



**UNIVERSIDAD NACIONAL AUTONOMA
DE MEXICO**

FACULTAD DE INGENIERIA

**DISEÑO Y CONSTRUCCION DE UNA TARJETA
CONTROLADORA PARA UNA IMPRESORA
DE CARACTERES ESPECIALES**

T E S I S
QUE PARA OBTENER EL TITULO DE
INGENIERO EN COMPUTACION
P R E S E N T A :
JAVIER GAMEZ GARCIA

DIRECTOR DE TESIS: M.J. JESUS M. DOPADOR GONZALEZ



MEXICO, D. F.

1994

**TESIS CON
FALLA DE ORIGEN**



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

AGRADECIMIENTOS

A mis padres Javier Gámez V. y Bertha García de G. por haberme dado la vida, por guiar mis pasos y apoyarme en todo momento; porque gracias a ellos logré llegar a la meta y cultivar la más grande de las herencias.

A mi Universidad por haberme brindado la oportunidad de estudiar en sus aulas y que sus catedráticos me brindaran sus conocimientos.

A José A. Vargas y Jesús M. Dorador por su apoyo incondicional en el desarrollo del proyecto y por haberme aceptado en su equipo de trabajo.

A todas aquellas personas que colaboraron conmigo incondicionalmente en el seguimiento de este trabajo.

DEDICATORIAS

A mis padres con todo mi cariño amor y respeto.

A mis amigos Raúl, Juan Carlos y Cuauhtémoc.

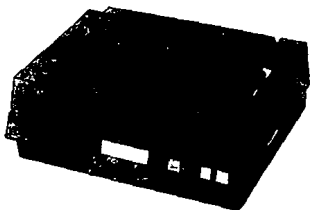
A mi novia con todo mi amor.

"Si tenemos la suerte de prestar algún servicio a nuestros compañeros en desgracia, nunca nos cansaremos de repetir que se debe al Sr. Barbier, el primero en inventar un sistema de escritura por medio de puntos para uso de los ciegos."

Luis Braille.

"Nunca podrán alabarse bastante los buenos sentimientos que le mueven a prestar servicio a aquellos que comparten su desgracia...pudiéndose esperar mucho de los elevados sentimientos que le guían."

Charles Barbier.



INDICE

CAPITULO 1

Introducción y Objetivos del trabajo

1.1 Introducción	1
1.2 Objetivos	3
1.2.2 Objetivos Específicos	3

CAPITULO 2

Generación de Diseño Conceptual

2.1 El sistema Braille	4
2.2 Conceptualización del sistema de impresión Braille	6
2.3 Conceptualización de los alcances este trabajo	12
2.4 Métodos de transferencia de datos de forma paralela	16
2.4.1 Simple Input And Output	16
2.4.2 Single Handshake Input Output	17
2.4.3 Double Handshake Data Transfer	18
2.5 Características de las hojas para la impresión en Braille	20

CAPITULO 3

Hardware del Sistema de Impresión Braille

3.1 Descripción de los sistemas de impresión	23
3.1.1 Impresoras de Impacto	23
3.1.2 Impresoras de matriz	25
3.2 Conceptualización de la impresora Braille prototipo	26
3.3 Diagrama de Bloques de la Arquitectura del Sistema de Impresión Braille	27
3.3.1 Buses del sistema	29
3.3.2 Unidad Central de Proceso	30
3.3.3 Circuitos de selección	30
3.3.4 Memorias	31
3.3.5 Sistema de Entrada/Salida	31
3.3.6 Interfaz Paralela Programable (PPI)	31
3.4 Descripción de la Arquitectura del Sistema de Impresión Braille	32
3.4.1 Buses del Sistema	33
3.4.2 Generador de reloj	35
3.4.3 Memoria ROM	35

3.4.4 Acceso a memoria de datos	36
3.4.5 Comunicación con la computadora	39
3.5 Direccionamiento de las agujas de impresión	44
3.6 Accionamiento de los actuadores	45
3.7 Sensores del sistema:	46
3.8 Direccionamiento a los circuitos del hardware	47

CAPITULO 4

Programación del Sistema de Impresión

4.1 Introducción	50
4.2 Conjunto básico de instrucciones	51
4.2.1 Instrucciones de transferencia	52
4.2.2 Instrucciones de operación	52
4.2.3 Instrucciones de control	53
4.3 Algoritmo de Inicialización del Sistema	54
4.4 Algoritmo de Recepción de Datos	56
4.5 Algoritmo de Traducción Braille - Ascii	63
4.5.1 Representación binaria del alfabeto Braille.	63
4.2.2. Almacenamiento del resultado de la conversión Ascii - Braille	68
4.6 Activación de las agujas de Impresión	71
4.7 Algoritmo de verificación de sensores	73
4.7.1 Sensor para chequeo de papel	73
4.7.2 Sensor de on-line y line-feed	74
4.8 Ensamblado de los programas	76

CAPITULO 5

Conclusiones	78
--------------	----

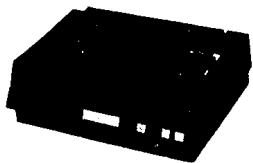
APENDICE A

Características del microcontrolador 8031 de Intel	81
--	----

APENDICE B

EL PPI 8255A	94
--------------	----

Bibliografía	98
--------------	----



CAPITULO 1

Introducción y Objetivos del trabajo

1.1 Introducción

El uso de las computadoras se ha extendido a todas las áreas de la actividad humana incluso en el área de edición de bibliografía. Para ello existen una innumerable cantidad de editores y procesadores de textos tales como Word Perfect, Chi-Writer, Word-Star o bien un editor como el Edlin de MSDOS que nos permiten imprimir en negro el documento utilizando infinidad de marcas, tipos de letras y modelos de impresoras.

Con el uso de las computadoras en este campo se han editado infinidad de libros en procesadores de textos, no obstante, la información que se ha publicado va dirigida principalmente a la gente que goza del sentido de la vista.

Los invidentes no pueden disfrutar de una buena lectura fácilmente puesto que las publicaciones en código Braille son muy reducidas y muy costosas.

El código Braille es un sistema desarrollado para que las personas invidentes puedan palpar relieves en una hoja a través de las yemas de sus dedos e interpretarlas como letras o números.

El tacto es realmente un factor de suma importancia para determinar la amplitud del uso del Braille. Los lectores se pueden clasificar en grupos de acuerdo a sus capacidades de lectura, van desde aquellos en quienes el Braille es algo así como una segunda naturaleza (como ocurre con quienes aprenden Braille desde temprana edad); hasta los que, por haber perdido la vista en edad adulta, tienen que pasar del

método de lectura visual al táctil. Estos últimos son invariablemente lectores lentos, siendo los de más edad los que mayores dificultades encuentran para dominar el Braille.

El código Braille está formado por un número limitado de caracteres, donde, cada carácter del alfabeto tiene un código especial formado por 6 puntos distribuidos en un arreglo de dos columnas y tres líneas, y para cada carácter corresponde una y solo una combinación de estos puntos; la combinación se da por la presencia o ausencia de cada punto, lo cual permite diferenciar entre una letra y otra.

El procedimiento de impresión Braille resulta costoso, encargándose casi totalmente de su ejecución por organizaciones privadas, a menudo con ayuda de subsidios del Estado. Los textos Braille se estereotipan sobre planchas de un metal blando, a mano o por medio de máquinas. Estas planchas se colocan sobre una prensa plana o rotativa que repuja los puntos en un papel especial para dicha impresión.

Por lo antes mencionado se pensó en desarrollar una impresora, que pueda convertir el código ASCII (American Standard Code Interchange Information) a código Braille, de tal forma que todos los textos que fueren editados en código ASCII sean factibles de ser impresos en relieve utilizando el código Braille y que el costo para imprimirlo sea menor.

1.2 Objetivos

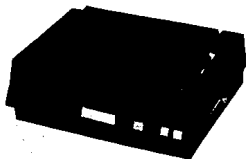
1.2.1 Objetivos Generales

- Diseñar y construir una impresora en código Braille con tecnología nacional.
- Promover la impresión de bibliografía en Código Braille.

1.2.2 Objetivos Especificos

- Desarrollar el hardware necesario para el control y operación de una impresora Braille prototipo.
- Desarrollar la microprogramación necesaria para la operación del hardware de la impresora Braille prototipo.

Nota: Cabe destacar que los alcances de este trabajo solo cumplen con los objetivos específicos, para cumplir con los objetivos generales, es necesaria la participación de áreas de la ingeniería mecánica y electrónica.



CAPITULO 2

Generación de Diseño Conceptual

2.1 El sistema Braille

El sistema Braille se compone de sesenta y tres signos, o sea sesenta y tres de las sesenta y cuatro combinaciones de puntos que componen en dominó de seis. Para facilitar la descripción de los signos por separado, se enumeraron los puntos convencionalmente en dos columnas de arriba hacia abajo como 1 2 3 para la columna del lado izquierdo y 4 5 6 para la columna del lado derecho.

En el Braille latino el alfabeto abarca veintiséis caracteres de los sesenta y tres, otros diez sirven internacionalmente como signos de puntuación y los veintisiete restantes se usan diversamente, para satisfacer las necesidades especiales de cada lengua o bien para las abreviaturas.

Los números se hallan representados por las diez primeras letras del alfabeto, con la diferencia de que son precedidas de un signo numeral, mismo que está incluido en los signos de puntuación.

En algunas lenguas existen dos grados de Braille:

- a) En el que se codifica letra por letra como se realiza en la escritura visual.
- b) En el que existen un número más o menos grande de signos abreviados para la expresión de conjuntos preposiciones, pronombres, prefijos, sufijos, etc.; su propósito fundamental consiste en reducir el volumen de los libros de Braille, lo cual supone una economía en los gastos de producción, así como de espacio para su almacenamiento y para los

gastos de distribución. Al mismo tiempo ahorra al braillista algunos esfuerzos en la lectura y la escritura.

Algunas otras lenguas han establecido sistemas muy abreviados que suelen considerarse como el Braille "grado tres", el cual, constituye una especie de taquigrafía, la cual se conoce como estenografía, ésta es demasiado complicada para los lectores que no reúnan las siguientes tres condiciones: amplio dominio del idioma, buena memoria y un sentido del tacto muy desarrollado; este sistema no es muy utilizado, debido a que las abreviaturas varían de idioma a idioma e inclusive, cada invidente puede definir su propio sistema estenográfico.

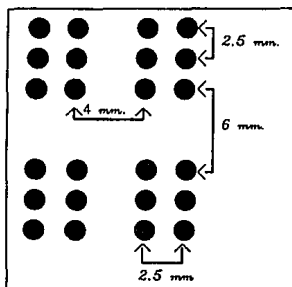
La razón por la cual el sistema de Luis Braille ha cobrado tanta superioridad sobre todas las otras formas de escritura para ciegos, estriba en que es el resultado de la pericia con una labor paciente y metódica.

Antes de que el sistema Braille se diera como tal, se habían realizado algunos otros intentos por dar a los ciegos una forma de comunicación escrita, sin embargo, estos no habían sido tan eficientes.

Para que una persona pueda diferenciar un caracter de otro relativamente fácil, se requiere, ordenar los puntos en grupos que corresponden exactamente a las necesidades del sentido del tacto. Si se reduce el número de puntos, los signos de que se dispone resultan insuficientes, en tanto, si se aumentan los signos, éstos no pueden abarcarse con la yema de los dedos, dificultando de esta manera su lectura.

Desde un principio, Braille asignó valores dobles o triples a cada signo, presentando un sistema de transcripción musical, una colección de símbolos matemáticos elementales y un sistema de taquigrafía.

Los seis puntos que componen cada caracter del sistema Braille se hallan distribuidos en dos columnas y tres líneas. En base a mediciones de diferentes sistemas de impresión, la separación más común entre cada punto es de 2.5 mm entre sí; la separación entre cada caracter es de 4 mm. y entre cada renglón es de 6 mm. el diámetro de cada punto es 1 mm. y la altura del relieve de 1 mm. como se muestra en la figura 2.1 .



Separación de los puntos que componen un caracter Braille
Figura 2.1

2.2 Conceptualización del sistema de impresión Braille

El problema que resuelve este trabajo, es fundamentalmente proporcionar un dispositivo que imprima la bibliografía que es editada por

computadora, en código Braille con costos reducidos. Por lo cual el sistema de impresión Braille prototipo se diseñó para que sea capaz de recibir datos de una computadora personal del tipo PC vía puerto paralelo de la misma, tal y como si ésta fuera una impresora estándar.

El sistema de impresión Braille está diseñado y programado para imprimir solamente caracteres alfabéticos, números y signos de puntuación, pensando en que en un futuro se puedan imprimir toda clase de caracteres especiales.

El diagrama de bloques correspondiente al prototipo se muestra en la figura 2.2

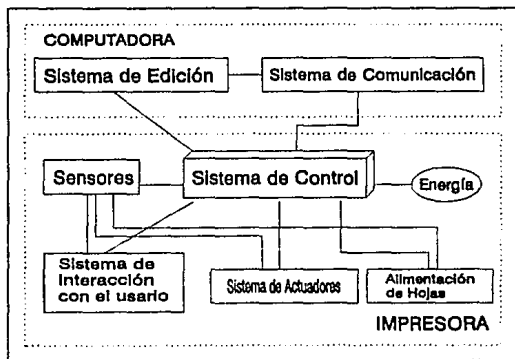


Diagrama de Bloques de la Impresora Braille
Fig. 2.2

a) Sistema de Control:

El sistema de control es el encargado de generar las señales necesarias para establecer: la comunicación del sistema de impresión con la computadora a través de una interface paralela programable (PPI), recibir datos del computador, traducir los datos recibidos, activar las agujas de impresión, checar el estado de los sensores y activar el sistema de alimentación de hojas para avance de papel o de renglón.

El sistema de control consta de un microprograma (programa en lenguaje de bajo nivel) almacenado en una memoria ROM que es el BIOS (Basic Input Output System) del sistema. Las palabras almacenadas en la memoria de control son microinstrucciones y cada una de ellas especifica una o más microoperaciones para los componentes en el sistema.

La conexión entre el sistema de control y los demás sistemas se da a través de un Bus de datos común a todos.

b) Sistema de Edición:

El sistema de edición consta de un procesador de texto especial para invidentes que tiene las funciones necesarias para impresión de documentos en Braille, inserción de palabras y funciones para almacenar, reemplazar o recuperar documentos. Para que un invidente pueda hacer uso del procesador de textos, está en etapa de diseño un cubreteclado que tenga los caracteres Braille en relieve.

c) Sistema de Comunicación:

Es el sistema a través del cual se establece el intercambio de señales de control entre el sistema de impresión y la computadora para el envío y recepción de datos en código ASCII a través del puerto paralelo de la misma pasándolos posteriormente al Bus de datos del sistema.

El sistema de comunicación a nivel Hardware utiliza una Interfaz Paralela Programable (PPI) para recibir los datos de la PC y enviarle las señales de control necesarias para establecer el protocolo que permite la transferencia de información paralela.

Para la mayoría de las impresoras como la IBM, EPSON FX-80 y la NEC 8023, los datos que han

de ser impresos son enviados en ocho líneas paralelas.¹

La transferencia del código Ascii hacia la impresora debe ser realizada a través de utilizar una técnica de transferencia de datos en los cuales intercambian información de control el sistema de impresión y la computadora, ya que ésta última puede enviar caracteres más rápidamente de lo que pueden ser impresos.

La impresora recibe los caracteres a ser impresos y los almacena en una memoria RAM (Random Access Memory) interna que sirve como buffer² de datos para que posteriormente sean impresos.

d) Sistema de Alimentación de hojas:

El avance de línea y de página se da a través de éste sistema, el cual consta de elementos mecánicos, eléctricos y electrónicos. Las señales de control que son necesarias para la activación de estos elementos son dadas por el sistema de control.

¹ V. Hall Douglas; Microprocessors and Interfacing Programming & Hardware; Edit. Mc. Graw Hill; 1986; Pág 271

² Un buffer de datos es una localidad de memoria en la cual se almacenan datos temporales.

e) Sistema de Energía:

Los elementos electrónicos y digitales del sistema son alimentados por el sistema de energía, el cual consta de una fuente de voltaje regulada.

f) Sistema de Actuadores:

El prototipo está diseñado mecánicamente para que la deformación del papel se realice a través de la activación de una serie de sesenta y cuatro bobinas solenoides que son activadas y desactivadas por el sistema de control.

Cada bobina podrá deslizar una aguja de impresión, misma que servirá para ejercer presión al papel a través de un rodillo que avanzará de manera bidireccional.

El sistema genera las señales necesarias para la activación del motores utilizados en: el avance de papel por línea de impresión, por renglón de texto y avance de hoja.

g) Sistema de Interacción con el usuario:

El sistema de Impresión Braille en su etapa de interacción con el usuario indica el momento en el cual ha ocurrido un error que requiere de la intervención directa del usuario para corregirlo. Para ello el sistema cuenta con una serie de

sonidos que le indican al usuario el momento en el cual se ha agotado la reserva de papel y si el sistema se halla en estado de on-line para continuar imprimiendo. En el momento que la impresora detecta que ha ocurrido un error el sistema de control se encarga de notificarlo a la computadora.

El usuario es capaz de poner a la impresora en estado de off-line (fuera de línea), en estado de on-line y darle a la impresora órdenes de avance de renglón o de papel.

h) Sistema de Sensores:

El sistema hace uso de sensores para saber el momento en el cual se ha terminado el papel, cuando el usuario desea detener la impresión y avanzar un renglón o una página completa.

Este trabajo comprende los sistemas de control, comunicación, interacción con el usuario, actuadores y sensores.

2.3 Conceptualización de los alcances este trabajo

El sistema de impresión Braille se subdividió en cuatro etapas como se muestra en la figura 2.3.

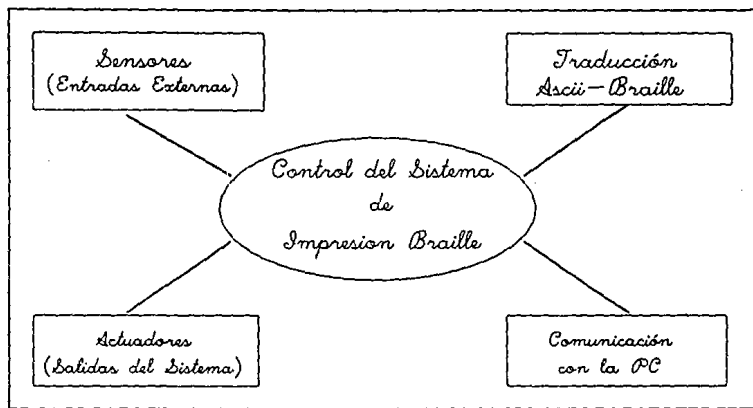


Diagrama de Bloques del Sistema de Impresión Braille
Fig. 2.3

a) Etapa de Comunicación con la PC

Aquí es donde se tiene el protocolo³ de comunicaciones entre la PC y nuestro sistema de impresión, en esta etapa se estudió la forma en que una PC transmite datos a través del puerto paralelo, que señales de control envía y cuales recibe y la manera en que las interpretan las impresoras estándar, la velocidad a que envía los datos y cuantos bits

³ Un protocolo de comunicación son las reglas y procedimientos utilizados para establecer una comunicación entre dos o más elementos. En los protocolos se definen los parámetros correspondientes al número de bits por palabra, velocidad de transmisión y número de bits de control.

por caracter utiliza. Por otro lado, la manera en que una impresora recibe datos y como los almacena en el buffer.

La mayoría de los dispositivos que utilizan puertos, como es el caso del circuito 8255A (Interfaz Paralela Programable) en la tarjeta SDK-86 contiene dos o tres puertos que pueden ser programados para operar en uno de los distintos modos que tienen, permitiendo así, utilizar el PPI para la transferencia de información paralela con alguno de los protocolos que existen para tal efecto.

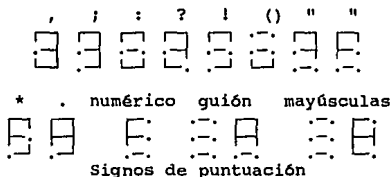
b) Traducción Ascii-Braille

En la etapa de conversión de Código los datos que son recibidos de la computadora que se hayan en el buffer de datos son interpretados por nuestro sistema para hacer una traducción de los caracteres ASCII a Braille.

La representación de cada caracter es un arreglo especial de tres líneas y dos columnas que corresponde exclusivamente a ese caracter como se muestra a continuación:

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q
⠁	⠃	⠉	⠑	⠅	⠋	⠗	⠎	⠚	⠞	⠊	⠍	⠏	⠕	⠇	⠖	⠙
r	s	t	u	v	x	y	z	ñ	á	é	í	ó	ú	w		
⠗	⠎	⠞	⠕	⠖	⠭	⠮	⠚	⠞	⠁	⠃	⠉	⠑	⠅	⠖		

Alfabeto Braille



donde: la línea continua denota la ausencia de deformación de papel.

c) Actuadores (Salidas del Sistema)

Cada una de las bobinas solenoides que mueven a las agujas de impresión tiene un valor posicional dentro de un byte. Las bobinas están divididas en conjuntos de ocho de tal forma que cada uno de los conjuntos tiene una dirección especial. Cuando se desea la activación de una de las bobinas, bastará con colocar un "1" lógico en la posición correspondiente dentro del byte y direccionarla.

La activación de los motores para el avance de línea y de papel, así como la del rodillo, corresponden a otra etapa del proyecto.

d) Sensores (Entradas Externas)

En esta etapa se detecta la existencia de papel a través de un fototransistor, si el usuario desea detener la impresión (sensor de on-line), el avance de línea (line-feed), el avance de forma (form-feed) o en el caso del rodillo bidireccional anteriormente mencionado, el momento en el es que ha llegado a uno de los extremos.

2.4 Métodos de transferencia de datos de forma paralela

2.4.1 Simple Input And Output

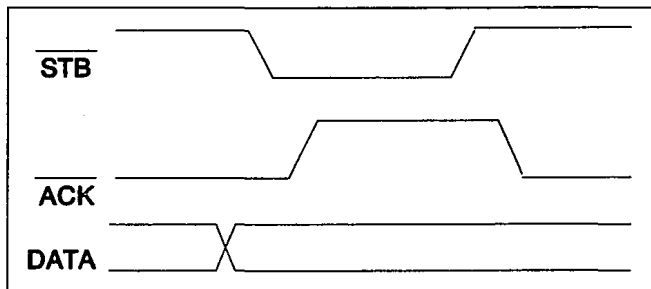
En muchas aplicaciones los datos que son enviados por un dispositivo son válidos solamente en un cierto momento y deben de ser leídos en ese tiempo. Un ejemplo claro de ello es el teclado de una computadora. Cuando una tecla es presionada, el circuito del teclado envía la información en ocho líneas paralelas de datos (Bus). El teclado envía entonces una señal de STROBE⁴ en otra línea para indicar que existe un dato válido en el bus de datos.

Para velocidades de transferencia de datos pequeñas, como de un teclado a un microprocesador, éste método funciona bien. Sin embargo, para velocidades de transferencia más rápidas no funciona porque no existe una señal que indique al dispositivo que envía la información cuando se está listo para recibir el siguiente dato.

⁴ La señal de STROBE es una señal de control utilizada en la transferencia de datos de forma paralela para indicarle al dispositivo receptor que hay un dato en las líneas de comunicación.

2.4.2 Single Handshake Input Output

En este método el periférico envía los datos y la señal de STROBE. El elemento que los recibe detecta la señal de STROBE a través de poleo o de interrupciones⁵ y entonces lee el byte⁶ del dato. El elemento receptor envía una señal de ACKNOWLEDGE (conocimiento) al periférico emisor para indicarle que puede enviar el siguiente byte de datos tal y como se muestra en la figura 2.4



Simple Strobe I/O

Fig. 2.4

⁵ Una interrupción se refiere a la transferencia de control de un programa que está trabajando corrientemente a otro programa de servicio como resultado de una señal de control generada externamente.

⁶ Un BYTE es el conjunto de bits que se requieren para representar un carácter en una computadora en particular.

Este diagrama puede representar el envío de datos de un microprocesador a una impresora. En este caso, el microprocesador envía los datos y la señal de STROBE a la impresora indicándole que hay un nuevo dato en el bus. Cuando la impresora se halla lista, ésta responde con una señal de ACKNOWLEDGE para indicarle al microprocesador que ya recibió el dato y que puede enviar el siguiente dato.

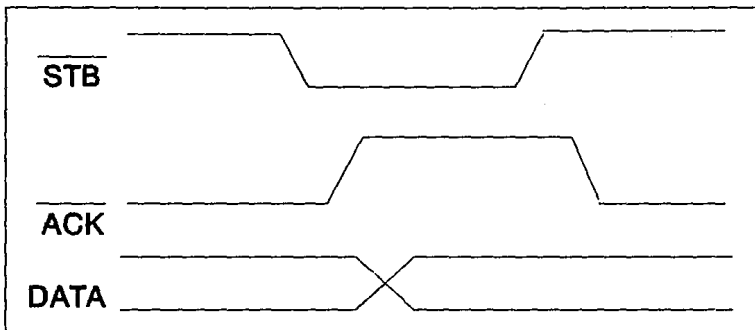
La característica principal de este método es que el dispositivo emisor no puede enviar el siguiente byte de datos hasta que el receptor le indica a través de la señal de ACKNOWLEDGE que está listo para recibir el siguiente byte de datos.

2.4.3 Double Handshake Data Transfer

Para la transferencia de datos en la que se requiere de una mayor coordinación entre el emisor y el receptor se utiliza el método *double handshake* la figura 2.5 la muestra gráficamente. Se puede hacer una analogía con una conversación entre dos personas. Cada una de las señales tiene un significado especial. El emisor manda la señal de STB (Strobe) en un nivel lógico "0" para preguntar "¿Estás listo?" el receptor contesta con la señal de ACK (Acknowledge) "Si, estoy listo", entonces, el emisor envía el byte de datos, el emisor coloca su señal de STB en un nivel lógico "1" para decirle al receptor "Aquí tienes un dato" , después de esto, el sistema receptor indica con la línea de ACK

en un nivel lógico "0" que ya ha recibido el dato y que espera recibir el siguiente dato.

Para utilizar este método de un microprocesador hacia un periférico, las señales son las mismas con la diferencia de que el microprocesador envía la señal de STB, y el periférico envía la señal de ACK.



Double Handshake I/O

Fig. 2.5

Un microprocesador puede determinar cuando es tiempo de enviar el siguiente byte de datos, basándose en técnicas de poleo o de interrupciones. Usualmente el método se basa en interrupciones ya que se maximiza el tiempo de procesador.

2.5 Características de las hojas para la impresión en Braille

El tamaño de papel que se utilizará para la impresión en código Braille tiene un tamaño de 28.5 cm. por 21.75 cm, utilizándose de dos maneras: que la longitud de la hoja sea de 21.75 cm y su amplitud de 28.5 cm. o al revés. Los márgenes izquierdo, derecho, superior e inferior son de 2.5 cm. cada uno.

Considerando que se utiliza la primer forma de impresión: S e tiene que cada caracter utiliza 6.5mm y que el margen derecho e izquierdo utilizan 50 mm. se tiene que el número máximo de caracteres que se puede imprimir por renglón es de:

$$285mm - 50mm = 235mm$$

$$\frac{235mm / renglon}{6.5mm / caracter} = 36 caracteres / renglon$$

y el número máximo de renglones es de:

$$217.5mm - 50mm = 166.5mm$$

$$\frac{166.5mm / hoja}{11mm / renglon} = 15 renglones / hoja$$

De lo anterior se obtiene que el número máximo de caracteres por hoja es de :

$$36 \text{ caracteres/renglon} \cdot 15 \text{ renglones/hoja} = 540 \text{ caracteres/hoja}$$

$$217.5 \text{ mm.} - 50 \text{ mm.} = 167.5 \text{ mm.}$$

Análogamente si se utiliza la segunda forma de impresión:

$$\frac{167.5 \text{ mm/renglon}}{6.5 \text{ mm./caracter}} = 25 \text{ caracteres}$$

y el número máximo de renglones es de:

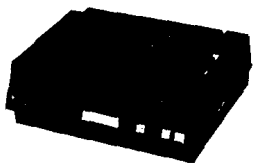
$$285 \text{ mm.} - 50 \text{ mm.} = 235 \text{ mm.}$$

$$\frac{235 \text{ mm./hoja}}{11 \text{ mm./renglon}} = 21 \text{ renglones/hoja}$$

de lo anterior se obtiene que el número máximo de caracteres que se pueden imprimir por hoja es de:

$$25 \text{ caracteres/renglon} \cdot 21 \text{ renglones/hoja} = 525 \text{ caracteres}$$

de tal forma que el uso de la hoja es óptimo si se utiliza de la primera forma.



CAPITULO 3

Hardware del Sistema de Impresión Braille

3.1 Descripción de los sistemas de impresión

Una impresora es capaz de recibir datos de una computadora, almacenarlos en un buffer, procesarlos e imprimirlos, asimismo, es capaz de detener la impresión en el momento en que el usuario lo desee, indicar cuando se requiere de papel para seguir imprimiendo o bien avanzar una página o una línea según se requiera enviando a la computadora señales de control que le indiquen en que momento ocurrió un error en el sistema.

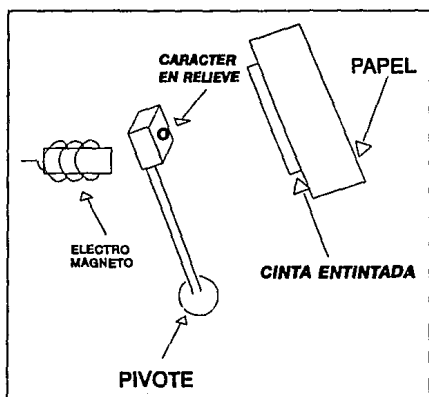
Existen diferentes tipos de impresoras, clasificadas según el principio de funcionamiento de las mismas, así por ejemplo, encontramos impresoras de matriz de puntos, impresoras de impacto o bien las impresoras láser, entre otras.

3.1.1 Impresoras de Impacto

Las impresoras de impacto son la clase más antigua de impresoras, éstas trabajan análogamente a una máquina de escribir: una pieza de metal o plástico que contiene al carácter en relieve golpea una cinta entintada contra una hoja de papel, como se muestra en la figura 3.1.

Algunos otros sistemas de impresión utilizan el principio de la margarita, llamada así porque contiene una rueda con pétalos como una margarita con los caracteres en relieve en los pétalos. Para imprimir un carácter, la impresora gira la rueda colocando el carácter apropiado enfrente del electroimán y acto seguido, energiza al magneto presionando la letra contra la cinta.

Existe otro tipo de impresoras de impacto, como las impresoras de martillo, éstas últimas utilizan una cadena de acero, en ésta, se hallan grabadas las letras. Una impresora de 80 columnas tendrá 80 martillos, uno para cada columna; una línea se imprime activando a cada martillo para que golpee tan pronto como la letra apropiada esté frente a él.



Corte transversal de una impresora de impacto.

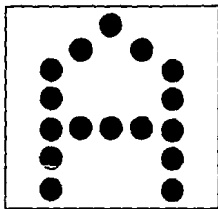
Fig. 3.1

3.1.2 Impresoras de matriz

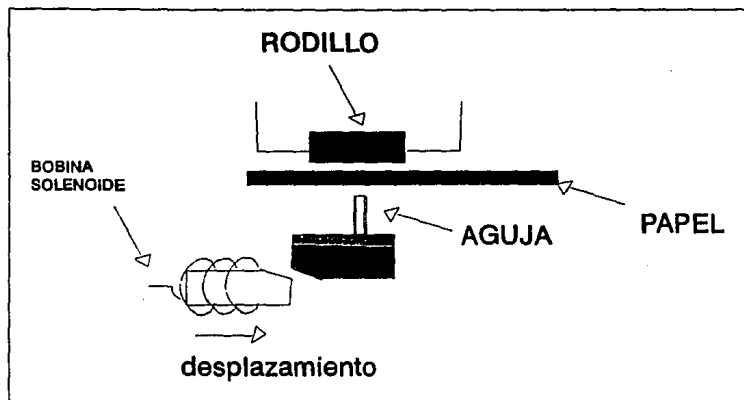
En las impresoras de matriz, se tiene una cabeza de impresión que se desplaza a lo largo de cada línea de impresión, la cabeza de impresión puede tener entre 7 y 24 agujas activadas de manera electromagnética.

Una impresora barata puede tener 7 agujas para imprimir líneas de 80 caracteres en un formato de 5 x 7, por lo tanto, una línea de impresión de estas características consiste de siete líneas horizontales, cada una consistente de 5 x 80 = 400 puntos. Cada punto se puede o no imprimir dependiendo de los caracteres que contenga la línea. La figura 3.2 muestra la letra 'A' impresa en una matriz de 5x7.

La calidad de impresión puede mejorarse por medio del uso de más agujas, o bien, sobreponiendo los círculos; sin embargo, el incremento en la calidad va al parejo con la velocidad de impresión del sistema, ya que normalmente se necesitan múltiples pasadas sobre cada línea para producir puntos sobrepuestos.



La letra "A" en una matriz de 5x7
Fig. 3.2



Principio de funcionamiento de la impresora Braille
La bobina solenoide al ser activada desplaza a la base de la aguja levantándola.

Fig. 3.3

3.2 Conceptualización de la impresora Braille prototipo

La impresora Braille es una combinación de los principios de funcionamiento de las impresoras de matriz de puntos y de impacto. Este sistema activa una serie de 64 solenoides, y a su vez, cada bobina solenoide levanta a una aguja metálica. Una vez que todas las agujas correspondientes a los caracteres de la línea a imprimir han sido activadas, un rodillo se desplaza de un extremo de la impresora hacia el otro, ejerciendo presión sobre el papel y las agujas, de tal forma,

que se deja marcado en relieve un punto, tal y como se muestra en la figura 3.3.

Cuando el rodillo ha llegado al otro extremo, el sistema realiza un avance de papel de 2.5 mm. para imprimir la segunda línea⁷ del renglón repitiendo el proceso anteriormente descrito también para la tercer línea del renglón.

Una vez que el sistema ha terminado de imprimir las tres líneas del renglón se realiza un avance de papel de 6mm. para imprimir otro renglón y se continua con el proceso sucesivamente (véase la figura 2.1).

3.3 Diagrama de Bloques de la Arquitectura del Sistema de Impresión Braille

El diagrama de bloques del sistema de impresión Braille se muestra en la figura 3.4, el sistema cuenta con un bus de datos, un bus de direcciones y un bus de control para la comunicación de los elementos constituyentes del sistema.

Como se puede notar en el diagrama de la figura 3.4, el bus de direcciones está conectado a la memoria ROM (Read Only Memory), la memoria RAM (Random Access Memory), el sistema de sensores, PPI (Parallel Programmable Interface), los circuitos de selección y el sistema de Actuadores utilizándose para seleccionar al subsistema con

⁷ El término renglón hace referencia al conjunto de las tres líneas de puntos que forman a un carácter Braille.

el cual se realiza la transferencia de información a través del bus de datos.

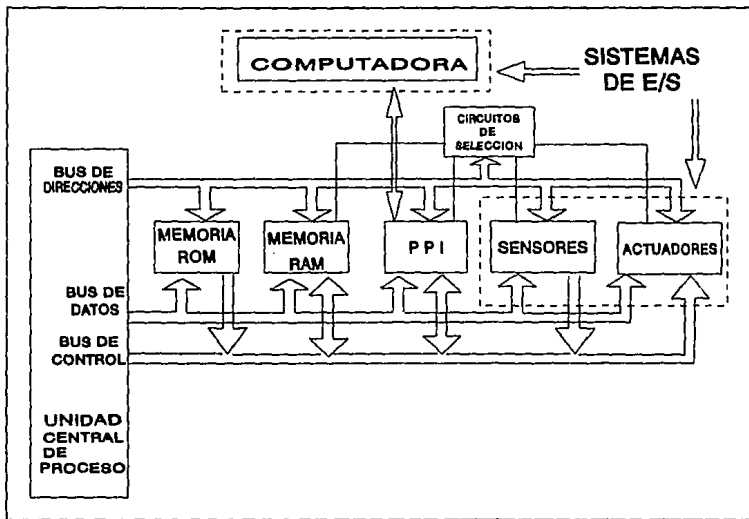


Diagrama de bloques del Sistema de Impresión Braille
Fig. 3.4

3.3.1 Buses del sistema

"Un bus es una ruta eléctrica común entre múltiples dispositivos"⁸. La información desde o hacia la unidad central de proceso puede ser de tres tipos: datos, direcciones y control, aunque no estén físicamente siempre separados, ya que están multiplexados⁹ en el tiempo, como explicaremos más adelante.

a) *Bus de direcciones*: son líneas del microcontrolador (unidad central de proceso) a las memorias o los demás sistemas que sirven para enviar la dirección del elemento seleccionado para una transferencia de datos.

b) *Bus de datos*: son líneas bidireccionales (o dos buses unidireccionales independientes) para el envío de información (instrucciones y datos) entre los subsistemas.

c) *Bus de control*: son líneas de entrada o salida de la unidad central de proceso de órdenes o de información de estado que permiten la operación de todo el sistema. El bus de control es utilizado para dar las señales de sincronización que gobiernan el intercambio de información entre los distintos subsistemas. Estas señales son las de validación de lectura o escritura de memoria y Entrada/Salida.

⁸ Andrew S. Tanenbaum, *Organización de Computadoras un enfoque estructurado*, pág. 128

⁹ Multiplexar significa transmitir una gran cantidad de unidades de información por un número pequeño de canales o líneas. Un multiplexor digital es un circuito combinacional que selecciona información binaria de una de muchas líneas de entrada para dirigirla a una sola línea de salida.

3.3.2 Unidad Central de Proceso

La unidad central de proceso es el sistema central del sistema de impresión Braille, ya que es capaz de interpretar las instrucciones y coordinar la transferencia de la información con otros subsistemas. La unidad central de proceso está constituida por tres subsistemas funcionales:

a) *Unidad Aritmética y Lógica*: es el elemento en donde se llevan a cabo las operaciones aritméticas y lógicas.

b) *Unidad de Control*: es el elemento que controla el flujo de la información (instrucciones y datos). Las funciones principales de la unidad de control son las siguientes:

b.1) búsqueda de instrucciones en memoria;

b.2) decodificación, interpretación y ejecución de las instrucciones;

b.3) control de la secuencia de ejecución

b.4) reconocimiento de primitivas externas de control: interrupciones, petición de acceso directo a memoria, órdenes de paro, espera, inicialización, etc.

c) *Generador de reloj*: es una fuente de pulsos de reloj común, por lo general un oscilador, el cual genera un tren periódico de pulsos eléctricos.

3.3.3 Circuitos de selección

Un circuito de selección permite al sistema a través del bus de direcciones seleccionar al subsistema que debe de utilizar el bus de

datos para transmitir o enviar información habilitando a uno y sólo un subsistema.

3.3.4 Memorias

Las memorias son utilizadas para almacenar las instrucciones que forman el programa y almacenar los datos y resultados intermedios. Las memorias están organizadas en palabras, cada una con una única dirección, la longitud de la palabra, en general, es la misma que la del bus de datos.

Las memorias de sólo lectura (ROM) se utilizan para almacenar programas, tablas y constantes. Las memoria de escritura y lectura (RAM) tienen como finalidad el almacenamiento de datos y resultados intermedios, siendo éstas del tipo volátil, es decir, la información almacenada se pierde si falla la alimentación del sistema.

3.3.5 Sistema de Entrada/Salida

El sistema de entrada y salida permite al sistema de impresión la comunicación con el mundo exterior utilizando para ello los subsistemas de sensores y actuadores, el sistema de E/S permite también la comunicación con la computadora a través del PPI (Interfaz Paralela Programable).

3.3.6 Interfaz Paralela Programable (PPI)

Una interfaz Paralela Programable es un circuito LSI (Large Scale Integration) compatible con el resto de los elementos del sistema a fin

de potenciar la comunicación con la computadora. Las características fundamentales de este subsistema son:

- a) Facilidad de interconexión a los buses de comunicación: el PPI se halla conectado directamente a la computadora para establecer la comunicación entre el sistema de impresión y la misma. El PPI es el encargado de recibir los datos provenientes de la computadora y pasarlos al bus de datos del sistema, estableciendo el protocolo de comunicaciones para el intercambio de datos y señales de control (véase la sección 2.4).
- b) Posibilidad de Programación: la interfaz cuenta con varios puertos que pueden ser programados como puertos de entrada o de salida; la programación consiste de cargar de una forma determinada un registro interno de modo de operación.

3.4 Descripción de la Arquitectura del Sistema de Impresión Braille

La arquitectura del sistema de impresión Braille se halla basada en el uso del microcontrolador 8031 de Intel (circuito u11) que es la unidad central de proceso del sistema en la que se llevan a cabo las operaciones aritméticas y lógicas y se generan las señales de control necesarias para habilitar a cada uno de los circuitos de la tarjeta controladora.

Se seleccionó el uso de este microcontrolador debido a que en un sólo circuito se incluye una CPU, RAM, 24 líneas de entrada/salida paralelas y una UART interna, permite la programación de hasta 4KB de ROM y el acceso a 128 Bytes de memoria interna, para este proyecto no se requieren de convertidores analógicos digitales (que ya vienen

integrados en algunos microcontroladores como el HC11) ni tampoco el direccionamiento de mucha memoria, existe en el mercado nacional por lo que no es necesario pedirlo al extranjero y además proporciona un set de instrucciones que permiten operaciones aritméticas, lógicas, de bifurcación y de transferencia de datos; con todas estas características mencionadas, el microcontrolador 8031 es el más económico y satisface las necesidades del proyecto.

3.4.1 Buses del Sistema

El microcontrolador 8031 tiene una capacidad de direccionamiento de hasta 64 KB con un bus de direcciones de 16 bits distribuido en dos puertos (Puerto 0 y Puerto 2). El bus de direcciones corresponde a las líneas etiquetadas como ADD0-ADD15.

El bus de datos de E/S es de 8 bits y se halla en el puerto 0 multiplexado con el bus de direcciones bajas (líneas D0-D7).

El microcontrolador 8031 genera señales de lectura (~RD), escritura (~WR) y de habilitación de circuitos latch (ALE- Ale Latch Enable) que forman parte del bus de control para hacer uso de memorias externas de programas y de datos.

Como el puerto 0 es utilizado para el bus de datos y direcciones del sistema se hace necesario realizar un multiplexaje en el tiempo de las señales para la separación de los datos y las direcciones, para facilitar esta tarea, el microcontrolador genera la señal de control ALE que está conectada directamente al pin de enable (pin de habilitación del circuito) del latch (circuito u1) mientras esta señal se encuentra en alto el latch permite el paso de las señales que se

encuentren en las entradas D_0 - D_7 , hacia las salidas Q_0 - Q_7 , las cuales forman la parte baja del bus de direcciones. Cuando la señal de ALE está en un nivel lógico bajo, las líneas son utilizadas por el bus de datos manteniendo las direcciones bajas hasta que la señal de ALE vuelva al estado alto para obtener otra dirección.

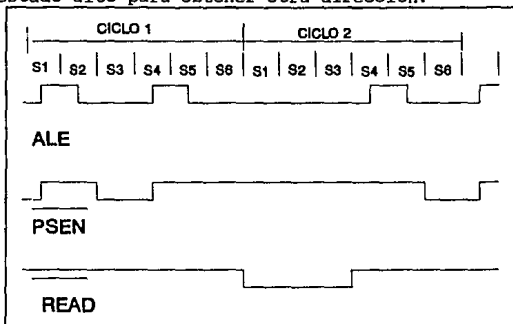


Diagrama de tiempo de transferencia de datos

Fig. 3.5

Como se muestra en la figura 3.5 cuando se habilita la señal de lectura, la señal de ALE y -PSEN (Program Strobe Enable) se deshabilitan, permitiendo a los datos el uso del bus de datos/direcciones, una vez que se ha realizado la transferencia de los datos, las señales de ALE y -PSEN son habilitadas nuevamente.

La señal de RESET (inicialización del sistema) la proporciona el usuario a través de un interruptor del tipo push-button conectado a una compuerta NOT (circuito u19A), en el momento en el que se proporciona la señal de RESET al sistema todos los puertos del microcontrolador

tienen en sus pines niveles lógicos altos, al igual que las señales de -RD y -WR. La señal de RESET se halla conectada también al PPI (circuito u14), ocasionando que los puertos de éste se utilicen como entradas para evitar que el circuito se dañe.

3.4.2 Generador de reloj

La tarjeta controladora del sistema de impresión Braille opera a una velocidad de 6.6 MHz. utilizando un cristal oscilador de 8 Mhz. conectado entre los pines 18 y 19 del microcontrolador como se puede observar en el diagrama de conexión del circuito (no se anexa el diagrama de conexión del circuito debido a que se trata de un proyecto en proceso de patente). Las señales de reloj que proporciona el sistema servirá para sincronizar a los elementos del hardware que hacen uso del bus de datos a través del bus de control del sistema.

3.4.3 Memoria ROM

El microcontrolador 8031 de Intel no cuenta con memoria de programa interna de tal forma que se utiliza una memoria ROM 2764 (circuito u12) en la que se halla almacenado el BIOS del sistema. El BIOS es el programa que el microcontrolador ejecuta empezando desde la dirección 0000H cuando se proporciona el RESET al sistema. La memoria ROM proporciona a través del bus de datos del sistema las instrucciones del programa al microcontrolador para que éste las interprete y las ejecute.

El programa del sistema de impresión Braille está escrito en el lenguaje ensamblador para el 8031. Un programa en lenguaje ensamblador

es una representación simbólica de algún lenguaje máquina subyacente. Un lenguaje ensamblador puro es aquel en el que cada sentencia produce exactamente una instrucción máquina, existiendo una correspondencia biunívoca entre las instrucciones de máquina y las sentencias del programa. Lo traduce a lenguaje máquina un programa denominado ensamblador. Los programas escritos en lenguaje máquina pueden ejecutarse directamente en los circuitos electrónicos.

3.4.4 Acceso a memoria de datos

El microcontrolador tiene integrada una memoria interna de datos de 256 Bytes la cual es insuficiente para nuestro sistema debido a que se requiere de una capacidad mayor para almacenar los datos que se reciben de la computadora para posteriormente procesarlos, por lo que se utiliza una memoria RAM externa de 8kx8 (circuito u13)¹⁰ que sirve como buffer de datos, de tal forma que los datos que envíe la computadora son guardados en la memoria RAM para posteriormente procesarlos y traducirlos a Braille. El rango de direcciones con los cuales se accesa a la memoria RAM es de la dirección 0000₁₆ hasta la dirección 1FFF₁₆, utilizando la señal de lectura del microcontrolador para habilitar el pin de salidas (output enable) y la de escritura para almacenar los datos en la RAM.

El 8031 cuenta con un registro interno de direcciones de 16 bits conocido como DPTR que se utiliza para direccionar los 64 KB de memoria

¹⁰El primer número indica la cantidad de palabras que tiene la memoria; el segundo número indica el número de bits de la palabra de la memoria.

de datos y de programa. Cuando se desea acceder a un elemento externo, el registro DPL (parte baja del registro DPTR) se hallará en el puerto 0 y el DPH (parte alta del registro DPTR) se encontrará en el puerto 2. Todos los elementos del hardware como son los actuadores y los sensores son accedidos de la misma manera que se accesa a la memoria de datos.

En el set de instrucciones del 8031 permite hacer transferencias de datos entre el microcontrolador y los demás elementos del hardware utilizando la instrucción MOVX y el registro DPTR. Cuando se ejecuta una transferencia de este tipo, son habilitadas en bajo la señal de lectura o de escritura en el caso de hacer una transferencia de una dirección hacia el 8031 o viceversa. El programa mostrado en la figura 3.6 escribe 255 datos en la memoria RAM externa y posteriormente los lee.

MOV DPTR, #DATO: mueve hacia el registro de direcciones el valor de DATO

MOVX @DPTR, A : transfiere la información contenida en el acumulador hacia la localidad direccionada por el registro DPTR.

MOVX A, @DPTR : transfiere el dato que se halla en la dirección que tiene el DPTR hacia el acumulador.

La señal de -RD (lectura) y -WR (escritura) son generadas por el propio microcontrolador cuando se realiza una transferencia de información de un circuito hacia el microcontrolador o del microcontrolador hacia un circuito respectivamente. Todos los elementos del hardware que no tienen en su diseño un pin para las señales de lectura o escritura


```

        ORG      0000H
        SJMP    0040H
        ORG      0040H
INICIO:  MOV     DPTR, #0000H
        MOV     A,   #00H
ESCRIBE: MOVX   @DPTR, A
        INC     A
        INC     DPTR
        CJNE   A,   #0FFH, ESCRIBE
LEE:     DEC     DPTR
        MOVX   A,   @DPTR
        CJNE   A,   #00H, LEE
        SJMP   INICIO

        END

```

Programa que transfiere datos entre el 8031 y la memoria RAM
Fig. 3.6

y que utilizan el bus de datos del sistema usan estas señales de control combinadas con señales lógicas generadas por el bus de direcciones a través de decodificadores¹¹ (circuitos u16 y u17) con compuertas lógicas OR o NOR para sincronizar la transferencia de la información entre los elementos del sistema; de tal forma que para enviar o recibir datos de cualquier elementos del sistema se utilizará el contenido del registro DPTR del microcontrolador.

La sincronía entre los elementos del hardware para la transferencia de la información a través del bus de datos es necesaria puesto que el

¹¹ Un decodificador es un circuito combinatorial que tiene n líneas de entrada y 2^n de salida, numeradas desde 0 hasta $2^n - 1$. Si el número que hay en las líneas de entrada es k , entonces se verifica en alto o bajo (dependiendo del tipo de circuito) la salida k , permaneciendo las demás sin verificarse.

puerto 0 del microcontrolador es utilizado como bus de datos y direcciones, en secciones posteriores se explica como se lleva a cabo.

3.4.5 Comunicación con la computadora

El circuito 8255A (circuito u14) es un PPI (Interface Paralela Programable) a través del cual el sistema establece la comunicación con la computadora vía puerto paralelo y recibe los caracteres del texto que se desea imprimir. Asimismo, a través de él se envían y reciben las señales de control que requiere la computadora utilizando el protocolo de comunicaciones estándar de las impresoras.

El 8255A tiene tres puertos de E/S interconectados entre sí por un bus interno; éstos son direccionados a través de las líneas A_0 y A_1 del 8255A, cada uno de los puertos es de 8 bits siendo posible su programación en tres modos diferentes accedando la palabra de control del PPI.

Los puertos del PPI están conectados a un conector Centronics, utilizándose en el sistema de impresión Braille: el puerto A para la recepción de los caracteres a imprimir, el puerto B para el envío de señales de control hacia la computadora y el puerto C para la recepción de señales de control de la computadora. Los datos se envían a través del conector Centronics, utilizándose los pines 2 al 9 para los datos, la señal de -STROBE en el pin 1 y la de inicialización (-INIT) en el pin 31, las señales de -ERROR, PE, SLCT, BUSY y -ACK a través de los pines 32, 12, 13, 11 y 10 respectivamente (véase la tabla 3.1).

Para acceder al PPI se requiere que las líneas de direcciones A_{13} - A_{14} tengan el valor de 001 respectivamente, esto habilitará la señal Y_1 del

decodificador en un nivel lógico bajo (circuito u16) y ésta a su vez al -CS (circuito seleccionado) del PPI.

El puerto A del 8255A (dirección 2000) es el utilizado para recibir los datos directamente de la computadora a través del puerto paralelo de la misma y pasarlos al bus de datos del sistema (véase el diagrama de conexión del circuito).

La dirección 2001H hace referencia al puerto B del PPI y a través de él se envían las siguientes señales de control:

- a) -Error : verificándose en bajo cuando en el sistema ha ocurrido algún error.
- b) PE : se verifica en alto e indica que la impresora no tiene papel.
- c) SLCT : se verifica en alto cuando la impresora está seleccionada.
- d) Busy : esta señal indica que la impresora no puede recibir datos. Esta señal se verifica en los siguientes casos:
 - 1. Durante la llegada de un dato.
 - 2. Durante la operación de impresión.
 - 3. En el estado de off-line.
 - 4. Durante un error en el estado de la impresora.
- e) -Ack : esta señal tiene una duración aproximada de 5 us e indica que el carácter se ha recibido y se está listo para recibir el siguiente.

En la figura 3.7 se muestra el diagrama de tiempos utilizado para la recepción de los datos y el envío de las señales de control. La computadora verifica que la señal de busy de la impresora esté en bajo (no busy) para enviarle un dato, posteriormente, la señal de -strobe se verifica en bajo indicándole a la impresora de que hay nuevos datos en el puerto. La impresora lee los datos y coloca en alto la señal de busy para indicarle a la computadora de que no envíe más datos hasta que se

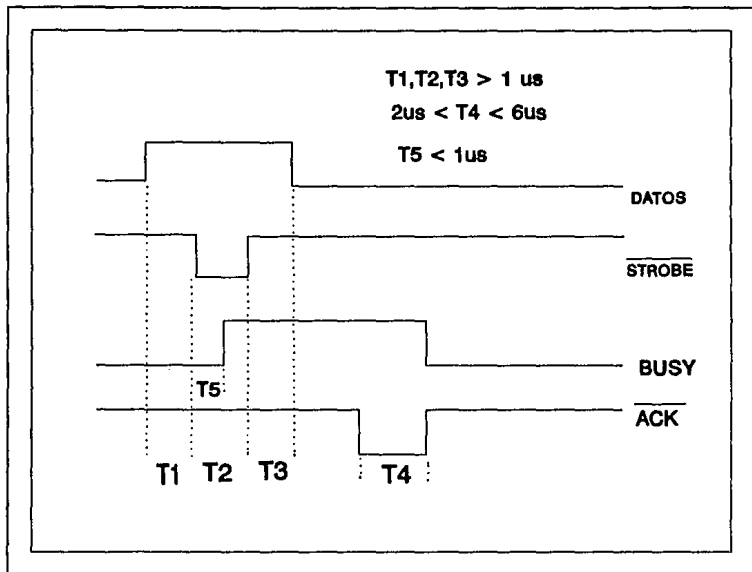


Diagrama de Tiempos de Recepción de Datos
Fig. 3.4

le indique a través de la señal de `-acknowledge` indicándole así que la impresora está lista para recibir un nuevo dato.

La señal de `-strobe` tiene una duración mayor a 1 microsegundo, en tanto que la señal de `-acknowledge` debe de tener una duración entre 2 y 6 microsegundos.

La dirección 2002H accesa al puerto C del PPI el cual se utiliza para recibir las señales de control de -strobe e -init provenientes de la computadora.

Cuando se accesa a la dirección 2003H se direcciona la palabra de control del PPI, a través de ésta podemos programar al 8255A para utilizarlos como puertos de entrada o salida.

Al igual que en la figura 3.6, para leer los datos que se reciben de la computadora y para enviar las señales de control se utiliza el registro DPTR a través del set de instrucciones de transferencia de datos del microcontrolador.

PIN	SEÑAL	DIRECCION	DESCRIPCION
1	STB'	ENTRADA	Pulso en bajo que indica que un dato esta listo para ser leído.
2	DATA1	ENTRADA	Dato
3	DATA2	ENTRADA	Dato
4	DATA3	ENTRADA	Dato
5	DATA4	ENTRADA	Dato
6	DATA5	ENTRADA	Dato
7	DATA6	ENTRADA	Dato
8	DATA7	ENTRADA	Dato
9	DATA8	ENTRADA	Dato
10	ACK'	SALIDA	Pulso en bajo para indicar que el dato ha sido recibido y se esta listo para recibir otro.
11	BUSY	SALIDA	Si el BUSY esta en alto indica que la impresora, por cualquier razón, no puede recibir datos.
12	PE	SALIDA	Si esta señal esta en alto indica que no hay papel para imprimir.
13	SLCT	SALIDA	Esta indica que la impresora esta en estado de selección.
14	AUTO FEED XT'	ENTRADA	La señal es puesta a bajo cuando el papel debe ser alimentado automáticamente después de imprimir

PIN	SEÑAL	DIRECCION	DESCRIPCION
15	NC		No conectada.
16	OV		Tierra lógica del circuito.
17	CHAS. GND	-	Tierra física de la impresora, normalmente es el chasis.
18	NC	-	No conectada.
19-30	GND	-	Tierra de retorno para cada una de las líneas de datos y señales.
31	INIT'	ENTRADA	Cuando esta señal es puesta a bajo, el buffer de la impresora es limpiado, y la rutina de inicialización de la impresora es puesta en marcha.
32	ERROR'	SALIDA	Esta señal va a bajo cuando se presenta algún error de impresión, tal como falta de papel, impresora fuera de línea, etc.
33	GND	-	Misma conexión que los pines 19 al 30
34	NC	-	No conectada.
35			Resistencia Pull-up a +5V
36	SLCT IN'	ENTRADA	La entrada de datos a la impresora es posible si esta señal está en bajo.

Tabla de conexión y descripción¹² de los pines de la interface paralela para las impresoras IBM PC y EPSON FX-100
Tabla 3.1

La interfaz Centronics hace uso de 36 pines. La tabla 3.1 muestra la asignación de los pines y la descripción para este tipo de conector tal y como lo utilizan las impresoras IBM y EPSON, sin embargo, algunos fabricantes utilizan uno o dos pines de manera diferente.

La razón por la cual se utilizan treinta y seis pines de conexión resulta de la necesidad de que cada una de las señales tenga su propia señal de tierra de retorno para evitar al máximo los niveles de ruido

¹² "Dirección" se refiere al flujo de la señal vista desde la impresora

Todas las condiciones de la interface se halla basada en los niveles TTL

que pudieran existir en la transmisión de datos entre la computadora y la impresora.

Las señales de salida la impresora de mayor importancia son:

1. La señal de -ACKNOWLEDGE en el pin 10, que cuando se halla en un nivel lógico "0", indica que el carácter recibido ha sido aceptado y la impresora está lista para recibir el siguiente dato.
2. La señal de BUSY en el pin 11, que si se halla en un nivel lógico "1" indica por alguna razón que la impresora no está lista para recibir el siguiente carácter.
3. La señal PE en el pin 12, que va alto (un nivel lógico "1") si la impresora no tiene papel.
4. La señal SLCT en el pin 13, que está en alto si la impresora ha sido seleccionada para recibir datos.
5. La señal de -ERROR en el pin 32 que se halla en el nivel lógico "0" por una variedad de condiciones en problemas en la propia impresora.

En el capítulo siguiente se explica la manera en que se programan a los puertos del PPI y las señales de control que se reciben y se generan para asegurar la recepción de los datos que envía la computadora a nuestra impresora Braille.

3.5 Direccionamiento de las agujas de impresión

La impresión que se realiza a través del sistema es en relieve, para ello, la deformación del papel se lleva a cabo a través de serie de 64 solenoides, cada solenoide activa una aguja (véase la figura 3.3).

Estas se hallan divididas en 8 conjuntos de 8. Las señales de control para la activación de los solenoides están dadas por latches de ocho bits cada uno (circuitos u2, u3, u4, u5, u6, u7, u8 y u9 del diagrama de conexiones), ya que estos circuitos mantienen la información hasta que son habilitados para cargar nueva información. Los latches son direccionados con las direcciones 4000H, 4400H, 4800H, 4C00H, 5000H, 5400H, 5800H y 5C00H para los conjuntos 1, 2, 3, 4, 5, 6, 7 y 8 respectivamente.

La señal de enable de los latches está dada por la señal V_n del circuito u17 que es un decodificador, y la señal -WR (escritura) que genera el microcontrolador, estas dos señales se combinan a través de una compuerta NOR sincronizando así la carga de los datos en los latches.

Un "1" lógico activa al solenoide correspondiente al resultado de la traducción ASCII - Braille .

3.6 Accionamiento de los actuadores

Para el avance de renglón o de forma, indicar si la impresora se halla en estado de on-line (en línea), en estado de off-line (fuera de línea) y/o sin papel se proveen las señales lógicas necesarias a través del bus de datos y la dirección 8000H que corresponde a un circuito latch (circuito u10) que activa los actuadores del sistema. Los actuadores son simulados a través de leds ya que en esta etapa se hace el accionamiento de los motores y las alarmas que le indican al usuario que ha existido algún problema en la impresora correspondiendo esta parte de la impresora Braille a otro bloque del sistema general.

Los bits correspondientes a los actuadores son los siguientes:

- a) B_0 y B_1 : Corresponden al rodillo que presiona al papel contra las agujas activadas. Con los valores 00 y 11 el motor del rodillo no se activará, con los valores 01 el rodillo avanzará de izquierda a derecha y con los valores 10 el movimiento será en sentido contrario.
- b) $B_2 - B_4$: Proporciona la secuencia que mueve al motor de pasos encargado de hacer el avance de línea.
- c) B_3 : Enciende un led que le indica al usuario que la impresora se halla en línea.
- d) B_4 : Enciende el led que le indica al usuario que se encontró que no existe papel en la impresora.

La asignación de cada uno de los bits asignados para esta etapa pueden ser modificados a partir de la programación del sistema.

3.7 Sensores del sistema:

El sistema cuenta con cinco sensores que se hallan conectados al sistema a través de un circuito del tipo transceiver 74LS244 (circuito u15 del diagrama de conexiones), éstos envían una señal entre 0 y 5 Volts y le indican al sistema que es lo que sucede en el mundo exterior.

Los datos son leídos de los sensores cuando se verifican la señal de -RD y la dirección 8000_{16} , éstas señales se combinan en una compuerta OR para proporcionar la señal lógica que habilita al transceiver para permitir el paso de datos de los sensores hacia el bus de datos del sistema.

El prototipo tiene dos sensores del tipo switch que indican el momento en que el rodillo ha llegado a uno de los extremos de la impresora abriendo el circuito para evitar el flujo de corriente hacia el rodillo; un interruptor del tipo push button indica a la impresora

en que momento el usuario desea detener una impresión y coloca a la impresora en estado de off-line u on-line, otro más es utilizado para indicarle a la impresora que el usuario desea que la impresora realice un avance de línea y, por último un sensor del tipo fototransistor que se utiliza para detectar la existencia de papel en la impresora para imprimir. Los sensores ocupan únicamente cinco bits (B_0 - B_4 respectivamente) de tal forma que se pueden agregar al sistema tres sensores más para la comunicación del sistema con el mundo exterior.

3.8 Direccionamiento a los circuitos del hardware

El sistema utiliza dos decodificadores 74LS138 (circuitos u16 y u17) para direccionar a la memoria RAM 2764 (circuito u13), al PPI 8255A (circuito u14), a los circuitos del tipo latch (circuitos u2-u9) que corresponden a los solenoides que activan a las agujas, al circuito transceiver (circuito u15) que permite el paso de las señales provenientes de los sensores y al latch que corresponde a los actuadores (circuito u10). Cada uno de los elementos utiliza el bus de datos del sistema en el momento en el que se le indica a través del bus de direcciones.

Las líneas de direcciones A_{15} , A_{14} y A_{13} permiten al circuito u16 seleccionar la memoria RAM, el PPI, los latches o el transceiver, tal y como se muestra en la tabla 3.2.

A_{15}	A_{14}	A_{13}	Dirección	Elemento
0	0	0	0000-1FFF H	Memoria de datos
0	0	1	2000-3FFF H	P P I 8255A
0	1	0	4000-5FFF H	Solenoides
0	1	1	6000-7FFF H	Actuadores
1	0	0	8000-9FFF H	Sensores
1	0	1	A000-BFFF H	No usado
1	1	0	C000-DFFF H	No usado
1	1	1	E000-FFFF H	No usado

Tabla 3.1

Por ejemplo, para acceder a la memoria RAM es necesario que en las líneas de direcciones A_{15} - A_{13} existan un valor 000 respectivamente, con lo que tendremos posibilidad de acceder desde la dirección 0000H hasta la dirección 1FFFH puesto que se habilita en bajo la señal Y_0 del decodificador u16 habilitándose con ello el pin -CS1 (habilitar circuito) de la memoria RAM utilizando las líneas de dirección A_0 - A_{12} para direccionar a cada una de las palabras de la misma.

Para activar los solenoides de la impresora es necesario que las entradas del decodificador u16 tengan un 1002, con lo que accedaremos a partir de la dirección 4000₁₆ habilitándose la salida Y_2 que a la vez habilita a los pines de enable (habilitación) del circuito u17. El decodificador u17 accesa a uno y sólo un latch de los ocho que se utilizan para los solenoides utilizando las líneas de dirección A_{10} - A_{12} . Las líneas A_0 - A_9 se utilizan como condiciones de DON'T CARE (no importa su valor).

Cada uno de los circuitos latch que activan a las solenoides cargan la información que se encuentra en el bus de datos cuando estos son

A ₁₅	A ₁₄	A ₁₃	Dirección	Elemento
0	0	0	0000-1FFF H	Memoria de datos
0	0	1	2000-3FFF H	P P I 8255A
0	1	0	4000-5FFF H	Solenoides
0	1	1	6000-7FFF H	Actuadores
1	0	0	8000-9FFF H	Sensores
1	0	1	A000-BFFF H	No usado
1	1	0	C000-DFFF H	No usado
1	1	1	E000-FFFF H	No usado

Tabla 3.1

Por ejemplo, para acceder a la memoria RAM es necesario que en las líneas de direcciones A₁₅-A₁₃ existan un valor 000 respectivamente, con lo que tendremos posibilidad de acceder desde la dirección 0000H hasta la dirección 1FFFH puesto que se habilita en bajo la señal Y₀ del decodificador u16 habilitándose con ello el pin -CS1 (habilitar circuito) de la memoria RAM utilizando las líneas de dirección A₀-A₁₂ para direccionar a cada una de las palabras de la misma.

Para activar los solenoides de la impresora es necesario que las entradas del decodificador u16 tengan un 1002, con lo que accedaremos a partir de la dirección 4000_h habilitándose la salida Y₂ que a la vez habilita a los pines de enable (habilitación) del circuito u17. El decodificador u17 accesa a uno y sólo un latch de los ocho que se utilizan para los solenoides utilizando las líneas de dirección A₁₀-A₁₂. Las líneas A₀-A₉ se utilizan como condiciones de DON'T CARE (no importa su valor).

Cada uno de los circuitos latch que activan a las solenoides cargan la información que se encuentra en el bus de datos cuando estos son

seleccionados con las líneas Y_0 - Y_7 , del circuito u17 en combinación con la señal de $\sim WR$ a través de una compuerta OR.



CAPITULO 4

Programación del Sistema de Impresión**4.1 Introducción**

El sistema de impresión Braille ejecuta un programa almacenando en la memoria ROM del sistema. El diseño de la programación trata del desarrollo de los programas para una aplicación particular¹³. Este programa configura a la unidad de control en el CPU para programar al PPI de tal forma que se pueda realizar la comunicación de la computadora con el sistema de impresión, se realice la traducción a código Braille de los datos recibidos, checar el estado de los sensores, se activen a los solenoides y se active el rodillo para la impresión.

Escribir programas para el sistema de impresión es esencialmente igual que hacerlo para cualquier otra computadora. La única diferencia es que el programador del sistema debe estar familiarizado con la configuración de los circuitos (en cuanto a como se direccionan y se accesan) y no debe perder de vista que se trata de una aplicación particular.

La figura 4.1 muestra en general, el procedimiento que se lleva a cabo para lograr los objetivos propuestos en este trabajo (véase la sección 1.2).

Cada uno de los bloques mostrados en la figura 4.1 son explicados de manera más explícita en las siguientes secciones de este capítulo. Cada uno de los algoritmos de programación que se explican tienen su correspondiente subrutina en ensamblador, sin embargo, por tratarse de

¹³ M. Morris Mano; *Lógica Digital y Diseño de Computadores*, pág. 520.

un proyecto de innovación tecnológica nos reservamos el derecho de duplicar algunos de ellos en este trabajo.

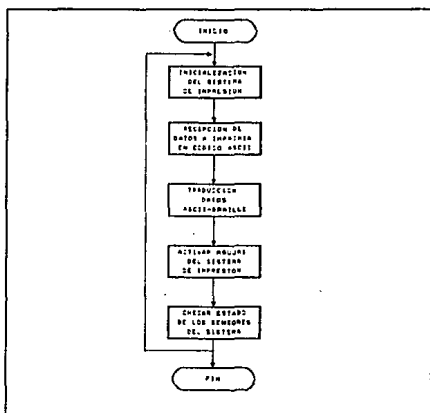


Diagrama de flujo del funcionamiento del Sistema de Impresión Braille
Fig. 4.1

4.2 Conjunto básico de instrucciones

Las instrucciones utilizadas en la programación del sistema, pueden clasificarse en tres tipos diferentes¹⁴:

- a) Instrucciones de transferencia que mueven datos entre registros, palabras de memoria y registros de interconexión sin cambiar el contenido de la información.

¹⁴ *Ibid*; pág. 535.

- b) Instrucciones de operación que realizan operaciones con los datos almacenados en los registros o palabras de memoria.
- c) Instrucciones de control utilizadas para causar un cambio en la secuencia del programa dependiendo de los resultados.

4.2.1 Instrucciones de transferencia

Una instrucción de movimiento (MOV) causa una transferencia de datos desde la fuente hasta su destino. La fuente o el destino puede ser un registro o un lugar de memoria. La instrucción de movimiento externo (MOVX) funciona análogamente a la instrucción MOV, con la diferencia que ésta se utiliza para hacer transferencias a direcciones externas utilizando el registro de 16 bits DPTR. Las instrucciones de insertar (PUSH) y sacar (POP) transfieren datos entre los registros procesadores y la pila¹⁵ de memoria.

4.2.2 Instrucciones de operación

Las instrucciones del tipo operativo ejecutan operaciones aritméticas y de desplazamiento entre los registros de palabras de memoria. Las pone a cero, a uno o complementan los bits de condición. Las instrucciones de operación típicas son: sumas (ADD), restas (SUBB), AND (ANL), OR (ORL), corrimiento de bits (RL), complementar y poner a cero (CLR C, CLR A).

¹⁵ Una pila consta de un bloque de memoria contigua accesada por una dirección que siempre se incrementa o decrementa después del acceso a memoria. El registro que almacena la dirección para la pila se llama STACK POINTER (Apuntador a la Pila) debido a que su valor indica siempre al ítem superior de la pila.

4.2.3 Instrucciones de control

Las instrucciones de tipo control tienen características para tomar decisiones y cambiar el camino (branch) tomado por el programa cuando se ejecuta. Las instrucciones de control se utilizan cada vez que el control debe ser transferido a una instrucción que esté fuera de la secuencia normal. Las instrucciones de control pueden ser condicionales (JG, CJNE, DJNZ, etc.)

o incondicionales (SJMP, LJMP, etc.).

Una instrucción de control cambia al contador del programa de manera que éste contenga la dirección de la instrucción que esté próxima a ejecutarse.

Hay tres tipos de instrucciones de control:

- a) Instrucciones de salto o bifurcación.
- b) Instrucciones de llamado y regreso a la subrutina.
- c) Instrucciones de omisión.

Las instrucciones de salto o bifurcación están asociadas con una dirección que especifica dónde se debe de hacer el salto o la bifurcación. Las instrucciones de subrutina, llaman a una subrutina¹⁶ y regresan a ejecutar la instrucción inmediata posterior de donde fue llamada. Una instrucción de omisión permite hacer una bifurcación a uno de dos lugares posibles, dependiendo del valor especificado de la condición del bit de status.

¹⁶ Una subrutina es una secuencia que contiene en sí instrucciones para ejecutar una tarea dada.

4.3 Algoritmo de Inicialización del Sistema

La inicialización del sistema Braille consta de varias etapas.

- 1.- Movimiento del rodillo a su posición original.
- 2.- Alineación y carga del papel.
- 3.- Eliminar los datos que puedan existir en los latches que activan a las agujas de impresión.
- 4.-Chequeo de los sensores para verificar la existencia de papel.
- 5.- Inicialización de las variables del programa. (cada una de las variables se halla en una localidad de Memoria).

Las etapas 1 y 2 corresponden al control de movimientos de motores de paso y de corriente directa, éstas etapas corresponden a otras personas del área electrónica de tal forma que no están dentro de los alcances de este trabajo. En la etapa 3 solo se deben direccionar los circuitos correspondientes a los solenoides y enviar que carguen un dato "0", con lo cual los solenoides se desactivan. La etapa 4 se halla como una subrutina, ya que ésta se ejecuta cada vez que se termina de imprimir un renglón y se realiza un avance de papel. En la etapa 5, se inicializan algunas variables que son importantes para la ejecución del programa, ya que de ella dependen algunas decisiones, tal es el caso de la variable ACTUAW (Palabra de actuadores) y RENGLONES (será el número de renglones por hoja).

La mayoría de los segmentos del programa que se presentan en este trabajo finalizan con una instrucción END (fin), sin embargo, si se desean utilizar como subrutinas habrá que cambiar ésta instrucción por RETURN (Retorno de Subrutina).

A continuación se muestra el programa de inicialización del Sistema de Impresión Braille.

```

;*****
;*****          Subrutina de inicialización
;*****
INICIALIZA:  MOV     RO, #ACTUAW      ;ACTUAW se halla en la
;localidad 24H
             MOV     A, #8AH
             MOV     @RO, A
             MOV     DPTR, #ACTUADORES ;ACTUADORES se halla
;en la localidad 6000H
             MOVX    @DPTR,A

             MOV     DPTR, #SENSORES  ;SENSORES se halla en
;la localidad 8000H
             MOVX    A, @DPTR
             ANL     A, #01H
             CJNE    A, #00H, CONTINUA
             MOV     DPTR, #ACTUADORES
             MOV     A, #0AH
             MOV     RO, #ACTUAW
             ORL     A, @RO
             MOV     @RO, A
             MOVX    @DPTR,A
             MOV     RO, #RENGLONES
             MOV     @RO, #0FH
CONTINUA:    LCALL   CHECKP          ;Manda a ejecutar la
;subrutina para checar
;papel.

             RET                      ;Retorno de la
;subrutina

```

La variable ACTUADORES activan las señales de alarma del sistema para indicar si la impresora se halla en estado de on-line u off-line o si existe un error en el sistema, tal y como se explicó en la sección 3.6.

La variable ACTUAW tiene almacenado el estado de los actuadores y es modificada cuando se manda a llamar a la subrutina CHECKP y la impresora no tiene papel o ha pasado al estado de off-line.

4.4 Algoritmo de Recepción de Datos

Para la mayoría de las impresoras comunes como la IBM PC, la Epson FX-80, y la NEC 8023, los datos a ser impresos son enviados en código ASCII en ocho líneas paralelas. La impresora recibe los caracteres y los almacena en un buffer de datos. Cuando la impresora detecta un carácter de retorno de carro (ODH carriage return), imprime el primer renglón de caracteres. Cuando detecta el segundo retorno de carro, se imprime el segundo renglón de caracteres. El proceso continúa hasta que todos los caracteres han sido impresos.

La transferencia de datos de una computadora hacia la impresora debe de estar hecha en base a algún método de handshake de transferencia de datos de forma paralela (véase la sección 2.4) ya que la computadora puede enviar los caracteres más rápidamente de lo que la impresora puede imprimirlos.

La impresora debe de alguna manera indicar a la computadora que no puede aceptar más caracteres hasta que haya impreso los que tiene en el buffer cuando éste se encuentre lleno.

El programa del sistema de impresión Braille tiene una subrutina que le permite recibir datos de la computadora, almacenarlos en memoria RAM, procesarlos e imprimirlos.

Lo primero que se hace es programar al 8255A (PPI) a través de su palabra de control para utilizar sus tres puestos de E/S y entablar la comunicación con la computadora.

```
CTRLW      EQU      2003H
ORG        0000H
SJMP      0040H
ORG        0040H
MOV        A,      #99H
MOV        DPTR,  #CTRLW
MOVX      @DPTR, A
END
```

Procedimiento de programación
para los puertos del PPI.
Fig. 4.2

La figura 4.2 muestra un programa para programar a los puertos del PPI. La dirección de la palabra de control del PPI es 2003H, con la instrucción MOVX @DPTR, A se transfiere el contenido del registro acumulador (que es 99H) hacia la dirección que tiene el registro DPTR (2003H).

Cada uno de los bits de la palabra de control de PPI tienen una función específica para la programación de los puertos¹⁷:

B₀: Parte baja del puerto C, si es "1" es entrada, si es "0" es salida.

B₁: Programación del puerto B, si es "1" es entrada, si es "0" es salida.

B₂: Modo de operación del puerto B, si es "0" modo 0 si es "1" modo 1.

B₃: Parte alta del puerto C, si es "0" es salida.

¹⁷ Para más información sobre los modos de funcionamiento de los puertos del PPI, consúltese el apéndice B.

B₄: Programación del puerto A, si es "1" es entrada.

B_{3,6}: Modo de operación del puerto B (Modo 0, Modo 1, Modo 2).

B₇: Indicador de MODE SET WORD.

De tal forma que si la palabra de control tiene un 99H (10011001) se tiene:

B₀: Parte baja Puerto C = Entrada

B₁: Puerto B = Salida

B₂: Puerto B = Modo 0

B₃: Parte alta Puerto C = Salida

B₄: Puerto A = Entrada

B_{5,6}: Puerto A = Modo 1

B₇: Set palabra de control

El puerto A del PPI tiene la dirección 2000H y se utiliza para la recepción de datos, el puerto B la dirección 2001H usándose para el envío de señales de control hacia la computadora, y la parte baja del puerto C para la recepción de señales de control de la computadora.

BIT	PUERTO A	PUERTO B	PUERTO C
0	Dato	~Error	~STB
1	Dato	PE	INIT
2	Dato	SLCT	
3	Dato	BUSY	
4	Dato	~ACK	
5	Dato		
6	Dato		
7	Dato		

Utilización de los bits de cada puerto del PPI 8255A
Tabla 4.1

Asumiendo que la impresora ha sido inicializada la computadora primero chequea la señal de BUSY (ocupado) para ver si la impresora está lista para recibir los datos. Si esta señal se encuentra en bajo, indicando que la impresora está lista (no ocupada), la computadora envía un código ASCII en las ocho líneas paralelas de datos. Después de al menos 0.5 microsegundos la computadora envía la señal de -STB en bajo para indicarle a la impresora que un carácter ha sido enviado. La impresora al detectar la señal de -STB en bajo coloca inmediatamente la señal de BUSY en alto. Después de un mínimo tiempo de 0.5 microsegundos la señal de -STB puede ir de nuevo en un nivel alto. Los datos son mantenidos válidos en las líneas de datos por 0.5 microsegundos después de que la señal de -STB se coloca en alto.

Cuando la impresora está lista para recibir el siguiente carácter, su señal de -ACK (acknowledge) se mantiene en bajo por 5 microsegundos. La señal de -ACK en bajo le indica a la computadora que puede enviar el siguiente carácter.

La señal de BUSY en bajo es otra indicación de que la impresora está lista para aceptar el siguiente carácter.

El diagrama de flujo de la figura 4.3 muestra el procedimiento de como se lleva a cabo la recepción de datos de la computadora utilizando el método Handshake Simple de Entrada Salida (Single Handshake Input Output)¹⁵.

¹⁵ Para mayor información consulte la sección 2.4

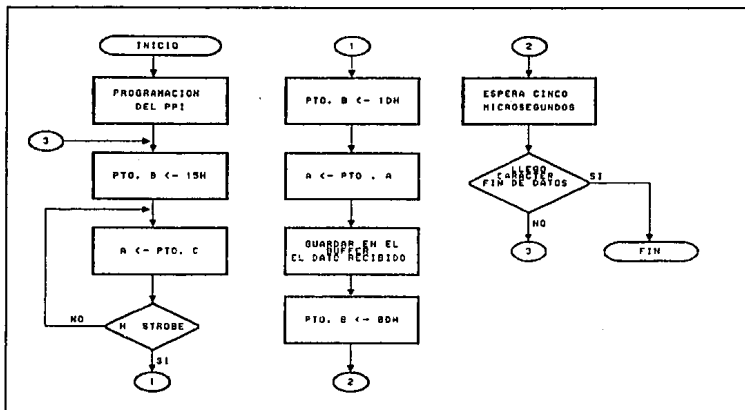


Fig. 4.3

Para llevar a cabo la recepción de datos, lo fundamental es tomar en cuenta en que momento se verifican las señales de control que emite y recibe el sistema. No se debe de perder de vista que cada una de las señales de control se halla en un sólo bit de los puertos B o C del PPI.

El puerto B tiene las señales de control que envía el sistema de impresión como se indica a continuación:

- B₀ : Señal de -ERROR
- B₁ : PE
- B₂ : SLCT
- B₃ : BUSY
- B₄ : -ACK
- B₅₋₇ : No usados

Así por ejemplo, cuando se encuentre un 15H por el puerto B, en realidad se colocan las señales de control como se muestra a continuación.

- 1 : Indica a la computadora que no hay error en el sistema de impresión.
- 0 : Indica que la impresora si tiene papel.
- 1 : Indica que la impresora está seleccionada.
- 0 : Indica que la impresora está no BUSY.
- 1 : Indica que la impresora no ha reconocido el carácter enviado.

A continuación se muestra el código del programa que ejecuta el microcontrolador para la recepción de datos.

```

; Programa de recepción de datos del sistema de
; Impresión Braille
; Este programa recibe los datos por el puerto A del
; PPI. Basado en el método de transferencia de datos
; Single Handshake Input Output

    BANKR      EQU      20H      ;# de bancos de 255
    ;datos recibidos
    DATAR      EQU      21H      ;contador de datos
    ;recibidos
    RENGLONES  EQU      25H
    PTOA       EQU      2000H
    PTOB       EQU      2001H
    PTOC       EQU      2002H
    CTRLW      EQU      2003H
    DATAINI   EQU      0000H    ;Inicio de datos en RAM

    ORG        0000H
    SJMP       0040H
    ORG        0040H

PROGPPI: MOV     A,      #99H
          MOV     DPTR,  #CTRLW
          MOVB   @DPTR, A
          LCALL  INICIALIZA ;Esta subrutina sirve para
          ;chechar el estado de los
          ;sensores del sistema
          MOV     RO,   #RENGLONES
          MOV     @RO,  #0FH
IRECIBE: MOV     RO,   #BANKR
          MOV     @RO,  #00H

```

```

MOV     R1,     #DATAR
MOV     @R1,    #00H
MOV     DPTR,   #DATAINI
PUSH    DPH
PUSH    DPL
BUSYLOW: MOV    DPTR,   #PTOB    ; 2 ciclos
MOV     A,     #15H    ; ack,-busy,slct,-pe,error
MOVX   @DPTR,  A      ; 2 ciclos
        ;-error,pe,slct,busy,-ack
MOV     DPTR,   #PTOC
STROBE: MOVX   A,     @DPTR    ;
ANL    A,     #01H    ;
CJNE   A,     #00H,   STROBE
BUSYH:  MOV     A,     #1DH
MOV     DPTR,   #PTOB    ; Salida de control
MOVX   @DPTR,  A      ; ack,busy,slct,-pe,error
LEE:    MOV     DPTR,   #PTOA   ; Puerto de datos
MOVX   A,     @DPTR    ; Recibir dato

METERAM: POP    DPL
POP     DPH
MOVX   @DPTR,  A
MOV     R2,    A
INC    DPTR
PUSH   DPH
PUSH   DPL
ACK:    MOV     DPTR,   #PTOB
MOV     A,     #0DH    ;-ack,busy,slct,-pe,error
MOVX   @DPTR,  A
NOP
NOP
NOP
INC    @R1
CJNE   @R1,   #0FFH,   CEDT
MOV     @R1,   #00H
INC    @R0
CJNE   @R0,   #08H,   CEDT
CEDT:  LJMP    CALLIMP
        R2,    #24H,   OTHDAT
OTHDAT: LJMP    BUSYLOW
CALLIMP: POP    DPL
POP     DPH
MOV     A,     #0AFH
MOVX   @DPTR,  A
LCALL  TRADUCE    ; Llama a la subrutina de
                  ; traducción ASCII-BRAILLE
CJNE   R2,    #24H,   RMD
RMD:   LJMP    ER
ER:    NOP     ; End Receive
LJMP   MAIN
END

```

La recepción de datos se lleva a cabo mientras no se reciba el carácter de fin de transmisión '\$' (carácter ASCII 24H) o se termine la capacidad del buffer de datos; éste tiene asignado 2KB en memoria RAM,

el resto del espacio de la memoria volátil se puede utilizar para el almacenamiento de datos intermedios. Una vez, que se ha cumplido alguna de las dos condiciones anteriores, el programa llama a la subrutina de traducción Ascii-Braille.

Mientras se está llevando a cabo la traducción Ascii-Braille, la impresora indica a la computadora que está en estado de Busy (ocupada) y que no puede recibir más caracteres. Una vez que termina de ejecutar la traducción, la impresora pasa al estado de no Busy (desocupada) para recibir más datos.

4.5 Algoritmo de Traducción Braille - Ascii

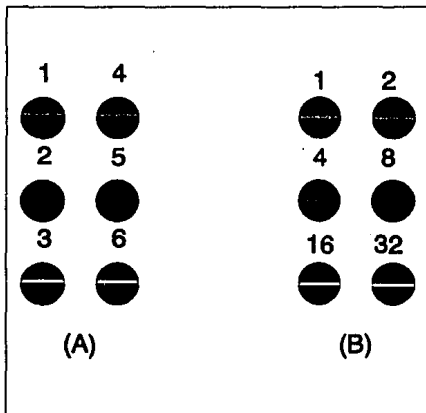
4.5.1 Representación binaria del alfabeto Braille.

El código ASCII se representa en ocho bits siendo posible tener hasta $2^8=256$ combinaciones diferentes posibles, cada combinación representa a uno y solo un carácter, de tal forma que el código Ascii tiene 256 caracteres diferentes.

Cada bit dentro del arreglo de ocho tiene un valor posicional en base 2 de tal forma que el bit 0 tiene el valor posicional $2^0 = 1$, el bit 1 el valor $2^1 = 2$ y así sucesivamente, así por ejemplo la letra "A" está representada en el código Ascii por el número 41_{10} (65_{10} o bien 01000001_2), la letra "B" por el número 42_{10} (66_{10} o su representación binaria 01000010_2), etc.

El sistema Braille, como se ha mencionado anteriormente, se forma a través de la combinación de 6 puntos en un arreglo matricial de 3×2 , de tal forma que se pueden tener hasta un máximo de 64 caracteres posibles, dependiendo de la presencia o ausencia de un punto, dando

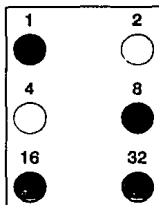
paso así a que cada carácter Braille pueda tener una representación análoga al código Ascii, la figura 4.4a muestra la asignación convencional de los puntos de la matriz Braille, sin embargo, para que los caracteres Braille tengan una representación binaria para hacer la traducción Ascii-Braille a cada punto de la matriz que componen a los caracteres Braille les asigné un valor posicional en base 2, de tal forma que el 1er. punto tenga un valor posicional $2^0=1$, el segundo $2^1=2$, y así sucesivamente como se muestra en la figura 4.4b.



(a) Valor posicional convencional de los puntos Braille; (b) valor posicional binario para la impresora diseñada

Fig. 4.4

Por ejemplo, la letra "Z" representada en la figura 4.5 tiene un valor numérico 57, que corresponde a la representación binaria 00111001, para nuestra tabla Braille, éste se obtiene de sumar cada uno de los valores posicionales correspondiente a cada punto en la matriz que representa al carácter. Cuando el punto se halla presente, se multiplica el valor posicional del mismo por "1" y cuando hay ausencia por "0", es decir:



Letra Z en Braille

Fig. 4.5

$$\text{Posición (1,1)} = 1 \times 1 = 1$$

$$\text{Posición (1,2)} = 2 \times 0 = 0$$

$$\text{Posición (2,1)} = 4 \times 0 = 0$$

$$\text{Posición (2,2)} = 8 \times 1 = 8$$

$$\text{Posición (3,1)} = 16 \times 1 = 16$$

$$\text{Posición (3,2)} = 32 \times 1 = 32$$

$$\text{Suma: } 1+0+0+8+16+32=57$$

De lo anteriormente mencionado se obtiene la tabla 4.2, en la cual se le asigna a cada carácter uno y sólo un valor que representa a cada uno de los caracteres y que permite al sistema de impresión traducir el código Ascii a Braille.

A	01H	B	05H
C	03H	D	0BH
E	09H	F	07H
G	0FH	H	0DH
I	06H	J	0EH
K	11H	L	15H
M	13H	N	1BH
O	19H	P	17H
Q	1FH	R	1DH
S	16H	T	1EH
U	31H	V	35H
W	2EH	X	33H
Y	3BH	Z	39H
0	0EH	1	01H
2	05H	3	03H
4	0BH	5	09H
6	07H	7	0FH
8	0DH	9	06H

Representación numérica de los caracteres para la conversión
 Ascii - Braille de acuerdo al diseño
 Tabla 4.2

El valor posicional en base 2 para el arreglo matricial no está dado al azar; la matriz de los seis puntos Braille esta distribuida en 3 renglones y 2 columnas, de tal forma que las primeros dos puntos (con valores 1, y 2) forman la 1er. línea del carácter, los puntos con valores 4 y 8 forman la segunda línea y los puntos con valor de 16 y 32 la tercera, las cuales son fáciles de obtener por separado a partir de la representación binaria del carácter utilizando para ello funciones de mascareo¹⁹ y de corrimiento²⁰ de registros.

¹⁹ Una función de mascareo es aquella que se utiliza para borrar selectivamente los bits de un registro a través de operaciones AND.

Por ejemplo: supóngase que se quiere obtener las 3 líneas que componen al carácter "Z" a partir de la representación binaria Braille del carácter.

Solución:

De la tabla 4.2 se obtiene que este carácter tiene la siguiente representación binaria: 00111001,

Para la primer línea se tendría que realizar una operación AND entre el valor binario y el número 00000011, bit a bit, puesto que se desean los 2 bits menos significativos, de tal forma que se obtiene:

```

00111001, Valor binario de "Z"
  AND
00000011,
-----
00000001, Valor binario de la 1a. línea de "Z"

```

Para la segunda línea se ejecuta similarmente una operación AND pero como se desean obtener los puntos cuyos valores son 4 y 8 (b2 y b3 bits de Byte) respectivamente, ésta tiene que realizarse con un 12:

```

00111001,
  AND
00001100,
-----
00001000,

```

Sin embargo estos bits tienen los valores posicionales 4 y 8 y se requiere que ocupen los bits B0 y B1 del Byte respectivamente, de tal forma que es necesario ejecutar 2 corrimientos de bits a la derecha dentro del byte, quedando el valor:

² Un corrimiento de bits en un registro se realiza cuando se desplaza la información binaria del mismo hacia la izquierda o hacia la derecha.


```

                00001000,
1er. corrimiento: 00000100,
2do. corrimiento: 00000010,   Valor de la 2a. línea de "Z"

```

Análogamente para la tercera línea:

```

                00111001,
                 AND
                00110000,
                -----
                00110000,

```

Pero ahora se ejecutan 4 corrimientos de bits dentro del byte a la derecha, dando como resultado:

```

1er. corrimiento           00110000,
2do. corrimiento          00011000,
3er. corrimiento          00001100,
4to. corrimiento          00000011,

```

De tal forma que se obtiene:

```

01 1ª línea
10 2ª línea
11 3ª línea

```

Como es de notarse cada línea del carácter esta representada por únicamente 2 bits, de tal forma que en un byte se puede almacenar el resultado de la traducción de cuatro caracteres.

4.2.2. Almacenamiento del resultado de la conversión Ascii - Braille.

El microcontrolador 8031 de Intel tiene más memoria RAM interna de 256 bytes, de los cuales 32 están agrupados en 4 bancos de 8 registros, que son conocidos como R0, R1,....., R7. La palabra de estatus de Programa (PSW), selecciona que banco de registros esta en uso.

Tomando en cuenta esto, se dio a cada banco de registros un propósito especial:

BANCO	DIRECCION	PROPOSITO
0	00H - 07H	Registros de propósito general.
1	08H - 0FH	1er. línea de la conversión A - B
2	10H - 17H	2da. línea de la conversión A - B
3	18H - 1FH	3er. línea de la conversión A - B

De tal forma que se tiene:

$$1 \text{ banco} * \frac{8 \text{ registros}}{1 \text{ banco}} * \frac{8 \text{ bits}}{1 \text{ registro}} = 64 \text{ bits}$$

El banco 0 es utilizado para variables del programa, en tanto que los bancos 1, 2 y 3 son utilizados para almacenar el resultado de la conversión de la 1er., 2da. y 3er. línea respectivamente de los 32 caracteres que caben en los 24 bytes de los tres bancos. Cada byte de cada registro almacena una línea de los caracteres Braille.

El primer carácter de los cuatro que caben en un byte ocupa los dos bits menos significativos del byte, en tanto que el cuarto carácter ocupará los dos bits más significativos. Así por ejemplo, el resultado de la impresión en Braille del lema universitario "POR MI RAZA HABLARA EL ESPIRITU" se obtendría tal y como se describe a continuación:

1er. byte de cada banco:

P	O	R	-	
11	10	10	00	= 17 ₁₆
10	01	11	00	= 39 ₁₆
10	10	10	00	= 15 ₁₆

2do. byte de cada banco:

M	I	-	R	
				= 4B ₁₆
				= C4 ₁₆
				= 41 ₁₆

3er. byte de cada banco:

A	Z	A	-	
				= 15 ₁₆
				= 08 ₁₆
				= 0C ₁₆

4to. byte de cada banco:

H	A	B	L	
				= 55 ₁₆
				= 53 ₁₆
				= 40 ₁₆

5to. byte de cada banco:

A	R	A	-	
				= 15 ₁₆
				= 0C ₁₆
				= 04 ₁₆





6to. byte de cada banco:

E	L	-	E	
				= 45 ₁₆
				= 86 ₁₆
				= 04 ₁₆

7to. byte de cada banco:

S	P	I	R	
				= 6E ₁₆
				= D5 ₁₆
				= 45 ₁₆

8vo. byte de cada banco:

I	T	U	.	
				
01	01	10	00	= 1A ₁₆
10	11	00	00	= 0D ₁₆
00	10	11	10	= 47 ₁₆

donde:

: Espacio en blanco

1 : Presencia de un punto

0 : Ausencia del punto

la línea continua denotata ausencia de deformación de papel

Cada uno de los 1's activan a un solenoide para que se levante una aguja de impresión. Cuando el carácter a imprimir es un retorno de carro (carriage return), o se han traducido a Braille treinta y dos caracteres, el sistema activa a las agujas de impresión de acuerdo al resultado de la traducción obtenida.

4.6 Activación de las agujas de Impresión

En el programa del sistema de impresión Braille, se halla escrita una subrutina que manda el resultado de la traducción a los latches que activan a las agujas de impresión. Hay que recordar que de acuerdo al diagrama de conexiones, cada uno de los circuitos latch tiene una dirección específica y que cada uno activan también a ocho solenoides (puesto que el bus de datos y los circuitos latch son de ocho bits).

Las direcciones en la que se hallan guardados el resultado de la traducción Braille-Ascii son de la 08H hasta la 20H y se direccionan con el registro R0.

;*****

; Subrutina que envía el resultado de la traducción

; Ascii-Braille a los circuitos que activan a los

; Solenoides

;*****

```

IMPRIME:  PUSH      R1
          PUSH      R2
          PUSH      R3
          PUSH      DPL
          PUSH      DPH
          MOV       RO,      #08H
          MOV       A,      @RO
          MOV       DPTR,    #4000H
          MOVX      @DPTR,   A
          INC       RO
          MOV       A,      @RO
          MOV       DPTR,    #4400H
          MOVX      @DPTR,   A
          INC       RO
          MOV       A,      @RO
          MOV       DPTR,    #4800H
          MOVX      @DPTR,   A
          INC       RO
          MOV       A,      @RO
          MOV       DPTR,    #4C00H
          MOVX      @DPTR,   A
          INC       RO
          MOV       A,      @RO
          MOV       DPTR,    #5000H
          MOVX      @DPTR,   A
          INC       RO
          MOV       A,      @RO
          MOV       DPTR,    #5400H
          MOVX      @DPTR,   A
          INC       RO
          MOV       A,      @RO
          MOV       DPTR,    #5800H
          MOVX      @DPTR,   A
          INC       RO
          MOV       A,      @RO
          MOV       DPTR,    #5C00H
          MOVX      @DPTR,   A
          INC       RO
          CJNE     RO, #20H, ESPERA ; Cuando RO es 20H
          ; indica que ya se
          ; imprimieron los 24
          ; bytes de datos
          LCALL    AVANZARENLON
          LCALL    CARGAHOJA
          LCALL    CLEANAGU
          POP      DPH
          POP      DPL
          POP      R3
          POP      R2
          POP      R1
          RET
          ; RET de imprime

```

4.7 Algoritmo de verificación de sensores

Los sensores son verificados cuando se inicializa el sistema o bien cuando se termina de imprimir un renglón de 32 caracteres Braille, como se explicó en el capítulo 3, los sensores se hallan conectados al sistema a través de un circuito transceiver y tienen la dirección 8000H.

El sensor del bit 0 es del tipo fototransistor n-p-n y es utilizado para checar que la impresora tenga papel para continuar con la impresión. El bit 1 es el utilizado por el sensor que sirve para indicarle a la impresora si el usuario presionó el botón de on-line. El 3er. bit es utilizado en el momento en que la impresora se halla en estado de off-line y le indica a la impresora que ejecute un avance de renglón. Los bits 4 y 5 son usados para detectar el momento en el que el rodillo de impresión ha llegado a uno de los extremos del mueble de la impresora.

4.7.1 Sensor para chequeo de papel

El sensor que detecta papel emite un valor lógico de "1" mientras que se pueda detectar el mismo, en el momento en que deja de existir papel en el sistema, el sensor emite un "0".

Cuando el sistema de impresión no tiene papel, la impresora Braille envía a la computadora una señal en la cual se le indica que ya no hay papel, en la impresora se activa la alarma correspondiente a que ha ocurrido un error en el sistema y la impresora queda automáticamente fuera de línea hasta que se vuelva a detectar la existencia de papel. La subrutina CHECKP (checa papel) realiza estas operaciones:

```

;*****
; Esta subrutina sirve para realizar el chequeo de papel
; ya sea después de un avance de renglón manual, automático
; o bien cuando se realiza una carga de papel
;*****
CHECKP:  PUSH    DPL
        PUSH    DPH
        PUSH    RO
CHECKP2: MOV     DPTR, #SENSORES
        MOVX   A, @DPTR
        ANL   A, #01H
        CJNE  A, #00H, SIHAY
        MOV   RO, #REGLONES
        MOV   @RO, #0FH ; Número de renglones en una hoja
        MOV   RO, #ACTUAW
        MOV   A, #OCFH
        ANL   A, @RO
        MOV   @RO, A
        MOV   A, #20H
        ORL   A, @RO
        MOV   @RO, A
        MOV   DPTR, #ACTUADORES
        MOVX  @DPTR, A
        MOV   DPTR, #PTOB ; Envía a la computadora la señal
                           ; de que ha ocurrido un error

        MOV   A, #1AH
        MOVX  @DPTR, A
        SJMP  CHECKP2
SIHAY:  MOV   A, #OCFH
        ANL   A, @RO
        MOV   @RO, A
        MOV   A, #10H
        ORL   A, @RO
        MOV   @RO, A
        MOV   DPTR, #ACTUADORES
        MOVX  @DPTR, A
        MOV   @RO, A
        POP   RO
        POP   DPH
        POP   DPL
        RET   ; Retorno de CHECKP

```

4.7.2 Sensor de on-line y line-feed

Se ha mencionado con anterioridad que la tarjeta del sistema de impresión trabaja a una velocidad de 6.6 MHz. (6,600,000 ciclos por segundo) por lo cual cuando el usuario presione un botón el sistema detectaría un dato erróneo debido a la velocidad del sistema, por lo cual para que se detecte el momento en que el usuario ha presionado el botón de on-line o de line-feed se realizó un procedimiento en el cual el sistema verifica que efectivamente se ha presionado el botón y que

el usuario desea poner a la impresora de un estado a otro. El procedimiento que se utiliza es generar un flip-flop por software, es decir, un procedimiento a través en el cual se garantice que se ha recibido la señal de +5V que proviene del sensor, el algoritmo para realizar esto es:

```
ESPERA: LEE SENSOR ONLINE
        SI SENSOR ONLINE SIGUE PRESIONADO SALTA A ESPERA
        SALIR
```

Tomando en cuenta el algoritmo anterior, la programación para checar el botón de on-line es:

```
ESPERA1: MOVX A, @DPTR
         ANL A, #02H
         CJNE A, #00H, ESPERA1
```

donde:

DPTR: Tiene la dirección de los sensores del sistema.
y el sensor de on-line ocupa el segundo bit.

La subrutina que ejecuta el chequeo de los sensores de on-line y de line-feed se muestra a continuación:

```
;*****
; Subrutina de chequeo de sensores de on-line y line-feed
;*****
ONLINE:  PUSH    DPL
         PUSH    DPH
         PUSH    R0
         PUSH    R1
         CLR     C
         MOV     RO,    #REGLONES

CHKOL:   MOV     R1,    #ACTUAW
         MOV     A,     #10H
         ANL    A,     @R1
         CJNE   A,     #10H,    NLOL
         MOV    DPTR,   #SENSORES
         MOVX   A,     @DPTR
         ANL    A,     #02H
         CJNE   A,     #02H,    SALIROL

ESPERA1: MOVX   A,     @DPTR
         ANL    A,     #02H
         CJNE   A,     #00H,    ESPERA1
```



```

MOV      A,      #0EFH
ANL      A,      @R1
MOV      @R1,    A
MOV      DPTR,   #ACTUADORES
MOVX     @DPTR,  A

NLOL:    MOV      DPTR,   #SENSORES
MOVX     A,      @DPTR
ANL      A,      #04H
CJNE     A,      #04H,    SLOL

ESPERA2: MOVX     A,      @DPTR
ANL      A,      #04H
CJNE     A,      #00H,    ESPERA2
MOV      RO,     #REGLONES
DEC      @RO
CJNE     @RO,    #00H,    MUESTRA
MOV      @RO,    #0FH
;LCALL   CARGAPAPEL
MUESTRA: MOV      @RO,    A
MOV      P1,     A

SLOL:    MOVX     A,      @DPTP
ANL      A,      #02H
CJNE     A,      #02H,    CHKOL

ESPERA3: MOVX     A,      @DPTR
ANL      A,      #02H
CJNE     A,      #00H,    ESPERA3

MOV      A,      #10H
ORL      A,      @R1
MOV      @R1,    A
MOV      DPTR,   #ACTUADORES
MOVX     @DPTR,  A

SALIROL: POP      R1
POP      RO
POP      DPH
POP      DPL
LCALL   CHECKP

RET

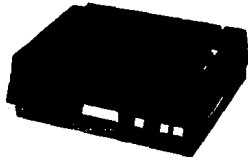
```

4.8 Ensamblado de los programas

Los programas son editados en cualquier editor de texto que deje los archivos en código ASCII y se nombra al archivo con la extensión .ASM, posteriormente se utiliza el programa CYS8051.EXE que es el que ensambla al programa fuente generando los archivos con extensiones .LST y .HEX .

El archivo .LST es un listado del programa fuente en el que se indican las direcciones de las etiquetas utilizadas y los errores que se tienen en el programa.

El archivo .HEX tiene un formato especial en el cual se hallan las instrucciones del programa en su correspondiente valor hexadecimal del set de instrucciones del microcontrolador; éste se utiliza para generar un archivo binario utilizando el programa HEXBIN y que sirve para programar a la memoria ROM.



CAPITULO 5

Conclusiones

Los sistemas de cómputo a nivel de hardware son desarrollados básicamente por los países que tienen la tecnología necesaria para hacerlo, en México se requiere del desarrollo de tecnología nacional para no llegar a ser únicamente un país maquilador e importador de productos extranjeros y éste es uno de los objetivos fundamentales de la Facultad de Ingeniería de la UNAM: formar a ingenieros capacitados para desarrollar nuevas tecnologías para el progreso del país.

Uno de los objetivos planteados en este trabajo fue precisamente el desarrollar y construir con tecnología nacional una impresora de código Braille con elementos que se encuentren en el mercado nacional; obviamente falta aún mucho por hacer para poder llegar a competir con las grandes potencias tecnológicas del mundo, sin embargo este es un comienzo. La impresora Braille que se planteó construir es aún muy modesta en comparación con otras impresoras ya existentes en el mercado pero ésta es la primera construida en México con tecnología nacional, este trabajo únicamente abarcó el desarrollo de la tarjeta controladora para este sistema.

El desarrollo de tecnología se da gracias a las necesidades de solucionar algún problema de la sociedad y requiere de una gran imaginación y experiencia por parte de los desarrolladores para lograrlo. Este trabajo me proporcionó la oportunidad de aprender y reafirmar mis conocimientos obtenidos en la facultad en áreas de la programación de sistemas y diseño de sistemas digitales en la arquitectura de computadoras.

Existe un refrán popular el cual dice: "si las cosas difíciles se hicieran fáciles cualquiera las haría" y esto es cierto en este caso, ya que el diseñar en papel una arquitectura para un sistema de cómputo aparentemente es sencilla, sin embargo en el momento de la implementación de los elementos del hardware se presentaron algunos problemas que solamente se pueden solucionar a través de la práctica y a la experiencia, tal como fue la sincronía de los elementos del sistema para realizar de una manera correcta la transferencia de los datos entre ellos.

La arquitectura basada en el uso de microcontroladores permitió desarrollar este trabajo de una manera más eficiente y rápida ya que generan las señales necesarias para su fácil implementación con circuitos combinatoriales además de contar con su propia unidad de control, registros internos, memoria y buses de datos, direcciones y control; su set de instrucciones es sencillo y fácil de aprender proporcionando las instrucciones aritméticas, lógicas, de bifurcación y de transferencia necesarias.

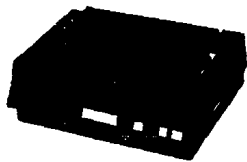
La implementación del software con el hardware del sistema requiere únicamente de algunos fundamentos de programación a nivel de ensamblador y de conocer perfectamente la arquitectura que se está programando de tal forma que es posible programar cualquier sistema de cómputo en lenguaje de bajo nivel si se conoce de que manera funciona internamente.

Desafortunadamente, aún no se puede mostrar el sistema de impresión completo debido a que es un proyecto multidisciplinario en el cual

intervienen ingenieros mecánicos, electrónicos y en computación, requiriendo un poco más de tiempo para desarrollar cada una de sus etapas correspondientes.

No dudo que posteriormente existan modificaciones al diseño del hardware proporcionado en este trabajo debido a nuevas necesidades del Sistema de Impresión para mejorarlo.

Sinceramente espero, que este trabajo sirva de alguna manera como guía para las generaciones posteriores a la mía y que tengan la inquietud de hacer de México un país cada vez mejor, más independiente tecnológicamente; y que los ingenieros estemos siempre al servicio de la sociedad.



APENDICES

Características del microcontrolador 8031 de Intel

a) Características:

El microcontrolador 8031 de Intel permite la implementación de sistemas digitales permitiendo con ello el fácil desarrollo de procedimientos algorítmicos a nivel microprogramación para el manejo de datos e información de control.

El conjunto de instrucciones que utiliza es sencillo de entender y programar además provee las señales de control necesarias para la escritura y lectura de datos hacia otros circuitos digitales utilizando un puerto de 8 bits para el manejo de datos y direcciones bajas y un segundo puerto (puerto 2) para el manejo de direcciones altas.

b) Arquitectura Básica

El 8031 es parte de la familia de microcontroladores MCS 8051 de Intel, sin embargo, no tiene memoria EEPROM interna, por lo cual se hace necesaria que el BIOS del sistema se halle en una memoria EEPROM externa de hasta 8KB de capacidad. En la tabla A.1 se encuentra una comparación de esta familia de microcontroladores.

Dispositivo	Memoria de Programa	Memoria de Datos (RAM)	Timers/- Counters	Interrupciones
8052 AH	8Kx8 ROM	256x8	3x16	6
8051 AH	4Kx8 ROM	128x8	2x16	5
8051	4Kx8 ROM	128x8	2x16	5
8032 AH	Ninguna	256x8	3x16	6
8031 AH	Ninguna	128x8	2x16	5
8031	Ninguna	128x8	2x16	5
8751 H	4Kx8 EPROM	128x8	2x16	5
8751H-8	4Kx8 EPROM	128x8	2x16	5

Tabla A.1

El 8031 tiene la siguiente arquitectura base:

- Bus datos de 8 bits.
- Capacidad de realizar operaciones Aritmético-lógicas.
- Capacidad de direccionamiento de hasta 64 KB utilizando 16 líneas de direccionamiento distribuidas en 2 puertos de 8 bits cada uno.
- Dos timers/contadores de 16 bits.
- Tiene 5 vectores de interrupción con dos niveles de prioridad.
- Un oscilador

b.1) Distribución de la memoria:

Como se puede observar en la tabla A.1 el microcontrolador cuenta con una memoria RAM interna la cual se divide en localidades para instrucciones y datos, las líneas de direcciones permiten acceder hasta 8KB de memoria RAM externa mediante el empleo del registro denominado DPTR que es de 16 bits.

i) Memoria de programa:

Después de un RESET el CPU inicia la ejecución del programa a partir de la localidad 0000H de memoria. Cada interrupción es asignada a una localidad fija en la memoria de programa. La interrupción causa que el CPU salte a aquella localidad donde iniciará la ejecución del o los programas.

Los 4KB de memoria de programa pueden estar incluidos en el microcontrolador o ser provistos por una memoria ROM externa, dependiendo de la versión del microcontrolador que se este empleando.

Mediante la conexión de EA' (External Acces, pin 31) a Vcc ó a Vss se le indica al microcontrolador donde debe buscar el programa a ejecutar. Si EA' se conecta a Vcc, el programa deberá estar en memoria interna. El programa podrá ocupar desde la dirección 0000H y hasta la FFFFH en la memoria interna de programa. Si EA' se conecta a Vss, entonces el programa será buscado en memoria externa y se habilitara la señal para lectura de memoria externa: PSEN' permite leer las instrucciones del programa contenido en la ROM externa. El circuito básico para la ejecución de un programa en memoria externa es el de la figura A.1. La señal ALE' permite la separación de direcciones y datos, habilitando o deshabilitando un latch.

ii) Memoria de datos:

En la figura A.2 se muestra el diagrama básico para acceder a memoria de datos externa. El programa en este caso esta contenido en memoria interna. El Puerto 0 sirve como multiplexor del bus direcciones/datos hacia la memoria de datos externa (RAM) y tres líneas del Puerto 2 son

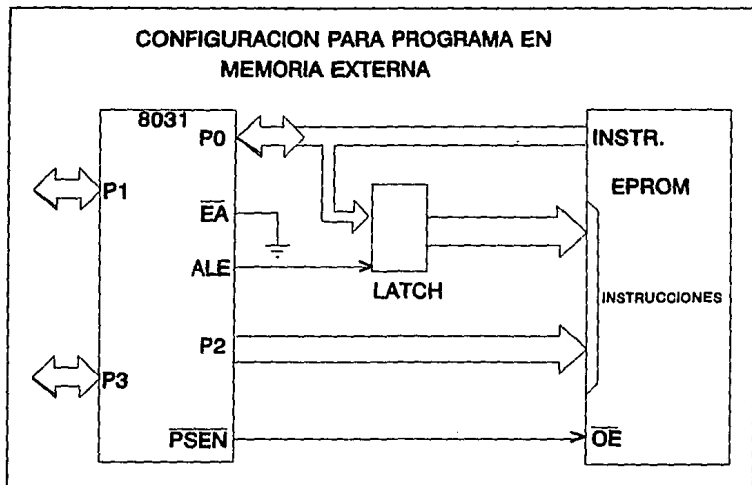


Fig. A.1

usadas para direccionar la página en la RAM. El microcontrolador genera las señales RD' y WR', que son necesarias durante el acceso a RAM externa.

El microcontrolador puede direccionar hasta 64KB de memoria externa de datos y el direccionamiento puede ser con uno o dos bytes de longitud (8 ó 16 líneas respectivamente); cuando se direcciona con 8 líneas (1 byte) se utiliza el Puerto 0, para direccionar con 16 líneas las 8 líneas más significativas (byte superior) son emitidas por el Puerto 2 y las 8 líneas menos significativas (byte inferior) por el Puerto 0. La longitud del dato leído es de ocho bits.

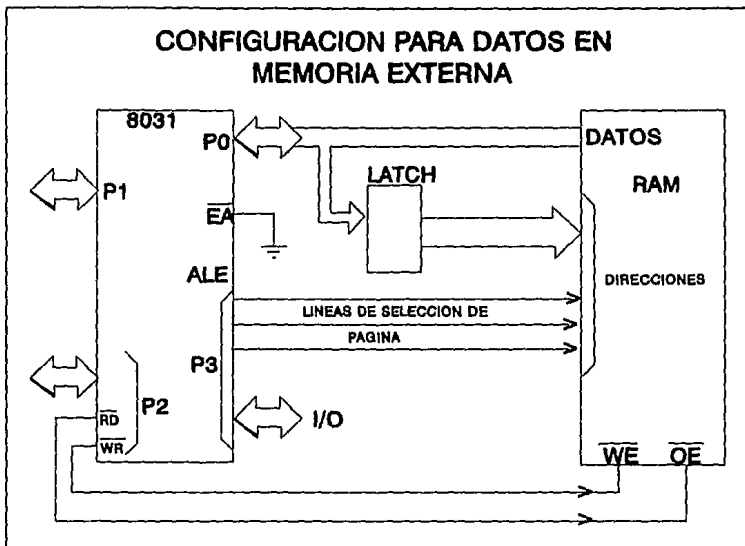


Fig. A.2

El microcontrolador tiene una memoria interna para datos de 256 bytes, lo cual implica que se puede direccionar máximo con 8 líneas, sin embargo usando los modos de direccionamiento para memoria interna se puede simular que se tienen 384 bytes de memoria RAM, ya que direccionando directamente por arriba de la dirección 7FH se accesa una localidad de memoria y haciendo el direccionamiento en forma indirecta se accesa a una localidad diferente de memoria. En la figura A.3 se

observa que los 128 bytes altos y el espacio SFR (Special Function Register), ocupan el mismo bloque de memoria, de la 80H a la FFH.

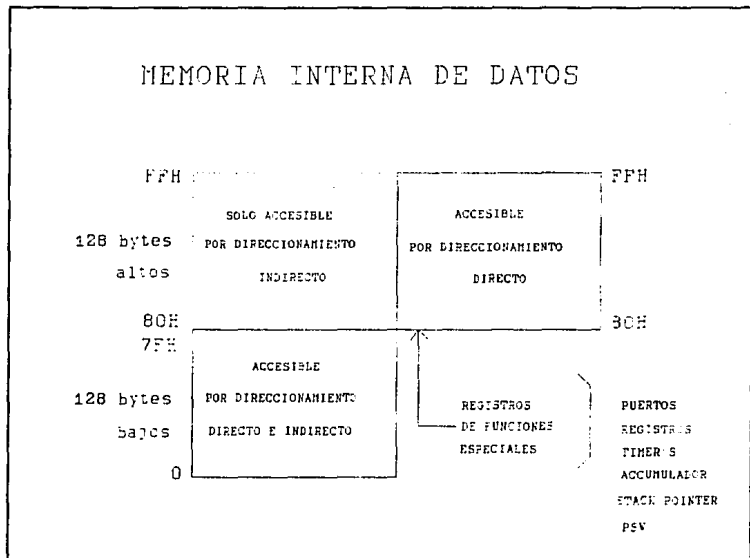


Fig. A.3

Los 128 bytes bajos de la RAM interna encontramos que los 32 primeros bytes son agrupados en 4 bancos de 8 registros cada uno. La instrucciones de programa pueden referirse a esos registros como R0 hasta R7. Dos bits en el PSW (Program Status Word) seleccionan que banco será usado, tal y como se muestra en la figura A.4, los siguientes 16 bytes son un

bloque de memoria direccionable directa o indirectamente al igual que los 128 bytes, de la dirección 0000F hasta la FFFFH.

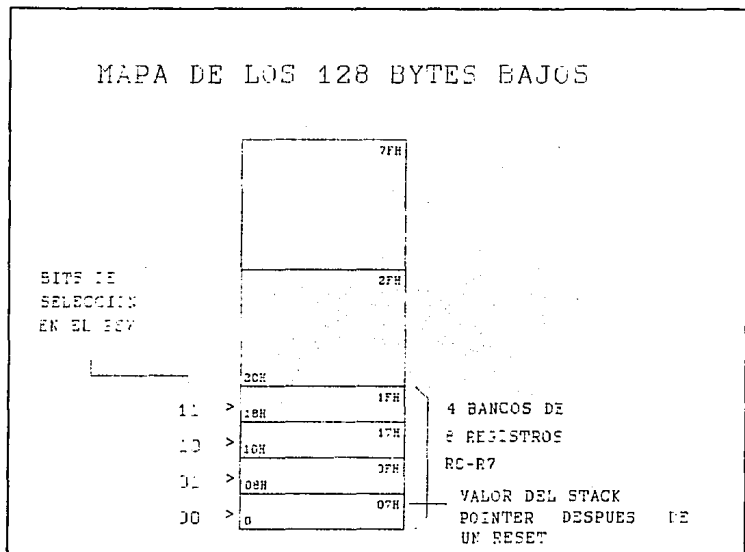


Fig. A.4

Program Status Word. El PSW contiene los bits de estado que reflejan la corriente de estados de ejecución en el microcontrolador. Reside en el área de memoria denominada SFR. Contiene el bit de acarreo, un bit auxiliar de acarreo (para operaciones BCD), dos bits para la selección

del banco de registros, el bit o bandera de Overflow, el bit de paridad y dos bits que pueden ser definidos por el usuario.

b.2) Modos de direccionamiento:

En el microcontrolador 8031 se tienen básicamente 6 modos de direccionamiento:

i) Direccionamiento Directo. Se especifica la dirección donde se encuentra el operando:

```
ADD    A,7FH
```

ii) Direccionamiento Indirecto. Se especifica un registro que contiene la dirección del operando, el símbolo @ indica este tipo de direccionamiento:

```
ADD    A,@R0
```

iii) Instrucciones sobre registros. Emplea un registro que contiene al operando, sobre el cual se realizara una determina operación:

```
ADD    A,R7
```

iv) Instrucciones sobre registros específicos. Algunas instrucciones del microprocesador 8031 solo pueden realizarse sobre un único registro, tal es el caso de la operación ADD, que hemos estado empleando para ejemplificar, que solo aceptara el registro ACUMULADOR (A) como primer operando y en el colocara el resultado de la operación.

v) Direccionamiento Inmediato. Puede realizarse una operación sobre un valor constante, el símbolo # indica que la operación se realizara sobre el valor especificado:

```
ADD    A,#100
```

vi) Direccionamiento indexado. Este modo de direccionamiento es muy utilizado en los casos en donde se tengan que acceder y leer tablas de datos especificadas dentro de la memoria de programa.

b.3) Instrucciones:

El set de instrucciones²¹ para el ensamblador del 8031, cuenta como en todos los lenguajes de programación con instrucciones aritméticas, lógicas, de salto y de transferencia de datos (interna y externa):

i) Aritméticas: las instrucciones aritméticas se listan en la tabla A.2, que indica los modos de direccionamiento que pueden ser empleados sobre la instrucción.

Nemónico	Operación	Modos direccionamiento
ADD A, <by>	$A = A + \text{<by>}$	D/I/R/Im
ADDC A, <by>	$A = A + \text{<by>} + C$	D/I/R/Im
SUBB A, <by>	$A = A - \text{<by>} - C$	D/I/R/Im
INC A	$A = A + 1$	ACUMULADOR
INC <by>	$\text{<by>} = \text{<by>} + 1$	D/I/R
INC DPTR	$DPTR = DPTR + 1$	SOLO DPTR
DEC A	$A = A - 1$	ACUMULADOR
DEC <by>	$\text{<by>} = \text{<by>} - 1$	D/I/R

²¹Para mayor información sobre el set de instrucciones del 8031, referirse a PROGRAMMER'S GUIDE AND INSTRUCTION SET del 8051.

MUL AB	B:A = B x A	ACC y B
DIV AB	A=INT(A/B) B=MOD(A/B)	ACC y B
DA A	Ajuste Decimal	ACUMULADOR

Tabla A.2

donde:

- <by> : operando,
- #d : es un valor constante,
- D : dir. directo,
- I : dir. indirecto,
- R : dir. de registro e
- Im : dir. inmediato.

ii) Instrucciones Lógicas: aquellas que realizan operaciones Boleanas sobre bytes o sobre bits. Para ejemplificar podemos mencionar la instrucción ANL o la ORL que realizan una AND y OR lógica respectivamente.

Nemónico	Operación	Modos direccio-
ANL A, <by>	A=A AND <by>	D/I/R/Im
ANL <by>, A	<by>=<by> AND A	D
ANL <by>, #d	<by>=<by> AND #d	D
ORL A, <by>	A=A OR <by>	D/I/R/Im
ORL <by>, A	<by>=<by> OR A	D
ORL <by>, #d	<by>=<by> OR #d	D
XRL A, <by>	A=A XOR <by>	D/I/R/Im
XRL A, <by>	<by>=<by> XOR A	D
XRL A, <by>	<by>=<by> XOR #d	D
CLR A	A=00H	Acumulador

CPL A	A=NOT A	Acumulador
RL A	Rota izq 1 bit de A	Acumulador
RLC A	Rota izq hasta Carry	Acumulador
RR A	Rota der 1 bit de A	Acumulador
RRC A	Rota der hasta Carry	Acumulador
SWAP A	Intercam. Nibbles A	Acumulador

Tabla A.3

iii) Instrucciones de transferencia: las instrucciones que permiten las construcciones iterativas y de salto dentro de un programa; tal es el caso de MOV que transfiere un byte en la memoria interna del microcontrolador y JMP que realiza una bifurcación dentro del programa a donde se indique.

Nemónico	Operación	Modo Direccionamiento
MOV A, <sc>	A = <sc>	D/I/R/Im
MOV <de>, A	<de> = A	D/I/R
MOV <de>, sc	<de> = <sc>	D/I/R/Im
MOV DPTR, #dato 16	DPTR = Constante de 16 bits	Im
PUSH <sc>	INC SP:MOV "@SP", <sc>	D
POP <de>	MOV <de>, "@SP":DEC SP	D
XCH A, <by>	ACC y <by> intercambian dato	D/I/R
XCHD A, @Ri	ACC y @Ri intercambian nibble bajo	I

Tabla A.4

c) Distribución de pines en el 8031:

El 8031 se presenta con una distribución de 40 pines para las señales de control, puertos de I/O y reloj.

- Vcc es voltaje de alimentación (+5V) y tiene asignada el pin 40.
- Vss es la tierra del circuito y esta en el pin 20.
- El Puerto 0 tiene asignadas 8 pines, del 32 al 39, de tipo I/O.
- El Puerto 1 es del mismo tipo que el cero, solo que con asignación de pines del 1 al 8.
- Para el Puerto 2 se tienen asignadas los pines del 21 al 28, y son de I/O.
- Para el Puerto 3 se tiene la distribución que indica la siguiente tabla:

No. de pin	Pin en puerto	Función
10	P3.0	RDX (entrada del puerto serie)
11	P3.1	TXD (salida del puerto serie)
12	P3.2	INT0' (línea para interrupción 0)
13	P3.3	INT1' (línea para interrupción 1)
14	P3.4	TO (entrada para el TIMER 0)
15	P3.5	T1 (entrada para el TIMER 1)
16	P3.6	WR' (señal de escritura para RAM externa)
17	P3.7	RD' (señal de lectura para RAM externa)

Tabla A.5

- RST es la señal para RESET del microcontrolador y esta asignada al pin 9.
- ALE (Adress Latch Enable), es la señal que emite el microcontrolador, hacia el Latch que hará la separación en los puertos de las direcciones y los datos. Esta asignada al pin 30.
- PSEN' (Program Store Enable) es la señal que provee el microcontrolador para la lectura del programa en memoria externa. Es el pin 29.

- EA' (External Access). Sirve para indicar al microcontrolador donde debe buscar el programa (en memoria interna o externa); si se conecta a Vcc indica que el programa esta en memoria interna, si se conecta a Vss indica que la ejecución del programa será externa. Se encuentra en el pin 31.

- XTAL1 es la entrada hacia el oscilador del microcontrolador.

- XTAL2 es la salida del oscilador.

Para proveer la señal de reloj al microcontrolador, se puede hacer de dos maneras, una es con una señal externa de oscilación conectada a XTAL2 y conectando XTAL1 a tierra. Otra forma es con un cristal, conectando las terminales de este a XTAL1 y XTAL2 y dos capacitores de los pines XTAL1 y XTAL2 a tierra.

EL PPI 8255A

El circuito integrado 8255 es un multiplexor de tres puertos a uno; cada puerto es de 8 líneas de datos y/o señales. Este circuito es de gran ayuda en las implementaciones digitales que requieren de la manipulación y control de una cantidad considerable de datos y señales de entrada y salida. Para el caso de una intercomunicación con el puerto paralelo, se requiere del manejo de 8 líneas de datos y 8 señales de control, de las cuales 5 se manejan como señales de salida y tres como señales de entrada.

ARQUITECTURA BASICA

En el 8255 se tienen 24 líneas de entrada/salida, distribuidos en tres puertos definidos como puerto A, B y C de 8 líneas cada uno. Los puertos A y B se pueden utilizar como entrada o salida; en tanto que el puerto C se puede utilizar en tres formas diferentes: A) Como un puerto de ocho bits de entrada o salida, B) Como dos puertos independientes²² de 4 bits cada uno, y C) Para proveer las señales de protocolo estándar de comunicación en paralelo² para los puertos A y B.

El 8255 cuenta con buses internos de datos y direcciones, para acceder a cualquiera de los tres puertos. Las señales de control para

²² Esta opción nos serviría para el caso en que se utilizaran los puertos A y B para recepción de datos, donde se necesitaría un puerto para las señales de protocolo de cada puerto, y con la división del puerto C se tienen esos dos puertos.

² Llamaremos protocolo de comunicación paralelo a las señales de control y sincronización que se sensan en la transmisión paralelo de información. Esta serie de señales es conocida en el medio del diseño electrónico como "HANDSHAKE". De ahora en adelante nos referiremos a ello simplemente como protocolo.

los buses del 8255 permiten leer o escribir datos en cualquier puerto, o controlar y leer el registro de estado. Las líneas de direccionamiento A0 y A1 permiten el acceso exclusivo a uno de los tres puertos o al registro de control. Las direcciones internas para cada puerto son: PUERTO A: 00, PUERTO B: 01, PUERTO C: 10, REGISTRO DE CONTROL: 11. La señal de CS' en bajo habilita el 8255. Cuando se activa la línea de RESET todas la líneas de los puertos son inicializadas como líneas de entrada, para prevenir posibles daños a los dispositivos que estén conectados a cada uno de los puertos. La figura B.1 muestra el diagrama de bloques interno del 8255A.

El 8255 puede ser inicializado en tres modos diferentes dependiendo de la aplicación específica para la cual se aplique:

MODOS DE OPERACION

a) MODO 0:

Se emplea para cuando el puerto se utilizara para entrada o salida. Si los puertos A y B son inicializados en Modo 0, entonces el puerto C puede ser utilizado como un solo puerto de 8 bits o como dos puertos independientes de 4 bits cada uno. Se dice que los dos puertos en los que se divide el puerto C son independientes porque, mientras uno puede ser inicializado como entrada el otro podría ser inicializado como salida.

b) MODO 1:

Este modo se emplea cuando los puertos A y B manejaran un protocolo de comunicación para su operación, y funcionan como entrada o salida. La parte baja del puerto C funcionara como receptor/emisor de la señales de protocolo para puerto B. Si el puerto A es inicializado como

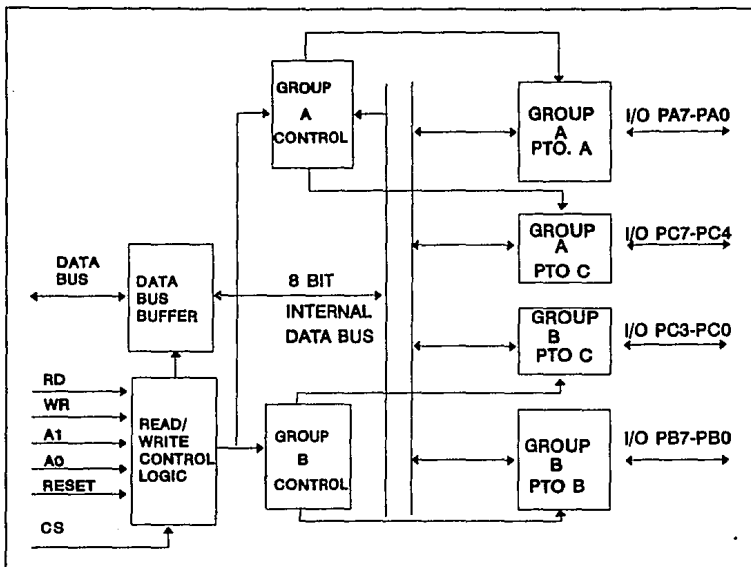


Fig. B.1

entrada, la parte alta del puerto C podría funcionar como receptor/emisor de las señales de protocolo para este puerto.

c) MODO 2:

En este modo el puerto A puede ser usado para establecer un protocolo de transferencia de datos bidireccional. El 8255 en este modo puede ayudar a extender el bus del sistema. Si el puerto A es inicializado en este modo, los pines PC3 y PC7 pueden ser usados como líneas de

protocolo para al puerto A. Tres pines del puerto C se pueden usar como líneas de entrada/salida si C esta inicializado en modo 0. Los tres pines restantes se podrían usar para las señales de protocolo del puerto B, si este esta en modo 1.

PALABRA DE CONTROL DEL 8255A

A continuación se enuncian la posición y el uso de cada uno de los bits de la palabra de control para programar a los puertos del 8255A.

Grupo B:

D₀: Parte baja del puerto C (1 = entrada 0 = salida)

D₁: Puerto B (1 = entrada 0 = salida)

D₂: Selección de modo (0 = modo 0, 1 = modo 1)

Grupo A:

D₃: Parte alta del puerto C (1 = entrada, 0 = salida)

D₄: Puerto A (1 = entrada, 0 = salida)

D_{5,6}: Selección de modo (00 = modo 0, 01 = modo 1, 1x = modo 2)

D₇: Bit de programación (1 = activo)



BIBLIOGRAFIA

BIBLIOGRAFIA

Bibliografía

Introducción al Microprocesador 8086/8088; Morgan, Christopher L.; Mc. Graw Hill, México 1985; 339 págs.

La Escritura Braille en el mundo: estudio de los esfuerzos realizados de la uniformidad de la notación Braille; Mackenzie, Sir Clutha Nantes; París:UNESCO; 1985, 181 págs.

Microprocessors and Interfacing Programming and Hardware; Hall, Douglas V.; Edit. Mc. Graw Hill; 1988; 554 págs.

Organización de computadoras: un enfoque estructurado; Tanenbaum, Andrew S.; 3a. Ed. ; Edit. Prentice Hall; México; 1992; 658 págs.

Lógica Digital y Diseño de Computadores; Mano, Morris M.; Edit. Prentice Hall; México; 1982; 636 págs.

The TTL Data Book for Design Engineers; 2a. Ed. ; Texas Instruments; USA; 1981

8 Bit Embedded Controllers; Intel; USA; 1990.

ECG Semiconductors Master Replacement Guide; Edit. Phillips ECG Inc.; USA; 1985; 654 págs.

BIBLIOGRAFIA

Logic Databook Volume II; National Semiconductor Corporation; 1984.

Principios de microprocesadores; Guilmore Charles M.; Edit. Limusa; México 1989; 297 págs.

Fujitsu: User's Manual and Programmer's Manual DL3600.