

**CENTRO DE INVESTIGACION CIENTIFICA Y DE  
EDUCACION SUPERIOR DE ENSENADA**

**DISEÑO Y CONSTRUCCION DE UN CONTROLADOR  
PROGRAMABLE DE BAJO CONSUMO DE ENERGIA  
PARA USO EN INSTRUMENTACION**

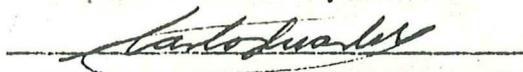
**T E S I S  
MAESTRIA EN CIENCIAS**

**Fernando Favela Vera**

RESUMEN de la Tesis de Fernando Favela Vara presentada como requisito parcial para la obtención del grado de MAESTRO EN CIENCIAS en FISICA APLICADA con opción en ELECTRONICA Y TELECOMUNICACIONES. Ensenada, Baja California, México, julio de 1984.

DISEÑO Y CONSTRUCCION DE UN CONTROLADOR PROGRAMABLE DE BAJO CONSUMO DE ENERGIA PARA USO EN INSTRUMENTACION.

Resumen aprobado:



M. en C. Carlos Duarte M.

Director de Tesis

Se describe el diseño de un sistema de control digital programable de propósitos generales y de bajo consumo de energía para utilizarse en aplicaciones de instrumentación.

El sistema se desarrolló con la arquitectura de una microcomputadora basada en el microprocesador MC146805E2 y contiene 16 líneas de control digital bidireccionales, un reloj de tiempo real, 2K octetos de memoria de lectura/escritura y 2K de memoria de programación; además de

la capacidad para direccionar hasta 32K de memoria externa. La comunicación del sistema con el exterior se realiza a través del ducto estándar industrial C-44, recomendado para instrumentos de bajo consumo de potencia a base de microprocesadores.

La circuitería desarrollada se complementó con un sistema de programación basado en el concepto de lenguajes de estado, con el objeto de expeditar el desarrollo de aplicaciones orientadas hacia el control digital.

Como ejemplo de aplicación del controlador se desarrolló un instrumento completo. Este consistió en una base de tiempo con calidad patrón capaz de sincronizarse automáticamente con señales transmitidas por la estación WWVB. Actualmente el instrumento está siendo utilizado en investigaciones de la División de Ciencias de la Tierra del CICESE, y se espera poder difundirlo a otras instituciones que tengan necesidad de registrar con exactitud señales de tiempo.

CENTRO DE INVESTIGACION CIENTIFICA Y DE  
EDUCACION SUPERIOR DE ENSENADA

DIVISION DE FISICA APLICADA

DISEÑO Y CONSTRUCCION DE UN CONTROLADOR PROGRAMABLE  
DE BAJO CONSUMO DE ENERGIA PARA USO EN INSTRUMENTACION.

TESIS

que para cubrir parcialmente los requisitos necesarios para  
obtener el grado de MAESTRO EN CIENCIAS presenta

FERNANDO FAVELA VARA

Ensenada, B. C., julio de 1984

TESIS APROBADA PARA SU DEFENSA POR:

---

M.C. Carlos Duarte Muñoz, Director del Comité

---

Dr. José Carlos Pomalaza Díaz, Miembro del Comité

---

M.C. José de Jesús Ibarra Villaseñor, Miembro del Comité

---

M.C. Jaime Sánchez García, Miembro del Comité

---

Dr. José Carlos Pomalaza Díaz, Jefe del Depto. de Electrónica y  
Telecomunicaciones

---

M.C. Martín Celaya Barragán, Director de la División de Física  
Aplicada

---

M.C. Gilberto Gaxiola Castro, Director Académico Interino

Tesis presentada en Julio 6, 1984

## DEDICATORIA

A mi padre y a mi madre por todo lo que me han dado durante el transcurso de mi vida

A mi esposa Bety que ha tenido que pasar soledades en mis desvelos y me ha dado aliento y alegría en esta nueva etapa de nuestras vidas.

A mi hijo o hija que pronto compartirá todo esto con nosotros.

## AGRADECIMIENTOS

A mi director de tesis y amigo M. en C. Carlos Duarte M. por la gran ayuda y apoyo que me ha brindado durante el tiempo que hemos trabajado juntos

A los miembros de mi comité de tesis Dr. José Pomalaza, M. en C. Jaime Sánchez y, M. en C. Jesús Ibarra por la revisión, crítica y comentarios a este trabajo y a mi asesor académico Dr. Enrique Mitrani por el tiempo dispensado durante mi estancia en el C. I. C. E. S. E.

A los miembros de las Divisiones de Física Aplicada y de Ciencias de la Tierra por el apoyo y la amistad que me han ofrecido.

Al personal de Coordinación Académica por la ayuda que me dieron desde mi llegada a este Centro y en especial a la Sra Socorro R. de Carrillo, a Nety y a Luz Aurora por el tiempo y esfuerzo extra dedicado a la revisión final de este trabajo.

Al C. D. N. A. C. y T. y a todos aquellos que por falta de espacio o memoria no pude agradecer aquí.

## CONTENIDO

	Página
1 PRESENTACION	1
1.1 Introducción	1
1.2 Estructura del trabajo	3
2 UNIDAD CENTRAL DE PROCESAMIENTO (CUTCMOS)	5
2.1 Descripción del sistema	5
2.2 Microprocesador	7
2.3 Reloj de tiempo real	10
2.4 Memoria de lectura/escritura	13
2.5 Memoria de lectura exclusiva	13
2.6 Manejo del mapa de memoria	13
2.7 Conexiones externas	16
2.8 Interrupciones externas	16
2.9 Modos de muy bajo consumo de energía	17
2.9.1 Modo WAIT	17
2.9.2 Modo STOP	18
2.10 Conclusiones	18
3 PROGRAMACION DE DIAGRAMAS DE ESTADO	20
3.1 Introducción	20
3.2 Aspectos generales	20
3.3 Las técnicas de descripción de estados	23
3.4 Aplicaciones apropiadas	25
3.5 Descripción del lenguaje de estados	27

	Página
3.5.1 Declaración ESTADO	27
3.5.2 Declaración de condición	28
3.6 Funciones lógicas	30
3.7 Traducción del lenguaje de estados	32
3.7.1 La macroinstrucción ESTADO	33
3.7.2 La Macroinstrucción de condición	33
3.8 Beneficios derivados del uso de la técnica de descripción de estados	35
3.9 Implementación del Procesador de Estados	38
3.9.1 Subrutina principal. (PROCED)	40
3.9.2 Rutina de ejecución de acciones. (EXECU)	41
3.9.3 Subrutina de validación de condición (CONDIC)	42
3.9.4 Subrutina MANO	43
3.10 Implementación del traductor del lenguaje a la tabla de estados	43
3.10.1 ESTADO	43
3.10.2 CONDIC	44
4 APLICACION GEOFISICA	47
4.1 Introducción	47
4.2 Objetivos de diseño	48
4.3 Formato de la señal de tiempo	49
4.4 Actividades del instrumento	50
4.5 Sistema propuesto	54

	Página
4.6 Programación	58
4.6.1 Rutina de Inicialización general. (RESET)	59
4.6.2 Rutina principal de despacho de interrupciones. (INTERR)	61
4.6.3 Rutina de atención a la interrupción del botón. (ATNINT)	62
4.6.4 Rutina de atención al reloj en Tiempo Real	63
4.6.5 Rutina de atención a la interrupción de la Alarma del RTR. (ALARMA)	63
4.6.6 Rutina de atención a la interrupción de actualización. (ACTUAL)	64
4.6.7 Rutina para determinar la calidad de la marca transmitida. (CALIDA)	66
4.6.8 Rutina de atención a la interrupción periódica. (INPERI)	67
4.6.9 Subrutina de envío de dígitos a los visualizadores. (SACA)	67
4.6.10 Subrutina de despliegue de un dígito (DIGIT)	68
4.6.11 Rutina de atención a la interrupción del UART. (ATNURT)	69
4.6.12 Rutina de atención a la señal de la WWVB	69
4.6.13 Rutina de tratamiento del flanco positivo	70
4.6.14 Rutina de tratamiento del flanco negativo	71

	Página
4.6.15 Subrutina para calcular la duración de los pulsos. (ANCHOP)	72
4.6.16 Rutina de atención a la interrupción del temporizador interno. (TIMER1)	73
4.6.17 Rutina de preparación a la sincronización	74
4.6.18 Rutina de sincronización de la base de tiempo. (CRONOS)	75
El procesador de estados	76
4.7.1 Rutinas ejecutables por el procesador de estados.	77
4.7.1.1 Subrutina NADA	79
4.7.1.2 Subrutina CARGAN	79
4.7.1.3 Subrutina NERROR	79
4.7.1.4 Subrutina ERROR	79
4.7.1.5 Subrutina CARDIG	80
4.7.1.6 Subrutina DECENE	80
4.7.1.7 Subrutina SEG35	80
4.7.1.8 Subrutina PREPAR	80
4.7.1.9 Subrutina DUPLIC	81
5 CONCLUSIONES	82
APENDICE I	90
Conexiones al ducto principal C-44	99
Conexiones al ducto auxiliar	100

	Página
APENDICE II	102
Especificaciones del ducto C-44	102
Prefacio	102
Objetivos de diseño	102
Consumo de energía mínimo	102
Amplia compatibilidad	103
Flexibilidad en el tamaño del sistema	103
Consideraciones generales	104
Tipos de tarjetas	104
Tarjetas de UCP	104
Tarjetas periféricas	105
Estados activos e inactivos	105
Asignación de patas al ducto	106
APENDICE III	110
Arquitectura de procesador MC146805E2	110
El acumulador (A)	110
El registro de índice (X)	110
El contador de programa (PC)	110
El apuntador de pila (S)	111
El registro de códigos de condición (CC)	111
Modos de direccionamiento	112
Modo de direccionamiento inmediato	112
Modo de direccionamiento directo	112

	Página
Modo de direccionamiento extendido	113
Modo de direccionamiento indicado	113
Indicado sin desplazamiento	113
Indicado con desplazamiento de 8 bits	114
Indicado con desplazamiento de 16 bits	114
Modo de direccionamiento relativo	115
Modo de direccionamiento de control de bits	115
Modo de direccionamiento de prueba de bits individuales	116
Modo de direccionamiento inherente	116
Las instrucciones del MC147806E2	116
Instrucciones de registro/memoria	117
Instrucciones lee/modifica/escrbe	118
Instrucciones de brincos condicionados	119
Instrucciones de manipulación de bits individuales	120
Instrucciones de prueba de bits individuales	121
Instrucciones de control	122
Sistema de desarrollo de MC146805E2	123
Operación	123
APENDICE IV	125
Tiempo y Frecuencia	125
El meridiano estándar	125
El Tiempo Universal	125
El tiempo Universal Coordinado (UTC)	126

	Página
El Tiempo y su medición	126
Conceptos y caracterización de la divulgación de tiempo y frecuencia.	130
Conceptos fundamentales	130
Relojes	130
Fecha o Epoca	131
Frecuencia	132
Sincronización	132
Técnicas de divulgación de Tiempo y Frecuencia	134
APENDICE V	139

## LISTA DE FIGURAS

	Página
Figura 1. Diagrama a bloques de la CUTCMOS	6
Figura 2. Diagrama de transiciones del oscilador al ducto externo del MCl46805E2	9
Figura 3. Arquitectura interna del MCl46805E2	11
Figura 4. Manejo de memoria de la CUTCMOS	15
Figura 5. Diagrama de un circuito secuencial	24
Figura 6. Diagrama de estados	24
Figura 7. Programacion de estados representada en diagrama de flujo	26
Figura 8. Implementación de la función negado	31
Figura 9. Implementación de la función 'Y'	31
Figura 10. Traducción del lenguaje de estados a la tabla de estados	34
Figura 11. Diagrama de flujo del procesador de estados	39
Figura 12. Diagrama del código de tiempo de la WWVB	51
Figura 13. Diagrama a bloques del reloj sincronizado con la WWVB	55
Figura 14. Diagrama de estados de la WWVB	78
Figura 15. Diagrama de localización de partes de la CUTCMOS	93
Figura 16. Tarjeta CUTCMOS	94

Figura 17. Diagrama electrónico de la  
tarjeta de Consola 98

Figura 18. Esquema de divulgación de tiempo 135

# DISEÑO Y CONSTRUCCION DE UN CONTROLADOR PROGRAMABLE DE BAJO CONSUMO DE ENERGIA PARA USO EN INSTRUMENTACION.

## 1 PRESENTACION.

### 1.1 Introducción.

Este trabajo presenta el desarrollo de un controlador programable de bajo consumo de energía orientado hacia aplicaciones de instrumentación y control industrial. El sistema desarrollado consiste en una microcomputadora de una tarjeta que emplea únicamente componentes de tecnología CMOS. la cual se ha denominado CUTCMS (Computadora en Una Tarjeta CMOS), y tiene las siguientes características.

2K de Memoria de lectura / escritura.

2K de Memoria de lectura exclusiva programable.

Un reloj en tiempo real (RTR).

Capacidad para direccionar hasta 32 K de memoria adicional

16 líneas de control digital bidireccionales.

La CUTCMOS se ha desarrollado como un prototipo reproducible a nivel industrial, lo que permitirá su difusión inmediata en caso de que exista interés por parte de otros investigadores o instituciones en adquirirla. Además, para la comunicación de la CUTCMOS con el exterior se ha adoptado el ducto estándar C-44, lo que permite su compatibilidad con otros productos CMOS adquiribles comercialmente. Esto ampliará su gama de aplicaciones al simplificar requerimientos en el diseño de módulos periféricos.

Para complementar la circuitería desarrollada y enfocar las aplicaciones de ésta hacia el control e instrumentación, se ha implantado en la CUTCMOS un sistema de programación basado en el concepto de lenguajes de estado. Mediante este sistema se simplifica la programación de procesos que puedan modelarse mediante diagramas de estados. Esto facilita el empleo inmediato de la CUTCMOS en el desarrollo de sistemas de control industrial.

Adicionalmente, y con el objeto de demostrar la aplicabilidad del controlador a sistemas de instrumentación, se desarrolló una aplicación concreta, la cual consistió de un reloj digital sincronizado por el código transmitido por

la estación de radio WWVE.

El reloj digital objeto de este trabajo presenta una serie de características que lo hacen interesante e innovador. Este sistema, una vez sincronizado por el código de tiempo de la WWVE, continúa llevando la cuenta del tiempo con o sin la referencia de la señal de radio, y dispone de un registro de antigüedad que permite estimar la calidad de las muestras de tiempo proporcionadas.

## 1.2 Estructura del trabajo.

El capítulo 2 contiene la descripción del diseño de la computadora de una sola tarjeta CUTCMOS. Se muestra su arquitectura y las características más importantes de la misma.

El capítulo 3 está dedicado a la descripción detallada del concepto de lenguaje de estados. Se incluye la idea de la cual surge el lenguaje de estados así como su implementación en la CUTCMOS.

En el capítulo 4 se hace la descripción completa de la aplicación que se le ha dado a la CUTCMOS, el reloj digital sincronizado con la señal de la WWVE. Se muestra la circuitería adicional que se ha requerido para tal efecto.

Se hace una descripción completa del problema de instrumentación que se pretendió resolver, y se presenta una descripción de la programación que fue necesaria para lograr resolver el problema.

Finalmente en el capítulo 5 se hace una sumarización del trabajo desarrollado. Se presentan las conclusiones y se hace mención del trabajo que queda por hacer.

Se presentan además varios apéndices.

En el apéndice I, se muestran las características de construcción de la CUTCMOS, y de la circuitería adicional requerida para el desarrollo del reloj sincronizado con la WWVB.

En el apéndice II se presenta una descripción de las líneas del ducto que se eligió para el diseño de la CUTCMOS junto con su filosofía de diseño.

En el apéndice III se hace una descripción del microprocesador MC146805E2 que se utilizó como parte principal de la CUTCMOS. Se describe también el sistema que se utilizó para su desarrollo.

En el apéndice IV se presentan algunos conceptos relativos a los estándares de tiempo y su difusión.

Y para terminar, en el apéndice V se muestran los listados de los programas generados.

## 2 UNIDAD CENTRAL DE PROCESAMIENTO (CUTCMOS).

### 2.1 Descripción del sistema

La Computadora en Una Tarjeta CMOS (CUTCMOS) es una microcomputadora orientada hacia aplicaciones de control e instrumentación diseñada para emplearse en situaciones donde el bajo consumo de energía sea un factor decisivo. La tarjeta contiene 2K de memoria de lectura/escritura (RAM), 2K de memoria de lectura exclusiva (EPROM), 16 líneas bidireccionales de entrada/salida digital y un reloj de tiempo real (RTR). Estas cualidades le dan la capacidad de trabajar por sí sola como controlador básico.

Adicionalmente, la CUTCMOS puede formar parte de sistemas de computación más complejos, ya que su capacidad puede ampliarse mediante la inclusión de módulos de expansión que pueden conectarse a un ducto común. En estos módulos pueden existir hasta 32K de memoria externa seleccionable en bancos de 2K.

La figura 1 es un diagrama a bloques de la arquitectura de la computadora CUTCMOS. En este diagrama se muestran las componentes principales del sistema, las cuales se describen a continuación con mayor detalle.

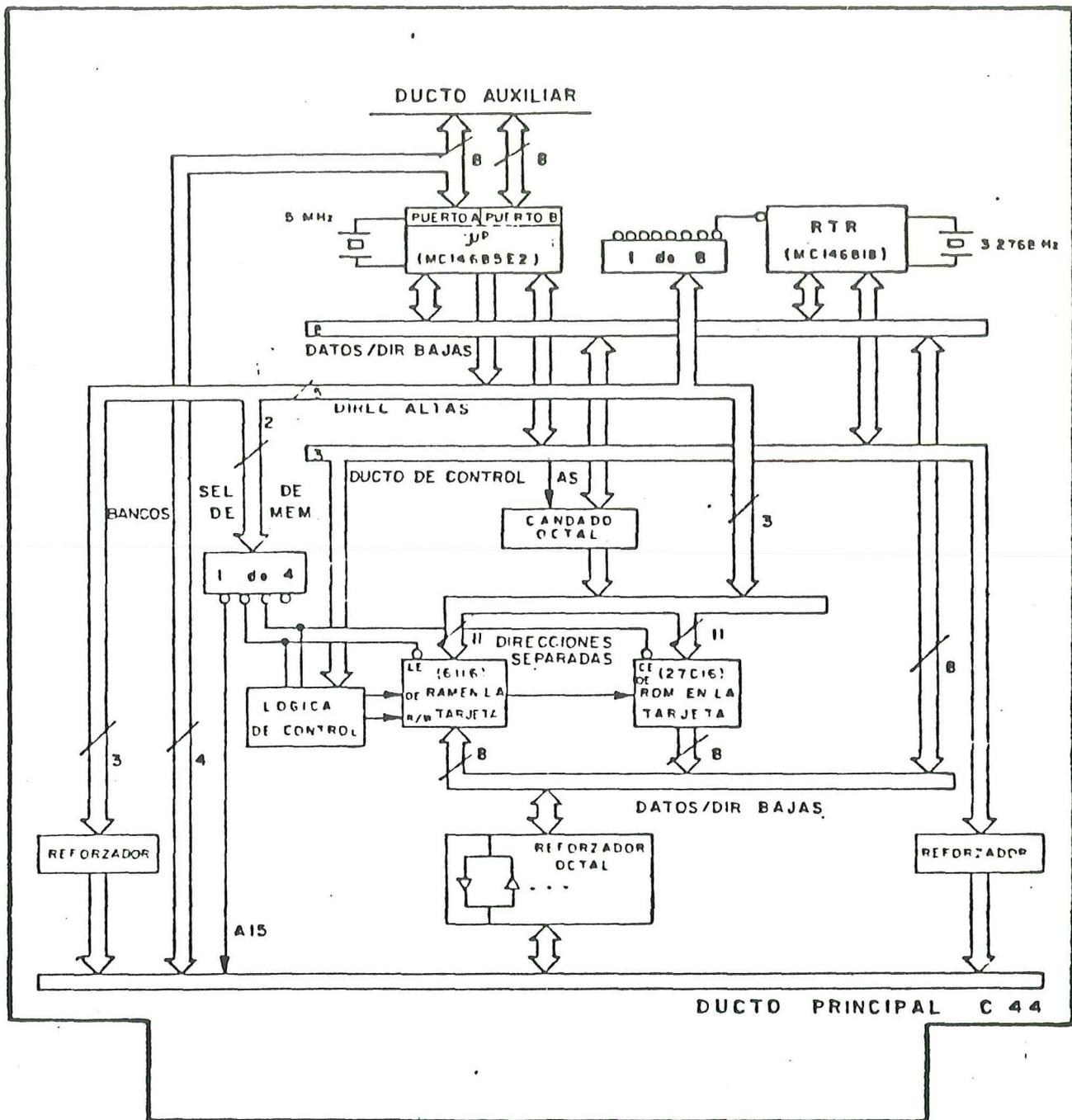


Figura 1. Diagrama a bloques de la CUTCMOS.

## 2.2 Microprocesador

El microprocesador empleado en la microcomputadora CUTCMOS, es un dispositivo MC146805E2, fabricado por Motorola y orientado fundamentalmente hacia aplicaciones de control (Anón., 1981)

Este microprocesador fue seleccionado en base a la disponibilidad de herramientas para hacer desarrollos con él, ya que el Laboratorio de Instrumentación Geofísica de la División de Ciencias de la Tierra del CICESE cuenta con un sistema de desarrollo de microprocesadores 'EXERCISER II'. Este sistema permite la elaboración de diseños alrededor de la familia 6800 de Motorola, por lo que la elección del MC146805E2 resultó natural. Esto es debido a que este microprocesador es un miembro de la familia 6800 disponible en tecnología CMOS.

El MC146805E2 contiene 2 registros internos de 8 bits (acumulador y registro de índice X), 112 octetos de RAM interna, que incluyen una pila (stack) de 64 octetos, un temporizador programable de 8 bits con capacidad de generar interrupciones, y 16 líneas de entrada/salida (Ver apéndice III). Este microprocesador es un dispositivo CMOS completamente estático y puede operar a diferentes velocidades, desde DC hasta 1 MHz (ciclo de máquina). En la

CUTCMOS, la frecuencia de diseño es la máxima (1 MHz) y está controlada por un cristal de 5 MHz mostrado en la figura 1. Esto no obsta para que pueda operar a velocidades menores.

El MC146805E2 tiene una arquitectura de 8 bits con un ducto de datos/direcciones-bajas compartido. El funcionamiento de este ducto es similar al de otros microprocesadores que comparten datos y direcciones en las mismas líneas. En un ciclo de acceso a memoria, los bits menos significativos de la dirección aparecen en el ducto durante la primera porción del ciclo, mientras que los bits de datos comparten las mismas líneas durante la segunda porción del ciclo. Para distinguir entre envío de direcciones y lectura/escritura de datos, el microprocesador activa en su oportunidad las señales AS (ADDRESS STROBE) y DS (DATA STROBE), respectivamente. Estas señales, junto con la línea R/W, constituyen el ducto de control del microprocesador, y sirven para indicar el tipo de actividad de los ductos de datos/direcciones-bajas y direcciones altas, en un momento dado. Ver figura 2

El hecho de que las señales de datos y direcciones-bajas compartan las mismas líneas, obliga a que al conectar dispositivos externos al microprocesador, que requieran de datos y direcciones separadas, se haga necesario utilizar un candado octal (octal latch) para

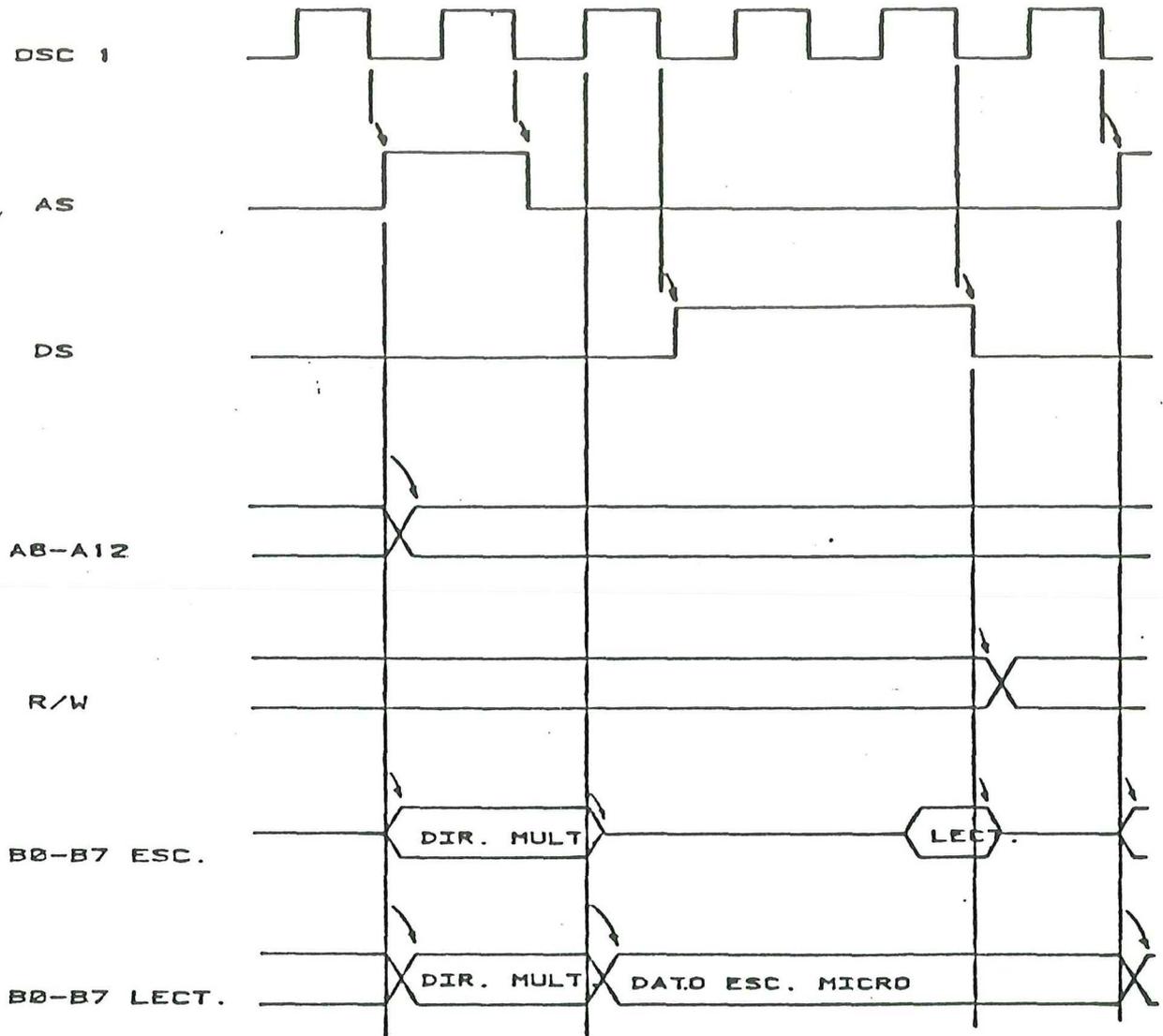


Figura 2. Diagrama de transiciones del oscilador al ducto externo del MC146805E2.

efectuar la separación. Este es el caso de las memorias RAM y EPROM de la CUTCMOS, y la inclusión del candado octal activado por la línea AS para capturar las direcciones bajas, se muestra en la figura 1.

La figura 3 muestra un diagrama de la arquitectura interna del MC146805E2.

El conjunto de instrucciones del MC146805E2 es similar al del conocido microprocesador 6800 y se podría decir que el MC146805E2 es una versión degradada del 6800, ya que no cuenta con todos los modos de direccionamiento de éste. Sin embargo, al MC146805E2 se le han añadido un grupo de instrucciones de manipulación de bits que permiten modificar cualquier bit individual de RAM interna o de los puertos con una sola instrucción. Similarmente, el MC146805E2 dispone de instrucciones que efectúan brincos condicionados dependiendo del estado de un bit. La capacidad de ejecutar estas instrucciones hace que la CUTCMOS sea ideal para aplicaciones de control, en donde detectar o controlar un bit particular resulta esencial.

### 2.3 Reloj de tiempo real.

La CUTCMOS contiene un reloj de tiempo real (RTR) basado en el dispositivo MC146818 de Motorola. Este

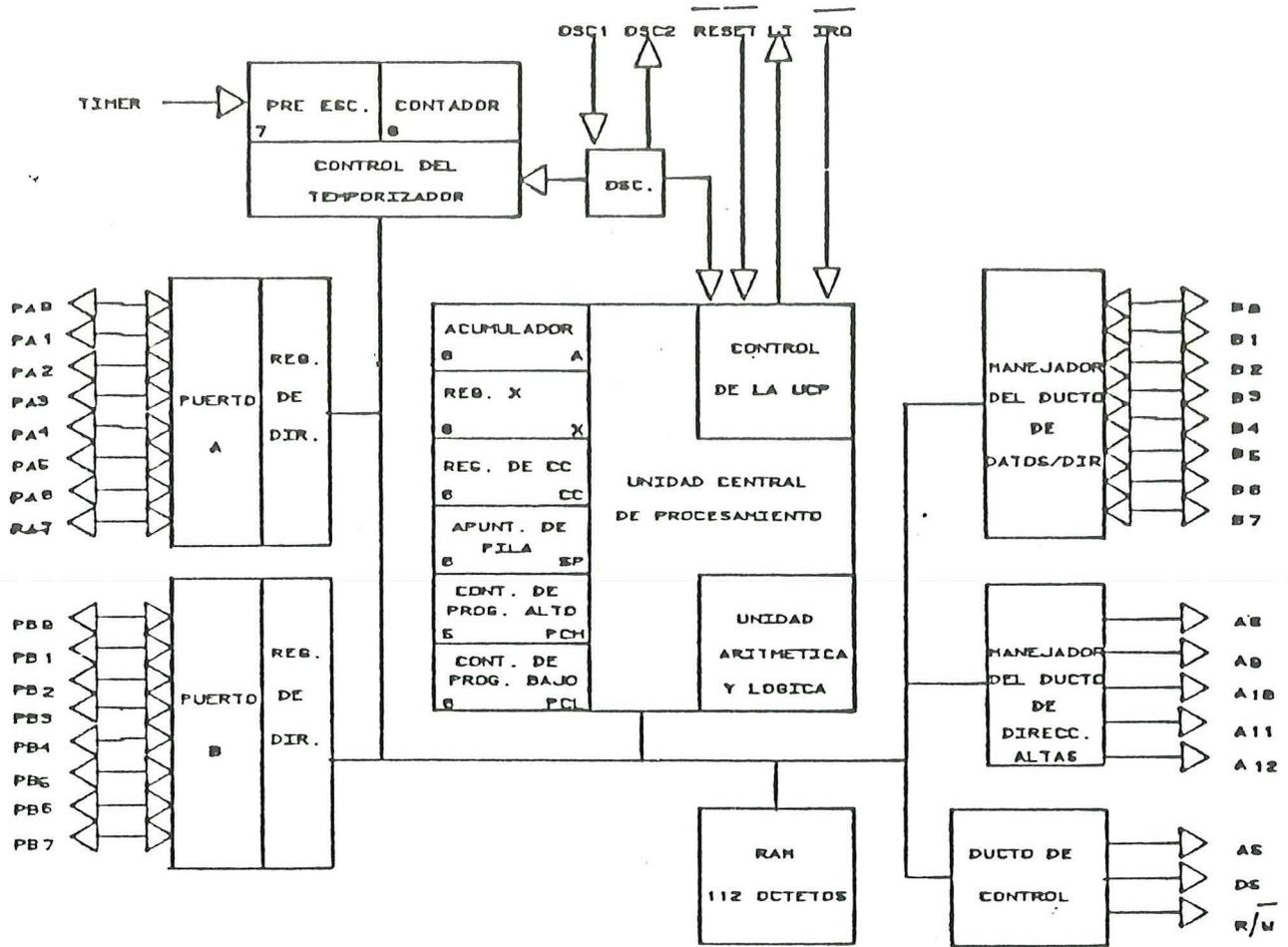


Figura 3. Arquitectura interna del MC146805E2.

dispositivo se puede conectar directamente al ducto de datos/direcciones del procesador, y únicamente requiere de un decodificador de direcciones para su inclusión en la CUTCMOS (selector 1 de 8 en Figura 1).

El RTR puede programarse para llevar a cabo funciones tales como generación de pulsos periódicos y registro del tiempo (Día, mes, año, hora, minuto, segundo). La capacidad de generar pulsos periódicos puede utilizarse para interrumpir al procesador a intervalos que varían desde 122.07 microsegundos ( $1/2^{*}13$  seg.) hasta uno por día (Anón., 1981). Los intervalos de interrupción de menos de uno por segundo se establecen al usar la capacidad de interrupción periódica del RTR que puede seleccionarse en intervalos que sean submúltiplos binarios de un segundo. Si se desean intervalos mayores, éstos se pueden seleccionar utilizando la capacidad del RTR de interrumpir a una hora determinada (función de alarma).

La base de tiempo que controla al RTR se puede seleccionar de un conjunto de frecuencias. En la CUTCMOS, la frecuencia de operación elegida fue la menor posible (32 768 Hz), con objeto de minimizar el consumo de energía total de la tarjeta (Mcbride, D., 1982).

#### 2.4 Memoria de lectura/escritura.

La microcomputadora CUTCMOS contiene 2K octetos de memoria de lectura/escritura (RAM) residente en un dispositivo 6116. Este dispositivo gasta típicamente 10 mW en estado activo y unos cuantos nW cuando no se direcciona (Anón., 1982a).

#### 2.5 Memoria de lectura exclusiva.

La computadora CUTCMOS contiene 2K de memoria de lectura exclusiva reprogramable (EPROM) que está contenida en el dispositivo 27016. Este dispositivo tiene las mismas características funcionales y de programación que la memoria estándar 2716 pero está fabricado con tecnología CMOS, por lo que su consumo de energía es un orden de magnitud menor que la 2716 (Anón., 1982b).

#### 2.6 Manejo del mapa de memoria.

El MC146805E2 contiene únicamente 13 líneas de direcciones, y por lo tanto está limitado a acceder directamente 8K octetos de memoria. Este hecho no se refleja en la capacidad de manejo de memoria de la CUTCMOS, ya que la tarjeta se ha diseñado para que pueda direccionar hasta 32K de memoria externa, aparte de la memoria incluida

en la tarjeta.

La figura 4 ilustra la forma en que se maneja la memoria en la CUTCMOS y la manera mediante la cual se pueden direccionar los 32K adicionales. En primer lugar se puede observar que el espacio de 8K direccionado por el microprocesador se ha dividido en 4 segmentos de 2K cada uno, seleccionados por el selector 1 de 4 de la Figura 1. El primer segmento, correspondiente a las direcciones 0-7FF, direcciona la memoria interna del microprocesador e incluye el manejo de los puertos de entrada/salida y el temporizador, además del control del RTR. El segundo y cuarto segmentos están asignados para direccionar la RAM y EPROM en la tarjeta, respectivamente. El tercer segmento se utiliza para direccionar memoria externa a la tarjeta en bancos de 2 K que son seleccionados por las líneas de entrada/salida PA0, PA1, PA2 Y PA3, que actúan como las líneas de dirección externa A11, A12, A13 Y A14, respectivamente. De esta forma, cada uno de los 16 bancos de 2K de memoria externa que pueden seleccionarse, comparte las localidades 1000-17FF del procesador y es controlado por un código de 4 bits que puede manejarse por programación.

DIRECCIONES AL DUCTO C-44 BITS DE SELEC.  
DE BANCO

BANCOS DE PAS PA2 PA1 PA0  
MEMORIA EXT. A14 A13 A12 A11

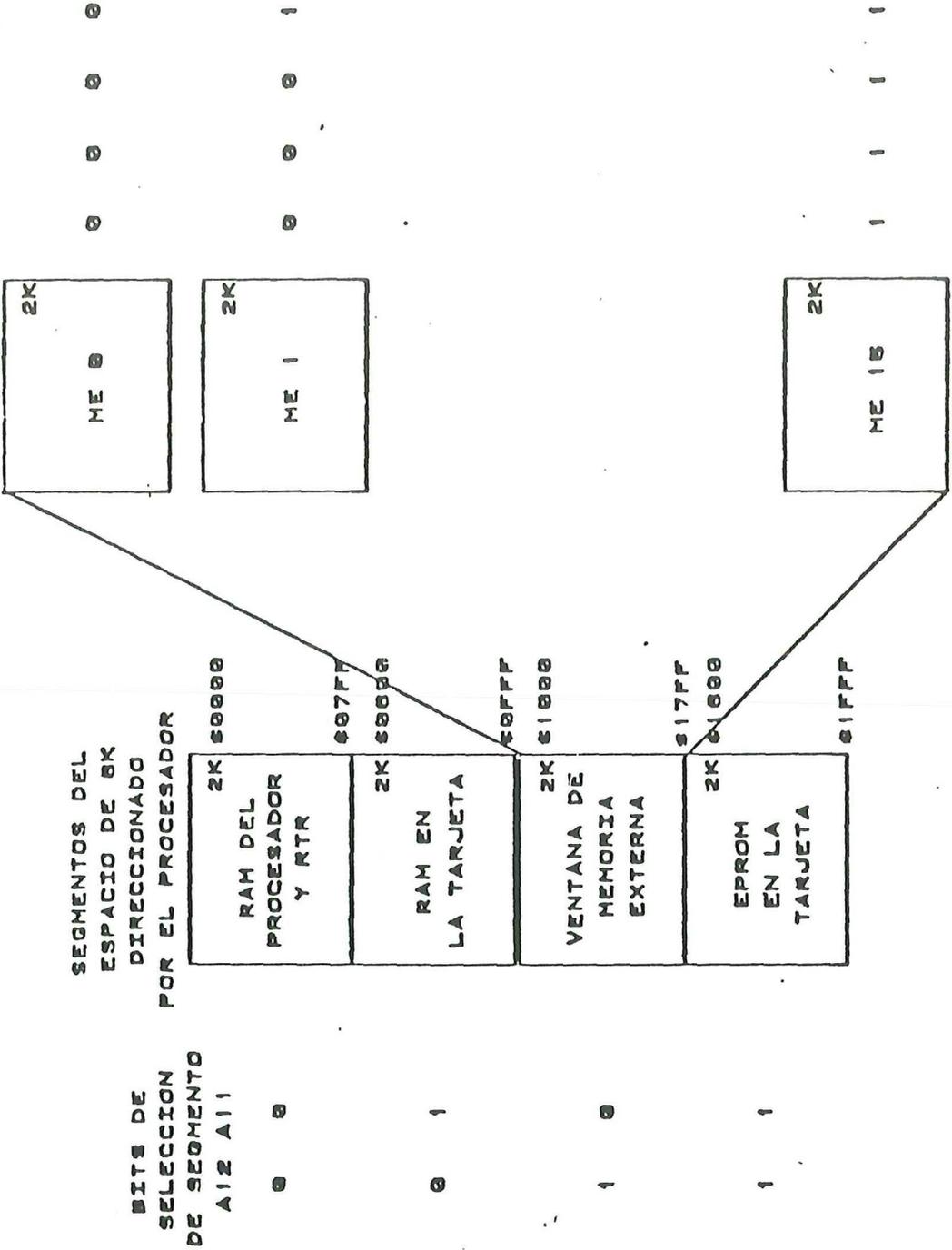


Figura 4. Manejo de memoria de la CUTCMOS.

## 2.7 Conexiones externas.

La CUTCMOS ha sido diseñada para formar parte de un sistema modular en el cual diversas tarjetas se puedan comunicar a través de un ducto (bus) común. El ducto empleado es un ducto de 44 líneas denominado C-44 (Ver apéndice II), que ha sido propuesto por la compañía ONSET Corp. como estándar para microcomputadoras de bajo consumo de energía (Anón., 1982). Este ducto conecta las señales de datos y direcciones del microprocesador, así como señales de sincronización y control para permitir la comunicación entre módulos. En la CUTCMOS, las conexiones al ducto C-44 están reforzadas. Esto hace que la carga capacitiva del ducto se mantenga en un valor bajo y permite conectar módulos de expansión sin causar retrasos excesivos en las señales transmitidas.

La CUTCMOS posee además un conector de 26 patas destinado para usos generales al cual están conectadas, entre otras señales, las 16 líneas de entrada salida del microprocesador (Ver apéndice I).

## 2.8 Interrupciones externas

A diferencia de otros diseños de Motorola, el MC146805E2 contiene sólo una línea de interrupción externa

(no existe NMI ni FIRQ). Esta línea se denomina IRQ y es activada por un nivel bajo. Las interrupciones en el MC146805E2 pueden ser autorizadas o inhibidas por programación, y son atendidas en la misma forma que la IRQ del 6800. La CUTCMOS tiene capacidad para recibir una línea de interrupción externa, la cual está conectada directamente a la línea IRQ del MC146805E2.

## 2.9 Modos de muy bajo consumo de energía.

La microcomputadora CUTCMOS puede operar en dos modos en los que su consumo de energía se reduce al mínimo y en los que la actividad del microprocesador está prácticamente cancelada. Estos modos resultan útiles en aplicaciones donde no es necesario que el procesador esté operando continuamente, y pueda suspender su actividad en espera de algún evento externo. Los modos se derivan directamente de las instrucciones WAIT y STOP del MC146805E2.

### 2.9.1 Modo WAIT.

En el modo WAIT, el procesador cancela sus accesos al ducto, pero el reloj interno se mantiene en operación. Esto reduce el consumo de energía de la tarjeta en un factor de 10. La tarjeta puede ser 'despertada' por una interrupción externa, una interrupción del temporizador o

una inicialización general (RESET).

### 2.9.2 Modo STOP

El usar la instrucción STOP en el MC146805E2 causa una reducción mucho más drástica en el consumo de energía del sistema, ya que esta instrucción hace que el microprocesador realmente detenga toda su actividad. Esto se debe a que el oscilador que controla los ciclos de acceso a la memoria se detiene. En estas condiciones, el consumo de energía de la tarjeta se reduce a menos de 1 mW. El modo STOP se termina cuando se activa la línea de RESET, o se efectúa una interrupción externa. El tiempo de recuperación del sistema, depende del tiempo de recuperación del oscilador, y es típicamente de unos 15 milisegundos.

### 2.10 Conclusiones.

Este capítulo ha descrito la organización de la circuitería del controlador programable, objeto de este trabajo. De la descripción anterior, se deriva que este controlador, constituye una microcomputadora completa por sí sola. En su diseño se han aprovechado al máximo las características del microprocesador empleado, para lograr una tarjeta que se pueda utilizar en el control de

instrumentos y procesos. Además, se alivió la limitación del manejo de sólo 8 K de memoria mediante el direccionamiento de bancos externos.

En el capítulo siguiente se dará una descripción del sistema de programación que complementa la circuitería de la CUTCMOS para orientar su aplicación a propósitos de instrumentación y control.

### 3 PROGRAMACION DE DIAGRAMAS DE ESTADO.

#### 3.1 Introducción.

Se describe aquí, un lenguaje de estados para desarrollar programación para aplicaciones de control que son fácilmente representadas por diagramas de estados. Este método permite pasar directamente de los diagramas de estados a un lenguaje de alto nivel apropiado, independiente del procesador que se esté utilizando.

Después de llevar a cabo la descripción se presenta la implementación particular que se llevó a cabo en el sistema EXERCISER II del Laboratorio de Instrumentación Geofísica de la División de Ciencias de la Tierra del CICESE. El sistema se implementó en el Microensamblador RASMOS.

#### 3.2 Aspectos generales

Existe una gran diversidad de formas a través de las cuales uno se aproxima a la tarea de desarrollar programación para microprocesadores. El más común de los enfoques es el de simplemente escribir el programa en lenguaje ensamblador basado en cualesquier metodología con la que esté familiarizado el programador. Por otra parte, la programación estructurada basada en el uso de los lenguajes de alto nivel apropiados, como PASCAL, es de

ayuda. Sin embargo, en muchos casos se prefiere utilizar técnicas de lenguaje ensamblador ad hoc porque:

1. El programador tiene poco entrenamiento en programación y las técnicas estructuradas pueden parecerle un poco elevadas.
2. El sistema de desarrollo utilizado puede carecer de un compilador de lenguajes de alto nivel o puede ser que produzca demasiado código y ocupe mucha memoria.
3. El lenguaje en sí, puede parecer inapropiado ya que no está relacionado con los elementos de la circuitería que controla el microprocesador.

La técnica aquí descrita, basada en la descripción de estados, es un punto intermedio entre las dos. Los diagramas de estados son una forma natural de describir máquinas o procesos que trabajan en forma secuencial y forman parte del legajo de conocimientos de cualquier persona entrenada en el diseño digital. Esta aproximación permite la generación de código bastante eficiente directamente a partir del diagrama de estados, utilizando únicamente el programa ensamblador disponible en la mayoría

de los sistemas de desarrollo de microprocesadores. Se pueden obtener beneficios tales como los atribuidos a los lenguajes estructurados y se puede generar código tanto o más eficiente que el desarrollado directamente en el lenguaje ensamblador.

Los diagramas de estados son ampliamente utilizados en el diseño y análisis de sistemas digitales. En este capítulo veremos como aplicar estas técnicas a ciertos tipos de programación y nos introduciremos a un lenguaje de estados que posea las siguientes características:

1. Describa diagramas de estados, en una forma conveniente y entendible para la máquina y el usuario
2. Traduzca fácilmente la información contenida en los diagramas de estados a una forma tabular (llamada tabla de estados) por medio de un programa fácilmente adaptable en un macroensamblador.

Se describe también un pequeño programa en ensamblador (Procesador de estados) que interpreta la tabla de estados. El procesador de estados evalúa las condiciones requeridas

para las transiciones de estados y lleva a cabo las acciones descritas en la tabla de estados para esa transición.

### 3.3 Las técnicas de descripción de estados.

El concepto de estado tal y como lo conocemos en circuitería digital, se refiere a la colección de los valores de los elementos de memoria de un circuito en un instante dado. De aquí que un circuito de  $N$  biestables tenga, en teoría  $2^N$  estados posibles. La información relativa a los estados y entradas previos que necesita recordar el circuito para responder correctamente a entradas futuras, se encuentra implícita en los estados del sistema.

Un circuito se diseña de tal forma que para un estado dado, digamos  $S_a$ , una entrada específica le causa un salida  $s$  y una transición a un nuevo estado  $S_b$ . Este ejemplo se muestra en la figura 5, para un circuito sencillo conteniendo sólo dos biestables,  $A$  y  $B$ . El estado  $S_a$  está representado por  $A$  encendido y  $B$  apagado y el estado  $S_b$  por el caso opuesto.

En la figura 6 se muestra como se representa la situación anterior en un diagrama de estados. Cada estado está representado por un círculo en cuyo interior se escribe el nombre del estado, con líneas que interconectan los estados indicando posibles transiciones. Cada línea de transición

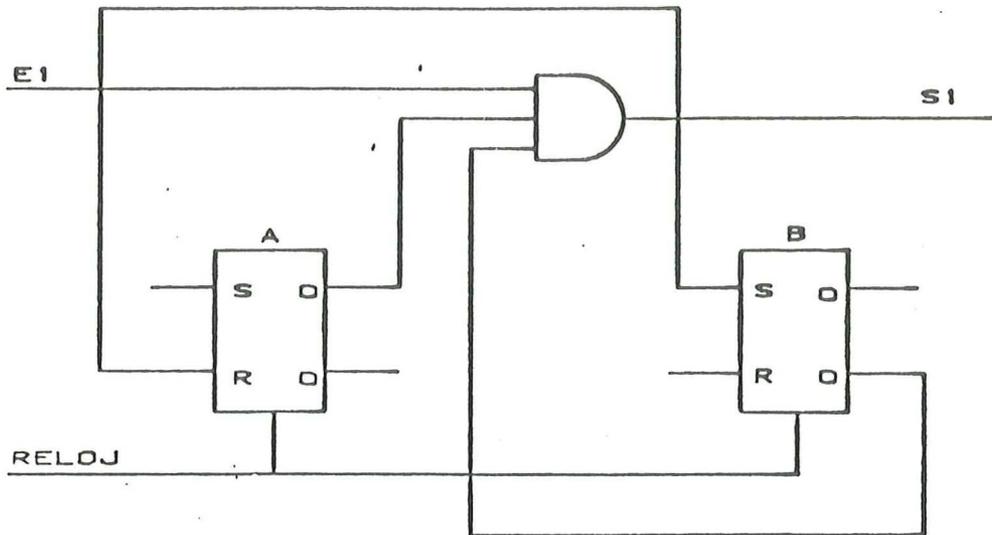


Figura 5. Diagrama de un circuito secuencial.

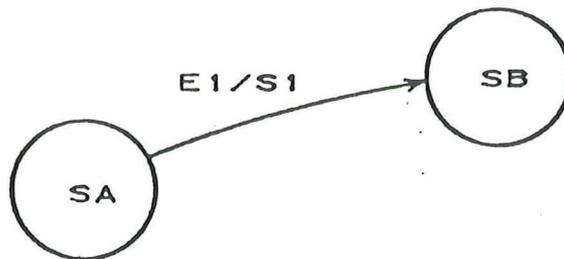


Figura 6. Diagrama de estados.

tiene asociada la entrada que la provoca y, separada por una diagonal, la salida que activa.

La figura 7 muestra un diagrama de flujo que ilustra la misma situación desde el punto de vista programación. Aquí suponemos que la computadora está controlando a una máquina que debe esperar en un estado  $S_0$  hasta que alguna de las dos condiciones es satisfecha. Dependiendo de qué condición ocurra, la máquina ejecutará ciertas acciones y pasará a alguno de los nuevos estados. Las condiciones  $c_1$  y  $c_2$  pueden referirse a alguna propiedad de la máquina que el programa detecta a través de la entrada de algún sensor, a la entrada de un operador, o sólo a esperar algún tiempo determinado.

A diferencia de un circuito digital, donde todas las entradas pueden cambiar casi simultáneamente, una computadora ejecuta sus tareas secuencialmente, de tal forma que las acciones desencadenadas por una transición de estado deben mostrarse como un lista de acciones que se llevan a cabo en un orden dado.

#### 3.4 Aplicaciones apropiadas.

Aunque se pueden utilizar los diagramas de estados para describir cualquier tipo de circuito digital secuencial,

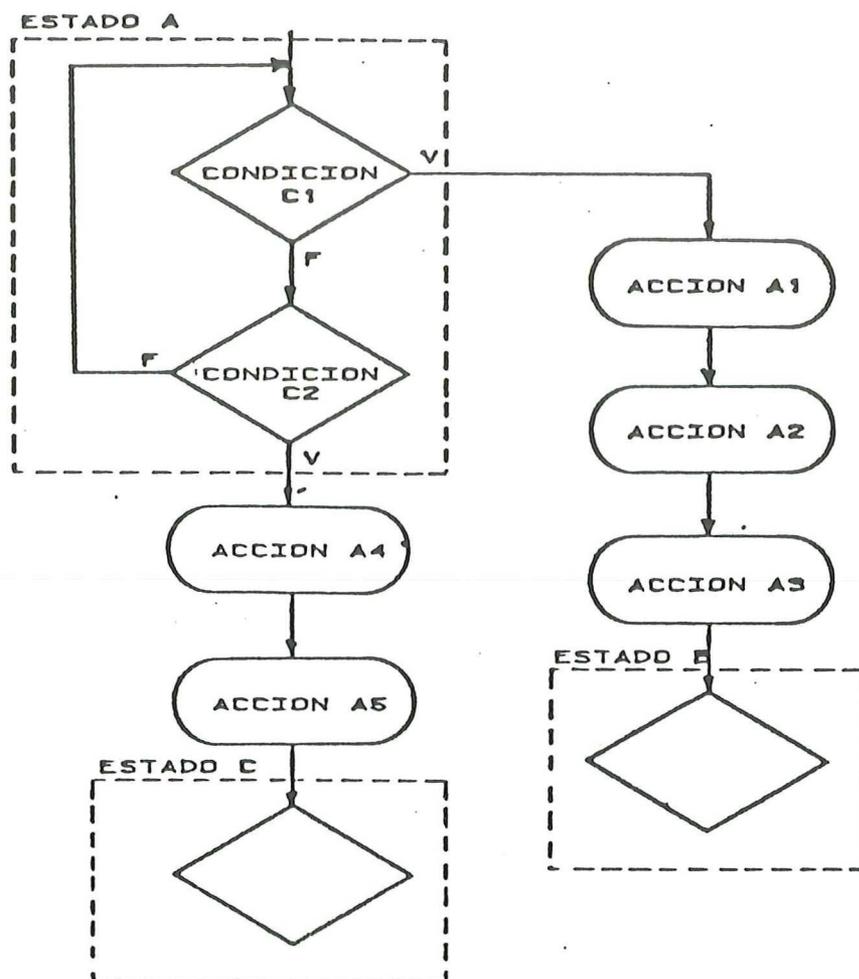


Figura 7. Programacion de estados representada en diagrama de flujo.

éstos típicamente se utilizan cuando hacen más claro el funcionamiento del circuito para el diseñador. Este mismo razonamiento se puede aplicar a la programación. La técnica de diagramas de estado, junto con el lenguaje de estados y el procesador de estados, sólo es útil para programas que se pueden visualizar convenientemente como secuencias de estados. Esto rige generalmente para las aplicaciones de control en instrumentación.

### 3.5 Descripción del lenguaje de estados.

Pasaremos ahora a la descripción de los conceptos elementales que forman parte del lenguaje de estados.

Los elementos esenciales del lenguaje de estados son las declaraciones ESTADO y CONDIC. Cada declaración ocupa una línea y está compuesta por un campo de nombre, un campo de mando y un campo de argumento.

#### 3.5.1 Declaración ESTADO.

La declaración ESTADO identifica el principio de la descripción de un nuevo estado y le asigna un nombre a este estado a través del campo de nombre.

nombre      ESTADO      parámetros opcionales

Los parámetros opcionales que aparecen en la declaración ESTADO pueden utilizarse para identificar eventos que ocurren en un gran porcentaje de los estados, condiciones que se activan en un gran porcentaje de los mismos, o mensajes o datos que deben de aparecer en alguna salida.

### 3.5.2 Declaración de condición.

Generalmente, una o más declaraciones de condición preceden a la declaración de ESTADO para describir la forma en que han de llevarse a cabo las transiciones entre estados. En este caso no se utiliza el campo de nombre:

tipo      identificador, LISTA DE ACCIONES, SIGEST

La condición es identificada por el campo tipo junto con el identificador. En el caso de que se trabaje con máquinas muy sencillas que contengan pocas condiciones, basta con que exista un solo tipo y se identifique a las condiciones por el identificador. Sin embargo, cuando existen muchas condiciones es conveniente agruparlas de

acuerdo a sus funciones, y se puede asignar un tipo para cada función para mejorar la legibilidad. En algunos casos el lenguaje de estados requiere de efectuar llamadas que son siempre ciertas. Por claridad conviene utilizar un tipo especial que se podría denominar SIGUE, donde se omiten los argumentos del identificador.

El argumento SIGEST indica el nombre del que deberá ser el siguiente estado cuando la condición descrita se cumpla. Si la condición no se cumple se evalúa la siguiente declaración de condición. Si ninguna de las condiciones es cierta, la máquina permanece en el mismo estado, y se procede a evaluar de nuevo la primera declaración de condición.

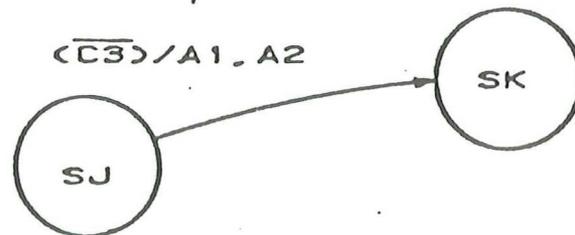
La LISTA DE ACCIONES contiene los nombres de las subrutinas que deberá llamar el procesador de estados en el momento en que la condición descrita en la declaración de condición sea verdadera. Esto corresponde a las salidas provocadas por una transición de estado. Estas acciones son las que se deberán ejecutar antes de pasar al siguiente estado. El número de elementos en la LISTA DE ACCIONES no está limitado, e inclusive puede ser nulo.

### 3.6 Funciones lógicas.

Como se ha hecho notar, cada una de las condiciones de un estado son evaluadas secuencialmente por el procesador de estados. Este hecho nos lleva a una situación completamente distinta a la de los circuitos digitales, donde las condiciones de cada estado son evaluadas simultáneamente. En la figura 8 se muestra como implementar una función negada aprovechando el hecho de que la primera declaración de condición siempre es evaluada antes que la segunda. Mientras la condición  $C3$  es válida, la máquina permanece en el mismo estado  $Sj$ . La segunda condición es evaluada únicamente cuando  $C3$  no es verdadera y entonces se transfiere a la máquina al estado  $Sk$  (incondicionalmente) ejecutando las acciones  $A1$  y  $A2$ .

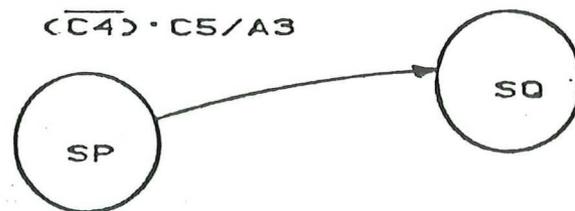
En el ejemplo de la figura 9 se utiliza el mismo concepto para lograr una función 'Y'. No se evaluará ninguna declaración de condición hasta que la condición  $C4$  sea falsa. La transferencia del estado  $Sp$  al estado  $Sq$  requiere entonces que también la condición  $C5$  sea verdadera, en cuyo caso se ejecutará la acción  $A3$ .

En la práctica, la necesidad para este tipo de funciones lógicas es relativamente poca, suponiendo que se eligen apropiadamente los tipos de las condiciones. Estas técnicas, que emplean las propiedades inherentes al procesamiento secuencial del procesador de estados, son



SJ ESTADO  
 CONDIC C3, SJ  
 CONDIC SIGE, A1, A2, SK

Figura 8. Implementación de la función negado.



SP ESTADO  
 CONDIC C4, SP  
 CONDIC C5, A3, SQ

Figura 9. Implementación de la función 'Y'.

preferibles a tener que hacer más complicado el mismo procesador de estados, por lo que se requeriría de más memoria y de tiempos de ejecución mayores.

### 3.7 Traducción del lenguaje de estados.

Es importante y deseable que la traducción del lenguaje de estados a la tabla de estados codificada se ejecute en forma automática. Esto elimina virtualmente la posibilidad de errores humanos y ahorra una cantidad considerable de tiempo cuando se requiere de una gran cantidad de estados. Es importante también el poder utilizar la programación estándar con la que se cuenta en la mayoría de los sistemas de desarrollo, ya que de otra forma la técnica del lenguaje de estados no sería una herramienta de fácil utilización en el desarrollo de programación, como se pretende.

El definir las declaraciones ESTADO y CONDIC con campos como los utilizados por el ensamblador, nos permiten implementar un programa en el cual, mediante el uso de la definición de macroinstrucciones, se pueda lograr una traducción automática del lenguaje de estados a la tabla de estados.

Para lograr esto hemos de definir concretamente los requerimientos formales de cada una de las declaraciones.

### 3.7.1 La macroinstrucción ESTADO.

La figura 10 muestra los requerimientos para la traducción del lenguaje de estados.

La Macro ESTADO tiene como principal objeto el definir una dirección asociada al nombre que aparece en el campo nombre de la declaración ESTADO. Esta dirección y el nombre asociado son almacenadas en la tabla de símbolos del ensamblador. Los parámetros opcionales pueden ser codificados típicamente en el primer octeto de el estado.

### 3.7.2 La Macroinstrucción de condición.

La macro de condición debe, para cada tipo de condición, generar un octeto para identificar el tipo y el identificador relacionados a esta condición. Los mnemónicos a utilizar en los distintos argumentos deberán ser igualados previamente a los valores deseados del código de la tabla de estados. A continuación se escribe la dirección relativa de cada una de las subrutinas de la LISTA DE ACCIONES. Para esto es necesario conocer el origen de las subrutinas y la dirección absoluta de cada una de ellas.

Se utilizarán sólo los últimos 7 bits de cada octeto ya que el octavo se empleará como bandera para indicar que es la última acción de la lista. Este bit puesto en '1'

TABLA DE ESTADOS

nombre ESTADO parametros op.	PARAMETROS
CONDIC ID, ACC, SEST	IDENTIFICADOR
	ACCION A3
	ACCION A2
	SIG. ESTADO
CONDIC ID, A4, SJ	IDENTIFICADOR
	ACCION A4
	ESTADO SJ
ESTADO FINAL	00
nombre ESTADO	IDENTIFICADOR
CONDIC ID, A1, A2, SK	ACCION A1
	ACCION A2
	ESTADO SK
ESTADO FINAL	00

Figura 10. Traducción del lenguaje de estados a la tabla de estados.

indicará que es la última acción. Esto nos da hasta 127 posibles entradas a la subrutinas de acciones. En caso de que la lista sea nula se dará por omisión una entrada nula (80).

Finalmente se coloca la dirección absoluta del estado correspondiente a SIGEST en los siguientes dos octetos.

Para permitir un número cualquiera de declaraciones de tipo CONDIC en cada estado, se ha reservado el código 00 para indicar el final del estado. Este octeto es generado mediante la macro ESTADO con la palabra FINAL en el campo de los parámetros opcionales.

## 2.8 Beneficios derivados del uso de la técnica de descripción de estados

Es claro, como se mostró en la figura 7, que el utilizar los diagramas de estado es una forma bastante más compacta de describir un proceso secuencial que el método de diagramas de flujo. Además, una vez que se ha generado este diagrama, se evita una gran cantidad de trabajo, ya que sólo se tiene que transcribir el diagrama al lenguaje de estados.

En la mayoría de los diseños de aplicaciones de microprocesadores no es posible el uso de los programas de alto nivel tales como FORTRAN o BASIC, ya sea porque no

están a la mano o porque el código objeto que generan es bastante grande y ocupa gran cantidad de memoria.

El uso del lenguaje de estados es mejor que estas técnicas debido a que:

1. Es más natural para la descripción de las máquinas secuenciales. Este lenguaje de estados parece ser más comprensible para las personas que no están familiarizadas con el desarrollo de programación (Gander, J. G., Liechti, H. U., 1981).

2. Impone una cierta estructura al programa resultante, logrando algunos de los beneficios de la programación estructurada. Es más fácil coordinar equipos de programadores que trabajan en un mismo proyecto y hay menos oportunidad de que se influencie el código con el estilo propio del programador.

3. Se obtienen algunos de los beneficios de los lenguajes de alto nivel al sustituir con una línea de declaración varias líneas de código de ensamblador.

Además de los beneficios obtenidos por los diagramas de estados como una fuente de programación, se obtienen una serie de beneficios extra en el curso de la implementación.

1. El lenguaje de estados es fácilmente transportable entre diferentes microprocesadores, ya que no contiene ninguna información relacionada con el conjunto de instrucciones del microprocesador.
2. Como en la mayoría de las técnicas que utilizan tablas, la descripción de estados requiere menos memoria que la que requiere en lenguaje ensamblador.
3. El procedimiento para verificar el programa se hace de una manera ordenada debido a la organización del sistema. La operación del procesador de estados es simple y puede ser verificada rápidamente. Las subrutinas que evalúan las condiciones y ejecutan las acciones se verifican individualmente, y finalmente es más sencillo verificar el correcto funcionamiento del diagrama de estados sabiendo que las subrutinas funcionan correctamente.

El beneficio máximo se logra cuando se utilizan todas las técnicas juntas, los diagramas de estados, el lenguaje de estados, la tabla de estados y el procesador de estados.

Estas técnicas no requieren de ninguna programación o circuitería complejas para su implementación y, el trabajo extra desarrollado se justifica claramente con los beneficios obtenidos al generar programas estructurados y ahorro de memoria.

### 3.9 Implementación del Procesador de Estados.

En la figura 11 se muestra el diagrama de flujo de las acciones que debe ejecutar el procesador de estados para la tabla de estados descrita en la figura 10. Al inicio de cada estado, el procesador de estados debe verificar la existencia de parámetros iniciales. Si estos existen, tomar las medidas correspondientes. Si no hay parámetros iniciales, proceder a la evaluación de la primera condición. Cada condición es evaluada en secuencia. Cuando una condición resulta falsa, se brinca a evaluar la siguiente condición.

Si el procesador llega al final del estado y no se presentó ninguna condición verdadera, se regresa al principio del estado y reinicia la evaluación de condiciones. Cuando se alcanza una condición que se evalúa como verdadera, se

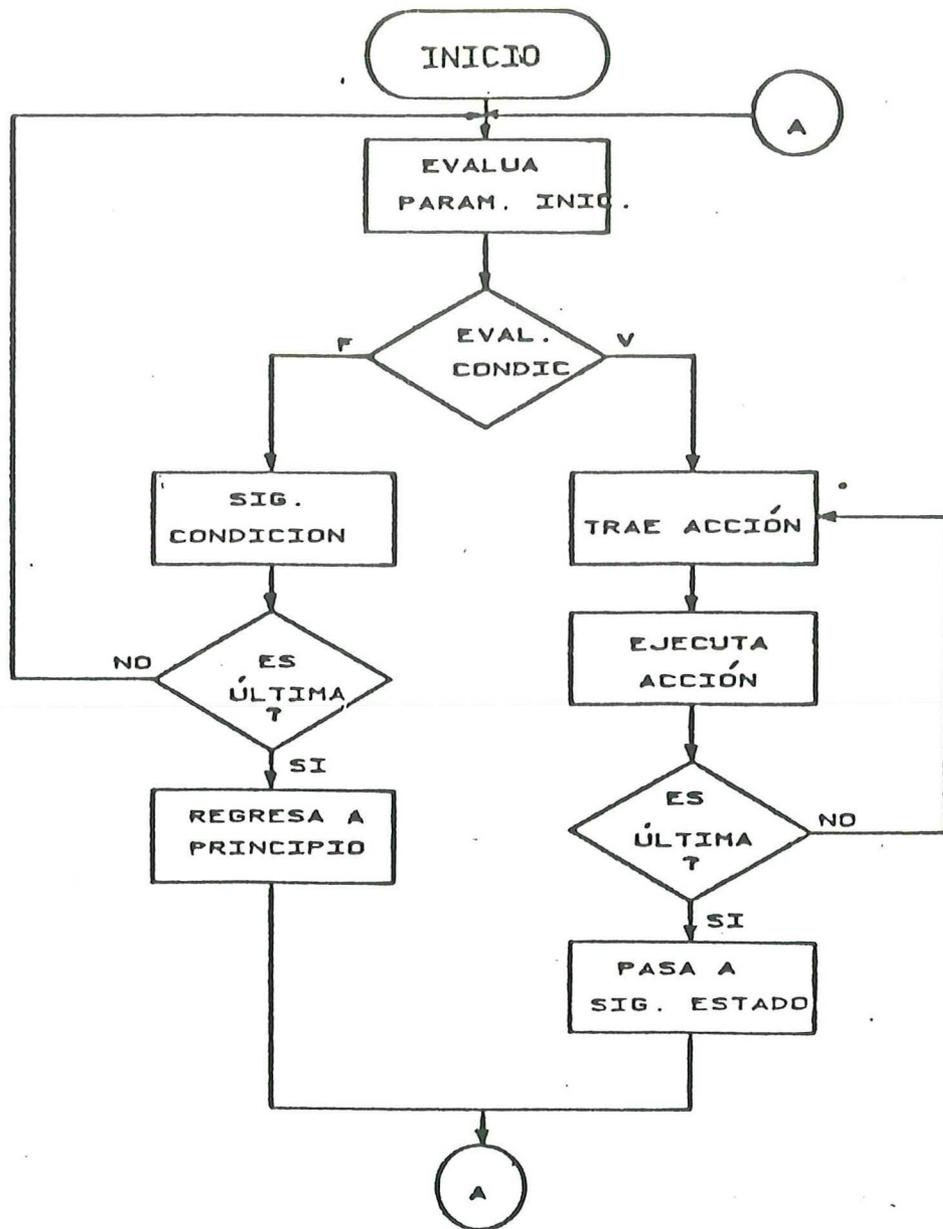


Figura 11. Diagrama de flujo del procesador de estados.

procede a ejecutar las acciones de la LISTA DE ACCIONES y al término de ellas se transfiere el procesador al estado marcado como siguiente en esa condición.

### 3.9.1 Subrutina principal. (PROCED)

La rutina principal del procesador de estados es la encargada de efectuar las transiciones entre los estados. Primero, se inicializa el apuntador que va a recorrer el estado que se va a procesar. Se llama a la subrutina MAND y se inicia el procesamiento del dato que se adquirió con MAND.

Primero se pregunta por la palabra de final de estado (OO), si es final se transfiere el control del programa a la finalización y se retorna de la llamada de la subrutina. En caso de no ser final se prosigue con el procesamiento. Se pregunta si no hay condiciones iniciales. En este caso las condiciones iniciales son el valor de la repetición ENE en caso de que se requiera. De no haber condiciones iniciales, se hace la llamada a la subrutina CONDIC que se encarga de determinar si la condición presentada es la que se espera. La subrutina CONDIC regresa un valor booleano. Este valor será verdadero si la entrada que se procesa es la que se espera o será falso en caso contrario.

Si el valor es falso, el procesador se regresa a conseguir otro dato de la tabla de estados y procesarlo. Si el valor es verdadero, se transfiere el control del procesador a la rutina EXECU.

### 3.9.2 Rutina de ejecución de acciones. (EXECU)

Esta rutina se encarga de hacer las llamadas a las subrutinas especificadas en la tabla de estados para la condición que se acaba de declarar verdadera.

Se incrementa el valor del apuntador del estado y se adquiere otro dato de la tabla de estados.

Se interroga primero si es la última condición, de ser así, se prende la bandera que indica última condición. En caso de no ser la última se procede a hacer la llamada. Para hacer la llamada se genera el apuntador a la subrutina con el dato que se adquirió de la tabla y se efectúa la llamada. Al regresar de la llamada se pregunta por la bandera de última, si fué la última se procede a transferir el valor del nuevo estado a los registros de estado corriente, se pone en ceros el apuntador que recorre el estado y se hace el retorno de la llamada a la subrutina. Si no fué la última llamada, se procede a adquirir el dato de la siguiente llamada y se repite el proceso.

### 2.9.3 Subrutina de validación de condición. (CONDIC)

Esta subrutina se encarga de verificar que la entrada que se está procesando corresponda a la que se espera según los datos de la tabla de estados. En caso afirmativo, se retorna con una bandera en estado verdadero y en caso de no coincidir se regresa el valor falso. En el caso particular del reconocimiento de la WWVB, las entradas pueden ser :

- a. No importa, en cuyo caso siempre se regresa con valor verdadero.
- b.  $ENE = 0$ , que será verdadera cuando el valor de la variable ENE sea igual a 0.
- c.  $ENT = Condición$ , en este caso se compara la entrada que se está procesando con la entrada que se espera en ese momento (Uno, Cero o Marca) y si coinciden se declara condición verdadera.

### 3.9.4 Subrutina MAND.

Esta subrutina es la encargada de adquirir los datos del estado que se está procesando.

### 3.10 Implementación del traductor del lenguaje a la tabla de estados.

Como se mencionó en la descripción, es deseable que la traducción del lenguaje a la tabla de estados se haga en forma automática.

Para tal efecto se han implementado las dos macroinstrucciones definidas anteriormente. Se diseñó la macroinstrucción ESTADO y la macroinstrucción CONDIC, ambas implementadas en el macroensamblador RASMO5 de Motorola.

#### 3.10.1 ESTADO

Esta macroinstrucción tiene el siguiente formato:

nombre del estado    ESTADO    parámetros opcionales

Donde el campo nombre del estado coincide con el campo de etiqueta, ESTADO coincide con el campo de

instrucción, y parámetros opcionales coincide con el campo del operando.

### 3.10.2 CONDIC

Esta macroinstrucción tiene el siguiente formato

CONDIC identificador, lista de acciones, estado sig.

CONDIC corresponde al campo de instrucción, identificador, lista de acciones y estado siguiente corresponden al campo de operando. Por restricciones propias del macro sólo se permiten un máximo de 8 elementos en la lista de acciones. Todos los elementos que aparecen en el campo del operando deben de ir separados por comas (,). Para hacer uso de las macroinstrucciones es necesario lo siguiente:

1. Dar el origen de la tabla de estados antes de la primera declaración ESTADO.
2. Igualar los códigos asignados a los identificadores de condición que aparecen en el

campo de identificador de la macro CONDIC, e igualar los códigos asignados a los parámetros de la macro ESTADO.

3. Dar el origen y la dirección de cada una de las subrutinas que se van a ejecutar como acciones y que se van a especificar en la lista de acciones de la declaración CONDIC.

Para una más fácil utilización de estas herramientas se recomienda lo siguiente:

1. Implementar en primer término el programa principal.
2. Colocar el procesador de estados
3. Colocar las subrutinas de evaluación de condición.
4. Colocar las subrutinas correspondientes a la lista de acciones.
5. Colocar las macroinstrucciones ESTADO y CONDIC.

6. Colocar las declaraciones correspondientes a la tabla de estados que se desea implementar.

Es necesario también reservar las direcciones 0010, 0011, y 0012 ya que se utilizan por la subrutina MAND, y colocar como primera subrutina la instrucción de retorno de subrutina, para tomar en cuenta el caso cuando no haya acciones a ejecutar.

Finalmente se recomienda que el programa que capture las entradas a procesar por el método descrito, las capture todas al mismo tiempo y no permita que cambien hasta que se haya completado un circuito completo al estado en proceso. Esto es con el objeto de prevenir saltos aleatorios a estados que no corresponden.

## 4 APLICACION GEOFISICA.

### 4.1 Introducción.

En este capítulo se describe una aplicación del controlador objeto de este trabajo. La aplicación escogida consiste de un reloj que se pone a tiempo con la señal de radio transmitida por la estación WWVB de la Oficina Nacional de Estándares de los E. U. A. localizada en Fort Collins, Colorado.

Esta aplicación se seleccionó en base a los siguientes criterios:

1. El requerimiento de un reloj de este tipo constituye una necesidad inmediata para muchas investigaciones realizadas en la División de Ciencias de la Tierra del CICESE.
2. El desarrollo de este sistema constituye una aplicación típica del controlador diseñado en este trabajo ya que involucra el procesamiento de entradas digitales asíncronas, el control de salidas y el mantenimiento del tiempo.

#### 4.2 Objetivos de diseño

Se propuso que el sistema a desarrollar cumpliera con los siguientes objetivos de diseño:

- 1.- Capacidad para interpretar el código de tiempo transmitido por la WWVB y sincronizar una base de tiempo interna cada vez que se tenga éxito en el reconocimiento de la señal recibida.
- 2.- Mantenimiento de la hora a pesar de no existencia de la sincronización externa.
- 3.- Capacidad para registrar la antigüedad de la sincronización más reciente, a fin de estimar errores en el registro del tiempo.
- 4.- Capacidad para comunicarse con un instrumento digital a través de una interfaz serie. Según el formato comercial 46B-DC de KINEMATRICS (Anón., 1978).

A continuación se describe el formato de la señal de la WWVB, para comprender el diseño con más claridad.

#### 4.3 Formato de la señal de tiempo.

La WWVE transmite continuamente un estándar de Tiempo con una frecuencia portadora de 60 KHz y una potencia efectiva de radiación de 13 Kw (Howe, Sandra, 1976):

El código de transmisión se encuentra sincronizado con la portadora y se transmite continuamente a razón de un pulso por segundo.

Cada pulso se genera reduciendo la potencia de la portadora en 10 dB al inicio de cada segundo, de tal forma que el flanco inicial de cada pulso negativo está 'a tiempo'.

La estación transmite un código de tiempo tipo IRIG B modificado el cual es un código decimal cifrado binario (DCE) en marcos de tiempo de un minuto (Howe, Sandra, 1976)

La información contenida en cada marco consta de 60 marcas y se transmite a razón de una marca por segundo. La información de cada marca se distingue por la duración del pulso, y puede ser de 3 tipos.

1. Cero binario si su duración es de 0.2 segundos.
2. Uno binario si su duración es de 0.5 segundos.
3. Marca de posición de cada 10 segundos o de

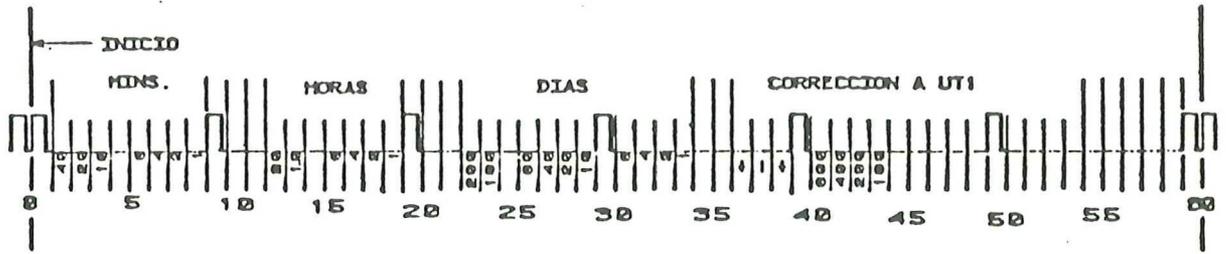
inicio del minuto si su duración es de 0.8 segundos.

Cada minuto, el código presenta la información de tiempo en minutos, horas, día del año y la diferencia existente entre el tiempo transmitido (UTC) y el UT1 (Ver apéndice IV). En la figura 12 vemos que las cifras primera y segunda especifican el minuto de la hora, las cifras tercera y cuarta, la hora del día; las cifras quinta, sexta y séptima, especifican el día del año. Las cifras novena, décima y undécima especifican los milisegundos que deberán ser sumados o restados a la señal de tiempo para obtener la hora UT1. La octava cifra indica el signo de la corrección. Si UT1 se encuentra atrasado (-) con respecto a UTC la marca número 38 será un uno. Si UT1 está adelantado (+) con respecto a la señal de UTC las marcas 37 y 39 serán unos.

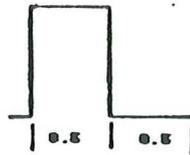
#### 4.4 Actividades del instrumento.

Para que el instrumento sea capaz de cumplir con los objetivos de diseño descritos en el inciso 4.2, deberá de efectuar las siguientes tareas:

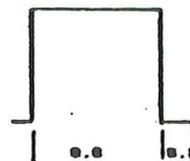
1. Detectar y distinguir los flancos de subida y bajada de la señal de radio, así como determinar la duración de los pulsos.



CERO



UNO



MARCA

Figura 12. Diagrama del código de tiempo de la WWVB.

2. Clasificar la información recibida en ceros, unos, marcas de tiempo o errores de transmisión.
3. Verificar el formato de ocurrencia de las marcas y extraer las cifras del marco de tiempo.
5. Almacenar las cifras recibidas correctamente.
6. Efectuar la corrección UT1 en la referencia de sincronización. (Si así se desea)
7. Sincronizar la base de tiempo interna con la señal recibida.

Finalmente, es necesario que la información de tiempo que se tiene se transmita a otros dispositivos. En nuestro caso específico y debido a que el instrumento desarrollado será compatible con un sistema comercial, existe ya un formato determinado para efectuar la transmisión de las marcas de tiempo (Anón., 1978). Este formato consiste en una cadena de caracteres ASCII enviados en serie bajo la norma RS232C en la siguiente secuencia:

Ctrl A      DDD : HH : MM : SS      Q      CR LF

Donde:

Ctrl A	Es un caracter de control que encabeza la secuencia de datos.
DDD	Especifica el día del año.
HH	Especifica la hora del día.
MM	Especifica el minuto de la hora.
SS	Especifica el segundo del minuto.
Q	Establece la calidad de la señal de tiempo.
CR	Establece la sincronización (el bit de inicio).
LF	Establece el fin de la marca de tiempo.

Esto es, la marca de tiempo se inicia con un caracter de control de inicio de encabezado (SOH o Ctrl. A) seguido inmediatamente por el día del año, la hora del día, el minuto de la hora, y el segundo del minuto, cada cifra separada por dos puntos (:) posteriormente se envía un caracter que indica la calidad de la hora bajo la siguiente convención:

Caracter	Calidad
?	Error de +- 500 milisegundos.
#	Error de +- 50 milisegundos
*	Error de +- 5 milisegundos

Error de  $\pm 1$  milisegundo

ESPACIO Error de menos de un milisegundo.

Finalmente se transmite un retorno de carro (CR), que proporciona la sincronización al momento de aparecer su bit de inicio y para terminar el marco de datos se transmite un avance de línea (LF).

#### 4.5 Sistema propuesto.

En la figura 15 se muestra un diagrama a bloques del sistema desarrollado. El sistema consta de un receptor comercial capaz de recibir la señal de la WWVB. La señal recibida es alimentada a la microcomputadora CUTCMOS que se encarga de extraer la información del tiempo del código de la WWVB y sincronizar la base de tiempo interna. La señal de tiempo registrada por el sistema puede desplegarse en un visualizador numérico luminoso y/o transmitirse a un instrumento digital a través de una interfaz serie RS232C.

Una vez inicializado el sistema, la CUTCMOS va a estar esperando la ocurrencia de alguna de las siguientes señales para que trabaje el proceso correspondiente.

Por una parte se reciben las señales de la WWVB, la interfaz serie (UART), y la interrupción externa

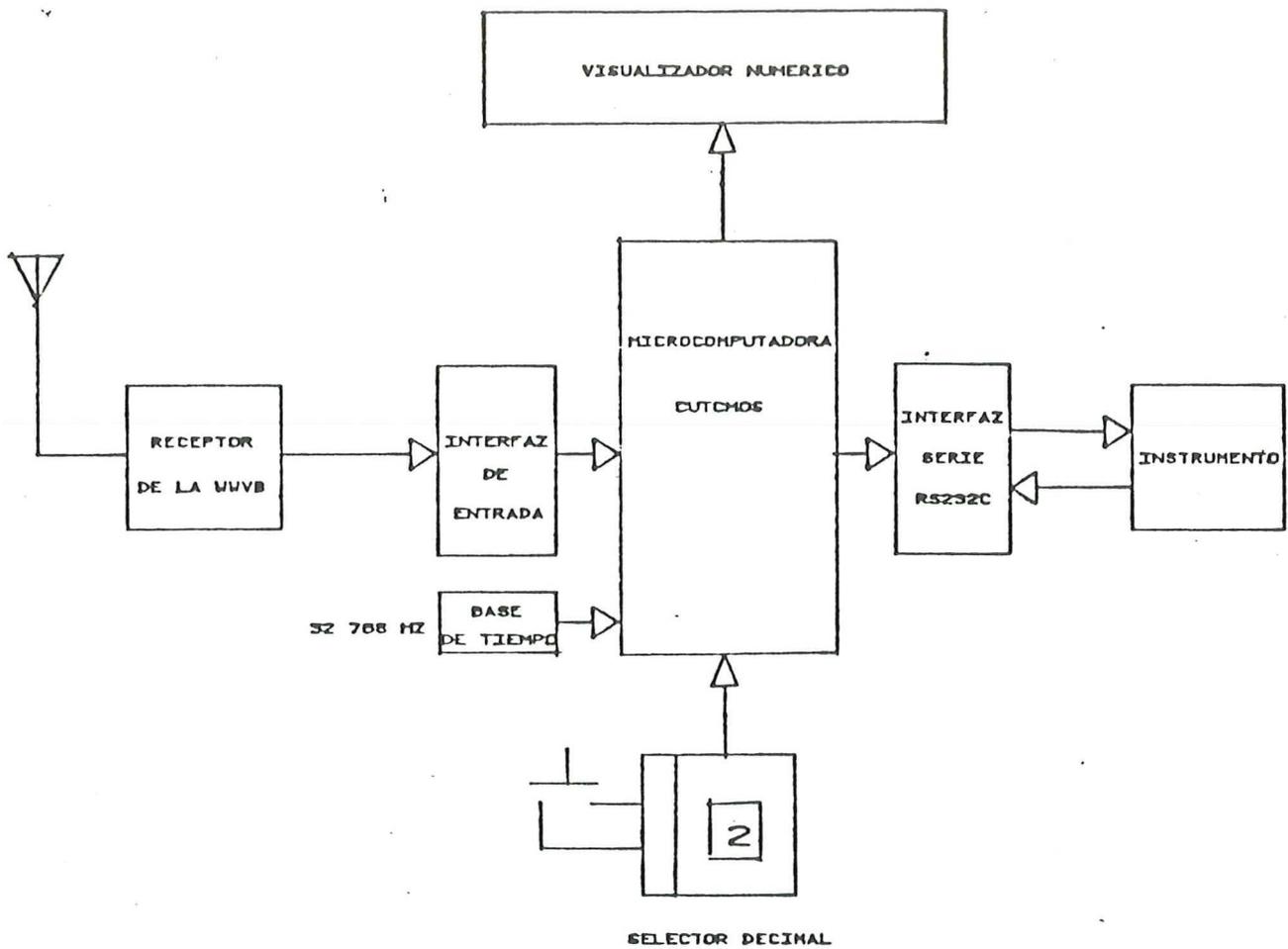


Figura 13. Diagrama a bloques del reloj sincronizado con la WWVB.

(botón).

Y por otro lado se tiene la interrupción del temporizador. La separación entre estas señales se debe a que en el procesador, estas interrupciones vienen por líneas distintas. Las 3 primeras llegan por la línea IRQ y la segunda por una línea interna. Cada una de estas interrupciones tiene su propio vector, de manera que son atendidas en forma independiente por el microprocesador.

En la línea IRQ concurren tres señales distintas, WAVE, CART, y ECTON. Dado que cada una de ellas requiere de atención individual se tiene la necesidad de distinguirles mediante un programa distribuidor de tareas. Este programa tiene la responsabilidad de identificar el tipo de señal que está causando la activación de la línea de IRQ, y transferir el control del programa a la rutina de servicio correspondiente.

El reloj en tiempo real tiene varios papeles a desempeñar.

Por medio de la interrupción periódica proporcionará una señal de sincronización cada segundo, que servirá para mantener la cadencia de transmisión de las marcas de tiempo que se enviarán al exterior. Se utilizará la interrupción de Alarma para incrementar internamente la

cuenta del día del año. Esto es para llevar la cuenta de los días son en el caso de que no se pueda recibir la señal de sincronización de la WWVB. Finalmente se utilizará la interrupción periódica para generar una serie de pulsos que controlarán el refrescamiento de los visualizadores.

La interrupción externa generada por el botón servirá para despertar la rutina que activa los visualizadores. Esta rutina permitirá el despliegue del tiempo durante cuatro segundos, período después del cual la rutina se desactivará.

El temporizador interno se programará para que sirva de base de tiempo para medir el ancho de los pulsos de la WWVB. El temporizador interno recibe una frecuencia efectiva de 556 Hz. Debido a sus características de diseño, el temporizador provoca una interrupción cada vez que pasa por cero y puede ser leído en cualquier instante (Anón., 1981).

La interrupción provocada por el UART actualmente sólo activa una rutina que manda un eco del código recibido. En un futuro se planea utilizar este puerto para que el sistema pueda ser programado desde una terminal exterior.

#### 4.6 Programación.

La programación del instrumento se ha dividido en las siguientes tareas:

1. Rutina de inicialización general (Reset).
2. Despachador de interrupciones.
3. Rutina de atención al reloj en tiempo real (RTR).
4. Rutina de atención a la interfaz serie (UART).
5. Rutina de atención a la WWVB.
6. Rutina de atención al exterior.
7. Rutina de atención al temporizador interno.

Las cuales se describen a continuación:

#### 4.6.1 Rutina de Inicialización general. (RESET)

Esta rutina es llamada por la aparición de la señal RESET o en el instante en que se aplica la energía al sistema. Las actividades desarrolladas por esta rutina son las siguientes:

Primeramente se hace una reinicialización del apuntador de pila (STACK POINTER); a continuación se programa el puerto A como salida y se inicializan los registros LAPSOA, TSEGUN, REGEDO, ENE, ENT, DIAJUL, DIAJUH, ERWW, ESTATU, TEMPAN y SINCRD con cero.

Se inicializa el registro de antigüedad con la antigüedad 255 así como la calidad con la peor calidad 19.

A continuación se procede a inicializar al RTR. Se programa para que funcione con la base de tiempo de 32768 Hz, en formato DCB (Decimal Cifrado en Binario) de 24 Hs.

Se programan las interrupciones de actualización (UPDATE) y de ALARMA. Se limpian las localidades del RTR y se programa la alarma para las 00 Hs 00 mins. 00 segs.

A continuación se inicializan los divisores del contador de la base de tiempo y se arranca el reloj.

Seguidamente se hace la programación del puerto B y se revisa que no exista interrupción por parte de la señal de la WWVB.

Se hace la llamada a la rutina que envía el mensaje inicial hacia la interfaz serie.

Se programan los divisores del contador del temporizador interno y el mismo temporizador interno de tal forma que se obtenga una frecuencia efectiva de conteo de 256 Hz por la entrada externa, interrumpiendo cada vez que se complete un ciclo de conteo (el par de 01 a 00).

Se hace la llamada a la subrutina de inicialización del procesador de estados. A partir de ese momento se autorizan las interrupciones y finalmente se coloca al procesador en un estado de bajo consumo de energía (WAIT).

El procesador es sacado de este estado de bajo consumo por cualquier interrupción, ya sea externa o del temporizador interno. Al regresar de atender a cualquier interrupción el programa coloca de nuevo al procesador en el estado de bajo consumo de energía (WAIT).

#### 4.4.2 Rutina principal de despacho de interrupciones. (INTERR)

El control del programa se transfiere a esta rutina cuando se activa la línea IRQ del microprocesador. La rutina INTERR se encarga de determinar la fuente que generó la señal de IRQ y transferir el control del programa a la rutina correspondiente.

Se llevan a cabo las siguientes acciones:

Primeramente pregunta por el RTR (ya que tiene la primera prioridad). Si este es el dispositivo causante de la interrupción, se transfiere el control a la rutina apropiada. (ATNRTR)

Si la fuente de interrupción no es el RTR, se inicia una interrogación a los demás dispositivos con el objeto de determinar al causante de la interrupción y efectuar la transferencia de control correspondiente.

La interrogación se realiza en el siguiente orden:

1. Si es la señal de la WWVB, se transfiere el control a la rutina ATNWW.
2. Si es la señal de Dato Recibido (DR) del UART, se transfiere el control a la rutina ATNURT.
3. Si es la señal de interrupción externa (botón), se transfiere el control a la rutina ATNINT.

Finalmente, si no se detecta ningún dispositivo causante de la interrupción se decreta una falsa alarma y se retorna de la interrupción.

#### 4.4.3 Rutina de atención a la interrupción del botón. (ATNINT)

Esta rutina se encarga de permitir de el RTA genere una interrupción periódica a razón de 32 interrupciones por segundo. Se inicializa también el contador del número de veces que se va a permitir la interrupción con un \$80 (128 decimal), de tal forma que el tiempo efectivo que estará interrumpiendo la interrupción periódica es, de  $128/32 = 4$  segundos.

#### 4.6.4 Rutina de atención al reloj en Tiempo Real. (ATNRTR)

Esta rutina es la encargada de distinguir, dentro de la interrupción del RTR el tipo de interrupción generada y transferir el control a la rutina correspondiente.

La transferencia se ejecuta bajo las siguientes prioridades:

1. Interrupción de la alarma. (ALARMA)
2. Interrupción de Actualización. (ACTUAL)
3. Interrupción periódica. (INPERI)

#### 4.6.5 Rutina de atención a la interrupción de la Alarma del RTR. (ALARMA)

Esta rutina se habilita cada vez que en el RTR se tiene la lectura de 00 hs. 00 mins. 00 segs., con el objeto de actualizar en ese momento el día del año.

Al momento de transferirse el control del programa a este punto se incrementan los registros donde se mantiene la cuenta del día del año (DIAJUL y DIAJUH). Se verifica que la cuenta no exceda de 365 y si es así, se cambia al día

000. La posibilidad de años bisiestos no está contemplada.

#### 4.6.6 Rutina de atención a la interrupción de actualización. (ACTUAL)

La rutina de ACTUAL es la encargada de atender a la petición de interrupción del RTR cada vez que ha ocurrido un ciclo de actualización en el reloj interno. Esto sucede cada segundo, y se utiliza para generar la transmisión de tiempo al exterior.

La rutina inicia preguntando por el estado de una bandera que indica si es un primer ciclo de interrupción o no. El primer ciclo se presenta cada vez que se reinicializa el RTR, y esto ocurre cuando se sincroniza con el código de la WKB o cuando se inicializa el sistema.

La necesidad de distinguir la primera interrupción del RTR de las subsecuentes se debe a una anomalía del RTR (Anón., 1983). En el momento en que se pasa del estado de reestablecimiento a estado normal de funcionamiento en los contadores de la base de tiempo del RTR, el primer ciclo, y sólo el primer ciclo de actualización ocurre después de un intervalo de 1/2 segundo de duración, ya que los ciclos siguientes ocurren cada segundo. Esto obliga a que haya

que detectar este comportamiento y repetir el primer ciclo 2 veces seguidas para completar el intervalo de un segundo que es el que nos interesa obtener.

Inmediatamente después, se transmite el Retorno de Carro (CR) y el Avance de Línea (LF) correspondientes al formato de transmisión de la hora. Esto termina el marco de transmisión que se inició en el segundo anterior.

Seguidamente se inicia la transmisión del siguiente marco. Para tal efecto se transmite primeramente un Inicio de Encabezado (Start of Header, Ctrl A en código ASCII) para avisar del inicio del marco. En seguida se verifica que se haya completado el marco anterior de información. De no ser así se considera que es un error y se termina la ejecución de la transmisión. Si se procedió normalmente (la transmisión anterior se completó con éxito) entonces se crea la nueva lista de transmisión y se inhibe su escritura. Esto es para prevenir que se sobre escriba la tabla mientras se transmite. La calidad se determina al inicio de cada minuto (en el segundo 00). Para determinar la calidad de transmisión se revisa el contador que indica el número de minutos que han transcurrido desde la última sincronización. Se revisa la bandera que indica la ocurrencia inmediata anterior de

una sincronización exitosa. Si la bandera está activa se pone en cero el contador de antigüedad y se desactiva la bandera, pero si no está activada se incrementa el registro de antigüedad. Una vez determinada la antigüedad de la hora transmitida, se calcula la calidad en base a la antigüedad, y se transmite. Una vez transmitida toda la información, se libera la tabla, y se retorna de la interrupción.

4.2.7 Rutina para determinar la calidad de la marca transmitida (CALIDA)

Esta rutina determina la calidad de la marca transmitida en base a la antigüedad de la hora mantenida. La calidad se determina de acuerdo al siguiente patrón:

Antigüedad	Valor de Q (carácter ASCII)
000	ESPACIO
001	.
004	*
128	#

D=255 ?

(Ver convención de transmisión)

#### 4.6.8 Rutina de atención a la interrupción periódica.

(INPERI)

Esta rutina es invocada cada vez que el RTR pide servicio por medio de una interrupción periódica. La rutina INPERI se encarga en primer término de habilitar a los visualizadores. Seguidamente se llama a la subrutina SACA la cual envía los dígitos de una tabla a los visualizadores. Al regresar de SACA se deshabilitan los visualizadores, y se revisa un contador que indica el número de interrupciones periódicas que se van a permitir (128 interrupciones a razón de 22 por segundo para encender los visualizadores durante cuatro segundos). Acto seguido se retorna de la interrupción. Cuando se cumple el número de interrupciones permitidas, se deshabilita la interrupción periódica.

#### 4.6.9 Subrutina de envío de dígitos a los visualizadores

(SACA)

Esta subrutina envía por una sola vez tres octetos a los visualizadores, conteniendo cada octeto dos caracteres

DCE (decimal cifrado en binario).

Primeramente se inicializan los registros correspondientes a los contadores de cifras y el contador de la posición. Los dígitos se despliegan del menos significativo al más significativo y de derecha a izquierda.

Se toma el primer octeto y se recorre para posicionar en los cuatro bits más significativos el primer dígito DCE, se llama entonces a la subrutina DIGIT para el envío del dígito.

Al regreso de la subrutina DIGIT se toma de nuevo el octeto y se envía el dígito DCE más significativo. Acto seguido se revisa el contador de octetos, si no se ha terminado se inicia de nuevo el proceso enviando la parte menos significativa del siguiente octeto. Al término del despliegue de los tres octetos se retorna de la subrutina.

#### 4.6.10 Subrutina de despliegue de un dígito. (DIGIT)

Esta subrutina toma el contenido del acumulador y lo envía a los visualizadores en la posición indicada por el registro POSIC.

Al retornar incrementa el registro de posición.

#### 4.6.11 Rutina de atención a la interrupción del UART. (ATNURT)

Esta rutina es llamada cuando el circuito UART activa la línea de IRQ por medio de la señal 'DATA RECEIVED', la cual se activa cuando el circuito receptor del UART se encuentra lleno.

La rutina lee el dato del UART y lo vuelve a transmitir, es decir hace eco de lo recibido.

En esta rutina se deja abierta la posibilidad de incluir un manejador de mandos, de tal forma que se pueda programar el instrumento desde la interfaz serie.

#### 4.6.12 Rutina de atención a la señal de la WWVB. (ATNWW)

Esta rutina es el punto de entrada para el tratamiento de la señal de la WWVB. Este tratamiento puede ser de dos tipos. El primero cuando se presenta un flanco positivo, el cual indica un principio de marca como se discutió anteriormente.

Y el segundo, que es cuando se presenta un flanco negativo.

Al momento de atender la rutina ATNWW lo primero que hace es leer el registro de la base de tiempo (TIMER).

Esto es para determinar el tiempo relativo en el que se recibió la marca y servirá para calcular la duración de las marcas de la WAVE como se verá más adelante.

Inmediatamente después determina si el flanco que ha arribado es positivo. En caso positivo se transfiere el control a la rutina que procesa los flancos positivos (SUBIDA). En caso contrario, se transfiere el control a la rutina correspondiente (BAJADA).

#### 4.6.13 Rutina de tratamiento del flanco positivo. (SUBIDA)

Primeramente se cancela la interrupción, mediante un cambio de estado en el bit 3 del puerto B.

Seguidamente se ejecuta una llamada a la subrutina ANCHOP que se encarga de determinar la duración de los pulsos. Se revisa que no se haya regresado con error de la subrutina ANCHOP. Si hay error se descarta el pulso recibido y se inicializa el procesador de estados.

En caso de no haber error, se calcula el tiempo total de duración de la marca y se verifica que cumpla con los límites establecidos (+5 por ciento).

En caso de exceder de los límites de error se transfiere el control a la rutina de error (ERRORW). Si se cumple con la duración establecida, se pregunta si en ese momento se está procesando la marca de la referencia para la

sincronización (segundo 35), y si es así se almacena el valor del registro de la base de tiempo que se tomó al inicio del tratamiento de la WWVB.

Finalmente se retorna de la interrupción.

#### 4.6.14 Rutina de tratamiento del flanco negativo. (BAJADA)

De la misma manera que en la rutina SUBIDA, se cancela la interrupción cambiando el estado del bit 3 del puerto B.

Se invoca a la subrutina ANCHOP y se revisa que no retorne con error.

De no existir error se almacena la duración del pulso alto que se acaba de recibir en la variable LAPSOA. Seguidamente se determina el tipo de señal que se recibió, clasificándola de acuerdo a su duración.

La clasificación se realiza de acuerdo a la siguiente tabla

Tipo	Duración.
Error	duración menor a .05 seg.
Cero	.05 <= duración < .35 seg.

Uno .35  $\leq$  duración  $<$  .65 seg.

Marca .65  $\leq$  duración  $<$  .95 seg.

Una vez determinado el tipo de pulso recibido, se almacena en el registro de entrada al procesador de estados(ENT). Se hace la llamada a la subrutina PROCED la cual activa el procesador de estados, quien determina si la marca (señal) actualmente recibida encaja dentro del código de la WWVE.

Regresando de la subrutina PROCED se retorna de la interrupción.

#### 4.6.15 Subrutina para calcular la duración de los pulsos (ANCHOP)

Esta subrutina se utiliza para determinar la duración en tiempo de los pulsos de la WWVB.

Primeramente se almacena en el registro TEMPAC el valor de la lectura del registro de la base de tiempo tomada en el inicio de la atención a la WWVE.

Esta lectura se compara con TEMPAN que es la lectura del registro de la base de tiempo tomada en la llamada a la rutina WWVE inmediata anterior.

Si TEMPAC  $<$  TEMPAN se procede a verificar que el

valor del registro de paso por cero (TSEGUN) sea cero, de no ser así se marca un error.

Si se cumple la condición de que TSEGUN = 0 entonces se calcula la duración de la siguiente forma:

$$\text{duración} = \text{TEMPAN} - \text{TEMPAC}.$$

Si TEMPAC >= TEMPAN se verifica que TSEGUN = 1. Si TSEGUN < 1 entonces se declara error. En caso de no existir error se calcula la duración de la siguiente manera

$$\text{duración} = \text{TEMPAN} + \$FF - \text{TEMPAC}.$$

Una vez determinada la duración, se almacena en la variable LAPSO, se limpia el registro TSEGUN, se actualiza el valor de TEMPAN y se retorna de la llamada a la subrutina

4.6.16 Rutina de atención a la interrupción del temporizador interno. (TIMER1)

Esta rutina se activa por medio del temporizador interno. La activación se realiza cuando el registro de conteo del temporizador pasa de 01 a 00. En primer término se efectúa la cancelación de la

interrupción, esto se hace para evitar que la misma interrupción sea atendida más de una vez.

Después se incrementa el registro TSEGUN que es donde se almacena el número de veces seguidas que ocurre la transición de 01 a 00.

A continuación se revisa si la bandera del contador de sincronización (TIMOS) está prendida. En caso de que sí lo esté, se decrementa dicho contador. Después se pregunta por el estado del contador de sincronización y se toman las siguientes decisiones:

1. Si el contador es mayor que cero, se retorna de la interrupción.
2. Si el contador es igual a cero, se transfiere el control a la rutina PRESIN, la cual prepara la sincronización.
3. Si el contador es menor que cero, se transfiere el control del programa a la rutina CRONOS, la cual se encarga de efectuar la sincronización.

#### 4.6.17 Rutina de preparación a la sincronización. (PRESIN)

En esta rutina se para el temporizador interno para

cargarlo con el valor calculado de defasamiento entre la señal recibida (el valor del registro del contador del temporizador en el segundo de sincronización) y la señal de sincronización propia (Ciclo de Actualización del RTR). Se para también el RTR para que no envíe ninguna interrupción durante el período de sincronización y se vuelve a arrancar el temporizador interno.

En este momento se inicia el proceso de sincronización el cual termina con la siguiente interrupción del Temporizador interno.

#### 4.6.18 Rutina de sincronización de la base de Tiempo. (CRONDS)

La rutina CRONDS es la responsable de sincronizar el reloj interno (RTR) con la señal de divulgación de tiempo (WWVB).

La primera acción a efectuar por la rutina es llamar a la subrutina LIMPIA que se encarga de inicializar los contadores del RTR en ceros y programarlo de nuevo para sincronizarlo con la señal de la WWVB.

Seguidamente se inicializa el temporizador interno y se programa para que trabaje en el modo normal. Recordemos que se inicializó de manera diferente al prepararlo para la sincronización.

Seguidamente se transmite un Retorno de Carro (CR) y un Avance de Línea (LF) para indicar la sincronización, ya que en este momento se encuentra parado el RTR y es tiempo de efectuar una sincronización (ciclo de actualización). Se transfiere la fecha recibida al reloj y se inicializan los segundos. Se hace una llamada a la subrutina INIPRO para inicializar el procesador de estados y, finalmente se avisa a la rutina CALIDA que se llevó a cabo una sincronización exitosa. Seguidamente se transfiere el control del programa a la rutina de ACTUAL entrando por TRHORA para inicializar la transmisión de la nueva marca de tiempo.

El procesador de estados.

El caso del reconocimiento de secuencias es una aplicación clásica de la técnica de diagramas de estado. En esta aplicación, se ha utilizado el lenguaje de estados para reconocer la sucesión de las marcas de la señal de la WWVB. Para llevar a cabo esta acción se han elegido las siguientes condiciones:

1. UNO. Esta condición es verdadera cuando la entrada es un pulso codificado como uno.
2. CERO. Esta condición es verdadera cuando la

entrada es un pulso codificado como cero.

3. MARCA. Esta condición es verdadera cuando la entrada es un pulso codificado como marca.

4. ENEO. Esta condición es verdadera cuando el contador de repetición de estados (ENE) es cero.

5. SIGA. Esta condición siempre es verdadera.

Tomando en cuenta estas condiciones se ha generado el diagrama de estados que se presenta en la figura 14.

El diagrama de estados se ha traducido al lenguaje de estados y por medio de las macros definidas en el capítulo anterior se produjo la tabla de estados (Ver apéndice V) necesaria para su procesamiento. De esta tabla de estados se tienen las siguientes acciones a ejecutar por el procesador de estados.

#### 4.7.1 Rutinas ejecutables por el procesador de estados.

DIAGRAMA WWVB

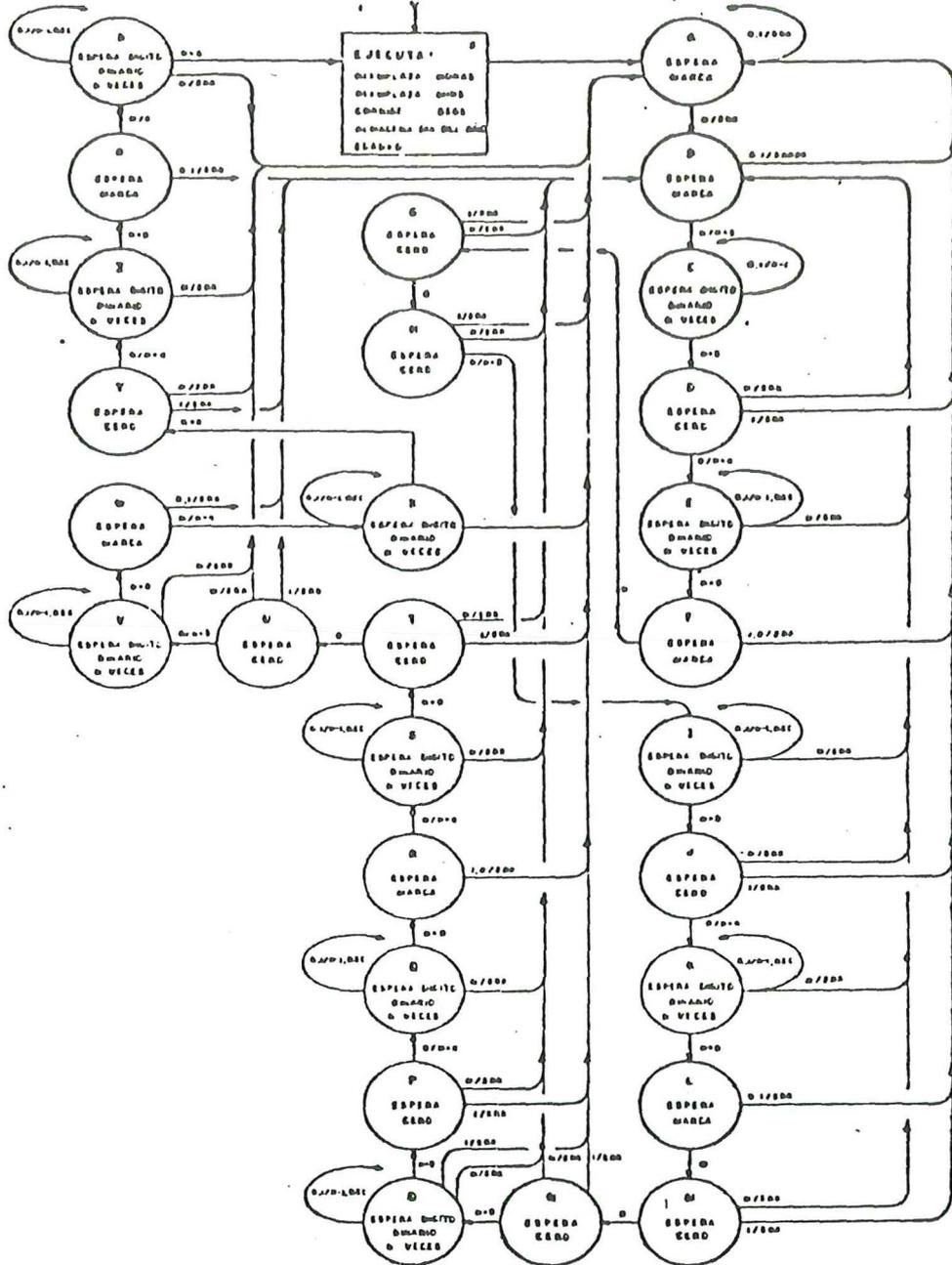


Figura 14. Diagrama de estados de la WWVB.

#### 4.7.1.1 Subrutina NADA.

Esta subrutina como su nombre lo indica no hace nada. La existencia de esta subrutina (que sólo consta de un retorno de subrutina) se debe a la estructura del procesador de estados.

#### 4.7.1.2 Subrutina CARGAN

Esta subrutina carga el valor de N del estado en la variable ENE. Esta variable sirve de contador para un estado.

#### 4.7.1.3 Subrutina NERROR.

Esta subrutina limpia las banderas de error.

#### 4.7.1.4 Subrutina ERROR.

Esta subrutina enciende los bits de error y limpia las localidades de recepción de datos. Además inicializa las banderas que indican la marca de sincronización y la habilitación del contador de sincronización e inicializa el contador SEG35.

#### 4.7.1.5 Subrutina CARDIG.

Esta subrutina es la encargada de recibir los bits de información decodificados de la WWVB e irlos almacenando en una tabla de recepción para después poder efectuar la sincronización.

#### 4.7.1.6 Subrutina DECENE.

Esta subrutina decrementa el valor de ENE.

#### 4.7.1.7 Subrutina SEG35.

Esta subrutina activa las banderas que indican que la siguiente marca es la de sincronización y que se habilite el contador a sincronización.

#### 4.7.1.8 Subrutina PREPAR.

Esta subrutina tiene por objeto preparar la cantidad de milésimas de segundo de corrección a UT1 decodificadas de la WWVB. Esto es debido a que en el sistema se utiliza una base de tiempo de 256 Hz y es necesario efectuar la conversión de milésimas de segundo a 256avos de segundo, además se efectúa una conversión de DCB, que es el código en que viene la

compensación, a binario puro que es lo que utiliza el contador de la sincronización.

En la subrutina PREPAR se aprovechan algunas de las propiedades de la conversión; ésta funciona de la siguiente forma:

Se toman directamente las milésimas como enteros y se empiezan a convertir a binario mediante la multiplicación por dos, se acumulan los residuos y se obtiene el resultado en binario. Esta operación se repite tantas veces como número de bits de resolución se necesiten en el resultado (8 en nuestro caso), a razón de un bit por vuelta.

Se hace uso de la subrutina DUPLIC para efectuar la multiplicación.

#### 4.7.1.9 Subrutina DUPLIC.

Esta subrutina se utiliza para multiplicar un número decimal en DCE por dos. Los números deben de estar uno en cada octeto y pueden ser n cifras. Al final se obtiene el número duplicado y un acarreo si es que ocurrió.

## 5 CONCLUSIONES.

El presente trabajo ha descrito el desarrollo de un controlador digital de propósitos generales para ser aplicado al diseño de instrumentos de bajo consumo de potencia.

El sistema se diseñó con la arquitectura de una microcomputadora, lo cual obliga a que gran parte del esfuerzo dedicado a adaptarlo a una aplicación dada recaiga sobre la programación. Es por eso que también se desarrolló un sistema de programación basado en el concepto de lenguajes de estado para dotar al usuario del sistema con una herramienta de programación más ad hoc al control de procesos.

Para demostrar la factibilidad de la utilización del controlador en situaciones reales, se desarrolló una aplicación que involucró en su diseño el uso de las herramientas de circuitería y programación que fueron objeto de este trabajo.

El resultado de esta aplicación es una base de tiempo con calidad patrón que se corrige continuamente al recibir las señales de la estación transmisora del tiempo universal WWVB.

Este instrumento particular, una vez que se fabrique como

prototipo reproducible a nivel industrial promete difundirse entre otros usuarios, ya que:

Su empleo no se limita a la sismología o a la geofísica, debido a que existe un gran variedad de aplicaciones en las que es vital disponer de señales de tiempo con calidad patrón, como lo son en la industria eléctrica, la radio y la televisión, la navegación y las comunicaciones.

El costo estimado de un prototipo reproducible a nivel industrial es de \$500 Dlls, mientras que un dispositivo comercial con características similares como el Reloj Sincronizado con la WWVB Modelo B170, fabricado por la compañía Spectracom Corp. , se cotiza actualmente en 2 100 Dlls (Anón., 1982c).

La inclusión del procesador de estados en la CUTCMOE probó ser de gran utilidad en el desarrollo de la base de tiempo, ya que simplificó la programación del reconocimiento de la señal transmitida por la WWVB. Esto se debió a que la traducción del algoritmo de reconocimiento a un diagrama de estados y posteriormente a una tabla de estados, directamente ejecutable por el procesador, resultó una operación natural dada la característica secuencial del código de la WWVB. El proceso anterior resulta más tedioso si se describe el

proceso de reconocimiento del código mediante otro método, como puede ser el de diagramas de flujo, y a partir de ello se escribe un programa en lenguaje ensamblador que lo ejecuta (Mitrani, E. 1983).

Actualmente, la CUTCMOS se está empleando en otras aplicaciones. Entre ellas se puede mencionar un sistema de control de un reflectómetro automático, que se está desarrollando (Salazar M., David, 1984) en el Laboratorio de Películas Delgadas del Departamento de Óptica de la División de Física Aplicada del CICESE.

Como en todos los casos, cualquier diseño se puede mejorar. El caso de la CUTCMOS no es una excepción y a lo largo del tiempo que se lleva utilizando se han propuesto varias mejoras.

- 1.- La inclusión de una interfaz serie para que la CUTCMOS tenga capacidad de comunicarse con otros dispositivos que la puedan programar directamente en el campo.

- 2.- La adición de resistencias sensoras de corriente en las líneas de alimentación, con objeto de medir directamente el consumo de potencia.

En forma personal, el autor cree que el desarrollo de este tema de tesis le ha permitido:

1. Obtener habilidad y experiencia en el manejo de circuitos de tecnología HCMOS.
2. El familiarizarse con un sistema de desarrollo de microprocesadores completamente nuevo, como lo es el sistema empleado por el MC146805E2 de Motorola.
3. El desarrollar una serie de criterios que permitan el diseño de subsecuentes módulos o el mejoramiento de los ya existentes.

Finalmente, pero sin restarle importancia, se espera haber dejado un precedente que sirva de inicio en la creación de una infraestructura para el desarrollo de instrumentos basados en microprocesadores de baja potencia en el Laboratorio de Instrumentación Geofísica de la División de Ciencias de la Tierra del CICESE. La CUTCMOS es pues, el primer dispositivo de una serie de tarjetas que están siendo diseñadas en este Laboratorio con la intención de disponer de un catálogo de tarjetas compatibles entre sí que permitan a los usuarios el desarrollar instrumentos con relativa facilidad. Esto evitará a los diseñadores el tener que

desviar su tiempo y esfuerzo en campos en los que probablemente no se encuentren suficientemente familiarizados.

## LITERATURA CITADA

- Morgan, A. H. 1967. Distribution of Standard Frequency and Time signals. Proc. of the IEEE 55(6):827-836.
- Haymes, Rogert C. 1971. Introduction to space science. John Wiley and Sons, Inc., New York. 556 pp.
- Chi, A. R., Fosque, H. S. 1972. A step in time. IEEE spectrum. 9(1):82-86
- Smith, Humphry M. 1972. International time and frequency coordination Proc of the IEEE. 60(5):479-486
- Howe, Sandra L. 1976. NBS time and frequency dissemination services. NBS special publication no. 432. U. S. Department of Commerce. Washington, D. C. 16 pp.
- Anón. 1978. Model 46B-DC User's manual. Kinematics. True time division. Santa Rosa, Cal. 42 pp.
- Anón. 1981. CMOS versions of the MC6805 development systems

- user's guide. Motorola Inc. Austin, Texas. 56 pp.
- Gander, J. G., Liechti, H. U. 1981. State language for real-time process control. Microprocessors and microsystems. 5(1):27-28
- Anón. 1982a. NM27C16 UV Erasable CMOS PROM Preliminary data. National semiconductor. Santa Clara, Cal. 6 pp.
- Anón. 1982b. Hitachi IC memories HM6116LP Data. Hitachi America Ltd. San Jose, Cal. 5 pp.
- Anón. 1982c. Short form catalog. Spectracom Corp. Rochester, N. Y. 1 pp.
- McEride, David. 1982. Low power considerations for MC146818 real time clock applications. Motorola Inc. Austin, Texas. 2 pp.
- Anón. 1983. C-44 bus data book, Onset Computer Corp., N. Falmouth Mass. 35 pp.
- Frank, Robert L. 1982. Current developments in Loran-C. Proc. of the IEEE. 71(10):1127-1139
- Mitrani, A. Enrique. 1983. Comunicación personal.

Salazar M., David. 1984. Reflectometría aplicada al estudio de capas delgadas. Escuela Superior de Ciencias Biológicas. Ensenada, B. C. Manuscrito.

APENDICE I.

CUTCMDS

Inventario de componentes

Circuitos integrados.

Núm.	Cant.	Num. parte	Fabricante	Descripción
U1	1	MC146805E2	Motorola	Microprocesador 8 bits
U2	1	MC146818	Motorola	Reloj de tiempo real
U3	1	MC74HC138	Motorola	Decodificador 3 a 8
U4	1	MM74HC139N	National	Decodificador 2 a 4
U5	1	MM74HC004N	National	Inversor
U6	1	MC74HC74	Motorola	Báscula 'D'
U7	1	MC74HC00	Motorola	Compuerta NAND 2 ent.
U8	1	MC74HC373	Motorola	Candado octal de 3 est
U9	1	NMC27C16QE-45	National	Mem. EPROM de 2k por 8
U10	1	HM2116LF-3	Hitachi	Memoria RAM de 2K por 8
U11, U12	2	MC74HC243	Motorola	Manejador de ducto
U13	1	SPI74HC541N	SPI	Reforzador octal

Bases para circuito integrado (DIP)

xx	1	40 patas
xx	3	24 patas

xx	2	20 patas
xx	2	16 patas
xx	5	14 patas

#### Cristales piezoeléctricos

XTAL1	1	5 MHz
XTAL2	1	32768 Hz

#### Capacitores

C1	1	1 uf
C2	1	22 pf
C3	1	15 pf
C4	1	33 pf
C5	1	5 pf
C6	1	0047 uf
C7-C8	2	2-25 pf
C9-C14	6	0.1 uf

#### Resistores

R1	1	56 K
R2	1	10 K
R3	1	10 M
R4	1	22 M

R5            1   220 K

R6            1   1 K

Varios

INi           1   Interruptor UPUT momentáneo, normalmente abierto

CNi           1   Conector de 26 patas (Ansley) para tarjeta de  
                  circuito impreso.

Tarjeta de circuito impreso de dos caras, con metalización  
(‘thru hole’) en los agujeros, de 11.4 cm. X 13.9. cm.

A continuación se presenta el diagrama de localización de  
partes y el diagrama electrónico detallado.

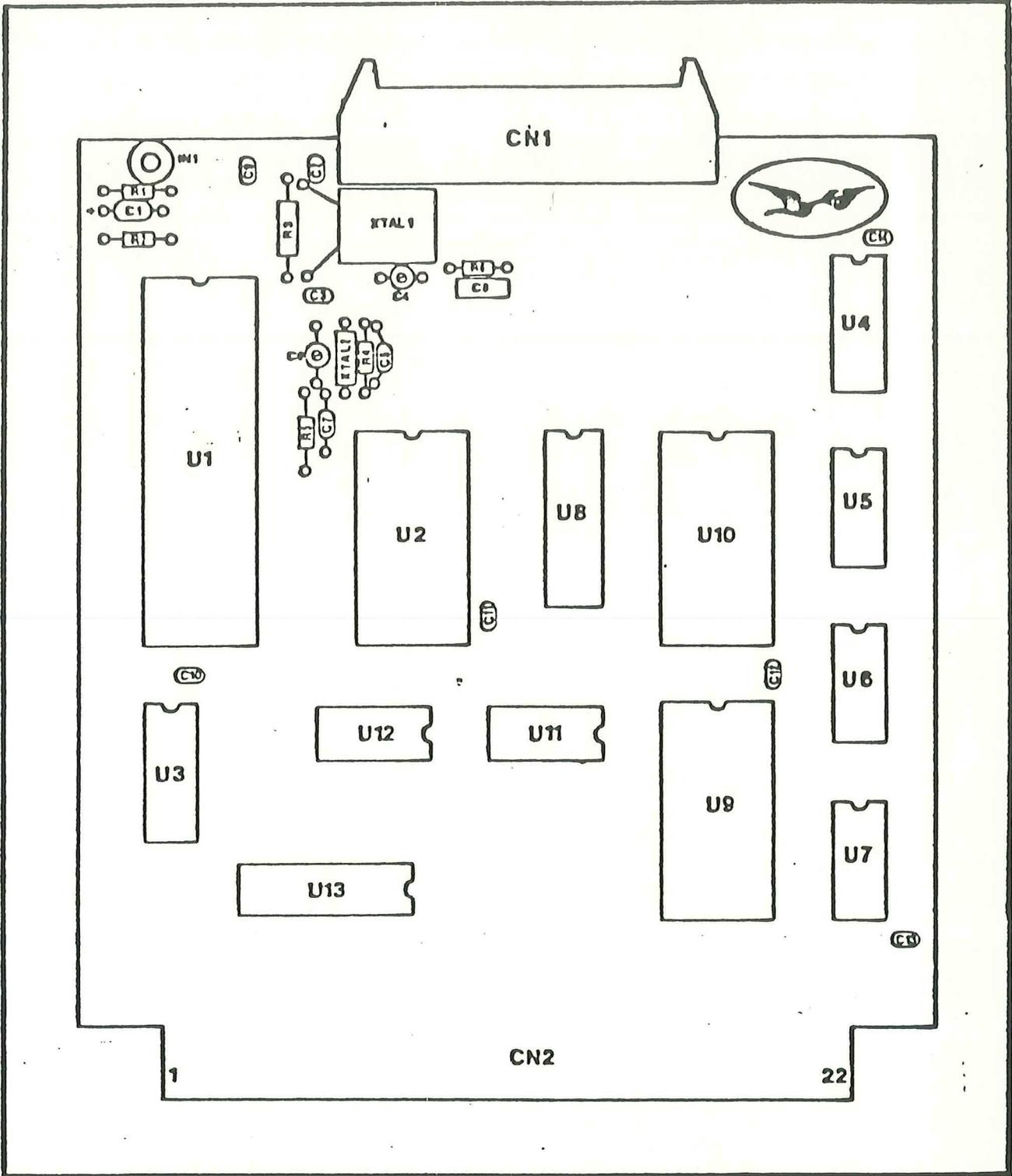


Figura 15. Diagrama de localización de partes de la CUTCMOS.

## Tarjeta Consola.

## Inventario de componentes.

## Circuitos Integrados.

Núm.	Cant.	Núm. parte	Fabricante	Descripción
U1	1	IM6402IFL	Intersil	UART
U2	1	MC74HC138	Motorola	Decodificador 3 a 8
U3	1	CD4511A	RCA	Manejador 7 seg.
U4	1	CD4013A	RCA	Báscula 'D'
U5	1	CD4098B	RCA	Monoestable doble.
U6	1	MC14077A	Motorola	Compuerta 'DEX' neg.
U7	1	CD4016A	RCA	Interruptor analógico
U8	1	CD4052B	RCA	Multicalalizador 4 a 1
U9	1	CD4060A	RCA	Oscilador con divisores

## Bases para circuito integrado (DIP)

xx	1	40 patas
xx	1	24 patas
xx	15	16 patas

## Cristales piezoeléctricos

XTAL1 1 2.4576 MHz

## Capacitores

C1, C2 2 1 uf

C3 1 0.1 uf

C4 1 0.001 uf

C5, C6 0.047 uf

## Resistores

R1 1 120 K

R2 1 470 k

R3, R4 10 M

R5 1 2.2 K

R6, R7 2 10 K

R8, R9 2 1 K

R9-R15 7 220

R16 1 3.7 K

R17        1    470 K

Varios

T1        1    transistor PNP 2N2401  
V1-V2     2    Visualizadores multicanalizados  
DC1       1    Convertidor DC-DC de 5V a -9V  
CN1       1    Conector de 26 patas (Ansley) para tarjeta  
          de circuito impreso.  
CN2       1    Conector BNC hembra  
CN3       1    Conector DE-25 hembra

A continuación se presenta el diagrama electrónico de la tarjeta de Consola.

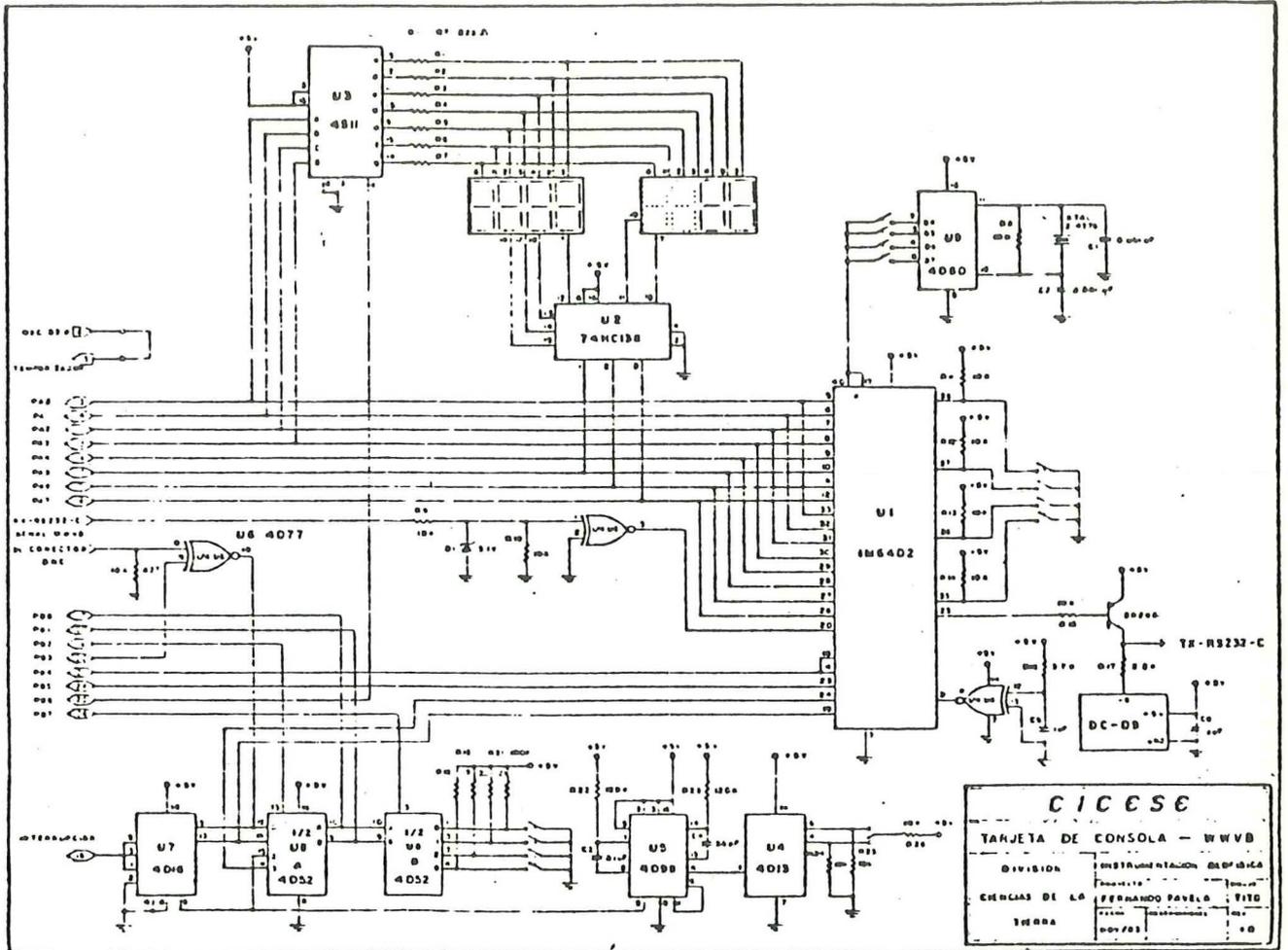


Figura 17. Diagrama electrónico de la tarjeta de Consola.

## Conexiones al ducto principal C-44

No.	Señal	Tipo	No.	Señal	Tipo
1	COMUN	Pot.	A	COMUN	Pot.
2	NC		B	NC	
3	NC		C	NC	
4	NC		D	NC	
5	NC		E	NC	
6	NC		F	INTERR. EXT. (B)	Ent.
7	NC		H	A13 (A)	Sal.
8	NC		J	A12 (A)	Sal.
9	NC		K	A11 (A)	Sal.
10	RELOJ EXT. (A)	Ent.	L	A10 (A)	Sal.
11	NC		M	A9 (A)	Sal.
12	NC		N	AB (A)	Sal.
13	NC		P	B7 (A)	Ent. /Sal.
14	NC		R	B6 (A)	Ent. /Sal.
15	R (A) / W (B)	Sal.	S	B5 (A)	Ent. /Sal.
16	DS (A)	Sal.	T	B4 (A)	Ent. /Sal.
17	AS (A)	Sal.	U	B3 (A)	Ent. /Sal.
18	NC		V	B2 (A)	Ent. /Sal.
19	A15 (A)	Sal.	W	B1 (A)	Ent. /Sal.
20	A14 (A)	Sal.	X	B0 (A)	Ent. /Sal.
21	COMUN	Pot.	Y	COMUN	Pot.
22	VCC	Pot.	Z	VCC	Pot.

## Conexiones al ducto auxiliar

No. de pata	Señal	Tipo
1	GNDA CUAD. (A)	Sal.
2	COMUN	Pot.
3	TEMPORIZADOR (A)	Ent.
4	DETECTA POTENC. (A)	Ent.
5	VCC	Pot.
6	OSC. 32K	Sal.
7	PA6	Ent. /Sal.
8	PA7	Ent. /Sal.
9	PA4	Ent. /Sal.
10	PA5	Ent. /Sal.
11	PA2	Ent. /Sal.
12	PA3	Ent. /Sal.
13	PA0	Ent. /Sal.
14	PA1	Ent. /Sal.
15	NC	
16	INTERRUPCION (B)	Ent.
17	PB0	Ent. /Sal.
18	COMUN	Pot.
19	PB2	Ent. /Sal.
20	PB1	Ent. /Sal.
21	PB4	Ent. /Sal.

22	PB3	Ent. /Sal.
23	PB6	Ent. /Sal.
24	PB5	Ent. /Sal.
25	COMUN	Pot.
26	PB7	Ent. /Sal.

## APENDICE II.

### Especificaciones del ducto C-44

#### Prefacio

El ducto C-44 fue introducido por la compañía 'Onset Computer Corp' para satisfacer los requerimientos de sistemas alimentados por baterías. El diseño del ducto se escogió para asegurar compatibilidad con un gran número de microprocesadores y dispositivos de soporte CMOS.

#### Objetivos de diseño

Las reglas de diseño del ducto C-44 imponen demandas adicionales en el diseñador de la tarjeta, lo cual resulta en simplificación en el diseño de sistemas y en el mantenimiento por parte del usuario.

#### Consumo de energía mínimo.

Ya que la operación por baterías es un objetivo principal, las tarjetas que se conecten al ducto C-44, deben diseñarse para utilizar un mínimo de energía. Debe tenerse cuidado en utilizar partes y técnicas de diseño

que minimicen el consumo de energía. Las tarjetas compatibles con el ducto C-44 deben tener algún medio para permitir que se desconecten aquellas funciones que no se requieran en un momento dado.

Amplia compatibilidad.

Aunque la estructura del ducto C-44 se parece mucho a la del microprocesador NEC-800, las especificaciones del ducto se escogieron para permitir que los microprocesadores 146805E2, 80C85, 80C51 y 6303 se pudieran adaptar fácilmente a este ducto. Este diseño flexible del ducto se adapta a cualquier periférico o microprocesador que use un ducto en donde los datos y las direcciones estén multicanalizados.

En el ducto C-44, no se dejaron posibilidades de hacer multiprocesamiento o DMA, ya que esto hubiera restringido el número de procesadores que podrían ser adaptados al ducto, y esta necesidad puede satisfacerse utilizando procesadores esclavos y/o memoria de doble acceso.

Flexibilidad en el tamaño del sistema.

Requerimientos simples en el voltaje de alimentación, y la obligación de tener reforzadores hace que los

sistemas que trabajen en este ducto se puedan expandir desde sistemas de una sola tarjeta, hasta sistemas de múltiples tarjetas con facilidad. Las tarjetas deben ser capaces de operar a partir de una batería que proporcione un voltaje entre 8 y 18 V y un voltaje regulado proporcionado por la tarjeta de la UCP.

Consideraciones generales.

Tipos de tarjetas.

El ducto C-44 actualmente especifica sólo dos tipos de tarjetas: Tarjetas de UCP y periféricos.

Tarjetas de UCP

Las tarjetas de UCP llevan a cabo la doble función de controlar las líneas de datos y direcciones a través de su UCF y de proporcionar una cantidad limitada de voltaje regulado al resto del sistema. Las tarjetas de UCP deben ser capaces de operar a partir de una fuente no regulada en el rango de 8 a 18 V, y deben ser capaces de proporcionar al menos 75mA de corriente al voltaje regulado  $V_{reg}$ .

### Tarjetas periféricas

Las tarjetas periféricas responden a los mandos de la tarjeta de la UCP. Pueden utilizar hasta 50 mA de corriente (promediada en un ciclo del ducto) a través de la línea de voltaje regulado cuando son accedadas, o hasta 10 mA con el CPU inactivo. Si se requiere más corriente o diferentes voltajes de alimentación, se deben obtener de la conexión para batería de cada tarjeta periférica. Las tarjetas de periféricos, deben ser capaces de operar apropiadamente utilizando unicamente el voltaje  $V_{reg}$  y la entrada de batería en cualquier voltaje dentro del rango 8-18 V.

### Estados activos e inactivos.

Las tarjetas de UCP del ducto C-44 tienen la opción de reducir el voltaje del sistema durante los periodos en que se encuentren inactivas. Tales periodos pueden ocurrir mientras se mantiene un reloj en tiempo real, o mientras se espera por una interrupción externa. Para ayudar a reducir el consumo de corriente durante estos periodos, la UCP puede suspender toda la actividad en el ducto y reducir el voltaje regulado  $V_{reg}$  a 3.0 V. Las tarjetas de periféricos, deben ser diseñadas para permitir operación estática en este voltaje. El consumo de corriente en este estado debe ser

menor de 30 microamperes por tarjeta de periférico, a menos que la tarjeta contenga algún dispositivo que tenga que permanecer activo, tal como un reloj de tiempo real.

Las tarjetas de UCP pueden opcionalmente entrar en el estado inactivo sin reducir la salida de Vreg. Esto es útil cuando alguna tarjeta periférica esta llevando a cabo alguna función que requiera de alimentación de 5.0 V, mientras que la UCP se mantiene inactiva para minimizar el consumo de energía.

Asignación de patas al ducto.

La designación de las patas por números se refiere al lado de componentes de la tarjeta. La lógica del ducto C-44 es generalmente positiva para las líneas de datos y direcciones y negativa para las líneas de control. La lógica negativa se designa con el sufijo B ('bajo') en el nombre de la señal. N. C. indica que no se han asignado estas líneas. Las señales entre paréntesis son opcionales y no necesitan ser proporcionadas por la UCP.

- |                        |                               |
|------------------------|-------------------------------|
| (1) COMUN              | (A) COMUN                     |
| (2) BATERIA A. C.      | (B) BATERIA A. C.             |
| (3) INTERRUPCION D (B) | (C) RESTABLECIMIENTO EXT. (B) |
| (4) N. C.              | (D) SALIDA RESTABL. EXT (B)   |

(5) INTERRUPCION C (B)	(E) INTERRUPCION B (B)
(6) DESPIERTA (B)	(F) INTERRUPCION A (B)
(7) N. C.	(H) A13
(8) N. C.	(J) A12
(9) (II (B))	(K) A11
(10) RELOJ	(L) A10
(11) N. C.	(M) A9
(12) BATERIA	(N) AB
(13) N. C.	(P) AD7
(14) N. C.	(R) AD6
(15) LEE (B)	(S) AD5
(16) ESCRIBE (B)	(T) AD4
(17) ALE	(U) AD3
(18) ID (A) /M (B)	(V) AD2
(19) A15	(W) AD1
(20) A14	(X) ADO
(21) COMUN	(Y) COMUN
(22) Vreg.	(Z) Vreg.

BATERIA A. C. se utiliza en las tarjetas periféricas que consumen mucha corriente en tiempos cortos y/o alimentan sensores de alta corriente.

Nótese que no se permite que la línea de reloj esté activa mientras la UCP está en su estado inactivo.

## LINEAS DE DIRECCIONES ALTAS (A8 -A15)

Las líneas A8-A15 llevan los 8 bits más significativos de las direcciones durante los accesos a memoria, y las direcciones de los puertos durante los accesos de entrada/salida.

## LINEAS DE DATOS/DIRECCIONES (A0-A7)

Las líneas de datos/direcciones están multicanalizadas en el tiempo y llevan las direcciones de memoria A0-A7 durante la primera parte del ciclo de acceso a la memoria y los datos durante la segunda parte. Durante los accesos a los puertos, estas líneas llevan los mismos datos que las líneas de dirección altas durante el primer ciclo, y datos durante el segundo.

## ALE

El flanco de bajada de esta señal puede utilizarse para atrapar la dirección de las líneas datos/direcciones.

## IO (A) / M (B)

Esta línea se pone en alto cuando las líneas de dirección están accésando un puerto, y en bajo cuando están accésando

memoria.

#### LEE (B)

Cuando la línea LEE (B) se pone en bajo, la UCP está esperando que un periférico le mande datos.

#### ESCRIBE (B)

Cuando la línea ESCRIBE está alta, la UCP está mandando datos a un periférico.

#### RESTABLECIMIENTO EXTERNO (B)

Esta es la línea de entrada al UCP que se utiliza para inicializar todo el sistema.

#### RESTABLECIMIENTO A PERIFERICOS (B)

Esta línea es una salida generada por la UCP y se usa para inicializar a las tarjetas de periféricos.

#### RELOJ

Esta línea proporciona una frecuencia en el alcance de 0.0 a 3 MHz para sincronizar al sistema.

## APENDICE III.

## Arquitectura de procesador MC146805E2

El procesador MC146805E2 contiene cinco registros (Anón., 1983):

## El acumulador (A)

El registro (A) es un acumulador de propósito general que se utiliza para realizar cálculos aritméticos y manipulación de datos.

## El registro de índice (X).

El registro X se utiliza durante el modo de direccionamiento indirecto y también puede utilizarse como un acumulador auxiliar. En la instrucción indicada, el registro X almacena un valor de E bits que se suma a un valor opcional contenido en la instrucción para crear una dirección efectiva.

## El contador de programa (PC).

El contador de programa es utilizado por el procesador para apuntar a la siguiente instrucción a ser ejecutada.

El apuntador de pila (S).

El apuntador de pila registra la dirección de la siguiente localidad a escribir en la pila.

La pila se utiliza para almacenar las direcciones de regreso de la subrutinas y para almacenar automáticamente los contenidos de los registros durante interrupciones.

El registro de códigos de condición (CC)

El registro de códigos de condición contiene varios bits que reflejan el estado actual del procesador.

Los bits del CC son:

- a) Medio acarreo (H).
- b) Máscara de interrupción (I).
- c) Bit negativo (N).
- d) Bit cero (Z).
- e) Bit de acarreo (C).

El acumulador (A) y el registro de índice (X) son de 8 bits, el registro de códigos de condición (CC) contiene 5 bits, el contador de programa (PC) es de 13 bits y el apuntador de pila (S) es de 6 bits.

Modos de direccionamiento.

Los modos de direccionamiento disponibles en el MC146805E2 son los siguientes:

Modo de direccionamiento inmediato.

La dirección efectiva de una instrucción en modo inmediato) es la localidad siguiente al código de operación. Este modo se utiliza para tomar un valor conocido en el momento en que el programa se escribe y no es cambiado durante la ejecución.

Estas instrucciones son de dos octetos; una para el código de operación y otro para el dato inmediato.

$PC + 1 \rightarrow PC$

$EA = PC$

$PC + 1 \rightarrow PC$

Modo de direccionamiento directo.

La dirección efectiva en el modo directo (DIR), es el contenido del siguiente octeto del código de operación. El direccionamiento directo puede ser usado para referirse a cualquiera de las primeras 256 (\$00-\$FF) localidades de memoria.

PC + 1 → PC

EA = (PC) + \$0000

PC + 1 → PC

Modo de direccionamiento extendido.

La dirección efectiva en una instrucción de modo extendido es el contenido de los 2 octetos siguientes al código de operación:

PC + 1 → PC

EA = (PC) : (PC + 1)

PC + 2 → PC

Modo de direccionamiento indicado.

El Modo de direccionamiento indicado en el registro "X" (registro de índice) para el cálculo de la dirección efectiva. En el MC146805E2 existen varias modalidades de direccionamiento indicado:

Indicado sin desplazamiento

En este modo, el contenido del registro de índice es la dirección efectiva. El modo indicado sin desplazamiento, requiere de solamente un octeto para codificar una

instrucción.

$$EA = X + \$0000$$

$$PC + 1 \rightarrow PC$$

Indicado con desplazamiento de 8 bits

En este modo, la dirección efectiva se calcula sumando el contenido del octeto siguiente al código de operación, al contenido del registro de índice X.

Esta es una instrucción de 2 octetos, el octeto de código de operación y el octeto de desplazamiento.

$$PC + 1 \rightarrow PC$$

$$EA = (PC) + X + \$0000$$

$$PC + 1 \rightarrow PC$$

Indicado con desplazamiento de 16 bits

La dirección efectiva en este modo de direccionamiento se calcula sumando los contenidos de los 2 octetos siguientes al código de operación al contenido del registro de índice X.

Esta es una instrucción de 3 octetos; uno para el código de operación y dos para el valor del desplazamiento.

$$PC + 1 \rightarrow PC$$

$$EA = (PC) : (PC + 1) + X$$

$$PC + 2 \rightarrow PC$$

Modo de direccionamiento relativo.

En el direccionamiento relativo, el octeto que sigue al código de operación se interpreta como un número de 8 bits codificado en complemento a dos. La dirección efectiva se calcula sumando los contenidos del contador del programa (PC) a éste número. Una instrucción de direccionamiento relativo ocupa 2 octetos: el de código de operación y el desplazamiento relativo.

$$(PC) + 1 \rightarrow PC$$

$$EA = PC + TEMP$$

Modo de direccionamiento de control de bits individuales

Mediante este modo de direccionamiento se pueden escribir bits individuales de cualquier localidad de memoria en página cero. Las instrucciones que utilizan este modo ocupan 2 octetos, uno para el código de operación (incluyendo el número de bit) y el otro para direccionar el octeto que contiene el bit de interés.

```

PC + 1 -> PC
EA = (PC) + $0000
PC + 1 -> PC

```

Modo direccionamiento de prueba de bits individuales.

Este modo es una combinación de direccionamiento directo, relativo y de bit. El octeto de dato ha ser probado se localiza mediante un apuntador que reside en la dirección siguiente al código de operación. Las instrucciones que utilizan este modo, ocupan 3 octetos.

```

PC + 1 -> PC
EA = (PC) + $0000
PC + 1 -> PC
(PC) -> TEMP
EA2= PC + TEMP

```

Modo de direccionamiento inherente.

En el modo de direccionamiento inherente, toda la información necesaria para llevar a cabo la instrucción, está contenida en el código de operación y por lo tanto no se requiere de campos adicionales para especificar operandos.

Las instrucciones del MC146805E2.

Instrucciones de registro/memoria.

Los modos de direccionamiento que son aplicables a las instrucciones de registro/memoria son:

Inmediato, directo, extendido, indicado sin desplazamiento, indicado con desplazamiento de 1 octeto e indicado con desplazamiento de 2 octetos.

El direccionamiento inmediato no se utiliza en las instrucciones de almacenar y saltar (STA, JMP, JSR y STX).

ADC - Suma memoria al acumulador y acarreo.

ADD - Suma memoria al acumulador.

AND - 'Y' lógico de memoria con el acumulador.

BIT - Prueba el bit de memoria con el acumulador (comparación lógica).

CMF - Compara acumulador con memoria (comparación aritmética).

CPX - Compara registro de índice X con memoria (Comparación aritmética).

EOR - 'O' exclusiva de memoria con el acumulador.

JMP - Salta.

JSR - Salta a subrutina.

LDA - Carga el acumulador.

LDX - Carga el registro X con contenido de memoria.

ORA - 'O' lógico de memoria con el acumulador.

SEC - Subtrae de memoria y toma en cuenta el acarreo.

STA - Almacena acumulador en memoria.

STX - Almacena registro de índice X en memoria.

SUB - Subtrae memoria del acumulador (ignora el estado del acarreo.)

Instrucciones lee/modifica/escrbe.

Estas instrucciones leen una dirección de memoria o registro, o prueban sus contenidos y después escriben un nuevo valor en la memoria o registro.

Los modos de direccionamiento disponibles para estas instrucciones son: directo, inherente, indicado sin desplazamiento e indicado con desplazamiento de un octeto.

ASL - recorrimiento aritmético a la izquierda (igual que LSL).

ASR - Recorrimiento aritmético a la derecha.

CLR - Pone en ceros.

COM - Complementa el contenido de la dirección efectiva.

DEC - Decrementa.

INC - Incrementa.

LSL - Recorrimiento lógico a la izquierda (igual que ASL)

LSR - Recorrimiento lógico a la derecha.

NEG - Negación (complemento a 2's).

RDL - Rotar a la izquierda con acarreo.

RDR - Rotar hacia la derecha con acarreo.

TST - Prueba si es negativo o cero.

#### Instrucciones de brinco condicionados

En estas instrucciones se prueba el estado de uno o varios de los bits del código de condición (CC) o el estado de una línea de control externa. Si el resultado de la prueba es positivo, el control del programa se transfiere a una localidad determinada; mientras que si la condición especificada no encuentra, la ejecución del programa continúa con la siguiente instrucción.

El único modo de direccionamiento aplicable a las instrucciones de brinco condicionados es el relativo.

BCC - Salta si no hay acarreo (igual que BHS).

BCS - Salta si hay acarreo (igual que BLO).

BEQ - Salta si igual que cero.

BHCC - Salta si no hay medio acarreo.

BHCS - Salta si hay medio acarreo.

BHI - Salta si es mayor que cero.

BHS - Salta si es mayor o igual que cero (igual que BCC).

BIH - Salta si la línea de interrupción externa está en alto.

BIL - Salta si la línea de interrupción externa está en bajo.

BLO - Salta si es menor que cero (igual que BCS).

BLS - Salta si es menor o igual que cero.

BMC - Salta si el bit de máscara de interrupción está en cero.

BMI - Salta si es menor.

BMS - Salta si el bit de máscara de interrupción está en uno.

BNE - Salta si no es igual a cero.

BPL - Salta si es mayor.

BRA - Salta siempre.

BRN - Nunca saltes.

BSR - Salta a subrutina.

Instrucciones de manipulación de bits individuales.

Estas instrucciones cambian el estado de un bit individual de cualquier localidad de página cero. El modo de direccionamiento aplicable es este grupo es el directo. El número de bit (0 - 7) es parte del código de operación.

BCLR N - Pon en cero el bit N (N=0...7)

BSET N - Pon en uno el bit N (N=0...7).

Instrucciones de prueba de bits individuales.

Estas instrucciones prueban el estado de un bit individual de una localidad en página cero, y dependiendo del resultado de la prueba, el control del programa se transfiere a una dirección determinada o continúa con la siguiente instrucción, en forma similar a las instrucciones de brincos condicionados. El modo de direccionamiento aplicable a estas instrucciones es el relativo.

Este tipo de instrucciones requieren de 3 octetos: uno para el código de operación que incluye la dirección del bit a probar; uno para especificar la dirección en página, cero del octeto a probar y, uno para especificar el desplazamiento relativo en caso de que se efectúe un brinco.

BRCLRN - Salta si el bit N en memoria está en cero.

BRSETN - Salta si el bit N en memoria está en uno.

Instrucciones de control.

Las instrucciones en este grupo tienen direccionamiento inherente. Estas instrucciones manipulan los bits de código de condición, controlan la pila y la operación de las interrupciones así como también transfieren datos desde el acumulador y el registro de índice.

CLR - Borra el bit de acarreo.

CLI - Borra el bit de la máscara de interrupción.

NOP - No operación

POP - Restablece el apuntador de pila (S).

RTI - Regresa de la interrupción.

RTS - Regresa de la subrutina.

SEC - Pon en uno el bit de acarreo.

SEI - Pon en uno el bit de la máscara de interrupción.

SWI - Interrupción programada.

TAX - Transfiere el acumulador al registro de índice.

TXA - Transfiere el registro de índice X al acumulador.

## Sistema de desarrollo de MC146805E2

El desarrollo de la circuitería y programación de la CUTCMOS se llevó a cabo en un sistema EXORCISER II con capacidad para emular al microprocesador MC146805E2.

El sistema consta de dos procesadores: el 6809 que controla el ducto del EXORCISER, y a través del cual pueden manejarse todos los recursos del EXORCISER, y el procesador MC146805E2, que reside en una tarjeta fuera del ducto. En esta forma, el procesador 6809 trabaja como amo y el MC146805E2 trabaja como esclavo y la interconexión entre ellos se realiza a través de una tarjeta interfaz.

### Operación.

Desde el punto de vista de desarrollo, la elaboración de la programación se efectúa en el EXORCISER, bajo el control del 6809, mientras que la función de emulación se hace en el MC146805E2.

Una de las características más importantes de este sistema es que la tarjeta que contiene al MC146805E2 puede utilizarse en forma autónoma como una microcomputadora independiente.

Cuando la tarjeta del procesador se conecta al ducto del EXORCISER a través de la interfaz, el EXORCISER controla su operación y permite al usuario la verificación y monitoreo de programas ejecutados por el MC146805E2.

## APENDICE IV.

## Tiempo y Frecuencia.

## El meridiano estándar.

La Tierra está dividida en 24 zonas de tiempo de 15 grados de ancho. Cada una tiene una diferencia estándar de un número, entero de horas con respecto al tiempo solar medio en el meridiano de Greenwich.

En 1884 se propuso al meridiano de Greenwich, Inglaterra como la referencia estándar en las mediciones de tiempo y longitud.

## El Tiempo Universal

En 1928, se recomendó utilizar la designación Tiempo Universal (Universal Time o Weltzeit, abreviado UT) para el tiempo solar medio en el meridiano de Greenwich, iniciado a partir de la media noche. Sin embargo, el término GMT (Tiempo medio de Greenwich) ha persistido, probablemente debido a que su significado es más claramente comprensible que el de UT en su utilización en el ámbito internacional de la navegación y las telecomunicaciones.

El tiempo Universal Coordinado (UTC).

El cambio más importante ocurrido en la comunidad de Tiempo y Frecuencia fue el inicio de la época del Tiempo Universal Coordinado (UTC). Esto ocurre el 1 de enero de 1972 (Chi, A. R., Fosque, H. S., 1972).

Gracias a este cambio en las escalas de tiempo se logra uniformizar (al menos en los países pertenecientes al CCIR) la divulgación de Tiempo. De acuerdo al nuevo sistema UTC, se adopta el segundo atómico como unidad eliminando así la necesidad de cambiar la frecuencia natural de oscilación de los relojes atómicos. Esto se hacía anteriormente en algunos lugares donde se tenía como escala de tiempo el sistema SA (Smith, Humphry M., 1972).

El Tiempo y su medición.

La medición del Tiempo se basa en la rotación de la tierra y por lo tanto en el movimiento aparente de la esfera celeste. El intervalo entre tránsitos sucesivos del meridiano de un objeto de referencia elegido se define como el Día. El objeto de referencia, como el sol o una estrella fija, determinará el tipo de día del que se trate. Sideral si se elige una estrella fija y Solar si se elige el sol. Cada tipo de día se encuentra dividido en 24 partes iguales.

llamadas horas. Definiremos ahora el tiempo (en términos geométricos) como la hora ángulo del objeto de referencia.

Sin embargo, la velocidad de rotación varía irregularmente, aunque en forma pequeña. Por ejemplo, el deshielo de las capas congeladas de Groenlandia cambia el momento de inercia del planeta y por tanto la velocidad angular de su rotación. Las mareas lunares (y en menor grado las solares) disminuyen también la velocidad de rotación de la tierra por fricción, principalmente en el profundo mar de Bering. Se vuelve claro entonces que se requiere de una referencia de tiempo más estable.

Se ha hablado del tiempo solar como si fuera constante. Sin embargo, el día Solar no es precisamente constante en cuanto a su duración a través del año. Hay varias causas para esto, aparte de los cambios de velocidad de rotación de la tierra.

Una de estas causas es que que la hora ángulo del sol se mide a lo largo del ecuador celeste, y consecuentemente está determinado por la proyección del eclíptico en el ecuador. El hecho de que el ángulo eclíptico no es cero sino 23.5 grados, significa que el valor de la proyección cambia de una manera sinusoidal a lo largo del año.

Además, resulta que la órbita de la tierra alrededor

del sol no es circular, sino que es una elipse ligeramente excéntrica. Las leyes del movimiento planetario causan que la velocidad orbital del planeta varíe a lo largo del año a consecuencia de esa excentricidad. Estos dos efectos, (la elipticidad y la variación de velocidad) cuando se superponen, causan que la longitud aparente del día solar varíe de manera irregular durante el año. El día aparente es un poco más largo en los meses de Invierno, alrededor de Diciembre.

Es deseable corregir este efecto, y para eso se ha desarrollado el concepto de Día Solar Medio. Este 'Día' son las 24 horas que transcurren entre los tránsitos del meridiano de un punto imaginario llamado 'sol medio'. El sol medio se mueve hacia el este a lo largo del ecuador celeste completando su ciclo de la misma manera que lo hace el sol a lo largo de la eclíptica.

El Tiempo Solar Medio es derivado de este sol ficticio. El segundo solar medio se define como la  $\frac{86}{400}$ ava parte del día solar medio, o del día solar aparente promedio. El Tiempo Medio solar es anunciado a través de varias estaciones de radio en todo el mundo. En los Estados Unidos, por ejemplo, se hace a través de las estaciones WWV, WWVB, WWVH.

Un caso especial muy útil es el Tiempo Universal (UT),

es el Tiempo solar Medio en la longitud de Greenwich, Inglaterra. UT es útil en la correlación de estudios de fenómenos mundiales en base a un tiempo común. El Tiempo Universal basado en la rotación de la tierra, es por lo tanto irregular y es corregido de esas irregularidades en la rotación mediante frecuentes observaciones estelares. Para una predicción correcta de alta precisión de los tiempos universales de los eventos celestes, se ha inventado una 'ficción matemática' llamada el tiempo 'efemérides'. Este tiempo transcurre uniformemente y su unidad arbitraria constante es la longitud del año tropical que se sucedió en el inicio del año 1900 dividido entre 31 556 925.97474, que fue el número de segundos ese año. Las correcciones para convertir el tiempo efemérides a UT se determinan por observaciones de los tiempos universales cuando la luna alcanza puntos en el espacio predichos por los tiempos de efemérides. En el presente siglo el tiempo efemérides se ha ido adelantando al UT. Particularmente, el tiempo efemérides iba adelante de UT por 35 segundos en 1970 (Haymes, Robert, 1971).

Como se vió anteriormente, la rotación de la tierra no es constante. Ha sido necesario para tal efecto distinguir tres tipos de escalas de tiempo para UT:

UTO es el tiempo observado (Este tiempo no es propiamente universal ya que es distinto en cada lugar de la Tierra).

UT1 es UTO tomando en cuenta la corrección por efectos de la variación polar (Este tiempo ya es uniforme en toda la Tierra).

UT2 es UT1 corregido con los cambios de velocidad en la rotación terrestre.

Conceptos y caracterización de la divulgación de Tiempo y Frecuencia.

Conceptos fundamentales.

En esta sección se considerarán algunos conceptos básicos y se establecerá la terminología necesaria para entender los conceptos de divulgación de Tiempo y Frecuencia.

Relojes.

El componente fundamental del rastreo del tiempo es el reloj. Un reloj es un dispositivo que acumula los ciclos de un oscilador y presenta el resultado de alguna forma conveniente. Generalmente, un reloj consiste de un estándar de frecuencia, un esquema de conteo de la frecuencia, y un dispositivo de visualización para el despliegue de las cuentas. Adicionalmente, el reloj puede contar con un dispositivo de interpolación para usarlo

entre cuentas.

El más antiguo de los relojes utiliza como oscilador la rotación de la tierra. El hombre cuenta los amaneceres y lleva la cuenta con un calendario. La unidad característica de este reloj es el día. Puede utilizarse un reloj de pared o un reloj de sol para interpolar entre amaneceres. Un reloj más reciente es el que utiliza la resonancia de los átomos de Cesio para controlar un oscilador, y utiliza circuitería electrónica para llevar la cuenta de las oscilaciones y desplegarla apropiadamente, el reloj de Cesio es mucho más estable que el reloj terraqueo, esto es, la longitud de un segundo atómico es casi la misma que cualesquier otro segundo atómico, en cambio la longitud del día solar puede variar considerablemente.

Fecha o Epoca.

Este concepto se aplica al nombre que se dá a un instante específico en una escala ordenada, por ejemplo, Noviembre 23, 03 hs, 15 min, 40.12 seg. En otras palabras, la fecha es el tiempo de un evento determinado relacionado a un origen arbitrario. Este concepto, del tiempo está generalmente relacionado con la posición del sol.

### Frecuencia.

Frecuencia e intervalo de tiempo están relacionados de manera inversa dimensionalmente. Para medir la frecuencia de ocurrencia de algún fenómeno se cuenta el número de veces que el evento se verifica durante un intervalo medido de tiempo. El concepto de intervalo de tiempo se refiere a duración, por lo que un intervalo de tiempo puede ser medido correctamente por un reloj que no esté a tiempo pero que trabaje a la velocidad correcta.

### Sincronización.

Para que dos observadores estén sincronizados en tiempo es necesario que sus relojes muestren la misma lectura a un tiempo, aunque eso no significa necesariamente que sea la misma lectura que alguna escala patrón. En principio no es necesario que en sistemas sincronizados se encuentren alineadas sus bases de tiempo u osciladores con los de la comunidad de usuarios de dicho sistema. Sin embargo es sumamente ventajoso para el sistema y los usuarios del mismo el permitir que los osciladores se encuentren alineados.

Las opciones para llevar cuenta del tiempo son muchas. El costo y la complejidad de los distintos

métodos se relaciona directamente con la precisión requerida, la localización del usuario, y el intervalo de resincronización requerido. Por ejemplo, un marinero navegando por el océano Pacífico requiere del tiempo para navegar. El marinero necesita de un reloj cuyo error acumulado sea pequeño. Este usuario tendría la opción de elegir un reloj caro que pudiera sincronizar en el momento de la partida o podría escoger un reloj más barato que sincronizara todos los días con alguna señal de estándar de tiempo como la WWVH o la JJY.

Debido a que la marca de la señal está a tiempo en el momento de ser transmitida es necesario hacer algunas correcciones por el retraso debido a la propagación de la onda de radio. El retraso de propagación de una onda de radio para una distancia de 1 km es de aproximadamente 3 microsegundos. Por lo tanto, una señal de tiempo llegará con 3 ms de retraso después de haber recorrido 1000 kms de distancia desde el transmisor. Se deberá hacer una consideración adicional al tomar en cuenta el retraso del paso de la señal a través del receptor (retraso de receptor).

Las transmisiones de radio de estándares de tiempo y frecuencia tales como la WWV, WWVH, CHU, y JJY son más que adecuadas para las necesidades diarias. Sin embargo con el lanzamiento de satélites a partir de finales de los 50s se requiere de sistemas sincronizados en todo el mundo.

La posibilidad de disponer de relojes atómicos portátiles estimuló el diseño de sistemas de navegación y comunicaciones sincronizados con precisión de microsegundos.

Una necesidad básica de todos estos sistemas es que sus relojes muestren el mismo tiempo. Si la sincronización se pierde debido a fallas en la electricidad o cualesquier otra causa el reloj debería de volverse a sincronizar. Un método para lograr esta sincronización sería el trasladar un reloj portátil de la estación de referencia a el reloj que se ha parado.

#### Técnicas de divulgación de Tiempo y Frecuencia.

Un sistema de divulgación de tiempo y frecuencia puede caracterizarse por su precisión, ambigüedad, repetibilidad, facilidad de uso, costo al usuario y número de usuarios servidos. Para explicar estas características veamos el siguiente ejemplo. Considérese un sistema simple consistente de una señal de 10 KHz sin modulación. Como lo muestra la fig. 15: Un flanco de subida de la señal, que sale del transmisor a las 0000 UT, deberá arribar al receptor en un tiempo posterior que será equivalente al retraso de propagación  $t_d$ . La exactitud de la medición del tiempo del usuario no podrá ser mejor que la exactitud con la que pueda predecir este retraso. Debido a que todos los

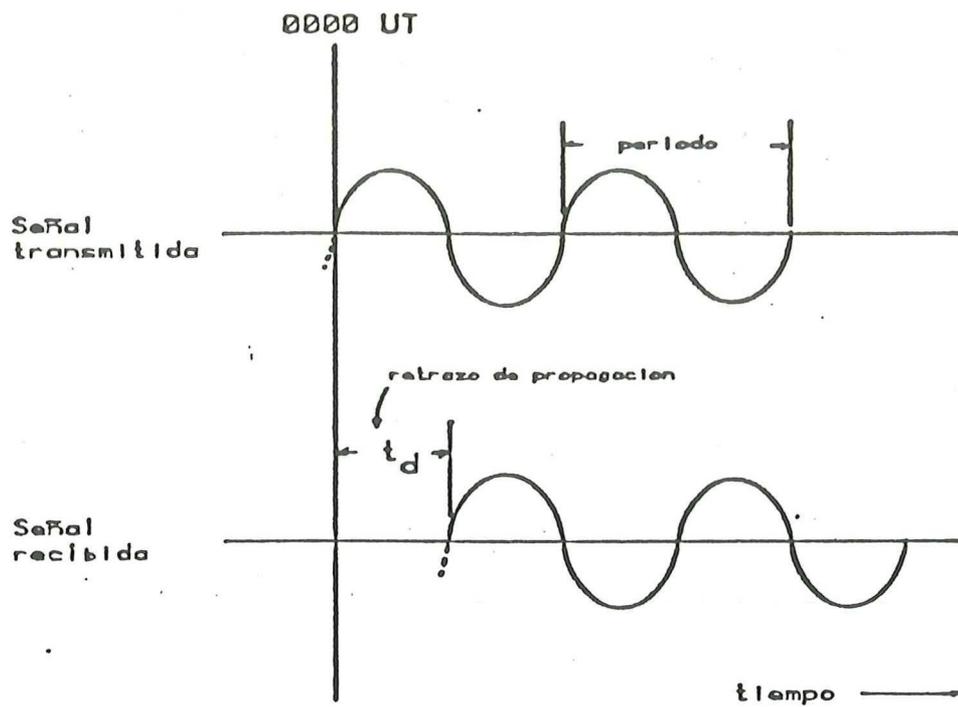


Figura 18. Esquema de divulgación de tiempo.

ciclos de la señal son idénticos, la señal es ambigua y el usuario tendrá que decidir de alguna manera cual es el ciclo que corresponde al ciclo de 'a tiempo'. Esto quiere decir que para el caso hipotético de nuestra señal de 10 KHz, el usuario deberá saber previamente el tiempo con una precisión de  $\pm 50$  microseg (medio período de la señal para poder sincronizar con la señal patrón). Supongamos ahora que el usuario no es capaz de medir el retraso de propagación pero que este retraso es más o menos el mismo día con día. En este caso se dice que el sistema tiene una buena repetibilidad. Si se puede transportar un reloj portátil de la fuente de la señal al receptor, éste puede medir el retraso de propagación y poner su reloj a tiempo con gran exactitud.

Algunos usuarios, como los Geofísicos, se interesan en hacer mediciones coordinadas en el tiempo sobre grandes áreas geográficas. Ellos necesitan que todas estas mediciones estén referidas a la misma base de tiempo, en este caso la cobertura del sistema es importante.

Otra característica importante de un sistema de divulgación de tiempo es el porcentaje en el cual el servicio de tiempo se encuentra disponible. Por ejemplo, el ciudadano común que tiene que atender a sus citas requiere de una precisión de unos minutos. Aunque requiere de una

base de tiempo burda, la necesita en todo momento de tal forma que carga con un reloj de pulsera que le proporciona el tiempo las 24 horas del día. Por otra parte, el usuario que requiere de precisiones de microsegundos tendrá un muy buen reloj que sólo necesitará sincronizar una o dos veces al día.

Una característica importante también en los sistemas de divulgación de tiempo es su confiabilidad, esto es la probabilidad de que una señal de tiempo esté disponible en el momento en el que se supone debe de estarlo. Por ejemplo, en algunas ocasiones uno no puede recibir señales de HF debido a desvanecimientos en la propagación.

El uso del radio en la divulgación del tiempo se remonta a principios del siglo 20.

En la propagación de señales de tiempo y frecuencia por medio de transmisión de radio se utiliza alta frecuencia (HF), baja frecuencia (LF), y muy baja frecuencia (VLF).

Actualmente existen alrededor de 15 estaciones de transmisión de estándares de tiempo y frecuencia en la banda de HF (3 a 30 MHz). Esta banda cuenta con un gran grupo de usuarios (Morgan, A. H., 1967).

En la banda de LF (30 a 300 KHZ) existe actualmente una tendencia hacia una mayor utilización. Con el advenimiento de la revolución de los circuitos integrados y los

microprocesadores, se ha dado un gran impulso a sistemas de divulgación de estándares de Tiempo y Frecuencia en la banda de LF (Frank, R. L., 1983).

En la banda de VLF (10 a 30 KHz) se cuenta con unas nueve estaciones de transmisión de estándares de Tiempo y Frecuencia.

En esta banda es donde se obtienen mejores resultados en la divulgación de tiempo y frecuencia a grandes distancias en cuanto a precisión. Esta banda es ampliamente utilizada para la sincronización de relojes a distancia (Morgan, A. H., 1967).

Dentro de las estaciones de transmisión de estándares de tiempo y frecuencia tenemos que las más cercanas a México son las que transmite la oficina de estándares de los Estados Unidos (NBS). Algunas en HF (WWV, WWVH) y otras en LF (WWVB, LORAN-C, NAA, NPG-NLK, NSS, NPM). En México no existe ninguna estación transmisora para la divulgación de estándares de tiempo y frecuencia.

## APENDICE V.

En este apéndice se presentan los listados del programa de la aplicación geofísica, los diagramas de flujo de los programas de la aplicación geofísica y, la tabla de estados utilizada para el procesamiento de la WWVB.

```

00001 *****
00002 D
00003 D
00004 D
00005 D
00006 D
00007 D
00008 D
00009 D
00010 D
00011 *****
00012 D

```

```

PROGRAMA
DEL RELOJ SINCRONIZADO CON
LA HHVE:

```

```

FERNANDO FAVELA U.
AERIL 84/ REV 1.2

```

```

00013 OPT CMO, LLE=120, S, G, NOMEX, F=50
00014 TTL RELOJ SINCRONIZADO CON LA HHVE: VER. 1M
00015
00016
00017

```

DECLARACION DE CONSTANTES

00018	0080	A OCHETA	EQU	\$80
00019	0070	A SETETA	EQU	\$70
00020	00EF	A EEFEE	EQU	\$EF
00021	0011	A HEX11	EQU	\$11
00022	000D	A CR	EQU	\$0D
00023	000A	A LF	EQU	\$0A
00024	00E1	A SOH	EQU	\$E1
00025	003A	A PUNTOS	EQU	\$3A
00026	00EF	A EEFEE	EQU	\$EF
00027	00FF	A EFEF	EQU	\$FF
00028	007E	A SIETEE	EQU	\$7E
00029	002E	A DOSEEE	EQU	\$2E
00030	0014	A CONT35	EQU	\$14
00031	0003	A TRES	EQU	\$03
00032	0004	A CUATRO	EQU	\$04
00033	0005	A CINCO	EQU	\$05
00034	0006	A SEIS	EQU	\$06
00035	0060	A SESENT	EQU	\$60
00036	0066	A SESEIS	EQU	\$66
00037	00F2	A NOVCIN	EQU	\$F2
00038	000C	A CINCOF	EQU	\$0C
00039	0059	A TRENCI	EQU	\$59
00040	00A5	A SEICUA	EQU	\$A5
00041	0002	A CEROTE	EQU	\$02
00042	0001	A UNOTE	EQU	\$01
00043	0003	A MARCOA	EQU	\$03
00044	0032	A TREDOS	EQU	\$32
00045	0031	A TRESEEE	EQU	\$31
00046	003F	A TREFEE	EQU	\$3F
00047	0000	A PRIMO	EQU	\$00
00048	0055	A SINSEC	EQU	\$55
00049	0005	A MAS	EQU	CINCO
00050	0000	A FTDA	EQU	\$0000

UN 20 DECIMAL.

```

EN PORCENTAJE 95%
EN PORCENTAJE 105% O 05%
35%
65%

```

00051	0001	A FTDE	ERU	\$0001
00052	0004	A FTDA	ERU	\$0004
00053	0005	A FTDF	ERU	\$0005
00054	0008	A TDR	ERU	\$0008
00055	0005	A TCF	ERU	\$0005
00056	0104	A RTKA	ERU	\$0104
00057	0108	A RTKB	ERU	\$0108
00058	010C	A RTKC	ERU	\$010C
00059	010D	A RTKD	ERU	\$010D
00060	0115	A SIGNO	ERU	\$0115
00061	0100	A SEGUND	ERU	\$0100
00062	0100	A RELOJ	ERU	\$0100
00063	0014	A RELEDO	ERU	\$0014
00064	0018	A ENT	ERU	\$0018
00065		»		
00066		»		RESERVACION DE REGISTROS Y
00067		»		LOCALIDADES TEMPORALES
00068		»		
00069A	001A		ORG	\$001A
00070		»		
00071A	001A	0001	A TSEGUN	RME: 1
00072A	001B	0001	A TTERC	RME: 1
00073A	001C	0001	A CONOVA	RME: 1
00074A	001D	0001	A QUALIT	RME: 1
00075A	001E	0001	A ANTIGU	RME: 1
00076A	001F	0001	A CIFRAS	RME: 1
00077		»		
00078A	0025		ORG	\$0025
00079		»		
00080A	0025	0001	A DIAJUL	RME: 1
00081A	0026	0001	A DIAJUN	RME: 1
00082A	0027	0001	A REGISC	RME: 1
00083A	0028	0001	A INTCON	RME: 1
00084A	0029	0001	A ENHW	RME: 1
00085A	002A	0001	A REGFTE	RME: 1
00086A	002B	0001	A CONT	RME: 1
00087A	002C	0001	A POSIC	RME: 1
00088A	002D	0001	A ESTATU	RME: 1
00089		»		
00090A	0120		ORG	\$0120
00091		»		
00092A	0120	0001	A TEMPAN	RME: 1
00093A	0121	0001	A TEMPAC	RME: 1
00094A	0122	0001	A SINCRD	RME: 1
00095A	0123	0001	A LAISSA	RME: 1
00096A	0124	0001	A LAISSO	RME: 1
00097		»		
00098	0024	A TTKANS	ERU	\$0024
00099		»		
00100		»		*****

PAGE 003 FINALVM.SA:1 RELOJ SINCRONIZADO CON LA WAVE VER. 1M

001014 1800		ORG	1800	ORIGEN DEL PROGRAMA
00101				*****
00103	007F	A LAJA	1007F	
00104	00F0	A FFE	1F0	
00105	000A	A NUMO	10A	NUMERO DE DATOS A DESPLEGAR(10)
00106				
00107				VARIABLES
00108				
00109	0012	A NUM	10012	LOCALIDAD DEL NUM. DE DATOS
00110	0080	A TEMP	10800	LOCALIDAD TEMPORAL DE DATOS
00111				
00112				CONSTANTES
00113				
00114	0024	A HORA	124	
00115	007E	A INL	17E	
00116	0010	A PCHAT	110	
00117	0030	A ASCII	130	
00118	00F3	A FTRES	1F3	
00119	000F	A UNID	10F	PALABRA PARA SEPARAR US.
00120	00F0	A DECE	1F0	PALABRA PARA SEPARAR DS.
00121				*****
00122				
00123				DOSDEC
00124				
00125				
00126				SUBROUTINA DE ENVIO DE DOS DIGITOS EN UN BYTE
00127				A LA TERMINAL
00128				
00129				
00130				*****
00131A 1810 F6		DOSDEC	10F6	CARGA EL DATO A DESPLEGAR
00132A 1801 44		LSRA		COLOCA LAS DECENAS
00133A 1802 44		LSRA		
00134A 1803 44		LSRA		
00135A 1804 44		LSRA		
00136A 1805 AD 01	1808	ESR	DECI	SALTA A LA RUTINA DE CONVERSION Y DESPLIEGUE
00137				
00138A 1807 F6		UNDEC	10F6	CARGA DE NUEVO EL BYTE PARA LA UNIDADES
00139				
00140A 1808 A4 0F		A DECI	AND 10UNID	CHECA QUE SEAN SOLO LAS UNIDADES
00141A 180A AE 30		A	ADD 10ASCII	SUMA 30 PARA CONVERTIR DECIMAL A ASCII
00142				
00143A 180C 10 03		A UART	ESET 10PTOR	FOR EL SELECTOR EN LECTURA DEL BIT QUE INDICA
00144A 180E 12 01		A	ESET 11PTOR	QUE EL BUFFER DE TRANSMISION ESTA VACIO.
00145A 1810 05 01	FD 1810	LOOP	TRCLP 11PTOR	QUE QUEDA QUE HASTA QUE PUEDAS ESCRIBIR EL DATO
00146A 1813 07 00		A	STA 11PTOR	SACA EL DIGITO
00147A 1815 1F 03		A	ECLR 11PTOR	CARGALO EN EL UART
00148A 1817 1A 01		A	ESET 11PTOR	MANDALO
00149A 1819 11 03		A	ECLR 11PTOR	
00150A 181B 13 03		A	ECLR 11PTOR	REGRESA A MONITOREAR LA WAVE.

```

00151A 181D 01          RTS
00152          B          FIN DE LA SUBROUTINA DE DESPLIEGUE DE DIGITOS
00153          B          *****
00154          B
00155          B
00156          B          ACTUAL
00157          B
00158          B          ATENCION A LA INTERRUPCION DE ACTUALIZACION
00159          B          *****
00160          B
00161A 181E 01 2D 0D 182E ACTUAL BRCLK PRIMO,ESTATU,UIDATA SALTA A EFECTUAR UIDATE SI ES NORMAL
00162A 1821 11 2D          A      BRCLK PRIMO,ESTATU LIMPIA LA BANDERA DE PRIMER UPDATE.
00163A 1823 0D 1A1A          A      JSR  LIMPIA REINICIALIZA EL RTR.
00164A 1827 16 0100          A      LDA  SEGUND REESTABLECE EL VALOR DE LOS SEGUNDOS.
00165A 1829 9A          DELA
00166A 182A 07 0100          A      STA  SEGUND
00167A 182C 00          RTI          RETORNA DE LA INTERRUPCION.
00168          B
00169A 182E A6 0D          A      UPDATA LDA  #CR  TRANSMITE UN RETORNO DE CARRO
00170A 1830 AD 0A          180C     BSR  UART
00171A 1831 A6 0A          A      LDA  #LF  Y UN AVANCE DE LINEA
00172A 1834 AD 0A          180C     BSR  UART
00173          B
00174A 1837 AC 81          A      TRHORA LDA  #SOH  INICIA LA TRANSMISION CON UN PRINCIPIO DE
00175A 1838 AD 02          180C     BSR  UART          ENCAJEZADO (CTRL A)
00176A 183A 0A 14 30 186D     BRSET  S,REGEDO,ERRKORT SALTA SI NO SE TERMINO DE TRANSMITIR A TIEMPO
00177A 183D 14 14          A      BRSET  S,REGEDO INHIBE POSIBILIDAD DE PERTURBAR LA TABLA DE TRANSMISION
00178A 183F 5E          A      SEI          DESHABILITA INTERRUPCIONES
00179A 1840 AE FF          A      TRLIST LDX  #EFEF  APUNTA A BASE PARA CREAR LA LISTA DE TRANSMISION
00180A 1841 16 01          A      LDA          #01,X  CARGA SEGUNDOS
00181A 1843 17 10          A      STA          (TTRANS-4) ALMACENALOS EN LA LISTA DE TRANSMISION
00182A 1845 18 0F          A      LDA          #03,X  CARGA LOS MINUTOS
00183A 1848 17 21          A      STA          (TTRANS-3) ALMACENALOS
00184A 184A E6 05          A      LDA          #05,X  CARGA LAS HORAS
00185A 184C E7 22          A      STA          (TTRANS-2) ALMACENALAS
00186A 184E E6 25          A      LDA          #25  CARGA LOS DIAS
00187A 1850 E7 23          A      STA          (TTRANS-1) ALMACENALOS
00188A 1852 E8 26          A      LDA          #26  CARGA LAS CENTENAS DE LAS DIAS
00189A 1854 E7 24          A      STA          TTRANS  ALMACENALAS
00190A 1856 9A          CLI          TERMINAMOS, PERMITE INTERRUPCIONES.
00191A 1857 A6 04          A      LDA          #CUATRO CARGA EN EL CONTADOR EL NUMERO DE CIFRAS A TRANSM.
00192A 1859 E7 1F          A      STA          CIFRAS
00193A 185B AE 24          A      LDX          #TTRANS. POSICIONA X EN EL PRINCIPIO DE LA TABLA A TRANSM.
00194A 185D AD 4D          1807     BSR  UNDEC  TRANSMITE CENTENAS
00195A 185F 5A          DECX          PASA A SIG. CIFRA
00196A 1860 AD 9E          1800     TRNSM  BSR  DOSDEC  TRANSMITE CIFRAS
00197A 1862 54          DECX          PASA A SIGUIENTE CIFRA
00198A 1863 57 1F          A      DEC          CIFRAS  DECREMENTA CONTADOR DE CIFRAS
00199A 1865 27 0F          1872     BSR  CALIDA  SI ES CERO TERMINO Y VA A TRANSMITIR LA CALIDAD
00200A 1867 A6 3A          A      LDA          #PUNTOS  SI NO TRANSMITE LOS DOS PUNTOS
    
```

PAGE 005 FINALVIA.SA:1

RELOJ SINCRONIZADO CON LA HWV: ULK. 1M

```

00201A 1864 A5 A1 180C ESK UART
00202A 186F 21 F3 1860 ERA TRANSF CONTINUA TRANSMITIENDO
00103
00204A 186D CC 1899 A ERRORI JMP LIMPER *****
00205A 1870 20 AC 1E1E UFIDAT ERA ACTUAL TRAMPOLIN A ACTUAL.
00206
00207
00208
00209A 1872 C6 0100 A CALIDA LDA SEGUNO VERIFICA SI INICIA EL MINUTO
00210A 1875 26 1A 1891 ENE TRACAL NO, TRANSMITE CALIDAD
00211A 1877 04 2D 15 188F ERSET 2,ESTATU,NOSIN VERIFICA SI HAY ERROR, NO>TRANSMITE CALIDAD
00212A 187A 3C 1E A INC ANTIGU SI ERROR, INCREMENTA LA ANTIGUEDAD DE LA HORA
00213A 187C 24 02 1880 ENE EQUIVA SI NO HAY SOBREFLUJO, DETERMINA CALIDAD
00214A 187E 3A 1E A DEC ANTIGU LEEGO A SU MAXIMO
00215
00216
00217
00218A 1880 A5 05 A EQUIVA LDX 4CUATRO INICIALIZA X
00219A 1802 E6 1E A LDA ANTIGU CARGA LA ANTIGUEDAD ACTUAL.
00220A 1884 D1 1FE0 A REUISA CMP TABCAL,X COMPARA A CONTRA LAS CALIDADES
00221A 1887 24 04 188D EHS CARCAL VE A CARGAR CALIDAD
00222A 1889 5A DECX REUISA EL SIGUIENTE
00223A 188A 24 F8 1884 EPL REUISA SIGUE VIENDO
00224A 188D 5F CLRX SI SE PASA LIMPIA X
00225A 188D E8 1D A CARCAL STX QUALIT ALMACENA EL APUNTAOR DE CALIDAD
00226A 188F 15 2D A USBIA ECLR 2,ESTATU AVISA QUE NO HAY SINCRONIZACION. (FOR OMISSION)
00227A 1891 E8 1D A TRACAL LDX QUALIT RECUPERA APUNTAOR DE CALIDAD
00228A 1893 D8 1FE5 A LDA TABCAL+5,X CARGA CALIDAD (D)
00229A 1894 CD 180C A JSF UART TRANSMITELO
00230A 1894 1E 14 A LIMPER ECLR 5,REGEDO INDICA QUE YA SE PUEDE CAMBIAR LA TABLA DE TRANSMICION
00231A 1898 8C RTI RETORNA DE LA INTERRUPCION (ACTUALIZACION)
00232
00233
00234
00235
00236
00237
00238
00239
00240
00241A 189C E7 27 A ATNRTR STA REGISC ALMACENA EL REG C DEL RTR EN PAG 0
00242A 189E E6 2A A LDA REGPTE CARGA EL VALOR QUE TENIA EL PTO E
00243A 18A0 E7 D1 A STA F10E PARA REINSTALARLO.
00244A 18A2 0A 27 70 1915 ERSET 5,REGISC,ALARMA SALTA A ATENDER A LA ALARMA (RTR2)
00245A 18A5 08 27 08 1D 0 ERSET 4,REGISC,UFIDAT SALTA A ATENDER LA ACTUALIZACION (RTR1)
00246
00247
00248
00249
00250

```

---

```

00251
00252
00253A 18A8 1C 01 A INFERI ESET 6,F10E HABILITA LOS VISUALIZADORES.
00254A 18AA AD 40 18E1 ESEF 6ACA MUESTRA LOS DATOS EN EL VISUALIZADOR
00255A 18AC 1D 01 A ECLR 6,F10E DESHABILITA LOS VISUALIZADORES.
00256A 18AE 5A 2E A DEC INTCON DECREMENTA EL CONTADOR DE VECES
00257A 18B0 26 0E 18E4 ENEC REGPER SI NO ES CERO RETORNADE LA INTERRUCCION
00258A 18B2 A6 EF A LDA 0EEFE APAGA LA INTERRUCCION PERIODICA
00259A 18B4 C4 010E A AND RTRE EN EL RTR
00260A 18B7 C7 010E A STA RTRE
00261A 18BA B0 REGPER RTI RETORNA DE AL INTERRUCCION PERIODICA
00262
00263
00264
00265
00266
00267
00268
00269A 18BE E6 2A A ATNINT LDA REGTE REINSTALA EL VALOR ORIGINAL DEL F10E.
00270A 18E0 E7 01 A STA F10E
00271A 18E2 A6 80 A LDA 00CHETA CARGA LA INICIALIZACION DEL CONTADOR
00272A 18E4 E7 2E A STA INTCON DE INTERRUCCIONES PERIODICAS EN INTCON
00273A 18E8 44 LSRA MUEVE EL BIT '1' A LA POSICION 6
00274A 18CA CA 010E A ORA RTRE FRENEDE EL BIT 6 DEL REG E DEL RTR
00275A 18C7 C7 010E A STA RTRE PARA PERMITIR LA INTERRUCCION PERIODICA
00276A 18CA B0 RTI RETORNA DE LA INTERRUCCION
00277
00278
00279
00280
00281
00282
00283
00284A 18CE E6 01 A INTERR LDA F10E ALMACENA EL ESTADO ACTUAL DE EL F10 E
00285A 18D0 E7 2A A STA REGTE PARA REINSTALARLO DESPUES.
00286A 18CF C6 010C A LDA RTRE LEE EL REGISTRO C DEL RTR PARA AVERIGUAR SI ESTA
00287A 18D2 2E CB 189C EMI ATNRTK INTERRUMPIENDO Y SALTA A ATENDERLO
00288A 18D4 11 01 A ECLR 6,F10E COLOCA EL MUX APUNTANDO A HWVE.
00289A 18D6 13 01 A ECLR 1,F10E
00290A 18D8 04 01 7A 1955 ERSET 2,F10E,ATNHW VE SI ES INTERRUCCION DE LA HWVE Y ATIENDE
00291A 18DE 10 01 A ESET 0,F10E CAMBIA EL SELECTOR PARA REVISAR SI ES EL UART
00292A 18DD 04 01 43 194B ERSET 2,F10E,ATNURT VE SI ES INTERRUCCION DEL UART Y ATIENDE
00293A 18E0 11 01 A ECLR 0,F10E
00294A 18E2 12 01 A ESET 1,F10E AFUNTA A EL INTERRUPTOR.
00295A 18E4 04 01 D9 18E1 ERSET 2,F10E,ATNINT VE A ATENDER AL INTERRUPTOR.
00296A 18E7 E6 2A A LDA REGTE REINSTALA EL VALOR ORIGINAL DEL F10E.
00297A 18E9 E7 01 A STA F10E FALSA ALARMA.
00298A 18EF B0 RTI
00299
00300

```

```

00301
00302
00303
00304
00305
00306
00307A 18E0 A6 03 A SACA LDA #TRES CARGA CONTADOR DE DATOS
00308A 18E1 E7 2E A STA CONT Y FONLO EN EL CONT
00309A 18F0 AC 20 A LDX # (TRANS-4) CARGA APUNTAOR DE TABLA
00310A 18F2 3F 2C A CLR FOSIC LIMPIA LA POSICION DE LOS DIGITOS
00211A 18F4 F6 VIENE LDA 0xX CARGA EL DATO A DESPLEGAR
00312A 18F5 4F LSLA
00313A 18F6 4B LSLA
00314A 18F7 4E LSLA
00315A 18F8 4B LSLA
00316A 18F9 AD 11 190C JSR DIGIT VE A LA SUBROUTINA QUE DESPLIEGA UN DIGITO
00317A 18FE CD 1F30 A JSR RETARDI SALTA A UN RETARDO PARA QUE SE VEA.
00318A 18FF F6 LDA 0xX
00319A 18FF AD 0C 190C JSR DIGIT DESPLIEGA EL OTRO DIGITO DEL OCTETO
00320A 1901 CD 1F30 A JSR RETARDI SALTA A UN RETARDO PARA QUE SE VEA.
00321A 1904 5C INX INCREMENTA APUNTAOR
00322A 1905 3A 2E A DEC CONT DECREMENTA CONTADOR
00323A 1907 26 EF 18F4 RNE VIENE NO TERMINO ? VIENE OTRA VEZ
00324A 1908 3C 00 A INC FTDA AFAZA LAS POSICIONES.
00325A 190E B1 RTS TERMINO Y RETORNA.
00326
00327
00328
00329
00330
00331
00332
00333
00334
00335A 190C A9 F0 A DIGIT AND #EFE SEPARA EL NIELE MAYOR
00336A 190E EA 2C A ORA FOSIC ANEXA LA POSICION DEL DIGITO EN EL DISPLAY
00337A 1910 E7 00 A STA FTDA FONLO EN EL FUERTO A
00338A 1912 3C 2C A INC FOSIC FON EL APUNTAOR DE DIGITO EN EL SIGUIENTE
00339A 1914 B1 RTS RETORNA DE LA SUBROUTINA
00340
00341
00342
00343
00344
00345
00346
00347
00348
00349
00350A 1915 E6 25 A ALARMA LDA DIAJUL INCREMENTA EL DIA JULIANO.

```



```

00401A 195E 07 01 22 1980      ERCLF 3,PTOE:BAJADA SALTA A ATENDER BAJADAS.
00402          »
00403          »          RUTINA DE ATENCION A LOS FRENTE DE ONDA POSITIVOS
00404          »
00405A 195E 17 01          A SUBIDA BCLR 3,PTOE: LIMPIA LA INTERRUPCION.
00406A 1960 AD 4E 1910      BSR ANCHOF SALTA A DETERMINAR EL ANCHO DEL PULSO.
00407A 1961 08 14 35 199A    ERSET 4,REGEDO,ERRORW ABORTA EL PROCESO SI REGRESO CON ERROR.
00408A 1965 01 0123      A ADD LAPSOA SUMALO AL ANCHO DEL PULSO ALTO.
00409A 196E 25 06 1970      ECS CIENTO A VER SI ES MAYOR DEL 105%.
00410A 196A A1 F2          A CMP QNOVCIN COMPARA CON EL 95% Y
00411A 196C 25 2C 199A      ELO ERRORW VERIFICA SI ES MAYOR, SI NO VE A ERROR.
00412A 196E 20 04 197A      ERA TRECEIN SALTA A CONTINUAR
00413A 1970 A1 0C          A CIENTO CMP QNCINCOI COMPARA CON EL 105%.
00414A 1972 22 26 199A      EHI ERRORW SALTA A ERROR SI ES MAYOR.
00415A 1974 0D 14 08 197F  TRCEIN ERCLF 6,REGEDO,RETHW ES TODO RETORNA DE LA INTERRUPCION.
00416A 1977 02 0120      A LDA TEMIAN SI ES EL SEG 35, ALMACENA EL TIEMPO DE
00417A 197A C7 0122      A STA SINCRO SINCRONIZACION.
00418A 197D 1D 14          A BCLR 6,REGEDO RETORNA A SU ESTADO NORMAL.
00419A 197F 80          » RETHW RTI          RETORNA DE LA INTERRUPCION POR HWVE.
00420          »
00421          »          RUTINA DE ATENCION AL FRENTE DE ONDA NEGATIVO
00422          »
00423A 1980 16 01          A BAJADA BSET 3,PTOE: AFAGA LA INTERRUPCION.
00424A 1982 AD 2C 1910      BSR ANCHOF SALTA A DETERMINAR EL ANCHO DEL PULSO.
00425A 1984 08 14 13 199A    ERSET 4,REGEDO,ERRORW ABORTA EL PROCESO SI REGRESO CON ERROR.
00426A 1987 C7 0123      A STA LAPSOA ALMACENA EL VALOR.
00427          »
00428A 198A A1 0C          A DESICI CMP QNCINCOI ESTABELECE SI ES *1*,*0*,MARCA O ERROR.
00429A 198C 25 0C 199A      ELO ERRORW SI ES MENOR AL 5%, ERROR.
00430A 198E A1 59          A CMP QTRENCI REvisa SI ES MENOR AL 35%. (5*AF<35)
00431A 1990 25 0C 199E      ELO CERDEN SI, ENTONCES ES UN *0*.
00432A 1992 A1 A5          A CMP QSEICUA REvisa SI ES MENOR AL 64%. (35*AF<64)
00433A 1994 25 0C 19A2      ELO UNDEN SI, ENTONCES ES UN *1*.
00434A 1996 A1 F2          A CMP QNOVCIN REvisa SI ES MENOR AL 95%. (64*AF<95)
00435A 199E 25 0C 19A6      ELO MARCAE SI, ENTONCES ES UNA MARCA.
00436          »
00437A 199A DC 1E:02      A ERRORW JMP ERRORI
00438          »
00439A 199D 80          » RTI          RETORNA DE LA INTERRUPCION.
00440          »
00441A 199E A6 02          A CEROTE LDA QCEROTE LA ENTRADA ES UN CERO. (02)
00442A 19A0 20 06 19AE      ERA ENTRA ENTRA
00443A 19A2 A6 01          A UNDEN LDA QUNOTE LA ENTRADA ES UN UNO. (01)
00444A 19A4 20 02 19AB      ERA ENTRA ENTRA
00445A 19A6 A6 03          A MARCAE LDA QMARCOA LA ENTRADA ES UNA MARCA. (03)
00446          »
00447A 19AB E7 1E          A ENTRA STA ENT ALMACENA EL VALOR EN LA VARIABLE DE ENTRADA.
00448A 19AD 9E          A SEI INHIBE LAS INTERRUPCIONES DURANTE EL PROC. DE EDOS.
00449A 19AE CD 1D:1F      A JSR PROCED VE A RECORRER EL PROCESADOR DE ESTADOS.
00450A 19AE 9A          A CLI PERMITE DE NUEVO LAS INTERRUPCIONES.
    
```

```

00451A 19A1 80          RTI          REGRESA DE LA INTERRUPCION DE LA MME.
00452          »
00453          »          SUBROUTINA DE ANCHO DE PULSO.
00454          »
00455A 19E0 C7 0121    A ANCHOP STA  TEMPAC  ALMACENA EL TIEMPO ACTUAL.
00456A 19E3 C1 0120    A          CMF  TEMPAN  COMPARALO CON EL TIEMPO ANTERIOR.
00457A 19E6 25 12      19CA    BLD  TSECON  REvisa SI REBASO UN SEGUNDO.
00458A 19E8 E6 1A      A          LDA  TSEGUN  REvisa SI TSEGUN ES CERO.
00459A 19EA 61 01      A          CMF  QUNOTE  COMPARA CON 001.
00460A 19EC 26 26      19E4    BNE  ERROKA  NO HAY CONCORDANCIA, ES ERROR.
00461A 19EE 3F 1A      A          CLF  TSEGUN  LIMPIA PASO POR CERO.
00462A 19F0 9A          A          CLI          PERMITE INTERRUPCIONES.
00463A 19F1 C6 0121    A CALCUM LDA  TEMPAC  CALCULA EL ANCHO ANCHO=ANT+FF-ACT
00464A 19F4 43          A          COMA  LDA          COMPLEMENTA ACUMULADOR.
00465A 19F5 CF 0120    A          ADD  TEMPAN  SUMALE EL TIEMPO ANTERIOR.
00466A 19F8 20 0D      19D7    BNA  TRANSI  TRANSI
00467A 19FA E6 1A      A TSECON LDA  TSEGUN  REvisa QUE SEA UN CERO EN TSEGUN.
00468A 19FC 26 16      19E4    BNE  ERROKA  Y MARCA ERROR SI NO ES.
00469A 19FE 3F 1A      A          CLR  TSEGUN  LIMPIA PASO POR CERO.
00470A 19F0 9A          A          CLI          PERMITE INTERRUPCIONES.
00471          »
00472A 19D1 C6 0120    A CALCUL LDA  TEMPAN  CALCULA ANCHO ANCHO=ANT-ACT.
00473A 19D4 C0 0121    A          SUR  TEMPAC
00474          »
00475A 19D7 C7 0124    A TRANSI STA  LAPSO
00476A 19DA C6 0121    A ASIVA  LDA  TEMPAC  ACTUALIZA EL VALOR ANTERIOR DE TIEMPO.
00477A 19DD C7 0120    A          STA  TEMPAN
00478A 19E0 C6 0124    A          LDA  LAPSO
00479A 19E3 D1          A          RTS          RETORNA DE LA SUBROUTINA DE ANCHO DE PULSO.
00480          »
00481A 19E4 1E 14      A ERROKA ISET 4, REGEDO FRENDE EL BIT DE ERROR.
00482A 19E7 3F 1A      A          CLR  TSEGUN  LIMPIA PASO POR CERO.
00483A 19EA 5A          A          CLI          PERMITE INTERRUPCIONES.
00484A 19ED 20 EF      19DA    BNA  ASIVA  SALTA A RENOVAR REGISTROS.
00485          »
00486          »
00487          »          TIMER1
00488          »
00489          »          RUTINA DE ATENCION AL TEMPORIZADOR INTERNO
00490          »          DEL PROCESADOR. (BASE INTERNA DE 1/256)
00491          »
00492          »
00493A 19EE 1F 09      A TIMER1 BCLR 7,TCR  LIMPIA LA INTERRUPCION.
00494A 19F1 30 1A      A          JNC  TSEGUN  AVISA QUE PASO POR CERO.
00495A 19F4 0F 14 21 1A12 BICLR 7,REGEDO,RETIM SI 1435, NADA MAS, RETORNA.
00496A 19F7 5A 1E      A          DEC  TIM35  NO, DECREMENTA TIM35. (CONTADOR DE 19 SEG.)
00497A 19FA 2B 31      1A27    BPI  CRONOS  SI NO HA LLEGADO A CERO, RETORNA.
00498A 19FD 26 1E      1A12    BNE  RETIM  SALTA A VER SI HAY QUE SINCRONIZARSE.
00499          »
00500          »          RUTINA PARA INICIAR LA SINCRONIZACION
    
```

```

00500
00502A 19FB 19 09 A FRENDI E:CLF 4,TCR PREPARA SINCRONIZACION, PARA TIMER INTERNO.
00503A 19FA 16 09 A E:SET 3,TCR PON EN CLIC LO CONTADORES.
00504A 19FC 06 0122 A LDA SINCRO CALCULA EL TIEMPO DE SINCRONIZACION
00505A 19FF 43 A COMA COMPLEMENTA LA LECTURA DEL TIMER.
00506A 1A00 0E 29 A ADD ERHW AGREGALE EL RETRAZO Y PONLO EN EL TIMER.
00507A 1A02 25 04 1A0B E:CS FRENDI SI HAY ACARREO AVISA EN LA BANDERA CONRESONDIENTE.
00508A 1A04 13 2D A E:CLF 1,ESTATU NO HAY ACARREO, AFAGA LA BANDERA DE ACARREO.
00509A 1A06 20 02 1A0A E:KA VO SALTA A CONTINUAR.
00510A 1A08 12 2D A FRENDI E:SET 1,ESTATU FRENDI LA BANDERA DE ACARREO EN SINCRONIZACION.
00511A 1A0A E7 0B A VO STA TDR
00512A 1A0C 18 09 A E:SET 4,TCR ARKANCA TIMER
00513A 1A0E A6 80 A LDA 00CHETA INHIBE AL RELOJ RTR.
00514A 1A10 C7 010E A STA RTRE
00515A 1A12 80 RETIM RTI RETORNA DE LA INTERRUPTION.
00516
00517
0051F
00519A 1A14 A6 70 A LIMPIA LDA 00SETTA CARGA PALABRA PARA
00520A 1A16 C7 010A A STA RTRE LIMPIA: LOS REGISTROS DE CONTEO.
00521A 1A18 A6 2D A LDA 00DSEI CARGA LA PALABRA DE CONTROL PARA
00522A 1A1E C7 010A A STA RTRE PROGRAMAR BASE DE TIEMPO Y PERIODICIDAD.
00523A 1A1E A6 32 A LDA 00TREDOS CARGA PALABRA PARA
00524A 1A20 C7 010E A STA RTRE PROGRAMAR INTERRUPTION DE ALARMA Y UPDATE.
00525A 1A22 C6 010C A LDA RTRE LIMPIA FUSIBLES INTERRUPTIONES.
00526A 1A24 81 RTS RETORNA DE SUBROUTINA.
00527
00528
00529
00530A 1A27 AD EE 1A1A CRONOS E:SR LIMPIA INICIALIZA RTR.
00531A 1A29 19 09 A E:CLI 4,TCR INICIALIZA EL TIMER EN 0120.
00532A 1A2B 16 09 A E:SET 3,TCR
00533A 1A2D A6 80 A LDA 00CHETA
00534A 1A2F E7 0B A STA TDR
00535A 1A31 18 09 A E:SET 4,TCR
00536A 1A33 A6 0D A LDA 00CR
00537A 1A35 CD 180C A JSR UART
00538A 1A38 A6 0A A LDA 00LF
00539A 1A3A CD 180C A JSR UART
00540
00541A 1A3D AE 0F A RENOVA LDX 00UNID POSICIONA X, EN LA BASE PARA USAR DESPLAZAMIENTO EN 1 EI
00542A 1A3F 3F 1C A CLR CONOVA LIMPIA EL CONTADOR
00543A 1A41 3C 1C A INC CONOVA PONLE UN UNO.
00544A 1A43 E6 FF A PASA LDA (RELOJ-1),X CARGA DECENAS.
00545A 1A45 48 LSLA COLOCA LA POSICION EN LAS DECENAS.
00546A 1A46 48 LSLA
00547A 1A47 48 LSLA
00548A 1A48 48 LSLA
00549A 1A49 5C INCX
00550A 1A4A EA FF A ORA (RELOJ-1),X

```

```

00551A 1A4C E7 F2      A      STA      (RELOJ-10E)*X
00552A 1A4E 5C          INDX
00553A 1A4F 3A 1C      A      DEC      CONOVA
00554A 1A51 2A F0      1A45  BIL      FASA
00555A 1A53 5C          INDX
00556A 1A54 5C          INDX
00557A 1A55 E6 FD      A      LDA      (RELOJ-3)*X COLOCA LAS CENTENAS A TRANSMITIR.
00558A 1A57 E7 26      A      STA      DIAJUH EN SU LUGAR.
00559A 1A59 E6 FE      A      LDA      (RELOJ-2)*X Y LUEGO LAS DECENAS.
00560A 1A5E 4B          LSLA
00561A 1A5C 4B          LSLA
00562A 1A5D 4B          LSLA
00563A 1A5E 4B          LSLA
00564A 1A5F E1 FF      A      ADD      (RELOJ-1)*X Y LAS UNIDADES.
00565A 1A61 E7 23      A      STA      DIAJUH
00566A 1A63 A2 55      A      LDA      *SINSEG PREPARA SEGUNDO DE SINCRONIZACION.
00567A 1A65 03 2D 01 1A73 BICLR 1,ESTATU,SEA SI NO HUBO ACARREO CARGA DIRECTAMENTE.
00568A 1A68 CE 0115    A      LD      SIGNU REvisa SI ES + O - LA SINCRONIZACION.
00569A 1A6E A3 05      A      CFX      *MAS
00570A 1A6D 27 03      1A72  BEQ      SUMA SI ES UN + SUMALE UN SEGUNDO.
00571A 1A6F 4A          DECA     SINO, FUE MENOS LE RESTAS UN SEGUNDO.
00572A 1A70 20 01      1A73  BNA     SEA CONTINUA NORMALMENTE.
00573A 1A72 4C          SUMA    INCA
00574A 1A73 C7 0100    A SEA  STA     SEGUNDO ALMACENA EL SEGUNDO QUE LE CORRESPONDI.
00575A 1A76 CD 18:00    A      JSR     INIPRO VE A INICIALIZAR EL PROCESADOR DE ESTADOS.
00576A 1A79 14 2D      A      BSET   2,ESTATU AVISA QUE SINCRONIZO. (AVISO AL PROGRAMA)
00577A 1A7E 1D 01      A      BCLR   6,FTOE AFAGA VISUALIZADORES. (AVISO AL USUARIO)
00578A 1A7D 15 14      A      BCLR   2,RECEO LIMPIA LA BANDERA DE ERROR.
00579A 1A7F 10 2D      A      BSET   PRIMO,ESTATU PRENDE BANDERA DE PRIMEF UPDATE.
00580A 1A81 3F 1D      A      CLR     QUALIT LIMPIA EL APUNTAOR DE CALIDAD.
00581A 1A83 3F 1E      A      CLR     ANTIGU LIMPIA EL REGISTRO DE ANTIGUEDAD DE LA HORA.
00582A 1A85 CC 1834    A      JMP     TRHORA SALTA A TRANSMITIR LA HORA. (UPDAT)
00583
00584
00585
00586
00587
00588
00589
00590
00591A 1A8E 9E          RESETA SEI
00592A 1A89 9C          RSP     INICIA STACK POINTER.
00593A 1A8A A6 FF      A      LDA     *EFEF
00594A 1A8C E7 04      A      STA     CFTOA PROGRAMA FTOA COMO SALIDAS.
00595A 1A8C 4F          CLEA
00596A 1A8F E7 00      A      STA     FTOA
00597A 1A91 C7 0123    A      STA     LAPSOA
00598A 1A94 E7 1A      A      STA     TSEGUN
00599A 1A96 E7 14      A      STA     RECEO
00600A 1A98 C7 0017    A      STA     ENL

```

```

00601A 1A9E E7 18 A STA ENT
00602A 1A9D E7 25 A STA DIAJUL
00603A 1A9E E7 26 A STA DIAJUH
00604A 1A9E E7 29 A STA ERHW
00605A 1A9B E7 2D A STA ESTATU
00606A 1A95 C7 0120 A STA TEMPAN
00607A 1A9B C7 0122 A STA SINCO
00608A 1A9E C7 0123 A STA LAPSOA
00609A 1A9E A6 FF A LDA 0EFEF INICIALIZA EL REGISTRO DE ANTIGUEDAD
00610A 1A9D E7 1E A STA ANTIGU
00611A 1A9E A6 04 A LDA 0CUATRO INICIALIZA EL REGISTRO DE CALIDAD.
00612A 1A94 E7 1D A STA QUALIT
00613A 1A9A A6 70 A LDA 0SETETA EFECTUA EL RESET DE LOS CONTADORES
00614A 1A9E C7 010A A STA RTKA
00615A 1A9E A6 80 A LDA 0OCHETA PERMITE MODIFICAR EL RELOJ.
00616A 1A9E C7 010E A STA RTRE:
00617A 1A9D AE 04 A LDA 0DIEZ
00618A 1A9E 6F FF A BLANCO CLR 0EFEFX LIMPIA LAS LOCALIDADES DEL RTR.
00619A 1A94 5A DECC
00620A 1A9E 26 FE 1A9E BLANCO FNE BLANCO
00621A 1A9E A6 01 A LDA 0UNOTE
00622A 1A9E C7 0100 A STA SEGUNO INICIA EL SEGUNDO EN UNO.
00623A 1A9E A6 32 A LDA 0TREDOS PROGRAMA INT DE UNFOT Y ALARMA.
00624A 1A9E C7 010E A STA RTRE:
00625A 1A9E A6 21 A LDA 0DUSEE PROGRAMA 32,768 HZ COMO BASE DE TIEMPO.
00626A 1A9E C7 010A A STA RTKA
00627A 1A9E C6 010C A LDA RTKC LIMPIA INTERRUPCIONES PENDIENTES.
00628A 1A9E A6 7E A LDA 0SIETE: PROGRAMA FTDE:
00629A 1A9E E7 05 A STA CFTDE:
00630A 1A9E A6 70 A LDA 0SETETA PROGRAMA PARAMETROS DE FTDE:
00631A 1A9E E7 01 A STA FTDE:
00632A 1A9E 21 02 1A9E BHI SIGUEL REvisa que no exista interrupcion
00633A 1A9E 16 01 A BSET 3,FTDE:
00634A 1A9E CD 1F00 A SIGUEL JSR MESS SACA EL MENSAJE DE INICIO. (JA)
00635A 1A9E 22 09 1AF3 BHI REVIN VERIFICA QUE NO EXISTA INTERRUPCION DE LA HWVE.
00636A 1A9E 07 01 04 1AF1 BRCLR 3,FTDE:FREEIT SI ESTA APAGADO VE A PRENDERLO.
00637A 1A9E 17 01 A BRCLR 3,FTDE: LIMPIALA SI EXISTE.
00638A 1A9E 20 02 1AF3 BRA REVIN
00639A 1AF1 16 01 A FREEIT BSET 3,FTDE: PRENDE EL BIT DE HWVE.
00640A 1AF3 A6 3F A REVIN LDA 0TREFEE PROGRAMA TIMER.
00641A 1AF5 E7 09 A STA TCR:
00642A 1AF7 A6 FF A LDA 0EFEF INICIALIZALO.
00643A 1AF9 E7 0E A STA TDR:
00644A 1AFB CD 1E:0C A JSR INIFRO INICIALIZA EL PROCESADOR DE ESTADOS.
00645A 1AFE 5A CLI
00646A 1AFF BF ESPERA WAIT
00647A 1E00 20 FD 1AFF BRA ESPERA SIGUE ESPERANDO.
0064E
00649A 1E02 19 14 A ERROR: BRCLR 4,REGEDO LIMPIA EL ERROR.
00650A 1E04 3D 1E A TST TIM35 REvisa que no sea durante la sincronizacion

```

PAGE 014 TEGJE .AL:1

PAGE 014 FINALVIM.SA:1

RELOJ SINCRONIZADO CON LA HWVE VER. JM

00651A	11:06	27	03	11:01	DEQ	REGRE	SI, REGRESA SIN HACER NADA YA QUE ES CASO ESPECIAL.
00652A	11:06	00	11:00	A	JSR	INIFRO	INICIALIZA EL PROCESADOR DE ESTADOS YA QUE ES ERROR.
00653A	11:06	00			REGRE	R1J	RETORNA A ESPERAR INTERRUPCION.
00654							

```

00656
00657
00658
00659
00660
00661
00662
00663
00664
00665
00666
00667
00668
00669
00670
00671
00672
00673
00674
00675
00676
00677
00678
00679
00680
00681A 1E:0C A6 D6
00682A 1E:0E E7 10
00683A 1E:10 A6 B1
00684A 1E:12 E7 13
00685A 1E:14 A6 1D
00686A 1E:16 E7 11
00687A 1E:18 A6 10
00688A 1E:1A E7 12
00689A 1E:1C CC 1E:86
00690
00691
00692
00693
00694
00695
00696
00697
00698
00699
00700
00701
00702
00703
00704
00705

```

```

*****
SUBROUTINA MANO
*****
ESTA SUBROUTINA ES LA QUE SE ENCARGA DE EXTRAER LOS OCTETOS
CORRESPONDIENTES EN LA TABLA DE ESTADOS.
LA SUBROUTINA SE CARGA EN MEMORIA DE LECT/ESC POR EL MODO DE
DIRECCIONAMIENTO UTILIZADO.
MANO LDA X:DEDO CARGA EN EL ACC EL DATO AFUNTADO POR X EN EL EDU DEDO
RTS RTS RETORNA
*****
INIFRO
*****
SUBROUTINA DE INICIALIZACION DEL PROCESADOR
DE ESTADOS
*****
A MEMO EQU $0010 DIRECCION DONDE SE VA A PONER LA SUBROUTINA MANO.
A TAELAH EQU $1D DIRECCION EN DONDE SE INICIA LA TABLA DE ESTADOS.
A TAELAL EQU $10
*****
A INIFRO LDA $1D6 CODIGO DE LDA EXTENDIDO
STA MEMO ALMACENADO EN RAM.
LDA $E81 CODIGO DE RTS.
STA MEMO+3
LDA $TAELAH
STA MEMO+1
LDA $TAELAL
STA MEMO+2
JMP ERROR SALTA A INICIALIZAR COMO ERROR.
*****
PROCED
*****
PROCESADOR DE ESTADOS
*****
FERNANDO FAVELA VANA
1 7 83
REV. 2R.10.83
*****
A ESTADO EQU $0015
A AUXH EQU $0016
A ENL EQU $0017
A RECEP EQU $0019

```

PAGE 016 FINALVM.SA:1 RELOJ SINCRONIZADO CON LA MMVE VER. 1M

```

00706      0034      A RHOBA EQU 40034
00707      0010      A MAND EQU 80010
00708      0011      A DEDOH EQU (MAND+1)
00709      0012      A DEDOL EQU (DEDOH+1)
00710
00711      000A      A DIEZ EQU 10A
00712      007F      A SIETEF EQU 87F
00713
00714
00715A 1E1F 5F          PROCED CLRX          LIMPIA EL APUNTAOR DEL ESTADO.
00716A 1E20 ED 10      A EUCLA JSR MAND     CARGA BYTE DE LA TABLA DE ESTADOS
00717A 1E22 27 38      1E5C  EER FINALX    TERMINA EL ESTADO SI BYTE=00
00718A 1E24 2A 03      1E29  EFL CONDI     SI BIT 7=0, SALTA A EVALUAR CONDICION
00719
00720
00721
00722
00723
00724
00725
00726
00727
00728A 1E26 5C          INCX                SI NO, EVALUA CONDICIONES INICIALES.
00729A 1E27 ED 10      A JSR MAND          EN ESTE CASO, NO HAY,SE USA EL BYTE PARA CARGAR ENC
00730A 1E29 AD 33      1E5E  CONDI ESR     CONDIC SALTA A LA RUTINA QUE EVALUA LA CONDICION
00731A 1E2E 00 14 0A 1E3E  ERESET 0*REGEDO,EXECU SI LA CONDICION ES VALIDA SALTA A EJEC. ACCIONES
00732A 1E2E 5C          SIGCON INCX        FASA AL SIGUIENTE OCTETO EN LA TABLA DE ESTADOS
00733A 1E2F ED 10      A JSR MAND          CARGA OCTETO
00734A 1E31 2A FE      1E2E  EFL SIGCON    SALTA OCTETOS HASTA ENCONTRAR EL SIG. COD. DE COND.
00735A 1E33 5C          INCX                SALTA TRES VECES
00736A 1E34 5C          INCX
00737A 1E35 5C          INCX
00738A 1E36 20 EB      1E20  ERA EUCLA    CONTINUA LA EVALUACION DEL ESTADO
00739
00740A 1E38 5C          EXECU INCX         FASA AL SIGUIENTE OCTETO
00741A 1E39 ED 10      A JSR MAND          CARGALO
00742A 1E3E 2A 04      1E41  EFL FASALE    SALTA A LA TRANSFERENCIA A SUBRUTINAS DIECTO
00743A 1E3D 14 14      A ESET 2*REGEDO   PRENDE EL BIT DE ULTIMO (ULTIMA SUBRUTINA)
00744A 1E3F A9 7F      A AND *SIETEF    LIMPIA EL BIT 7 DEL ACCUMULADOR
00745A 1E41 EF 15      A FASALE STX     ESTADO SALVA EL REG. X ACTUAL EN LA LOC. ESTADO
00746A 1E43 97          TAX              TRANSFIERE 'A' A 'X'
00747A 1E44 D0 1E7A    A JSR SUBRUT,X   VE A EJECUTAR LA ACCION CORRESPONDIENTE
00748A 1E47 EE 15      A LDX ESTADO     RECUPERA 'X'
00749A 1E49 05 14 EC 1E3E  BRCLR 2*REGEDO,EXECU SALTA SI EL BIT DE ULTIMO ESTA APAGADO. NO ULT
00750A 1E4E 15 14      A BCLR 2*REGEDO  LIMPIA EL BIT DE ULTIMO.
00751A 1E4E 5C          INCX              SI ES EL ULTIMO, FASA A TRANSFERIR AL OTRO EDO.
00752A 1E4F ED 10      A JSR MAND        TOMA LA DIRECCION ALTA DEL SIG. ESTADO
00753A 1E51 E7 16      A STA AUXH       ALMACENALA EN LA LOCALIDAD AUXH
00754A 1E53 5C          INCX
00755A 1E54 ED 10      A JSR MAND        TOMA LA DIRECCION BAJA DEL SIG. ESTADO Y

```

```

00756A 1E5E E7 12 A STA DEDD ALMACENALA EN EL REGISTRO DE DIRECCION DE ESTADO BAJO
00757A 1E5E E8 1E A LDA AUXH TOMA LA DIRECCION DE LA LOCALIDAD AUXILIAR
00758A 1E5A E7 11 A STA DEDDH ALMACENALA EN EL REGISTRO DE DIRECCION DE ESTADO ALTO
00759A 1E5C 5F FINALX CLRX INICIALIZA EL REGISTRO X
00760A 1E5D E1 RTS RETORNA DE LA VUELTA (FUERDE SER RTS O BRK O JMP)
00761 *****
00762 *
00763 * CONDIC
00764 * SUBROUTINA PARA EVALUAR
00765 * CONDICIONES
00766 *
00767 *****
00768A 1E5C 11 14 A CONDIC ECLR 0+REGEDO DE ENTRADA LA CONDICION ES FALSA (BIT 0 DE RECEDD=0)
00769A 1E60 A1 7F A CMF 4SIGTEF VERIFICA SI ES UN SIGA.
00770A 1E62 27 0E 1E6C EEO VERDAD SI ES, ENTONCES SALTE CON VERDADERO.
00771A 1E64 A4 03 A AND 4TRES LIMPIA LA CONDICION
00772A 1E66 26 07 1E6F ENE CHECK SALTA SI ES DISTINTA DE 00
00773A 1E68 E6 17 A LDA ENE CARGA ENE (NUMERO DE DATOS A RECIBIR)
00774A 1E6A 26 0D 1E79 ENE RETURN SI NO ES 00 REGRESA
00775A 1E6C 10 14 A VERDAD ESET 0+REGEDO SI =00 LA CONDICION SE CUMPLE (BIT 0 DE RECEDD=1)
00776A 1E6E E1 RTS REGRESA DE LA LLAMADA
00777A 1E6F E1 1E A CHECK CMF ENT COMPARA CONDICION CON EL VALOR DE LA ENTRADA
00778A 1E71 27 F9 1E6C EER VERDAD SALDA A VALIDAR CONDICION SI ES IGUAL
00779A 1E73 A1 03 A CMF 4TRES COMPARA A VER SI ES MAYOR QUE TRES
00780A 1E75 23 02 1E79 ELS RETURN SI ES MENOR O IGUAL REGRESA CON VALOR FALSO F/COND.
00781A 1E77 12 14 A ESET 1+REGEDO SI ES MAYOR HAY UN ERROR
00782A 1E79 E1 RETURN RTS REGRESA DE LA LLAMADA
00783 *****
00784 *
00785 * SUBROUTINAS EJECUTABLES
00786 * POR EL PROCESADOR DE ESTADOS
00787 *
00788 *****
00789 1E7A A SUBRUT EQU * AQUI ES SUBRUT
00790 *
00791 010E A TABLAR EQU 010E BASE DE LA TABLA DE RECEPCION.
00792 000E A ONCE EQU 11 NUMERO DE DATOS A RECIBIR.
00793 *
00794A 1E7A E1 NADA RTS NO HACE NADA
00795 *
00796A 1E7E 5F CARGAN CLRX LIMPIA EL APUNTAOR DEL ESTADO
00797A 1E7C E0 10 A JSI: MAND CARGA OCTETO
00798A 1E7E A4 03 A AND 4TRES LIMPIA
00799A 1E80 E7 17 A STA ENC CARGALO EN LA VARIABLE ENE
00800A 1E82 E1 RTS
00801 *
00802A 1E83 13 14 A NERROR ECLR 1+REGEDO APAGA EL BIT DE ERROR
00803A 1E85 E1 RTS
00804 *
00805A 1E86 12 14 A ERROF ESET 1+REGEDO PRENDE EL BIT DE ERROR

```

PAGE 018 FINALVJM.SA11

RELOJ SINCRONIZADO CON LA MANIPUL. 1M°

00006A	1E8B	1F	14	A	BCLR	7	REGEDO	APAGA	BIT	DE	SEG	35.							
00007A	1E8A	1D	14	A	BCLR	6	REGEDO	APAGA	BIT	DE	SEG	34.							
00000A	1E8C	A6	14	A	LDA	0	CONT35	CAMBIAR	CUANDO	SEA	NECESARIO.	*****							
00009A	1E8E	17	1E	A	STA	7	TIME	INICIALIZAR	EL	CONTADOR	DE	SEG. 35.							
00010A	1E90	3F	19	A	EQUIE	CLP	RCLP	LIMPIA	EL	AFUNTADOR	DE	RECEPCION	DE	DIGITOS.					
00011A	1E92	3F	34	A	CLP	RHORA		LIMPIA	EL	REGISTRO	DE	RECEPCION	DE	DIGITOS.					
00012A	1E94	B1			RTS														
00013																			
00014A	1E95	E6	18	A	CARDIG	LDA	ENT	CARGA	LA	ENTRADA	AL	SISTEMA							
00015A	1E97	A1	01	A	CMF			0	(UNO-140)	REVISAR	SI	ES	UN	UNO					
00016A	1E99	27	03	1E9E	EED			LIMP											
00017A	1E9E	98			CLC														
00018A	1E9C	20	01	1E9F	EKA			METE											
00019A	1E9E	99			LIMP	SEC													
00020A	1E9F	39	34	A	METE	ROL	RHORA												
00021A	1E91	3D	17	A	TSI			ENE											
00022A	1E93	26	10	1E95	ENE			MISMO											
00023A	1E95	1E	19	A	LDX			RECIB	CARGA	EL	AFUNTADOR	A	LOS	DIGITOS	RECIBIDOS.				
00024A	1E97	E6	34	A	LDA			RHORA	CARGA	EL	DIGITO	RECIBIDO.							
00025A	1E99	D7	010E	A	STA			TAELAR,X	GUARDA	LA	CIFRA	COMPLETADA.							
00026A	1E9E	3F	34	A	CLR			RHORA	LIMPIALA	PARA	EL	SIGUIENTE	DIGITO.						
00027A	1E9E	5C			INCX				INCREMENTA	EL	AFUNTADOR.								
00028A	1E9F	A1	0E	A	CMF			0	ONCE										
00029A	1E91	2A	0E	1E90	EFL			EQUIE											
00030A	1E93	E7	19	A	STX			RECIB	SALVA	EL	AFUNTADOR.								
00031A	1E95	B1			MISMO	RTS													
00032																			
00033A	1E96	3A	17	A	DECENE	DEC		ENE											
00034A	1E98	B1			RTS														
00035																			
00036A	1E99	1E	14	A	SEC35	BSET		7	REGEDO	PRENDE	EL	BIT	DE	QUE	YA	VIENE	EL	SEC	35.
00037A	1E91	1C	14	A		BSET		6	REGEDO	PRENDE	EL	BIT	PARA	ACTIVAR	LA	SINCRONIZACION.			
00038A	1E93	B1			RTS														
00039																			
00040A	1E9E	9D			FREFAR	NOF													
00041																			
00042																			
00043																			
00044																			
00045																			
00046																			
00047																			
00048																			
00049																			
00050																			
00051	0003	A	NOCIF	EQU		\$03													
00052	000E	A	OCHO	EQU		\$0E													
00053	000F	A	CEREF	EQU		\$0F													
00054	0115	A	MILIE	EQU		\$0115								LOCALIDAD DE LAS CIFRAS DE RECEPCION EN MILIS.					
00055	002F	A	COP1A	EQU		\$002F								LOCALIDAD DE TRABAJO DE LAS CIFRAS					

FERNANDO FAVELA 10.27.83

00056	0029	A	RESUL	ERU	00029	LOCALIDAD EN DONDE SE ALMACENARA EL RESULTADO
00057	0033	A	CJFCON	ERU	00033	LOCALIDAD DEL CONTADOR DE CIFRAS
00058						
00059A	1E1F AE 03	A	INAR	LDX	0NOCIF	CARGA EL NUMERO DE CIFRAS A CONVERTIR
00060A	1E1C D6 0115	A	PASALO	LDA	MILIS.X	TOMA LAS CIFRAS RECIBIDAS Y
00061A	1E1A E7 2F	A		STA	COFIA.X	PASALAS A PAGINA CERO PARA
00062A	1E1C 5A			DECX		TRABAJARLAS
00063A	1E17 26 FB	1E1C1		ENE	PASALO	
00064A	1E19 3F 29	A	CONVER	CLC	RESUL	LIMPIA LA LOCALIDAD DEL RESULTADO
00065A	1E1E A6 08	A		LDA	00CHO	CARGA LA RESOLUCION EN BITS (8)
00066A	1E1D E7 33	A		STA	CJFCON	ALMACENALA EN EL CONTADOR
00067A	1E1F AD 14	1E1E5	DOBLE	FSK	DUPLIC	LLAMA A LA SUBROUTINA DE X2
00068A	1E11 39 29	A		KOL	RESUL	CARGA EL BIT RESULTANTE
00069A	1E13 3A 33	A		DEC	CJFCON	DECREMENTA EL CONTADOR DE BITS
00070A	1E15 26 FB	1E1CF		ENE	DOBLE	CONTINUA HASTA TERMINAR
00071A	1E17 C6 0115	A		LDA	SIGNO	REVISAS SI VA SIMPLE O EN COMP. A 2.
00072A	1E1A 41 05	A		CMF	0MAS	
00073A	1E1D 27 06	1E1E4		EEQ	RETSU	REGRESA NORMAL
00074A	1E1E E6 29	A		LDA	RESUL	CALCULA EL COMPLEMENTO A 2.
00075A	1E1E 43			COMA		
00076A	1E11 4C			INCA		
00077A	1E1E E7 29	A		STA	RESUL	REGRESALO A SU REGISTRO
00078A	1E14 B1		RETSU	RTS		RETORNA DE LA LLAMADA
00079						
00080						
00081						
00082A	1E1E AE 03	A	DUPLIC	LDX	0TRES	ALMACENA EL NUMERO DE CIFRAS
00083A	1E17 9E			CLC		INICIALIZA EL ACARREO
00084A	1E1E E6 2F	A	DTCIF	LDA	COFIA.X	TOMA LA CIFRA
00085A	1E1A E9 2F	A		ADC	COFIA.X	SUMALA CONSIGO MISMA
00086A	1E1C A1 0A	A		CMF	0DIEZ	COMPARA EL RESULTADO CON DIEZ
00087A	1E1E 25 04	1E1F4		ELO	ALMAC	SI ES MENOR ESTA BIEN
00088A	1E1F AE 06	A		ADD	0SEIS	SI ES MAYOR O IGUAL SUMALE LA COMPENSACION
00089A	1E12 A4 0F	A		AND	0CEREF	SI HAY MEDIO ACARREO BORRALO
00090A	1E14 E7 2F	A	ALMAC	STA	COFIA.X	ALMACENA EL RESULTADO DE X2
00091A	1E16 25 03	1E1FE		ECS	INVER	INVIERTE EL VALOR DEL ACARREO
00092A	1E1E 9F			SEC		
00093A	1E19 20 01	1E1FC		EKA	VUELTA	
00094A	1E11 9E		INVER	CLC		
00095A	1E1D 5A		VUELTA	DECX		DECREMENTA EL CONTADOR DE CIFRAS
00096A	1E1D 26 E9	1E1EB		ENE	DTCIF	SI NO TERMINO SIGUELE
00097A	1E1F B1			RTS		AL TERMINAR RETORNA DE LA LLAMADA
00098						

```

00900 *****
00901 >
00902 >
00903 > PROGRAMA CON LOS MACROS
00904 > PARA GENERAR LA TABLA
00905 > DE ESTADOS
00906 >
00907 *****
00908 > DEFINICION DE MACROS
00909 >
00910 >
00911 >
00912 > DEFINICION DE VARIABLES
00913 >
00914 0091 A UNO EQU 191
00915 0092 A CERO EQU 192
00916 0093 A MARCA EQU 193
00917 0094 A ENERO EQU 194
00918 009F A SIGA EQU 197
00919 >
00920 *****
00921 >
00922 > MACRO ESTADO
00923 >
00924 *****
00925 0000 A FINAL EQU 100
00926 >
00927 ESTADO MACR
00928 IFGT NARG-1
00929 FAIL >DEMASIADOS ARGUMENTOS>
00930 ENDC
00931 IFER NARG-1
00932 JFC \0+FINAL
00933 FCE \0
00934 ENDC
00935 IFNC \0+FINAL
00936 FCE \0+OCTA
00937 ENDC
00938 ENDC
00939 ENDM FIN DEL MACRO DE ESTADO
00940 >
00941 *****
00942 >
00943 > MACRO CONDIC
00944 >
00945 *****
00946 0080 A OCTA EQU 180 IGUALA CANTIDAD QUE SIGNIFICA ULTIMA ACCION.
00947 >
00948 CONDIC MACR INICIA LA MACRO CONDICION
00949 IFLT NARG-3 VERIFICA QUE NO FALTEN ARGUMENTOS
    
```

```

00950 FAIL »FALTAN ARGUMENTOS»
00951 ENDC FIN DE LA PRUEBA DE ESCASES
00952 IFGT NARG-10 VERIFICA QUE NO SOBREN ARGUMENTOS
00953 FAIL »SOBRAN ARGUMENTOS»
00954 ENDC FIN DE LA PRUEBA DE EXCESO
00955 IFGE NARG-3 SI ES MAYOR DE 3 Y
00956 IFLE NARG-10 ES MENOR DE 10 ; ADELANTE
00957 FCE \0 FON LA CONDICION.
00958 IFGT NARG-3 SI SON MAS DE 1 ACCION
00959 FCE \1-SUERUT FON LA PRIMERA.ACCION.
00960 IFGT NARG-4 SI SON MAS DE 2 ACCIONES
00961 FCE \2-SUERUT FON LA SEGUNDA.
00962 IFGT NARG-5 SI SON MAS DE 3 ACCIONES
00963 FCE \3-SUERUT FON LA TERCERA.
00964 IFGT NARG-6 SI SON MAS DE 4 ACCIONES
00965 FCE \4-SUERUT FON LA CUARTA.
00966 IFGT NARG-7 SI SON MAS DE 5 ACCIONES
00967 FCE \5-SUERUT FON LA QUINTA.
00968 IFGT NARG-8 SI SON MAS DE 6 ACCIONES
00969 FCE \6-SUERUT FON LA SEXTA.
00970 IFGT NARG-9 SI SON MAS DE 7 ACCIONES
00971 FCE \7-SUERUT FON LA SEPTIMA Y
00972 FCE \8+OCTA-SUERUT LA OCTAVA COMO ULTIMA.
00973 FDE \9 FON EL SIGUIENTE ESTADO.
00974 ENDC FIN DE LA CONDICION DE 8 ACCIONES.
00975 IFEQ NARG-9
00976 FCE \7+OCTA-SUERUT FON LA SEPTIMA COMO ULTIMA.
00977 FDE \8 FON EL SIGUIENTE ESTADO.
00978 ENDC
00979 ENDC FIN DE LA CONDICION DE 7 ACCIONES.
00980 IFEQ NARG-8
00981 FCE \6+OCTA-SUERUT FON LA SEXTA COMO ULTIMA.
00982 FDE \7 FON EL SIGUIENTE ESTADO.
00983 ENDC
00984 ENDC FIN DE LA CONDICION DE 6 ACCIONES.
00985 IFEQ NARG-7
00986 FCE \5+OCTA-SUERUT FON LA QUINTA COMO ULTIMA.
00987 FDE \6 FON EL SIGUIENTE ESTADO.
00988 ENDC
00989 ENDC FIN DE LA CONDICION DE 5 ACCIONES.
00990 IFEQ NARG-6
00991 FCE \4+OCTA-SUERUT FON LA CUARTA COMO ULTIMA.
00992 FDE \5 FON EL SIGUIENTE ESTADO.
00993 ENDC
00994 ENDC FIN DE LA CONDICION DE 4 ACCIONES.
00995 IFEQ NARG-5
00996 FCE \3+OCTA-SUERUT FON LA TERCERA COMO ULTIMA.
00997 FDE \4 FON EL SIGUIENTE ESTADO.
00998 ENDC
00999 ENDC FIN DE LA CONDICION DE 3 ACCIONES.

```

PAGE 027 TELIS .AL:1

---

PAGE 022 FINAL VIR. SA:1

RELOJ SINCRONIZADO CON LA WWVF VER. 1K

01000	IFEQ	NARG-4
01001	FDE	\2+OCTA-SUERUT FON LA SEGUNDA COMO ULTIMA.
01002	FDE	\3 FON EL SIGUIENTE ESTADO.
01003	ENDC	
01004	ENDC	FIN DE LA CONDICION DE 2 ACCIONES.
01005	IFEQ	NARG-3
01006	FDE	\1+OCTA-SUERUT FON LA UNICA ACCION.
01007	FDE	\2 FON EL SIGUIENTE ESTADO.
01008	ENDC	FIN DE LA CONDICION CON UNA SOLA ACCION.
01009	ENDC	FIN DE <10.
01010	ENDC	FIN DE >3.
01011	ENDM	FIN DE LA MACRO DE CONDICION.
01012		

```

01014
01015
01016
01017
01018
01019
01020
01021A 1D10
01022 0001 A DONAS ERU 01
01023 0002 A TRIFAS ERU 02
01024 0003 A CUAJOS ERU 03
01025
01026
01027A 1D10
01028A 1D10
01029A 1D14
01030A 1D18
01031A 1D19
01032A 1D1A
01033A 1D1F
01034A 1D23
01035A 1D24
01036A 1D24
01037A 1D28
01038A 1D2C
01039A 1D21
01040A 1D32
01041A 1D33
01042A 1D37
01043A 1D3B
01044A 1D3F
01045A 1D40
01046A 1D40
01047A 1D44
01048A 1D48
01049A 1D4D
01050A 1D4E
01051A 1D4F
01052A 1D53
01053A 1D57
01054A 1D58
01055A 1D59
01056A 1D5D
01057A 1D61
01058A 1D65
01059A 1D69
01060A 1D6A
01061A 1D6A
01062A 1D6E
01063A 1D72

```

PROGRAMA EN LENGUAJE DE ESTADOS  
PARA EFECTUAR EL RECONOCIMIENTO DE  
LA SENAL DE LA HHV.

```

ORG 1D10
CONDIC MARCA,ERROR,EDDA
CONDIC ENED,ERROR,EDDA
ESTADO FINAL
ESTADO TRIFAS
CONDIC MARCA,ERROR,CARGAN,EDDC
CONDIC ENED,ERROR,EDDA
ESTADO FINAL
ESTADO
CONDIC MARCA,ERROR,EDDI
CONDIC ENED,CARDIG,EDDD
CONDIC SIGA,CARDIG,DECENE,EDDC
ESTADO FINAL
ESTADO CUAJOS
CONDIC MARCA,ERROR,EDDE
CONDIC ENED,CARDIG,EDDF
CONDIC SIGA,CARDIG,DECENE,EDDE
ESTADO FINAL
ESTADO
CONDIC MARCA,ERROR,EDDE
CONDIC ENED,CARDIG,EDDF
CONDIC SIGA,CARDIG,DECENE,EDDE
ESTADO FINAL
ESTADO DONAS
CONDIC MARCA,CARGAN,EDDG
CONDIC SIGA,ERROR,EDDA
ESTADO FINAL
ESTADO DONAS
CONDIC MARCA,ERROR,EDDE
CONDIC UNO,ERROR,EDDA
CONDIC ENED,CARGAN,EDDI
CONDIC CER0,DECENE,EDDG
ESTADO FINAL
ESTADO
CONDIC MARCA,ERROR,EDDI
CONDIC ENED,CARDIG,EDDJ
CONDIC SIGA,CARDIG,DECENE,EDDI

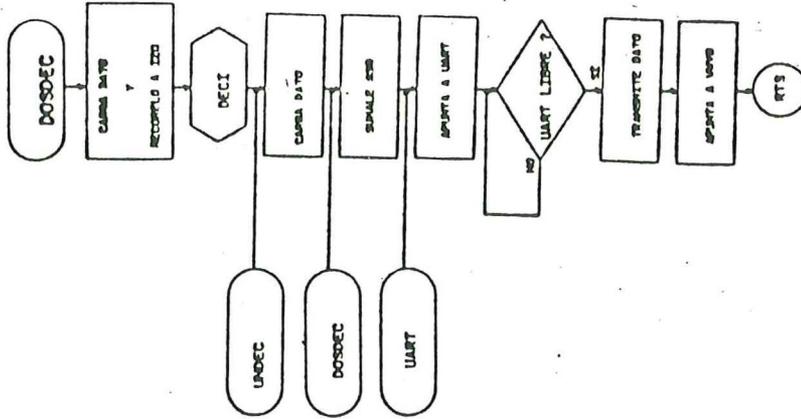
```

01064A 1D77		ESTADO FINAL
01065A 1D78	EDDJ	ESTADO CUAJOS
01066A 1D79		CONDIC CERD,CARGAN,EDDK
01067A 1D7C		CONDIC MARCA,ERROR,EDDL
01068A 1D81		CONDIC SIGA,ERROR,EDDA
01069A 1D85		ESTADO FINAL
01070A 1D86	EDDK	ESTADO
01071A 1D86		CONDIC MARCA,ERROR,EDDI
01072A 1D8A		CONDIC ENED,CARDIG,EDUL
01073A 1D8F		CONDIC SIGA,CARDIG,DECENE,EDDK
01074A 1D93		ESTADO FINAL
01075A 1D94	EDDL	ESTADO DONAS
01076A 1D95		CONDIC MARCA,CARGAN,EDDM
01077A 1D98		CONDIC SIGA,ERROR,EDDA
01078A 1D9D		ESTADO FINAL
01079A 1D9E	EDDM	ESTADO DONAS
01080A 1D9F		CONDIC MARCA,ERROR,EDDE
01081A 1DA5		CONDIC UNO,ERROR,EDDA
01082A 1DA7		CONDIC ENED,CARGAN,EDDD
01083A 1DAF		CONDIC CERD,DECENE,EDDM
01084A 1DAF		ESTADO FINAL
01085A 1DB0	EDDD	ESTADO
01086A 1DB0		CONDIC MARCA,ERROR,EDDE
01087A 1DB4		CONDIC ENED,CARDIG,EDDF
01088A 1DB8		CONDIC SIGA,CARDIG,DECENE,EDDD
01089A 1DBD		ESTADO FINAL
01090A 1DBE	EDDF	ESTADO CUAJOS
01091A 1DBF		CONDIC CERD,CARGAN,EDDD
01092A 1DC0		CONDIC MARCA,ERROR,EDDI
01093A 1DC7		CONDIC UNO,ERROR,EDDA
01094A 1DCI		ESTADO FINAL
01095A 1DCC	EDDR	ESTADO
01096A 1DCC		CONDIC MARCA,ERROR,EDDE
01097A 1DD0		CONDIC ENED,CARDIG,EDDR
01098A 1DD4		CONDIC SIGA,CARDIG,DECENE,EDDR
01099A 1DD9		ESTADO FINAL
01100A 1DDA	EDDR	ESTADO CUAJOS
01101A 1DDE		CONDIC MARCA,CARGAN,EDDS
01102A 1DDF		CONDIC SIGA,ERROR,EDDA
01103A 1DE3		ESTADO FINAL
01104A 1DE4	EDDS	ESTADO DONAS
01105A 1DE5		CONDIC MARCA,ERROR,EDDE
01106A 1DE9		CONDIC ENED,CARDIG,CARGAN,EDDT
01107A 1DEE		CONDIC SIGA,CARDIG,DECENE,EDDS
01108A 1DF3		ESTADO FINAL
01109A 1DF4	EDDT	ESTADO TRIFAS
01110A 1DF5		CONDIC MARCA,ERROR,EDDE
01111A 1DF9		CONDIC UNO,ERROR,EDDA
01112A 1DFD		CONDIC ENED,CARGAN,EDDV
01113A 1E01		CONDIC CERD,DECENE,SEG35,EDDT

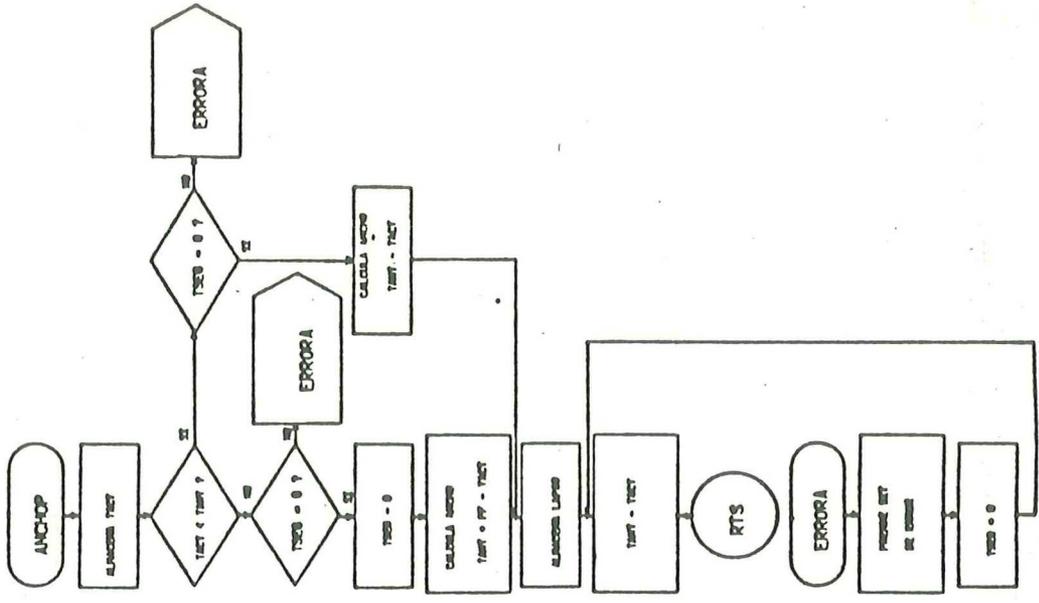
011144 1E06		ESTADO FINAL
011154 1E07	EDDV	ESTADO
011164 1E07		CONDIC MARCA,ERROR,EDD!
011174 1E08		CONDIC ENLD,CARDIG,EDDV
011184 1E0F		CONDIC SIGA,CARDIG,DECENE,EDDV
011194 1E14		ESTADO FINAL
011204 1E15	EDDW	ESTADO CUAJOS
011214 1E16		CONDIC MARCA,CARGAN,EDDX
011224 1E1A		CONDIC SIGA,ERROR,EDDA
011234 1E1E		ESTADO FINAL
011244 1E1F	EDDX	ESTADO
011254 1E1F		CONDIC MARCA,ERROR,EDD!
011264 1E23		CONDIC ENLD,CARDIG,EDDY
011274 1E27		CONDIC SIGA,CARDIG,DECENE,EDDX
011284 1E2C		ESTADO FINAL
011294 1E2D	EDDY	ESTADO CUAJOS
011304 1E2E		CONDIC CERD,CARGAN,EDDZ
011314 1E32		CONDIC MARCA,ERROR,EDDE
011324 1E36		CONDIC UNO,ERROR,EDDA
011334 1E3A		ESTADO FINAL
011344 1E3F	EDDZ	ESTADO
011354 1E3E		CONDIC MARCA,ERROR,EDD!
011364 1E3F		CONDIC ENLD,CARDIG,EDDAM
011374 1E43		CONDIC SIGA,CARDIG,DECENE,EDDZ
011384 1E48		ESTADO FINAL
011394 1E49	EDDAM	ESTADO CUAJOS
011404 1E4A		CONDIC MARCA,CARGAN,EDDEM
011414 1E4E		CONDIC SIGA,ERROR,EDDA
011424 1E52		ESTADO FINAL
011434 1E53	EDDEM	ESTADO DONAS
011444 1E54		CONDIC MARCA,ERROR,EDDE
011454 1E5E		CONDIC ENLD,CARDIG,PREFAR,CARGAN,EDDCH
011464 1E5E		CONDIC SIGA,CARDIG,DECENE,EDDEM
011474 1E63		ESTADO FINAL
011484 1E64	EDDCH	ESTADO
011494 1E64		CONDIC ENLD,NADA,EDDA
011504 1E68		CONDIC SIGA,DECENE,EDDCH
011514 1E6C		ESTADO FINAL
01152	»	
01153	»	FINALE
01154	»	





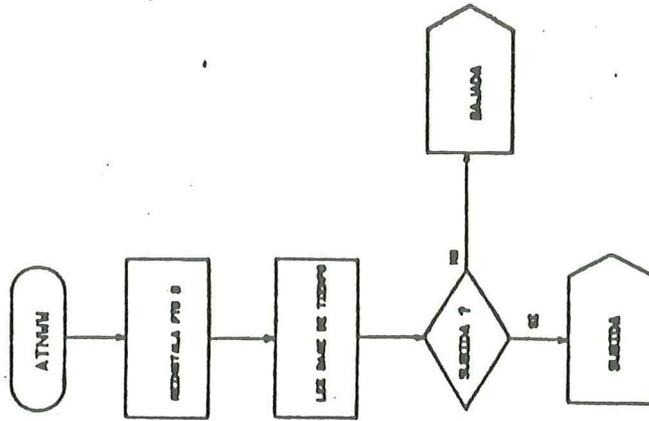


Rutina DOSDEC

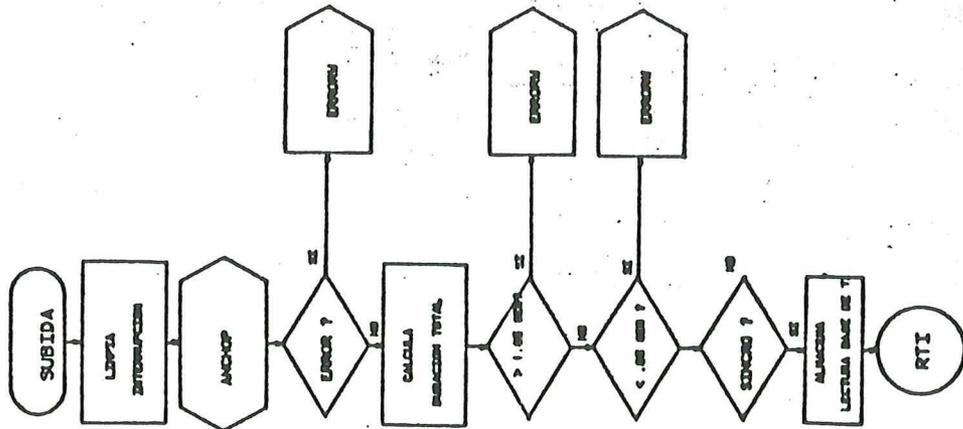


Rutina ANCHOP

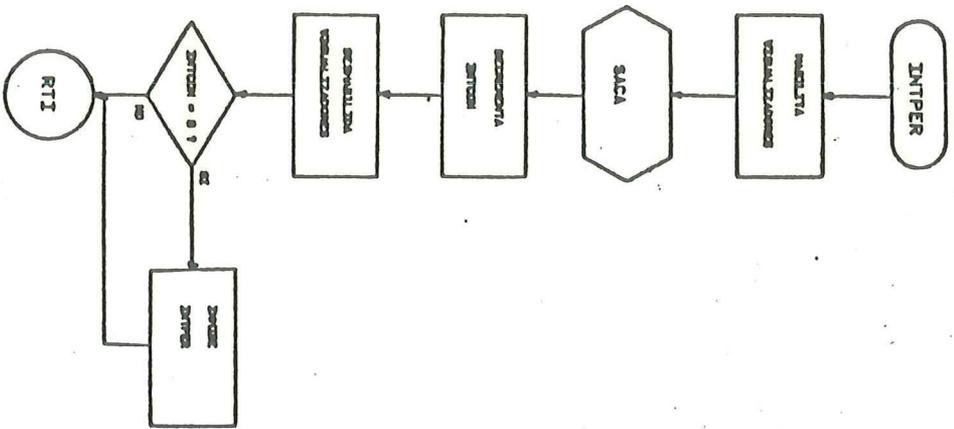




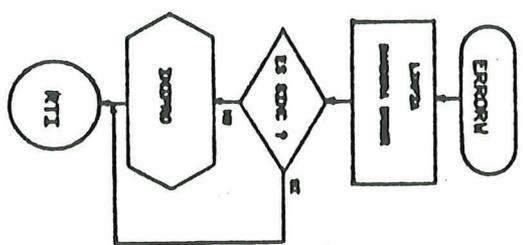
Rutina ATNHW



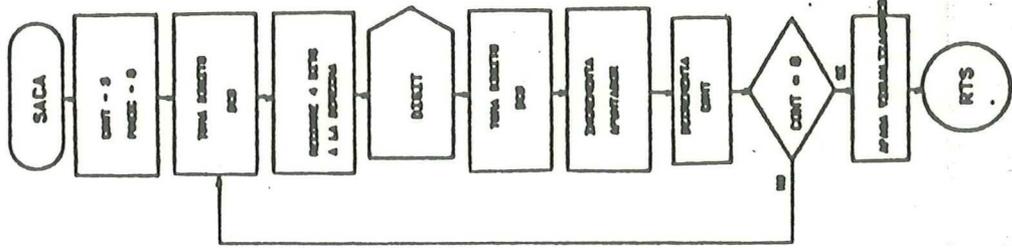
Rutina SUBIDA



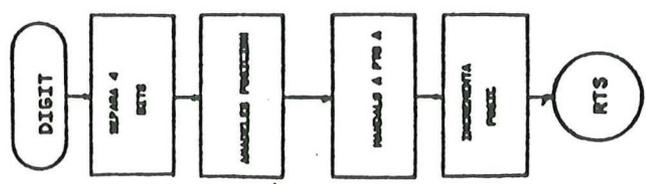
Rutina INTPER



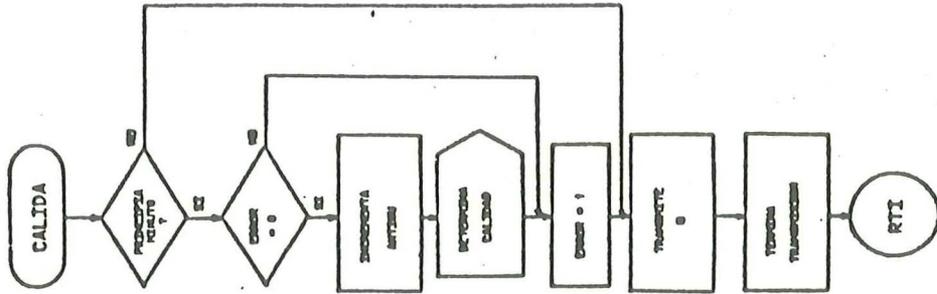
Rutina ERRORR



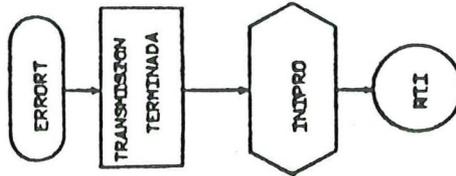
Rutina SACA



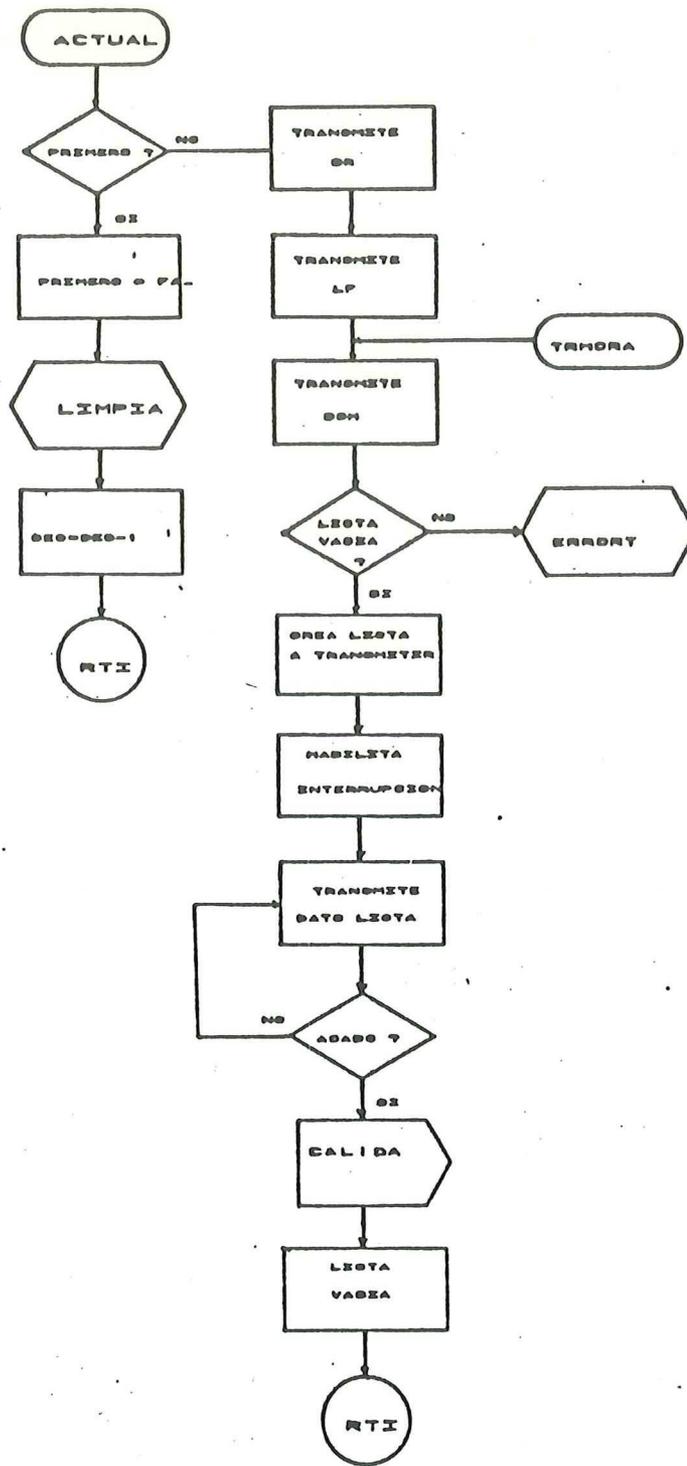
Rutina DIGIT



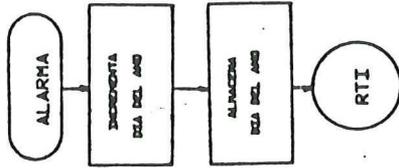
Rutina CALIDA



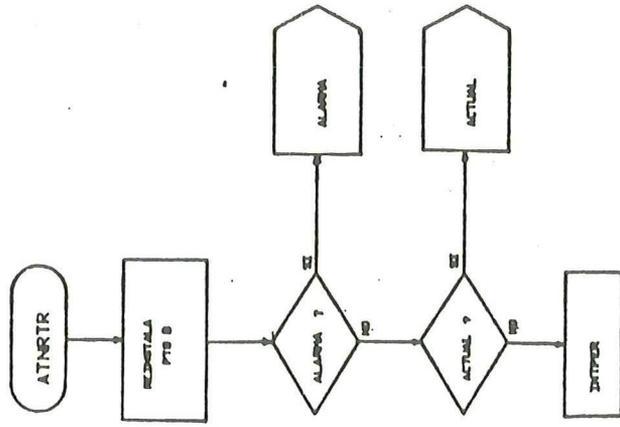
Rutina ERROR



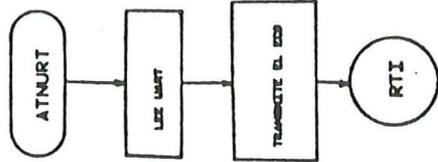
Rutina ACTUAL



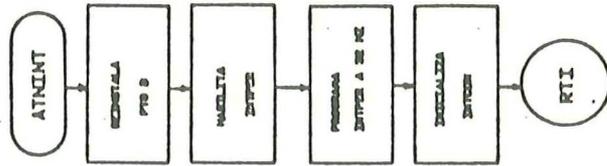
Rutina ALARMA



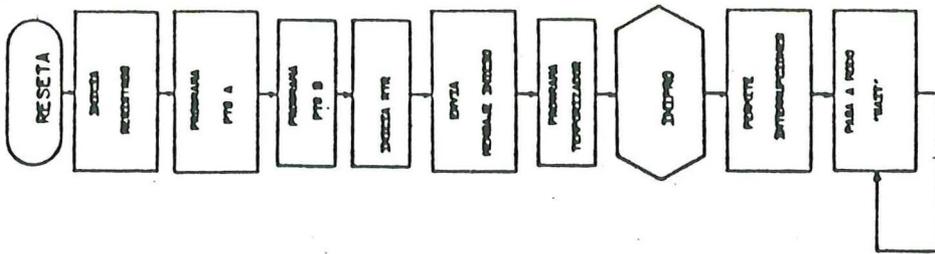
Rutina ATNRTR



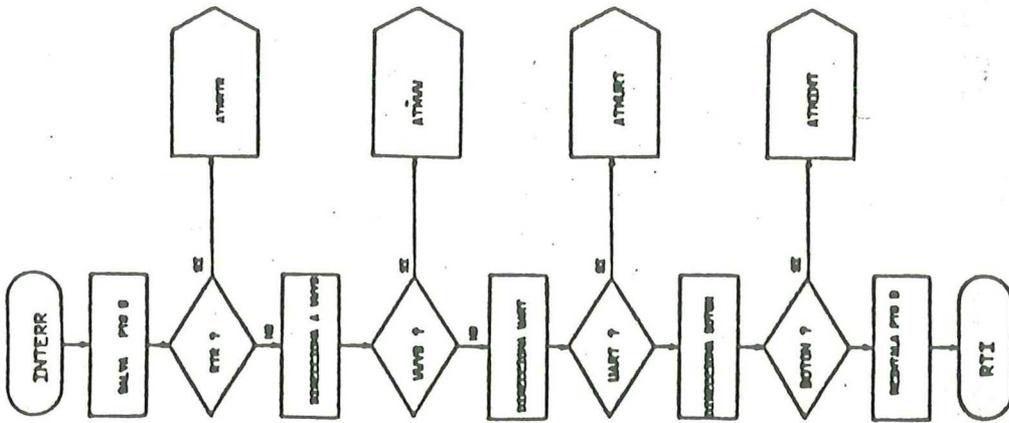
Rutina ATNURT



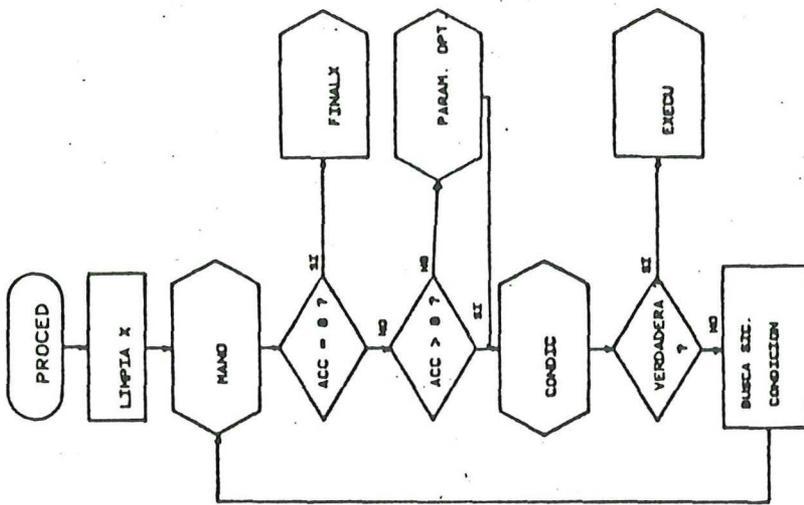
Rutina ATNINT



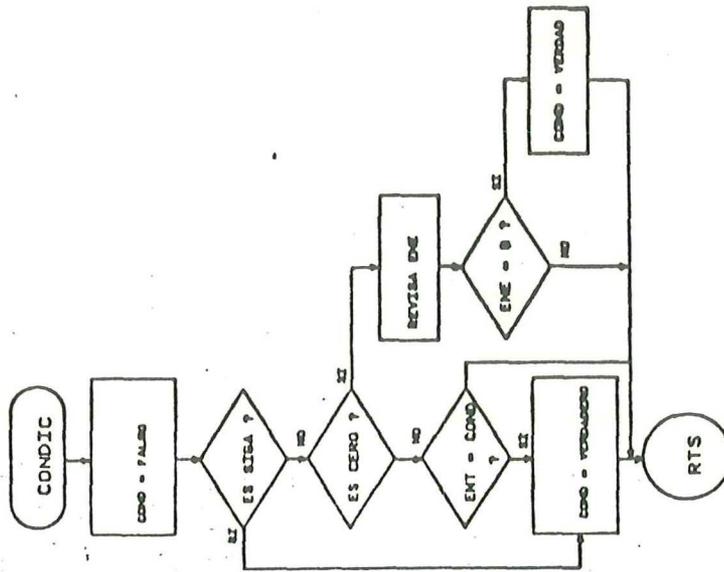
Rutina RESETA



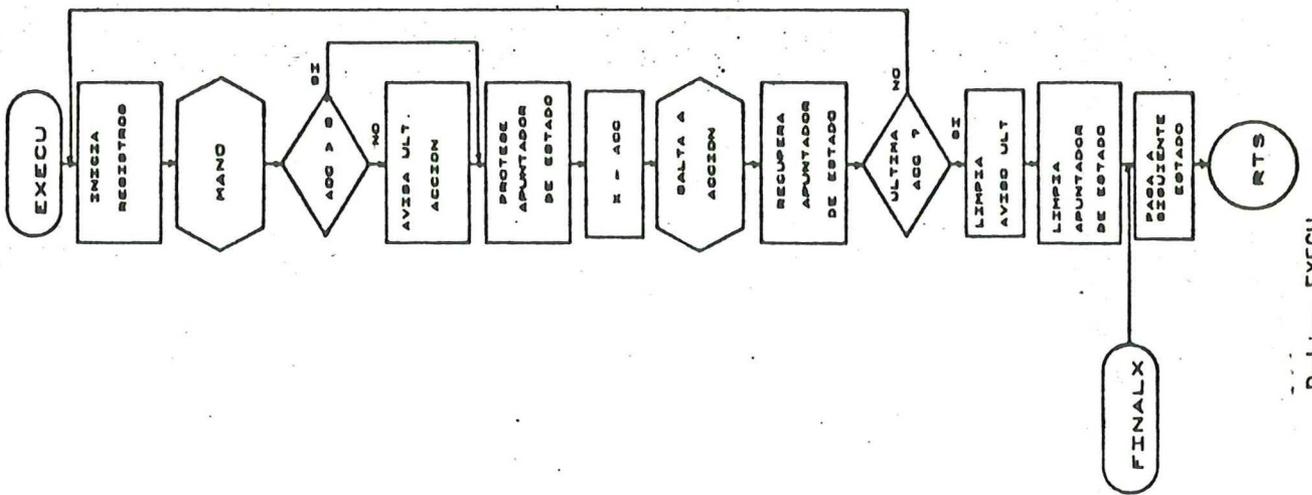
Rutina INTERR



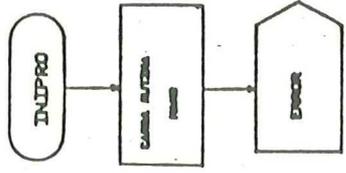
Rutina PROCED



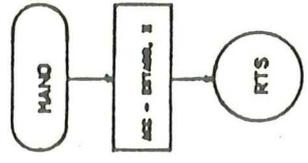
Rutina CONDIC



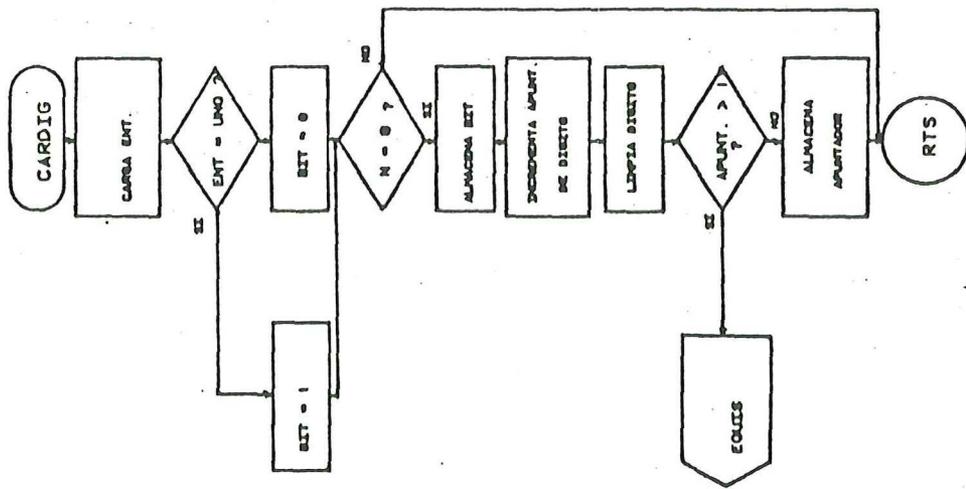
Rutina EXECU



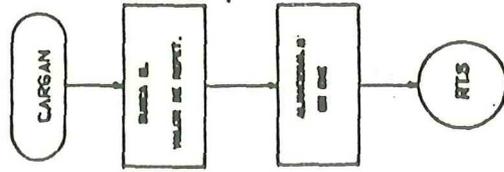
Rutina INIPRO



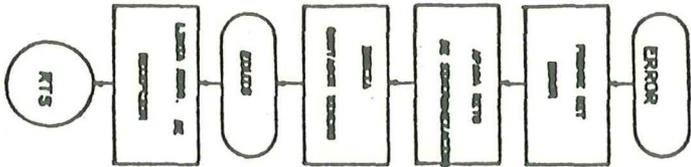
Rutina MANO



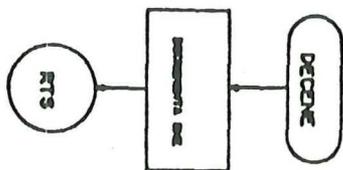
Rutina CARDIG



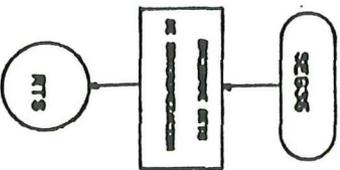
Rutina CARGAN



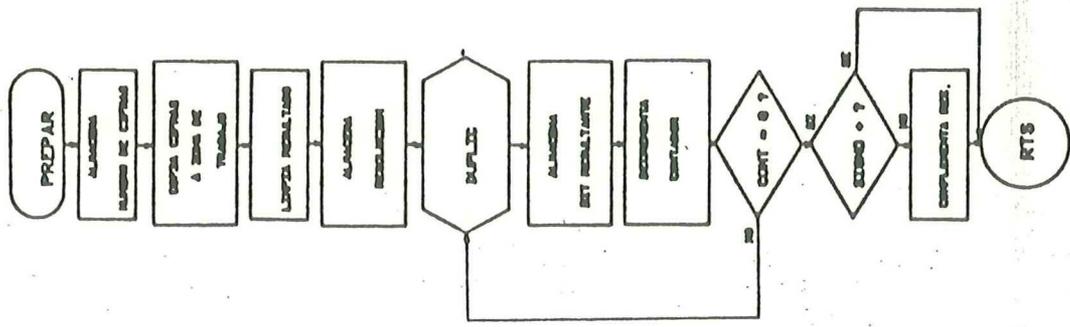
Rutina ERROR



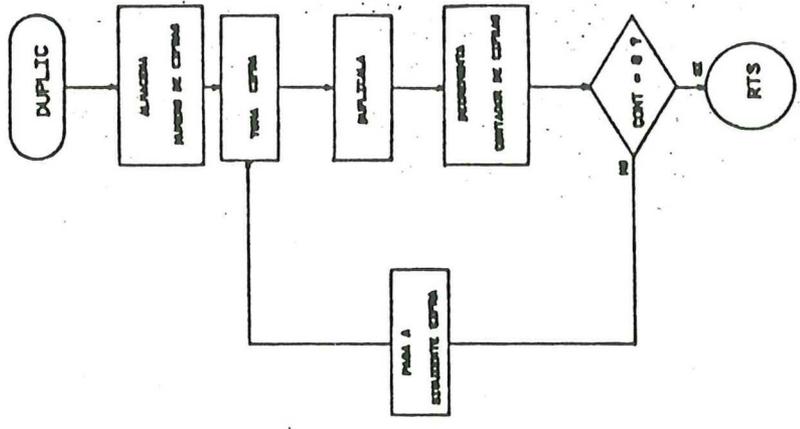
Rutina DECENE



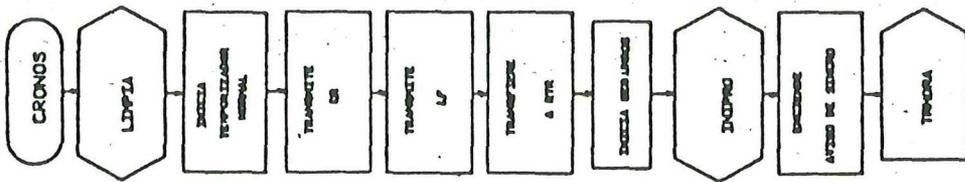
Rutina SEG3S



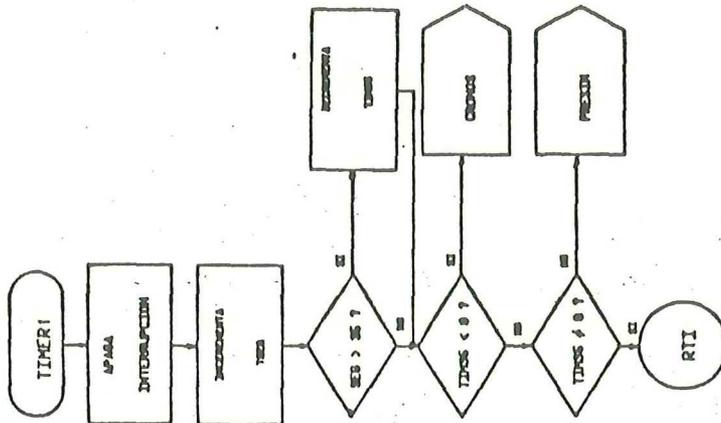
Rutina PREPAR



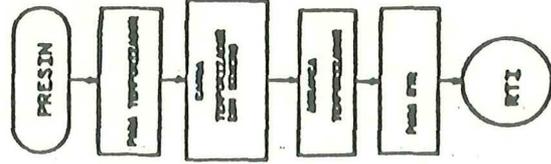
Rutina DUPLIC



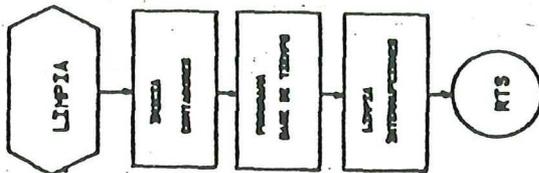
Rutina CRONOS



Rutina TIMER



Rutina PRESIN



Rutina LIMPIA