

881217  
16  
29

# UNIVERSIDAD ANAHUAC

ESCUELA DE INGENIERIA  
CON ESTUDIOS INCORPORADOS A LA  
UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO



UNIVERSIDAD ANAHUAC  
VINCE IN BONO MALUM

DISEÑO DE EQUIPO PROBADOR DE TARJETAS DE  
ABONADO DE LA CENTRAL TELEFONICA  
"CARMELITA"

T E S I S  
QUE PARA OBTENER EL TITULO DE  
INGENIERO MECANICO ELECTRICISTA  
P R E S E N T A :  
JESUS SALVADOR LOZANO GUAJARDO

México, D. F.

1993

TESIS CON  
FALLA DE ORIGEN



Universidad Nacional  
Autónoma de México



## **UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso**

### **DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis está protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

## I N D I C E

Objetivos	I
Alcances	II
<b>Capítulo I: Introducción</b>	
1.1) Generalidades	1
1.2) Antecedentes	1
1.3) Problemática	7
1.4) Características Técnicas	
<b>Capítulo II: Descripción de la Central Carmelita</b>	
2.1) Generalidades	13
2.2) Descripción de los Módulos de la Central	16
2.3) Arquitectura del Sistema	20
<b>Capítulo III: Descripción de la Tarjeta de Abonado de la Central Carmelita</b>	
3.1) Funciones de la Tarjeta de Abonado	25
3.2) Señalización Externa	32
3.3) Diagrama de Bloques de la Tarjeta	53
<b>Capítulo IV: Hardware del Sistema</b>	
4.1) Descripción de la Prueba	61
4.2) Descripción del Hardware	63

## Capítulo V: Software del Sistema

5.1) Introducción	100
5.2) Teoría de Automatas	100
5.3) Automata del Programa	102
5.4) Algoritmo de la Rutina de Prueba	127
5.5) Programa	136
Conclusiones	159
Nota sobre Diagramas Eléctricos	160
Bibliografía	164

## OBJETIVOS

El Objetivo del presente trabajo es el diseño de un equipo semi-automático de prueba que permita comprobar la calidad de un tipo de tarjeta de una Central telefónica CARMELITA.

El equipo será programable para permitir su uso posterior con tarjetas similares, cambiando la memoria de programa según la tarjeta que se desee revisar.

Diagnosticará las fallas que tenga y las indicará al operario.

## ALCANCES

El diseño que aquí se presenta consiste en la elaboración del Hardware y Software del equipo.

Entendemos por diseño la elaboración de planos completos que especifiquen totalmente la configuración eléctrica de los circuitos y documentos suficientes para especificar completamente el Software.

El trabajo será exclusivamente a nivel de diseño, sin incluir pruebas, prototipos ni documentos de fabricación, y no incluirá el diseño de alimentaciones por tomarse éstas de una fuente ya existente.

La prueba que se realice a la tarjeta será exterior solamente, esto es, a través de las terminales de entrada y/o salida.

El diseño está hecho alrededor del Microprocesador 8035.

## I) INTRODUCCION

### 1.1) Generalidades.

Todo producto terminado que funcione satisfactoriamente es el fruto de esfuerzos que se hicieron anteriormente para llegar a él. Los artículos electrónicos no son la excepción. Desde un radio hasta una Central telefónica, son el resultado de una cadena de actividades.

Muchas veces, al ver una obra consumada, -como puede ser un aparato electrónico terminado-, sólo apreciamos el paso previo a la consecución de dicho fin. Vemos una línea de producción, que arma en poco tiempo grandes cantidades de productos, y podemos fácilmente olvidar que antes de llegar a esos niveles de producción industrial, se necesitó de trabajo intenso en laboratorio: Diseño electrónico, armado de prototipos, pruebas, etc.

A continuación, describiremos los pasos principales que existen desde el inicio de un proyecto hasta su entrega para producción serie. (Trataremos aquí de un proyecto en particular: La Central CARMELITA).

### 1.2) Antecedentes.

#### a) Idea del Proyecto.

Tiene que haber una idea inicial. Ya sea por una proble-

mática que haya en ese momento, o bien para mejorar alguna condición existente, el hecho es que surge la idea.

En este caso, debido a la necesidad de comunicar a las poblaciones pequeñas con el resto del país, surgió la necesidad de una central automática rural de baja capacidad. TELMEX se encargó de este proyecto, al que asignó el nombre de Central CARMELITA.

b) Elaboración de Especificaciones.

TELMEX estableció los requisitos que debe satisfacer esta Central; como son: número de abonados por Central (150), número de troncales (16), teléfonos de teclado de multifrecuencia, capacidad de autodiagnóstico, mantenimiento remoto; destinado a trabajar como Central OTA (Oficina Terminal Aislada); bajo consumo, etc.

c) Selección del Grupo de Trabajo.

Para la realización de la idea, para que no se quede en el aire, se escoge para que trabajen en él a las personas idóneas.

En su mayor parte, este grupo debe estar compuesto por ingenieros capaces, con experiencia y bastante conocimiento de su campo.

Se trata de personal muy calificado de la Empresa.

En este caso, se dejó la responsabilidad de llevarlo a cabo al Centro de Investigación y Desarrollo de TELMEX.



d) Diseño del Proyecto.

Una vez que se ha analizado lo que se pide al proyecto como caja negra, teniendo en cuenta las especificaciones estipuladas se hacen borradores de circuitos. Se hacen cálculos. Se evalúan varias alternativas.

En este caso, personal de TELMEX definió la filosofía de la Central y realizaron después el diseño de los circuitos que la integran.

e) Alambrado del Circuito.

Por ser un proyecto tan grande, se alambró cada circuito en particular y fue probado aisladamente. Se procedió así con el circuito de abonado, el de troncal, los de señalización de abonado y de troncal; fuentes de alimentación, etc.

f) Prueba Alambrada.

Se prueba ahora el Circuito alambrado. Se corrigen las anomalías que se observan. De preferencia se le somete para esta prueba a las máximas condiciones adversas de operación que se pueda.

En este caso, por tratarse de un proyecto muy grande como es una Central telefónica, se prueba cada uno de los módulos aisladamente.

Una vez que cada uno de los módulos funcionó correctamente, se procede al siguiente paso.

g) Tarjetas de Circuito Impreso (Requisito).

Una vez terminadas, se revisan para corregir errores que

podieron existir en su ensamble.

Se mandan a fabricar unas cuantas tarjetas, sólo para el armado de prototipos.

Para la Central Carmelita, es éste un paso largo por la gran variedad de tarjetas que la componen. Se persiguen varias metas, como el tratar de integrar en una sola tarjeta el máximo número de abonados posible, para disminuir el número total de tarjetas de la Central, el costo de los Circuitos Impresos, simplificación de funciones, etc.

#### h) Diseño Industrial.

En esta etapa se elaboran los planos para el Bastidor o Gabinete que constituirá el elemento mecánico del equipo.

Esta armazón deberá proveer lugares adecuados para insertar las tarjetas, interconexión de las mismas, conectores a emplear; así como la adaptación al lugar donde va a estar localizado el equipo, si va a ser transportable o va a permanecer fijo; deberá prever labores de mantenimiento, etc.

Estos planos se envían para la fabricación de unos cuantos Bastidores.

#### i) Ensamblado de las Tarjetas de Circuito Impreso.

Una vez que están listas las Tarjetas de Circuito Impreso y las componentes que van a ir en ellas, se solda cada componente en su lugar en el número adecuado de tarjetas para el armado de prototipos.

Es necesario ensamblar los diferentes tipos de tarjetas. (Por Central):

Abonado (50)  
Troncales (8)  
Receptor DTMF (2)  
R2 (6)  
Generador 425Hz (2)  
Final (1)  
Transmisor DTMF (1)  
Microprocesador (1)  
Extensión de Memoria (1)  
Controlador del Bus Acoplador (1)  
Modem (Para comunicación con el  
Centro de Zona) (1)

j) Armado del Equipo.

Ahora se insertan las tarjetas en el Bastidor de prueba. Se instalan los conectores, así como la alimentación eléctrica del equipo. Ya se tiene un Prototipo.

k) Prueba del Prototipo en Laboratorio.

Se procede a la Prueba del Prototipo. Se le somete a diferentes condiciones de operación para ver cómo responde. Lo importante de este paso es que los circuitos que antes se habfan probado individualmente, ahora se prueban en conjunto. Puede suceder, que aunque cada circuito funcione correctamente, el Prototipo falle por algún error de concepto

general, de sistema.

Puesto que la Central CARMELITA está controlada por un Microprocesador, se observa cómo marcha la ejecución del programa.

Una vez que se ha probado exhaustivamente el Prototipo de Laboratorio, se retroalimentan los resultados obtenidos al Grupo de Diseño para que se hagan las correcciones pertinentes y se pruebe de nuevo al equipo.

Si se observa una falla muy grave a nivel de Sistema, éste es el momento de dar marcha atrás y plantear de nuevo el proyecto.

1) Prueba del Prototipo en Condiciones Reales.

Se lleva un Prototipo al campo para someterlo a sus condiciones reales de operación.

La mayoría de las veces, por más variadas y severas que se hagan las pruebas en laboratorio, no se asemejan a las condiciones reales a las que va a estar sometido el equipo.

Nuevamente, se hacen las modificaciones necesarias en base a los resultados obtenidos.

Una vez que el equipo se ha modificado lo suficiente y pasa con éxito las pruebas por un periodo largo de tiempo, se procede al siguiente paso.

m) Elaboración de la Documentación para Producción Industrial.

Se elaboran todos aquellos documentos, diagramas eléctricos, especificaciones, planos de fabricación definitivos, etc. para que se pueda pasar a la producción en serie de este producto.

### 1.3) Problemática.

La etapa particularmente crítica y sujeta a errores humanos fue el ensamblado de tarjetas de circuito impreso por personal de TELMEX.

El trabajo que hasta ahora se ha descrito se desarrolló en un laboratorio. Las tarjetas que se ensamblaron para el Proyecto en particular fue un pequeño número, sólo el suficiente para armar un Prototipo.

El personal que ensambló los componentes constantemente trabaja en diferentes tipos de tarjetas. Ello provocó que el número de errores que se produjeron en el ensamblado en laboratorio fue mucho mayor que los que se obtienen en producción serie, en la Industria.

El porcentaje de tarjetas buenas después del ensamblado depende de factores tales como Control de Calidad de Entrada y Niveles de Producción, Métodos de Armado, Supervisión, etc.

Para efectos de cálculo, vamos a suponer que 60% de las tarjetas ensambladas en laboratorio no tienen errores, (En la práctica este valor fue superior al 80%).

Un espectro de las fallas que se pueden encontrar a-  
parece en la figura 1.3.

Bajo esta suposición, tenemos que la probabilidad de  
que una tarjeta no presente errores de ensamblado es del 60%.

En un simple producto de sólo 5 tarjetas, armadas con  
este criterio, sin haber sido antes corregidas, observamos  
que:

La probabilidad de que todas las tarjetas que lo compo-  
nen fueran buenas, ésto es, de que no presenten errores, es  
de:

$$\left(\frac{60}{100}\right)^5 = 0.07 = 7\%$$

Luego, podemos predecir que de una muestra suficiente-  
mente amplia, 93 productos de cada 100 que armáramos presen-  
tarían al menos una tarjeta defectuosa.

Supóngase ahora que se tiene delante de uno, un Pro-  
totipo con muchas tarjetas. Si dicho Prototipo no funciona  
correctamente; hallar el error es sumamente difícil y requie-  
re de personal que conozca a fondo el Proyecto. Podríamos  
afirmar que sólo los ingenieros que participaron en el di-  
seño están capacitados para localizarlo (Tiempo de retraso,  
alto costo). Al realizar prueba a nivel de tarjetas re-  
cién ensambladas ya sabemos que las tarjetas que constitu-  
yen el Prototipo están en buen estado. Ya se elimina una  
variable que resolver. Por lo tanto, se emplea menos tiempo

en hacer funcionar el Prototipo. Esto justifica la prueba de tarjetas.

Luego, es necesario examinar cada tarjeta recién ensamblada y corregirla. Este es un procedimiento que consume mucho tiempo, que repercute en retraso del proyecto. Ciertamente lleva menos tiempo que hallar el error en un Prototipo terminado, pero de todas formas, es tardado. Además, requiere que personal muy capacitado, que conozca a fondo el circuito, lo realice. Este personal es caro para la Empresa. Para aprovechar mejor sus recursos, se necesita ocupar a estos ingenieros en labores más interesantes y productivas que corregir tarjetas de circuito impreso.

Como ejemplo, en la Central CARMELITA:

Al armar tan sólo 2 prototipos, se necesitarán probar 100 tarjetas de abonado. Para un ingeniero electrónico que conozca profundamente el circuito, hacerlo manualmente le llevaría unos dos meses.

La solución que proponemos en este trabajo es automatizar esta prueba. Es diseñar un equipo que pruebe las tarjetas automáticamente e indique en qué sección de la tarjeta se localiza una falla, cuando ésta exista.

Con la introducción de este equipo, se pretende disminuir el tiempo de desarrollo del Proyecto y aprovechar mejor el material humano con que cuenta la Empresa.

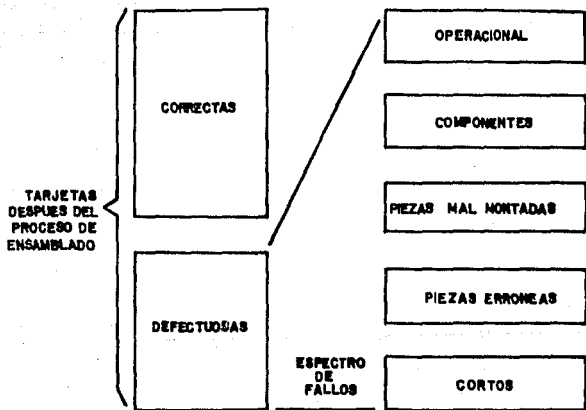


FIG. 1.3



**Características Técnicas del Equipo de Prueba:**

Para probar la tarjeta de abonado, el equipo aquí presentado debe ser capaz de lo siguiente:

a) Simular en la línea telefónica estado de colgado y descolgado;

b) Revisar la línea telefónica atendida por el Circuito de Abonado para verificar que proporcione alimentación eléctrica, que envíe satisfactoriamente la corriente de llamada;

c) Ejecutar operaciones de escritura sobre la tarjeta, ordenándole el envío o la suspensión de la Corriente de Llamada, y habilitando o deshabilitando la Programación del Codec;

d) Que ejecute operaciones de lectura sobre la tarjeta, para supervisar que el estado lógico de sus salidas sea correcto en todo momento.

e) Programar los Codecs en transmisión o recepción en cualquier ranura de tiempo o bien ordenarles que queden en reposo;

f) Enviar un tono por la línea telefónica atendida por la tarjeta (señal analógica) o en el Canal PCM (Señal digital).

g) Detectar el tono en la línea telefónica atendida por la tarjeta o en el Canal PCM.

#### 1.4) Características Técnicas.

Las funciones básicas que desempeña la tarjeta de abonado son las siguientes:

Hacia el abonado:

- Proporcionar alimentación eléctrica requerida por su teléfono.

- Supervisar en todo momento el estado de su teléfono (Colgado o Descolgado).

- Proporcionar la corriente de llamada.

- Supervisión de línea abierta.

Hacia la Central:

- Efectuar la función híbrida (Conversión de 2 a 4 hilos) .

- Efectuar el enlace necesario para la comunicación a través de la Red de Conmutación.

- Informar al Microprocesador del estado en que se encuentra el abonado.

- Ejecutar las órdenes que el Microprocesador le escriba.

## II) DESCRIPCION DE LA CENTRAL CARMELITA

### 2.1. Generalidades

Todos nosotros estamos muy acostumbrados al teléfono. Forma parte esencial del medio en que nos desenvolvemos, en que vivimos. Podemos afirmar que no se puede concebir la vida moderna sin este útil invento. Es ya tan natural y rutinario en nuestras vidas el hacer conversaciones telefónicas que no caemos en la cuenta de lo complejo que es un Sistema Telefónico, así como de la serie de eventos que ocurren al efectuarse una simple conversación.

Vamos a analizar un poco lo que sucede:

Todo teléfono está unido a una Central telefónica por un par de alambres, a los que llamaremos "Hilos de abonado".

Una Central telefónica atiende a un número limitado de líneas o abonados dependiendo de su propia capacidad y dentro de una cierta área, comprendida por un radio generalmente no mayor a 10 km.

La primera función que realiza una Central telefónica es la de supervisar el estado de los teléfonos atendidos por ella. En el momento en que descuelga un abonado, -que por ser el que inicia la llamada le llamaremos abonado A, -la Central detecta este evento, proporciona a su teléfono alimentación eléctrica y le envía un tono continuo, -tono de invitación a marcar, -con el que indica al abonado que está lista para recibir los números que él teclee. Una vez que el abonado A escucha el tono de invitación a marcar, teclea los dígitos que corresponden al número del abonado con quien quiere comunicarse, o abonado B.

La Central tiene la misión de recibir los dígitos y una vez que detecta que la marcación está completa, inicia la búsqueda del abonado B.

Toda Central telefónica se encuentra unida a una o más centrales de la red telefónica por unos canales llamados "troncales" o "hilos de troncal".

Primeramente, la Central decide si B está suscrito a ella o a otra Central. En el supuesto de que B se encuentre suscrito a una Central diversa, entonces la Central de A, -que llamaremos Central origen,- inicia la comunicación con otra Central a la que está enlazada, a la que llamaremos Central destino.

El Protocolo de comunicación se describe a continuación:

Inicialmente la Central origen busca una troncal disponible. Si no existe ninguna, es imposible lograr la comunicación y envía al abonado A un tono interrumpido a intervalos llamado "tono de ocupado". Si existe una troncal disponible, la toma y envía a la Central destino el primer dígito tecleado por el abonado A. Una vez que la Central destino recibe el primer dígito, envía a la Central origen una señal, indicándole que lo ha recibido bien. Entonces la Central origen procede a enviar el segundo dígito y así sucesivamente, hasta completar de enviar el número del abonado B. La Central destino se encarga de seguir la búsqueda, pero la troncal se queda pendiente para establecer la comunicación entre los abonados en caso de que ésta se llegue a efectuar.

En el caso de que el abonado B estuviera suscrito a la misma Central que A, entonces ésta investiga si existe alguna ruta

disponible por la cual se pudieran enlazar A y B. Si las rutas están congestionadas por el alto tráfico telefónico envía al abonado A tono de ocupado. En el caso de que existiera alguna ruta disponible, la Central la separa e investiga el estado en que se encuentra el teléfono del abonado B. Si está libre (colgado), entonces activa su campanario con la "corriente de llamada", indicándole que tiene una llamada entrante; simultáneamente envía a A un tono con intervalos de silencio, llamado "retorno de llamada", con el que A se entera de que se está activando el campanario de B y que en caso de que éste descuelgue, se puede efectuar la conversación.

Cuando B descuelga, la Central deja de enviar la corriente de llamada y el tono de retorno de llamada, proporciona alimentación eléctrica a B y los conecta a la ruta separada previamente por ella, pudiéndose así establecer la conversación.

La conversación ocupará la ruta asignada todo el tiempo que dure. Una vez que uno de los 2 abonados cuelgue, la Central da por acabada la conversación, deshace la ruta que conecta a ambos abonados, dejándola lista para otra conversación.

Ahora bien, si B estuviera ocupado, -teléfono descolgado, -en el momento en que la Central investiga su estado para ver si es factible la comunicación, entonces envía a A tono de ocupado. El abonado A colgará y realizará otro intento más tarde.

Podemos clasificar las funciones que desempeña una Central telefónica dentro de estas 2 categorías:

a) Funciones inteligentes: Involucra a todas aquellas funciones de búsqueda, supervisión y decisión. Por ejemplo: Búsqueda

de rutas disponibles, supervisión del estado de abonados, decisión de enviar tono de ocupado a A cuando la línea del abonado B está ocupada, etc.

b) Funciones de esclavo: Aquí se incluyen todas aquellas funciones que no requieren de "inteligencia" y que más bien dependen de algún órgano inteligente de la Central, como por ejemplo: Generación de tonos de ocupado, retorno de llamada; amplificación de la voz, recepción de dígitos tecleados por abonado y traducción a un lenguaje inteligible para la Central, etc.

## 2.2 Descripción de los Módulos de la Central

Ya hemos visto a grandes rasgos las funciones que desempeña una Central telefónica. Se entiende la necesidad para el desempeño de estas funciones de dividir las funciones en módulos, de tal forma que cada módulo se encargue de parte de estas funciones. De esta suerte, tenemos los siguientes módulos:

### 1) Módulo de Abono

También conocido como Circuito de Abonado. Existe uno por cada abonado o línea que atiende la Central. Por ser el Circuito sobre el que efectuaremos la prueba, hablaremos más a fondo de él en el siguiente capítulo. Ahora nos limitaremos a decir que básicamente su función es proporcionar alimentación eléctrica al teléfono, amplificar la voz de su abonado, cambiar el estado lógico de sus salidas cuando descuelga el abonado, enviar corriente de llamada cuando así se le ordene, etc.

## 2) Módulo Generador de 425 Hz.

Forma parte de una categoría conocida como: Circuitos de señalización de abonado.

El tono de 425 Hz. es el que escuchamos en la bocina del teléfono al descolgarlo. Es el medio que emplea la Central para proporcionar información al abonado.

El módulo genera las diferentes cadencias:

TIPO	SIGNIFICADO
Tono de invitación a marcar	Indica al usuario que puede comenzar a teclear dígitos.
Tono de Ocupado	Indica al abonado A que el abonado B está ocupado.
Tono de Congestión	Indica al abonado A que no existe una ruta disponible entre él y el abonado B por el alto tráfico telefónico.
Tono de Intervención de Operadora	Indica al abonado que la Operadora está interviniendo en su conversación.

## 3) Módulo DTMF

También es un Circuito de señalización de abonado.

En el proceso de una conversación, el usuario envía a la Central unos dígitos. En las Centrales antiguas, cuyos teléfonos tienen disco de marcación, este envío se realiza interrumpiendo

un determinado número de veces, -según el número de que se trate, - la corriente que circula por el bucle de abonado (llamamos bucle de abonado al conjunto de línea y aparato telefónico).

En las Centrales modernas, cuyos teléfonos tienen teclado en lugar del disco convencional, cada dígito tiene asignados 2 tonos. Al pulsar una tecla, el abonado envía a la Central los 2 tonos que la identifican (Multifrecuencia) por la línea telefónica.

La misión de este módulo es recibir la Multifrecuencia enviada por el abonado y traducirlo a su salida en un número en binario, según el dígito que se trate.

#### 4) Módulos R2

Estos módulos pertenecen a la categoría conocida como: Circuitos de señalización de troncal. Desempeñan la función de enviar y recibir Multifrecuencias por las troncales según el protocolo de comunicación entre Centrales. Para ello, se sigue una norma conocida como R2.

Estos módulos son: Transmisor R2 Adelante, Receptor R2 Adelante, Transmisor R2 Atrás, Receptor R2 Atrás.

En la fig. 2.1 se muestra la forma como trabajan estos módulos.



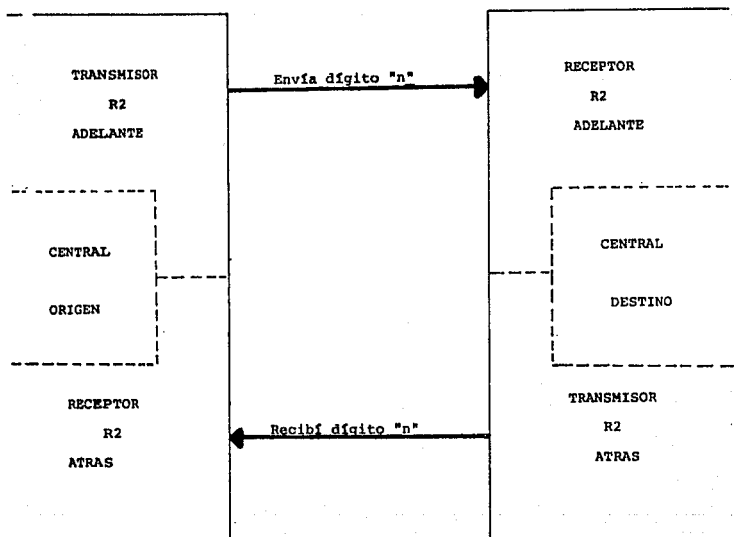


FIG. 2.1

### 5) Módulo de Troncal

Existe uno para cada troncal que tiene la Central y que la enlaza con otra Central. Este módulo es propiamente la vía por la cual viajan:

a) Multifrecuencias necesarias para envío de la información sobre la identificación del abonado con quien se desea iniciar una conversación;

b) En caso de que ésta se llegue a efectuar, la voz de los abonados mientras ésta dure.

Además, bajo las órdenes de la Central a la que está conectado, el módulo avisa a la Central distante que desea ocupar la troncal cuando así se le solicite.

A su vez es capaz de recibir la solicitud de toma de troncal de la Central distante y avisar a su Central para que ésta atienda a dicha petición.

Por todo lo expuesto anteriormente, se infiere la necesidad de estos módulos, que deben tener ciertas entradas y salidas, pudiendo ser éstas en general, analógicas o digitales.

### 2.3 Arquitectura del Sistema

Los módulos que componen la Central necesitan la coordinación de un controlador, de un órgano inteligente, capaz de escribirles órdenes y leer información sobre los mismos. Para la comunicación con este órgano inteligente, - que en la Central Carmelita es un Microprocesador 8086 de Intel, - todos estos módulos se conectan a un grupo común de líneas que se llama "Bus Red" o "Bus Acoplador".

El Microprocesador, como Órgano inteligente de la Central, requiere de una memoria PROM donde esté almacenado el programa, una memoria RAM donde pueda almacenar datos; es deseable también que la Central tenga órganos de entrada y salida como un teclado, visualizador, modem; para la comunicación del Microprocesador con toda esta serie de elementos se emplea un grupo de líneas que reciben el nombre de: Multibus Intel.

Ahora bien, la pregunta natural ahora es: ¿Porqué la existencia de 2 buses distintos: Bus Red y Multibus Intel?

¿Cómo se comunica el Microprocesador con los diversos módulos que componen la Central, si éstos se localizan en diversos buses?

La Respuesta es la siguiente: La existencia de tantos módulos provoca que las líneas del Bus Red tengan una cierta longitud que les introduce una resistencia y capacidad parásitas. Dicha constante RC del Bus Red lo limitan en frecuencia y por lo tanto las operaciones de lectura y escritura que realice el Microprocesador no podrán ser a la frecuencia máxima de trabajo de éste, 10 MHz. Se requiere, pues, de una interfaz entre ambos buses que dé la temporización adecuada para una lectura y escritura correctas.

Por otra parte, los diversos módulos que componen la Central están mapeados en RAM. Esta interfaz también decodifica las direcciones provenientes del Microprocesador y a sus salidas direcciona en forma directa a cada módulo, evitando así la introducción de circuitos integrados decodificadores en cada uno de ellos.

En resumen, ambos buses sí están efectivamente comunicados,

Pero a través de una interfáz.

También hemos hablado que una de las funciones que desempeña una Central telefónica es la de establecer rutas por las cuales se establezca comunicación entre los abonados. En general, podemos extender este concepto al establecer rutas para la intercomunicación de los diversos módulos entre sí; por ejemplo del Módulo que genera el tono de invitación a marcar a un módulo de abonado en particular; de un módulo de abonado al módulo receptor de dígitos telefónicos, de un módulo de abonado a otro. Para efectuar esto, los módulos se encuentran conectados a un grupo común de líneas que reciben el nombre de Bus PCM. En realidad toda la comunicación se realiza por un solo hilo conductor, los demás hilos del Bus PCM son para señales de sincronización. La forma en que se consigue que diversas conversaciones viajen por el mismo cable es por medio de la técnica de Multiplexado en el tiempo. Las señales analógicas internas de cada módulo se codifican en pulsos y se envían o reciben en la "ranura" de tiempo correspondiente. Dos módulos que se intercomuniquen entre sí deben estar programados para que uno transmita y el otro reciba en la misma ranura de tiempo, durante la cual los restantes módulos deben estar en un tercer estado de alta impedancia para no afectar la comunicación. A esta técnica de la conexión entre abonados por medio de multiplexado en el tiempo, compartiendo diversas conversaciones en el mismo cable se le conoce como conmutación temporal.

La Central Carmelita está organizada alrededor de los 3

Buses descritos previamente:

- a) Bus Red o Bus Aclopador
- b) Multibus Intel
- c) Bus PCM

Todas las señales que manejan estos buses son digitales.

Todos los módulos que componen la Central tienen acceso a uno o más de los buses anteriormente descritos y en general pueden tener además otras entradas y salidas, analógicas, como son los hilos de habla que conectan al teléfono con su módulo de abonado.

Todo lo expuesto anteriormente, se puede apreciar mejor si se hace referencia a la fig. 2.2.

En ella pueden apreciar los módulos que ya se han explicado en este capítulo; el concepto de Filtro y Codec se explicará más adelante.

El bloque "Controlador Bus Acoplador" es propiamente la interfaz entre el Multibus Intel y el Bus Red.

Obsérvese que los únicos módulos conectados al Bus Red con salida al exterior son el de Abonado y el de Troncal.

La forma en que otros módulos logran tener acceso a los hilos de abonado o de troncal es por medio de la red de conmutación temporal.

ARQUITECTURA CENTRAL CARMELITITA

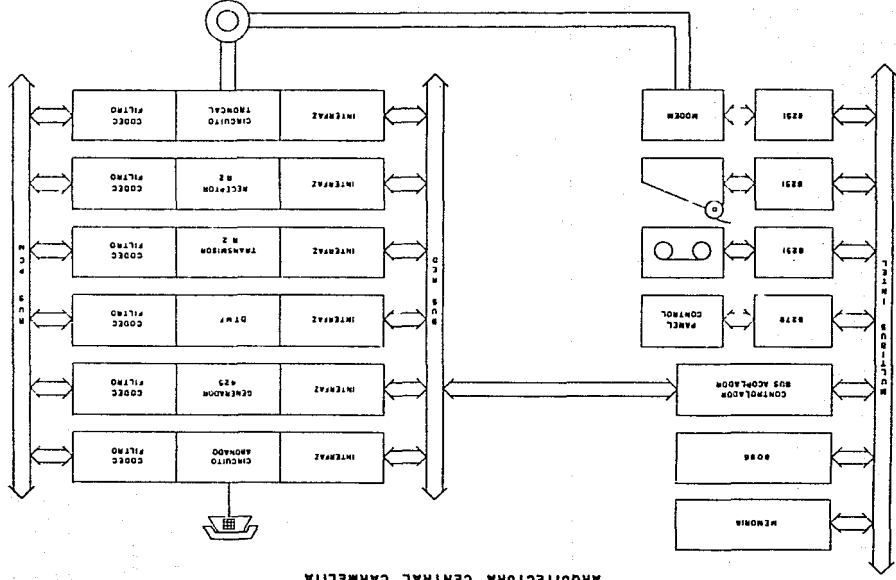


FIG. 2.2

### III) DESCRIPCION DE LA TARJETA DE ABONADO DE LA CENTRAL CARMELITA

#### 3.1) Funciones de la Tarjeta de Abonado

En el Capitulo anterior hablamos de las funciones de una Central telefónica. Para realizarlas la Central está dividida en módulos, encargándose cada uno de una misión en particular. Ahora hablaremos más a fondo del Módulo de Abonado, también conocido como Circuito o Interfaz de Abonado.

Cada abonado de la Central tiene un Circuito asignado a él individualmente y que es al fin y al cabo, el que lo atiende y le sirve de interfaz con el resto de la red telefónica.

Para que se pueda realizar satisfactoriamente una conversación, el teléfono del usuario deberá tener la alimentación eléctrica que le permita transmitir su voz hacia la Central y recibir la voz de la persona con quien está hablando. Una de las funciones del Circuito de Abonado es proporcionar esta alimentación al teléfono.

Cuando el abonado está colgado y tiene una llamada entrante, alguien desea comunicarse con él, - el Circuito de Abonado se encarga de avisarle activando el campanario de su teléfono. Este se activa enviando al teléfono una corriente que se llama convencionalmente: "Corriente de Llamada".

Otra función muy importante del Circuito es la de supervisar en todo momento el estado del teléfono del usuario. Así pues, debe detectar el descuelgue del mismo y avisar inmediatamente a la Central.

En el curso de una conversación normal, cuando el Circuito envía al teléfono del usuario la corriente de llamada, debe ser capaz de detectar el momento en que es descolgado el teléfono tanto si esto ocurre mientras está activado el campanario o en un intervalo de silencio. Debe también enterarse del momento en que se cuelga el teléfono, indicando de esa forma el abonado que desea acabar la conversación.

Una de las funciones más características que se realiza en telefonía es la función híbrida que explicaremos a continuación:

Supongamos que 2 abonados están efectuando una conversación, pero ya sea por la relativa cercanía entre ambos o el buen estado de la línea que los enlaza se entiende perfectamente, sin necesidad de amplificar la señal que viaja por la línea ya sea en un sentido o en otro.

El Diagrama de Conexión se indica en la fig. 3.1



Fig. 3.1

Ahora supongamos que los 2 abonados se encuentran muy apartados uno del otro. Si los conectamos así directamente, su conversación va a ser ininteligible, puesto que la línea que los une tiene una cierta impedancia que atenúa fuertemente la señal. Luego es necesario amplificarla a cada determinada distancia, según las



características de la línea. Sin embargo, nos topamos con un problema. En el par de hilos que unen a los abonados se encuentran mezcladas las señales que viajan en ambos sentidos. Si quisieramos insertar directamente en el par telefónico un amplificador, habría primero que pensar cuál es la entrada y cuál es la salida. Puesto que si abrimos imaginariamente la línea vemos que dicho par de terminales es tanto entrada como salida. Entrada de la señal de voz que debe escuchar el abonado y la salida es la voz que él habla.

Una solución, rápida pero errónea sería la de conectar 2 amplificadores, entrada con salida (Fig. 3.2)

En efecto, se observa que este sistema sería inestable y oscilaría fácilmente.

Para solucionar este problema en Telefonía se emplea la función Híbrida, que es la conversión de 2 a 4 hilos (Véase la Fig. 3.3).

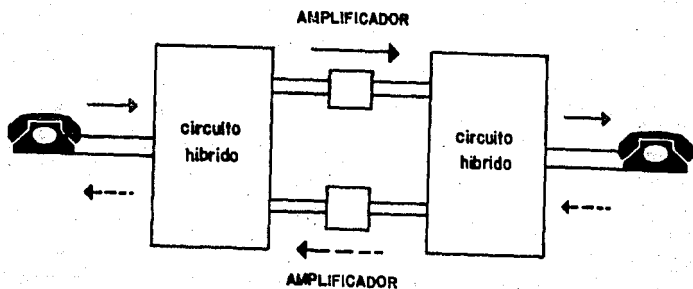
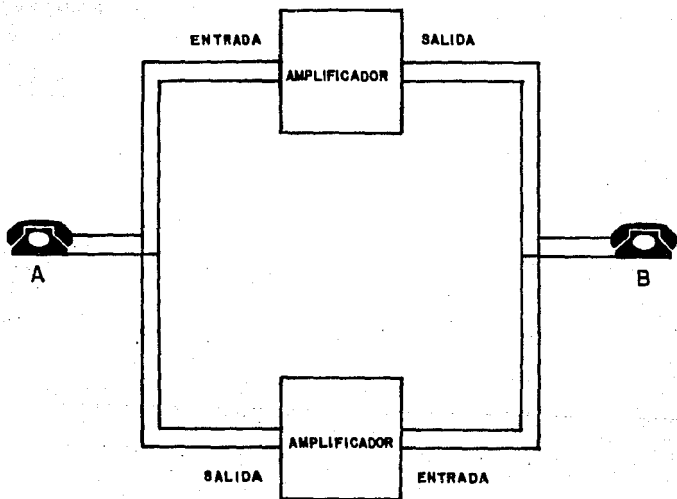


Fig. 3.3



La Híbrida realiza la función de separar la señal que viaja en un sentido de la que viaja en el otro sentido. Así ya se pueden amplificar por separado, sin peligro de oscilaciones.

El Circuito de Abonado está encargado de esta función.

Puede suceder que el bucle de abonado (llamamos bucle al conjunto de hilos y aparato) se abra por algún accidente y quede así aislado el usuario. El Circuito de Abonado debe ser capaz de detectar esta anomalía y dar aviso a la Central.

Dado que los abonados están enlazados a la Central por un par de alambres, cabe la posibilidad de que durante una tormenta eléctrica se induzcan voltajes muy altos o bien que por alguna razón la línea telefónica haga contacto con las líneas de distribución de energía. En cualquiera de los casos las consecuencias para los circuitos de estado sólido que constituyen la Central podrían ser fatales. Es por ello que el Circuito de Abonado debe contar con protección que limite tanto el voltaje como la corriente que absorban las líneas telefónicas.

Por último, el Circuito debe tener acceso a la Red de Conmutación, conectándose y desconectándose según sea necesario.

Recordando un poco el Capítulo anterior, vemos que en el proceso de una conversación es necesario que exista una comunicación interna entre los diversos módulos que componen la Central.

Vamos a suponer que un abonado desea iniciar una llamada. El Módulo Generador de Tono debe enrutarse internamente con su Módulo de Abonado para proporcionarle Tono de invitación a marcar.

Posteriormente se deshacerá esta ruta y se establecerá otro camino entre el Circuito de Abonado y el Módulo DTMF para detectar los dígitos que teclea el usuario. Una vez que éste termina la marcación, se deshace esta ruta y se establece otra entre el abonado A y el abonado B, si se encuentra en la misma Central o entre el abonado A y un Módulo de Troncal, si el abonado B está atendido por otra Central.

La Red de Conmutación tiene la misión de ofrecer estas rutas para la comunicación interna entre módulos y además deben ser éstas suficientes para manejar satisfactoriamente el tráfico telefónico, esto es, varias conversaciones simultáneamente.

En la Central que estamos estudiando, todas las conversaciones internas de los módulos se hacen a través del mismo hilo, pero en intervalos de tiempo diferentes. Esto es, se emplea la técnica de Multiplexado en el tiempo. Se dice que el hilo conduce el Canal PCM.

El Circuito muestrea la señal que va a transmitir, la codifica en una palabra de 8 bits y la transmite en la "ranura" de tiempo asignada únicamente a él por el Microprocesador. Así tiene que ser, puesto que la técnica de Multiplexado en el Tiempo exige que en cualquier momento solo se transmita una palabra. En otro caso, se perdería la señal.

Este proceso de Muestreo, Codificación y Transmisión se hace 8,000 veces por segundo. ¿Porqué 8,000 veces? ¿Porqué no más o menos veces?

La respuesta nos la da el Teorema de Muestreo:

Si una señal no contiene componentes de frecuencia para  $|f| \geq W$ , se describe en forma completa por medio de valores muestra instantáneos espaciados de manera uniforme en el tiempo con -

período  $T_s \leq 1/2W$ . Si una señal ha sido muestreada a la velocidad de Nyquist o a una mayor ( $f_s \geq 2W$ ) y los valores muestra se les representa como impulsos ponderados, la señal puede ser reconstruida en forma exacta a partir de sus muestras por medio de un filtro pasabajas ideal de ancho de banda  $B$ , donde  $W \leq B \leq f_s - W$ . \*

Puesto que el ancho de banda telefónica es de 300-3,400 Hz., una frecuencia de muestreo de 8,000 Hz asegura una transmisión sin distorsión.

Como el Canal PCM trabaja a una frecuencia de 2.048 Mbps y cada "ranura" consiste de 8 bits, tenemos que cada ranura ocupa (8,000) 8 = 64 Kbps.

Frecuencia de la ranura = 8,000 veces por segundo

Bits por ranura = 8

Luego:

$$2,048/64 = 32$$

Frecuencia de trabajo del Canal PCM en Kbps = 2,048

Espacio ocupado por una sola ranura en Kbps = 64

Ranuras disponibles = 32

Al conjunto de 32 ranuras de tiempo se le conoce como "Trama".

Ahora bien, una conversación normal entre 2 abonados requiere de 2 ranuras: Una para transmisión de A a B y otra para transmisión de B a A. Por lo tanto esta Central tiene una capacidad máxima de manejo de 16 conversaciones simultáneas.

\*Black (1953, Cap. 4)

$f_s$  = frecuencia de muestreo

### 3.2) Señalización Externa

#### 3.2.1) Descripción de terminales

La Tarjeta de Abonado contiene 3 Circuitos de Abonado. Esto es, el Circuito de Abonado cuya función describimos en detalle en el aparato anterior, ocupa físicamente el espacio de 1/3 de la Tarjeta de Abonado. Por lo tanto, una Tarjeta es capaz de atender a 3 abonados.

La Tarjeta tiene una serie de terminales que la habilitan para comunicarse con el mundo exterior y que puede ser de entrada y/o salida. Las terminales se dividen en 2 Grupos:

1) Terminales que se conectan al Bus Red y a través de las cuales el Módulo de Abonado intercambia información con el Microprocesador, recibe alimentación eléctrica y tiene acceso a la Red de Conmutación;

2) Terminales que se conectan a los hilos de habla de los usuarios atendidos por la Tarjeta.

En la página siguiente se incluye una tabla en la que se muestra la asignación de las terminales de la tarjeta que se conectan al Bus Red.

ASIGNACION DE TERMINALES DEL BUS RED (Conector 1)

1	Tierra Digital	2	Tierra Digital
3	+5V	4	+5V
5	+5V	6	+5V
7	+12V	8	+12V
9	+15V	10	+15V
11	Tierra Digital	12	Tierra Digital
13	-----	14	-----
15	LEC/	16	ESC/
17	CLKx	18	CLKr
19	Dx	20	Dr
21	FSx	22	FSr
23	Dc	24	CLKc
25	RESET C	26	RESET I
27	+5V	28	+5V
29	-----	30	-----
31	SE E	32	SE F
33	SE C	34	SE D
35	SE A	36	SE B
37	SE 8	38	SE 9
39	SE 6	40	SE 7
41	SE 4	42	SE 5
43	SE 2	44	SE 3
45	SE 0	46	SE 1
47	SG E	48	SG F
49	SG C	50	SG D
51	SG A	52	SG B
53	SG 8	54	SG 9
55	SG 6	56	SG 7
57	SG 4	58	SG 5
59	SG 2	60	SG 3
61	SG 0	62	SG 1
63	INF 6	64	INF 7

65	INF 4	66	INF 5
67	INF 2	68	INF 3
69	INF 0	70	INF 1
71	Tierra Digital	72	Tierra Digital
73	-5V	74	-5V
75	-12V	76	-12V
77	Tierra Digital	78	Tierra Digital
79	Tierra Analógica	80	Tierra Analógica
81	-48V	82	-48V
83	-48V	84	-48V
85	Tierra Analógica	86	Tierra Analógica



Obviamente surgen muchas dudas al observar la Tabla anterior.  
Trataremos de aclararlas:

Terminales	Significado
1 - 12	
27,28	Alimentación Eléctrica
73-86	

#### SEÑALES DIGITALES

15 LEC/	Señal de Lectura emitida por el Microprocesador para los Módulos de la Central.
16 ESC/	Señal de Escritura emitida por el Microprocesador para los Módulos de la Central.
19 Dx	(Estas terminales van unidas)
20 Dr	Línea que conduce todas las conversaciones internas de los módulos multiplexadas en el tiempo (Canal PCM).
17 CLKx	(Estas terminales van unidas)
18 CLKr	Reloj que sincroniza el envío y recepción de los bits del Canal PCM.
21 FSx	Pulso que sincroniza el inicio
22 FSr	de Trama (Terminales unidas)

23 Dc	Terminal através de la cual se le programa al Circuito de Abonado una ranura de transmisión, de recepción o se le manda quedar en reposo.
24 CLKc	Reloj para la programación del Circuito de Abonado
25 RESET C	Deshabilita la programación del Circuito de Abonado.
26 REET I	Elimina estado de corriente de llamada.
31-46 SE n	Señales empleadas por el
47-62 SG m	Microprocesador para direccionamiento de los módulos, puesto que cada uno tiene asignado un Grupo y Elemento únicos.
63-70 INF	Líneas a través de las cuales el Microprocesador lee o escribe datos a los módulos de la Central.

En la tarjeta de abonado se emplean solamente tres líneas para lectura:

DPN/	Estado del abonado sin corriente de llamada (Colgado o Descolgado)
DPI/	Estado del abonado con corriente de llamada
CPI/	(Colgado o Descolgado)

Y para escritura se emplean solamente 2:

PCO/                   Habilita a la Programación del Circuito de  
Abonado.

CLL/                   Envía corriente de llamada.

### 3.2.2) Diagramas de Tiempos

Mostraremos a continuación Diagramas de Tiempos de las operaciones que se pueden realizar sobre la Tarjeta de Abonado.

En la fig. 3.4 se puede observar una operación de lectura sobre el Módulo de Abonado. Nótese que la información interna del estado del abonado aparece en las terminales de salida solo durante la operación de lectura. En todo otro momento el Circuito presentará un estado de alta impedancia en dichas terminales.

En la fig. 3.5 aparecen algunos cálculos que se emplearon en la determinación de los tiempos.

En la fig. 3.6. podemos apreciar una operación de escritura sobre el Módulo de Abonado. Una vez que el Microprocesador ha ejecutado una orden de escritura sobre el Módulo de Abonado, la información queda almacenada internamente en la tarjeta. Por ello es que en el Diagrama aparecen 2 señales: INF INT e INF EXT.

INF INT = Información almacenada en el Módulo por la  
Operación de Escritura.

INF EXT = Información presente en las terminales de la tarjeta  
solo durante la Operación de Escritura.

En la fig. 3.7 se muestran algunos cálculos utilizados para establecer los tiempos específicos en la fig. 3.6

En la fig. 3.8 se enseña el Diagrama de Tiempos para la programación del Circuito de Abonado para establecer o deshacer rutas en la Red de Conmutación.

La información presente en la terminal Dc es válida para el Circuito solo en las transiciones negativas de CLKc.

Los primeros 2 bits definen si los 6 bits restantes se aplicarán para transmisión y recepción (00), transmisión solamente (01), recepción solamente (10) o si el Circuito deberá estar en reposo, aislándose de la Red, en cuyo caso los siguientes 6 bits son irrelevantes.

Los 6 bits restantes definen la asignación de la ranura de tiempo, desde 000000 (ranura 1) hasta 011111 (ranura 32).

Las señales de sincronía y de datos en el Bus PCM están indicadas en la fig. 3.9 En este caso la transmisión tiene lugar en la ranura 2 y la recepción en la ranura 3.

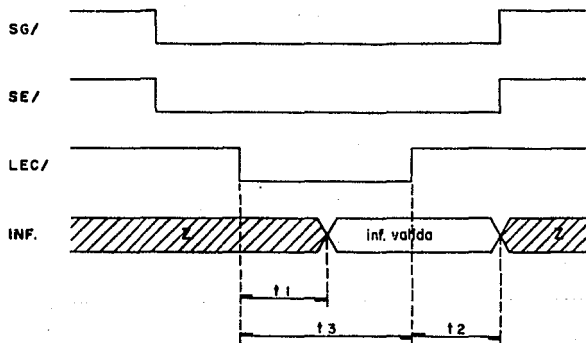


FIG. 3.4

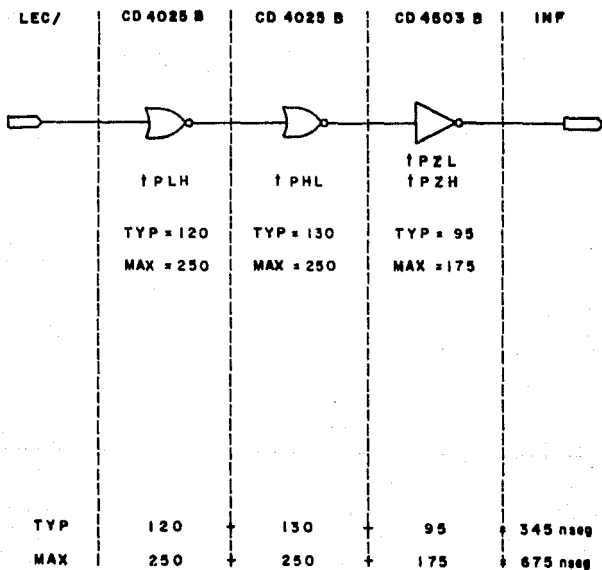


FIG. 3.5 (a)

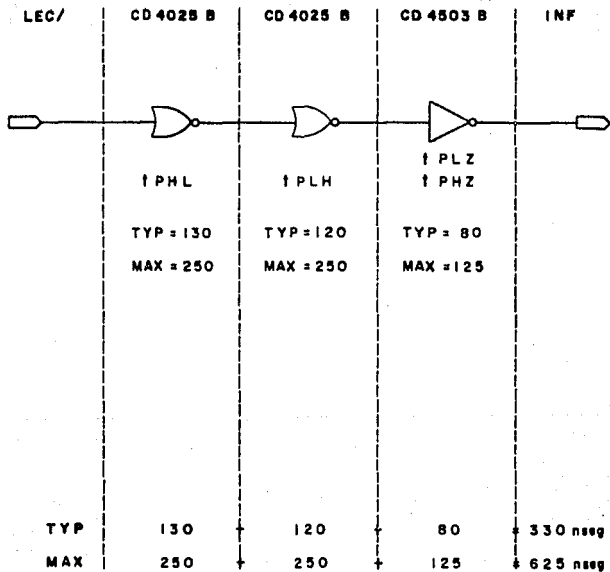


FIG. 3.5 (b)

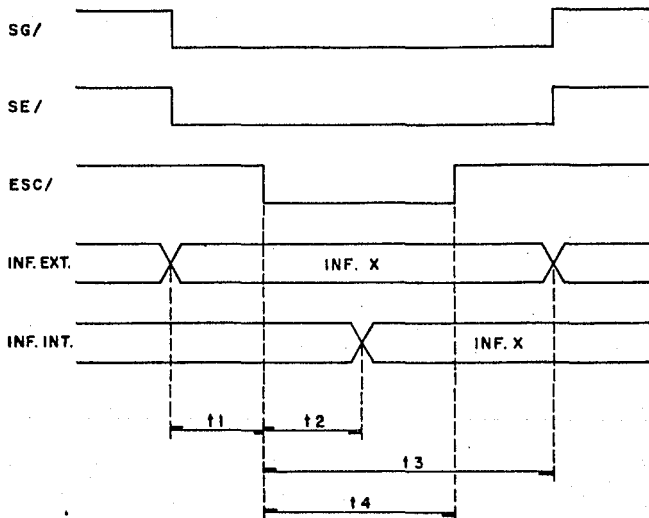


FIG. 3.6



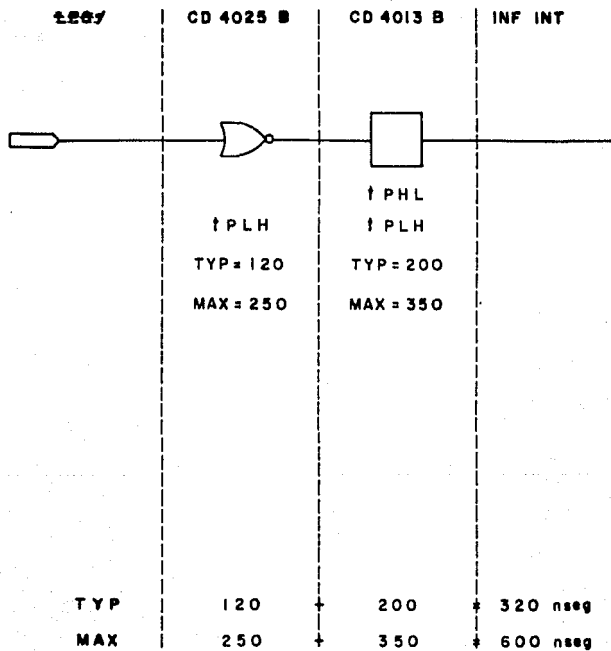


FIG. 3.7

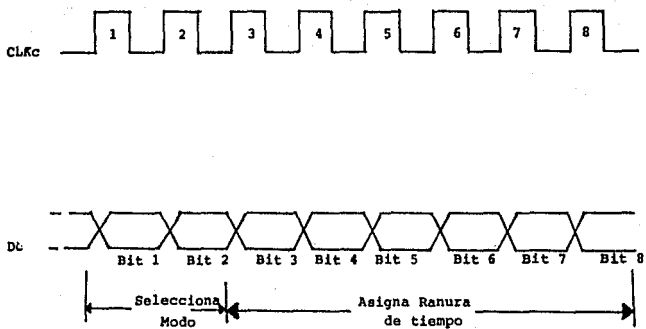


FIG. 3.8

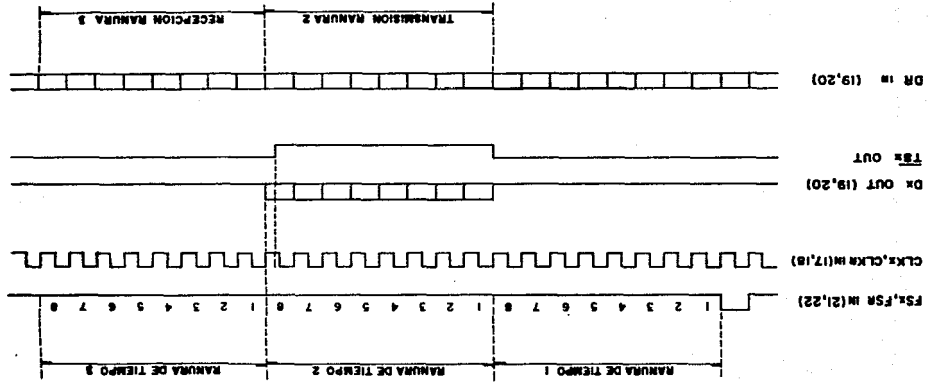


FIG. 3.9

DIAGRAMA DE TIEMPOS DEL ENVIO DE CORRIENTE DE LLAMADA

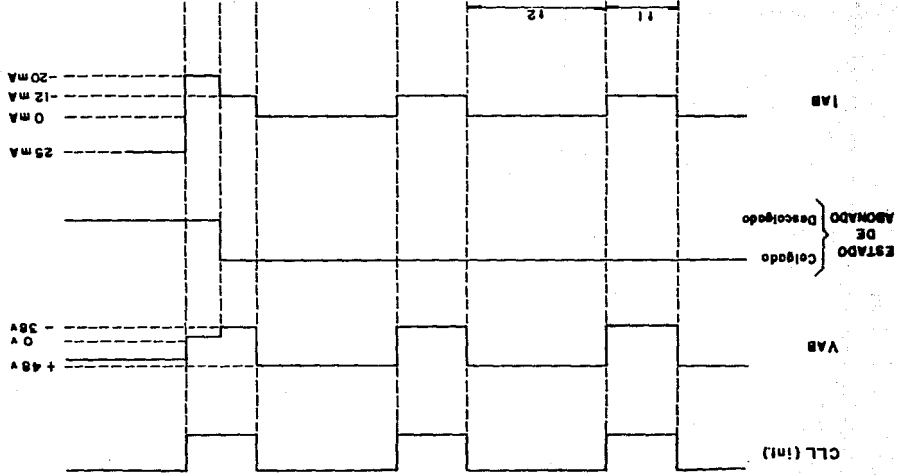


FIG. 3.10

La fig. 3.10 es un Diagrama de tiempos del envío de corriente de llamada. En dicho diagrama:

CLL (int) = Información interna del Módulo de Abonado escrita previamente por el Microprocesador.

$V_{AB}$  = Tensión presente en el par telefónico atendido por el Circuito de Abonado. Se toma la convención de considerar la positiva si el hilo A es más positivo que el hilo B.

$I_{AB}$  = Corriente que circula por el par telefónico.

En este diagrama se puede observar una cadencia típica de un estado de corriente de llamada: 1 segundo de sonido y 4 segundos de silencio. En la Central que estamos estudiando, la corriente de llamada se envía invirtiendo la polaridad en la línea. Es importante aclarar aquí que éste es un caso único. En las centrales ahora existentes, el campanario del teléfono se activa aplicando a la línea desde la Central un voltaje alterno de 90 Vrms a 25 Hz.

No es éste el caso de la Central CARMELITA. Por lo tanto, los teléfonos que vayan a conectarse a esta Central deberán tener en vez de un campanario un circuito que emita un sonido al invertirse la polaridad de la línea (Circuito Zumbador). De cualquier forma, para fines de descripción para nosotros serán equivalentes los siguientes términos:

Polaridad invertida de la

línea

=

Envío de corriente de llamada

Polaridad normal de la

línea

=

No existe corriente de llamada

Campanario del

teléfono

=

Circuito Zumbador

Obsérvese que cuando no existe corriente de llamada y el abonado está colgado, prácticamente no consume corriente su teléfono, por lo que para fines prácticos consideramos que la resistencia a D.C. que presenta a la línea es infinita.

Una vez que el abonado descuelga, debe alimentarse a su teléfono con una corriente constante de 25 mA, independientemente de la distancia que lo separe de la Central, dentro de un cierto margen, claro está.

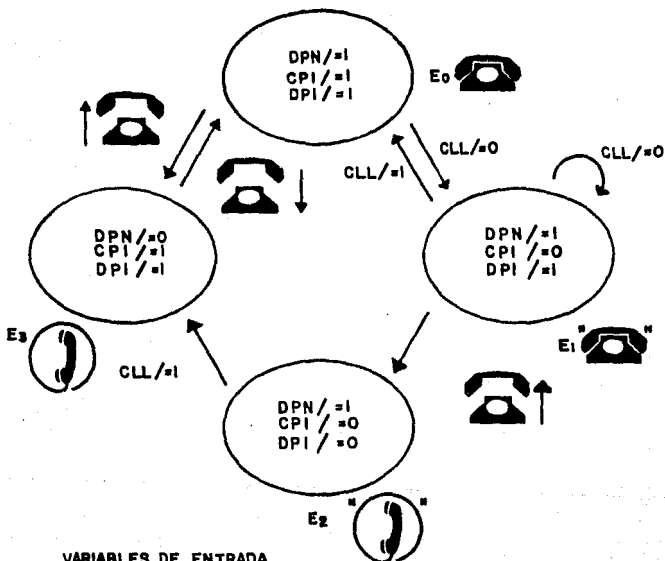
Cuando se envía corriente de llamada y el abonado está colgado, el Circuito Zumbador consume unos 12 mA. Una vez que el abonado descuelga, disminuye la resistencia a D.C. de su teléfono, consumiéndose ahora 20 mA.

Estos cambios en el consumo de corriente los supervisa el Módulo de Abonado para detectar el momento en que descuelga el usuario.

### 3.2.3) Diagramas de Estados

Vamos ahora a hacer referencia a la fig. 3.11. En ella podemos apreciar que el abonado se debe encontrar en uno de los 4 estados ahí especificados. El Circuito de Abonado detecta dicho estado (información DPN/,CPI/,DPI/ presente en el interior de los círculos) y puede así informar al Microprocesador cuando éste ejecute una lectura sobre él.

## DIAGRAMA DE ESTADOS



### VARIABLES DE ENTRADA



COLGAR (generadas por abonado)

DESCOLGAR

CLL/=0 ENVIO CORRIENTE DE LLAMADA (generadas por el microprocesador)

CLL/=1 SUSPENSION CORRIENTE DE LLAMADA

### ESTADOS



COLGADO SIN CORRIENTE DE LLAMADA

DESCOLGADO SIN CORRIENTE DE LLAMADA

COLGADO CON CORRIENTE DE LLAMADA

DESCOLGADO CON CORRIENTE DE LLAMADA

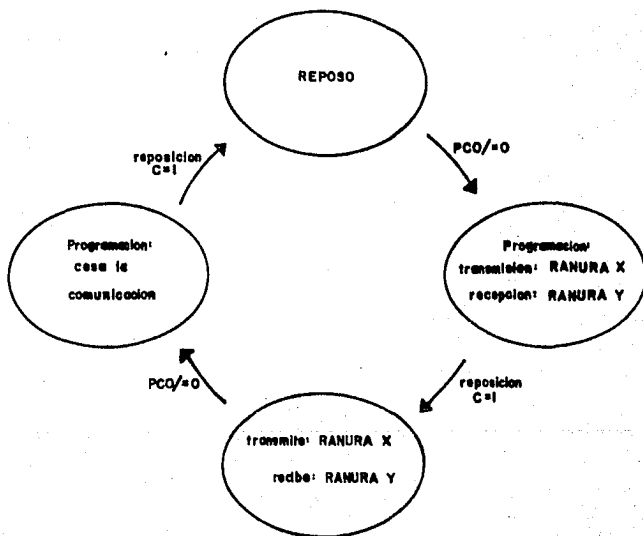
La fig. 3.12 nos enseña los diferentes estados y transiciones del Circuito de Abonado con respecto a la Red de Conmutación.

El Circuito presenta un estado de alta impedancia a la Red, y por lo tanto aislado de la misma, cuando se halla en reposo. Al escribir  $PCO/=0$  se puede programar, ya sea para asignarle ranuras para comunicación o bien ordenarle quedar en reposo. La señal Reposición C (0 RESET C) inhibe la futura programación del Circuito.



## DIAGRAMA DE ESTADOS

### RED DE CONMUTACION:



### VARIABLES DE ENTRADA:

PCO =0 habilita la programacion del abonado

REPOSICION C=1 deshabilita la programacion del abonado



### 3.3) Diagrama de bloques de la tarjeta

En la fig. 3.13 se puede apreciar el Diagrama de bloques del Circuito de Abonado.

Brevemente, explicaremos algunos:

a) Circuito de Protección: Tiene la función de proteger a la Interfaz de Abonado contra el exceso de voltaje o corriente que pudiera absorber la línea telefónica.

b) Circuito Híbrido: Es la parte de la Interfaz que se encarga de la función híbrida. En este caso se separan las señales que - viajan mezcladas en ambos sentidos en el par telefónico y se envían por 2 trayectorias separadas:

1) Transmisión desde el abonado hacia la Red de Conmutación.  
( Esta señal se codificará después para enviarla por el Bus PCM: Tx ).

2) Recepción desde la Red de Conmutación hacia el abonado, -  
( Esta señal proviene ya decodificada del Bus PCM: Rx ).

c) Circuito de Alimentación y Corriente de Llamada: Este circuito cumple con dos tipos funciones: Unas hacia el bucle de abonado y otras hacia el resto de la Interfaz.

1) Funciones hacia el abonado:

Las funciones hacia el bucle de abonado son principalmente las siguientes:

Supervisar el estado del abonado (Colgado o descolgado).

Detectar el estado de línea telefónica abierta.

Emitir la señal de corriente de llamada.

Proporcionar la corriente de polarización al aparato de abonado.

Las funciones hacia el resto de la interfaz son:

Detectar la señal CLL para activar envío de corriente de llamada

Proporcionar las señales:

DPN 1; para indicar estado de abonado sin envío de corriente de llamada.

EPI 1; para indicar el estado de abonado con envío de corriente de llamada.

Recuérdese que estas señales proporcionadas por el Circuito de Alimentación y Corriente de Llamada están aplicadas a otra parte del Circuito o Interfaz de abonado y no salen al exterior de la tarjeta.

d) Detector de estado de abonado sin corriente de llamada

La función de este circuito es identificar el estado de abonado descolgado sin corriente de llamada, a partir de la señal DPN 1 proveniente del Circuito de Alimentación y Corriente de Llamada y convirtiéndola a un nivel TTL (0 ó 5 Volts).

Esta nueva señal se llama DPN.

La señal DPN llegará hasta el exterior de la tarjeta, convirtiéndose en DPN/, - solo si se ha seleccionado el Grupo y Elemento adecuados, simultáneamente con la orden de lectura. De no ser así, un Acoplador de 3 estados la aísla de las terminales exteriores de la tarjeta.

Otra de las funciones de este circuito es la de controlar al circuito economizador de corriente.

e) Detector de Estado de Abonado con Corriente de Llamada

Básicamente la función de este circuito es la de identificar la señal EPI 1 proveniente del Circuito de Alimentación y Corriente de Llamada y a partir de ella reconocer uno de dos estados:

Abonado colgado con corriente de llamada

Abonado descolgado con corriente de llamada

De la misma manera que en el circuito anterior, las señales CPI y DPI están convertidas a niveles TTL y están aisladas del Bus de la Central por un Acoplador de 3 Estados. En el caso en que se seleccionara el Grupo y Elemento adecuados simultáneamente con la orden de Lectura se habilitaría el Acoplador de 3 Estados permitiendo a las señales que llegaran a las terminales externas.

f) Economizador de corriente para +12 V y -12 V.

Dado que el circuito híbrido está constituido por componentes activos, es necesario proporcionarles una alimentación, siendo en este caso de  $\pm 12 V_{DC}$ . Sin embargo, este circuito híbrido solo está en funcionamiento cuando el abonado está efectuando una conversación, lo cual hace solamente una parte de tiempo en el día. Para evitar el consumo de energía innecesario durante el resto del día, se cuenta con el Circuito Economizador de corriente para +12 V y -12 V, el cual es controlado por la señal DPN.

g) Multivibradores Biestables

Los Multivibradores cumplen con la función de retener en sus salidas las órdenes que hayan sido previamente escritas por el Microprocesador.

La Interfaz de abonado recibe solamente 2 Órdenes del Microprocesador:

- CLL/, para envío de corriente de llamada y
- PCO/, para habilitar la programación del CODEC.

h) Acoplador de Tres Estados

Unicamente sirve para evitar cargar el infoducto al cual se van a enviar las señales: DPN, CPI y DPI cuando la salida de ellas no sea requerida por el Microprocesador.

Al pasar al estado de baja impedancia, el Acoplador de 3 Estados permite que la información interna de la tarjeta,- del estado en que se encuentra el abonado,- llegue a las terminales externas y pueda ser accesada por el Microprocesador.

i) Compuertas Lógicas NOR

Estas compuertas permiten que se realicen las operaciones de lectura y escritura sobre el circuito de abonado cuando el Microprocesador así lo solicita.

En la Central existen muchos módulos distintos que son atendidos por el Microprocesador y es la misión de las compuertas lógicas NOR el detectar cuándo ha sido direccionado dicho circuito en particular.

Además, controlan al Acoplador de 3 Estados y a los Multi-vibradores cuando el Microprocesador ejecute una lectura o escritura, respectivamente.

j) CODEC

Este Circuito Integrado está encargado de la conversión Analógica/Digital y Digital/Analógica necesarias para la interfaz

del Circuito de Abonado con la Red de Conmutación temporal.

Analizando un poco más a detalle su funcionamiento, observamos:

1) El Circuito Integrado codifica la señal analógica de entrada proveniente del Filtro (Tx) y la convierte a una palabra de 8 bits. Esta codificación se realiza una vez por trama y por lo tanto su frecuencia está determinada por el pulso de inicio de trama aplicado al Codec a través de las terminales 21,22 de la tarjeta (Señal FSx, FSr).

2) Internamente, el Codec lleva a cada instante un conteo de la ranura de tiempo por la que está pasando. Dicho conteo se hace con la ayuda de una señal de sincronía (terminales 17, 18 de la tarjeta) que valida la aparición de cada bit en el Canal PCM. Una vez que le toca su turno de transmitir, habilita al Buffer que aislaba la señal Tx de la Red y envía una palabra de 8 bits al Canal PCM.

3) Cuando le toca en turno recibir, admite una palabra de 8 bits proveniente del Canal PCM en su terminal Rx. Decodifica esa palabra, la convierte a un valor analógico que presenta en su salida Rx aplicada al Filtro. Este valor se mantiene constante hasta que se recibe una nueva palabra de 8 bits.

#### k) Filtro

Para entender mejor la misión que desempeña el Filtro, repasaremos algunos conceptos:

Cualquier señal que esté definida para cualquier tiempo "t" tiene componentes de energía para diferentes frecuencias.

En la fig. 3.14 (a) se muestra una función  $f(t)$  y en la fig. 3.14 (b) aparece su transformada de Fourier. La Transformada de Fourier nos enseña la densidad de energía espectral de la señal.

Obsérvese que en este caso la señal en estudio no tiene energía para frecuencias mayores a  $W_m$ .

El Teorema de Muestreo nos indica que para reconstruir totalmente esta señal a partir de sus muestras, éstas han de ser tomadas al menos con una frecuencia de  $2W_m$ . En la fig. 3.15 (a) se observa como se toman muestras a determinados intervalos de tiempo, menores a  $1/2 W_m$ .

En la fig. 3.15 (b) se enseña la Transformada de Fourier de la fig. 3.15 (a). Nótese cómo se "multiplica" la forma individual de la señal sin muestrear, cada  $W_0$ . Puesto que  $W_0 > 2W_m$ , la señal original puede ser reconstruida pasando la señal muestreada por un filtro pasabajos ideal de ancho de banda  $B$  tal que  $W_m < B < W_0 - W_m$ .  $W_0$  es la frecuencia de muestreo.

Sin embargo, las señales reales no son de banda estrictamente limitada. Pero un espectro de mensaje como el de la fig. 3.16 (a) se considera que, prácticamente, es de banda limitada si el contenido de frecuencia arriba de  $W$  es pequeño y para nuestros fines no es importante en lo que a conducción de información se refiere. Cuando se muestrea tal mensaje, habrá un traslape inevitable de componentes espectrales, fig. 3.16 (b). En la reconstrucción, las frecuencias originalmente fuera de la banda nominal del mensaje ( $f > w$ ) aparecerán a la salida del filtro



en la forma de frecuencias mucho menores. Así, por ejemplo,  $f_1 > W$  resulta ser  $f_s - f_1 < W$ , como se indica en la figura. \*

Este fenómeno de traslación de frecuencia hacia valores menores ocurre siempre que se submuestra una componente de frecuencia, o sea que,  $f_s < 2f$ , y se le da el nombre descriptivo de "interferencia de colas espectrales (aliasing)". Lo grave de este problema es que las frecuencias interferentes caen dentro de la banda del mensaje.

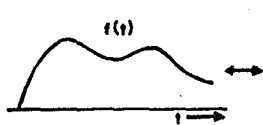
Existen 2 formas de remediar este problema: Filtrar el mensaje tanto como sea posible antes del muestreo y muestrear a una velocidad mucho mayor que la nominal de Nyquist.

La función del Filtro, en cuanto le atañe a transmisión hacia el Bus PCM, -es filtrar el mensaje que se va a muestrear para eliminar la interferencia de colas espectrales. Si por ejemplo, se aplica a la entrada del Filtro una señal como la que aparece en la fig. 3.16 (a), éste atenuará fuertemente (digamos unos 30 db) aquellas componentes de frecuencia superiores a  $W$ . Esta nueva señal ya filtrada se aplicará después al Codec para su muestreo y codificación.

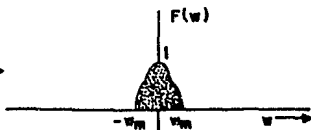
El espectro de voz promedio se extiende más allá de los 10 kHz, aunque mucha de la energía se concentra en el intervalo de 100 a 600 Hz y un ancho de banda de 3.4 kHz es suficiente para inteligibilidad. Cuando una onda se muestrea a  $f_s = 8$  kHz, lo normal para sistemas telefónicos, las componentes interferentes de colas espectrales están en forma típica 30 dB debajo de la señal deseada y prácticamente no se perciben.

Ahora bien, en cuanto a recepción, el Filtro pasa la señal proveniente del Codec por un Filtro Pasabajos que construye la señal original que se muestreó en el otro módulo para ser transmitido por el Bus PCM.

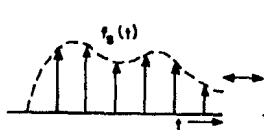
\*  $f_s$  = frecuencia de muestreo



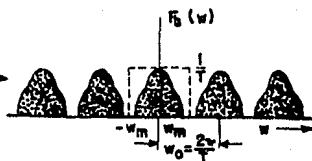
3.14 (a)



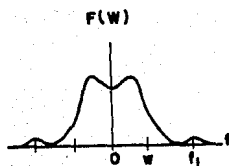
3.14 (b)



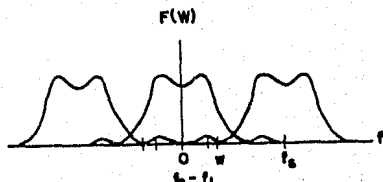
3.15 (a)



3.15 (b)



3.16 (a)



3.16 (b)

#### IV) HARDWARE DEL SISTEMA

##### 4.1) Descripción de la Prueba.

Ya hemos explicado en el Capítulo anterior las funciones del circuito de abonado. Para probarlo será necesario someterlo a sus condiciones de operación y comprobar si las ejecuta satisfactoriamente.

Estas son principalmente:

- Verificar que el circuito proporcione alimentación eléctrica a la línea y que envíe corriente de llamada cuando a sí se le ordene;

- Simularle al circuito las diferentes condiciones en que se puede encontrar el usuario: Colgado o Descolgado. En cada una de ellas se ejecutará una lectura sobre la tarjeta para observar si la información a su salida es correcta. Esta operación se hará con y sin envío de corriente de llamada para probarla en todas las condiciones posibles.

- Además, se comprobará que la etapa de conmutación de la tarjeta está en buen estado proporcionando a los Codecs de la tarjeta señales de sincronía, programándoles en transmisión en una ranura determinada y enviando un tono por la línea telefónica atendida por la tarjeta. Se verificará que dicho tono se transmita al Canal PCM.

Posteriormente se programará al Codec una ranura para recepción y se le enviará una señal codificada en dicha ranura

por el Canal PCM. Se comprobará que la señal aparezca en la lí  
nea telefónica atendida por la tarjeta.

En resumen, para la prueba que vamos a hacer sobre la tar  
jeta de abonado se precisa que nuestro equipo sea capaz de lo  
siguiente:

a) Simular en la línea telefónica estado de colgado y des  
colgado;

b) Revisar la línea telefónica atendida por el Circuito  
de Abonado para verificar que proporcione alimentación eléc  
trica y que envíe satisfactoriamente la corriente de llamada;

c) Ejecutar operaciones de escritura sobre la tarjeta, o  
denándole el envío o la suspensión de la Corriente de Llamada  
y habilitando o deshabilitando la Programación del Codec;

d) Que ejecute operaciones de lectura sobre la tarjeta,  
para supervisar que el estado lógico de sus salidas sea el co-  
rrecto a todo momento;

e) Programar los Codecs en transmisión o recepción en  
cualquier ranura de tiempo o bien ordenarles que queden en re-  
poso;

f) Enviar un tono por la línea telefónica atendida por la  
tarjeta (señal analógica) o por el canal PCM (señal digital);

g) Detectar el tono en la línea telefónica atendida por  
la tarjeta o en el Canal PCM;

h) Ser lo suficientemente "inteligente" para decidir si  
la tarjeta está defectuosa. En este caso, debe detectar qué  
parte de la tarjeta es la que no está funcionando bien;

i) Interactuar con un operario, el que debe avisar de los

resultados obtenidos en la prueba y recibir de él diferentes órdenes, como por ejemplo: Comienza la prueba, suspende la prueba, prueba el abonado 8, 2 o 3 de la tarjeta; así como información acerca del Grupo y Elemento que caracterizan a cada abonado.

#### 4.2) Descripción del Hardware.

Para cumplir con todas las funciones mencionadas anteriormente, nuestro Probador necesitará estar dividido en diferentes secciones:

##### Microprocesador 8035.

Ya veremos en el próximo Capítulo que para el Planteamiento de la Rutina de Prueba utilizaremos como modelo un Automata, esto es, un sistema compuesto de diversos estados por los que va pasando el Probador en función de las excitaciones de entrada que se le presentan. Existen varias formas de implementar esto físicamente.

Una de ellas puede ser en base a lógica cableada. El estado en que se encuentra el Probador estaría almacenado en unas variables de estado interno, - salidas de compuertas lógicas, - y el nuevo estado al que evolucionaría el Automata sería a función tanto de las variables de Entrada como de las variables de Estado interno.

Un sistema con lógica cableada presenta muchas desventajas, las principales son:

a) Poca Versatilidad para el Cambio. Una decisión de ha-

cer un cambio en la Rutina de Prueba repercutirá en modificar el Hardware del Sistema. Imagínese la gravedad del cambio si ya existen tarjetas de circuito impreso fabricadas.

b) Crecimiento desproporcionado del Hardware. Esto es obvio, puesto que toda la lógica es cableada, se precisa de muchos componentes y más espacio en tarjeta de circuito impreso.

Es por ello que nos inclinamos en implementar el Automata en programa. Esto es, nuestro sistema está desarrollado al rededor de un microprocesador, el 8035 de Intel.

Si a un futuro queremos efectuar un cambio en la Rutina de Prueba, bastará cambiar lo necesario del programa sin alterar el Hardware.

Tambien integramos mucho el diseño, al disminuir el número de componentes utilizados.

Vamos a describir nuestro microprocesador:

#### Memoria RAM.

El microprocesador tiene integrada una memoria RAM de 64 bytes. 8 localidades (0-7) son direccionables directamente por ciertas instrucciones y por lo tanto son ideales para almacenar datos a los que se hace referencia frecuentemente en el Programa.

Existen otras 8 localidades (24-31) que se pueden convertir en registros de trabajo tomando el lugar de los registros (0-7) con una orden de selección de Banco de Registros. De esta forma se puede utilizar esta facilidad para almacenar datos en la atención a subrutinas trabajando en el Programa Principal con un Banco de Registros, y en la Subrutina con otro Banco.

Todas las demás localidades de Memoria se pueden acceder indirectamente por medio de apuntadores almacenados ya sea en el Registro "0" o en el Registro "1".

16 localidades (8-23) están destinadas a almacenar datos necesarios para regresar de una subrutina. Esto es, al llamar a una subrutina, ya sea con un CALL o con una Interrupción, se almacenan el Contador del Programa y la mitad de la Palabra de Estados del Procesador. Para ello se precisan 2 bytes. El apuntador del stack se incrementa para apuntar a la siguiente localidad, y así guardar la información requerida para el regreso al final de la subrutina, y a su vez permitir almacenar nuevos datos en caso de llamarse a otra subrutina.

Puesto que se destinan 16 localidades para este fin y 2 se necesitan para cada CALL, se permite un máximo de 8 niveles de subrutinas.

#### Puertos 1 y 2.

El microprocesador tiene dos puertos de entrada o salida. (Puertos 1 y 2). La información que se escribe en dichos puertos queda almacenada ahí hasta que se escriba nueva información en ellos. En cambio, al emplearse como puertos de entrada, la información presente es tomada como válida sólo durante la instrucción de lectura.

La configuración de las líneas es tal, que para que las líneas del puerto funcionen como entradas se debe escribir en ellas previamente en una lógica. Esta configuración permite que algunas líneas de un puerto puedan emplearse como entradas, o--

tras como salidas, etc.

### BUS.

8 líneas del circuito integrado están dedicadas al BUS. En general, este puerto se emplea para operaciones de escritura y de lectura con la instrucción MOVX.

Durante una operación de escritura, la información en el BUS es válida en el flanco de subida de  $\overline{WR}$ .

Durante una operación de lectura, se toma la información presente en el BUS en el momento en que sube la señal  $\overline{RD}$ .

Para cumplir su función, el Microprocesador lee las instrucciones del programa almacenado generalmente en una memoria EPROM externa.

Para ello, es preciso que inicialmente dirija la localización de la instrucción que va a leer.

Los pasos que se siguen para una búsqueda de una instrucción son los siguientes:

1) El contenido del Contador de Programa (12 bits) aparece en el BUS y en la parte baja del puerto 2.

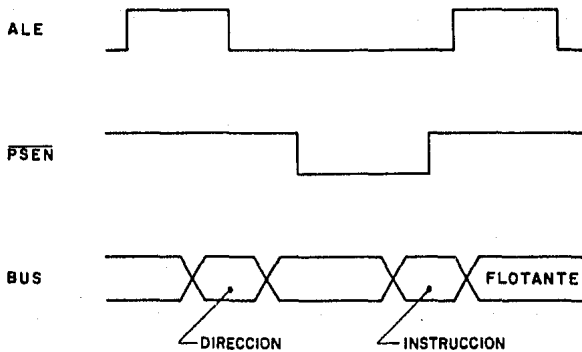
2) La señal ALE (Address Latch Enable) indica el momento en que es válida la dirección. El flanco de bajada de ALE se emplea para almacenar la dirección externamente.

3) La señal  $\overline{PSEN}$  (Program Store Enable) indica que está en curso una búsqueda de instrucción externa y se debe emplear para habilitar la memoria que contiene el programa.

4) El BUS vuelve a su modo flotante de entrada para estar listo a recibir la instrucción.

(Ver Figura 4.1.)





**BUSQUEDA DE UNA INSTRUCCION A MEMORIA EXTERNA**

FIG. 4.1

### CONTADOR.

Otra facilidad que presenta el 8035 es la presencia de un contador que auxilia al usuario a contar exentos externos y generar retardos de tiempo muy precisos; sin representar una carga para el Procesador. En ambos casos la operación del contador es la misma, siendo la única diferencia la entrada que tenga.

Si al usuario le interesa emplearlo para generar retardos, existen instrucciones que permiten que el contador sea incrementado por un reloj interno, de 12.5 KHz.

Para la otra aplicación, de contar exentos externos, otras instrucciones permiten que el contador sea incrementado en el flanco de bajada de la señal que se aplique en T1.

Una vez que el contador, - que consiste en una palabra de 8 bits, - pase su cuenta máxima, se altera el estado de un biestable interno y se hace una solicitud de interrupción. Si dicha interrupción está habilitada, entonces al llegar a su tope el contador se hará una llamada a la subrutina almacena da en la localidad 7 de memoria.

Por otra parte, el estado del biestable interno puede ser probado por ciertas instrucciones.

### T0, T1, $\overline{\text{INT}}$ .

Tres terminales del procesador, - T0, T1 e  $\overline{\text{INT}}$ , - pueden ser probadas durante la ejecución del programa. Existen instrucciones especiales por medio de las cuales se supervisa el

## REPOSICIÓN

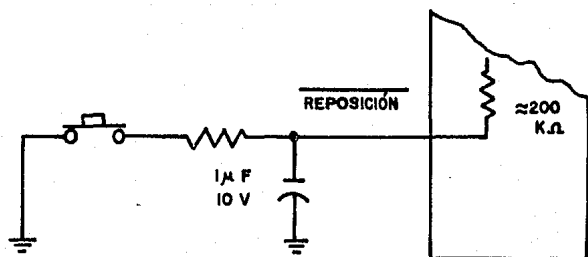


FIG. 4.2

estado lógico que aparece en ellas y en base a esta información el procesador continúa con la siguiente instrucción o salta a otra parte del programa.

Además, si las interrupciones están habilitadas, la presencia de un cero lógico en  $\overline{\text{INT}}$  causa un "salto a la subrutina" presente en la localidad 3 de Memoria.. El Contador de Programa y la Palabra de Estados se guardan automáticamente en el Stack para poder volver al Programa Principal una vez que se atiende a la interrupción.

#### REPOSICION.

Al pulsar este botón, el Contador de Programa se iguala a cero y se inicia la ejecución del Programa desde el principio. (Ver Figura 4.2.)

#### EXPANSOR DE PUERTO 8243.

En un sistema con Microprocesador 8035, una forma muy eficaz de lograr una expansión de entrada/salida es por medio del uso de expansores 8243.

El expansor 8243 se une al Microprocesador con sólo 5 líneas, 4 de las cuales se emplean para intercambiar información con él (Parte baja del puerto 2) y la restante proporciona la temporización adecuada (PROG) para el intercambio.

Es por medio de estos Expansores que el Microprocesador se comunica con:

a) Receptor de Tono (Entrada, 1 bit)

Un uno lógico en esta entrada indica que se detecta una multifrecuencia válida en la línea a la que está conectado el Receptor DTMF. Así se indica que la Etapa de Conmutación y de Adaptación a la línea telefónica del circuito en estudio trabaja satisfactoriamente, ya sea en transmisión o recepción, depende de la etapa de la prueba.

Un cero lógico quiere decir que el Receptor de tono no de tecta una multifrecuencia válida y, por lo tanto, la Etapa de Conmutación y de Adaptación a la línea del circuito estudiado está defectuosa.

b) Bus Red (Entrada/Salida, 8 bits)

Estas líneas se emplean para:

- Llevar información al Circuito de Abonado, durante una operación de escritura.

- Recoger información del Circuito de Abonado, durante una operación de lectura.

Se conectan directamente a las terminales 63 a 70 de la tarjeta.

c) LEC/, ESC/ (Salidas, 2 bits)

Líneas que son las señales de escritura y de lectura de la tarjeta, de la información presente en las 8 líneas denominadas Bus Red. Se conectan a las terminales 15 y 16 de la tarjeta.

d) 8279 Habilita (Salidas, 3 bits)

8279 C/D

8279 Reposición

La primera línea , - 8279 Habilita , - habilita o deshabilita el 8279 para comunicarse con el Microprocesador.

La segunda línea , - 8279 C/D , - avisa al 8279 si la señal que se le va a leer o escribir es una orden , - comando , - o un dato.

La tercera línea , - 8279 Reposición , - da la orden de reposición al 8279.

e) Selección de Grupo (Salida, 4 bits)

Selección de Elemento (Salida, 4 bits)

Con estas líneas se direcciona al Circuito de Abonado que será examinado.

f) Colgado/Descolgado (Salida, 1 bit)

Para la prueba existe un circuito que simula los estados de colgado y descolgado que puede tener el teléfono.

Dicho circuito es controlado por esta línea.

g) Programador Codec Habilita.

El flanco de subida de esta línea habilita una operación de escritura al Circuito Programador de Codecs.

h) "Crosspoint" Habilita.

Línea que habilita el "Crosspoint" para ejecutar operaciones de escritura sobre él.





## TECLADO.

El Teclado es el medio por el cual el Operador escribe instrucciones y datos al equipo.

Consta de las siguientes partes:

- 1) Teclado Hexadecimal
- 2) Botón: Limpia Visualizador
- 3) Botón: Válida Información

- 1) Teclado Hexadecimal.

Consta de 16 teclas. Por medio de este teclado el usuario informará al equipo del Grupo y Elemento que caracterizan a cada circuito de abonado de la tarjeta que será sometida a prueba. Además le informará del Grupo y Elemento que caracterizan a un circuito de abonado que conste que está en buen estado y que ayudará en la prueba.

Después de haber proporcionado esta información al equipo, el usuario tecleará el número de circuito que desea probar, tecleando un 1, 2 o 3.

Una vez iniciada la prueba, si se encuentra alguna falla la rutina se suspende momentáneamente y se informa al Operario sobre el desperfecto hallado. El Operario tendrá la opción de continuar la prueba (tecleando un 1) o suspenderla (tecleando un 9)

2) Botón: Limpia Visualizador.

La información que es teclada por el usuario y que es reconocida por el Microprocesador aparece en el Visualizador, para comodidad de él. En el momento en que él cometa algún error al teclear, puede limpiar el Visualizador pulsando el Botón: Limpia Visualizador.

3) Botón: Válida Información.

Una vez que el Operario está satisfecho con la información que aparece en el Visualizador, puede alimentarlo al equipo pulsando el Botón: Válida Información. En ese momento, el Microprocesador ejecutará una operación de lectura sobre el Visualizador para enterarse de la orden o dato que escribió el Operario.

VISUALIZADOR.

Es éste el medio de salida que emplea el Microprocesador para comunicarse con el usuario.

Cumple con los siguientes fines:

1) Con los primeros 4 dígitos el Microprocesador informa al Operario de las teclas que ha reconocido que han sido pulsadas. De esta forma el operario puede fácilmente comprobar si la entrada que ha hecho ha sido interpretada correctamente por el Sistema.

2) Los últimos 4 dígitos se utilizan para dar avisos al usuario como:

- Pedirle datos referentes al Grupo y Elemento que caracterizan a cada circuito de la tarjeta que será sometido a prueba; así como del Grupo y Elemento que identifican al circuito en buen estado

- Preguntar qué circuito será sometido a prueba

- Informarlo de los errores encontrados durante la ejecución de la Rutina.

### 8279

Para el manejo de Teclado y Visualizador se empleó el 8279. El 8279 es un circuito integrado especialmente diseñado para este propósito. Su empleo presenta las siguientes ventajas:

#### 1) Simplificación del Software.

El 8279 se encarga de supervisar el teclado y escribir en el Visualizador los caracteres que deben aparecer. Con ello libera al Microprocesador de esta tarea.

Cuando el 8279 valida la pulsación de una tecla, sube una señal que va conectada al Microprocesador (terminal T1) y almacena el carácter tecleado a una memoria FIFO. De esta suerte, una vez que el Microprocesador revisa la señal T1 y detecta que la FIFO no está vacía, ejecuta una operación de lectura sobre el 8279 para informarse de la tecla que fue pulsada.

Para hacer aparecer un número en el Visualizador, al Microprocesador le basta escribir una sola vez el caracter adecuado. El 8279 tiene una Memoria RAM a la que alternativamente consulta para escribir sobre los dígitos del Visualizador.

## 2) Versatilidad en el Diseño.

Es recomendable que todo diseño sea versátil, esto es, que la introducción de una nueva facilidad o de un cambio en general no le represente modificaciones significativas, ya sea en el Hardware o en el Software.

El empleo del 8279 da mucha versatilidad en el diseño, pues permite la ampliación del teclado o del visualizador sin alterar prácticamente el Software del Sistema.

Además, el cambio en el Hardware sería mínimo, limitándose a hacer las conexiones necesarias de las nuevas teclas o dígitos del Visualizador al 8279.

No será necesario ocupar nuevas líneas de entrada o salida de puertos del Microprocesador.

CONTROLADOR DE CODECS.

Este Módulo está encargado de las siguientes funciones:

- 1) Ser capaz de programar al Codec del Circuito de Abogado que está siendo sometido a prueba ya sea una ranura para transmisión, una de recepción, o bien quedar en reposo;
- 2) Proveer las señales de sincronía para el funcionamiento del Bus PCM.

A continuación describimos la serie de eventos que tiene lugar al programarse un Codec:

- 1) Se habilita la programación del Codec de la tarjeta escribiendo la señal PCO/=0 sobre el Circuito
- 2) El Microprocesador selecciona el Módulo Controlador de Codecs levantando la señal PCEN (Programador de Codec Enable; línea de salida de un Expansor de Puerto).
- 3) Posteriormente, el Microprocesador escribe una palabra de 8 bits con una orden MOVX.
- 4) El Módulo Programador de Codecs convierte la palabra de paralelo a serie y alimenta cada bit al Bus PCM, terminal "Dc", simultáneamente con las transiciones negativas de la señal "CLKc", también generada por él.
- 5) Una vez que termine con la programación, el Módulo Controlador de Codecs da un pulso en la terminal "RESET C" para deshabilitar la Programación de Codecs.
- 6) El Microprocesador genera por software un retardo de

tiempo lo suficientemente amplio (250 microsegundos: 2 tramas) para permitir que la próxima programación a dicho Codec sea válida.

Para componer el Circuito Controlador de Codecs se es cogió el mismo circuito que se emplea en la Central Carmelita para este fin. (Figuras 4.4. y 4.5.)

Sin embargo, hicimos unos pequeños cambios para adaptarnos a este diseño, puesto que:

1) Sólo precisamos de una línea de direccionamiento, mientras que el diseño original contempla muchas líneas para este fin;

2) En el Diseño original, el Circuito Controlador de Codecs baja una señal (ACK) para indicar al Microprocesador que ya puede levantar la señal de escritura ( $\overline{WR}$ ).

Como nuestro Microprocesador no maneja la señal ACK, implementamos un pequeño arreglo para levantar la señal de escritura en cuanto llegue el ACK. (Figura 4.6.).

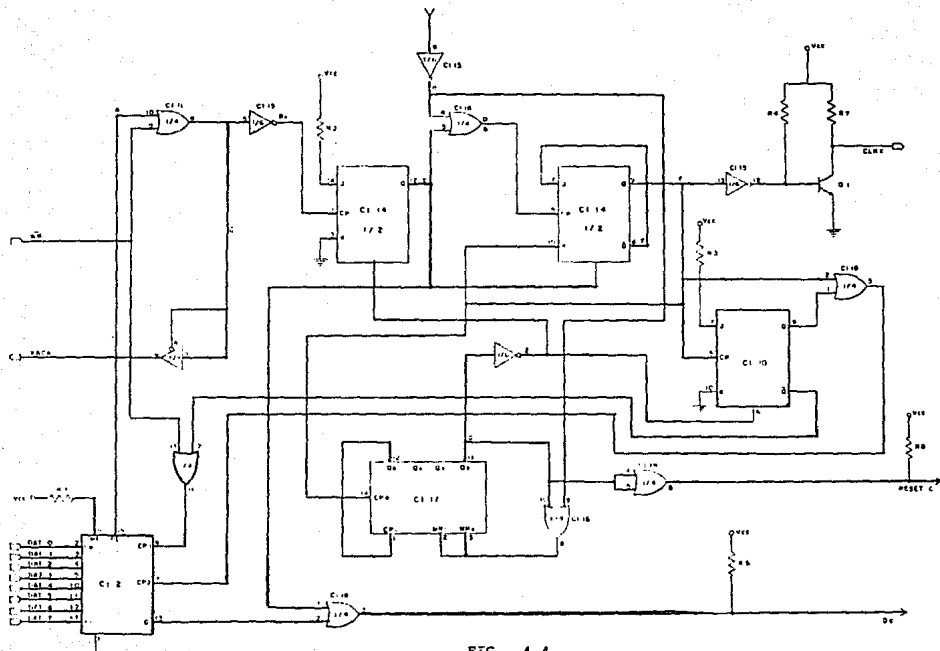


FIG. 4.4

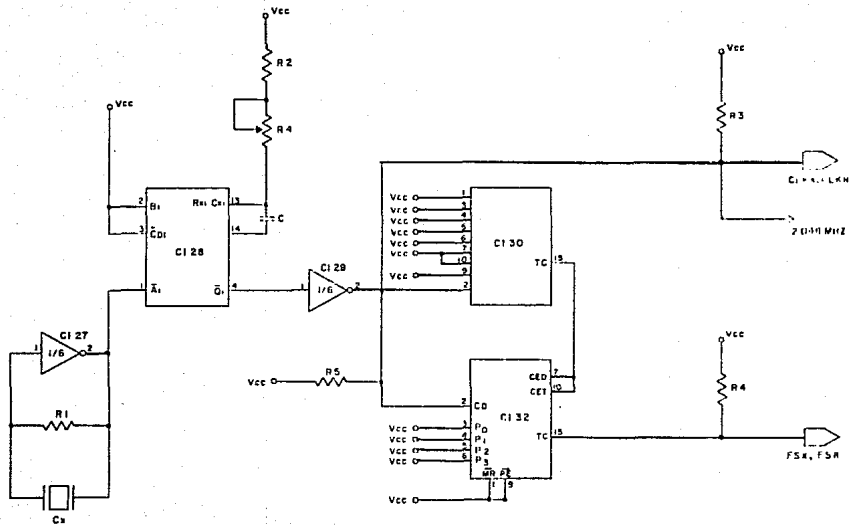


FIG. 4.5



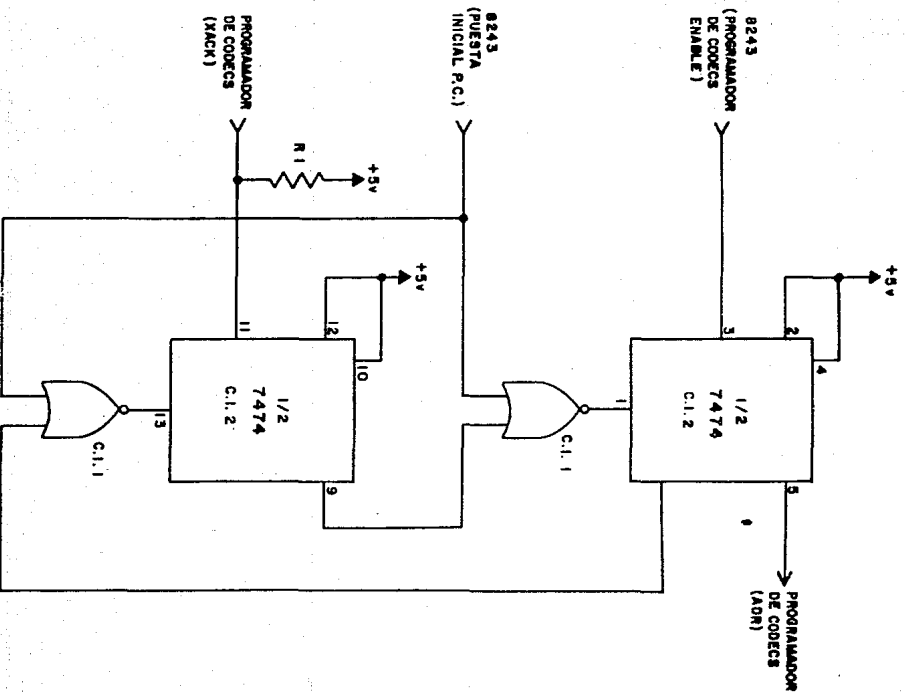


FIG. 4.6

## SIMULADOR.

Es la parte del Sistema encargada de simular las condiciones de colgado o descolgado.

Se empleó el circuito de la figura 4.7.

Según vimos en el Capítulo anterior, en el Diagrama del Envío de Corriente de llamada, para simular las diferentes condiciones en que se puede encontrar el usuario, es necesario presentar diferentes impedancias a la línea telefónica, según los diferentes estados en que se puede encontrar el abonado:

Sin envío de corriente de llamada:

Colgado: No hay circulación de corriente por la línea

Descolgado: Circulan 25 mA por la línea.

Con envío de corriente de llamada:

Colgado: Circulan 12 mA por la línea

Descolgado: Circulan 20 mA por la línea.

Los diodos se encargan de hacer que, dependiendo de la polaridad de la línea, una de las ramas del circuito presente una impedancia prácticamente infinita.

Los transistores se mantienen en corte para simular estado de colgado.

Para simular descolgado, se saturan.

Es muy conveniente el uso de optoacopladores, porque la línea maneja voltajes fuera del rango de los que emplea nuestro Sistema. El optoacoplador permite aislar eléctricamente nuestro Sistema de la línea, acoplándose sólo ópticamente.

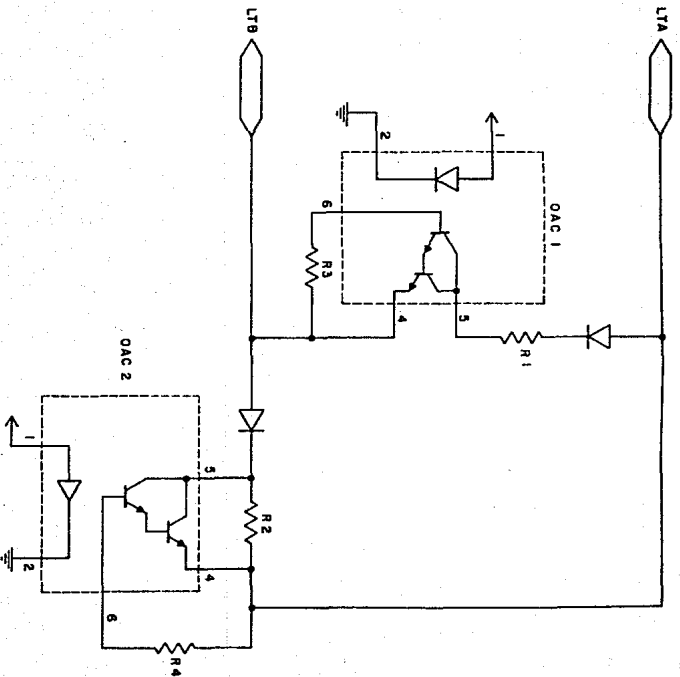
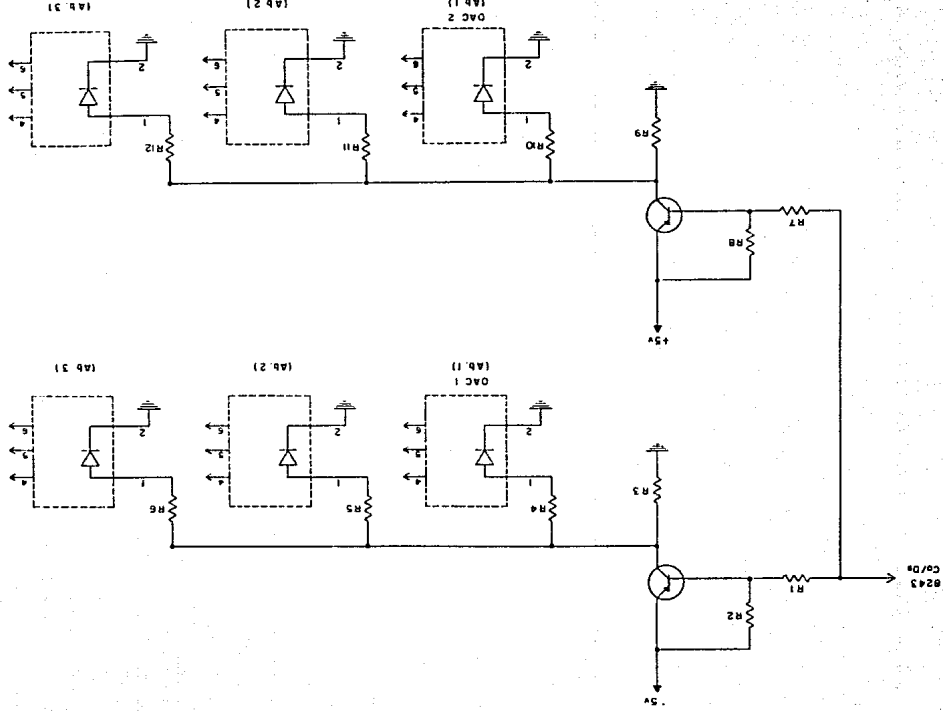


FIG. 4.7 (a)

FIG. 4.7 (b)



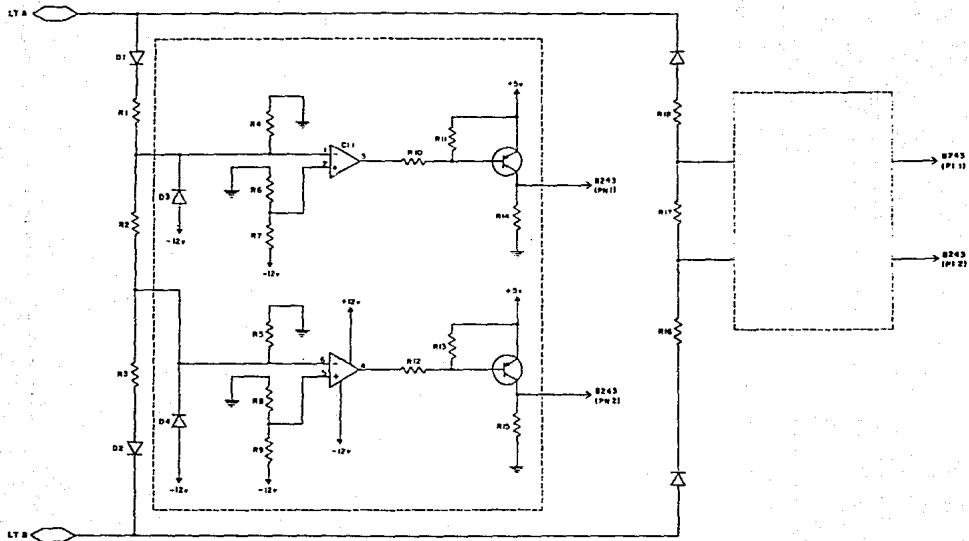


FIG. 4.8

#### COMPROBADOR DE ESCRITURA.

Este circuito se muestra en la figura 4.8.

La misión de este circuito es comprobar que el Circuito de Abonado proporcione alimentación eléctrica a la línea y envíe corriente de llamada correctamente.

Los diodos permiten que una de las ramas del Circuito Comprobador trabaje en estado de ausencia de corriente de llamada mientras la otra presenta una impedancia prácticamente infinita. Al enviarse corriente de llamada, se cambian los papeles de estas ramas.

Las salidas de este circuito se conectan a un Expansor de Puerto 8243.

#### ALMACENADOR DE DIRECCIONES: 74LS373.

La función que desempeña este Circuito integrado es muy sencilla: almacena la dirección que aparece en el Bus en el momento adecuado, ésto es, en el flanco de bajada de ALE. (Figura 4.9.).

#### MEMORIA EPROM 2716.

Este Circuito integrado almacena la memoria del programa, consistente tanto en instrucciones ejecutables por el Mi

croprocesador y en datos. La estructura interna del programa se analizará en el siguiente Capítulo. (Figura 4.9.).

#### BUFFER BIDIRECCIONAL DE 8 BITS: 8286.

Puesto que la carga que se conecta al Bus en el sistema excede su capacidad de manejo de corriente, es preciso ampliarla con la inserción de un Buffer para el Bus de datos.

Además, es deseable que se incluya el Buffer para brindar una mayor seguridad al Sistema. Así es, si se desea que el Bus de datos viaje exterior a la tarjeta, entonces es fácil que exista un corto circuito en alguna línea; si no existiera el Buffer el daño se haría directamente al Microprocesador. (Figura 4.9.)



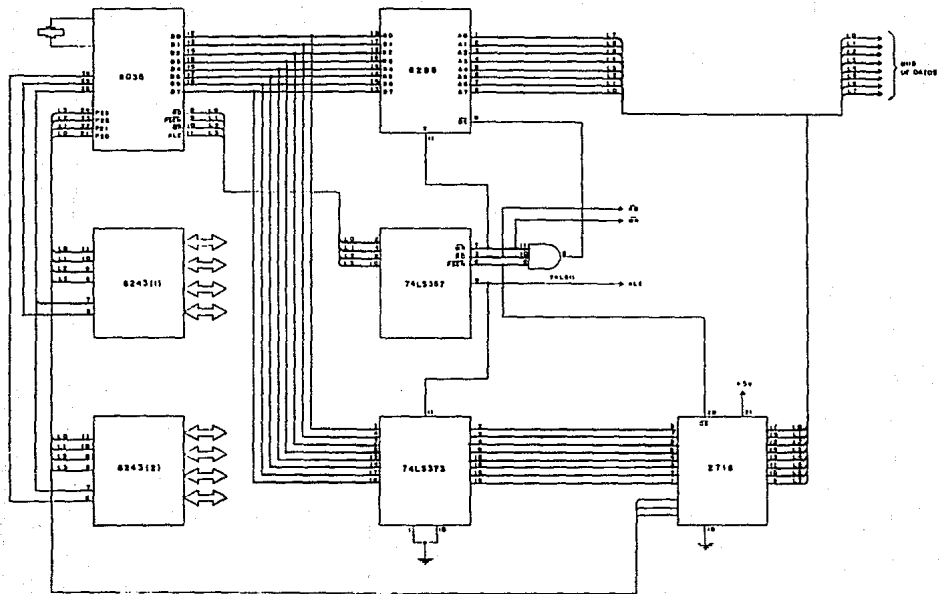


FIG. 4.9

## TRANSMISOR DE TONO, RECEPTOR DE TONO Y "CROSSPOINT".

Ya hemos hablado que una de las misiones que tiene la interfaz de Abonado es la recepción de señales analógicas en el par telefónico, conversión a señal digital y transmisión adecuada al canal PCM.

Asimismo, debe ser capaz de recibir una señal digital del canal PCM, convertirla a señal analógica y aplicarla a la línea telefónica.

Para probar esta función, nos ayudamos de un circuito de Abonado que previamente sabemos que está en buen estado.

Para probar la transmisión adecuada de un tono de la línea telefónica hacia el Bus PCM se establecen las siguientes conexiones:

- 1) Transmisor de tono a la línea telefónica atendida por el circuito en prueba.
- 2) Receptor de tono a la línea telefónica atendida por el circuito en buen estado.

Para probar la recepción correcta de una señal en el Bus PCM y aplicación al par telefónico, se establecen las siguientes conexiones:

- 1) Transmisor de tono a la línea telefónica atendida por el circuito en buen estado.
- 2) Receptor de tono a la línea telefónica atendida por el circuito en prueba.

El "Crosspoint" (RCA CD22101) es un Circuito integrado que provee las rutas necesarias para conectar al transmisor y receptor de tono con el par telefónico (Ya sea del circuito en prueba o del circuito en buen estado).

Es un circuito integrado especialmente diseñado para aplicaciones de telefonía.

Tiene 4 pares de líneas: X1,X1'

X2,X2'

X3,X3'

X4,X4'

de los cuales cualquiera se puede conectar con alguno de los siguientes pares:

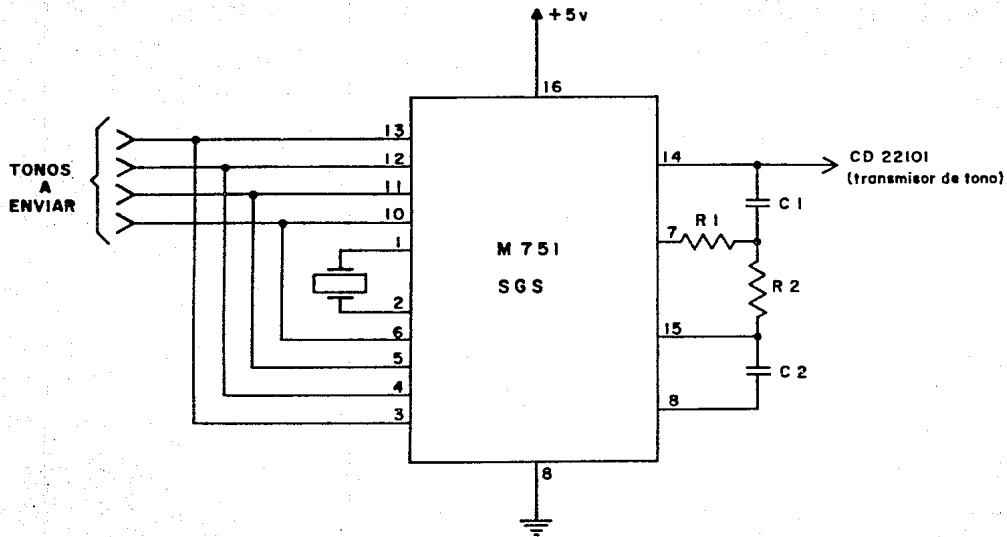
Y1,Y1'

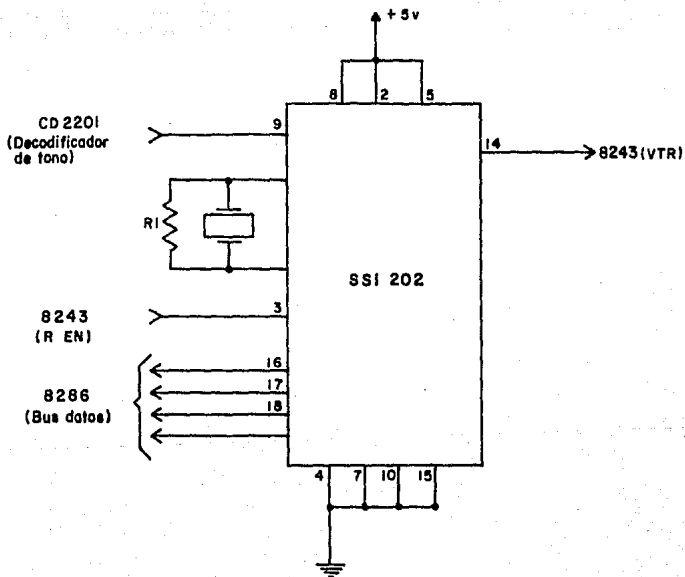
Y2,Y2'

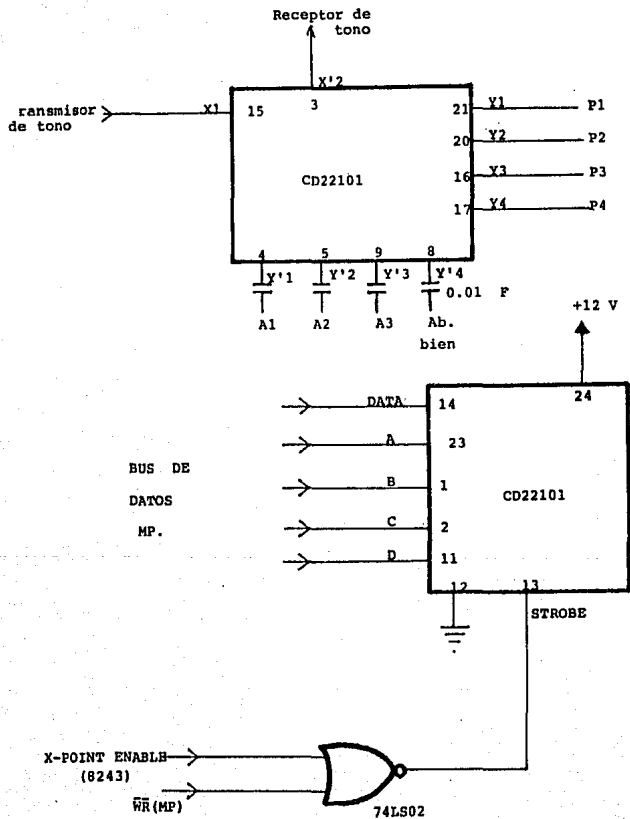
Y3,Y3'

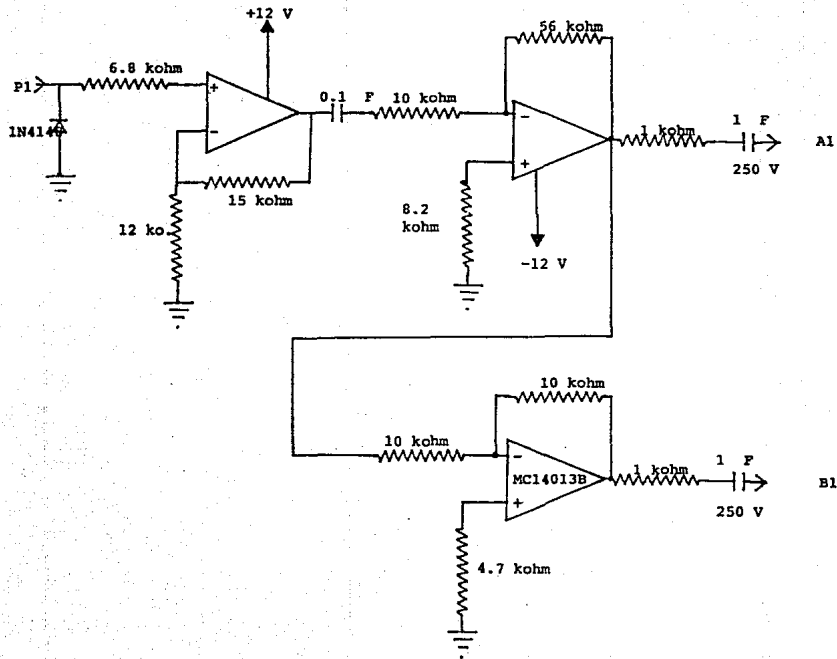
Y4,Y4'

Lo más novedoso de este Circuito integrado es que tiene memoria. Para establecer una conexión basta una operación de escritura y la ruta se mantendrá hasta que una nueva orden la cancele.









## DECODIFICADORES.

La interfaz de Abonado tiene 32 líneas de direccionamiento: 16 para selección de Grupo y 16 para selección de Elemento.

Una forma de direccionar el circuito es conectando éstas líneas directamente a expansores 8243.

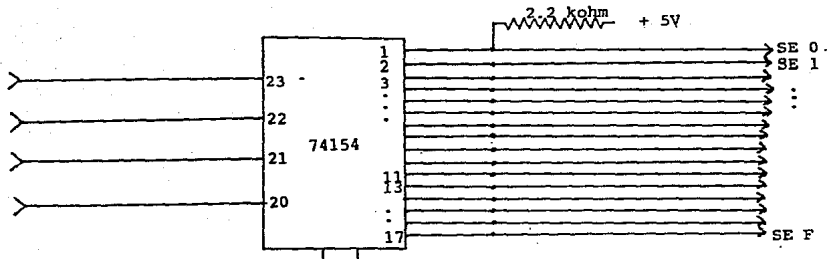
Sin embargo, ésto sería un desperdicio, pues 2 expansores 8243 estarían dedicados solamente a ese fin.

Por ello, empleamos solamente 8 líneas de los expansores, que entregan la dirección del circuito codificada.

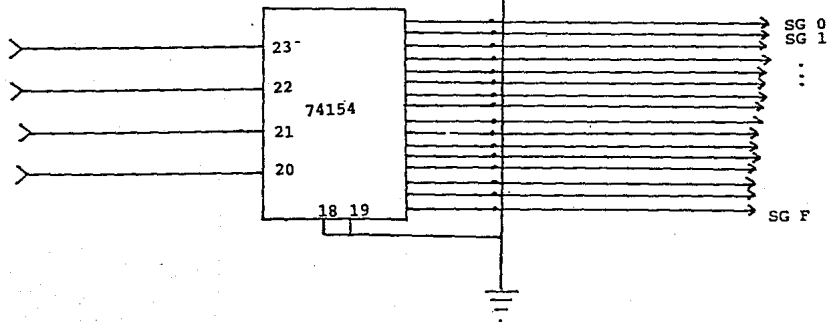
Los decodificadores 74154 tienen la misión de recibir a su entrada la información proveniente de los expansores, decodificarla, y direccionar en sus salidas directamente al circuito en prueba.



8243



8243



## V) SOFTWARE DEL SISTEMA

### 5.1) Introducción

Para el desarrollo del Programa se tomaron en cuenta los siguientes factores:

#### 1) Claridad en el Programa

Es ésta una de las características más importantes que debe tener un programa. Es probable que en el futuro se desee cambiar o mejorarlo. Si el Programa está muy enredado y confuso se va a dificultar mucho la tarea e incluso se prefiera hacer uno totalmente nuevo.

Un programa claro está sujeto a cambio aún por personas que no hayan colaborado inicialmente a su elaboración, puesto que lo entenderán con facilidad.

#### 2) Facilidad de cambio

Un buen programa debe ser versátil, debe poder ser cambiado con facilidad. Ya sea que queramos añadir algo o simplemente modificarlo, es deseable que se pueda hacer rápidamente, sin cambiar la estructura principal del Programa.

Para cumplir con estos objetivos se desarrolló el Programa según el Modelo Matemático de Autómata finito y se hizo lo más modular posible.

### 5.2) Teoría de Autómatas

Ahora damos un breve repaso a la Teoría de Autómatas, para entender mejor el funcionamiento sobre el que se armó el Programa.

Un Automata finito está formado por 6 componentes  $(Q, \&, \$, q_0, F, \varepsilon)$

donde:

a) "Q" representa un conjunto finito de estados:  $q_0, q_1, q_2, \dots, q_n$ .

b) " $q_0$ ", -que es parte del conjunto "Q", -es el estado inicial,

c) " $F$ "  $\subseteq$  "Q" es el conjunto de estados finales.

d) " $\&$ " es un alfabeto finito de entrada: Colección de símbolos que son variables de entrada o de excitación para el Automata (a, b, c, d, ...)

e) " $\$$ " es una función de transición que asocia a cada par posible formado por cada estado y cada símbolo de entrada, otro estado. Supongamos que "x" pueda ser un símbolo de entrada y "q" cualquier estado contenido en Q. Luego,  $\$(q, x)$  corresponde a un estado para cada "q" y cada símbolo de entrada "x".

f) " $\varepsilon$ " es la salida del automata. Tenemos dos casos:

Automata de Moore: La salida del automata depende solamente del estado en que éste se encuentre. " $\varepsilon$ " se especifica para:  $\varepsilon(q_0), \varepsilon(q_1), \varepsilon(q_2), \dots, \varepsilon(q_n)$ .

Automata de Mealy: La salida del Automata no solamente depende del estado, sino también de la transición. " $\varepsilon$ " se especifica para:  $\varepsilon(q_0, a), \varepsilon(q_1, b), \dots, \varepsilon(q_n, x)$ .

Existen varias formas de representar un Automata finito gráficamente. Una de ellas es encerrando cada estado en un círculo. De cada círculo parten flechas que pueden llegar a otros círculos o regresar al mismo círculo.

A cada flecha se le escribe un símbolo de entrada, tomado del Alfabeto  $\&$ , y una salida, si se trata de un Automata de Mealy.

Más adelante aplicaremos mucho este tipo de diagramas como modelo de nuestro programa.

Otra forma muy ilustrativa de representar un Automata es la siguiente:

El Automata consiste en un Control Finito que, encontrándose en un estado  $q$  contenido en  $Q$ , va leyendo una secuencia de símbolos de  $\Sigma$  escritos en una "cinta". Según se desplaza cada posición de la cinta, el Control finito puede cambiar de estado.

Una vez que el estado representado por el Control finito es un estado final, se dice que la secuencia hasta entonces recorrida es un lenguaje aceptado por el Automata. Observemos la figura 5.1.

### 5.3) Automata del Programa

A continuación presentamos todas las variables de entrada o de excitación que tenemos en nuestro automata. Recuérdese que para cada estado existe una transición, -ya sea al mismo o a otro estado, - asociado a cada excitación.

Variables de Entrada ( 0 Excitaciones):

"a": Esta variable nos señala que el operario ha pulsado alguna tecla, no importa cual. Podemos adelantar que la presencia de esta excitación conducirá al Control a la Rutina de atención a Teclado y Visualizador.

"b" : Indica que el Control recién ha salido de una Rutina de atención a Teclado y Visualizador. Esto es, el usuario ya ha tecleado la información que cree conveniente.

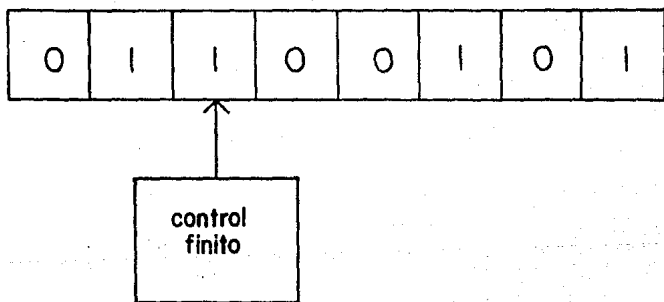


FIG. 5.1

"c" : Recuérdese que al pisar una tecla, el 8279 la almacea inmediatamente en una memoria FIFO (FIFO de teclado o FIFO de lectura). Esta variable es activa una vez que existe un solo caracter en la Memoria FIFO.

"d" : Existe más de 1 caracter en la FIFO de teclado.

"e" : La FIFO de teclado está vacía.

"f" : El caracter que se obtuvo al hacer una operación de lectura sobre la FIFO fue un dígito.

"g" : El caracter que se obtuvo al hacer una operación de lectura sobre la FIFO fue un \*. (Limpia Visualizador)

"h" : El caracter que se obtuvo al hacer una operación de lectura sobre la FIFO fue un #. (Valida información).

Las siguientes excitaciones se van a emplear para que el Control finito decida cuál abonado de la tarjeta va a probar: el 1, el 2 o el 3 :

"i" : El usuario acaba de pulsar un "1" y validarlo.

"j" : El usuario acaba de pulsar un "2" y validarlo.

"k" : El usuario acaba de pulsar un "3" y validarlo.

Las excitaciones que describiremos a continuación se utilizarán en la rutina de prueba:

"l" : La información que presenta a sus salidas el Circuito en prueba es correcta.

"m" : La información que presenta a sus salidas el Circuito bajo prueba es incorrecta.

"i" : El usuario acaba de teclear un "1" y validarlo. (Indicando que desea continuar la prueba una vez que ésta se ha suspendido para darle algún aviso).

"n" : El usuario acaba de pulsar un "9" y validarlo. (Indicando que desea que ya no continúe la prueba).

"p" : Se detecta un tono válido. (Transmisión correcta de una señal por el Bus PCM).

"q" : No se detecta tono. (No hay transmisión a través del Canal PCM).

"r" : El Circuito proporciona alimentación eléctrica correcta al teléfono.

"s" : El Circuito no proporciona alimentación eléctrica adecuada al teléfono.

"t" : El Circuito envía correctamente la Corriente de Llamada.

"u" : El Circuito no envía correctamente la Corriente de Llamada.

Ahora vamos a analizar por etapas el autómata que compone el Programa principal.

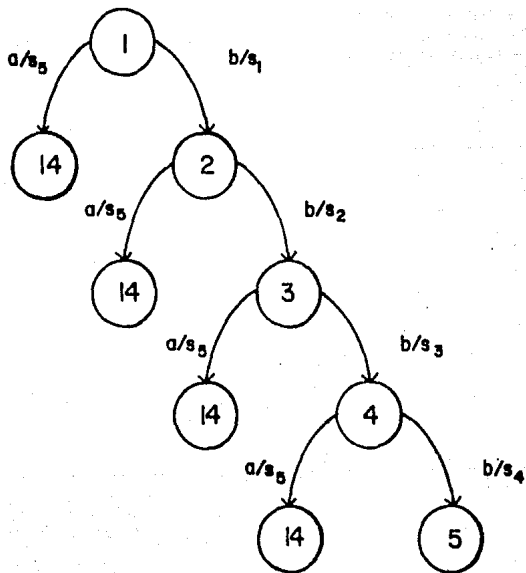


FIG. 5.2



La siguiente explicación se refiere a la fig. 5.2

Explicación de las salidas:

"S<sub>1</sub>" : Almacena Grupo y Elemento (Ahora en R7) en Registro Abonado 1. (R61)

Desactiva "b".

Avisa al usuario que la ejecución se detiene a esperar a que se teclee la información del siguiente abonado (Abonado 2).

Limpia el Registro que almacenaba los datos validados en el Visualizador. (R7)

"S<sub>2</sub>" : Almacena Grupo y Elemento (Ahora en R7) en Registro Abonado 2 (R62).

Desactiva "b".

Avisa al usuario que la ejecución se detiene a esperar a que se teclee la información del siguiente abonado (Abonado 3).

Limpia el Registro que almacenaba los datos validados en el Visualizador (R7).

"S<sub>3</sub>" : Almacena Grupo y Elemento (Ahora en R7) en Registro Abonado 3 (R63).

Desactiva "b".

Avisa al usuario que la ejecución se detiene a esperar a que se teclee la información del siguiente abonado (Abonado 4 ó Abonado en buen estado).

Limpia el Registro que almacenaba los datos validados en el Visualizador (R7).

"S<sub>4</sub>" : Almacena Grupo y Elemento (Ahora en R7) en Registro Abonado bien (R60).

Desactiva "b".

Limpia el Registro que almacenaba los datos validados en el Visualizador (R7).

Avisa al usuario que la ejecución se detiene a esperar que el operario elija qué circuito de la tarjeta desea probar: 1, 2 ó 3.

La salida "S<sub>5</sub>" se explicará hasta el siguiente diagrama, para mayor claridad.

#### Explicación del diagrama 5.2:

Este bloque del Autómata tiene la función de almacenar en localidades de RAM datos referentes al Grupo y Elemento que caracterizan a cada circuito de abonado de la Tarjeta que será examinada; así como el Grupo y Elemento que identifican al Circuito de Abonado que se sabe está en buen estado y que nos será de ayuda para la Rutina de Prueba.

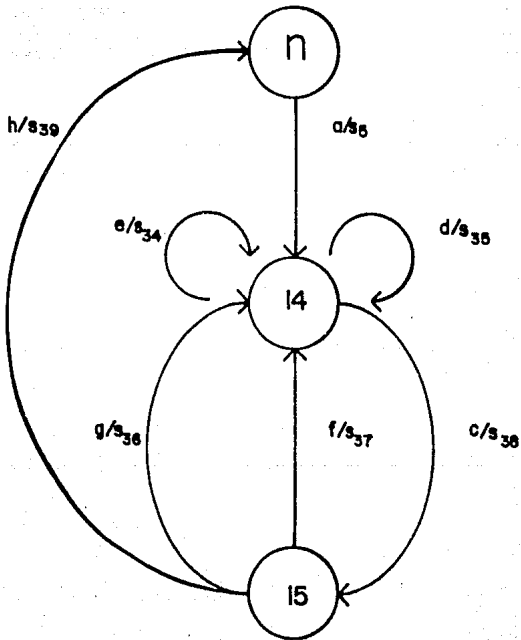


FIG. 5.3

La siguiente explicación se refiere a la figura 5.3

Explicación de las salidas:

"S<sub>5</sub>" : Almacena el Estado "n" en un Registro especial, -R5 de RB1, - para poder volver a él al finalizar la Rutina de atención a Teclado y Visualizador.

Escribe en el 8279 instrucciones para estar listo a escribir en el primer dígito del visualizador.

"S<sub>34</sub>" : No hace nada.

"S<sub>35</sub>" : Limpia la FIFO de Teclado en el 8279.

Escribe en el 8279 instrucción para estar listo a escribir en el primer dígito del visualizador.

"S<sub>36</sub>" : Limpia las primeras 4 localidades de RAM del Visualizador. (Equivale a borrar la información que hasta ahora haya tecleado el usuario).

"S<sub>37</sub>" : Busca en una tabla el dígito específico que se tecleó según la localización de la tecla pulsada.

Escribe el dígito tecleado en el Visualizador.

"S<sub>38</sub>" : Lee la localización de la tecla pulsada en la FIFO de Teclado del 8279. Almacénala en un Registro especial quizá para futura referencia (Si se trata de un dato).

Define la clasificación de dicha tecla. Puede ser: Un dato, botón de limpia de Visualizador o botón de validación de información.

Almacena en el Registro 37 la clasificación que corresponde.

"S<sub>39</sub>" : Lee el primer dígito del Visualizador y almacénalo en los 4 LSB\* del R7.

Lee el segundo dígito del Visualizador y almacénalo en los 4 MSB\* del R7.

Activa "b", avisando que se ha salido de atender una Rutina de Teclado y Visualizador.

Pasa el contenido de R5(RBI) a R3(RBO) para volver a la Rutina de Prueba.

Limpia los últimos 4 dígitos del Visualizador, avisando así al usuario que se vuelve a la Rutina de Prueba.

Explicación de la fig. 5.3 :

Esta parte del Automata es la que se encarga de atender a Teclado y Visualizador.

Se llega de un estado "n",-existen varios del que se puede hacer la transición,- y una vez que el usuario está satisfecho con la información que aparece en el Visualizador, la valida y regresa al estado "n".

Constantemente se investiga cuántos caracteres hay en la FIFO de Teclado. Si hay uno se lee y se pasa al estado 15. Si hay más de uno se limpia la FIFO y se espera a que se tecleen nuevos datos, puesto que no se le atendió a tiempo.

Se prevee que el operario pueda equivocarse al teclear algún dato, teniendo la facilidad de borrar los datos anteriores y alimentar nuevos datos.

\* LSB = Bits menos significativos. ( De menor valor )

MSB = Bits más significativos. ( De mayor valor ) .

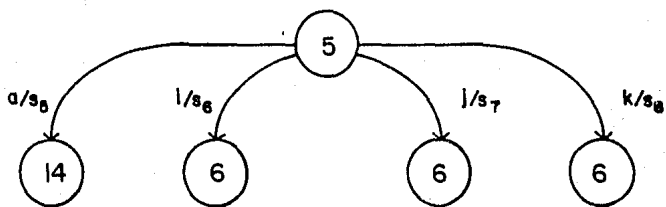


FIG. 5.4

La siguiente explicación se refiere a la fig. 5.4

Explicación de las salidas:

"S<sub>6</sub>" : Cuelga (Simula estado de abonado colgado).

Escribe en el Circuito 1 que suspenda el envío de corriente de llamada.

Inserta en el Registro 59, el Grupo y Elemento que caracterizan al abonado 1, almacenados en el Registro 61.

Limpia el Visualizador, informando así al operario que la prueba ha comenzado.

Bajo estas condiciones,-abonando colgado y así corriente de llamada,- almacena lo que debe ser la Información correcta leída en el circuito en el Registro 5.

Programa ambos Codecs,-tanto el del circuito en prueba como el del circuito en buen estado,- para quedar en reposo.

"S<sub>7</sub>" : Cuelga (Simula estado de abonado colgado).

Escribe en el Circuito 2 que suspenda el envío de corriente de llamada.

Inserta en el Registro 59, el Grupo y Elemento que caracterizan al abonado 2, almacenados en el Registro 62.

Limpia el Visualizador, informando así al operario que la prueba ha comenzado.

Bajo estas condiciones,-abonado colgado y sin corriente de llamada,- almacena lo que debe ser la Información correcta leída en el circuito en el Registro 5.

Programa ambos Codecs,-tanto el del circuito en prueba como el del circuito en buen estado,- para quedar en reposo.

"S<sub>g</sub>" : Cuelga (simula estado de abonado colgado).

Escribe en el Circuito 3 que suspenda el envío de corriente de llamada.

Inserta en el Registro 59, el Grupo y Elemento que caracterizan al abonado 3, almacenados en el Registro 63.

Limpia el Visualizador, informando así al operario que la prueba ha comenzado.

Bajo estas condiciones,-abonado colgado y sin corriente de llamada,- almacena lo que debe ser la Información correcta leída en el circuito en el Registro 5 .

Programa ambos Codecs,-tanto el del circuito en prueba como el del circuito en buen estado,-para quedar en reposo.

Explicación de la fig. 5.4 :

Esta parte del Autómata define cuál circuito de abonado de la tarjeta es el que va a estar sometido a prueba: Puede ser el 1, el 2 ó el 3.



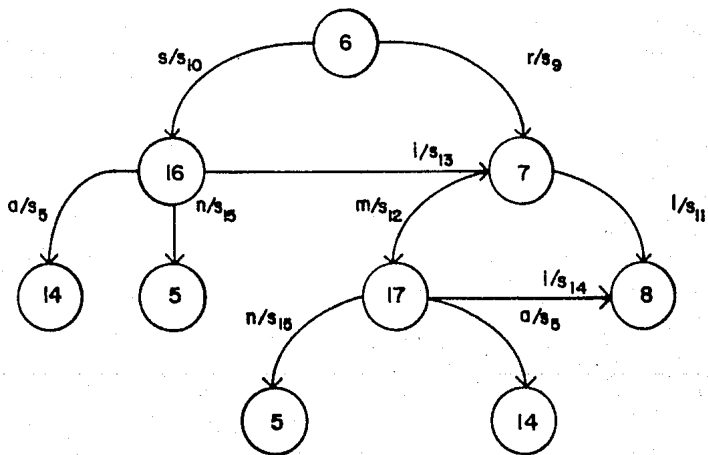


FIG. 5.5

La siguiente explicación se refiere a la fig. 5.5

Explicación de las salidas:

"S<sub>9</sub>" : No hace nada.

"S<sub>10</sub>" : Enseña al usuario el Error # 1 en el Visualizador.

(El Circuito no proporciona alimentación al teléfono).

Limpia el Registro 7, que almacenaba los datos validados anteriormente en el Visualizador.

"S<sub>11</sub>" : Escribe sobre el circuito bajo prueba la orden de envío de corriente de llamada.

Almacena en el Registro 5 lo que debe ser ahora la Información correcta que entregue el circuito bajo prueba al ejecutar una lectura sobre él (Colgado y con envío de corriente de llamada).

"S<sub>12</sub>" : Enseña al operario el Error # 2. (El Circuito no proporciona información correcta a sus salidas del estado en que se encuentra el abonado: (Colgado y sin corriente de llamada).

Limpia el Registro 7, que almacenaba los datos validados anteriormente del Visualizador.

"S<sub>13</sub>" : Limpia el Visualizador, indicando así al usuario que la prueba se ha reanudado.

"S<sub>14</sub>" : Limpia el Visualizador e indica al usuario de esta forma que la prueba continúa.

Escribe sobre el circuito bajo prueba la orden de enviar corriente de llamada.

Almacena en el Registro 5 lo que debe ser ahora la Información correcta que entregue el circuito bajo prueba al ejecutar una lectura sobre él (Colgado y con envío de corriente de llamada).

"S<sub>15</sub>" : Limpia el Visualizador e indica al usuario que el equipo está listo para recibir la elección del nuevo circuito que será sometido a prueba.

Limpia el Registro 7, que almacenaba los datos validados anteriormente en el Visualizador.

Explicación de la fig. 5.5 :

Esta parte del Autómata verifica que el circuito esté proporcionando alimentación eléctrica adecuada al abonado. Después, supervisa con una operación de lectura que la información proporcionada por éste sea correcta: Abonado colgado sin envío de corriente de llamada.

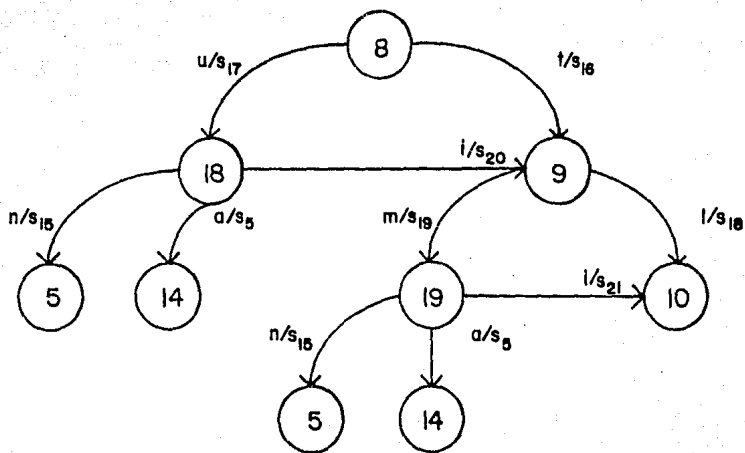


FIG. 5.6

La siguiente explicación se refiere a la fig. 5.6

Explicación de las salidas:

"S<sub>16</sub>" : No hace nada.

"S<sub>17</sub>" : Enseña al usuario el Error # 3. (El Circuito no envía corriente de llamada).

Limpia el Registro 7, que almacenaba los datos validados anteriormente del Visualizador.

"S<sub>18</sub>" : Descuelga. (Simula estado de Abonado descolgado).

Almacena en el Registro 5 lo que debe ser ahora la Información correcta que entregue el circuito bajo prueba al ejecutar una lectura sobre él (Descolgado y con envío de corriente de llamada)-

"S<sub>19</sub>" : Enseña al usuario el Error # 4. (El Circuito no proporciona información correcta a sus salidas del estado en que se encuentra el abonado: Colgado con envío de corriente de llamada).

Limpia el Registro 7, que almacenaba los datos validados anteriormente en el Visualizador.

"S<sub>20</sub>" : Limpia el Visualizador indicando así que la prueba se ha reanudado.

"S<sub>21</sub>" : Limpia el Visualizador, indicando así que la prueba se ha reanudado.

Descuelga (Simula estado de abonado descolgado)

Almacena en el Registro 5 lo que debe ser ahora la Información correcta que entregue el circuito bajo prueba al ejecutar una lectura sobre él (Descolgado y con envío de corriente de llamada).

Explicación de la fig. 5.6 :

Esta parte del Autómata verifica que el circuito envíe correctamente la corriente de llamada cuando así se le ordena. Después, supervisa con una operación de lectura que la información proporcionada por éste sea la correcta: Abonado colgado con envío de corriente de llamada.

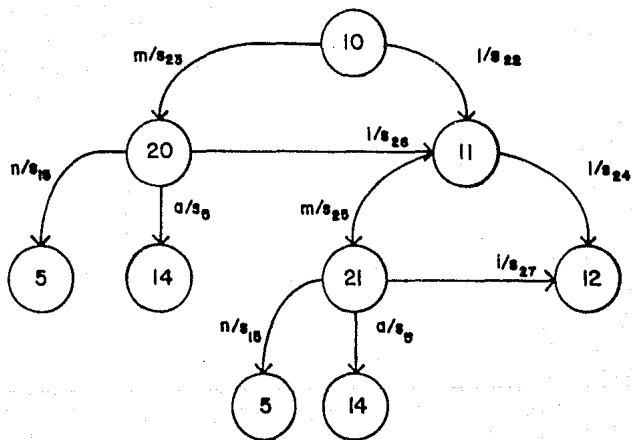


FIG. 5.7

La siguiente explicación se refiere a la fig. 5.7

Explicación de las salidas:

"S<sub>22</sub>" : Escribe sobre el Circuito bajo prueba la orden de suspender el envío de corriente de llamada.

Almacena en el Registro 5 lo que debe ser ahora la Información correcta que entregue el circuito bajo prueba al ejecutar una lectura sobre él (Descolgado y sin envío de corriente de llamada).

"S<sub>23</sub>" : Enseña al usuario el Error # 5 (El circuito no presenta a sus salidas la información correcta del estado en que se encuentra el abonado: Descolgado con envío de corriente de llamada).

Limpia el Registro 7, que almacenaba los datos validados anteriormente en el Visualizador.

"S<sub>24</sub>" : Primero se le ordena al Codec del Circuito bajo prueba quedar en reposo. Después, se le programa una ranura "Y" de recepción.

Se ordena al Codec del Circuito del Abonado bien quedar en reposo y posteriormente se le asigna una ranura "Y" de transmisión. Se aplica un tono a la línea telefónica atendida por el circuito en buen estado por el "Crosspoint" (Estableciendo una conexión entre el transmisor de tono y la línea).

Se establece una conexión entre el receptor de tono y la línea telefónica atendida por el circuito en prueba por medio del "Crosspoint"

"S<sub>25</sub>" : Enseña al usuario el Error # 6 (El Circuito no presenta a sus salidas la información correcta del estado en que se encuentra el abonado: Descolgado sin envío de corriente de llamada).

Limpia el Registro 7, que almacenaba los datos validados anteriormente en el Visualizador.



"S<sub>26</sub>" : Limpia el Visualizador, indicando así al usuario que la prueba continúa.

Escribe sobre el circuito en prueba la orden de suspender el envío de corriente de llamada.

Almacena en el Registro 5 lo que debe ser ahora la Información correcta que entregue el circuito bajo prueba al ejecutar una lectura sobre él. (Descolgado y sin envío de corriente de llamada).

"S<sub>27</sub>" : Limpia el Visualizador e indica al operario que la prueba sigue su curso.

Ordena al Codec del Circuito bajo prueba quedar en reposo. Después, se le programa una ranura "Y" de recepción.

Se ordena al Codec del Circuito del Abonado bien quedar en reposo y posteriormente se le asigna una ranura "Y" de transmisión. Se aplica un tono a la línea telefónica atendida por el circuito en buen estado por el "Crosspoint" (Estableciendo una conexión entre el transmisor de tono y la línea).

Se establece una conexión entre el receptor de tono y la línea telefónica atendida por el circuito en prueba por medio del "Crosspoint".

Explicación de la fig. 5.7 :

Esta parte del Automata verifica que la información que presente el circuito bajo prueba sea la correcta en los siguientes estados:

Abonado descolgado con envío de corriente de llamada

Abonado descolgado sin envío de corriente de llamada

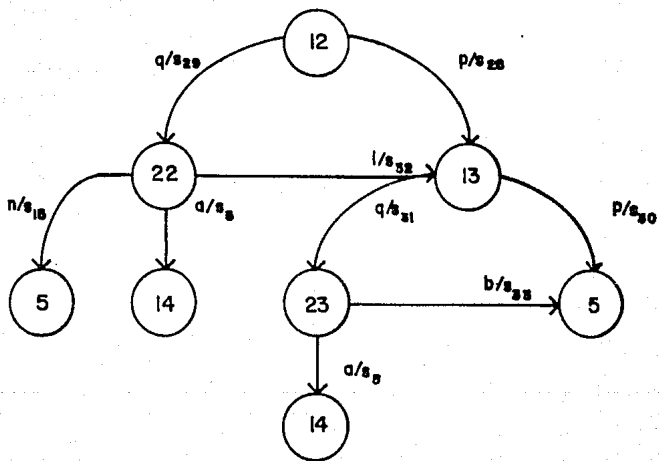


FIG. 5.8

La siguiente explicación se refiere a la fig. 5.8

Explicación de las salidas:

"S<sub>28</sub>" : Se ordena al Codec del circuito en prueba quedar en reposo. Después, se le programa una ranura "X" de transmisión.

Se ordena al Codec del circuito en buen estado quedar en reposo y posteriormente se le asigna una ranura "X" de recepción.

Se aplica un tono a la línea telefónica atendida por el circuito en prueba por el "Crosspoint". (Estableciendo una conexión entre el transmisor de tono y la línea).

Se establece una conexión entre el receptor de tono y la línea telefónica atendida por el abonado en buen estado por medio del "Crosspoint".

"S<sub>29</sub>" : Enseña el Error # 7 al usuario (La Etapa de Conmutación del circuito en prueba no recibe satisfactoriamente el tono enviado por el Bus PCM).

Limpia el registro que almacena los datos validados anteriormente en el Visualizador (R7).

"S<sub>30</sub>" : Limpia el Registro que almacena los datos validados anteriormente en el Visualizador (R7).

Informa al usuario que el equipo está listo para probar a otro circuito puesto que ya finalizó la actual.

"S<sub>31</sub>" : Enseña el Error # 8 al usuario (La Etapa de Conmutación del circuito en prueba no transmite satisfactoriamente el tono enviado desde la línea telefónica hasta el Bus PCM).

Desactiva "b".

Limpia el Registro que almacena los datos validados anteriormente en el Visualizador (R7).

"S<sub>32</sub>" : Limpia el Visualizador e indica así al operario que la prueba continúa.

Ordena al Codec del circuito bajo prueba quedar en reposo posteriormente se le asigna una ranura "X" de transmisión.

Programa al Codec del circuito del abonado en buen estado quedar en reposo y después asignarle una ranura "X" de recepción.

Se aplica un tono a la línea telefónica atendida por el circuito en prueba por el "Crosspoint". (Estableciendo una conexión entre el transmisor de tono y la línea).

Se establece una conexión entre el receptor de tono y la línea telefónica atendida por el abonado en buen estado por medio del "Crosspoint".

"S<sub>33</sub>" : Desactiva "b".

Informa al usuario que el equipo ha terminado la rutina de prueba y está listo para probar otro circuito.

Explicación de la fig. 5.8 :

Esta parte del Automata es útil para ver si la Etapa de Conmutación de la tarjeta está trabajando satisfactoriamente. Primero se verifica que el circuito bajo prueba reciba bien el tono que se envía por el Bus PCM. Después se envía un tono por la línea atendida por él y se supervisa que sea transmitido adecuadamente al Bus PCM.

#### 5.4) Algoritmo de la Rutina de Prueba

Ahora bien, el método tradicional para implementar un Autómata es por medio de una Tabla de transiciones. En una tabla de este tipo, se asignan para cada par de estado y variable de entrada posibles, un cuadro en el que se escribe el estado al que se transferirá el Control al presentarse dicha combinación.

Sin embargo en nuestro modelo es fácil observar que para cada estado solo existen transiciones con la presencia de unas pocas variables. Por lo tanto la realización de una tabla implicaría que aproximadamente un 80% de la misma se desperdiciaría. Puesto que ya en la práctica esta tabla se almacena en memoria PROM, nos conviene implementar nuestro modelo de otro modo.

Por ello, optamos representar el Autómata por medio de listas ligadas. Se precisa de 4 listas:

La información que contienen estas listas es la siguiente:

Lista # 1 : Apuntadores.

Lista # 2 : Excitaciones que se pueden presentar para cada estado.

Lista # 3 : Estados a los que se puede transferir el Control para cada transición.

Lista # 4 : Direcciones de las subrutinas de las salidas que están asociadas a cada transición, puesto que se trata de un Autómata de Mealy.

Lista#1

1	3	5	7	9	13	15	17	19	21	23	25	27	29	32	35	38	41	44	47	50	53	56	
---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	--

Lista#2

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
a	b	a	b	a	b	a	b	a	i	j	k	s	r	m	l	u	t	m	l	m	l	m	l

Lista#3

14	2	14	3	14	4	14	5	14	6	6	6	16	7	17	8	18	9	19	10	20	11	21	12
----	---	----	---	----	---	----	---	----	---	---	---	----	---	----	---	----	---	----	----	----	----	----	----

Lista#4

2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34	36	38	40	42	44	46	48
---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

Lista#1

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

Lista#2

25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
q	p	q	p	e	d	c	h	g	f	a	n	i	n	a	i	n	a	i	n	a	i	n	a

Lista#3

22	13	23	5	14	14	15	-	14	14	14	5	7	5	14	8	5	14	9	5	14	10	5	14
----	----	----	---	----	----	----	---	----	----	----	---	---	---	----	---	---	----	---	---	----	----	---	----

Lista#4

50	52	54	56	58	60	62	64	66	68	70	72	74	76	78	80	82	84	86	88	90	92	94	96
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----



Podemos afirmar que durante su ejecución el Programa hace constantemente referencia a estas 4 listas.

Las localidades de la Lista # 1 almacenan apuntadores, uno para cada estado del Automata. Estos apuntadores señalan lugares específicos de las listas 2,3 y 4.

Ya hemos dicho para cada estado de nuestro modelo matemático de Automata solo existen transiciones para unas cuantas variables, siendo éstas un pequeño porcentaje comparado con el total de variables existentes.

Damos ahora un ejemplo: Vamos a suponer que el Control se halla en el estado q(16). Para este estado solo existen 3 transiciones posibles. Estas transiciones ocurren con las variables "a", "n", "i".

En la localidad 16 de la Lista # 1 aparecerá un número, por ejemplo el 35.

El Programa investiga inicialmente la excitación presente en el Sistema, en sus terminales de entrada. (Excitación real).

Después la compra con la excitación en la tabla 2 (Excitación en tablas) presente en la localidad 35 de la Lista # 2.

Si no son iguales, incrementa en uno el apuntador ( $35+1=36$ ) y ahora compra la Excitación real con la Excitación en tablas almacenada en la localidad 36 de la Lista # 2.

Si no son iguales, incrementa en uno el apuntador ( $36+1=37$ ) y ahora compara la Excitación real con la Excitación en tablas almacenada en la localidad # 37 de la Lista # 2.

Si no fué igual, entonces vuelve a investigar la Excitación real que se presenta en el Sistema y otra vez realiza la rutina de



compararlo contra las 3 excitaciones almacenadas en la Lista # 2.

Este proceso se repite hasta que se detecta una igualdad en la comparación realizada. Esto indica que se debe ejecutar una transición, por lo tanto se consultan ahora las listas 3 y 4.

La lista # 3 nos dice a qué nuevo estado se transferirá el Control.

La lista # 4 nos dice qué dirección contiene la salida asociada a dicho transición y por lo tanto debe ejecutarse.

Esto es a grandes rasgos lo que hace el Programa. Las listas se hallan en la pág. 3 de la Memoria PROM de Programa. Si en un futuro se decide alterar el programa, ya sea para añadir nuevos estados o quitar algunos ya existentes; o bien simplemente modificar los diagramas, bastará añadir o quitar datos a estas listas, pero la estructura principal del programa permanece inalterada.

Ahora escribiremos el Algoritmo del Programa para realizar el proceso explicado anteriormente:

1) Inicia con el estado 1.

RAM (3) = 1

2) Busca el valor del apuntador en la lista # 1 que corresponde al estado actual. (Estado actual = RAM (3)). Almacena el valor encontrado en RAM (4).

ACUM = RAM (3) +LOCI

RAM (4) =ROM(ACUM)

3) Busca la primera excitación,-información leída del circuito de abonado,- y almacénala en RAM(32). (Lectura correcta o incorrecta).

4) Busca la segunda excitación,- verificación de existencia de tono en el Receptor DTMF,-y almacénala en RAM(33).

5) Busca la tercera excitación,-información validada en la última Rutina de Teclado y Visualizador,-y almacénala en RAM(34).

6) Busca la cuarta excitación,-solicitud de atención a Teclado y Visualizador hecha por el usuario,-y almacénala en RAM(35).

7) Busca la quinta excitación,-número de caracteres almacenados en la FIFO de Teclado del 8279,- y almacénala en RAM(36).

8) Busca la octava excitación,-alimentación eléctrica correcta proporcionada por el Circuito de Abonado,- y almacénala en RAM(39).

9) Busca la novena excitación,-envío de corriente de llamada satisfactorio del circuito de abonado,y almacénala en RAM(40).

10) Consulta la excitación presente en tablas (lista # 2). La localidad que se va a consultar está señal por la variable RAM(4).

Almacena la excitación en RAM(2)

ACUM = RAM(4) + LOC2

RAM(2) = ROM(ACUM)

11) Esta excitación en tablas,- la hallada en el paso 10,- puede ser cualquiera de las que ya hemos visto: solicitud de atención a Teclado y Visualizador, o de información que debe ser leída en el circuito en prueba, o de recepción correcta de tono, etc.

En este paso se decidirá contra cuál excitación real de las buscadas anteriormente, -pasos 3 a 9,- se debe comparar.

Los 4 MSB sirven para este fin.

ACUM = RAM(2)

Pasa los 4 MSB a los 4 LSB

Haz los 4 MSB igual a cero

ACUM = ACUM + 20

Compara RAM(2) con RAM(ACUM). Si son iguales salta al paso # 15. Si no son iguales, continúa con el paso # 12.

12) Puesto que la excitación consultada en tablas no correspondió a la excitación real, se incrementará en uno el valor del apuntador, -almacenado en RAM(4), -de tal forma que a la siguiente vuelta se lea la siguiente localidad en la lista # 2.

RAM(4) = RAM(4) + 1

13) Debemos verificar por supuesto que el apuntador no sobrepase el valor máximo que puede tomar para el estado en que nos encontramos.

ACUM = RAM(3) + 1 + LOCI

14) Compara ROM(ACUM) con RAM(4)

En este momento ROM(ACUM) = Valor máximo que puede tomar RAM(4) y que señala el apuntador para el siguiente estado.

Si ROM(ACUM) RAM(4) salta al paso #10.

En otro caso salta al paso # 2.

15) Puesto que la excitación real correspondió a la excitación en tablas, se debe producir una transición.

Lo primero que hacemos es almacenar el estado actual, -antes de la transición,- en otra variable: RAN(3), para poder volver a él en caso de que vayamos a atender una Rutina de Teclado y Visualizarlo.

RAN(3) = RAM(3)

16) Pasamos ahora el nuevo estado al que el Control se va a transferir a RAM(3). Para ello precisamos de consultar la lista # 3.

ACUM = RAM(4) + LOC3

RAM(3) = ROM(ACUM)

17) Ahora ejecutaremos la salida que corresponde a esta transición. Para ello consultamos la lista # 4, que almacena las direcciones de las subrutinas a las que debe transferirse la ejecución del programa para llevar a cabo las instrucciones de salida.

ACUM = RAM(4) + LOC4

ACUM = ROM(ACUM)

Ahora ACUM es la dirección a la que se debe saltar para proseguir con la ejecución del programa.

18) Salta a la localidad del programa: ACUM.

Al final de la ejecución de la salida se salta al paso # 2.

**NOTA:**

LOC1 = Constante cuyo valor es el de la dirección de la localidad anterior al comienzo de la lista # 1.

LOC2 = Constante cuyo valor es el de la dirección de la localidad anterior al comienzo de la lista # 2.

LOC3 = Constante cuyo valor es el de la dirección de la localidad anterior al comienzo de la lista # 3.

LOC4 = Constante cuyo valor es el de la dirección de la localidad anterior al comienzo de la lista # 4.

RAM(3) = Variable que almacena el estado en el que se está trabajando.

RAM(4) = Variable que almacena el valor que toma el apuntador.

RAM(2) = Variable que almacena la excitación consultada en la lista # 2.

Las excitaciones reales 6 y 7 no se incluyen en esta Rutina y se buscan solo al ejecutar ciertas salidas:

Excitación 6: Caracter teclado: #, \* ó un dato. (Se guarda en el Registro 37).

Excitación 7: Ya se ha pasado por una Rutina de atención a Teclado y Visualizador (Almacenada en el Registro 38).

ROM(n) = Constante dimensionada cuyo valor representa el dato almacenado en la dirección "n" de la memoria PROM

RAM(n) = Variable dimensionada cuyo valor representa el dato almacenado en la localidad "n" de la memoria RAM

## 5.5) Programa

LO1	EQU	00H
LO2	EQU	23D
LO3	EQU	80D
LO4	EQU	137D
TRU	EQU	57D
ABPRU	EQU	59D
ABBIE	EQU	60D
AB1	EQU	61D
AB2	EQU	62D
AB3	EQU	63D

	ORG	000H
	JMP	INIC;
	DS	8
JETE:	ORG	00AH
	MOVP	A, &A;
	RET;	
	ORG	010H
	DB	77H
	DB	88H
	DB	99H
	DB	0AAH
	DB	44H
	DB	55H
	DB	66H
	DB	0BBH
	DB	11H
	DB	22H
	DB	33H
	DB	0CCH
	DB	00H
	DB	0FFH
	DB	0EEH
	DB	0DDH
	ORG	020H
SOLU:	JMP	&A;
	JMP	SOL1;
	JMP	SOL2;

	JMP	SOL3;
	JMP	SOL4;
	JMP	SOL5;
	JMP	SOL6;
	JMP	SOL7;
	JMP	SOL8;
	JMP	SOL9;
	JMP	SOL10;
	JMP	SOL11;
	JMP	SOL12;
	JMP	SOL13;
	JMP	SOL14;
	JMP	SOL15;
	JMP	SOL16;
	JMP	SOL17;
	JMP	SOL18;
	JMP	SOL19;
	JMP	SOL20;
	JMP	SOL21;
	JMP	SOL22;
	JMP	SOL23;
	JMP	SOL24;
	JMP	SOL25;
	JMP	SOL26;
	JMP	SOL27;
	JMP	SOL28;
	JMP	SOL29;
	JMP	SOL30;
	JMP	SOL31;
	JMP	SOL32;
	JMP	SOL33;
	JMP	SOL34;
	JMP	SOL35;
	JMP	SOL36;
	JMP	SOL37;
	JMP	SOL38;
	JMP	SOL39;
	ORG	084H
PLU:	MOVP	A, &A;
	RET;	
	ORG	087H
	DB	40H
	DB	41H
	DB	42H
	DB	42H
	DB	42H
	DB	42H
	DB	42H
	DB	42H

INIC:

```
ORL      P2,#10H;
ORL      P2,#20H;
ANL      P2,#0EFH;
MOV      A,#0FFH;
MOVD     P4,A;
MOVD     P7,A;
MOV      A,#0FEH;
ANLD     P7,A;
ORL      P2,#10H;
ANL      P2,#0DFH;
MOV      A,#0FFH;
MOVD     P5,A;
MOVD     P4,A;
ORL      P2,#20H;
CALL     HAB79;
CALL     COM79;
CLR      A;
MOVX     &R0,A;
MOV      A,#24H;
MOVX     &R0,A;
MOV      A,#0C2H;
MOVX     &R0,A;
CALL     TARDA;
CALL     DES79;
MOV      R7,#00H;
MOV      R0,#37D;
MOV      &R0,50H;
MOV      R0,#39D;
MOV      &R0,#70H;
MOV      R0,#38D;
MOV      &R0,#60H;
CALL     CLDIR;
MOV      R0,#55D;
MOV      A,#10H;
CALL     ALSDI;
MOV      R0,#55D;
CALL     ENSEN;
```



	MOV	R3, #01H;
COMBO:	MOV	A, R3;
	ADD	A, #L01;
	MOVP3	A, &A;
	MOV	R4, A;
	MOV	R2, #00H;
BUEx:	MOV	R0, #20H;
	CALL	BUEx1;
	MOV	R0, #21H;
	CALL	BUEx2;
	MOV	R0, #22H;
	CALL	BUEx3;
	MOV	R0, #23H;
	CALL	BUEx4;
	MOV	R0, #24H;
	CALL	BUEx5;
	MOV	R0, #27H;
	CALL	BUEx8;
	MOV	R0, #28H;
	CALL	BUEx9;
URA:	MOV	A, R4;
	ADD	A, #L02;
	MOVP3	A, &A;
	MOV	R2, A;
	SWAP	A;
	ANL	A, #0FH;
	ADD	A, #20H;
	MOV	R0, A;
	MOV	A, &R0;
	XRL	A, R2;
	JZ	IG;
NOIG:	MOV	A, R4;
	INC	A;
	MOV	R4, A;
	MOV	A, R3;
	INC	A;

	ADD	A, #L01;
	MOVP3	A, &A;
	MOV	R0, #TRU;
	MOV	&R0, A;
	MOV	A, R0;
	INC	A;
	MOV	R0, A;
	MOV	A, R4;
	MOV	&R0, A;
	MOV	R0, #TRU;
	CALL	SUB;
	JC	COMBO;
	JMP	URA;
IG:	MOV	A, R3;
	SEL	RB 1;
	MOV	R3, A;
	SEL	0;
	MOV	A, R4;
	ADD	A, #LO3;
	MOVP3	A, &A;
	MOV	A, R3;
	MOV	A, R4;
	ADD	A, #LO4;
	MOVP3	A, &A;
	ADD	A, #1FH;
	JMP	SOLU;

SUB:	CLR	C;
	MOV	A, &R0;
	CPL	A;
	INC	A;
	MOV	R1, A;
	MOV	A, R0;
	INC	A
	MOV	R0, A;
	MOV	A, &R0;
	ADD	A, R1;
	RET;	
BUEX1:	MOV	R1, #ABPRU;
	MOV	A, &R1;
	ANL	P2, #0EFH;
	MOVD	P6, A;
	SWAP	A;
	MOVD	P5, A;
	ORL	P2, #10H;
	ANL	P2, #0DFH;
	MOV	A, #0EH;
	ANLD	P4, A;
	MOVD	A, P6;
	SWAP	A;
	MOV	R6, A;
	MOVD	A, P7;
	ADD	A, R6;
	MOV	R6, A;
	MOV	A, #01H;
	ORLD	P4, A;
	ORL	P2, #20H;
	MOV	A, R6;
	XRL	A, R5;
	JZ	LB;
LE:	MOV	A, #00H;
	MOV	&R0, A;
	RET;	
LB:	MOV	A, #01H;
	MOV	&R0, A;
	RET;	
BUEX2:	ANL	P2, #0EFH;
	MOVD	A, P4;
	ANL	A, #01H;
	ADD	A, #10H;
	MOV	&R0, A;
	ORL	P2, #10H;
	RET;	

BUEX3:	MOV ANL ADD MOV RET;	A,R7; A,#0FH; A,#20H; &R0,A;
BUEX4:	JT1 MOV JMP MOV RET;	\$ + 6; &R0,#30H; \$ + 4; &R0,#31H;
BUEX5:	CALL CALL MOVX ANL ADD CALL MOV CALL RET;	HAB79; COM79; A,&R0; A,#07H; A,#87H; PLU; &R0,A; DES79;
BUEX8:	ANL MOVD ANL XRL JZ MOV JMP MOV RET;	P2,#0DFH; A,P5; A,#03H; A,#01H; \$ + 6; &R0,#70H; \$ + 4; &R0,#71H;
BUEX9:	ANL MOVD ANL XRL JZ MOV JMP MOV RET;	P2,#0DFH; A,P5; A,#0CH; A,#04H; \$ + 6; &R0,#80H; \$ + 4; &R0,#81H;

SOL1:	MOV CALL MOV CALL MOV MOV MOV CALL JMP	R0, #3DH; GE; R7, #00H; LIMGA; R0, #55D; A, #20H; R0, #55D; ENSEN; COMBO;
SOL2:	MOV CALL CALL MOV MOV MOV CALL MOV CALL JMP	R0, #3EH; GE; LIMGA; R7, #00H; R0, #55D; A, #30H; ALSDI; R0, #55D; ENSEN; COMBO;
SOL3:	MOV CALL CALL MOV MOV MOV CALL MOV CALL JMP	R0, #3Fh; GE; LIMGA; R7, #00H; R0, #55D; A, #40H; ALSDI; R0, #55D; ENSEN; COMBO;
SOL4:	MOV CALL CALL MOV JMP	R0, #3cH; GE; LIMGA; R0, 55D; SOL15;
SOL5:	CALL CALL JMP	GU; ESD11; COMBO;
SOL6:	SEL MOV SEL MOV CALL JMP	RB1; R4, #10H; RB0; R0, #3DH; TECL1; COMBO;

SOL7:	SEL MOV SEL MOV CALL JMP	RB1; R4,#12H; RB0; R0,#3EH; TECL1; COMBO;
SOL8:	SEL MOV SEL CALL JMP	RB1; R4,#11H; RB0; TECL1; COMBO;
SOL9:	NOP; JMP	COMBO;
SOL10:	MOV MOV CALL MOV CALL MOV JMP	R0,#58D; A,#10H; ALMDI; R0,#55D; ENSEN; R7,#00H; COMBO;
SOL11:	MOV CALL MOV JMP	R0,#3BH; POLIN; R5,#0BFH; COMBO;
SOL12:	MOV MOV CALL MOV CALL MOV JMP	R0,#58D; A,#20H; ALMDI; R0,#55D; ENSEN; R7,#00H; COMBO;
SOL13:	CALL JMP	CLDIR; COMBO;
SOL14:	CALL MOV CALL MOV JMP	CLDIR; R0,#3BH; POLIN; R5,#0BFH; COMBO;

SOL15:	MOV MOV INC MOV INC MOV INC MOV MOV CALL MOV JMP	R0, #55D; &R0, #50H; R0; &R0, #50H; R0; &R0, #50H; R0; &R0, #50H; R0, #55D; ENSEN; R7, #00H; COMBO;
SOL16:	NOP; JMP	COMBO;
SOL17:	MOV MOV CALL MOV CALL MOV JMP	R0, #58D; A, #30H; ALMDI; R0, #55D; ENSEN; R7, #00H; COMBO;
SOL18:	CALL MOV JMP	DESCU; R5, #3Fh; COMBO;
SOL19:	MOV MOV CALL MOV CALL MOV JMP	R0, #58D; A, #40H; ALMDI; R0, #55D; ENSEN; R7, #00H; COMBO;
SOL20:	CALL JMP	CLDIR; COMBO;
SOL21:	CALL JMP	CLDIR; COMBO;
SOL22:	CALL MOV JMP	POLNO; R5, #0FDH; COMBO;

ORG	300H
DB	00H
DB	01H
DB	03H
DB	05H
DB	07H
DB	09H
DB	0DH
DB	0FH
DB	11H
DB	13H
DB	15H
DB	17H
DB	19H
DB	1BH
DB	1DH
DB	20H
DB	23H
DB	26H
DB	29H
DB	2CH
DB	2FH
DB	32H
DB	35H
DB	38H

DB	31H
DB	61H
DB	31H
DB	61H
DB	31H
DB	61H
DB	31H
DB	61H
DB	31H
DB	21H
DB	22H
DB	23H
DB	70H
DB	71H
DB	00H
DB	01H
DB	80H
DB	81H
DB	00H
DB	01H
DB	00H
DB	01H
DB	00H
DB	01H
DB	21H
DB	29H



DB	31H
DB	21H
DB	29H
DB	31H
DB	21H
DB	31H
DB	61H

DB	0EH
DB	02H
DB	0EH
DB	03H
DB	0EH
DB	04H
DB	0EH
DB	05H
DB	0EH
DB	06H
DB	06H
DB	06H
DB	0FH
DB	07H
DB	11H
DB	08H
DB	12H
DB	09H
DB	13H
DB	0AH
DB	14H
DB	0BH
DB	15H
DB	0CH
DB	16H
DB	0DH
DB	17H
DB	05H
DB	0EH
DB	0EH
DB	0FH
DB	00H
DB	0EH
DB	0EH
DB	0EH
DB	05H
DB	07H
DB	05H
DB	0EH
DB	08H
DB	05H
DB	0EH
DB	09H

DB	05H
DB	0EH
DB	0AH
DB	05H
DB	0EH
DB	0BH
DB	05H
DB	0EH
DB	0CH
DB	05H
DB	0EH
DB	0DH
DB	0EH
DB	05H

DB	02H
DB	04H
DB	06H
DB	08H
DB	0AH
DB	0CH
DB	0EH
DB	10H
DB	12H
DB	14H
DB	16H
DB	18H
DB	1AH
DB	1CH
DB	1EH
DB	20H
DB	22H
DB	24H
DB	26H
DB	28H
DB	2AH
DB	2CH
DB	2EH
DB	30H
DB	32H
DB	34H
DB	36H
DB	38H
DB	3AH
DB	3CH
DB	3EH
DB	40H
DB	42H
DB	44H
DB	46H
DB	48H
DB	4AH

DB	4CH
DB	4EH
DB	50H
DB	52H
DB	54H
DB	56H
DB	58H
DB	5AH
DB	5CH
DB	5EH
DB	60H
DB	62H
DB	64H
DB	66H
DB	68H
DB	6AH
DB	6CH
DB	6EH
DB	70H
DB	72H

SOL23:	MOV	RO, #58D;
	MOV	A, #50H;
	CALL	ALMDI;
	MOV	RO, #55D;
	CALL	ENSEN;
	MOV	R7, #00H;
	JMP	COMBO;

SOL24:	MOV	RO, #3BH;
	CALL	PRORE;
	MOV	RO, #3CH;
	CALL	PROTR;
	CALL	LIMXP;
	CALL	HABXP;
	MOV	A, #13H;
	MOVX	&RO, A;
	SEL	RB1;
	MOV	A, R4;
	SEL	RB0;
	ADD	A, #08H;
	MOVX	&RO, A;
	CALL	DESXP;
	JMP	COMBO;

SOL25:	MOV	RO, #58D;
	MOV	A, #60H;
	CALL	ALMDI;

	MOV	RO,#55D;
	CALL	ENSEN;
	MOV	R7,#00H;
	JMP	COMBO;
SOL26:	CALL	CLDIR;
	JMP	SOL22;
SOL27:	CALL	CLDIR;
	JMP	SOL24;
SOL28:	MOV	RO,#ABBIE;
	CALL	STAND;
	MOV	RO,#3BH;
	CALL	STAND;
	MOV	RO,#3BH;
	CALL	PROTR;
	MOV	RO,#3CH;
	CALL	PRORE;
	CALL	LIMXP;
	CALL	HABXP;
	MOV	A,#1BH;
	MOVX	&R0,A;
	SEL	RB1;
	MOV	A,R4;
	SEL	RB0;
	MOVX	&R0,A;
	CALL	DESXP;
	JMP	COMBO;
SOL29:	MOV	RO,#58D;
	MOV	A,#70H;
	CALL	ALMDI;
	MOV	RO,#55D;
	CALL	ENSEN;
	MOV	R7,#00H;
	JMP	COMBO;
SOL30:	JMP	SOL15;
SOL31:	MOV	RO,#58D;
	MOV	A,#80H;
	CALL	ALMDI;
	MOV	RO,#55H;
	CALL	ENSEN;
	MOV	R7,#00H;
	CALL	LIMGA;
	JMP	COMBO;

SOL32:	CALL JMP	CLDIR; SOL28;
SOL33:	CALL JMP	LIMGA; SOL15;
SOL34:	NOP; JMP	COMBO;
SOL35:	CALL CALL MOV MOVX CALL CALL JMP	HAB79; COM79; A,#0C2H; &R0,A; DES79; ESD11; COMBO;
SOL36:	CALL CALL MOV MOVX CALL MOV MOVX MOVX MOVX MOVX CALL CALL JMP	HAB79; COM79; A,#90H; &R0,A; DAT79; A,#0FFH; &R0,A; &R0,A; &R0,A; &R0,A; &R0,A; DES79; ESD11; COMBO;
SOL37:	SEL MOV SEL RRC RRC RRC RL RL ANL CALL MOV CALL CALL MOV MOVX CALL JMP	RB1; A,R6; RB0; A; A; A; A; A; A; A,#0FH; JETE; R1,A; HAB79; DAT79; A,R1; &R0,A; DES79; COMBO;

SOL38:	CALL	LEEFI;
	JB2	GALI;
	MOV	RO,#25H;
	MOV	&RO,#53H;
	JMP	SIGU;
GALI:	JB3	GA;
	MOV	RO,#25H;
	MOV	&RO,#51H;
	JMP	SIGU;
GA:	MOV	RO,#25H;
	MOV	&RO,#52H;
SIGU:	JMP	COMBO;

SOL39:	CALL	HAB79;
	CALL	COM79;
	MOV	A,#70H;
	MOVX	&RO,A;
	MOVX	A,&RO;
	SWAP	A;
	ANL	A,#0FH;
	MOV	R7,A;
	MOVX	A,&RO;
	ANL	A,#0F0H;
	ADD	A,R7;
	MOV	R7,A;
	CALL	DES79;
	MOV	RO,#38D;
	MOV	&RO,#61H;
	SEL	RB1;
	MOV	A,R5;
	SEL	RB0;
	MOV	R3,A;
	CALL	CLDIR;
	JMP	COMBO;

GE:	MOV MOV RET;	A, R7; &R0, A;
ALSDI:	MOV INC MOV INC MOV INC MOV RET;	&R0, A; R0; &R0, #00H; R0; &R0, #00H; R0; &R0, #00H;
ENSEN:	CALL CALL MOV MOVX CALL MOV CALL CALL CALL CALL CALL RET;	HAB79; COM79; A, #94H; &R0, A; DAT79; R0, #55D; ESCDI; ESCDI; ESCDI; ESCDI; DES79;
ESCDI:	MOV MOVX INC RET;	A, &R0; &R0, A; R0;
ESDI1:	CALL CALL MOV MOVX CALL RET;	HAB79; COM79; A, #90H; &R0, A; DES79;
STAND:	CALL MOV MOV CALL CALL CALL MOV MOVX CALL RET;	ESGS1; R0, #TRU; &R0, #0FDH; ESGIN; SENES; HABPC; A, #0FFH; &R0, A; TARDA;

TECL1:	CALL MOV CALL CALL CALL MOV CALL MOV CALL MOV CALL CALL CALL CALL RET;	AVPRU; R5, #0FFH; CUELG; POLNO; CLDIR; R0, #AB1; STAND; R0, #AB2; STAND; R0, #AB3; STAND; R0, #ABBIE; STAND; LIMXP;
TARDA:	MOV MOV STRT MOV JZ JMP STOP RET;	A, #0FH; T, A; T; A, T; \$ + 4; \$ - 3; TCNT;
ALMDI:	MOV DEC MOV DEC MOV DEC MOV RET;	&R0, A; R0; &R0, #00H; R0; &R0, #00H; R0; &R0, #00H;
GU:	SEL MOV MOV RETR;	RB1; A, R3; R5, A;
AVPRU:	MOV MOV MOV RET;	A, &R0; R0, #3BH; &R0, A;
CUELG:	ANL MOV ORLD ORL RET;	P2, #0EFH; A, #08H; P7, A; P2, #10H;



POLNO:	MOV CALL MOV MOV CALL CALL RET;	R0,#3BH; ESGS1; R0,#TRU; &R0,#OFFH; ESCIN; SENES;
ESGS1:	ANL MOV MOVD SWAP MOVD ORL RET;	P2,#0EFH; A,&R0; P6,A; A; P5,A; P2,#10H;
ESCIN:	ANL MOV MOVD SWAP MOVD ORL RET;	P2,0DFH; A,&R0; P7,A; A; P6,A; P2,#20H;
SENES:	ANL MOV ANLD MOV ORLD ORL RET;	P2,#0DFH; A,#0DH; P4,A; A,#02H; P4,A; P2,#20H;
CLDIR:	CALL CALL MOV MOVX CALL CALL CALL RET;	HAB79; COM79; A,#90H; &R0,A; DAT79; HEXDI; DES79;
HAB79:	ANL MOV ANLD ORL RET;	P2,#0DFH; A,#0BH; P4,A; P2,#20H;

COM79:	ANL MOV ORLD ORL RET;	P2,#0DFH; A,#08H; P4,A; P2,#20H;
DAT79:	ANL MOV ANLD ORL RET;	P2,#0DFH; A,#07H; P4,A; P2,#20H;
HEXDI:	MOV MOVX MOVX MOVX MOVX MOVX MOVX MOVX RET;	A,#0FFH; &R0,A; &R0,A; &R0,A; &R0,A; &R0,A; &R0,A; &R0,A; &R0,A;
DES79:	ANL MOV ORLD ORL RET;	P2,#0DFH; A,#04H; P4,A; P2,#20H;
POLIN:	MOV CALL MOV MOV CALL CALL RET;	R0,#3BH; ESGS1; R0,#TRU; &R0,#0BFH; ESCIN; SENES;
DESCU:	ANL MOV ANLD ORL RET;	P2,#0EFH; A,#07H; P7,A; P2,#10H;
PRORE:	CALL MOV MOV CALL CALL	ESGS1; R0,#TRU; &R0,#0FDH; ESCIN; SENES;

	CALL	HABPC;
	MOV	A, #58H;
	MOVX	&R0, A;
	CALL	DESPC;
	CALL	TARDA;
	RET;	
PROTR:	CALL	ESGS1;
	MOV	RO, #TRU;
	MOV	&R0, #OFDH;
	CALL	ESCIN;
	CALL	SENES;
	CALL	HABPC;
	MOV	A, #44H;
	MOVX	&R0, A;
	CALL	DESPC;
	CALL	TARDA;
	RET;	
HABPC:	ANL	P2, #0EFH;
	MOV	A, #0BH;
	ANLD	P7, A;
	CPL	A;
	ORLD	P7, A;
	ORL	P2, #10H;
	RET;	
LEEFI:	CALL	HAB79;
	CALL	COM79;
	MOV	A, #40H;
	MOVX	&R0, A;
	CALL	DAT79;
	MOVX	A, &R0;
	SEL	RB1;
	MOV	R6, A;
	RETR;	
HABXP:	ANL	P2, #0EFH;
	MOV	A, #0DDH;
	ANLD	P6, A;
	ORL	P2, #10H;
	RET;	
DESP:	ANL	P2, #0EFH;
	MOV	A, #22H;
	ORLD	P6, A;
	ORL	P2, #10H;
	RET;	

LIMXP:	CALL	HABXP;
	CER	A;
MAST:	MOVX	&R0, A;
	INC	A;
	JB4	\$ + 4;
	JMP	MAST;
	CALL	DESP;
	RET;	

## CONCLUSIONES.

Hasta aquí se ha presentado el diseño de un equipo que pruebe automáticamente las tarjetas de abonado de la Central CARMELITA.

Los diferentes circuitos que se presentaron se probaron aisladamente y funcionaron correctamente.

No me fue posible construir un Prototipo de este proyecto debido a que se me asignó otro trabajo más importante: la Industrialización de la Central CARMELITA.

Este nuevo trabajo involucra aspectos como: supervisión del diseño de tarjetas de circuito impreso, revisión del diseño electrónico, vigilar los aspectos de control de calidad de la Central, tener contacto con el cliente, vigilar que existan las componentes que se necesitan, controlar el diseño industrial, etc.

Sin embargo, dejo aquí este desarrollo teórico para que se le dé continuidad y así con la construcción de este equipo de prueba, se ahorre tiempo y recursos en la prueba de tarjetas de la Central CARMELITA la empresa TELMEX.

NOTA SOBRE DIAGRAMAS  
ELECTRICOS

Las alimentaciones eléctricas de algunos circuitos integrados no se incluyen en los diagramas para ayudar a la claridad de los mismos.

Cada circuito integrado debe llevar conectado entre sus terminales de alimentación un capacitor de cerámica de 0.1 microfaradio . Además, este capacitor deberá localizarse tan cerca del circuito integrado como sea posible.

Las resistencias cuya potencia no se especifica pueden tomarse como de 1/4 Watt.

Observaciones a algunos diagramas

Fig. 3.4

	Min	Típ	Máx	Unidades
$t_1$	---	345	675	nseg
$t_2$	---	330	625	nseg
$t_3$	700	---	---	nseg

Fig. 3.6

	Min	Típ	Máx	Unidades
$t_1$	0	---	---	nseg
$t_2$	---	320	600	nseg
$t_3$	600	---	---	nseg
$t_4$	600	---	---	nseg

Fig. 3.10

$t_1 = 1 \text{ seg.}$

$t_2 = 4 \text{ seg.}$

Figura del teclado y visualizador

R = 2.2 kohms

Fig. 4.6

C.I. 1 .....74LS02

R1 .....10 kohms

Fig. 4.7 (a)

R1	560 ohms (1/2 Watt)
R2	2.7 kohms (1/2 Watt)
R3	1 Mohm
R4	1 Mohm
Diodos	SK3100

Fig. 4.7 (b)

R1,R7	2.7 kohms
R2,R8	1 Mohm
R3,R9	10 Kohms
R4,R5,R6,R10,R11,R12	560 ohms
OAC1,OAC2	MCA255 (General Instruments)



Fig. 4.8

D1,D2	1N4148
R1,R16	5.6 kohms
R2,R17	15 Kohms
R3,R18	82 kohms
R4,R5	100 kohms
R6,R8	180 kohms
R7,R9	82 kohms
R10,R12	330 kohms
R11,R13	1 Mohm
R14,R15	10 kohms
C.I. 1	RC4136

Fig. del C.I. M751

R1	430 kohms $\pm$ 2%
R2	82 kohms $\pm$ 2 %
C1	820 pF $\pm$ 10 %
C2	120 pF $\pm$ 10 %
Xtal	$f_0 = 4.4336619$ MHz

Fig. del C.I. SSI 202

R1	1Mohm
Xtal	3.58 MHz

## Bibliografía

B. P. LATHI

Introducción a la teoría y Sistemas de Comunicación

Limusa

A. BRUCE CARLSON

Sistemas de Comunicación

Mc Graw-Hill

TELEFONOS DE MEXICO, S.A. (GERENCIA DE CAPACITACION)

Telefonía Elemental

JOHN E. HOPCROFT, JEFFREY D. ULLMAN

Introduction to Automata Theory, Languages and Computation

Addison Wesley

THOMAS BARTEE

Introduction to Computer Science

Mc Graw-Hill

FAIRCHILD

TTL Data Book

INTEL

Component Data Catalog

GENERAL INSTRUMENT

Catalog of Optoelectronic Products 1980

HEWLETT PACKARD

Cómo incrementar Calidad y Productividad con Equipos Electrónicos  
de Prueba

NATIONAL SEMICONDUCTOR CORPORATION

CMOS Databook

SCHILLING, BELOVE

Electronic Circuits: Discrete and Integrated

Mc Graw-Hill

INTEL

Manual del 8748

## GLOSARIO

**HARDWARE.** Se refiere a los elementos físicos que constituyen los circuitos:

a) Componentes que lo integran: Ya sea transistores, teclado, visualizador, circuitos integrados, optocopladores, etc.

b) Interconexión entre los mismos.

**SOFTWARE.** Con este nombre se designa al Programa, y a su filosofía, que controla la operación del circuito.

Es invisible al usuario.

Normalmente se almacena en una memoria ROM, que forma parte del circuito.