

UNIVERSIDAD NACIONAL DE INGENIERIA

Facultad de Ingeniería Eléctrica y Electrónica



INTERFAZ INTELIGENTE MICROPROCESADA ENTRE COMPUTADOR Y TELEX

T E S I S

Para optar el Título Profesional de
INGENIERO ELECTRONICO

VICTOR ENRIQUE MUÑOZ ROMAN

PROMOCION 1982 - 2

Lima . Perú
1987

Dedico éste trabajo
a mis padres:

VICTOR Y ANTONINA

Y tambien a mi esposa

ELENA.

TABLA DE CONTENIDO

	PAGINA
PROLOGO	
CAPITULO I: INTRODUCCION	3
1.1 Consideraciones Preliminares	3
1.2 Reseña histórica del servicio telex	8
1.3 El teleimpresor	9
1.4 Descripción básica del Sistema Dise ñado.	12
1.5 Esquemas	17
CAPITULO II: LA UNIDAD CENTRAL DEL SISTEMA	20
2.1. Generación de frecuencia de trabajo	20
2.2 Procesador Z-80 C.P.U.	20
2.3 Memoria Eprom	24
2.4 Memoria Dinámica	24
2.5 Generación de señales de control	25
2.6 Esquemas	31
2.7 Programa monitor	33
CAPITULO III: INTERCONEXION DE PROCESADOR A TECLADO HEXADECIMAL Y DISPLAYS	48
3.1 Funciones de teclado	48
Funciones del Display	
3.2 Circuito usado como interfaz para la conexión del teclado y displays	59
3.3 Interacción del procesador con el te- clado y display	63
3.4 Esquemas	

CAPITULO IV: INTERFAZ A COMPUTADOR O TERMINAL INTELIGENTE	78
4.1 Generalidades	78
4.2 Circuito usado como interfaz	83
4.3 Interacción del procesador con el computador o terminal inteligente	86
4.4 Esquemas	101
CAPITULO V: INTERFAZ A LINEA TELEX	103
5.1 Generalidades	103
5.2 Circuito usado como interfaz	107
5.3 Interacción del procesador con la línea Telex	108
5.4 Tabla del código Baudot	118
5.5 Esquemas	119
CAPITULO VI: PROGRAMACION DEL SISTEMA	122
6.1 Sistema operativo que permite interactuar al interfaz inteligente con un terminal inteligente	122
6.2 Sistema operativo que permite interactuar al interfaz inteligente con un computador	131
6.3 Sistema operativo parte I	135
6.4 Sistema operativo parte II	159
CAPITULO VII: RESULTADOS EXPERIMENTALES	182
CONCLUSIONES	187
BIBLIOGRAFIA	
APENDICE: CIRCUITO DEL PROCESADOR CENTRAL	

UNIVERSIDAD NACIONAL DE INGENIERIA

Facultad de Ingeniería Electrónica

"Interfaz Inteligente Microprocesada entre
Computador y Telex"

Autor : Víctor Enrique Muñoz Román

Proyecto para optar el Título de Ingeniero Electrónico

Lima 1986

EXTRACTO

Este trabajo tiene por objetivo desarrollar una interfaz que permita captar información de la línea Telex asignada a un abonado cualesquiera para ser transferida hacia un computador; también el computador puede enviar información proveniente de sus archivos almacenados en unidades de disco magnético o cinta magnética hacia cualquier abonado remoto.

Tal interfaz simula ser un teleimpresor (telex). Conectado por un lado a una línea Telex y por otro a un port de computador. La línea telex usada tiene la característica de trabajar a 2 (dos) hilos y corriente simple de 40mA. La línea conectada al port del computador esta constituida por 3 (tres) hilos, uno para la transmisión, otro para la recepción y el último usado como retorno común. Los niveles de tensión están de acuerdo a las normas E.I.A. además la transmisión entre computador e interfaz es a full-duplex.

La línea telex, en Lima, utiliza el código BAUDOT

para transmitir la información, cada caracter es representado por solo 5 (cinco) bits, mas 1 1/2 bits de STOP con una velocidad de transmisión de 50 Baudios half-duplex - (en transmisión telex se usan un total de 57 caracteres, notar que cinco caracteres solo permiten 32 combinaciones, mas adelante se explicará como es posible tal transmisión)

La línea a computador utiliza al código ASCII para transmitir la información, cada caracter esta representado por 7 bits mas 2 bits de STOP, la velocidad de transmisión puede ser seleccionable desde 1,800 a 9,600 baudios full-duplex. (notar que al control de la transmisión de caracteres no usa señales como data terminal ready - (DTR) ni request to send (RTS), sino la misma línea de recepción y transmisión).

En telefonía, a cada abonado se le asigna 6 (seis) digitos de identificación. En telegrafía a cada abonado se le asigna 5 (cinco) digitos, siendo así para cualquier abonado del Perú. La selección se realiza por disco marcador (39% de MARK y 61% de SPACE, al igual que en telefonía).

La interfaz es capaz de realizar la selección de un abonado cualesquiera, con solo enviarle los 5 (cinco) digitos necesarios por medio del teclado del computador. - Mostrando por pantalla el proceso del enlace (notará - posteriormente que no se usa ningún circuito integrado especial para generar la selección del abonado, sino que es realizado por programa propio del interfaz.

En lo que concierne a los modos de uso, el terminal de computador presenta tres modos; la primera nos permite dialogar y/o transmitir un documento, el segundo modo nos permite recepcionar automaticamente los documentos - que nos envíen y el tercer modo de uso nos brinda la posibilidad de editar un documento.

En el primer modo, desde el terminal de computador se puede establecer un intercambio de mensajes con cualquier abonado antes seleccionado (simulando así un teleimpresor), estando en este modo y presionando dos teclas de control podemos enviar un documento que tenga aproximadamente 57,344 caracteres. Para esto el computador nos pedirá primero el nombre de tal documento (que normalmente está almacenado en la unidad de disco). Luego el computador enviará tal documento al interfaz a gran velocidad (para los pruebas he transmitido a 4,800 baudios usando el protocolo Xon-Xoff), y entonces preguntará si Ud. desea mantener el enlace después que el interfaz haya terminado de enviar todo el documento, si su respuesta es si, el computador esperará que el interfaz termine la transferencia, de lo contrario si la respuesta suya fue no, el computador se libera del interfaz y esta último se dedicará solo a enviar el documento. La opción de liberar al computador mientras se transmite un documento, es la más conveniente, ya que la transmisión de un documento a una velocidad de transmisión de 50 baudios toma un tiempo más o menos notorio.

En el segundo modo, al ser seleccionado este modo el computador envía un carácter de control por su port de salida hacia el interfaz, este último al recibir tal carácter queda programado en el modo recepción automática, además queda liberado el computador, en este modo el interfaz almacena toda la información recibida en su memoria (hasta 57,344 caracteres). Si Ud. desea transferir todo el mensaje desde el interfaz hacia el computador, basta con enviar por el port del computador un carácter de control y luego el terminal de computador presentará su menú (lista de opciones o modos de uso) para que Ud. elija el modo de trabajo.

El tercer modo, es propio del computador, ya que para generar un documento se hace uso de un editor de tex-

tos el cual facilita el usuario redactar un documento, para luego ser enviado al interfaz para que este ultimo transmita al abonado remoto seleccionado.

Otra facilidad que nos proporciona este interfaz es la siguiente: si el computador estuviera siendo reparado, entonces se puede conectar un terminal de computador DIRECTAMENTE al interfaz utilizando la misma línea del computador. Para esto se tendrá que cambiar de posición un switch en el interfaz. En esta nueva opción de uso, Ud. podrá solo intercambiar mensajes con el abonado seleccionado, haciendo uso del teclado del terminal de computador.

El computador necesita de un programa sencillo (comunmente llamado paquete de comunicación) para poder interactuar con el interfaz. Este programa puede ser realizado en lenguajes de alto nivel, como BASIC o bajo nivel, como ASSEMBLER.

Para finalizar esta interfaz ha sido implementado y probado en el laboratorio de la compañía DIGICOMP S.A.. Los circuitos integrados estan colocados en sockets que están distribuidos sobre una tarjeta de propositos generales cableados por el lado de los componentes. Se usa una fuente de alimentación externa de tres salidas, + 5, + 12 y - 12 voltios.

PROLOGO

Este trabajo consta de dos partes bien marcadas, el diseño e implementación de procesador central y los circuitos que permiten interactuar con la línea telex y la línea del computador. Cabe mencionar que no se ha desarrollado ningún estudio teórico, ya que no era necesario porque el desarrollo a sido en su totalidad práctico, midiendo señales y niveles de corriente directamente en el aparato. Además este producto es totalmente nuevo en su concepción ya que se ha tratado de utilizar un mínimo de circuitos integrados, compensando tal ahorro incrementando el programa monitor (programa de control). El hecho que se use la mínima cantidad de circuitos integrados le da al aparato una mayor fiabilidad.

El procesador central inicialmente fue diseñado como un procesador de DESARROLLO con todas sus señales de control, bus de datos y bus de direcciones disponibles al diseñador. El procesador en uso es el Z-80A. Se puede acceder a cualquier localidad de su memoria dinámica de 64KBytes de capacidad por medio de un teclado hexadecimal y observar por 6 (seis) displays, cuatro asignados a las direcciones y dos a los datos, la representación en estos displays es mostrada en hexadecimal. Este procesador central le permite a Ud. aprender a programar los circuitos integrados programables como el SIO (interfaz serial asincrónica/sincrónica, es un circuito integrado de 40 pines, altamente programable), también puede practicar con el C.I. 1793 que es un controlador de drive de diskette y así con muchos más.

La segunda parte fue diseñada por la necesidad de almacenar en la unidad de memoria del computador los mensajes que llegaban vía TELEX y también el caso inverso. Se

puede entender facilmente que esta era verdaderamente una necesidad ya que se tenia que dedicar una persona a introducir al computador todos los mensajes recibidos por su teleimpresor y lo mismo en el sentido inverso. Pudiendo esta persona realizar un trabajo mas humano y así poder seguir desarrollandose como tal.

El circuito que permite interactuar con la linea telex y la linea del computador esta realizada en base a un SIO trabajando asincronamente ademas usa un circuito generador de baudios también programable, dos switch que sirven para cambiar las impedancias hacia la linea telex y permiten también enviar telegraficamente la información habriendo y cerrando un switch bajo control del procesador, el acoplamiento a la linea telex es por medio de un acoplador optico y a la linea del computador por medio de un integrado que traslada los niveles TTL a niveles E.I.A..

La interfaz inicialmente fue probada conectandola a un microcomputador, se hicieron mejoras, se analizaron las facilidades de operación y lo mas importante se mostro su confiabilidad.

Aprovecho esta oportunidad para agradecer a la Compañía DIGICOMP S.A. por haberme permitido el uso de su laboratorio y el uso de su línea telex.

También agradecer al Ingeniero Alberto Guillén C. por su desinteresada cooperación, a mis compañeros de trabajo del Area de INGENIERIA y de manera especial a mi compañero Luis Campos que trabaja en el área de Sistemas por haberme ayudado a desarrollar los programas que corresponden al computador. Y a todas aquellas que de una manera u otra contribuyeron para que este proyecto se lograra. Muchas Gracias.

CAPITULO I

INTRODUCCION

1.1 Consideraciones Preliminares

Analizemos la necesidad del uso de un computador: El hombre desde su inicio ha realizado actividades de todo tipo utilizando siempre su mente y su fuerza, a la vez; siendo utilizado uno de ellos en mayor o menor proporción

Aquí en nuestra sociedad se puede observar dos grupos bien marcados, el primero que podríamos llamar intelectual y que subsiste gracias a su capacidad mental y el segundo al que podríamos llamar grupo manual y que subsiste gracias a sus fuerzas físicas. Notándose con gran claridad en nuestro medio como el segundo grupo lucha constantemente por migrar hacia el grupo intelectual. Otra de las características de nuestro país en vía de desarrollo es que el grupo intelectual es minoritario, en tanto que el grupo manual es el que existe en mayor proporción.

En estos momentos si se toca el tema de computadoras, la mayoría de personas se entusiasman, les gusta, pero hasta allí nomas, porque PIENSAN que tal es solo para los intelectuales, es demasiado caro o desean hacer las cosas como lo hacian desde antes, se van a lo seguro, no arriesgan, pudiendo encontrar en el computador una gran ayuda hasta para el carpintero, bueno, todos debemos de ver en el computador una ayuda eficaz a nuestras necesidades y el Gobierno Central del Perú también debe de ver esta ayuda.

Veamos la necesidad de desarrollar la industria informática en el Perú bajo el apoyo del Gobierno Central.

El problema de mercado radica en el hecho de que naciones como el Perú, con una población muy pequeña y con

una población económica activa de aproximadamente 20%, no pueden absorber ni sustentar una industria sólida y auto-sostenida. Los productos de una industria dedicada únicamente a satisfacer las necesidades internas no podrán normalmente aprovechar las ventajas de la economía de escala ya que sus volúmenes de producción serán pequeñas y estarán orientados a un mercado pequeño, por lo tanto los precios serán muy altos y difícilmente podrán competir con los precios y calidades de productos importados producidos masivamente. Paralelamente la calidad de estos productos será baja, ya que las normas de control de calidad, para pequeños volúmenes de producción implica una elevación en el costo de los productos. Por lo tanto la industria tradicional, como dedicarnos a fabricar carros, televisores, etc. no nos saldria nada a cuenta, mucha es la ventaja que otros países desarrollados tienen con respecto a nosotros.

Ahora se nos presenta una oportunidad para salir del sub-desarrollo usando otro camino, que es el de desarrollar la industria informática. Primero para entender con mayor claridad a que nos referimos con desarrollo de la industria informática. Veamos, como el hombre que se diferencia de otros seres, por su inteligencia y creatividad, ha venido encontrando continuamente medios para delegar a terceros, ya sea animales o máquinas, tanto la realización de los esfuerzos físicos como la de esfuerzos mentales que deben realizar para sobrevivir. Se liberó de la realización de esfuerzos físicos domesticando animales y luego inventando máquinas que lo reemplazan con mucha ventaja en la tarea de cargar o trasladar pesos, En el aspecto intelectual inventa por ejemplo las matemáticas, y cuando sus cuentas crecen en volumen y complicación inventa máquinas que lo relevan del esfuerzo de calcular y recordar. Así la gran civilización china inventa y utiliza el ábaco, la civilización incaica utiliza el quipu, base de datos original peruana. Estos son los orígenes -

de la fascinante historia de los computadores. Estas veloces máquinas de calcular procesan datos, los almacenan y los transforman en información, es decir datos procesados de tal manera que ayudan en la tarea de tomar decisiones.

Vemos también que el costo de los equipos electrónicos cada vez se abaratan mas y mas lo cual hace posible que los países del tercer mundo tengan mayor posibilidad de adquirirlos. Las computadoras electrónicas constituyen inicialmente privilegio de una aristocracia poseedora de los capitales necesarios para adquirirlos. Las investigaciones, los logros de nuevas tecnologías y la gran demanda en el mercado mundial, dan como resultado no solamente la disminución del tamaño físico de las computadoras sino sobre todo la disminución de sus costos.

Entonces podemos notar claramente que la oportunidad se nos presenta, tenemos capacidad porque somos seres humanos, los costos de los computadores disminuyen (ya no necesitamos la tecnología para armar nuestros propios computadores), por lo tanto si el país con ayuda del gobierno central empiezan a desarrollar la Industria Informática podremos gozar de las facilidades y ayudas que nos brinda un computador.

Por ejemplo la administración en países como el nuestro ha generado el fenómeno de la burocracia, los computadores y los avances de la ingeniería de sistemas e informática, hacen que los sistemas administrativos apliquen conceptos de productividad, métodos y procedimientos. Las oficinas usarán cada vez más medios computarizados. El tradicional trabajo de la mecanógrafa, será simplificado por el empleo de procesadores de palabra, máquinas que ya existen en nuestro medio. Esto no significa desempleo sino un mejor empleo de la capacidad humana, porque cuanto mayor sea la eficacia en los procedimientos administrativos y en la producción, se generarán mayor número

y mayor número de puestos de trabajo, donde la labor principal consistirá en la realización de TAREAS CREATIVAS y de mayor productividad. Los sistemas administrativos podrán pues ser altamente mecanizados con un mínimo de mano de obra, la cual permitirá el pago de mejores sueldos, bajo costos de operación; mayores utilidades, es decir mayor rentabilidad de la empresa, mayor disponibilidad de capitales, los cuales con una legislación adecuada, significará a su vez mayores inversiones y mayores fuentes de trabajo bien remunerados.

Como segundo ejemplo tenemos beneficiada la Educación. El proceso educativo de nuestro país requiere desbordar las aulas, las universidades, los institutos que sólo pueden ser financiados y existir en las ciudades de mayor importancia. En cualquier aldea de la Selva, en cualquier puna de la Sierra, en cualquier poblado de la Costa, puede funcionar un radio, un televisor a batería y puede llegar por lo tanto al beneficio de la educación por el aire. A la vez las escuelas deberán educar ciudadanos que integran una sociedad informatizada por lo que deberán estar equipados con uno o mas computadores. No habrá excusa para no tenerlo, sus costos son cada vez más bajos, su mantenimiento cada vez mas sencillo, pero es también un instrumento indispensable para sentar las bases de la formación mental de nuestros futuros ciudadanos profesionales gobernantes, científicos, tecnológicos, políticos y muchos mas.

Las Universidades no deberian de marchar aisladamente. Un computador central en cada universidad enlazados entre si permitira transferir información entre sí y además una vez que Entel Perú brinde el uso de la red de transmisión de datos se podrán accesar bancos de datos de otras universidades del mundo.

Así la Industria Informática logrará que las dife-

rencias entre el grupo intelectual y manual desaparezcan. Ambos grupos podrian hacer uso de su inteligencia y PROGRAMAR los computadores para la realizaci3n de diferentes tareas, ya sea en el hogar, en la oficina, en la escuela, en el hospital, en el campo, es decir el computador ir3 haciendo posible la desaparici3n de la clase manual y la ir3 transformando en una clase que tambi3n vivirá de su intelecto, podrá ADQUIRIR CULTURA, lograra mayores ingresos y en consecuencia su calidad de vida mejorará aceleradamente.

Por lo tanto la industria informatica nos da la oportunidad de competir con los paises industrializados, por que para desarrollar PROGRAMAS, solo necesitamos lapiz y papel. Comp prueba de todo este cambio, Brasil es un ejemplo. La principal empresa brasileña de PROGRAMAS para micros comenzo a exportar a Estados Unidos de Norteamérica y se prepara para ingresar a los mercados de Australia, Jap3n y América Latina. Despues de conquistar en menos de cinco años, la posici3n lider del area de software en Brasil, la empresa Datal3gica Desarrollo Tecnol3gico S.A. Ltda., con sede en Sao Pablo, parti3 decidida rumbo a la disputa de una porci3n expresiva del mercado de los Estados Unidos. Aprovechando una laguna del mercado Norteamericano, aparentemente recen-tido por la ausencia de software empresarial, para alimentar mas de 600 mil microcomputadores Macintosh, de la Apple, la empresa brasileña abri3 una sucursal llamada Datalogica en el Valle del Silicio, California.(En este proyecto he minimizado el hardware y en consecuencia se ha incrementado el software de control).

Adem3s, si alguna vez una máquina nos reemplaza de nuestra actividad rutinaria, no deberia ser motivo de tristeza ni de preocupaci3n, al contrario, nos deberíamos alegrar ya que nos estaria dando la posibilidad de conocer otras especialidades y dejar la rutina.

1.2 Reseña histórica

La historia del servicio telex se remonta al año - 1887, cuando la American Telephone and Telegraph Company construye una línea privada entre New York y Philadelphia. Los equipos terminales del circuito estaban dispuestos de tal manera que el usuario, en ese entonces un cambista, - pudiera conmutar de telegrafía a voz, según sus necesida des, estableciéndose así por primera vez el servicio con mutable alternadamente de telegrafía-voz.

Para atender estos equipos se necesitaban técnicos especiales en pruebas y regulación de líneas Telegráficas. En Telegrafía, a diferencia de la telefonía, no se llegó a separar las funciones de operación de las funciones de prueba de los circuitos, lo que requería trabajo altamen te especializado.

La aparición en 1917 de máquinas Teleimpresoras de arranque y parada (START-STOP), que operaban con un códi go de 5 unidades, dio impulso a la conmutación de canales telegráficos de teleimpresión, puesto que, por prime ra vez, todas las estaciones de la red podían recibir con precisión bajo condiciones cambiantes de línea. Asimismo, todas las estaciones podían transmitir o recibir con el mismo tiempo de retorno y la estación receptora podía in terrumpir a la estación transmisora.

Otras técnicas nuevas fueron introducidas en este pe ríodo, por ejemplo: el teleimpresor de cinta, que impr imia el texto del mensaje en una tira angosta de papel - que después se pegaba sobre el formato respectivo.

En 1931 la American Telephone and Telegraph Co. in trodujo un nuevo servicio de conmutación manual de circui to telegráficos, que permitía a sus abonados el uso de - circuitos telegráficos en períodos cortos de tiempo, a se mejanza del servicio de línea privada arrendada que nor malmente se usaba para tráfico de gran volumen, con la di

ferencia trascendental que el usuario podía llegar a cualquier abonado del sistema. Así nace el servicio público de teleimpresión que más tarde tomó el nombre de TELEX. a partir de 1962, el servicio manual se convirtió en automático.

El término TELEX, proviene del inglés "TELEGRAPH EXCHANGE". En los Estados Unidos se le conoce con las siglas de TWX, forma abreviada de Teletypewrite Exchange. En Europa, con la denominación de GENTEX, "GENERAL TELEGRAPH EXCHANGE"

En el Perú el servicio telex data del año 1959, año en que All America Cables And Radio Inc. empieza a operar una central manual, Cable West Coast hace lo mismo e inicia la operación de un tablero manual de 30 líneas de capacidad. Posteriormente, en 1970, ambas empresas inauguran un servicio automático; finalmente, en 1975, ENTEL PERU asume totalmente dicho servicio.

1.3. El Teleimpresor

Para empesar damos el significado la palabra TELEIMPRESION que se deriva del griego TELE que significa 'a distancia y del IMPRESSI, que significa "impresión", o sea, impresión a distancia. Una definición más amplia define al teleimpresor como un mecanismo que escribe directamente caracteres tipográficos correspondientes a las series de impulsos que recibe.

El teleimpresor electrónico, modelo 1000, es otro de los aparatos terminales de abonado que utiliza ENTEL-PERU en la red nacional del servicio telex, en el cuál todos las funciones lógicas están a cargo de circuitos electrónicos y el accionamiento mecánico se aplica sólo a aquellas funciones que por naturaleza todavía no pueden realizarse electrónicamente.

Algunas Características importantes del
TELEIMPRESOR 1000 de la Compañía
SIEMENS AKTIENGESELLSCHAFT

Alfabeto telegráfico	Alfabeto telegráfico internacional N° 2 (CCITT) y otros códigos de 5 bits.
Velocidad telegráfica	50 baudios, conmutable discrecionalmente a 75 y 100 baudios.
Tensión y frecuencia de la red.	187 a 264V; 40 a 70HZ o bien. 93.5 a 140V; 40 a 70HZ.
Consumo	En estado de reposo 40W en estado de impresión 90/120W
Mando de impresión	El texto transmitido se imprime en rojo y el texto recibido se imprime en negro.
Transmisor de indicativo	Hasta 20 signos elegibles discrecionalmente.
Niveles de ruido sin/con equipos adicionales a 100 baudios (DIN 45635)	$\frac{51}{53}$ db
Posición de servicio	Cualquier inclinación hasta de 25°
Condiciones ambientales de Servicio.	0 a 50°C, humedad relativa hasta 90%

Algunas funciones del teclado

- a) Tecla de llamada al colateral: Al pulsar esta tecla se acciona un timbre en el teleimpresor del colateral y en el suyo propio.
- b) Tecla de repetición: Mientras presiona esta tecla, su teleimpresor repite el último signo o la última función pulsada.
- c) Tecla de retorno de carro: pulsando esta tecla retorna el carro sobre el mismo reglón de uso.

- d) Tecla cambio de Línea: Pulsando esta tecla se cambia a la siguiente línea, sin mover el carro de su posición física.
- e) Tecla de nuevo reglón: Al presionar esta tecla se realiza un retorno de carro seguido de un cambio de línea.
- f) Tecla de espacio: Pulsando esta tecla se separa una palabra de la otra.
- g) Tecla de identificación del colateral: Al pulsar esta tecla el teleimpresor del colateral transmite automáticamente su indicativo; de esta manera puede informarse quién es su colateral.
- h) Tecla de identificación propia: Pulsando esta tecla su teleimpresor transmite en forma automática su propio indicativo; así puede identificarse cuando efectúe una llamada.
- i) Tecla de cambio a letras/a cifras: Puede suceder que cuando su teleimpresor está recibiendo un mensaje, en el texto aparezcan sólo cifras y símbolos; en este caso, pulse esta tecla y su máquina empesará a escribir correctamente el mensaje.

Algunas abreviaturas mas usadas en el servicio telex y que tienen vigencia en cualquier país son:

<u>Abreviatura</u>	<u>Significado</u>
ABS	Teleimpresor desconectado
BK	Corte. Suspenda la transmisión.
CFM	Confirmar
COL	Repetición de datos importantes del mensaje recibido.
DER	Fuera de servicio por avería.
DF	Esta conectado con el número solicitado.
EEE	Error

GA	Puede transmitir.
MINS	Minutos
MOM	Espere un momento
MUT	Mutilación.
NA	No hay conexión con este número.
NC	Ningún circuito libre
NCH	Número cambiado
NP	El número solicitado ya no es abonado.
DCC	Abonado ocupado
OK	De acuerdo
RAP	Volveré a llamar
RPT	Repita por favor
THRU	Esta en conexión con ...
WRU	¿Quién contesta?
+?	Ha terminado de transmitir. Puede usted empezar.

1.4 Descripción Básica del sistema diseñado

Esta interfaz consta de cuatro módulos, el primero es el procesador central, controlado por un microprocesador Z-80, que maneje 72KByte de memoria; el segundo módulo es el teclado y displays que permiten el acceso a TODAS las localidades de la memoria dinámica de 64KBytes, además sus teclas de control facilitan el cargar programas de prueba en diseños de cualquier tipo de interfaces; el tercer módulo lo conforma el circuito que permite interactuar al procesador central con un computador o terminal de computador, dependiendo cual fuera la modalidad de trabajo; como último módulo tenemos al circuito que interactúa con el procesador central y la línea telex. La figura 1.1 muestra los dos modalidades de uso de la interfaz y la figura 1.2 muestra el diagrama de bloques de la interfaz.

1.4.1 Procesador central

Utiliza como unidad central de procesos el micropro

cesador Z-80A. Se utiliza un cristal de 13.33MHz de frecuencia que dividido entre cuatro por un circuito integrado de Código 74LS93 nos da la frecuencia de trabajo que es 3.3325MHz.

Utilizo un C.I. 74LS74 para generar dos señales MUX y CAS que me permitirán controlar las memorias dinámicas, estas señales son consecuencia de haber ingresado al C.I. 74LS74 las señales RFSH, MREQ y el clock de 13.33MHz de frecuencia. Se utiliza dos tipos de memoria, la primera es una memoria EPROM de código 2764 de capacidad 8K Bytes (se escoge esta capacidad por ser este un equipo de desarrollo), el segundo tipo de memorias es un conjunto de 8 (ocho) circuitos integrados de 64Kbits cada uno que dan un total de 64KBytes (memorias RAM dinamicas). Como el microprocesador Z-80A solo puede direccionar un máximo de 64KBytes y tenemos un total de 72KBytes agrupamos a las memorias en dos grupos llamandole banco de memoria a cada grupo. Para seleccionar los bancos de memorias utilizamos otro C.I. 74LS74 mas unas instrucciones de control. Mediante una lógica combinacional se obtiene la señal de control a memoria RAS, las memorias dinamicas son de código 4164 que aceptan el bus de direcciones de 16 bits en dos grupos de 8bits cada uno. Para lo cual se utiliza un multiplexor de código 74LS157. Como último utilizo un C.I. de código 74LS138 que me permite seleccionar los dispositivos que serán controlados por el procesador.

1.4.2 Teclado y Display

El teclado consta de 24 (veinte y cuatro) teclas distribuidas en 6 (seis) columnas y 4 filas de las cuales 6 (seis) teclas sirven para el control y 16 (dieciseis) para ingresar datos o alterar direcciones en hexadecimal, quedando 2 (dos) teclas para usos futuros.

La tecla RUN, permite que corra el progra-

ma a partir de la dirección que este indicando el display. Por lo tanto se puede correr un programa a partir de cualquier dirección (esto es muy útil para el diseño de controladores).

La tecla DEC, permite decrementar en una unidad la dirección mostrada en el display, como consecuencia se muestra automáticamente el dato correspondiente a esa nueva dirección en los dos display restantes.

La tecla INC, incrementa en una unidad la dirección mostrada en el display.

La tecla DIR, permite ingresar el modo de selección de una localidad de memoria ya sea para observar que dato hay en ella o alterar tal dato. Trabaja de la siguiente manera: una vez presionada la tecla se enciende intermitentemente un led junto al display que va a ser alterado si vuelve presionar la tecla dir el led del display contiguo hacia la derecha comenzará a encenderse intermitentemente, si se sigue presionando la tecla el led del siguiente display encenderá indicando que ya se puede alterar su valor y así sucesivamente hasta llegar al cuarto display, si sigue presionando, se volverá al primer display y todo el ciclo se repite. Para alterar el valor del display seleccionado bastará con presionar cualquiera de las dieciseis teclas en hexadecimal, y automáticamente el led del display contiguo hacia la derecha comenzará a encenderse intermitentemente. Para salir de esta opción. presione cualquiera de las teclas de la última fila del teclado.

La tecla DAT, permite alterar el dato de la localidad de dirección mostrada en el display, su uso es semejante al uso de la tecla DIR.

Las teclas de datos están numeradas desde 0 hasta F en hexadecimal.

Los displays me permiten observar el dato y la dirección de cualquier localidad de la memoria RAM de 64KBytes. Cuatro displays para las direcciones y dos displays para los datos, ambos en hexadecimal.

Cada display consta de siete segmentos mas un led indicador. Los siete segmentos permiten formar los número en hexadecimal.

1.4.3 Circuito interfaz, procesador central y línea del Computador

Se utiliza un C.I. SIO (serial in/out, ver indice para ver especificaciones) solo uno de sus ports de I/O es utilizado, el otro lo utilizara la línea telex. Para generar los niveles E.I.A. (-25 hasta -3 voltios para el cero lógico y desde 3 hasta 25 voltios para el uno lógico) se utiliza un C.I. de código MC1488 al cual le ingresa un nivel T.T.L. y sale un nivel E.I.A., para la recepción se utiliza al MC1489 que recibe un nivel E.I.A. y se obtiene un nivel T.T.L., ambos trabajan como inversores. Para generar la velocidad de transmisión en baudios se utiliza un C.I. programable a obtener hasta 16 velocidades distintas. El programa monitor selecciona la velocidad de uso, como generador de frecuencia base utiliza un Cristal de 5.0688MHz.

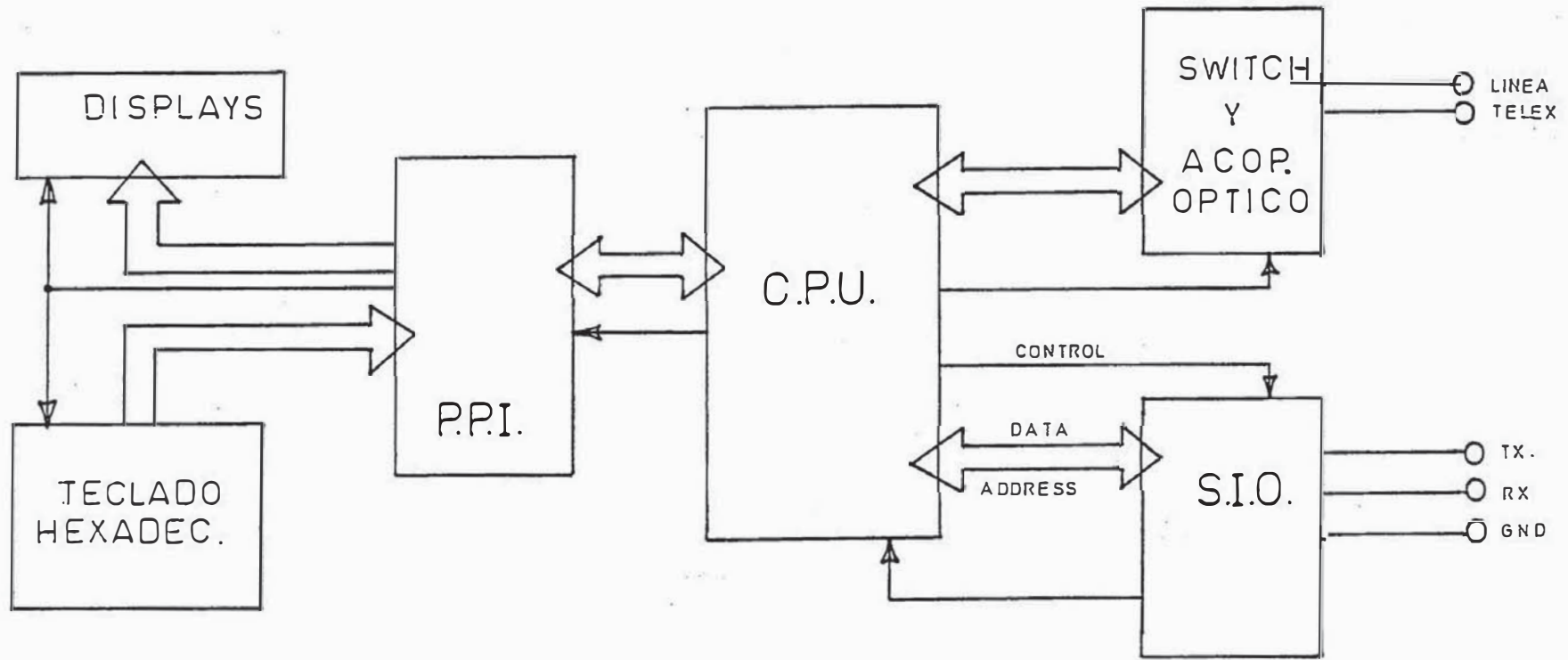
1.4.4 Circuito interfaz, procesador central y línea telex

Se utiliza el port del SIO que queda libre. Para conseguir los niveles de corriente SMA en reposo y 40mA en uso, se utilizan dos cargas de 22K Ω y 220 Ω cada uno - en serie con un switch controlado por el procesador central.

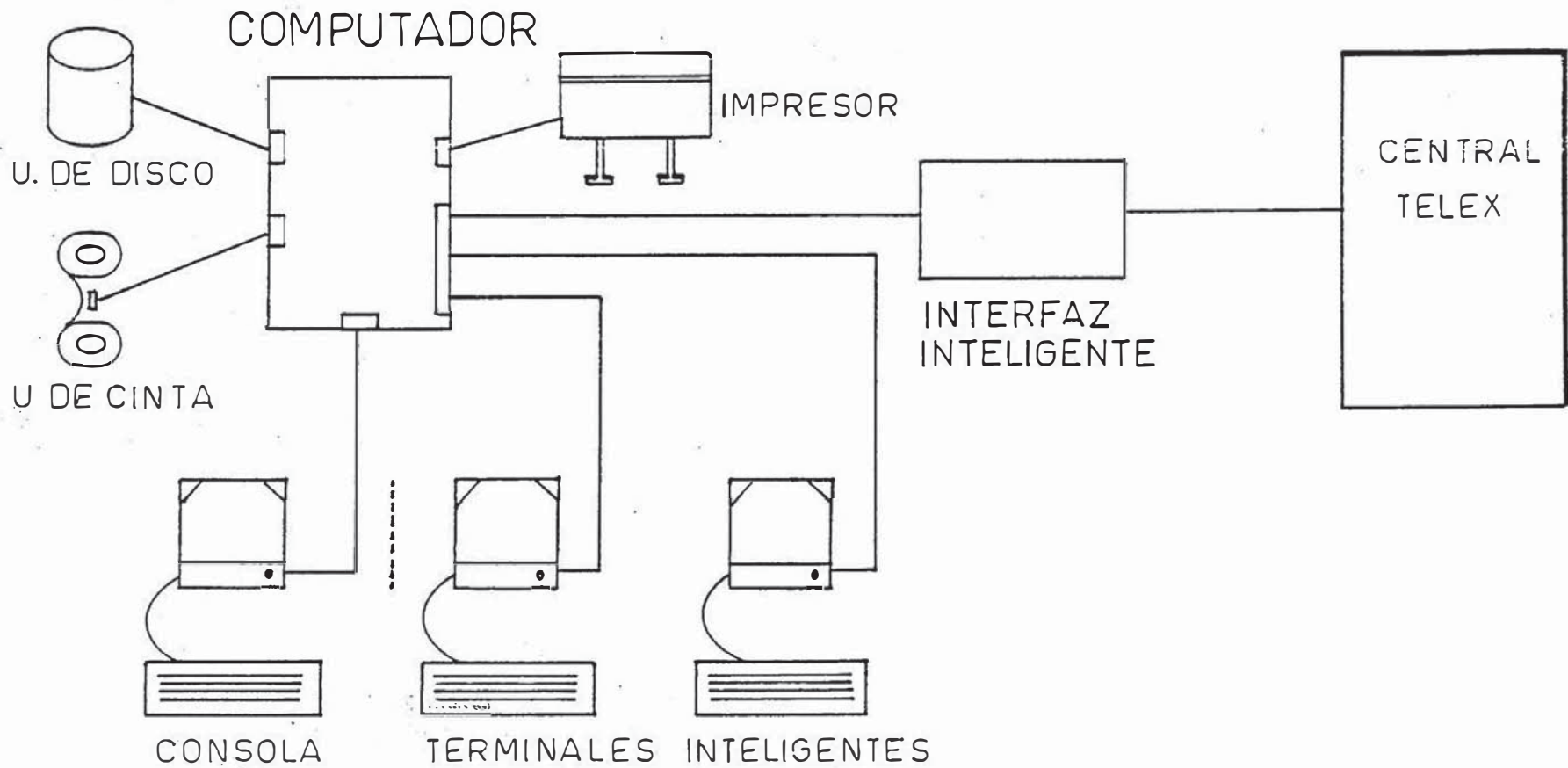
La tensión que aparece en la línea telex proveniente de la central es de 120 voltios en vacio y 110 volt. con carga. El Switch en serie a la carga de 220 Ω permite realizar la llamada a un abonado cualquiera siendo este controlado por el procesador, también este switch permite

enviar los mensajes a una velocidad de 50 baudios. En serie a este circuito sencillo de switch se ha instalado un circuito acoplador optico que esta encapsulado en un circuito integrado de 6 (seis) pines con código MCT-2E, que permite tomar los mensajes que llegan al interfaz.

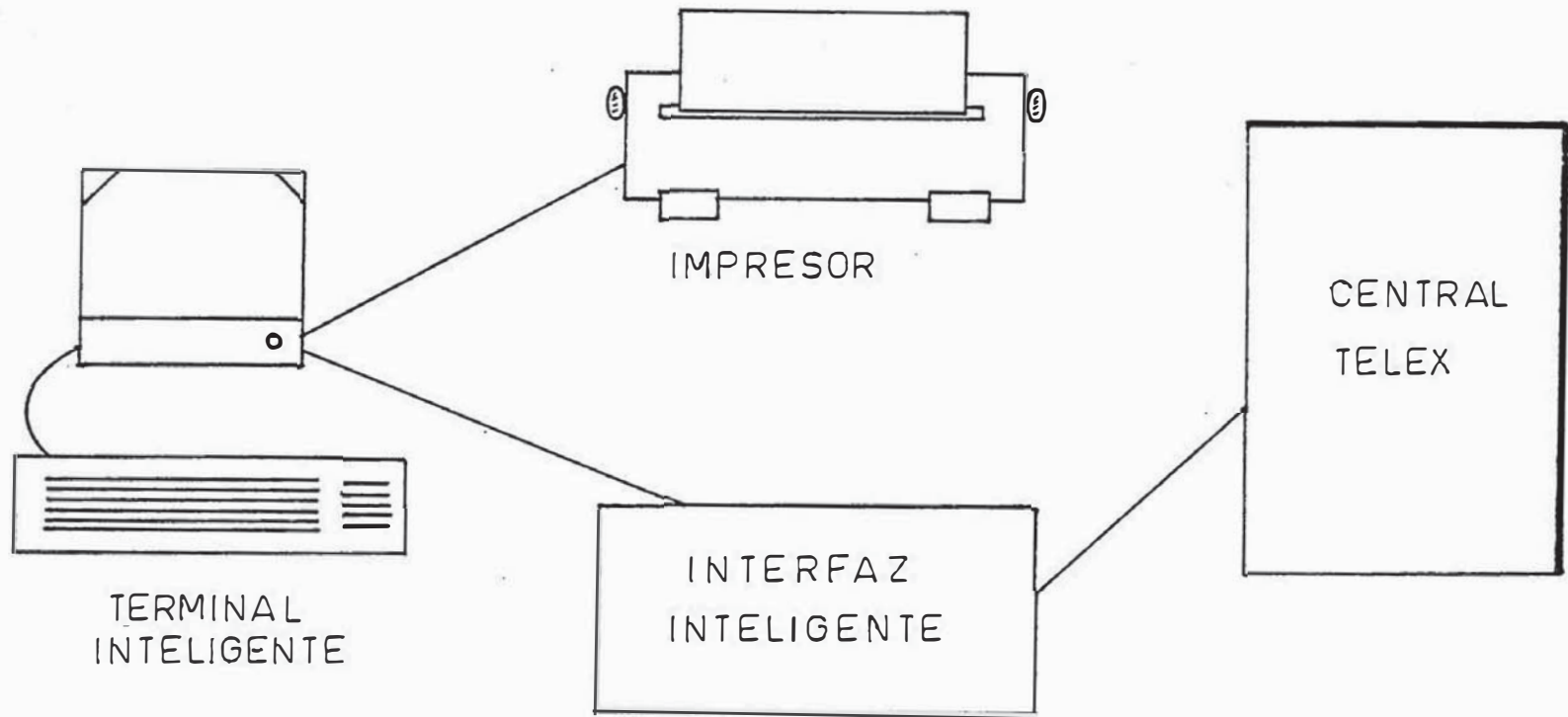
INTERFAZ - INTELIGENTE



PRIMERA - CONFIGURACION



SEGUNDA CONFIGURACION



CAPITULO II

LA UNIDAD CENTRAL DEL SISTEMA

2.1 Generación de Frecuencia de Trabajo

Se utiliza un generador de frecuencia que usa un cristal interno, este generador tiene tres pines útiles, uno para ser alimentado, otro conectado a tierra y el tercero nos brinda la señal de 13.33MHz esta señal es inyectada a un divisor entre cuatro que nos da una frecuencia de trabajo de 3.3325 MHz, que es utilizado por el Z-80CPU y por el Z-80SI0 ($\emptyset=3.325\text{MHz}$).

También veremos después que la frecuencia fundamental de 13.33MHz es utilizada para generar las señales de control MUX y CAS para la memoria dinamica.

2.2 Procesador Z-80CPU

La literatura al respecto es abundante, por lo que no tratare de explicar lo que en muchos libros tratan en detalle, (ver bibliografía). Para mi personalmente, este procesador es el que mayor se adapta para realizar un microcomputador de desarrollo de "8 (ocho) bits" por su "variedad de instrucciones" lo cual hace de este, un microcomputador muy versátil adaptándose así al diseño de controladores de diversos usos. Por ejemplo: usa tres modos de interrupción enmascarada (estos tipos de interrupción pueden ser seleccionados y habilitados por programa, como también deshabilitados) conocidos como MODO 0, MODO 1, MODO 2; siendo el mas útil y poderoso el modo 2 el cual permite intervectuar con 128 dispositivos distintos. Este modo usa el direccionamiento vectorizado el cual es ensamblado por 2 bytes uno grabado en el registro I del microprocesador y el otro lo da el dispositivo (sirviendo este último byte como código de dispositivo.

- Existen dos clases básicas de procesamiento de las in

terrupciones en el Z-80 que se utilizan en el diseño tratado.

- a) Procesamiento bajo el control del software.
- b) Procesamiento controlado mediante hardware y software.

He aquí una definición general:

- a) Procesamiento bajo el control del software.

Procesamiento en el cual todas las funciones entre las tareas se producen totalmente bajo el control del software. Como ejemplos de transiciones controladas por el software tenemos las llamadas a subrutina y los retornos, y saltos. Una técnica corrientemente usada que ayude en las transiciones controladas por software, es el sistema de muestreo.

- En este diseño es así como se interactúa con el teclado hexadecimal y los displays, realizándose un muestreo de alta velocidad.

- b) Procesamiento controlado mediante hardware y software.

Procesamiento en el cual las transiciones entre las tareas se producen bajo el control del software o del hardware. Como ejemplos de las transiciones controladas por hardware tenemos las interrupciones y también las transferencias por acceso directo a memoria (DMA). Dos circuitos periféricos del Z-80, llamados SIO y DMA, están diseñados específicamente para las aplicaciones de interrupción y DMA, respectivamente.

- También en este diseño son usados las interrupciones - por medio del C.I. SIO, el terminal de computador puede acceder al microcomputador (interfaz inteligente) enviando caracteres produciendo una interrupción de recepción a la tarea que esté realizando el microcomputador, cada vez que el terminal de computador envía un carácter; también se producen interrupciones de transmisión cada vez que el microcomputador necesita enviar una serie de caracteres, que se mostrara como un mensaje en la pantalla del

terminal.

El tema de muestreo e interrupciones es un tema muy poco tratado en la mayoría de libros de electrónica, por lo que voy a dar algunas definiciones y un ejemplo de cada caso.

Muestreo: El muestreo se refiere a una técnica de software utilizada para determinar si un dispositivo externo necesita servicio. El dispositivo externo está unido a la C.P.U. mediante una vía de comunicación que puede ser examinada periódicamente para detectar señales que indiquen que se necesita servicio.

Interrupción: Una interrupción es una señal que es introducida en la C.P.U. para indicar que un dispositivo externo necesita servicio, es decir, los recursos del computador. Típicamente, durante la ejecución de cada instrucción la C.P.U. muestrea su entrada de interrupción para determinar si es activa. Si es así, se termina la instrucción actual antes de que empiece el servicio a la interrupción.

Acceso directo a la memoria (DMA): Esta técnica es utilizada para transferir grandes bloques de datos entre la C.P.U. y el dispositivo periférico. Después de que la C.P.U. la inicia, la transferencia está controlada totalmente por una circuitería especial. El resultado es que posiblemente no se necesitan los recursos de la C.P.U. para efectuar la transferencia. Sin embargo, puesto que los buses de dirección, de datos y de control, participan en la transferencia de datos, a menudo la C.P.U. debe compartirlos con el dispositivo de DMA. Este proceso de compartir los recursos degrada la eficacia.

Ejemplo de muestreo: Si Ud. tuviera que admitir llamadas telefónicas utilizando la técnica de muestreo, usted debería tomar periódicamente el receptor telefónico y preguntar "¿Hay alguien allí?"

Ejemplo de interrupción: Si Ud. tuviera que admitir llamadas telefónicas utilizando la técnica de procesamiento de interrupciones, usted debería aguardar hasta que oyera

el timbre del teléfono y tomar entonces el receptor. Ejemplo DMA: Si Ud. tuviera que aceptar las llamadas telefónicas utilizando la técnica de acceso directo a memoria, usted guardaría los mensajes telefónicos en una cinta magnética y accedería a ellos siempre que lo necesitara. Observe que esto no incluye un intercambio altamente interactivo de mensajes cortos entre usted y el que efectúa la llamada. El método DMA no está diseñado para atender este tipo de tratamiento.

En los ejemplos previos del teléfono, usted puede inmediatamente discernir varias ventajas de procesado de las interrupciones con respecto al muestreo. La mayor desventaja del sistema de muestreo es que utiliza recursos aunque no se necesite ningún servicio. Otra desventaja es que el tiempo de respuesta del servicio puede ser relativamente largo. Por ejemplo, si en el ejemplo del teléfono, usted muestrea su línea de teléfono una vez cada hora entonces es posible que la persona que llame deba aguardar 59 minutos antes de poder hablar con usted.

El procesado de las interrupciones tiene varias ventajas importantes con respecto al muestreo. En primer lugar, la repartición de los recursos está basado estrictamente en las necesidades. Los recursos no se gastan en dispositivos que no necesitan servicio. El servicio puede empesar inmediatamente después de que se ha detectado la necesidad, de forma que el tiempo de respuesta puede ser mas rápido, sin la sobrecarga incluida en el muestreo frecuente. La desventaja del procesamiento de las interrupciones es que puede incrementar tremendamente la complejidad del software necesario para soportarlo. Puesto que la interrupciones son acontecimientos que pueden potencialmente, interrumpir un programa en cualquier etapa de su ejecución, algunas veces se necesita una lógica adicional de software para asegurar que el programa pueda resumir la ejecución despues de que pasó el servicio de interrupción. Además esta aleatoriedad puede hacer que los defectos intermitentes del software sean muy dí-

fáciles de reproducir, aislar y reparar. Una desventaja adicional es que el servicio de las interrupciones requiere a menudo un hardware externo bastante caro que se utiliza para dar prioridades e identificar los dispositivos interruptivos.

En este caso el SIO es un circuito especializado - que puede trabajar por muestreo o por interrupciones, dependiendo de su programación. Por lo que no necesita circuitería externa.

2.3 Memoria Eprom

Se utiliza la memoria de código 2764 tiene una capacidad de 8kbits (1byte consta de 8 (ocho) bits), con un tiempo de acceso a la data de 300nseg. Recordemos que la letra E significa erasable, traducido al castellano, borrable; mejor dicho, se puede grabar en la memoria y luego borrarla usando luz ultravioleta. Es una memoria no volátil.

En esta memoria esta grabada basicamente el programa monitor. Este programa monitor solo usa aproximadamente 1.3Kbytes quedando todo el resto de memoria para grabar programas de desarrollo (aproximadamente 6.7Kbytes).

Esta memoria, es considerada por el microprocesador, como un banco (banco de memoria 1).

2.4 Memoria dinamica

Se utiliza la memoria de código 4164, tiene una capacidad de 64Kbits, por lo que para formar 64Kbytes, necesitamos 8 (ocho) de estos circuitos, y así tendremos 64Kbytes en memoria dinamica RAM (memoria de acceso aleatorio) volátil. Esta memoria usa tres señales de control RAS, CAS, y WE (notar que no usa una entrada especial para el refresco RFSH), además tiene que direccionar 64K bits para lo cual necesitaria 16 bit de direcciones, osea

16 pines solo para las direcciones y este chip solo tiene 16 pines y nace la pregunta ¿como puede direccionar tantas celdas de memoria?, bueno este integrado usa la técnica del multiplexado de direcciones y asi solo necesita 8 pines para las direcciones. Veamos como lo hace: ver figura 2.1, una vez generada la señal RAS, la memoria toma los 8 bits menos significativos del bus de direcciones, luego se genera la señal MUX que presenta los 8 bits mas significativos del bus de direcciones a la memoria. Y cuando se genera la señal CAS, la memoria toma estos últimos 8 bits, completandose asi los 16 bits de direcciones. Tambien sabemos que toda memoria dinamica necesita refrescar sus datos almacenados, esto se realiza por medio del pin destinado al RAS. Notar que el RAS para refresco es generado por la combinación de las señales RFSH y MREQ (provenientes del microprocesador Z-80), encambio el RAS usado para la transferencia de datos es generado por la señal MREQ.

El pin de escritura (WE) esta conectado directamente al uprocesador Z-80.

Para multiplexar los bit de direccionamiento se usan dos C. integrados de código 74LS157 estos circuitos multiplexan a la orden de la señal MUX.

2.5 Generación de señales de control

(Ver circuito general y diagrama de tiempo)

Para generar las señales MUX y CAS se utiliza un C. I. 74LS74 FLIP-FLOP que procesa las señales RFSH, MREQ y la frecuencia fundamental.

En el diagrama de tiempos se muestra un ciclo M1 (busqueda del código de operación). Como se puede ver en el diagrama en los dos primeros ciclos T siempre existiran las señales MUX y CAS, esto debido a que la señal RFSH esta en uno LOGICO y una vez que MREQ habilita el

FLIP-FLOP TIPO D la señal MUX aparecera después de un tiempo $P/2$ (donde P es el periodo de la frecuencia fundamental) y luego aparecerá la señal CAS después de un tiempo P. En los ciclos T tres y cuatro estas señales desaparecen primero porque MREQ al deshabilitar el FLIP-FLOP coloca estas señales a cero y luego porque la señal RFSH en estos dos ciclos se coloca a cero.

Para tener mas claro el concepto de ciclo veamos lo siguiente: un ciclo M1 esta formado por cuatro ciclos de frecuencia de trabajo \emptyset cada ciclo es llamado ciclo.

Un arreglo lógico permite generar la señal RAS. Primero cuando el selector de bancos de memoria activa el banco 1 (memoria Eprom) y el bit A15 del bus de direcciones esta en cero la señal RAS es generado por la señal RFSH y MREQ solo en los dos ciclos T últimos, destinados al refresco de memoria. Esto es lógico que suceda ya que se esta trabajando con el banco 1. Además notar que en estos dos últimos ciclos T las señales MUX y CAS están deshabilitadas por lo que no hay ningún acceso externo a la memoria dinamica, solo el refresh esta activo. En los dos primeros ciclos T la señal RAS también esta deshabilitada por las señales RFSH, BANK y A15.

Estando activo el banco 1 y el bit A15 activo (uno lógico) se puede accesar simultaneamente a la memoria EPROM y a la mitad superior de la memoria dinamica (8000H -FFFFH) esto es permitido ya que no hay sobreposición de localidades de memoria y también como se verá esta es una forma de pasar la información de la memoria EPROM a la memoria dinamica.

Si el selector de Bancos activa la memorias dinamicas en su plenitud (BANCO 2) la lógica combinacional permite que la señal RAS sea un fiel reflejo de la señal MREQ y por lo tanto la señal RAS aparecera esta vez en los dos primeros ciclos T.

La señal EPROMCE esta generada por la señal BANK y el bit mas significativo del bus de direcciones. Si la señal BANK tiene un nivel uno lógico (BANCO 1 habilitado) y el bit A15 tiene un nivel cero lógico (direcciones bajas habilitadas) entonces la señal EPROMCE habilitara la memoria EPROM. Si el bit A15 esta a uno lógico (seleccionando direcciones altas, 32K superiores) entonces deshabilitará la memoria EPROM. Otra posibilidad es que la señal BANK tenga un nivel 0 (BANCO 2 habilitado) lo cual hara que la señal EPROMCE deshabilite la memoria EPROM en todo momento, sin importar cual sea el estado del bit mas significativo A15.

Otra señal generada y util es MREQ-RD, generada por dos señales MREQ y RD. Esta señal habilita la salida de la memoria EPROM (salida TRI-STADO) permitiendo asi poder leer la información almacenada en sus celdas.

Las dos señales antes mencionadas EPROMCE y MREQ-RD generan otra señal útil, la señal OE (OUT-PUT enable) que habilita un conjunto de 8 (ocho) circuitos TRI-STADOS instalados cada uno a cada salida de datos de las memorias RAM dinamicas.

Se utiliza un C.I. de código 74LS138 usado como decodificador (3 entradas y 8 salidas de señal).

Para entender claramente, como se usa este C.I. en este diseño, explicaré brevemente la instrucción de salida OUT(n), A:

Esta instrucción esta diseñada para sacar por el bus de datos el valor del acumulador, hacia el dispositivo externo cuyo código de dispositivo ha sido denominado n. El código hexadecimal para esta instrucción es D3n, donde n es el código de dispositivo formado por dos digitos hexadecimal. Al ejecutar esta instrucción el C.P.U. realiza

las siguientes operaciones:

- a) Coloca el código del dispositivo n en los ocho bits de menor peso del bus de direcciones (A0-A7).

Coloca el contenido del acumulador en los ocho bits de mayor peso del bus de direcciones (A8-A15).

- b) Activa las señales IORQ y WR.
- c) Coloca el contenido del acumulador en el bus de datos y es colocado en el dispositivo de código de dispositivo n.

El uso de este circuito en el presente diseño nos permite usar la combinación software y hardware para seleccionar un dispositivo cualquiera.

Los términos software y hardware se usan y se usaron en el presente tratado por lo tanto he aquí sus definiciones simplificadas.

Hardware: Los dispositivos mecánicos, magnéticos, e electrónicos a partir de las cuales se fabrica un computador; el ensamblado de materiales que constituyen un computador.

Software: La totalidad de programas y rutinas utilizadas para ampliar las habilidades de los computadores, tales como compiladores, rutinas y subrutinas.

Tal circuito usa los bits A2, A3, A4, A7 IOREQ y M1R como entradas. El circuito es activo si la señal IOREQ está a cero lógico, M1R está a uno lógico y el bit A7 del bus de direcciones está a cero lógico, es decir cuando se produce un requerimiento de entrada o salida (recordar que un reconocimiento del microprocesador a una interrupción, se manifiesta poniendo a cero lógico las señales IOREQ y M1R). Los valores lógicos en las líneas de direc

ción desde A2 hasta A3 determinan cual de las ocho líneas de salida está activada. El circuito integrado 74LS138 - es un decodificador/demultiplexor con lógica negativa, lo que significa que sus ocho salidas están normalmente altas y que son activas a nivel bajo (cero lógico). Notar algo importante, los bits A0, A1, A5, A6 no son usados por lo tanto varios códigos de dispositivo pueden direccionar un dispositivo, así, si deseamos direccionar el primer dispositivo conectado a la línea 01, tenemos que 04,05,06,07,24,25,26,27,40,...66,67 como valores de n activan tal dispositivo.

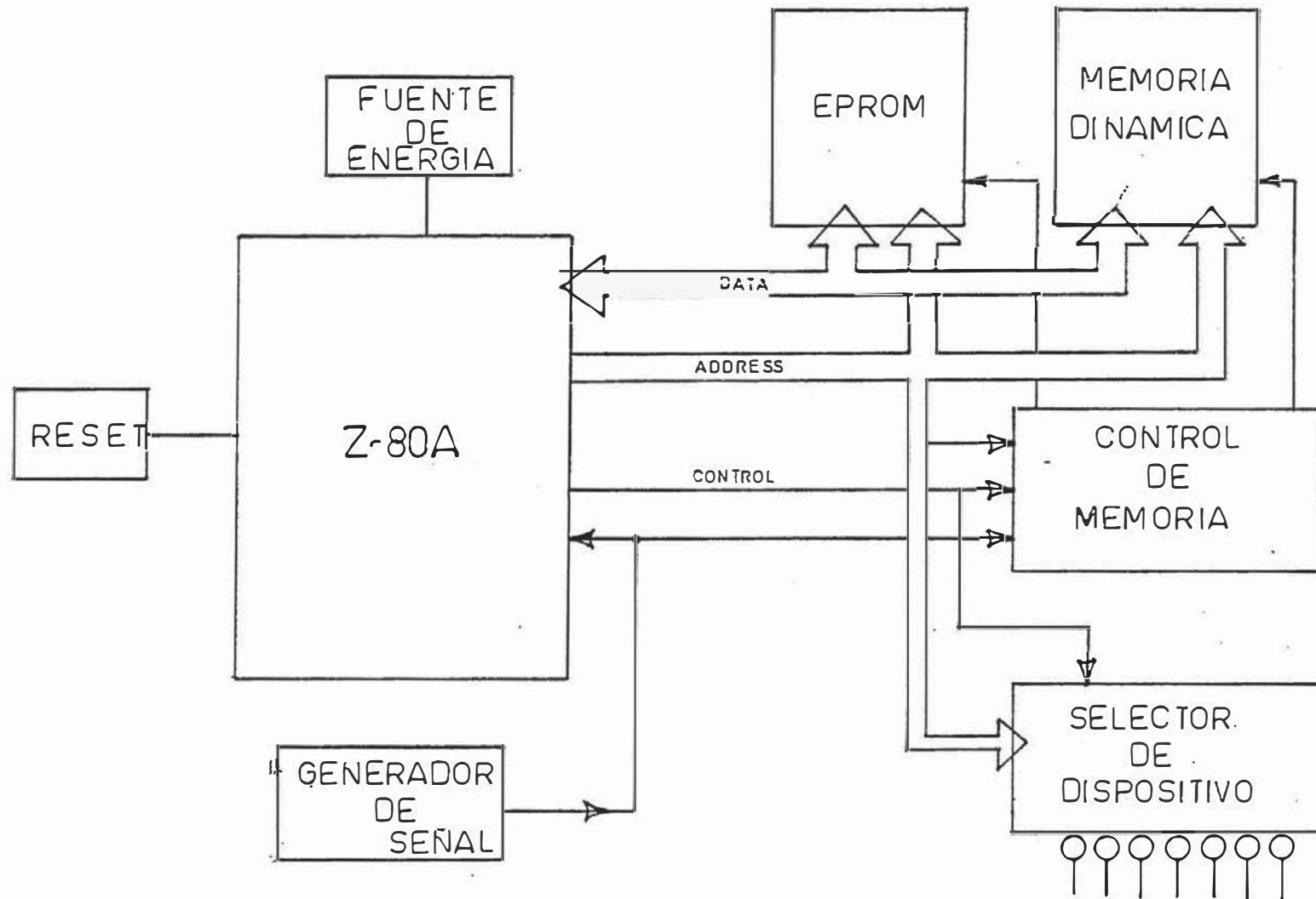
Esto significa que el dispositivo de entrada conec-tado a la línea uno es direccionado por 16 códigos de dispositivos diferentes, lo cual no es catastrófico pero si a menudo indeseable.

Similarmente, el dispositivo unido a cualquier salida del 74LS138 no es direccionado en solitario. El cir-cuito decodificador utilizado en este diseño utiliza la técnica de DECODIFICACION AMBIGUA.

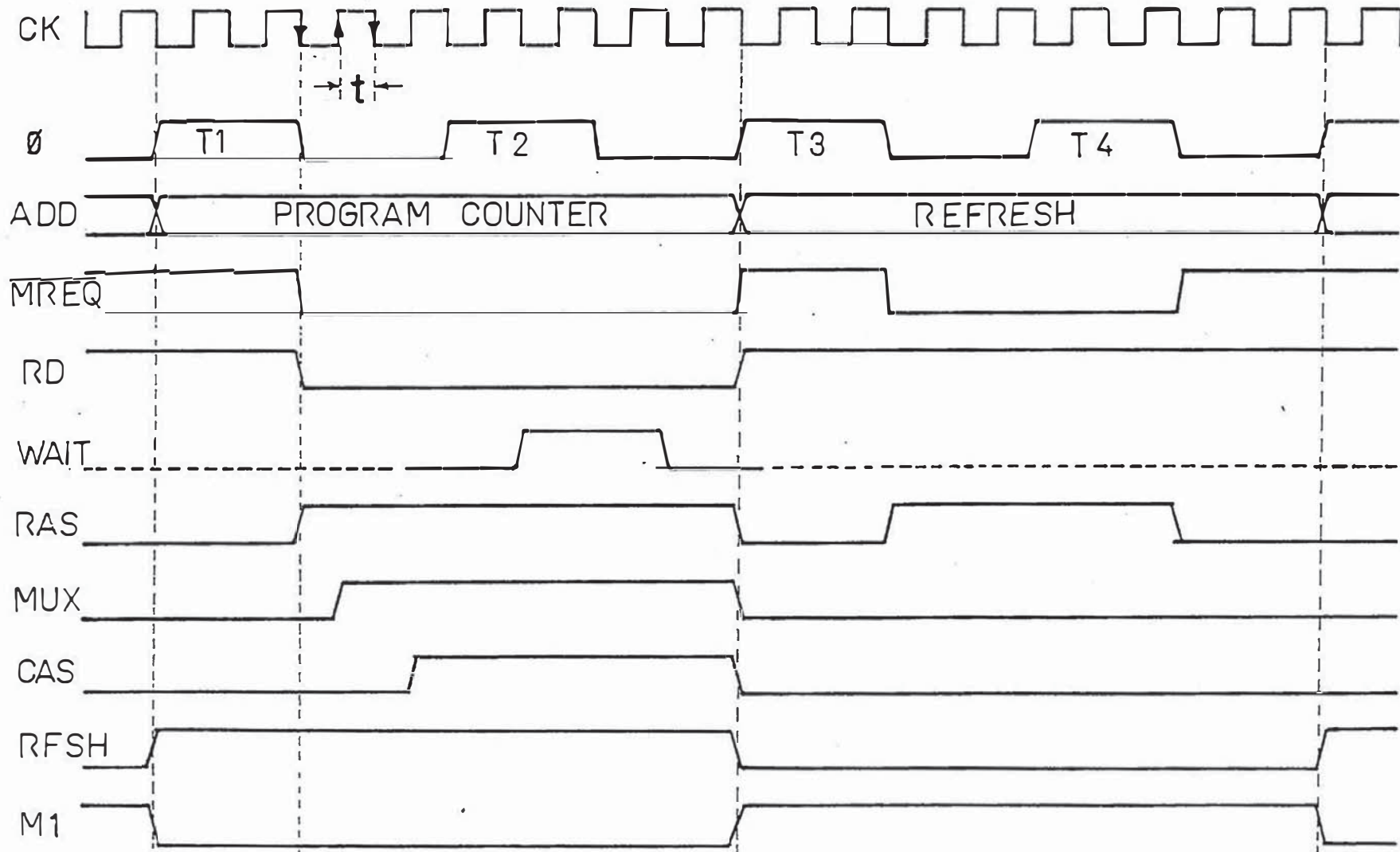
Otra señal importante es BANK que selecciona cualquiera de los dos bancos de memoria. Cuando esta en nivel ba-jo selecciona el banco 2 que son las memorias dinamicas y cuando está a nivel alto selecciona el banco 1 que es la memoria EPROM. Esta señal es generada por una combina-ción software y hardware. Por el lado del hardware tene-mos un FLIP-FLOP tipo D y un OR-INVERSOR, por el lado del software uso la instrucción OUT (SYSPRT), A. SYSPRT es el código de dispositivo y vale 14H (en exadecimal), el va-lor del acumulador A permite seleccionar al banco de me-moria deseado y mantenerlo mediante el FLIP-FLOP, solo un bit del acumulador es usado, es el bit mas significa-tivo D7. Notar que esta selección de banco también es ambigua ya que varios valores pueden habilitar o deshabilitar. Cuando se presiona el botón de RESET coloca la sali

da BANK del FLIP-FLOP en nivel ALTO seleccionando así la memoria EPROM el cual contiene el programa monitor. Por lo cuál, cuando se enciende el microcomputador inicialmente queda seleccionada la memoria EPROM.

DIAGRAMA DE BLOQUES DEL PROCESADOR



CICLO M1 DEL PROCESADOR



PROGRAMA MONITOR

PROGRAMA MONITOR

0000		NOP	00
0001		NOP	00
0002		NOP	00
0003		NOP	00
0004		NOP	00
0005		NOP	00
0006		JP , SALUDO	c3 B0 01
0100		nop	00
0101	ARRANQUE :	LD A, 90H	3E 90
0103		OUT(N4), A	D3 13
0105		LD HL, DIRECCION INICIAL	21 00 20
0108		LD (XXXX), HL	22 20 80
010B		LD A, (HL)	7E
010C		LD (XXXX+02H)	32 22 80
010F		LD A, C3H	3E C3
0111		LD (XXXX-01H),A	32 1F 80
0114	INICIO :	LD A, CDH	3E CD
0116		LD (8001H),A	32 01 80
0119		LD IX, TECLA 1	dd 21 4D 02
011D		LD (8002H),IX	DD 22 02 80
0121		LD A, C9H	3E C9
0123		LD (8004H),A	32 04 80
0126		LD SP, POINTER	31 16 80
0129		JP , START	C3 60 01
012C	YYYY :	DEFB	FE
012D		DEFB	FD
012E		defb	FB
012F		DEFB	F7
0130		DEFB	EF
0131		DEFB	DF
0132		DEFB	BF
0133		defb	3F
0134	TABLA :	DEFB	3F
0135		DEFB	06

0136		DEFB	5B
0137		DEFB	4F
0138		DEFB	66
0139		DEFB	6D
013a		DEFB	7D
013B		DEFB	07
013C		DEFB	7F
013D		DEFB	67
013E		DEFB	77
013F		DEFB	7C
0140		DEFB	39
0141		DEFB	5E
0142		DEFB	79
0143		DEFB	71
0144		NOP	00
0145	RETARDO :	PUSH AF	F5
0146		PUSH BC	C5
0147		LD BC, 0118H	01 18 01
014A	DIS :	DEC BC	0B
014B		LD A,B	78
014C		OR C	B1
014D		JRNZ, DIS	20 FB
014F		POP BC	C1
0150		POP AF	F1
0151		RET	C9

0160	START :	LD HL, YYYY	21 2C 01
0163		LD C, N2	0E 12
0165		EXX	D9
0166		LD HL, XXXX	21 20 80
0169		LD B, 01H	06 01
016B	NIBBELS :	LD D, 02H	16 02
016D	RASTREO	LD C, TABLA	0E 34
016F		EXX	D9
0170		XOR A	AF
0171		OUT (N1),A	D3 11
0173		BIT 6, (HL)	CB 76
0175		JRZ, START	28 E9
0177		LD A, (HL)	7E
0178		OUT (C), A	ED 79
017A		PUSH BC	C5
017B		PUSH HL	E5
017C		CALL RAM	CD 01 80
017F		POP HL	E1
0180		POP BC	C1
0181		XOR A	AF
0182		OUT (N1), A	D3 11
0184		OUTI (C)	ED A3
0186		EXX	D9
0187		BIT 1, D	CB 4A
0189		JRZ, MAS	28 09
018B		RLD	ED 6F
018D		CALL, EQUIV	CD A1 01
0190		RRD	ED 67
0192		JR , SAL	18 07
0194	MAS :	RRD	ED 67
0196		CALL, EQUIV	CD A1 01
0199		RLD	ED 6F
019B	SAL :	DEC D	15
019C		JRNZ, RASTREO	20 CF
019E		INC HL	23
019F		JR , NIBBELS	18 CA

01A1	EQUIV :	LD E,A	5F
01A2		ADD A,C	81
01A3		LD C,A	4F
01A4		LD A, (BC)	0A
01A5		OUT (N1),A	D3 11
01A7		LD A,E	7B
01A8		CALL, RETARDO	cD 45 01
01AB		RET	C9
01B0	SALUDO :	LD A, 90H	3E 90
01B2		OUT(N4), A	D3 13
01B4		LD HL, 8031H	21 31 80
01B7		LD (HL), 76H	36 76
01B9		INC HL	23
01BA		LD (HL), 3FH	36 3F
01Bc		INC HL	23
01BD		LD (HL), 38H	36 38
01BF		INC HL	23
01C0		LD (HL), 77H	36 77
01C2		inc HL	23
01C3		LD (HL), 00H	36 00
01C5		INC HL	23
01C6		LD (HL), 00H	36 00
01C8		LD SP, POINTER	31 16 80
01CB		LD C, N2	0E 12
01CD		LD A, 02H	3E 02
01CF		LD(80fdH),A	32 FD 80
01D2	START 1 :	LD A, 40H	3E 40
01D4		LD (80FCH), A	32 FC 80
01D7		XOR A	AF
01D8		LD HL, 80FAH	21 FA 80
01DB		LD (HL), A	77
01DC		INC HL	23
01DD		LD (HL), A	77
01DE	INT :	LD HL, 8036H	21 36 80

01E1		LD IX, (80FCH)	DD 2A FC 80
01E5	SAL:	XOR	AF
01E6		OUT (N1), A	D3 11
01E8		LD A, (HL)	7E
01E9		LD D, (IX +00H)	DD56 00
01EC		OUT (C), D	ED 51
01EE		OUT (N1), A	D3 11
01F0		DEC HL	2B
01F1		INC IX	DD 23
01F3		CALL, RETARDO	CD 45 01
01F6		LD IY, 80FBH	FD 21 FB 80
01FA		INC (IY+00H)	FD 34 00
01FD		LD A, AEH	3E AE
01FF		CP (IY+00H)	FD BE 00
0202		JRNZ, PROX	20 18
0204		XOR A	AF
0205		LD (IY+00H) , A	FD 77 00
0208		LD IY, 80FCH	FD 21 FC 80
020C		INC (IY+00H)	FD 34 00
020F		LD A, 47H	3E 47
0211		CP (IY+00H)	FD BE 00
0214		JRZ, START 1	28 BC
0216		LD A, (IY+00H)	FD 7E 00
0219		LD (80FCH), A	32 FC 80
0216	PROX. :	LD IY, 80FAH	FD 21 FA 80
0220		INC (IY+00H)	FD 34 00
0223		LD a, 06H	3E 06
0225		CP (IY+00H)	FD BE 00
0228		JRNZ, SAL 1	20 06
022A		XOR A	AF
022B		LD (IY+00H),A	FD 77 00
022E		JR ,INIT	18 AE
0230	SAL 1 :	IN A,(N3)	DB 10
0232		CP F7H	FE F7
0234		JPNZ, SAL	C2 E5 01
0237		JP ARRANQUE	C3 1F 05

0240		DEFB		DF
0241		DEFB		EF
0242		DEFB		FD
0243		DEFB		FE
0244		DEFB		F7
0245		DEFB		FBI
0246		DEFB		DF
0247		DEFB		EF
0248		DEFB		FD
0249		DEFB		FE
024A		DEFB		F7
024B		DEFB		FB
024D	TECLA 1 :	IN A, (N3)		DB 10
024F		CP FEH		FE FE
0251		RETNZ		C0
0252		LD A, (HL)		7E
0253		LD HL, ssss-02H		21 5F 02
0256	NEXT :	INC HL		23
0257		INC HL		23
0258		RRCA		0F
0259		JRC, NEXT		38 FB
025B		LD D, (HL)		56
025C		INC		23
025D		LD E, (HL)		5E
025E		LD L,D		6A
025F		LD H, E		63
0260		JP (HL)		E9
0261	ssss :	DEFW	(INICIO)	14 01
0263		DEFW	(RUN)	9002
0265		DEFW	(DEC)	81 02
0267		DEFW	(INC)	70 02
0269		defw	(DIRECCION)	10 03
026B		DEFW	(DATA)	00 03

0270	INC :	LD HL, (8020H)	2A 20 80
0273		INC HL	23
0274	COMUN :	LD A, (HL)	7E
0275		LD (8022H), A	32 22 80
0278		LD (8020H), HL	22 20 80
027b		CALL LIBERACION	CD A0 02
027E		JP ,INICIO	C3 14 01
0281	DEC :	LD HL, (8020H)	2A 20 80
0284		DEC HL	2B
0285		JR , COMUN	18 ED
0290	RUN :	LD SP, 8016H	31 16 80
0293		POP AF	F1
0294		POP BC	C1
0295		POP DE	D1
0296		POP HL	E1
0297		LD IY, (8020H)	FD 2A 20 80
029B		JP (IY)	FD E9
0010	rst16 :	LD SP, 801E	31 1E 80
0013		PUSH HL	E5
0014		PUSH DE	D5
0015		PUSH BC	C5
0016		PUSH AF	F5
0017		CALL, LIBERACION	CD A0 02
001A		JP , INICIO	C3 01 01

02A0	LIBERACION :	LD IX, LBT	DD 21 B3 02
02A4		LD (8002H),IX	DD 22 02 80
02A8		POP HL	E1
02A9		LD (8028H),HL	22 28 80
02AC		XOR A	AF
02AD		LD (802AH), A	32 2A 80
02B0		JP 0126H	C3 26 01
02B3	LBT :	IN A, (N3)	DB 10
02B5		CP FFH	FE FF
02B7		JRNZ, ENABLE	20 11
02B9		LD A, 11H	3E 11
02BB		LD HL, 802AH	21 2A 80
02BE		INC (HL)	34
02BF		CP (HL)	BE
02C0		RET NZ	C0
02C1		LD HL, (8028H)	2A 28 80
02C4		LD (800CH),HL	22 0C 80
02C7		LD SP, 800CH	31 0C 80
02CA	ENABLE :	XOR A	AF
02CB		LD (802AH), A	32 2A 80
02CE		RET	C9
02D0	PUNTO :	PUSH HL	E5
02D1		LD A, 70H	3E 70
02D3		LD HL, 8025H	21 25 80
02D6		INC (HL)	34
02D7		CP (HL)	BE
02D8		LD HL, 8026H	21 26 80
02DB		JRNZ INT	20 08
02DD		INC (HL)	34
02DE		LD HL, 8025H	21 25 80
02E1		XOR (HL)	AE
02E2		LD (HL), A	77
02E3		POP HL	E1
02E4		RET	C9
02E5	INT :	BIT 0, (HL)	CB 46
02E7		POP HL	E1
02E8		RETZ	C8

02E9		LD A, 80H	3E 80
02EB		OUT (C), B	ED 41
02ED		OUT (N1),A	D3 11
02EF		LD B, 50H	06 50
02F1	STATIC :	DJNZ, STATIC	10 FE
02F3		XOR A	AF
02F4		OUT(N1), A	D3 11
02F6		LD A, (HL)	7E
02F7		OUT (C), A	ED 79
02F9		ret	C9
0300	DATA :	CALL LIBERACION	CD A0 02
0303		LD IX, DATO 1	DD 21 70 03
0307		1d (8002H), IX	DD 22 02 80
030B		RET	C9
0310	DIRECCION :	CALL, LIBERACION	CD A0 02
0313		LD IX, DIRECCION 1	DD 21 E0 03
0317		LD (8002H)	DD 22 02 80
031B		RET	C9
0320	MATRIZ :	LD D, (HL)	56
0321		BIT 3, A	CB 5F
0323		JRZ, OPERATOR	28 0C
0325		BIT 2,A	CB 57
0327		JRZ, MEMEX	28 04
0329		LD A, 1FH	3E 1F
032B		JR, BIT	18 06
032D	MEMEX :	LD A, 15H	3E 15
032F		JR , BIT	18 02
0331		LD A, FBH	3E FB
0333	BIT :	ADD A, 01H	C6 01
0335		RRC, D	CB 0A
0337		JRC, BIT	38 FA
0339		CP 00H	FE 00
033B		JPZ, INICIO	CA 14 01
033E		CP 01H	FE 01
0340		JPZ, INICIO	CA 14 01
0343		RET	C9

0345	TECLA 12 :	LD HL, (8020H)	2A 20 80
0348		LD A, (8022H)	3A 22 80
034B		LD (HL), A	77
034C		RET	C9
034E	TECLA 13 :	LD HL, (8020H)	2A 20 80
0351		LD A, (HL)	7E
0352		LD (8022H), A	32 22 80
0355		RET	C9
0357	MASSIG :	RLD	ED 6F
0359	DUAL :	LD B, 04H	06 04
035B	BETA :	RRC (HL)	CB 0E
035D		DJNZ, BETA	10 FC
035F		RET	C9
0360	MENSIG :	RRD	ED 67
0362		JR, DUAL	18 F5

0370	DATO 1 :	LD B, EFH	06 EF
0372		CALL, PUNTO	CD D0 02
0375		IN A, (N3)	DB 10
0377		CP FFH	FE FF
0379		RET Z	C8
037A		BIT 0,A	CB 47
037C		JRNZ, BRINCO 1	20 0A
037E		BIT 5, (HL)	CB 6E
0380		JRZ, SUMATORIA 1	28 12
0382		BIT 0, (HL)	CB 46
0384		JPZ, INICIO	CA 14 01
0387	RET C9	ret	C9
0388	BRINCO 1 :	CALL MATRIZ	CD 20 03
038B		LD HL, XXXX+02H	21 22 80
038E		CALL, MASSIG	CD 57 03
0391		CALL TECLA 12	CD 45 03
0394	SUMATORIA 1 :	CALL LIBERACION	CD A0 02
0397		LD IX, DATO 2	dd 21 A0 0
039B		LD (8002H), IX	DD 22 02 8
039F		RET	C9
03A0	DATO 2 :	LD B,DF	06 DF
03A2		CALL, PUNTO	CD D0 02
03A5		IN A, (N3)	DB 10
03A7		CP FFH	FE FF
03A9		RET Z	C8
03AA		BIT 0, A	CB 47
03AC		JRNZ , BRINCO 2	20 0A
03AE		BIT 5, (HL)	CB 6E
03B0		JRZ, SUMATORIA 2	28 12
03B2		BIT 0,(HL)	CB 46
03B4		JPZ, I NICIO	CA 14 01
03B7		RET	C9
03B8	BRINCO 2:	CALL MATRIZ	CD 20 03
03BB		LD HL, XXXX+02H	21 22 80
03BE		CALL MENSIG	CD 60 03

03C1		CALL TECLA 12	CD 45 03
03C4	SUMATORIA 2 :	CALL LIBERACION	CD A0 02
03C7		LD HL, (8020H)	2A 20 80
03CA		INC HL	23
03CB		LD A, (HL)	7E
03CC		LD (8022H), A	32 22 80
03CF		LD (8020H), HL	22 20 80
03D2		LD IX, DATO 1	DD 21 70 03
03D6		LD (8002H), IX	DD 22 02 80
03DA		RET	C9
03E0	DIRECCION 1 :	LD B, FBH	06 FB
03E2		CALL PUNTO	CD D0 02
03E5		IN A, (N3)	DB 10
0E37		CP FFH	FE ff
03E9		RET Z	C8
03EA		BIT 0,A	CB 47
03EC		JRNZ, BRINCO 3	20 0A
03EE		BIT 4, (HL)	CB 66
03F0		JRZ, SUMATORIA 3	28 12
03F2		BIT 0,(HL)	CB 46
03F4		JP Z, INICIO	CA 14 01
03F7		RET	C9
03F8	BRINCO 3 :	CALL MATRIZ	CD 20 03
03FB		LD HL, XXXX+0iH	21 21 80
03FE		CALL MASSIG	CD 57 03
0401		CALL TECLA 13	CD 4E 03
0404	SUMATORIA 3 :	CALL LIBERACION	CD A0 02
0407		LD IX, DIRECCION 2	DD 21 10 04
040B		LD (8002H), IX	DD 22 02 80
040F		RET	C9
0410	DIRECCION 2 :	LD B, F7H	06 F7
0412		CALL PUNTO	CD D0 02
0415		in A, (N3)	DB 10
0417		CP FFH	FE FF
0419		RET Z	C8
041A		BIT 0,A	CB 47
041C		JRNZ, BRINCO 4	20 0A

041E		BIT 4, (HL)	CB 66
0420		JRZ, SUMATORIA 4	28 12
0422		BIT 0,(HL)	CB 46
0424		JPZ, INICIO	CA 14 01
0427		RET	C9
0428	MATRIZ :	CALL, MATRIZ	CD 20 03
042B		LD HL, XXXX+01H	21 21 80
042E		CALL MENSIG	CD 60 03
0431		CALL tecla 13	CD 4E 03
0434	SUMATORIA 4 :	CALL LIBERACION	CD A0 02
0437		LD IX, DIRECCION 3	DD 21 40 04
043B		LD (8002H),IX	DD 22 02 80
043F		RET	C9
0440	DIRECCION 3 :	LD B, FEH	06 FE
0442		CALL PUNTO	CD D0 02
0445		IN A, (N3)	DB 10
0447		CP FFH	FE FF
0449		RET Z	C8
044A		BIT 0,A	CB 47
044C		JRNZ, BRINCO 5	20 0A
044E		BIT 4, (HL)	CB 66
0450		JRZ, SUMATORIA 5	28 12
0452		BIT 0,(HL)	CB 46
0454		JPZ, INICIO	CA 14 01
0457		RET	C9
0458	BRINCO 5 :	CALL MATRIZ	CD 20 03
045B		LD HL, XXXX	21 20 80
045E		CALL MASSIG	CD 57 03
0461		CALL TECLA 13	CD 4E 03
0464	SUMATORIA 5 :	CALL LIBERACION	CD A0 02
0467		LD IX, DIRECCION 4	DD 21 70 04
046B		LD (8002H),IX	DD 22 02 80
046F		RET	C9

0470	DIRECCION 4 :	LD B, 7DH	06 FD
0472		CALL PUNTO	CD D0 02
0475		IN A, (N3)	DB 10
0477		CP FFH	FE FF
0479		RET Z	C8
047A		BIT 0,A	CB 47
047C		JRNZ, BRINCO 6	20 0A
047E		BIT 4, (HL)	CB 66
0480		JRZ, SUMATORIA 6	28 12
0482		BIT 0, (HL)	CB 46
0484		JPZ, INICIO	CA 14 01
0487		RET	C9
0488	BRINCO 6 :	CALL MATRIZ	CD 20 03
048B		LD HL, XXXX	21 20 80
048E		CALL MENSIG	CD 60 03
0491		CALL TECLA 13	CD 4E 03
0494	SUMATORIA 6 :	CALL LIBERACION	CD A0 02
0497		LD IX, DIRECCION 1	DD 21 E0 03
049B		LD (8002H), IX	DD 22 02 80
049F		RET	C9

PROGRAMA DE AUTOCARGA

CAMBIO DE BANCO 1 A BANCO 2 , EPROM a DINAMICAS

0501		LD A, 00H	3E 00
0503		OUT (SYSBANK), A	D3 14
0505		LD HL, 9000H	21 00 90
0508		LD DE,0000H	11 00 00
050B		LD BC, 2000H	01 00 20
050E		LDIR	ED B0
0510		JP ARRANQUE	C3 01 01
051F	AUTOCARGA :	LD HL, 0000H	21 0000
0522		LD DE, 9000H	11 00 90
0525		LD BE, 2000H	01 00 20
0528		LDIR	ED B0
052A		JP 9501H	C3 01 95

CAPITULO III

INTERCONEXION DE PROCESADOR A TECLADO

HEXADECIMAL Y DISPLAYS

3.1 Funciones del Teclado

Este teclado usa 6 (seis) funciones (ver figuras) básicas y necesarias para el desarrollo de programas (ver 1.4.2).

RST: Esta tecla es sensada y al ser detectada activa, se ejecuta una instrucción de salto a la dirección 0000H. -
(JP 0000H = C3 00 00)

RUN: Permite correr un programa a partir de la dirección que es mostrada por el display. Este salto y arranque se realiza de la siguiente manera: La dirección que esta siendo mostrada en el display y su dato estan almacenados en las direcciones 8020H, 8021H, 8022H, por lo tanto esto facilita el trabajo ya que mediante una instrucción LD IY, (8020H) almaceno la dirección que es mostrada en el display en el registro interno del CPU Z-80, esta dirección es formada por los bytes almacenados en las localidades de memoria 8020H y 8021H. Luego se ejecuta la instrucción JP(IY) que hace cargar el contador de programa P.C. con el valor de IY (PC ← IY). Así de esta manera sencilla se pasa el control del procesador a un programa cualquiera.

Aquí el programa completo.

0290	RUN	:	LD SP,8016H	31 16 80
0293			POP AF	F1
0294			POP BC	C1
0295			POP DE	D1
0296			POP HL	E1
0297			LDIY,(8020H)	FD 2A 20 80
029B			JP (IY)	FD E9

Se aconseja terminar un programa utilizando la instrucción "RST16" ó de lo contrario presionar la tecla RESET. Al terminar un programa utilizando como última instrucción RST"16" se vuelve al control del programa monitor y ademas se almacenan los valores de los registros a partir de la dirección 8016H.

Aquí el programa completo.

0010	"RST16" :	LDSP, 801EH	31 1E 80
0013		PUSH HL	ES
0014		PUSH DE	DS
0015		PUSH BC	CS
0016		PUSH AF	FS
0017		CALL LIBERACION	CD A0 02
001A		JP ARRANQUE	C3 01 01

8016 F

8017 A

8018 C

8019 B

801A E

801B D

801C L

801D H

801E

Y como el control del computador esta bajo el programa principal se pueden leer lo valores de los registros accedando a las direcciones 8016H hasta 801EH.

Notar que, cuando se corre un programa los registros se cargan con los valores de estas direcciones (esto inicialmente). El programa puede asignar nuevos valores a los registros de trabajo y no tomar en cuenta los valores que inicialmente tomo.

Liberación es una subrutina que detiene todo el pro

ceso hasta que Ud. deje de presionar la tecla. Esto evita que se ejecuten nuevas ordenes después de haberse ejecutado alguna por la presión de algunas de las teclas de control. El programa monitor tiene como dirección inicial 0101H por lo tanto se carga al contador de programa con aquella dirección y así el microcomputador estará bajo el control del programa monitor (programa principal).

DEC: La tecla DEC, permite decrementar en una unidad la dirección mostrada en el display, mostrando su respectivo dato almacenado en los display de data.

INC: Incrementa el valor mostrado en los display que muestra direcciones.

Aquí el programa.

0270	INC	:	LD HL, (8020H)	2A	20	80
0273			INC HL	23		
0274	COMUN	:	LDA, (HL)	7E		
0275			LD (8022H), A	32	22	80
0278			LD (8020H), HL	22	20	80
027B			CALL LIBERACION	CD	A0	02
027E			JP ARRANQUE	C3	01	01
0281	DEC	:	LD HL, (8020H)	2A	20	80
0284			DEC HL	2B		
0285			JR COMUN	18	ED	

La primera instrucción carga en el par de registros "HL" el contenido de las localidades de memoria 8020H y 8021H; en "L" lo almacenado en 8020H, en "H" lo almacenado en 8021H.

Recordar que lo almacenado en las direcciones 8020 y 8021 se displayan en los display correspondientes a direcciones y en 8022 su data respectiva que también es mostrado en los displays. Ver esquema.

		DISPLAYS							
8020	7B	1	2	3	4	5	6		
8021	02	0	2	7	B	C	D		
8022	CD								

Esto quiere decir que en la dirección 027BH esta al macenada la data CDH.

Bueno, sigamos con el analisis del programa INC, luego la instrucción INC HL incrementa en una unidad el contenido del "HL", con la instrucción LDA, (HL) carga el valor de la data de la nueva dirección en el acumulador A.

Entonces ya tenemos la nueva dirección de memoria y su dato almacenado respectivo. Luego mediante las instrucciones LD(8020H), HL y LD (8022H), A almacenamos es tos nuevos valores para que el programa principal los tome de las direcciones 8020H, 8021H y 8022H y los muestre en los display.

		DISPLAYS							
8020	7C	1	2	3	4	5	6		
8021	02	0	2	7	C	A	0		
8022	A0								

Para decrementar se procede de manera muy semejante; como ambos programas usan instrucciones comunes, se crea la subrutina común que empieza en la dirección 0274H.

En el caso que se ejecute el programa DEC (por efecto de presionar la tecla DEC) el valor da la dirección almacenado en 8020 y 8021 decrementara en una unidad en la dirección 8022 se almacenara su respectivo valor de da ta. Para que así el programa principal pueda tomar los valores almacenados en tales direcciones y mostrarlos.

		DISPLAYS					
		1	2	3	4	5	6
8020	7A						
8021	02						
8022	80	0	2	7	A	8	0

DIR: Esta tecla es usada para poder acceder a una localidad de memoria.

Si Ud. presiona la tecla DIR, se encenderá intermitentemente el led del primer display.

DISPLAYS					
1	2	3	4	5	6
0	2	8	4	2	B

led que se enciende intermitentemente indicando que el dígito puede ser alterado, tomando el valor de la tecla que sea presionada (valor en hexadecimal).

Si Ud. no desea alterar el valor del primer display entonces puede presionar nuevamente la tecla DIR, entonces se encenderá intermitentemente el led del segundo display que indicará que el dígito puede ser alterado, tomando el valor de la tecla que sea presionada (valor en hexadecimal). Si tampoco desea alterar el valor del segundo display puede presionar nuevamente la tecla DIR, entonces se encenderá intermitentemente el led del tercer display, si presiona nuevamente la tecla DIR se encenderá intermitentemente el led del cuarto display que corresponden a las presentaciones de las direcciones. Una vez que esté en este cuarto display, si vuelve a presionar la tecla DIR se volverá a encender el led del primer display y así sucesivamente.

En los casos anteriores, si Ud. presiona alguna de las teclas de datos en hexadecimal después de haber presionado la tecla DIR, este valor sera almacenado y el dis

play lo mostrara, además, el led dejará de encenderse intermitentemente y comenzará a encenderse así el led del siguiente display de direcciones, pudiendo así completarse una dirección. Si Ud. ya no desea alterar mas ningún valor mostrado por los display, entonces presione cualquier tecla de la última fila, (las teclas libres) que le permitirá salir de esa opción y así podrán habilitarse las teclas de funciones, para que el computador responda a sus requerimientos.

Cada vez que se altere un valor del display de direcciones, automáticamente se actualizarán los datos mostrados en los displays de datos.

Aquí el programa dirección 1.

03E0	DIRECCION 1 :	LDB,FBH	06 FB
03E2		CALL PUNTO	CD DO 02
03E5		INA, (N3)	DB 08
03E7		CP FFH	FE FF
03E9		RETZ	C8
03EA		BIT 0, A	CB 47
03EC		JR NZ, BRINCO 3	20 OA
03EE		BIT 4, (HL)	CB 66
03FO		JRZ, 3	28 12
03F2		BIT 0, (HL)	CB 46
03F4		JPZ, INICIO	CA 14 01
03F7		RET	C9
03F8	BRINCO 3 :	CALL MATRIZ	CD 20 03
03FB		LDHL, XXXX+01H	21 21 80
03FE		CALL MASSIG	CD 57 03
0401		CALL TECLA 13	CD 4E 03
0404	3 :	CALL LIBERACION	CD AO 02
0407		LD IX, DIRECCION 2	DD 21 10 04
040B		LD (8002H), IX	DD 22 02 80
040F		RET	C9

Este programa es utilizado para alterar el primer dí

gito de direcciones mostrado en hexadecimal (DIRECCION 1). El programa que altera el segundo dígito se llama DIRECCION 2, para el tercer dígito el programa se llama DIRECCION 3 y para el cuarto dígito, DIRECCION 4. Si se llegara a ejecutar el primer programa cambiando algún valor, el control pasa al siguiente programa (en este caso pasará a DIRECCION 2).

Al ejecutar el programa se utiliza la subrutina PUNTO que hace, encender intermitentemente el led indicado por el valor cargado en el registro B (FBH en hexadecimal indica el primer led). Luego se realiza una entrada de data proveniente del teclado mediante la instrucción IN A, (N3) luego se detectará mediante la instrucción CP FF H si el teclado ha enviado información o no, si no hay información entonces mediante la instrucción RETZ devuelve, el control al programa principal, el cual luego de ejecutar su función de mostrar en los display alguna dirección y su data almacenado, salta a la subrutina DIRECCION 1 y mientras no se presione alguna tecla este lazo de programa se seguirá ejecutando (programa principal y subrutina DIRECCION 1).

Si alguna tecla se presiona, entonces se averigua, si se ha presionado la tecla RESET o DIR.

Si se presiono RESET, mediante la instrucción, (JPZ, INICIO), se salta al programa principal inicial.

Si se presiono DIR se salta a la subrutina (3) que cambia de subrutina de lazo cerrado al programa principal. Por lo tanto se ejecutará el programa principal y la subrutina DIRECCION 2 manteniéndose así este lazo hasta que no se salga de esta opción DIR.

Si alguna tecla de datos hexadecimales se presiona, entonces se ejecutará la subrutina BRINCO 3 el cual acu

dirá a una subrutina llamada MATRIZ para decodificar por SOFTWARE la información proveniente del teclado, luego la subrutina MASSIG realiza el cambio en el nibble mas significativo (ver esquema), colocando allí, el nuevo valor hexadecimal que será mostrado, y la subrutina TECLA 13 actualizará el valor contenido en la dirección 8022H, almacenando en la dirección 8022H, el valor de la data almacenada en la nueva dirección mostrada.

```
8020 E9
8021 03 nibble mas significativo
8022 C8
```

Si se hubiera presionado la tecla 4, se tendría

8020	E9			DISPLAYS			
8021	43	1	2	3	4	5	6
8022	FF	4	3	E	9	F	F

notar que en la dirección 8022 se carga el valor almacenado en la dirección de memoria 43E9H, automáticamente, por efecto de la subrutina TECLA 13.

Por último se ejecuta la subrutina 3 que permite cambiar de subrutina de lazo al programa principal. Realizándose el lazo siguiente: programa principal y subrutina de DIRECCION 2 y así sucesivamente.

DAT: Esta tecla es usada para cambiar el contenido del dato almacenado en alguna localidad de memoria mostrada en los displays.

Si Ud. presiona la tecla DAT, se encenderá intermitentemente el led del display 5.

DISPLAYS

1	2	3	4	5	6
0	3	7	9	C	8

led que se enciende intermitentemente, indicando que el dígito puede ser alterado, tomando el valor de la tecla que sea presionada (valor en hexadecimal).

Si Ud. no desea alterar el valor del quinto display, entonces, puede presionar nuevamente la tecla DAT, para lo cual, se encenderá intermitentemente el led del segundo display que indicara que el dígito puede ser alterado, tomando el valor de la tecla que sea presionada (valor hexadecimal). Si tampoco desea alterar el valor del display 6, puede presionar nuevamente la tecla DAT, que hará que se incremente en uno, la dirección mostrada y actualizará automáticamente el valor mostrado en los displays de datos, además, el led del quinto display se encenderá intermitentemente, indicando así, que puede ser alterado.

Este incremento automático de dirección al alterar un byte, facilita el cargado de programas en memoria, porque evita así, el tener que incrementar la dirección manualmente, lo cual evita errores al copiar, ya que la atención esta dedicada al ingreso de datos.

En los casos anteriores, si Ud. presiona alguna de las teclas de datos en hexadecimal después de haber presionado la tecla DAT, este valor será almacenado y el display lo mostrara, además, el led dejará de encenderse intermitentemente y comenzará a encenderse así el led del siguiente display. Si Ud. carga con un dato el sexto display, automáticamente se incrementará la dirección mostrada en los displays y aparecerá su respectiva data alma-

cenada en los displays correspondientes, encendiéndose intermitentemente el led del quinto display y esto continua así sucesivamente con las direcciones siguientes. Si Ud. ya no desea alterar mas ningún valor de datos mostrados por los displays, entonces presione cualquier tecla de la última fila (las teclas libres) que le permitirá salir de esa opción y así podrán habilitarse las teclas de funciones, para que el computador responda a sus requerimientos.

Aqui el programa DATO 1.

0370	DATO 1	:	LDB, EFH	06 EF
0372			CALL PUNTO	CD DO 02
0375			IN A, (N3)	DB 08
0377			CP FFH	FE FF
0379			RET Z	C8
037A			BIT 0, A	CB 47
037C			JRNZ, BRINCO 1	20 0A
037E			BIT 5, (HL)	CB GE
0380			JRZ, 1	28 12
0382			BIT 0, (HL)	CB 46
0384			JP Z, INICIO	CA 14 01
0387			RET	C9
0388	BRINCO 1	:	CALL MATRIZ	CD 20 03
038B			LD HL, XXXX+02H	21 22 80
038E			CALL MASSIG	CD 57 03
0391			CALL TECLA 12	CD 45 03
0394	1	:	CALL LIBERACION	CD A0 02
0397			LD IX, DATO 2	DD 21 A0 03
039B			LD (8002H) IX	DD 22 02 80
039F			RET	C9

Este programa altera el quinto dígito de los displays, el programa que altera el sexto dígito se llama DATA 2.

Al ejecutar el programa se utiliza la subrutina PUNTO que hace encender intermitentemente el led indicado por el valor cargado en el registro B (EFH en hexadecimal, indica el quinto led).

Luego mediante las instrucciones INA, (n3) y CP FFH se detectará, si el teclado ha sido usado o no para ingresar datos o se han presionado las teclas RESET o DAT. Si ninguna tecla fue preseionada, se devuelve al control al programa principal usando la instrucción RETZ. Por lo tanto se produce el lazo siguiente: programa principal y subrutina DATA 1.

Si alguna tecla se presiona, entonces se averigua, si se presiono la tecla RESET o DAT; para esto mediante la instrucción BIT 0, A se averigua si se presiono alguna tecla de funcion situadas en la "primera fila" del teclado si fue asi las instrucciones BIT 0, (HL) y BIT 5, (HL) nos indica que "columna" fue presionada (como Ud. puede ver en los esquemas "0" corresponde a la tecla RESET y "5" a la tecla DAT).

Si se presiono RESET, mediante la instrucción, (JPZ, inicio), se salta el programa inicial.

Si se presiono DAT se salta a la subrutina (1) que cambia la subrutina de lazo cerrado al programa principal. Por lo tanto se ejecutará el programa principal y la subrutina DATO 2 manteniéndose así este lazo, hasta que no se salga de esta opción DAT.

Si alguna tecla de datos hexadecimales se presiona, entonces se ejecutará la subrutina BRINCO 1, al cual acudiré a una subrutina MATRIZ para decodificar por software la información proveniente del teclado, luego la subrutina MASSIG realiza el cambio en el nibble mas significativo (8021H, ver esquema), colocando allí el nuevo valor hexadecimal que será mostrado y la subrutina TECLA 12 actualizará el contenido de la dirección de memoria mostrada.

8020 E9

8021 03

8022 98 nibble mas significativo

Por último se ejecuta la subrutina 1, que permite cambiar la subrutina de lazo al programa principal. Realizándose el lazo siguiente: Programa principal y subrutina de DATA 2 y así sucesivamente.

3.2 Circuito usado como interfaz para la conexión del teclado y display

Se utiliza el circuito integrado de código 8255A de 40 patillas de nombre interfaz periférica programable (P.P.I.). En este trabajo solo es usado (programado) el modo 0. Tiene 3 port de salida, el A, B y C, los ports B y C están programados para que trabajen como port de salida y el port A como entrada.

La programación es bien sencilla, solo se utiliza una palabra de control.

Control Word.

D7	D6	D5	D4	D3	D2	D1	D0
1	X	X	1	0	X	0	0

que sera enviada por el BUS de datos desde el microprocesador hacia el P.P.I.

El bit D7 esta a un nivel uno lógico, indicando que se quiere seleccionar el modo 0 y los bits D0, D1, D3, D4 la combinación seleccionada de los ports A, B, C. Notar que los bits D2, D5, D6 son irrelevantes.

Las señales usadas en este circuito integrado son RD, WR, A0, A1, RESET, Cs; notar que la señal RESET actua cuando el nivel es uno lógico, contrario al usado en el microprocesador Z-80. A0, A1 son los bits menos significativos del bus de direcciones que seran usados para colocar en el bus de datos uno de los ports o colocar el contenido del bus de datos en cualquiera de los ports A, B o C

OPERACION BASICA

A1	A0	RD	WR	CS	Operación de lectura	
0	0	0	1	0	Port A	Data Bus
0	1	0	1	0	Port B	Data Bus
1	0	0	1	0	Port C	Data Bus
					Operación de escritura	
0	0	1	0	0	Data Bus	Port A
0	1	1	0	0	Data Bus	Port B
1	0	1	0	0	Data Bus	Port C
1	1	1	0	0	Data Bus	Control
					Función deshabilitada	
X	X	X	X	1	Data Bus	Tri-State
1	1	0	1	0	Condición ilegal	
X	X	1	1	0	Data Bus	Tri-State

Por lo tanto Ud. podrá utilizar los bits desde A2 hasta A7 para seleccionar el P.P.I. y así conseguir la señal PPICE.

Supongamos que deseamos sacar por el port C el valor 7EH en hexadecimal, además supongamos que el decodificador selecciona con la combinación 000101 (A7=0, A6=0, A5=0, A4=1, A3=0, A2=1), para sacar el contenido del bus de datos por el port C los bits A1, A0 deben tomar los valores 10 (A1=1, A0=0) entonces tendremos 00010110 como byte selector, al cual llamamos N. En este ejemplo el byte selector, N, vale 16H en hexadecimal. Por lo tanto mediante las siguientes instrucciones lograremos presentar por el port C el valor 7EH:

```

N : EQU 016H
MUESTRA : DB 07EH
LD A, MUESTRA      3E 7E
DUT (N), A         D3 16
    
```

Y como programa de programación del PPI los valores de los bits A1, A0 deben ser 11 (DATA BUS → CONTROL) y

así tenemos como byte selector 00010111 = 17H, la palabra de control es 90H y así tenemos el programa:

```
      N4 : EQU 017H
CONTROL : DB 090H
          LDA, CONTROL      3E 90
          OUT (N4), A       D3 17
```

Bueno para nuestro caso, el P.P.I. es seleccionado - por los bits 000100 (A7-A2), entonces tenemos:

	7	6	5	4	3	2	1	Ø		
PORT B (Salida)	0	0	0	1	0	0	0	1	11H	N1
PORT C (Salida)	0	0	0	1	0	0	1	0	12H	N2
PORT A (Entrada)	0	0	0	1	0	0	0	0	10H	N3
CONTROL	0	0	0	1	0	0	1	1	13H	N4

por lo tanto tenemos cuatro bytes selectores. En este diseño trabaja en MODO "0" y con los ports designados como entrada, salida y salida para los ports A, B y C respectivamente, y así la palabra de control es 90H.

CONTROL WORD 1 0 0 1 0 0 0 0 90H.

ver esquemas al final de este capítulo.

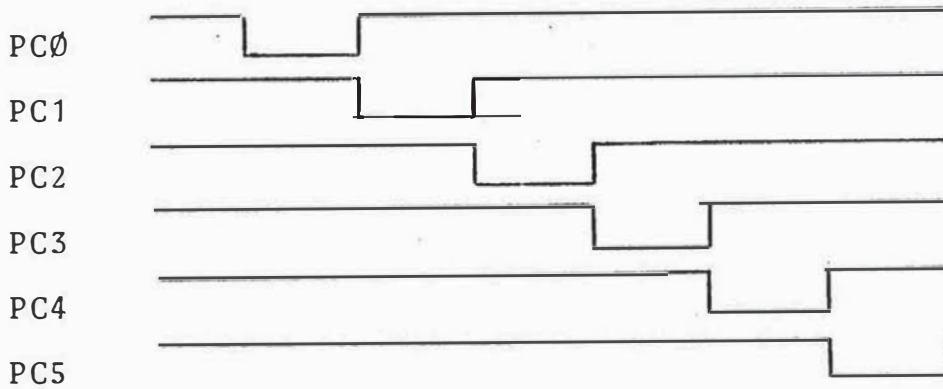
El teclado consta de cuatro filas. Cada fila está conectada por un lado a una resistencia de 10KN y esta a la fuente de 5 voltios y por el otro lado hacia una línea del port A (entrada PPI). Las filas están conectadas a las líneas menos significativas PAØ, PA1, PA2, PA3 llamándose fila uno a la fila conectada a la línea PAØ del port A y así sucesivamente con las demás filas.

Este teclado también consta de seis columnas. Cada columna está conectada a una línea del port C. Las columnas están conectadas a las líneas menos significativas - PCØ, PC1, PC2, PC3, PC4, PC5 llamándose columna uno a la columna conectada a la columna PCØ del port C y así sucesivamente con las demás columnas.

El teclado tiene 24 (veinte y cuatro) switches, cada

uno, una fila y una columna si es presionada.

El port C, solo utiliza 6 líneas que producen las siguientes señales



que muestrean los switch, si han sido presionados o no, con la ayuda de la lectura del port A.

Supongamos que se presione el switch ubicado en la segunda fila y tercera columna.

Entonces, solo cuando PC2 este en nivel bajo, solo en ese momento la segunda fila tendrá el valor de cero lógico. Y así se forma una matriz: El port A toma el valor 11111101 y el port C da el valor 11111011. El port A da el número de fila y el port C el número de columna.

El port C muestra valores siguientes

1 1 1 1 1 1 1 0	F E H
1 1 1 1 1 1 0 1	F D H
1 1 1 1 1 0 1 1	F B H
1 1 1 1 0 1 1 1	F 7 H
1 1 1 0 1 1 1 1	E F H
1 1 0 1 1 1 1 1	D F H

la presentación es recirculante para así generar las señales antes mostradas.

El port A puede tomar solo los valores

1 1 1 1 1 1 1 0	F E H
1 1 1 1 1 1 0 1	F D H
1 1 1 1 1 0 1 1	F B H

1 1 1 1 0 1 1 1 F 7 H

Entonces utilizando una tabla, se puede identificar el switch presionado.

Estas señales que muestra el port C, se aprovecha también para realizar un SCAN a los seis displays, habilitando solo un display por vez. Según puede verse en el diagrama de señales, solo una línea cae a cero por vez.

Cada display consta de siete segmentos, mas un led (que es presentado como un punto). El display usado es de ánodo común. Todos los displays estan conectados en paralelo a el port B del P.P.I. (byte selector N1). La conexión se hace a travez de ocho inversores (siete segmentos mas un led).

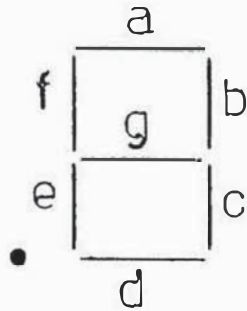
La conexión de los displays al port C (port que habilitara a solo un display) se hace a travez de seis inversores open colector, utilizando seis resistencias de 56N cada una, este valor de resistencia nos permita graduar la luminosidad de los displays. Recordar que solo un display es habilitado por vez, y por lo tanto, el consumo de corriente es mínimo.

Como ejemplo supongamos que deseamos mostrar en el segundo display la letra F: para lo cual los ports B y C deben mostrar (ver diagrama circuital).

El port C deberá mostrar el byte F7H

El port B deberá mostrar el byte 71H

A cada segmento del display se le designa una letra minuscula.



además cada segmento o punto se ilumina cuando el pin correspondiente al segmento es llevado a un nivel bajo.

Como cada segmento o punto pasa a travez de un inversor hacia la línea del port B. Esta será iluminado cuando en la línea del port B aparezca un nivel lógico uno. Por lo tanto para mostrar la letra F en el display, el port B tendrá que mostrar el byte 71H.

```
. f g e d c b a      " F "  
0 1 1 1 0 0 0 1
```

Cuando se seleccione otro display por otro byte del port C el port B mostrará al Byte que se desea representar en tal display. Y toda esta secuencia ocurre a alta velocidad, de tal manera, que parece que los 6 display están activos a la vez.

3.3 Interacción del procesador con el teclado y display

El procesador central ve el teclado y display como dispositivos de entrada y salida respectivamente. El procesador usa el procesamiento de muestreo para atender al teclado; aprovechando estos tiempos de muestreo, para mostrar en los displays la dirección y su dato almacenado.

Inicialmente el procesador decodifica las instrucciones a partir de la dirección 0000H.

Se muestra el programa principal al final del capítulo.

Las primeras instrucciones son de no operación (NOP) esto para asegurar la estabilización del circuito. En la dirección 0006H encontramos la instrucción JP SALUDO, que hace que el control lo tome la subrutina SALUDO, su función es la de presentar el saludo HOLA, recirculando de derecha a izquierda, como se muestra:

(1)	H O L A _ _	(4)	A _ _ H O L
(2)	O L A _ _ H	(5)	_ _ H O L A
(3)	L A _ _ H O	(6)	_ H O L A _

Vea el diagrama de flujo siguiente, que le permite a Ud. analizar la secuencia del programa, que realiza la presentación HOLA, recirculantes.

1. Primero se programa el PPI mediante el uso de las instrucciones:

```
LDA, Control Word. 3E90
OUT (N4), A        D313
```

2. Se llena la tabla que permita mostrar hola

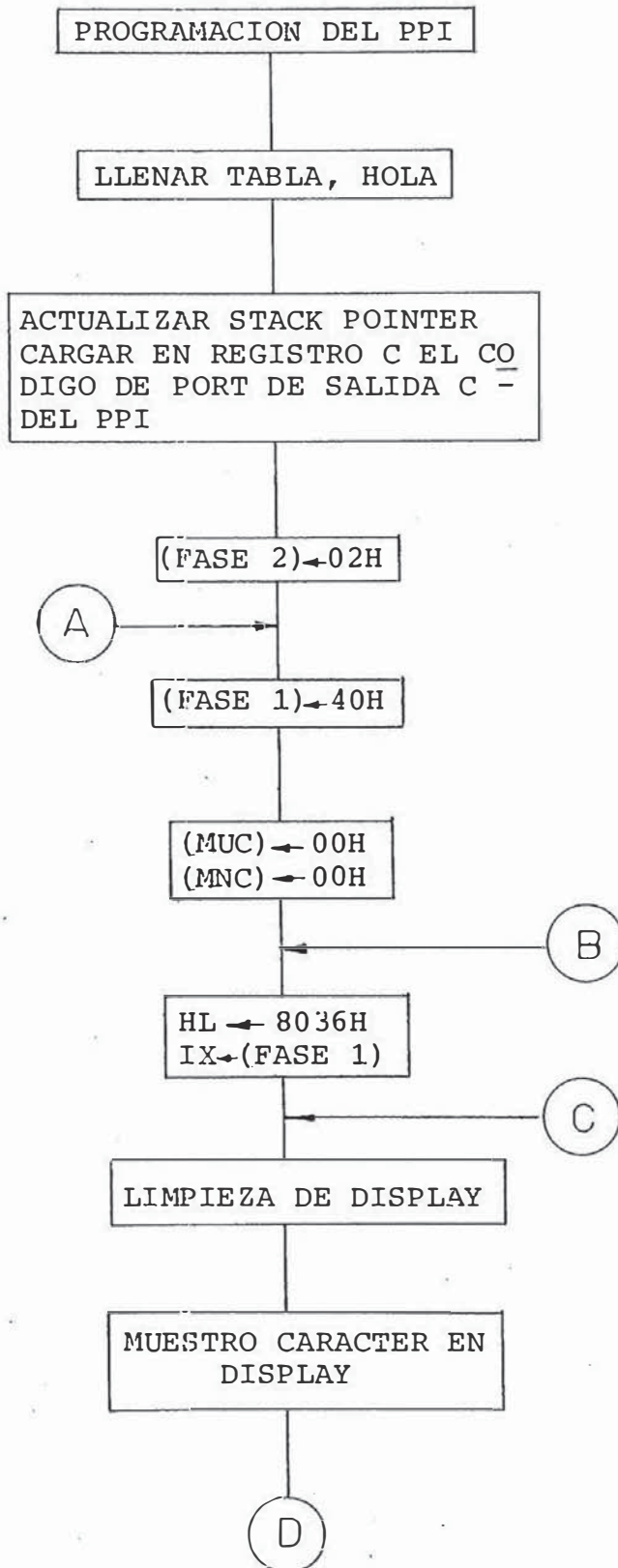
```
8031 H
8032 O
8033 L
8034 A
8035 "CERO"
8036 "CERO"
```

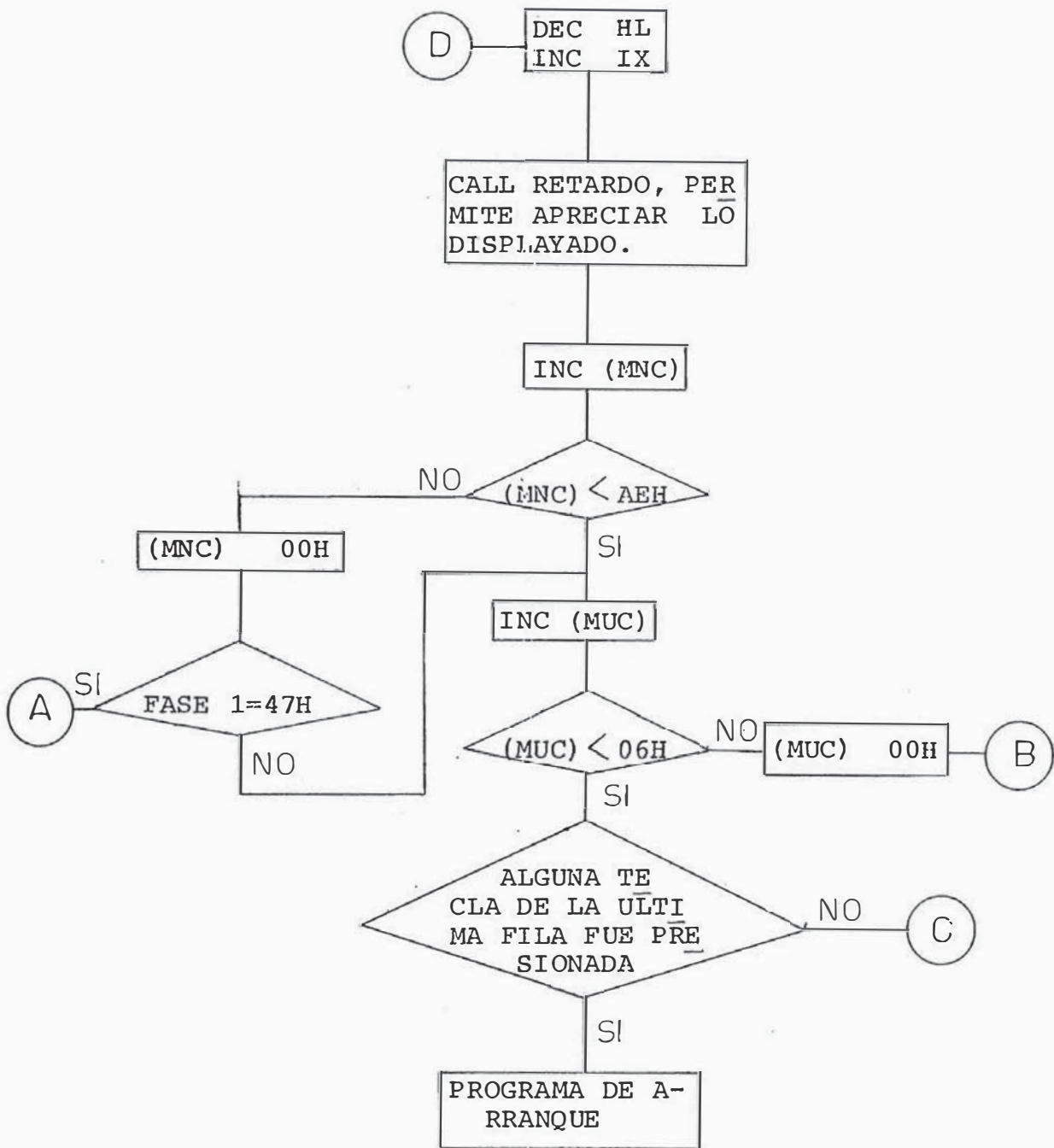
Por ejemplo para cargar H se utilizan las instrucciones.

```
LDHL, 8031H        21 31 80
LD(HL), byte "H"   3676
```

para cargar el byte "O" solo incrementamos el valor de Hl en una unidad, para así obtener la siguiente dirección de almacenaje:

DIAGRAMA DE FLUJO DEL PROGRAMA QUE PERMITE
PRESENTAR "HOLA" RECICLANDO





```
INC HL          23
LD (HL), byte "0" 36 3F
```

y así sucesivamente hasta completar la tabla.

3. Se carga el indicador de stack con la dirección 8016H

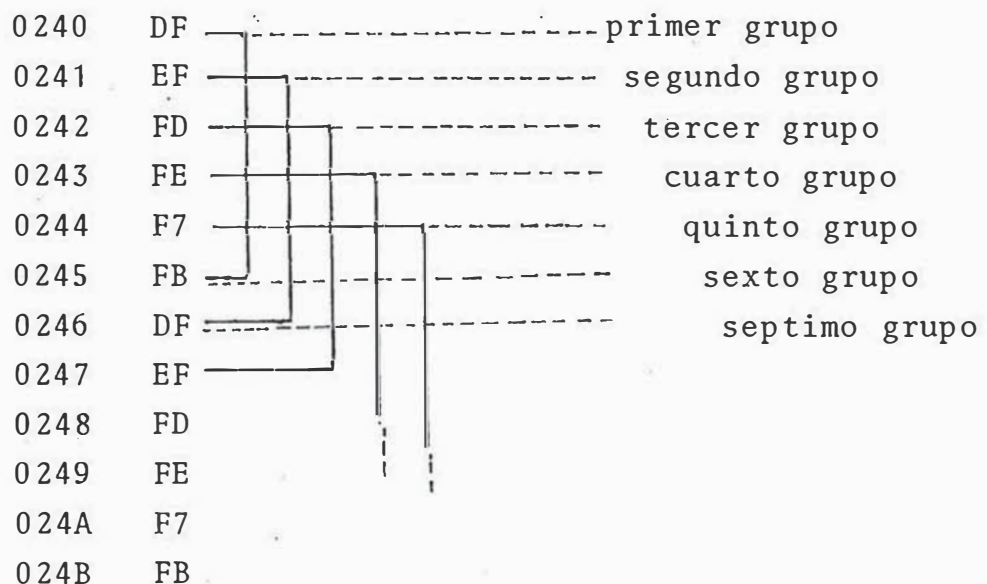
Como este programa usa la instrucción de salida OUT (C), D donde C y D son registros. D estara cargado con la data a presentar y C con el byte selector. En este caso C es cargado con el byte selector N2 = 12H.

4. Se creara una tabla de control, que mas adelante se utilizará. Cargado inicialmente así:

MUC	:	80FA	00	muestra un ciclo
MNC	:	80FB	00	muestra N ciclos
FASE 1	:	80FC	40	
FASE 2	:	80FD	02	

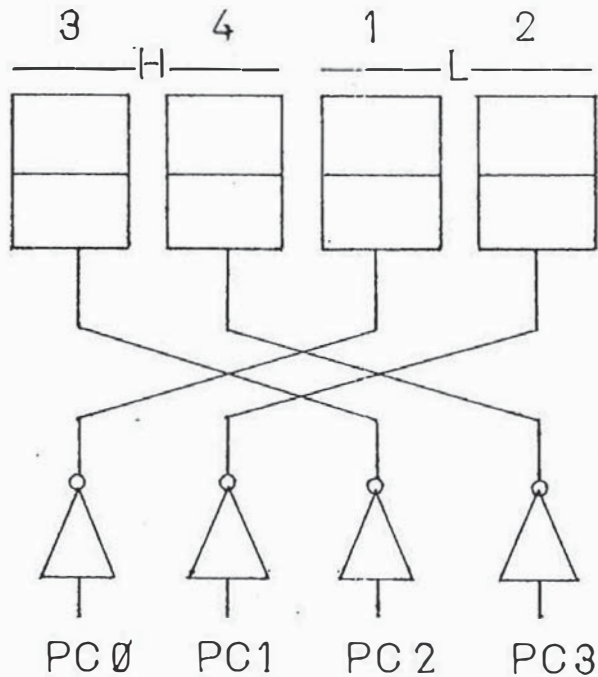
MUC son las iniciales de muestra de un ciclo

MNC muestra N ciclos, FASE 2 y FASE 1 cargan dos bytes - que forman una dirección de inicio de una tabla usada - para habilitar los displays requeridos.



Hagamos un parentésis al diagrama de flujo, y aprovechemos estas tres tablas para entender el metodo de programación.

Hay un detalle muy importante en la presentación de los displays. Se habia dicho que los display estan seleccionados por las líneas del port C del PPI , pero notar en el diagrama la reubicación de los display de direcciones.



Esto es debido a que cuando almacenamos alguno dirección a mostrar en las direcciones 8020H y 8021H se hace por medio de instrucciones LD (8020H), HL; la cual carga en 8020H el valor del registro L y en 8021H el valor del registro H. Cuando el programa principal muestra en los display de direcciones los valores almacenados en los direcciones 8020H y 8021H primero mostrara el valor del registro L y luego el del registro H lo cuál no facilita la lectura de la dirección observada ya que primero se estaria mostrando el valor menos significativo y luego el mas significativo, entonces, es por eso que se hace el intercambio en las parajes de display. Notar que este cambio no altera al programa principal en lo absoluto. Bueno, ahora continuemos con el analisis:

Para mostrar HOLA, el programa presentara por el -

port B del PPI, los valores de la tabla 8031H, empezando desde 8036 hasta llegar a 8031H y así repetir. A la vez utilizara la tabla 0240H para habilitar los displays correspondientes, presentando los valores almacenados en esta tabla por el port C del P.P.I..

Primero se mostrara lo siguiente en los displays:

```
3 4 1 2 5 6
H O L A _ _
```

El port C del P.P.I. mostrara el byte DFH que habilitara al display 6.

El port B del PPI mostrara al byte 00H que se displaya solo en el display habilitado, (Recordar que todos los display estan conectados en paralelo hacia el port B del P.P.I. por medio de 8 inversores simples, además el byte 00H no enciende ningún segmento del display).

Luego el port C mostrara el byte EFH que habilitara el display 5. El port B mostrara el byte 00H, no encendiéndose ningún display.

Luego al port C mostrara el byte FDH que habilitara el display 2. El port B mostrara el byte 77H que representa la letra "A".

Luego el port C mostrara al byte FEH que habilitara el display 1. El port B mostrara el byte 38H que representa la letra "L"

Luego el port C mostrara el byte F7H que habilitara el display 4. El port B mostrara el byte 3FH que representa

ta la letra "0".

Luego al port C mostrara el byte FBH que habilita ra el display 3. El port B mostrara el Byte 76H que re presenta la letra "H".

Luego el port C mostrara el byte DFH que habilitara el display 6. El port B mostrar al byte 00H, no encen diéndose ningún segmento de display. Y este ciclo comple to se repite AEH veces, (en hexadecimal), durando aproxi madamente 1.8 segundos. Además la presentación entre ca racter y caracter toma un tiempo de aproximadamente 1.78 milisegundos, para que así se pueda ver mayor iluminado ya que así, el afecto optico permite ver con mayor clari dad cada segmento de display.

Entonces después de presentar un caracter se llama a una subrutina llamada retardo el cual permite que se a precie el caracter por un corto tiempo necesario para que el ser humano defina lo que ve.

Calculamos el tiempo:		N° de Stados T
RETARDO :	PUSH AF F5	11
	PUSH BC C5	11
	LDBC, 0118H 01 18 01	10
DIS :	DEC BC 0B	6
	LDA, B 78	4
	OR C B1	4
	JR NZ, DIS 20FB	7,12
	POP BC C1	10
	POP AF F1	10

$$t = 11T + 11T + 10T + (0118H-1) (6T + 4T + 4T + 7T) \\ + (GT + 4T + 4T + 12T) + 10T + 10T$$

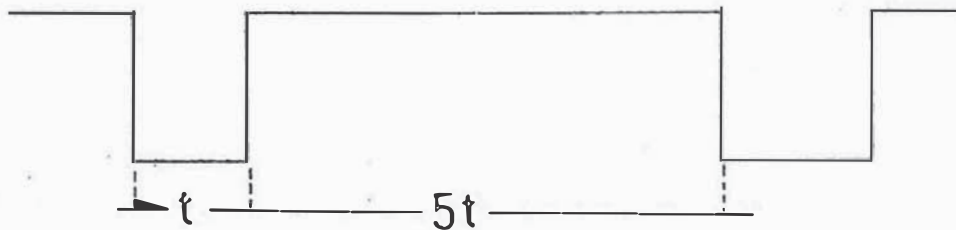
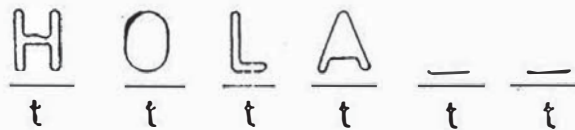
$$0118H = 16^2 + 16 + 8 = 280 \quad (10)$$

$$T = 5937T \quad T = \frac{1}{3.325MHz}$$

$$t = \frac{5937}{3.3325} \times 10^{-6} \text{ segundos}$$

$$t = 1.78 \text{ milisegundos}$$

Por lo tanto, si se presenta el caracter "A" el programa lo presentara por 1.78 milisegundos luego presentará el caracter "L" y también lo presentará por 1.78 milisegundos y así sucesivamente.



obteniéndose una frecuencia de 93.6Hz aproximadamente. No tar que el ciclo no es simétrico.

La primera muestra de HOLA XX dura 1.8 segundos aproximadamente, esto por que se usa un contador almacenado en la dirección 80FBH llamado MNC (muestra N ciclos) al cual muestra 174 caracteres (AEH en hexadecimal) mientras se mantiene la palabra completa estática.

$$t_1 = 174 \times 1.78 \text{ milisegundos}$$

$$t_1 = 1.8 \text{ segundos}$$

Después de toda esa primera presentación, FASE 1 - osea (80FCH), es incrementado en 1 al valor del byte almacenado, por lo tanto el inicio de la tabla ya no será sino 0241H lo cual permitirá lograr un afecto optico de corrimiento de la palabra HOLA XX

Ahora tendremos mostrado:

```
3 4 1 2 5 6
O L A _ _ H
```

El port C mostrará el byte EFH que habilitará el display 5. El port B mostrará el byte 00H que no ilumina ningún segmento de display.

Luego el port C mostrará el byte FDH que habilitará el display 2. El port B mostrará el byte 00H.

Luego el port C mostrará el byte FEH que habilitará el display 1. El port B mostrará el byte 77H que representa la letra "A".

Y así hasta llegar al display 6.

El port C mostrará el byte DFH que habilitará el display 6. El port B mostrará el valor 76H que representa al caracter "A".

Como ya se indicó anteriormente, cada carácter es presentado por 1.78 milisegundos además este segundo grupo de presentación durará también 1.8 segundos aproximadamente.

Después de esta segunda presentación, FASE 1 osea - (80FCH), es incrementados en 1 el valor del byte, almacenado, por lo tanto el inicio de la tabla ya no será 0241H sino 0242H el cual permitirá lograr el efecto optico de corrimiento.

Y así se tendrá la siguiente presentación

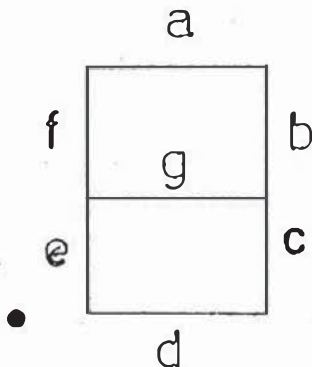
3 4 1 2 5 6
L A _ _ H O

El byte almacenado en FASE 1 es incrementado hasta 46, y así se tendrán siete grupos de presentación. Las tablas que inician con la dirección 0240H y 0246H son iguales por lo tanto la primera presentación dura 3.6 segundos aproximadamente y es así que se tienen siete presentaciones repitiéndose una vez la primera presentación.

FASE 1 toma los valores desde 40 hasta 46, cuando llega a tomar el valor de 46 nuevamente se carga con 40 y así se tiene un corrimiento uniforme hacia la izquierda la palabra mostrada.

Como ejemplo: podemos mostrar la palabra "digi" alterando ciertas localidades de memoria (tabla 8031H hasta 8033H).

Para hallar los bytes que representan los caracteres "d", "i", "g" e "i" se procede de la siguiente manera:



. g f e d c b a					
0 1 0 1 1 1 1 0	5EH	representa	al	caracter	"d"
0 0 0 0 0 1 0 0	04H	"	"	"	"i"
0 1 1 0 1 1 1 1	6FH	"	"	"	"g"
0 0 0 0 0 1 0 0	04H	"	"	"	"i"

Entonces debemos de alterar

01B8	5E	por	76
01BB	04	por	3F
01Be	6F	por	38
01C1	04	por	77

Y corremos al programa a partir de la dirección 0000H el cual cargara la nueva tabla (8031H hasta 8033H) con la palabra a mostrarse.

También podemos hacer otra prueba, para afianzar mas el analisis de este programa, podríamos hacer que recircule a mayor velocidad, cambiando el valor de un solo byte.

El byte AEH nos da un tiempo de 1.8 segundos. Entonces para aumentar la velocidad debemos disminuir este valor por ejemplo a 50H, esto se logra alterando la localidad de memoria 01FEH.

01FE 50 por AE

Luego corre el programa a partir de la dirección - 0000H y podrá ver la palabra mostrada recirculando a mayor velocidad.

TECLADO HEXADECIMAL

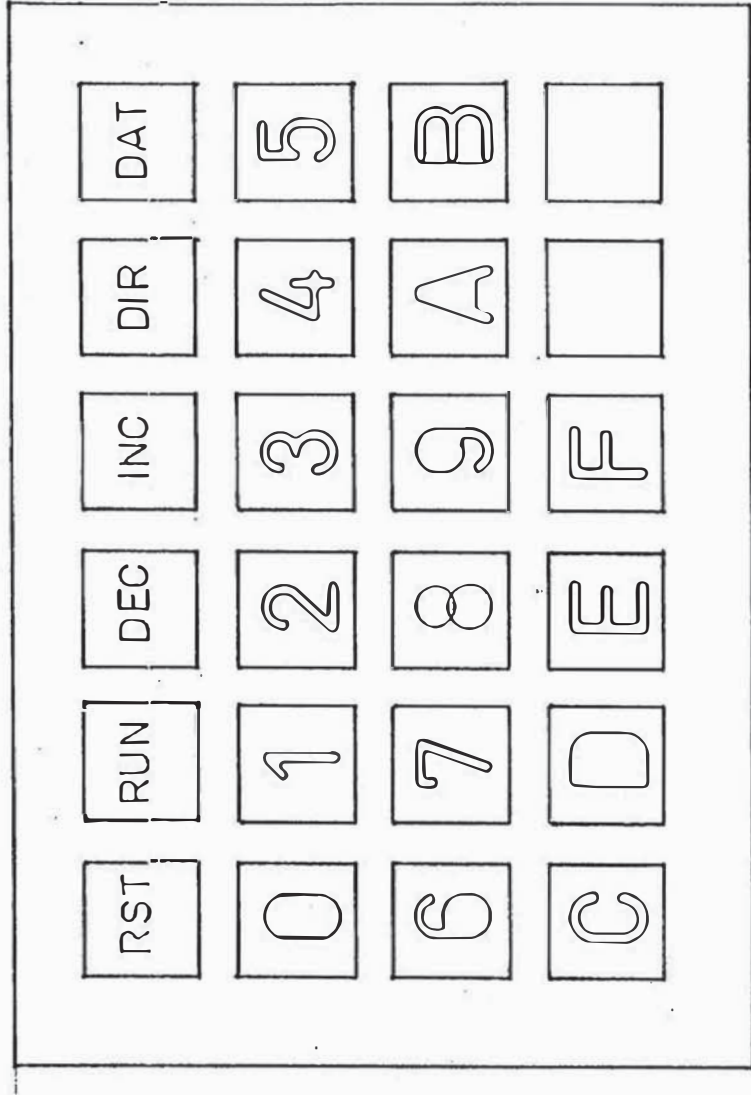
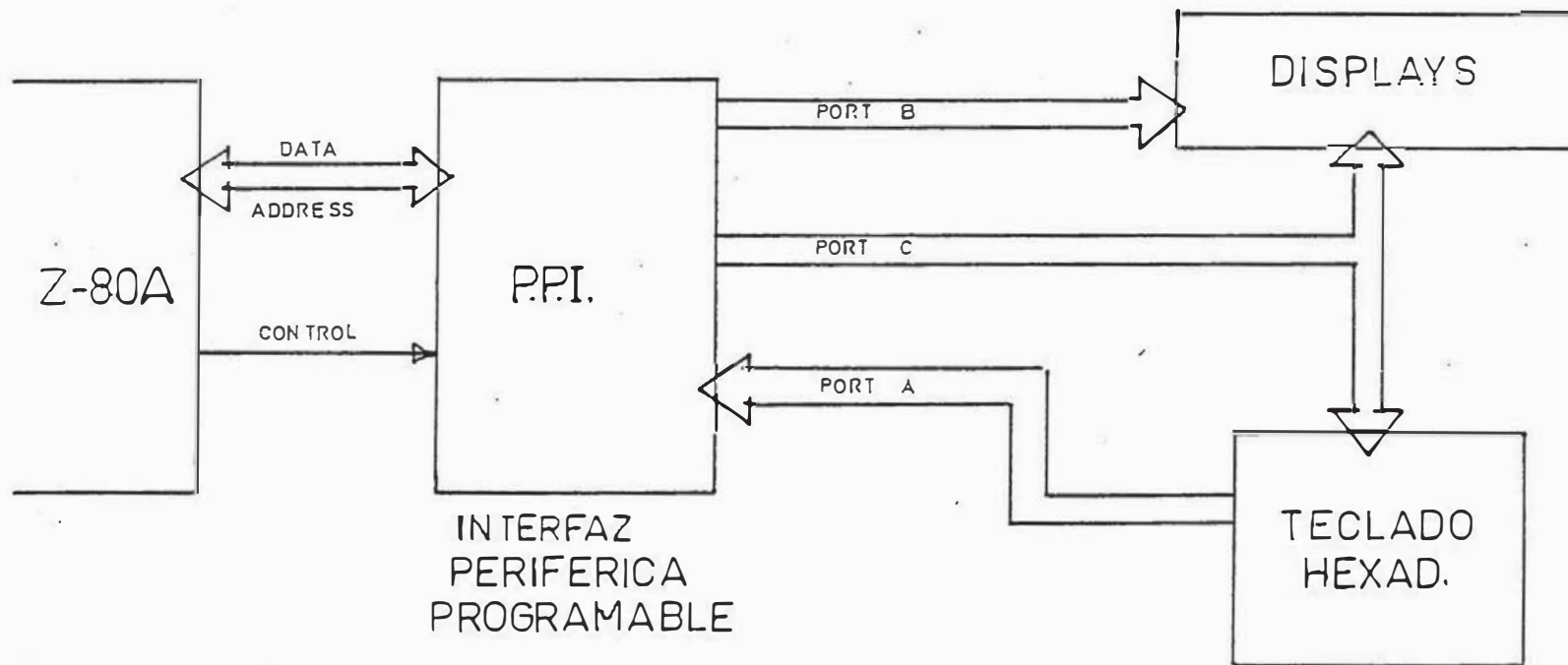
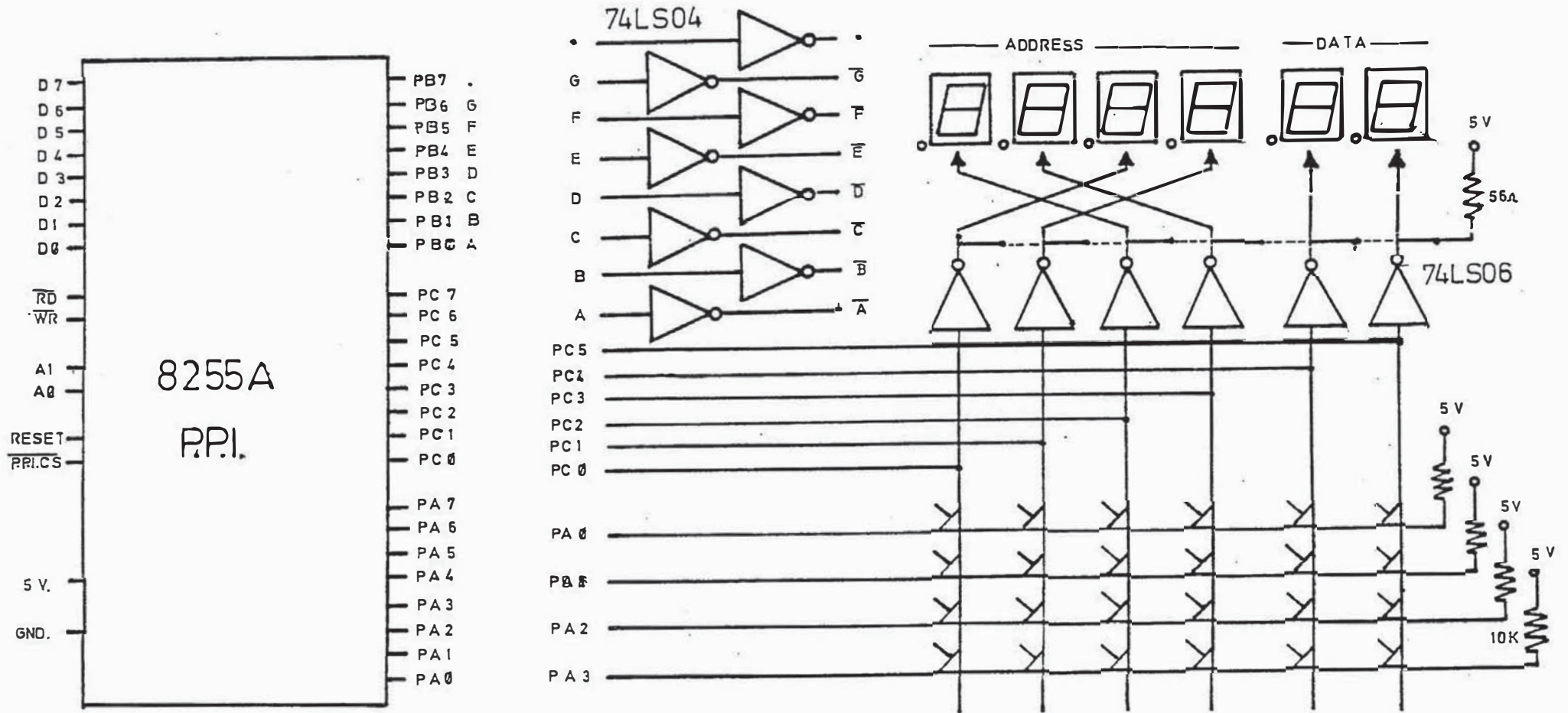


DIAGRAMA DE BLOQUES - TECLADO Y DISPLAYS



INTERFAZ A TECLADO Y DISPLAYS



CAPITULO IV

INTERFAZ A COMPUTADOR O TERMINAL INTELIGENTE

4.1 Generalidades

Un computador en general puede transmitir o recibir información por medio de un canal de comunicaciones, comúnmente llamado port, los computadores pueden atender a 4, 8, 16, 64, 128 o mas terminales inteligentes. El control de cada terminal esta a cargo de un programa de atención, llamados procesos, así, si por ejemplo 27 terminales desean comunicarse con el computador, cada uno conectado al computador independientemente, se crearan 27 procesos.

Estos 27 procesos son atendidos por el procesador central, cuanto mas procesos esten activos, mas recursos del computador se estarán utilizando (mas tiempo del C.P. U., mas memoria).

El procesador central interactua con los terminales por medio de tarjetas controladoras (hardware), cada tarjeta puede soportar 8 ó 16 terminales, estas tarjetas pueden usar señales de control de MODEM como no usarlas, otras tarjetas solo usan tres hilos por canal para su interacción con los terminales, realizando el control de caracteres enviados o recibidos por medio de caracteres de control, conocidos con el nemonico DC1 y DC2 (mas conocidos en el ambiente de computo como XON y XOFF respectivamente).

El computador puede enviar caracteres a el terminal a distintas velocidades, estas pueden ser:

110	Baudios	2400	Baudios
150	"	3600	"
300	"	4800	"
600	"	7200	"
1200	"	9600	"

1800 Baudios
2000 Baudios

19200 Baudios

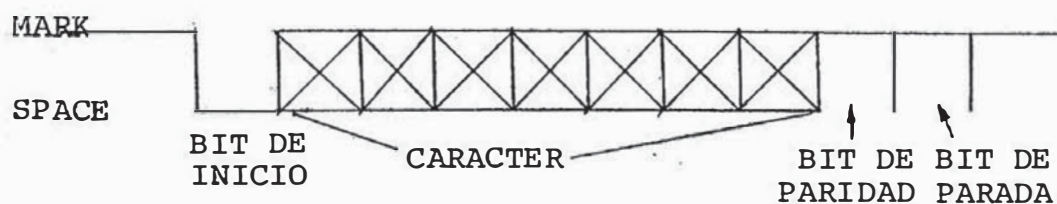
La transmisión y recepción pueden ocurrir a la vez (mientras se transmite un caracter, se esta recibiendo otro caracter), llamandose a esta manera de transmisión, full duplex. Normalmente la seleccion de velocidad es por software lo cuál permite el fácil manejo del port (Las características del port tales como paridad, números de bit de stop, si se habilitan o no las señales de control de MODEM, etc. son seleccionados desde la consola del computador por medio de programas).

La transmisión puede ser sincrona ó asincrona, en este trabajo se utiliza solamente la transmisión asincrona.

Una transmisión sincrona se caracteriza por la transmisión de caracteres en BLOQUES, cada bloque consta de varias partes básicas.

Primero se envian caracteres que informan acerca del bloque que se esta enviando luego los caracteres que forman el mensaje uno tras otro y por último caracteres de control que sirven para chequear si los caracteres han llegado correctamente tal como se enviaron o no.

Una transmisión asincrona se caracteriza por la transmisión caracter por caracter de la información. Cada caracter es enviado acompañado de bits que permiten sincronizar el inicio y parada del caracter mas un bit llamado de paridad.



Se acostumbra llamar al nivel uno lógico, con el nombre MARK y el nivel cero lógico, con el nombre SPACE.

Por caracter, se utiliza un sólo bit de inicio que es reconocido por un SPACE, luego llega el caracter con formado por 8 ó 7 bits y luego llega al bit de paridad que es opcional y por último el bit de parada que es reconocido por un MARK. Entre caracter y caracter puede no enviarse ningún mensaje por lo que el nivel se mantiene en MARK.

Ahora, supongamos que el computador esta enviando caracteres en form asincrona hacia un terminal. Normalmente la velocidad de procesamiento de un computador es mayor que la velocidad de tratamiento de un periférico. Por lo que existe la posibilidad de que el terminal pierda caracteres. Los puede perder porque mientras al terminal esta displayando "un caracter", a el la pueden estar llegando "dos caracteres". Bueno una solución es utilizar un terminal que permita almacenar los caracteres recibidos para luego mostrarlos. Otra solución es añadir hilos de control que la indiquen al computador, mediante cambios de nivel, que ya no envíe mas caracteres por el momento ó usar una combinación de Buffer mas hilos (de control , nivel "0" no envíe caracteres por el momento). Además de estos metodo existe el metodo de control por software usando los caracteres de control DC1 y DC2. Si el computador esta enviando caracteres (por el hilo de transmisión) el puede recibir simultáneamente (por el hilo de recepción) un caracter de control DC2 que la pide que deje de enviar caracteres, luego que el terminal este listo para seguir recibiendo, este puede enviar hacia el computador el caracter de control DC1 que le indica listo para recibir caracteres. La ventaja de este método es que no interesa si el terminal utiliza Buffer (pequeña cantidad de memoria) o no y además no utiliza ningún hilo adicional.

Este diseño utiliza este último método de control de caracteres, por software.

Algunos terminales inteligentes transmiten y reciben asincrónicamente. Estos constan de un teclado (ASCII) un monitor y una lógica de control. Este último le da ca pacidad autónoma de control.

El terminal recibe un caracter en código ASCII, le hace un tratamiento a este caracter y lo presenta por el monitor. Además de llegar caracteres ASCII pueden también llegar caracteres de control.

Algunos caracteres de control:

0EH	Empezar a encender intermitentemente los caracteres que se impriman.
0FH	Deja de encender intermitentemente los caracteres siguientes.
14H	Subraya los caracteres mostrados
15H	Deja de subrayar los caracteres mostrados siguientes .
1EH, 44H	Muestra los caracteres en iluminación reversa.
1EH, 45H	Deja de mostrar los caracteres <u>siguien</u> tes en iluminación reversa.
1CH	Disminuye la intensidad de muestra de los caracteres.
1DH	Deja de disminuir la intensidad de <u>mues</u> tra de los caracteres.

Ejemplo: Supongamos que el computador envíe al terminal esta serie de caracteres 0DH, 0AH, 0AH, 20H, 20H, 44, 41, 54, 41, 20H, 14H, 4CH, 49H, 4EH, 4BH, 15H, 2CH, 20H, 1EH, 44H, 32H, 33H, 3AH, 31H, 34H, 1EH, 45H, 20H, 2AH.

En la pantalla se mostrara lo siguiente:

DATA LINK, 23:14 *

El terminal envia caracteres por medio del port de comunicaciones haciendo uso del teclado. (port por el cual recibe información desde el computador). El teclado esta bajo el control de la lógica de control. Cuando el terminal esta interactuando con el computador, se dice que esta trabajando en Línea; en este caso todo caracter enviado por el teclado va directamente hacia el computador por medio del port de comunicaciones. Si el computador devuelve el mismo caracter hacia el terminal esta ultima es mostrada en el monitor y recién podrá apreciar Ud. el caracter que presiono en el teclado (se dice que se ha habilitado el eco del computador).

Otra manera de usar el teclado es usar el terminal fuera de línea (no interacción con el computador), así

de esta manera todo caracter que sea presionado en el teclado será mostrado directamente en el monitor del terminal.

La interfaz inteligente diseñada, tiene un port para comunicación asincrona el cual puede ser conectado al canal de un computador, para que así al computador pueda acceder bidireccionalmente a la linea telex o puede conectarse a un terminal inteligente, para que así el terminal inteligente pueda acceder también bidireccionalmente a la línea telex.

El port del interfaz inteligente recibe o transmite caracteres ASCII, la velocidad del canal es seleccionado por programa, escogiendose para las pruebas trabajar a 4,800 baudios, ademas se trabaja con caracteres de 7 y 8 bits sin bit de paridad, se utilizó 2 bits de parada, parada, para así poder trabajar indistintamente con 7 ó 8 bits por caracter. El control de caracteres se realiza utilizando los caracteres de control DC1 y DC2 (XON, XOFF respectivamente), los cuales evitan la pérdida de caracteres si la interfaz inteligente esta muy ocupada.

Entiéndase por interfaz inteligente a todo el conjunto de interfaces más lógica programada de control.

4.2 Circuito usado como interfaz

La interfaz inteligente usa como interfaz al computador o terminal de computador un circuito integrado de código Z-80A-SIO/1 de 40 pines. Esta es un circuito altamente programable, tiene la capacidad de manejar dos ports (dos canales) con las características que uno desee (según programación) y se conecta al procesador central por medio de un bus de datos mas unas señales de control. Cabe recordar que los dos ports son totalmente independientes. Para este caso, solo usamos un port (port "A").

El circuito necesita un generador de señales, estas velocidades estarán dados en baudios, será este generador quien de la velocidad de transmisión o recepción.

El generador (BAUD RATE) usado es un circuito integrado de código 8116 de 18 pines el cual necesita un cristal de 5.0688 MHz ó de 4.9152 MHz para generar sus ondas cuadradas. Este circuito también es programable para así generar la velocidad en baudios deseada.

El port "A" del circuito SIO se conecta a la línea del computador o a la línea del terminal inteligente por medio de unos circuitos cambiadores de nivel.

Los C. Integrados cambiadores de nivel son el MC1488 que pasa los niveles T.T.L. a niveles E.I.A. y el MC1489 que cambia los niveles E.I.A. a niveles T.T.L..

Las líneas hacia el lado del procesador están divididos en 3 grupos.

- a) Bus de datos (8 líneas)
- b) Control desde el procesador
- c) Control de cadena de interrupción

Las líneas hacia el port A ó B están divididos en 2 grupos.

- d) Líneas de canal
- e) Líneas de control de Modem

- a) Bus de datos: Por este bus se transmite bidireccionalmente la información enviada o recibida, entre el procesador y el computador o terminal. Además por este bus se envían las palabras de control que programan el SIO.
- b) Control desde el procesador: Se refiere al grupo de señales como, M1, RESET, CE, IORQ, RD, C/D, B/A. Aquí - C/D significa que si el nivel es bajo entonces en el bus hay un dato, si el nivel es alto, en el bus hay una palabra de control. Con respecto a B/A si el nivel

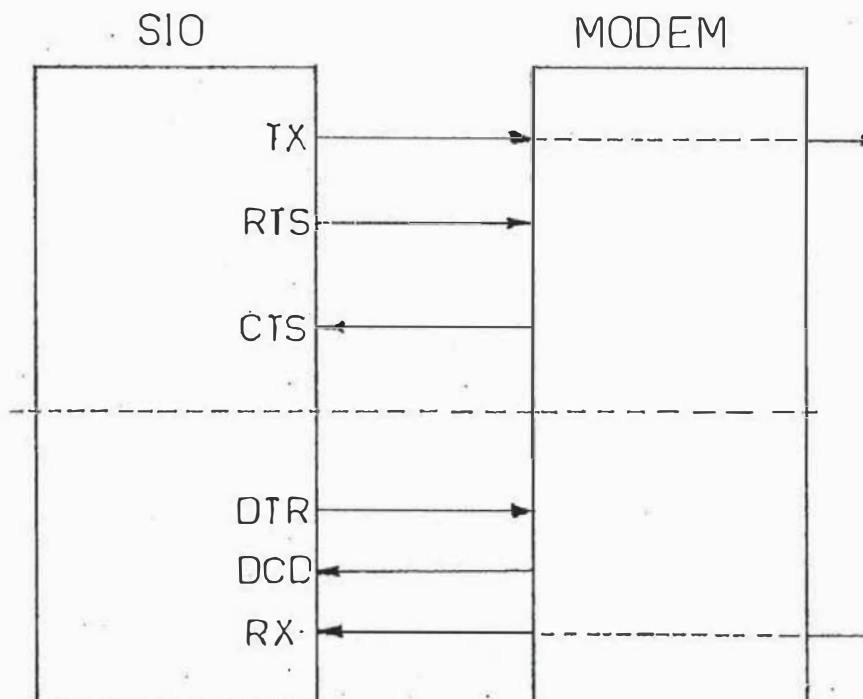
es bajo entonces el bus de datos se enlaza con el port A; de lo contrario con el port B. Normalmente se conecta el bit A1 del bus de direcciones al pin indicado con C/D y el bit \emptyset con el pin B/A.

- c) Control de cadena de interrupción: Lo conforman las líneas INT, IEI, IEO.

Si INT baja su nivel, significa que el SIO requiere atención, (señal de interrupción) IEI es usado con IEO para formar una cadena de interrupción con prioridades, cuando hay mas de un dispositivo que interrumpe. Un nivel alto en este pin indica que el dispositivo de mas alta prioridad no requiere servicio del procesador. IEO esta a nivel alto si IEI esta a nivel alto y el SIO no requiere servicio.

- d) Lineas del canal: Tenemos las líneas de transmisión Tx, recepción Rx, frecuencia de Rx, frecuencia de Tx.

- e) Líneas de control de Modem: Lo conforman las señales RTS, CTS, DTR, DCD.



La programación usa una serie de comandos que inicializan el modo básico de operación. Por ejemplo, el modo asincrono, longitud de caracter, clock rate, número de bits de parada, paridad par o impar, modo de interrupción finalmente la habilitación de recepción y transmisión.

Ambos canales cuentan con registros que pueden ser programados por el programa principal.

Estos registros pueden ser de escritura o de lectura. Los registros de lectura me indican los estados de transmisión; como, existe un dato en el port o SIO listo para recibir datos del procesador. También me indica si hay error de paridad, framing error. El registro 2 me da el vector de interrupción utilizado. Los registros de escritura son 8 para el canal B y 7 para el canal A. El registro WR2 contiene el vector interrupción que es exclusivo del canal B (este vector lo utilizan ambos port A y B). Con excepción del registro WRO, la programación de los otros registros requieren dos bytes. El primer byte es para el WRO y contiene tres bits (D0-D2) que seleccionan el registro, el segundo byte contiene la palabra de control que es escrita en el registro seleccionado para configurar el SIO.

4.3. Interacción del procesador con el computador o terminal inteligente

Inicialmente para aprender como programar el SIO realizo varios programas de prueba, empezando a programarlo en su modo mas simple, que es el modo muestreo en comunicación asincrona, comunmente llamado modo polling (término muy usado). Luego que domine la programación de ese modo, comenze a realizar pruebas de programación usando el modo de interrupción.

Por lo tanto, para deshabilitar el modo interrupción (y trabajar en el modo polling), para deshabilitar la paridad, utilizar 2 bits de parada, usar el clock mode standard, transmitir y recibir a 7 bits por caracter, tenemos que programar el SIO cargando los registros antes mostrados.

WR0	con	18H
WR1	"	00H
WR4	"	4CH
WR3	"	41H
WR5	"	A8H

Para lo cual creamos el programa siguiente llamado SIOPOLLINGA:

PROGRAMA DE PROGRAMACION DEL SIO POLLING

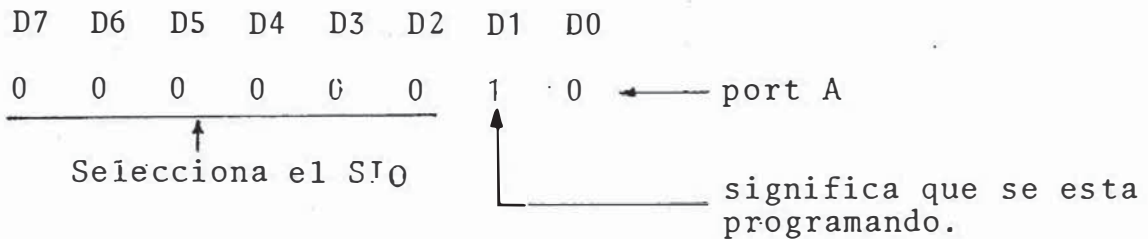
```
1E50 SIOPOLLINGA : LDC, SIOCTRLA      0E 02
1E52                LDB, PROGRAM      06 09
1E54                LD HL, TABLA PRG.  21 60 1E
1E57                OUTIR              ED B3
1E59                LDA, 4,800 BAUDIOS 3E 0C
1E5B                OUT (04H), A       D3 04
1E5D                RET                C9
1E60 TABLA PRG.    : DEFB              18
1E61                DEFB              01
1E62                DEFB              00
1E63                DEFB              04
1E64                DEFB              4C
1E65                DEFB              03
1E66                DEFB              41
1E67                DEFB              05
1E68                DEFB              A8
```

Este programa hace uso de la instrucción "OUTIR"; toma el contenido de la dirección de memoria indicada por el par de registros HL y lo muestra por el port indicado

por el byte selector almacenado en el registro C. Además esta instrucción automáticamente incrementa en una unidad el registro HL, decrementa en una unidad el registro B y todo esto se repite si B ≠ 0.

OUTIR : OUTPUT, INC HL, DEC B, SE REPITE SI B ≠ 0

Como hallar el byte selector para programar el port A:



entonces N = 02H, y como esta instrucción requiere el byte selector cargado en el registro C, usamos la instrucción LD:

LD C, SIOCTRLA OE 02

Luego notemos que se necesita enviar desde el procesador hacia el SIO 9 bytes.

- 18 RESET al canal A
- 01 Seleccione el registro 1
- 00 Valor que se carga al registro 1
- 04 Seleccione el registro 4
- 4C Valor que se carga el registro 4
- 03 Seleccione el registro 3
- 41 Valor que se carga al registro 3
- 05 Seleccione el registro 5
- A8 Valor que se carga al registro 5

Estos bytes son enviados uno tras otro hacia el SIO los cuales programaran el SIO. La instrucción enviará los bytes contenidos en la tabla TABLA PRG. cuya dirección de inicio es 1E60H los cuales son 9.

Y así utilizamos las siguientes instrucciones.

```
LD B, PROGRAM          0609
LD HL, TABLA PRG       21 60 1E
```

entonces OUTR se encargara de enviar los 9 bytes cargados en la tabla cuya dirección inicia en 1E60H hacia el control del port A direccionado así por el byte selector N=02.

Por lo tanto ya tenemos un programa de programación en polling del SIO.

Ahora ya podemos desarrollar un programa de prueba pequeño. Este programa va ha mostrar en la pantalla del terminal la siguiente frase, repetidas veces.

```
"ESTA ES LA PRUEBA DEL SIO  *"  
on reverse
```

El programa de prueba es el siguiente:

```
1E37 PRUEBA : CALL STOPOLLINGA      CD 50 1E
1E3A      ♂ : LD HL, 1E15H          21 15 1E
1E3D      LD B, LONGITUD DE LINEA  06 20
1E3F      LD C, PORT DE SALIDA     0E 00
1E41      ♂ : IN A, (SIOCTRLA)     DB 02
1E43      BIT 2, A                 CB 57
1E45      JRZ, ♂                    28 1A
1E47      OUTI                      ED A3
1E49      JRZ, ♂                    28 1E
1E4B      JR, ♂                      1B 1A
1E15H     45H                       H
1E16     53                          S
1E17     54                          T
1E18     41                          A
1E19     20                          ESPACIO
1E1A     45                          H
1E1B     53                          S
1E1C     20                          ESPACIO
1E1D     4C                          L
```

1E1E	41	A
1E1F	20	ESPACIO
1E20	50	P
1E21	52	R
1E22	55	U
1E23	45	E
1E24	42	B
1E25	41	A
1E26	20	ESPACIO
1E27	44	D
1E28	45	E
1E29	4C	L
1E2A	20	ESPACIO
1E2B	1E	CARACTERES DE CONTROL PARA INICIAL VIDEO INVERSO
1E2C	44	
1E2D	53	S
1E2E	49	I
1E2F	4F	O
1E30	1E	CARACTERES DE CONTROL PARA FINALIZAR VIDEO INVERSO
1E31	45	
1E32	20	ESPACIO
1E33	2A	*
1E34	0A	NUEVA LINEA

El programa primeramente usa la instrucción "CALL" para llamar a la subrutina "SIOPOLLINGA", que programara el port A del SIO. Luego utilizara la instrucción OUTI para mostrar por el port A los 32 bytes (20H en hexadecimal) almacenados a partir de la dirección 1E15H.

LD HL, 1E15H	21 15 1E	carga en "HL" la dirección de inicio.
LD B, 20H	06 20	carga en "B" la longitud de la frase.
LD C, 00	0E 00	carga en "C" el byte selector del port A.

Las siguientes instrucciones se encargarán de mostrar en cada línea una frase completa.

⌘ : IN A, (SIOCTR LA)
BIT 2, A
JRZ, ⌘

"⌘", es una subrutina muy importante; el procesador es mucho mas rápido que el terminal por lo cual si se envían todos los caracteres en una sola vez hacia el terminal, este no mostraria nada legible ya que habría superposición de caracteres. Para esto el SIO recibe caracteres solo si el terminal ya puede mostrar otro mas, esto se consigue accedando al registro RRO mediante la instrucción IN y luego preguntando por el bit D2 del byte leído, indicando, si es uno, que el SIO puede recibir otro caracter para ser mostrado y si es cero que el SID no puede recibir ningún caracter mas ya que su buffer de tres bytes esta lleno; paralelamente el SIO esta enviando los caracteres almacenados en su buffer pero ni bien haya un lugar libre en su buffer, el SIO cargará el bit D2 del byte del registro RRO con un uno LOGICO, indicando al procesador que ya puede enviar otro caracter al SIO para luego ser mostrado. Y así este programa entra en un lazo repetitivo comunmente llamado LOOP, entonces se dice que el programa entro en un LOOP.

PROGRAMAS DE PRUEBA USANDO INTERRUPCION

Primero programaremos al SIO port A en el modo interrupción; para lo cual se tendrá que cargar los registros WR0, WR4, WR3, WR5, WR1 del canal A y los registros WR0, WR2, WR1 del canal B.

Los Registros del Canal A

WR0	18H	RESET el canal A
WR4	4CH	2 bits de parada, paridad deshabilitada, x16 clock mode.
WR3	41H	7 bits por caracter de Rx, Rx habilitada
WR5	A8H	7 bits por caracter de Tx, Tx habilitada señal DTR habilitada
WR1	1CH	Habilita interrupción de Rx, Tx simple

queda habilitada.
06 Habilita interrupción de Tx, Rx simple queda habilitada.
1EH habilita interrupción de Rx y Rx.

Los Registros del Canal B

WRO 18H RESET al canal B
WR2 00H Vector interrupción
WR1 04H El vector interrupción quedará afectado por el tipo de interrupción. (Sea de Tx ó Rx).

Canal A

Canal B

WRO	18A	WRO	18H
WR4	4CH	WR2	00H
WR3	41H		
WR5	A8H		
WR1	1CH, 06H ó 1EH		

Para programar el port A en el modo interrupción, se desarrollo el siguiente programa: (notar que es necesario usar los registros del port B).

PROGRAMA DE PROGRAMACION SID (INTERRUPCION)

1E93	SIOINT	:	LDC, SIOCTRLA	0E 02
1E95			LDB, PROGRAM-A	06 09
1E97			LDHL, TABLA PRG	21 B0 1E
1E9A			OUTIR	ED B3
1E9C			INC C	OC
1E9D			LDB, PROGRAM-B	06 05
1E9F			OUTIR	ED B3
1EA1			LDA, 4, 800 BAUDIOS	3E OC
1EA3			OUT (04H), A	D3 04
1EA5			LDA, 1FH	3E 1F
1EA7			LDI, A	ED 47
1EA9			IM2	ED 5E

1EAB	EI	FB
1EAC	RET	C9
1EB0	DEFB	18
1EB1	DEFB	14
1EB2	DEFB	4C
1EB3	DEFB	13
1EB4	DEFB	41
1EB5	DEFB	15
1EB6	DEFB	A8
1EB7	DEFB	11
1EB8 (*)	DEFB	1C ó 06 ó 1E
1EB9	DEFB	18
1EBA	DEFB	12
1EBB	DEFB	00
1EBC	DEFB	11
1EBD	DEFB	04

Notar que solo estamos programando el port A

Los registros WRO, WR4, WR3, WR5 se cargan de igual manera que cuando programamos el port A en polling. Además de estos registros se carga un nuevo registro WR1 el cual permite habilitar las interrupciones ya sea solo de Tx ó solo de Rx ó ambos.

Una interrupción de Tx, se entiende así: Cuando el programa muestra un byte por el port seleccionado, haciendo uso de la instrucción OUT se produce una interrupción de transmisión, tomando el control del procesador la subrutina de interrupción de transmisión que esta direccionada por el emsamblado del registro I y el vector interrupción afectado.

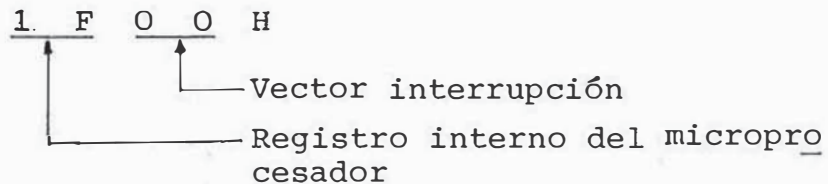
Una interrupción de Rx, se entiende así: cuando el SIO recibe un caracter, se produce una interrupción, tomando el control del procesador la subrutina de interrupción

ción de recepción que esta direccionada por el ensamblado del registro I y el vector interrupción afectado.

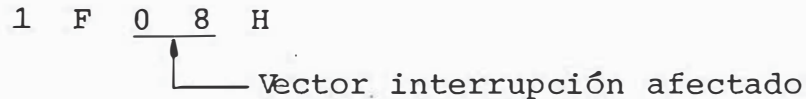
Veamos que se entiende por vector interrupción afectado.

Tenemos por el programa anterior que I vale 1FH y el vector interrupción vale 00H (WR2 del canal B)

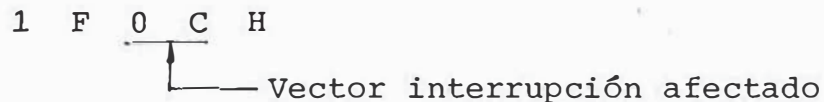
Entonces ensamblando tenemos:



Si ocurre una interrupción de Tx el vector interrupción queda afectado tomando el valor 08H y tenemos:



Si ocurre una interrupción de Rx el vector interrupción queda afectado tomando el valor 0CH y tenemos:



Así en estas direcciones almacenamos las direcciones de las subrutinas de interrupción de Tx e interrupción de Rx respectivamente.

1F08	40	2040H	Dirección inicial de la subrutina de interrupción de transmisión.
1F09	20		
1F0C	70	2070H	Dirección inicial de la subrutina de interrupción de recepción
1F0D	20		

Notar la analogia con un vector.



Recordemos el registro WR1 del canal A nos permite seleccionar que tipo o tipos de interrupción deseamos habilitar. En el programa de programación del SIO (interrupción) notamos que el registro WR1 toma el byte almacenado en la dirección 1EB8H.

Como deseamos realizar experimentos podemos desarrollar tres programas pequeños que permitan habilitar las interrupciones que deseamos.

Aquí los programas:

a) Programación del SIO canal A usando interrupción de Rx y Tx simple.

```
1E7A  INT.Rx      : LD A, INT DE RX      3E 1C
1E7C                LD (1EB8H), A      32 B8 1C
1E7F                JR SIO INT        18 12
```

b) Programación del SIO canal A usando interrupción de Tx y Rx simple.

```
1E81  INT.Tx     : LD A, INT DE TX      3E 06
1E83                LD (1EB8H), A      32 B8 1E
1E86                JR SIO INT        18 0B
```

c) Programación del SIO canal A usando interrupción de Tx e interrupción de Rx.

```
1E88  INT.Rx.Tx : LDA, INT Rx, Tx     3E 1E
1E8A                LD(1EB8H), A      32 B8 1e
1E8D                JR SIO INT        18 04
```

Como ya podemos programar el SIO usando interrupciones.. Se desarrollaron dos programas de prueba, uno usa in

terrupción de Recepción (Rx) y el otro usa interrupción de Transmisión (Tx).

Programa de Prueba que usa Interrupción de Recepción

2050	PRUEBA INTRx	:	CALL INT.Rx	CD 7A 1E
2053	LOOP	:	NOP	00
2054			NOP	00
2055			NOP	00
2056			JR LOOP	18 FC

Cuando ocurre una interrupción de recepción por la llegada de un caracter al port A del SIO, se tiene el vector siguiente.



- a) 1FOCH es la direccion en samblada que contiene la direcci3n 2070H.
- b) 2070H es la direcci3n de la subrutina de atenci3n en la interrupci3n de recepci3n.

SUBROUTINA DE ATENCION A LA INTERRUPCION DE RECEPCION

2070	PUSH AF	F5
2071	PUSH HL	ES
2072	IN A, (SIO DATA A)	DB 00
2074	CP CR	FE 8D
2076	JPZ, 0000H	CA 00 00
2079	LD HL, (RBuffPtr)	2A 05 1F
207C	LD(HL), A	77
207D	INC HL	23
207E	LD (RBuffPtr), HL	22 05 1F
2081	POP HL	E1
2082	POP AF	F1
2083	EI	FB
2084	RET I	ED 4D

Este programa usa dos direcciones de trabajo.

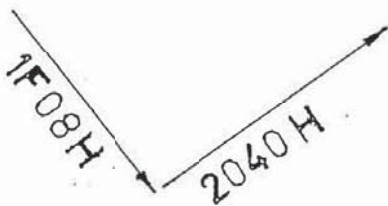
1F0CH	70H	1F05H	00H
1F0D	20	1F06	30

En conclusión, cada vez que llegue un caracter al port A del SIO, (cuando se este ejecutando el programa PRUEBA INT Rx), estos se almacenarán a partir de la dirección 3000H.

Programa de Prueba que usa Interrupción de Transmisión

2060	PRUEBA INT.Tx	:	CALL INT.TX	CD 81 1E
2063			LDA, NUEVA LINEA	3E OA
2065			BUT (SID DATA A), A	D3 00
2067	LOOP 1	:	NOP	00
2068			NOP	00
2069			NOP	00
206A			JR, LOOP1	18 FC

Cuando se termine de ejecutar la instrucción de salida "OUT", ocurre una interrupción de transmisión por la salida de un caracter por el port A del SID, teniendose asi el vector siguiente:



- a) 1F08H es la dirección ensamblada que contiene la dirección 2040H.
- b) 2040H es la dirección de la subrutina de atención a la interrupción de transmisión.

También, una vez cargado el buffer (3 bytes) del SIO por el programa de interrupción, este devuelve el control al programa LOOP1, paralelamente el SIO estara ocupado en viando caracteres almacenados en su buffer. Una vez que el SIO tenga lugar en su buffer para almacenar un caracter proveniente del procesador, el SIO interrumpira nuevamente.

Todo este intercambio de programas de control sucederá hasta que se termine de enviar todo el mensaje deseado.

Notar que la interrupción al programa LOOP 1 se produce en cualquier momento de ejecución de este programa. He allí la eficacia de este método de transmisión. Solo interrumpe cuando es necesario.

SUBROUTINA DE ATENCION A LA INTERRUPCION
DE TRANSMISION

2040		PUSH AF	F5
2041		PUSH DE	D5
2042		PUSH HL	E5
2043		PUSH BC	C5
2044		LD HL, (XBuffØPtr)	2A 03 1F
2047		LDA, (HL)	7E
2048		CP Nueva Línea	FE 0A
204A		JR NZ, SIGUE	20 06
204C		LDA, Reset int/perding	3E 28
204E		OUT (SIOActr1A),A	D3 02
2050		JR EXIT	18 03
2052	SIGUE	OUT (SIODATA A), A	D3 00
2054		INC HL	23
2055	EXIT	LD (XBuffØPtr), HL	22 03 1F
2058		POP BC	C1
2059		POP HL	E1
205A		POP DE	D1
205B		POP AF	F1
205C		EI	FB
205D		RET I	ED 4D

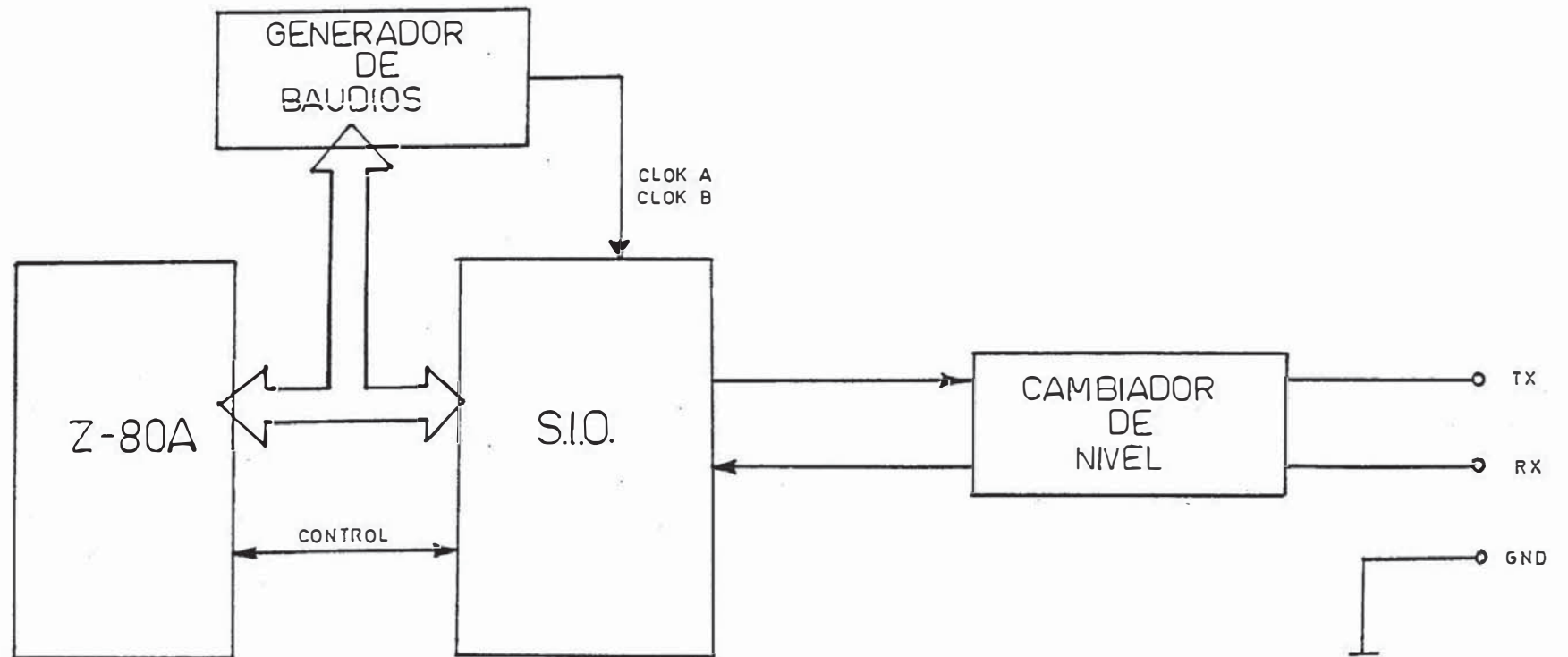
Este programa usa dos direcciones de trabajo:

1F08H	40H	1F03H	15H
1F09	20	1F04	1E

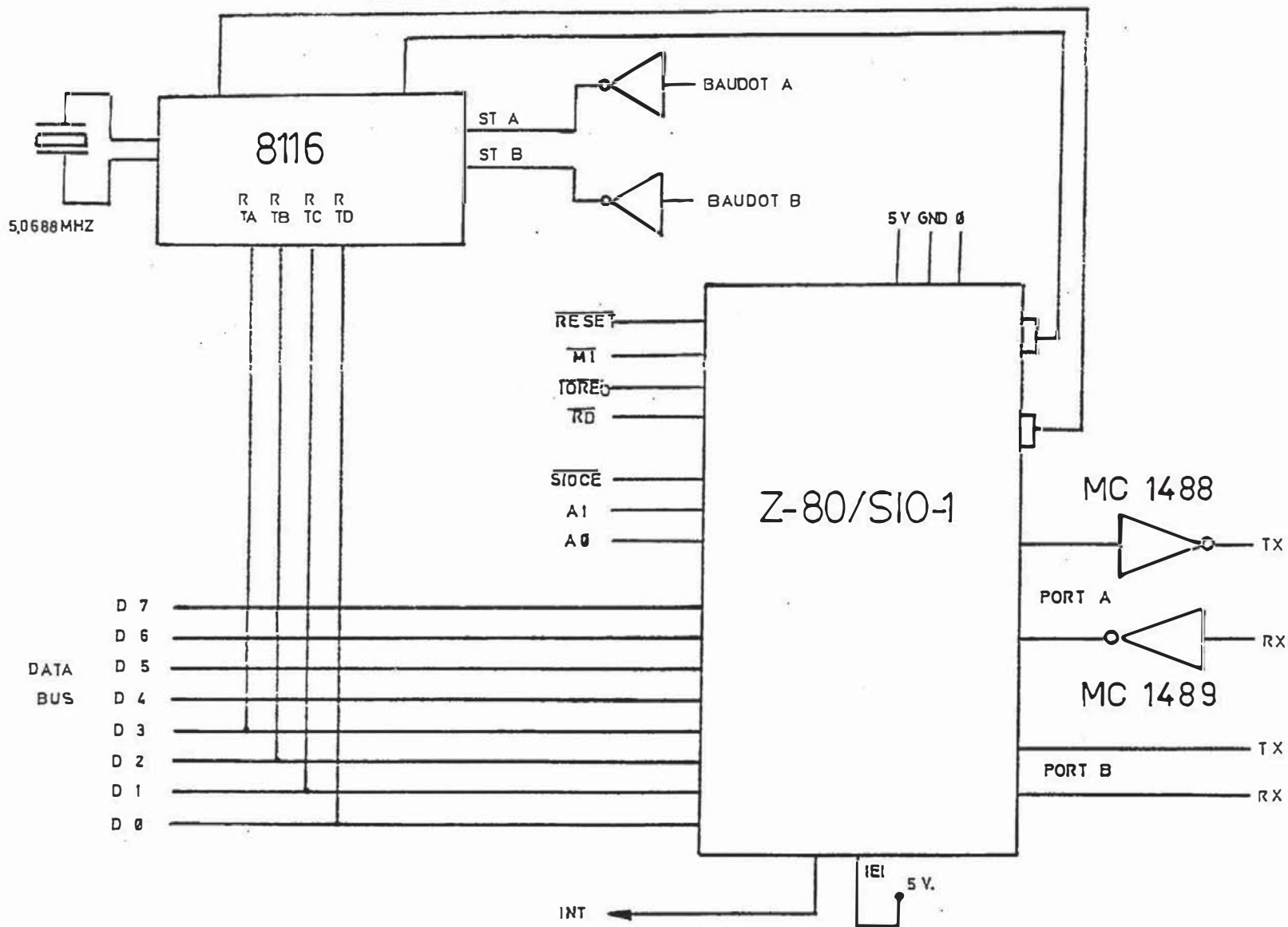
En conclusión, cada vez que se requiere enviar un caracter, la subrutina de interrupción tomara estos caracteres de la tabla que inicia con la dirección 1E15H y esto sucedera hasta encontrar el caracter NL (nueva linea). Que para nuestro programa de Prueba significa: fin del mensaje. Se pueden utilizar muchas maneras de finalizar, como por ejemplo utilizar contadores.

Estos programas de prueba ayudan notablemente al dominio de la programación, usando el modo interrupción.

DIAGRAMA DE BLOQUES - INTERFAZ A COMPUTADOR



INTERFAZ A COMPUTADOR



CAPITULO V

INTERFAZ A LINEA TELEX

5.1 Generalidades

El sistema telex es muy usado aquí en el Perú, gran cantidad de compañías lo consideran como uno de sus principales medios de comunicación para todo lo que es texto. Cabe recordar que esta red Peruana se inicio el año 1975 por primera vez, con un servicio a Nivel Nacional y con mas de 1,600 abonados en operación, actualmente se sigue ampliando esta capacidad de acuerdo a los requerimientos del país con la instalación de la nueva Central Internacional EDX.

El sistema telex en el Perú esta dividido en tres zonas (zona I, II y III). La zona II tiene una central nodal internacional ubicada en Lima, y las zonas I y III tienen cada una, una central nodal simple ubicadas en Trujillo y Arequipa respectivamente.

Este sistema es economico, el cual trabaja las 24 horas del día y no necesita un operador que manipule la recepción, ya que esto se hace por medio de un teleimpresor.

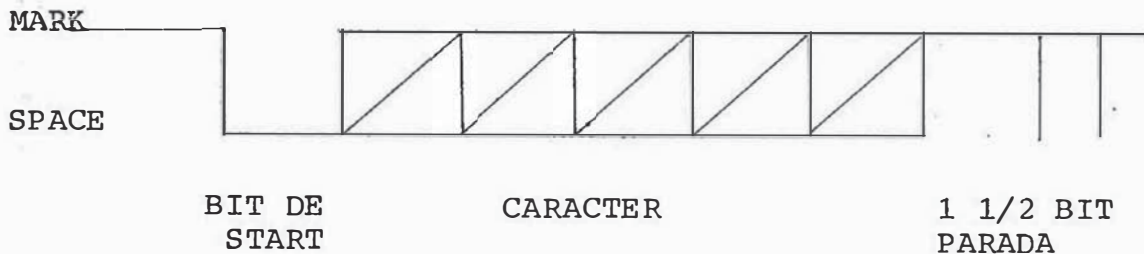
Cada teleimpresor puede comunicarse a la central de las siguientes formas:

- a) Corriente simple a 2 hilos
- b) Corriente doble a 2 hilos
- c) Corriente doble a 4 hilos

La forma escogemos en este diseño es la primera, corriente simple a 2 hilos por ser la mas usada en nuestro medio.

Para indicar el nivel uno lógico (MARK) se usa una corriente de 40 miliamperios y para el nivel cero lógico (SPACE) 0 miliamperios. La comunicación es asíncrona con las siguientes características:

- 5 bits por caracter
- Codigo BAUDOT para el caracter
- 1 bit de START (inicio)
- 1.5 bits de SPOP (parada)
- La velocidad de transmisión es 50 Baudios.



El código BAUDOT es el código usado tradicionalmente en transmisiones telex, vease al final del capítulo la tabla del código BAUDOT. Este código solo utiliza cinco bits para representar todo el alfabeto, número y signos que sumados dan 57. Estos 5 bits solo me permiten tener 32 combinaciones diferentes, pero la solución que se da es la siguiente.

Existen dos códigos que selecciona si el caracter es letra o simbolo; si el caracter es 1FH significa que los siguientes caracteres representan LETRAS, y si el caracter que precede e 1BH, entonces los siguientes caracteres representaran simbolos.

Por ejemplo: si tenemos la secuencia

- 1FH, 10H, 01H, 12H, 01H, 1DH, 1BH, 10H, 01H, 12H, 01H, 1DH.

Representará:

- LETRAS T E L E X SIMBOLOS 5 3] 3 /

En el teleimpresor se mostrara.
TELEX53)3/

Las señales de enrutamiento se muestran el final del capítulo.

Los experimentos se realizaron usando corriente simple, 2 hilos.

La línea en vacío da una tensión de 120 voltios. Cuando se conecta el teleimpresor a la línea esta tensión cae a 110 voltios consiguiendo una corriente de 5MA. Luego si se oprime en su teleimpresor el botón de llamada. se encenderá la lámpara de ese botón lo que indica que esta en condición de marcar el número que desea; en otras palabras la corriente cambia a 40MA (10 volt) y después de unos segundos aparece un pulso de 25 milisegundos de ancho (estrictamente, ovolt y oma); este pulso inicia la invitación a marcar el número de abonado. Luego se marca inmediatamente el número de abonados, (5 dígitos para un abonado del Perú) los pulsos del código de abonado son similares a los pulsos usados en telefonía con la diferencia que aquí trabajamos con corriente y en telefonía con voltajes.

Los pulsos de marcado tienen las mismas caracteristicas y son las siguientes:

MARK	39% del ciclo
SPACE	61% del ciclo
FRECUENCIA 10HZ	

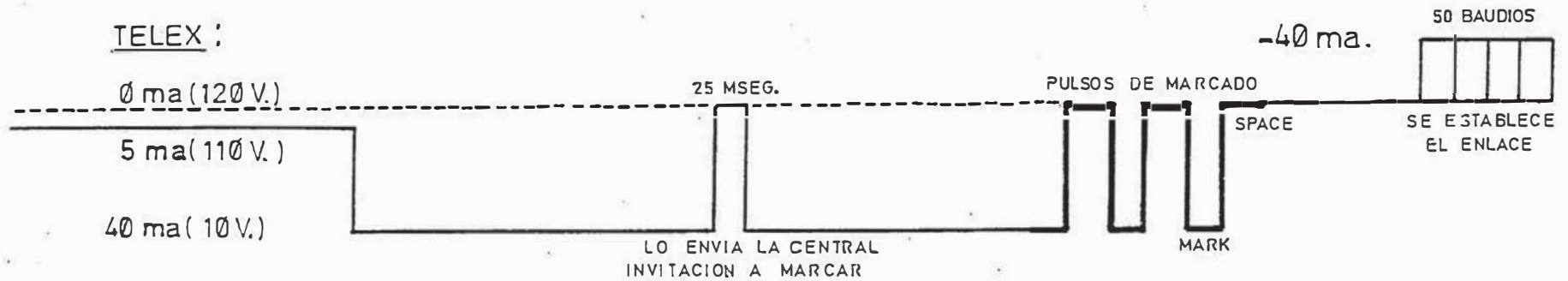
Estos diagramas se consiguieron experimentalmente, gracias a la ayuda del osciloscopio y el multímetro digital.

Además notar en el gráfico de señales que una vez conseguido el enlace, la corriente cambia de sentido to

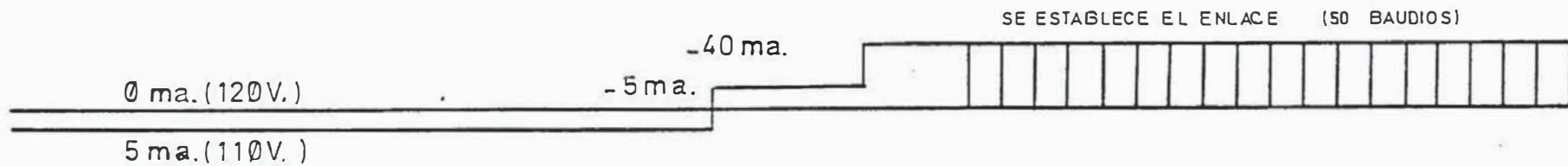
TELEFONIA:



TELEX:



RECONOCIMIENTO DE LLAMADA



mando el valor de -40mA y en ese momento todo mensaje enviado o recibido se realiza por comunicación asincrona y a una velocidad de 50Baudios usando el código RAU DOT.

5.2 Circuito usado como Interfaz

Al final del capítulo podemos apreciar el circuito usado como interfaz a la línea telex.

El acoplador óptico sensa la corriente de -40 miliamperios y sus variantes enviadas hacia el port B del SIO (pin de recepción), para que este transfiera el carácter en código BAUDOT hacia el microprocesador Z-80 por medio del bus de datos. Además este acoplador detecta el cambio de sentido de corriente (74LS374), para esto el procesador central ejecuta una instrucción IN que permite leer el nivel de la línea que es cargado en el bit 2 del acumulador (D2 del bus de datos). Esta detección del cambio de sentido de corriente indica que el enlace se ha conseguido.

Los switches son utilizados para "simular" los estados del teleimpresor.

<u>SWITCH A</u>	<u>SWITCH B</u>	<u>ESTADO</u>	<u>CORRIENTE</u>
ABIERTO	ABIERTO	TELEIMPRESOR	DESCONECTADO 0MA
ABIERTO	CERRADO	"	CONECTADO 5MA
CERRADO	ABIERTO	"	EN LINEA 40MA -40MA

Para conseguir la corriente de 5MA usamos una resistencia de 22 Kohmios. Para conseguir la corriente de +40 miliamperios utilizamos una resistencia de 220 ohmios.

El switch A permite generar los pulsos de selección de abonado, siendo controlado este switch por el procesador central a través del bit D0 del bus de datos. Además este switch permite generar los caracteres en código BAUDOT, siendo controlado ahora por el port B del SIO

(pin de transmisión).

El switch B es controlado por el procesador central a travez del bit D1 del bus de datos.

Las Bobinas de los switchs son actividades a travez de circuitos integrados open colector (driver 74LS06) y estos dejan pasar la señal de control conveniente del circuito integrado 74LS377 que graba los estados de los bits D0 y D1 del bus de datos.

5.3 Interacción del procesador con la Línea Telex

El procesador podrá enviar o recibir mensajes usando la línea TELEX. Los bits, en la línea estan en serie y los bits que el procesador trata deben estar en paralelo, para poder hacer este cambio de paralelo a serie y de serie a paralelo de los caracteres tratados, se utiliza el port B del SIO.

Para el diseño de esta interfaz se tuvo que investigar la transmisión de caracteres usando el port B del SIO. Para lo cual se realizo un programa de prueba que a continuación presentamos.

Programa del Port B del SIO:

Las características de este canal son las siguientes:

WR1	: No interrupción, activado el polling	00H
WR4	: Paridad no habilitada, 1 1/2 bits de stop, x16 clock mode	48H
WR3	: Recepción habilitada, 5 bits/caracter	01H
WR5	: Transmisión habilitada, 5 bits/caracter señal D.T.R. habilitada	88H

velocidad de transmisión 50 Baudios, recordar que WR1, WR4, WR3, WR5 con registros de programación del SIO, en este caso para el port B.

Aquí se muestra la subrutina que programara el port

B del SIO.

PROGRAMA DE PROGRAMACION SIO PORT B
(POLLING)

```

0600 SIOPOLLINGB : LDC, SIOCTRLB    OE 03
0602                LDB, PROGRAM    06 09
0604                LDHL, TABLA PRG. 21 0E 06
0607                OUTIR           ED B3
0609                LDA, 50 BAUDIOS  3E 00
060B                DUT (Band B)    C3 18
060D                RET             C9
060E TABLA PRG     : DEFB           18
060F                DEFB           01
0610                DEFB           00
0611                DEFB           04
0612                DEFB           48
0613                DEFB           03
0614                DEFB           01
0615                DEFB           05
0616                DEFB           88
    
```

Notar que el byte selector de control vale 03H

```

03H = 0 0 0 0 0 0 1 1
      |
      |----- B/A  seleccionado el port B
      |
      |----- C/D  seleccionado el control
    
```

El circuito integrado 8116 se programa usando 4 bits menos significativos del bus de datos.

D3	D2	D1	D0		D3	D2	D1	D0		
0	0	0	0	50	Baudios	1	0	0	0	1800 baudios
0	0	0	1	75	"	1	0	0	1	2000 "
0	0	1	0	110	"	1	0	1	0	2400 "
0	0	1	1	134.5	"	1	0	1	1	3600 "
0	1	0	0	150	"	1	1	0	0	4800 "

0	1	0	1	300	Baudios	1	1	0	1	7200	Baudios
0	1	1	0	600	"	1	1	1	0	9600	"
0	1	1	1	1200	"	1	1	1	1	19200	"

Para nuestro caso, lo conseguimos enviando el byte 00H para así conseguir programarlo a una velocidad de 50 Baudios

Notar en el diagrama (al final del capítulo) que este circuito generador tiene 2 salidas independientes, conectados al port A y el port B cada uno. La programación de velocidad de cada señal generada es independiente para lo cual se utilizan dos bytes selectores, que son los siguientes:

BAUD	SIOA	:	04H	Para port A
BAUD	SIOB	:	18H	Para port B

Bueno, ahora que ya hemos programado el port B del SIO; se puede ya utilizar el SIO para transferir datos almacenados en la memoria, enviándolos a través de la línea telex. Para esto se realizó el siguiente programa de prueba que permite enviar la siguiente "frase" repetidas veces a un teleimpresor ya enlazado:

"ENLACE AUTOMATICO DIGICOMP"

Esta frase lo recibe el teleimpresor enlazado repetidas veces uno tras otro y solo dejará de imprimir si uno de los dos, el interfaz inteligente o el teleimpresor receptor se desconecta.

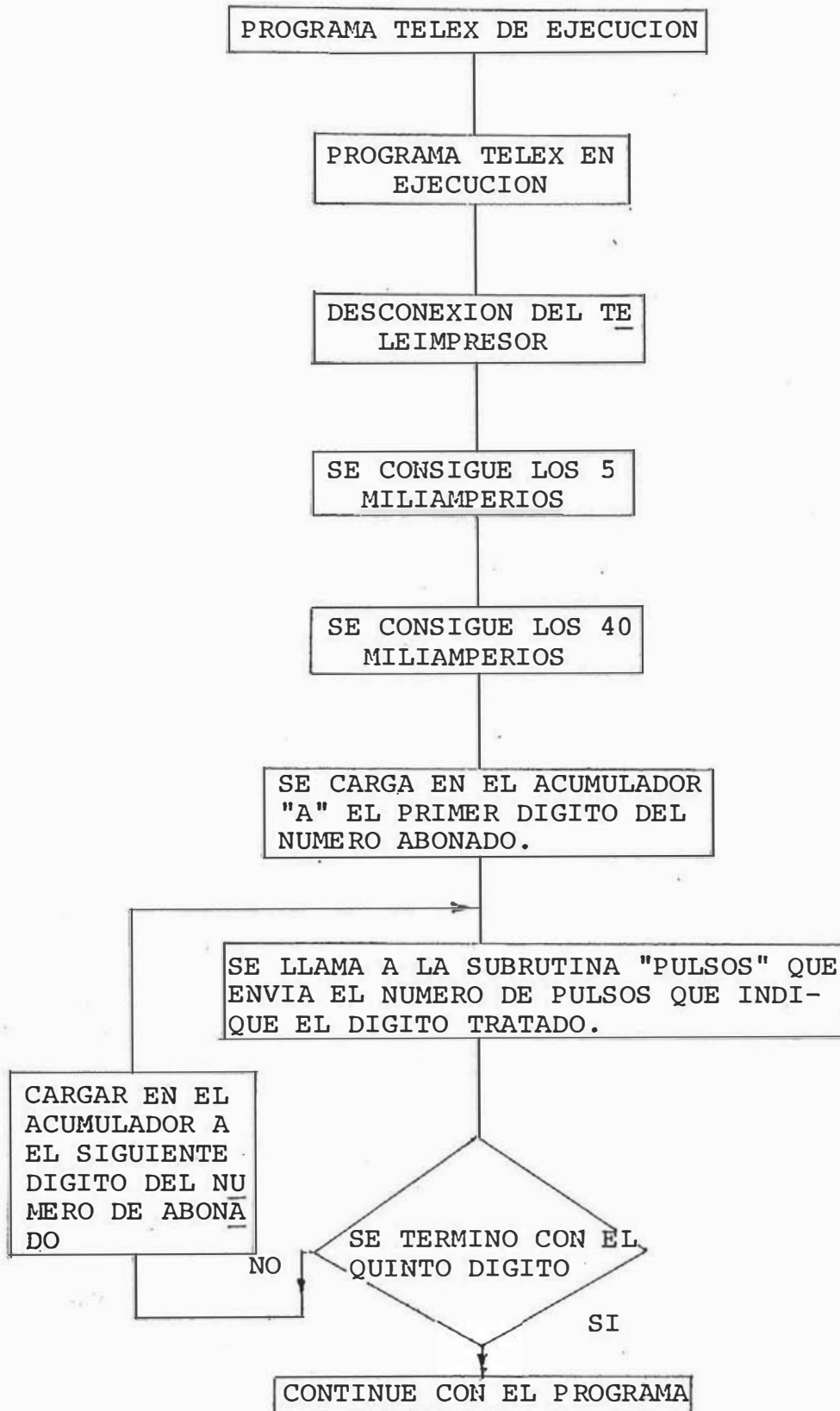
PROGRAMA DE PRUEBA EN POLLING (TELEX)

061A	CALL SIOPOLLINGA	CD 00 06
061D	α : LDHL, ARCHIVO	21 30 06
0620	LD B, LONGITUD DE LINEA	06 1D
0622	LD C, SIODATA B	0E 01
0624	⋈ : INA, (SIOCTRLB)	DB 03
0626	BIT 2, A	CB 57
0628	JRZ, ⋈	28 FA

062A	OUTI	ED A3
062C	JRZ, α	28 EF
062E	JR, ⸗	18 F4
0630	LETRAS	1F
0631	CR	08
0632	NL	02
0633	E	01
0634	N	0C
0635	L	12
0636	A	03
0637	C	0E
0638	E	01
0639	ESPACIO	04
063A	A	03
063B	U	07
063C	T	10
063D	O	18
063E	M	1C
063F	A	03
0640	T	10
0641	I	06
0642	C	0E
0643	O	18
0644	ESPACIO	04
0645	D	09
0646	I	06
0647	G	1A
0648	I	06
0649	C	0E
064A	O	18
064B	M	1C
064C	P	16

Aprovecharé esta sección para presentar el "programa que realiza la llamada automática a un abonado". El interfaz inteligente simulará los estados de conexión y desconexión del teleimpresor.

AQUI EL DIAGRAMA DE FLUJO

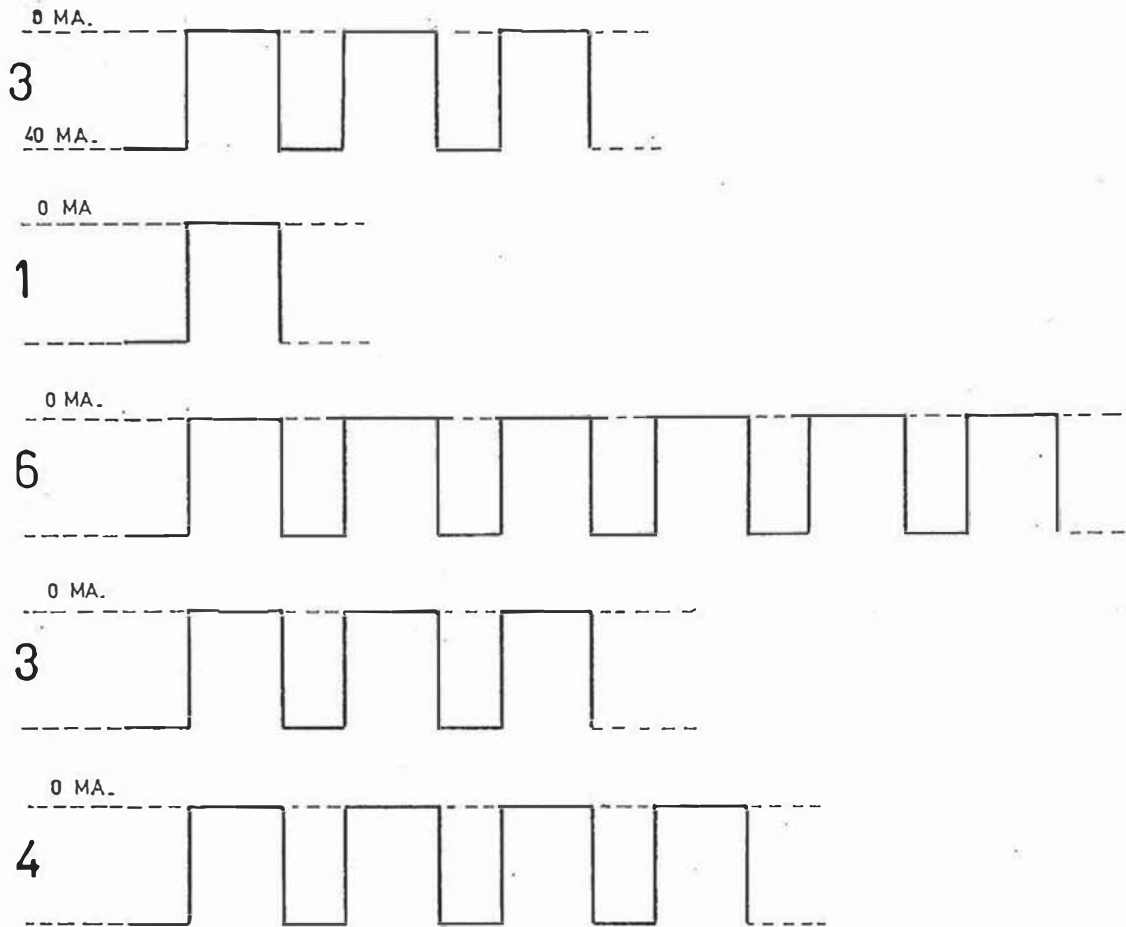


PROGRAMA QUE REALIZA LLAMADA AUTOMATICA

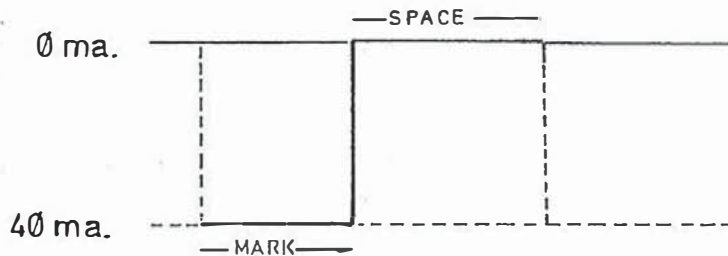
0760	START	:	NOP	00 4
0761			LD A, 00H	3E 00
0763			DUT (TELEX), A	D3 08
0765			CALL 1 SEG	CD 30 07
0768	CONECTADO	:	LDA, 02H	3E 02
076A	(5 MA)		OUT (TELEX), A	D3 08
076C			CALL 2 SEG	CD 40 07
076F	40mA	:	LD A, 03H	3E 03
0771			DUT (TELEX), A	D3 08
0773			LDA, 01H	3E 01
0775			OUT (TELEX), A	D3 08
0777			CALL 3 SEG	CD 50 07
077A	LLAMADA	:	LD A, (200H)	3A 00 20
077D			NOP	00
077e			CALL PULSOS	CD 08 07
0781			LD A, 01H	3E 01
0783			DUT (PULSO), A	D3 08
0785			CALL 1 SEG	CD 30 07
0788			LD HL, 077BH	21 7B 07
078B			INC (HL)	34
078C			LDA, 05H	3E 05
078E			CP (HL)	BE
078F			JR NZ, LLAMADA	20 E9
0791			LD (HL), 00H	36 00
0793			RET	C9

Este último programa (START) es una subrutina del programa principal llamado telex, que también llama a otra subrutina llamada "PULSOS". Los cinco dígitos del número de abono de telex se almacenan a partir de la dirección 2000H de la memoria, esto por acción del programa principal. La subrutina START tomara los dígitos uno por uno y para cada cuál llamará a la subrutina PULSOS para que éste envíe los pulsos que indican ese dígito.

Supongamos que tenemos el número de abonado 31634, entonces tendremos los pulsos siguientes:



NUMERO DE ABONADO 31634



MARK ES EL 39% DEL PERIODO DE UN CICLO
SPACE ES EL 61% DEL PERIODO DEL MISMO CICLO

LA FRECUENCIA DE LA ONDA ES 10HZ.

Por lo tanto mediante el programa "PULSOS" se enviaran tantos pulsos de este tipo como lo indique el digito tratado. Si el digito tratado fuese cero entonces se en viaran 10 pulsos de estos.

Ahora calculemos los tiempos de MARK y SPACE

$$\text{PERIODO} : \frac{1}{10\text{Hz}} = 0.1 \text{ segundo} = 100 \text{ milisegundos}$$

$$\text{MARK} = \frac{39}{100} \times 100 = 39 \text{ milisegundos}$$

$$\text{SPACE} = \frac{61}{100} \times 100 = 61 \text{ milisegundos}$$

El programa "PULSOS" genera los pulsos indicados por el valor del acumulador A. También la subrutina "PULSOS" utiliza otra subrutina llamada "PULSO" que se utiliza co mo base de tiempo.

Primero mostraré la subrutina "PULSO" y luego la sub rutina "PULSOS".

SUBROUTINA PULSO

0700	PULSO	PUSH BC	C5
0701		LD B, C3H	0603
0703	TEMPO	NOP	00
0704		DJNZ, TEMPO	10 FD
0706		POP BC	C1
0707		RET.	C9

Calculemos la base de tiempo que se consigue llaman

do a esta subrutina.

PUSH BC	11T	C3H = 195
LDB, C3H	7T	
NOP	4T	
DJ.NZ	13T, 8T	
POP BC	10T	
RET	10T	

$$\begin{aligned} \text{tiempo} &= 11T + 7T + (C3H-1) (4T + 13T) + (4T + 8T) \\ &\quad + 10T + 10T \\ &= 18T + 194 (17T) + 12T + 20 T \\ &= 3348T \\ &= 3348 \times \frac{1}{3.3325 \text{ MHz}} \end{aligned}$$

tiempo = 1.004651 milisegundo

SUBROUTINA "PULSOS"

0708	PULSOS	:	AND OFH	E6 OF
070A			LD D,A	57
070B			XOR A	AF
070C			OR D	B2
070D			JR NZ, AGAIN	20 02
070F			LDD, OAH	16 0A
0711	AGAIN	:	LDE, 27H (MARK 39%)	1E 27
0713			LDA, 01H	3E 01
0715			OUT (TELEX), A	D3 08
0717	MARK	:	CALL PULSO	CD 00 07
071A			DEC E	1D
071B			JR NZ, MARK	20 FA
071D			LDE, 3DH (SPACE 61%)	1E 3D
071F			LDA, 00H	3E 00
0721			OUT (TELEX), A	D3 08
0723	SPACE		CALL PULSO	CD 00 07
0726			DEC E	1D
0727			JRNZ, SPACE	20FA

0729	DEC D	15
072A	JRNZ, AGAIN	20 E5
072C	RET	C9

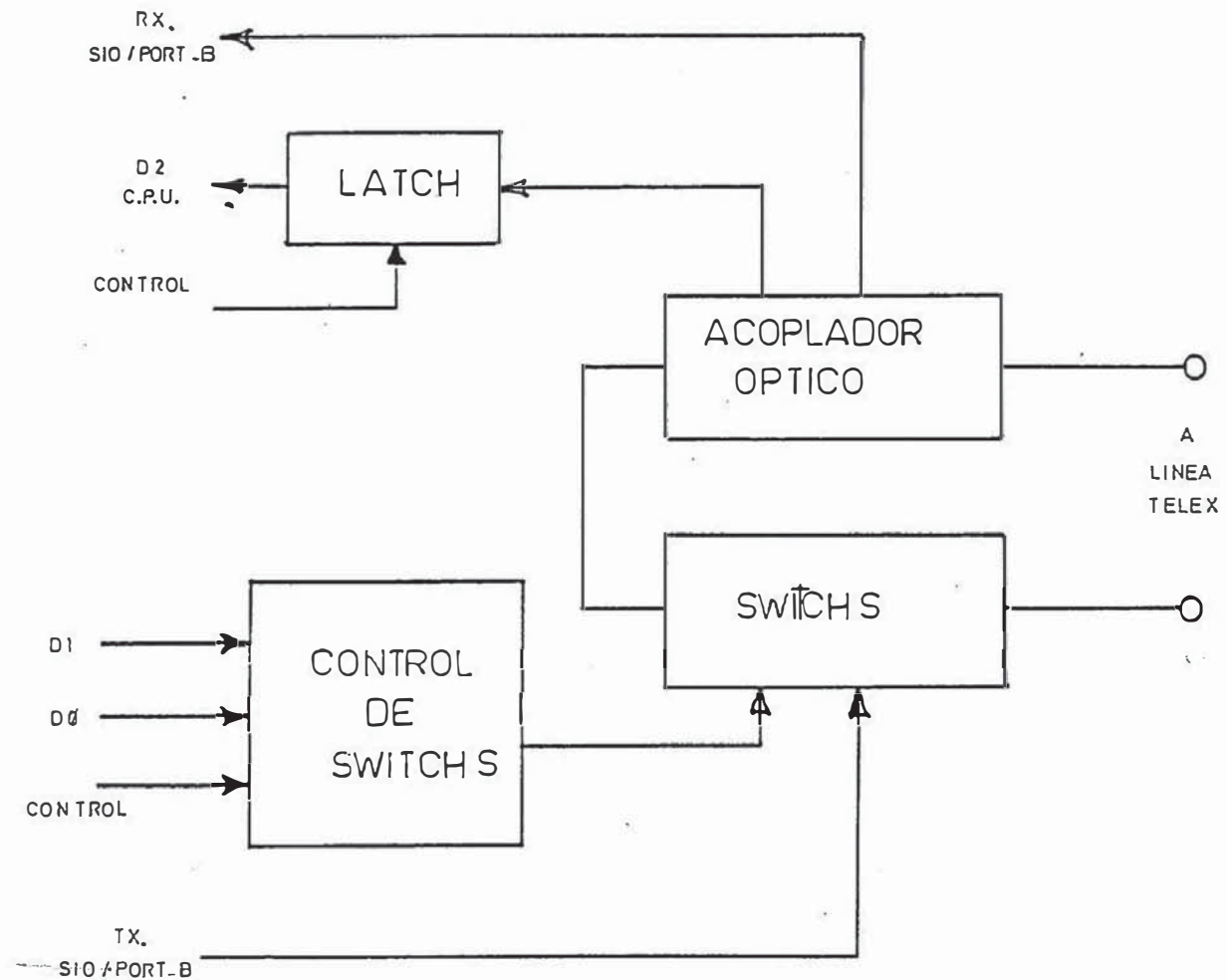
Estos programas se llegaron a terminar después de muchas pruebas de enlace. Notar que el byte selector 08H corresponde al switch A.

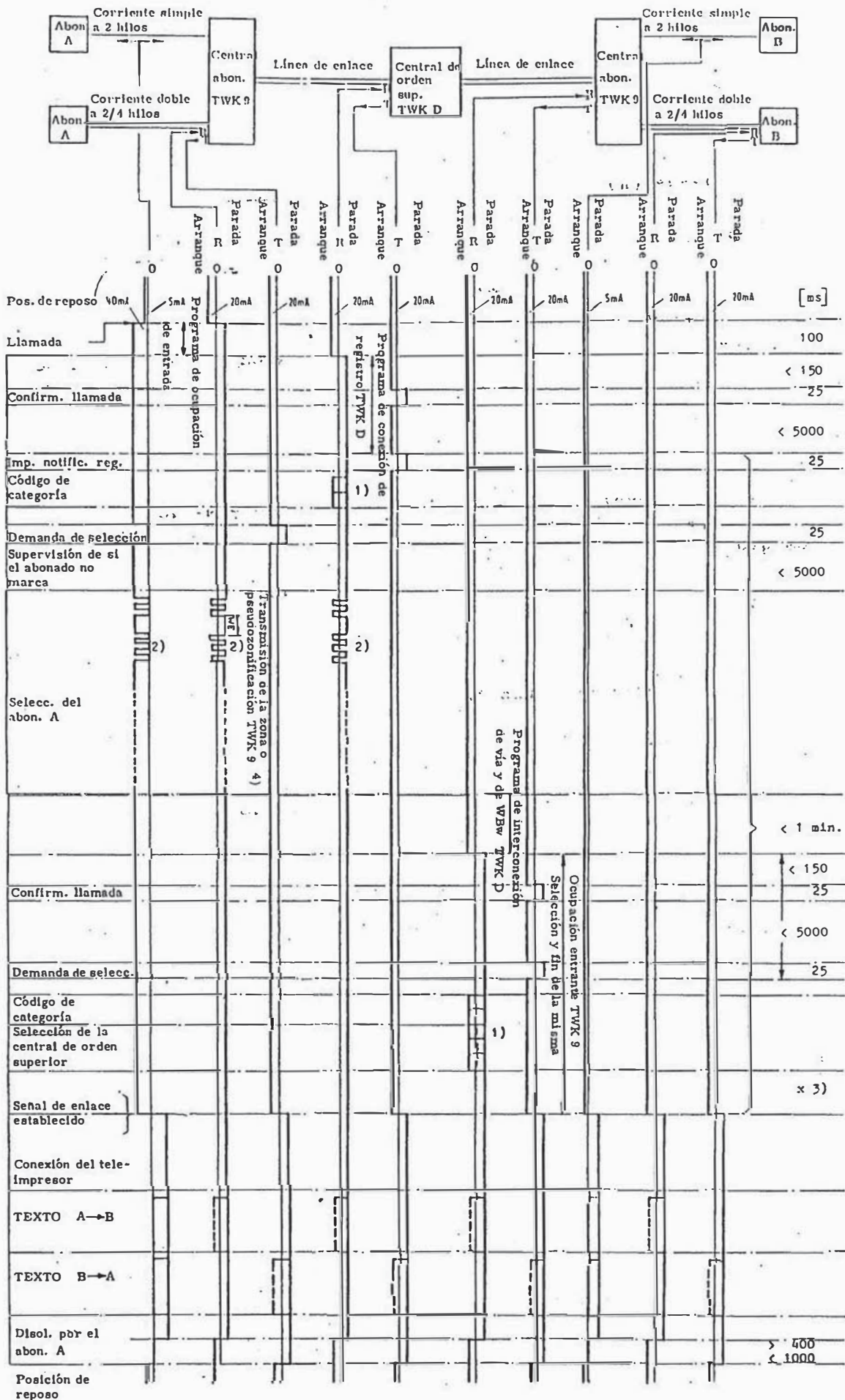
TELEX = 08H (byte selector)

CODIGO BAUDOT

<u>HEXADECIMAL</u>	<u>LETRAS</u>	<u>SIMBOLOS</u>
03	A	-
19	B	?
0E	C	:
09	D	IDENTIFICACION
01	E	3
0D	F	\$
1A	G	;
14	H	N
06	I	8
0B	J	BELL
0F	K	(
12	L)
1C	M	.
0C	N	,
18	O	9
16	P	0
17	Q	1
0A	R	4
05	S	'
10	T	5
07	U	7
1E	V	=
13	W	2
1D	X	/
15	Y	6
11	Z	+
08	RETORNO DE CARRO	
02	NUEVA LINEA	
04	ESPACIO	
1B	SIMBOLOS	
1F	LETRAS	
00	VACIO	

DIAGRAMA DE BLOQUES - INTERFAZ A TELEX





1) en código "2 de 5"

2) Selección por disco marcador

3) x = Tiempo máximo de interconexión en la central de abonados

4) El programa "Fin de selección - Transmisión de la zona" o "Fin de selección - Pseudozonificación" puede solicitarse después de la 1ª, la 2ª, la 3ª... o la 8ª cifra.

CAPITULO VI

PROGRAMACION DEL SISTEMA

6.1 Sistema Operativo que permite interactuar al interfaz inteligente con un terminal inteligente.

Este sistema operativo permite poder acceder al procesador por medio del teclado del terminal (teclado ASCII) por medio de unos comandos fáciles de usar.

Este sistema operativo se inicia en la dirección 0800H de la memoria. Entonces inicialmente cuando se enciende este equipo de prueba al presionar la tecla RESET, el procesador está bajo el control del programa monitor que tiene como dirección inicial 0000H, luego de presionar la tecla RESET se digita la dirección 0800H en los display de direcciones por medio del teclado hexadecimal para luego presionar la tecla RUN que hará correr el programa grabado en la memoria a partir de la dirección 0800H que en este caso es el sistema operativo. Una vez que el control del procesador lo tiene este programa, en la pantalla de el terminal aparecerá el siguiente mensaje.

DIGICOMP S.A.

LIMA PERU

* MICROCOMPUTADOR DE DESARROLLO *

XODIAC, 64K

*)

Los caracteres de invitación a teclear algún comando comúnmente llamado prompt aparece en la pantalla, estos son

*)

Los comandos son los siguientes:

HELP, MENU, JUEGOS, SISBOOT

Si Ud. digita HELP y luego la tecla CR (retorno de carro) o NL (nueva linea) se mostrara lo siguiente en la pantalla:

COMANDOS

- MENU
- JUEGOS
- SYSBOOT

Si Ud. digita cualquier otra palabra que no sea HELP o alguno de los comandos el sistema operativo lo detectará y enviará el siguiente mensaje que será mostrado en la pantalla:

ERROR : DIGITE "HELP"

*)

Así Ud. podrá digitar HELP y aparecerá la lista de comandos. Si Ud. ha digitado por ejemplo HELM lo cual es un comando equivocado, puede usar la tecla DEL que enviará un caracter hacia el procesador, el que será decodificado por el sistema operativo e interpretado como: borrar el último caracter y así se tendrá HEL.

Si Ud. presiona nuevamente DEL, el sistema operativo borrará de su memoria y de la pantalla el caracter L y quedará:

*) HE

Si continuamos presionando DEL borraremos los caracteres E y H, si sigue presionando la tecla DEL el cursor no se moverá de su lugar límite y sonará el timbre cada vez que se presione tal tecla, todo esto controlado por el sistema operativo. Además por programa se puede controlar el máximo número de caracteres que pueden ser

almacenados en memoria que a la vez seran mostrados. Así, en este diseño se escogio 30 caracteres por línea, los de mas caracteres que se presionan no serán ni mostradas ni almacenados esta opción es también controlada por el sistema operativo.

Supongamos que digitamos SYSBOOT

*) SYSBOOT

si es así después de presionar la tecla CR ó NL el control del programa lo tendrá el programa monitor por que este comando carga el contador de programa del microprocesador con la dirección 0000H.

Supongamos ahora que digitamos JUEGOS y luego presionamos la tecla CR ó NL, entonces se presentara en la pantalla lo siguiente:

JUEGOS:

- (1) GUERRA DE LAS GALAXIAS
- (2) MISION IMPOSIBLE
- (3) EL ELEFANTE Y LA PULGA
- (4) EXIT

Seleccione un número:

Entonces dependiendo del número que Ud. presione el contador de programa del microcomputador se cargara con la respectiva dirección de inicio del programa creado para cada juego. Como no se ha realizado ningún desarrollo de programas de juego, en las cuatro opciones hemos colocado la dirección de inicio del sistema operativo, por lo que se reinicializara y aparecera el prompt.

*)

Si ahora digitamos el comando MENU, tendremos en la pantalla lo siguiente:

MENU:

- (1) CARACTERES ASCII
- (2) TELEX
- (3) INSTALACION CP/M
- (4) EXIT

Seleccione un número:

El seleccionar un número, hara que el programa prin cipal tome una dirección correspondiente al número y sea ese el programa de control actual. Así si Ud. digita 1 el procesador iniciara a ejecutar el programa que empieza en la dirección 1CA0H, que es un programa que pre senta todos los caracteres ASCII de la siguiente forma: 10 líneas de caracteres de un solo tipo, cada linea consta de 80 caracteres, luego se presentan 3 espacios y aparecen 10 lineas mas del siguiente caracter y así sucesivamente. Este proceso de lazo se puede interrumpir presionando cualquier tecla y se presentara el prompt

*)

Si nuevamente digitamos MENU y luego seleccionamos el número 2 el contador de programa se cargara con la dirección 0650H el cuál es la dirección de inicio del programa de comunicación que sirve para enlazar el interfaz inteligente a un abonado telex remoto.

Este programa TELEX automáticamente mostrara en la pantalla el siguiente mensaje:

" TELEX-2000 "

" Esta opción le permite llamar a un abonado TELEX "

" DIGITE EL NUMERO: "

luego Ud. podrá digitar el número de abonado que Ud. desee, el sistema operativo también controla la presentación y el almacenaje, pudiendo borrar algún digito equi-

vocado. Cabe recordar que el sistema 0. Almacena estos 5 digitos en la memoria del interfaz a partir de la dirección 2000H.

Una vez digitado los 5 digitos si Ud. esta seguro que ese es el número de abonado con el cual desea enlazarse, no tiene mas que presionar la tecla CR y NL y el interfaz comenzará a realizar el proceso de llamado automático. (visto en el capítulo anterior). Mientras este proceso se ejecuta se presenta el siguiente mensaje en la pantalla:

" SE ESTA REALIZANDO LA LLAMADA"

una vez que se consigue el enlace, el interfaz detectará el sentido de la corriente y si se mantiene o no enlazado por medio segundo (500ms.) mientras tanto aparecerá en la pantalla el siguiente mensaje:

"CENTRAL TELEX LLAMANDO AL ABONADO"

a) Si el enlace se completa entonces aparecerá el siguiente mensaje:

ENVIE SU MENSAJE

>

El prompt en este caso es el simbolo ">". Ud. podrá digitar cualquier mensaje, ya sean letras o números todo digitado en el teclado ASCII del terminal inteligente.

Si Ud. desea esperar la respuesta a su mensaje, entonces Ud. deberá presionar las teclas "CONTROL" y "A" - (se denota ↑A), así el interfaz estará en el modo repción y para indicar este, en la pantalla aparecerá el prompt ")". Por lo tanto se podrán repcionar mensajes y estos aparecerán en la pantalla del terminal. Además estando en este modo de repción Ud. puede pedir el indicativo de su colateral. Una vez que Ud. se ha enlazado con el abonado deseado (colateral) y estando en el mo

do recepción, si presiona las teclas "control" y "Z" - (↑ Z) el interfaz inteligente enviará los caracteres - en código BAUDOT: SIMBOLO Y PETICION DE INDICATIVO (1BH, 09H) hacia el teleimpresor remoto (colateral) enviando este su indicativo, automáticamente.

Por ejemplo: El indicativo del Banco Continental es CONTIBANK. Estando en el modo recepción, tecleamos dos - veces Z. Entonces en la pantalla se mostrará lo siguiente.

```
)  
21281 CONTIBK  
21281 CONTIBK
```

Luego si Ud. desea enviar algún mensaje, deberá pasar al modo transmisión, presionando las teclas A y así aparecerá el prompt " > " que le indica a Ud. que ya puede enviar su mensaje. Si Ud. desea cortar el enlace entonces deberá presionar la tecla ESCAPE (denotada por "ESC") y el sistema operativo al detectar ese caracter, dará - por finalizada la telecomunicación.

b) Ahora, supongamos que el colateral esta ocupado y por lo tanto el enlace no se llega a realizar, entonces en la pantalla aparecerá el mensaje:

```
"LINEA OCUPADA"
```

y luego de 3 segundos el control vuelve al Sistema Operativo, desde un inicio.

Aquí se presenta la distribución de memoria para el sistema operativo (primera parte)

```
1E15  DATOS A SER DISPLAYADOS PORT SIO POLLING  
1E34  (PARA PROGRAMA DE PRUEBA)  
1E37  PROGRAMA: DISPLAY  
1E4C  D/200 (POLLING)  
1E50  INICIALIZACION DEL SIO  
1E68  D/200 (POLLING) PORT A  
1E7A  INICIALIZACION DEL SIO  
1E80  INTERRUPCION DE Rx  
1E81  INICIALIZACION DEL SIO
```

1E87 INTERRUPCION DE Tx.
1E88 INICIALIZACION DEL SIO
1E8E INTERRUPCION DE Tx y Rx
1E90 SUBROUTINA OE
1EAC INICIALIZACION
1EBO DATOS DE SUBROUTINA
1EBD DE INICIALIZACION
1ECO RETARDO HL FFFFH
1EC8 (SUBROUTINA DE ESTABILIZACION)
1EC9 SUBROUTINA DISPLAY
1ED4 (RESET/INT PENDING)
1ED7 SUBROUTINA BUFFER PORT A
1EDD
1DOO PROGRAMA DE PRUEBA
1D72 INT DE Rx
1D75 SUBROUTINA
1D87 HELP
1D90 SUBROUTINA
1DBD MENU
1DCO SUBROUTINA
1DD8 JUEGOS
1F03 VECTORES
1FOD
1F10 SUBROUTINA DE INTERRUPCION
1F2E DE TRANSMISION
1F30 SUBROUTINA DE INTERRUPCION
1F97 DE RECEPCION
1FAO PROGRAMA DE PRUEBA
1FBS INT DE Tx
1CAO CARACTERES
1CDF ASCII
0000 PARA
 . USO
 . EXCLUSIVO
 . DEL
05FF PROGRAMA MONITOR.

1800 DATOS A DISPLAYAR
1866 POR LA SUBROUTINA DE TRANSMISION
1867 USO ESPECIAL, CUENTA DE COMANDOS COMPARADOS
1868 DIRECCIONES DONDE
: ALMACENAN
: LOS DATOS TECLEADOS
1899
189A DATOS A DISPLAYAR
18B6 ERROR: DIGITE HELP *)
1879 USO ESPECIAL, DATO QUE INDICA SALIR DE SUBROUTINA
18BA DATOS DE
18FO COMANDO
18F2 DATOS DE
1975 JUEGO
1977 DATOS DE
19E5 MENU
1A00 COMANDOS GRABADOS
1A1F QUE SON COMPARADOS

VECTORES INTERRUPCION

1F03 00 DIRECCION A PARTIR DEL CUAL
1FO4 18 SE DISPLAYARA
1FO5 68 DIRECCION A PARTIR DEL CUAL
1FO6 18 SE ALMACENAN LOS DATOS TECLEADOS
1F08 11 DIRECCION DE SUBROUTINA
1FO9 1F DE INTERRUPCION DE Tx
1FOC 30 DIRECCION DE SUBROUTINA
1FOD 1F DE INTERRUPCION DE Rx

VER SISTEMA OPERATIVO AL FINAL DE CAPITULO

Aquí se presenta la distribución de memoria para el sistema operativo (Segunda parte).

0600 PROGRAMACION SIO
0616 POLLING PORT B
061A PROGRAMA DE PRUEBA
062F SIOPOLLING

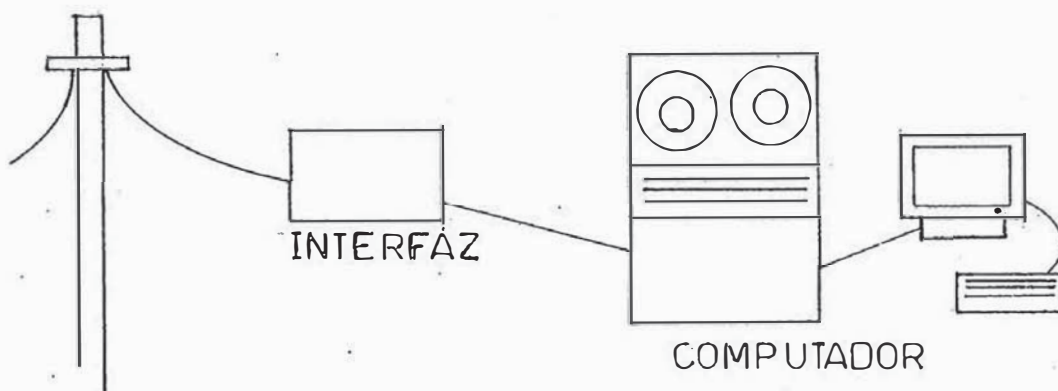
0630 DATOS DE PRUEBA
064D SIO POLLING PORT B
0650 PROGRAMA TELEX
06CE DIALOGO
06D5 OCUPADO
06EA
0700 SUBROUTINA
0707 PULSO = 1.0046511 segundos
0708 SUBROUTINA REGISTRO A = # de pulsos
072C PULSOS
0730 1 SEGUNDOS
073D
0740 2 SEGUNDOS
074D
0750 3 SEGUNDOS
075D
0760 SUBROUTINA START 2000 = 5 digitos
0793 LLAMADA AUTOMATICA
0794 SUBROUTINA RECEPCION TELEX
07BF CHEQUEA CAMBIO DE SENTIDO DE CORRIENTE
07CO PROGRAMA
07FB TRANSFERENCIA (MODO TRANSMISION)
0800 SIMULA TELEIMPRESOR DESCONECTADO
0806 Y SE ENTRA AL PROGRAMA PRINCIPAL
080A 00 SIMBOLOS
01 LETRAS
00 SIMBOLOS
080B 01 LETRAS
080C PROGRAMA
082B LETRA
0830 PROGRAMA
0857 SIMBOLO
085A PROGRAMA RESPONDA
086B ENVIA AL COLATERAL CR, NL, NL
0870 SUBROUTINA BUFFER PORT B
0876
0880 PROGRAMA RESPONDA 1
0890

089A PROGRAMA RECEPCION (MODO RECEPCION)
090C
0910 PROGRAMA
0921 LETRA 1
092A PROGRAMA
0935 SIMBOLO 1

VER SISTEMA OPERATIVO AL FINAL DE CAPITULO

6.2 SISTEMA OPERATIVO QUE PERMITE INTERATUAR AL INTERFAZ INTELIGENTE CON UN COMPUTADOR.

Otra manera de usar es la de conectar un port del computador a el interfaz inteligente y este ultimo a la línea telex.



Esta es una forma de uso muy útil ya que toda la información recibida puede ser almacenada en la memoria del computador. O la información grabada en la memoria puede ser enviada hacia un teleimpresor remoto por medio de la interfaz.

El programa que controla este modo de operación empieza en la dirección 0A00H.

Una vez corrido este programa el computador también puede correr su programa de comunicación para con el in-

terfaz.

Su operación es la siguiente:

El computador este encendido y realizado algún proceso, entonces Ud. puede ingresar a controlar un proceso por cualquiera de los terminales inteligentes conectados a el COMPUTADOR.

Recordar que el interfaz inteligente esta conectado a uno de los ports del computador, estos ports son canales de comunicación asincrona o sincrona serial, a estos port, normalmente se les conecta terminales inteligentes.

Bueno, una vez que usted tiene el control del terminal, puede invocar al programa de comunicaciones grabado en el computador para poder entablar comunicación con el interfaz.

Entonces, supongamos que el programa se llama TELEXØ, Ud. digitará este nombre y luego de presionar CR o NL, se ejecutará.

)TELEX Ø

entonces este programa inicializará (programará) la interfaz y mostrará en el terminal del computador lo siguiente:

- 1.- MODO TRANSMISION/DIALOGO
- 2.- MODO RECEPCION AUTOMATICA
- 3.- EXIT

Luego el computador esta a la espera de uno de los 3 números (1, 2 ó 3) para que el programa de comunicaciones, programe la interfaz en el modo seleccionado y el mismo programa del computador pase el control del proceso a la subrutina indicada por el número seleccionado.

Si Ud. escosio "1":

- a) MODO TRANSMISION/DIALOGO: En este modo Ud. desde su terminal de computador (terminal inteligente conectado a un port del computador) podrá digitar el número de abonado con el cual desea comunicarse, luego del enlace conseguido podrá intercambiar mensajes con el abonado telex seleccionado; Ud. desde su terminal de computador y su colateral desde su teleimpresor. Ahora si Ud. desea transmitir un documento, no tiene mas que presionar las teclas "CONTROL" y "B" (denotado - por B) y en la pantalla aparecerá:

"NOMBRE DEL DOCUMENTO"

Entonces Ud. dará el nombre del documento almacenado en algún medio magnético (unidad de disco o unidad de cinta), luego, si Ud. esta seguro de haber digitado el nombre de documento correcto presione la tecla CR ó NL y en la pantalla se mostrará lo siguiente:

"TRANSMITIENDO ARCHIVO HACIA CONVERTOR"

La transmisión de el documento desde el computador hacia el conversor es rápida, ya que tal transmisión se realiza a 4,800 Baudios (se puede escoger otra velocidad). Al finalizar la transmisión del documento hacia el conversor en la pantalla se mostrará:

DESEA LIBERAR EL COMPUTADOR MIENTRAS EL INTERFAZ
TRANSMITE (SI/NO)?

- Si la respuesta es SI (DIGITADO), el terminal de computador quedará liberado del proceso TELEX Ø y Ud. podrá utilizar tal terminal para realizar cualquier otra tarea con el computador, mientras el interfaz inteligente estará transmitiendo todo el documento almacenado en su memoria de 56KBytes, esta transmisión tomará un tiempo mayor, ya que se transmitirá a una velocidad de 50Baudios.

- Si la respuesta hubiese sido NO, el proceso TELEX Ø hará que se muestre en la pantalla el mensaje siguien

te:

"TELEX-2000, ENVIANDO ARCHIVO AL ABONADO"
y una vez que el interfaz inteligente ha terminado de enviar todo el documento, este avisará al computador (por medio del envío de un carácter) que ya terminó. Y el modo de TRANSMISION/DIALOGO seguirá activa, pudiendo así continuar con el dialogo.

Para escapar del control de este proceso Ud. solo - debe presionar la tecla ESCAPE (ESC).

Si Ud. escogio "2":

- b) MODO RECEPCION AUTOMATICA: En este modo al programa TELEX 0 que se esta corriendo en el COMPUTADOR, programa el interfaz en este modo por medio del port usado por el interfaz. Además el terminal de computador quedará liberado. Entonces todo mensaje o documento - que envíen al interfaz, este lo almacenará en su memoria de 56Kbytes; además este avisará al computador envia un caracter, para que el operador del computado se entere de que el interfaz tiene información almacenada. Luego el operador puede invocar al programa TELEX 0, apareciendo en la pantalla del terminal lo siguiente:

"DAR NOMBRE AL DOCUMENTO"

Con este nombre Ud. podrá identificar el documento que será almacenado en la memoria del COMPUTADOR que es proveniente de la memoria del interfaz. Y así Ud. tendrá ya almacenada tal información, además estará identificada.

Si Ud. escogio "3":

- c) EXIT: Esta selección, le permite a Ud. poder hacer terminar el proceso TELEX 0.

SISTEMA OPERATIVO - PARTE I

SUBROUTINA DE INTERRUPCION (TX)

1F10		DEFS	00
1F11	TX EMPTY:	PUSH AF	F5
1F12		PUSH HL	E5
1F13		LD HL, (XBUFFPTR)	2A 03 1F
1F16		LD A, (HL)	7E
1F17		CP Carriage return	FE 0D
1F19		JR NZ, SIGUE	20 09
1F1b		LD A, RESET INT/PENDING	3E 28
1F1D		OUT(SIOCTRL),A	D3 02
1F1F		LD(1F10H), A	32 10 1F
1F22		JR, EXIT	18 06
1F24		OUT(SIODATA), A	D3 00
1F26		INC HL	23
1F27		LD(XBUFFPTR), HL	22 03 1F
1F2A	EXIT:	POPHL	E1
1F2B		POP AF	F1
1F2C		EI	FB
1F2D		RETI	ED 4D

SUBROUTINA DE INTERRUPCION (RX)

1F30	EMPTY:	PUSH AF	F5
1F31		PUSH BC	C5
1F32		PUSH HL	E5
1F33		IN A,(SIODATAA)	DB 00
1F35		CP Carriage return	FE 8D
1F37		JRZ, SCAPE	28 04
1F39		CP New Line	FE 8A
1F3B		JRNZ, SALTO	20 07
1F3D	SCAPE:	LD A, 8DH	3E 8D
1F3F		LD(2011H), A	32 11 20
1F42		JR, SALIDA	18 19
1F44	SALTO:	CP DELETE	FE FF
1F46		JRZ, BORRAR	28 1B
1F48		XOR B	A8
1F49		LD HL, (RBUFFPTR)	2A 05 1F
1F4C		LD BC, 2030H	01 30 20
1F4F		SUBC BC	ED 42
1F51		JRZ, SALIDA	28 0A
1F53		OUT(SIODATA), A	D3 00
1F55		LD HL, (RBUFFPTR)	2A 05 1F
1F58		LD (HL), A	77
1F59		INC HL	23
1F5A		LD (RBUFFPTR), HL	22 05 1F
1F5D	SALIDA:	POP HL	E1
1F5E		POP BC	C1
1F5F		POP AF	F1
1F60		E1	FB
1F61		RETI	ED 4D

(este programa continua en la siguiente página)

1F63	BORRAR:	LD HL, (RBUFFPTR)	2A 05 1F
1F66		LD BC, 2000H	01 00 20
1F69		XOR A	AF
1F6A		SUBC BC	ED 42
1F6A		JRNZ, SIGUA	20 06
1F6E		LD A, 87H	3E 87
1F70		OUT(SIODATA A),A	D3 00
1F72		JR, REGRESO	18 1E
1F74	SIGUA:	LD HL, (RBUFFPTR)	2A 05 1F
1F77		DEC HL	2B
1F78		LD (HL), 00H	36 00
1F7A		LD (RBUFFPTR), HL	22 05 1F
1F7D		LD A,--	3E 99
1F7F		OUT(SIODATA A),A	D3 00
1F81		CALL, BUFPTR	CD D7 1E
1F84		LD A, SPACE	3E A0
1F86		OUT(SIODATA A),A	D3 00
1F88		CALL, BUFFER de TX	CD D7 1E
1F8B		LD A,--	3E 99
1F8d		OUT(SIODATA A),A	D3 00
1F8F		CALL, BUFFER de TX	CD D7 1E
1F92	REGRESO:	POP HL	E1
1F93		POP BC	C1
1F94		POP AF	F1
1F95		EI	FB
1F96		RETI	ED 4D

PROGRAMA DE PRUEBA (usa interrupcion de TX)

TX : Transmisión

RX : Recepción

1FA0	*** :	LD HL, 1800H	21 00 18
1FA3		LD (if03H), HL	22 03 1F
1FA6		LD SP, FFF0H	31 F0 FF
1FA9		CALL INT de TX	CD 81 1E
1FaC		LD A, ERASE PAG.	3E 0C
1FaE		OUT(SIODATA A),A	D3 00
1FB0		CALL DISPLAY	CD c9 1E
1FB3		JP P.P.I. RX	C3 00 1D

PROGRAMA DE PRUEBA (usa interrupción de RX)

1D00	P.P.I. RX :	LD HL, COUNTER	21 28 1D
1D03		LD (HL), 00H	36 00
1D05		LD HL, DIR ALMACEN	21 00 20
1D08		LD(RBUFFPTR), HL	22 05 1F
1D0B		LD SP, FFF0H	31 F0 ff
1D0E		CALL INT de RX	CD 7A 1E
1D11		XOR A	AF
1D12		LD DE, 1FFEh	11 FE 1F
1D15		LD B, 13H	06 13
1D17	CEROS :	INC DE	13
1D18		LD (DE), A	12
1D19		DJNZ, CEROS	10 FC
1D1B	COM :	LD A, (2011H)	3A 11 20
1D1E		CP Carriage return	FE 8D
1D20		JRNZ, COM	20 F9
1D22		NOP	00
1D23		NOP	00
1D24	INICIO :	LD DE, 2000H	11 00 20
1D27		LD HL, 1A00H	21 00 1A
1D2A		LD BC, 0008H	01 08 00
1D2D	ADE :	LD A, (DE)	1A
1D2E		CPI	ED A1
1D30		INC DE	13
1D31		JRNZ, CONTINUE	20 06
1D33		LD A, B	78
1D34		OR C	B1
1D35		JPZ, PROG SEL	28 26
1D37		JR, ADE	18 F4

(continua en la siguiente página)

1D39	CONTINUE :	LD HL, COUNTER	21 28 1D
1D3C		LD A, 08H	3E 08
1D3E		ADD (HL)	86
1D3F		LD (HL), A	77
1D40		LD HL, 1FFFH	21 FF 1F
1D43		INC (HL)	34
1D44		LD A, (HL)	7E
1D45		CP 04H	FE 04
1D47		JRNZ, INICIO	20 DB
1D49	ERROR :	LD HL, 189AH	21 9A 18
1D4C		LD B, # de Caracteres	06 1D
1D4E		LD C, 00H	0E 00
1D50	⌘ :	IN A, (02H)	DB 02
1D52		BIT 2, A	CB 57
1D54		JRZ, ⌘	28 FA
1D56		OUTI	ED A3
1D58		JRNZ, ⌘	20 F6
1D5A		JP P.P.I. RX	c3 00 1D
1D5D	PROG SEL :	LD HL, 1FFFH	21 FF 1F
1D60		LD A, (HL)	7E
1D61		CP 00H	FE 00
1D63		JPZ, HELP	CA 75 1D
1D66		CP 01H	FE 01
1D68		JPZ, MENU	CA 90 1D
1D6B		CP 02H	FE 02
1D6D		JPZ, SYSBOOT	CA 00 00
1D70		JP , JUEGOS	c3 C0 1D

MENSAJE ALMACENADO EN MEMORIA ,
LO DA EL SISTEMA OPERATIVO EN UN INICIO

1800	ERASE PAG	0C
1801	WRITE CURSOR ADDRESS	10
1802	POSITION	03
1803		00
1804	D	44
1805	I	49
1806	G	47
1807	I	49
1808	C	43
1809	O	4F
180A	M	4D
180B	P	50
180C	SPACE	20
180D	S	53
180E	.	2E
180F	A	41
1810	.	2E
1811	HOME (Cursor 0,0)	08
1812	WRITE CURSOR ADDRESS	10
1813	POSITION	40
1814		00
1815	L	4C
1816	I	49
1817	M	4D
1818	A	41
1819	-	2D
181A	P	50
181B	E	45
181C	R	52
181D	U	55
181E	HOME	08
181F	WRITE CURSOR ADDRESS	10
1820	POSITION	17
1821		03

1822	*	2A
1823	SPACE	20
1824	START UNDERSCORE	14
1825	M	4D
1826	I	49
1827	C	43
1828	R	52
1829	o	4F
182A	C	43
182B	O	4F
182C	M	4D
182D	P	50
182E	U	55
182F	T	54
1830	A	41
1831	D	44
1832	O	4F
1833	R	52
1834	END UNDERSCORE	15
1835	SPACE	20
1836	START UNDERSCORE	14
1837	D	44
1838	E	45
1839	END UNDERSCORE	15
183A	SPACE	20
183B	START UNDERSCORE	14
183C	D	44
183D	E	45
183E	S	53
183F	A	41
1840	R	52
1841	R	52
1842	O	4F
1843	L	4C
1844	L	4C
1845	O	4F

1846	END UNDERSCORE	15
1847	SPACE	20
1848	*	2A
1849	HOME	08
184A	WRITE CURSOR ADDRESS	10
184B	POSITION	00
184c		06
184D	START REVERSE	1E
184E		44
184F	X	58
1850	O	4F
1851	D	44
1852	I	49
1853	A	41
1854	C	43
1855	-	2D
1856	∅	30
1857	1	31
1858	END REVERSE	1E
1859		45
185A	SPACE	20
185B	,	2C
185C	6	36
185D	4	34
185E	K	4B
185F	.	2E
1860	New Line	0A
1861	New Line	0A
1862	Start Blink	0E
1863	*	2A
1864	End Blink	0F
1865)	29
1866	Carriage Return	0D

MENSAJE ALMACENADO EN MEMORIA ,
LO ENVIA EL SISTEMA OPERATIVO CUANDO DETECTA ERROR
AL DIGITAR ALGUN COMANDO

189A	New Line	0A
189B	E	45
189C	R	52
189D	R	52
189E	o	4F
189F	R	52
18A0	:	3A
18A1	SPACE	20
18A2	D	44
18A3	I	49
18A4	G	47
18A5	I	49
18A6	T	54
18A7	E	45
18A8	SPACE	20
18A9	SPACE	20
18AA	"	22
18AB	H	48
18AC	E	45
18AD	L	4C
18AE	P	50
18AF	"	22
18B0	SPACE	20
18B1	New Line	0A
18B2	New Line	0A
18B3	*	2A
18B4)	29
18B5	BELL	07
18B6	bell	07

PROGRAMA HELP

1D75	HELP :	LD HL, 18BAH	21 BA 18
1D78		LD(1F03H), HL	22 03 1F
1D7B		CALL INT de TX	CD 81 1E
1D7E		LD A, BELL	3E 07
1D80		OUT(SIODATA A),A	D3 00
1D82		CALL DISPLAY	CD c9 19
1D85		JP p.P.I. RX	C3 00 1D

MENSAJE ALMACENADO EN MEMORIA

18BA	ERASE PAG.	0C
18BB	WRITE CURSOR ADDRESS	10
18BC	POSITION	12
18BD	"	00
18BE	START UNDERSCORE	14
18bF	C	43
18C0	o	4F
18C1	M	4D
18C2	A	41
18C3	N	4E
18C4	D	44
18C5	Q	4F
18C6	S	53
18c7	END UNDERSCORE	15
18C8	SPACE	20
18C9	:	3A
18CA	CURSOR DOWN	1A
18CB	CURSOR DOWN	1A
18CC	-	1D
18CD	SPACE	20
18CE	M	4D
18CF	E	45

18D0	N	4E
18D1	U	55
18D2	WRITE CURSOR ADDRESS	10
18D3	POSITION	1C
18D4		03
18D5	-	2D
18D6	ESPACIO	20
18D7	J	4A
18D8	U	55
18D9	E	45
18DA	G	47
18DB	O	4F
18DC	S	53
18DD	WRITE CURSOR ADDRESS	10
18DE	POSITION	1C
18DF	!!	04
18E0	-	2D
18E1	SPACE	20
18E2	S	53
18E3	Y	59
18E4	S	53
18E5	B	42
18E6	O	4F
18E7	O	4F
18E8	T	54
18E9	WRITE CURSOR ADDRESS	10
18EA	POSITION	00
18EB		06
18EC	START BLINK	0E
18ED	*	2A
18EE	END BLINK	0F
18EF)	29
18F0	CR	0D
18F1	00H	00

PROGRAMA MENU

1D90	MENU :	LD HL, 1977H	21 77 19
1D93		LD (1F03H), HL	22 03 1F
1D96		CALL INT de TX	CD 81 1E
1D99		LD A, BELL	3E 07
1D9B		OUT (SIODATA A),A	D3 00
1D9D		CALL, DISPLAY	CD c9 1E
1DA0	PAS :	IN A, (SIOCTRLA)	DB 02
1DA2		BIT 0,A	CB 47
1DA4		JRZ, PAS	28 FA
1DA6		IN A, (SIODATA A)	DB 00
1DA8		NOP	00
1DA9		NOP	00
1Daa		AND OFH	E6 0F
1DAC		CP 01H	FE 01
1DAE		JPZ, CARACTERES ASCII	CA A0 1C
1DB1		CP 02H	FE 02
1DB3		JPZ, TELEX	CA 50 06
1DB6		CP 03H	FE 03
1DB8		JPZ, INSTALACION CP/M	CA A0 1F
1DBB		JP , PROGRAMA PRINCIPAL	C3 A0 1F

MENSAJE ALMACENADO EN MEMORIA (MENU)

1977	ERASE PAG	0C
1978	WRITE CURSOR ADDRESS	10
1979	POSITION	1D
197A		02
197B	START REVERSE	1E
197C		44
197D	M	4D
197E	E	45
197F	N	4E
1980	U	55
1981	END REVERSE	1E
1982		45
1983	WRITE CURSOR ADDRESS	10
1984	POSITION	14
1985		05
1986	(28
1987	1	31
1088)	29
1989	SPACE	20
198A	c	43
198B	A	41
198C	R	52
198D	A	41
198E	C	43
198F	T	54
1990	E	45
1991	R	52
1992	E	45
1993	S	53
1994	SPACE	20
1995	A	41
1996	S	53
1997	C	43
1998	I	49
1999	I	49
199A		

199A	WRITE CURSOR ADDRESS	10
199B	POSITION	14
199C		06
199D	(28
199E	2	32
199F)	29
19A0	SPACE	20
19A1	T	d4
19A2	E	C5
19A3	L	CC
19A4	E	C5
19A5	X	D8
19A6	New Line	A0
19A7	WRITE CURSOR ADDRESS	10
19A8	POSITION	14
19A9		07
19AA	(28
19AB	3	33
19AC)	29
19AD	SPACE	20
19AE	I	49
19AF	N	4E
19B0	S	53
19B1	T	54
19B2	A	41
19B3	L	4C
19B4	A	41
19B5	C	43
19B6	I	49
19B7	o	4F
19B8	N	4E
19B9	SPACE	20
19BA	(28
19BB	C	43
19BC	P	50
	/	2F

19BD	/	2F
19BE	M	4D
19BF)	29
19C0	WRITE CURSOR ADDRESS	10
19C1	POSITION	14
19C2		08
19C3	(28
19C4	4	34
19C5)	29
19C6	SPACE	20
19C7	E	45
19C8	X	58
19C9	I	49
19CA	T	54
19CB	WRITE CURSOR ADDRESS	10
19CC	POSITION	0F
19CD		0B
19CE	S	53
19CF	E	45
19D0	L	4C
19D1	E	45
19D2	C	43
19D3	C	43
19D4	I	49
19D5	O	4F
19D6	N	4E
19D7	E	45
19D8	SPACE	20
19D9	U	55
19DA	N	4E
19DB	SPACE	20
19DC	N	4E
19DD	U	55
19DE	M	4D
19DF	E	45
19E0	R	52

19E1	0	4F
19E2	SPACE	20
19E3	:	3A
19E4	SPACE	20
19E5	CR (Carriage rreturn)	0D

PROGRAMA JUÉGOS

1DC0	JUEGOS :	LD HL, 18F2H	21 F2 18
1DC3		LD (1F03H), HL	22 03 1F
1DC6		CALL INT de TX	CD 81 1E
1DC9		LD A, BELL	3E 07
1DCB		OUT (SIODATA A),A	D3 00
1DCD		CALL DISPLAY	CD C9 1E
1DD0	% :	IN A,(SIOCTRL A)	DB 02
1DD2		BIT 0,A	CB 47
1DD4		JRZ, %	28 FA
1DD6		JP PROGRAMA PRINCIPAL	C3 A0 1F

MENSAJE ALMACENADO EN MEMORIA (JUEGOS)

18F2	ERASE PAG	0C
18F3	WRITE CURSOR ADDRESS	10
18F4	POSITION	12
18F5		00
18F6	START UNDERSCORE	14
18F7	J	4A
18F8	U	55
18F9	E	45
18FA	G	47
18FB	O	4F
18FC	S	53
18FD	END UNDERSCORE	15
18FE	SPACE	20
18FF	:	3A
1900	CURSOR DOWN	1A
1901	CURSOR DOWN	1A
1902	(28
1903	1	31

1904)	29
1905	SPACE	20
1906	G	47
1907	U	55
1908	E	45
1909	R	52
190A	R	52
190B	A	41
190C	SPACE	20
190D	D	44
190E	E	45
190F	SPACE	20
1910	L	4C
1911	A	41
1912	S	53
1913	SPACE	20
1914	G	47
1915	A	41
1916	L	4C
1917	A	41
1918	X	58
1919	I	49
191A	A	41
191b	S	53
191C	WRITE CURSOR ADDRESS	10
191D	POSITION	1A
191E		03
191F	(28
1920	2	32
1921)	29
1922	SPACE	20
1923	M	4D
1924	I	4E
1925	S	53
1926	I	49
1927	o	4F

1928	N	4E
1929	SPACE	20
192A	I	49
192B	M	4D
192C	P	50
192D	O	4F
192E	S	53
192F	I	49
1930	B	42
1931	L	4C
1932	E	45
1933	WRITE CURSOR ADDRESS	10
1934	POSITION	1A
1935		04
1936	(28
1937	3	33
1938)	29
1939	SPACE	20
193A	E	45
193B	L	4C
193C	SPACE	20
193D	E	45
193E	L	4C
193F	E	45
1940	F	46
1941	A	41
1942	N	4E
1943	T	54
1944	E	45
1945	SPACE	20
1946	Y	59
1947	SPACE	20
1948	L	4C
1949	A	41
194A	SPACE	20
194B	P	50

194C	U	55
194D	L	4C
194E	G	47
194F	A	41
1950	WRITE CURSOR ADDRESS	10
1951	POSITION	1A
1952		05
1953	(28
1954	4	34
1955)	29
1956	SPACE	20
1957	E	45
1958	X	58
1959	I	49
195A	T	54
195B	WRITE CURSOR ADDRESS	10
195C	POSITION	12
195D		07
195E	S	53
195F	E	45
1960	L	4C
1961	E	45
1962	C	43
1963	C	43
1964	I	49
1965	o	4F
1966	N	4E
1967	E	45
1968	SPACE	20
1969	U	55
196A	N	4E
196B	SPACE	20
196C	N	4E
196D	U	55
196e	M	4D
196F	E	45
1970	R	52

1971	0	4F
1972	SPACE	20
1973	:	3A
1974	SPACE	20
1975	CR (Carriage Return)	0D

PROGRAMA
QUE PRESENTA LOS CARACTERES ASCII

1CA0	ASCII :	CALL SIOPOLLING	CD 50 1E
1CA3		LD HL, DIRECCION DE TRABA.	21 78 18
1CA6	START :	LD (HL), SPACE	36 20
1CA8		LD E, 5FH	1E 5F
1CAA	BETA :	LD D, 0BH	16 0B
1CAC		LD C, 00H	0E 00
1CAE	ALFA :	LD B, 50H	06 50
1CB0		LD A, New Line	3E 0A
1CB2		OUT(SIODATA A),A	D3 00
1CB4	GAMA :	IN A, (SIOCTRL A)	DB 02
1CB6		BIT 2,A	CB 57
1CB8		JRZ, GAMA	28 FA
1CBA		BIT 0,A	cB 47
1CBC		JRZ, #	28 05
1CBE		IN A, (SIODATA A)	DB 00
1CC0		JP , PROGRAMA PRINCIPAL	c3 A0 1F
1CC3	# :	OUTI	ED A3
1CC5		DEC HL	2B
1CC6		JRNZ, GAMA	20 EC
1CC8		DEC D	15
1CC9		JRNZ, ALFA	20 E3
1CCB		INC (HL)	34
1CCC		LD C, 03H	0E 03
1CCE	* :	LD A, New Line	3E 0A
1CD0		OUT(SIODATA A),A	D3 00
1CD2	SUMATORIA :	IN A,(SIOCTRL A)	DB 02
1CD4		BIT 2,A	CB 57
1CD6		JRZ, SUMATORIA	28 FA
1CD8		DEC C	0D
1CD9		JRNZ , *	20 F3
1CDB		DEC E	1D
1CDC		JRNZ, BETA	20 CC
1CDE		JR , START	18 C6

SISTEMA OPERATIVO - PARTE II

PROGRAMACION DEL SIO(POLLING PORT B)

0600	SIOPOLLING B :	LD C, SIOCTRL B	0E 03
0602		LD B, PROGRAM	0609
0604		LD HL, TABLA PRG	21 0E 06
0607		OUTIR	ED B3
0609		LD A, 50 Baudios	3E 00
060B		OUT (BAUDOT B),A	D3 18
060D		RET	C9
060E		DEFB	18
060F		DEFB	01
0610		DEFB	00
0611		DEFB	04
0612		DEFB	48
0613		DEFB	03
0614		DEFB	01
0615		DEFB	05
0616		DEFB	88

0650	TELEX :	LD HL, DATOS a DISPLAYAR 1;21 60 1A	
0653		LD (1F03H),HL	22 03 1F
0656		CALL INT de TX	CD 81 1E
0659		LD a, BELL	3E 07
065B		OUT (SIODATA A),A	D3 00
065D		CALL display	CD c9 1E
0660		LD HL, DIRECC. #TELEX	21 00 20
0663		LD (RBUFPTR),HL	22 05 1F
0666		LD SP, FFF0H	31 F0 FF
0669		CALL INT de RX	CD 7A 1E
066C		XOR A	AF
066D		LD (2011H),A	32 11 20
0670		LD DE, 1FFEH	11 FE 1F
0673		LD B, 13H	06 13
0675	CEROS :	INC DE	13
0676		LD (DE), A	12
0677		DJNZ CEROS	10 FC
0679	COM 1 :	LD A, (2011H)	3A 11 20
067C		CP CR	FE 8D
067E		JRNZ, COM 1	20 F9
0680		LD HL, DATOS a DISPLAYAR 2;21 C0 1A	
0683		LD (1F03H), HL	22 03 1F
0686		CALL INT de TX	CD 81 1E
0689		LD A, BELL	3E 07
068B		OUT (SI)DATA A),A	D3 00
068D		CALL DISPLAY	CD C9 1E
0690		CALL SIOPOLLONG B	CD 00 06
0693		CALL START	CD 60 07
0696		CALL RX TEL	CD A0 07
0699		CALL INT de TX	CD 81 1E
069C		LD HL, DATOS A DISPLAYAR 3;21 F0 1A	
069F		LD (1F03H),HL	22 03 1F
06A2		LD A, BELL	3E 07
06A4		OUT (SIODATA A),A	D3 00
06A6		CALL DISPLAY	CD C9 1E
06A9		CALL 2 SEC	CD 40 07
06AC		CALL 3 SEC	CD 50 07
06AF		IN A, (MENSAJE ?)	DBOC

06B1	BIT 2,A	CB 57
06B3	JPZ, OCUPADO	CA D5 06
06B6	LD HL, DATOS a DISPLAYAR 4;21 20 1B	
06B9	LD (1F03H), HL	22 03 1F
06B6	CALL INT de TX	CD 81 1E
06BF	LD a, BELL	3E 07
06C1	OUT (SIODATA A), A	D3 00
06C3	CALL DISPLAY	CD c9 1E
06c6	CALL 1 SEC	CD 30 07
06C9	CALL SIOPOLLING A	CD 50 1E
06CC	JP TRANSFERENCIA	C3 C0 07

06D5	OCUPADO :	LD HL, DATOS a DISPLAYAR 5;21 3D 1B	
06D8		LD (1F03H), HL	22 03 1F
06DB		CALL INT de TX	CD 81 1E
06DE		LD A, BELL	3E 07
06E0		OUT (SIODATA A), A	D3 00
06E2		CALL DISPLAY	CD C9 1E
06E5		CALL 3 SEC	CD 50 07
06E8		JP PROGRAMA PRINCIPAL	c3 00 08

0730	1 SEC :	PUSH HL	E5
0731		LD HL, 0400H	21 00 04
0734	BETA :	CALL PULSO	CD 00 07
0737		DEC HL	2B
0738		LD A, H	7C
0739		OR L	B5
073A		JRNZ, BETA	20 F8
073C		POPHL	E1
073D		RET	C9
0740	2 SEC :	PUSH HL	E5
0741		LD HL, 0800H	21 00 08
0744	ALFA :	CALL PULSO	CD 00 07
0747		DEC HL	2B
0748		LD A, H	7C
0749		OR L	B5
074A		JRNZ, ALFA	20 F8
074C		POP HL	E1
074D		RET	C9
0750	3 SEC :	PUSH HL	E5
0751		LD HL, 0B00H	21 00 0B
0754	GAMA :	CALL PULSO	CD 00 07
0757		DEC HL	2B
0758		LD A, H	7C
0759		OR L	B5
075A		JRNZ, GAMA	20 F8
075C		POP HL E1	
075D		RET	C9

0794	RX TEL :	LD HL, 4300H	21 00 43
0797	SUMATORIA :	CALL PULSO	CD 00 07
079A		in A, (MENSAJE ?)	DB 0C
079C		BIT 2,A	CB 57
079E		JRNZ, SURE	20 07
07A0		DEC HL	2B
07A1		LD A,H	7C
07A2		OR L	B5
07A3		JRNZ, SUMATORIA	20 F2
07A5		JR INICIAR	18 0E
07A7	SURE :	CALL PULSO	CD 00 07
07AA		LD b, 64H	06 64
07Ac	ENGANCHE :	IN A, (MENSAJE?)	DB 0C
07AE		BIT 2,A	CB 57
07B0		JRZ, INICIAR	28 03
07B2		DJNZ, ENGANCHE	10 F8
07B4		RET	C9
07B5	INICIAR :	CALL SIOPOLLING A	CD 50 1E
07B7		JP , CARACTER "d"	C3 F0 0B
0BF0	CARACTER "d" :	LD A, "d"	3E E4
0BF2		OUT (SIODATA A),A	D3 00
0BF4		CALL 1 SEC	CD 30 07
0BF7		JP , PROGRAMA PRINCIPAL	C3 00 08

07C0	TRANSFERENCIA :	LD A, LETRAS	3E 1F
07C2		OUT (SIODATA B),A	D3 01
07C4	ESPERO :	IN A, (SIOCTRL A)	DB 02
07C6		BIT 0,A	CB 47
07C8		JRZ, ESPERO	28 FA
07CA		IN A, (SIODATA A)	DB 00
07CC		OUT (SIODATA A),A	D3 00
07CE		CP CONTROL A	FE 81
07D0		JPZ RECEPCION	CA 9A 08
07D3		CP SCAPE	FE 9B
07D5		JPZ DESCONECTAR	CA 00 08
07D8		LD B,A	47
07D9		AND FOH	E6 F0
07DB		CP 80H	FE 80
07dd		JPZ, SIMBOLO	CA 30 08
07E0		CP 90H	FE 90
07E2		JPZ SIMBOLO	CA 30 08
07E5		CP A0H	FE A0
07E7		JPZ , SIMBOLO	CA 30 08
07EA		CP B0H	FE B0
07EC		JPZ , SIMBOLO	CA 30 08
07EF		CP C0H	FE C0
07F1		JPZ, LETRA	CA 0C 08
07F4		CP D0H	FE D0
07F6		JPZ, LETRA	CA 0C 08
07F9		JP , ESPERO	C3 C4 07
0800	DESCONECTAR :	LD A, 00H	3E 00
0802		OUT (TELEX),A	D3 08
0804		JP , PROGRAMA PRINCIPAL	C3 A0 1F

080C	LETRA :	LD HL, 080AH	21 0A 08
080F		LD A, (HL)	7E
0810		CP 01H	FE 01
0812		JRZ , LETRAS	28 09
0814		LD (HL), 01H	36 01
0816		LD A,1FH	3E 1F
0818		OUT (SIODATA B),A	D3 01
081A		CALL BUFFER B	CD 70 08
081D	LETRAS :	LD A, B	78
081E		SUB COH	D6 C0
0820		LD C,A	4F
0821		LD B, 10H	06 10
0823		LD a, (BC)	0A
0824		OUT (SIODATA B),A	D3 01
0826		CALL BUFFER B	CD 70 08
0829		JP ESPERO 03 04 07	C3 C4 07
0830	SIMBOLO :	LD HL, 080AH	21 0A 08
0833		LD a, (HL)	7E
0834		CP 00H	FE 00
0836		JRZ, SIMBOLOS	28 09
0838		LD (HL),00H	36 00
083A		LD A, 1BH	3E 1B
083C		OUT (SIODATA B),A	D3 01
083E		CALL BUFFER B	CD 70 08
0841	SIMBOLOS :	LD A,B	78
0842		AND 3FH	E6 3F
0844		LD HL, 1021H	21 21 10
0847		LD BC, 0020H	01 20 00
084A		CPIR	ED B1
084C		DEC HL	2B
084D		LD A,L	7D
084E		SUB 20H	D6 20
0850		OUT (SIODATA B),A	D3 01
0852		CALL BUFFER B	CD 70 08
0855		JP , ESPERO	C3 C4 07

085A	RESPONDA :	LD HL, ARCHIVO	21 6C 08
085D		LD B, LONGITUD BE LINEA	06 03
085F		LD C, PORT DE SALIDA B	0E 01
0861	GAMA :	IN A, (SIOCTRL B)	DB 03
0863		BIT 2,A	cb 57
0865		JRZ, GAMA	28 FA
0867		OUTI	ED A3
0869		RETZ	C8
086A		JR GAMA	18 F5
086C		CR	08
086D		NL	02
086E		NL	02
0880	RESPONDA 1 :	LD a, NL	3E 0A
0882		OUT (SIODATA A),A	D3 00
0884		CALL BUFFER A	CD D7 1E
0887		LD A, PROMT	3E 3E
0889		OUT (SIODATA A),A	D3 00
088B		CALL BUFFER A	CD D7 1E
088E		JP ESPERO	C3 C4 07
0870	BUFFER B :	IN A, (SIOCTRL B)	DB 03
0872		BIT 2,A	CB 57
0874		JRZ, BUFFER B	28 FA
0876		RET	C9

089A	RECEPCION :	LD A,New Line	3E 0A
089C		OUT (SIODATA A),A	D3 00
089E		CALL BUFFER A	CD D7 1E
08A1		LD a, PROMT	3E 29
08A3		OUT (SIODATA A),A	D3 00
08A5		CALL BUFFER A	CD D7 1E
08A8		CALL RESPONDA	CD 5A 08
08AB	VUELVE :	CALL 1 SEC	CD 30 07
08AE	CLEAR :	CALL SIOPOLLING B	CD 00 06
08B1	CAMBIO :	IN A, (SIOCTRLA)	DB 02
08B3		BIT 0,A	CB 47
08B5		JRZ CAMBIO 1	28 0C
08B7		IN A, (SIODATA A)	DB 00
08B9		CP CONTROL A	FE 81
08BB		JPZ , RESPONDA 1	CA 80 08
08BE		CP CONTROL Z	FE 9A
08C0		JPZ , PEDIR IDENTIF.	CA 7A 09
08C3	CAMBIO 1	IN A, (SIOCTRL B)	DB 03
08C5		BIT 0,A	CB 47
08c7		JRZ, CAMBIO	28 E8
08C9		IN A,(SIODATA B)	DB 01
08CB	BAUDOT :	AND 1FH	E6 1F
08CD		CP 08H (CRB)	FE 08
08CF		JRNZ CRNLSP	20 07
08D1		LD A, ODH	3E 0D
08D3		OUT (SIODATA A)	D3 00
08D5		JP CAMBIO 2	C3 29 09
08D8	CRNLSP :	CP 02H (NLB)	FE 02
08DA		JRNZ, CRNLSP 1	20 07
08DC		LD A, OAH	3E 0A
08DE		OUT (SIODATA A),A	D3 00
08E0		JP CAMBIO 2	c3 29 09
08E3	CRNLSP 1 :	CP 04H (SPB)	FE 04
08E5		JRNZ, ANALISIS	20 07
08E7		LD A, 20H (SPA)	3E 20
08E9		OUT (SIODATA A),A	D3 00
08EB		JP CAMBIO 2	C3 29 09
08EE	ANALISIS :	CP 1FH (LETRAS B)	FE 1F
08F0		JRNZ, ANALISIS 1	20 09

08f2		LD A, 01H (LETRAS A)	3E 01
08F4		LD HL, 080BH	21 0B 08
08F7		LD (HL),A	77
08F8		JP CAMBIO 2	C3 29 09
08FB	ANALISIS 1 :	CP 1BH (SIMBOLOS B)	FE 1B
08FD		JRNZ, CAMBIAR	20 09
08FF		LD A, 00H (SIMBLOS A)	3E 00
0901		LD HL, 080B	21 0B 08
0904		LD (HL), A	77
0905		JP CAMBIO 2	C3 29 09
0908	CAMBIAR :	LD B,A	47
0909		LD HL, 080BH	21 0B08
090C		LD A, (HL)	7E
090D		CP 00H	FE 00
090F		JPZ, SIMBOLO 1	CA 30 09
0912		JP LETRA 1	C3 1A 09
091A	LETRA 1 :	LD A, B	78
091B		LD HL, 1001H	21 01 10
091B		LD BC, 001BH	01 1B 00
0921		CPIR	ED B1
0923		DEC HL	2B
0924		LD A, COH	3E C0
0926		ADD L	85
0927		OUT (SIODATA A),A	D3 00
0929	CAMBIO 2 :	CALL BUFFER A	CD D7 1E
092C		JP , CAMBIO	C3 B1 08
0930	SIMBOLO 1 :	LD A,B	78
0931		ADD 20H	C6 20
0933		LD C, A	4F 00
0934		LD B,10H	06 10
0936		LD A, (BC)	0A
0937		CP 2AH ("*")	FE 2A
0939		JPZ, DAR IDENTIF.	CA 4A 09
093C		OUT (SIODATA A),A	D3 00

093E		CALL BUFFER A	CD D7 1E
0941		JP CAMBIO	C3 B1 08
094A	DAR IDENTIF.	LD HL, # de ABONADO	21 63 09
094D		LD B, LONGITUD DE LINEA	06 13
094F		LD C, PORT DE SALIDA B	0E 01
0951	GAMA :	IN A, (SIOCTRL B)	DB 03
0953		BIT 2,A	CB 57
0955		JRZ, GAMA	28 FA
0957		OUTI	ED A3
0959		JRNZ, GAMA	20 F6
095b		LD A, 01H	3E 01
095D		LD (080BH),A	32 0B08
0960		JP VUELVE	C3 AB 08
0963	# de ABONADO :	SIMBOLOS	1B
0964		2	13
0965		1	17
0966		2	13
0967		4	0A
0968		2	13
0969		ESPACIO	04
096A		LETRAS	1F
096B		D	09
096C		I	06
096D		G	1A
096E		I	06
096F		C	0E
0970		O	18
0971		M	1C
0972		P	16
0973		SPACIO	04
0974		S	05
0975		A	03

097A	PEDIR IDENTIF.	LD A, 00H	3E 00
097C		LD (080BH), A	32 0B08
097F		LD A, 1BH (simbolos)	3E 1B
0981		OUT (SIODATA B),A	D3 01
0983		CALL BUFFER B	CD 70 08
0986		LD A, 09H (identific)	3E 09
0988		OUT (SIODATA B),A	D3 01
098A		JP CLEARBUFFER	C3 Ac 09
099E	1/2 SEC :	PUSH HL	E5
099F		LD HL, 0.5 segundos	21 00 02
09A2	\$:	CALL PULSO	CD 00,07
09A5		DEC HL	2B 2
09A6		LD A,H	7C
09A7		OR L	B5
09A8		JRNZ, \$	20 F8
09AA		POP HL	E1
09AB		RET	C9
09AC	CLEARBUFFER :	CALL 1/2 SEC	CD 9E 09
09AF		JP CLEAR	C3 AE 08
09E0	NUMERO DE AB. :	LD HL, 2000H	21 00 20
09E3	WAIT 2 :	IN A, (SIOCTRL A)	DB 02
09E5		BIT 0,A	CB 47
09E7		JRZ, WAIT 2	28 FA
09E9		IN A, (SIODATA A)	DB 00
09EB		LD (HL), A	77
09EC		INC HL	23
09ED		LD A,L	7D
09EE		CP 05H	FE 05
09F0		RET Z	C8
09F1		JR WAIT 2	18 F0

TABLA USADA POR EL SISTEMA OPERATIVO
PARA CONVERTIR CODIGOS ASCII A BAUDOT

10c1	A	03
10c2	B	19
10C3	C	0E
10C4	D	09
10c5	E	01
10C6	F	0D
10C7	G	1A
10C8	H	14
10C9	I	06
10CA	J	0B
10CB	K	0F
10CC	L	12
10CD	M	1C
10CE	N	0C
10CF	O	18
10D0	P	16
10D1	Q	17
10D2	R	0A
10D3	S	05
10D4	T	10
10D5	U	07
10D6	V	1E
10D7	W	13
10D8	X	1D
10D9	Y	15
10DA	Z	11
10DB	(00

TABLA USADA POR EL SISTEMA OPERATIVO
PARA CONVERTIR CODIGOS BAUDOT A ASCII

10E1	3	33
10E2	New Line	0A
10E3	-	2D
10E4	ESPACIO	20
10E5	'	27
10E6	8	38
10E7	7	37
10E8	Carriage Return	0D
10E9	Pedir Identificación	2A
10EA	4	34
10EB	BELL	07
10EC	,	2C
10ED	\$	24
10EE	:	3A
10EF	(28
10F0	5	35
10F1	+	2B
10F2)	29
10F3	2	32
10F4	Ñ	4E
10F5	6	36
10F6	Ø	30
10F7	1	31
10F8	9	39
10F9	?	3F
10FA	;	3B
10FB	SIMBOLOS	04
10FC	.	2E
10FD	/	2F
10FE	=	3D
10FF	LETRAS	05

MENSAJE ALMACENADO EN MEMORIA
UTILIZADO POR LA SUBROUTINA DISPLAYAR 1

1A60	ERASE PAG	0C
1A61	WRITE CURSOR ADDRESS	10
1A62	POSITION	1B
1A63		03
1A64	START REVERSE	1E
1A65		44
1A66	T	54
1A67	E	45
1A68	L	4C
1A69	E	45
1A6A	X	58
1A6B	-	2D
1A6C	2	32
1A6D	0	30
1A6E	0	30
1A6F	0	30
1A70	END REVERSE	1E
1A71		45
1A72	WRITE CURSOR ADDRESS	10
1A73	POSITION	0D
1A74		07
1A75	E	45
1A76	S	53
1A77	T	54
1A78	A	41
1A79	ESPACIO	20
1A7A	0	4F
1A7B	P	50
1A7C	C	43
1A7D	I	49
1A7E	0	4F
1A7F	N	4E
1A80	ESPACIO	20

1A81	L	4C
1A82	E	45
1A83	ESPACIO	20
1A84	P	50
1A85	E	45
1A86	R	52
1A87	M	4D
1A88	I	49
1A89	T	54
1A8A	E	45
1A8B	ESPACIO	20
1A8C	L	4C
1A8D	L	4C
1A8E	A	41
1A8F	M	4D
1A90	A	41
1A91	R	52
1A92	ESPACIO	20
1A93	A	41
1A94	ESPACIO	20
1A95	U	55
1A96	N	4E
1A97	ESPACIO	20
1A98	A	41
1A99	B	42
1A9A	O	4F
1A9B	N	4E
1A9C	A	41
1A9D	D	44
1A9E	O	4F
1A9F	ESPACIO	20
1AA0	T	54
1AA1	E	45
1AA2	L	4C
1AA3	E	45
1AA4	X	58
1AA5	WRITE CURSOR ADDRESS	10

1AA6	POSITION	19
1AA7		0A
1AA8	D	44
1AA9	I	49
1AAA	G	47
1AAB	I	49
1AAC	T	54
1AAD	E	45
1AAE	ESPACIO	20
1AAF	E	45
1AB0	L	4C
1AB1	ESPACIO	20
1AB2	N	4E
1AB3	U	55
1AB4	M	4D
1AB5	E	45
1AB6	R	52
1AB7	O	4F
1AB8	:	3A
1AB9	ESPACIO	20
1ABA	Carriage Return	0D
1ABB	00H	00

MENSAJE ALMACENADO EN MEMORIA
UTILIZADO POR LA SUBROUTINA DISPLAYAR 2

1AC0	New Line	0A
1AC1	"	0A
1AC2	"	0A
1AC3	ESPACIO	20
1AC4	S	53
1AC5	E	45
1AC6	ESPACIO	20
1AC7	E	45
1AC8	S	53
1AC9	T	54
1ACA	A	41
1ACB	ESPACIO	20
1ACC	R	52
1ACD	E	45
1ACE	A	41
1ACF	L	4C
1AD0	I	49
1AD1	Z	5A
1AD2	A	41
1AD3	N	4E
1AD4	D	44
1AD5	O	4F
1AD6	ESPACIO	20
1AD7	L	4C
1AD8	A	41
1AD9	ESPACIO	20
1ADA	L	4C
1ADB	L	4C
1ADC	A	41
1ADD	M	4D
1ADE	A	41
1ADF	D	44
1AE0	A	41
1AE1	Carriage Return	0D

MENSAJE ALMACENADO EN MEMORIA
UTILIZADO POR LA SUBROUTINA DISPLAYAR 3

1AF0	ESPACIO	20
1AF1)	3E
1AF2	"	3E
1AF3	"	3E
1AF4	"	3E
1AF5	"	3E
1AF6	"	3E
1AF7	"	3E
1AF8	ESPACIO	20
1AF9	C	43
1AFA	E	45
1AFB	N	4E
1AFC	T	54
1AFD	R	52
1AFE	A	41
1AFF	L	4C
1B00	ESPACIO	20
1B01	T	54
1B02	E	45
1B03	L	4C
1B04	E	45
1B05	X	58
1B06	ESPACIO	20
1B07	L	4C
1B08	L	4C
1B09	A	41
1BOA	M	4D
1BOB	A	41
1BOC	N	4E
1BOD	D	44
1BOE	O	4F

1B0F	ESPACIO	20
1B10	A	41
1B11	L	4C
1B12	ESPACIO	20
1B13	A	41
1B14	B	42
1B15	O	4F
1B16	N	4E
1B17	A	41
1B18	D	44
1B19	O	4F
1B1A	Carriage Return	0d

MENSAJE ALMACENADO EN MEMORIA
UTILIZADO POR LA SUBROUTINA DISPLAYAR 4

1B20	New Line	0A
1B21	"	0A
1B22	"	0A
1B23	ESPACIO	20
1B24	"	20
1B25	"	20
1B26	"	20
1B27	E	45
1B28	N	4E
1B29	V	56
1B2A	I	49
1B2B	E	45
1B2C	ESPACIO	20
1B2D	E	45
1B2E	L	4C
1B2F	ESPACIO	20
1B30	M	4D
1B31	E	45
1B32	N	4E
1B33	S	53
1B34	A	41
1B35	J	4A
1B36	E	45
1B37	New Line	0A
1B38	"	0A
1B39	"	0A
1B3A)	3E
1B3B	Carriage Return	0D

MENSAJE ALMACENADO EN MEMORIA
UTILIZADO POR LA SUBROUTINA DISPLAYAR 5

1B3D	New-Line	0A
1B3E	"	0A
1B3F	"	0A
1B40	ESPACIO	20
1B41	ESPACIO	20
1B42	"	20
1B43	L	4C
1B44	I	49
1B45	N	4E
1B46	E	45
1B47	A	41
1B48	ESPACIO	20
1B49	START REVERSE	1E
1B4A		44
1B4B	O	4F
1B4C	C	43
1B4D	U	55
1B4E	P	50
1B4F	A	41
1B50	D	44
1B51	A	41
1B52	END REVERSE	1E
1B53		45
1B54	BELL	07
1B55	Carriage Return	0D

CAPITULO VII

RESULTADOS EXPERIMENTALES

Este proyecto se inicio desarrollando la unidad central de procesos, utilizando como microprocesador al Z-80A (trabaja hasta 4MHz). La memoria usada fue la 2114 (1K x 4bits) por lo que el sistema era simple ya que estas memorias son RAM estáticas y no necesitan circuiteria especial, como lo requieren las memorias dinamicas. Se tenia pensado usar memorias dinamicas para el diseño final, pero como se estaba conociendo algo nuevo, el diseño de un procesador central, se quizo empezar a trabajar con tales memorias (2114). Lo cual trajo beneficios ya que me dedique a conocer más al asembler del microprocesador Z-80.

Este circuito solo contaba con los siguientes componentes:

	CANTIDAD
Z-80A	1
MEMORIA RAM 2114	2
MEMORIA EPROM 2716	1
DECODIFICACION 74LS139	1
GENERADOR DE SEÑAL	1

esto para realizar la unidad central de procesos; un programa se tenia que grabarlo, en la memoria EPROM, para ver que tal funcionaba.

Y como todo programa se perfeccionava o se altera para superar algunos errores, se tenia que estar borrando y grabando la memoria (2716), lo cual no era nada práctico. Para salvar esa dificultad diseñe un circuito interfaz que me permitiria accesar a la memoria del procesador, pudiendo grabar en la memoria RAM por medio de un teclado en hexadecimal, bajo el control de un programa que controla tal teclado. También usa 6 displays de los cuales cual

tro me permitieron observar las direcciones y los otros 2 los datos almacenados en tales direcciones.

Este interfaz cuenta con los siguientes componentes:

Interface periferica programable 8255
74LS04
74LS06
DISPLAYS (ANODO COMUN)
TECLAS

este para realizar la interfaz a teclado y displays.

Estos dos circuitos, el procesador central y la interfaz se armaron totalmente en protoboards. El cableado lo realice usando cables de calibre 30, la fuente fue de propósitos generales (+5voltios).

Este cableado no me ocasiono ningún problema de ruido. Por lo que pude desarrollar programas y grabados fácilmente en la memoria RAM. Así cada vez que se encontraba algún error en el programa, se podría corregir rápidamente por medio del teclado hexadecimal y corroborar tal cambio por medio de los displays. Notar que los programas son escritos en lenguaje de máquina, lo cual da mucho poder al programa, ya que se podía realizar cualquier instrucción de la tabla de instrucciones y no se dependía de ningún ensamblador o compilador.

Luego comenze a investigar el uso de las memorias dinamicas y en base a las señales requeridas para su control diseñe un circuito que las controlara. Para esto tuve que modificar ciertas direcciones de control para hacer compatible el programa monitor que se uso inicialmente. Al armar el circuito que utilizaba memorias dinamicas tuve problemas con el cableado, no por el ruido, sino que la gran cantidad de cables hizo que confundiera una conexión.

Bueno, una vez superado este problema (lo cual no - rápido de detectar por el multiplexaje de señales) el circuito trabajo correctamente. El uso del osciloscopio fue fundamental para el desarrollo de este circuito.

Una vez que queda bien probado el procesador y su interfaz a teclado y displays se procedio a armar el circuito en una tarjeta de proposito general, que tomo un buen - tiempo el armarlo; una vez armado el procesador y la interfaz fueron probados, realizando allí programas de prueba, los cuales resultaron bien.

Después de esto tuve que aprender a usar y programar el SIO, este circuito integrado me permitiría comunicarme con el procesador central por medio de un terminal inteligente o un computador. El aprender a programarlo me tomo también un buen tiempo, pero aprendí lo suficiente gracias a la ayuda del procesador central que me permitia - realizar programas de prueba para así inicializar el SIO en polling, en interrupción. Luego que pude dominar la programación del SIO pase a desarrollar programas de pruebas que mostraban mensajes en el terminal, provenientes de - la memoria del computador (esto en polling) luego desarrolle programas que usaban el método de interrupción, lograndose también enviar mensajes al terminal, pero esta vez produciendo interrupciones al procesador por cada caracter enviado. También se realizaron pruebas de interrupción de recepción los que permitian grabar en la memoria del procesador todos los caracteres enviados por el teclado como consecuencia de haber presionado alguna tecla.

Este interfaz cuenta con los siguientes componentes:

	CANTIDAD
Z-80A-SIO/1	1
COM 8116	1
74LS04	1
MC 1488	1
MC 1489	1
Cristal de 5.0688MHz	1

Este circuito fue armado en otra tarjeta de propósitos generales, su implementación fue rápida, para chequear el buen armado se desarrollaron programas que probaban las etapas que lo constituyen.

Luego se procedió a diseñar la interfaz a telex, es se diseño fue hecho en base a las señales que se debían de generar para conseguir el enlace y la comunicación a la velocidad de 50 baudios. Para conseguir un enlace via telex, primero enlacé el circuito a un abonado telefónico, para lo cual use una línea telefónica de la compañía, realizando programas de llamada automática, Y así es como pude desarrollar el programa "PULSOS" que me permite llamar a un abonado cualquiera enviando los pulsos que indique el número de abonado digito a digito. Escogí desarrollar el programa "PULSOS" usando la línea telefónica ya que podía así hacer comprobaciones utilizando la otra línea telefónica de la compañía.

Una vez que el procesador central, el interfaz a terminal y el interfaz a línea telex trabajaron bien independientemente, se procedió a desarrollar el sistema operativo que me permitiría la interacción de estas tres partes.

Este sistema operativo está almacenado en la memoria EPROM, en ubicación, después del programa monitor. Inicialmente después de utilizar la tecla RESET el procesador ejecuta la primera instrucción contenida en la dirección 0000H. Corriéndose así el programa de presentación que permite mostrar el mensaje HOLA en los displays. Este programa (parte del monitor) se está ejecutando por lectura de la memoria EPROM (banco 1) y una vez que Ud. presiona alguna tecla de la última fila del teclado hexadecimal, todo el programa almacenado en la memoria EPROM pasará a grabarse en la memoria dinámica; esto se hizo porque la memoria EPROM y dinámica no podían estar trabajando a la vez ya que el procesador solo puede direccionar 64K y cuando direcciona una direc

TARJETA PROCESADORA

4164

4164

74LS244

74LS74

OSC.

74LS138

4164

4164

74LS00

74LS04

74LS93

2764

4164

4164

74LS157

74LS02

Z-80A

4164

4164

74LS157

74LS74

8255 P.P.I.

ción menor a 8K (2764) ocurría tal sobreposición de bytes en el bus de datos. La manera de solucionar ese problema fue la siguiente: Primero mediante programa se paso este programa contenido en la memoria de 8K (2764) hacia las localidades superiores de la memoria dinámica, las cuales si podían ser accedadas teniendo seleccionado el banco 1 (2764), luego estos programas almacenados en la parte superior de la memoria dinámica se pasara a sus direcciones bajas habilitando para esto el banco 2 (4164). Por lo tanto el programa monitor se correra, ahora ya se tendrá acceso a todos las localidades de memoria (4164). Todo este proceso es automático y transparente al usuario que ocurre a gran velocidad.

Este sistema operativo facilita la comunicación vía telex ya que al final de todas las explicaciones técnicas el uso de este sistema por un usuario cualesquiera, es fácil.

CONCLUSIONES

Este sistema diseñado me ha permitido dominar mucho más lo que es conocido por lógica programada. En otras palabras, he podido programar los circuitos integrados programables de tal manera que cumplieron con los requerimientos que se necesitaban para la interacción con el computador y la línea telex.

El procesador central con su interfaz a displays y teclado hexadecimal, permitirá a cualquier usuario diseñar y programar mas interfaces que sería diseñadas de acuerdo al requerimiento del dispositivo a controlar. Esta tarjeta que contiene el procesador central, tiene disponibles las señales de control del microprocesador Z-80A como también las señales de dirección y de data, - pudiendo así utilizar las señales que se necesitan, utilizando un conector y cables de extensión hacia el proto-board que contendrá el circuito que está siendo diseñado.

El desarrollo y diseño de los circuitos interfaces los he realizado en base a las señales que debían de recibir y controlar tales circuitos, pudiendo afirmar por experiencia que es la mejor manera de diseño en lo que concierne a lógica programada.

Este procesador central puede trabajar a otras velocidades, dependiendo solo de la selección del Cristal, el cual no afectará el funcionamiento del programa monitor (el microprocesador Z-80A trabaja a un máximo de frecuencia de 4MHz).

El decodificador utilizado (74LS138) tiene 8 salidas; se podría utilizar otro decodificador para así controlar 8 líneas mas, habilitando o deshabilitando uno de ellos por medio del bit A7 del bus de direcciones. Con respecto a la memoria EPROM, solo 1.5Kbytes se han utilizado quedando 6.5Kbytes para grabar programas de desarrollo.

Con respecto al SIO, existen tres modelos el 0,1 y 2; se diferencian entre sí porque alguna señal de control lo tiene uno de ellos pero el otro no lo tiene. Esto es debido a que el circuito integrado solo tiene 40 pines y un SIO con todas las señales de control sobrepasaría el número de pines y se necesitarían 41 pines. El SIO usado en este diseño es el SIO-1, al cual le falta la señal Data terminal ready (DTR) del port B.

Ahora el hecho de que el generador de baudios y el SIO sean programables, le da a esta interfaz gran capacidad de poder controlar cualquier dispositivo serial.

Supongamos que tenemos un impresor serial que trabaje a 134.5 baudios y que utiliza 6 bits por caracter, entonces para nuestro circuito no le sería ningún problema el diagnosticar tal impresor.

También este sistema me permitió conocer el uso de los caracteres de control DC1 (Xon) y DC2 (Xoff). Los cuales permiten controlar el envío de caracteres hacia o desde el computador. Si el computador recibiera el caracter DC2 este dejaría de enviar información (pero si puede recibir por su línea de recepción) hasta que recibiera el caracter DC1 y así continuaría enviando información. Recordar que uso comunicación full-duplex (significa que mientras se transmite por la línea de transmisión se puede recibir a la vez información por la línea de recepción).

La interface a telex no utiliza ningún circuito programable, pero en su totalidad está controlado por el procesador central.

Los componentes que lo conforman son los siguientes:

	CANTIDAD
Acoplador optico	1

Relés	2
74LS06	1
74LS377	1
74LS374	1
Resistencias 22K, 220 , 150K 6K, 1K, 1K,	6
Diodos 1N4001	3

El circuito 74LS377 es de gran ayuda ya que graba los comandos provenientes del procesador central; pudiendo así generar las señales necesarias para conseguir el enlace y lograr la comunicación en baudios por medio de los switches. Si se desea conectar este circuito en serie con un teleimpresor cualquiera, lo único que hay que hacer es añadir una resistencia de 60 ohmios a la salida del interfaz para que así este no altere notoriamente la impedancia del teleimpresor conectado. El uso de la interfaz en esta modalidad permite que se pueda presentar por la pantalla del terminal todos los mensajes recibidos y transmitidos hacia y donde el teleimpresor, para lo cual solo se corre el programa de recepción del interfaz.

La interfaz se conecto a un microcomputador al cual se le desarrollo un programa de comunicaciones en BASIC (a estos programas mayormente se les conoce como paquetes de comunicación), trabajó como se había previsto. Esta interfaz es gran utilidad ya que toda la información recibida puede ser almacenada en su memoria y luego usando un simple comando del microcomputador se podrá grabar esta información en algún medio magnético, en este caso, en una unidad de disco. También la información almacenada en la unidad de disco proveniente del telex, puede ser impreso en una unidad impresora y sacar las copias que se desean. También para transmitir algún documento, Ud. primero puede editar esta, usando los recursos del computador, luego almacenarlo en la unidad de disco y mediante el uso de un comando podrá enviarlo hacia el abonado telex deseado. Algo más, si el computador se malogra, el

interfaz seguirá almacenando información que le llegue. Y una vez reparado el computador, este último podrá tomar tal información y almacenarla en algún medio magnético.

Espero que el orden de presentación de este trabajo les sirva de modelos para sus diseños ya que yo lo encuentro muy apropiado.

BIBLIOGRAFIA

- The Z-80 Microcomputer Handbook
Willian Barden
- Handbook Z-80 SIO
Zilog
- Interconexion de Periféricos a Microprocesadores
Coleccion Mundo Electrónico
- Teleinformática y Redes de Computadores
Coleccion Mundo Electrónico