



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERIA

DISEÑO Y CONSTRUCCION DE UNA
TERMINAL PORTATIL

T E S I S

QUE PARA OBTENER EL TITULO DE:

INGENIERO MECANICO ELECTRICISTA

P R E S E N T A N:

GUSTAVO GERARDO ITURBIDE RUIZ
MIGUEL ANGEL OLGUIN ORTEGA

MEXICO, D. F.

1983



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

I N D I C E

INTRODUCCION.....	1
CAPITULO 1. CARACTERISTICAS GENERALES OPERATIVAS DE UNA TERMINAL	
1.1. Generalidades.....	3
CAPITULO 2. TECLADOS Y DECODIFICADORES	
2.1. Teclados.....	8
2.1.1. Estilo.....	9
2.1.2. Tipo de Contactos.....	11
2.1.3. Técnicas de Multitecleo.....	13
2.1.4. Minimización de Errores.....	14
2.1.5. Mayúsculas y/o Minúsculas	14
2.2. Decodificadores.....	14
2.2.1. Estático.....	15
2.2.2. Rastreador.....	17
2.2.3. Rastreador con Memoria.....	18
2.2.4. Rastreador con Memoria y Latch.....	20
CAPITULO 3. MECANISMOS DE IMPRESION E INTERFASE	
3.1. Mecanismos de Impresión.....	21
3.1.1. Electrosensitivo.....	21
3.1.2. Térmico.....	22
3.1.3. De Impacto.....	23
3.2. Interfase de Impresión.....	28

CAPITULO 4. CONTROLADORES	
4.1. Generalidades.....	32
4.2. Microprocesadores.....	33
4.2.1. Arquitectura.....	33
4.2.2. Tecnología.....	34
CAPITULO 5. ARQUITECTURA DE LA TERMINAL PROPUESTA	
5.1. Hardware.....	42
5.1.1. Operación.....	54
5.1.2. Diagrama.....	59
5.2. Software.....	60
5.2.1. Programa.....	64
5.2.2. Diagramas.....	76
5.2.3. Descripción.....	84
CONCLUSIONES.....	87
APENDICE A: MODEM.....	89
APENDICE B: FUENTES DE ALIMENTACION.....	98
BIBLIOGRAFIA.....	101

I N T R O D U C C I O N

Uno de los avances tecnológicos más importantes en los últimos años ha sido el desarrollo y uso de las computadoras. A la par con estas ha surgido la necesidad de poner más énfasis en el perfeccionamiento de los dispositivos de entrada y/o salida, llamados periféricos, así como de las técnicas de comunicación de datos para facilitar y optimizar su uso.

El uso de las computadoras y por tanto de los periféricos, se ha ido adelantando a todos los campos, por lo que se hace necesario crear sistemas complejos que faciliten su operación. Debido a los altos costos de un centro de cómputo personal, ha surgido la idea de crear un periférico para el manejo de datos que esté al alcance de todos, siendo este el objetivo de nuestro trabajo.

En muchas actividades la necesidad del manejo de datos en un momento dado y desde cualquier lugar, haría necesario llevar consigo un centro de cómputo, lo cual sería poco realizable.

Con la ayuda de un medio de transmisión capaz de lograr la comunicación y una terminal portátil, se puede lograr lo anterior.

Uno de los medios de transmisión altamente usados para este propósito, son las líneas telefónicas, ya que conforman una gran red de comunicación y por tanto una transmisión puede realizarse tanto a cortas distancias como a cientos de kilómetros.

En el desarrollo de este proyecto se pensaron en ciertas características con las que debe contar la terminal para tener una buena aceptación. Deberá ser portátil y por lo mismo tendrá poco peso y volumen, ahora bien, por el hecho de ser portátil deberá poder realizar funciones de teleproceso y por esto contará con un editor para poder hacer correcciones en la terminal y no en el C.P.U. Una última característica sería la de poder expandirse en cuanto a funciones, aumentando el hardware o modificando el firmware.

En el capítulo 1 se describen los principales conceptos generales para el diseño de una terminal de datos. En el capítulo 2 y 3 se describen las partes fundamentales de una terminal (teclados, mecanismos de impresión e interfases). En el capítulo 4 se hace mención de los microprocesadores más importantes para el control de un sistema y en el capítulo 5 se explica el desarrollo de la terminal propuesta.

C A P I T U L O 1

CARACTERISTICAS GENERALES OPERATIVAS DE UNA TERMINAL

1.1 GENERALIDADES

Cuando un teclado es combinado con una unidad de video o un mecanismo de impresión y ciertos dispositivos electrónicos, que pueda ser capaz de mandar información a un procesador central y recibir de éste e imprimir en el video o mecanismo de impresión, el teclado-impresora o teclado-video es llamado una terminal.

Las terminales son altamente usadas para introducir programas y datos en una computadora, habilitándola para que ésta responda al usuario.

Un diagrama a bloques general de una terminal podría ser el siguiente:

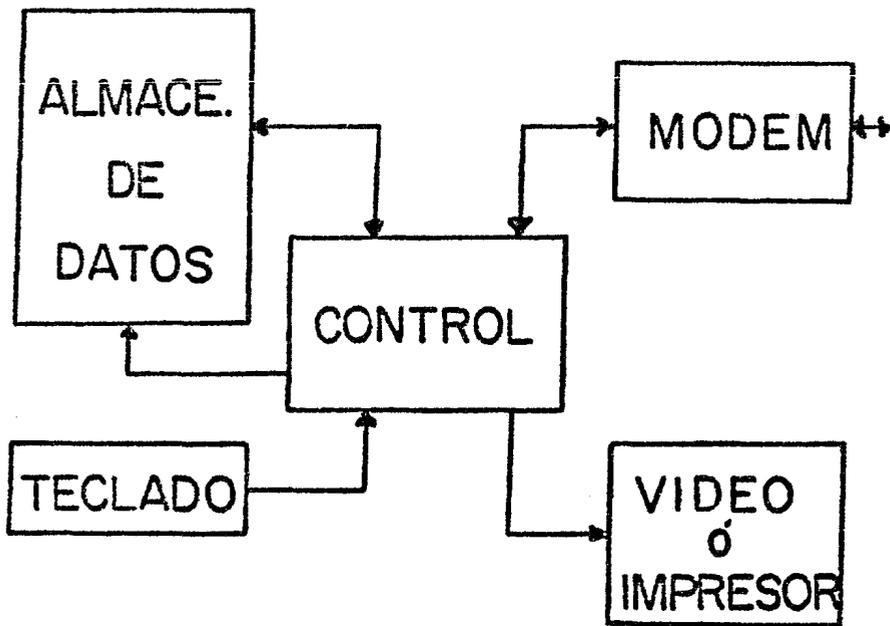


DIAGRAMA A BLOQUES DE UNA TERMINAL

El cerebro de la terminal, por así decir, es el controlador, que realiza la tarea de toma de decisiones y supervisión mediante la ayuda de un programa almacenado (monitor).

Existen varios elementos periféricos que realizan la comunicación con el exterior, así pues, el teclado es un elemento que realiza la función de traducir letras de un lenguaje cualquiera a caracteres en código entendibles por el sistema. Por otro lado el mecanismo de impresión o video, realiza la misma función pero en sentido inverso, es decir, del sistema al usuario. Existe la posibilidad de integrar otros elementos que ayuden a realizar la comunicación (MODEM, acoplador acústico, interface P/P, etc.), dependiendo de las necesidades del sistema.

El principal objetivo para el uso de una terminal, es la transmisión y recepción de datos. Existen una gran variedad de técnicas para realizar dichas funciones. El uso de una técnica determinada estará en función de cinco parámetros básicos, que son:

1. Distancia de transmisión
2. Características del medio de transmisión
3. Modo de transmisión
4. Modulación
5. Códigos de transmisión

Los tres primeros están plenamente relacionados entre sí, puesto que la distancia de transmisión determina el medio más adecuado para transmitir y el medio que se seleccione determina cuál es el modo más eficaz de transmisión. Dicho modo de transmisión estará condicionado a otras dos características, la simultaneidad y la sincronización en la transmisión.

A continuación se establece un cuadro en el cual se consideran los parámetros antes mencionados y sus características.

MEDIOS DE TRANSMISION

- Líneas Abiertas
 - Par de alambres
 - Susceptibles a inducciones electromagnéticas.
- Cables
 - Varios pares de alambres entrelazados
 - Incrementa la diafonía
 - Incremento de atenuación
- Espacio Libre
 - Capacidad de multiplexaje indefinido
 - Relativo bajo costo

MODULACION

- Por Amplitud
 - La portadora varía en amplitud
 - Recomendada para bajos niveles de ruido y cambios lentos en la moduladora.
- Por Frecuencia
 - La portadora varía en frecuencia
 - Modos síncrono o asíncrono
 - Mejora la relación señal/ruido al menos 3 db.
- Por Fase
 - La portadora varía en fase
 - Utilizada únicamente en transmisiones digitales.

MODOS DE TRANSMISION

- Simplex
 - Capacidad de recibir o transmitir
 - Utilizada en aplicaciones específicas
- Half-Duplex
 - Capacidad de recibir o transmitir no simultáneamente.
 - Siempre es a 2 alambres
- Full-Duplex
 - Capacidad de recibir o transmitir simultáneamente.
 - Generalmente a 4 alambres
- Asíncrono
 - No existe sincronización entre transmisor y receptor.
 - Señal de marca (nivel alto), señal de arranque (nivel bajo, 1 bit), señal de datos y señal de parada (nivel alto, 1, 1½ ó 2 bits).
- Síncrono
 - Existe sincronización entre transmisor y receptor.
 - La palabra SYNC es previamente definida

CODIGOS DE TRANSMISION

- ASCII (AMERICAN STANDARD CODE FOR INFORMATION INTERCHANGE):
 - Longitud: 8 bits con paridad (128 caracts.)
 - Paridad: par o impar
- EBCDIC (IBM): - Longitud: 8 bits (256 caracts.)
 - Paridad: no dispone
- HOLLERITH (12 bits, 1 a 19 letras y números)
- BAUDOT: - 5 bits (64 cts con SHIFT LETTERS Y SHIFT FIGURES)
 - Obsoleto

Como se dijo anteriormente, el elemento fundamental en la toma de decisiones es el programa monitor, estableciendo así la comunicación entre el usuario y la terminal misma. El diálogo entre estos consiste de comandos que alimenta el usuario, usando el lenguaje del monitor, y el monitor a su vez responde ya sea con un mensaje impreso o una acción.

Entre las funciones más elementales que generalmente el programa monitor le permite al usuario, se pueden mencionar las siguientes:

1. Cargar un programa codificado a partir de la localidad que le debe indicar el usuario.
2. Iniciar la ejecución de los programas del usuario.
3. Modificar los contenidos de localidades de memoria o de los registros de la CPU.
4. Imprimir en el periférico de salida los contenidos de ciertas localidades o de los registros de la CPU.
5. Otras funciones especiales, como es el caso de los editores.

El programa Monitor generalmente se encuentra cargado en una memoria PROM y que se instala a partir de la localidad 0000H.

C A P I T U L O 2

TECLADOS Y DECODIFICADORES

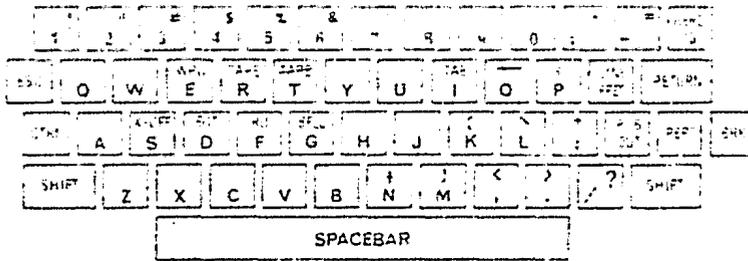
2.1 TECLADOS

Un teclado es usualmente el medio por el cual se puede introducir datos en una terminal. La mayoría de los teclados consisten de dos partes, las teclas y los decodificadores. Las teclas son simples contactos normalmente abiertos, y el decodificador es la parte que convierte un simple cierre del contacto en un código ASCII. Un diseño apropiado del decodificador elimina el ruido y anula el efecto de presionar dos teclas a la vez, también podrá dar un bit extra llamado de paridad, el cual podrá checar si hay algún error en la transmisión. Un sistema trimodo puede lograr que por medio de una llave de control, cada tecla tenga dos diferentes significados y si tiene dos llaves se podrán tener tres diferentes significados para cada tecla, estas llaves son generalmente "CONTROL" y "SHIFT". En la transmisión de datos se utiliza generalmente el código ASCII, el cual usa 7 bits para representar todos los caracteres y un octavo bit de paridad.

Hay varios factores que influyen en el diseño de un teclado entre los que podemos mencionar:

1. Estilo
2. Tipo de Tecla
3. Técnicas de Multitecleo
4. Minimización de Errores
5. Mayúsculas y/o Minúsculas

2.1.1. Estilo. Hay dos tipos básicos de arreglos que deben ser considerados, el lineal o calculadora y el común de máquina de escribir, llamado "Qwerty". Las ventajas del tipo calculador son su bajo costo y pequeño tamaño, ya que necesitan de media pulgada entre centros de las teclas, mientras que el tipo "Qwerty" necesita de tres cuartos de pulgada, además la parte numérica es mucho más rápida de operar. El color de las teclas y el agrupamiento de las mismas hacen fácil su operación. El tipo "Qwerty" es comúnmente usado en mensajes largos. Hay dos arreglos básicos de este tipo el ANSI y el Data Communication ASR-33. El ANSI (American National Standart Institute) ofrece dos tipos, uno de sólo mayúsculas y otro de combinación de mayúsculas y minúsculas y el ASR-33 ofrece sólo mayúsculas.



TECLADO ASR-33

2.1.2. Tipo de Contactos. Hay tres tipos básicos de contactos que son: los contactos mecánicos, los elastoméricos y los de tecnología sofisticada.

Los del tipo mecánico son contactos de metal sobre metal. Ejemplos típicos son alambre de oro-plata, entre dos contactos de oro, contacto oro-plata, "Ree-Bar", disco endentado de acero inoxidable y contacto.

Para un buen diseño es aconsejable usar un contacto de punto de cruce, deberá tener algún modo de acción positiva y un sensor al tacto para indicar que el contacto ha sido hecho, deberá tener protección contra corrosión y degradación y deberá minimizar el ruido.

Las técnicas elastoméricas usan una pieza de conductor flexible, plástico o espuma. De un tipo, el material elastomérico tiene que ser movido contra un contacto y tiene una constante y relativa baja impedancia. Un segundo tipo es sensitivo a la presión y su resistencia baja drásticamente bajo la misma. Los contactos elastoméricos tienen una resistencia de varios cientos y hasta miles de ohms, lo cual los hace compatibles con decodificadores ASCII particularmente MOS Y CMOS. Varios problemas limitan el uso de contactos elastoméricos, el primero es el de poder obtener una aceptable carrera y sensibilidad al tacto, un segundo problema es la degradación térmica, esto es particularmente verdadero en la espuma de alta conductividad, que cargada con partículas de plata, los efectos de tecleo disminuyen su resistividad. Esta degradación puede ser contaminación en la superficie o corrosión de las partes mecánicas con las cuales el elastómero está en contacto.

Hay muchas técnicas sofisticadas para contactos, como son los contactos inductivos y capacitivos, ambos usados mucho en el pasado y manejados por osciladores y monitoreados por amplificadores sensibles. Otra técnica es la de usar bloques conductores en tubos de mercurio, otras usan un código mecánico en forma matricial con contactos, que son ahora los más usados.

2.1.3. Técnicas de Multitecleo. Existen tres técnicas para eliminar el error de teclear más de una llave a la vez. Estas técnicas son N-Llaves "Lockout", Dos-Llaves "Rollover" y N-Llaves "Rollover".

En N-Llaves "Lockout", cualquier llave que sea tecleada genera un comando de tecleo o "STROBE", cualquier llave extra presionada no genera dicha señal y es ignorada, dicha llave debe ser restablecida antes del siguiente tecleo. Esta es la forma más simple y barata.

En Dos-Llaves "Rollover", cualquier llave que sea tecleada genera un comando de tecleo, un segundo comando de tecleo activa una lógica que envía un nuevo comando hasta que la primera llave es restablecida. Esta forma es fácil de construir y deberá ser incluida en el decodificador.

En N-Llaves "Rollover", al instante que una llave es presionada, el código es almacenado (cualquier número de llaves puede ser presionada por cualquier longitud de tiempo) cambiando los códigos de salida cada vez que una llave sea presionada. Esta técnica toma un Latch especial de salida y puede necesitar Buffers externos, así como diodos en serie con cada llave para eliminar el efecto de "Llave Fantasma". No es muy usada esta técnica porque es de mucho lujo y no tiene ventajas reales sobre las Dos-Llaves "Rollover".

2.1.4. Minimización de Errores. La mayoría de los contactos mecánicos genera ruido, para minimizarlo, todo teclado debe contar con un circuito eliminador de ruido.

Un bit adicional puede usarse para eliminar los errores sistemáticos, haciendo con este bit que los unos sean pares y en caso de que en una transmisión los unos sean nones, se puede saber que hubo un error en la transmisión. Si el teclado no genera el bit de paridad por sí mismo, existen dos integrados típicos para hacerlo, el CMOS 4531 y el TTL 74180. Existen detectores de error que rectifican los errores existentes en bloques de datos.

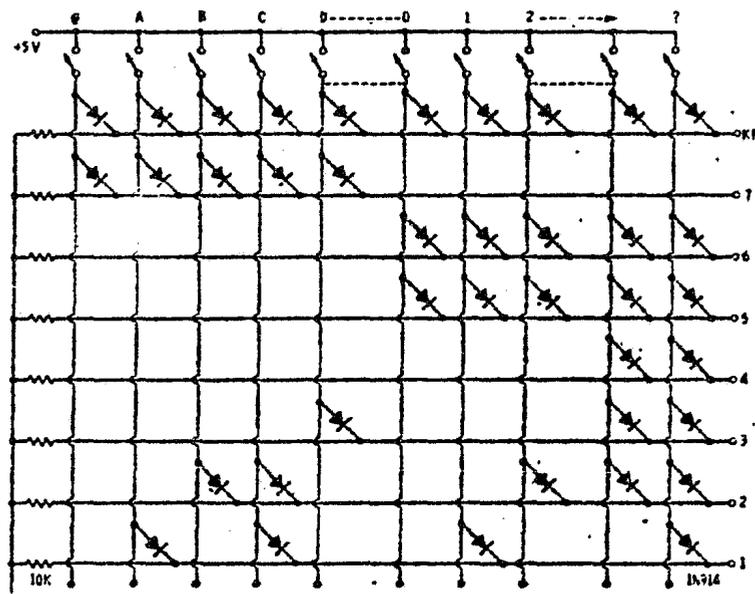
2.1.5. Minúsculas y/o Mayúsculas. Un factor importante es el de decidir si debe haber o no minúsculas. La mayoría de los Teletipos usan sólo mayúsculas, ya que resulta más económico, sobre todo cuando se usan decodificadores matriciales o generadores de caracteres en POM.

2.2. DECODIFICADORES.

La función de estos es cambiar la simple presión de un contacto a un código de 7 u 8 bits (ASCII). Deberán tener corandos de control, circuito eliminador de ruido y algunas técnicas de anti-multitecleo. Hay cuatro tipos básicos de decodificadores,

estos son en orden de costo y complejidad, el rastreador y convertidor con latch, el rastreador y convertidor, el rastreador y el estático.

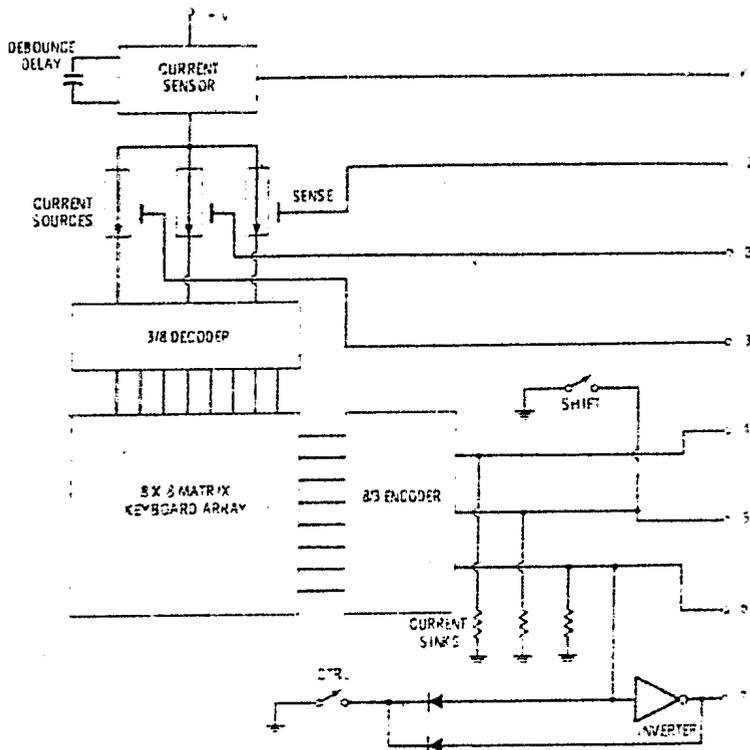
2.2.1. Decodificador Estático. En este tipo de decodificadores, es usada una lógica digital convencional, generando directamente el código. Solamente se usan diodos y resistencias como se muestra:



MATRIZ PARA CODIGO ASCII

Cada llave está conectada a una fuente de voltaje y a la vez el bus de dicha llave a determinado número de diodos, que al cerrar el contacto activa ciertos bits, siendo el código ASCII del carácter especificado. Esta forma es muy laboriosa ya que para 50 ó 60 llaves que se usan se necesitarían varios cientos de diodos.

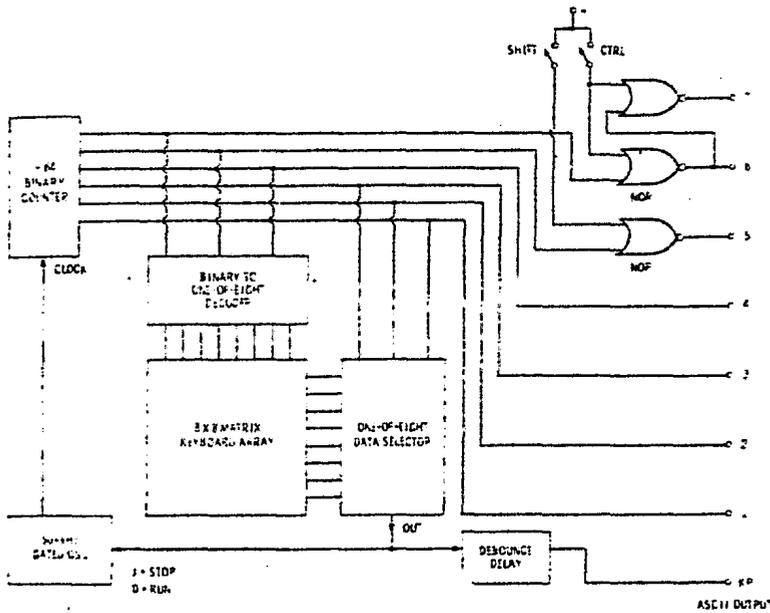
Hay otros tipos de decodificadores estáticos que usan un arreglo matricial de 8 X 8, como se muestra en la figura. El circuito tiene 3 fuentes de corriente conectadas a los bits 1, 2 y 3, limitadores de corriente en los bits 4, 5 y 6, un codificador 8 X 3 y un decodificador 3 X 8. Para sólo mayúsculas el bit 7 es el inverso del bit 6, los bits 6 y 7 pueden ser puestos a tierra por una llave de control, lo mismo que el bit 5. Este tipo de decodificador es simple y barato pero tiene algunas desventajas, como el no poder usar todos los caracteres del código ASCII. Consta además de una fuente de corriente común, que detecta la presión de una llave, un capacitor da el retraso y eliminación de ruido. El uso de estos decodificadores tiene dos inconvenientes, primero, que se dificulta al usar CMOS y TTL y segundo, que se construye en forma semidiscreta usando 40 ó 50 integrados por el hecho de usar RTL.



DECODIFICADOR ESTÁTICO

2.2.2. Decodificador Rastreador. Cada llave es continuamente interrogada en una secuencia, a través de un oscilador y de un multiplexor. El primero trabaja a 50 khz. y por tanto el contador binario lo hace a la misma velocidad. Los tres primeros bits de dicho contador entran a un decodificador (3 X 8) el cual hace que una columna y sólo una, se vaya activando, los otros tres bits del contador entran al multiplexor que preguntará por renglón. Al oprimirse una llave, ésta es sensada por el multiplexor, poniendo en su salida un "1" lógico, que detiene al oscilador y quedando así el contador con el código de la llave deseada. Se requieren de otras tres compuertas para poder tener las llaves de "Shift" y Control". Si una segunda llave es oprimida, no habrá problema, ya que el oscilador está detenido hasta que

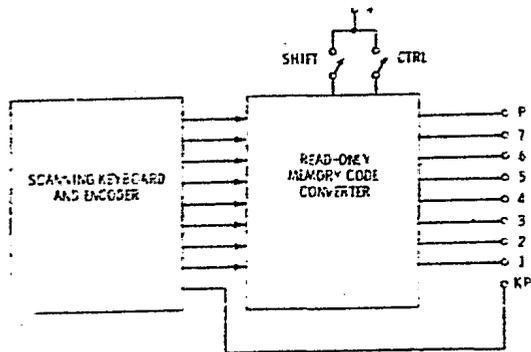
la primera tecla sea restablecida, con esto mismo se provee al circuito de la técnica de Dos-Llaves "Rollover". Este tipo de teclado es generalmente usado en operación de control remoto. El circuito integrado CMOS 4051 es particularmente atractivo para este caso.



DECODIFICADOR RASTREADOR

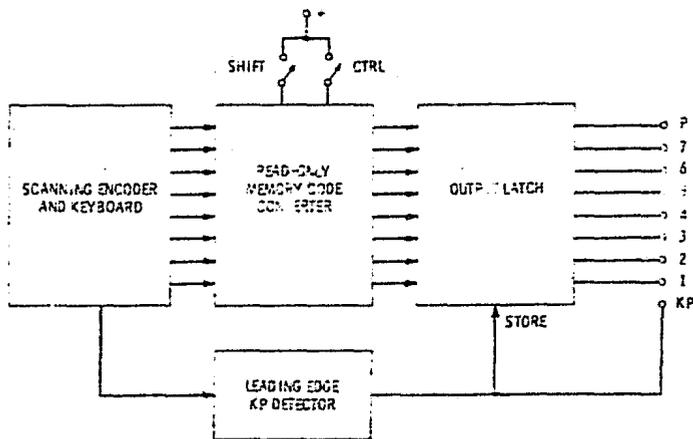
2.2.3. Decodificador Rastreador con Memoria. La forma del teclado anterior se puede reducir más aún, con el uso de una memoria ROM, ya que en lugar de que el contador de el código directamente, da la dirección de la localidad donde se encuentra dicho código.

Tiene una gran ventaja ya que el arreglo de llaves con el tipo anterior puede ser de una sola forma, en cambio con éste, se puede hacer cualquier arreglo y sólo dependería de la programación de la ROM, otra ventaja es que las llaves de control podemos localizarlas donde nosotros queramos. El circuito integrado 2376 es un típico ejemplo de decodificador rastreador con memoria.



DECODIFICADOR RASTREADOR CON MEMORIA

2.2.4. Decodificador Rastreador con Memoria y Latch. Es parecido al anterior, la diferencia estriba en que éste usa un latch a la salida de la ROM, el cual sirve para evitar errores de multitecleo y llave fantasma, ya que sólo al activarse la señal KP tenemos salida del latch.



DECODIFICADOR RASTREADOR CON MEMORIA Y LATCH

C A P I T U L O 3

MECANISMOS DE IMPRESION E INTERFASE

3.1. MECANISMOS DE IMPRESION

Por mecanismo de impresión, entendemos el conjunto de cabeza impresora y sistemas de movimiento de cabeza y papel.

En mecanismos de impresión existen tres tipos básicos que son: electrosensitivos, térmicos y de impacto.

3.1.1. Mecanismo Electrosensitivo. La cabeza impresora de este mecanismo consta de 7 ó 9 electrodos colocados verticalmente, los cuales, con el movimiento mismo de la cabeza, forman los caracteres en matriz de 5 X 7 ó 7 X 9, según sea el caso. Los electrodos tienen un diámetro aproximado de 0.010", y construídos con una resistencia de tungsteno de aproximadamente 39 ohms. El circuito consta además de una fuente de entre 40 y 60 volts, una resistencia limitadora de corriente, un switch que permita el paso de corriente y el papel mismo que hace la función de conductor. El papel electrosensitivo consiste de una película de aluminio metálico que lo hace conductor. Los puntos que forman el caracter son escritos en un tiempo aproximado de 7 mseg., con un

pico de corriente de aproximadamente 400 mA. Estas características varían de acuerdo al papel empleado ya que existen papeles que responden a 3 volts, otros a 1 mA. de corriente, etc. Los motores de movimiento de cabeza y de salto de línea, son de corriente directa, ya que dichos rotores tienen la ventaja de alcanzar rápidamente la velocidad deseada y responden eficientemente a una señal de retroalimentación para el control de velocidad, estos motores son habilitados desde la interfase con las señales de avance de carro y de papel.

3.1.2. Mecanismo Térmico. La cabeza impresora de este mecanismo consta de 7 ó 9 circuitos colocados verticalmente, o bien en una matriz de 3 X 5 ó 5 X 7, para aplicaciones del tipo "exploración por columna" o "exploración por renglón", respectivamente. Dicho circuito consta de una resistencia del tipo "wear" que se calienta al cierre del switch lógico y al paso de la corriente, necesitando aproximadamente de 7 mseg. para lograr la impresión. Tiene además una fuente de voltaje de entre 20 y 30 volts. La corriente es limitada internamente por la resistencia propia de la cabeza impresora. El papel térmico tiene una película sensible al calor, oscureciendo el punto donde fue aplicado directamente el mismo, respondiendo dicho papel a una temperatura de entre 70° y 100° C. Los motores de avance de carro y de papel son del mismo tipo que para los mecanismos electrosensitivos.

3.1.3. Mecanismo de Impacto. Existen varios tipos de mecanismo dentro de este grupo, uno de ellos puede ser el de tipo cañón, en el cual la cabeza de impresión consta de un grupo de 7 ó 9 agujas, que por medio de un solenoide propio, es impulsada sobre el papel, entre la aguja y el papel existe una cinta de carbón la cual produce la impresión. Un segundo tipo es el llamado "margarita", el cual consiste de una esfera en donde están grabados los caracteres, dicha esfera gira según el caracter a imprimir. Otro tipo es el "multicañon"; un mecanismo bastante sofisticado para impresoras de alta velocidad o trabajo pesado, consta de una serie de martillos que golpean el papel contra una banda giratoria, en la cual están grabados todos los caracteres, por tanto, en lugar de escribir los caracteres en secuencia, los escribe aleatoriamente dentro de un renglón. El papel empleado es generalmente del tipo "Bond". Los motores son tan diferentes como sofisticado sea el mecanismo.

Después de un estudio de los diferentes tipos de mecanismos de impresión, se puede observar que existe uno para cada necesidad. El tipo electrosensitivo se adapta a las necesidades requeridas para este proyecto, por lo siguiente:

1. Por su poco volumen y peso.
2. Por su velocidad de operación, que aunque no es muy rápida cumple con la velocidad requerida, que son 300 bauds.
3. Por su bajo costo.
4. Por su durabilidad.
5. Por su poca necesidad de mantenimiento, ya que tiene poco desajuste mecánico.

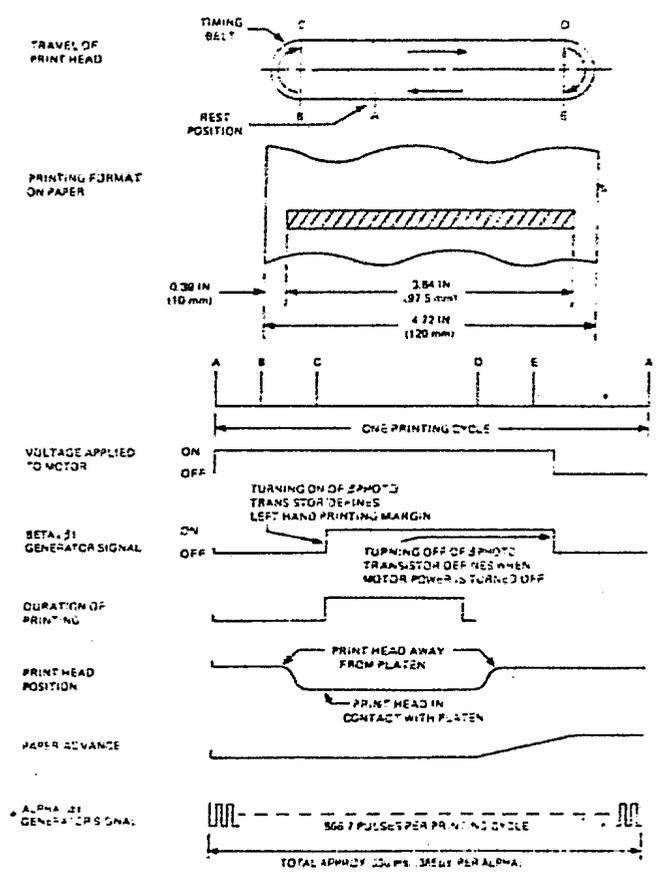
Dentro de los mecanismos electrosensitivos existen diferentes marcas y modelos, el modelo DC-4004 de Hycom es un ejemplo, en el cual nos basamos para el desarrollo de este proyecto, ya que cumple con las características requeridas.

La operación del mecanismo de impresión se puede dividir en 3 fases que son:

1. Inicio de funcionamiento del motor.
2. Impresión cuando las señales de tiempo lo indiquen.
3. Alto del motor, cuando la impresión es completada, la cabeza impresora ha regresado y el papel ha avanzado.

El ciclo de impresión comienza con el restablecimiento de la cabeza impresora a su posición original (punto A). Para imprimir, el motor es encendido y la cabeza se mueve hacia el punto B y más adelante manejada por un diente sobre una correa de transmisión. Al pasar el punto B, el diente móvil permite bajar la cabeza impresora hacia el punto C, para comenzar la fase de impresión.

Sólo después de haber llegado al punto C, un fototransistor provee la señal beta para empezar la línea de impresión. Con esta señal dada, la señal de posición de punto, puede localizar la posición de la matriz seleccionada. Así, la cabeza barre la línea de impresión, la cabeza es retractada (D), el papel avanzado (F), y la cabeza inicia el regreso al punto A. Poco después la cabeza pasa a E, la señal beta es apagada y el motor desenergizado si no hay una impresión adicional. Cuando el motor es apagado la cabeza impresora regresa por inercia al punto A.



CICLO DE IMPRESION Y DIAGRAMA DE TIEMPOS

La impresión ocurre cuando la cabeza impresora es alimentada por pulsos de voltaje. La disposición de los alambres de la cabeza impresora están en posición vertical a semejanza de un peine, coordinando los pulsos que serán usados en la matriz de puntos para formar el carácter. Como la cabeza impresora tiene 9 electrodos, pueden formar una matriz de puntos de 7 X 9, y no necesariamente usar todos. También pueden formarse las matrices de puntos de 5 X 8 y 5 X 7 en este mecanismo.

La impresión ocurre cuando la cabeza impresora toca ligeramente la superficie de aluminio del papel. Los electrodos emiten durante cerca de 170 mseg. pulsos de voltaje, los cuales provocan una fuerte corriente a través de la superficie del papel, provocando la evaporización del aluminio instantáneamente y tornando oscura la parte expuesta.

Alfa, es la señal de posición de la cabeza impresora, originada por la interrupción de un haz lumínico a través de un disco ranurado y sensada por un fototransistor. El disco ranurado es montado en la base del motor, las variaciones de velocidad del motor no afectan la posición de ningún pulso alfa. Definiendo la posición de puntos, y contando las alfas podemos definir la posición de un carácter o la generación del tiempo necesario para generar

la matriz de puntos del caracter. La salida del generador de caracteres es alimentada al circuito de potencia de la cabeza impresora, creando así la impresión.

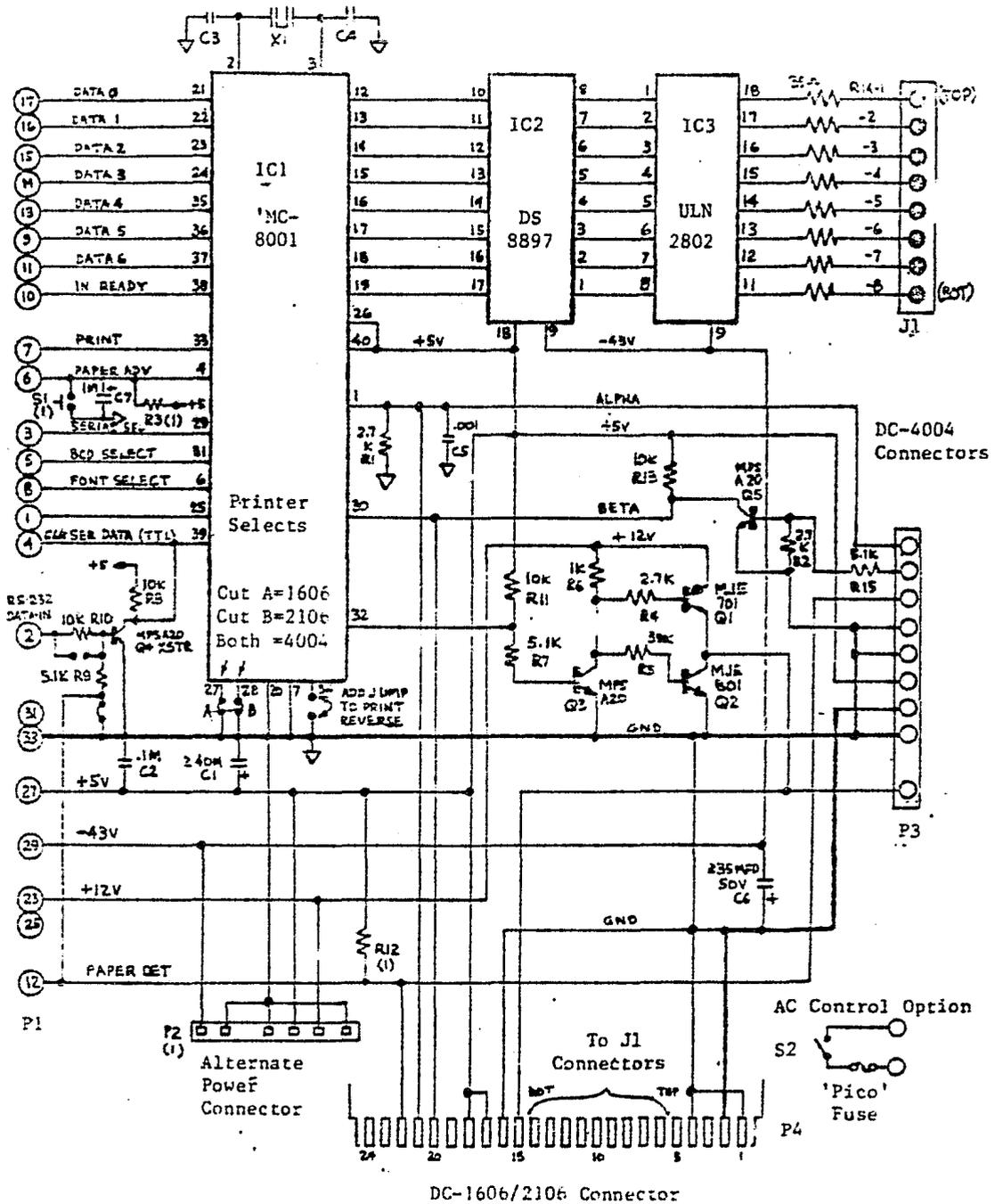
Beta, como alfa es generada por un disparador pasando entre un fototransistor y un haz lumínico. El disparador es hecho a la medida apropiada para indicar el comienzo de una línea de impresión y además indicar la posición en la cual será apagado el motor.

3.2. INTERFASE DE IMPRESION

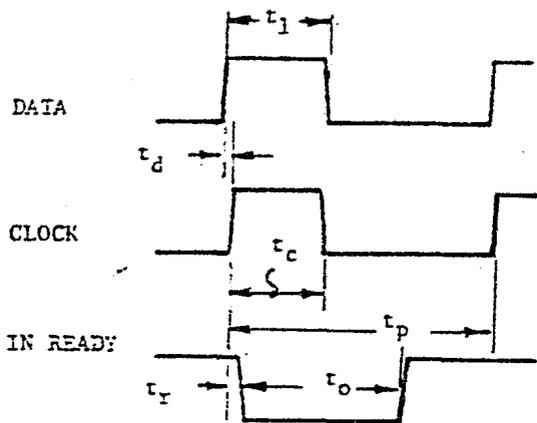
Habiendo seleccionado el mecanismo de impresión de Hycom se usará la tarjeta de interfase IB-40 por su compatibilidad con el mismo.

En entrada de datos paralelos (que puede ser también serial) los pines 9, 11 y 13 a 17 permiten la entrada de caracteres ASCII de 7 bits y señales de control. El dato entra al buffer de la interfase cuando la señal de reloj presenta un nivel alto y durante 25 microseg.

Las señales que cargan el buffer de una línea pueden ser un CP (OD) o un nivel alto en el pin 7 (siendo bajo para el primer caso). La salida READY con un nivel alto (pin 10) indica que un dato puede ser cargado en el buffer y cuando se está imprimiendo pasa a nivel bajo.



INTERFASE DE IMPRESION



*Hold pin 7 'Lo' if software command used.

Timing Relationships
in usec

	MIN	NOM
t_d	0	
t_1	15	25
t_p	75	80
t_c	20	
t_r	11	20
t_o	.	55

DIAGRAMA DE TIEMPOS

Después de presentarse el dato (mínimo 15 microseg.), un nivel alto en la entrada de reloj lo almacena en memoria, en respuesta, la interfase da un nivel bajo en la salida READY (aprox. 55 microseg.), y hasta que regresa a un nivel alto se permite la entrada de un nuevo dato. La velocidad nominal a la que se permite la entrada de datos es de 12 Khz.

Para inicializar la interface bastará con alimentar la línea de +5 Volts (cualquier interrupción en la alimentación causará un RESET). Este también podrá darse con un comando software (1BH) c por medio de la autopruueba.

TABLA DE COMANDOS SOFTWARE

COMANDO	HEX	EFEECTO
Impresión	0D	Guardar en RAM un dato
Enlongado	08	Caracter Enlongado
Limpiar RAM	1B	Limpiar el buffer
Deletear	7F	Borra último Byte (caso serie)
Avance Papel	0D	Cuando no hay entrada de datos avance de papel.

C A P I T U L O 4

CONTROLADORES

4.1. GENERALIDADES

En todo sistema que realice funciones específicas, es necesario contar con una parte que se encargue del control. Dicho control será el encargado de tomar las decisiones en un proceso a seguir.

A lo largo de la historia han existido diferentes tipos de controladores. Estos se pueden dividir para su estudio en:

a) Controladores Alambrados. Fueron los primeros que se usaron en el control de sistemas, y son actualmente usados en aplicaciones de propósito específico, que no requieran gran cantidad de elementos, ya que redundaría en complicación de diseño, alto costo y consumo de potencia.

b) Controladores Microprogramados. Surgieron a partir de la invención de los microprocesadores y son actualmente los más empleados en aplicaciones de propósito general. Con este tipo de control surge la necesidad de crear un microprograma auxiliar del

microprocesador y que le indicará a éste las tareas a seguir y las decisiones a tomar.

4.2. MICROPROCESADORES

En la actualidad los microprocesadores juegan un papel muy importante en sistemas de control. Desde la aparición del primer microprocesador a la fecha, han surgido un sin número de ellos, perfeccionándose las técnicas de diseño y construcción, lo cual los ha hecho más eficientes. Las características a considerar de un microprocesador son: Arquitectura y Tecnología.

4.2.1. Arquitectura. Existen numerosas arquitecturas siendo las más comunes la Von Neumann y la de Stack. Las arquitecturas más sofisticadas definen jerarquía de memoria y de procesos para optimizar el sistema. La arquitectura Von Neumann se basa en la secuencia de instrucciones para el control, llamado programa. Cada instrucción consta de un operador y un operando y la estructura de hardware es: registro de instrucciones, contador de programa, acumulador, ALU, y buses por los cuales se interconectan. El control realiza la supervisión y sincronización de los demás elementos del sistema. La arquitectura de Stack está orientada

hacia lenguajes de alto nivel basándose en un registro del tipo LIFO, llamado Stack Pointer, el cual contiene la dirección de memoria en donde se encuentra la dirección a la cual tiene que regresar el programa. Para optimizar la transferencia de datos y procesos se utilizan arreglos como el DMA (acceso directo a memoria), que es una transferencia directa de datos de un periférico a la memoria principal sin control directo del procesador, otro arreglo es el de prioridad de interrupciones, las cuales pueden ser simple, vectorial o prioritaria. La estructura de hardware es similar a la anterior.

4.2.2. Tecnología. Existen diferentes tecnologías de las cuales depende; la velocidad de operación, consumo de potencia, inmunidad al ruido y costo. Estas tecnologías son: Planar, Bipolar, MOS, CMOS y otras tecnologías MOS como son, SOS (silicio sobre safiro), I²L (lógica de inyección integrada), etc.

Como se puede observar, con la combinación de las diferentes tecnologías y arquitecturas, surgen un sin número de microprocesadores. Dependiendo de la aplicación, se deberá escoger el microprocesador más adecuado, teniendo en cuenta su capacidad, su soporte software y la facilidad de integrar el hardware. Algunos de los microprocesadores más comerciales son:

INTEL 8008. Utiliza la tecnología MOS canal P, con un bus de datos de 8 bits y un bus de direcciones de 14 bits, con el cual se podrá direccionar una memoria de 16kbytes. Tiene un acumulador y dos registros temporales de 8 bits cada uno, 4 banderas (acarreo, paridad, signo y cero), un registro de instrucciones de 8 bits, 6 registros de propósito general de 8 bits, dentro de los cuales están los registros H y L para almacenar direcciones. Además contiene 3 señales (S0, S1 y S2) para definir su palabra de estado.

INTEL 3002. Es un microprocesador de 2 Bits-Slice construido en base a la tecnología SCHOTTKY Bipolar. El 3002 (elemento de procesamiento central) es implementado junto con el 3001 (unidad de control microprogramado) para formar el microprocesador, por tanto, es un microprocesador contenido en dos Circuitos Integrados, por tanto, para implementar una microcomputadora de 16 bits habrá que conectar 8 C.I. 3002 (CPU) y un 3001 que convierte instrucciones de usuario a funciones de control del microprograma.

MOTOROLA 6800. Es un microprocesador de 8 bits construido con tecnología MOS canal N. Consta de un contador de programa de 16 bits, apuntador de Stack, registro indexado de 16 bits, 2 acumuladores de 8 bits cada uno y un registro de código de condición.

TI SBP 0400. Es un microprocesador de 4 Bits-Slice construido con tecnología I2L. Su arquitectura está organizada alrededor de un bus interno múltiple y una memoria PLA, registros de propósito específico y general y un multiplexor de interface. La ventaja de usar un PLA es que emula un gran sistema de cómputo sin degradar las características del microprocesador.

Z-80. Es el primero de la tercera generación de microprocesadores, introducido en abril de 1976, fabricado en base a la tecnología MOS canal N. Requiere de una sola fuente de alimentación de 5 Volts y de una sola señal externa de reloj de 5 Volts. Las señales de salida de control adicionales son generadas por el Z-80, lo cual elimina la necesidad de multiplexar la información de la palabra de estado sobre el bus de datos. El Z-80 es contenido en un integrado de 40 patas tipo DIP.

El Z-80 tiene 16 líneas de dirección con lo cual puede direccionar 64 Kbyte de memoria y 8 líneas de datos bidireccionales. Consta de 6 entradas de control que son: entrada de reloj, \overline{INT} (interrupt request), \overline{NMI} (nonmaskable interrupt), \overline{WAIT} , \overline{BUSREQ} (bus request) y \overline{RESET} , y 8 salidas de control que son: \overline{HALT} , \overline{MEMREQ} (memory request), \overline{IORQ} (input/output request), \overline{RD} (read data), \overline{WP} (write data), \overline{BUSACK} (bus acknowledge) y \overline{RFSH} (refresh).

El Z-80 opera con un reloj de 4 Mhz para tener un mínimo de ciclo de instrucción de un microsegundo.

I-8080. Es el primero de la segunda generación de microprocesadores y fue introducido en noviembre de 1973. Hoy en día, es uno de los más usados en el mundo, fabricado usando la tecnología MOS Canal N, en un circuito integrado de 40 patas, las cuales pueden ser divididas en 5 categorías:

1. Bus de direcciones de 16 líneas.
2. Bus bidireccional de datos de 8 líneas.
3. Fuente de alimentación.
4. Bus de control de entrada.
5. Bus de control de salida.

Requiere de tres fuentes de voltaje que son: +12 V @ 40 mA., +5 V @ 60 mA. y -5 V @ 10 mA., estando estos voltajes referidos a tierra. Seis patas son usadas como entradas de control, que son dos entradas de reloj, HOLD, READY, INT (interrupt) y RESFT. Las dos señales de reloj son requeridas por el 8080 para secuencia y tiempo interno. Estas dos señales de reloj están defasadas en un tren de pulsos de 12 V. La entrada HOLD es mantenida en un cero lógico en la operación normal, y cuando se realiza un acceso

directo a memoria cambia a un nivel alto para que el microprocesador permanezca en un estado de flotación, cesando su actividad poniendo el bus de datos y de direcciones en un estado de alta impedancia. La entrada READY es mantenida en un estado alto durante la operación normal. Un nivel bajo es aplicado a esta entrada para hacer que el procesador entre en un estado de espera. Una vez que está en el estado de espera el procesador cesa su actividad hasta que la entrada READY retorna a un estado alto. Un uso de la entrada READY es sincronizar el 8080 con memorias o periféricos de acceso lento. Después de que una dirección fue puesta en el bus con anterioridad a una lectura de memoria, el 8080 interroga la línea READY para ver si la línea está preparada para responder con el dato del Byte designado. READY se mantiene en un nivel bajo, hasta que el dato se encuentre en el bus de datos.

Una diferencia entre el estado HOLD y el estado WAIT, es que el primero sólo puede ser habilitado después de completar un ciclo de instrucción, y el segundo, puede ser habilitado a la mitad de la ejecución de una instrucción, sólo después de que una nueva dirección sea puesta en el bus. Una segunda diferencia es que en el estado HOLD, el bus de datos y de direcciones permanecen en modo de alta impedancia, mientras que en el estado WAIT no lo están.

Mientras el 8080 está ejecutando un programa, es posible interrumpir su ejecución y brincar a un nuevo programa. Esta interrupción es inicializada por un nivel alto en la entrada INT. El 8080 no reconocerá interrupción alguna mientras esté en el estado de HOLD, y sólo reconocerá una interrupción cuando el flip-flop INTE (interrupt enable) esté puesto. Este FF puede ser limpiado o puesto por medio del programa que el 8080 esté ejecutando.

Durante la ejecución de un programa la entrada RESET es mantenida en un nivel bajo. Para limpiar el 8080 y comenzar la ejecución de un programa en la localidad 000H de memoria, es necesario que la entrada RESET permanezca en nivel alto al menos tres ciclos de reloj. La ejecución de un programa comienza en la localidad 000H, cuando la señal en la entrada RESET regresa a un nivel bajo, y también los FF de INTE y HLDA son limpiados.

El bus de control de salida tiene seis líneas: DRIN, INTE, SYNC, \overline{VR} , WAIT y HLDA. La salida DRIN es usada para indicar a los circuitos externos que el bus de datos está en modo de entrada o de salida.

La salida de INTE (interrupt enable) indica el estado del FF interno INTE, este FF puede ser puesto o limpiado por las instrucciones del microprocesador de habilitar o deshabilitar interrupción. Cuando INTE=0 el microprocesador no reconocerá ninguna solicitud de interrupción.

La salida SYNC (synchronize) aparece al comienzo de cada ciclo de máquina y es usada para indicar la presencia de la palabra de estado en el bus de datos.

La señal \overline{WR} (write) es usada para indicar que un dato está disponible en el bus de datos.

La salida WAIT es usada para indicar que el procesador está en estado de espera, mediante un nivel alto en esta salida. Hay dos formas mediante las cuales la salida WAIT puede ser alta, una es aplicando una señal baja a la entrada READY y la otra es ejecutando la instrucción HALT en el programa que se está ejecutando.

La salida HLDA (hold acknowledge) va a un nivel alto para indicar que el procesador ha reconocido una solicitud de HOLD.

Agregadas a las seis líneas de control de salida del 8080, existen 8 bits adicionales de información de control, que son puestos en el bus de datos al comienzo de cada ciclo de máquina, los cuales forman la palabra de estado y son usados como sigue:

D0: INTA (interrupt acknowledge) indica que el procesador ha respondido a una interrupción de la línea de control INT.

D1: \overline{WO} indica, por medio de un nivel bajo, que el procesador está escribiendo en memoria o que está en un ciclo de salida.

D2: STACK indica que el procesador está realizando una lectura de Stack o una operación de escritura.

D3: HLTA (halt aknowledge) significa que el procesador ha parado como resultado de la ejecución de una instrucción HALT.

D4: OUT indica que un dato está siendo enviado a un puerto.

D5: M1 es una señal para indicar que la CPU se encuentra en un ciclo de FETCH para leer el primer Byte de una instrucción.

D6: INP indica que el bus de dirección contiene el código de selección de un dispositivo de entrada y que un dato para leerse se colocará en el bus de datos cuando se active la señal DBIN.

D7: MEMR indica que el bus de datos se usará para leer datos de la memoria.

La arquitectura del 8080 consta de 7 registros de 8 Bits (incluyendo el acumulador), un contador de programa de 16 Bits, un registro de 16 Bits llamado STACK POINTER (registro externo) que guarda la localidad del STACK en memoria externa. El número de instrucciones básicas del 8080 es de 78 con un tiempo de ciclo de instrucción de 2 microsegundos.

C A P I T U L O 5

ARQUITECTURA DE LA TERMINAL PROPUESTA

5.1. HARDWARE.

La configuración que hemos desarrollado para este proyecto está basada en el funcionamiento general de una terminal, es decir, la transmisión y recepción de datos. Para llevar a cabo esta función es necesario contar con una unidad de control y ciertos periféricos de entra/salida, dicha configuración, es la mínima requerida por una terminal de datos.

Los periféricos de esta terminal han sido explicados con anterioridad, por lo que corresponde ahora, explicar la unidad de control.

Dicha unidad de control la hemos dividido, para su mejor comprensión, en unidad central de procesos con sus circuitos auxiliares y en puertos de entrada o salida y sus agregados.

Los circuitos que corresponden a la unidad central son los siguientes: Microprocesador 8080, Driver y generador de reloj 8224, Controlador del sistema y Driver del bus de datos 8228, Memoria RAM 421 y la Memoria EPROM 2708.

Driver y Generador de Reloj. El 8224 es un integrado que proporciona las señales de reloj $\phi 1$ y $\phi 2$ para la CPU 8080A. Proporciona también las funciones de reposición de energía (Reset), muestreo de la palabra de estado (\overline{STSTB}) y la sincronización de listo (READY) para las memorias lentas. La salida del oscilador se conecta a un generador de reloj divisor entre 9 para generar las señales de reloj $\phi 1$ y $\phi 2$ y el decodificador de disparo proporciona un patrón 2-5-2 en sus salidas $\phi 1$ y $\phi 2$, en otras palabras, la señal $\phi 2$ sube cuando la señal $\phi 1$ baja. La señal $\phi 2$ permanece en alto por 5 unidades de tiempo y la señal $\phi 1$ por 2 unidades. La entrada SYNC cuando toma el nivel alto genera la señal \overline{STSTB} con nivel bajo, logrando así la sincronización entre el 8080A y el 8224. Un nivel bajo en la entrada \overline{RESIN} también genera la señal \overline{STSTB} , pero por un periodo de tiempo más largo. Esto también permite limpiar automáticamente a la 8080A sin necesidad de una pata terminal especial para esta función.

Controlador del Sistema y Driver del Bus de Datos. El 8228 genera todas las señales de control requeridas para la interfaz directa con los componentes RAM, EPROM y de E/S, y también alta capacidad de manejo para soportar sistemas pequeños y medianos. El 8228 proporciona un driver bidireccional para efectuar la función de BUFFER entre el bus de datos del 8080A y los demás componentes y los niveles de voltaje requeridos para la 8080A y las memorias. Al comienzo de cada ciclo de máquina, la 8080A envía una palabra de estado interno por el bus de datos, indicando el tipo de actividad que ocurrirá durante el ciclo. El 8228 almacena esta información en el latch de estado cuando su entrada \overline{STSTB} pasa a nivel bajo. Las salidas del latch de estado están conectadas a un arreglo de compuertas para generar las señales de control. La entrada \overline{STSTB} del 8228 se debe conectar a la salida \overline{STSTB} del 8224, de tal manera que cuando la señal SYNC de la 8080A pasa a un nivel alto se habilita la entrada SYNC del 8224, el cual enviará un nivel bajo por la salida \overline{STSTB} del 8228 para cargar la palabra de estado en su latch.

La palabra de estado en combinación con las salidas DBIN, \overline{WR} y HLDA de la 8080A generan las 5 señales de control: \overline{MEMW} , \overline{MEMR} , $\overline{I/OW}$, $\overline{I/OR}$ e \overline{INTA} , la cual se usa normalmente para permitir la entrada del puerto de la instrucción de la interrupción.

La entrada BUSEN es una entrada síncrona que fuerza, si tiene nivel alto, a los buffers de salida del bus de datos y a los buffers de control al estado de alta impedancia.

Memoria RAM 421. Es una memoria rápida del tipo estático organizada en 1024 palabras de 8 bits, una de sus características incluye baja disipación de potencia (20%) en el modo de no operación. En esta memoria estará almacenado el texto por transmitir y además es auxiliar en el almacenamiento de datos variables de proceso.

Memoria EPROM 2708. Memoria programable de sólo lectura con capacidad de ser borrada con rayos ultravioletas. Opera con una sola fuente y ahorra un 75% en el modo de no operación. En esta memoria estará almacenado el programa monitor, es decir, las instrucciones necesarias para controlar todas las operaciones del microprocesador.

Los circuitos que corresponden a los puertos de entrada o salida son: Decodificador de Puertos 8205, Interfaz de E/S 8212 e Interfaz de Comunicación Programable 8251.

Decodificador de Puertos 8205. Es un decodificador de 3 a 8 binario, el cual dependiendo del valor en binario de las entradas A0, A1 y A2, hace que una de sus 8 salidas pase a nivel bajo para seleccionar uno de los puertos del sistema. Para realizar esta función las entradas de habilitar ($\overline{E1}$ y $\overline{F2}$) deberán tener nivel bajo y la entrada E3 nivel alto.

Interfaz de E/S 8212. Consiste de un circuito de 8 latches de 1 bit con salidas buffer de 3 estados con lógica de selección y control. Incluye también un flip-flop para generar y controlar las solicitudes de interrupción. Una característica importante es la de poder trabajar en multimodo, es decir, se puede implementar en varias configuraciones.

Una parte importante del 8212 es la lógica de control, la cual se maneja con los valores de las 5 entradas: $\overline{DS1}$, DS2, MD, STP y \overline{CLR} , las cuales se definen a continuación:

$\overline{DS1}$, DS2. Selección del dispositivo. Estas dos señales se usan para seleccionar el dispositivo.

MD. Modo. Esta entrada se utiliza para controlar el estado de la salida del buffer y para determinar la fuente que debe generar la señal de reloj para habilitar la entrada de los datos a los latches.

STB. Muestrear. Esta entrada se usa como señal de reloj de los latches de datos para el modo de entrada y para limpiar el flip-flop de solicitud de servicio.

CLR. Limpiar. Esta señal limpia los 8 latches y pone al F/F SR.

Interfaz de Comunicación Programable 8251. La 8251 es un dispositivo conocido como Receptor/Transmisor Síncrono/Asíncrono Universal (USART). La 8251 acepta caracteres de datos paralelos de la CPU y los convierte en un flujo de datos serie para transmisión. Simultáneamente puede recibir un flujo de datos serie y convertirlos en caracteres de datos paralelos para la CPU. Las señales de la 8251 se pueden dividir en 4 categorías:

1. Control e Interfaz con la CPU
2. Entrada Serie
3. Salida Serie
4. Control del Modem

Cuando la CPU envía un dato paralelo a la 8251 éste puede ser un dato al periférico o una palabra de control, así mismo ocurre cuando la 8251 envía un dato a la CPU.

La 8251 se puede acceder como dos puertos, uno que se utilizará para la transferencia de datos y el otro para la transferencia de las palabras de control y de estado. La lógica de selección consiste de dos entradas: \overline{CS} y C/\overline{D} , que combinadas con los valores de las entradas \overline{RD} y \overline{WR} indican la dirección de la transferencia del dato y la interpretación del mismo.

La 8251 tiene otras dos señales adicionales; RESET y CLK. La primera es una señal típica del sistema, que cuando tiene nivel alto permanece en estado inactivo. La señal CLK es una señal de entrada de reloj que se utiliza únicamente para los tiempos internos del USART, pero debe ser a lo menos 13 veces mayor que la velocidad de transmisión o recepción.

Los datos serie se reciben en la terminal PxD y las transiciones bajo-alto del reloj \overline{RxC} permiten leer los bits de un dato. La lógica de recepción utiliza la señal de control RxRDY. Esta salida toma el nivel alto en el momento en que el buffer de entrada envía el dato al buffer de salida.

El dato serie se envía por la terminal TxD. La velocidad de transmisión es controlada por la señal de reloj que alimenta la entrada TxC. Las transiciones alto-bajo del reloj TxC permiten la transferencia de los bits del dato. Existen 2 señales de control asociadas con la lógica de transmisión; TxRDY y TxE. La señal TxRDY pasa a nivel alto tan pronto el contenido del buffer de entrada se ha corrido al buffer de salida. La salida TxE se pone en alto tan pronto como el dato en el buffer de salida se ha enviado al periférico y permanece en alto mientras no se corra un nuevo dato del buffer de entrada al buffer de salida.

Las señales de control del MODEM son estandar. La 8251 utiliza la salida $\overline{\text{DTR}}$ para indicar que se encuentra lista y la entrada $\overline{\text{DSR}}$ la usa para probar el estado en el que se encuentra el MODEM. Una vez que la 8251 y el MODEM se encuentran listos, la 8251 inicia la transmisión enviando una solicitud de transmitir al MODEM por medio de $\overline{\text{RTS}}$ y la entrada $\overline{\text{CTS}}$ la utiliza el MODEM para indicarle a la 8251 que inicie la transmisión.

Antes de comenzar la transmisión o recepción de datos la 8251 se deberá cargar con un conjunto de palabras de control, estas palabras de control se dividen en dos formatos:

1. Instrucción de Modo.
2. Instrucción de Comando.

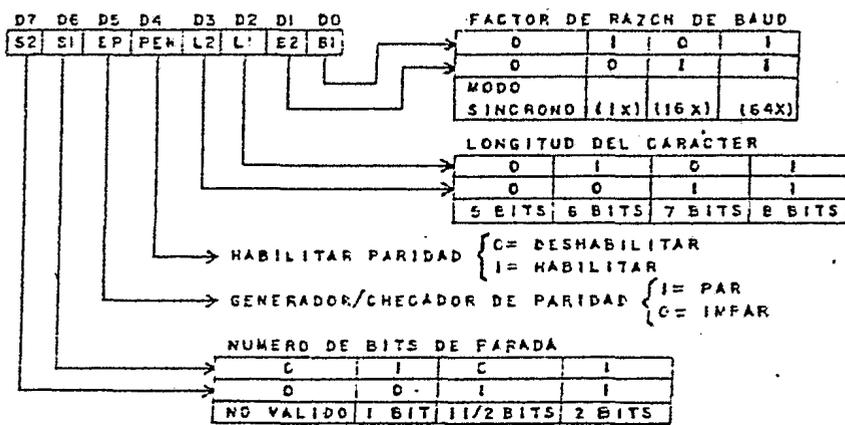
El primer formato define las características operacionales generales de la 8251. Una vez que la Instrucción de modo fue enviada a la 8251 por la CPU, se pueden insertar instrucciones de comando.

La Instrucción de Comando define una palabra de estado que se usa para controlar la operación actual de la 8251.

Siempre que la CPU envía un carácter de dato, la 8251 le agrega automáticamente un bit de arranque y un número programado de bits de parada. También un bit de paridad par o impar se inserta antes del bit de parada, según se defina en la instrucción de modo. El dato serie se corre a una razón igual a 1, 1/16 ó 1/64 de la de TxC, como se hubiere definido durante la Instrucción de Modo.

Si ocurre un error de paridad en la recepción, la bandera de error de paridad toma el nivel alto. Si un carácter previo no ha sido atrapado por la CPU, el nuevo carácter lo reemplaza en el

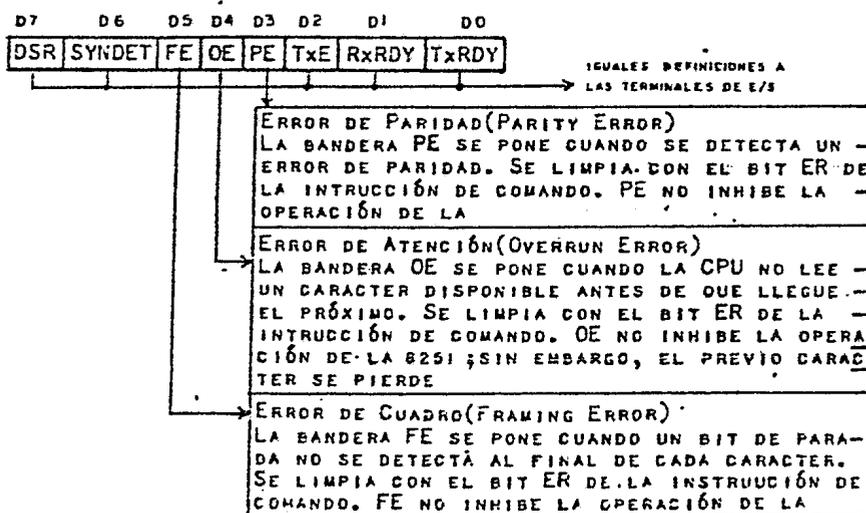
buffer de E/S y la bandera de OVERRUN se levanta. Todas las banderas de error se pueden limpiar con una Instrucción de Comando.



FORMATO DE LA INSTRUCCION DE MODO ASINCRONO

D0	TxE	HABILITAR TRANSMISIÓN(TRANSMIT ENABLE) 1 = HABILITAR 0 = DESHABILITADO
D1	DTR	TERMINAL DE DATOS LISTA(DATA TERMINAL READY) "ALTO" FORZA A LA SALIDA DTR A CERO
D2	RxE	HABILITAR RECEPCIÓN(RECEIVE READY) 1 = HABILITADO 0 = DESHABILITADO
D3	SBRK	ENVIAR CARACTER DE RUPTURA(SEND BREAK CHARACTER) 1 = FORZA A TxD A NIVEL BAJO 0 = OPERACIÓN NORMAL
D4	ER	LIMPIAR ERRORES(ERROR RESET) 1 = LIMPIA TODAS LAS BANDERAS DE ERROR; PE, OE, FE
D5	RTS	SOLICITUD PARA ENVIAR(REQUEST TO SEND) "ALTO" FORZA A LA SALIDA RTS A CERO
D6	IR	LIMPIAR INTERNO(INTERNAL RESET) "ALTO" REGRESA A LA B251 A FORMATO DE INST. DE MODO
D7	EH	ENTRA MODO HUNT(ENTER HUNT MODE) 1 = HABILITA PARA BUSCAR CARACTERES DE SINCROÑIA (SYNC)

FORMATO DE LA INSTRUCCION DE COMANDO



FORMATO DE LA LECTURA DE ESTADOS

En los sistemas de comunicación de datos es necesario examinar los estados del dispositivo para verificar si han ocurrido errores u otras condiciones que requieren la atención del procesador. El 8251 tiene un registro de banderas que permiten leer los estados del dispositivo en cualquier tiempo.

5.1.1. OPERACION HARDWARE.

Para poder explicar el funcionamiento operativo de la terminal, desde un punto de vista técnico, llevaremos la secuencia de acciones que realiza la terminal en la recepción y transmisión de datos.

Basándonos en la idea de que la transmisión es primero que la recepción de información, la secuencia será la siguiente:

1. Encendido de la terminal.

Al momento de encender la terminal existirá un RESET automático, ya que en un tiempo inicial el capacitor C1 conduce, esto trae como consecuencia que la entrada RESIN tome un nivel bajo, originando a su vez que la entrada RES de la 8080 tome un nivel alto, inicializando así el programa monitor.

La primera función de este programa será la de limpiar la memoria RAM y registros de la 8080, como también la de definir parámetros constantes.

2. Espera por interrupción.

Una vez realizada la rutina de inicialización la 8080 esperará una interrupción, ya sea de transmisión o de recepción. Suponiendo que se realizará una interrupción de transmisión, el teclado lo efectuará por medio de un STROBE, éste entrará al CI8, el cual, pondrá en su salida $\overline{\text{INT}}$ un nivel bajo, generando así la entrada del dato (RST 7) y la interrupción de la CPU.

3. Atención de Interrupción.

Al ser interrumpida la 8080 pondrá un nivel bajo en la salida $\overline{\text{INTA}}$ del CI5, sacando éste al bus de datos la información para el RST 7. Con dicho RESTART el programa efectuará un brinco incondicional a la rutina de atención del teclado. Esta rutina hará una lectura del puerto correspondiente, una escritura al puerto de impresión y una edición en memoria RAM.

4. Atención al Teclado.

En el inicio de esta rutina, se hará una lectura del dato presentado por el teclado de la siguiente manera: la instrucción IN activará la señal de control $\overline{\text{I/OR}}$, la cual habilitará al IC7,

para decodificar la dirección del puerto, por medio de A2, A1 y A0. Esta decodificación permitirá habilitar el CI10 y presentará el dato para ser leído por el microprocesador. Una vez leído el dato, la 8080 realizará la escritura del mismo, por medio de dos instrucciones OUT, una para la escritura en impresión y otra en memoria. La primera activará la señal $\overline{I/O}$, para habilitar el decodificador de puertos y seleccionar el de impresión. Al realizar esta actividad, hay que tener en consideración que el mecanismo de impresión es mucho más lento que la CPU, por lo cual tendrá que investigar ésta, si el dato ya fue escrito. Esto se logra mediante la seña \overline{RDY} del mecanismo. Cuando esta señal está en bajo, el mecanismo estará listo para recibir otro dato, de lo contrario el puerto estará deshabilitado. La CPU se dará cuenta de ello por medio de la salida \overline{INT} de CI19, dicha señal permanecerá en bajo hasta que el dato haya sido escrito. La segunda instrucción OUT activará la señal de control \overline{MEMR} , para activar la memoria RAM con el objeto de escribir el dato en ella.

5. Transmisión.

Cuando se teclea el comando de transmisión, todos los caracteres que conforman el texto almacenado en memoria RAM, serán enviados a la computadora central a través de una línea de transmisión, de

la siguiente forma: El microprocesador lee de los switches externos la instrucción de modo, que le indicará al USART el modo de transmisión (baud rate, sinc/asinc, bits de parada, etc.). El dato estará permanentemente entrando al 8212 y sólo cuando se hace la lectura del puerto respectivo, el dato entra al canal de los mismos. Dicho dato es enviado al USART como palabra de control, al habilitar el puerto correspondiente, en su entrada C/D. Después de realizar esto, la 8080 enviará a la 8251, un dato previamente establecido (instrucción de comando), el cual le indicará las condiciones en que deberá operar, la forma de comunicárselo será de igual manera que para la instrucción de modo. Establecidas la forma y condiciones de operar, se realizará una investigación de las condiciones en las que se encuentra el USART, para poder tomar la decisión de transmitir o esperar. Una vez preparada la 8251 para transmitir, la CPU direcciona el primer carácter del texto leyéndolo de la memoria PAM, colocándolo en el bus de datos, para que sea leído por la 8251 al momento de ser habilitada con las señales \overline{CS} y \overline{DATA} y siendo enviado al MODEM en la bajada del reloj TxC como un flujo de datos serie en la salida TxD, agregándole automáticamente un bit de arranque, normalmente de nivel bajo, y un número programado de bits de parada normalmente altos, como también un bit de paridad par o impar se inserta antes del bit de parada.

6. Recepción

En el momento que un dato llega al USART por medio del MODEM, la salida RxDY de la 8251 se pondrá en alto, siendo el STROFF del CI8, para generar una interrupción (RST 5) en la 8080. Al ser identificada esta interrupción la CPU ejecutará la rutina de recepción. En esta rutina ejecutará los mismos pasos que en la de transmisión, con la salvedad de que la palabra de estado es utilizada para saber si hubo error en la recepción. Los datos recibidos no se procesarán, exclusivamente se imprimirán de la forma antes mencionada. En caso de que en la recepción exista un error (de cuadro, de paridad o de alcance), éste no será corregido, solamente se desplegará en el mecanismo de impresión, por lo que se sugerirá repetir la consulta.

BUS DE DIRECCIONES

BUS DE DATOS

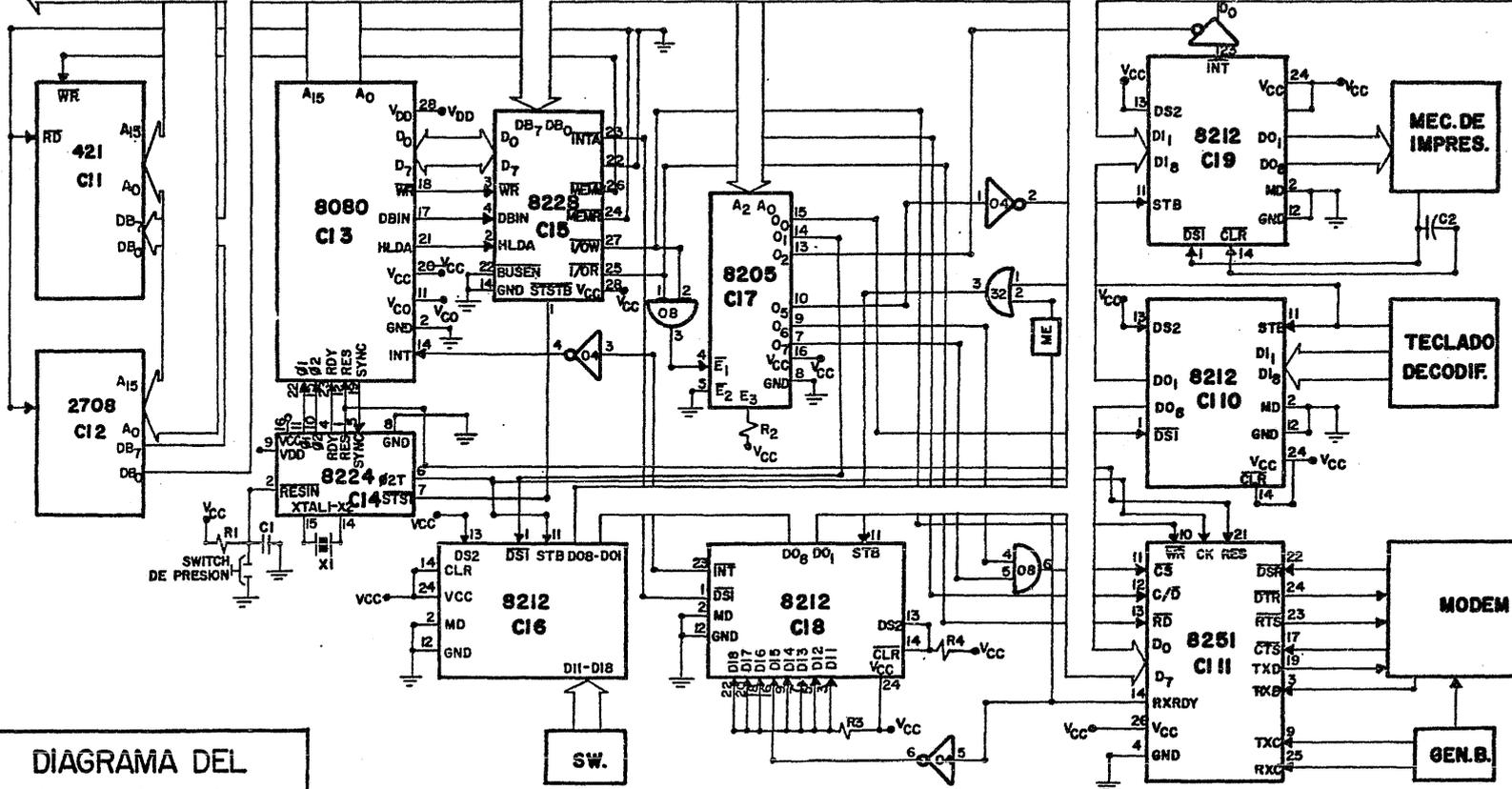
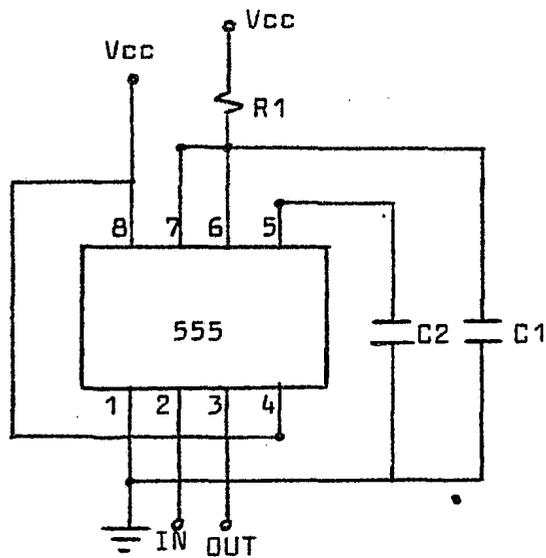


DIAGRAMA DEL CONTROLADOR

G. ITURBIDE M.A. OLGUIN

NOTA: CONSULTAR ANEXO

1. El bloque M.F. corresponde al siguiente circuito:



2. La línea de dirección A14 es la habilitadora de las memorias PAM y EPROM. La línea A15 no se utiliza.
3. Es recomendable utilizar capacidad distribuida para eliminar ruido lógico en los circuitos integrados.
4. X1 es un cristal oscilador que deberá cumplir con la especificación de ser 9 veces mayor en frecuencia que con la que se desea trabajar.

5.2. SOFTWARE.

En todo sistema controlado con microprocesador existe la necesidad de contar con un programa monitor. Dicho programa será el traductor entre las instrucciones ejecutadas por el operador y las que realizará el sistema.

El software realizado en este proyecto está almacenado en una memoria EPROM, siendo el cerebro, por así decir de la terminal.

Este programa realizará los procedimientos necesarios para atender el teclado y la impresora, así como la transmisión y recepción de datos.

El programa cuenta con una característica especial, la de poder editar textos en la misma terminal. Esta característica es importante ya que el tiempo de máquina (o el tiempo que una terminal tenga ocupado un canal de la computadora) es muy costoso. El editor podrá corregir errores en la terminal sin ocupar tiempo de máquina, ya que de no existir éste, las correcciones se harían en la computadora.

Los caracteres introducidos a la terminal por medio del teclado pueden dividirse en dos tipos, los caracteres imprimibles, que son todos los caracteres que existen en la formación de un texto, y los comandos de control, que sin imprimirse ejecutan una función específica.

Los comandos de control los hemos subdividido en cinco grupos, que son: los de Control mismos, los de Acción, los de Posición, el de Impresión y el de Transmisión.

1. Control.

- a) CR, es el comando para inicialización de una nueva línea.
- b) BS, es el comando para borrar el último carácter escrito.

2. Acción.

- a) Inserción con este comando se puede insertar un carácter dentro de un texto.
- b) Deletear, con este comando se puede borrar un carácter sin dejar espacio en donde fue borrado.

3. Posición.

- a) Empiezo, con este comando se deja la posibilidad de escribir desde el principio del texto.
- b) Regreso, deja la posibilidad de posesionarse sobre cualquier caracter del texto, regresando de uno en uno.
- c) Avance, deja la posibilidad de posesionarse sobre cualquier caracter del texto, avanzando de uno en uno y siendo el tope máximo el último caracter imprimido.
- d) Fin, se localiza sobre una posición adelante del último caracter impreso.

4. Impresión. Con este comando manda a impresión la parte del texto deseada para su verificación sin habilitar la transmisión, imprime desde la posición actual hasta el último caracter impreso.

5. Transmisión. Salida por MODEM desde el inicio hasta el fin del texto.

A continuación se da una tabla de los comandos de control con su modo operativo, su representación ASCII y representación hexadecimal.

		MODO		
		OPTIVO	ASCII	HEX
CONTROL	CR	BS	CR	0D
	RS	BS	RS	08
ACCION	INSERTAR	^I	HT	09
	DELETEAR	^D	EDT	04
POSICION	EMPIEZO	^D	STX	02
	REGRESC	^R	DC2	12
	AVANCE	^K	VT	0F
	FIN	^E	ENC	05
IMPRESION	IMPRESION	^P	DLE	10
TRANSMISION	TRANSMISION	^T	DC4	14

^ Indica (en modo operativo) la acción de dos teclas simultáneamente, la tecla CTRL y la que en seguida se indica.

5.2.1. PROGRAMA

DIREC.	ETIQ.	CODIGO	NEM.	COMENTARIO
0000	princ.	C3	JMP	Salto a
0001		00		0100
0002		01		Inicio
0020	rst 5	C3	JMP	Salto a
0021		84		C284
0022		02		*reces
0038	rst 7	C3	JMP	Salto a
0039		22		0122
003A		01		Teclado
0090	subcare	DB	IN	Entrada
0091		02		A1 C2
0092		E6	ANI	A=A.01
0093		01		
0094		CA	JZ	A=C
0095		90		Princa
0096		0C		A la 0000
0097		C9	RET	Recesa
0100	inicio	31	LXI SP	SP=FFFF
0101		FF		
0102		FF		
0103		11	LXI D	D-F=0000
0104		00		
0105		00		
0106		21	LXI H	H-I=FC00
0107		00		
0108		FC		
0109		01	LXI B	B-C=03FC
010A	F0			
010B	03			
010C	blancos	36	MVI M	blanco
010D		20		20 ASCII
010E		23	INC H	
010F		0B	DCX,B	
0110		7E	MOV A,B	
0111		CA	JZ	A=0
0112		17		brinca a
0113		01		0117
0114		C3	JMP	brinca
0115		0C		010C
0116		01		
0117	79	MOV A,C	blancos	
0118	C2	JNZ	a≠C	
0119	0C		brinca	
011A	01		blancos	

011B		06	MVI B	P=48
011C		30		
011D	loop int	FB	FI	hab.int.
011E		00	NOP	
011F		C3	JMP	brinca
0120		1D		loop int.
0121		01		
0122	teclado	F3	DI	des.int.
0123		DB	INPUT	entrada
0124		00		al 00
0125		4F	MOV C,A	
0126		E6	ANI	A=A 60
0127		60		
0128		CA	JZ	A=0
0129		4F		brinca
012A		01		014F
012B		78	MOV A,B	
012C		CA	JZ	B=0
012D		3C		brinca
012E		01		013C
012F	subimpca	05	DCR B	
0130		13	INX D	
0131		79	MOV A,C	
0132		77	MOV M,A	
0133		23	INX H	
0134		D3	OUT	salida
0135		05		al 05
0136		CD	CALL	subcare
0137		90		0090
0138		00		
0139		C3	JMP	brinca
013A		1D		loop int.
013B		01		011D
013C	auto cr	3E	MVI A	A=0D
013D		0D		
013E		D3	OUT	salida
013F		05		al 05
0140		CD	CALL	subcare
0141		90		0090
0142		00		
0143		3E	MVI A	A=0A
0144		0A		
0145		D3	OUT	salida
0146		05		al 05
0147		CD	CALL	subcare
0148		90		0090
0149		00		
014A		06	MVI B	B=48
014B		30		
014C		C3	JMP	brinca
014D		2F		subimpca
014E		01		012F

014F	idctrol	79	MOV A,C	
0150		E6	ANI	A=A:72
0151		72		ident.cr
0152		CA	JZ	A=0
0153		8E		brinca
0154		01		a la 018F
0155		79	MOV A,C	
0156		E6	ANI	A=A:77
0157		77		ident.bs
0158		CA	JZ	A=0
0159		A4		brinca
015A		01		a la 01A4
015B		79	MOV A,C	
015C		E6	ANI	A=A:76
015D		76		ident.^I
015E		CA	JZ	A=0
015F		B9		brinca
0160		01		a la 01B9
0161		79	MOV A,C	
0162		E6	ANI	A=A:7F
0163		7B		ident.^D
0164		CA	JZ	A=0
0165		EF		brinca
0166		01		a la 01EF
0167		79	MOV A,C	
0168		E6	ANI	A=A:7D
0169		7D		ident.^F
016A		CA	JZ	A=0
016B		15		brinca
016C		02		a la 0215
016D		79	MOV A,C	
016E		E6	ANI	A=A:6D
016F		6D		ident.^P
0170		CA	JZ	A=0
0171		1B		brinca
0172		02		a la 021F
0173		79	MOV A,C	
0174		E6	ANI	A=A:74
0175		74		ident.^K
0176		CA	JZ	A=0
0177		2B		brinca
0178		02		a la 022F
0179		79	MOV A,C	
017A		E6	ANI	A=A:7A
017B		7A		ident.^E
017C		CA	JZ	A=0
017D		3E		brinca
017E		02		a la 023E
017F		79	MOV A,C	
0180		E6	ANI	A=A:6F
0181		6F		ident.^P
0182		CA	JZ	A=0

0183		46		brinca
0184		02		a la 0246
0185		79	MOV A,C	
0186		E6	ANI	A=A-6F
0187		6B		ident.^T
0188		CA	JZ	A=0
0189		5F		brinca
018A		02		a. la 025F
018F		C3	JMP	brinca a
018C		1D		loop int
018D		01		011D
018E	*cr	79	MOV A,C	
018F		77	MOV M,A	
0190		23	INX F	
0191		D3	OUT	salida
0192		05		al 05
0193		CD	CALL	subcare
0194		90		
0195		00		
0196		3E	MVI A	A=0A
0197		0A		LF
0198		77	MOV M,A	
0199		23	INX H	
019A		D3	OUT	salida
019E		05		al 05
019C		CD	CALL	subcare
019D		90		
019E		00		
019F		06	MVI B	col=48
01A0		30		
01A1		C3	JMP	brinca a
01A2		1D		loop int
01A3		01		
01A4	*bs	7D	MOV A,L	
01A5		CA	JZ	A=0
01A6		B0		brinca
01A7		01		a la 01B0
01A8		2B	DCX F	
01A9		05	DCR P	
01AA		1B	DCX D	
01AB		36	MVI M	M=blanco
01AC		20		
01AD		C3	JMP	brinca
01AE		1D		loop int
01AF		01		
01B0	comp bs	7C	MOV A,H	
01B1		E6	ANI	A=A-03
01B2		03		iden FC00
01B3		CA	JZ	A=0
01B4		1D		brinca
01P5		01		loop int
01E6		C3	JMP	brinca

01B7		A8		a dec h5
01B8		01		
01B9	*insert	D5	PUSH D	
01BA		7B	MOV A,F	
01BE		C6	ADI	A=A+FF
01BC		FF		
01BD		5F	MOV F,A	
01BE		7A	MOV A,D	
01BF		CE	ACI	A=A+FF+CY
01C0		FB		
01C1		57	MOV D,A	
01C2		7F	MOV A,E	
01C3		95	SUB L	A=A-I
01C4		5F	MOV E,A	
01C5		7A	MOV A,D	
01C6		9C	SBB H	A=A-H-CY
01C7		57	MOV D,A	
01C8		E1	POP H	
01C9		E5	PUSH H	
01CA		7D	MOV A,L	
01CB		C6	ADI	A=A+FF
01CC		FE		
01CD		6F	MOV L,A	
01CE		7C	MOV A,H	
01CF		CE	ACI	A=A+FF+CY
01D0		FB		
01D1		67	MOV H,A	
01D2	comp ^I	7B	MOV A,F	
01D3		CA	JZ	A=0
01D4		DF		brinca
01D5		01		cod ^I
01D6	inserta	7E	MOV A,M	
01D7		23	INX H	
01D8		77	MOV M,A	
01D9		2F	DCX H	
01DA		2B	DCX H	
01DB		1B	DCX D	
01DC		C3	JMP	brinca
01DD		D2		comp ^I
01DE		01		01D2
01DF	cod ^I	7A	MOV A,D	
01E0		CA	JZ	A=0
01E1		E6		brinca
01E2		01		real ^I
01E3		C3	JMP	brinca
01E4		D6		a inserta
01E5		01		
01E6	real ^I	23	INX H	
01E7		36	MVI M	M=20
01E8		20		
01E9		D1	POP D	
01EA		13	INX D	

01EB		03	INX B	
01EC		C3	JMP	brinca a
01ED		1D		loop int
01EE		01		
01EF	*delete	D5	PUSH D	
01F0		7B	MOV A,E	
01F1		C6	ADI	A=A+FE
01F2		FE		
01F3		5F	MOV E,A	
01F4		7A	MOV A,D	
01F5		CE	ACI	A=A+FB+CY
01F6		FB		
01F7		57	MOV D,A	
01F8	comp h	BC	CMP H	H=A
01F9		CA	JZ	brinca a
01FA		05		comp L
01FB		02		0205
01FC	deletea	23	INX H	
01FD		7E	MOV A,M	
01FE		2B	DCX H	
01FE		77	MOV M,A	
0200		23	INX H	
0201		7A	MOV A,D	
0202		C3	JMP	brinca a
0203		F8		comp h
0204		01		
0205	comp L	7B	MOV A,E	
0206		BD	CMP L	
0207		CA	JZ	brinca a
0208		0D		ejecuta
0209		02		
020A		C3	JMP	brinca a
020B		FC		deletea
020C		01		
020D	ejecuta	36	MVI M	M=20
020E		20		
020F		D1	POP D	
0210		1B	DCX D	
0211		0B	DCX B	
0212		C3	JMP	brinca a
0213		1D		loop int
0214		01		
0215	*empiezo	21	LXI H	HL=FC00
0216		00		
0217		FC		
0218		C3	JMP	brinca
0219		1D		loop int
021A		01		
021B	*regreso	7D	MOV A,L	
021C		CA	JZ	A=0
021D		23		brinca a
021E		02		compara

021F	decrem	2F	DCX H	
0220		C3	JMP	brinca
0221		1D		loop int
0222		01		
0223	compara	7C	MOV A,H	
0224		E6	ANI	A=A:03
0225		CA	JZ	brinca a
0226		1D		loop int
0227		01		
0228		C3	JMP	brinca a
0229		1F		decrem
022A		02		
022B	*avance	7A	MOV A,D	
022C		C6	ADI	A=A+FC
022D		FC		
022E		BC	CMP H	
022F		CA	JZ	A=0
0230		36		brinca a
0231		02		comp L
0232	ejecuta	23	INX H	
0233		C3	JMP	brinca a
0234		1D		loop int
0235		01		
0236	comp L	7B	MOV A,E	
0237		BD	CMP L	
0238		CA	JZ	A=0
0239		1D		brinca
023A		01		loop int
023B		C3	JMP	brinca a
023C		32		ejecuta
023D		02		
023E	*fin	6B	MOV L,E	
023F		7A	MOV A,D	
0240		C6	ADI	A=A+FC
0241		FC		
0242		67	MOV H,A	
0243		C3	JMP	brinca a
0244		1D		loop int
0245		01		
0246	*regreso	7B	MOV A,E	
0247		BD	CMP L	
0248		CA	JZ	brinca a
0249		55		sique
024A		02		
024B	ejecuta	7E	MOV A,M	
024C		D3	OUT	salida
024D		05		al 05
024E		CD	CALL	subcare
024F		90		
0250		00		
0251		23	INX H	
0252		C3	JMP	brinca a

0253		46		*regreso
0254		02		
0255	sigue	7A	MOV A,D	
0256		C6	ADI	A=A+FC
0257		FC	-	
0258		BC	CMP F	
0259		CA	JZ	A=0
025A		1D		brinca a
025B		01		loop int
025C		C3	JMP	brinca a
025D		4E		ejecuta
025E		02		
025F	*transm	21	LXI H	
0260		00		HL=FC00
0261		FC		
0262		DB	IN	entrada
0263		01		al 01
0264		D3	OUT	salida
0265		07		al 07
0266	regreso	3E	MVI A	
0267		36		A=36
0268		D3	OUT	salida
0269		07		al 07
026A	listo tr	DB	IN	entrada
026B		07		al 07
026C		E6	ANI	A=A-01
026D		01		
026E		CA	JZ	A=0
026F		6A		brinca a
0270		02		listo tr
0271		1E	DCX D	
0272		7B	MOV A,E	
0273		CA	JZ	A=0
0274		7D		brinca a
0275		02		sigue
0276	trx	7E	MOV A,M	
0277		D3	OUT	salida
0278		06		al 06
0279		23	INX H	
027A		C3	JMP	brinca a
027B		66		regreso
027C		02		
027D	sigue	7A	MOV A,I	
027E		CA	JZ	A=0
027F		00		brinca a
0280		01		inicio
0281		C3	JMP	brinca a
0282		76		trx
0283		02		
0284	*recep	F3	DI	
0285		DE	IN	entrada
0286		01		al 01

0287	D3	OUT	salida
0288	07		al 07
0289	3E	MVI A	A=36
028A	36		I.comando
028B	D3	OUT	salida
028C	07		al 07
028D	DB	IN	entrada
028E	07		al 07
028F	4F	MOV C,A	
0290	F6	ANI	A=A*08
0291	08		
0292	CA	JZ	A=0
0293	4A		brinca a
0294	03		rcx
0295	79	MOV A,C	
0296	F6	ANI	A=A*10
0297	10		
0298	CA	JZ	A=0
0299	4A		brinca
029A	03		a rcx
029B	79	MOV A,C	
029C	E6	ANI	A=A*20
029D	20		
029E	CA	JZ	A=0
029F	4A		brinca a
02A0	03		rcx
02A1	3E	MVI A	A=76
02A2	76		
02A3	D3	OUT	salida
02A4	07		al 07
02A5	DE	IN	entrada
02A6	01		al 01
02A7	D3	OUT	salida
02A8	07		al 07
02A9	3E	MVI A	A=36
02AA	36		I.comando
02AB	D3	OUT	salida
02AC	07		al 07
02AD	3E	MVI A	A=2A
02AE	2A		
02AF	D3	OUT	
02B0	05		
02B1	CD	CALL	subcare
02B2	90		
02B3	00		
02B4	3E	MVI A	A=20
02B5	20		
02B6	D3	OUT	
02B7	05		
02B8	CD	CALL	subcare
02B9	90		
02BA	00		
02BB	3E	MVI A	

02BC	45		A=45
02BD	D3	OUT	
02BE	05	.	
02BF	CD	CALL	subcare
02C0	90		
02C1	00		
02C2	3E	MVI A	A=52
02C3	52		
02C4	D3	OUT	
02C5	05		
02C6	CD	CALL	subcare
02C7	90		
02C8	00		
02C9	3E	MVI A	A=52
02CA	52		
02CB	D3	OUT	
02CC	05		
02CD	CD	CALL	subcare
02CE	90		
02CF	00		
02D0	3E	MVI A	A=4F
02D1	4F		
02D2	D3	OUT	
02D3	05		
02D4	CD	CALL	subcare
02D5	90		
02D6	00		
02D7	3E	MVI A	A=52
02D8	52		
02D9	D3	OUT	
02DA	05		
02DB	CD	CALL	subcare
02DC	90		
02DD	00		
02DE	3E	MVI A	A=20
02DF	20		
02E0	D3	OUT	
02E1	05		
02E2	CD	CALL	subcare
02E3	90		
02E4	00		
02E5	3E	MVI A	A=45
02E6	45		
02E7	D3	OUT	
02E8	05		
02E9	CD	CALL	subcare
02EA	90		
02EB	00		
02EC	3F	MVI A	A=4F
02ED	4E		
02EE	D3	OUT	
02EF	05		

02F0	CD	CALL	subcare
02F1	90		
02F2	00		
02F3	3E	MVI A	A=20
02F4	20		
02F5	D3	OUT	
02F6	05		
02F7	CD	CALL	subcare
02F8	90		
02F9	00		
02FA	3E	MVI A	A=52
02FB	52		
02FC	D3	OUT	
02FD	05		
02FE	CD	CALL	subcare
02FF	90		
0300	00		
0301	3E	MVI A	A=45
0302	45		
0303	D3	OUT	
0304	05		
0305	CD	CALL	subcare
0306	90		
0307	00		
0308	3E	MVI A	A=43
0309	43		
030A	D3	OUT	
030B	05		
030C	CD	CALL	subcare
030D	90		
030E	00		
030F	3E	MVI A	A=45
0310	45		
0311	D3	OUT	
0312	05		
0313	CD	CALL	subcare
0314	90		
0315	00		
0316	3E	MVI	A=50
0317	50		
0318	D3	OUT	
0319	05		
031A	CD	CALL	subcare
031B	90		
031C	00		
031D	3E	MVI A	A=43
031E	43		
031F	D3	OUT	
0320	05		
0321	CD	CALL	subcare
0322	90		
0323	00		

0324		3E	MVI A	A=49
0325		49		
0326		D3	OUT	
0327		05		
0328		CD	CALL	subcare
0329		90		
032A		00		
032B		3E	MVI A	A=4F
032C		4F		
032D		D3	OUT	
032E		05		
032F		CD	CALL	subcare
0330		90		
0331		00		
0332		3E	MVI A	A=4F
0333		4E		
0334		D3	OUT	
0335		05		
0336		CD	CALL	subcare
0337		90		
0338		00		
0339		3E	MVI A	A=20
033A		20		
033B		D3	OUT	
033C		05		
033D		CD	CALL	subcare
033E		90		
033F		00		
0340		3E	MVI A	A=2A
0341		2A		
0342		D3	OUT	
0343		05		
0344		CD	CALL	subcare
0345		90		
0346		00		
0347		C3	JMP	brinca
0348		1D		loop int
0349		01		
034A	rcx	DB	IN	entrada
034B		06		al 06
034C		D3	OUT	salida
034D		05		al 05
034E		CD	CALL	subcare
034F		90		
0350		00		
0351		C3	JMP	brinca a
0352		1D		loop int
0353		01		

DIAGRAMA GENERAL A BLOQUES

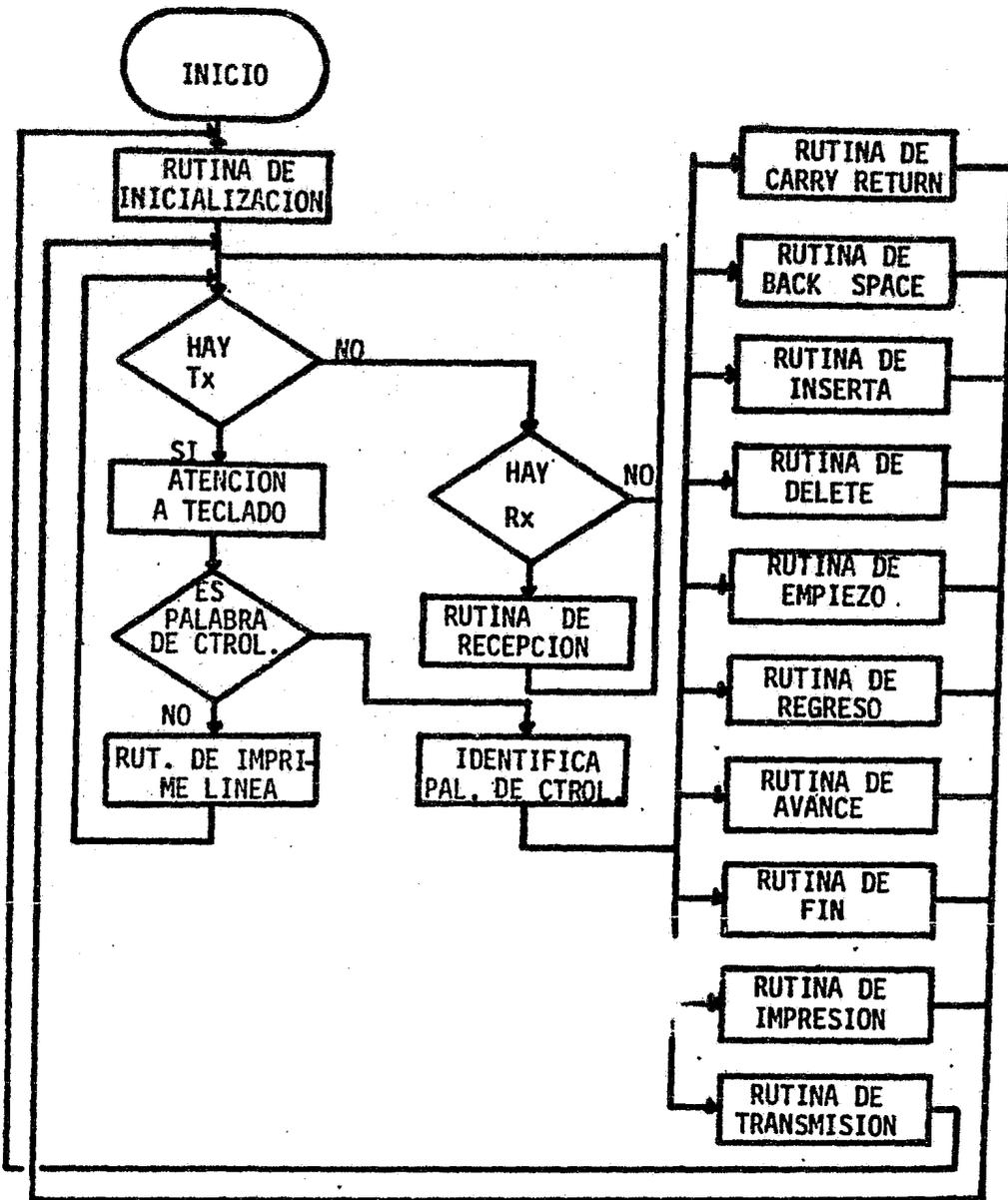


DIAGRAMA RUTINA DE INICIALIZACION A BLOQUES

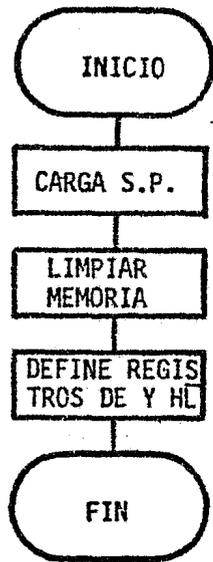


DIAGRAMA RUTINA IMPRIME LINEA A BLOQUES

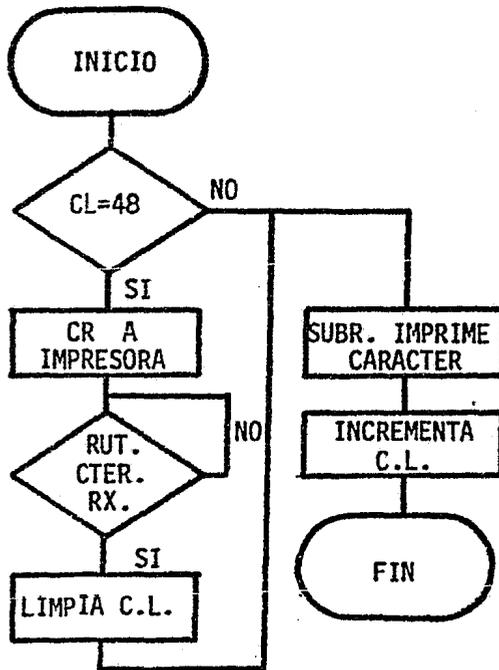


DIAGRAMA Rutina B.S. A BLOQUES

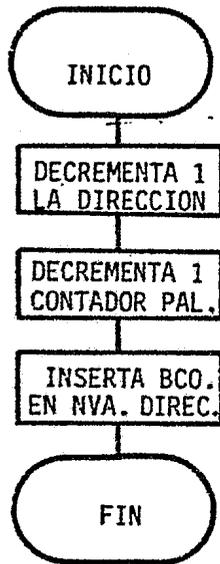


DIAGRAMA Rutina INSER. A BLOQUES

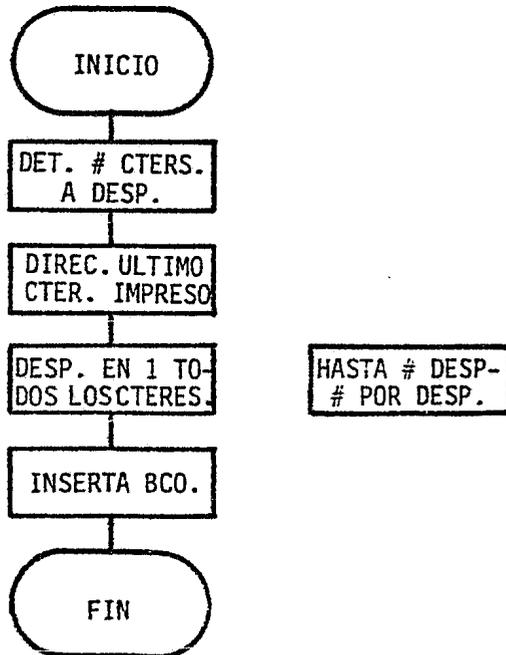


DIAGRAMA SUBROUTINA IMPRIME CARACTER A BLOQUES

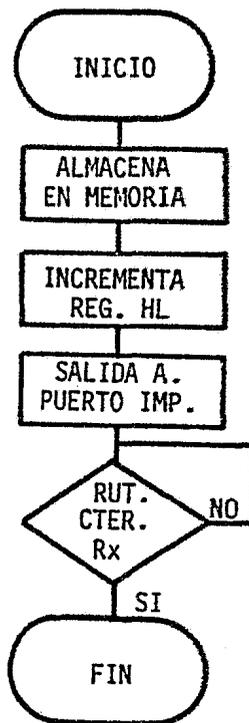


DIAGRAMA RUTINA C.R. A BLOQUES

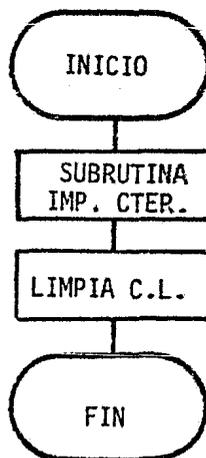


DIAGRAMA DE RUTINA DELETE A BLOQUES

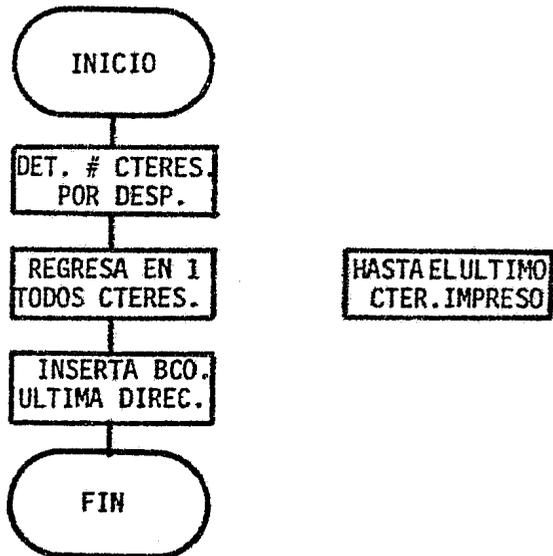


DIAGRAMA RUTINA EMPIEZO A BLOQUES



DIAGRAMA RUTINA REGRESO A BLOQUES

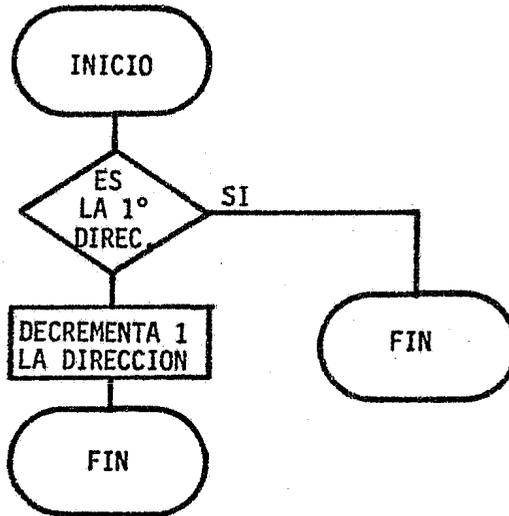


DIAGRAMA RUTINA AVANCE A BLOQUES

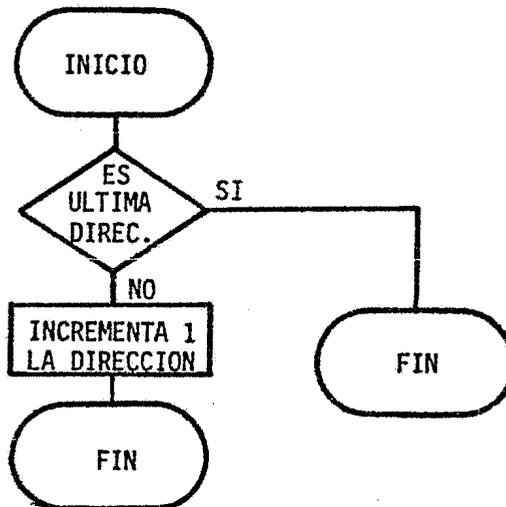


DIAGRAMA RUTINA FIN A BLOQUES

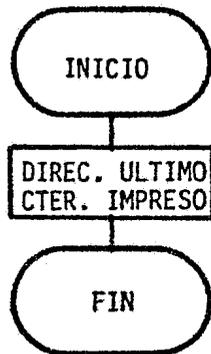


DIAGRAMA RUTINA RECEPCION A BLOQUES

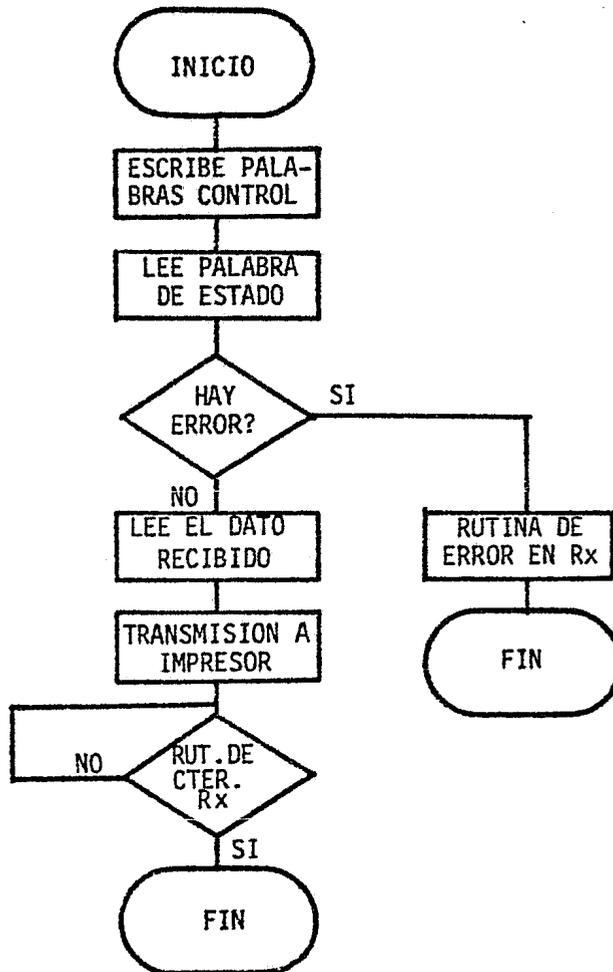


DIAGRAMA Rutina Impresion a Bloques

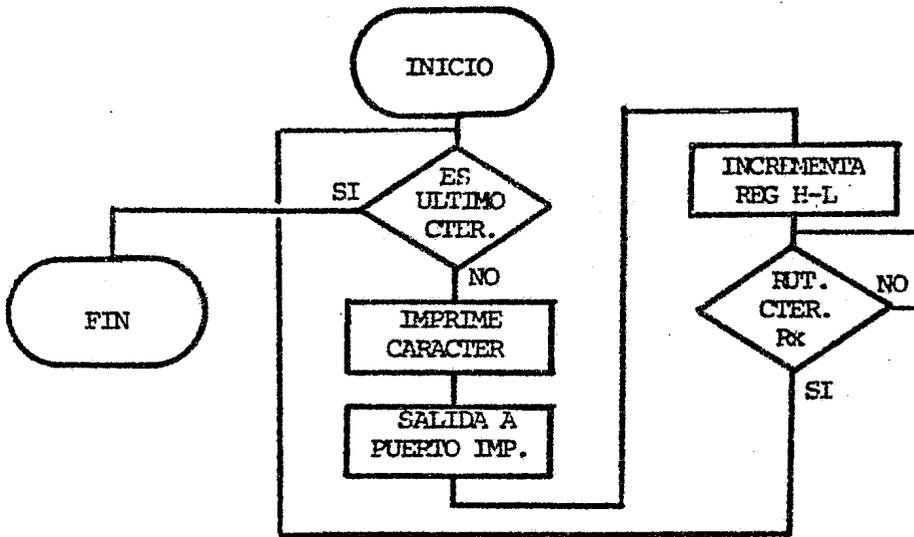
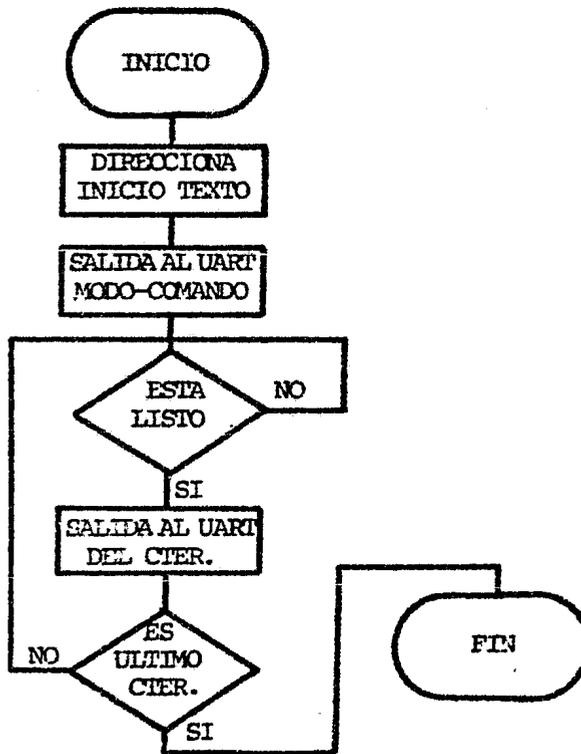


DIAGRAMA Rutina Transmision a Bloques



5.2.3. DESCRIPCION DEL SOFTWARE.

En esta parte se pretende hacer una explicación del programa Monitor, el cual se realizó en forma estructurada para mayor facilidad en diseño y comprensión.

El programa comienza con una rutina que hemos llamado de inicialización, en la cual se limpian los registros y memoria y se definen los siguientes parámetros: contador de caracteres, contador de columnas, inicio de texto y de Stack. Después de esto, el microprocesador permanece en un estado de espera para ser interrumpido por una señal de recepción o un comando de transmisión, ejecutando un RST 5 ó un RST 7, respectivamente. En el caso de RST 7 se procederá a la identificación del carácter teclado, teniendo dos opciones, la primera, que sea imprimible, en cuyo caso se realizará la rutina de imprime línea, y la segunda que sea de control, ejecutando la rutina correspondiente a cada código.

Dentro de la rutina de imprime línea, se contabiliza el número de caracteres impresos en una línea; que en caso de ser 48 se dará un CR automático sin almacenarse en memoria. Esta rutina contiene las subrutinas de imprime carácter y carácter recibido.

En la primera almacena el dato y lo imprime; en la segunda espera a que el dato haya sido impreso.

Cuando el caracter es de control, hace un análisis del mismo con el objeto de ejecutar la rutina correspondiente. Dentro de cada rutina se hacen los pasos necesarios para realizar las características propias de cada comando de control antes mencionadas.

En el caso de que se genere un RST 5 se interrumpe la 8080 (sólo en caso de estar en el loop de interrupción) y procede a ejecutar la rutina de recepción. En esta rutina se hacen los pasos necesarios para la programación de la 8251 así como la lectura de su palabra de estado, para tomar las decisiones necesarias y poder después mandar el dato a impresión. En caso de que en la lectura de la palabra de estado se detecte un error, mandará un mensaje de error ("ERROR EN RECEPCION").

En todo sistema, a cada periférico le corresponde determinado número de puerto, en este caso hemos usado 6 puertos, siendo éstos:

FUERTO**PERIFERICO**

00	Teclado
01	Switch Externo
02	Listo para Imprimir
05	Mecanismo de Impresión
06	USART Dato
07	USART Control

Los registros tienen una función específica dentro del sistema, siendo ésta:

REGISTRO**FUNCION**

A	Proceso
B	Contador de Columnas
D-E	Contador de Caracteres
H-L	Dirección en RAM
FFFF	Stack Pointer

C O N C L U S I O N E S

En el presente trabajo se sentaron las bases para el desarrollo de terminales portátiles, dejando la alternativa de mejorarla para su proyección en México.

Se ha introducido una nueva función dentro de las terminales portátiles, siendo ésta, la capacidad de editar dentro de la misma. Característica importante en el manejo de datos a distancia por la necesidad de contar con canales disponibles en cualquier momento (por medio del editor no se utiliza el canal de computadora para corrección de textos erróneos).

El hardware que se empleó para este proyecto tiene como principal objetivo, el poderse ampliar o cambiar en cuanto a funciones realizables. Se diseñó con la idea de que todos los componentes fueran de fácil adquisición y bajo costo.

En la actualidad han surgido un sin número de componentes de propósito específico, mediante los cuales los costos en diseño de cualquier sistema se han abatido. Un ejemplo de esto podría ser el microprocesador 8048, y en sí toda la familia, que en un solo

circuito integrado, están contenidos: un microprocesador, una memoria ROM y una memoria RAM. Sin embargo, el uso de este circuito limitaría la expansión en cuanto a Software. Existen también una gran cantidad de circuitos programables, como es el caso del 8255 (puerto de entrada/salida), los cuales amplían en cuanto a posibilidades de funciones, pero al mismo tiempo reducen en un software sofisticado, lo que trae como consecuencia un Programa Monitor más extenso y, por tanto, una memoria EPROM de mayor capacidad.

En cuanto al Software de la terminal, se deja la posibilidad de aumentar las características operativas, como también las del editor mismo, desarrollando, claro está, un programa más complejo.

El desarrollo de este trabajo es sólo el principio de lo que podría ser una terminal que ayudará a cualquier persona a introducirse al mundo de la computación.

Agradecemos las facilidades que nos fueron brindadas, al Departamento de Electrónica del IIMAS para la realización de este proyecto.

Apéndice A

M O D E M

La comunicación entre una terminal remota y una unidad de procesamiento de datos es muy frecuente, las líneas telefónicas proporcionan un medio económico bastante confiable para realizar dicha comunicación.

Existen algunas desventajas: un limitado ancho de banda del canal telefónico, la existencia de atenuación y la presencia de ruido, lo que nos obliga a acondicionar la información digital por medio de un MODEM (modulador-demodulador). El tipo de modulación puede variar (PSK, FSK, etc.), y en este caso nos referimos a un MODEM que opera conmutando frecuencias, modulación FSK, por medio del circuito MC 6860.

Este contiene circuitos para la modulación y demodulación digital, y un control supervisor que permite manejar los protocolos de funcionamiento en FULL DUPLEX en modo origen, en respuesta automática y desconexión automática.

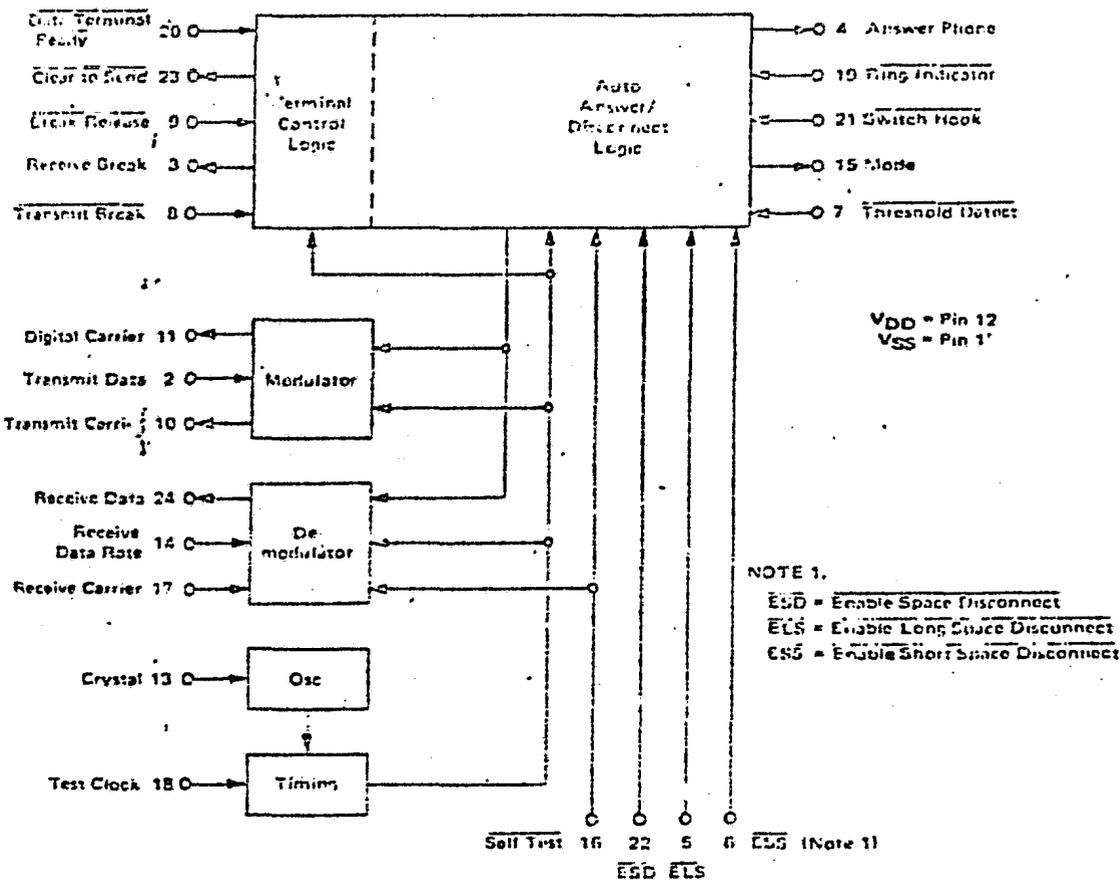


DIAGRAMA MC 6860

Señales de Entrada y Salida.

Recepción de Portadora (RxCAR). Es una entrada al demodulador. Cualquier semiperiodo mayor o igual a 429 1 microseg. en la banda inferior ó 235 1 microseg. en la banda superior, será detectado como espacio.

Indicador de Llamadas (RI). Reconocerá petición de llamada con por lo menos 20 ciclos de una señal de 20-47 Hz (nivel bajo = 50% del ciclo de trabajo), y dispone al MODEM en modo respuesta.

Solicitar Llamada (SH). Entrada para pedir llamada. Si en 17 seg., luego de hacerla no se estableció el enlace, se cuelga automáticamente.

Detector de Umbral (TD). Mediante esta señal se le informa al integrado que está presente la portadora. Debe estar baja por lo menos 20 microseg. cada 32 mseg. para tener operación normal. Si esto no sucede por más de 32 mseg. se invalidan los datos y a los 51 mseg., se pierde el CTS.

Velocidad de Recepción de Datos (RxRate). Es una entrada para optimizar la calidad del funcionamiento, de 0-300 bps. debe estar alta y de 0-600 bps. debe estar baja.

Datos Transmitidos (Txdata). Es una entrada al modulador, alta significa marca, baja es espacio.

Terminal de Datos Lista (DTR). Debe estar baja para operar, luego de 3 segs. de subir hay desconexión.

Liberar Solicitud de Interrupción (BrkR), Luego de recibir una petición de interrupción, una señal baja por lo menos 20 microseg., libera la bandera de interrupción.

Solicitar Interrupción (TxBrk). Una señal baja de más de 34 mseg. en esta entrada, hace que el MODEM envíe un espacio de 233 mseg., siendo interpretado como una solicitud al otro MODEM de que detenga su envío de datos.

Habilitación de Desconexión por Espacio ($\overline{\text{ESD}}$). Si está baja al pulsar $\overline{\text{DTR}}$ para iniciar desconexión, el MODEM transmite un espacio de 3 seg., o hasta recibir pérdida de umbral, lo que ocurra primero. Si está alta, se siguen transmitiendo datos hasta que se cumplan 3 seg. donde se produce la interrupción.

Habilitación de Desconexión por Espacio Corto ($\overline{\text{ESS}}$). Si está bajo se cuelga el teléfono automáticamente luego de 0.3 seg. de espacio continuo. $\overline{\text{ESS}}$ y $\overline{\text{ELS}}$ no deben ser bajas simultáneamente.

Habilitación de Desconexión por Espacio Largo ($\overline{\text{ELS}}$). Igual a la anterior, pero con un periodo de 1.5 seg.

Cristal (XTAL). Esta entrada es para un cristal de 1 MHz, también puede introducirse una onda cuadrada de 1 MHz. El cristal debe cumplir algunos requerimientos.

Prueba de Reloj (TST). Es una entrada para disminuir el tiempo de prueba del integrado. Normalmente debe estar bajo.

Auto Prueba (\overline{ST}). En caso de estar baja, nos permite recibir la misma transmisión del MODEM. Una vez alta sigue con la situación anterior.

Contesta Teléfono (AnPh). Esta salida es $((\overline{SP+RI})\overline{DTR})$, se utiliza para conectar al MODEM con la línea telefónica.

Modo (mode). Indica si el MODEM opera en modo respuesta (bajo) u origen (alto). Esta salida cambia de nivel al activar \overline{ST} .

Listo para Envío (\overline{CTS}). Un nivel bajo indica que está listo a transmitir.

Datos Recibidos (RxData). Es la salida del demodulador FSK. Un alto es una marca, un nivel bajo es un espacio.

Recibo de Solicitud de Interrupción (RxBrk). Luego de recibir un espacio de más de 150 mseg. se levanta esta bandera a un nivel alto. Sigue así hasta que se establece el \overline{CTS} .

Portador Digital (FC). Es una salida para disminuir el tiempo de prueba, es una onda cuadrada a la frecuencia de transmisión.

Portadora de Transmisión (TxCar). Es una senoide sintetizada a partir de escalones, sus características son:

MODO	DATO	FREC.	TOLERANCIA
Origen	Marca	1270 Hz	-0.15 Hz
Origen	Espacio	1070 Hz	0.09 Hz
Respuesta	Marca	2225 Hz	-0.31 Hz
Respuesta	Espacio	2025 Hz	-0.71 Hz

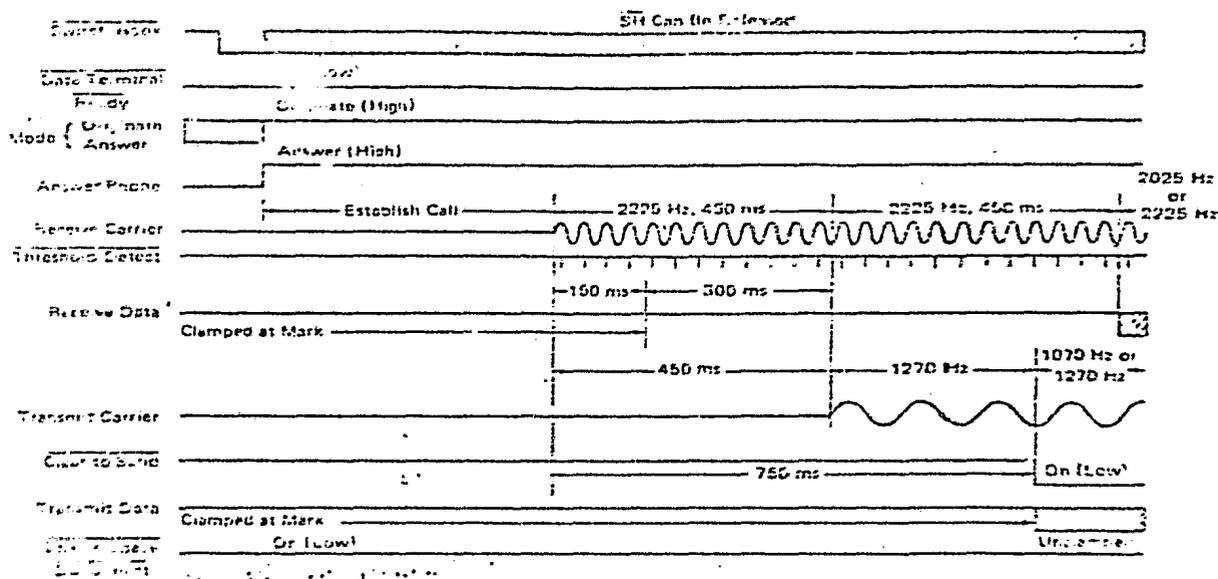


DIAGRAMA DE TIEMPOS MODO ORIGEN

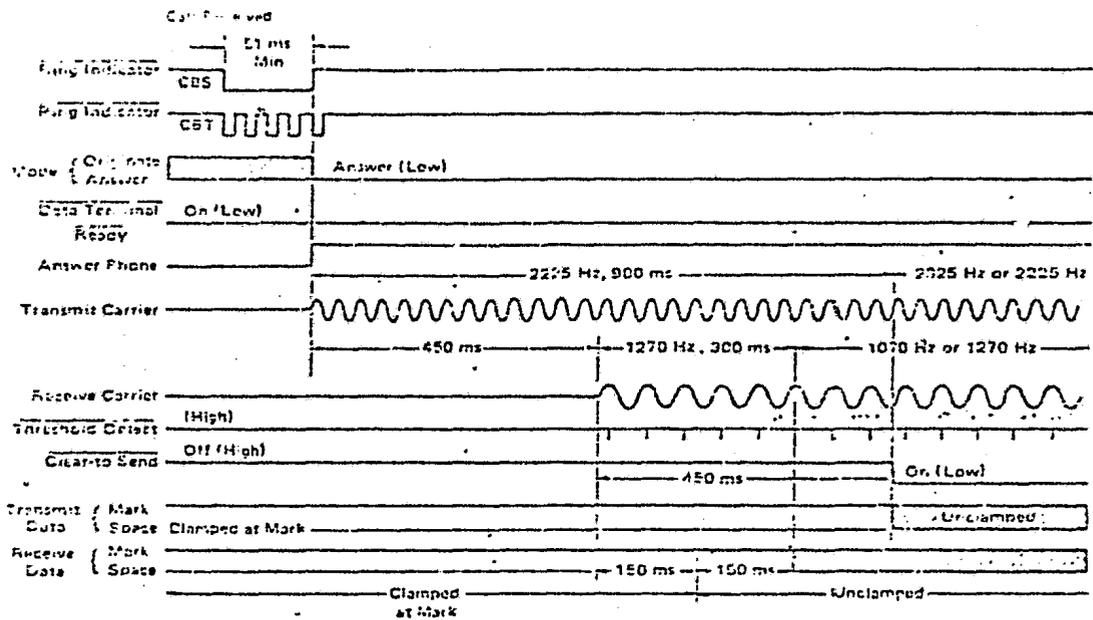


DIAGRAMA DE TIEMPOS MODO RESPUESTA

El C.I. MC 6860 requiere de ciertos elementos para mejorar su operación, como son:

1. Buffer y Filtros Pasabajos. La función de éstos es por un lado presentar alta impedancia en la salida del C.I., ya que por ser de tecnología MOS no puede manejar grandes cargas y, por otro lado, filtrar los componentes de alta frecuencia de la portadora modulada, pues ésta es sintetizada a partir de escalones.

2. Duplexer. Su función es la de presentar a la línea, la portadora transmitida y a los filtros de recepción, la portadora recibida, sin que la transmitida sea al mismo tiempo recibida.

3. Filtros de Recepción. Cada filtro selecciona un canal, separándolos.

4. Limitador, Detector de Umbral y Lógica de Selección. Las dos señales moduladas están presentes a la salida de los filtros. Para poder recuperar la información es preciso cuadrificarlas, para ser demoduladas por el MODEM. Para ello se las compara con un nivel de referencia nulo, a través de comparadores que admiten amplio rango de entrada en modo común. Además, se las compara con el nivel de referencia para saber si la intensidad de la señal recibida es aceptable.

Luego de hecho esto, tenemos dos parejas de señales a llevar al demodulador y a la lógica de control, una correspondiente al canal donde estamos transmitiendo y otra al canal de recepción; hay que seleccionar la pareja del canal de recepción, lo que se hace con lógica gobernada por la señal de modo de operación (Mode).

5. Acoplador. Este circuito enlaza la línea telefónica con el MODEM y el teléfono. A través de un relevador, seleccionamos la conexión de la línea telefónica a uno u otro. Este relevador está gobernado por el control del MC 6860 a través de la señal de Conteste Teléfono (AnPh).

Hay un circuito que detecta la señal de llamado enviándola al MODEM a través del indicador de llamada (RI).

Hay un switch para solicitar manualmente comunicación, Solicitud de Llamada (SR).

El acoplamiento del MODEM a la línea se hace a través de un transformador, pues el MODEM no es un circuito balanceado y además la central telefónica debe ver una baja impedancia en DC.

6. Interfaz. Las señales de recepción y transmisión de datos, así como las de control de terminal, son adaptadas para que puedan ser recibidas y transmitidas por el MODEM, con los niveles de voltaje y corriente adecuados. La interface usada es la RS232 con los circuitos 1488 y 1489.

Apéndice B

FUENTES DE ALIMENTACION

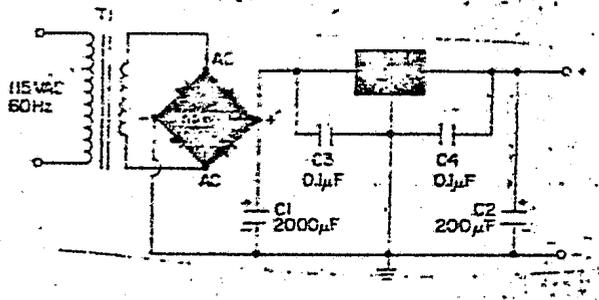
El propósito de este apéndice, no es el de dar una explicación detallada del diseño de una fuente, sino el de mostrar la fuente seleccionada para nuestro propósito.

Existe una gran variedad de arreglos, que van desde los más sencillos y baratos, hasta los más sofisticados, reeditando por supuesto en el costo.

Al escoger una fuente u otra, dependerá de la seguridad eléctrica y la eficacia que se desee.

Cada circuito integrado tiene un rango de voltaje para su correcto funcionamiento, por ejemplo, los C.I. TTL deberán funcionar entre los 5.25 y 4.8 volts, siendo ésta la primera condición para una fuente de alimentación. Si alguno de los elementos del sistema, cayera en un estado de corto permanente, o si alguno de los elementos de referencia se abriera, podrá producir un daño irreparable en los circuitos del sistema. Por tanto, la fuente deberá contar con cierta seguridad para evitar dicho daño.

Las fuentes a usar en este proyecto son tres: +5 Volts, +12 Volts y -5 Volts, que serán iguales en su configuración.



FUENTE REGULADA

Esta fuente es probablemente la más usada en sistemas chicos microcomputarizados. El transformador podrá ser cualquier transformador de filamento, con un rango de 2 amperes o más.

El rectificador es un puente de diodos a 50 volts PIV, cumpliendo con el rango de 1 ampere o más. El capacitor usado en el filtro cumple con la regla 2000 microF/amp., por supuesto no sufrirá daño alguno si se asegura con más capacitancia. El capacitor de salida es estrictamente opcional, siendo recomendable, ya que mejora la respuesta al transitorio. Este tendrá un valor que sigue la regla de 200 microF/amp., teniendo para nuestro caso un valor entre los 100 y 500 microF. Los pequeños capacitores son usados para mejorar la inmunidad al ruido de los reguladores. Estos deberán ser montados tan cerca como sea posible, físicamente. La mayoría de los fabricantes los localizan directamente sobre los mismos reguladores. Los reguladores podrán ser cualquiera de los estandares de tres terminales, LM 309, 7805/7812, LM340T5/LM340T12, etc.

BIBLIOGRAFIA

280 ASSEMBLY LANGUAGE PROGRAMMING.

Lance A. Leventhal

Osborne/Mc Graw-Hill, 1979.

MICROPROCESADORES 8080 E INTERFACES.

García Narcia Octavio F.

Preedición, 1980.

DIGITAL COMPUTER FUNDAMENTALS.

Thomas C. Bartee.

Mc. Graw-Hill, 1977.

INTRODUCTION TO MICROPROCESSOR SYSTEM DESIGN.

Harry Garland.

Mc Graw-Hill, 1979.

COMPUTER SYSTEM ARCHITECTURE.

M. Morris Mano.

Prentice-Hill, Inc., 1976.

TV TYPEWRITER COOKBOOK.

Don Lancaster.

Howard W. Sams & Co., Inc., 1976

PROGRAMACION 8080 Y 8085.

García Narcía Octavio F.

Bioediciones, S. A., 1980.

COMPUTERS AND PROGRAMMING.

Septiembre/Octubre 1981.

HYCOM, MODEL DC-4004.

Information Manual.

TELEGRAPHY.

Renton, R. N.

Pitman Publishing, 1976.

NEC MICROCOMPUTERS, INC.

1981 Catalog.

**TELEVIDEO MODEL 950 CRT TERMINAL INSTALLATION
AND USER'S GUIDE**

Televideo Systems, Inc.

SILENT 700 ELECTRONIC DATA TERMINALS.

Texas Instrument, Inc.

DATA COMMUNICATIONS.

Hewlett Packard, 1976

DISEÑO DE UN MODEM ASINCRONO, FULL-DUPLEX PARA 300 bps.

Ing. A. González Hermosillo.

UNAM, marzo de 1979.

COMPUTERS AND PROGRAMMING.

October 1981.