista de cada módulo que se encuentra en memoria por medio de un directorio. Cuando los módulos son cargados a memoria, son añadidos al directorio de módulos.

Cada entrada del directorio contiene la dirección y un contador de los procesos que están utilizando ese módulo. Este contador es llamado el contador de enlaces (link count).

Cuando un proceso se enlaza a un módulo en memoria, el contador de enlace del módulo se incrementa en uno. Cuando un proceso se desliga de un módulo, su contador de enlaces se decrementa en uno. Cuando un módulo ya no es requerido (contador de enlaces igual a 0), la memoria que utiliza es recuperada por el sistema y su entrada en el directorio es borrada.

3.1.1.2.1 Estructura de un Módulo Básico

Cada módulo tiene tres partes : un encabezado, el cuerpo del módulo y un valor de Verificación de Redundancia Cíclica (CRC). (ver figura 4)

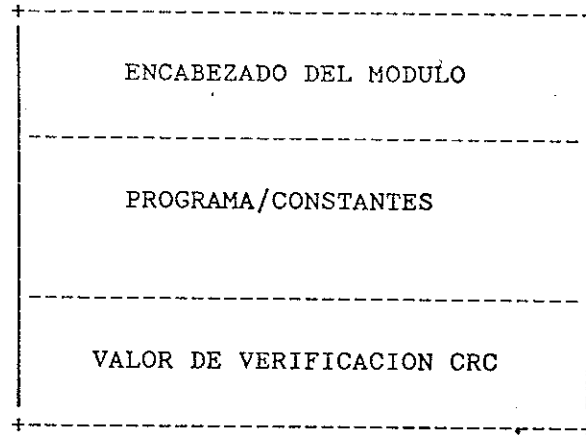


Figura 4. Formato de Módulo Básico de Memoria

- 1.-El encabezado del módulo contiene información que describe el módulo y su uso. El encabezado es creado por el cargador a la hora de ser cargado. La información contenida en el encabezado incluye el nombre del módulo, tamaño, tipo, lenguaje, requerimientos de memoria y punto de entrada. En el manual "OS9-9/68000 Operating System Technical Manual" páginas 1-4 a 1-11, vienen explicados los diferentes campos que integran el encabezado de los módulos.
- 2.-El cuerpo del módulo contiene datos de inicialización, instrucciones del programa, tablas de constantes, etc.
- 3.-Los tres últimos bytes del módulo albergan un valor de CRC (Cyclic Redundancy Check) verificación de redundancia cíclica para certificar la integridad del módulo.

3.1.1.2.2 Requerimientos de los Módulos.

Existen diferentes tipos de módulos. Cada tipo tiene diferente uso y función. Los módulos no tienen que ser programas completos o inclusive estar escritos en lenguaje de máquina. Los requisitos principales de los módulos son: que no se automodifiquen y que sean independientes de posición en memoria. Esto permite a OS-9 cargarlos en cualquier área de memoria disponible.

El conjunto de instrucciones del 68000 soporta un estilo de programación llamado "código re-entrante"; Módulos que no se automodifican son llamados módulos re-entrantes. Esto permite que una misma copia de un módulo sea compartida por dos o más procesos diferentes en forma simultánea. Los procesos no se interfieren uno a otro, puesto que cada uno tiene una área separada para sus variables.

OS-9 sabe automáticamente cuántos procesos está usando un módulo por medio de el contador de enlaces. También borra un módulo, liberando su memoria si su contador de enlaces se vuelve 0.

NOTA. Los módulos de datos son la excepción a la restricción de no automodificación. Es indispensable una cuidadosa coordinación para que varios procesos compartan los mismos módulos de datos. En la implementación del sistema de captura de datos de RESNOR, este tipo de módulos son la parte medular del diseño del sistema, ya que existe un módulo de datos que es compartido y accesado en forma concurrente por tres procesos.

Al hablar de código independiente de su posición, se quiere decir que un programa no sabe donde será cargado en memoria. OS-9 determina una dirección de cargado apropiada sólo cuando un programa se va a correr. Los compiladores e interpretes de OS-9 generan código independiente de posición en forma automática.

3.1.1.3 Mapa de Memoria de OS-9/68000

OS-9 utiliza un sistema de administración de memoria por programa donde toda la memoria está contenida dentro de solo un mapa de memoria. Por lo tanto, todas las tareas de usuario comparten un espacio de direccionamiento común.

Un mapa típico de memoria para OS-9/68000 se muestra en la figura 5. Las varias secciones mostradas como ROM, RAM, E/S, etc., no requieren una dirección específica a excepción de donde se indique. Es recomendado que cada sección sea guardada en bloques reservados contiguos.

3.1.1.4 Asignación de Memoria por el Sistema.

Durante la secuencia de inicialización de OS-9, los bloques de RAM y ROM son encontrados por una función automática de búsqueda que se encuentra en el ROM de inicialización. Algo de RAM es reservado por OS-9 para sus propias estructuras de datos y también se buscan bloques de ROM. (aquéllos que tengan módulos válidos de OS-9)

La cantidad de memoria requerida para OS-9 es variable. Requerimientos dependen de la configuración del sistema y del número de tareas activas y archivos abiertos. A continuación se describe cantidades aproximadas de memoria utilizada por varias partes de OS-9.

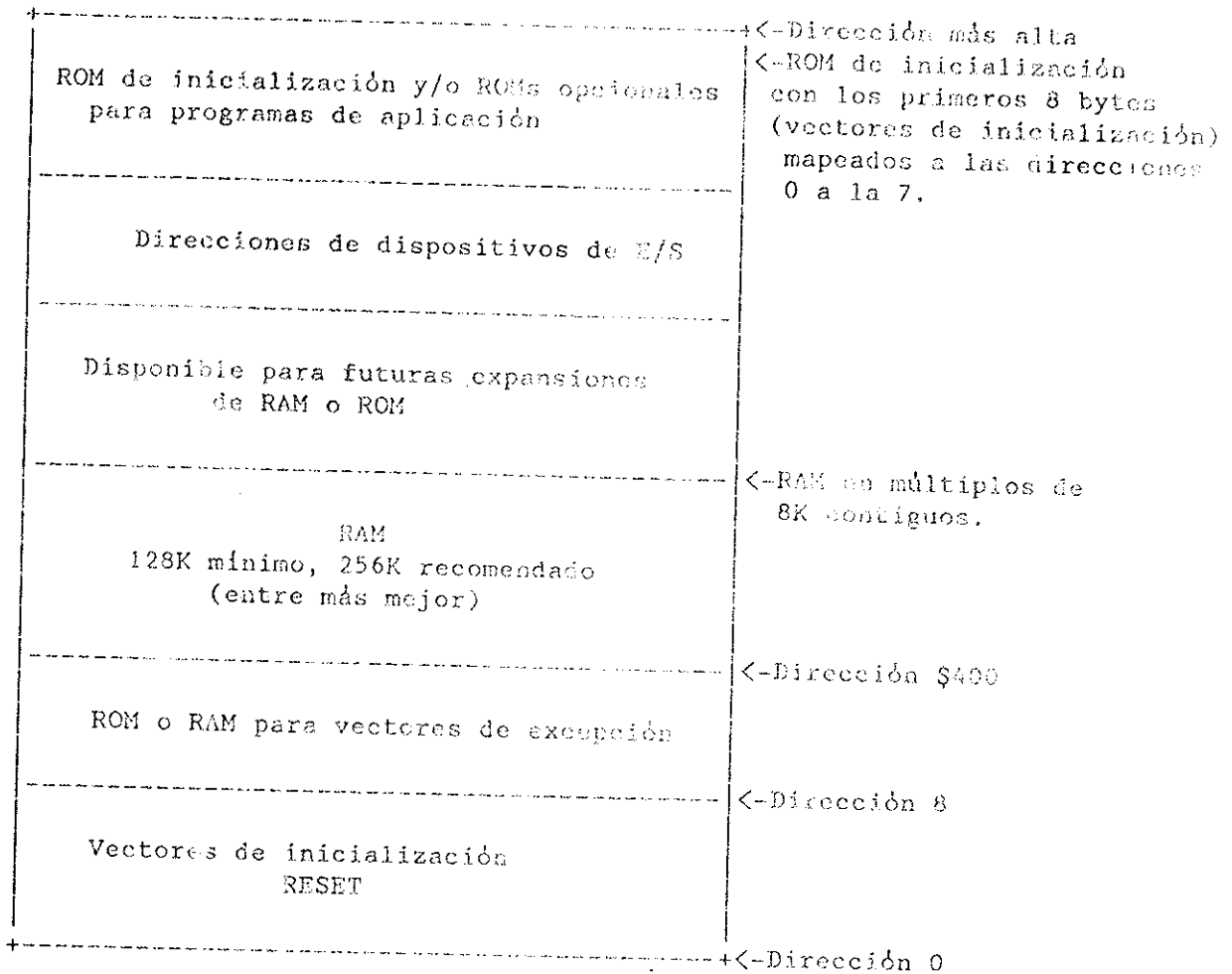


Figura 5. Mapa de Memoria Típico para OS-9/68000

3.1.1.4.1 Código Objeto de Sistema Operativo.

Un conjunto completo típico de los módulos del sistema (NUCLEO, administradores de E/S, manejadores de dispositivos, etc.) ocupa entre 24K y 32K de memoria. Estos módulos son cargados en RAM durante inicialización en sistemas que usan discos. OS-9 no carga dinámicamente sobreposiciones (overlays) o intercambia código del sistema. Esto quiere decir que no se requiere memoria adicional para código del sistema.

OS-9 puede residir en ROM para sistemas que no tengan disco. El tamaño típico de OS-9 en un sistema basado en ROM varía entre 20K y 24K.

3.1.1.4.2 Memoria Global del Sistema.

OS-9 usa una sección de 8K de RAM para uso interno. Esta área de memoria está generalmente localizada en las direcciones más bajas de RAM. Esta contiene una tabla de saltos por excepción y variables globales del sistema. Variables en esta área están definidas simbólicamente en la librería "sys.1" y son aquellas que tienen el prefijo "D_". Por medio de llamadas al sistema se pueden leer variables en esta área.

3.1.1.4.3 Memoria Dinámica del Sistema.

OS-9 mantiene estructuras de datos de tamaño dinámico (como áreas de almacenamiento temporal para E/S, descriptores de trayectorias, descriptores de procesos, etc.) y son asignadas de el área general de RAM cuando son solicitadas. Apuntadores a estas direcciones de estructuras de datos son mantenidas en el área de memoria global del sistema. En un sistema pequeño típico, la memoria RAM será de aproximadamente de 16K. Tamaños exactos de todas las estructuras de datos del sistema pueden encontrarse estudiando los listados de los archivos fuentes que forman el archivo de librería "sys.1"

3.1.1.4.4 Memoria de Usuario.

Toda la memoria de RAM no usada se asigna a un banco de memoria libre. El espacio de memoria es reservado y liberado del banco al ser asignado y desasignado para propósitos varios. OS-9 automáticamente asigna memoria del banco cuando ocurren cualquiera de los casos siguientes:

- 1.-Cuando son cargados módulos a RAM.
- 2.-Cuando nuevos procesos son creados.
- 3.-Cuando procesos existentes demandan memoria adicional.
- 4.-Cuando OS-9 necesita más memoria de almacenamiento temporal para E/S o sus estructuras de datos internos requieren ser expandidas.

El almacenamiento para módulos de programas de usuario y espacio para datos es dinámicamente asignado y desasignado del banco de memoria libre.

3.1.2 Llamadas al Sistema

3.1.2.1 Estados de Sistema y de Usuario

Existen 2 distintos ambientes en OS-9 en los que el código objeto puede ser ejecutado:

Estado de Usuario: Estado de usuario es el ambiente normal en que programas se ejecutan. Generalmente, procesos de estado de usuario no se involucran directamente con la configuración específica de la electrónica de la máquina OS-9.

Estado de Sistema: Las llamadas al sistema de OS-9 y las rutinas de servicio a interrupciones se ejecutan en estado de sistema. En la familia de procesadores 68000, es un sinónimo de estado de supervisor. Las rutinas de estado de sistema están involucradas con equipo colgado al sistema.

Las funciones que se ejecutan en estado de sistema tienen varias ventajas sobre aquellas que corren en estado de usuario:

- 1.-Una rutina en estado de sistema tiene acceso irrestricto a todas las capacidades del procesador.
- 2.-Existe un grupo de llamadas al sistema OS-9 que son accesibles solo desde estado de sistema.
- 3.-Rutinas de estado de sistema nunca son detenidas por el recalendarizador (despachador). Una vez que una rutina ha entrado a estado de sistema, ningún otro proceso se ejecutará hasta que el proceso de estado de sistema haya terminado. Solo si el proceso de estado de sistema se va a dormir, se podría ejecutar otro proceso. El único procesado que podría recalendarizar una rutina de estado de sistema es una rutina de servicio a interrupciones.

3.1.2.1.1 Instalación de Rutinas de Estado de Sistema.

Con acceso directo a toda la electrónica del sistema, cualquier rutina de estado de sistema tiene el poder de tomar el completo control sobre toda la máquina. Esto puede ser peligroso. En rutinas de estado de sistema es un reto mantener a la maquina funcionando y prevenir que se cuelgue en algún proceso.

En OS-9, existen cuatro formas de proveer rutinas de estado de sistema.

- 1.-Instalar un módulo que se llame "OS9P2" en el archivo del sistema de inicialización o en ROM. Durante el encendido, el NUCLEO de OS-9 se enlazará a este módulo, y si lo encuentra, llamará a su punto de entrada de ejecución.
- 2.-Utilizar el sistema de E/S como puerta a estado de sistema. Administradores de archivos y manejadores de dispositivos se ejecutan siempre en estado de sistema. Se podría escribir un manejador de dispositivos que lo único que hiciera es darnos acceso a estado de sistema.
- 3.-Escribir un módulo manejador de trampas que se ejecute en estado de sistema.
- 4.-Un programa se ejecutará en estado de sistema si el bit de estado de "supervisor" en el campo de atributo/revisión del módulo está prendido y además sea poseído por el superusuario.

De cualquier manera que se ejecuten las rutinas de estado de sistema hay que recordar que éstas no son detenidas por el despachador, por lo tanto tienen que ser escritas lo más cortas y rápidas posibles.

3.1.2.2 Procesado de Llamadas al Sistema por el Nucleo.

Todas las solicitudes de servicio a OS-9 (llamadas al sistema) son procesadas por el NUCLEO. Las solicitudes de servicio son usadas para comunicarse entre OS-9 y programas para aspectos como reservado de memoria o creación de procesos. Además de funciones para E/S y administración de memoria, existen otras funciones de solicitud de servicio para control entre procesos y control de agenda para el recalendarizador.

Las librerías del sistema "sys.1" y "usr.1" definen nombres simbólicos para todas las solicitudes de servicio. Las librerías son ligados con lenguaje ensamblador o código generado por compilador. OS-9 tiene un macro para generar llamadas al sistema:

```
OS9 I$Read
```

Esta sentencia es reconocida y traducida al siguiente código:

```
TRAP #0  
dc.w I$Read
```

El compilador de C incluye funciones estándar para acceder casi todas las llamadas al sistema de OS-9 en modo de usuario.

Los parámetros para llamadas al sistema son generalmente pasados y regresados en los registros del procesador. Las llamadas al sistema son

divididas en tres grupos:

- 1.-Llamadas al sistema en estado de usuario: Estas funciones realizan administración de memoria, multiprogramación y otras funciones para programas de usuario. Estas son procesadas generalmente por el núcleo. Los nombres simbólicos para esta categoría inician con "F\$" (Ejemplo: F\$Link).
- 2.-Llamadas al sistema E/S: Estas solicitudes realizan varias funciones de E/S y son procesadas por un administrador de archivos o por un manejador de dispositivos. Los nombres simbólicos para esta categoría de llamadas inician con "I\$" (Ejemplo: I\$Read).
- 3.-Llamadas al sistema en estado de sistema: Estas solicitudes son llamadas especiales al sistema que sólo pueden invocarse desde estado de sistema. Estas regularmente operan sobre estructuras internas de OS-9. Los nombres simbólicos de estas llamadas inician con "F\$".

3.1.2.3 Entrada/Salida desde Estado de Sistema.

Todas las llamadas de E/S al sistema pueden invocarse desde rutinas de estado de sistema. Existe una pequeña diferencia cuando estas son ejecutadas desde estado de usuario. La diferencia es que todos los números de trayectoria usados en estado de sistema son números de trayectoria de "sistema". Cada descriptor de proceso tiene un número de trayectoria que es utilizado para convertir los números de trayectoria locales de procesos a números de trayectoria del sistema. A su vez, el sistema tiene una tabla global de números de trayectoria para convertir números de trayectoria de sistema a direcciones actuales de descriptores de trayectorias. Llamadas al sistema de E/S desde estado de sistema deben hacerse con números de trayectoria de sistema.

—
 Cuando un proceso de estado de usuario termina con una trayectoria de E/S abierta, ésta es cerrada automáticamente por la rutina F\$Exit. Esto

ocurre porque OS-9 graba las trayectorias abiertas en la tabla de trayectorias de procesos. En estado de sistema, las llamadas al sistema I\$Open y I\$Create regresan un número de trayectoria de sistema. Este no es grabado en la tabla de trayectorias de procesos ni en ninguna otra parte por OS-9. Es responsabilidad de la rutina de estado de sistema que abra cualquier trayectoria de E/S se asegure que en algún momento se cierre.

3.1.3 Administración del Procesador y Ejecución de Procesos.

3.1.3.1 Multi-Tareas en OS-9

OS-9 es un sistema operativo de multitareas que permite a varios programas independientes llamados procesos o tareas, se ejecuten simultáneamente. Cada proceso puede tener acceso a cualquier recurso del sistema por medio de una solicitud de servicio apropiada a OS-9.

El tiempo de UCP (Unidad Central de Procesamiento alias CPU) es un recurso finito que debe ser distribuido eficientemente para maximizar el uso de la computadora. OS-9 asigna automáticamente tiempo de UCP sólo aquellos procesos que pueden utilizar ese tiempo eficientemente. Para realizar esto, OS-9 utiliza una técnica llamada de rebanadas de tiempo (timeslicing).

El rebanado de tiempo permite a los procesos compartir el tiempo de UCP con todos los otros procesos activos. Es implementado usando funciones de mecánica (hardware) y programática (software).

La UCP es interrumpida por un reloj en tiempo real a una tasa regular de (usualmete) 100 veces por segundo. Esta unidad básica de tiempo es llamada "tick".

En cualquier ocurrencia de un tick, OS-9 puede suspender la ejecución de un proceso e iniciar otro. La suspensión y arrancado de procesos es de tal manera que no afecta la ejecución del proceso.

Los procesos que son activos (no están esperando por algún evento) se corren por un período asignado por el sistema llamado rebanda de tiempo (time slice). Qué tan frecuente un proceso recibe una rebanada de tiempo depende de la prioridad relativa del proceso con respecto a los demás procesos activos. Existen solicitudes de servicio a OS-9 para crear, terminar y controlar procesos.

3.1.3.1.1 Areas de Memoria para Procesos.

Todos los procesos son divididos en dos áreas lógicas de memoria: una área para código y otra para datos.

Un programa debe de estar en la forma de un módulo de memoria ejecutable para que pueda ser corrido y es llamado el módulo primario. Puede ser ligado y ejecutar código de otros módulos. Es considerado independiente de su posición y de lectura exclusiva.

El área de datos de un proceso es un espacio separado de memoria donde

todas las variables del programa se guardan. La parte alta de esta área, es usada para la pila del programa. La dirección de esta área es asignada a la hora de ejecutarse el programa. Una dirección base es guardada en el registro a6 para acceder el área de datos. Esta zona sí es modificable (lectura/escritura).

Si un programa requiere que sean inicializadas variables, los valores de inicialización son copiados del área de lectura exclusiva (módulo primario).

3.1.3.1.2 Creación de Procesos

Los nuevos procesos se crean con la llamada al sistema "F\$Fork". El parámetro más importante pasado a F\$Fork es el nombre del módulo primario que el nuevo proceso iniciará ejecutando. La creación de un proceso se describe a continuación:

- 1.-Localizar y cargar el programa. OS-9 primero busca el módulo en memoria. Si no lo encuentra, OS-9 carga en memoria un archivo de disco que concuerde con el nombre del módulo.
- 2.-Almacena e inicializa un descriptor de proceso. Una vez que el módulo primario ha sido encontrado, una estructura de datos llamada descriptor de proceso es asignada al nuevo proceso. El descriptor de proceso es una tabla que contiene información del proceso, su estado, dirección de memoria, prioridad, trayectorias de E/S, etc. El descriptor de proceso es iniciado y mantenido por OS-9.
- 3.-Reservado del área de datos y pila. El encabezado del módulo primario contiene el tamaño del área de datos y pila. OS-9 reserva una área contigua de memoria del tamaño requerido. La memoria es reservada del banco de memoria libre.
- 4.-Inicializa proceso. Los registros del nuevo proceso son inicializados con las direcciones apropiadas del área de datos y del módulo de

código. (ver figura 6). Si el programa requiere que se inicialicen variables y/o apuntadores, estos son copiados del módulo de código a su dirección apropiada en el área de datos.

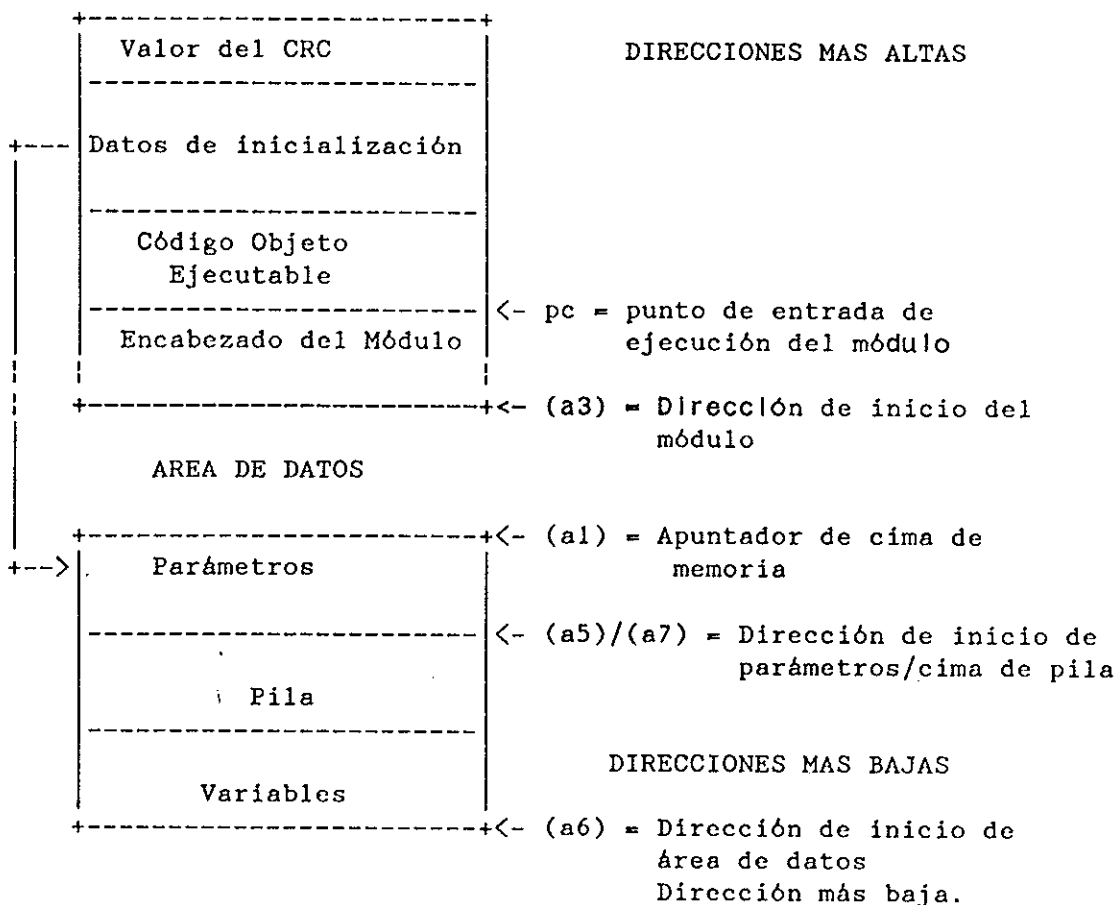
Si alguno de los pasos anteriores no puede ser realizado, la creación del nuevo proceso es abortado, y el proceso que originó la llamada `F$Fork` es notificado del error. Sino, el nuevo proceso es añadido a la cola de procesos activos para su calendarización de ejecución (`execution scheduling`).

Al nuevo proceso también se le asigna un número único llamado identificación de proceso (`Process ID`). Otros procesos se pueden comunicar con él por medio de su identificación. El proceso también tiene asociado una identificación de Grupo y de Usuario. Estas últimas identificaciones son heredadas del proceso padre que lo creo.

Los procesos terminan cuando ejecutan una solicitud de servicio `F$Exit` o cuando reciben señales fatales o de error. La terminación del proceso cierra cualquier trayectoria abierta, libera su memoria y desliga el módulo primario.

MODULO PRIMARIO

CONTENIDO DE REGISTROS



Registros pasados al nuevo proceso

sr =0000	(a0)=no definido
pc = punto de entrada de ejecución del módulo	(a1)=apuntador a cima de memoria
d0.w=Identificación de proceso	(a2)=no definido
d1.w=ID de grupo/usuario	(a3)=apuntador a módulo primario
d2.w=prioridad	(a4)=no definido
d3.w=# de trayectorias de E/S heredadas	(a5)=apuntador a parámetros
d4.l=no definido	(a6)=apuntador a área de datos almacenamiento estático
d6.l=Total de memoria inicial reservada	(a7)=apuntador de pila (igual que a5)

NOTA: (a6) siempre es polarizado por \$8000 para permitir programas objeto acceder 64K de datos usando direccionamiento indizado. Generalmente esta polarización puede ser ignorada porque el cargador se ajusta automáticamente a ella.

Figura 6. Mapa de memoria inicial y contenido de registros de un nuevo proceso.

3.1.3.2 Estado de Procesos

En cualquier instante, un proceso se encuentra en uno de los tres siguientes estados:

- ACTIVO El proceso es activo y listo para ejecutarse
- ESPERANDO EL proceso está inactivo hasta que un proceso hijo termine o una señal sea recibida.
- DURMIENDO El proceso está inactivo por un período de tiempo específico o hasta que una señal sea recibida.

Existe una cola para cada estado de procesos. Cada cola es una lista ligada de descriptores de procesos correspondientes a todos aquellos procesos con el mismo estado. Los cambios de estado se llevan a cabo moviendo un descriptor de proceso de su cola corriente a otra.

Estado Activo

El estado activo incluye todos los procesos ejecutándose. A estos procesos les son dadas rebanadas de tiempo para ejecutarse de acuerdo a su prioridad relativa con respecto a todos los procesos activos. El calendarizador (scheduler) usa un método que involucra la comparación de edades con cada proceso activo en la cola. Le da a todos los procesos algo de tiempo de UCP, incluso a aquéllos con un muy baja prioridad relativa.

Estado de Espera

Se entra al estado de espera cuando el proceso ejecuta una solicitud de servicio F\$Wait. Este proceso permanece inactivo hasta que cualquiera de sus procesos descendientes termine o hasta que reciba una señal.

Estado Durmiente

Se entra al estado durmiente cuando un proceso ejecuta una solicitud de servicio F\$Sleep. La solicitud de servicio F\$Sleep especifica un intervalo de tiempo que el proceso debe permanecer inactivo. Cero ticks especifica un periodo infinito de tiempo. El proceso permanece dormido hasta que el tiempo especificado haya transcurrido o hasta que se reciba una señal.

En nuestro sistema, cada vez que se aplica el algoritmo de detección a las señales sísmicas, y la condición de disparo no es satisfecha, Se hace una solicitud para entrar en estado durmiente por un periodo de un promedio corto (se explica en la sección 5.3.3).

3.1.3.3 Calendarización de Ejecución.

El NUCLEO es responsable de distribuir tiempo de UCP a todos los procesos activos. OS-9 usa un algoritmo de calendarización que asegura que todos los procesos obtengan algo de tiempo de ejecución.

Todos los procesos activos son miembros de una cola de procesos activos,

que se mantiene ordenada (clasificada) por la edad de los procesos. La edad de un proceso es un contador de cuantos cambios de procesos han ocurrido desde que el proceso entro a la cola, más la prioridad inicial del proceso.

Cuando un proceso es movido a la cola de procesos activos, su edad es inicializada con la prioridad asignada. Procesos que tengan prioridad relativa mayor son puestos en la cola con una edad artificial mayor. Siempre que un nuevo proceso es puesto en la cola de activos, las edades de todos los otros procesos son incrementadas. Las edades nunca son incrementadas mas allá de \$FFFF.

Al terminar la rebanada de tiempo del proceso que se está ejecutando, el calendarizador escoge el proceso que tenga la mayor edad para que entre a ejecutarse. Como la cola se mantiene ordenada por edades, el proceso más viejo será el que esté al frente de la cola.

3.1.3.3.1 Terminación Prematura de Procesos.

En aplicaciones de tiempo real, es necesario tener una respuesta rápida a interrupciones. OS-9 nos permite terminar en forma prematura el proceso que actualmente se está ejecutando, cuando un proceso de más alta prioridad se vuelve activo. El proceso de más baja prioridad pierde el resto de su rebanada de tiempo y es reinsertado a la cola de activos.

El relevo de procesos o tareas es afectado por dos variables globales del

sistema: `D_MinPty` y `D_MaxAge`. Ambas variables son accesibles por usuarios con identificación de grupo cero (super usuarios) a través de la llamada al sistema `F$SetSys`.

`D_MinPty` define una prioridad mínima por debajo de la cual los procesos no son añejados (no envejecen) ni tampoco son candidatos para ejecución. `D_MinPty` es generalmente puesto en cero en la mayoría de los sistemas. Cuando es puesto con algún nivel de prioridad, todos los procesos que estén corriendo por debajo de ese nivel, son parados por completo mientras que tareas críticas corren hasta completarse.

Esto es peligroso porque si la prioridad mínima del sistema es puesto por encima de la prioridad de todos los procesos que están corriendo, el sistema se cerrará por completo y sólo podrá ser recuperado con una reinicialización del sistema. Es crucial reestablecer `d_MinPty` a un nivel normal cuando las tareas críticas hayan terminado.

`D_MaxAge` define una edad máxima a la cual procesos no se les permite madurar por encima de esta edad. Generalmente esta edad es puesta en cero, pero cuando está activa, `D_MaxAge` divide las tareas en dos grupos: baja y alta prioridad. Las tareas de baja prioridad dejan de envejecer cuando llegan a `D_MaxAge`. Tareas de alta prioridad recibirán todo el tiempo disponible de UCP. Tareas de alta prioridad nunca envejecen. Todas las tareas de baja prioridad se ejecutarán sólo cuando las de alta prioridad estén inactivas.

La excepción a estas reglas es para cualquier proceso que este realizando una llamada al sistema. Este no tendrá una terminación prematura y terminará de ejecutarse a menos que voluntariamente renuncie a su rebanada de tiempo.

3.1.3.4 Procesamiento de Excepciones e Interrupciones.

OS-9 soporta en forma extensiva el sistema avanzado de interrupciones/excepciones del 68000. Rutinas para manejar una excepción en particular pueden ser instaladas usando varias llamadas al sistema OS-9 para los diferentes tipos de excepciones. La tabla 4 muestra la relación entre el vector número, mnemónico de la llamada al sistema de OS-9 y la asignación que OS-9 le vinculó.

Por el momento los que nos interesan son vectores 25 a 31 y 64 a 255 que están relacionados con los manejadores de puertos serie (dispositivos) de PC68K.

Vectores 25 a 31 son autovectores de interrupción. Estas excepciones proveen escrutinio (polling) de interrupciones para dispositivos de E/S que no generan vectores de interrupción. Internamente se manejan como vectores de interrupción. En particular, todos los puertos serie de la tarjeta de expansión de la PC68K utilizan el mismo autovector 29.

Vectores 64 a 255 son 192 vectores de interrupción. Proveen de un mínimo de carga al sistema para llamar un módulo manejador de dispositivos para

atender una interrupción. Rutinas de atención a interrupciones son ejecutadas en estado de sistema sin ningún proceso corriente asociado. El manejador de dispositivo debe proveer un entrada de error para que el sistema la ejecute en caso de que ocurra una excepción de error durante el procesado de interrupción. La llamada al sistema F\$IRQ es utilizada para instalar el manejador de interrupción en la tabla de interrupciones. Múltiples dispositivos pueden usar el mismo vector si es necesario.

3.2 Sistema de Entrada/Salida E/S

El NUCLEO realiza algo del procesado de E/S (como reservar espacio para estructuras de datos), y luego llama al módulo administrador de archivos apropiado que a su vez llama al módulo manejador de dispositivos. El administrador de archivos, el manejador de dispositivo y el descriptor de dispositivo son módulos estándar de memoria que pueden ser instalados o removidos en forma dinámica mientras el sistema esté corriendo.

El NUCLEO provee el primer nivel de servicio para las llamadas de E/S al sistema al direccionar los datos entre procesos y su administrador de archivos y manejador de dispositivos apropiado. El NUCLEO mantiene dos importantes estructuras de datos internos de OS-9: La tabla de dispositivos y la tabla de trayectorias.

Cuando se abre una trayectoria, el NUCLEO intenta enlazarse al módulo de memoria que tiene el nombre del dispositivo dado (o implícito) en la lista de trayectoria. El módulo a enlazarse es el descriptor del dispositivo,

que contiene los nombres del manejador de dispositivo y administrador de archivo para ese dispositivo. La información del descriptor de dispositivo se guarda por el NUCLEO para que las llamadas subsecuentes al sistema puedan ser direccionadas a estos módulos.

Tabla 4. Relación entre el número del vector de excepción del 68000, mnemónico de la llamada a OS-9 asociada y la función del vector.

Vector Número	Llamada a OS9 Asociada	Asignación
0	Ninguna	Apuntador de pila inicial del sistema
1	Ninguna	Contador de programa inicial del sistema
2	F\$STrap	Error de ducto
3	F\$STrap	Error de dirección
4	F\$STrap	Instrucción ilegal
5	F\$STrap	División por cero
6	F\$STrap	Instrucción CHK
7	F\$STrap	Instrucción TRAPV
8	F\$STrap	Violación de privilegios
9	F\$DFork	Trazado de instrucciones
10	F\$STrap	Emulador de línea 1010
11	F\$STrap	Emulador de línea 1111
12-13	Ninguna	Reservado
14	Ninguna	Reservado (error de formato)
15	Ninguna	Interrupción no inicializada
16-23	Ninguna	Reservado
24	Ninguna	Interrupción espuria
25-31	F\$IRQ	Autovectores de interrupción niveles 1-7
32	F\$OS9	Instrucción trampa de usuario #0 (llamada a OS-9)
33-47	F\$TLink	Trampas de usuario #1-#15 Vectores de instrucción
48	F\$STrap	Ramificación FPCP (Floating Point Coprocessor)
49	F\$STrap	Resultado inexacto FPCP
50	F\$STrap	FPCP división por cero
51	F\$STrap	FPCP Error de 'Underflow'
52	F\$STrap	FPCP error de operandos
53	F\$STrap	FPCP error de sobreflujo
54	F\$STrap	FPCP NAN señalado
55-63	Ninguna	Reservados
64-255	F\$IRQ	Vectores de interrupción

3.2.1 Administradores de Archivos.

La función de un administrador de archivos es procesar los datos crudos de ó hacia el manejador de dispositivos para un grupo similar de dispositivos.

Los administradores de archivos generalmente retienen los datos y solicitan al NUCLEO áreas temporales de trabajo en forma dinámica para almacenarlos. Pueden también examinar y procesar los datos.

Los administradores de archivos son re-entrantes. Un administrador de archivos puede ser utilizado por un grupo de dispositivos que tengan características operacionales similares. OS-9 puede tener cualquier número de módulos administradores de archivos.

Los cuatro administradores de archivos que son incluidos en un sistema típico son:

- 1.-Administrador de archivos con bloques aleatorios RBF (Random Block File Manager): Este administrador opera sobre dispositivos de acceso aleatorio de bloques estructurados como sistemas de disco.
- 2.-Administrador de archivos de caracteres secuenciales SCF (Sequential Character File Manager): Este administrador es utilizado en dispositivos que operan caracter por caracter como terminales, impresoras, modems, etc. Para nuestra aplicación tendremos que utilizar a SCF o una versión modificada de éste para capturar y procesar los datos de RESNOR.
- 3.-Administrador de archivos pipa PIPEMAN (Pipe File Manager): Este soporta comunicación entre procesos por medio de áreas temporales de trabajo llamadas pipas.
- 4.-Administrador de archivos de bloques secuenciales SBF (Sequential Block

File Manager): Para dispositivos que operan con bloques estructurados secuenciales como las unidades de cinta.

3.2.1.1 Funciones y Organización de Administradores de Archivos.

Un administrador de archivos es una colección de subrutinas accedidas a través de una tabla de desplazamientos (offset). La tabla contiene la dirección de inicio de cada subrutina relativa al inicio de la tabla. La posición de la tabla está especificada por el punto de entrada de ejecución en el encabezado del módulo.

Cuando las rutinas individuales de un administrador de archivos son llamadas, parámetros estándar son pasados en los siguientes registros:

- (a1) Apuntador al descriptor de trayectoria
- (a4) Apuntador al descriptor de proceso corriente
- (a5) Apuntador al registro de pila de usuario.
Registros de usuario pasan/reciben parámetros como se
mostró en la sección de llamadas al sistema.
- (a6) Apuntador al área de datos global del sistema.

3.2.1.2 Funciones de Rutinas de Administradores de Archivos.

A continuación sólo se hace mención a las funciones de interés que contiene el manual técnico de OS-9 para la aplicación a desarrollar.

LEER. Leer es responsable de regresar el número de bytes solicitados al área de datos del usuario. Este regresa un error de fin de archivo (EOF), si no hay datos disponibles. Leer debe ser capaz de copiar datos binarios

crudos y generalmente no efectúa ninguna edición sobre ellos. El administrador de archivos, generalmente llamará al manejador de dispositivo para leer los datos a una área de trabajo. Posteriormente copiará estos datos del área de trabajo al espacio de datos del usuario.

ESCRIBIR. La solicitud de escritura, como la de lectura, debe ser capaz de escribir datos binarios sin ninguna alteración.

GETSTAT y SETSTAT. Las llamadas al sistema Getstat (averigua el estado) y Setstat (Pon el estado) son llamadas comodines diseñadas para proveer un método de acceder aspectos del dispositivo (o administrador de archivos) que generalmente son dependientes del dispositivo.

El administrador de archivos puede realizar algunas funciones específicas como definir el tamaño de un archivo a un valor dado. Llamadas de estado que no son reconocidas por el administrador de archivos son pasadas al manejador.

3.2.2 Módulos Manejadores de Dispositivos de E/S.

Módulos manejadores de dispositivos de E/S realizan funciones básicas físicas de E/S. Estos módulos también son re-entrantes, lo cual permite que sólo una copia del módulo soporte en forma simultánea varios dispositivos que utilizan controladores idénticos de E/S.

3.2.2.1 Requerimientos Funcionales Básicos de un Manejador.

Un módulo manejador es un paquete de siete subrutinas que son llamadas por un administrador de archivos en estado de sistema. Sus funciones son:

- 1.-Inicializar la electrónica del controlador del dispositivo y variables relacionadas del manejador.
- 2.-Leer una unidad estándar física (un caracter o sector, dependiendo del dispositivo).
- 3.-Escribir una unidad estándar física (un caracter o sector, dependiendo del dispositivo).
- 4.-Regresar un estado especificado del dispositivo.
- 5.-Poner un estado especificado en el dispositivo.
- 6.-Liberar un dispositivo. Se asume que el dispositivo no va a ser utilizado posteriormente a menos que se re-inicialice.
- 7.-Procesa una excepción de error generada durante la ejecución del manejador.

En nuestra aplicación los módulos manejadores que nos interesan son los que atienden a los puertos serie. Como se vió en la sección de descripción del equipo, existen en la PC68K dos diferentes tipos de dispositivos para los puertos serie: el MK68564 y el MC68681. Hay un manejador para cada tipo de dispositivo. Estos se analizan en la sección OS9-PC68K.

Un módulo manejador puede manejar varias interfaces idénticas de dispositivos. La información específica para cada unidad física (dirección del puerto, constantes de inicialización, etc.) esta dada en un módulo llamada descriptor de dispositivo.

El nombre por el cual el dispositivo es conocido por el sistema es por el nombre del módulo descriptor del dispositivo. OS-9 copia algo de la información contenida en el descriptor de dispositivo a la estructura de datos descriptor de trayectoria para facilitar el acceso a manejadores.

3.2.2.2 Formato de un Módulo Manejador de Dispositivo.

Todos los manejadores se deben sujetar al formato estándar de módulos de memoria en OS-9. El tipo de código en el manejador es "Drivr". Un ejemplo de un encabezado se muestra en figura 7.

```

*   Encabezado del Módulo

      Type_Lang  equ  (Drivr<<8) + Objet
      Attr_Revs  equ  (RcEnt<<8) + 0

      psect  Acia,Type_Lang,Attr_Revs,Edition,0,AciaEnt

* Tabla de desplazamientos a puntos de entrada
AciaEnt dc.w  Init      Rutina de inicialización
         dc.w  Leer     Rutina de lectura
         dc.w  Escritura Rutina de escritura
         dc.w  GetStat  Trae el estado del dispositivo
         dc.w  SetStat  Establece estado de dispositivo
         dc.w  Termina Rutina de terminación
         dc.w  Error    Rutina que maneja errores

```

Figura 7. Formato del Encabezado de un Módulo Manejador de Dispositivo.

El desplazamiento de ejecución en el encabezado del módulo (M\$Exec) nos da

la dirección de una tabla de desplazamiento, que especifica la dirección de inicio de cada una de las siete subrutinas del manejador.

El tamaño del almacenamiento estático (M\$Mem) especifica la cantidad de almacenamiento local requerido por el manejador. Esta es la suma del almacenamiento requerido por el administrador de archivos (variables V_XXX) más cualquier variables o tablas declaradas en el manejador.

Las subrutinas del manejador son llamadas por su administrador de archivos asociado a través de la tabla de desplazamientos. Las rutinas de manejador son siempre ejecutadas en estado de sistema. Sin importar que tipo de dispositivo, los parámetros estándar mencionados a continuación son pasados al manejador en los registros. Otros parámetros que dependen del tipo de dispositivo y subrutina llamada pueden también ser pasados:

Inicializa y Termina

- (a1) Dirección del módulo descriptor del dispositivo
- (a2) Dirección del área de almacenamiento estático de variables del manejador.
- (a4) Dirección del descriptor de proceso que solicitó la función de E/S.
- (a6) Dirección del área de almacenamiento de variables globales del sistema.

Leer, Escribir, Getstat y Setstat

- (a1) Dirección del descriptor de trayectoria.
- (a2) Dirección del área de almacenamiento estático de variables del manejador.
- (a4) Dirección del descriptor de proceso que solicitó la función de E/S.
- (a5) Apuntador a la pila de registros del proceso llamador.
- (a6) Dirección del área de almacenamiento de variables globales del sistema.

Cada rutina termina con una instrucción RTS. Un estado de error es regresado utilizando el bit de acarreo CCR con un código de error en dl.w.

Una rutina adicional que se encuentra dentro del manejador de dispositivo, pero no está incluida en la tabla de desplazamientos a las 7 subrutinas, es la de servicio de atención a interrupciones para los dispositivos que son guiados por interrupciones. En nuestra aplicación, esta rutina juega un papel muy importante y se analiza en la sección de implementación.

3.2.2.3 Interrupciones

Como OS-9 es un sistema operativo de multi-tareas, un comportamiento óptimo del sistema se obtendrá cuando todos los dispositivos de E/S operen guiados por interrupciones.

Para dispositivos que operan caracter por caracter, el controlador deberá generar una interrupción al recibir un caracter y al terminar de transmitir un caracter. Los datos de entrada y salida deberán ser almacenados temporalmente por el manejador.

Generalmente, el módulo INIT añade las rutinas de servicio a interrupciones de dispositivos al sistema de escrutiño (polling) de interrupciones de OS-9 por medio de la llamada de sistema F\$IRQ. Las rutinas de LECTURA y ESCRITURA habilitan y deshabilitan las interrupciones del controlador según sea solicitado.

El manual técnico de OS-9 recomienda los siguientes niveles de prioridad en las interrupciones:

Reloj en tiempo real	Nivel 6
Puertos de terminales/impresoras	Nivel 4
Controladores de disco	Nivel 3
Procesadores de E/S	Nivel 2

Para los dispositivos de la PC68K los niveles de interrupción están alambrados de la siguiente manera:

Puertos serie MC68681	Nivel 5
Memoria de doble acceso MDP	Nivel 4
Puertos serie MC68564	Nivel 3
Adaptador de interfaz paralelo, (PIA) MC6821	Nivel 2
Temporizador (tick timer)	Nivel 1

3.2.2.4 Módulos Descriptores de Dispositivos

Módulos descriptores de dispositivos son módulos pequeños no ejecutables que proveen información que asocia un dispositivo específico de E/S con un nombre lógico, dirección de la electrónica del controlador, nombre del manejador de dispositivo, nombre del administrador de archivos y parámetros de inicialización.

Manejadores de dispositivos y administradores de archivos operan sobre grupos generales de dispositivos, no sobre puertos específicos de E/S. Los módulos descriptores de dispositivos ajustan sus funciones a dispositivos específicos de E/S. Debe existir un módulo descriptor de

dispositivo para cada dispositivo de E/S en el sistema. Sin embargo, un dispositivo puede tener varios descriptores de dispositivo con diferentes constantes de inicialización. Para los puertos serie de la PC68K, los nombres de los módulos descriptores de dispositivos son los siguientes:

T1, T2 Para puertos serie de dispositivo MK68564
 T3, T4, T5, T6, T7, T8 Para puertos serie de dispositivos MC68681

El nombre del módulo es utilizado como el nombre lógico del dispositivo por el sistema y el usuario. Su formato consiste de un encabezado estándar de módulo que tiene como tipo de código descriptor de dispositivo (Devic). El resto de los campos del encabezado se muestran en la figura 8 y son descritos a continuación.

Desplazamiento	Utilización
\$30	M\$Port Dirección del puerto
\$34	M\$Vector Número de vector trampa
\$35	M\$IRQLvl Nivel de interrupción IRQ
\$36	M\$Prior Prioridad en escrutinio de IRQ
\$37	M\$Mode Capacidades de modo del dispositivo
\$38	M\$FMgr Desplazamiento del nombre del administrador de archivos
\$3A	M\$PDev Desplazamiento del nombre del manejador de dispositivo
\$3C	M\$DevCon Desplazamiento de configuración del dispositivo
\$3E	Reservado
\$46	M\$Opt Tamaño de tabla de inicialización
\$48	M\$DTyp Tipo de dispositivo

Figura 8. Campos Estandar Adicionales al Encabezado de Módulos Descriptores de Dispositivo.

Nombre	Utilización
M\$Port	Dirección del puerto. Esta es la dirección física absoluta del controlador.
M\$Vector	Vector Trampa. 25 a 31 para autovectores de interrupción. 64 a 255 para vectores de interrupción.
M\$IRQLvl	Nivel de interrupción.
M\$Prior	Prioridad de escrutinio (polling) de interrupción. Números más pequeños son escrutados primero si más de un dispositivo se encuentra en el mismo vector. Una prioridad de cero indica que el dispositivo requiere uso exclusivo del vector.
M\$Mode	Este byte es utilizado para validar el byte de modo de acceso a los llamadores de rutinas I\$Create e I\$Open. Si un bit está prendido, el dispositivo es capaz de llevar a cabo su función solicitada.
M\$FMgr	Desplazamiento del nombre del administrador de archivos. Este es el desplazamiento al nombre del módulo administrador de archivos a utilizar.
M\$PDev	Desplazamiento del nombre del manejador de dispositivo. Este es el desplazamiento al nombre del módulo manejador de dispositivo a utilizar.
M\$DevCon	Configuración del dispositivo. Reservado.
M\$Opt	Tamaño de la tabla. Contiene el tamaño de la tabla de inicialización.
M\$DTyp	Tabla de inicialización. Esta tabla es específica del dispositivo.

La tabla de inicialización es copiada a la sección de opciones de un descriptor de trayectoria cuando una trayectoria al dispositivo es abierta (se explica en la siguiente sección). Los valores de esta tabla pueden usarse para definir los parámetros de operación que son accesibles por las llamadas al sistema I\$Getstat y I\$Setstat. Por ejemplo, los parámetros de inicialización para una terminal define que caracteres de control son usados para borrar un caracter, control de flujo, etc. El tamaño máximo de la tabla de inicialización es de 128 bytes.

Descriptores de dispositivos pueden estar en ROM, en el archivo de

encendido o cargados de disco a memoria mientras el sistema esté corriendo.

3.2.2.5 Descriptores de Trayectorias

Cada trayectoria abierta es representada por una estructura de datos llamada descriptor de trayectoria (PD). Este contiene información requerida por el administrador de archivos y manejador de dispositivos para realizar funciones de E/S. Descriptores de trayectoria son puestos y removidos de memoria en forma dinámica al abrir y cerrar trayectorias.

Descriptores de trayectoria tienen tres secciones: la primera sección es definida en forma universal para todos los administradores de archivo y manejadores de dispositivos como la muestra la figura 9:

	Desplazamiento	Utilización
\$00	PD_PD	Número de trayectoria
\$02	PD_MOD	Modo de acceso (R W E S D)
\$03	PD_CNT	# de Trayectorias usando este PD
\$04	PD_DEV	Dirección de inicio de tabla de dispositivo asociada
\$08	PD_CPR	ID de proceso solicitante
\$0A	PD_RGS	Dirección de la pila de registros del llamador.
\$0E	PD_BUF	Dirección del área de datos (buffer)
\$12	PD_USER	ID de grupo/usuario del propietario original de la trayectoria.
\$16	PD_FST	Area de almacenamiento del administrador de archivos
\$80	PD_OPT	Tabla de opciones

Figura 9. Definiciones Universales para Descriptores de Trayectorias.

La sección llamada "PD_FST" está reservada y definida por cada tipo de administrador de archivo para apuntadores de archivo, variables permanentes, etc.

La sección de 128 bytes llamada "PD_OPT", es usada como área de opciones para alterar dinámicamente parámetros de operación del archivo o dispositivo. Estas variables son inicializadas cuando una trayectoria es abierta y son copiadas de la tabla de inicialización del módulo descriptor de dispositivo, y pueden ser examinadas o modificadas posteriormente con ayuda de las llamadas al sistema GETSTAT y SETSTAT.

4 OS-9 y PC68K

OS-9 se comunica con el mundo exterior a través de los dispositivos de entrada/salida. Como se mencionó en el capítulo sobre sistema operativo OS-9, los manejadores de dispositivos son dependientes de las características físicas y operacionales de los dispositivos. En las próximas líneas se mencionan los manejadores de la tarjeta PC68K que le permiten a ésta por medio de OS-9 comunicarse con el mundo exterior. Algunos de estos dispositivos forman parte de la PC68K, pero otros son de la computadora huésped IBM-PC.

Existen dos clases de manejadores que son distribuidos con el sistema PC68K y OS-9.

Manejadores de tipo 1

Manejadores de tipo 1 comparten dispositivos de la computadora huésped IBM-PC, y utiliza la memoria de doble puerto MDP para comunicarse con el dispositivo. La siguiente tabla contiene los nombres de los módulos manejadores de dispositivos de este tipo y sus módulos descriptores asociados:

Manejador	Descriptor(es)	Dispositivo
-----	-----	-----
pc68cio	TERM	Consola de E/S
p68fd	DO, D1	Unidad de disco flexible
p68hd	HO, H1	Unidad de disco duro
lptptr	LPT1, LPT2	Impresoras paralelo de PC

Un breve resumen de un manejador que utiliza el MDP es como sigue:

La tarea que se está ejecutando gestiona una solicitud a OS-9.

OS-9 interpreta la solicitud y pasa el control al manejador apropiado o al módulo de soporte al periférico.

El manejador averigua si la MDP está disponible y de ser así, marca a la MDP como ocupada, manda la información requerida a la MDP y genera una interrupción a la computadora huésped PC.

La computadora huésped PC pasa el control a la interfaz PCI que interpreta la solicitud, efectúa las llamadas apropiadas para satisfacer la solicitud y luego genera una interrupción de regreso a la PC68K.

El manejador de la PC68K recibe el control, libera a la MDP, y pasa los datos pertinentes a la tarea que gestionó la solicitud de servicio.

El procedimiento normal para un manejador de OS-9 cuando encuentra el MDP ocupado, es ponerse a dormir, permitiendo así que otras tareas que no requieren el MDP se ejecuten. Cuando se recibe un reconocimiento de interrupción de la IBM-PC, las tareas que estaban esperando en PC68K son despertadas para que terminen su procesado. Los manejadores que nos

interesan son los de tipo 2 y en particular los que manejan los puertos serie.

Manejadores de tipo 2.

Manejadores de tipo 2 utilizan dispositivos en la tarjeta PC68K y no usan la memoria MDP. Estos utilizan protocolos normales de manejadores de OS-9. La siguiente tabla contiene los nombres de los módulos manejadores de dispositivos de este tipo y sus módulos descriptores asociados:

Manejador	Descriptor(es)	Dispositivo
-----	-----	-----
sc68564	T1, T2	Puertos serie
sc68681	T3, T4, T5, T6, T7, T8	Puertos serie
p68clk		Reloj
p68pptr	P	Impresora paralelo

Estos módulos manejadores al igual que los de tipo 1, son proporcionados por el fabricante de la tarjeta PC68K. En el manual "68000 Coprocessor Board. Installation and Technical Reference" de Emerald Computer Inc., viene una descripción de estos manejadores así como el mapeo de los registros de los dispositivos a memoria del 68000, niveles y vectores de interrupción asociadas y otros aspectos operacionales relacionados con los dispositivos para los dos tipos de manejadores.

Como se mencionó en el capítulo 2, referente a la descripción del equipo, los 8 puertos serie que contienen la PC68K junto con la tarjeta de expansión, involucran a 2 diferentes tipos de dispositivos: el MC68564 y el MC68681. El primero es un dispositivo de E/S serie de doble canal

capaz de manejar protocolos síncronos y asíncronos. Existe uno de estos en la tarjeta principal que incorpora 2 puertos serie al sistema, cuyos módulos descriptores son T1 y T2. El manejador para estos puertos es el módulo SC68564.

El segundo tipo de dispositivos es el MC68681, el cual es un receptor/transmisor universal asíncrono de 2 canales (DUART). De éstos, existen 3 en la tarjeta de expansión los cuales incorporan 6 puertos serie (2 por dispositivo) adicionales al sistema PC68K. Los módulos descriptores para estos puertos son T3, T4, T5, T6, T7 Y T8. El manejador para estos dispositivos es el SC68681.

Después de haber efectuado unas pruebas preliminares, nos dimos cuenta que los manejadores proporcionados (SC68564 y SC68681) no satisfacían los requerimientos de la aplicación. Se sometió el sistema a capturar información sísmica continua a través de varios puertos serie. De los pruebas se detectó la pérdida de información y el administrador de archivos de caracteres secuenciales SCF reportaba errores de los cuales no se puede tener control ni corregir. Se presentó la misma situación que ocurrió cuando se estaba desarrollando el sistema de captura y detección alrededor del equipo PRIME 350 (Duarte C. et al 1981).

Con ayuda de un depurador, se desensambló los módulos de los manejadores y se encontró mucho código redundante que consumía mucho tiempo para nuestra aplicación. También se observó que los manejadores, en sus áreas de almacenamiento estático (una por puerto serie), retienen los caracteres de

entrada (medias muestras) en una sección muy reducida de memoria (80 caracteres). Si los caracteres de entrada no son leídos o copiados a otro lado lo suficientemente rápido, estas secciones reducidas (colas) se llenan, produciendo así la pérdida de información. Además, sobre estos datos de entrada que se albergan temporalmente en las pequeñas secciones de memoria de los manejadores, no se puede operar directamente con ellos. Es necesario copiarlos a otra área de usuario para poder operar sobre ellos. Esto es muy costoso para nuestra aplicación, ya que la mayoría de los datos de entrada se desechan.

Por las deficiencias encontradas en las pruebas preliminares, fue necesario desarrollar nuestros propios manejadores de dispositivos. Esta acción también tuvo que tomarse en el desarrollo del sistema de captura y detección con el equipo PRIME (Hinojosa A. et al 1983). Las características generales de los manejadores a implementar fueron las siguientes:

- Código rápido, eficiente y específico para la aplicación.
- Capaz de efectuar edición (reformatar) en las muestras recibidas,
- Una área de almacenamiento temporal para los datos de entrada de 4 Kbytes aproximadamente. Existe una área por estación o puerto serie.
- Esta área podrá ser accesada directamente y en forma concurrente por otros procesos.

-Dadas las facilidades de OS-9, el código de los manejadores deberá ser re-entrante. Es decir, solo habrá una copia del manejador para todos los dispositivos que sean idénticos (MC68681 y MC68564).

En el capítulo 5 se plantean las especificaciones y diseño del sistema donde algunas de las características se trasladan directamente a los manejadores. En el capítulo 6, se describe la implementación de los manejadores y del sistema de captura en su conjunto. Los nombres que asigné a los módulos de los nuevos manejadores son: ma68681 y ma68564. También generé unos nuevos descriptores de dispositivos, que especifican las características de operación de las señales digitales provenientes de las estaciones de RESNOR como: número de bits de alto, número de bits por caracter, baud rate y otros parámetros muy específicos de la aplicación. Los nombres que asigné a los descriptores son: EST1, EST2, EST3, EST4, EST5, EST6, EST7, EST8 y RELOJ.

Los manejadores de los puertos serie (SC68681 y SC68564) proporcionados por el fabricante, se comportan satisfactoriamente cuando le dan servicio a terminales en sesiones interactivas (multiusuarios).

Parámetros operacionales de los puertos serie como bits de alto, número de bits de datos, tasa de señalización (baud rate), etc. pueden ser alterados dinámicamente por medio de los mandos de OS-9: TMODE y XMODE. Estos mandos modifican la tabla de opciones en los módulos descriptores de dispositivos que a su vez afectan los descriptores de trayectorias abiertas. Para mayores detalles sobre estos mandos, consultar el manual de Microware "Using Professional OS-9".

5 DISEÑO DEL SISTEMA DE CAPTURA

5.1 Especificaciones

5.1.1 Objetivo

El objetivo del sistema de captura y procesamiento de datos es detectar aquellas señales que, provenientes de la red telemétrica digital RESNOR, se identifiquen como eventos sísmicos para almacenarlos en forma apropiada y efectuar su posterior procesamiento.

5.1.2 Entrada

A cada puerto serie del sistema, entra información sísmica digital de una estación sismológica. La tarjeta coprocesador PC68K cuenta con 8 puertos serie, por lo tanto tiene una capacidad máxima para 8 estaciones por tarjeta PC68K. Como comentario, el fabricante de la tarjeta coprocesador PC68K especifica que hasta 16 tarjetas PC68K pueden correr en forma concurrente desde un sólo ducto de IBM PC/XT/AT. Cada estación tiene 3 sensores (1 vertical y 2 horizontales) los cuales son muestreados 40 veces por segundo cada sensor. La información de estos tres sensores se transmite en forma multicanalizada.

Los datos digitales (muestras) que transmiten las estaciones telemétricas son números enteros de 12 bits con signo, expresados en código binario exceso 2048. Por lo tanto, tenemos que la codificación en binario es la siguiente:

0000 0000 0000 representa el valor mas pósitoivo
 1000 0000 0000 representa el valor 0
 1111 1111 1111 representa el valor más negativo

Cada muestra está codificada en 2 bytes que contienen 8 bits cada uno. Un byte contiene únicamente información parcial relativa al valor de la muestra. El otro tiene agregado un código que identifica al canal (componente) y contiene el resto de la información de la muestra.

El formato de los datos de entrada es la siguiente:

byte1	byte2	byte3	byte4	byte5	byte6	byte7	...
LSBCH1	MSBCH1	LSBCH2	MSBCH2	LSBCH3	MSBCH3	LSBCH1	...

El byte1 es el primero en entrar a la computadora. El byte menos significativo (LSBCHx) y el más significativo (MSBCHx) de cada muestra contienen la siguiente información:

Byte menos significativo.

D6 D5 D4 D3 D2 D1 D0 1
 1 identifica al LSByte

El byte más significativo.

ID1 D11 D10 D9 D8 D7 ID2 0
 0 identifica al MSByte

D11 es el bit más significativo y D0 es el menos significativo

ID2, ID1 identifican el canal	ID1	ID2	CANAL	COMPONENTE
Horizontal	0	0	1	Norte-Sur
Vertical	0	1	2	Vertical
Horizontal	1	0	3	Este-Oeste

- Cada byte es recibido en forma estandar asincrona (LSB primero, MSB despues) palabras de 10 bits, que contiene:
 - bit de inicio
 - 8 bits de datos
 - Sin paridad
 - 1 bit de alto

La entrada es continua y constante a una velocidad de 2400 bits por segundo. La entrada de los datos es asincrona con la norma RS-232.

5.1.3 Salida

La salida son archivos salvados en medio masivo de almacenamiento (disco duro) que contienen la información sísmica que ha sido detectada como evento sísmico. Se grabará un archivo por estación y por evento detectado. Es decir, cada vez que el algoritmo de detección declara que algo parecido a un evento sísmico está ocurriendo, se generan un conjunto de archivos (uno por estación) que contienen la información de preventivo y el evento sísmico. Independientemente de que estaciones declaren el evento, se graba la información de todas las estaciones. El nombre de cada archivo, asocia la información que contiene con la estación a la que pertenece y una fecha y hora aproximada de la ocurrencia. Por ejemplo, un archivo con nombre:

ENX_1028871121

Pertenece a la estación ENX_ (Ensenada) y contiene información ocurrida en el mes 10, día 28, del año 87 a las 11 horas con 21 minutos.

Con el objeto de optimizar el espacio de almacenamiento, la información es grabada en formato binario. Debido a que la información sísmica que arriba al CICESE no tiene alguna referencia con respecto al tiempo, se incluirán marcas de tiempo a la información sísmica con el objeto que ubicar precisamente en el tiempo la ocurrencia de eventos sísmicos. La base de tiempo deberá estar sincronizada con el Tiempo Universal Coordinado (UTC) (Favela F. 1984).

5.2 Algoritmo de Detección

Los datos que llegan al sistema se prueban para determinar si son eventos sísmicos o si son solamente ruido. El método para efectuar la detección es conocido como algoritmo de "promedios cortos contra promedios largos" y en términos generales consiste en observar cuando la amplitud de la señal recibida (promedio corto) sobrepasa por un cierto umbral al promedio del ruido ambiental (promedio largo). En otras palabras, se compara el promedio de la magnitud de la señal (valor absoluto) evaluada en un tiempo corto, con el promedio de la magnitud evaluada en un tiempo largo. El uso de promedios, en vez de las señales instantáneas tiene el objeto de insensibilizar la detección a variaciones muy rápidas en la señal de entrada.

Este algoritmo se aplica a las trazas que entran al sistema de adquisición

en una de sus tres componentes (seleccionable). Si el umbral ha sido sobrepasado por las estaciones definidas en la función de detección, se procede a declarar la ocurrencia de un evento sísmico (condición de disparo). Una vez que se decidió que se satisface la condición de disparo, se procede a almacenar los datos en disco. Periódicamente se vuelve al proceso de detección para investigar hasta cuando se continúa con el almacenamiento de los datos.

La función de detección es una función booleana, que puede ser configurable por el administrador del sistema. Esta función permite especificar situaciones bajo las cuales se inicie el proceso de grabación (condición de disparo). En la sección de "Módulo de detección grabación" se explica con más detalle la mencionada función, pero para darles un ejemplo, esta función se podría configurar de tal manera que se iniciara la grabación de información cuando las estaciones que están involucradas en la función de detección sobrepasen el umbral y además satisfagan la siguiente relación booleana:

$$\text{Disparo} = (\text{ST1}) \text{ o } (\text{ST2 y ST3}) \text{ o } (\text{ST2 y ST5 y ST6})$$

Esto quiere decir que Disparo va a ser verdadera si:

-En la estación ST1, el cociente del promedio corto sobre el promedio largo fue mayor que el umbral ($\text{umbral} < \text{pcorto}/\text{plargo}$),

-O si en la estación ST2 y en la ST3 el $\text{umbral} < \text{pcorto}/\text{plargo}$

-O si en la estación ST2 y en la ST5 y en la ST6 el $\text{umbral} < \text{pcorto}/\text{plargo}$

Esta función implica que si hay 8 estaciones, las estaciones ST8, ST7 y ST4 no se están considerando en nuestra relación booleana. También implica que todas las otras situaciones que se presenten no van a ser consideradas, por ejemplo cuando en la estación ST3 y en la ST5 el umbral <pcorto/plargo. La razón de esta función de detección es darle flexibilidad a la condición de disparo, ya que se monitorean varias estaciones y puede haber casos en los que queramos observar con mayor atención una estación o deshabilitar otra que no esté funcionando correctamente. Esta herramienta es nueva, ya que no está implementada en el sistema PRIME 350.

Los tamaños de las ventanas de tiempo en el promedio corto, promedio largo y el valor del umbral de disparo son seleccionables por el usuario, pero se ha observado del sistema actual (PRIME 350) que valores de 1.6 segundos, 1.7 minutos y 3 respectivamente han tenido resultados satisfactorios. (valores por omisión)

Se hicieron dos variantes a este algoritmo, sugeridas por la experiencia adquirida con el sistema que actualmente esta en operación. Primero, el algoritmo no se aplica a la señal, sino a la primera diferencia de la señal. Esto es con el objeto de aproximar la señal a su derivada y así insensibilizar al algoritmo de la componente de frecuencia cero, conocida en instrumentación como componente DC que ocasionalmente tiene deriva en las estaciones de RESNOR. Segundo, el promedio largo es un promedio de promedios cortos.

5.3 Procesos Principales

Desde un punto de vista lógico, el sistema de captura siempre se encuentra detectando eventos de la información sísmica que recibe o grabando la información detectada. Podemos decir que se encuentra en uno de dos estados: detección o grabación. El hecho de que se encuentre en uno de los dos estados, implica un manejo diferente de variables de estado del sistema (se analiza más adelante).

El sistema está compuesto de módulos de programas (procesos) y módulos de datos. Los módulos de programas se ejecutan en forma "simultánea" y operan concurrentemente sobre los módulos de datos, los cuales son compartidos por los procesos. El sistema operativo OS-9 con el que se implementó el sistema, es un sistema de multitareas que permite que varios procesos compartan el procesador 68000, y se ejecuten en forma "simultánea". Además permite que módulos de datos y de procesos sean compartidos. Existen 3 procesos principales que siempre se están ejecutando dentro de ciclos infinitos. Estos son:

- 1.-Adquisición y reformatización
- 2.-Estampado de marcas de tiempo
- 3.-Detección-grabación

5.3.1 Módulo Adquisición y Reformatización

La función de este módulo es capturar los datos que se reciben en forma continua de las estaciones telemétricas a través de los puertos serie. Capturado el dato, hay que reformatizarlo de tal manera que nos permita realizar fácilmente operaciones aritméticas con ellos (se preparan los datos para módulo de detección). Una vez capturado y reformatizado el dato, se escribe en una área de almacenamiento temporal. Esta área está implementada en una estructura de datos conocida como cola con representación secuencial (alias buffer circular o cola circular). Esta estructura de datos está contenida dentro del módulo de datos que es compartido con los otros dos procesos. Hay una cola por puerto serie. Esta estructura de datos nos permite retener información de pre-evento y el tamaño de éstas, determina que tanta información de pre-evento puede almacenarse. Se puede ver como una ventana corrida de tiempo que almacena la señal sísmica más recientemente recibida, que será sobrescrita por futura información en caso de no ser de utilidad o será transferida a disco en caso de contener un evento sísmico. El tamaño de las colas así como su manejo se explican más adelante en la sección de estructuras de datos de apoyo.

Este proceso decodifica la información tal y como llega de las estaciones. Tiene la lógica necesaria para quitar bits de utilería y etiquetar los tres diferentes canales durante la reformatización. Debido al orden en que llega la muestra: LSB primero y MSB después, el llenado de las colas tendrá que contemplar esta situación. Como se mencionó anteriormente, los

datos llegan con el siguiente formato:

+ reciente

```
+-----+
| LSB1 | MSB1 | LSB2 | MSB2 | LSB3 | MSB3 | LSB1 | MSB1 | ....
+-----+
```

FORMATO DE LA MUESTRA ES:

```
+-----+-----+
|           LSB           |           MSB           |
+-----+-----+
| D6 D5 D4 D3 D2 D1 D0 1 | ID1 D11 D10 D9 D8 D7 ID2 0 |
+-----+-----+
```

ID1 ID2

```
-----
0 0  CANAL 1  Norte-Sur
0 1  CANAL 2  Vertical
1 0  CANAL 3  Este-Oeste
```

La reformatización de las muestras consiste en transformar esas mismas muestras a otro formato que permita realizar operaciones aritméticas fácilmente. El resultado es una muestra que se inicia en una dirección par y tiene el siguiente formato generalizado:

```
+-----+
| I3 I2 I1 I0 | D11 D10 D9 D8 D7 D6 D5 D4 D3 D2 D1 D0 |
+-----+
```

Los símbolos D11 a D0 son los bits de la muestra RESNOR siendo D11 el bit más significativo y D0 es el bit menos significativo. La identificación de canal se define en la siguiente tabla binaria:

I3	I2	I1	I0	Identificadores
0	0	0	0	Canal 1. Horizontal (Norte-Sur)
1	x	x	x	Marca de tiempo (ver módulo estampado..)
0	1	0	0	Canal 2. Vertical
0	0	0	1	Canal 3. Horizontal (Este-Oeste)

Este proceso se encuentra en un ciclo infinito. En cada ciclo revisa si hay datos en cada uno de los puertos y de existir los lee, reformatiza y escribe en su respectiva cola.

Dada la naturaleza de la información que es capturada, en la cual la mayoría de ella es desechable (no eventos sísmicos), el manejo de las áreas de almacenamiento temporal (las colas) es diferente al acostumbrado para este tipo de estructura de datos. El manejo de éstas depende del estado en que se encuentre el sistema: detección o grabación. Si el sistema se encuentra en estado de detección, la escritura en las colas se hace sin verificar sobreflujo de estas (que se llenen). Si el sistema se encuentra en estado de grabación, la escritura en las colas se hace verificando sobreflujo, por lo tanto la lectura (vaciado) de las colas tiene que ser más rápida que la escritura (el llenado). También el tamaño de éstas tiene que ser lo suficientemente grande que permita al procedimiento de vaciado de las colas trabajar con olgura, evitando que éstas se llenen provocando así la pérdida de información.

5.3.2 Módulo de Estampado de Marcas de Tiempo

La función de este proceso es incluir marcas de tiempo a la información sísmica. Cada segundo se incluyen estampas de tiempo que se mezclan con la información sísmica en el área de almacenamiento temporal (colas).

La información de tiempo se obtiene de un reloj que está sincronizado con el tiempo universal coordinado. Un radio-reloj muy preciso (Favela F. 1984) está conectado a uno de los puertos serie. Esto implica sacrificar un puerto destinado a una estación para conectar el reloj. Este radio-reloj decodifica la señal de radio WWVB transmitida por el National Bureau of Standards (NBS) desde Fort Collins, Colorado, E.U.A. La decodificación de esta señal permite sincronizar el reloj con el tiempo universal coordinado. Este reloj manda información muy exacta de la hora y fecha en tiempo real. Esta información está en ASCII y es una sarta de 16 caracteres que tienen el siguiente orden:

```
CTRL(A) DDD:HH:MM:SS Q (CR) (LF)
  0      123456789012 3 4    5    . caracter número
```

Donde CTRL(A) = inicio de información, DDD = día juliano,
: = dos puntos, HH = horas del día, MM = minutos,
SS = segundos; Q = Calidad de la hora proporcionada,
CR = regreso de carro, LF = nueva línea.

y se transmite cada segundo por el reloj. Cada vez que se recibe el CR, indica que es el tiempo exacto de la información transmitida por el reloj.

El símbolo Q trae información sobre la última actualización con la señal

WWVB que repercute directamente en la calidad del tiempo proporcionado por el reloj. El reloj manda símbolos ASCII que representan la precisión de la hora. Estos caracteres se transforman a otro código más comprimido que se establece en en la siguiente tabla:

Código 3 Bits	Símbolo ASCII 8 bits	
000	Espacio	posibilidad de error menor de 1 miliseg.
001	.	posibilidad de error de + o - 1 miliseg.
010	*	posibilidad de error de + o - 5 miliseg.
011	#	posibilidad de error de + o - 50 miliseg.
100	?	posibilidad de error de + o - 500 miliseg.
101	no definido	Algo anda mal. pues solo están definidos los símbolos anteriores

El resultado del proceso de estampado es convertir 16 caracteres ASCII que manda el reloj, a dos marcas de tiempo binarias (de 16 bits cada una) que estarán mezcladas con información sísmica. Es decir, contiene la misma información pero en forma comprimida. Estas estampas siempre se encuentran una junto a la otra. El formato de las dos estampas de tiempo en octal es el siguiente:

1DDDDH

1QMMSS

Las estampas se insertan cada vez que se detecta un (CR) y se actualiza cada segundo. Por lo tanto deben ocurrir las marcas de tiempo cada 120 muestras de información sísmica.

Donde DDD = día juliano, HH= horas del día, Q = Calidad de información del reloj, MM = Minutos, SS = Segundos.

El 1 al principio significa que es marca de tiempo y esto lo diferencia de la información sísmica. La información sísmica tiene otro formato.

El resultado es que tendremos en el área de almacenamiento temporal (cola) información sísmica y de tiempo mezcladas. Por la frecuencia con que se ponen las marcas y la tasa constante de entrada de los datos sísmicos, esperaremos encontrar marcas de tiempo cada 120 muestras de información sísmica (de los tres canales).

Como se mencionó en la descripción del equipo, la tarjeta coprocesador PC68K tiene un reloj interno. La razón por la cual no se utilizó este reloj, es que no está sincronizado con el tiempo universal coordinado.

5.3.3 Módulo de Detección-Grabación.

La función de este módulo es aplicar el algoritmo de detección a las trazas sísmicas en la componente seleccionada y determinar si ha ocurrido un evento sísmico. Este proceso opera sobre los datos que han sido capturados por el proceso de adquisición reformatización y que correspondan a las estaciones involucradas en la función de detección. Trabaja por medio de ventanas corridas de tiempo de diferentes tamaños que ven a diferentes parámetros del algoritmo de detección.

Este proceso se encuentra en un ciclo infinito, dentro del cual puede estar detectando eventos sísmicos o grabando información detectada. Cuando está en detección, éste proceso se activa cada intervalo de tiempo especificado por el promedio corto. Al activarse, primero calcula para cada estación involucrada en la función de detección el promedio de las primeras diferencias. Utiliza las muestras del canal seleccionado más

recientemente recibidas. La muestra más recientemente recibida estará apuntada por el apuntador ATRAS de cada cola (se explica con más detalle en la sección de estructura de datos de apoyo).

El número de muestras a considerar en el promedio corto están implícitas en el intervalo de la ventana del promedio corto. El valor que se utiliza es 65 muestras equivalente a 1.6 segundos aproximadamente. Este parámetro es global; el mismo para todas las estaciones. La fórmula generalizada para calcular el promedio corto en cada estación es la siguiente:

$$\frac{1}{P_{\text{corto}} - 1} \sum_{i=2}^{P_{\text{corto}}} |X_i - X_{i-1}|$$

donde P_{corto} es el número de muestras en promedio corto y X son las muestras del canal seleccionado más recientemente recibidas.

Una vez calculado el promedio corto a cada estación habilitada para detección, se procede a actualizar el promedio largo. Como se mencionó con anterioridad, el promedio largo es un promedio de promedios cortos.

El número de promedios cortos en el promedio largo es configurable. El valor que se utiliza es de 64 promedios cortos en el promedio largo, equivalente a 1.7 minutos aproximadamente (esto es con 65 muestras en el promedio corto). Este parámetro también es global para todas las estaciones.

Los valores asignados a las ventanas del promedio corto y del promedio largo, además de que han dado buenos resultados con el sistema actual (PRIME 350), se escogieron también porque la división que se tiene que llevar a cabo en el cálculo del promedio, se puede realizar aplicando un corrimiento en seis bits a la derecha al resultado de la sumatoria. Este corrimiento es equivalente a dividir entre 64. El corrimiento de un registro se ejecuta más rápidamente que la división. La diferencia en tiempo entre las dos operaciones es del orden 8:1 (Motorola 1985).

Debido a que el promedio largo es un promedio de promedios cortos, se requiere preservar los resultados de los últimos promedios cortos que están contenidos dentro de la ventana del promedio largo. Estos se mantienen en una cola (una por estación). Entonces tenemos dos colas por cada estación. Una que almacena la información sísmica y la otra guarda los promedios cortos. El tamaño de las segundas colas está definido por el número de promedios cortos en el promedio largo. Estas siempre se encuentran llenas. Parte de la inicialización de este módulo consiste en llenar estas colas.

Cada vez que se calcula un nuevo promedio corto, éste último substituye al más antiguo dentro de la ventana (cola). El promedio corto más reciente y el más antiguo se utilizan para actualizar el promedio largo. Este se actualiza en dos pasos. El primer paso se obtiene con la siguiente fórmula:

```
Plargo_nuevo <- Plargo_anterior + Pcorto (t) - Pcorto (-nt)
```

Donde $P_{largo_anterior}$ es la sumatoria de promedios cortos calculada en la iteración anterior, $P_{corto}(t)$ es el promedio corto más recientemente calculado y $P_{corto}(-nt)$ es el promedio corto más antiguo dentro de la ventana del promedio largo. Hay que hacer notar que $P_{largo_anterior}$ y P_{largo_nuevo} son los valores de la sumatoria de promedios cortos antes de efectuar la división. Falta llevar a cabo la división de la sumatoria entre el número de promedios cortos en el promedio largo. Esta se realiza en el segundo paso aplicando un corrimiento de P_{largo_nuevo} en seis posiciones a la derecha (equivalente a dividir entre 64). Antes de ejecutar la división, se salva el P_{largo_nuevo} para que en la próxima iteración se convierta en $P_{largo_anterior}$.

Como se puede observar de la fórmula para estimar el promedio largo, se requiere de un $P_{largo_anterior}$ original. Este se calcula durante la inicialización de este módulo, después de haber llenado la cola de promedios cortos que forman parte de la ventana del promedio largo original.

Teniendo ya el promedio corto (el más reciente) y el promedio largo, se ejecuta el cociente del primero entre el segundo. Recuerde que esto se hace en una ventana de tiempo del tamaño del promedio corto para todas las estaciones que forman parte de la función de detección. Si para una estación del conjunto el resultado del cociente es mayor que el umbral, una variable booleana que corresponde a esa estación y a esa ventana en particular se vuelve verdadera (1), sino permanece falsa (0). En la

figura 10 se muestra para una estación X, el estado de las ventanas y de los parámetros del algoritmo de detección.

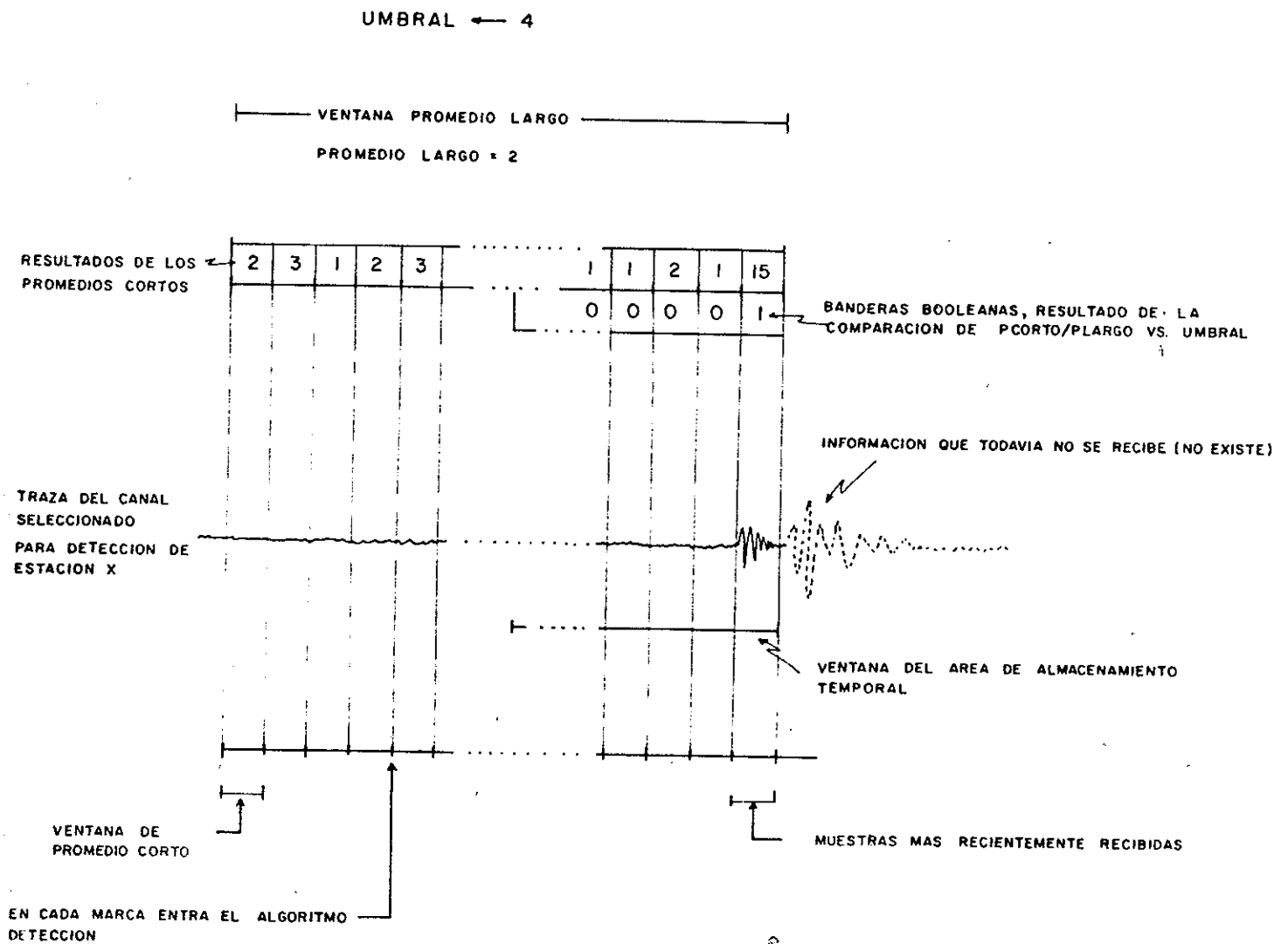


Figura 10. Parámetros del algoritmo de detección para una estación. Se muestra la ventana corrida de promedios cortos y la de resultados de la comparación de pcorto/plargo vs. umbral.

El resultado de las variables booleanas del conjunto de estaciones se guarda en un arreglo de bits (booleano). Este arreglo de bits pertenece a una ventana de promedio corto y al conjunto de estaciones que están incluidas en la función de detección. Debido al carácter regional de la red, un mismo evento puede detectarse por diferentes estaciones con varios segundos de diferencia, dependiendo de la posición relativa del hipocentro y cada una de las estaciones.

Para poder determinar si un mismo evento está siendo reportado por más de una estación, se requiere preservar los resultados booleanos de las previas aplicaciones del algoritmo. Los resultados booleanos representados en un arreglo de bits se guardan en una cola. Esta nueva cola se puede ver nuevamente como una ventana corrida de tiempo que contiene los resultados de las últimas aplicaciones del algoritmo de detección.

Cada vez que se aplica el algoritmo de detección a todas las estaciones incluidas en la función de detección, el resultado obtenido sobrescribe al resultado más antiguo de la cola. Cada elemento de la cola contiene un arreglo booleano correspondiente a una ventana de promedio corto. El tamaño de la cola determinará que tantos resultados previos se preserven. En virtud de que sólo se puede recuperar la información que se encuentra en el área de almacenamiento temporal (cola de datos), el tamaño de la ventana de resultados tendrá que ser menor o igual al equivalente de la ventana de la cola de datos.

A diferencia de las colas de datos y de promedios cortos de las cuales existe una por estación, existe solo una cola de resultados por sistema. En la figura 11 se muestra una gráfica de un evento sísmico en varias estaciones y la ventana corrida (cola) que contiene los resultados de la aplicación del algoritmo de detección. Como se puede ver en la figura, un mismo evento es reportado en diferentes tiempos en distintas estaciones. Por eso es necesario almacenar de alguna forma los resultados de las aplicaciones previas del algoritmo.

Hasta ahora, no hemos visto como evaluar la función de detección para determinar si se satisface la condición de disparo. Para poder evaluarla, primero necesitamos hacer un resumen de lo que ha ocurrido durante el tiempo que abarca la ventana de resultados. Este resumen se obtiene por medio de la aplicación de una operación "O" lógica a todos los arreglos booleanos que se encuentran en la ventana corrida de resultados. Este resumen es un nuevo arreglo booleano que contiene un valor acumulativo de lo que ha ocurrido en cada estación durante el tiempo que abarca a la ventana corrida de resultados.

En el siguiente ejemplo se muestra como se calcula el resumen a partir de los resultados que se encuentran en la cola de la figura 11.

VENTANA CORRIDA DEL AREA DE ALMACENAMIENTO TEMPORAL

VALOR ACUMULATIVO EN LA VENTANA

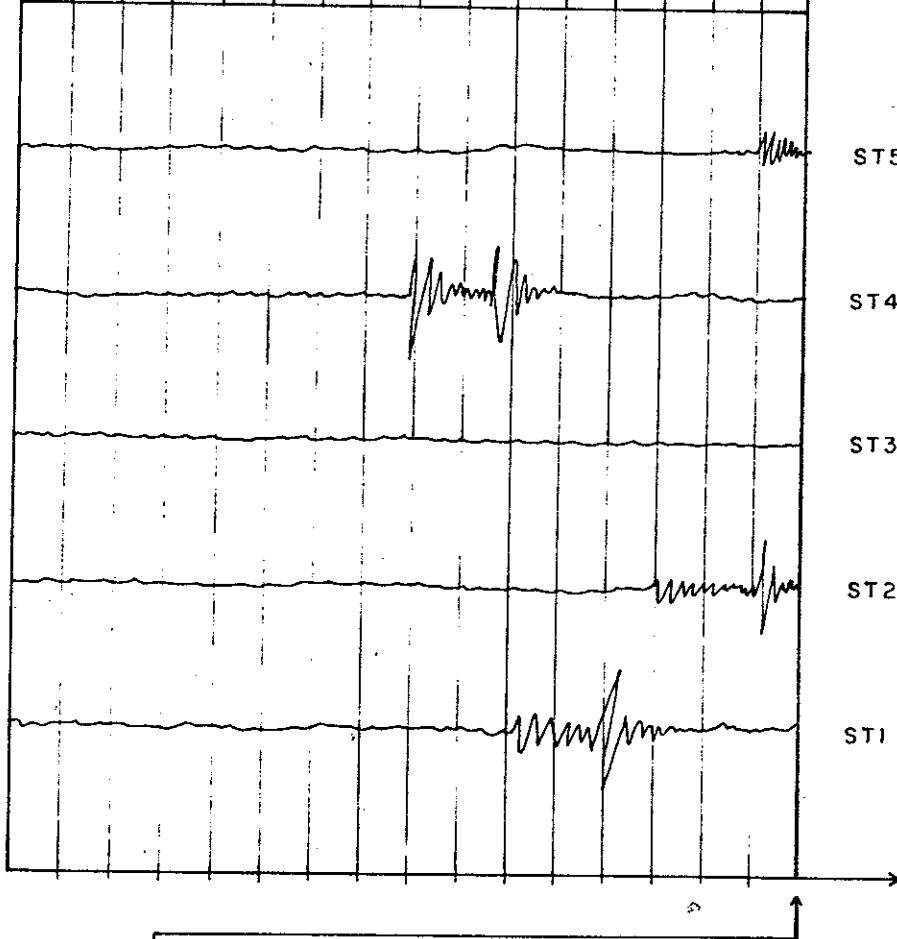
1
1
0
1
1

VENTANA CORRIDA DE LOS RESULTADOS DE LAS COMPARACIONES PCORTO/PLARGO VS. UMBRAL

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0

MAS RECIENTE

ST5 } ESTACIONES INVOLUCRADAS EN FUNCION DE DETECCION
ST4 }
ST3 }
ST2 }
ST1 }



$$\text{DISPARO} = (\text{ST4 Y ST5}) \circ (\text{ST1 Y ST2 Y ST5})$$

Figura 11. Evento sísmico reportado en 4 de las 5 estaciones de la gráfica. En la parte superior de la figura se muestra la ventana corrida de resultados y el valor acumulativo (resumen) de la ventana.

Obviamente, las estaciones que se consideran en el resumen forman parte de la función de detección:

ST5	ST4	ST3	ST2	ST1	
1	0	0	1	0	Arreglo de bits más reciente
0	0	0	1	0	
0	0	0	1	1	
0	0	0	0	1	
0	1	0	0	1	
0	1	0	0	1	
0	1	0	0	0	Operación 0 lógica con todos los arreglos de la ventana
0	1	0	0	0	
0	0	0	0	0	
0	0	0	0	0	
0	0	0	0	0	
0	0	0	0	0	
0	0	0	0	0	
0	0	0	0	0	
0	0	0	0	0	
0	0	0	0	0	
0	0	0	0	0	Arreglo de bits más antiguo

1	1	0	1	1	Resumen de lo ocurrido

Si contamos el número de unos presentes en el resumen, obtenemos el número de estaciones que reportan evento sísmico en la ventana que se está considerando. Este esquema es el que se utiliza en el sistema que opera actualmente alrededor del máquina PRIME 350. Cuando el número de estaciones que reportan evento sísmico es mayor o igual a dos, se procede a grabar la información.

Para el caso de la función de detección, se isola cada uno de los términos de la función y se crea una "máscara" para cada uno de ellos. Con cada máscara se efectua una operación "Y" lógica con el resumen. Si el resultado de la operación es la misma máscara, se satisface la condición de disparo y se pasa al estado de grabación. Se van probando cada uno de

los términos hasta que se satisfaga una de las condiciones de disparo o se agoten los términos. Si se agotaron los términos de la función y ninguno de ellos fue verdadero, se permanece en estado de detección y el proceso duerme hasta que sea el tiempo de aplicar de nuevo el algoritmo de detección. El tiempo para despertar el proceso, es cuando haya transcurrido un tiempo equivalente al promedio corto. Cuando despierta el proceso, se repite todo de nuevo.

Para ilustrar como se generan las máscaras de los términos de la función de detección, supongamos que tenemos la siguiente función:

$$\text{DISPARO} \leftarrow (\text{ST4 y ST5}) \text{ o } (\text{ST1 y ST2 y ST5})$$

Esto quiere decir que DISPARO va a ser verdadero si durante la ventana de resultados, la estación ST4 y la estación ST5 reportan evento sísmico o para el mismo período las estaciones ST1 y ST2 y ST5 reportan evento sísmico. El segundo término de la función (ST1 y ST2 y ST5) generaría la siguiente máscara binaria:

8	7	6	5	4	3	2	1	Estación Número

0	0	0	1	0	0	1	1	Máscara.

Al aplicar la operación "Y" lógica de la máscara con el resumen de la figura 11 tendríamos el siguiente resultado:

8 7 6 5 4 3 2 1	Estación Número

0 0 0 1 1 0 1 1	Resumen
0 0 0 1 0 0 1 1	Máscara

0 0 0 1 0 0 1 1	Resultado

Como el resultado fue igual a la máscara, se satisface uno de los términos de la función, por lo tanto DISPARO es verdadero.

Si este proceso determina que se ha satisfecho la condición de disparo especificada por la función de detección, se procede a transferir la información sísmica que se encuentra en el área de almacenamiento temporal (colas de datos) a archivos de disco.

Al pasar de estado de detección a grabación, lo primero que se hace es definir que estamos en grabación. Esto repercute en que el proceso de adquisición y reformatización empezará a verificar el sobreflujo de las colas de datos. En el cambio de estado de detección a grabación se establece que las colas de datos están a un 75% llenas, con el objeto de recuperar la información de preevento.

La grabación de información sísmica continua hasta que el algoritmo decide que el evento ha terminado. Al ocurrir esto, nuevamente se tiene que definir que regresamos a detección, para que el proceso de adquisición-reformatización deje de considerar el sobreflujo en las colas de datos.

5.4 Estructuras de datos de Apoyo

Para llevar a cabo todas estas tareas y decisiones, los tres procesos mantienen un banco de datos. Este banco de datos es accesado en forma concurrente por los tres procesos y está organizado en varias estructuras de datos. Esta área común de memoria es compartida y actualizada por los tres procesos.

El sistema operativo OS-9 se encarga de arbitrar el acceso al banco de datos con el objeto de que sólo un proceso acceda en un momento dado esta información. En el capítulo del sistema operativo se describe como ocurre la calendarización de los procesos y la prioridad de unos sobre otros.

En la descripción de las estructuras de datos que forman el banco de información, se mencionan los nombres de las variables o estructuras, para que sirva y por que módulos son leídas o modificadas.

5.4.1 ESTADO de Sistema

Esta es una variable booleana que nos indica en que estado se encuentra el sistema. Si la variable es verdadera, el estado es de grabación y si es falsa nos encontramos en detección.

Esta variable es modificada por el módulo de detección-grabación. Al satisfacerse la condición de disparo la variable se vuelve verdadera y se mantiene así mientras que el algoritmo de detección lo deteremine.

Es leída por los módulos de adquisición-reformatización y el de estampado de marcas de tiempo. Dependiendo de su valor estos dos últimos módulos escribirán datos sísmicos y marcas de tiempo verificando o no, el sobreflujo en las colas de datos. Si el estado es de grabación se verifica y si es de detección (falso) no se verifica.

Cuando en una cola ocurre un sobreflujo, quiere decir que está llena y no puede aceptar más datos para almacenar. La razón de las pruebas de sobreflujo, es que mientras nos encontremos en detección, no nos importa que la información que estamos recibiendo se pierda. Por perder quiero decir que sea sobrescrita por información futura sin haberla guardado en alguna otra parte. El hecho de estar en detección implica que la información que estamos recibiendo no es de utilidad.

Por otro lado, si el algoritmo decide que un evento sísmico está ocurriendo, tenemos que salvar la información que se ha retenido en el área de almacenamiento temporal (cola de datos) antes de ser sobrescrita por nuevos datos que se están recibiendo.

5.4.2 Colas de DATOS

Hay una por cada estación (puerto serie). Estas estructuras de datos, que se han mencionado en el texto como áreas de almacenamiento temporal, retienen la información sísmica y las marcas de tiempo más recientemente recibidas. Que tanta información retiene, depende del tamaño asignado

para cada una de ellas. Un elemento de una cola es un entero de 16 bits que puede almacenar una muestra sísmica o una marca de tiempo.

Estas colas tiene representación secuencial y están caracterizadas por: un tamaño, una dirección de inicio, un apuntador ATRAS y otro FRENTE. El tamaño es el mismo para todas y es de 2048 elementos. Esto equivale a 16.5 segundos de información aproximadamente. Esta dimensión es seleccionable por el usuario, pero el valor asignado ha reportado buenos resultados con el sistema actual y la configuración de la red RESNOR.

El inicio de cada cola está definido por un arreglo de apuntadores. Uno por cola. El acceso al arreglo de apuntadores es a través de un índice (1..8) PUERTO, que está determinado por el puerto al que pertenece la información. Cada cola comienza en una dirección par, con el objeto de habilitar al procesador 68000 para que pueda efectuar direccionamiento de palabras y dobles palabras.

Los apuntadores ATRAS (uno por cola) están guardados en un arreglo cuyo acceso es a través del índice PUERTO. Lo que almacenan es el desplazamiento a partir del inicio de la cola, donde se escribirá la siguiente muestra por recibir. Cada vez que se recibe una muestra o marca de tiempo, ATRAS se incrementa en una posición y se guarda en el lugar indicado por este apuntador. Como se puede ver, este apuntador siempre nos indica cual es la muestra más recientemente recibida en la cola. La fórmula generalizada para indicarnos el lugar en que se guardará un muestra recibida será:

```

ATRAS <- (ATRAS + 1) mod TAMANO
LUGAR <- INICIO + ATRAS

```

Los apuntadores FRENTE (uno por cola) también están guardados en un arreglo y su acceso es por medio del índice PUERTO. Lo que almacenan es el desplazamiento a partir del inicio de la cola, donde se encuentra el elemento más antiguo de la cola. Cada vez que se quiere leer un elemento, se lee del lugar indicado por este apuntador. Después de leído el valor, se incrementa en una posición. La fórmula generalizada para indicarnos el lugar en que se encuentra el valor más antiguo será:

```

LUGAR <- INICIO + FRENTE
FRENTE <- (FRENTE + 1) mod TAMANO

```

El sobreflujo en una cola de este tipo se verifica con los apuntadores FRENTE y ATRAS. Al querer escribir una muestra que acabamos de recibir, primero incrementamos ATRAS como se explicó en la fórmula generalizada. Si el nuevo valor de ATRAS es igual a FRENTE, significa que la cola está llena. En estos casos, la muestra que queríamos escribir se pierde. Por eso es importante que la lectura sea más rápida que la escritura para que no ocurran estas situaciones. Esta prueba de cola llena, sólo ocurre cuando el estado es de grabación.

Cuando nos encontramos en estado de grabación, por un lado el módulo de adquisición-reformatización escribe en las colas los datos que se están recibiendo y por el otro lado -el módulo de detección-grabación está vaciando (leyendo) los datos que se encuentran en las colas a archivos en

disco. La lectura se hace de la muestra más antigua (FRENTE) a la más reciente (ATRAS).

En estado de detección, no se hace la verificación de sobreflujo, simplemente se está sobrescribiendo la información contenida en las colas. Cada vez que se recibe una muestra, ésta sobrescribe a la más antigua. El apuntador FRENTE ni se mantiene ni se considera.

En la transición de detección a grabación, el apuntador FRENTE se posiciona de tal manera que establece que la cola se encuentra en un 75% llena. El otro 25%, que son las muestras más antiguas en la cola se pierden. El objetivo de dejar éste 25% del espacio disponible, es reservar algo de espacio para que las muestras que se están recibiendo tengan lugar para almacenarse.

El apuntador ATRAS es modificado (actualizado) por los módulos de adquisición-reformatización y el de estampado de marcas de tiempo. Esto ocurre, cada vez que se escribe una muestra que se recibió de los puertos serie o se escriben las marcas de tiempo. ATRAS es leído pero no modificado por el módulo de detección-grabación y lo utiliza para calcular el promedio corto a las muestras más recientes. ATRAS siempre apunta a la muestra más reciente.

FRENTE es modificado solamente por el módulo de detección-grabación. Esto ocurre al pasar de estado detección a grabación posicionando a FRENTE como se mencionó en los párrafos anteriores. También es modificado al estar

transfiriendo los datos que se encuentran en las colas a archivos de disco. Cada vez que se lee una muestra o marca de tiempo de alguna cola para copiarla en disco, FRENTE es modificado.

Como se puede ver, estas estructuras de datos son las básicas para nuestra aplicación ya que contienen la señal sísmica que estamos probando y en su caso salvando en disco. Además los tres procesos operan sobre ellas en forma intensa y concurrente.

5.4.3 Colas de PROMEDIOS CORTOS

Hay una por cada estación (puerto serie). Estas estructuras de datos, retienen los promedios cortos más recientemente calculados que están contenidos dentro de la ventana de promedio largo. Un elemento de una cola es un entero sin signo de 16 bits que almacena el valor de un promedio corto.

Estas colas tiene representación secuencial y están caracterizadas por: un tamaño, una dirección de inicio y un apuntador ATRAS. El tamaño es el mismo para todas y es de 64 elementos. Esto equivale a 1.7 segundos de información aproximadamente (esto es con un promedio corto de 65 muestras). El valor asignado ha reportado buenos resultados con el sistema actual y la configuración de la red RESNOR.

El inicio de cada cola está definido por un arreglo de apuntadores. Uno por cola. El acceso al arreglo de apuntadores es a través de un índice

(1..8) PUERTO, que está determinado por el puerto al que pertenece la información. Cada cola comienza en una dirección par, con el objeto de habilitar al procesador 68000 para que pueda efectuar direccionamiento de palabras y dobles palabras.

Los apuntadores ATRAS (uno por cola) están guardados en un arreglo cuyo acceso es a través del índice PUERTO. Lo que almacenan es el desplazamiento a partir del inicio de la cola, donde se escribirá el siguiente promedio corto a calcular. Cada vez que se calcula un promedio corto, ATRAS se incrementa en una posición y se guarda en el lugar indicado por este apuntador. Como se puede ver, éste apuntador siempre nos indica cual es el promedio corto más recientemente calculado en la cola. La fórmula generalizada para indicarnos el lugar en que se guardará el siguiente elemento será:

$$\begin{aligned} \text{ATRAS} &\leftarrow (\text{ATRAS} + 1) \bmod \text{TAMANO} \\ \text{LUGAR} &\leftarrow \text{INICIO} + \text{ATRAS} \end{aligned}$$

Estas colas después de ser inicializadas, siempre se mantienen llenas. Esto implica que cada vez que se escribe un promedio corto (el más reciente) en la cola, éste sobrescribe al más antiguo. Debido a que el cálculo del promedio largo utiliza el promedio corto más reciente y al más antiguo, no se requiere de otro apuntador que nos indique cual es el más antiguo, puesto que las colas siempre se mantienen llenas. Ambos promedios se pueden acceder por medio del apuntador ATRAS. Antes de ser sobrescrito el más antiguo por el más reciente, se utilizan ambos valores para calcular el promedio largo con la fórmula anteriormente mencionada:

$$Plargo_nuevo \leftarrow Plargo_anterior + Pcorto(t) - Pcorto(-nt)$$

Donde $Plargo_anterior$ es la sumatoria de promedios cortos calculada en la iteración anterior, $Pcorto(t)$ es el promedio corto más recientemente calculado y $Pcorto(-nt)$ es el promedio corto más antiguo dentro de la ventana del promedio largo. Una vez calculada la nueva sumatoria ($Plargo_nuevo$), el promedio corto más reciente sobrescribe al más antiguo.

La actualización de los elementos en las colas la efectúa el módulo de detección-grabación, y solo actualiza las colas de las estaciones que están involucradas en la función de detección. La actualización ocurre cada intervalo de tiempo especificado por el promedio corto.

5.4.4 Tabla de PROMEDIOS LARGOS

Esta es una tabla almacenada en un arreglo cuyo acceso es a través del índice PUERTO. El arreglo tiene 8 elementos. A cada elemento del arreglo está asociada una estación sismológica (puerto serie). Cada celda de la tabla es un entero sin signo de 32 bits y lo que almacenan es la sumatoria de promedios cortos que están contenidos dentro de la ventana del promedio largo.

La actualización de los elementos en la tabla la efectúa el módulo de detección-grabación, y solo actualiza las celdas de las estaciones que

están involucradas en la función de detección. La actualización ocurre cada intervalo de tiempo especificado por el promedio corto. La forma como actualiza cada elemento está especificada en la sección anterior.

En la fase de inicialización del módulo de detección-grabación, se genera la primera sumatoria de los promedios cortos para todas las estaciones, con el objeto de que funcione nuestra fórmula recurrente.

5.4.5 FUNCION DE DETECCION

Esta es una tabla almacenada en un arreglo de 8 elementos. Cada elemento es un arreglo de 8 bits (un byte) y almacena un término de la función de detección. Esto quiere decir que la función de detección aceptará un máximo de 8 términos. Cada término está representado con una máscara.

Existe un mapeo entre la estación número (PUERTO) y el bit número de un elemento del arreglo. El bit menos significativo del byte está asociado con la estación conectada al puerto serie 1. El bit más significativo (8) está asociado a la estación conectada al puerto 8.

Supongamos que se especifica la siguiente función de detección:

DISPARO \leftarrow (ST7 Y ST8) O (ST1 Y ST2 Y ST3) O (ST4 Y ST5 Y ST6)

Esta función cuenta con 3 términos y la generación de las máscaras sería la siguiente:

TABLA	MASCARAS								TERMINO
	Bit Número								
	8	7	6	5	4	3	2	1	
[1]	1	1	0	0	0	0	0	0	(ST7 y ST8)
[2]	0	0	0	0	0	1	1	1	(ST1 y ST2 y ST3)
[3]	0	0	1	1	1	0	0	0	(ST4 y ST5 y ST6)
[4]	0	0	0	0	0	0	0	0	
[5]	0	0	0	0	0	0	0	0	
[6]	0	0	0	0	0	0	0	0	
[7]	0	0	0	0	0	0	0	0	
[8]	0	0	0	0	0	0	0	0	
	1	1	1	1	1	1	1	1	Acumulativo

Si se efectúa una operación "0" lógica con todos los elementos de la tabla, el resultado acumulativo contendrá a todas las estaciones que están involucradas en todos los términos de la función de detección. En este ejemplo, son las 8 estaciones.

La modificación de la tabla generalmente se hace durante la inicialización del módulo de detección-grabación, pero se desarrollarán futuras herramientas (programa MONITOR) para modificarla en forma dinámica.

La lectura de esta tabla es realizada por el módulo de detección-grabación. Se efectúa después de haber aplicado el algoritmo de detección y haber calculado el resumen de lo ocurrido en la ventana de resultados con las estaciones involucradas en función de detección. La lectura se hace en forma secuencial y termina cuando ocurren cualquiera de las tres siguientes situaciones:

- Se satisfizo la condición de disparo con uno de los términos de la función (ocurre transición de detección a grabación).

-El término de prueba es igual a cero (ya no hay más términos)

-Se llegó al final de la tabla.

5.4.6 Cola de RESULTADOS

Hay una cola por sistema. Esta estructura de datos retienen los resultados obtenidos de las aplicaciones previas del algoritmo de detección a las estaciones involucradas en la función de detección. Nuevamente, esta cola es la implementación de una ventana corrida de tiempo, que en su conjunto contiene un resumen de lo que ha ocurrido con el algoritmo durante el período de tiempo que la ventana abarca. El funcionamiento y mantenimiento de esta estructura de datos está explicado en la sección de módulo de detección y grabación. En esta sección se menciona su implementación y caracterización.

Un elemento de la cola es un byte que almacena un arreglo booleano de 8 bits. Un arreglo booleano contiene el resultado de la aplicación del algoritmo de detección en una ventana de promedio corto.

Esta cola tiene representación secuencial y esta caracterizada por: un tamaño; una dirección de inicio y un apuntador ATRAS. El apuntador ATRAS almacena el desplazamiento a partir del inicio de la cola, donde se escribirá el siguiente arreglo booleano. La fórmula generalizada para indicarnos el lugar donde se guardará el siguiente elemento será:

```

ATRAS <- (ATRAS + 1) mod TAMANO
LUGAR <- INICIO + ATRAS

```

El tamaño de la cola determinará que tantos resultados previos se preserven. En virtud de que sólo se puede recuperar la información que se encuentra en el área de almacenamiento temporal (cola de datos), el tamaño de la ventana de resultados tendrá que ser menor o igual al equivalente de la ventana de la cola de datos.

A la cola de datos se le asignó una dimensión de 2048 muestras, que puede retener 16.5 segundos de señal sísmica aproximadamente. Cada elemento de la cola de resultados equivale a una ventana de promedio corto. Entonces tenemos que la dimensión de la cola de resultados tendrá que ser menor o igual a:

$$\frac{16.5 \text{ segundos en ventana de datos}}{1.6 \text{ segundos en ventana de promedio corto}} \sim 10$$

Esta cola siempre se mantiene llena. La actualización de los elementos en la cola la efectúa el módulo de detección-grabación. Esto ocurre cada intervalo de tiempo especificado por el promedio corto, después de haber aplicado el algoritmo de detección a todas las estaciones que están contenidas en la función de detección.

Un elemento de la cola no tiene sentido por sí sólo, sino el conjunto de todos arreglos booleanos almacenados. Como se mencionó en la sección del módulo de detección-grabación, se obtiene un resumen a partir de la

aplicación de la operación "O" lógica con todos los elementos de la cola.
Este resumen es comparado con cada uno de los términos de la función de
detección para determinar si se satisface la condición de disparo.

6 IMPLEMENTACION

La implementación se hizo en el ambiente provisto por la conjunción de la tarjeta coprocesador PC68K y por el sistema operativo OS-9. Los lenguajes que se utilizaron fueron "C" y ensamblador para el 68000.

6.1 Módulo de DATOS Compartido.

La clave en la implementación está en la coexistencia de varios procesos que se ejecutan en forma "simultánea" (multitareas) y que comparten un mismo módulo de datos que lo accesan en forma concurrente.

El módulo de datos es creado al inicializar el sistema y permanece residente en memoria mientras el sistema esté corriendo. Existe un archivo con nombre "Startup" que al inicializar el sistema, ejecuta todas las directrices que en él existen. Una de ellas es invocar un programa en "C" que lo único que hace es crear el módulo de datos con la dimensión especificada. Este módulo lleva el nombre de "DATOS". El módulo de datos es creado a través de una llamada al sistema F\$DatMod. Es indispensable que exista éste módulo en memoria para cubrir uno de los requisitos de la implementación. Se puede verificar la existencia del módulo por medio del comando de OS-9: mdir -e (directorio de memoria con opcion extendida).

Una vez que tenemos el módulo de DATOS en memoria, otros procesos pueden enlazarse a él con la solicitud de servicio al sistema F\$Link incrementando así el contador de enlaces del módulo. F\$Link nos regresa

la dirección del inicio del módulo en memoria.

Si recuerda del capítulo 3, todos los módulos tienen un encabezado. El acceso al área disponible de memoria del módulo (estructuras de datos de la aplicación) es por medio de la suma de la dirección del inicio del módulo más el contenido del campo M\$Exec (\$30) del encabezado.

Al crear el módulo de DATOS con la llamada F\$DatMod, se especifica la dimensión requerida en bytes. El módulo de datos alberga las siguientes estructuras de datos del sistema:

- ESTADO. Como se definió en la sección 5.4.1, ésta variable booleana nos indica si el sistema se encuentra en detección (0) o en grabación (1). El valor de ésta variable repercute directamente en la forma de manejar las colas. Esta variable ocupa 2 bytes.
- Indices ATRAS de las colas de datos. Existe uno por cada puerto (cola de datos). Están conformadas en un arreglo de 8 elementos. Este índice sumado a la dirección de inicio de su respectiva cola, nos indica el lugar donde se escribirá la siguiente muestra a recibir para un puerto en particular. Cada índice ATRAS ocupa 2 bytes.
- Indices FRENTE de las colas de datos. Existe uno por cada puerto (cola de datos). Están conformadas en un arreglo de 8 elementos. Este índice sumado a la dirección de inicio de su respectiva cola, nos indica el lugar donde se encuentra la próxima muestra a leer en una cola de datos en particular. Cada índice FRENTE ocupa 2 bytes.
- Colas de datos. Existe uno por cada puerto. En esta primera versión del sistema, todas las colas son del mismo tamaño. El tamaño es de 4096 bytes (2048 muestras) que equivale a aproximadamente 16.5 segundos de información sísmica.

Si sumamos las dimensiones de las varias estructuras de datos, tenemos que en su conjunto el módulo de datos ocupa un poco más de 32K bytes. La figura 12 nos muestra la organización impuesta al módulo de datos.

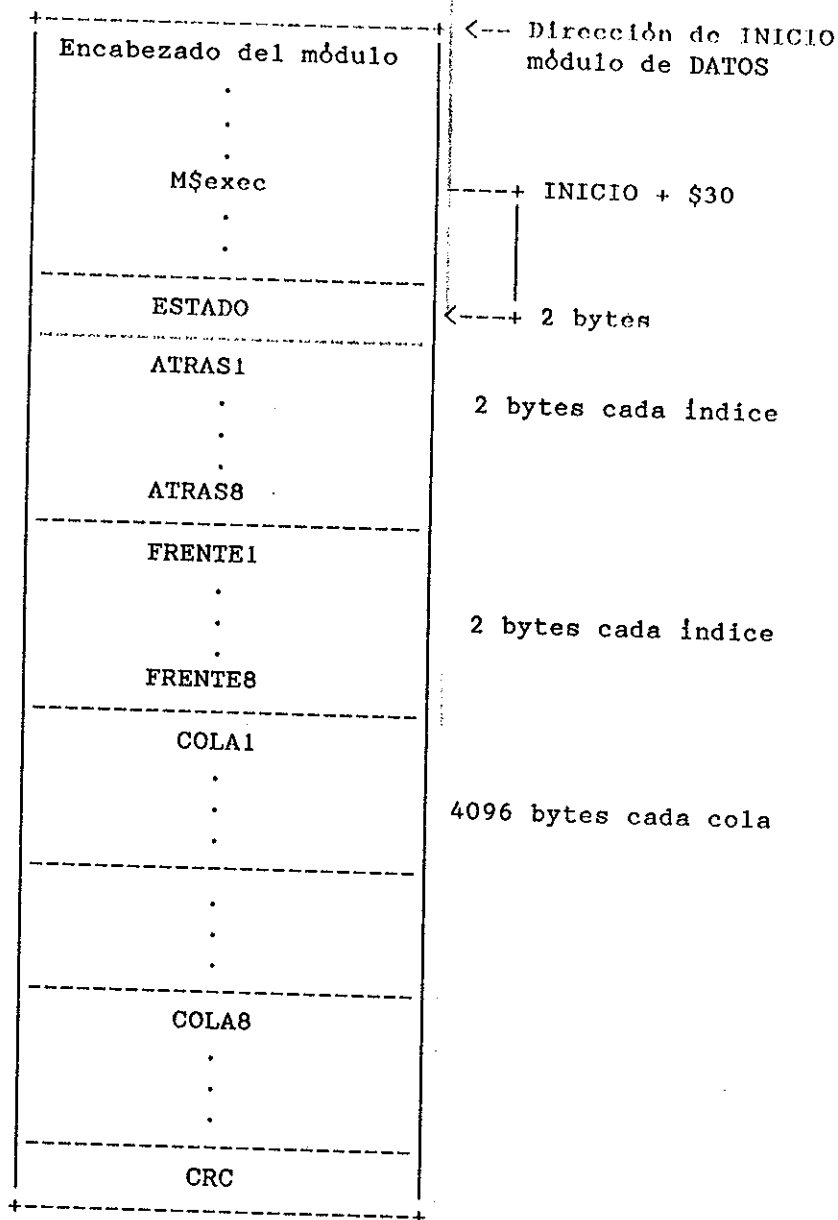


Figura 12. Organización de las varias estructuras de datos que se encuentran en el módulo de DATOS que es accesado en forma concurrente por los tres procesos: detección/grabación, adquisición/reformatización y estampado de marcas de tiempo.

Otro de los requisitos para que la implementación funcione, es que los tres procesos (descritos en el capítulo 5), tienen que estar en completa

coordinación y de acuerdo respecto a las estructuras de datos que ahí se almacenan y de sus dimensiones. Si se efectúa un cambio en uno de los tres procesos en las definiciones relacionadas con el módulo de datos, los otros dos tendrán también que reflejar el cambio. El parámetro más importante es el tamaño de las colas de datos.

6.2 Manjedores de Dispositivos.

Por las razones expuestas en el capítulo 4, fue necesario desarrollar manejadores propios con características muy específicas de nuestra aplicación. Estos se desarrollaron en lenguaje ensamblador del 68000. En virtud de no tener disponible la información de los dispositivos MK68564, solo se desarrollaron manejadores para los DUARTs MC68681. Esto implica que solo se podrá atender a 5 estaciones sismológicas con su reloj sincronizado con la WWVB para poner las estampas de tiempo. Esto es porque se tienen 6 puertos serie en los 3 dispositivos MC68681. Con la experiencia adquirida con los manejadores implementados para los dispositivos MC68681, se podrán desarrollar otros para el MK68564 una vez que se tenga la información. Cuando se desarrolle un manejador tipo RESNOR para el MK68564, se podrán incorporar 2 estaciones más.

Si recuerda del capítulo 3, una sola copia del manejador puede controlar varios dispositivos idénticos con características operacionales similares. Este es el caso para las estaciones de RESNOR.

Se desarrollaron 2 manejadores para el MC68681. Uno para procesar la

entrada de la información sísmica (ma68681) y otro (re68681) para manejar la entrada del reloj sincronizado con la WWVB (Favela F. 1983). Se implementaron parcialmente de acuerdo a las normas estándar de OS-9. De las 7 rutinas que forman un manejador estándar, solo se desarrollaron 2 ya que las otras no son de utilidad para nuestra aplicación. Se implementó la rutina de inicialización del dispositivo y la que deshabilita o termina al dispositivo. Las otras 5 (Leer, Escribir, Getstat, Setstat y Traperr) son sustituidas por otras funciones de la aplicación o son eliminadas por ser innecesarias. Por ejemplo: no se requiere de salida por los puertos serie (Escribir), la lectura (Leer) es efectuada por el proceso de detección-grabación.

Para los dos manejadores, la rutina que efectúa todo el trabajo de adquisición-reformatización y estampado de marcas de tiempo respectivamente, es la de atención a interrupción, que forman parte del manejador pero no es accesible por medio de la tabla de desplazamiento a los 7 puntos de entrada de las rutinas del manejador de dispositivo.

Al inicializar un dispositivo por medio del comando de OS-9:

Iniz nombre_módulo_descriptor_del_dispositivo

el NUCLEO llama a la rutina de inicialización (la primera de las 7) del manejador del dispositivo, el cual en forma general efectúa las siguientes faenas:

- Reserva espacio para el area de almacenamiento estático de éste dispositivo en particular.
- Traduce el código de OS-9 de parámetros operacionales de dispositivo de caracteres secuenciales a código que entiende el dispositivo. Con código traducido escribe en los registros apropiados de éste.
- Instala en la tabla de interrupciones de OS-9, la rutina de atención a interrupción para este dispositivo por medio de la llamada al sistema F\$IRQ.
- Inicializa variables locales del manejador para este dispositivo en particular. En nuestro caso, parte de la inicialización es enlazarse al módulo de DATOS.

Los nombres de los módulos descriptores de dispositivos asociados con los puertos serie de los dispositivos MC68681 para las estaciones de RESNOR y el reloj sincronizado con la WWVB son los siguientes:

EST3, EST4, EST5, EST6, EST7, EST8, RELOJ

Los módulos descriptores de dispositivos de caracteres secuenciales (nuestro caso), tienen una serie de campos que definen caracteres de control de flujo, edición, etc. En nuestra aplicación, son de nula utilidad. Por ejemplo: el caracter de aborto, X_ON, X_OFF, etc. Para los módulos descriptores de las estaciones de RESNOR, se utilizó el campo del caracter X_OFF para definir que cola de datos le corresponde a ese dispositivo dentro del módulo de DATOS compartido. Este caracter X_OFF, tiene un valor cuyo rango es entre 1 y 8. La rutina de inicialización utiliza este valor para calcular la dirección de sus índices ATRAS, FRENTE e INICIO de cola.

Al inicializar un dispositivo por medio de la invocación del comando de OS-9 "iniz" con un módulo descriptor de las estaciones de RESNOR o RELOJ, los procesos de adquisición-reformatización y/o estampado de marcas de tiempo se establecen, ya que la rutina de atención a interrupción es la que efectúa toda la faena descrita en la sección 5.3.1 y 5.3.2 de la presente tesis. Estos procesos son guiados por interrupciones. La interrupción ocurre cuando se recibe un un caracter en ese puerto y es generada por el dispositivo.

El hecho de que ésta parte de la implementación (adquisición reformatización y estampado de marcas de tiempo) se efectúe por rutinas de atención a interrupciones, implica que realmente no existe un proceso asociado a éstas tareas. Si se invoca el comando de OS-9 "procs" (despliega procesos), no se encontrará ninguno que esté llevando a cabo estas funciones. Solo se verá al proceso que está efectuado la detección grabación.

6.3 Manejador MA68681. Adquisición-Reformatización.

A continuación se describe en forma general los algoritmos de las rutinas de inicialización del dispositivo (Init), terminación (Termina) del dispositivo y la de atención a interrupción (Aten_Irq). El código en mnemónicos del ensamblador del 68000 para estas rutinas se encuentra en el apéndice 1.

Init.- Inicializa el dispositivo con la información contenida en el módulo descriptor del dispositivo que se quiere inicializar (dirección del puerto, número de bits de alto y bits por carácter, paridad, baud rate, etc.). Traduce el código de los parámetros de operación de OS-9 al código del MC68681 y escribe en sus respectivos registros.

-Instala en la tabla de interrupciones de OS-9, la rutina de atención a interrupción para éste dispositivo por medio de la llamada al sistema F\$IRQ.

-Enlázate al módulo de DATOS por medio de la llamada al sistema F\$Link y copia en variables estáticas locales las direcciones de ESTADO, ATRAS, FRENTE e inicio de COLA que le corresponden a éste puerto con ayuda del contenido del carácter X_OFF.

-En caso de ocurrir algún error en llamadas a F\$Link o F\$IRQ, notifica al llamador.

Termina. Deshabilita el puerto. Retira de la tabla de interrupciones de OS-9 la rutina de atención a interrupción a éste dispositivo, por medio de llamada al sistema F\$IRQ.

Aten_Irq. En virtud de que todos los puertos serie de los dispositivos MC68681 están alambrados al mismo autovector (29), verifica si el dispositivo en prueba realmente interrumpió. Si no fue así, regresa inmediatamente con el registro de código de condiciones (ccr) en 1. El sistema interrogará al dispositivo con la siguiente prioridad de escrutinio en el mismo autovector.

-En caso de haber interrumpido (recibió carácter), verifica si el dispositivo reporta errores (paridad, framing, overrun). En caso de existir, escribe el error en el campo V_ERR del descriptor del dispositivo (aquí se acumulan errores).

-Prueba si se recibió el LSB o el MSB.

+Si LSB, reformatea la media muestra y escríbela en una variable local para éste dispositivo.

+Si MSB, reformatea media muestra, etiqueta canal y complementa muestra con el LSB. Escribe muestra verificando sobreflujo en COLA si ESTADO es de grabación. Si se escribió la muestra en su respectiva COLA, actualiza ATRAS. Si ocurrió un sobreflujo (COLA llena) no escribas muestra ni actualices ATRAS (se pierde la muestra). Además reporta error a V_ERR en descriptor de dispositivo.

6.4 Manejador RE68681. Estampado de Marcas de Tiempo

Este manejador tiene que ver con todas las colas de datos, pero está asociado a un solo puerto, el que está conectado al reloj sincronizado. Su función es decodificar y comprimir la sarta de 16 caracteres que el reloj manda cada segundo. Los formatos de la sarta, estampas binarias de tiempo y el proceso en general están descritos en la sección 5.3.2.

Las rutinas de inicialización y termina son similares a las del manejador MA68681 ya que se trata del mismo dispositivo. La diferencia estriba en que este manejador solo necesita la dirección de inicio del arreglo de índices ATRAS. Toda la demás información que requiere la puede inferir de esta dirección.

La rutina de atención a interrupción decodifica la información transmitida por el reloj, validando los caracteres que son invariantes en cada paquete de 16 caracteres. Estos son: control A CTRLA (\$01), dos puntos : (\$3a), regreso de carro CR (\$0D) y salto de línea LF (\$0A).

Si un carácter invariante no es recibido en el orden que le corresponde, la rutina se resincroniza con el próximo CTRLA. Si se validan los caracteres invariantes correctamente, al recibir el CR la rutina pone las 2 estampas binarias de tiempo en cada una de las 8 colas de datos proporcionando así una referencia precisa en el tiempo de la información sísmica que se está recibiendo de las estaciones de RESNOR. El código en mnemónicos del ensamblador del 68000 para este manejador se encuentra en

el apéndice 2.

6.5 Módulo de Detección-Grabación.

Para poder lograr el objetivo de detectar en tiempo real eventos sísmicos en las señales digitales transmitidas por las estaciones de RESNOR, el sistema en su conjunto se basa en ventanas corridas de tiempo de la señal sísmica y de información extraída de ésta.

Las ventanas de tiempo, como fue explicado en el capítulo 5, fueron implementadas con la estructura de datos llamada cola. Entonces tenemos que se mantienen colas de datos, promedios cortos y de resultados de aplicaciones previas del algoritmo de promedios cortos contra promedios largos en las estaciones incluidas en la función de detección.

Las mejoras del sistema con respecto al que opera en la actualidad con el equipo PRIME son las siguientes:

- Función lógica de detección programable por el usuario.
 - El canal sobre el cual se aplica el algoritmo de promedios cortos contra promedios largos es seleccionable por el usuario.
 - Algoritmo de promedios cortos contra promedios largos se aplica a la primera diferencia de la señal del canal seleccionado.
-

Este módulo fue codificado en lenguaje "C" y opera sobre los datos que fueron escritos en las colas de datos por la rutina de atención a interrupción del manejador ma68681.

Este proceso al inicializarse, también se enlaza al módulo de DATOS copiando en forma local las direcciones de ESTADO, índices ATRAS y FRENTE e inicio de cada una de las 8 colas.

Mantiene en forma local (no compartida) las colas de promedios cortos, de resultados, la tabla que almacena la función de detección y la tabla de promedios largos.

El algoritmo de este módulo en su conjunto se describe en forma general a continuación:

- Inicializa

- Repite para SIEMPRE

- Aplica algoritmo de promedios cortos vs. promedios largos a estaciones involucradas en función de detección

- Si no se reporta EVENTO sísmico, duerme por período equivalente a un promedio corto. Después despierta y repite ciclo infinito.

- Si hay EVENTO sísmico:

- Genera los nombres de los archivos donde se almacenará la información detectada. Abre los nuevos archivos.

- Cambia a ESTADO de grabación (1). Establece a las colas como si estuvieran a un 75% llenas.

-Lee de las colas y escribe en los archivos mientras persista la condición de disparo. Independientemente de quien haya reportado el evento sísmico, graba de todas las estaciones.

-Cierra archivos generados

-Cambia a ESTADO de detección (0).

-Repite ciclo infinito.

6.6 Concurrencia al Módulo de DATOS

Conflictos entre procesos debido al acceso concurrente del módulo de datos fueron eliminados en parte por el diseño del sistema y por otro lado por la arquitectura de la misma tarjeta coprocesador PC68K.

La variable ESTADO y los índices FRENTE, solo son modificados por el proceso de detección-grabación. Esto ocurre al declarar la ocurrencia de un evento. La variable ESTADO es puesta en 1 y los índices FRENTEs son movidos de tal manera que se establece que las colas se encuentran en un 75% llenas. Los índices FRENTE también son modificados al estar transfiriendo los datos de las colas a archivos en disco. Los otros 2 procesos leen solamente éstas variables. Estos lo hacen cada vez que quieren escribir una muestra en las colas. Revisan el valor de ESTADO, si es diferente de cero, leen índices FRENTE para establecer si la cola está llena. Al terminar la condición de evento sísmico, el proceso de detección-grabación pone ESTADO en 0, repercutiendo en los otros dos procesos. Cuando ESTADO es 0, los índices FRENTEs no son considerados para nada por ninguno de los procesos.

Los índices ATRAS son modificados por los procesos de adquisición reformatización y estampado de marcas de tiempo. Aquí es donde podrían surgir conflictos, pues los dos modifican los índices ATRAS al escribir en las colas por un lado muestras sísmicas y por otro estampas de tiempo.

Como se recordará del capítulo 2, los 6 puertos serie de los 3 dispositivos MC68681 están alambrados al mismo vector de excepción. Este es el autovector 29 que asigna a todos el nivel de interrupción 5. Este hecho nos resolvió el problema de la concurrencia al módulo de datos, ya que mientras se esté dando servicio de interrupción a uno de los puertos serie (todos con nivel 5), solo estarán permitidas atenciones a interrupciones con nivel 6 o 7(NMI), evitando así el conflicto en la competencia por un mismo recurso (índices ATRAS).

El proceso de detección-grabación lee pero no modifica los índices ATRAS. Esto lo hace en dos situaciones. Una es para aplicar el algoritmo de promedios cortos contra promedios largos a las muestra más recientemente recibidas. La muestra más reciente está apuntada por el índice ATRAS. La otra situación cuando se leen los índices ATRAS, es para determinar si la cola está vacía con ayuda de se respectivo índice FRENTE.

8 CONCLUSIONES y RECOMENDACIONES

Con el presente trabajo permitirá tener a la División de Ciencias de la Tierra del CICESE un sistema redundante en el subsistema de adquisición de la red sísmológica RESNOR. El subsistema de comunicaciones localizado en el observatorio de San Pedro Martir que se encarga de concentrar las señales de las estaciones que se encuentran en el valle de Mexicali y retransmitirlas a Ensenada, sufre con frecuencia averías en la época de tormentas y en el invierno, evitando así que se reciban las señales en el CICESE. El sistema desarrollado se podría instalar en un lugar estratégico en el valle de Mexicali para capturar la información que se pierde cuando el subsistema de comunicaciones sufre un desperfecto.

El presente desarrollo está lo suficientemente documentado para que otros usuarios (UNAM UAP) puedan implementar sus subsistema de captura para redes similares a la de RESNOR (RESMAC).

Una de las razones que permitieron cumplir el objetivo planteado con las herramientas seleccionadas, fue la buena y extendida documentación técnica de la empresa Microware del sistema operativo OS-9 que en realidad mostró ser modular, flexible y poderoso.

La presente tesis abarcó solamente los aspectos de captura y detección. Entre las directrices a seguir desarrollando en éste nuevo sistema están las siguientes:

- Las estructuras de datos que mantiene en forma privada el módulo de detección grabación (colas de promedios cortos, función de detección, etc.) hacerlas públicas por medio de otro módulo de datos o el mismo pero expandido, que permita a otros procesos (monitor del sistema) acceder ésta información.
 - Implementación del detector secundario (Hinojosa A. 1983), que revisa los archivos generados por el detector primario (Plargos vs. Pcortos) para verificar su utilidad.
 - Implementación de utilerías de graficado, manejo de información, cálculo de magnitudes, etc.
 - Investigación y desarrollo en la comunicación de la PC68K con la IBM a través de la memoria de doble acceso MDP y la red Ethernet del Centro de Cómputo del CICESE, para hacer más expedita la transferencia de archivos entre máquinas.
-

9 LITERATURA CITADA

- Dallas Semiconductor. "DS1216 Smartwatch" Preliminary.
- Duarte C., Hinojosa A. 1980. "Evaluación del Sistema PRIME 350 en la Adquisición de Datos en Tiempo Real". CICESE Reporte Técnico.
- Duarte, C., Hinojosa, A. e Inzunza, L. 1981. "MUPPET: programa de apoyo para el procesamiento de datos de RESNOR". CICESE, Informe Especial GEO81-02.
- Emerald Computers Inc. 1987. "68000 CoProcessor Board. Installation and Technical Reference" Portland Oregon.
- Favela F. Duarte C. 1983 "Base de Tiempo Geofísica Sincronizada con la WWVB". Memorias del VI Congreso Nacional de Ingeniería Sísmica. pp 525-536. Noviembre 1983. Puebla, Puebla.
- Favela F. 1984. "Diseño y Construcción de un Controlador de Bajo Consumo de Energía para uso en Instrumentación". Tesis de Maestría en Ciencia de Instrumentación Electrónica. CICESE.
- Harman Thomas I., Lawson Barbara. 1985. "The Motorola MC68000 Microprocessor Family: Assembly Language, Interface Design and System Design". Prentice-Hall Inc., Englewood Cliffs, New Jersey.
- Hinojosa A., Duarte C., Inzunza L. 1982. "Sistema de Adquisición y Procesamiento RESNOR". Reporte técnico interno del CICESE.
- Hinojosa A, Duarte C. 1982. "PRIMOS Modifications for a Real Time Data Acquisition with a PRIME 350". Reporte técnico interno CICESE.
- Hinojosa A. Duarte C. 1984. "Procesamiento Automático de Banco de Datos de RESNOR". Memorias del congreso de la Unión Geofísica Mexicana 1983.
- Hinojosa A., Duarte C., Inzunza L. 1987. "Sistema de Captura y Procesamiento RESNOR". Memorias de la Reunión Anual de la Unión Geofísica Mexicana. Ensenada B. C.
- Medina M., Duarte C. 1979. "Red Sismológica de Telemedición Digital del Campo Geotérmico de Cerro Prieto". Mexicali, B.C. Octubre 1979, pp. 335-368.
- Medina M., Duarte C. 1984. "RESNOR: Red de Telemetría Sísmica del CICESE". Memorias de la Reunión 1982-1983 de la Unión Geofísica Mexicana, pp. 254. México D.F.

- Microware. 1986. "OS-9/68000 Assembler/Linker/Debugger User's Manual". Des Moines ", Iowa.
- Microware. 1986. "OS-9/68000 C Compiler User's Manual". Des Moines, Iowa.
- Microware. 1986. "OS-9/68000 Operating System Technical Manual". Des Moines, Iowa.
- Microware. 1986. "Using Professional OS-9". Des Moines, Iowa
- Motorola. 1985 Septiembre. "MC68681 Dual Asynchronous Receiver/Transmitter (DUART)". Austin, Texas.
- Motorola. 1985. "MC68000 12-/32 Bit Microprocessor". Octubre 1985. Motorola Inc. Austin, Texas.
- United Technology. "MK68564 MOSTEK Serial Input Output Technical Manual".
- Valerdi J. 1978. "Red Sismológica Digital en Tiempo Real". Revista de Ingeniería, Vol. 48, No. 2, UNAM. abril-junio 1978, México D.F., pp 192-202.
- Vidal A., Munguía I. 1987. "Local Magnitud and Source Parameter for Earthquakes in the San Miguel Fault Zone, Baja California, México". Memorias del Encuentro de Otoño de American Geophysical Union. Diciembre 1987. San Francisco, California.
- VLSI Technology. "VT2130/VT2131 1,024 x 8 Dual Port Ram." Preliminary. 1109 Mckay Drive, San Jose Ca. 95131.
- Triebel W. A., Singh A., 1986. "The 68000 Microprocessor. Architecture, Software and Interfacing Techniques". Prentice-Hall Inc., Englewood Cliffs, New Jersey.
- Williams Steve. 1985. "Programming the 68000". SYBEX Inc. Berkeley, California.