

**UNIVERSIDAD NACIONAL DE INGENIERIA**

**Facultad de Ingeniería Eléctrica y  
Electrónica**



**Diseño del Circuito de Control basado en  
Microcontroladores Intel MCS-51 para el equipo  
Telmet del Sistema de Servicio Medido de  
las Centrales Telefónicas Pentaconta**

**TESIS**

**Para Optar el Título Profesional de**

***INGENIERO ELECTRONICO***

**LUIS ALBERTO URA ENRIQUEZ**

**Promoción 1989 - II**

**Lima - Perú  
1994**

## SUMARIO

El presente proyecto parte como necesidad de implementar una tasación flexible ofrecida por CPTSA, ésta solicitó se desarrollara un sistema que permita medir el tiempo de uso del servicio por los abonados, y tasarlo en función de la hora y tipo de día en que ocurre el evento.

Evaluado el estado inicial del "Sistema de Servicio Medido de CPTSA, se adoptó como solución modernizar el sistema, en particular el equipo tasador **Telmet**, así el presente trabajo tiene como objetivo el desarrollo de un circuito que repotencie las facilidades del equipo Telmet. Técnicamente, el circuito a diseñar es la unidad de control central del Telmet, y es desarrollado sobre la base de microcontroladores en relación maestro-esclavo, cada uno procesando un software particularizado: el controlador esclavo interactuando con el Telmet y procesando el tiempo de uso de servicio **por** cada abonado, y el maestro manteniendo la cuenta total de cada abonado, administrando el nuevo plan tarifario, comunicaciones y otras tareas.

Este enfoque permitió un fácil desarrollo e implementación del diseño, éste cubre los objetivos del proyecto y viene operando en el campo en forma satisfactoria.

DISEÑO DEL CIRCUITO DE CONTROL BASADO EN  
MICROCONTROLADORES INTEL MCS-51 PARA EL EQUIPO  
TELMET DEL SISTEMA DE SERVICIO MEDIDO DE  
LAS CENTRALES TELEFONICAS PENTACONTA

"DISEÑO DEL CIRCUITO DE CONTROL BASADO EN  
MICROCONTROLADORES INTEL MCS-51 PARA EL EQUIPO TELMET DEL  
SISTEMA DE SERVICIO MEDIDO DE LAS CENTRALES TELEFONICAS  
PENTACONTA"

AUTOR : LUIS ALBERTO URA ENRIQUEZ

GRADO AL QUE OPTA : INGENIERO ELECTRONICO

ASESOR : ING. AURELIO MORALES

UNIVERSIDAD NACIONAL DE INGENIERIA  
FACULTAD DE INGENIERIA ELECTRICA Y ELECTRONICA  
LIMA - 1994

XTRACTO

La medición del uso de línea por abonado es una necesidad fundamental en las redes conmutadas que prestan servicio público de telefonía. En nuestro medio, buena parte de las centrales de conmutación está atendida por equipos que no ofrecen una tasación flexible. Este trabajo señala como solución la modernización de éstos equipos, y desarrolla el diseño de un módulo circuital para repotenciarlos.

Este módulo es un pequeño sistema basado en microcontroladores que facilita y hace más sencilla su implementación en hardware y permite un amplio control al ser gobernado por software.

Para su desarrollo, el presente trabajo se ha dividido en tres partes:

1. Se hace una descripción sencilla del equipo Telmet, equipo a repotenciar, sus antecedentes y principio de funcionamiento. Así mismo, se ha planteado la solución en microcontroladores, detallándose los recursos que usaremos de ellos.
2. Sobre las condiciones de interacción con el resto del equipo, tiempos de observación y magnitud de tareas, se reparte la carga de trabajo en dos microcontroladores, uno esclavo del otro. El microcontrolador esclavo interactúa directa-

mente con el Telnet mientras el microcontrolador maestro maneja la cuenta de los abonados, interactúa con otros periféricos, y el exterior por su puerto serial.

3. Se presenta un esquema de tarificación de "tasación flexible", desarrollándose el software para que ambos microcontroladores se complementen en la ejecución del plan tarifario. Además se desarrolla el protocolo de comunicaciones, etc.

Finalmente, en los anexos se incluye los esquemas eléctricos, listado de programas y hojas técnicas de los circuitos integrados usados.

## TABLA DE CONTENIDO

	PAG
INTRODUCCION	1
CAPITULO I ASPECTOS GENERALES	3
1.1 El Sistema de servicio medido	3
1.1.1 Antecedentes	3
1.1.2 Estado actual	4
1.1.3 Principio de funcionamiento	5
1.2 El sistema de servicio medido a diseñar	5
1.2.1 El equipo Telmet modernizado	6
1.3 Los microcontroladores	8
1.3.1 La familia Intel MCS-51	10
1.3.2 Selección de los microcontroladores 8032AH y 80C452	11
1.4 Descripción de los microcontroladores Intel 8032AH y 80C452	12
1.4.1 Arquitectura del 8032 y 80C452	13
1.4.2 Organización de memoria de programa y datos	14
1.4.3 Instrucciones de la familia MCS-51	19
1.4.4 Temporización de la CPU	22
1.4.5 Estructura de interrupciones	24
1.5 Transmisión de datos	28
1.5.1 Modos de transmisión	28
1.5.2 Interfaz físico de comunicación de datos	30
1.5.3 Protocolos de comunicación	30
CAPITULO II DISEÑO DE HARDWARE	32
2.1 Condiciones de temporización	32
2.1.1 Señales de control ISCA', HASCA e INSCA	32
2.2 Diagrama de bloques de la unidad de control	34
2.3 Módulo del controlador maestro	37
2.3.1 Unidad central de procesamiento maestro µC 8032	37

2.3.2	Memoria de programa	42
2.3.3	Memoria de datos de abonados RAM	42
2.3.4	Circuito de corrección y detección de errores	44
2.3.5	Reloj de tiempo real RTC DS1285	45
2.3.6	Circuito supervisor DS1236	46
2.3.7	Batería de respaldo	47
2.3.8	Interfaz de comunicación serial	48
2.4	Módulo del controlador esclavo	49
2.4.1	Unidad de procesamiento periférico $\mu$ C 80C452	49
2.4.2	Memoria de programa del esclavo	53
2.4.3	Memoria de datos temporales externa RAM	53
2.4.4	Generador de señales de sincronismo	54
2.4.5	Circuito de redundancia y multiplexaje	55
CAPITULO III DISEÑO DE SOFTWARE		58
3.1	Estructura del plan de tarifas	59
3.2	Descripción general de funciones	62
3.2.1	Funciones del procesador maestro 8032	62
3.2.2	Funciones del procesador esclavo 80C452	63
3.3	Módulos del sistema - diagramas de flujo	64
3.3.1	Programa supervisor e interrupciones del $\mu$ C maestro	66
3.3.2	Programa supervisor e interrupciones del $\mu$ C esclavo	69
3.4	Configuración de entorno e inicialización	71
3.4.1	Distribución de memoria RAM para $\mu$ C maestro	71
3.4.2	Maestro: inicialización y tareas en el programa supervisor	75
3.4.3	Distribución de memoria para el $\mu$ C esclavo	80
3.4.4	Esclavo: inicialización y entrada al programa supervisor	82
3.5	Protocolo de comunicación entre procesadores	84
3.5.1	Comunicación del maestro al esclavo	84
3.5.2	Comunicación del esclavo al maestro	86
3.6	Protocolo de comunicación serial	89
3.6.1	Formato de comunicación	89
3.6.2	Estructura de mensajes	90

3.6.3 Código de los mensajes	91
3.6.4 Criterio para calcular el CRC	95
3.6.5 Criterio para el protocolo de comunicación	95
3.7 Interrupción del reloj RTC y del detector corrector de errores DCU	99
3.7.1 Interrupción de DCU por error en RAM externa	99
3.7.2 Interrupción del reloj RTC DS1285	99
3.8 Ejecución de plan tarifario	101
3.8.1 Acumulación de unidades de costo por el maestro	101
3.8.2 Cuenta de tiempo de conversación por el esclavo	103
CONCLUSIONES	110
ANEXOS:	112
A. TELMET: OPERACION DEL SUBSISTEMA INTERFAZ DE LINEA	113
B. DIAGRAMAS CIRCUITALES DE LA UNIDAD DE CONTROL	120
C. PROGRAMAS DE LOS MICROCONTROLADORES MAESTRO Y ESCLAVO DE LA UNIDAD DE CONTROL	123
D. REFERENCIA DE INSTRUCCIONES MCS-51	186
E. CARACTERISTICAS TECNICAS DE COMPONENTES	191
F. LISTADO DE COMPONENTES Y DISPOSICION	282
BIBLIOGRAFIA	286

## INTRODUCCION

El presente trabajo consiste en el diseño de un circuito que repotencia al equipo tasador Telmet de consumo de servicio de las centrales telefónicas Pentaconta, ofreciéndole nuevas y mayores facilidades de tasación y operación.

El diseño de éste módulo -circuito Unidad de Control- se basa en microcontroladores, los que permiten un fácil y confiable desarrollo del hardware por los variados y generosos recursos que ofrecen, junto con su facilidad de manejar por software los requerimientos que implica la modernización del Telmet.

Modernizar el sistema de servicio medido para las centrales telefónicas Pentaconta significa hacer que los antiguos equipos de tasación (con sólo dos planes de tarifa, uno el doble del otro, y consumo redondeado a múltiplos de tres minutos) sean capaces de registrar el consumo del servicio de los abonados que efectúan una llamada -cada central Pentaconta atiende más de 10,000 abonados- con planes de tasación versátiles y flexibles en función de la hora del día (horas pico, media tarde, noche, etc.) y en función del día de la semana (se puede diferenciar planes de tarifa para un día laborable, un sábado, un domingo) ó de un día feriado.

La modernización del sistema de servicio medido involucra:

- a) Ampliar el acceso a los Telmet desde dos estaciones: una estación remota que centralice el acceso a todos los Telmet y una estación local situada en la misma oficina donde se encuentre un Telmet o grupo de ellos.

- b) Acondicionar la captación de una señal -proveniente de la central telefónica- que se mantenga en condición estable durante el tiempo de conversación.
- c) Mantener invariable la generación de señales de reloj y sincronismo para el resto de los circuitos/unidades que componen el Telmet.
- d) Diseñar un módulo unidad de control altamente confiable en hardware y software, capaz de leer e interpretar las señales que ofrece el Telmet, capaz de manejar planes de tarifa programables, y que pueda ser accesible por comunicación serial con cualquier computador IBM PC compatible.
- e) Diseñar un protocolo de comunicación serial que permita el enlace desde dos computadores (estaciones de interacción hombre-máquina) varias unidades de control.
- f) Diseñar un circuito conmutador de canales de comunicación serial, una unidad de interfaz entre computadores y unidades de control.

El sistema completo abarca todos los puntos trabajando interrelacionados, sin embargo, el presente trabajo sólo pretende cubrir los items mencionados en los puntos "d)" y "e)". Estos serán desarrollados en detalle en los capítulos II y III. El capítulo I hace una breve introducción al sistema de servicio medido, los microcontroladores y la teoría de transmisión de datos.

## **CAPITULO I**

### **ASPECTOS GENERALES**

#### **1. El sistema de servicio medido**

Parte de la red de telefonía que posee la ciudad de Lima, está constituida por centrales de conmutación analógica, las que para tasar la duración de llamadas locales, interactúan con los equipos Telmet, los cuales se encargan de registrar el consumo del servicio de los abonados generadores de llamadas.

El Sistema de servicio medido de la CPTSA es el sistema encargado de tasar el tiempo de enlace de los abonados de centrales analógicas que originan llamadas.

#### **1. .1 Antecedentes**

En 1974 CPTSA inicia el servicio medido local con la puesta en operación de los equipos de tasación TELMET, de manufactura belga. La instalación termina dos años mas tarde y comprendía aproximadamente 240,000 abonados, en 25 oficinas centrales (13 centros de alambre). En principio se facturaba únicamente el número de llamadas efectuadas por cada abonado, pero debido a requerimientos del servicio se decidió efectuar una tasación por duración de llamadas y para tal efecto en 1981 se adquirieron los temporizadores telefónicos a una firma Española, estos equipos se instalaron durante 1982-1983.

En 1982 se puso en servicio el "Sistema Centralizado de Lectura Rápida" para los equipos TELMET, que permitió centralizar todas las operaciones de los equipos TELMET en un solo punto desde donde se podía efectuar la lectura de todos los abonados servidos por TELMETs, en un tiempo muy corto y así mismo permitía efectuar mantenimiento remoto. Este equipo es de procedencia nacional.

En 1986 la CPTSA decidió aplicar la tarifa reducida para el servicio medido local y para esto se adquirió en el mercado local el equipo necesario. La limitación es que éste equipo solo permite la emisión de dos tarifas y una el doble de la otra.

En 1990, debido a razones de repuestos para mantenimiento de los equipos TELMET, la CPTSA adquirió 25 subsistemas de memoria, para reemplazar los discos magnéticos originales que ya no se fabrican y cuyos repuestos son escasos y muy caros. Estos subsistemas son de fabricación nacional.

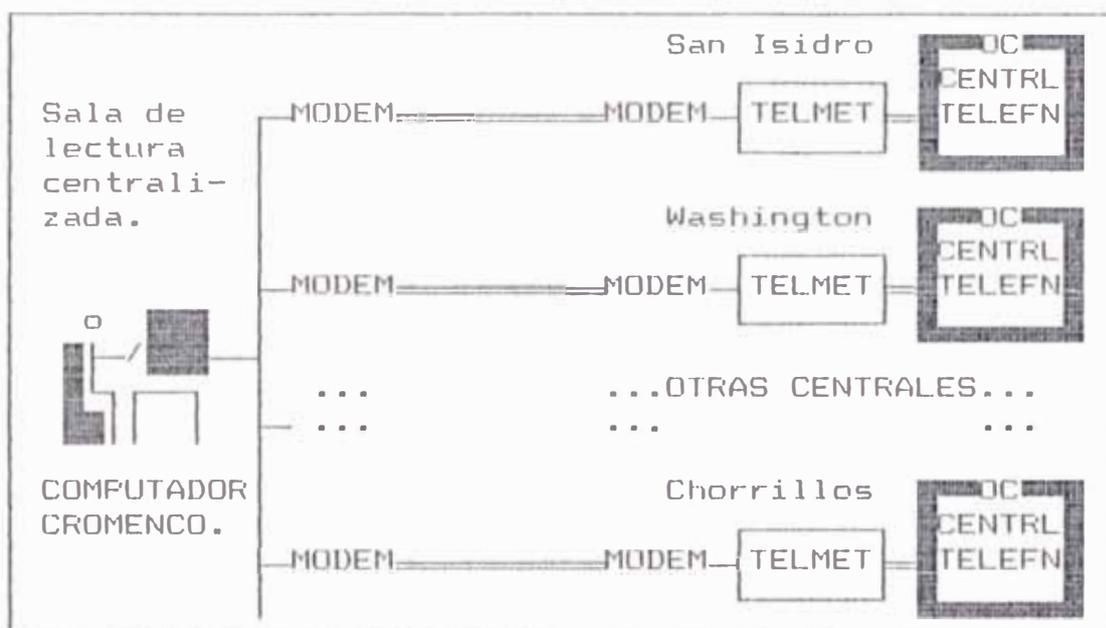


Fig. 1.1 Sistema de Servicio Medido

### 1.1.2 Estado actual

Como se ve en el actual Sistema de Servicio Medido existen hasta cuatro proveedores de subequipos "independientes". A esto se suma la incapacidad actual de los equipos referidos, para efectuar la TASACION FLEXIBLE ofrecida por la CPTSA (una unidad de llamada por los tres primeros minutos y una por cada minuto adicional, por ejemplo).

Así mismo, los equipos mencionados sirven a toda la red analógica de la CPTSA, es decir a las centrales electromecánicas en aconta (180 mil abonados) las cuales

están en la parte final de su vida útil y deben ser reemplazadas a corto ó mediano plazo.

La solución más adecuada para éste problema es la modernización de los equipos Telmet de modo que permitan la tasación flexible y la eliminación de los temporizadores ya que el mismo Telmet modernizado podrá encargarse de las diferentes tarifas y además se extenderá la vida útil de éste al incluirsele componentes de nueva tecnología.

### **1.1.3 Principio de funcionamiento**

El Sistema de servicio medido, basa la medición del tiempo de enlace/conversación, del abonado que origina la llamada, observando el el valor de tensión DC sobre el hilo "c" (también llamado hilo de cómputo de abonado) que proviene del centro de conmutación OC al equipo Telmet. Esto es, un hilo "c" por cada abonado, y un equipo Telmet por cada centro de conmutación OC. Significa que por cada Telmet se observa el estado de mas de diez mil abonados.

Asi, los equipos Telmet observan el tiempo de conversación/enlace cada tres minutos y guardan el resultado de su observación en forma acumulativa por abonado en su unidad de almacenamiento (memorias de estado sólido, antes discos duros magnéticos).

Luego, según una estrategia de procedimientos, periódicamente desde la sala de lectura se hace el acopio de datos en disco en un formato especial, le llamaremos formato CPT, y éste se entrega al centro de cómputo para su procesamiento final antes de llegar el recibo al usuario.

## **1.2 El sistema de servicio medido a diseñar**

El nuevo sistema comprende el acceso a cada Telmet desde dos estaciones de administración, una remota con un computador que centraliza el gobierno de todo el sistema, y una estación local, que con un computador puede acceder a los varios Telmet que pueden estar alojados en

el mismo local físico. La Fig. 1.2 muestra este nuevo entorno.

En el mismo diagrama podemos notar que el enlace de los Telmet con los computadores remoto ó local no es directo, aunque puede serlo. El enlace es mediante un circuito denominado Unidad de Interfaz, éste resulta ser un conmutador de canales de comunicación serial, permite el enlace de los computadores con cualesquiera de las unidades de control.

La comunicación con el computador remoto no es directa, necesita de un modem en cada extremo. La distancia que cubren son de varios kilómetros. Estos modem trabajan a 9600bps sobre línea dedicada.

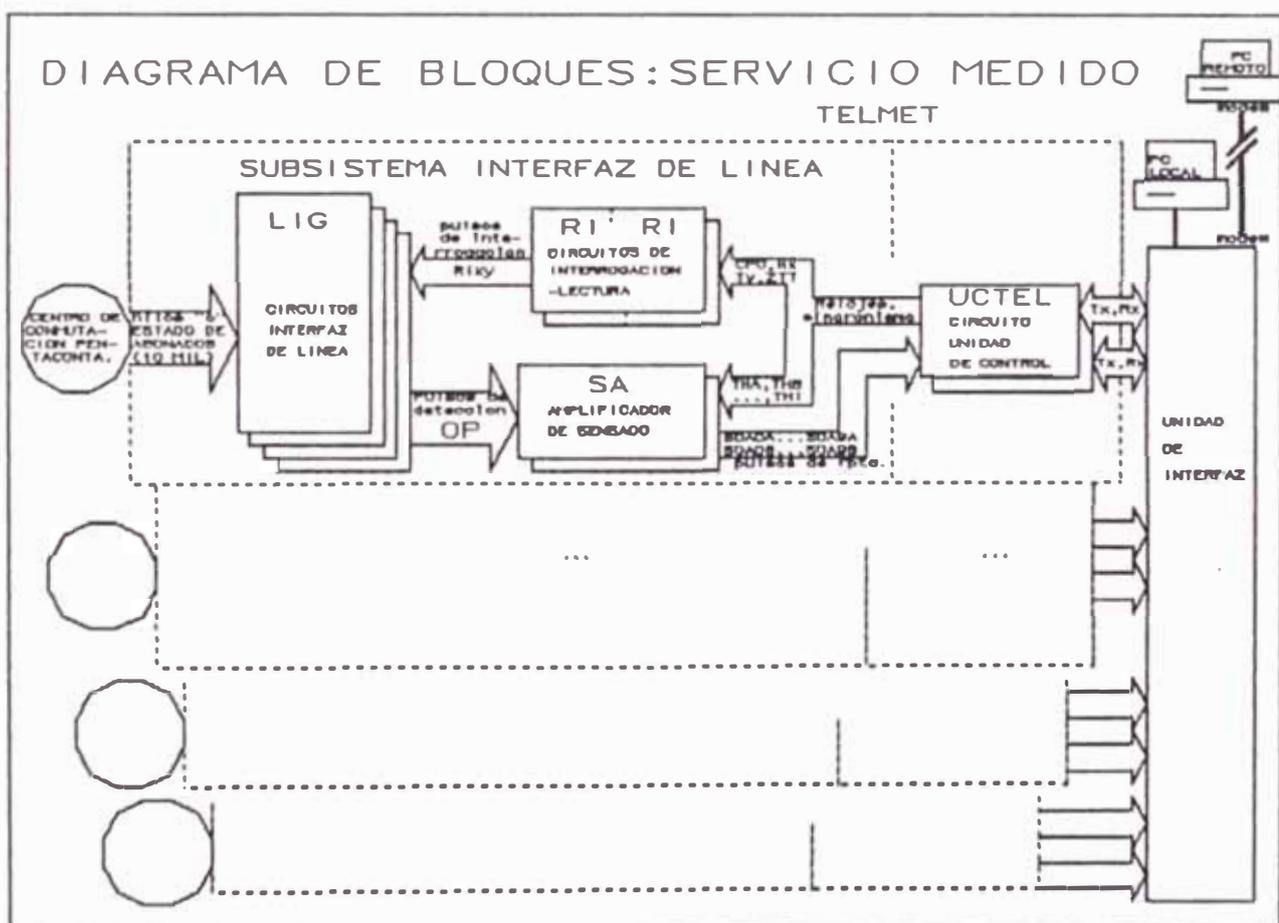


Fig. 1.2 Sistema de Servicio Medido y elme Modernizado

### 1.2.1 El equipo Telmet modernizado

El equipo Telmet observa el estado del hilo "c" proveniente del centro de conmutación (Pentaconta), tasa

y almacena el consumo del servicio. El **Telnet** está compuesto por: a) El Subsistema Interfaz de Línea y b) La Unidad de Control.

Como uno de los objetivos es la tasación programable, los tiempos de observación del estado del hilo "c" para cada abonado deben ser bastante pequeños, esto implica la eliminación de los temporizadores de tres minutos.

La Interfaz de línea queda invariable, la unidad de **control** le entrega las señales de control y sincronismo, y las respuesta que recibe son tratadas y procesadas por la unidad de control.

#### **1.2.1.1 El subsistema interfaz de línea**

La función del subsistema Interfaz de Línea es la de detectar la existencia de información de tasación en los hilos de cómputo de los abonados, e informar los resultados obtenidos a la Unidad de Control.

El Subsistema Interfaz de Línea está formado por:

- .LIG : Circuitos Line Interface Gate
- .RI : Matriz de selección de línea ó circuitos de Interrogación-Lectura
- .SA : Amplificador de Sensado

Los circuitos Line Interface Gate (LIGs) ó puerta de interfaz de línea son uno por cada abonado y están agrupadas en arreglos matriz de 1000 LIGs cada una para permitir ser leídos ordenada y sincronizadamente por los pulsos de interrogación provenientes de la matriz de selección de línea (RI). Físicamente los LIGs están organizados en tarjetas de 90 LIGs cada tarjeta. Así para OC Callao 69 de 1000 hilos "c" se necesita 12 tarjetas. Estas se denominan PPBG ó ILTEL100. En la Oficina Central OC Callao 65 de 10420 hilos "c", se han acomodado 120 tarjetas PPBG KC200886. **Las tarjetas ILTEL100 son copia de las KC200886.**

La **capacidad para tratar y multiplexar los estados del hilo "c"** por los circuitos RI y SA hace que las 10420

líneas se distribuyan en dos grupos de 5000 abonados incluyendo los 420 restantes en el primer grupo.

La Matriz de Selección de Línea es básicamente un arreglo de circuitos AND para que en base a las señales provenientes de los EPROMS de la Unidad de Control, pueda generar los pulsos de interrogación para lectura de los LIGs con las tensiones adecuadas (+5/-12 VDC). Estos circuitos arreglados, ordenados en las tarjetas KE200803J (con fuente de alimentación  $V_{DC}$ ) y KE200803K (sin fuente  $V_{DC}$ ) producen 100 pulsos de interrogación, 50 por tarjeta, éstas tarjetas también se denominan RITEL100. Se necesitan dos tarjetas por cada grupo de 5000 abonados (los 420 se suman al primer grupo). Total 4 tarjetas KE200803.

El Amplificador de Sensado es el que recibe los 10 pulsos de detección de cada grupo de 1000 LIGs, los discrimina por sus valores de tensión DC, validando sólo las que indican nivel VDC de conversación, y luego los multiplexa en función de los relojes de sincronismo de la Unidad de Control, para luego entregarle 20 líneas de respuesta, 10 como SOA0A...SOA9A con el estado de los primeros 5000 abonados más de los 420 últimos (una tarjeta KC200884A), y 10 líneas como SOA0B...SOA9B con el estado de los segundos 5000 abonados (otra tarjeta KC200884A). Las tarjetas también se denominan SATEL100.

A diferencia de Callao 65 que necesita de todas las tarjetas, Callao 69 se basta con dos RITEL100 y un SATEL100 por su menor número de abonados.

SOA0A...SOA9A y SOA0B...SOA9B son la entrada secuencial de datos de la Unidad de Control. Estos datos serán comprendidos por la Unidad de Control en función de las señales básicas de reloj ISCA', HASCA e INSCA más adelante explicadas en el acápite 2.1, Fig. 2.1.

### **1.3 Los microcontroladores**

También llamados microcomputadores ( $\mu C$ ) en un circuito integrado, son dispositivos más especializados e

integrados que los microprocesadores ( $\mu P$ ). Los microprocesadores, son básicamente el CPU que para completar un sistema necesita de unirse con otros periféricos. En cambio los microcontroladores son fabricados combinando un CPU, memorias, puertos de entrada/salida (E/S) y otros dispositivos como puertos seriales, temporizadores/contadores, convertidores análogo digital (ADC), etc. produciendo un auténtico microcomputador al nivel de un solo chip.

FABRICANTE	FAMILIA Y/O MICROCONTROLADOR MAS REPRESENTATIVO		CICLO INSTRUCCION ó FREC. DE OPERACION	MEMORIA INTERNA			TIMER COUNT	PUERTO SERIAL	A/D CON-VERT	WATCH DOG INC.
				EPROM	RAM	ROM				
INTEL	MCS-48	8048	2.5us		64B	1K	1			
	MCS-51	8051	1 us		128B	4K	2	SI		
	MCS-96	8098	1 us		232	8K	2	SI	SI	SI
MOTOROLA	6801	M6801U4	1 MHz		192B	4K	1			
	6804	M68HC04P3	2.2us		124B		1			
	6805	M68HC05A6	2.1MHz	2096B	176B	4K	1	SI		
	6811	M68HCS11E9FN	2.1MHz	512B	512B	12K	1	SI	SI	SI
GENERAL ELECTRIC	6805	CDP1804AC Seg-Fte. Motorola	3.2us		64B	2K	1			

TABLA 1.1 MICROCONTROLADORES MAS COMUNES DE ALGUNOS FABRICANTES.

La Tabla 1.1 es una tabla comparativa de los microcontroladores de algunos de los fabricantes más conocidos.

Las compañías fabricantes, ofrecen diversas familias de microcontroladores, con variaciones en capacidad de memoria interna (RAM/EPROM), más puertos de entrada/salida, temporizadores, contadores de 8 ó 16 bits, baja potencia de disipación, etc.

Los criterios fundamentales para escoger un microcontrolador adecuado para un determinado sistema son:

1ro) Que el microcontrolador y sus partes puedan satisfacer parcial ó totalmente las características y necesidades de la aplicación. (Por ejemplo, para



La familia Intel MCS-51 (una generación mas avanzada que la familia Intel MCS-48) está **constituida** por diversos chips microcontroladores, cada uno adaptado para un tipo específico **de** sistema. Las diferentes versiones son mostradas en la Tabla 1.2

Para éste diseño, por razones que detallaremos en el **siguiente capitulo**, la combinación de controlador con EPROM externo, es de lejos el de más bajo costo y más efectivo que una versión con EPROM interna.

### **1.3.2 Selección de los microcontroladores 8032AH y 80C452**

Las razones **para** escoger los microcontroladores 8032AH y 80C452 de entre los muchos miembros de la familia Intel MCS-51 son:

- a) Las características de las necesidades a cubrir y de las tareas a efectuar, las cuales exigen compartir la carga entre dos procesadores en relación maestro-esclavo, el 8032AH como maestro, y conectado a su bus como un periférico esclavo el 80C452.
- b) Por la característica del tamaño relativamente grande del programa de control del procesador maestro, así como para direccionar a otros dispositivos externos, es preferible una combinación de microcontrolador con EPROM externo, que además habilita el acceso a data externa. El 8032 direcciona sus 64K bytes de programa externamente.
- c) Por la necesidad de una RAM interna que maneje muchas variables y un stack grande. Lo más que ofrece la familia es 256 bytes.
- d) Por la necesidad de usar varios temporizadores para programar diversos eventos y tareas. El 8032 posee 3 temporizadores de 16 bits, con ventaja sobre otros microcontroladores.

Existe **también** el 80C32 que es fabricado con **tecnología CHMOS**, mientras que el 8032AH es fabricado con **tecnología HMOS**, ésto le permite al 80C32 trabajar en modo Idle y modo Power Down, reduciendo así su consumo de

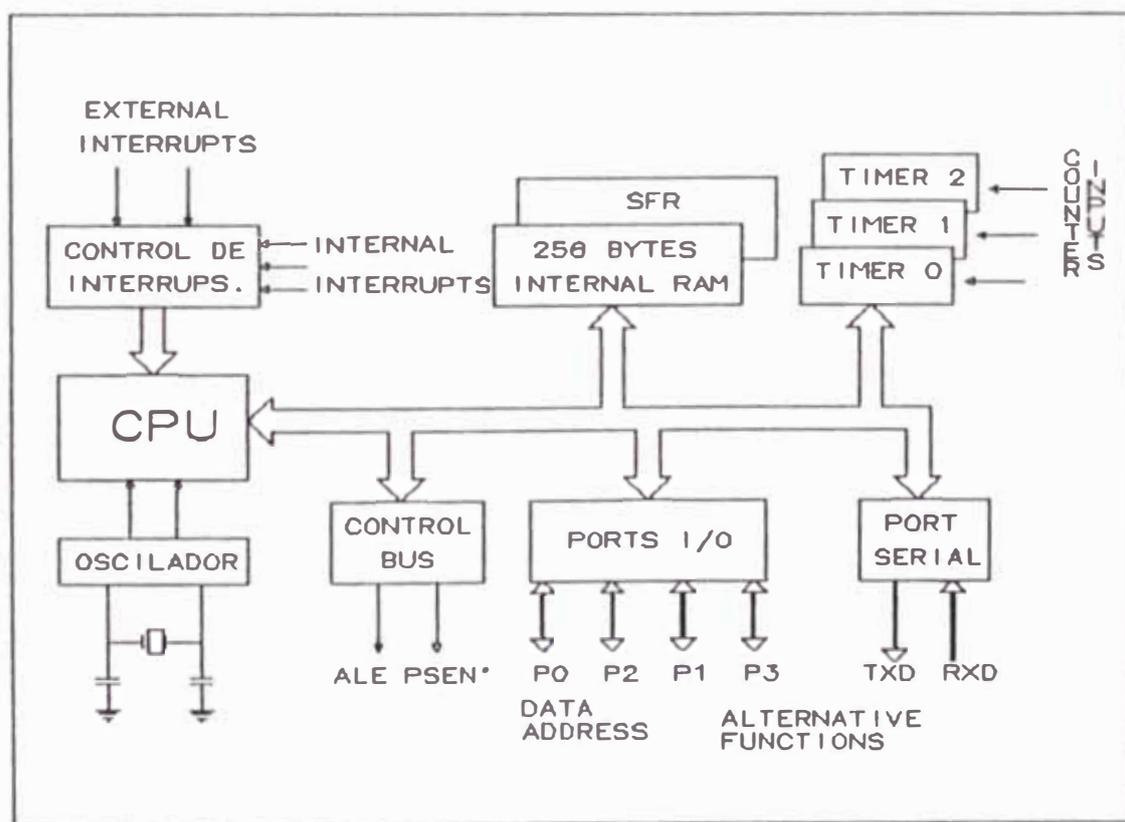
energía. Esta facilidad, es innecesaria para la presente aplicación, optando así por el microcontrolador Intel 8032AH (al cual nos referiremos en adelante simplemente como 8032).

#### 1.4 Descripción de los microcontroladores

##### Intel 8032AH y 80C4 2

El microcontrolador 8032AH es idéntico al 8052AH ó al 8752BH, pero no tiene memoria ROM ó EPROM interna, el 8032 direcciona sus 64KBytes de programa externamente, también el 80C452.

El microcontrolador 80C452 es idéntico al 83C452 ó al 87C452P, pero sin ROM ó EPROM interna, el 80C452 está preparado para trabajar como un dispositivo periférico paralelo conectado al bus de direcciones y datos, y con posibilidad de generar interrupciones. Intel lo nombra también UPI452: Universal Peripheral Interface, es un controlador esclavo de propósito general.



ig. 1.3 estructura básica del microcontrolador Intel 8032

### 1.4.1 Arquitectura del 8032 y 80C452

La estructura básica de la arquitectura del 8032 es mostrada en la Fig. 1.3.

El 80C452 se basa en la misma arquitectura, con los siguientes adicionales y diferencias: posee un canal de comunicación paralelo hacia su procesador maestro (bus hacia el host) basado en un buffer FIFO y Registros de Comando Inmediato (ICR). Posee también dos canales de Acceso Directo a Memoria (DMA) y un Puerto paralelo adicional. Comparado con el 8032, el 80C452 tiene solo dos temporizadores/contadores.

Para un mayor detalle respecto a la arquitectura interna de los microcontroladores 8032 y 80C452 consultar el anexo E de características técnicas de componentes.

En ambos controladores, su CPU lee el programa del usuario desde su bus local y ejecuta las instrucciones allí almacenadas. Los elementos primarios que lo componen son la Unidad Aritmética Lógica de ocho bits, asociado con los registros A, B, PSW, SP, y los registros de 16 bits Program Counter y Data Pointer.

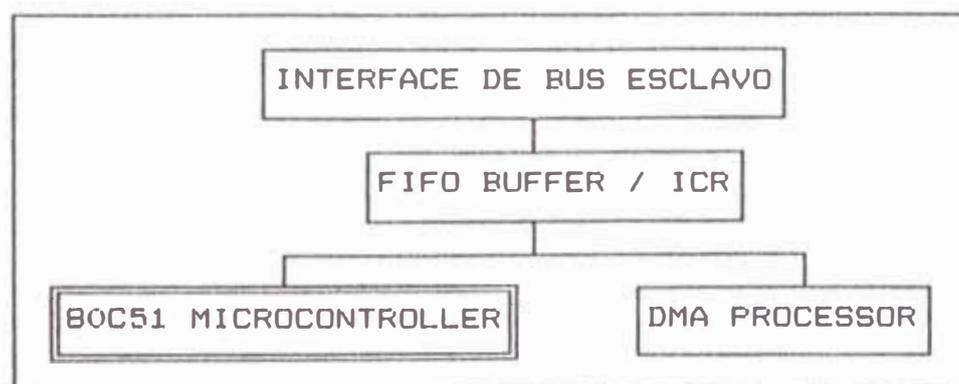


Fig. 1.4 Estructura del microcontrolador UPI 80C452

La arquitectura del MCS-51 soporta diversas direcciones de memoria en espacios "físicos" distintos, funcionalmente separados en el nivel de hardware por diferentes mecanismos de direccionamiento, señales de control de lectura-escritura, o ambos; estos espacios de memoria son:

- Memoria de datos dentro del chip (interna)

- .Registros de función especial dentro del chip (interna)
- .Memoria de programa fuera del chip
- .Memoria de datos fuera del chip

#### 1.4.2 Organización de memoria de programa y datos

Los microcontroladores 8032 y 80C452 poseen tres espacios de direcciones: para Programa, para Datos externo y para Datos interno, como se muestra en la Fig. 1.5. Mientras el programa es direccionado por los 16 bits del registro PC, la memoria de Datos externo también es direccionada por 16 bits del registro Data Pointer (DPTR).

-La Memoria de Programa (Program Memory) puede ser sólo leída, y tiene una capacidad máxima de 64K bytes de ROM externa. El indicador (strobe) para lectura del ROM externo es la señal PSEN' (Program Store ENable).

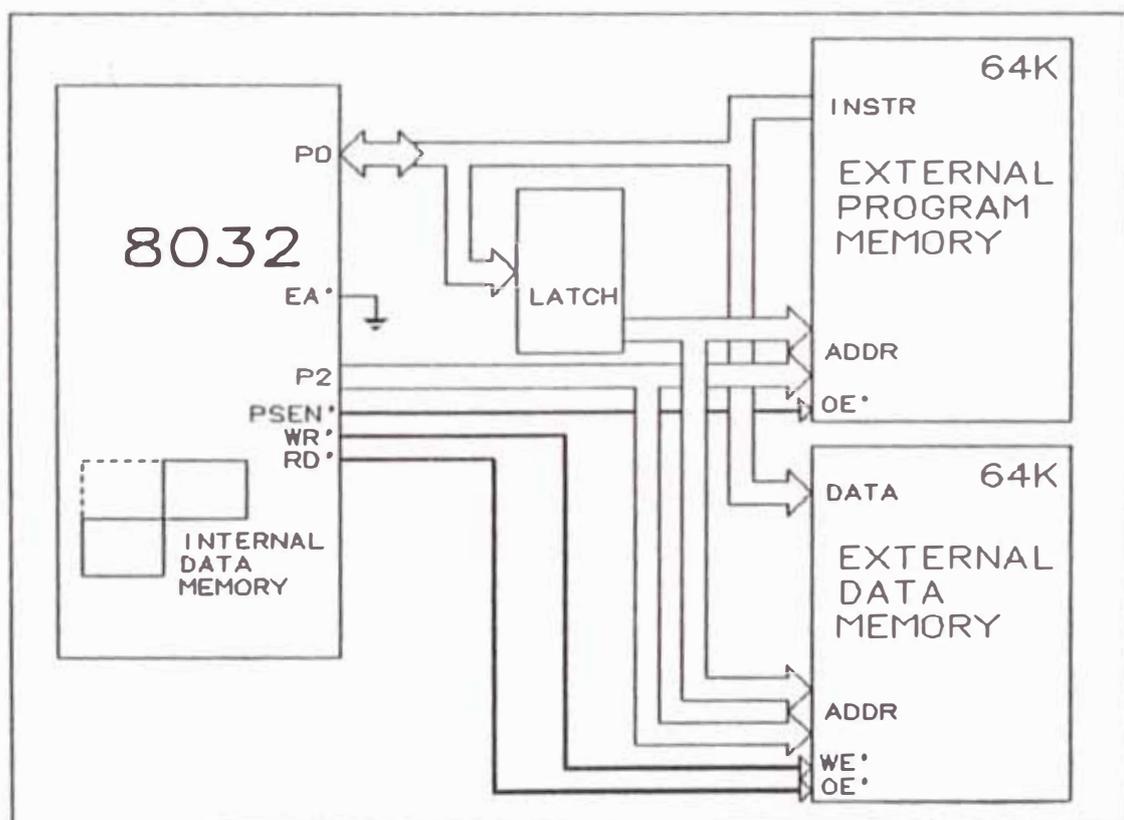


Fig. 1.5 Memoria de Programa y Datos Externos

-La Memoria de Datos Externa (External Data Memory) ocupa un espacio separado de direcciones distinto de la memoria

de Programa. El espacio de la memoria de Datos puede ser direccionado hasta en 64K bytes de RAM externa; para accederla, el CPU genera señales de lectura y escritura, RD' y WR' respectivamente.

-La Memoria de Datos Interna (Internal Data Memory) físicamente ubicada dentro del mismo microcontrolador, está a su vez dividida en tres bloques, uno de los cuales, el SFR (Space Function Register) es particularmente diferente e importante porque contiene registros de función especial del microcontrolador.

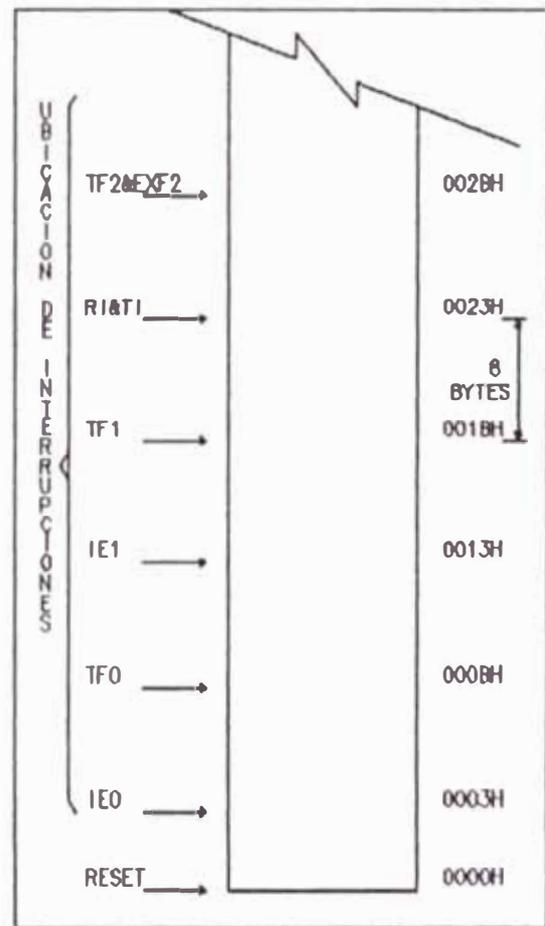
**1.4.2.1 Memoria de programa**

La Fig. 1.6 muestra un mapa de la parte inferior de la memoria de Programa (PROGRAM MEMORY). Después del reset, el CPU comienza la ejecución desde la ubicación 0000H.

Cada interrupción es asignada a una ubicación fija en la Memoria de Programa. Una interrupción ocasiona que la CPU salte a esta ubicación, donde comienza la ejecución de una rutina de servicio.

La Interrupción Externa 0, por ejemplo, es asignada a la ubicación 0003H. Si la Interrupción Externa 0 es usada, su rutina de servicio debe comenzar en la ubicación 0003H.

Si la Interrupción no es usada, su ubicación de servicio es disponible para propósito general de la Memoria de Programa.



F g. 1.6 Ubicación de Interrupciones en la Memoria de Programa.

Las ubicaciones de los "interrupt service" están espaciados a intervalos de 8 bytes:

0003H para la Interrupción Externa 0

000BH para el Temporizador 0

0013H para el Interrupción Externa 1

001BH para el Temporizador 1, etc.

Si una rutina de servicio a Interrupción es pequeña, puede residir completamente en el intervalo de 8 bytes. Rutinas de servicios más grandes pueden usar una instrucción de salto (jump) para avanzar por encima de las inmediatas siguientes ubicaciones de interrupción, si estas otras están en uso.

El pin EA' (Externa Address) en el 8032 y en el 80C452 está puesto a tierra para indicar que todo el espacio de Program Memory esta fuera del chip.

El pin PSEN' (Program Storage ENable) es la señal de lectura de EPROM externa.

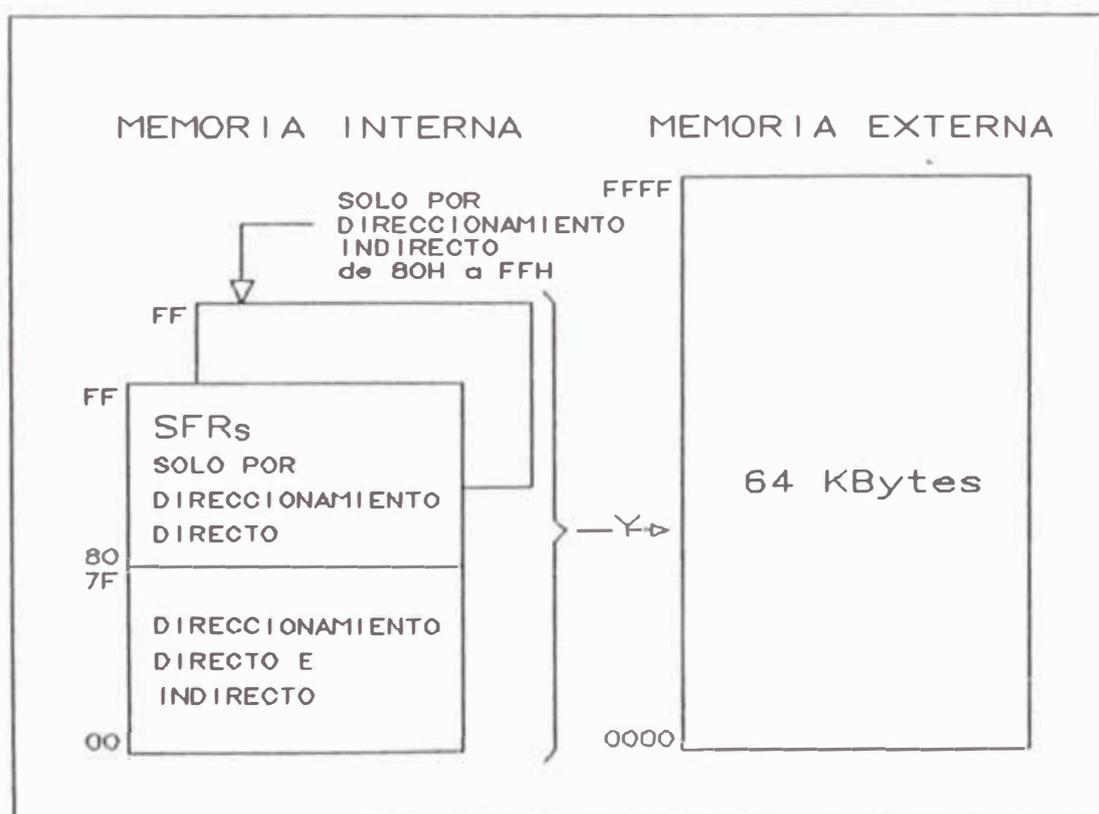
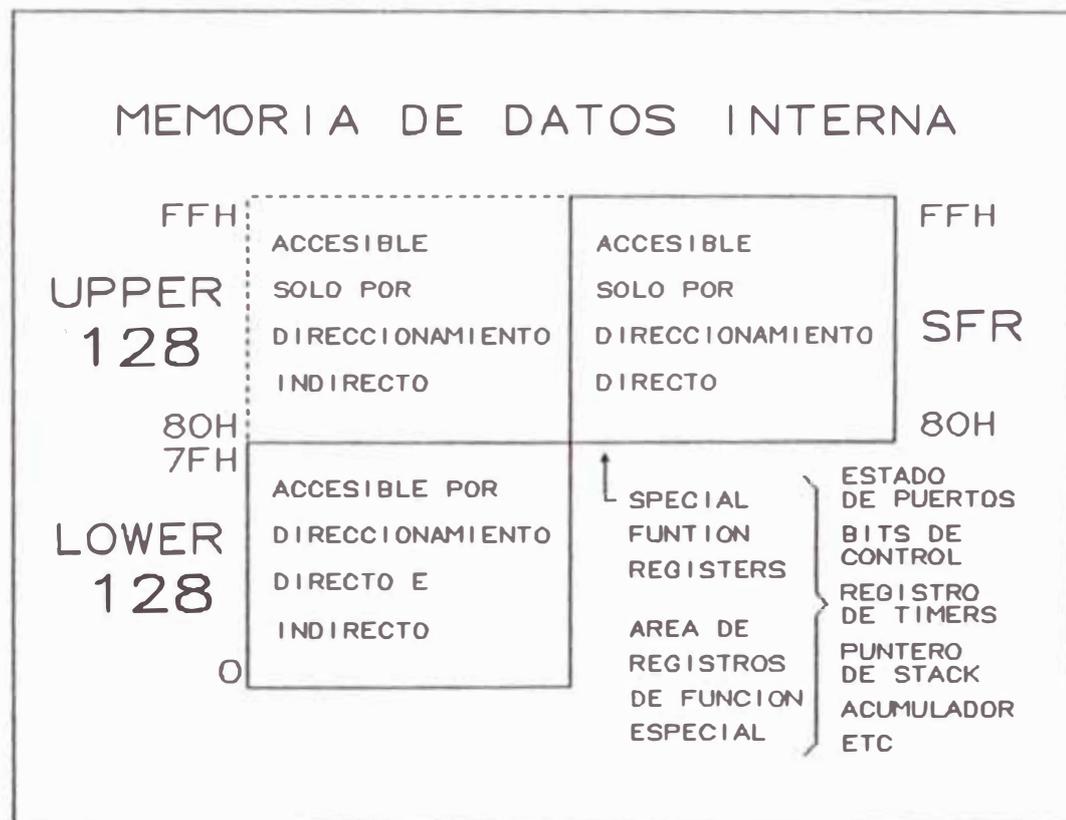


Fig. 1.7 Memoria de datos del 8032 y 80C452

### 1.4.2.2 Memoria de datos externa

Para **direccionar** la memoria de datos externa (External Data Memory) se hace a través del Puerto 0, que multiplexa las direcciones y datos. El Puerto 2 sirve para emitir los bits de dirección más significativos.

A éste espacio no solo pueden conectarse dispositivos RAM, sino también cualquier otro dispositivo que pueda ser **seleccionado** por su dirección y accesado sus registros mediante el bus de datos. La CPU generará señales de RD' y WR' necesarias durante el acceso a los datos externos



F'g. 1.8 Memoria de da os 'nterna In el 8032 80C452.

### 1.4.2.3 Memoria de datos interna

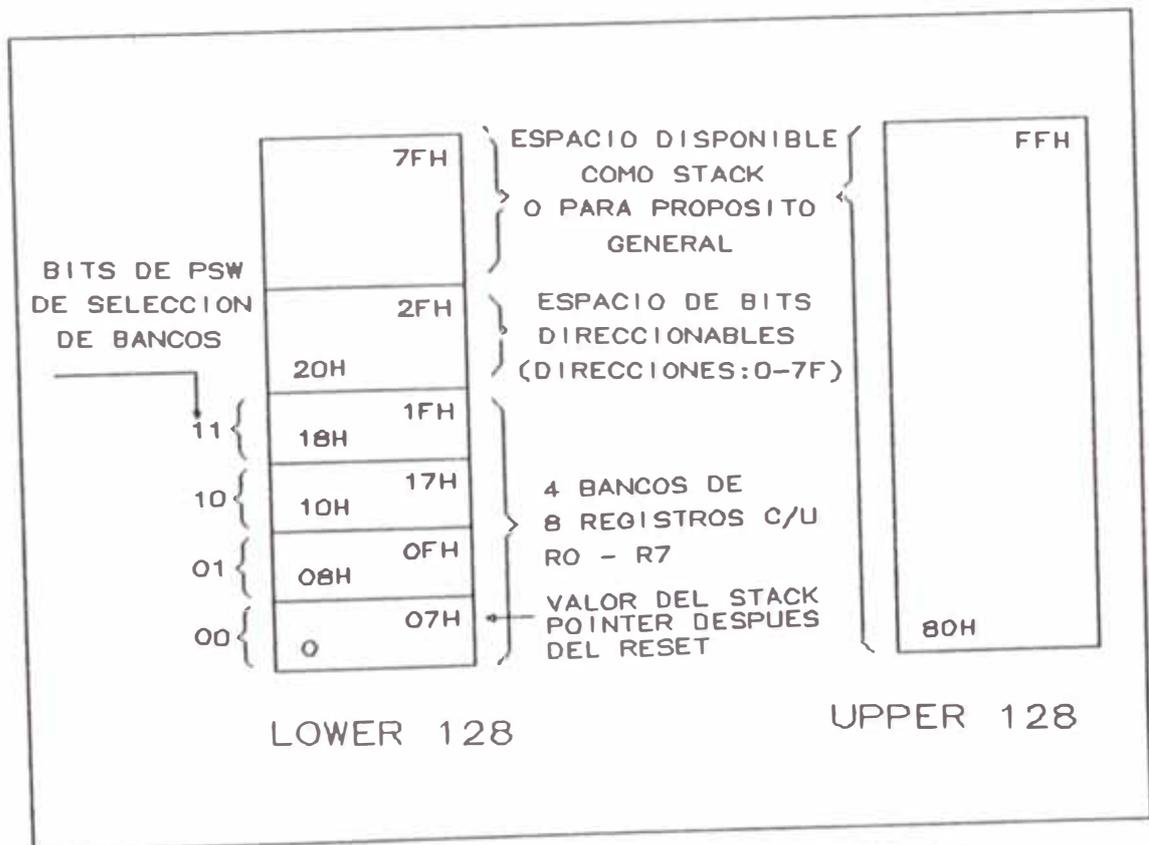
La Memoria de datos interna (Internal Data Memory) está dividida en **tres bloques** los cuales nos referiremos como espacios LOWER 128, UPPER 128 y espacio SFR (Special Function Registers). Esta memoria interna siempre se direcciona con un byte. De acuerdo al modo de

direccionamiento interno, con éste byte (8 bits) se puede direccionar a 384 bytes.

### Espacio LOWER 128:

Esta memoria de datos interna es una RAM de 128 bytes. Los 32 primeros bytes están agrupados en 4 bancos de 8 registros cada uno. El programa puede llamar a éstos registros como R0 a R7. Dos bits en el Program Status Word (PSW) seleccionan cual banco de registro es el usado, esto permite más eficiencia en el uso de variables tamaño registro.

Los siguientes 16 bytes encima del banco de registros forman un bloque de memoria direccionable bit a bit. Las instrucciones de la familia MCS-51, incluyen instrucciones de selección de bit, y los 128 bits de esta área pueden ser directamente accedidos por estas instrucciones. Las direcciones de los bits en esta área van de 00H a 7FH.



ig. 1.9 -espacio LOWER 128 y espacio UPPER 128

Lo que queda de la parte superior puede ser usada para propósito general ó como pila de almacenamiento (stack).

Todos los bytes del espacio LOWER 128 pueden ser accesados por los direccionamientos directo e indirecto.

#### **Espacio UPPER 128:**

Esta memoria de datos interna es una RAM de 128 bytes, los cuales pueden ser accesados solo por direccionamiento indirecto. Está disponible para propósito general ó como pila de almacenamiento.

#### **Espacio SFR:**

Es la memoria de datos interna que contiene el estado de los registros función especial del 8032 y 80C452, por lo cual se le denomina Special Function Register (SFR).

En éste espacio se ubican los registros acumulador ACC, B, la palabra de estado PSW, incluye los puertos paralelos programados temporizadores, control de interrupciones, periféricos, etc.

Estos registros pueden ser accesados solamente por direccionamiento directo.

Por otro lado los registros del espacio de SFR, cuya dirección termina en x0H ó x08H, son direccionables bit a bit; las direcciones de éstos bits en ésta área van de 80H a FFH

El 80C452 posee más de 30 registros adicionales que el 8032, necesarios para controlar su buffer FIFO, interrupciones del host y sus canales DMA.

### **1.4.3 Instrucciones de la familia MCS-51**

Todos los miembros de la familia ejecutan las mismas instrucciones (111 instrucciones, 64 de ellas se ejecutan en un solo ciclo de máquina). Las instrucciones de MCS-51 son apropiadas para aplicaciones de control de 8 bits. Por su aplicación se agrupan en:

- .Operaciones Aritméticas
- .Operaciones Lógicas para variables tipo Byte

- .Transferencia de Datos
- .Manipulación de variables Booleanas
- .Control de programa.

Virtualmente todas las instrucciones se ejecutan en uno ó dos ciclos de máquina -uno o dos microsegundos con cristal de 12Mhz- con la sola excepción de la multiplicación y división que se completan en cuatro ciclos de máquina.

La familia MCS-51 también proporciona de rápidos modos de direccionamiento para el acceso a la RAM interna y facilitar las operaciones de byte.

#### 1.4.3.1 La palabra de estado (PSW)

Es un registro cuyos bits reflejan permanentemente el estado del CPU y la ejecución del programa, el PSW reside en el espacio SFR. La Fig. 1.10 muestra al PSW.

MSB				LSB			
CY	AC	FO	RS1	RS0	OV	-	P
SMB POS	NOMBRE Y/O SIGNIF.		SMB POS	NOMBRE Y/O SIGNIF.			
CY	PSW.7	Carry flag		OV	PSW.2	Overflow flag	
AC	PSW.6	Auxil. Carry flag		-	PSW.1	Flag definible para usuario	
FO	PSW.5	Flag 0 (Disp. para propósito general)		P	PSW.0	Flag de paridad Puesto y limpiado por hardware cada ciclo de instrucción para indicar un número impar o par de bits "uno" en el acumulador	
RS1	PSW.4	Bit 1 selector de banco de registros					
RS2	PSW.3	Bit 0 selector de banco de registros					

RS1 y RS0 son reseteados o limpiados por software para determinar el banco de registros de trabajo.

NOTA: El contenido de RS1 y RS0 habilitan al banco de registros de la manera siguiente:  
(RS1,RS0) BANCO DIRECC.

{	0,0	0	00H-07H
{	0,1	1	08H-0FH
{	1,0	2	10H-17H
{	1,1	3	18H-1FH

Fig. 1.10 Palabra de estado (PSW) Intel 8032

#### 1.4.4 Temporización de la CPU

Todos los microcontroladores de la familia MCS-51 poseen internamente un oscilador el cual puede ser usado como fuente de reloj para el CPU. Para usar su oscilador se conecta un cristal entre los pines XTAL1 y XTAL2 del  $\mu\text{C}$ , y estos a su vez, con condensadores a tierra. También es posible trabajar con un reloj externo, en algunas aplicaciones puede ser necesaria ésta opción para sincronizar al  $\mu\text{C}$  con una referencia externa.

##### 1.4.4.1 Ciclo de máquina

Un ciclo de máquina consiste de una secuencia de 6 estados numerados de S1 a S6. Cada estado contiene 2 periodos del oscilador. Cada estado es dividido en fase 1 y fase 2.

En las Fig. 1.11 y Fig. 1.12 se muestra la ejecución de algunas instrucciones en el tiempo, así como la composición de los ciclos de máquina para las mismas instrucciones.

Como vemos de la Fig. 1.11, normalmente dos bytes de programa son leídos durante cada ciclo de máquina, excepto en la instrucción MOVX.

La instrucción MOVX toma dos ciclos de máquina para ejecutarse, sea en lectura ó escritura; es importante notar que en el acceso a datos externos, la señal ALE se suspende una vez, y no hay ciclo FETCH o lectura de byte de programa en dos oportunidades consecutivas.

Si ocurre acceso a la memoria de datos externa, la señal PSEN' es obviada, permitiendo el direccionamiento y el uso del bus de datos a la memoria externa.

El ciclo de máquina es equivalente a 12 periodos de oscilación del cristal (ó reloj) conectado externamente.

---


$$T_{\text{ciclo de máquina}} = 12/f_{\text{oscil}} \quad \text{Ecuc. 1.1}$$


---

así, para una  $f_{\text{oscil}} = 12 \text{ MHz}$  se tiene  $T_{\text{cm}} = 1 \mu\text{s}$ , y para una  $f_{\text{oscil}} = 11.0592 \text{ MHz}$  se tiene  $T_{\text{cm}} = 1.085 \mu\text{s}$ .

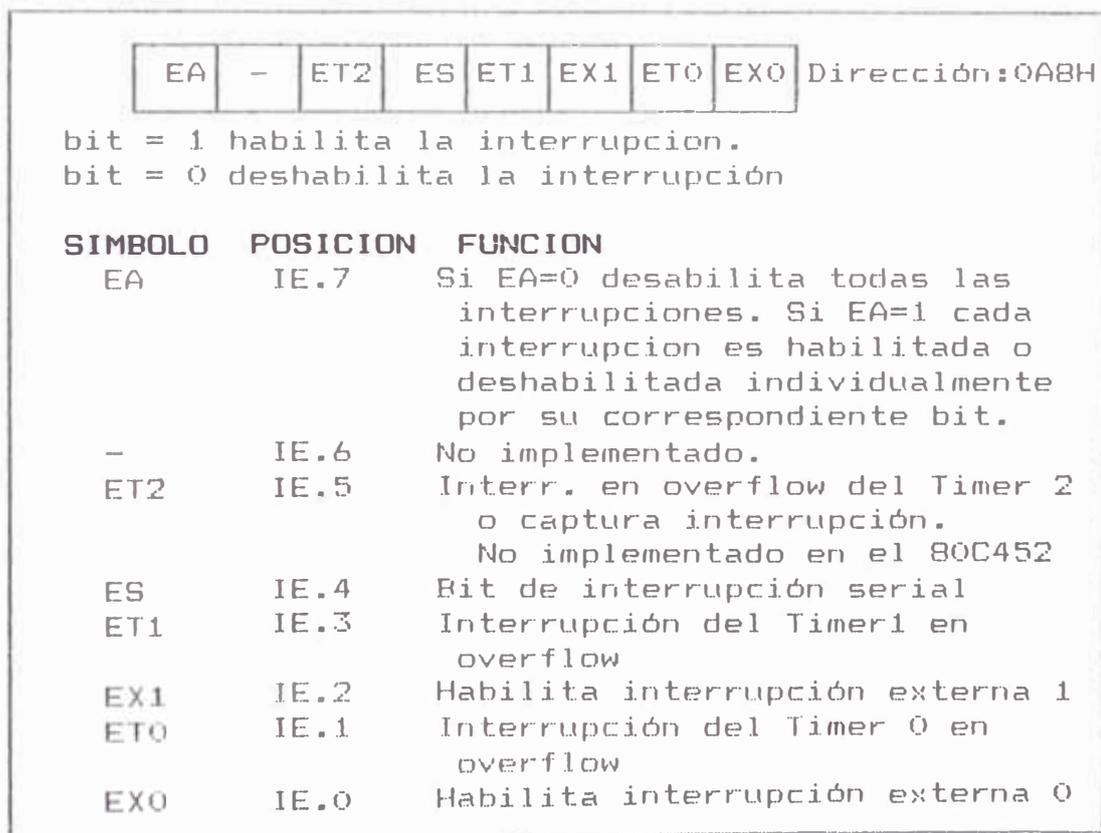
### 1.4.5 Estructura de interrupciones

El 8032 provee 8 vectores de interrupción, que atienden a 8 fuentes, estos vectores son:

- 2 interrupciones externas.
- 3 interrupciones de temporizadores internos.
- 1 interrupción de puerto serial UART interno.

El 80C452 provee un total de 8 vectores de interrupción, que atienden 9 fuentes, su operación la misma que la del 8032, con la adición de tres nuevas fuentes:

- 2 interrupciones de los canales 0 y 1 de DMA respectivamente.
- 1 interrupción de la interface de bus esclavo-FIFO.



**F'g. 1.13 Reg. I de habilitación de Interrupciones**

El 8032 y 80C452 administran las interrupciones mediante los bits de dos registros: IE para habilitar/deshabilitar e IP para priorizar.

El 80C452 adicionalmente provee del registro EIP en SFR para habilitar y priorizar estas nuevas fuentes de interrupción; las mismas están globalmente habilitadas o deshabilitadas por el bit EA del registro IE SFR

### Habilitación de interrupciones:

Cada fuente de interrupción puede ser individualmente habilitada o deshabilitada seteando o limpiando un bit del registro IE (Interrupt Enable) en el SFR. Este registro también contiene un bit para deshabilitación global, el cual puede ser puesto a 0 para deshabilitar todas las interrupciones a la vez. La Fig. 1.13 muestra al registro IE del 8032 y 80C452

- - PT2 PS PT1 FX1 PTO FX0 Dirección:0B8H

bit = 1 asigna alta prioridad.

bit = 0 asigna baja prioridad.

SIMBOLO	POSICION	FUNCION
-	IP.7	reservado
-	IP.6	revertado
PT2	IP.5	Bit de prioridad del Timer2 No implementado en el 80C452
PS	IP.4	Bit de prioridad de Puerto serie
PT1	IP.3	Bit de prioridad del Timer1
FX1	IP.2	Bit de prioridad de INT external
PT0	IP.1	Bit de prioridad del Timer0
PX0	IP.0	Bit de prioridad de INT externa0

Fig. 1.14 Registro IP Priorizador de Interrupciones

### Prioridad de Interrupciones:

También es posible asignar una prioridad mayor a una interrupción, seteando el bit correspondiente en el registro IP que administra la prioridad de las interrupciones. Esto permite que mientras un servicio de interrupción se está ejecutando, no puede ser interrumpido por otra interrupción de nivel inferior ó igual.

Para el 8032 la prioridad de interrupciones si no se altera por programación, es como sigue:

Fuente	Nivel de prioridad
1 External Interrupt 0	mas alto
2 Timer Counter 0	
3 External Interrupt 1	
4 Timer Counter 1	
5 Tx o Rx por Canal Serial	V
6 Timer 2 o External Interrupt 2	mas bajo

Para el 80C452 la prioridad de interrupciones también es programable en IP e EIF, pero por omisión, su prioridad es como se muestra:

Fuente	Nivel de prioridad
1 External Interrupt 0	mas alto
2 Timer Counter 0	
3 DMA Canal 0	
4 External Interrupt 1	
5 DMA Canal 1	
6 Timer Counter 1	
7 Interface de Bus Esclavo FIFO	V
8 Tx o Rx por Canal Serial	mas bajo

---

Reg. EIF del 80C452: Esp SFR

- - PFIFO EDMA0 EDMA1 PDMA0 PDMA1 EIFIFO Dir:F8H

---

SIMBOL	POSICION	FUNCION
-	IEP.7	reservado
-	IEP.6	reervado
PFIFO	IEP.5	Prioridad de Interrupción de la interfaz de bus esclavo FIFO
EDMA0	IEP.4	Habilita interrupt de canal0 DMA
EDMA1	IEP.3	Habilita interrupt de canal1 DMA
PDMA0	IEP.2	Prioridad de canal 0 de DMA
PDMA1	IEP.1	Prioridad de canal 1 de DMA
EIFIFO	IEP.0	Habilita Interrup: de interfaz de bus esclavo del FIFO

---

F g. 1. 5 Registro IP de Habilitación y Priorización de Interrupciones de 80C452

## **1.5 Transmisión de datos**

Es el movimiento de información que ha sido y va a ser procesada y codificada generalmente en forma binaria, sobre un sistema de transmisión eléctrica.

Transmitir datos significa transmitir información entre equipos ó dispositivos. A los paquetes mínimos de información o data intellegible se les denomina byte. Toda vez que existe transmisión de datos de un punto a otro, establecemos un conjunto de especificaciones que deben concretarse para mantener la comunicación entre éstos puntos. Así tenemos especificaciones sobre el modo de transmisión, interfaz físico y protocolo de comunicación.

### **1.5.1 Modos de transmisión**

Los modos de transmisión más importantes son:

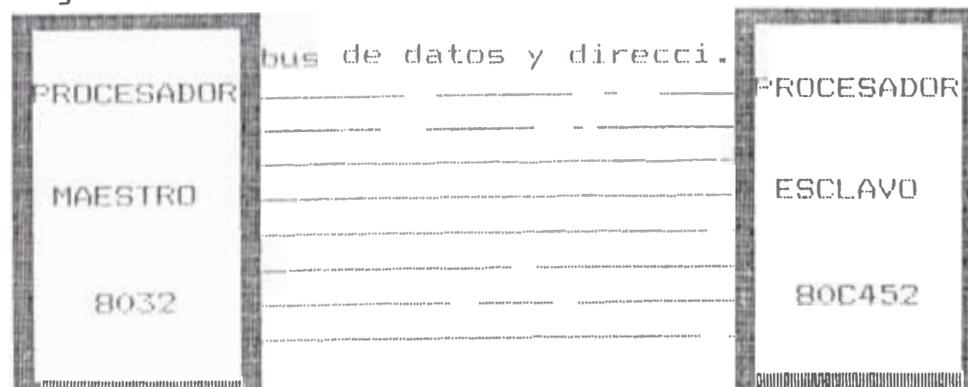
- Transmisión serie/paralelo
- Transmisión Simplex/Half y Full Duplex
- Transmisión Síncrona/Asíncrona

#### **Transmisión serie/paralelo:**

En función del número de líneas físicas usadas así como del envío de "bit a bit" la transmisión puede ser paralela ó serial.

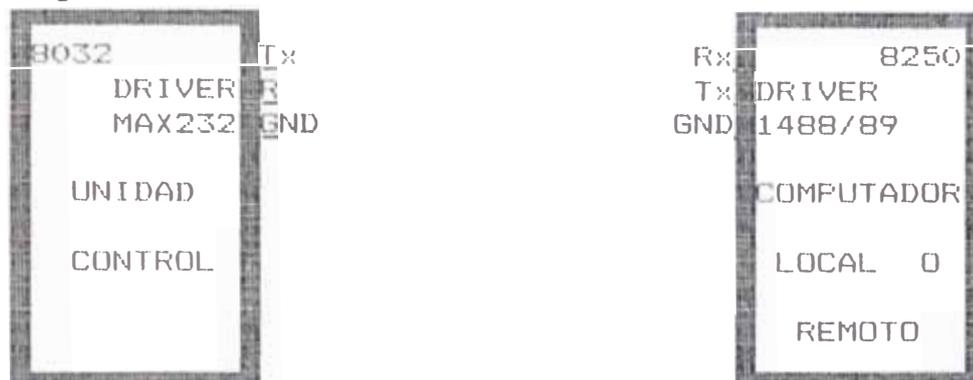
Es paralela cuando se usan varias líneas para comunicar dispositivos que están bastante cerca entre sí, un paquete mínimo de información -un byte- puede viajar entero en un mismo instante de tiempo, sin necesidad de ser particionado en bits en el tiempo, por ejemplo un procesador y su RAM, ROM, periféricos se comunican en palelo vía su bus de direcciones y datos.

**Fig. .17 TRANSMISION EN PARALELO**



La transmisión de datos es serial cuando los datos tienen que ser enviados "bit a bit". Es ventajoso cuando los dispositivos a comunicar se encuentran alejados entre sí, pues usa un menor número de hilos/cables para el enlace. Tomemos como ejemplo el enlace entre nuestra Unidad de Control y el Computador consola de operador.

**Fig. 1.18 TRANSMISION EN SERIE**



#### **Transmisión Simplex/Half y Full Duplex:**

La modalidad de transmisión de datos observada en función de la simultaneidad de envío de datos, podemos señalarla como:

Transmisión simplex, ó siempre en un sólo sentido.  
 Transmisión Half Duplex ó en ambos sentidos pero uno a la vez, y Transmisión Full Duplex ó en ambos sentidos y simultáneamente.

#### **Transmisión Síncrona/Asíncrona:**

En transmisión serial, para poder interpretar correctamente la información transmitida, es necesario que el transmisor y el receptor tengan una base de tiempos común, es decir, estén sincronizados.

Según el método de sincronización que use el emisor y receptor, la transmisión puede ser síncrona ó asíncrona.

Es asíncrona cuando se usa bits señalizadores de inicio (start) y parada (stop) a los extremos de cada carácter transmitido. La sincronización es implícita con

el bit de start en cada caracter. La transmisión de datos se puede considerar de caracter a caracter.

Es síncrona cuando se transmite bloques grandes y completos de caracteres, la sincronización no se da para cada caracter, por el contrario, para todo el bloque ó trama.

### **1.5.2 Interfaz físico de comunicación de datos**

El interfaz físico de comunicación de datos, está definido por los aspectos mecánicos, eléctricos y los procedimientos de intercambio de circuitos.

Para el caso de transmisión serial, países y organismos internacionales han establecido muchas recomendaciones para uniformizar el interfaz físico entre equipos, así por ejemplo, normas como EIA RS232-C, EIA 232-D, CCITT V.24/V.28, X.21bis, etc., sugieren el tipo de conector mecánico, longitud y velocidad de transmisión, rango de tensiones para Marca-Espacio, etc.

### **1.5.3 Protocolos de comunicación**

Se denomina así al conjunto de métodos ó procedimientos que gobiernan la transmisión de datos entre equipos. Son las reglas que hacen que la comunicación trabaje asegurando que los datos enviados sean recibidos correctamente.

Es parte del protocolo la especificación de las características de la información ó datos que se transmiten.

Así, cuando se establece el protocolo para una comunicación serial, se debe especificar el modo de transmisión, características del formato, velocidad, que caracteres de control se usa, la estructura del contenido de un mensaje como: los caracteres de cabecera, direcciones del transmisor-receptor, la data del usuario, los caracteres cola de mensaje, los caracteres de chequeo de mensaje, etc.

**Fig. 1.19 jemplo de una estructura de mensaje:**

CONTROL DIRECCIONES	DATA DE USUARIO	CONTROL CHEQUEO MENSAJE
CARECERA	DATA DE USUARIO	COLA

En la Fig. 1.19 se muestra una cadena de caracteres asíncronos ( por ejemplo, de 8 bits cada caracter, más bits de inicio y parada) para transmisión-recepción que en su conjunto forman un mensaje comprensible por transmisor-receptor.

Son tambien parte del protocolo las especificaciones de la acción a tomar después de remitido un mensaje, cómo se asegura el emisor la correcta recepción por el destinatario ó de la necesidad de que el mensaje sea retransmitido. Estas informaciones pueden ser logradas mediante el uso de caracteres de control.

Tendiendo siempre a uniformizar las normas, los fabricantes y organismos han establecido protocolos estándar. Así algunos de los protocolos síncronos más comunes son: SDLC de IBM, HDLC de CCITT. De los asíncronos podemos mencionar: el Tinet, el ISO ASYNC (NCR/ISO).

En general, cualquiera que desee interconectar sus equipos, debe repetar un estándar ó crear su propio protocolo.

## **CAPITULO II**

### **DISEÑO DE HARDWARE**

La Unidad de Control del Telmet (en adelante UCTEL) es diseñada para hacer la lectura de los hilos "c" de los abonados, aplicar el plan de tarifas programado, calcular los pulsos e incrementar la cuenta acumulativa (del consumo de servicio) de cada abonado. La Unidad de Control genera las señales que requiere el Telmet, así como leerá el resultado del muestreo de los hilos "c" que llegan a los LIG circuitos Line Interface Gate (Ref. 1.2.1.2 y 1.2.1.3). La UCTEL basa sus cálculos en éste muestreo (señales SA: **SOAOA...SOA9A** y **SOAOB...SOA9B**) bajo el principio de que la condición del hilo "c" proveniente de la central telefónica, se mantiene en un estado permanente durante la conversación.

Esta Unidad de Control -UCTEL- va a ser diseñada para trabajar en paralelo con otra unidad idéntica para proveer una función duplicada, sin embargo ésta unidad también puede trabajar sola.

La UCTEL posee un canal serial para recibir/enviar información, sea a través de la Unidad de Interfaz UITEL, ó directamente a un terminal (computador) inteligente.

#### **2.1 Condiciones de temporización**

Son varias las señales de tiempo (de reloj y sincronismo) que se tienen que respetar para mantener el funcionamiento de la Matriz de Selección de Línea (RI) y del Amplificador de Sensado (SA) (ver Ref. 1.2.1.1 y Fig. 1.2). **Para el diseño, partiremos del diagrama de tiempos mostrado en la Fig. 2.1.**

##### **2.1.1 Señales de control ISCA', HASCA e INSCA**

Delimitando las condiciones de interacción de la Unidad de Control con el resto del sistema, tendremos que

los datos de los abonados proviene de las líneas **SOA0A...SOA9A** y **SOA0B...SOA9B** de los Amplificadores de Sensado (SA), y son interpretados en función de las señales de control y sincronismo ISCA, HASCA e INSCA tal como se muestra en la Fig. 2.1.

**ISCA'** indica la presencia de datos válidos de un nuevo grupo de 10 abonados. Se considera datos al estado de conversación o no de un abonado originante de llamada. ISCA está presente  $24\mu\text{s}$  cada  $480\mu\text{s}$ .

**HASCA** indica la lectura del primer ó segundo circuito Amplificador de Sensado. **HASCA=1** por  $240\text{ms}$  sobre un periodo de  $500.16\text{ms}$ .

**INSCA** en combinación con HASCA, fija si se está leyendo a los primeros ó segundos 5000 abonados ó al último grupo de 420 abonados. **INSCA=1** por  $144\mu\text{s}$  sólo en el inicio de los primeros y segundos **5000 abonados**, su periodo de **500.16ms**.

De acuerdo a la temporización de HASCA e INSCA (Fig. 2.1), la lectura de los datos de los 10420 abonados se efectúa en dos grupos de 5000 y uno de 420. En la Fig. 2.1 se muestra la relación entre los cambios de estado de las señales HASCA e INSCA cuando se inicia con la lectura de los datos de los primeros **5000 abonados**, de los segundos 5000 y de los últimos 420 abonados.

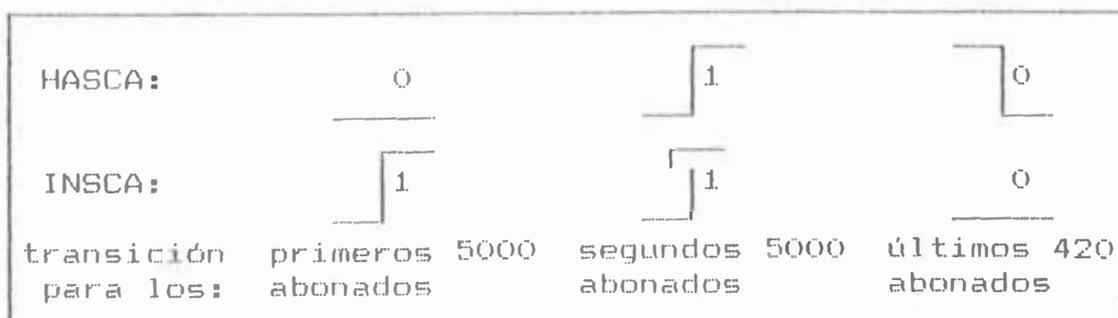


Fig. 2.2 Relación de cambio de estado entre HASCA e INSCA

## 2.2 Diagrama de bloques de la unidad de control

La Fig. 2.3 muestra el diagrama de bloques de la Unidad de Control, donde definimos un sistema basado en **compartir** las tareas entre dos microcontroladores, ésto

básicamente porque el tiempo de observación y procesamiento es demasiado corto para ser asumido por un solo procesador. La relación es de procesador maestro a procesador esclavo, este último, el 80C452 conectado como un periférico adicional al bus del maestro 8032. Los datos acumulados de los abonados son asegurados con una memoria de respaldo y un detector y corrector de errores independiente.

Aunque la confiabilidad de los circuitos integrados microcontroladores es sumamente alta, es condición que sistemas como éste no lleguen a colapsar bajo ninguna circunstancia, es por eso que además esta Unidad de Control se ha diseñado para trabajar con una gemela en forma totalmente independiente, pero procesando los mismos datos en paralelo (simultáneamente), de tal manera que si una Unidad de Control falla, se tiene la otra que contiene la misma data y sigue manteniendo en funcionamiento normal todo el Telmet.

La Unidad de Control está funcionalmente dividida en dos partes, controlador maestro y controlador esclavo.

•El Módulo del Controlador Maestro está compuesto por:

- Unidad Central de Procesamiento Maestro, ( $\mu$ C Intel 8032).
- Memoria de programa, EPROM 32K bytes, 27C256.
- Memoria datos de abonados RAM 64KB y 64KB de respaldo
- Circuito de corrección y detección de errores 8206.
- Reloj en tiempo real (Real Time Clock) RTC DS1285.
- Circuito supervisor, DS1236.
- Interfaz RS232, MAX232.

•El Módulo de Controlador Esclavo está compuesto por:

- Unidad de Procesamiento Periférico Esclavo ( $\mu$ C Intel 80C452).
- Memoria de programa de esclavo EPROM, 32KB 27C256.
- Memoria de datos temporales RAM, 32K bytes.
- Generador de señales de sincronismo.
- Circuito de redundancia y multiplexaje.

## 2.3 Módulo del controlador maestro

Este es el módulo principal, preparado para manejar el acumulado global de los contadores de tasación de los abonados, así como la comunicación con el exterior. Los datos acumulados son muy importantes. Este módulo es crítico en confiabilidad.

### 2.3.1 Unidad central de procesamiento maestro $\mu$ C 8032

Usaremos como CPU Maestro al  $\mu$ C N8032AH perteneciente a la familia Intel MCS-51. Este microcontrolador es el nexo directo entre el software de control y el gobierno del hardware, básicamente controla los contadores de los abonados. El 8032 incluye dentro suyo, tres temporizadores que permiten cronometrar diversas tareas. También contiene un controlador de interrupciones que permite arbitrar óptimamente muchos de los eventos que deben ser atendidos en tiempo real. Posee además un UART (Universal Asynchronous Receiver Transmitter) incorporado, el cual es usado para interconectarse una consola inteligente. La velocidad del canal serial es programada a 9600 bps, por lo que el cristal del circuito oscilador se fija en 11.0592MHz.

Los componentes externos que interactúan con el microcontrolador (U3 en la Fig. 2.4), a través de sus puertos paralelo P0 y P2 son: Una memoria EPROM que almacena su programa y los periféricos paralelos accesados también por su bus datos y direcciones son listados en la Tabla 2.1.

Mapa de Memoria $\mu$ C Maestro 8032	Desde Hasta
.RAM de abonados vigilado por DCU 8206 U14	0000h FBFFh
.RAM no accesible. RAMs: U10,U11,U12,U13	FC00h FDFFh
.Reloj en Tiempo Real RTC DS1285	U8 FE00h FEFFh
.Procesador Esclavo 80C452 Fig. 2.5:U25	FF00h FFFFh

**Tabla 2.1 Dispositivos conectados al bus del Maestro**

El circuito Supervisor DS1236 (U2 en la Fig. 2.4) **resetea** al Maestro si no recibe señal de STROBE. Y el

Detector de Errores (DCU 8206) sólo interactúa interrumpiendo al Maestro sobre un ERROR, el Maestro averigua si el error es corregible, si CE=1 autoriza a corregirlo, haciendo P1.3 cero (P1.3=CRTC'=0).

El circuito interfaz de comunicación serial que complementa la labores del UART interno del 8032, es el integrado MAX232CPE (U9 en la Fig. 2.4).

### **2.3.1.1 Descripción de señales de control de $\mu$ C maestro**

Las líneas de salida del puerto 0 (P0) y puerto 1 (P1) del microcontrolador son usados como direccionadores de memoria y periféricos externos así como para la entrada de datos (El puerto 0 se comparte como bus de datos y direcciones). El 8032 provee de las señales ALE, PSEN, RD' y WR' para el acceso a datos externos.

ALE (Address Latch Enable) es la señal que, con sus flanco de bajada, obliga al 74HC373 (U4 en la Fig. 2.4) a retener en sus salidas el valor de las direcciones, separándola de los datos del puerto P0.

PSEN' es la señal que fuerza al EPROM (U6) a entregar el byte-código del programa almacenado según la dirección señalada por ADDR BUS (líneas de salida de U4 y U5). PSEN (Program Storage ENable) es activo cuando PSEN'=0.

RD' y WR' son las señales que controlan el acceso a lectura ó escritura de memorias ó periféricos externos a través del del bus de datos.

El puerto P0 y el puerto P2 del Maestro pueden manejar hasta 8 y 4 entradas LS TTL respectivamente, como el número de periféricos a manejar es una carga mayor, optamos por añadir un octal driver 74HC244 (U5 en el esquema circuital Fig. 2.4) para el puerto P2, y un octal transceiver 74HC245 (U7) para el puerto P0. La dirección de acceso del transceiver es manejada por el RD' del 8032, y es puesto en alta impedancia sólo cuando el  $\mu$ C accesa al reloj U8.

Las entradas de interrupción al  $\mu\text{C}$  8032 (U3) son distribuidas como se muestra en la Tabla 2.2.

Interrupt Atiende a

INT0'	U8 Reloj en Tiempo Real(RTC DS1285) U14 Detector y Corrector de Errores de la RAM (DCU 8206)
INT1	U25 Procesador Esclavo (UPI 80C452)

Tabla 2.2 Periféricos asociados a INT0 e INT1

Donde:  $\text{INT0}' = (\text{INT}_{\text{RTC}})' * (\text{ERROR}_{\text{DCU}})'$

$\text{INT0} = \text{INT}_{\text{RTC}} + \text{ERROR}_{\text{DCU}}$

Cuando el Reloj ó el detector de errores interrumpen al procesador, éste discrimina consultando sus entradas P1.0 y P1.1 respectivamente; además tiene la posibilidad de consultar en P1.2 si el error es corregible, y autorizar su corrección haciendo  $\text{P1.3} = \text{CRCT}' = 0$  (Ver Tabla 2.3).

De igual manera, cuando el controlador esclavo interrumpe, P1.5, P1.6 y P1.7 le permiten al maestro discriminar el tipo de solicitud del esclavo.

El pin P1.4 lo usa como salida para generar la señal continuamente variable entre 0 y 1 lógico, el strobe que le indica al circuito (U2) supervisor -watchdog DS1236- que el software del maestro no ha perdido el control.

P3.4 es usado para resetear al procesador esclavo, el control de este Reset es por software.

P3.5 se usa para detectar el estado fallido "battery fail" de la batería (U1) que alimenta las RAM y al Reloj. Esta es una batería inteligente, la DS1260-100 de 960mAH a 3V.

**Tabla 2.3 SEÑALES DE CONTROL DEL PROCESADOR MAESTRO**

SÍMBOLO	PIN#	TIPO	DIRECCION	FUNCIÓN
			PUERTA	
AD0-AD7		IN/OUT	P0	Bus bidireccional para los datos, es compartido con la salida señales de dirección (8 bits menos significativos).
aA8		OUT	P2	Salida de los 8 bits más significativos del el bus de direcciones.
PSEN'	32	OUT		Controla la lectura del programa almacenado.
ALE	33	OUT		Su flanco de bajada sirve, para que con un LATCH externo, se retenga las direcciones, separándola de los datos.
RD'	19	OUT		Controla la lectura en el bus de datos.
WR'	18	OUT		Controla la escritura sobre el sobre cualquier dispositivo en el bus.
INT0'	14	IN	INT0' = INT <sub>RTC</sub> + ERROR <sub>DCU</sub>	Interrupción generada por el RELOJ ó por DETECTOR DE ERRORES. Con "0" se activa el interrup.
INT <sub>RTC</sub> '	2	IN	P1.0	Interrupción generada por el IC REAL TIME CLOC .
ERROR'	3	IN	P1.1	Si se detecta un error durante la lectura RAM, entonces ERROR'=0
CE	4	IN	P1.2	Cuando ERROR'=0 está activo, el CE indica si el error es corregible ó no. CE=0 No corregible CE=1 Corregible
CRCT'	5	OUT	P1.3	CRCT'=0 habilita la corrección de errores, si CRCT'=1 el 8206 trabaja en el modo "solo detectar" errores.
INT1'	15	IN	INT1' = INT <sub>RDIN</sub> +INT <sub>RDOUT</sub> +INT <sub>RD</sub>	Interrupciones del Esclavo al Maestro. Este para discriminar el tipo de interrupción debe observar las puertas P1.5 P1.6 y P1.7 Con "0" se activa el Interrup.
RST SLV	16	OUT	P3.4	Control de RESET sobre el procesador Esclavo. Un RESET válido es un "1" lógico.
BF'	17	IN	P3.5	Indica batería baja. BF'=0 batería baja BF'=1 batería buena.
STROBE'	6	OUT	P1.4	Señal usada para el circuito de WATCHDOG y SF (Sistema Funcional) El período es de 100mseg, el ancho de pulso 1useg una instrucción.
INT <sub>RD</sub>	9	IN	P1.7	El Esclavo está enviando un Immediate Command ó ha aparecido alguna condición de error. INT <sub>RD</sub> =1.
INT <sub>RDOUT</sub>	8	IN	P1.6	El FIFO de salida contiene igual ó más bytes que los especificados como límite para interrupción. INT <sub>RDOUT</sub> =1.
INT <sub>RDIN</sub>	7	IN	P1.5	Interrupción de canal FIFO de entrada, éste requiere data. INT <sub>RDIN</sub> =1.

### 2.3.2 Memoria de programa

Esta es la memoria de sólo lectura que contiene el software del controlador maestro  $\mu\text{C}$  8032. Está conformado por un solo integrado (U6 en la Fig. 2.4) el EPROM 27C256 de 32 KBytes, esta memoria entrega el byte de programa - señalado por las direcciones A0..A14- al bus de datos AD0..AD7, cada vez que su entrada OE' (Output Enable) es puesta a "0" lógico por la señal PSEN' del microcontrolador.

La conexión del Eprom 27C256 con el microcontrolador 8032 es por el address latch U4 74HC373 que separa los datos de las direcciones del P0 del 8032, reteniendo las direcciones según la señal ALE del  $\mu\text{C}$  maestro.

Los ocho bits superiores de dirección viajan a través del octal driver U5 74HC244, pudiendo éste soportar un carga mayor que el P2 del  $\mu\text{C}$  8032.

### 2.3.3 Memoria de datos de abonados RAM

Compuesta por cuatro circuitos integrados Static RAM SRM20256LM12 CMOS de bajo consumo de 120ns de acceso y de 32KBytes cada una. Se distribuyen en 64KBytes (U12+U13) de área de trabajo, y 64Kbytes (U10+U11) de data de respaldo. Las RAM que contienen los datos de respaldo son directamente administradas por la (U14) Unidad de Detección y Corrección de errores DCU (Data Correction Unit), IC 8206-1.

Los 64Kbytes de área de trabajo se usan para almacenar la cuenta acumulada de los abonados y las tablas de plan de tarifas, horarios, feriados, etc., usadas por los procedimientos de software almacenados en EPROM.

Toda la RAM es alimentada por el VCC0 del IC Supervisor DS1236 (U2), lo que les permitirá mantener los datos de los abonados y tablas de configuración en caso de falla de la fuente de alimentación ó retiro de la tarjeta UCTEL del Telmet. En ésta circunstancia, el

DS1236 conmuta la alimentación por batería DS1260 (U1 en la Fig. 2.4) llevando a la memoria al estado DATA RETENTION.

Las señales de acceso para lectura y escritura son OE' (Output Enable) y WE' (Write Enable) respectivamente, ambas están conectadas a los controles RD' (Read) y WR' (Write) del microcontrolador vía circuitos que habilitan e inhiben éstos controles según la dirección de acceso. La señal Chip Select de las RAM está permanentemente activada. Es necesario que CS'=0 y RD'=0 para leer, así también es necesario CS'=0 y WR'=0 para escribir en RAM.

Los circuitos de acceso permiten el direccionamiento de RAM desde 0000h hasta FBFFh. Posiciones superiores son para otros dispositivos, así para:

-escritura en RAM 32KBytes inferior:

$$\overline{WE} = \overline{A_{15}} * \overline{WR'}$$

$$\overline{WE} = \overline{A_{15}} + \overline{WR'}$$
 para escribir ambos deben ser 0

-lectura en RAM 32KBytes inferior:

$$\overline{OE} = \overline{A_{15}} * \overline{RD'}$$

$$\overline{OE} = \overline{A_{15}} + \overline{RD'}$$
 para leer RAM ambos deben ser 0

En la Fig. 2.4, los ICs que forman la RAM en las direcciones más altas (U11 y U13), son parametradas en su acceso por un arreglo de NANDs (U19A, U19B, U20).

-escritura en RAM 32KBytes superiores:

$$\overline{WE} = \overline{A_{15}} * \overline{WR'} * \overline{A_{15} * A_{14} * A_{13} * A_{12} * A_{11} * A_{10}}$$

$$\overline{WE} = \overline{A_{15}} + \overline{WR'} + \overline{A_{15} * A_{14} * A_{13} * A_{12} * A_{11} * A_{10}}$$

$$\overline{WE} = \overline{A_{15}} + \overline{WR'} + \overline{A_{14} * A_{13} * A_{12} * A_{11} * A_{10}}$$

Si  $A_{15}=1$  y  $WR'=0$  las direcciones a acceder están limitadas por el AND lógico de  $A_{14} * A_{13} * A_{12} * A_{11} * A_{10}$ . Así direcciones superiores a partir de FC00h no serán accesadas. Para el caso de lectura, sucede el mismo caso.

-lectura en RAM 32KBytes superiores:

$$\overline{OE} = \overline{A_{15}} * \overline{RD'} * \overline{A_{15} * A_{14} * A_{13} * A_{12} * A_{11} * A_{10}}$$

$$\overline{OE} = \overline{A_{15}} + \overline{RD'} + \overline{A_{15} * A_{14} * A_{13} * A_{12} * A_{11} * A_{10}}$$

$$\overline{OE} = \overline{A_{15}} + \overline{RD'} + \overline{A_{14} * A_{13} * A_{12} * A_{11} * A_{10}}$$

Tabla 2.4

**MAPA DE LA MEMORIA RAM EXTERNA Y DE LOS  
PERIFERICOS QUE MANEJA EL MAESTRO 8032**

Direcciones			
Dec'm.	Hexa.		
65535	FFFF	80C452 (ESCLAVO):	5
65280	FF00		256 BYTES
			Hasta
			Desde
65279	FEFF	RELOJ EN TIEMPO REAL:	64
65024	FE00		256 BYTES
65023	FDFE	RAM NO ACCESIBLE	
64512	FC00		512 BYTES
64511	FBFF	RAM PARA Tx / Rx Y AREA DE TRABAJO	
62784	F540		1728 BYTES
62783	F53F	DIAS FERIADOS	
62656	F4C0		128 BYTES
62655	F4BF	HORARIOS	
62592	F480		64 BYTES
62591	F47F	PLAN DE TARIFAS	
62528	F440		64 BYTES
62527	F43F	RAM LIBRE	
62520	F438		8 BYTES
62519	F437	CONTADORES DE LOS ABONADOS	
			Hasta
00000	0000		62520 BYTES
			Desde

### 2.3.4 Circuito de corrección y detección de errores

Conformado por el circuito integrado DCU (Data Correction Unit) 8206-1 de Intel (U14 en el esquema), dos octal tri-state driver 74HC244 (U15 y U16) y circuitos de lógica discreta para inhabilitación del 8206 en direcciones superiores a partir de FC00h.

El 8206 es un circuito especializado que se encarga de **detectar errores** en RAM, identificar si son corregibles, avisar de esto al  $\mu\text{C}$  maestro, y **si éste** así lo indica, corregir el error.

En el caso de escritura, los datos se almacenan vía U15 en RAM de abonados, en paralelo éstos datos son leídos por el 8206, encriptados en código Hamming y almacenados en cinco bits en la RAM de respaldo vía U16. Ambos drivers salen de su estado de alta impedancia cuando se activa escritura del 8032  $\text{WR}'=0$  en las direcciones permitidas.

En el caso de lectura, el 8206 lee los datos de RAM de abonados y de la RAM de respaldo vía U16 deencriptándola, si ambos coinciden lo entrega al bus de datos en forma normal, **de otro modo** avisa (vía pin  $\text{ERROR}'$ ) con una interrupción de alta prioridad al  $\mu\text{C}$  maestro, avisando además si el error es corregible (pin  $\text{CE}=1$ ). El  $\mu\text{C}$  maestro debe ordenar su corrección (haciendo  $\text{CRCT}'=0$ ) cuando sea posible.

Las entradas  $\text{R/W}'$  y  $\text{BM}_0'$  usualmente en lógico hacen al 8206 trabajar en modo escritura. El 8206 sólo pasa al estado lectura cuando el  $\mu\text{C}$  Maestro hace  $\text{RD}'=0$  en el rango 0000h a FFFFh.

### 2.3.5 Reloj de tiempo real RTC DS1285

Es un IC encargado de mantener la **fecha** y hora, inclusive los años bisiestos. Este reloj de tiempo real RTC (Real Time Clock), U8 en el esquema de la Fig. 2.4, es 1 IC DS1285Q de Dallas Semiconductor. Ofrece un bus **compatible a** Intel (también a Motorola), y su acceso es decodificado por el inversor U17D y el NAND de ocho entradas U23.

$$\overline{\text{CS}} = \overline{\text{A}_0' \cdot \text{A}_2' \cdot \text{A}_{10}' \cdot \text{A}_{11}' \cdot \text{A}_{12}' \cdot \text{A}_{13}' \cdot \text{A}_{14}' \cdot \text{A}_{15}}$$

Esta misma señal es **invertida** con un arreglo de resistencias y transistor (12K $\Omega$  y 5.1K $\Omega$  que **saturan o no** al 2N2907) para poner en alta impedancia al transceiver y

evitar colisión con el bus del Reloj cuando éste es seleccionado.

El DS1285 para su operación necesita de un crystal de 32.768KHz éste conectado al circuito oscilador interno le provee la base de tiempo que necesita el circuito reloj.

Para mantener su funcionamiento en apagado (sin alimentación de 5v), su entrada  $V_{BAT}$  es alimentada por la batería DS1260 (U1), de otra forma perdería los datos de sus 14 registros de las 50 posiciones de memoria que ofrece.

Su salida de interrupción es puesta a 1" lógico con ayuda de una resistencia externa de  $15K\Omega$ , y conectada la entrada INTO y al pin P1.0 del  $\mu C$  8032 maestro. La entrada de interrupción es compartida con el corrector de errores.

### 2.3.6 Circuito supervisor DS1236

Las funciones del IC DS1236 (U2) son supervisar la tensión de alimentación VCC +5V tolerándolo en  $\pm 10\%$ , administrar el circuito de watchdog reseteando al  $\mu C$  maestro 8032 si las condiciones así lo requieren, y conmuta a alimentación por batería llevando a las memorias al estado "Data Retention" cuando VCC sale de su rango ó desaparece. El DS1236 también provee una entrada PBRST' para un pulsador de reset (S1 en el esquema de la Fig. 2.4), éste también genera la activación de la salida RST=1, eliminando los rebotes de un reseteo manual.

VCCI es la entrada monitoreada 5VDC  $\pm 10\%$ , y la salida que alimenta a las memorias estáticas es VCCO.

La entrada CEI' cableada a tierra, está permanentemente activada, y gobierna con CEO' los controles Chip Enable CE' de los 128KB de RAM para llevar a estas memorias estáticas al estado Retención de Datos alimentados por batería en caso de falla ó ausencia de alimentación +5VCC.

La entrada RC = le indica al DS1236 que el procesador que está manejando es NMOS y no un CMOS.

ST' es la entrada watchdog que vigila la presencia de una señal STROBE' (P1.4) que debe ser generada por software del procesador maestro, ésta señal debe presentar un flanco de bajada a lo más cada 100ms para resetear al temporizador del watchdog, ésto será sinónimo que el software del 8032 aún mantiene el control. El mínimo tiempo que el STROBE debe estar en cero es 20ns, de otra forma, el DS1236 reseteará al procesador maestro. Se ha previsto que este evento (reinicialización del  $\mu$ C maestro) se refleje como la inicialización de la Unidad de Control emitiéndose un reporte al mundo exterior.

En caso de reset sobre el maestro, la señal de RST generada por el supervisor (U2 en Fig. 2.4) es mantenida durante 100mseg -mínimo 25mseg- tiempo suficiente para un RESET efectivo del  $\mu$ C 8032, el cual especifica que basta mantener dos ciclos de máquina ( $\sim 2\mu$ s) en alto su entrada RESET para una reinicialización efectiva.

La señal RST también maneja la base del transistor darlington MPS A64 para tenerlo en saturación, manteniendo prendido al LED D1 en operación normal; éste se apagará mientras un RESET del supervisor IC DS1260 (U2) sobre el procesador maestro.

### 2.3.7 Bateria de respaldo

Es una batería de litio, el DS1260, U1 en la Fig. 2.4, que da respaldo a los bancos de memoria RAM así como al Reloj en Tiempo Real cuando no hay alimentación VCC. Esta batería tiene una capacidad de 960MAH, que considerando el consumo de las cargas en standby, puede llegar a retener los datos de las RAM y mantener la fecha y hora del reloj hasta en más de 10 años.

Consumo cada IC RAM SRM20256L12: 2.0 $\mu$ A Típico

Consumo máximo del RTC DS1285Q: 0.5 $\mu$ A

**Consumo total:**  $4 \times 2.0 + 0.5 = 8.5\mu$ A

Capacidad batería

Tiempo de respaldo =  $\frac{\text{Capacidad batería}}{\text{Consumo total}}$

Tiempo de respaldo =  $960\text{mAh}/8.5\mu\text{A}=12.9$  Años

De otro lado, se aprovecha la salida BATTERY FAIL (BF) que va cero lógico cuando la batería se ha consumido. También  $\text{BF}'=0$  cuando la energía es suplida por la batería de litio.

### 2.3.8 Interfaz de comunicación serial

El  $\mu\text{C}$  8032 posibilita fácilmente la comunicación serial, pues viene preparado con un full duplex UART, el cual es programable por software para funcionar en diferentes modos y velocidades. El uso de éste UART requiere de tres pequeñas secciones de software:

- a) Sobre el encendido-inicio ó reset de hardware, el puerto serial y las palabras de control sobre el temporizador de velocidad de transmisión, deben ser inicializados con valores apropiados.
- b) Una rutina para cargar el byte-dato a ser enviado en el registro **buffer** SBUF, antes de ser transmitido vía puerto serial.
- c) Otro **rutina** es necesaria en la recepción para recoger el byte-dato que llegan en el registro SBUF.

SCON y SBUF son los registros de función especial (SFR) usados para éste fin. SCON es el Registro de Control de Puerto Serial, y SBUF es el buffer donde se carga el dato enviar ó de donde se lee el dato recibido. El UART trabajando en modo 1 (1 bit de inicio, 8 de datos y 1 de parada), usará el Temporizador 2 del  $\mu\text{C}$  8032 para **generar el** Baud Rate, siendo ésta función de los registros RCAP2High y RCAP2Low según **la ecuación:**

$$\left[ \text{Baud Rate}_{\text{TIMER2}} = \frac{\text{Frecuencia de Oscilación}}{32 * [65536 - (\text{RCAP2H}, \text{RCAP2L})]} \right] \text{Ecuac. 2.1}$$

La velocidad de transmisión/recepción, es programable por software, pero tiene dependencia de la **frecuencia de**

oscilación del cristal del  $\mu\text{C}$  Maestro, así, para la velocidad con la que trabajaremos (9600bps) y con la frecuencia de oscilación de 11.0592MHz resultan RCAP2H=FFh y RCAP2L=0DCh.

Para completar el circuito de comunicación serial, se usa el circuito integrado MAX232CPE. Este es un interfaz driver/receiver de niveles TTL (5v y 0v) a niveles RS232 (-12v a -3v y +3v a +12v). Este circuito integrado MAX232C, hace la tarea que comunmente era asumida por los circuitos integrados tradicionales 1488 y 1489, con la ventaja de no necesitar las fuentes de  $\pm 12\text{v}$ , sino sólo la de 5v, involucrando un ahorro en el costo de la fuente de alimentación. El MAX232C internamente posee un doblador de voltaje de +5v a +10v y un inversor de +10v a -10v, permitiéndole una salida de  $\pm 9\text{v}$ . En la recepción pueden manejar niveles de hasta  $\pm 30\text{v}$ .

Los dobladores e inversores de tensión del del IC MAX232 son completados con condensadores externos. Este hardware habilita así la conexión al puerto serial de cualquier PC compatible.

#### **2.4 Módulo del controlador esclavo**

Las funciones de este módulo (ver Fig. 2.5) son leer las líneas de datos SOA0A...SOA9A y SOA0B...SOA9B provenientes de los Amplificadores de Sensado (Sense Amplifier), estos datos son interpretados según las señales de ISCA', HASCA e INSCA también generadas por éste módulo. En este módulo también se generan las señales de reloj, control y sincronismo que necesita el Telmet. Lo más crítico en este módulo son los tiempos de observación y procesamiento de la datos del estado de los abonados.

#### **2.4. Unidad de procesamiento periférico $\mu\text{C}$ 80C452**

Este procesador será el encargado de preprocesar los datos de los abonados antes de entregarsela al Maestro  $\mu\text{C}$  8032. El procesador periférico que usaremos es un micro-

controlador diseñado para ser un periférico inteligente, el  $\mu\text{C}$  UPI 80C452 de Intel (Universal Peripheral Interface UPI), en adelante procesador Esclavo (U25 en la Fig. 2.5). Este UPI posee una arquitectura interna similar al del Maestro 8032, con un bus adicional orientado al maestro. Contaremos entonces con un bus del maestro y un bus local (bus del esclavo) con su propia RAM y ROM.

La interacción del 80C452 con su bus local es simple, sólo interactúa con una RAM estática (U27 en la Fig. 2.5) y un EPROM (U28), ayudándose con un latch de direcciones 74HC373 gobernado directamente por su ALE (Address Latch Enable).

#### **2.4.1.1 Señales de control del $\mu\text{C}$ esclavo**

El procesador Esclavo es encargado de leer los datos de los abonados que ingresan por sus entradas P1.0...P1.7, P4.0 y P4.1, ya multiplexadas según HASCA e interpretadas según ISCA', HASCA e INSCA. Aquí controlaremos según el plan de tarifas si el abonado que inicia una conversación conversa el intervalo de tiempo mínimo ó más intervalos de tiempo, avisando de esto al procesador Maestro.

El 80C452 puede solicitar atención del Maestro interrumpiéndolo (tres líneas interrupción), y éste puede comunicarse con el Esclavo mediante su bus de direcciones y datos, leyendo ó escribiendo sobre los registros del Esclavo. Este también ofrece un canal FIFO de comunicación hacia el Maestro, todo a través del Bus Maestro. Se usa U24 un NAND 74HC30 para activar el CS' del esclavo. En la Fig. 2.5 (esquina superior izquierda) se pueden ver todas las líneas de interacción Maestro-Esclavo.

La Tabla 2.5 muestra en detalle las señales de control que maneja el procesador Esclavo.

Para satisfacer la necesidad de hacer más rápida la operación del 80C452, usamos un cristal de 12Mhz (los tiempos son críticos, se debe leer y preprocesar 10 abo-

**Tabla 2.5      S ÑA L S DE CON RO    DEL PROCESADOR ESCLAVO**

SIMBOLO	PIN#	TIPO	DIRECCION	FUNCION
INT <sub>ROIN</sub>	49	OUT		Interrumpe al procesador Maestro cuando el canal FIFO de entrada requiere data.
INT <sub>ROOUT</sub>	48	OUT		Interru pe al procesador Maestro cuando el canal FIFO de salida contiene igual ó mas bytes bytes que los especificados como límite.
INT <sub>RO</sub>	50	OUT		Interru pe al procesador Maestro cuando sale un Immediate Command ó aparece alguna condición de error.
CS'	44	IN		Habilita el acceso de lectura ó escritura sobre ciertos recursos del procesador Esclav. Visto por el Maestro sólo en las direcciones FFXXh
A2A1A0	42/41/40	IN		Direcciones con las que el procesador Maestro accesa a los canales FIFO y a los registros SFR del 80C452.
RST SLV	20	IN		El RESET puede ser activado por el P3.4 (T0) del procesador Maestro. Un RESET es con una duración por lo menos 2 ciclos de máquina.
CTRL LED	62	OUT	P3.5	Con 1/0 se prende/apaga al LED de "Esclavo en buen funcionamiento".
ISCA'	65	IN	INT0'	Interrumpe al Esclavo cada 480µs, indica que hay data de un nuevo grupo de 10 abonados.
ISCA'	35	IN	P4.5	Idem.
HASCA	36	IN	P4.6	Señal periódica que se presenta como "0" durante los primeros 5000 abonados y durante los últimos 420, y como "1" durante los 5000 intermedios. Duty cycle: 240ms/500.16ms
INSCA	37	IN	P4.7	Nor al ente en "0" lógico, se activa durante 144µs sólo al inicio del primer y segundo grupos de 5000 abonados.
3SL	32	IN	P4.2	Indica el estado tri-state (local) de la salida de las señales de reloj y sincronismo que salen como data de las memorias U40, U41, U42 y U43 locales. 3SL=0 Nor al(Activado)      3SL=1 Alta impedancia (Desactivado).
3SE	33	IN	P4.3	Indica el estado tri-state (remot) de la salida de las memorias de la Unidad de Control UCTEL100 gemela que trabaja en paralelo. 3SE=0 Normal(Activado)      3SE=1 Alta impedancia (Desactivado).
SL	34	IN	P4.4	Selecciona las señales periódicas ISCA, HASCA, INSCA y CPO' locales ó externas. Indica si se va a sincronizar local ó externamente. SL=1 Local, SL=0 Externo.
SAL	63	OUT	P3.4	Sistema Local Activado:    SAL=1 Activado.    SAL=0 Desactivado.
S00..S09		IN		P1.0-P1.7 Cada pin corresponde al estado de un abonado, éstos ofrecen P4.0-P4.1 señal 1/0 según su estado ocupado/libre.

nados en menos de 480µs), no interesa otro valor de cristal, pues no usamos el UART del Esclavo.

Para visualizar el **correcto** funcionamiento del Esclavo, hacemos que éste gobierne a un LED D2 manteniéndolo en un prende-apaga visible (flashing). Una indicación de que el Esclavo mantiene el control de sus acciones, es vigilada por el Maestro, dándole un tiempo para que el Esclavo le indique su estado de actividad, de lo contrario el 8032 activará el pin de RESET (RST SLV) del Esclavo. Estas acciones son realizadas por software.

El 80C452 también puede trabajar como controlador de DMA (acceso directo a memoria) transfiriendo los datos de su bus local vía FIFO al bus maestro y viceversa, ésta opción no es necesaria en nuestro sistema, por lo que la deshabilitamos, haciendo su pin DACK'=1.

#### **2.4.2 Memoria de programa del esclavo**

Es una memoria EPROM de 32KBytes que contiene el software del controlador Esclavo µC 80C452. Está conformado por un solo integrado (U27 en la Fig. 2.5) el EPROM 27C256 de 120ns de acceso, ésta memoria entrega el byte de programa -señalado por las direcciones SA0..SA14- al bus de datos SADO..SAD7, cada vez que su entrada OE' (Output Enable) es puesta a "0" lógico por la señal PSEN' del microcontrolador esclavo.

La conexión de ésta Memoria 27C256 con el microcontrolador 80C452 es similar a la del Eprom U6 con el µC 8032. El cableado de los ocho bits superiores del bus de direcciones es directo.

#### **2.4.3 Memoria de datos temporales externa RAM**

Esta compuesta por una RAM estática externa (U28 en la Fig. 2.5), el IC SRM20256LM12 de 32KBytes, directamente conectada al Esclavo. Esta RAM no es respaldada por batería, pues es más usada como área de trabajo donde los datos cambian permanentemente (a diferencia de la RAM del **Maestro que es** más usada para almacenar los **datos** acumulados). Esta RAM está

direccionada en los primeros 32K y se activa su CS' con SA15=0 (Slave fddress 15).

#### 2.4.4 Generador de señales de sincronismo

Este módulo se encarga de generar todas las señales de sincronismo que necesita el **Telnet partiendo** de los datos contenidos en un banco (U39, U40, U41, U42 y U43) de memorias EPROM 27C256L que son leídos secuencialmente por los ICs U37 y U38 contadores binarios 74HC4040 y 74HC4024 y cuyas salidas sirven de direccionador a las EPROM, éstas deben ser leídas cada 24us. Así, los contadores binarios tienen a su vez su entrada de reloj CLK' marcada cada 24us por la salida del IC divisor de frecuencia 74HC4059, éste a su vez tiene como base de tiempos a un oscilador (U35 en Fig. 2.5) de 1MHz.

El divisor de frecuencia 74HC4059, fija su salida en función de la conexión a "0" ó de sus entradas J1...J16, y del MOD0 de operación programado con sus entrada de selección Ka, Kb y Kc. Como necesitamos una salida con pulsos CLK de periodo 24us y el oscilador ofrece un pulso por cada 1us (f=1MHz), el divisor debe ser programado en **n=24**. Esto nos permite ingresar a la fórmula del divisor así:

$$n = \text{MOD0} \cdot (1000 \cdot D5 + 100 \cdot D4 + 10 \cdot D3 + D2) + D1 \quad \text{Ecuac. 2.2}$$

Si definimos MOD0=5, entonces podemos hacer D5=D4=D3=0, con lo que la Ecuac. 2.2 queda reducida a  $n=24=5 \cdot (D2) + D1$ , de donde podemos hacer: D2=4 y D1=4. La Tabla 2.6 muestra como quedan las conexiones de las entradas de divisor de frecuencia para una salida de periodo 24us.

MOD0	D1	D5	D2	D3	D4
5	4	0	4	0	0

LEKaKbKc	J1	J2	J3	J4	J5	J6	J7	J8	J9	J10	J11	J12	J13	J14	J15	J16
0101	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0

Tabla 2.6 Programación del Divisor de Frecuencia 74HC4059

Así, con ésta señal de reloj hacemos avanzar a los contadores binarios 74HC4040 y 74HC4024. Estos en cascada manejan las direcciones de acceso a las EPROM U39 a U43 (Fig. 2.5). La primera de éstas EPROM cuando las direcciones alcanzan un valor determinado (20840=5168h) se genera una señal RST que modo de realimentación vuelve a los contadores al estado de RESET en sus cuentas y sus salidas apuntan otra vez a la dirección 0000h.

El primer EPROM U39 se encarga de crear las señales locales ISCAL', HASCAL, INSCAL, RST e ISCAL. Con esta última, y ayudada por los monoestables HC4538, crea otras señales más pequeñas: CPO'L, EU3L y KLL cuyas "ventana de tiempo" más pequeñas que las otras hace preferible crearlas con monoestables, así tenemos:

$$t_{\text{SALIDA DE MONOESTABLE}} = 0.7 \cdot R \cdot C \quad \text{Ecuac. 2.3}$$

Donde R: Resistencia y

C: Capacidad conectados a monoestable

Ancho de pulso en Ec.2.3:  $CPO'L = 0.7 \cdot 20K\Omega \cdot 330pF = 4.62\mu s$

CPO'L, EU3L y KLL pasando através del 74HC373 U46, así como el resto las otras salidas de los EPROMs, van a salir hacia la Matriz de Selección de Línea RI y al Amplificador de Sensado SA vía los drivers sumidero de corriente UDN2595A (U47 a U51 en la Fig. 2.5) capaces de manejar corrientes de hasta 100mA por puerto, simultáneamente cuando la salida de los EPROMs está en tierra.

La salida de los EPROMs generadores de señales U40 a U43 y del 74HC373 U46 pueden ser deshabilitadas (señal 3SL) cuando el control de éstas señales es asumido por la Unidad de Control gemela y viceversa.

#### 2.4.5 Circui o de redundancia y multiplexaje

El Subsistema Interfaz de Línea del Telmet, vía el módulo /mplificador de Sensado (Sense Amplifier), entrega a la Unidad de Control veinte señales S0A0A...S0A9A y S A0 ...S A9 , que son datos en tiempo real del estado de

los abonados. Estas veinte líneas son multiplexadas en diez para ser procesadas por el CPU esclavo 80C452 según el valor de HASCA:

HASCA Habilita lectura de los 74HC373:

0 U29 y U30

1 U31 y U32 Hasca es invertido por U17F

En ambos casos, el flanco de bajada de CPO' (4.62us  $\approx$  5us después del inicio de cada ISCA) es el que permite la captura (latching) de los datos en los 74HC373, y en ambos casos sus entradas están conectadas a tierra con un conjunto de resistencias, indicando el estado del abonado en reposo (abonado que no genera llamada). Así, el procesador Esclavo considerará efectivo la conversación por originante cuando las salidas de los latches S00...S09 sean "1" lógicos.

De otro lado, el circuito de redundancia coordina las señales de las dos Unidades de Control trabajando en paralelo para que no haya conflicto en el tratamiento de las señales como en el sincronismo. Aquí también tenemos un módulo multiplexor, el U33 74HC157 con cuatro mux de 2x1. El circuito de redundancia hace que las dos UCTEL trabajen con las señales de control y sincronismo (ISCA', HASCA, INSCA y CPO') de sólo un único UCTEL al mismo tiempo, para llevar ambas la misma tratativa de los datos. Así, si una UCTEL está activada, ésta genera el sincronismo; si falla ó pierde el control del STROBE', el circuito de redundancia transferiría el gobierno de las señales de sincronismo a la UCTEL de respaldo, conmutando su mux 74HC157 de Externo (A'/B=0) a Local (A'/B=1). En el UCTEL fallido sucedería lo contrario. Esta conmutación del gobierno de las señales de reloj y sincronismo, también es válida para las líneas tri-state local 3SL y tri-state externo 3SE cableadas a los OE' Output Enable de los EPROMs generadores de señales de reloj.

En el diagrama esquemático Fig. 2.5, nuestro control 3SL (tri-state local) no deshabilita al primer EPROM U39, ésto permite mantener permanentemente las señales locales

ISCAL', HASCAL, INSCA, RST e ISCA, que marcarían el temporizado si ésta unidad trabajando de respaldo tuviese que asumir el control por falla de su gemelo principal.

El U45B monostable HC4538 está siendo redispuesto permanentemente a "1" por el flanco de bajada del STROBE' generado por el Maestro. Si en un tiempo igual a  $0.7 \cdot 110K\Omega \cdot 3.3\mu F \approx 254.1ms$  no se presenta el STROBE , la salida Q=SFL se hace "0" provocando:

- a) 3SL=1 deshabilita EFRoMs locales, y
- b) SFE=0 en UCTEL de respaldo, obligando su mux (74HC157 A'/B=1) a que tome referencias locales.

En resumen, en éste capítulo hemos delimitado las condiciones de temporización que debe respetar la Unidad de Control para interactuar con el resto del equipo. Para cumplir con éstos tiempos, se ha repartido la carga de trabajo en dos microcontroladores, cuidando que los datos en RAM de abonados estén bien protegidos. Sobre esta plataforma de hardware se desarrollará software para dos procesadores. El uso de los microcontroladores y otros componentes integrados que realizan funciones complejas y especializadas facilitarán el desarrollo del software.

## CAPITULO III DISEÑO DE SOFTWARE

En este capítulo estableceremos la formulación del diseño de software de la Unidad de Control (UCTEL), así como las consideraciones que se establecieron para su desarrollo.

En el diseño de Hardware, acápite 2.2, establecimos un sistema basado en doble procesador, además de redundante, por el tiempo de observación y procesamiento que exigían ser muy rápidos, siendo más crítico para el procesador esclavo, porque éste es el que leerá el estado de los abonados en las líneas S00...S09 según la interrupción ISCA' y señales HASCA e INSCA. El Esclavo medirá el consumo inmediato del servicio para cada abonado según el plan de tarifas que el Maestro le alcance y reportará este consumo al Maestro. El Esclavo también es responsable de discriminar un falso inicio ó término de conversación del originante. El tiempo que dispone para observar los 10420 abonados es de 500.16ms, cumpliendo su objetivo en dos de éstos ciclos.

El Maestro en cambio es el responsable de administrar el plan de tarifas, registrar el consumo del servicio, interactuar con el exterior vía su puerto serial, supervisar la operatividad del Esclavo, de la RAM, etc.

Estableceremos primero la estructura del plan tarifario por pulsos, el espacio de almacenamiento y procesamiento para los datos de los abonados, así como el espacio para las tablas del sistema. Luego veremos la interacción del procesador Esclavo con el Maestro para la ejecución del plan tarifario por pulsos, además del diseño y consideraciones para el protocolo de comunicación serial con las consolas local y remota, etc.

### 3.1 Estructura del plan de tarifas

El Plan de Tarifas (PT) consta de ocho tarifas diferentes, según la configuración de la Tabla 3.1.a.

PT <sub>i</sub>	1er Intervalo		2do Intervalo	
PT <sub>0</sub>	n <sub>01</sub>	c <sub>01</sub>	n <sub>02</sub>	c <sub>02</sub>
PT <sub>1</sub>	n <sub>11</sub>	c <sub>11</sub>	n <sub>12</sub>	c <sub>12</sub>
PT <sub>2</sub>	n <sub>21</sub>	c <sub>21</sub>	n <sub>22</sub>	c <sub>22</sub>
PT <sub>3</sub>	n <sub>31</sub>	c <sub>31</sub>	n <sub>32</sub>	c <sub>32</sub>
PT <sub>4</sub>	n <sub>41</sub>	c <sub>41</sub>	n <sub>42</sub>	c <sub>42</sub>
PT <sub>5</sub>	n <sub>51</sub>	c <sub>51</sub>	n <sub>52</sub>	c <sub>52</sub>
PT <sub>6</sub>	n <sub>61</sub>	c <sub>61</sub>	n <sub>62</sub>	c <sub>62</sub>
PT <sub>7</sub>	n <sub>71</sub>	c <sub>71</sub>	n <sub>72</sub>	c <sub>72</sub>

**Tabla 3.1.a PLAN DE TARIFAS**

Donde:

PT<sub>i</sub> = Plan de Tarifas i-ésimo.

n<sub>11</sub> = Tiempo en segundos del 1er Intervalo Programable de 0-4095 seg.

c<sub>11</sub> = # de Pulsos de Cómputo (x100) del 1er Intervalo de Tiempo. Programable de 0 a 4095 pulsos.

n<sub>12</sub> = Tiempo en segundos de los Intervalos subsiguientes. Programable de 0 a 4095 segundos.

c<sub>12</sub> = # de Pulsos de Cómputo (x100) de los Intervalos Subsiguientes. Programable de 0 a 4095 pulsos.

Cuando un abonado inicia una conversación de una llamada, se le "cobra" c<sub>11</sub> pulsos por los primeros n<sub>11</sub> segundos, aún si cuelga antes de los n<sub>11</sub> segundos; si continúa conversando se le "carga" c<sub>12</sub> pulsos dándole un plazo de n<sub>12</sub> segundos por éste intervalo; para los intervalos subsiguientes se repite con c<sub>12</sub> con un plazo de n<sub>12</sub>. La Tabla 3.1.b es una tabla ejemplo:

PT <sub>i</sub>	1er Intervalo		2do Intervalo	
PT <sub>0</sub>	60	1200	60	400
PT <sub>1</sub>	120	1200	60	400
PT <sub>2</sub>	120	900	60	300
PT <sub>3</sub>	180	900	60	300
PT <sub>4</sub>	180	600	60	200
PT <sub>5</sub>	180	450	60	150
PT <sub>6</sub>	180	360	60	120
PT <sub>7</sub>	180	300	60	100

**Tabla 3.1.b PLAN DE TARIFAS**

Por los primeros 3 minutos se "cobra" 600 pulsos, por cada minuto adicional 200 pulsos

Por los primeros 3 minutos se "cobra" 300 pulsos, por cada minuto adicional 100 pulsos

Podemos observar, por ejemplo, que el PT6 es 20% más elevado que el PT7. Que PT5 es 50% más elevado que el PT7. El PT3 es 50% más que el PT4. El PT2 es igual en número de pulsos al PT3, pero la bajada de bandera se ha reducido a 120seg.

Para asignar los PT a las diferentes horas y días de la semana se tendrán la Tabla 3.2.a.

**Tabla 3.2.a**  
**HORARIOS POR**  
**TIPO DE DIA:**

DIA 0		DIA 1		DIA 2	
h00	PTp	h10	PTr	h20	PTw
:	:	:	:	:	:
:	:	hij	PTx	:	:
:	:	:	:	:	:
h07	PTq	h17	PTs	h27	PTz

**Lunes a Viernes      Sábados      Domingos**

Donde: hij = Es la hora de comienzo del horario j del día i (i=0,1,2)

PTx = Es el Plan de Tarifa asignado a partir de la hora hij

En total puede haber hasta 8 horarios por día y 3 días diferentes. El primer día (DIA 0) será asignado de Lunes a Viernes, el segundo día (DIA 1) será asignado a los Sábados y el tercer día (DIA 2) a los Domingos.

Como ejemplo, consideremos la siguiente Tabla 3.2.b:

**Tabla 3.2.b**  
**HORARIOS**  
**POR TIPO**  
**DE DIA**

	DIA 0		DIA 1		DIA 2	
hi0	00	7	00	7	00	7
hi1	07	4	07	6	08	6
hi2	10	0	10	4	22	7
hi3	11	3	17	6	99	99
hi4	15	4	22	7	99	99
hi5	17	2	99	99	99	99
hi6	20	5	99	99	99	99
hi7	22	6	99	99	99	99

**Lunes a Viernes      Sábados      Domingos**

En el ejemplo tenemos que para el Día 0, a partir de las 00 horas se aplicará el Plan de Tarifa 7, a partir de

las 07 hrs el PT4, de las 10 hrs hasta las 11 hrs el PT0, y así sucesivamente.

Se completarán los datos con la tabla de feriados

		DF	..
<b>Tabla 3.3.a DIAS FERIADOS</b>			
	DF00		DDMM00
	DF01		DDMM01
Donde:	..		.
DFi = Día feriado i (i=00-31)	DFi		DDMMi
DDMMi= Día y Mes del día feriado i	..		.
	DF31		DDMM31

Son hasta 32 días feriados.

Nota.- El Día Feriado usará el mismo Plan de Tarifario PT que el de los Domingos.

Como ejemplo consideraremos los feriados en el Perú del año 1994, la tabla sería:

		DF	..
<b>Tabla 3.3.b DIAS FERIADOS</b>			
Primer de Enero	DF00		0101
	DF01		3103
	DF02		0104
	DF03		0105
	DF04		2406
	DF05		2906
	DF06		2807
	DF07		2907
	DF08		3008
	DF09		0807
	DF10		2907
	DF11		0111
	DF12		0812
25 de Diciembre	DF13		2512
	DF14		9999
	.		.
	.		.
	DF31		9999

La selección del Plan Tarifario se detalla más adelante, en el acápite 3.4.2.2, Fig. 3.4.

### **3.2 Descripción general de funciones**

El software es desarrollado orientándose fundamentalmente a la tasación de las llamadas, casi todos los eventos deben ser tratados en tiempo real y la ejecución del software debe estar adecuadamente distribuida en el tiempo, de acuerdo al orden de prioridades de eventos a procesar.

Así, la tarea básica de tasación es compartida entre ambos procesadores. El Esclavo efectúa la lectura del estado de los abonados y temporiza éstas llamadas, avisando al Maestro de los abonados que culminan sus intervalos ( $n_{11}$   $n_{12}$ ) de conversación. El Maestro incrementa el número de pulsos ( $c_{11}$   $c_{12}$ ) del PT vigente a los registros acumuladores del abonado respectivo.

#### **3.2.1 Funciones del procesador maestro 8032**

##### **.Sobre el encendido:**

- Inicializa recursos propios y periféricos de su entorno, registros de control en RAM interna y externa, Timers, UART, Reloj, procesador Esclavo, etc.

##### **.A ención de In erupciones Prioritarias:**

- Atiende interrupciones del Reloj DS1285 para cuando sea necesario seleccionar el Plan de Tarifa ( $c_{i1}, n_{i1}$  y  $c_{i2}, n_{i2}$ ), y ordena al Esclavo actualizarse con éste cambio.

- Sobre un error en RAM de abonados, atiende al IC Detector de Errores y si es posible, habilita su corrección. Informa de éste evento y la acción tomada al computador externo.

- Atiende las solicitudes de atención del procesador Esclavo.

##### **.Manejo de Con adores**

- Inicializa contadores por petición externa.
- Incrementa contadores por tasación señalada por procesador Esclavo.

-Modifica contadores por petición externa.

-Corrige contador cuando se detecta error.

#### .Comunicaciones con el Exterior.

-Maneja protocolo de comunicaciones.

-Recibe mensajes (genera tareas).

-Envía mensajes (resultado de otras tareas ó eventos).

-Convierte hexadecimal en ASCII.

-Convierte ASCII en hexadecimal.

#### .Supervisión

-Maneja supervisor de todo el programa

-Vigila la correcta operativa del controlador Esclavo.

-Genera una señal de STROBE sobre el circuito de watchdog DS1236, éste con un LED indica el funcionamiento del sistema.

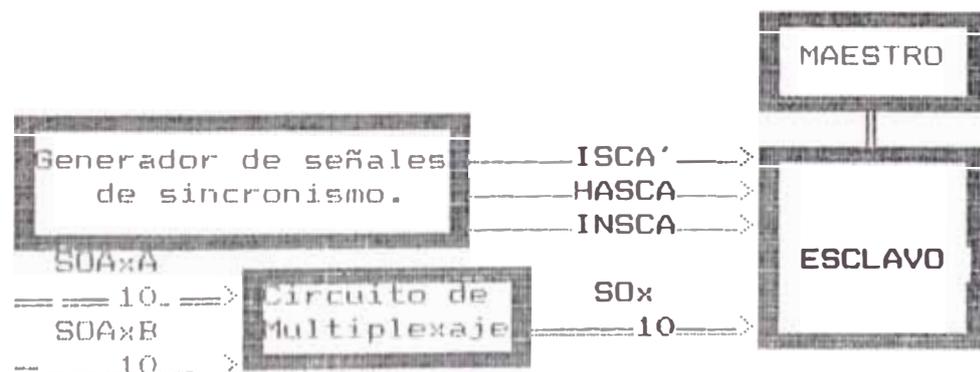
Prendido: Sistema operando correctamente.

Oscila : Sistema en mal estado.

Apagado : No hay +5V<sub>DC</sub> o éste es defectuoso.

### 3.2.2 Funciones del procesador esclavo 80C452

El Esclavo recibe las siguientes señales:



ISCA' señal de interrupción, pte. 24µs cada 480µs para leer un grupo de 10 abonados por 500...509.

HAS A Indica si la cuenta se encuentra o no en los segundos 5000 abonados.

INSCA Indica el inicio de los primeros 5000 ó de segundos 5000 abonados.

**S0x10** Son diez líneas (S00...S09) que indican el estado Libre/Ocupado de los 10 abonados observados en la ventana ISCA'.

En base a estas señales, podemos definir las funciones de procesador Esclavo, como sigue:

**.Sobre el encendido:**

-Inicialización de recursos internos, timers, canal FIFO, registros de control, RAM interna, inicializa contadores en RAM externa, etc.

**.Lectura de estado de abonados**

-Se efectúa la lectura de S00...S09 sobre cada interrupción ISCA'.

-Supervisa el correcto sincronismo para validar la lectura sobre los primeros ó segundos 5000 abonados.

-Calcula el número del abonado (10 veces).

-Verifica la condición de Libre/Ocupado (10 veces).

**.Incrementa con adores**

-Atiende interrupción cada 1000ms.

-Verifica bit de ocupado

-Incrementa contador de número de abonado.

-Temporiza si el abonado alcanzó un tiempo de conversación ni1 ó ni2.

-Avisa al maestro el número del abonado que inicia una conversación, del abonado que ha cumplido una cuenta ni1 ó ni2.

**.Supervisión**

-Vigila por la ejecución de tareas de menor prioridad.

### 3.3 Módulos del sistema - diagramas de flujo

El principio que manejaremos será que el procesador Maestro y Esclavo tendrán un programa base llamado "Supervisor" que estará explorando secuencialmente en un bucle ó nido, las tareas que soliciten su atención obser-

vando si su bit bandera "Bit de Ejecución" de estas tareas está activado.

Para considerar los varios grados de urgencia asegurar que no se pierda ninguna información externa y que las tareas sean llevadas en un orden correcto de prioridad, los programas serán agrupados en varios niveles de interrupción, con programas de alto nivel a los cuales se les permite interrumpir la ejecución de un programa de menor nivel.

Así, cuando ocurra una interrupción, el programa corrientemente en ejecución es detenido, y los contenidos de de los registros usados por el programa son guardados. El código de interrupción es ejecutado, luego los contenidos son restaurados y el control es retornado al programa interrumpido.

### **3.3.1 Programa supervisor e interrupciones del $\mu$ C maestro**

Sobre el Power On, empiezan a ejecutarse las rutinas de inicialización de registros y RAM interna del maestro, el UART, el reloj, el corrector de errores, resetea al esclavo y hace un test de la operatividad de éste. Entra luego a un bucle permanente denominado SUPERVISOR.

El programa Supervisor cumple dos tareas básicas:

a) Genera la señal de STROBE complementando la salida (Port.Pin) P1.4 para el circuito WATCHDOG Supervisor DS1236.

b) Pregunta secuencialmente por el Bit de Ejecución  $T_{NN}$  de la tarea NN-sima; si encuentra activado éste bit ( $T_{NN}=1$ ), ejecuta la tarea asociada, y cuando la termina desactiva su BE ( $T_{NN}=0$ ).

La ejecución de algunas tareas puede activar el BE de otras tareas, ésta últimas serían atendidas cuando el Supervisor pregunte y encuentre activado su BE respectivo.

Las tareas más urgentes ó que requieren atención más inmediata, han sido separadas del bucle Supervisor, y son

ejecutadas atendiendo a la interrupción que las provoque. Así, requieren prioridad de atención al siguiente orden:

```
.INT0  Error de memoria RAM externa e      Mayor prioridad
        Interrupción del RTC DS12850
.INT1  Esclavo envia data a Maestro
.TO    Timer0 Vigila actividad de Esclavo
.T1    Timer1 Temporiza intentos de Tx Serie  V
.Tx/Rx Interrupción de canal serial      Menor prioridad
```

**INT0:** Están asociadas a INT0 dos tareas. La tarea considerada de mayor prioridad es la detección y/o corrección de un error de memoria. Esto es para no perder la dirección que el puntero de RAM (DPTR) tiene en ése instante. También es considerada prioritaria la interrupción del Reloj para que a) A través del Maestro, el Esclavo temporice a los abonados que efectúan llamada, y b) El Maestro se actualice con el plan de tarifa con el cambio de hora y envíe los nuevos intervalos de temporización al Esclavo.

**INT1:** La lectura de la información que el Esclavo entrega al Maestro también es una tarea urgente y de importancia, por lo que esta tarea se ejecuta como atención a la INT1, interrupción del FIFO del Esclavo. El Maestro debe analizar el mensaje, y según lo que lee, debe proceder como sigue:

\*Si un abonado empieza una llamada, el Maestro debe cargarle en su cuenta el número de pulsos ni1 del PT vigente.

\*Si un abonado ha cumplido su intervalo de tiempo y sigue conversando, se le carga el número de pulsos ni2 del PT en vigencia.

\*Si el abonado termina de conversar, éste aviso le puede servir al Maestro para Reportar a la consola este evento.

Existen otros mensajes de Esclavo a Maestro, pero los mencionados son los más importantes.

**TIMER0:** Vigila la actividad del Esclavo. Si el Esclavo no se comunica con el Maestro en un lapso de 5 segundos, se

reseteará al Esclavo, se envía un mensaje de alarma Consola y se permite la inicialización del Esclavo.

**TIMER1:** Temporiza los intentos de retransmisión serial. Si un mensaje enviado a consola no recibe respuesta en el lapso de un segundo, se reintenta la transmisión del mensaje.

**SERIAL PORT INT:** La atención a la Tx/Rx serial es la tarea de menor prioridad dentro de las urgentes. Esta tarea atiende las interrupciones del UART del Maestro toda vez que se transmite ó recibe información por el canal serial.

### **3.3.1.1 Tareas del programa supervisor del maestro**

Las tareas del programa Supervisor son ejecutadas cuando éste encuentra activado su bit de ejecución BE. El programa Supervisor explora según la relación siguiente:

#### **BE TAREA**

- T05D Selecciona plan de tarifas.
- T08D Envía mensaje al Esclavo con intervalos de tiempo nil ni2
- T10D Protocolo de recepción serial, discrimina el BE T<sub>NND</sub> a setear según código de recepción serial.
- T11D Conversión de la recepción serial de ASCII a hexadecimal.
- T12D Activar UCTEL para que señales de control sean locales. Solicita el resultado al Esclavo.
- T13D Desactivar UCTEL, señales de control serán externas. Solicita el resultado al Esclavo.
- T14D Modifica fecha y hora a solicitud de la consola.
- T15D Modifica tabla de plan de tarifas nil cil ni2 ci2 con la i-esi a enviada por la consola.
- T16D Modifica tabla de plan de horarios por tipo de día.
- T17D Modifica tabla de días feriados
- T18D Modifica los contadores de un abonado y envía resultado.
- T19D Inicializa los contadores de los abonados número N<sub>inicial</sub> a N<sub>final</sub>.
- T22D Inicializa los contadores de todos los abonados.
- T24D Envía a consola solicitante el estado del bit de Alarma activada ó desactivada.
- T25D A petición de consola, Maestro pide a Esclavo el estado activado ó desactivado del sistema.
- T26D Envía fecha y hora a la consola.
- T27D Envía tabla plan de tarifas de RAM externa a consola.
- T28D Envía tabla de horarios por tipo de día a consola.
- T29D Envía tabla de feriados a consola.
- T30D Leer el contador del abonado N-simo y enviarlo a consola.
- T31D Leer los contadores de los abonados de N<sub>inicial</sub> a N<sub>final</sub> y enviarlo a consola.
- T32D Leer los contadores de todos los abonados (10420) y enviarlo a consola.
- T35D Conversión de hexadecimal a ASCII para transmisión serial
- T36D Inicio de transmisión serial.
- T37D Envía a consola el estado del bit de Reporte.

- T38D Maestro solicita estado de activado ó desactivado a Esclavo y controla el tiempo de rpta.  
 T39D Maestro envía a Esclavo el último estado activado/desactivado.  
 T40D Maestro ordena al Esclavo activar el sistema.  
 T41D Maestro ordena al Esclavo desactivar el sistema.

Los BE como T01D, T02D, T07D, etc. no se toman en cuenta, porque sus tareas asociadas ha sido asumidas con prioridad por las interrupciones.

### **3.3.2 Programa supervisor e interrupciones del $\mu$ C esclavo**

De manera similar al Maestro, en el Esclavo, el programa de control empieza inicializando su entorno, ésto es, RAM interna, registros, recursos como sus Puertos que interactúan con hardware externo, el canal de comunicación FIFO, etc. Luego de habilitar interrupciones entra en su bucle Supervisor.

El Control del programa en el Supervisor interroga por el estado del bit de ejecución  $BEL_N$ , si lo encuentra activado ( $BEL_N=1$ ), ejecuta la tarea asociada a éste bit.

Las tareas urgentes y necesarias de ejecución inmediata, son ligadas en orden de prioridad, a las siguientes interrupciones:

**INTO:** Provocada por el flanco de bajada de la señal ISCA'. Controla el sincronismo y lee el estado S00...S09 de los abonados.

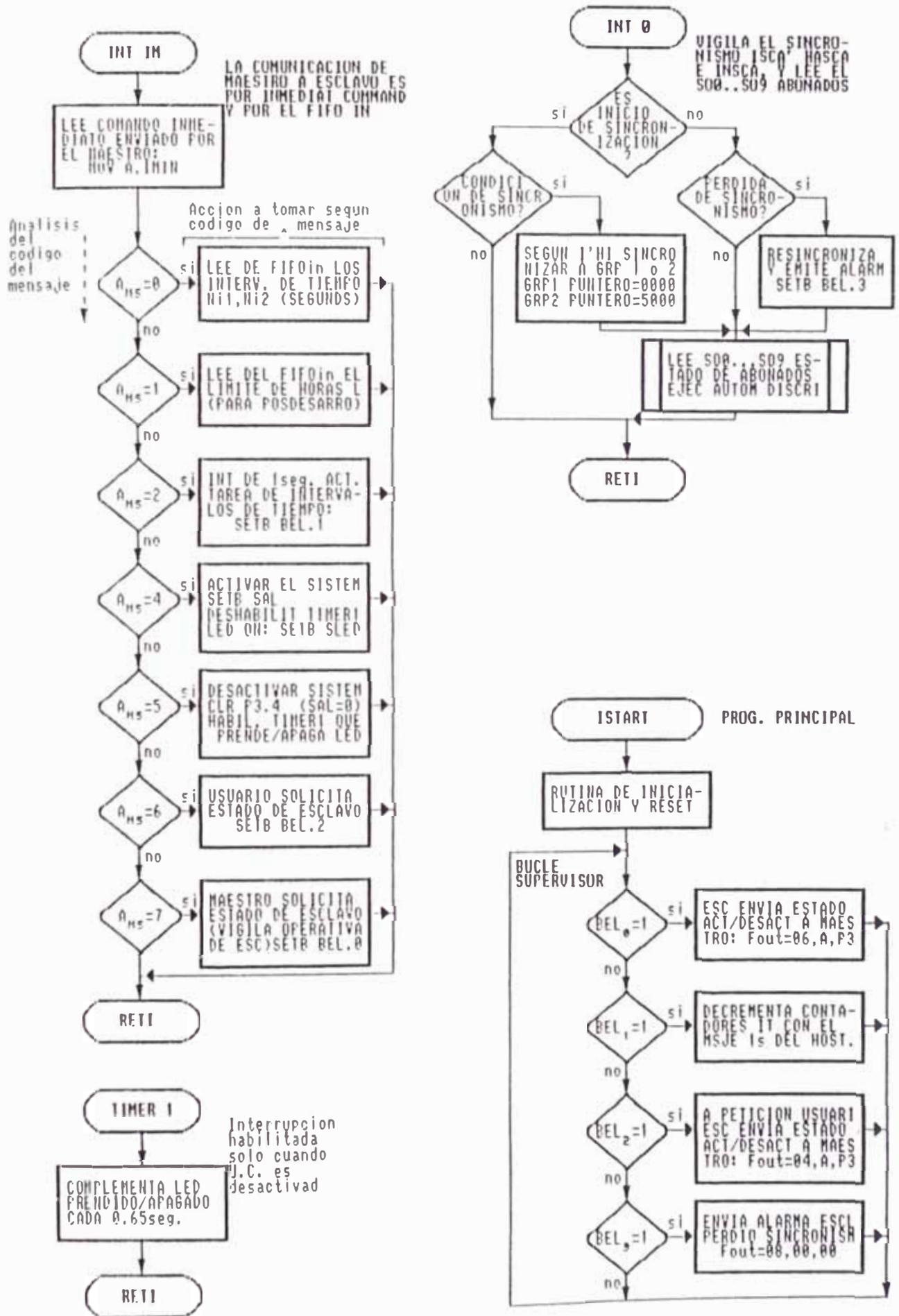
**INT IM:** Ejecuta la lectura y análisis del mensaje enviado por el Maestro. Este mensaje es escrito por el Maestro sobre el registro IM del Esclavo, generándole una interrupción de Comando Inmediato (IM), si hay data asociada a éste Comando -caso de envío de  $n_{11}$  y  $n_{12}$ - es recibida por el canal FIFO de entrada.

**TIMER1:** Su tarea asociada, no es prioritaria, sólo se usa para temporizar el Led señalizador de UCTEL activado (Led on) ó desactivado (Led blinking).

#### **3.3.2.1 Tareas del programa supervisor del Esclavo**

Son menores en número que las del Maestro, y son ejecutadas cuando el programa Supervisor del Esclavo encuentra activado su bit de ejecución  $BEL.N$ .

Fig. 3.2 DIAGRAMA DE BLOQUES SOFTWARE DEL uC ESCLAVO 80C452



**BE TAREA**

- BELO Esclavo envía estado activado ó desactivado a Maestro.  
 BEL1 Decrementa contadores de IT con el mensaje 1s del Maestro.  
 BEL2 Envía estado al Maestro a petición de consola.  
 BEL3 Envía alarma Esclavo perdió sincronismo.

**3.4 Configuración de entorno e inicialización**

Estableceremos aquí la distribución de los recursos de RAM externa e Interna para cada uno de los procesadores. Con ésto podremos entrar en las rutinas de Inicialización, donde se configura el ambiente donde trabajará el programa de control.

**3.4.1 Distribución de memoria RAM para  $\mu$ C maestro**

La Memoria disponible del Maestro es en su mayor parte en RAM externa, la respaldada por batería, destinada en su mayor parte para los Contadores de abonado, el resto para las tablas de Planes de Tarifa PT, horarios por tipo de día, tabla de feriados, además para la Data ASCII a transmitir ó recibir.

La RAM interna en cambio, no es respaldada por batería, y se usa para trabajar las variables, los datos hexadecimales en Tx/Rx serial, el stack, el espacio de registros de función SFR.

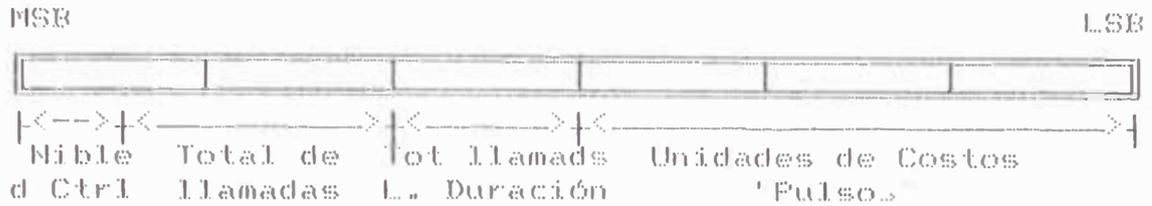
.RAM ó Periférico asignado		Bytes	Desde	Hasta
.RAM	RAM de abonados	63000	0000h	F437h
ctrlId	Libre RAM accesible	8	F438h	F43Fh
x	Plan de Tarifas	64	F440h	F47Fh
DCU	Horarios	Día0 Lu-Vi	16	F480h F48Fh
		por tipo Día1 Sáb	16	F490h F49Fh
	de Día.	Día2 Dom	16	F4A0h F4AFh
		64 Día3 Previsib	16	F4B0h F4BFh
Días Feriados	Tabla1	64	F4C0h F4FFh	
	128 Tb12 Prev	64	F500h F53Fh	
	Tbl. Data ASCII en Rx	128	F540h F5BFh	
	Tbl. Data ASCII a Tx	128	F5C0h F63Fh	
	Data de Alarmas(1B/3B)	1024	F640h FA3Fh	
	RAM Libre accesible		FA40h FBFFh	
	RAM no accesible		FC00h FDFFh	
.Reloj RTC DS1285	14Reg+50bytes	64	FE00h FEFFh	
.Procesador Esclavo 80C452		5	FF00h FFFFh	

Tabla 3.4 Mapa de Datos Externos del  $\mu$ C 8032

**3.4.1.1 RAM externa: contadores de abonado y tablas**

Aquí consideramos el espacio básicamente para los contadores, plan de tarifas, horarios y días feriados.

**CONTADORES**



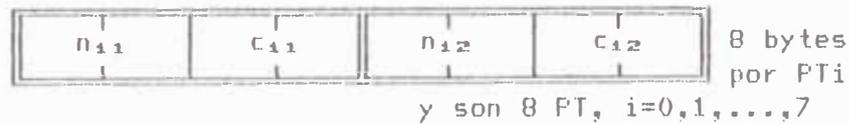
Por abonado estimaremos 6 bytes, distribuidos en tres contadores, así:

- .3 bytes para las Unidades de Costo "Pulsos"
- .1 byte para total Llam. Larga Durac. *Possible uso...*
- .1½ bytes para total de llamadas *a implementar*
- .½ byte para control de reporte de llamadas

Total 6Bytes \* 10500 abonados = 63000Bytes

**TABLA PARA TARIFAS**

-Planes de Tarifa:



Se necesita 8 bytes por cada PT, y son 8 PT

Total: 8 \* 8 = 64Bytes

-Horarios por tipo de Día: HH PT



Cada Horario con su Plan Tarifario consta de 2 Bytes, tenemos 8 horarios por día (j=0...7) y 3 tipos de día (i=0,1,2), considerar 4 tipos. Total 2 \* 8 \* 4 = 64Bytes

-Días Feriados: DF



Con 2 Bytes por feriado, y una tabla de 32 feriados. Considerar 2 tablas de feriados:

Total 2 \* 32 \* 2 = 128Bytes

## TABLA DE DATOS ASCII

Los datos que viajan por el canal serial, en general deben estar constituidos de caracteres ASCII e imprimibles, así, los datos que se reciben, antes de ser convertidos hexadecimal deben estar alojados en RAM externa. Lo mismo se aplica los datos que se transmiten, antes de ser enviados, se deben convertir en ASCII formando una tabla en RAM externa.

La máxima longitud de los mensajes serie será de 128 bytes. con dos tabla, una para recepción y otra para transmisión.

Total  $2 * 128 = 256$  Bytes

Resumiendo, la distribución total de memoria externa del  $\mu$ C Maestro sería:

Contadores	6 * 10500 =	63000
Plan de Tarifas	-	64
Horarios x tipo de Día =		64
Días Feriados	=	128
Tablas para Tx/Rx ASCII=		256

Total 63512 Bytes

Adicionalmente, el Maestro tiene 50 bytes libres en el Reloj RTC DS1285, algunos son usados para almacenar el estado Activado ó Desactivado de la Unidad de Control, su identificación, etc.

### 3.4.1.2 RAM interna: Área de trabajo

La RAM interna del Maestro es usada para trabajar con variables, punteros, procesar las cadenas de mensajes a transmitir ó recibir, área de stack para interrupciones, y trabajar los registros de función SFR.

El espacio Lower 128 es destinado para variables, y los registros R0 y R1 son usados como punteros. El rango de bytes de 20h a 2Fh es variables tipo bit, usados como bandera para diversa tareas como los BE del Supervisor por ejemplo.

La primera mitad del espacio Upper 128 es usado para tratar los datos hexadecimales que interactúa con su correspondiente ASCII de RAM externa. El resto, 64 bytes, es usado como zona de Stack.

La tabla 3.5 muestra los registros SFR y el espacio Lower 128 y Upper 128 en la RAM interna del  $\mu$ C 8032.

### **3.4.2 Maestro: inicialización y tareas en el programa supervisor**

En la rutina de inicialización es donde se establece la configuración del entorno del Maestro. Después de ejecutar la rutina de inicialización, el programa entra en el bucle Supervisor, y aquí veremos la ejecución de una de sus tareas, la de selección de plan de tarifas en función de la fecha y hora vigentes.

#### **3.4.2.1 Rutina de inicialización**

Esta rutina sólo se ejecuta sobre el Power On ó Reset. Así el Maestro inicia la ejecución del programa efectuando un retardo (~1seg.) para que todo el sistema Telmet se estabilice antes que UCTEL empiece a trabajar su software.

- .Con las interrupciones deshabilitadas, el Maestro prioriza éstas en el siguiente orden: INTO, INT1, TMRO, TMR1 y TxRx. (MOV IP,#0C
- .Habilita el inicio del Esclavo.
- .Se inicializan registros internos, se fija Stack 64bytes (SP=#BFh), etc.
- .Lee el último estado de UCTEL almacenado en la RAM externa que ofrece el Reloj DS1285 (#LAST\_STATE #FE10h), si el valor no es "0", se setea bit T12D, activar sistema.
- .La RAM interna es limpiada.
- .Timer 0 y Timer 1 son configurados para trabajar en modo 16 bits.

```
MOV  TMOD,#11H ;Timer1 y 0 en modul: 16bits
SETB TCON.4   ;Activa al Timer 0: TRO ON
SETB TCON.6   ;Activa el Timer 1: TR1 ON
```

Fig. 3.3 INICIALIZACION DEL MAESTRO



- Se inicializa variables tipo byte y bit (p.e.)

```

IUC  <- #LAST_IUC ;Default # de identif. UC
JCONS<-#LAST_JCO ; y # de identif. CONSOLA
CI1  <-#0C8h ;Tarifa 200 pulsos default x
CI2  <-#0C8h ; 1er y 2do Interval d Tiemp
SETB ALARMACTIV ;Bit de Alarmas Activado
PE   <-#F640h ;Punteros d Entrada y Salida
PS   <-#F640h ; para emisión de alarmas
LIMI <-#FA40h ;Límite de profundidad para
      ; los punteros PE y PS
SETB SW000 ;Para EMPEZAR análisis de
      ;abonado a reportar.

```

- Inicializa vars. de Tx/Rx serial y configura el Timer2 como generador de Baud Rate:

```

MOV  RCAP2L,#0DCH ;BAUD RATE=9600 con
MOV  RCAP2H,#0FFH ;xtal de 11.0592MHz.
MOV  T2CON,#034H ;Timer2 como BAUD RATE
MOV  SCON,#050H ;UART en Modo 1
SETB ES ;Prepara habil. INT SERIE

```

- Configura el RTC DS1285 para interrumpir cada segundo y cada hora, haciendo diferencia entre ambas cuando sea leído su Registro C.

```

MOV  A,#82H ;Desabilita Update Transfer
MOV  DPTR,#RTC_RB ; e interrupts, configura en
MOVX @DPTR,A ; modo 24horas BCD.
MOV  A,#0A0H ;Pone en ON al oscilador de
MOV  DPTR,#RTC_RA ; RTC y deshabilit PERIODIC
MOVX @DPTR,A ; INTERRUPTS y SQW.
MOV  A,#0F0H ;Pone código "don't care"
MOV  DPTR,#RTCAH ; en alar a horas.
MOVX @DPTR,A
CLR  A ;Pone 00 en alarma minutos
MOV  DPTR,#RTCAM ; y alarma segundos.
MOVX @DPTR,A
MOV  DPTR,#RTCAS
MOVX @DPTR,A
MOV  A,#32H ;Habilita update (SET=0) e
MOV  DPTR,#RTC_RB ; interrupciones de RTC por
MOVX @DPTR,A ; Alarma AIE y Update UIE.

```

- Activa tarea de selección de plan de tarifas.
- Prepara Alarma de que UC ha sido inicializada, la alarma es enviada a Consola por Pto serial.
- Hace lectura de RAM externa para observar el pin de "Error" del DCU 8206, y prepara aceptar interrupciones del RTC ó del DCU.

- Consulta por el registro HSTAT del Esclavo y espera por que en re en modo normal.
- Habilita atención a todas las interrupciones.
- El control del programa entra al bucle Supervisor.

### 3.4.2.2 Rutina de selección del plan de tarifas

La entrada al programa Supervisor del Maestro es siempre complementando el pin de STROBE para señalar que no se ha perdido el control del programa.

```

    ) SUPERVIS: CPL STROB      ;Genera Señal de STROBE
      T05: JNB T05D,T06
          LCALL SELPT ;Selecciona plan de tarifas
      . . .
      Tx : JNB TxD,Ty
          LCALL TAREAx ;Ejecutar la subrutina TAREAx
      . . .
    (< L JMP SUPERVIS
  
```

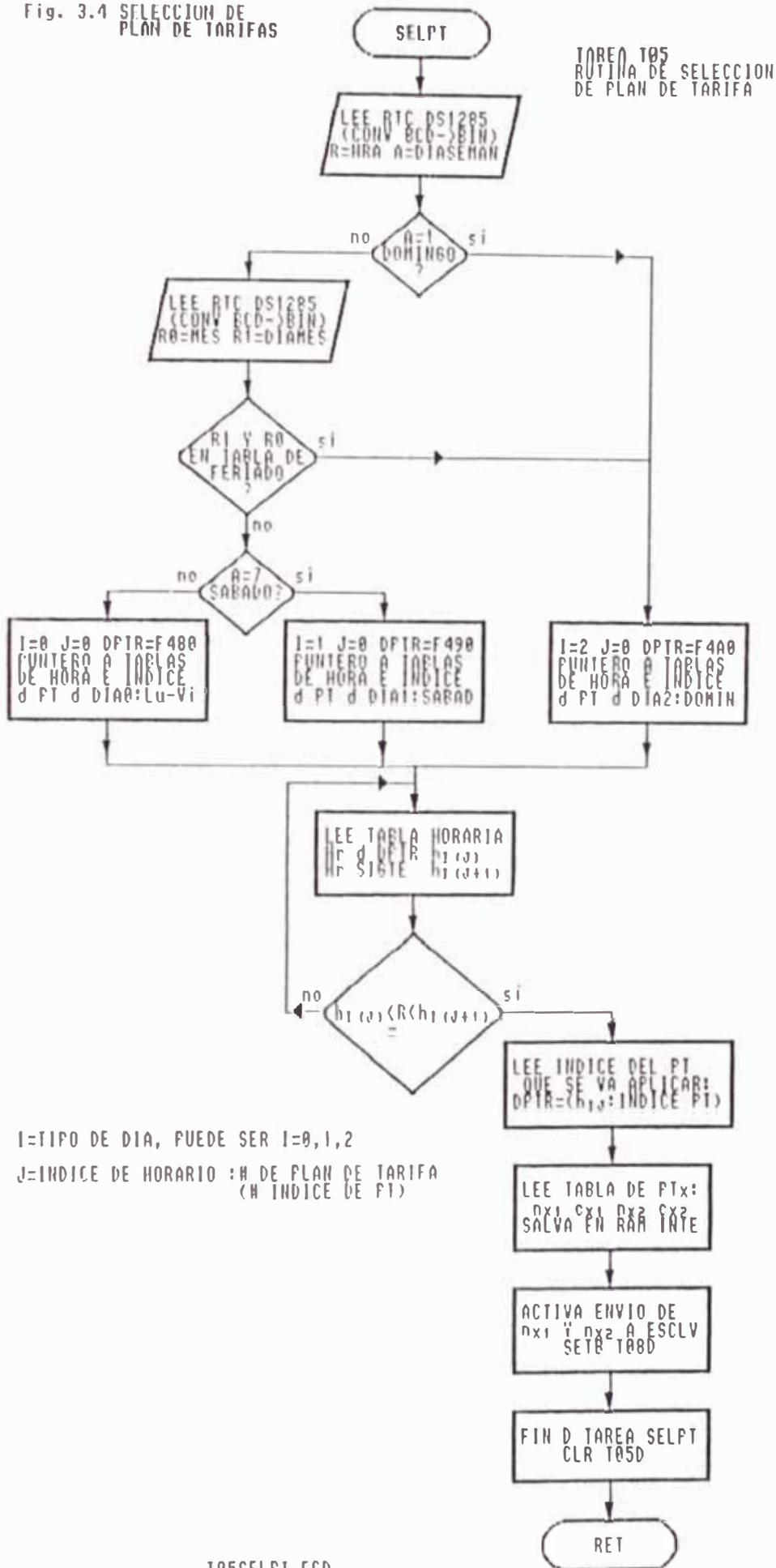
La rutina de selección del plan de tarifas SELPT, es ejecutada cuando el programa Supervisor encuentra T05D en ON.

Seleccionar el Plan de Tarifa consta básicamente de tres etapas: a)Determina con que tipo de Día (0,1,2) se ha aplicar el PT. b)Dentro de la tabla tipo de Día, se ubica el intervalo horario que corresponde a la hora leída del RTC, éste nos dará el índice del PT ha aplicar. c)Se lee el PT de RAM externa y se carga las variables de RAM interna correspondientes. Se termina la rutina activando la tarea de envío de "Intervalos de Tiempo" ni1 e ni2 al Esclavo.

Para determinar la Tabla tipo de Día, se lee del RTC el día de la semana, si no es domingo, se lee de RTC el día y mes y se busca en la tabla de feriados por ésta fecha, si la encuentra se aplica la tabla de los domingos (Día 2); de lo contrario pregunta por si es sábado para aplicar tabla de sábados (Día 1), en caso opuesto, aplica tabla de D'a 0 que corresponde de lunes a viernes.

Con la tabla escogida y la hora leída del RTC, se ubica en el intervalo horario que le ofrece la tabla para determinar el índice del Plan Tarifario a aplicar. Final-

Fig. 3.4 SELECCION DE PLAN DE TARIFAS



mente lee el PT de RAM externa, carga sus variables con ni1, ci1, ni2, ci2 y activa (SETB TO8D) el BE de envío de los intervalos de tiempo ni1 ni2 al Esclavo.

### 3.4.3 Distribución de memoria para el $\mu$ C esclavo

La memoria de trabajo del Esclavo, al igual que la del Maestro está conformada por RAM externa e interna, la RAM externa es más pequeña, y no es necesario el respaldo con batería.

#### 3.4.3.1 RAM externa: contadores y control

Usado solamente para los contadores decrementales de intervalo de tiempo ni1 ni2, y para controlar el inicio ó término de cada conversación por abonado.

Por abonado se asignará tres bytes, donde:

\*BYTE 2 y BYTE 1 : Serán usados para la cuenta del tiempo (ni1 ó ni2) por cada abonado

\*BYTE 0 : Administrado bit a bit será usado para el control de la llamada, así...

B7,B6,B5,B4: Control de la llamada por el autómata discriminador

B3,B2,B1,B0: Control de las llamadas de larga duración (Uso potencial).



Total de bytes necesarios:

3 Bytes \* 10420 abonados = 31260 Bytes  
 ≈ 31KBytes

Con 31KB no exedemos el total disponible de memoria externa 32KBytes.

### a la 3.6 Mapa e Memoria Externa del $\mu$ C 80C452

RAM Externa del Esclavo	Bytes Desde Hasta
.Contadores de abonados	31260 0000h 7A1Bh
.RAM libre disponible	1508 7A1Ch 7FFFh



```

    Deshab Int. de FIFO de entrada
    Deshab Int. de FIFO de salida
    Deshab Int. cuando un Comando
    inmediato es escrito.
    Deshab Int si IMout disponibl
    SLCON 00000100b Slave Control reg.
    LReq rto si FIFOout vacío
    LReq rto si FIFOin lleno
    Reservado, siempre lee 1
    Modo Freeze (congelado)
  
```

-Deshabilitar Interrupciones al Host Maestro  
provenientes de FIFO ó Comando Inmediato del Esclavo.  
HCON=0000000b Host Control reg.

-El FIFO de 128bytes, se distribuye en 99bytes para  
FIFOout (máximo 33 mensajes de 3 bytes c/u). Los 29  
1Dh restantes son el FIFOin, así CBP=1Dh.

```

MOV CBP,#01DH ;Chanel Boundary Pointer además automática-
MOV OWPR, 01DH ; mente hace Input y Output FIFO Read/Write
MOV ORPR,#01DH ; Pointers IRPR=IWPR=0 y ORPR=OWPR=CBP=#1DH
MOV ITHR,#00H ;Input/Output FIFO Threshold. ITHR no impor.
MOV OTHR,#02H ;3 bytes ó mas en FIFOout genera INT a Host.
  
```

.Limpieza de RAM

```

MOV RO,#0FFH
ICLR IRAM: MOV @RO,#00H ;Bucle de Limpieza de RAM INTERNA
            DJNZ RO,ICLR_IRAM
            MOV DPTR,#0000H
            CLR A
ICLR XRAM: MOVX @DPTR,A ;Bucle de Limpieza de RAM EXTERNA
            INC DPTR
            MOV RO,DPH
            CJNE RO,#080H,ICLR_XRAM
  
```

.Inicialización de variables.

```

CLR SS ;Sincronismo aún no alcanzado (bit)
BE <=#0000h ;Reset a todos los bit de Ejecución
ni1<=#00B4h ;Intervalos de tiempo para tasación
ni2<=#00B4h ; por default: 180segs. (#B4h).
CAR<=#0000h;Contador de Núm. de Abonado en Rutina de lectura
  
```

.Inicializa el Timer1

```

MOV TMOD#10H ;Timer 1 en modo 1, modo de 16 bits.
SETB TCON.6 ;Activa Timer1: TR1 ON
SETB TCON.0 ;INT0 de ISCA' será activd por flanco de bajada
  
```

.CONFIGURACION FINAL INTERFAZ MAEST-ESC

```

MOV HCON,#080H ;El HOST Atenderá solo al FIFO de salida,
                ;cuando éste requiere servicio (INTRQOUT).
MOV SLCON,#029H ;ESC a modo normal y sólo acepta Int. IMIN
MOV MODE #0BFH ;Permite interrumpir al Maestro en INT1
  
```

.Avisa a Maestro de ESC inicializado. Habilita Int. EA=1.

.Entra al bucle Supervisor de Esclavo.

### 3.4.4.2 Lazo del programa supervisor

Las interrupciones tiene prioridad sobre el Supervisor, dentro de éste, se ha priorizado a la Tarea0.

```

ISUPERVIS:JB BEL.0,TAREA00 ;Envía al Maestro el estado solicitud
          JB BEL.1,TAREA01 ;Decrem Contad IT con sje ls de Host
V        JB BEL.2,TAREA02 ;Envía a Maestro est. solíc. x usar.
          JB BEL.3,TAREA03 ;Envía ALARMA x perdida de sincronismo
<        SJMP ISUPERVIS

```

### 3.5 Protocolo de comunicación entre procesadores

La comunicación entre el Maestro y el Esclavo es bidireccional, y en ambos casos la presencia de un mensaje significa una interrupción sobre el destinatario.

#### 3.5.1 Comunicación del maestro al esclavo

El Maestro escribe sobre el registro Comando Inmediato y sobre el FIFOin del Esclavo. El Esclavo será interrumpido cuando reciba un byte código en su registro IMIN. Sólo se puede escribir un byte a la vez en IMIN, hasta que sea leído y se vacíe, por eso la data adicional al código (si la hay) viaja por FIFOin. Los códigos y data que el Maestro envía al Esclavo son:

#### Código

x	IMIN	FIFOin	Descripción
0	ni1,ni2		Envía valores de ni1 y ni2
1	h		Envía límite de horas(PostDesarrollo.)
2			Interrupción de 1sg
3			Interrupción de 1h (PostDesarrollo.)
4			Activa el sistema
5			Desactiva el sistema
6			Maestro solicita estado(chqueo)
7			Usuario solicita estado

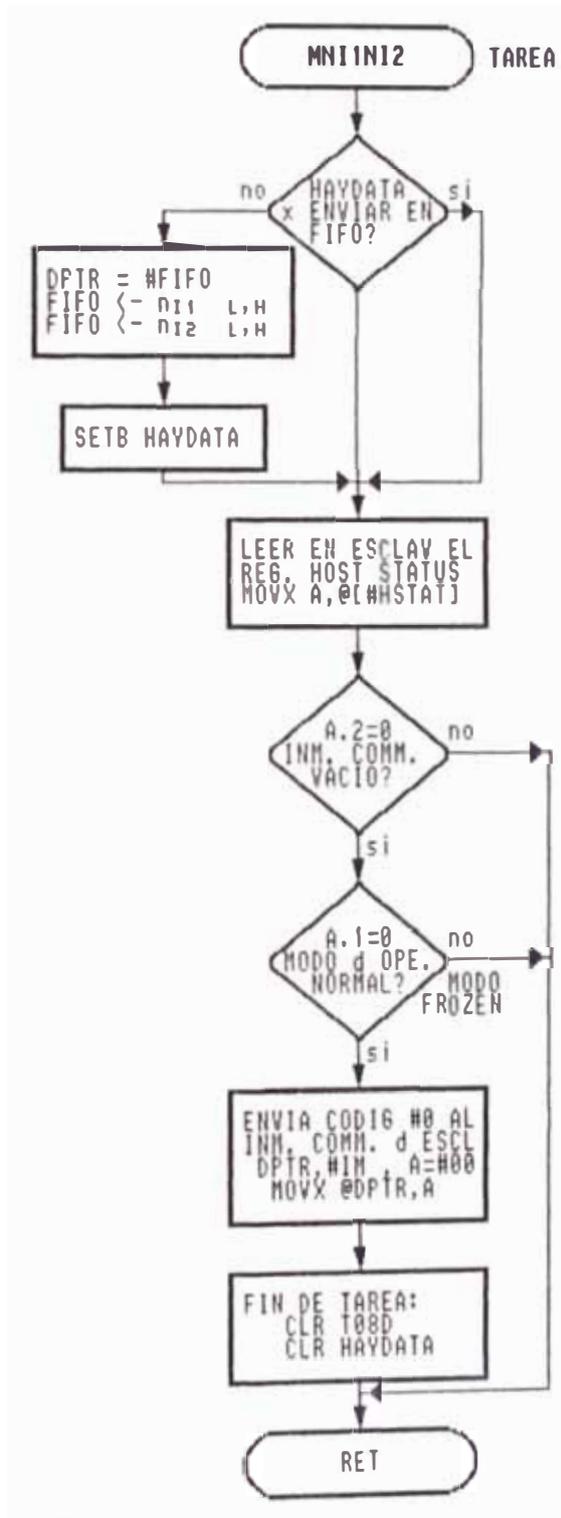
#### 3.5.1.1 Tarea envío a esclavo de los intervalos de tiempo

La tarea consiste en enviar un mensaje de código "0" al esclavo.

El BE de ésta tarea es activado después que la rutina de selección de Plan de Tarifas ha actualizado ni1, ci1, ni2, ci2 en su RAM interna.

La rutina actual escribe en la cola FIFO los IT ni1 y ni2 y usa una bandera señalizadora HATDATA para no reescribir en la cola. Se considera enviado el mensaje

Fig. 3.5 MAESTRO ENVIA MENSAJE A ESCLAVO CON INTERVALOS DE TIEMPO  $n_1$   $n_2$  segundos



T08MNI1N2.FCD

cuando se escribe el código #00 en la dirección #IM del registro Comando Inmediato IMIN del Esclavo.

Si el reg. IMIN está ocupado, se sale de la rutina a limpiar las banderas bit HAYDATA ni TOSD de tarea ejecutada, así el Supervisor en la siguiente vuelta volverá a ejecutar la rutina pero sin reescribir data en el FIFO. El diagrama de flujo (Fig 3.5) de la Tarea 08 muestra el detalle.

### 3.5.2 Comunicación del esclavo al maestro

Esta es por el Canal FIFOout, y a diferencia del caso anterior es en un formato específico de bytes. Escritos los tres bytes, el Esclavo genera interrupción al Maestro. Si éste no tiene tiempo para atender la interrupción, el Esclavo puede aún seguir enviando mensajes, pues la profundidad del FIFOout es mucho mayor. Los códigos y datos que el Esclavo envía al Maestro son:

FIFOout	
Códg	Da a Descripción
0	CAIT Envía # de abonado que entró en estado de conversación (ocupado).
1	CAIT Envía # de abonado que cumplió su IT (Intervalo de Tiempo) ni1 ó ni2.
2	CAIT Envía # de abo. que cumplió las h hrs límite de llamada de L.Du.(Post. uso)
3	CAIT Envía # de abonado que acabó la conversación (libre).
4	STAT Envía estado para ser Tx a Consola.
5	ENTO Envía reporte de entorno (Post. uso).
6	STAT Envía estado al Maestro (chequeo).
7	x Indica que Esc. ha sido inicializado.
8	x Contador de Esc. perdió sincronismo.

CAIT que significa el número de abonado, es compuesto por dos bytes, y éste número identificador de abonado se expresa en hexadecimal.

#### 3.5.2.1 Esclavo envía estado del sistema solicitado por el maestro

La tarea consiste en que el Esclavo envía un mensaje de código "6" seguido por el estado STAT activado/desactivado del sistema.

Fig. 3.6 ATENCION DEL MAESTRO A MENSAJES DEL ESCLAVO

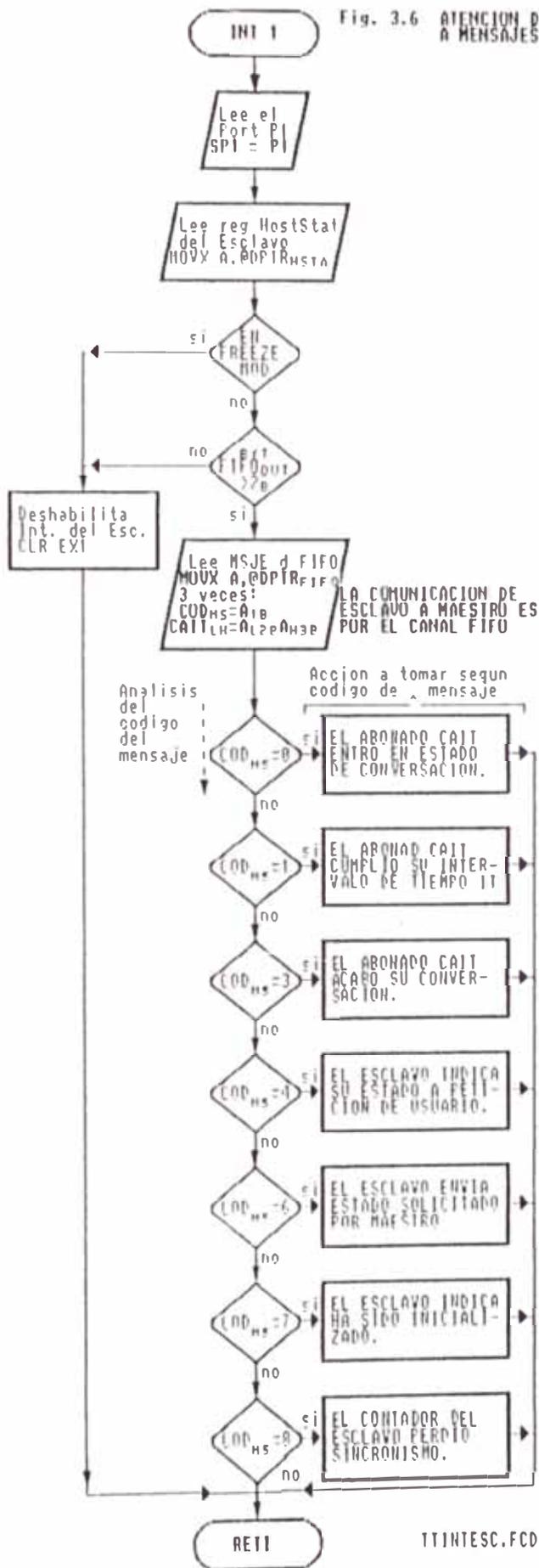
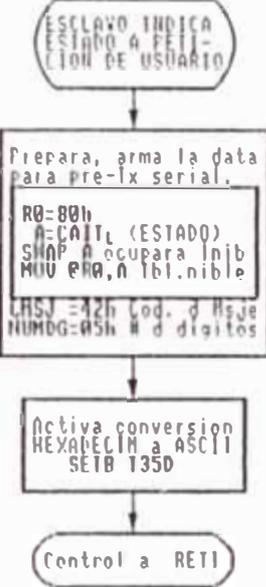
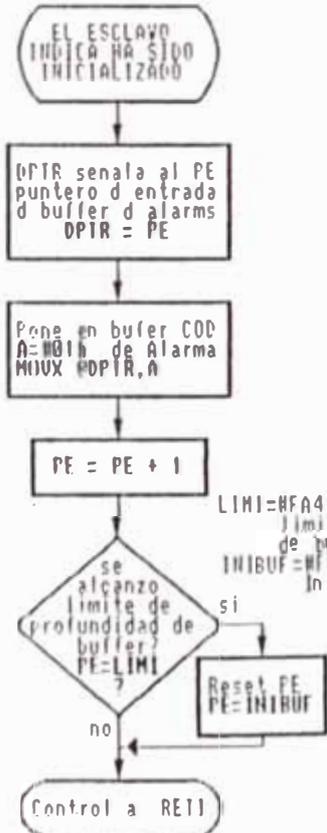


Fig. 3.7 ATENCION DEL MAESTRO A MENSAJES EN EL FINO DEL ESCLAVO

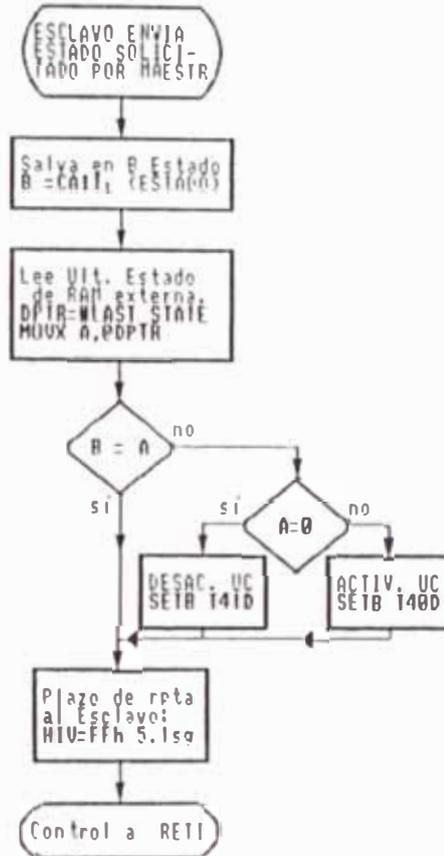
MAESTRO RECEPCIONA DE ESCLAVO: A=CODHS=4



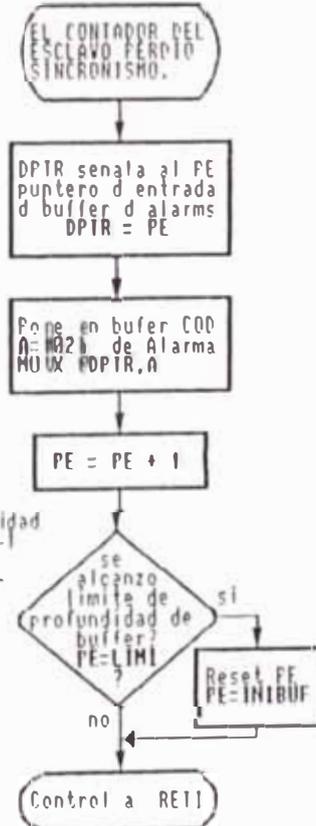
MAESTRO RECEPCIONA DE ESCLAVO: A=CODHS=7



MAESTRO RECEPCIONA DE ESCLAVO: A=CODHS=6



MAESTRO RECEPCIONA DE ESCLAVO: A=CODHS=8



Se observa en el reg. SSTAT (Slave Status) que el FIFOout no esté lleno, luego según el estado del pin SAL=P3.4 se carga el reg. A con activado (A=#00) ó desactivado (A=#01).

Se escribe sobre el FIFOout con tres bytes: El código, el estado STAT en el reg. A, y un byte más para completar el formato de comunicación Esclavo a Maestro.

```
TAREA00:  MOV A,SSTAT      ;Lee reg. status de Esclavo
          JNB ACC.4,SALTINO ;Si ACC.4=SST4=1, el FIFO OUT está lleno
          SJMP LUPCIT      ; espera que se desocupe.
SALTINO:  JB SAL,SI_ACTIVA ;Lectura del pin SAL=P3.4
DESACT1:  MOV A,#00H
          SJMP OKEY
SI_ACTIVA: MOV A,#01H
OKEY:     MOV FOUT #06H ;Envía al Maestro con COD=#06 el
          MOV FOUT,A   ; estado de UCTEL en el reg. A
          MOV FOUT,P3  ; relleno (P3) completa el formato.
          CLR BEL.0    ;Tarea terminada: BE=0
          SJMP ISUPERVIS ;Retorno a Supervisor.
```

### 3.6 Protocolo de comunicación serial

Los lineamientos a seguir para que nuestra Unidad de Control establezca comunicación con el exterior los fijaremos en:

#### 3.6.1 Formato de comunicación

Con el UART del Maestro trabajaremos en:

.Asíncrono

.9600 BPS

.1 bit de inicio, 8 bits de datos, y 1 bit de stop

Esto es, en la inicialización del Maestro significa:

```
MOV RCAP2L,#0DCH ;BAUD RATE=9600 con
MOV RCAP2H,#0FFH ;xtal de 11.0592MHz.
MOV T2CON,#034H ;Timer2 como BAUD RATE
MOV SCON,#050H  ;UART en Modo 1
```

Con el Temporizador 2 (del 8032) usado como generador de Baud Rate, los valores de RCAP2L y RCAP2H vienen de despejar la Ecuac. 2.1 (acápite 2.3.8) para una frecuencia de oscilador de 11.0592Mhz, de donde resulta:

$$RCAP2 = 65500 = FFDCh.$$

SCON=#50h configura al UART interno en modo 1: 1 start bit(0), 8 data bits, 1 stop bit(1). Total 10 bits.

### 3.6.2 Estructura de mensajes

Considerando que varias Unidades de Control (hasta ocho UCTEL por cuatro TELMET en planta) deben ser manejados por la PC Consola Local vía la Unidad de Interfaz, y esta además debe tener la capacidad de conmutar la conexión de las Unidades de Control con las PC consola remota y local, trabajaremos con una estructura de datos con una cabecera que señale la identificación de la Unidad de Control y el PC Consola involucrados en el mensaje.

STX,i,j,CD,DATA,ETX,CRC

Donde:

STX = Comienzo del mensaje

i = Número de la Unidad de Control (ó de interface)

i=0,1,...,7 para Unidades de Control

(i=8 para Unidad de Interfaz).

j = Número de la PC Consola.

j=0 Consola Local/j=1 Consola Remota

CD = Código del mensaje que precede la DATA y facilita su interpretación.

DATA = Datos transmitidos.

ETX = Fin de mensaje.

CRC = Byte de Control de Error calculado como el XOR de todos los bytes de la trama post STX a ETX: <STX,ETX>.

#### Señales de reconocimiento:

A la recepción de un mensaje, el destinatario debe comparar su CRC calculado con el CRC recibido, y según el resultado, debe responder con:

ACK	Respuesta a un mensaje bien recibido.
NACK	Respuesta a un mensaje incorrecto.

### 3.6.3 Código de los mensajes

Todos los bytes del mensaje deben estar en código ASCII y deben ser imprimibles. La información es tratada en hexadecimal sólo en CPU de UCTEL ó en CPU de Consola.

En el caso de recepción de mensajes, es la Unidad de Control la encargada de convertir los ASCII hexadecimal.

#### Mensajes enviados hacia la Unidad de Control:

<u>CD</u>	<u>Código de Mensaje</u>
00	= Activar Unidad de Control
01	= Desactivar Unidad de Control
02	= Solicita estado de Unidad de Control
03	= Modifica fecha y hora
04	= Solicita fecha y hora
05	= Modifica plan de tarifas
06	= Solicita plan de tarifas
07	= Modifica horarios
08	= Solicita horarios
09	= Modifica días feriados
10	= Solicita días feriados
11	= Solicita lectura de los contadores del abonado Ni
12	= Modifica contadores del abonado Ni
13	= Solicita lectura de los contadores de un grupo de abonados: desde Ni al Nf
14	= Inicializa los contadores de los abonados Ni al Nf
15	= Solicita lectura total de contadores
19	= Inicialización total de los contadores
24	= Activa / Desactiva alarmas
25	= Solicita estado de alarmas
27	= Activa / Desactiva reportes, involucra mensajes 70 y 71
28	= Solicita estado de reportes

Después de la recepción del mensaje, y convertido de ASCII a hexadecimal, una rutina tarea debe discriminar el código del mensaje para activar el BE la tarea que se solicita al Maestro.

La descripción de los mensajes que puede recibir la Unidad de Control (-lo mismo-, ó los mensajes que envía el computador) son mostrados en detalle en la Tabla 3.8.

TABLA 3.8 DESCRIPCION DE MENSAJES ENVIADOS HACIA UNIDAD DE CONTROL

ACTIVAR UNIDAD DE CONTROL	BTX,1,J,00,ETX,CRC
DESACTIVAR UNIDAD DE CONTROL	BTX,1,J,01,ETX,CRC
SOLICITA ESTADO DE LA UNIDAD DE CONTROL	BTX,1,J,02,ETX,CRC
MODIFICA FECHA Y HORA	BTX,1,J,03,AAMDDDDwhhmmss,ETX,CRC
DONDE, AA = AÑO MM = MES DD = DIA Dw = DIA DE SEMANA hh = HORA mm = MINUTO ss = SEGUNDO	
SOLICITA FECHA Y HORA	BTX,1,J,04,ETX,CRC
MODIFICA PLAN DE TARIFAS	BTX,1,J,05,0,0nnn(01)0ccc(01)0nnn(02)0ccc(02),ETX,CRC BTX,1,J,05,1,0nnn(11)0ccc(11)0nnn(12)0ccc(12),ETX,CRC BTX,1,J,05,2,0nnn(21)0ccc(21)0nnn(22)0ccc(22),ETX,CRC BTX,1,J,05,3,0nnn(31)0ccc(31)0nnn(32)0ccc(32),ETX,CRC BTX,1,J,05,4,0nnn(41)0ccc(41)0nnn(42)0ccc(42),ETX,CRC BTX,1,J,05,5,0nnn(51)0ccc(51)0nnn(52)0ccc(52),ETX,CRC BTX,1,J,05,6,0nnn(61)0ccc(61)0nnn(62)0ccc(62),ETX,CRC BTX,1,J,05,7,0nnn(71)0ccc(71)0nnn(72)0ccc(72),ETX,CRC
DONDE (POR EJEMPLO): - nnn(21) = SON TRES DIBITOS EN HEXADECIMAL,QUE EXPRESAN UN NUMERO BINARIO, QUE CORRESPONDEN AL LOS SEGUNDOS DEL PRIMER INTERVALO DE TIEMPO DEL PLAN DE TARIFAS 2. - ccc(62) = SON TRES DIBITOS EN HEXADECIMAL,QUE EXPRESAN UN NUMERO BINARIO DE PULSOS, QUE CORRESPONDEN A LOS PULSOS DEL SEGUNDO INTERVALO DE TIEMPO DEL PLAN DE TARIFAS 6.	
SOLICITA PLAN DE TARIFAS	BTX,1,J,06,ETX,CRC
MODIFICA HORARIOS	BTX,1,J,07,0,hhtt(00)hhtt(01)hhtt(02)hhtt(03) hhtt(04)hhtt(05)hhtt(06)hhtt(07),ETX,CRC BTX,1,J,07,1,hhtt(10)hhtt(11)hhtt(12)hhtt(13) hhtt(14)hhtt(15)hhtt(16)hhtt(17),ETX,CRC BTX,1,J,07,2,hhtt(20)hhtt(21)hhtt(22)hhtt(23) hhtt(24)hhtt(25)hhtt(26)hhtt(27),ETX,CRC BTX,1,J,07,3,hhtt(30)hhtt(31)hhtt(32)hhtt(33) hhtt(34)hhtt(35)hhtt(36)hhtt(37),ETX,CRC
DONDE (POR EJEMPLO): - hhtt(37) = hh ES LA HORA DEL HORARIO 7 DEL DIA 3, Y tt ES EL PLAN DE TARIFAS CORRESPONDIENTE A ESTE HORARIO. - EN GENERAL hh TOMARA LOS VALORES DE 00-23 Y tt DE 0-7. - NO SERA NECESARIO QUE UN DIA, TENGA 8 HORARIOS DIFERENTES. SI SE ESPECIFICA, POR EJEMPLO PARA EL DIA 2, 007125, ESTO SIGNIFICARA QUE DESDE LAS 00 HRB HASTA LAS 12 HRB SE APLICARA LA TARIFA 7 Y DESDE LAS 12 HRB HASTA LAS 24 HRB SE APLICARA LA TARIFA 5.	
SOLICITA HORARIOS	BTX,1,J,08,ETX,CRC
MODIFICA DIAS FERIADOS	BTX,1,J,09,0,DDMM(000).....DDMM(031),ETX,CRC BTX,1,J,09,1,DDMM(100).....DDMM(131),ETX,CRC DONDE, DDMM(031) = DIA Y MES DEL FERIADO NUMERO 31 DE LA LISTA DE FERIADOS 0.
SOLICITA DIAS FERIADOS	BTX,1,J,10,ETX,CRC
SOLICITA LECTURA DE LOS CONTADORES DEL ABONADO NI	BTX,1,J,11,NNNN,ETX,CRC
DONDE, NNNN = NUMERO DEL ABONADO (0000h - 28B3h) (00000 - 10419)	
MODIFICA CONTADORES DEL ABONADO N	BTX,1,J,12,NNNN,UUUUUU,ETX,CRC
DONDE, NNNN = NUMERO DEL ABONADO EN HEXADECIMAL UUUUUU = SON LAS UNIDADES DE COSTO EN HEXADECIMAL(X100) EN DECIMAL=16 <sup>7</sup> 777,215	
SOLICITA LECTURA DE LOS CONTADORES DE LOS ABONADOS NI AL NI	BTX,1,J,13,NNNNI,NNNNf,ETX,CRC
DONDE, NNNNI = ES EL NUMERO DEL ABONADO DE INICIO DE LECTURA NNNNf = ES EL NUMERO DEL ABONADO DE FIN DE LECTURA	
INICIALIZA LOS CONTADORES DE LOS ABONADOS NI AL NI	BTX,1,J,14,NNNNI,NNNNf,ETX,CRC
SOLICITA LECTURA TOTAL NO DESTRUCTIVA	BTX,1,J,15,ETX,CRC
INICIALIZACION TOTAL DE LOS CONTADORES	BTX,1,J,19,ETX,CRC
PETICION ALARMAS	BTX,1,J,24,0,ETX,CRC
DESACTIVA ALARMAS	BTX,1,J,24,1,ETX,CRC
ACTIVA ALARMAS	BTX,1,J,24,1,ETX,CRC
SOLICITA ESTADO DE ALARMAS	BTX,1,J,25,ETX,CRC

Si la Unidad de Control es la emisora de mensajes, será la consola receptora la encargada de convertir los caracteres ASCII a hexadecimal.

En cada mensaje que recibe la Unidad de Control, debe guardar la identificación (j) de la Consola que hace el requerimiento para tomarla como **destinataria** en el mensaje que le envía como respuesta (respuesta de la Unidad de Control a la Consola). Referencia: Ver tarea T10, rutina DISCRIMI en apéndice C.

#### Mensajes enviados desde la Unidad de Control:

CD	<u>Código de Mensaje</u>
42	Envía estado de la Unidad de Control (activada ó desactivada)
44	= Envía fecha y hora
46	= Envía plan de tarifas
48	= Envía horarios
50	= Envía días feriados
51	Envía <b>lectura</b> de los contadores del abonado Ni.
53	= Envía lectura de los contadores de los abonados Ni al Nj.
55	= Envía lectura total no destructiva.
65	= Envía estado de las alarmas
68	= Envía estado de habilitación de reportes (activado ó desactivado)
70	NNNN = Rprta Ab NNNN inició conversación
71	NNNN = Reporta Abonad NNNN terminó conv.
80	= Alarmas enviadas por Unidad de Control según la DATA, las alarmas pueden ser:
00	: Unidad de Control inicializada
01	: Procesador Esclavo inicializado
02	: Procesador Esclavo perdió sincronismo
03	: Esclavo sin comunicación, no responde
10	NNNN : RAM Error no corregibl. Pos NNNN
11	NNNN : RAM Error corregible. Pos NNNN

TABLA 3.9 DESCRIPCION DE MENSAJES ENVIADOS DESDE UNIDAD DE CONTROL

ENVIA ESTADO DE LA UNIDAD DE CONTROL			
UNIDAD 1 DEBACTIVADA			BTX,1,J,42,0,BTX,CRC
UNIDAD 1 ACTIVADA			BTX,1,J,42,1,BTX,CRC
ENVIA FECHA Y HORA			BTX,1,J,44,AAMDDDDwhhmmss,ETX,CRC
	DONDE,	AA = ARO	hh = HORA
		MM = MES	mm = MINUTO
		DD = DIA	ss = SEGUNDO
		Dw = DIA DE SEMANA	
ENVIA PLAN DE TARIFAS			
			BTX,1,J,46,0,0nnn(01)0ccc(01)0nnn(02)0ccc(02),ETX,CRC
			BTX,1,J,46,1,0nnn(11)0ccc(11)0nnn(12)0ccc(12),ETX,CRC
			BTX,1,J,46,2,0nnn(21)0ccc(21)0nnn(22)0ccc(22),ETX,CRC
			BTX,1,J,46,3,0nnn(31)0ccc(31)0nnn(32)0ccc(32),ETX,CRC
			BTX,1,J,46,4,0nnn(41)0ccc(41)0nnn(42)0ccc(42),ETX,CRC
			BTX,1,J,46,5,0nnn(51)0ccc(51)0nnn(52)0ccc(52),ETX,CRC
			BTX,1,J,46,6,0nnn(61)0ccc(61)0nnn(62)0ccc(62),ETX,CRC
			BTX,1,J,46,7,0nnn(71)0ccc(71)0nnn(72)0ccc(72),ETX,CRC
	DONDE (POR EJEMPLO),		
	- nnn(21)	= CORRESPONDE A LOS SEGUNDOS DEL PRIMER INTERVALO DE TIEMPO DEL PLAN DE TARIFAS 2.	
	- ccc(62)	= PULSOS DEL SEGUNDO INTERVALO DE TIEMPO, DEL PT 6	
ENVIA HORARIOS			
	BTX,1,J,48,0,	hhht(00)hhht(01)hhht(02)hhht(03)	
		hhht(04)hhht(05)hhht(06)hhht(07),	ETX,CRC
	BTX,1,J,48,1,	hhht(10)hhht(11)hhht(12)hhht(13)	
		hhht(14)hhht(15)hhht(16)hhht(17),	ETX,CRC
	BTX,1,J,48,2,	hhht(20)hhht(21)hhht(22)hhht(23)	
		hhht(24)hhht(25)hhht(26)hhht(27),	ETX,CRC
	BTX,1,J,48,3,	hhht(30)hhht(31)hhht(32)hhht(33)	
		hhht(34)hhht(35)hhht(36)hhht(37),	ETX,CRC
	- EN GENERAL	hh TOMARA LOS VALORES DE 00-23 Y tt DE 0-7.	
	- NO SERA NECESARIO	QUE UN DIA, TENGA 8 HORARIOS	
	DIFERENTES.	SI SE ESPECIFICA, POR EJEMPLO PARA EL DIA	
	2, 007123,	ESTO SIGNIFICARA QUE DESDE LAB 00 HRB	
	HASTA LAB 12 HRB	SE APLICARA LA TARIFA 7 Y DESDE LAB	
	12 HRB HASTA LAB 24 HRB	SE APLICARA LA TARIFA 3.	
ENVIA DIAS FERIADOS			
	BTX,1,J,50,0,	DDMM(000).....DDMM(031),	ETX,CRC
	BTX,1,J,50,1,	DDMM(100).....DDMM(131),	ETX,CRC
	DONDE,	DDMM(031)	= DIA Y MES DEL FERIADO NUMERO 31 DE LA LISTA DE FERIADOS O.
ENVIA LECTURA DE LOS CONTADORES DEL ABONADO N			
	DONDE,		BTX,1,J,51,NNNN,UUUUUU,ETX,CRC
			NNNN = NUMERO DEL ABONADO EN HEX
			UUUUUU = SON LAS UNIDADES DE COSTO EN HEXADECIMAL(X100)
			EN DECIMAL=16'777,215
ENVIA LECTURA DE LOS CONTADORES DE LOS ABONADOS NI AL NI			
	BTX,1,J,53,	NNNN1,UUUUUU01,UUUUUU11,...,UUUUUU91,	ETX,CRC
	BTX,1,J,53,	UUUUUU01,UUUUUU11,...,UUUUUU91,	ETX,CRC
	BTX,1,J,53,	NNNN?,UUUUUU?,UUUUUU1?,...UUUUUU9?,	ETX,CRC
	NOTA:	SIEMPRE SE TRANSMITIRA EN GRUPOS FIJOS DE 10,	
		COMENZANDO POR EL INICIO DE LA DECENA.	
		POR EJEMPLO, SI EL USUARIO PIDE EL CONTENIDO DE LOS	
		CONTADORES DEL 1254 AL 1282, LA CONSOLA HARA EL	
		CONSIDERENCIADO PARA PEDIR DE 1250 A 1289	
		SE ENVIA EL CONTENIDO DE LAS SIGUIENTES DECENAS:	
		1250,1260,1270,1280.	
	DONDE,	NNNN = NUMERO DEL ABONADO.SIEMPRE COMIENZA	
		EN EL INICIO DE LA DECENA. EN HEX.	
		UUUUUU = SON LAS UNIDADES DE COSTO EN	
		HEXADECIMAL(X100)	
		EN DECIMAL=16'777,215	
ENVIA LECTURA TOTAL NO DESTRUCTIVA			
	BTX,1,J,55,	NNNN1,UUUUUU01,UUUUUU11,...,UUUUUU91,	ETX,CRC
	BTX,1,J,55,	UUUUUU01,UUUUUU11,...,UUUUUU91,	ETX,CRC
	BTX,1,J,55,	NNNN?,UUUUUU?,UUUUUU1?,...UUUUUU9?,	ETX,CRC
	NOTA:	SIEMPRE SE TRANSMITIRA EN GRUPOS FIJOS DE 10,	
		COMENZANDO POR EL INICIO DE LA DECENA.	
		EN ESTE CASO EL INICIO SERA 0000 Y LA ULTIMA	
		DECENA SERA 28AA (0000-10410 DEC).	
	DONDE,	NNNN = NUMERO DEL ABONADO.SIEMPRE COMIENZA	
		EN EL INICIO DE LA DECENA. EN HEX.	
		UUUUUU = SON LAS UNIDADES DE COSTO EN	
		HEXADECIMAL(X100)	
		EN DECIMAL=16'777,215	
REPORTES: NUMERO DE ABONADO QUE INICIA CONVERSACION			BTX,1,J,70,NNNN,ETX,CR
QUE TERMINA CONVERSACION			BTX,1,J,71,NNNN,ETX,CR
	DONDE,	NNNN = NUMERO DEL ABONADO. EN HEX.	
		La consola receptora debe agregarle su fecha y hora	
		de recepción en es de mostrarla ó almacenarla en archivo	
ALARMAS: ALARMAS DEL SISTEMA			BTX,1,J,80,cc,ETX,CRC
	DONDE:		cc
	UNIDAD DE CONTROL INICIALIZADA		00
	ESCLAVO DE U.C. INICIALIZADO		01
	CONTADOR DE ESCLAVO PERDIO SINCRONISMO		02
	ESCLAVO SIN COMUNICACION, NO RESPONDE		03
ALARMAS DE ERROR EN RAM:			BTX,1,J,80,cc,NNNN,ETX,CRC
	DONDE:		cc
	ERROR NO CORREGIBLE EN DIRECCION NNNNH		10
	ERROR CORREGIBLE EN DIRECCION NNNNH		11
		La consola receptora debe agregarle su fecha y hora	
		de recepción antes de mostrarla ó almacenarla en archivo	

### 3.6.4 Criterio para calcular el CRC

Para nuestra estructura de mensaje, el último byte (CRC Check Redundancy Cycle) efectúa el control de error, que se calcula como el XOR de todos los bytes comprendidos entre STX y ETX, incluido éste último.

Por ejemplo, el mensaje de alarma (80) "Esclavo sin comunicación, no responde" (03) enviado por la Unidad de Control #2 al PC consola #1 remoto, sería:

```
STX, i, j, CD ,DATA ,ETX,CRC
STX, 2, 1, 80 , 03 ,ETX,CRC
02,32,31,38,30,30,33,03,crc
```

Con:  $crc=32 \text{ XOR } 31 \text{ XOR } 38 \text{ XOR } 30 \text{ XOR } 30 \text{ XOR } 33 \text{ XOR } 03 = 0B$

Recordemos que los números y/o letras comprendidos entre STX y ETX se transmiten todos en código ASCII.

### 3.6.5 Criterio para el protocolo de comunicación

El emisor (Consola ó Unidades de Control) envía un mensaje con formato:

```
STX,i,j,CD,DATA,ETX,CRC
```

espera que le respondan:

```
..ACK
..NACK ó
..NO HAY RESPUESTA en 1 segundo.
```

Si recibe ACK, considera su envío correcto.

Si se recibe NACK ó durante 1 seg. no hay respuesta, se intenta 2 veces más, después de lo cual se cancela la operación.

Si la recepción de un mensaje se interrumpe, la operación del sistema no debe ser afectada. El sistema debe considerar la resincronización con cualquier nuevo mensaje.

#### 3.6.5.1 Transmisión y recepción serial

La transmisión recepción de mensajes de Unidad de Control al exterior es ejecutada básicamente en la rutina de servicio las interrupciones del canal serial. Sin embargo ésta rutina de servicio está asociada a otras tareas.



En el caso de transmisión de un mensaje, se parte bajo la premisa de que cualquier tarea que desea enviar su mensaje, lo debe haber formado en hexadecimal en RAM interna. Los pasos son:

- a).Tarea 35 : Convertir de Hex->ASCII. Los datos ASCII forman una tabla en RAM externa con STX, ETX y CRC
- b).Tarea 36 : Efectuar el inicio de transmisión (1er byte STX) acondicionando variables y punteros.
- c).El resto de bytes se Tx en la rutina de servicio a la Int. serial.
- d).Timer1 temporiza los reintentos de Tx. en caso de no respuesta.

La Fig. 3.8 muestra la tarea Inicio de transmisión, y la rutina de atención a la interrupción del canal serial.

En el caso de recepción, la tarea global termina cuando se entrega al programa Supervisor el destinatario del mensaje (activando el BE de la tarea destinatario). Los pasos son:

- a).La recepción de los bytes, armado del mensaje en RAM externa, y validación del CRC lo hace la rutina de servicio a la Int. serial. Fig. 3.8.
- b).Tarea 11 : Convierte de ASCII->Hex El mensaje externa ASCII es depurada de STX, ETX y CRC, y convertida a nibbles hexadecimales en RAM interna.
- c).Tarea 10 : Discrimina según el código del mensaje el BE de la tarea solicitada al Maestro.

Por ejemplo, si se discrimina interpretando que se active el BE T16D, el programa Supervisor ejecutará la tarea de modificación de tabla de horarios con la data recibida Fig. 3.9. Debe llevar la data de RAM interna a su tabla de en RAM externa, termina activando T05D la selección de plan de tarifas para ponerla en vigencia.

### **3.7 Interrupción del reloj RTC y del detector corrector de errores DCU**

Ambos dispositivos, cuando solicitan atención, concurren sobre la interrupción 0 del Maestro. Esta interrupción tiene máxima prioridad. Ver Fig. 3.10. Así sobre una interrupción INTO, se observa en el puerto F1 si la interrupción es generada por el reloj RTC DS1285 ó la unidad de detección y corrección de errores DCU 8206.

#### **3.7.1 Interrupción de DCU por error en RAM externa**

La UDC interrumpe al Maestro cuando encuentra un error en lectura de RAM externa. Sobre la interrupción 'INTO=ERROR', la UDC 8206 avisa al Maestro si el error es corregible (CE=1) se habilita la corrección y luego se emite una alarma indicando el tipo de error. El acceso a los datos externos siempre es por el acumulador, y se sale de la rutina con el dato (byte) correcto en el acumulador.

Si la DCU indica error no corregible (CE=0) se emite una alarma indicando otro tipo de error. An ambos casos, error corregible ó no, la alarma también debe indicar la posición de RAM (dirección) donde ocurre el error. Esto puede llevar al reemplazo de memorias.

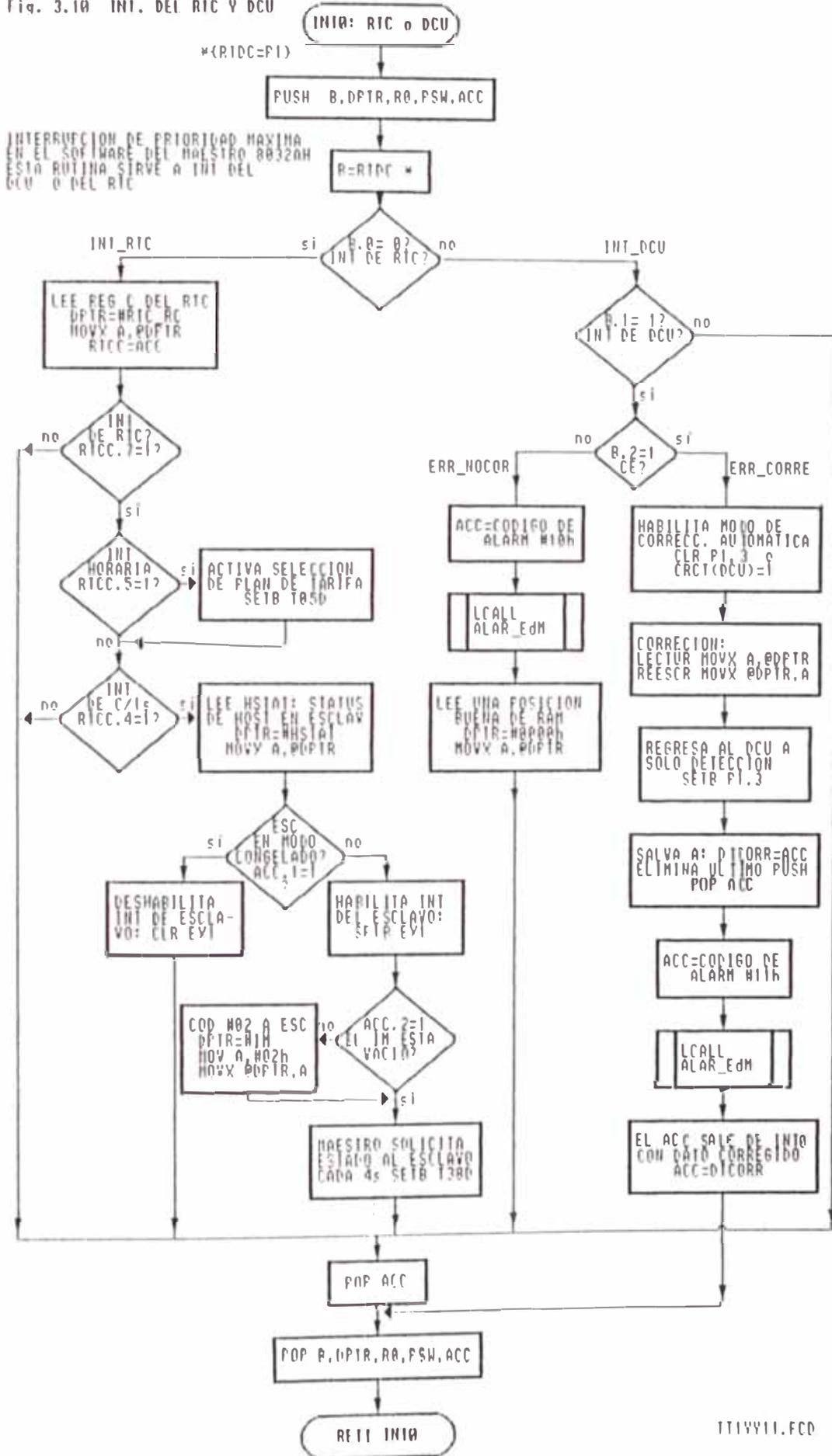
#### **3.7.2 Interrupción del reloj RTC DS1285**

Tiene dos objetos:

1. Dar base de tiempos de 1seg al Esclavo para temporizar los IT ni1, ni2, y
2. Actualiza cada hora el plan tarifario.

Si la interrupción es del reloj, se lee su registro "c" RTCC, y se verifica si es una interrupción por alarma horaria (HH:00:00 cambio de hora), RTCC.7=1, para provocar una actualización del Plan Tarifario i-esimo de acuerdo a la hora vigente (Selección del Plan de Tarifas: ni1 ci1 ni2 ci2). Si así fuese se activa el bit de ejecución T05D.

Fig. 3.10 INT. DEL RIC Y DCU



La interrupción del reloj, también se da cada un segundo,  $RTCC.4=1$ , con esto se avisa al Esclavo que debe iniciar su tarea de temporizado por IT con ni1 y ni2. El aviso se envía por el **registro Comando Inmediato del Esclavo**. Previamente debe verificarse en el registro de estado del host: HSTAT si el Esclavo no está en modo congelado y que el reg. de Comando Inmediato ya ha sido leído y se encuentre vacío para escribir sobre él. Cada 4 segundos solicita al estado del sistema al Esclavo.

### **3.8 Ejecución de plan tarifario**

Establecimos que la tasación se efectúa sólo sobre los abonados que originan la llamada. La tasación de una llamada se **efectúa** por número de pulsos ó también llamados unidades de costo que se acumulan como una cuenta por cada abonado. Existe otro procedimiento que traduce el total de pulsos acumulados por abonado a unidades monetarias. Este procedimiento es separado y totalmente independiente de este proyecto.

La tasación es un procedimiento combinado entre los procesadores Maestro y Esclavo, y para los efectos de tasación, consideraremos:

1. La acumulación de Unidades de Costo por el Maestro
2. La cuenta de tiempo de conversación por el Esclavo

#### **3.8.1 Acumulación de unidades de costo por el maestro**

El Maestro recibe el número de abonado CAIT que justo a iniciado una llamada para cargarle los ci1 pulsos, del PT vigente, a sus acumuladores de unidad de costo  $UC_{CAIT}$  ó **recibe** el número del abonado CAIT al que se le ha vencido (en el Esclavo) conversando su intervalo de tiempo IT ni1 debe **cargarle** ci2 pulsos su cuenta  $UC_{CAIT}$ .

La Fig. 3.11 muestra **los mensajes** que el Maestro recibe del **Esclavo**, de los cuales **interesan** para **incremento** de acumuladores de unidades de costo los mensajes de código 0 (00,CAITLH) ó código 1 (01,CAITLH).

### **3.8.2 Cuenta de tiempo de conversación por el esclavo**

El Esclavo observa el estado de todos y cada uno de los abonados, temporizando con  $ni1$  ó  $ni2$  segundos los abonados que entran en estado de ocupado (conversación), e informa al Maestro sólo de aquellos que inician una conversación ó de los que siguen conversando y se les ha vencido su IT intervalo  $ni$ .

Los procedimientos que sigue el Esclavo hasta generar su reporte al Maestro, son:

- Lectura del estado de abonados sobre INTO
- Autómata discriminador de libre/ocupado
- Temporización ITs (Intervalos de Tiempo) y reportes a Maestro

#### **3.8.2.1 Lectura de estado de abonados por el esclavo**

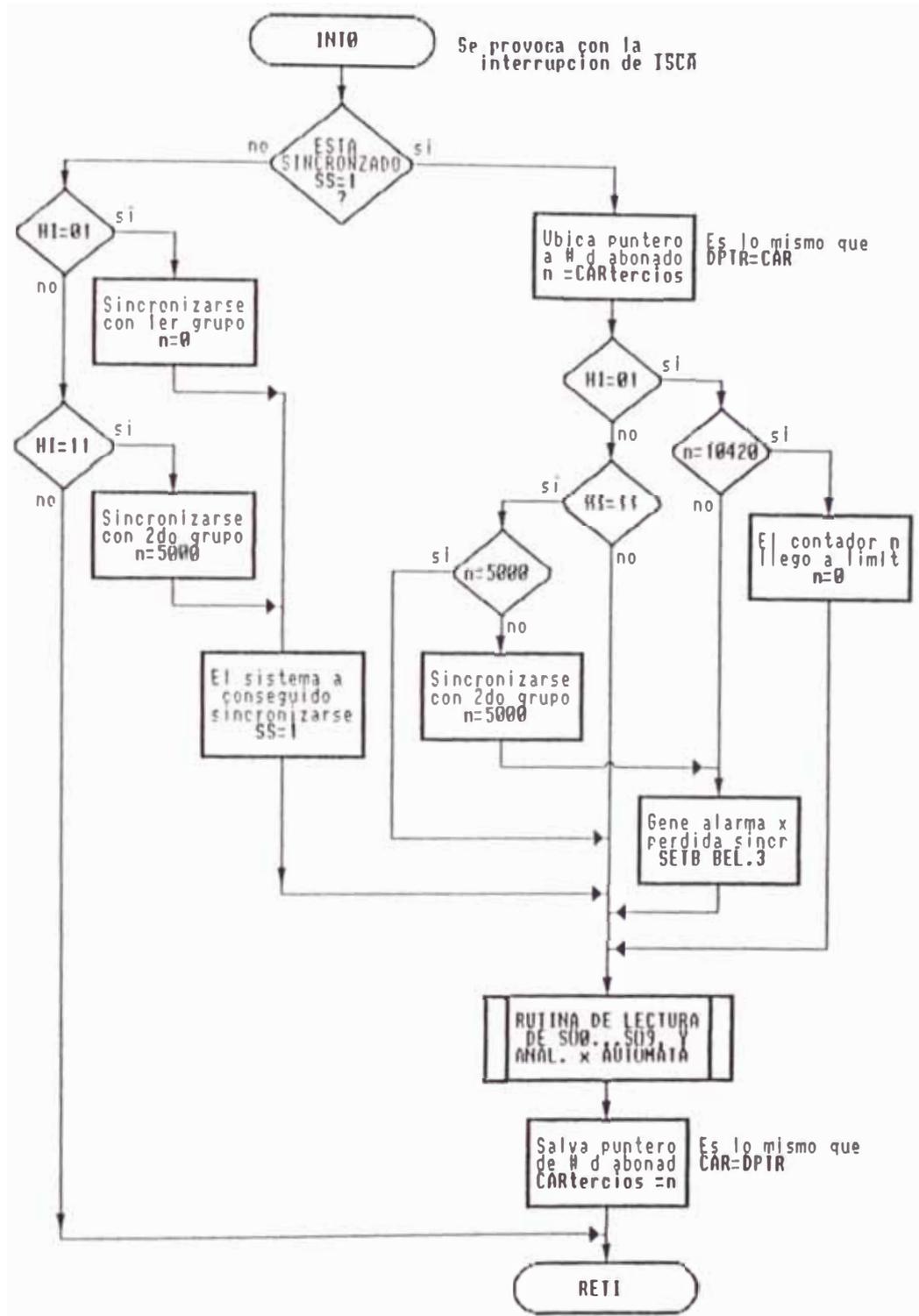
La lectura de  $S00...S09$  ó Ports  $F1$  y  $F4$  será con la interrupción ISCA' cada  $480\mu s$ , ver Fig. 2.1, y se repite para el mismo abonado (para el mismo grupo de 10 abonados) cada  $516mseg$ .

Previo a la lectura, lo inmediato a la interrupción de ISCA' es supervisar el correcto sincronismo de HASCA e INSCA para validar la lectura sobre los primeros ó segundos  $5000$  abonados. Ver el acápite 3.2.2.

En la Fig. 3.12 se muestra que cuando un reset ó Power on, el Esclavo debe iniciar un sincronizado, y vigilarlo con cada nuevo ISCA'. Si lo perdiera, debe resincronizarse y avisar al Maestro para que éste genere una alarma.

La lectura del estado de los abonados significa leer el estado de  $S00...S09$  de los ports  $F1$  y  $F4$  y comparalos con su lectura anterior ( $516ms$  antes) para discriminar un correcto cambio de estado. En la Fig. 3.13 y Fig. 3.14 el procedimiento Autómata Discriminador es el que se encarga de analizar los cambios de  $S0x$  para cada uno de los 10 abonados leídos ( $x=0,1,..9$ ).

Fig. 3.12 INTERRUPCION DE TSCA SOBRE EL ESCLAVO



SS : Bit de sincronizado.  
 HI : IAS  
 I : INS n  
 n : Contador del Num. de Abonado  
 CARtercios : Variable para almacenar n  
 CAR : Variable para guardar el DPTR  
 Como son 3 bytes por abonado,  
 se cumple la relacion: CAR=DPTR=3\*n

### 3.8.2.2 Autómata discriminador de libre/ocupado

Es el procedimiento que discrimina un franco cambio de estado de libre a ocupado y viceversa, de los ruidos, rebotes ó falsos cambios de estado. El procedimiento es ajecutado para cada abonado, usando los bits de control de llamada.

El tiempo de detección de contestación ó fin de conversación es fijada en 1 segundo . Así el Autómata Discriminador debe definir en éste tiempo (dos ciclos de lectura de los 10420 abonados) si el estado del abonado es de conversación o no, llamaremos estado de ocupado ó libre.

#### Bits de Control del Autómata:

- B4 Filtra rebotes de cambio de estado
- B5 Muestra el resultado: libre/OCUPADO
- B6 Almacena el valor anterior de B5.
- B6=1 cuando Autómata aprueba un real cambio de estado.
- B7 Usado para señalarizar 1 seg.

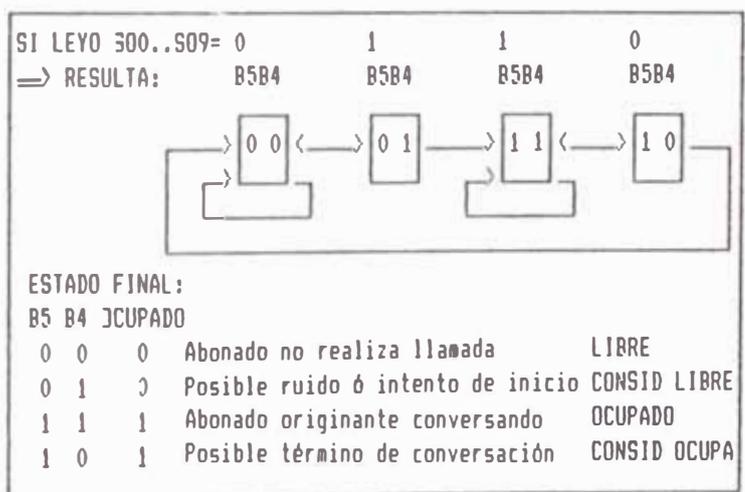
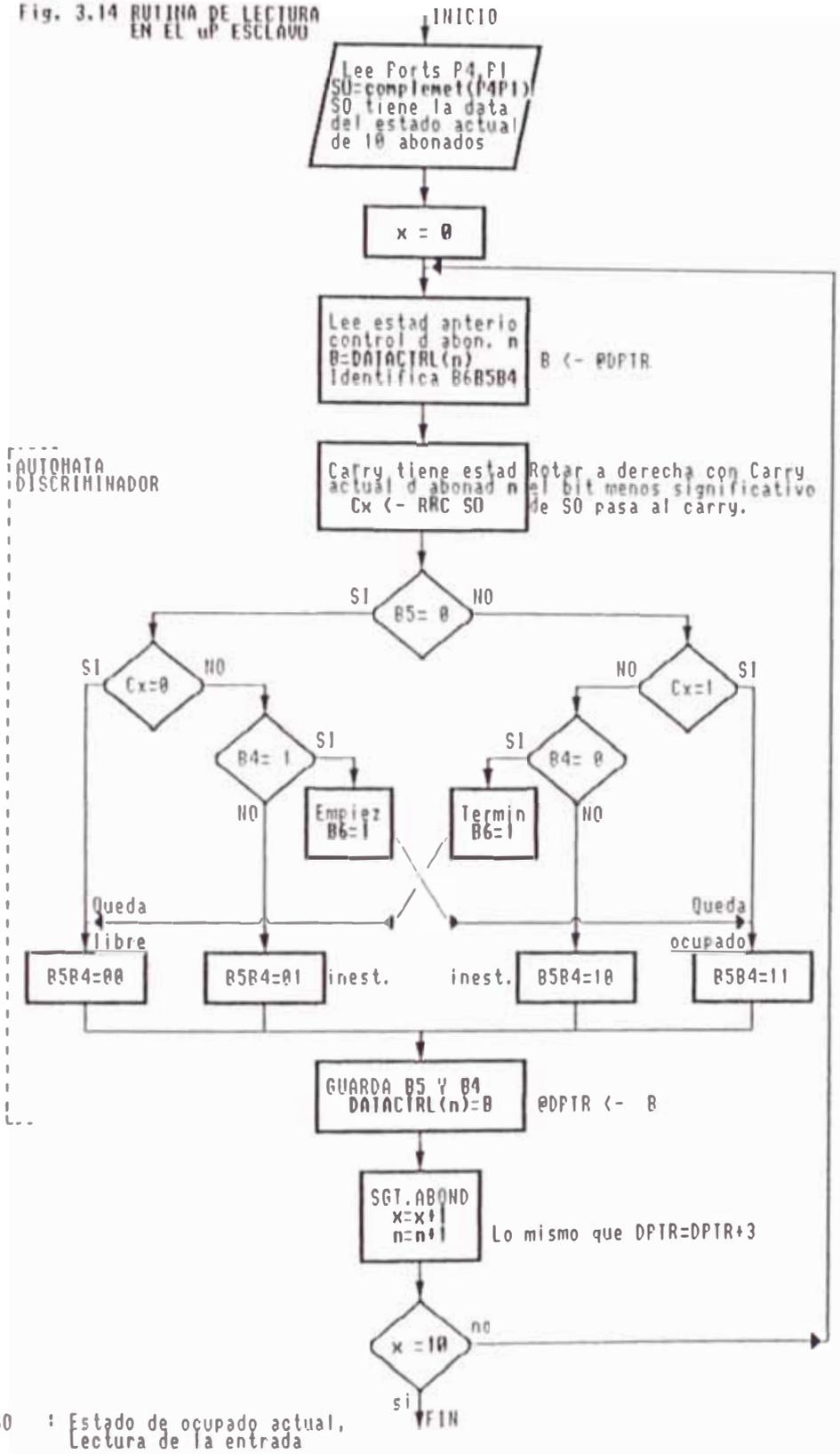


Fig. 3.13 Autómata Discriminador

Con Valor Y si lee S0x=0 Pero si S0x=1

Inic. B5B4	Val.Final B5B4	Val.Final B5B4	B5B4
0 0	0 0	0 1	
0 1	0 0	1 1	
1 1	1 0	1	
1 0	0 0	1 1	

Fig. 3.14 RUTINA DE LECTURA EN EL uP ESCLAVO



S0 : Estado de ocupado actual,  
Lectura de la entrada

x : Contador de bonado, dentro de  
los diez leidos.  
x = 1, ..., 10

B5, B4 : Bits de control del automata

B6 : Bit bandera de franco cambio de estado

Cx : carry contenendo add actual  
del abonado x den ro e os e os  
Cx = S0x S00, S 1, 02, ..., 09

En detalle, podemos decir que la observación de todos los abonados se completa cada 500.16ms, si en éste tiempo alguien cambia de estado, no se considerará válido sino hasta que lo mantenga en el siguiente ciclo. Así, si un abonado en la lectura de SOx (reflejo del hilo "c") cambia de estado por un tiempo menor a 0.516seg, se asumirá que es un ruido, se ignorará para fines de procesamiento. Y para asegurar que una transición de estado es válida, ésta debe durar más de 1.032seg. en el nuevo estado.

### **3.8.2.3 Temporización ITs y reportes a maestro**

La tarea de temporizar los Intervalos de Tiempo IT (Tarea01) de cada abonado es ejecutada por el programa supervisor del Esclavo, y su bit de ejecución BEL.1 se activa cada 1seg. por Comando Inmediato del Maestro.

En la Fig. 3.15, cuando un abonado CAIT empieza una conversación (B6B5=11), se le carga **n1** segundos en su contador IT de RAM externa, y se avisa al Maestro enviándole el código 0 y el número del abonado: 0,CAIT<sub>LH</sub>. El Maestro cargará c1 a la cuenta del abonado CAIT.

Así, cada un segundo se decrementa el tiempo IT del abonado en conversación "ocupado" (B6B5=01).

Cuando el contador IT del abonado CAIT llega a IT=0 y éste sigue conversando, se le carga **n2** segundos en su IT. Se envía al Maestro el mensaje 1,CAIT<sub>LH</sub>. El Maestro cargará c2 a la cuenta del abonado número CAIT. El procedimiento se repite **mientras el abonado siga conversando** (B6B5=01) y su IT decrementado cada segundo **llegue hasta cero.**

Cuando el abonado termina su conversación, de manera similar, por el FIFOout, se avisa al Maestro: 3,CAIT<sub>LH</sub>. El Maestro no hace tasación **con éste dato**, pero puede usarlo en caso de que se le pida generar reporte de eventos.



En este capítulo hemos visto que el software para ambos microcontroladores, ha sido desarrollado teniendo en cuenta una interacción muy frecuente para conseguir entre ambos realizar la tasación del servicio con diversos planes de tarifa. El Maestro atendiendo interrupciones del Reloj, selecciona el plan de tarifa, y hace conocer al Esclavo los nuevos intervalos de tiempo con los que debe medir el consumo (del servicio) de los abonados. El software del Esclavo lee el estado de los abonados, discrimina posibles ruidos ó falsos cambios de estado, y temporiza el consumo para aquellos que entran en estado de conversación.

Es tarea principal del software del maestro, acumular en la cuenta de cada abonado los consumos de servicio vigilados por el Esclavo. Se ha desarrollado también, para el Maestro, el protocolo de comunicaciones que le permite a la Unidad de Control conectarse a un computador e interactuar con la persona usuario que establece los planes de tarifa ó necesita saber el consumo por abonado, etc.

## CONCLUSIONES

1. El sistema de servicio medido de las centrales telefónicas Pentaconta, es potenciado en sus facilidades de tasación e interacción con el hombre.
2. La nueva Unidad de Control para el equipo Telmet le permite una programación más amplia y flexible de planes de tarifas. Se puede programar más planes tarifarios en función del tráfico de las diversas horas del día, y según el tipo de día.
3. El registro del consumo de servicio por duración de llamadas se efectúa sólo sobre los abonados que originaron la llamada, y se **sigue expresando en pulsos** unidades de costo como lo requiere el procedimiento externo que traduce el total de pulsos acumulados por abonado a unidades monetarias. Este otro procedimiento es separado y totalmente independiente de éste proyecto.
4. La información de tasación puede ser leída, almacenada e incluso modificada por un computador con interfaz serial RS-232C.
5. El sistema resulta ser altamente confiable y seguro, y es capaz de emitir alarmas de identificación de error **cada vez que reconozca una falla interna**. Así, la RAM de abonados es controlada por un detector y corrector **de errores, y junto con el reloj**, tienen una batería de respaldo. Un circuito supervisor vigila al software del microcontrolador principal y éste al resto. El software está preparado **para generar las alarmas de error ó falla por el puerto serial**. Adicionalmente se ha diseñado la Unidad de Control **para trabajar** en

paralelo con una gemela procesando simultáneamente ambas la misma data.

6. El uso de microcontroladores ofrece mayores facilidades para el diseño por su alto grado de integración de partes: RAM, UART, temporizadores/contadores, puertos de Entrada/Salida encapsulados junto con el CPU en un mismo IC, al mismo tiempo son cada vez una salida más económica.
7. El proyecto se desarrolla sobre la base de dispositivos de estado sólido que por su alto grado de integración, permiten una construcción sencilla del hardware del sistema, fácil ampliación y diseño de software más flexible.

Se han implementado con éxito, tres prototipos Unidad de Control para los Telmet que sirven a los abonados de la serie 65 y 69 de CPT. Se ha implementado también procedimientos externos para el análisis automático y diario de los datos.

Es posible implementar facilidades adicionales, como contar el total de llamadas, contar las llamadas de larga duración, etc. Se ha dejado el hardware preparado para ello, y el software organizado en tareas bajo un bucle del programa supervisor, el cual permite el fácil uso e inclusión de nuevos módulos.

## BIBLIOGRAFIA

1. 8 BIT EMBEDDED CONTROLLERS  
Intel Corporation
2. ARQUITECTURA DE COMPUTADORES  
M. Morris Mano
3. CMOS DATA BOOK  
RCA Semiconductors
4. DALLAS SEMICONDUCTOR  
DATA BOOK
5. MAXIM NEW RELEASES  
DATA BOOK VOL II
6. PERIPHERALS  
Intel Corporation
7. S-MOS SYSTEM CMOS DATA BOOK  
Silicon System
8. SPRAGUE  
Integrated Circuits  
Data Book
9. THE TIME SAVER  
For 8051 Microcontroller Development  
Archimedes Software Inc.
10. TRANSMISION DE DATOS  
INICTEL
11. WHY MICROCONTROLLERS?  
STEVEN A. CIARCIA  
BYTE, 8 - 1988
12. Z80 APPLICATIONS  
James W. Coffron