

30  
2ej.



# UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE INGENIERIA

## DISEÑO Y CONSTRUCCION DE UN SISTEMA DE GRABACION DIGITAL DE AUDIO

### T E S I S

QUE PARA OBTENER EL TITULO DE:  
INGENIERO MECANICO ELECTRICISTA  
P R E S E N T A N :  
RUBEN ALFONSO CARRILLO ZAZUETA  
BERTHA MEZA NUÑEZ  
YADIRA VERGARA BAUTISTA

DIRECTOR DE TESIS:  
M.I. JUAN CARLOS ROA BEIZA

MEXICO, D. F.

1992

**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.

# INDICE

PAGINA

I	MARCO TEORICO.....	1
I.1	CODIGOS DE TRANSMISION Y RECEPCION DE INFORMACION DIGITAL.....	2
I.2	TIPOS DE FLOPPIES.....	11
I.3	TECNICAS Y PROTOCOLOS DE GRABACION. EN FLOPPIES.....	19
I.4	CIRCUITOS DE CONVERSION A/D.....	27
I.5	CIRCUITOS DE CONVERSION D/A.....	37
II	ACCESO Y CONTROL DE INFORMACION.....	44
II.1	ACCESO DE INFORMACION POR MEDIO DE UN MPU.....	45
II.2	TECLADOS. SELECCION.....	52
II.3	DESARROLLO DEL SOFTWARE.....	69
III	DISENO Y CONSTRUCCION.....	105
III.1	SISTEMAS DE VISUALIZACION.....	106
III.2	DISENO DEL HARDWARE PARA EL CONTROL Y VISUALIZACION.....	113
III.3	DISENO DE LAS INTERFACES.....	122
III.4	DISENO DEL AMPLIFICADOR DE AUDIO.....	125
III.5	BLINDAJE DE SEÑALES.....	132

CONCLUSIONES.....	136
APENDICES.....	138
A. HOJAS DE ESPECIFICACIONES.....	139
B. NORMAS PARA TARJETAS DE INTERFACE EN UNA PC-XT.....	147
C. FUNCIONAMIENTO DEL SISTEMA.....	151
D. CALCULOS DEL AMPLIFICADOR DE POTENCIA.....	153
E. ALGORITMO DE COMPACTACION DIGITAL.....	156
GLOSARIO.....	159
BIBLIOGRAFIA.....	161

# CAPITULO I

## MARCO TEORICO

## II CODIGOS DE TRANSMISION Y RECEPCION DE INFORMACION DIGITAL

Hoy en día es muy frecuente transmitir información de forma digital, ya que de esta manera la señal será más inmune al ruido. En caso de que la señal sea analógica se recurre a la utilización de convertidores tanto analógicos/digitales como digital/analógicos. Estas cantidades físicas llamadas señales, cuando son procesadas digitalmente, se convierten en elementos discretos de información que pueden ser impulsos eléctricos, los dígitos decimales, las letras de un alfabeto o cualquier otro conjunto de símbolos significativos.

Debido a la restricción física de los componentes electrónicos, a la baja confiabilidad de los circuitos al manejar muchos valores y que la lógica humana frecuentemente tiende a ser binaria los sistemas digitales manejan valores binarios.

### CODIGOS BINARIOS

Los sistemas digitales electrónicos usan señales que tienen dos valores distintos y elementos de circuitos que tienen dos estados estables. Existe así una analogía directa entre estas señales, los elementos de circuito y los dígitos, por lo que cualquier elemento discreto de información específico entre un grupo de cantidades puede ser representado por un código de dos estados, dentro de este código se define un bit como un dígito binario.

Cuando se trabaja con un código de este tipo es más fácil pensar que denota una cantidad binaria igual a "0" ó "1".

Para representar un grupo de  $2^n$  elementos diferentes en este código se requiere un mínimo de  $n$  bits, ya que es posible arreglar  $n$  bits en  $2^n$  maneras diferentes. Las diferentes combinaciones en bits de un código de  $n$  bits puede encontrarse contando en forma binaria desde 0 hasta  $2^n - 1$ , cabe señalar que

algunas combinaciones de bits no se asignan cuando el número de elementos de un grupo que va a codificarse no es múltiplo de una potencia de 2. Aunque el número de bits, necesarios para codificar  $2^n$  cantidades diferentes, es  $n$ , no hay un número máximo de bits que puedan ser usados por un código binario.

Un grupo de cuatro cantidades diferentes puede ser representado por un código binario de cuatro bits con cantidad asignada a cada una de las siguientes combinaciones de bits:

primera cantidad	00
segunda cantidad	01
tercera cantidad	10
cuarta cantidad	11

Si se trata de un grupo de ocho elementos se requiere un código de ocho bits:

1a.- 000	5a.- 100
2a.- 001	6a.- 101
3a.- 010	7a.- 110
4a.- 011	8a.- 111

Sin embargo, muchas veces para simplificar operaciones de sustracción y manipulaciones lógicas se usan los complementos de números binarios. De estos existen dos tipos, COMPLEMENTO A DOS y COMPLEMENTO A UNO, el primero de estos se define de la siguiente manera:

Dado un número positivo  $N$  con parte entera de  $n$  dígitos su complemento sería  $2^n - N$  para  $N$  diferente de cero y cero para  $N=0$ . Por ejemplo para el número (101100)<sub>2</sub> su complemento es:

$$2^6 - (101100) = (1000000) - (101100) = 010100$$

Otra manera de obtener el complemento de un número es dejar todos los ceros menos significativos y el primer dígito diferente de cero sin cambio, para luego reemplazar unos por ceros y ceros por unos en el resto de signos más significativos.

En el complemento a uno la definición dice que dado un número positivo  $N$  con parte entera de  $n$  dígitos y una parte fraccionaria de  $m$  dígitos, el complemento se obtiene como  $2^n - 2^{-m} - N$ . Por ejemplo para el número binario (0.0110) su complemento es:

$$2^0 - 2^{-4} - (1.0110) = (0.1111 - 0.0110) = 0.1001$$

Otra forma más sencilla de obtener el complemento a uno de cualquier número se obtiene cambiando los ceros por unos y los unos por ceros.

Por otro lado también el signo existe en los números binarios. El signo es una cantidad discreta de dos valores: más o menos. Y la convención aquí es representar un más con un cero y un menos con un uno. Cuando un número binario es positivo, el signo se representa con un cero y la magnitud por un número binario positivo. Cuando el número es negativo, el signo se representa con un uno y el resto del número puede ser representado de las tres maneras siguientes:

- 1.- Signo--Magnitud
- 2.- Signo--Complemento a uno
- 3.- Signo--Complemento a dos

Cabe señalar que para un número positivo las tres representaciones son iguales. Todo esto se puede visualizar mucho mejor con el siguiente ejemplo:

	+	9	-	9
Signo--Magnitud	0	001001	1	001001
Signo--Complemento a 1	0	001001	1	110110
Signo--Complemento a 2	0	001001	1	110111

A esto se le conoce con el nombre de MAGNITUD SIGNADA.

## CODIGOS DECIMALES

Los códigos binarios para dígitos decimales requieren un mínimo de cuatro bits. Se pueden obtener numerosos códigos diferentes arreglando cuatro o más bits en diez posibles combinaciones. Así los bits obtenidos de la codificación son combinaciones de unos o ceros arreglados de acuerdo a las reglas del código usado.

### CODIGO BCD

Del tipo de códigos decimales tenemos al BCD (Binario Decimal Codificado) que es una forma directa de asignar a un decimal un equivalente binario. Dentro de éste código es posible asignar cargas a los bits binarios de acuerdo a sus posiciones y éstas son: 8, 4, 2, 1:

Número decimal	Código BCD
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
10	1010

## CÓDIGOS DE DETECCIÓN DE ERROR

La información binaria, siendo señales de pulsos modulados o señales de entrada y salida de un sistema digital, puede ser transmitida a través de algún medio de comunicación tal como ondas de radio o alambres. Cualquier ruido externo introducido en el medio de comunicación física cambia los valores de los bits de "0" a "1" y viceversa. Con el objeto de detectar los errores durante la transmisión se han creado los códigos de detección de error, lo que no implica que con este tipo de códigos se puedan corregir directamente los errores. Sin embargo, el procedimiento usual es observar la frecuencia del error, si este ocurre de vez en cuando, aleatoriamente y sin ningún efecto pronunciado sobre el total de la información transmitida puede no hacerse nada, pero si por el contrario el error distorsiona el significado de la información recibida, se prosigue a detectar la falla del sistema. En ocasiones se agrega un bit de paridad, es decir se incluye un bit extra con el mensaje para convertir el número total de unos en impar o par, y de esta forma detectar un error. Esto depende del código que se use para determinar así un error. Por ejemplo para determinado mensaje de 4 bits se puede agregar un bit de protección que será un cero si en el mensaje hay un número impar de ceros:

Mensaje	Bit de protección
0000	1
0001	0
0010	0
0011	1
0100	0
0101	1
0110	1
0111	0
1000	0
1001	1

## CODIGO ASCII

El código ASCII (American Standard Code for Information Interchange) consta de siete bits a los que se le suma un octavo de paridad. Puesto que los valores asignados a las letras del alfabeto, y a los restantes caracteres, siguen una secuencia binaria continua, los sistemas digitales no tienen que dejar su propio lenguaje binario para realizar operaciones secuenciales con estos caracteres. Además, debido al agrupamiento de sus funciones de control, con sólo analizar los dos primeros bits de una combinación codificada el sistema puede determinar si se trata de una función de control (dos ceros) o de un caracter (uno de los dos no es cero).

Existen 32 caracteres de control incluidos en el conjunto de caracteres. De los cuales existen cuatro clases para caracteres de control y un número de caracteres individuales. Las cuatro clasificaciones de control genéricas de caracteres son:

- \* Control de transmisión: que controlan el flujo de datos a través de la línea.

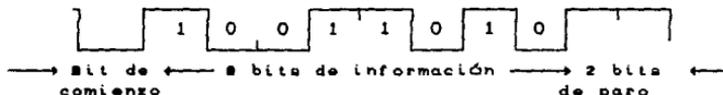
- \* Formato: usados para controlar el esquema físico de la información en la página impresa o en la pantalla de visualización.

- \* Control de dispositivos: utilizado primordialmente para controlar los dispositivos auxiliares en otros lugares de recepción.

- \* Separadores de información: utilizados para delimitar lógicamente los elementos de cada dato.

Los bits del 1 al 7 contienen la información que se desea transmitir, y el octavo bit es el de paridad. Su propósito es el de ayudarnos a contar con un elemento que nos permita detectar los errores de transmisión.

La transmisión del código ASCII está basada en el principio de "arranque-detección". Estas dos señales se denotan por un bit de comienzo y dos de paro cuando la transmisión se está llevando a cabo en forma serie. En forma automática el sistema gobierna la longitud de la conducción de paro. Cuando la transmisión se realiza en forma sincrónica, esto es cuando un sistema transmite a otro en forma continua con tiempos regulares (sin variaciones), se obtiene su más eficiente utilización y no son necesarios los bits de "arranque-detección"



En la transmisión en paralelo, todos los bits de un carácter codificado son transmitidos simultáneamente, lo cual significa que cada nivel de código tiene un único canal dedicado a él. Así para caracteres en ASCII se requerirá de ocho canales. Con esto se logra una alta densidad de información enviada en la transferencia de datos, pero el problema estriba en el alto costo de proveer un canal por cada nivel de codificación.



TABLA DEL CODIGO ASCII

					0	0	0	0	1	1	1	1	
					0	0	1	1	0	0	1	1	
					0	1	0	1	0	1	0	1	
					0	1	2	3	4	5	6	7	
b i t s	b4	b3	b2	b1	COL FILA								
	0	0	0	0	0	NUL	DLE	SP	0	*	P	'	P
	0	0	0	1	1	SOH	DC1	!	1	A	Q	a	q
	0	0	1	0	2	STX	DC2	"	2	B	R	b	r
	0	0	1	1	3	ETX	DC3	#	3	C	S	c	s
	0	1	0	0	4	EOT	DC4	\$	4	D	T	d	t
	0	1	0	1	5	ENQ	NAK	%	5	E	U	e	u
	0	1	1	0	6	ACK	SYN	&	6	F	V	f	v
	0	1	1	1	7	BEL	ETB	'	7	G	W	g	w
	1	0	0	0	8	BS	CAN	)	8	H	X	h	x
	1	0	0	1	9	HT	EM	(	9	I	Y	i	y
	1	0	1	0	10	LF	SUB	*	:	J	Z	j	z
	1	0	1	1	11	VT	ESC	+	;	K	[	k	{
	1	1	0	0	12	FF	FS	,	<	L	\	l	&
	1	1	0	1	13	CR	CS	-	=	M	]	m	}
	1	1	1	0	14	SD	RS	>	~	N	^	n	~
1	1	1	1	15	SI	US	/	?	O	_	o	DEL	

## 1.2 TIPOS DE FLOPPIES

Debido a la necesidad de memorias de gran capacidad, no volátiles, modificables y de bajo costo de los sistemas computacionales, fue necesaria la creación de diversos medios de almacenamiento masivo de información. Los más utilizados hoy en día son: las cintas magnéticas, los discos flexibles (floppies), los discos duros y los discos ópticos.

Dado que el tiempo de acceso a la cinta magnética es muy grande y resulta poco eficaz, las cintas magnéticas sólo son utilizadas para respaldo de información; así, los medios con los que contamos para el almacenamiento masivo de información se reduce a los tres tipos de almacenamiento en disco.

Para nuestro propósito, solamente nos abocaremos a la tecnología de floppies.

### FLOPPIES

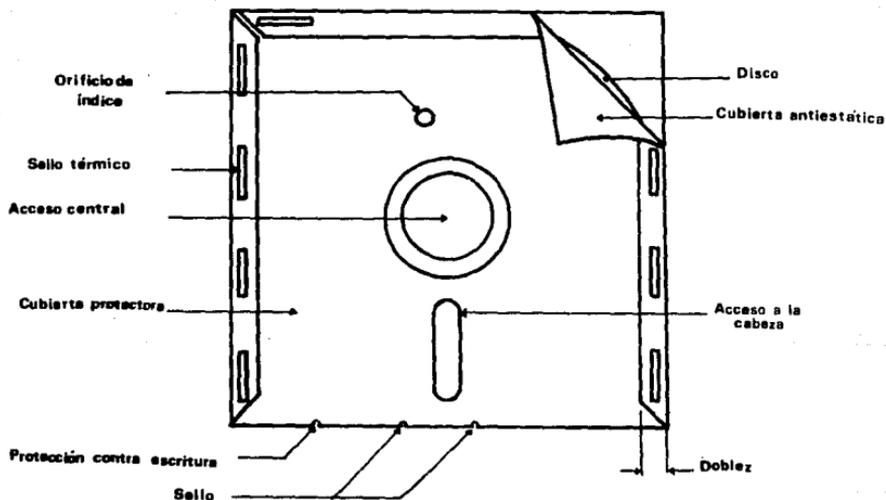
Se le denomina floppy a un disco de Mylar cubierto con una capa de óxido magnético, en el que la información es grabada en forma de unos y ceros representados mediante la magnetización de las partículas de óxido en direcciones opuestas.

Los floppies en general, vienen encerrados en una cubierta plástica que los protege del polvo y otros contaminantes que pudieran dañar la información ahí almacenada. La cubierta, sin embargo, no protege todo el disco debido a que requiere de algunas ranuras necesarias para su correcto funcionamiento pero que permiten la entrada de cierta cantidad de polvo al interior.

Las ranuras que presentan en general son: una que permite el contacto de la cabeza de lectura-escritura con la superficie del disco; un orificio de índice que proporciona información temporizadora para sincronía, su paso es detectado mediante un circuito fotoeléctrico que localiza el primer sector del disco; un

orificio central que sirve para hacerlo girar mientras la cubierta permanece inmóvil; una muesca lateral sobre la cubierta que sirve como una protección mecánica contra escritura, esta también es detectada mediante un circuito fotoeléctrico.

En la figura se muestra un esquema de la estructura de un floppy de los más utilizados.



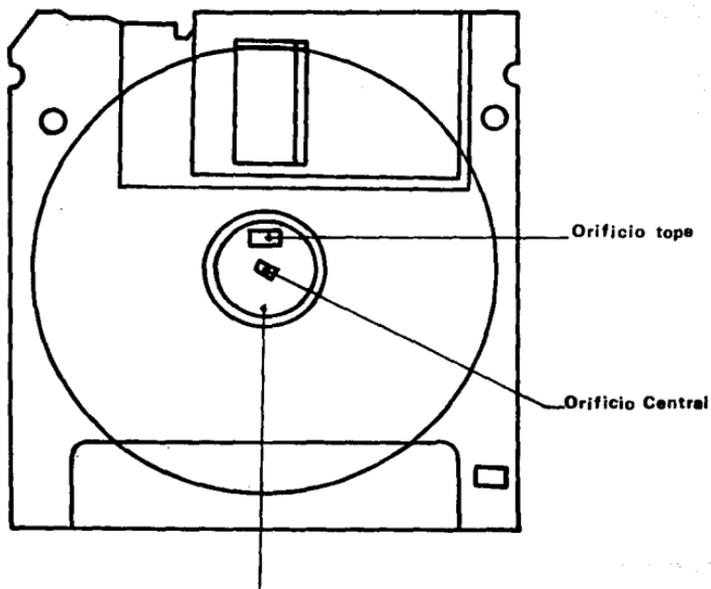
FLOPPY DE 5 1/4"

Hace 22 años, IBM introdujo este tipo de tecnología por medio de su disco flexible de 8", con una cubierta similar a la de las cintas magnéticas. Debido a los problemas de contacto directo con dicha superficie, IBM los empezó a producir con una cubierta plástica que los protegiera.

Para el año de 1976, Shugart Associates introdujo en el mercado un nuevo tamaño de disco flexible denominado mini-floppy o diskette de 5<sup>1</sup>/<sub>4</sub>". Originalmente, este tipo de diskettes sólo podían almacenar 50 Bytes de información por 40 pistas (tracks), lo cual sin duda era bastante restringido esto a venido a solucionarse mediante nuevas técnicas de grabación en doble, triple y cuádruple densidad que logran aumentar la capacidad de grabación del floppy, además de mejoras en la precisión del mecanismo del drive.

Con la introducción al mercado de computadoras de tamaño reducido, el tamaño de los diskettes también ha disminuido. Existen discos de alrededor de 4" y aún hasta de 3<sup>1</sup>/<sub>2</sub>" y en algunas pruebas hasta de 2" de diámetro. A diskettes de este tamaño también se les conoce como micro-floppies. El diskette de 3" viene empaquetado en plástico rígido de 3.2" por 4" con un espesor de 0.2", su drive está estandarizado de la misma forma que los de 5<sup>1</sup>/<sub>4</sub>" (300 rpm, 100 pistas/lado, 100 pistas/plg y 250 KBits/seg) utiliza la técnicas de grabación FM y MFM, logrando hasta 500 KBytes utilizando doble densidad y doble lado.

El diskette de 3.5" está empaquetado en un plástico rígido de 3.54" por 3.7" con un espesor de 0.133". Existen algunas variantes en cuanto a sus especificaciones, dependiendo del fabricante. En la figura se muestra un diagrama esquemático de este tipo de diskette.



Acceso Central

FLOPPY 3.5"

El diskette del sistema Tabor y Seagate de 3<sup>1</sup>/<sub>4</sub>" es casi desconocido y utiliza 3.27" por 3.43" y logra 140 pistas/pul con una velocidad de rotación de 300 rpm. Puede alojar hasta 500 KBytes en un sólo lado. Es compatible a los drives estandar de 5<sup>1</sup>/<sub>4</sub>" y utiliza un empaquetado de plástico flexible .

## FLOPPIES DE MUY ALTA DENSIDAD DE GRABACION

Se define como floppy de muy alta densidad de grabación (VHD Floppy Disk) a aquel tipo de floppy capaz de almacenar al menos 5 MBytes de datos en un diámetro de 3.5" o menos.

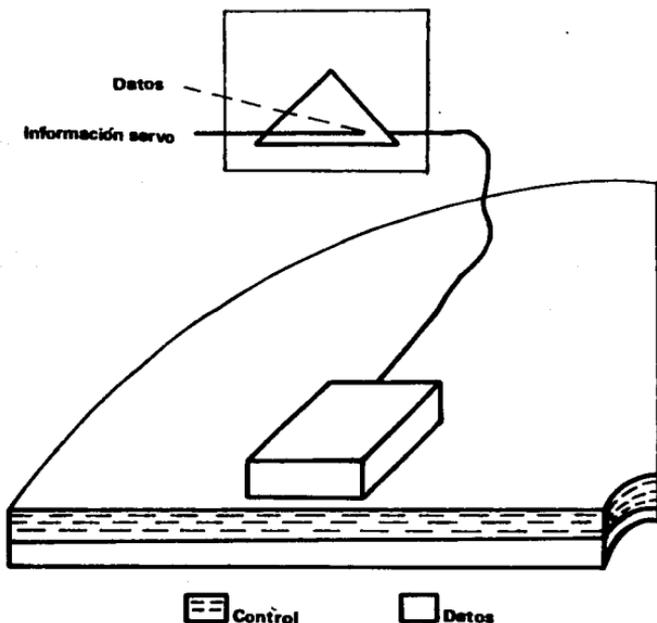
Una gran parte del trabajo de lograr diskettes con esta capacidad de grabación se debe a las técnicas de grabación utilizadas y a la construcción misma de los drives. Sin embargo, también es necesaria la modificación de la construcción misma del floppy en muchos de los casos.

Los diskettes estandar utilizan un recubrimiento de Cobalto modificada con partículas de óxido de hierro. En los floppies de muy alta densidad se utiliza un recubrimiento de Bario-Ferrita, la cual es muy delgada y de mayor coercitividad.

Toshiba propone un esquema en el que se utiliza una nueva cabeza y un controlador modificado. Los floppies a utilizar deben ser pre-procesados y deben de ser formateados en el mismo drive donde serán utilizados para lectura/escritura. Esta tecnología utiliza técnicas de grabación vertical, y sólo puede almacenar 2.88 MBytes de información.

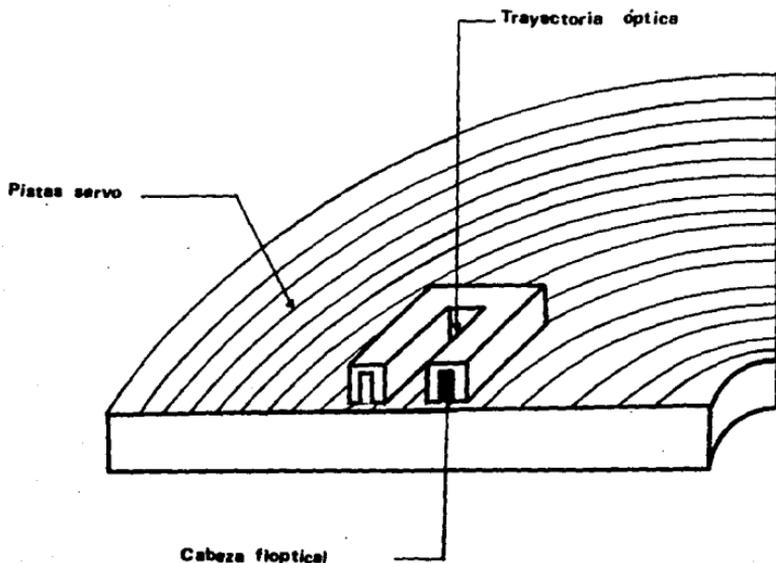
Desde hace algún tiempo NEC trabaja en un modelo de floppy de muy alta densidad utilizando el mismo medio de Bario-Ferrita de los floppies de 3.5". Utiliza 431 pistas/pul y aumenta la densidad de grabación un poco. Requiere de "marcas" para el posicionamiento servo entre los sectores de datos, con la finalidad de que la cabeza se sitúe exactamente sobre éstos. Para lograr que la grabación de dichas marcas sea tan precisa como se requiere, es necesario que dicha operación sea realizada en la fábrica. El sistema NEC puede almacenar hasta 12 MBytes de información. Las marcas son hechas independientes en un lado y en otro, de tal forma que la lectura/escritura en cada lado puede lograrse aunque las cabezas no esten alineadas entre sí. Una desventaja de este sistema es que si la información servo es borrada, el disco se vuelve inservible para el sistema.

De igual forma que el sistema NEC, Brier desarrolló un sistema en que pueden almacenarse hasta 21.4 MBytes de información. Utiliza patrones pregrabados en fábrica utilizando altas corrientes. El límite teórico para este sistema puede considerarse 100 MBytes, con una densidad de 6000 pistas/pul. El sistema Brier actualmente utiliza 555 pistas/lado. La información servo se graba a una frecuencia distinta de los datos con la finalidad de evitar interferencia. Para la extracción de los datos grabados sobre la información servo, dado que están grabados a diferentes frecuencias, un prisma electromagnético las separa. El controlador que logra esto está incluido en el drive. En lugar de la interfaz estandar, el sistema Brier utiliza una del tipo SCSI. El sistema Brier hace girar al diskette a 720 rpm y transfiere datos a 288 KBytes/seg.



SISTEMA BRIER

InSite produjo una nueva tecnología denominada "Floptical" que utiliza información servo de tipo óptico y grabación de tipo magnética para incrementar la densidad de las pistas. Utiliza floppies estándar de 5.25 pulgadas de 4 MBytes. La grabación de la información servo se realiza mediante un láser de forma similar a como se hace en los Compact Disc. Las cabezas de lectura/escritura están acopladas a un sensor óptico. Cuando las cabezas se aproximan a la posición deseada, el posicionamiento final se ajusta por medio de sensores ópticos. Las cabezas magnéticas sólo se ocupan entonces de la lectura/escritura de datos. Este sistema utiliza un único patrón servo sobre la superficie del disco para controlar la posición de ambas cabezas. Además utiliza un código de corrección de error con chequeo cíclico de redundancia (CRC). El propio drive puede detectar automáticamente un sector dañado y asignar uno nuevo.



#### SISTEMA FLOPTICAL

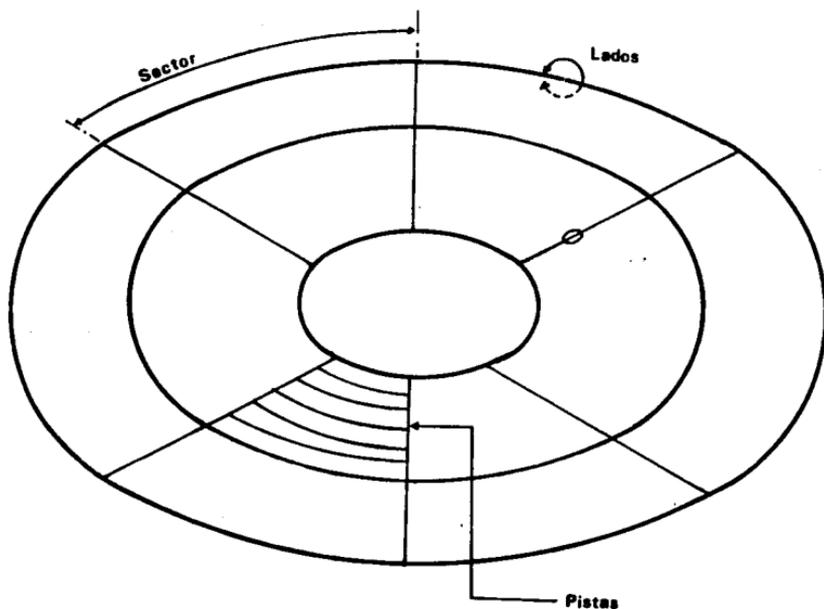
El drive de InSite usa un controlador SCSI con lo cual ni el BIOS ni el Sistema Operativo ni las aplicaciones tienen que cambiar para su utilización. La mayor desventaja de este sistema es que es mucho más caro que los demás tipos de drives.

Para lograr los objetivos de este trabajo, es de suma importancia que pueda almacenarse la mayor cantidad de información posible en el floppy, debido a que como se sabe, la señal de audio requiere de un muestreo de al menos 40 KHz, de tal manera que se cumpla el teorema de muestreo. Así, si se utiliza un convertidor de 8 bits, se tendrán 40 KBytes/seg de información por grabar, lo cual es demasiado para lograr grabar la señal de audio por el suficiente tiempo (utilizando una grabación directa de 8 bits en un floppy de 5 <sup>1</sup>/<sub>4</sub>" de 360 KBytes, podrían almacenarse solo 9.22 seg de audio). Así, el factor de más importancia para seleccionar el tipo de floppy es su densidad de grabación. Atendiendo a este factor, se observa que la mejor opción al respecto es la presentada por InSite, es decir la tecnología Floptical con la cual podrían almacenarse hasta 25 MBytes de información. ( 10 min, 55.3 seg de audio utilizando 8 bits por dato). Sin embargo, la limitante para poder utilizar esta tecnología para el desarrollo de nuestro prototipo del proyecto es que adquirir un drive de tipo floptical representaría un increíble incremento en el costo del prototipo de prueba. Por lo tanto, con el fin de realizar el prototipo, se decidió utilizar un drive convencional de 5 <sup>1</sup>/<sub>4</sub>" y 360 KBytes de capacidad, dejando claro que si en lugar de este se utilizara uno del tipo floptical, el sistema sería optimizado por mucho. Cabe mencionar que si se utilizan técnicas de compactación de la información digital, pueden lograrse mayores capacidades de almacenamiento de la información que grabando directamente dicha información.

### 1.3 TÉCNICAS Y PROTOCOLOS DE GRABACION DE FLOPPIES

Antes de hacer referencia sobre las técnicas de grabación convendría hacer mención de la estructura y capacidad de un disco flexible.

Los datos en un disco se guardan en una serie de divisiones concéntricas llamadas pistas (tracks). Cada una de estas pistas a su vez se dividen en segmentos más pequeños, llamados sectores. La cantidad de datos que pueden almacenarse en el disco dependerá del número de pistas así como del tamaño de los sectores. La siguiente figura mostrará dicha distribución.



ORGANIZACION DE UN DISCO

La capacidad total de un disco dependerá del tipo de drive así como del Sistema Operativo empleado, pero no de su estructura, que es la misma en todos los casos.

La localización de cada pista y el número de lados utilizables dependerá del hardware y de la construcción del disco. Un diskette puede ser clasificado como de "sector duro" o "sector suave". Los de sector duro tienen un orificio al inicio de cada sector para proveer de la información de tiempo al drive, y tiene además un orificio de índice para información de posición, de tal manera que cada sector tiene un tamaño fijo definido por los orificios. Los de sector suave sólo cuentan con el orificio de índice, y toda la localización de los sectores esta definida por el sistema controlador y el software. Así, los discos de sector suave tienen una mayor flexibilidad en el almacenamiento de la información, pero tienen menor capacidad que los de sector duro, esto es debido a que la información referente al tamaño de cada sector debe ser grabada junto con el sector mismo, lo cual ocupa parte de la capacidad total del disco.

Para lograr un esquema de grabación de alta densidad, el factor más importante es el método utilizado de grabación de datos.

El principio más simple de grabación directa en un medio magnético es el de la aplicación de pulsos a la cabeza de grabación la cual produce un cambio de flujo magnético originando una pequeña área magnetizada sobre la superficie del disco que se puede considerar como un "1" lógico. La ausencia de estos pulsos se considera como un "0" lógico. Para la lectura de información la cabeza se mueve sobre la superficie del disco el área magnetizada induce una corriente en la cabeza la cual es amplificada como un pulso. Existen varios métodos basados en este principio que son utilizados para la grabación magnética.

## GRABACION FM

El método más simple de grabación de datos es denominado grabación FM. Este método de grabación emplea celdas de bit para definir los bits de datos. Las fronteras entre las celdas de bit vienen determinadas por el flanco de un pulso de reloj. Si dentro de esta celda existe un cambio de flujo se considera como un "1" lógico y si no lo hay, como un "0" lógico.

Se le llama FM porque modula en frecuencia la información binaria, asignando dos pulsos para un "1" y un solo pulso para un "0".

Este método de grabación es confiable debido a que cada bit está acotado por el pulso de reloj. El problema que presenta es que los pulsos de reloj deben ser separados de la información al ser leídos, lo que consume tiempo y espacio en el disco.

En la búsqueda de mayor capacidad de grabación se encontró que se podrían almacenar muchos más datos si se pudieran eliminar los pulsos de reloj. El esquema podría utilizar un pulso a la mitad de la celda como un "1" y ninguna condición como un "0". El inconveniente de esto sería la pérdida de sincronía al presentarse largas cadenas de ceros con lo cual el sistema sería incapaz de distinguir las celdas subsiguientes.

## GRABACION MFM

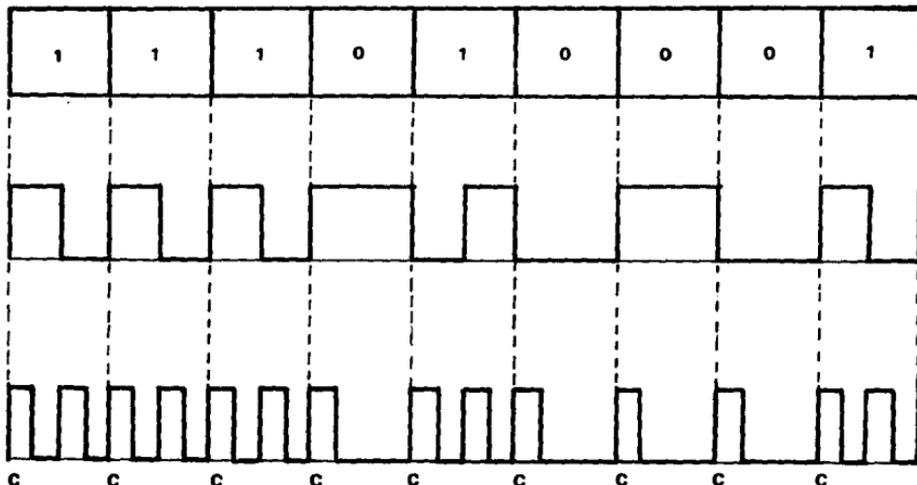
El método de grabación MFM (FM mejorado) es un esquema que resuelve la pérdida de sincronía utilizando el flanco positivo del pulso de reloj, como en la grabación FM, a partir del segundo cero adyacente y los demás ceros subsiguientes; esto elimina todos los pulsos de reloj para los unos y ceros acotados por unos debido a que los unos son una señal que contiene al reloj. En la figura se ilustra como 9 bits de datos requieren de 14 pulsos y 14 cambios

de flujo para el método FM y solamente 7 pulsos y 7 cambios de flujo para el método MFM, de tal manera que pueden ser grabados el doble de datos en la misma distancia lineal en una pista. Esto tiene como resultado una doble densidad de grabación en el mismo espacio.

Existe una mejora para el esquema de grabación MFM en el que pone un flanco de reloj solamente si la celda de bit anterior contiene un "1" o una señal de reloj, y si en la celda de bit presente hay un "0". Esto elimina los pulsos de reloj para cada cero adyacente y sólo se necesitan 6 pulsos y 6 cambios de flujo para grabar los 9 bits de datos, aumentandose así la capacidad de grabación.

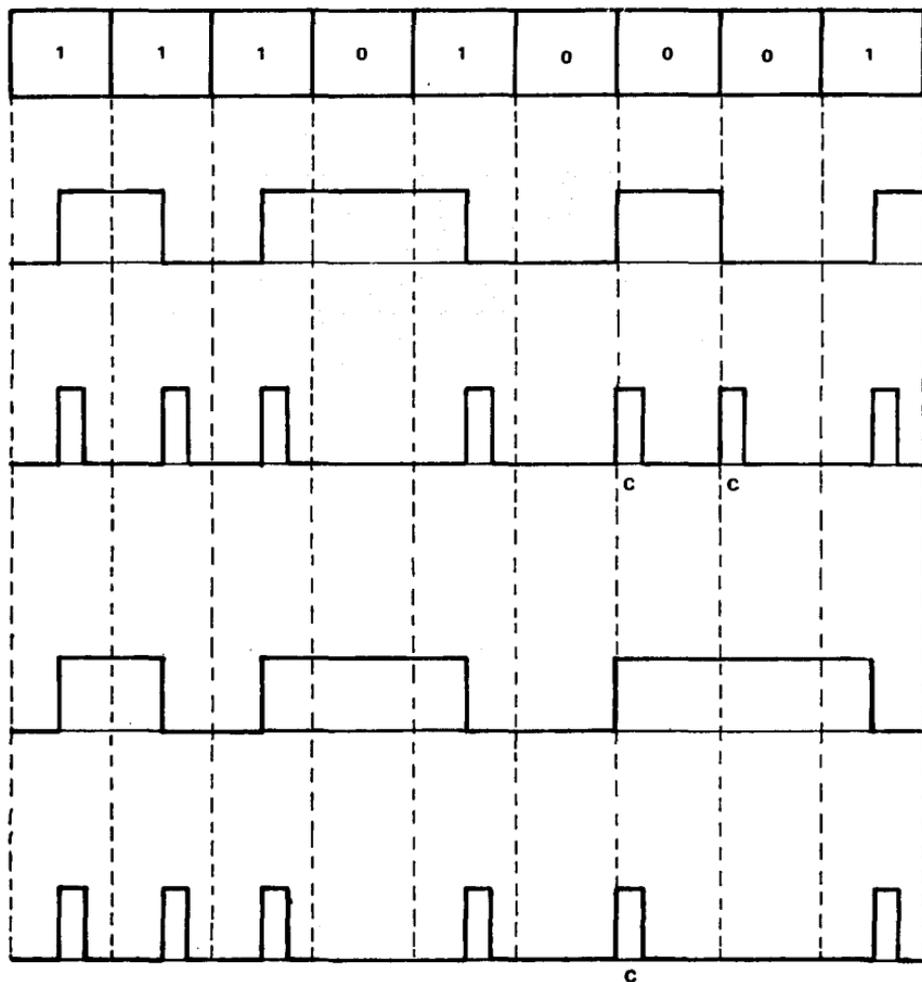
Para lograr lo anterior es necesario que la tarjeta controladora tenga la circuitería necesaria para la separación de los datos y los pulsos de reloj además de la circuitería para la separación de los datos de los pulsos de reloj antes de poder leerlos. El circuito electrónico más utilizado para realizar dicha separación es un oscilador de fase encadenada (PLO).

Espacios de bit



Grabación FM

Espacios de bit



Grabación MFM

## GRABACION PERPENDICULAR

El método de grabación perpendicular o vertical consiste en mantener la magnetización perpendicular a la superficie del disco en lugar de hacerlo a lo largo de ésta como sucede en la grabación longitudinal, lográndose que cada dominio magnético sea más delgado permitiendo mayor cantidad de información por unidad de longitud.

En este método de grabación los dominios magnéticos se disponen en ángulo recto con el sustrato, de tal manera que su longitud queda determinada por el espesor del recubrimiento magnético.

En la grabación longitudinal la relación entre el largo y el ancho de los dominios magnéticos hace que al ser estos muy pequeños se desmagneticen entre sí. Así, si se incrementa la densidad de grabación los dominios serán más pequeños y la probabilidad de desmagnetización aumentará. Problema que no se presentaría en la grabación vertical.

El inconveniente que presenta la grabación perpendicular es que al ser muy pequeño el campo magnético y aún más pequeña la salida de la cabeza de grabación es necesario amplificar la señal sin distorsionarla.

Bajo condiciones de laboratorio las técnicas de grabación longitudinal pueden lograr hasta 15 000 bits/pul y 800 pistas/pul. Utilizándose discos de película delgada y cabezas de ferrita se pueden esperar 25 000 bits/pul y 1200 pistas/pul. Mientras que en la grabación perpendicular pueden esperarse hasta 100 000 bits/pul y 1200 pistas/pul.

## SISTEMA SERVO CONTROLADO

Este es un esquema de control de disco duro que puede aplicarse también a discos flexibles, mediante el cual se pueden almacenar varios MBytes de información en un sólo floppy.

La limitación más importante inherente a los drives tradicionales es la forma en que su mecanismo es controlado. Cuando el mecanismo de posicionamiento de la cabeza de lectura-escritura se mueve, ésta sólo puede aproximarse a su posición correcta para cada pista. El mecanismo de posicionamiento nunca sabe si la cabeza se encuentra en el lugar correcto, y no hay forma de monitorear si la cabeza está en el sitio adecuado en la pista, es decir, se trata de un sistema en malla abierta.

Como alternativa a este método tradicional se utiliza el sistema servo controlado, el cual es realimentado por medio de sensores que proporcionan datos al mecanismo de posicionamiento indicándole exactamente donde se encuentra la cabeza de lectura/escritura, de tal manera que este servomecanismo se autoajuste constantemente de acuerdo a la información de realimentación.

El mecanismo servo controlado determina la posición de la cabeza mediante la detección de marcas especialmente grabadas entre las pistas de datos en el lugar correspondiente al final de cada sector en cada sector del disco. Si la cabeza está posicionada directamente sobre el dato y exactamente entre dos marcas, las señales producidas por cada una de ellas tendrán la misma magnitud. Si la cabeza se mueve en una trayectoria concéntrica una marca producirá una señal mayor que la otra y el servomecanismo moverá la cabeza en la dirección de la señal más débil. Utilizando este esquema de control la posición de la cabeza de lectura-escritura puede ser controlada hasta un diez milésimo de pulgada logrando que la capacidad del floppy aumente.

Un inconveniente de los sistemas servo controlados es que las marcas deben ser grabadas con mucha precisión y los drives por si

mismos no son capaces de grabarlas con la precisión requerida, por lo que los discos requieren que la información servo sea grabada de fábrica.

Hace algunos años Drivetec Inc. diseñó un drive comercial utilizando este tipo de técnicas aumentando la capacidad del floppy a 3.3 MBytes sin formatear y 2.78 MBytes formateado. Kodak ha fabricado este mismo tipo de drive con el nombre de Kodak 3.3 y existen versiones de 6, 12 y hasta 25 MBytes de capacidad.

## 1.4 CIRCUITOS DE CONVERSION A/D

Existen diferentes formas de clasificar a los convertidores A/D. Un método es separarlos en programables y no programables. En un convertidor A/D programable, el proceso de conversión se realiza en un número dado de pasos, cada uno de los cuales toma un cierto tiempo fijo. El convertidor no programable puede requerir que una secuencia de eventos tome lugar antes de completar la conversión; sin embargo esta secuencia no tiene un tiempo fijo, sino que depende del tiempo de respuesta del circuito de conversión.

Otra forma de clasificar a los convertidores A/D sería de acuerdo a si están realimentados o si son del tipo de lazo abierto. En este último, se hace una comparación directa entre la entrada analógica de voltaje y una referencia analógica. El resultado de la comparación es una palabra digital que es equivalente a la entrada analógica. En convertidores de lazo cerrado a medida que avanza el proceso, se genera un voltaje analógico internamente como una función de la palabra digital que se realimenta a una entrada del comparador, este voltaje es comparado contra la entrada que está siendo convertida y cuando el voltaje de realimentación iguala a la entrada, la conversión es completada. La palabra digital es entonces el equivalente digital de la entrada analógica.

Una tercera clasificación sería de acuerdo a si utilizan la carga de un capacitor o si se compara con un voltaje discreto. El primer tipo consiste básicamente en codificar el tiempo que tarda en cargarse un capacitor hasta cierto voltaje de referencia o al valor de la entrada analógica. El segundo tipo consiste en la generación de voltajes discretos cuyos niveles son equivalentes a palabras digitales, y la comparación de éstos niveles discretos contra la entrada determina la salida.

## CONVERTIDORES A/D POR CARGA DE CAPACITOR

Tres ejemplos de este tipo de convertidores son: el convertidor de voltaje a frecuencia, el convertidor por modulación de ancho de pulso y el convertidor por integración.

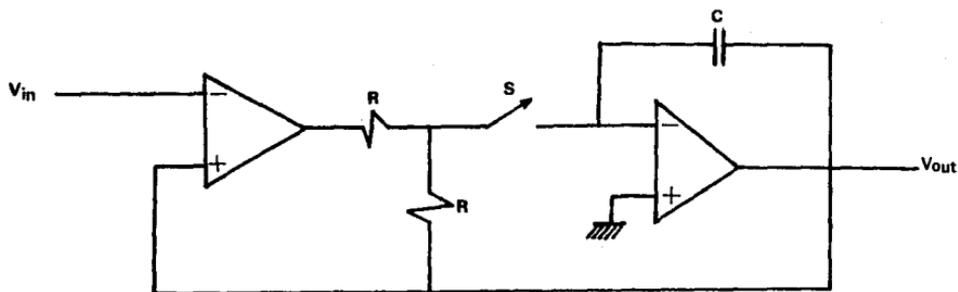
Convertidor de voltaje a frecuencia.

En este tipo de convertidor la entrada analógica de voltaje se convierte inicialmente, a una corriente constante proporcional, la cual es integrada por un amplificador-integrador de acoplamiento directo. La integración se lleva a cabo hasta que la salida excede  $+V_r$  o  $-V_r$ , instante en el cual uno de los comparadores genera un pulso de salida el cual es utilizado para reestablecer el integrador a cero, y vuelve a integrar la entrada. La operación continua produce de esta manera un cierto número de pulsos por segundo (frecuencia) que es proporcional a la entrada analógica. Estos pulsos pueden ser contados durante un periodo fijo por medio de un contador binario, cuya cuenta final es proporcional a la entrada analógica.

Convertidor por modulación de ancho de pulso

Este es uno de los dispositivos más simples para implementar un convertidor A/D. El nombre de este proceso de conversión proviene de el hecho de que el nivel de la señal analógica se transforma a un pulso cuya duración en el tiempo es una función del valor de la señal de entrada analógica. Dicho ancho se convierte a un formato digital por medio del conteo de los ciclos de una referencia de frecuencia que toma lugar entre el inicio y el final de dicho pulso.

El principio básico de operación de este convertidor se ilustra en la figura siguiente:



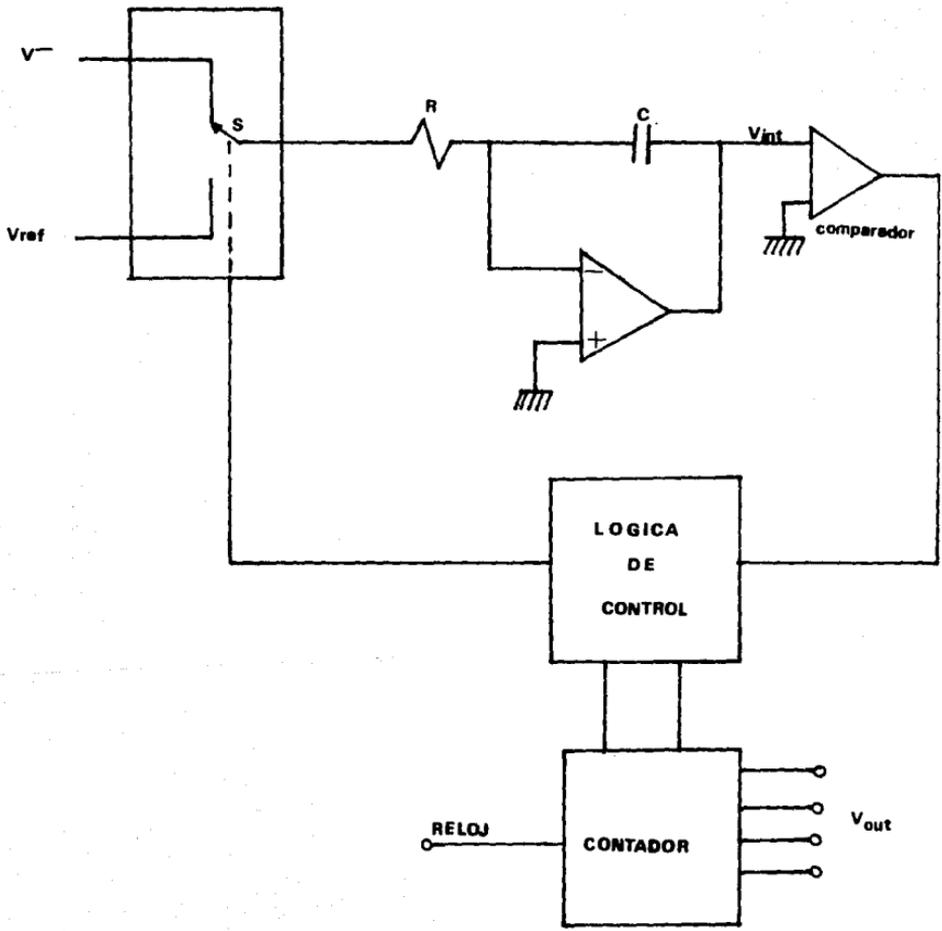
El interruptor de reset  $S_1$ , está cerrado hasta que empieza la conversión. En ese momento (el inicio del pulso) el interruptor se abre y el capacitor  $C_1$  se carga linealmente debido a la corriente constante  $I$ . El comparador analógico que se conecta al capacitor toma relativamente poca corriente. A medida que el capacitor se carga desde cero, el acumulador (generalmente un contador) cuenta los ciclos de la frecuencia de referencia. Cuando el voltaje en  $C_1$  iguala la entrada analógica de voltaje,  $V_{XA}$ , la salida del comparador cambia de estado (final del pulso). La señal

del comparador inhibe entonces la entrada al acumulador de la referencia de frecuencia, y la cuenta final en el acumulador es el equivalente digital de la señal analógica de entrada.

#### Convertidor por integración de doble rampa

Este es un modulador de amplitud de pulso. Su principal característica es la de proporcionar una mayor exactitud que el modelo anterior, pero con una mayor complejidad en equipo.

En este convertidor la señal de entrada es aplicada a un integrador; al mismo tiempo se inicia un conteo de pulsos de reloj. Cuando el contador llega a cierto número (en un periodo fijo de tiempo,  $T$ ), una referencia de voltaje con la polaridad opuesta se conecta al integrador. En ese instante, la carga acumulada en el capacitor integrador es proporcional al valor promedio de la entrada en el intervalo  $T$ . La integral de la referencia es una rampa descendiente con una pendiente  $-V_{ref}/RC$ . Al mismo tiempo un contador empieza a contar desde cero. Cuando la salida del integrador llega a cero, se detiene el conteo y se reestablece la circuitería analógica. Debido a que la carga ganada es proporcional a  $V_{in}T$ , y la cantidad de carga perdida es proporcional a  $V_{ref}\Delta t$ , entonces el número de ciclos en relación a todos los ciclos es proporcional a  $\Delta t/T$ , ó  $V_{in}/V_{ref}$ . Si la salida del contador es un número binario, éste será una representación binaria del número de entrada.



## CONVERTIDORES POR COMPARACION DE VOLTAJE DISCRETO

### Convertidor de rampa

Este es uno de los convertidores más simples, pero tiene la desventaja de ser el más lento. Cuando el voltaje de entrada es máximo se necesitan  $2^n - 1$  pasos para terminar la conversión. Como muestra la figura, el proceso de conversión inicia con un pulso de reset al contador en to. El contador empieza a recibir y a contar señales de reloj por la entrada 1. El convertidor D/A está encadenado al contador de manera tal que a medida que la cuenta se incrementa, así mismo el voltaje D/A crece. Cuando la cuenta ha aumentado lo suficiente como para que el voltaje en la salida del D/A sea ligeramente mayor que la entrada analógica de voltaje, el comparador de voltaje cambia de estado, e inhibe la entrada 1 para que no entren más pulsos de reloj al contador. En ese momento la palabra digital en el contador es el equivalente digital de la entrada analógica. Cuando se tiene a la entrada la escala completa de voltaje, se requiere que el contador cambie desde 000...00 hasta 111...11, lo cual toma, para un convertidor de 12 bits, 4096 pulsos de reloj, lo cual suele ser prohibitivo, en términos de tiempo.

## Convertidor de aproximaciones sucesivas

Los convertidores A/D de aproximaciones sucesivas son ampliamente utilizados, especialmente para interfaces con computadoras, porque son capaces de una alta resolución (16 bits) y una alta velocidad (1MHz). El tiempo de conversión es fijo e independiente de la magnitud del voltaje de entrada. Cada conversión es única e independiente de los resultados de la conversión anterior, porque la lógica interna es reiniciada antes de cada conversión.

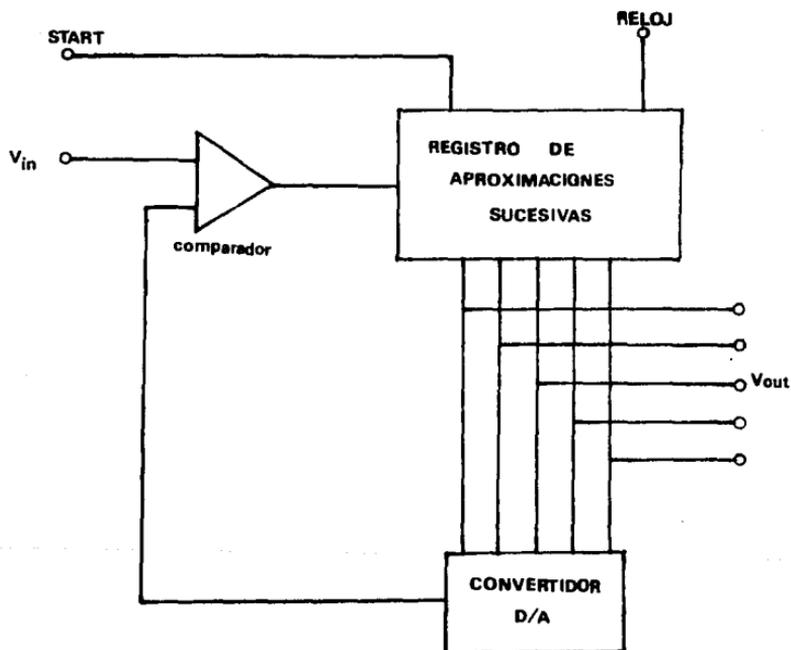
Los convertidores modernos CI incluyen salidas de datos en tres estados y bytes de control para facilitar la comunión con el microprocesador. Las salidas de tres estados tienen, además de los estados "1" y "0", cuando están habilitados, una condición deshabilitada, en la cual la salida es desconectada y presenta una alta impedancia, tal y como si fuera un circuito abierto.

La técnica de conversión consiste en comparar la entrada desconocida contra un voltaje o corriente conocida que es generada por un convertidor D/A. La entrada del convertidor D/A es un número digital proveniente de la salida del convertidor A/D. El proceso de conversión es notablemente similar al proceso de pesar usando una balanza química con un juego de  $n$  pesos.

Después de que el comando de conversión es aplicado y el convertidor ha sido inicializado, la salida MSB del convertidor D/A (1/2 escala) es comparada con la entrada. Si la entrada es más grande que el MSB, se registra una salida alta ("1"), y el siguiente bit (1/4 escala) es probado. Si la entrada es menor que el MSB se registra un cero a la salida ("0"), y se prueba el siguiente bit. Si el segundo bit no es suficientemente grande para exceder la entrada, se deja en "1" y el tercer bit es probado. Si el segundo bit sobrepasa la escala, se apaga ("0") y el tercer bit es probado. El proceso continúa hasta el último bit.

El contenido del registro de salida forma un código binario correspondiente a las magnitudes de la señal de entrada.

La siguiente figura muestra el diagrama de bloques de un A/D de aproximaciones sucesivas.



En el ejemplo de la figura, la entrada no cambia durante la conversión. Si la entrada está cambiando durante el tiempo de conversión, el número de salida sólo representará la entrada analógica vagamente. Sin embargo, aún cuando el valor final se igualara con el valor desconocido habría todavía la pregunta de si

dicho valor es el verdadero, especialmente si la conversión ha ocurrido en un tiempo específico. Para evitar problemas de este tipo, generalmente se usa un dispositivo de muestreo y retención antes del convertidor para retener el valor de entrada que estaba presente al inicio de la conversión, y mantenerlo constante hasta que ésta termine. La salida de estado del convertidor (ocupado/conversión completa) puede usarse para liberar el dispositivo de muestreo y retención de su modo de retención al final de la conversión. El dispositivo de muestreo y retención puede no ser necesario si la señal de entrada tiene variaciones suficientemente pequeñas y está suficientemente libre de ruido.

#### Convertidor flash

La siguiente figura muestra un convertidor A/D tipo flash.

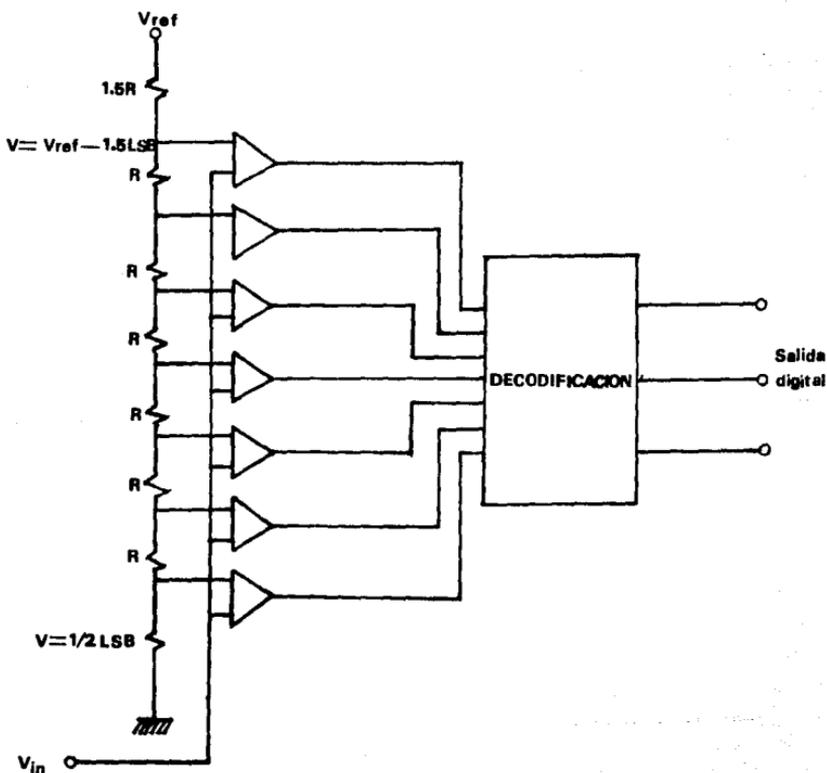
El funcionamiento es como se explica a continuación. Un divisor de voltaje fija las referencias de potencial para cada una de las entradas inversoras del bloque de comparadores. El voltaje que se encuentre hasta arriba de la red divisora representará el máximo valor de la escala para el convertidor. El voltaje a convertir es aplicado en las entradas no inversoras de todos los comparadores en paralelo. Si la entrada de voltaje en los comparadores es más grande que la referencia de voltaje en su entrada inversora, la salida del comparador será un nivel alto de voltaje.

La ventaja de este convertidor flash es la velocidad de conversión la cual es simplemente la propagación del tiempo de retardo de los comparadores.

La desventaja que presenta es el número de comparadores necesarios para producir un resultado con una resolución razonable. El convertidor de 2 bits requiere 3 comparadores, por lo que para producir convertidores con  $n$  bits es necesario utilizar  $2^n - 1$  comparadores. Los convertidores disponibles de 8 bits pueden hacer conversiones en 20 ns.

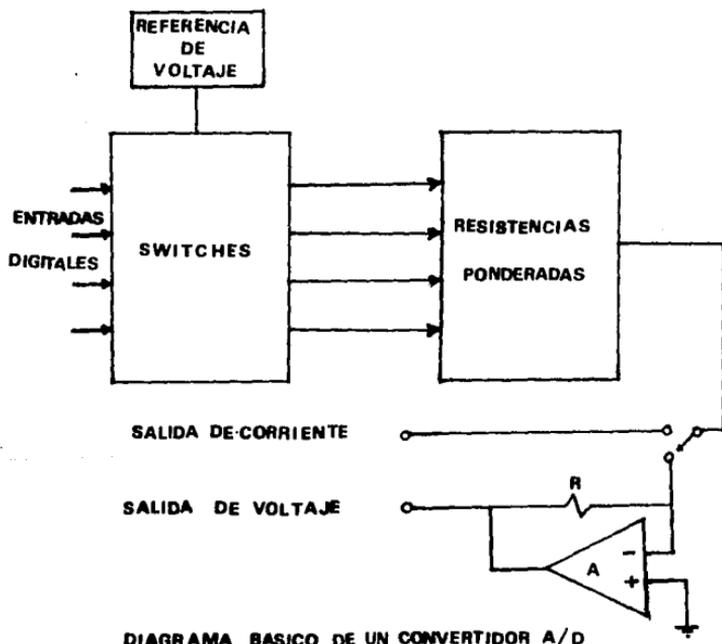
Un mejor muestreo se podría lograr con un mayor número de bits pero dado al incremento en el costo de los dispositivos convertidores A/D se eligió un convertidor de 8 bits, considerando estos suficientes para tener una resolución aceptable.

El convertidor utilizado fue un ADC0809 con un multiplexor. Dicho multiplexor se puede tomar para hacer que la salida sea de dos canales y no monoaural como esta en el proyecto terminal.



## 1.5 CIRCUITOS DE CONVERSION D/A

Un convertidor D/A básico se construye con una referencia de voltaje, un juego de resistencias ponderadas y un juego de interruptores. Su etapa de salida está constituida por un bloque que presenta alta impedancia de entrada, baja impedancia de salida, convierte corriente a voltaje y provee amplificación.



En este ejemplo, el amplificador operacional mantiene uno de los extremos de todas las resistencias de un juego de  $n$  resistencias a un potencial de tierra. Las resistencias están ponderadas binariamente, por ejemplo cada una está ponderada por  $2^j$  por lo tanto, si se aplica el mismo potencial a todas las

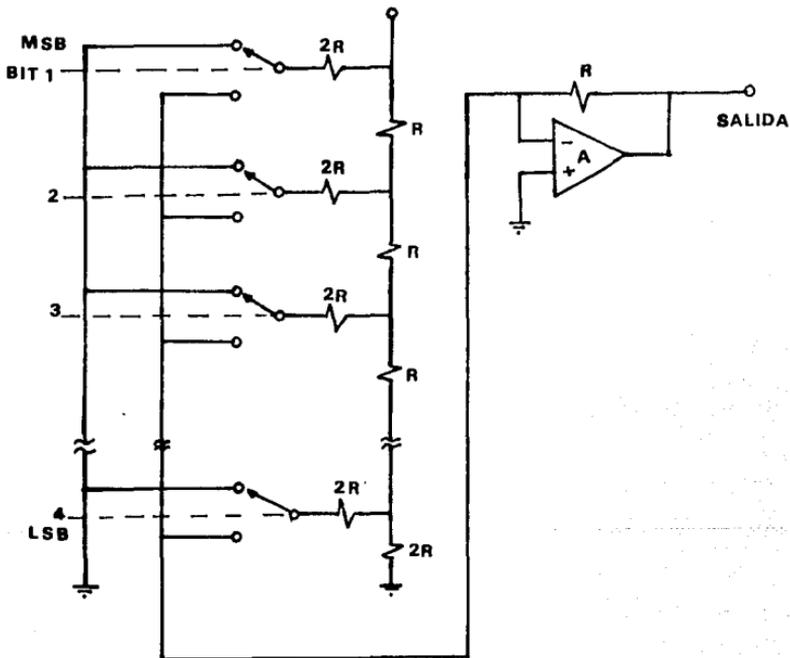
resistencias, las corrientes estarán ponderadas por  $2^{-j}$ . Los interruptores están operados por la lógica digital: abiertos para "0" y cerrados para "1". Cada interruptor que se cierra agrega una cantidad ponderada de corriente al nodo suma que está conectado a la entrada inversora del amplificador operacional. El voltaje de salida es proporcional a la corriente total, y por lo tanto a uno de los  $2^n$  posibles valores representables por el código de entrada.

En general, para convertidores de más de 4 bits este esquema no es práctico debido a las resistencias; por ejemplo, en un convertidor de 12 bits el rango requerido de valores de resistencias será de 2048 a 1, esto es 20 M $\Omega$  para LSB hasta 10 K $\Omega$  para el MSB. Si las resistencias han de ser fabricadas en película delgada o gruesa, ó si se va a fabricar un circuito integrado dicho rango sería totalmente impráctico. Si se utilizan resistencias discretas el costo y el tamaño se incrementan, y surgen problemas con el inventario y con las impedancias parásitas que se originan de la interconexión del gran número de elementos.

#### ARREGLOS DE RESISTENCIAS

Una manera de reducir tanto el número de resistencias como su rango es, utilizar un número limitado de valores repetidos en una configuración que provea una atenuación conveniente. Un buen método para lograr esto consiste en utilizar un grupo de cuatro valores escalados (por ejemplo 2R, 4R, 8R, 16R) para cada conjunto de 4 bits, con una atenuación de 16:1 para cada grupo sucesivo hasta llegar al menos significativo. Por lo tanto, los 4 bits más significativos se suman sin atenuación, los siguientes 4 bits se atenúan de 16:1 y se suman, y los últimos 4 bits se atenúan por 265:1 y se suman, o más bien se atenúan por 16:1 respecto a la etapa anterior, y se suman a la siguiente etapa. Una ventaja de este esquema es que se puede utilizar para una conversión de BCD, reemplazando la resistencia de atenuación con una que provea una reducción de 10:1. Si continuamos con este esquema de atenuación,

se llega a la escalera R-2R la cual es una forma conveniente y muy popular. La figura muestra una escalera con un amplificador operacional inversor en una configuración que emplea interruptores CMOS.



CONVERTIDOR D/A ESCALERA R-2R

Si todos los bits, salvo el MSB están apagados, es decir, aterrizados, el voltaje de salida será  $(-R/2R)E_{ref}$ . Para el segundo bit, el conjunto de resistencias aterrizadas de todos los bits menos significativos (debajo de la terminal correspondiente al segundo bit) es  $2R$ ; el divisor formado por la resistencia  $R$  en serie y los dos elementos  $2R$  en paralelo, provee una atenuación del 50% por lo tanto, la corriente a través del punto de suma será la mitad de la suministrada por el MSB siendo el voltaje de salida  $0.5(-R/2R)E_{ref}$ . A lo largo de la escalera, cada resistor  $2R$  tiene la mitad del potencial que el inmediatamente superior, por lo tanto atenúa la corriente a la mitad. El voltaje de salida es proporcional (por superposición) a la suma de todas las corrientes ponderadas que se hayan conectado.

La red  $R-2R$  puede ser utilizada para obtener una salida de voltaje no invertido y sin atenuación simplemente intercambiando las terminal de referencia con la salida. La terminal de referencia tiene una baja impedancia y la terminal de salida esta conectada a una carga de alta impedancia, tal como la entrada de un amplificador operacional conectado en configuración de seguidor. La resistencia efectiva a tierra de todos los resistores debajo de un nodo dado es  $2R$ . De esa forma, si el MSB esta encendido, la salida sera  $V_{ref}/2$ . Si el bit 2 esta encendido (con todos los otros bits aterrizados), su resistencia serie  $2R$  y las inferiores  $2R$  forman un generador de  $E_{ref}/2$  en serie con  $R$ . Debido a que  $R$  esta en serie con otra  $R$ , y una resistencia efectiva  $2R$  esta aterrizada por medio del interruptor del MSB, el voltaje en el nodo de salida es la mitad del generador en el nodo "2", o  $(1/4)V_{ref}$ . Las contribuciones de los subsecuentes bits de la suministrada por el MSB siendo el voltaje de salida  $0.5(-R/2R)E_{ref}$ . A lo largo de la escalera, cada resistor  $2R$  tiene la mitad del potencial que el inmediatamente superior, por lo tanto atenúa la corriente a la mitad. El voltaje de salida es proporcional (por superposición) a la suma de todas las corrientes ponderadas que se hayan conectado.

La red  $R-2R$  puede ser utilizada para obtener una salida de voltaje no invertido y sin atenuación simplemente intercambiando

las terminal de referencia con la salida. La terminal de referencia tiene una baja impedancia y la terminal de salida esta conectada a una carga de alta impedancia, tal como la entrada de un amplificador operacional conectado en configuración de seguidor. La resistencia efectiva a tierra de todos los resistores debajo de un nodo dado es  $2R$ . De esa forma, si el MSB esta encendido, la salida sera  $V_{ref}/2$ . Si el bit 2 esta encendido (con todos los otros bits aterrizados), su resistencia serie  $2R$  y corrientes de los bits se suman y se convierten a voltaje por medio de una resistencia de retroalimentación, el cual está incluido en el CI, al igual que el arreglo de resistencias en escalera, para una mejor estabilidad ante variaciones de la temperatura ambiente. La presencia en este circuito de los interruptores conmutadores de corriente provoca problemas con la linealidad debido a las desviaciones de los amplificadores operacionales, problemas con el ruido proveniente de los interruptores, y una respuesta más lenta debido a los mismos.

En el modo de voltaje la escalera se utiliza como un atenuador resistivo; los interruptores pueden funcionar como una referencia de voltaje de baja impedancia o como tierra. La magnitud de la corriente que fluye a través de los interruptores no influye en la precisión, pero la fuente de referencia y los interruptores deben tener una impedancia suficientemente baja como para que el paso de la corriente a través de ellos no provoque una caída de voltaje significativa.

Para convertidores CMOS en el modo de conmutación de voltaje, la resistencia constante en la entrada del amplificador conectada a la escalera elimina los problemas de linealidad causados por la modulación del voltaje de corrimiento del amplificador debida a la resistencia en el punto suma, que depende del código de entrada, tal como ocurriría en el modo de conmutación de corriente. Agregado a esto, la capacitancia del interruptor está más lejos del amplificador, y la carga se transfiere a la entrada de la fuente o a tierra, en lugar del punto de suma. Además la capacitancia de salida de la red es considerablemente más baja. Todo esto conduce a una respuesta más limpia y rápida del circuito ante cambios del código de entrada. Otra ventaja adicional es que la salida del sistema es de la misma polaridad que la referencia de voltaje: esto hace posible operar el convertidor y su amplificador con una fuente de voltaje monopolar. Se tienen unas pequeñas desventajas, como la no linealidad que ocurre si la referencia de voltaje es demasiado grande (debido a que la resistencia de conducción del FET es una función del voltaje), y la limitación a una sola polaridad en la entrada.

## REFERENCIAS DE VOLTAJE

Existen diversos tipos de circuitos que proporcionan referencias de voltaje, uno de los más simples es el diodo Zener, el cual puede ser utilizado en conjunto con amplificadores operacionales con la finalidad de estabilizar el punto de operación, aumentar la impedancia de la carga o para convertir el voltaje en corriente. Este tipo de referencia está siendo sustituido por nuevos diseños de circuitos integrados que actúan como diodos Zener sintéticos, de alto rendimiento, así como diodos Zener construidos en sustratos inmunes a los cambios de temperatura. En los circuitos integrados las referencias están construidas por diodos Zener muy estables, que han sido ajustados por medio de laser para minimizar el error.

Para convertidores que utilizan fuentes de corriente activas se debe proveer de una referencia que compense las características indeseables de dichas fuentes y de los interruptores asociados a ellas.

Se eligió un convertidor D/A de 8 bits con salida de corriente (DAC 0800) por sus características mencionadas en el apéndice y que además fuera compatible con el convertidor A/D utilizado el cual como se mencionó es de 8 bits.

## **CAPITULO II**

### **ACCESO Y CONTROL DE INFORMACION**

### III. ACCESO DE INFORMACION MEDIANTE UN MPU

Uno de los puntos más importantes en la evolución de los microprocesadores (MPU) ha sido la de orientarlos a trabajar como Unidades Centrales de Procesamiento (CPU) de propósito general. Después de la aparición del MC6800 de Motorola, Intel produjo el 8085, como una versión mejorada del 8080. Motorola produjo entonces el MC6809 que básicamente era un procesador de 8 bits con unas cuantas instrucciones de 16 bits. Para el año de 1978, Intel sacó al mercado el 8086, un microprocesador completo de 16 bits. Poco tiempo después Motorola puso a la venta el MC68000 como su entrada a la carrera de los MPU's de 16 bits. Estos MPU's trabajan directamente con palabras de 16 bits y pueden direccionar hasta 1 MByte de memoria, además de que pueden ejecutar instrucciones mucho más rápido que los MPU's de 8 bits. También poseen instrucciones simples para funciones que requerían una larga secuencia de instrucciones en procesadores de 8 bits. La evolución de estos dispositivos no ha terminado, se trabaja ahora con modelos de 32 bits, capaces de direccionar Giga y hasta Terabytes de memoria.

La familia de microprocesadores de Intel 8086, 8088, 80186, 80188, 80286, etc. son ampliamente utilizados en computadoras personales, sistemas computacionales, y en sistemas de control industrial. Para efectos de este trabajo, creemos que la utilización de un microprocesador de esta familia para desarrollar el hardware requerido, es adecuado dado que el hardware asociado al microprocesador es sencillo de implementar y resulta suficiente para realizar el control de todos los dispositivos que se utilizarán y que existe bastante información en cuanto a su utilización.

## EL MICROPROCESADOR 8088

El microprocesador 8088 es un microprocesador de 16 bits, lo cual quiere decir que tanto su Unidad Aritmética Lógica (ALU), sus registros internos, y la mayor parte de sus instrucciones están diseñadas para trabajar con palabras binarias de 16 bits. El 8088 posee un bus de direcciones de 20 bits, con el cual es capaz de direccionar hasta  $2^{20}$  (1 048 576) localidades de memoria de un byte de información cada una. Posee además un bus de datos de 8 bits (el 8086 lo tiene de 16 bits), el cual es más que suficiente para controlar los dispositivos a utilizar durante el desarrollo de este trabajo.

### ARQUITECTURA INTERNA

Como puede apreciarse en el diagrama de bloques mostrado, el CPU está dividido en dos partes funcionales independientes, la Unidad de Bus de Interfaz (BIU) y la Unidad de Ejecución (EU). El BIU manda las direcciones, lee las instrucciones de memoria, lee o escribe datos en memoria y puertos. En otras palabras, el BIU maneja todas las transferencias de datos y direcciones en los buses de la unidad de ejecución.

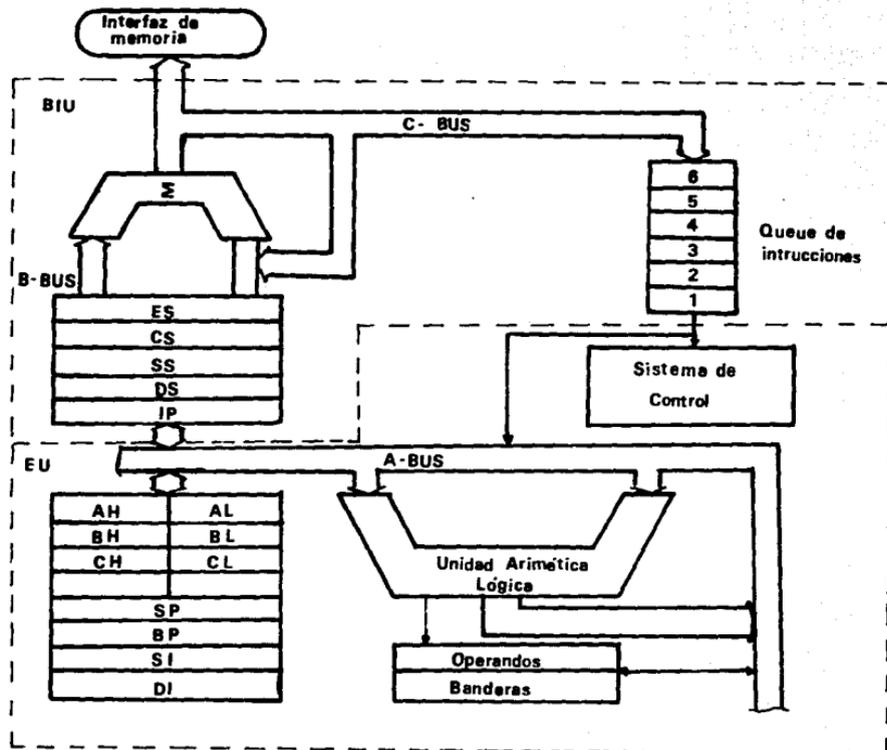


DIAGRAMA DE BLOQUES INTERNO  
DEL 8088

A continuación se presentan las funciones de cada una de las partes que forman el BIU:

#### EL QUEUE

Para acelerar la ejecución del programa, el BIU acomoda hasta seis bytes de instrucciones en orden, en un registro de primeras entradas- primeras salidas llamado 'Queue'. El BIU puede ir acomodando más conforme la EU va decodificando las instrucciones almacenadas o bien va ejecutando aquellas que no requieran del uso de los buses.

#### REGISTROS DE SEGMENTO

El BIU contiene cuatro registros de segmento de 16 bits. Estos son: Registro de Segmento de Código (CS), registro de Segmento de Stack (SS), registro de Segmento Extra (ES) y el registro de Segmento de Datos (DS). Estos registros de segmento son utilizados para almacenar los primeros 16 bits de las direcciones de inicio de los cuatro segmentos de memoria con los que puede trabajar el CPU. A la vez, el CPU puede trabajar con cuatro segmentos de memoria de hasta 64 Kbytes, los cuales de ser necesario pueden ser sobrepuestos o traslapados. A la parte de la dirección de inicio de un segmento, almacenada en los registros de segmento es denominada Base del Segmento.

#### APUNTADOR DE INSTRUCCIONES

El siguiente punto de interés en el BIU es el registro del Apuntador de Instrucciones (IP). Como se sabe, el CS almacena los primeros 16 bits de la dirección de inicio de los bytes de código de las instrucciones. El IP guarda los 16 bits correspondientes al siguiente byte de código correspondiente a este segmento de código. El valor contenido en el IP está referido como un *offset*, debido a que dicho valor requiere ser sumado a la dirección del segmento base en CS para poder producir la dirección física de 20

bits. Dicho de otra manera, el IP contiene la distancia desde la dirección base hasta la correspondiente a la siguiente instrucción a ser leída por el BIU. Debe notarse que ambos números no se suman en línea, sino que el contenido de CS debe ser corrido a la izquierda antes de que IP pueda ser sumado a él para obtener la dirección física correcta.

Ahora se presentan las partes constituyentes de la unidad de ejecución:

#### CIRCUITERIA DE CONTROL, DECODIFICADOR DE INSTRUCCIONES Y ALU

La circuiteria de control dirige las operaciones internas. El decodificador traduce las instrucciones en una serie de acciones que la EU realiza. La EU contiene además una Unidad Aritmética Lógica la cual se encarga de realizar las operaciones de suma, resta, AND, OR, XOR, incrementar, decrementar, complementar o hacer corrimientos de números binarios de 16 bits.

#### REGISTRO DE BANDERAS

Una bandera es un Flip-flop que indica alguna condición producida por la ejecución de cierta instrucción, o para el control de ciertas operaciones de la unidad de ejecución. El registro de banderas en la EU contiene nueve banderas activas, su localización es como sigue:

BIT 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

U	U	U	U	OF	DF	1F	TF	SF	ZF	U	AF	U	PF	U	CF
---	---	---	---	----	----	----	----	----	----	---	----	---	----	---	----

Donde:

CF            Bandera de acarreo. Acertada por el carry del MSB

PF            Bandera de paridad. Acertada si el resultado es de paridad par.

AF            Bandera auxiliar de acarreo para BCD.

ZF            Bandera de cero. Acertada si el resultado es cero

SF            Bandera de signo. Es igual al MSB del resultado.

TF            Single Step Trap.

1F            Habilidad de interrupción.

DF            Dirección de cadena.

OF            Sobreflujo.

Las banderas CF, PF, AF, ZF, SF y OF son banderas condicionales. Las tres banderas restantes son de control. Las condicionales son acertadas o no en base a los resultados de las operaciones realizadas por la ALU. Las banderas de control son afectadas por instrucciones específicas en un programa.

#### REGISTROS DE PROPOSITO GENERAL

La unidad de ejecución cuenta además con ocho registros de propósito general los cuales son: AH, AL, BH, BL, CH, CL, DH y DL. Estos registros pueden ser utilizados individualmente para almacenamiento temporal de datos de 8 bits. El registro AL es denominado también *acumulador*. Ciertos pares de registros pueden ser utilizados en conjunto para almacenar datos de 16 bits, estos son AH-AL (también llamado AX), BH-BL (BX), CH-CL (CX), y DH-DL (DX). Para las operaciones de 16 bits, AX es el *acumulador*.

La ventaja de la utilización de estos registros para el almacenamiento temporal de la información es que una vez que los

datos estan allí, es mucho más rápido su acceso que a partir de cierta localidad de memoria.

#### REGISTRO APUNTADOR DE STACK

El STACK es una sección de memoria cuya finalidad es la de almacenar direcciones y datos mientras una subrutina se está ejecutando. Los primeros 16 bits de la dirección base del Stack están almacenados en el SS. El registro Apuntador de Stack (SP) contiene el offset de 16 bits, desde el inicio del Stack hasta la localidad de memoria donde una palabra fue más recientemente guardada en el stack, es decir, el final del Stack. La dirección real de dicha localidad de memoria viene dada por la suma del SS corrido a la izquierda sumado con el SP.

#### OTROS APUNTAORES Y REGISTROS DE INDICE

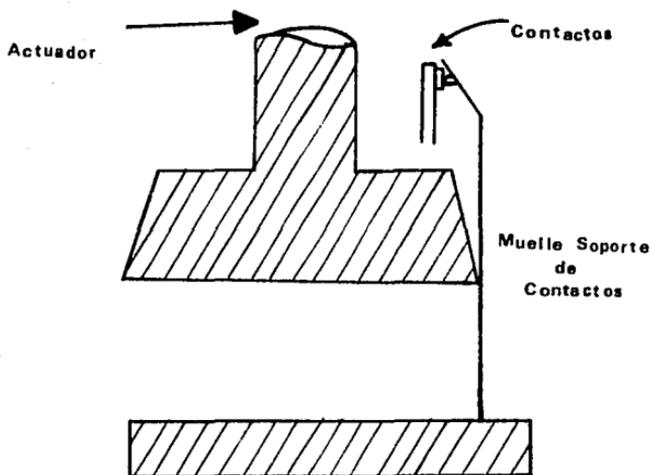
Además del apuntador de stack, la EU contiene un registro apuntador base (BP), un registro de Índice Fuente (SI) y un registro de Índice Destino (DI). Estos tres registros pueden ser utilizados para el almacenamiento temporal de información como los registros de propósito general vistos anteriormente. Sin embargo, su mayor aplicación se encuentra utilizandolos para almacenar el offset de 16 bits de una palabra de datos en uno de los segmentos.

## II.2 SELECCION DE TECLADOS

Se denomina TECLADO al género de periféricos de entrada, constituidos por un conjunto de botones pulsadores, de tal modo que cada botón corresponda con determinado carácter, función, o instrucción. El elemento unitario, en si, es el formado por el conjunto pulsador-tecla, que es a menudo denominado simplemente tecla. La tecla o capuchón, aún cuando no cumple ninguna función eléctrica, es absolutamente imprescindible con el fin de identificar cada pulsador, así como ofrecer una correcta superficie de apoyo apropiada a las características de los dedos de los operadores. En general, cada tecla controla un simple interruptor que permanece abierto mientras el pulsador es oprimido. Los teclados pueden ser clasificados dependiendo del pulsador empleado, del número de códigos generados por carácter o por la forma en que se codifica la información recibida.

### PULSADORES MECANICOS CONVENCIONALES

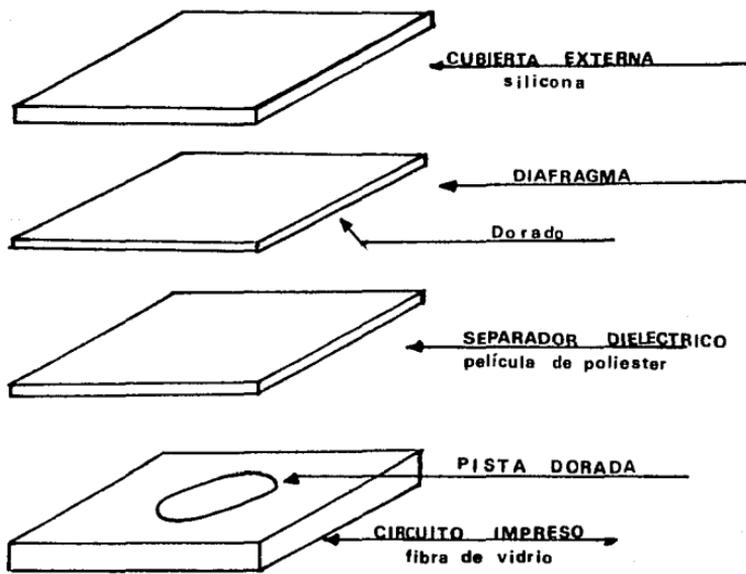
En los pulsadores mecánicos dos piezas de metal se separan cuando se oprime una tecla directamente. Son utilizados básicamente por su economía, además permiten, sin grandes complicaciones, configuraciones de contactos múltiples.



PULSADOR MECANICO

#### PULSADORES MECANICOS DE LAMINA FLEXIBLE

Este tipo de interruptor, formado por una serie de láminas sobrepuestas, se basa en la deflexión de un diafragma, dorado en su cara inferior, que permite establecer contacto con un circuito impreso a través de aperturas en un separador dieléctrico. Una cubierta silicona protege los contactos contra cualquier contaminante. Algunas versiones más económicas emplean láminas flexibles de silicona conductora, que sustituye la cubierta y el diafragma dorado. En algunos casos se emplea una base serigrafiada de tinta conductora como sustrato.



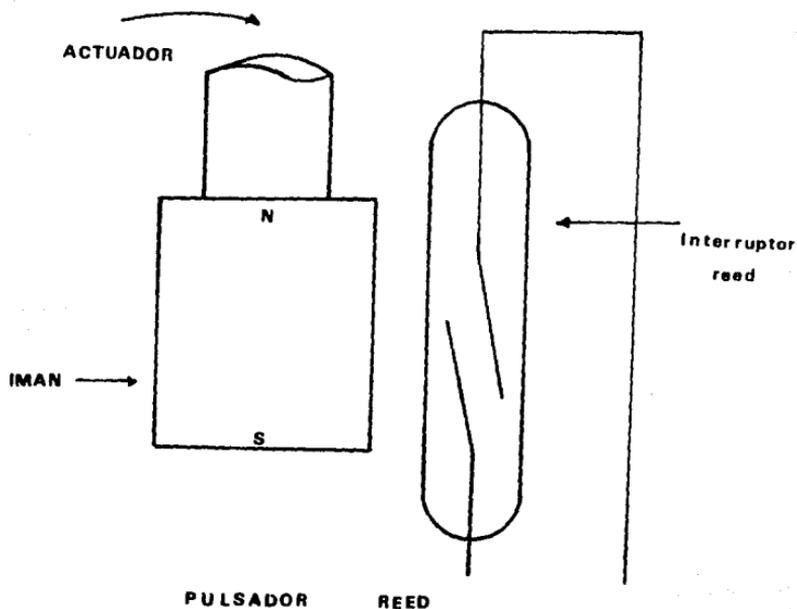
**PULSADOR DE LAMINA  
FLEXIBLE**

**PULSADORES MECANICOS DE MEMBRANAS**

Estos pulsadores emplean unos discos metalicos en forma de casquete esférico, que cuando son oprimidos pasan a una condición invertida, estableciendo el contacto requerido.

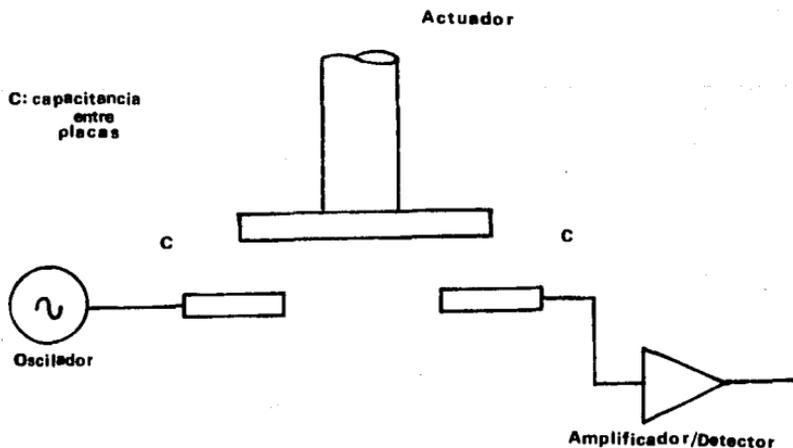
## PULSADORES REED

Están formados por contactos inmersos en una atmósfera inerte, sellados por una cápsula de vidrio hermética. El movimiento de la tecla desplaza un pequeño imán permanente que provoca el cierre de los contactos; dada la acción indirecta sobre estos últimos, no se transmiten sobrecargas mecánicas que provoquen fatiga y desgastes prematuros. Este tipo de contacto ofrece una vida útil cinco veces superior al clásico pulsador mecánico.



## PULSADORES CAPACITIVOS

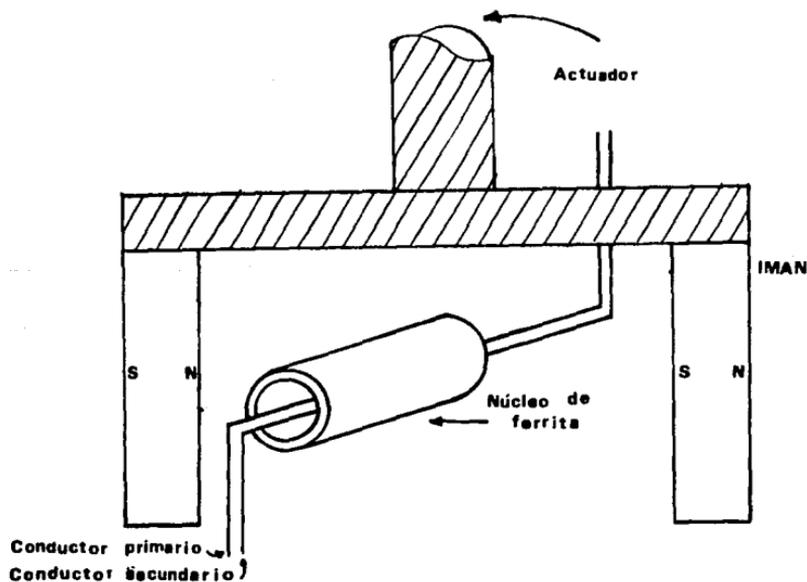
Este tipo de pulsadores emplea un cambio en la capacitancia de un condensador para entregar una salida. Se emplean dos superficies vecinas sobre un mismo circuito impreso, estando una de ellas excitada por la señal alterna de un oscilador. Si se aproxima paralelamente una placa conductora sobre ambas superficies, se provoca un acoplamiento entre ellas, \ la capacitancia de un condensador para entregar una salida. Se emplean dos superficies vecinas sobre un mismo circuito impreso, estando una de ellas excitada por la señal alterna de un oscilador. Si se aproxima paralelamente una placa conductora sobre ambas superficies, se provoca un acoplamiento entre ellas, con lo que aparece una fracción de la señal alterna en la salida, la cual debe ser convenientemente amplificada y convertida a niveles lógicos. Los pulsadores capacitivos ofrecen la elevada confiabilidad de los interruptores sin contactos móviles, sin embargo son muy sensibles a interferencias.



**PULSADOR CAPACITIVO**

## PULSADORES DE NUCLEOS MAGNETICOS

El elemento conmutador es un núcleo toroidal de ferrita empleado como transformador. El núcleo es atravesado normalmente por dos hilos: uno, energizado a alta frecuencia, se emplea como primario; el segundo como secundario. El acoplamiento es muy bajo en la posición normal del pulsador, donde el núcleo se encuentra saturado debido al campo del imán permanente. La presión de la tecla desplaza el imán, quitando la saturación del núcleo y generando una salida que debe ser amplificada, rectificada y trasladada a niveles lógicos; su confiabilidad es comparable a los pulsadores capacitivos.

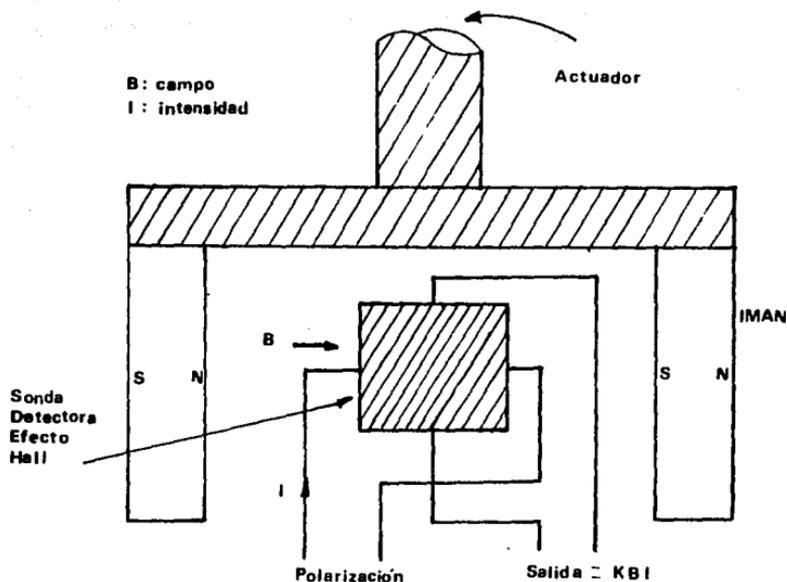


PULSADOR DE NUCLEO MAGNETICO

## PULSADORES DE EFECTO HALL

Los sensores de efecto Hall están formados por una pastilla semiconductor por la cual circula continuamente una corriente y un campo magnético perpendicular a ella que provoca una deformación de las líneas equipotenciales sobre la superficie del semiconductor, apareciendo a la salida una tensión proporcional al producto de la corriente de polarización por la intensidad del campo magnético aplicado. La conmutación se obtiene al aproximar un imán permanente al sensor, que desarrolla una tensión de salida que es amplificada y convertida en digital. Generalmente el conjunto formado por el sensor, amplificador, disparador de Schmitt, monostable opcional y etapa de salida, forma un circuito integrado monolítico asociado a cada pulsador.

Se distinguen dos tipos fundamentales: estático y dinámico. Los primeros generan una salida, mientras exista campo magnético a su entrada. Los pulsadores dinámicos conducen durante cierto periodo (20 $\mu$ s) cuando el campo de entrada supera el nivel de conmutación, pero no lo hacen durante el resto del tiempo que dicho campo permanezca a nivel elevado, ni durante el alejamiento del imán. A la salida puede colocar un transistor de colector simple o doble. Dada la ausencia de contactos, la baja impedancia de todas las señales de interconexión y la insensibilidad a polvo, suciedad y contaminantes, hacen de este tipo de pulsadores los más confiables aún cuando presentan desgaste del elemento móvil y del resorte de retorno. Este último, en algunos casos, es sustituido por el sistema magnético de retorno, que proporciona simultáneamente una realimentación al tacto.



### PULSADOR DE EFECTO HALL

#### MODOS

El número de códigos generados por la presión de una tecla es definido como el número de modos de un teclado.

Un teclado mono-modo genera solamente un código por cada tecla, son utilizados cuando el número de códigos es reducido. Los teclados de modos múltiples generan dos o más códigos por cada tecla en función de como se utilicen las teclas selectivas. Los teclados de dos modos se emplean para disponer de letras mayúsculas y minúsculas. Los conjuntos de modos múltiples reducen el trabajo realizado por el operador, puesto que puede utilizarse una sola mano para activar las teclas selectivas. Su ventaja consiste en la importante reducción del número de teclas, siendo este factor primordial cuando no se exige elevada velocidad al introducir datos. Los códigos estándar tales como ASCII son particularmente adecuados al empleo de múltiples modos,

puesto que por sus propia estructura, se obtienen los códigos de los diversos modos de cada tecla con solo modificar uno o dos bits.

## CODIFICACION

Como señales de salida de un teclado pueden utilizarse las conexiones correspondientes a todos y cada uno de los distintos conjuntos tecla-pulsador que lo constituyen. Esto puede ser válido para teclados simples formados por un reducido número de teclas; pero sería muy complicado si el número total de teclas supera ciertos umbrales. Evidentemente se han buscado soluciones efectivas que han sido centradas en la codificación de los datos de salida. Esta codificación consiste en numerar en forma binaria cada uno de los distintos códigos emitidos por el teclado, de tal modo que el número total de bits requeridos para expresar cualquier código no supera los umbrales de maniobrabilidad.

La codificación más usual para teclados numéricos reducidos es la hexadecimal; en el caso de teclados alfanuméricos se amplía la codificación, siendo el código más usual el ASCII de 6 o 7 bits según sea reducido o completo.

Cuando se trata de teclados más grandes se emplean circuitos integrados encadenados del tipo de decodificadores 74147.

## CONEXION MATRICIAL

Cuando el número de teclas sea más elevado no es práctico seguir empleando los circuitos codificadores antes descritos. La técnica más usual consiste en conectar las teclas en forma

matricial, de tal modo que el número total de teclas conectadas es igual al número de intersecciones, permitiendo la conexión de hasta 128 teclas repartidas en una matriz de 8 filas por 16 columnas, es decir las teclas forman una especie de red en donde cada tecla ocupa una posición bien definida y cuando esta sea presionada será posible conocer el renglón y la columna que ocupa y así poder determinar cual tecla fué presionada.

### EXPLORACION SECUENCIAL

La conexión matricial no puede generalizarse por lo que se acude como norma general a realizar los circuitos codificadores que emplean técnicas de exploración secuencial. Un circuito clásico se basa en un contador de 7 bits, un multiplexor y un decodificador de 4 a 16. Las teclas codificadas forman una matriz en la que cada tecla conecta una salida del decodificador con una entrada del multiplexor. El decodificador está seleccionado por los 4 bits menos significativos y el multiplexor del contador. Cuando se pulsa una tecla se cierra una conexión, de tal modo que cuando el contador alcanza el código apropiado el multiplexor conmuta y dispara un monoestable redispensible, que detiene el conteo. El monoestable se dispara continuamente mientras la tecla permanece oprimida.

Una PROM o circuitería equivalente realiza la codificación y adaptación de modos aunque esta función puede ser realizada mediante acceso a una tabla incluida en el microprograma.

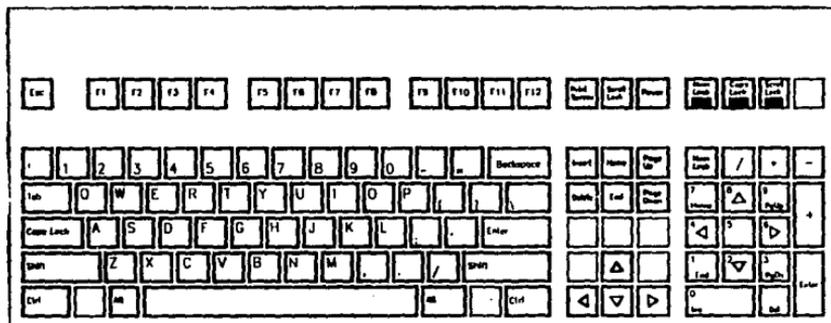
Un esquema completo para codificar un teclado de hasta 128 teclas por el método de exploración secuencial resultaría bastante voluminoso si se realiza mediante integrados simples SSI y MSI, por lo que se ha decidido utilizar circuitos integrados MOS\_LSI que realizan todas estas funciones.

## CODIFICACION POR MICROPROCESADORES

Una técnica que se está expandiendo fuertemente es el uso de microprocesadores como elementos de lógica activa en la codificación de teclados. Realiza varias tareas, que ayudan a disminuir el trabajo al microprocesador del sistema. Esta técnica permite con un mínimo de componentes, realizar funciones que hasta el presente raramente eran llevadas por un periférico. Entre estas funciones cabe mencionar: exploración secuencial, protección contra pulsaciones simultaneas, codificaciones, modos múltiples salidas en paralelo o en serie, autorepetición, detección de errores, etc.

### TECLADOS DE LAS PC' s

Los teclados para las PC's son de los más completos y funcionales, ya que en su mayoría poseen un microprocesador para uso específico, no es necesario una decodificación de las señales que se reciben y se puede conectar solo a través de una interfaz. Además hay una gran variedad de estos teclados en el mercado por lo que se facilita su adquisición. Son estas las razones por las que hemos optado por utilizar un teclado de este tipo para nuestro grabador digital de audio, quizás no todas las teclas que posee el teclado sean utilizadas, aunque el sistema será capaz de captar títulos en una pantalla, pero el usar un teclado sin estas características nos llevaría a realizar un extensivo trabajo de codificación y a otros problemas que pueden evitarse y no son de gran interés para nuestro objetivo final.

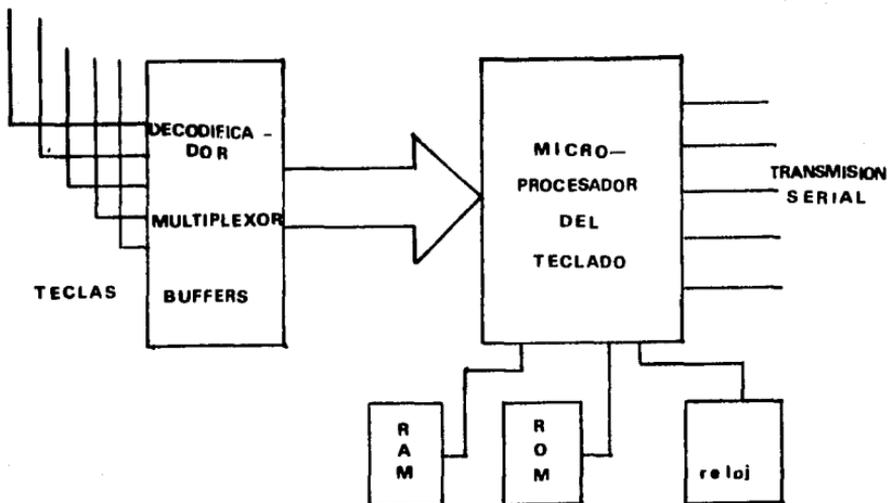


### CARACTERISTICAS GENERALES

Un teclado está constituido por un microprocesador el cual contiene dentro del mismo circuito un puerto serie, una memoria RAM, una memoria ROM y un puerto paralelo y su único propósito es el de estar explorando y verificando el estado de las teclas, obtener el código de la tecla y poder transmitir dicho código. Cabe señalar que los datos del teclado son enviados a una velocidad de 1200 bauds.

Otra parte del teclado es la interfaz de comunicación serial bidireccional, compuesta de 5 líneas de conexión, la cual es empleada para enviar señales entre el usuario y el sistema.

El teclado posee también un buffer de 16 caracteres first-in-first-out (FIFO), en donde se almacenan los códigos cuando el teclado no está siendo atendido por el sistema y hasta que la interfaz esté lista para recibirlos.



#### ORGANIZACION INTERNA DEL TECLADO

En el teclado las teclas son clasificadas como "make-break", es decir que cuando una tecla es presionada el teclado envía un código make para cada tecla y un código break para cuando es soltada. La velocidad de transmisión a la cual es enviado el código make es conocida como velocidad de transmisión de escritura automática (typematic rate), y es por default de 10 caracteres por segundo con un retraso de 500 mseg. Dependiendo de la posición de la tecla se emite el código make, el cual puede representarse por la siguiente tabla:

LETRA	CODIGO	NUMERO	CODIGO	SIMBOLO	CODIGO
A	E1	1	FD	ESC	FE
B	CF	2	FC	FLECHA	B7
C	D1	3	FB	ARRIBA	
D	DF	4	FA	FLECHA	AF
E	ED	5	F9	ABAJO	
F	DE	6	F8	ENTER	E3
G	DD	7	F7	BLANCO	C6
H	DC	8	F6	.	CB
I	E8	9	F5	,	CC
J	DB	0	F4	;	DB
K	DA			<---	F1
L	D9				
M	CD				
N	CE				
N	D7				
O	E7				
P	E6				
Q	EF				
R	EC				
S	E0				
T	EB				
U	E9				
V	D0				
W	EE				
X	D2				
Y	EA				
Z	D3				

El teclado y el sistema se comunican a través de las líneas de reloj y datos. La fuente de cada una de estas es un dispositivo de colector abierto en el teclado, los cuales permiten que tanto el

sistema como el teclado forcen una línea a nivel negativo. Cuando no hay comunicación, las líneas de reloj y datos están a nivel positivo.

Una transmisión de datos del y hacia el teclado consiste de un flujo de datos de 11 bits, los cuales son enviados en serie por la línea de datos y estos pueden ser estructurados de la siguiente forma:

B<sub>1</sub> B<sub>2</sub> B<sub>3</sub> B<sub>4</sub> B<sub>5</sub> B<sub>6</sub> B<sub>7</sub> B<sub>8</sub> B<sub>9</sub> B<sub>10</sub> B<sub>11</sub>

BIT	FUNCION	BIT	FUNCION
1	0 BIT DE INICIO	7	BIT DE DATO 5
2	BIT DE DATO 0	8	BIT DE DATO 6
3	BIT DE DATO 1	9	BIT DE DATO 7
4	BIT DE DATO 2	10	BIT DE PARIDAD
5	BIT DE DATO 3	11	BIT DE PARADA
6	BIT DE DATO 4		

El bit de paridad es 1 ó 0, pero los bits de datos altos más el bit de paridad siempre deben ser un número non. Cuando el sistema envía datos al teclado, la línea de datos es forzada a nivel negativo y permite a la línea de reloj ir a nivel positivo. Cuando el teclado envía datos al sistema o los recibe de este, es generada la señal de reloj al tiempo de dato.

#### ENCENDIDO DEL TECLADO

La lógica del teclado genera un POR ( Power-On-Reset ) cuando se le aplica la alimentación, este puede durar un mínimo de 300 mili segundos y un máximo de 9 segundos. Después el teclado ejecuta una prueba de seguridad básica BAT (Basic Assurance Test). La cual consiste en verificar toda la ROM y un bit estaca (stuck-bit), además de hacer una prueba de direccionamiento. La ejecución del BAT puede tomar de 600 a 900 mseg y también puede

iniciarse con en comando del sistema llamado RESET. Después del BAT y cuando la interfaz está habilitada el teclado envía un código completo para indicar que la realización fue correcta.

#### SALIDA DE DATOS DEL TECLADO

Cuando el teclado está listo para enviar datos, primero chequea un estado de teclado inhibido o una petición de envío del sistema en las líneas de reloj y datos. Si la línea de reloj está baja (estado inhibido) el dato es almacenado en el buffer del teclado. Si la línea de reloj es alta y la de datos baja (petición de envío), el dato es almacenado en el buffer del teclado y recibe el dato proveniente del sistema.

Si tanto el reloj y datos son altos el teclado envía el bit 0 de inicio, los 8 bits de datos, el bit de paridad y el bit de parada. Los datos son válidos antes del corte o flanco de caída y después del flanco de subida de cada pulso de reloj. Durante la transmisión, el teclado chequea la línea de reloj para un posible nivel positivo al final de cada 60 mseg. Si el sistema baja la línea de reloj de un nivel positivo después de que el teclado inicio el envío de un dato, una condición conocida como contención de línea ocurre, y el teclado detiene el envío de datos. Si la contención de línea ocurre antes del flanco de subida del décimo ciclo de reloj (bit de paridad), el buffer del teclado retorna a las líneas de reloj y datos a nivel positivo. Si la contención no ocurre por el décimo ciclo de reloj, el teclado termina la transmisión.

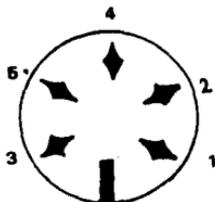
Después de una transmisión, el sistema puede inhibir el teclado hasta que el sistema procese el dato o hasta una petición de envío.

### CONECTOR DEL TECLADO

El teclado se conecta al sistema por medio de un cable con un conector DIN de cinco terminales:

PIN DEL CONECTOR	NOMBRE DE LA SEÑAL
1	RELOJ
2	DATOS
3	NO SE USA
4	TIERRA
5	5 Vpp

de las cuales solo conectaremos los pines 1, 2, 4 y 5.



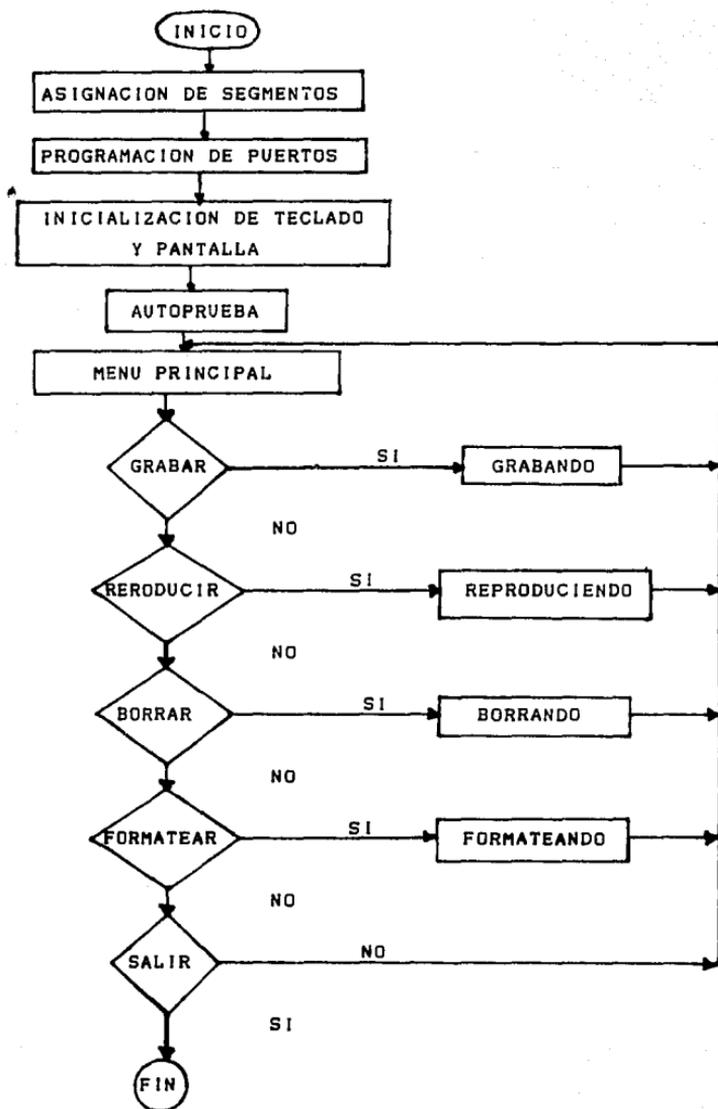
## II.3 DESARROLLO DEL SOFTWARE

El software desarrollado para el manejo del SGDA se realizó en lenguaje ensamblador Macro-Assembler para 8088. En el programa principal inicialmente se asignan las localidades en memoria para los segmentos DS, ES, SS. El segmento de código (CS) se localiza al inicio de la ROM y se asigna por medio de un salto inicial, el cual está escrito en la dirección 1FFF0 de la misma ROM y que corresponde a la dirección XFFF0 de nuestro sistema. A continuación se realiza la programación de los puertos del sistema y la inicialización de la pantalla y del teclado.

Con la finalidad de manejar adecuadamente la operación del sistema se realizaron subrutinas generales de soporte. En primera instancia se realiza una subrutina de inicialización de la pantalla y del teclado. En los dos dispositivos se envían comandos al puerto de comando con el fin de dejarlos en espera de datos.

Inmediatamente y con el fin de verificar la correcta operación del sistema se realiza una rutina de AUTOPRUEBA en donde se chequea la RAM escribiendo y verificando cada localidad con dos datos fijos complementarios: AAH y 55H, en seguida se verifica si la PC está trabajando en el programa de comunicación, en esta parte se envía el comando de no-acción (OAH) y se espera que la PC conteste con AOH.

Si la autoprueba finaliza con éxito se envía a pantalla un menú de opciones a elegir, en esta parte del programa se sensa el accionamiento de las flechas hacia arriba o hacia abajo, hasta que finalmente se elige un número para acceder a una subrutina. Si existiese algún error en el accionamiento de las teclas se llamará a una subrutina de BEEP. La parte correspondiente al programa principal se muestra a continuación:



```

datos segment 'data'
    cabeza db 00h
    pista db 00h
    sector db 00h
    cabeza_final db 00h
    pista_final db 00h
    sector_final db 00h
    numero db 00h
datos ends
;
extra segment 'extra'
    cancion db 00h
extra ends

;programación de puertos
ppa equ 00h
ppb equ 01h
ppc equ 02h
ppctrl equ 03h
pped equ 04h
ppsd equ 05h
ppci equ 06h
ppctrli equ 07h
psed equ 08h
psc equ 09h
pcp equ 0ah
pdp equ 0bh
;
stack_1 segment 'stack'
    dw 500 dup(00)
    stack_top label word
stack_1 ends
;
sgda segment 'code'
;inicialización de segmentos en memoria.
    assume cs:sgda,ds:datos,es:extra,ss:stack_1
    mov ax,04c0h
    mov ss,ax

```

```

mov ax,0000h
mov ds,ax
mov ax,0481h
mov es,ax
lea sp,stack_top
mov dx,ppctrl ;puerto paralelo de comunicación
mov al,8ah ;a=salida,b=entrada,cI=salida,ch=entrada
out dx,al
mov dx,offset inicio_pan_tec
call dx
inicio:mov al,0ah
mov dx,ppc
out dx,al ;manda a la pc no hacer nada
espa0:in al,dx ;espera respuesta de la pc
and al,0f0h
cmp al,0a0h
jnz espa0
mov dx,offset despliega_menu
call dx
cmp al,31h ;#1=grabar
jz grabar
cmp al,32h ;#2=reproducir
jz escuchar
cmp al,33h ;#3=borrar
jz borra
cmp al,34h ;#4=formatear
jz format
cmp al,35h ;#5=salir
jz salir
mov dx,offset beep
call dx
jmp inicio
grabar:mov dx,offset grabacion
call dx
jmp inicio
escuchar:mov dx,offset reproducir
call dx
jmp inicio

```

```

borra:mov dx,offset borrar
      call dx
      jmp inicio
format: mov bx,0f298h      ;'seguro deseas formatear?'
      mov dx,offset letrero
      call dx
      mov dx,offset lee_teclado
      call dx
      cmp al,53h          ;'s'
      jnz inicio
      mov dx,offset formato
      call dx
      jmp inicio
salir:mov dx,ppc
      mov al,0dh          ;comando de fin para la pc
      out dx,al
      mov bx,0fic0h      ;letrero 'no olvides apagarme'
      mov dx,offset letrero
      call dx
      hit

```

```

inicio_pan_tec proc near ;inicializa pantalla y
      pushf              ;puerto serie
      push ax
      push bx
      push cx
      push dx
;programación del puerto serie
      mov dx,psc
      mov al,00h
      out dx,al
      mov cx,02h
d0:   loop d0
      out dx,al
      mov cx,02h
d1:   loop d1
      out dx,al

```

```

    mov cx,02h
d2:  loop d2
    mov al,40h
    out dx,al
    mov cx,02h
d3:  loop d3
    mov al,01011101b
    out dx,al
    mov cx,02h
d4:  loop d4
    mov al,00010100b
    out dx,al

```

;programación de la pantalla

```

    mov cx,0cffh
esp1:loop esp1
    mov dx,pcp
    mov al,34h
    out dx,al
    mov cx,0cffh
esp2:loop esp2
    mov al,34h
    mov dx,pcp
    out dx,al
    mov cx,0cffh
esp3:loop esp3
    mov al,34h
    mov dx,pcp
    out dx,al
    mov cx,0cffh
esp4:loop esp4
    mov al,34h
    mov dx,pcp
    out dx,al
    mov cx,0cffh
esp5:loop esp5
    mov al,0ch
    mov dx,pcp

```

; para el manejo de la pantalla se siguieron los pasos indicados en el manual de operación de ésta. Al inicio se manda un tiempo de espera de aproximadamente 15mseg inmediatamente despues se envia un 34H que indica que se va a escribir en la pantalla y esto se va a realizar por medio de 8 bits. Se manda otro tiempo de espera de más de 4.1mseg y otro 34H, esta operación se repite dos veces más. Después se manda un 0CH para encender el display. Luego se manda un 01h para borrar la pantalla y para terminar la inicialización se manda un 03H.

```

    out dx,al
    mov cx,0cffh
esp6:loop esp6
    mov al,01h
    mov dx,pcp
    out dx,al
    mov bx,03h
etiqueta2:mov cx,0cffh
esp7:loop esp7
    dec bx
    cmp bx,00h
    jnz etiqueta2
    mov al,06h           ; se enciende la pantalla
    mov dx,pcp
    out dx,al
    mov bx,03h
etiqueta:mov cx,0cffh
esp8:loop esp8
    dec bx
    cmp bx,00h
    jnz etiqueta
    mov dx,pcp
    mov al,0eh           ; enciende pantalla sin cursor
    out dx,al
    mov cx,0cffh
espera1:loop espera1
    mov al,01h           ; se borra la pantalla
    mov dx,pcp
    out dx,al
    mov bx,03h
etiqueta3:mov cx,0cffh
espera2:loop espera2
    dec bx
    cmp bx,00h
    jnz etiqueta3
    mov al,80h           ; aparece cursor
    mov dx,pcp
    out dx,al

```

```

    mov cx,0cffh
espera3:loop espera3
    mov dx,offset autopruoba
    call dx
    pop dx
    pop cx
    pop bx
    pop ax
    popf
    ret
inicio_pan_tec endp

```

```

autopruoba proc near

```

```

    pushf
    push ax
    push bx
    push cx
    push dx
    push si
    mov si,0000h           ;prueba de RAM
    mov cx,7fffh

```

```

auto_aa:mov al,(si)

```

```

    mov ah,al
    mov al,0aah
    mov (si),al
    mov al,00h
    mov al,(si)
    cmp al,0aah
    jnz error_ram
    mov al,ah
    mov (si),al
    inc si
    loop auto_aa
    mov si,0000h
    mov cx,7fffh

```

```

auto_55:mov al,(si)

```

```

    mov ah,al
    mov al,55h

```

```

mov (si),al
mov al,00h
mov al,(si)
cmp al,55h
jnz error_ram
mov al,ah
mov (si),al
inc si
loop auto_55
auto_esp0:mov al,0ah
mov dx,ppc
out dx,al ;manda a la PC no hacer nada
in al,dx ;espera respuesta de la pc
and al,0f0h
cmp al,0a0h
jz auto_fin
mov bx,0f328h
mov dx,offset letrero ;letrero "la pc no esta lista"
call dx
jmp auto_esp0
error_ram:mov bx,0f340h ;letrero "error en ram"
mov dx,offset letrero
call dx
auto_fin:pop si
pop dx
pop cx
pop bx
pop ax
popf
ret
autoprueba endp

despliega_menu proc near
pushf
push bx
reinicio:mov bx,0f100h
mov dx,offset letrero
call dx

```

```

flecha_arriba:mov dx,offset lee_teclado
               call dx
               cmp al,0e0h           ;e0=flecha arriba
               jz flecha_arriba
               cmp al,0e2h           ;e2=flecha abajo
               jnz ruido
               add bx,00h
               mov dx,offset letrero
               call dx
tecla_equivocada:mov dx,offset lee_teclado
                  call dx
                  cmp al,0e0h
                  jz decremento
                  cmp al,0e2h
                  jz incremento
                  jmp ruido1
decremento:sub bx,30h              ;regresa al letrero anterior
             cmp bx,0f100h         ;si es el primero vuelve a empezar
             jz reinicio
             mov dx,offset letrero
             call dx
             jmp tecla_equivocada
incremento:cmp bx,0f1c0h          ;si es el último no avanza adelante
            jz tecla_equivocada
            add bx,00h
            mov dx,offset letrero
            call dx
            jmp tecla_equivocada
ruido1:cmp bx,0f1c0h
        jz fin_menu               ;si es el último letrero y se oprime
        mov dx,offset beep
        call dx                    ;otra tecla (no flechas) se elige en AL
        jmp tecla_equivocada
ruido:mov dx,offset beep
        call dx                    ;si se oprime una tecla no válida
        jmp flecha_arriba
fin_menu:pop bx
         popf

```

```
ret
despliega_menu endp
```

Para el manejo del teclado se realizó una subrutina de lectura. Así se inicializa el puerto serie para evitar lecturas erróneas, posteriormente se checa la bandera bit 2 en el puerto de control si está en uno, entonces hay un dato proveniente del teclado esperando en el buffer, se efectúa la lectura del dato pero solo en once bits ya que una tecla envía dos códigos, uno al oprimirse y otro al soltarse, así en nuestro sistema se lee solo un código y se traduce de acuerdo a una tabla almacenada en memoria este código se cambia por su correspondiente código para la pantalla. Esto se logra sumando el valor leído con un valor inicial (FO0H), el cual corresponde al inicio de la tabla de traducción, de tal forma que leyendo la localidad en memoria que corresponde a la suma de valores se obtiene un código ya traducido.

```
lee_teclado proc near      ; espera llegada de dato por teclado
    pushf                 ; en al regresa el código de la pantalla
    push bx                ; de la tecla oprimida
    push cx
    push dx
    push si
    mov dx,psc
    mov al,00h
    out dx,al
    mov cx,02h
dd0: loop dd0
    out dx,al
    mov cx,02h
dd1: loop dd1
    out dx,al
    mov cx,02h
dd2: loop dd2
    mov al,40h
    out dx,al
    mov cx,02h
dd3: loop dd3
```

```

mov al,01011101b
out dx,al
mov cx,02h
dd4: loop dd4
mov al,00010100b
out dx,al
verifica:mov dx,psc
in al,dx
and al,02h
jz verifica
mov dx,psed
in al,dx
mov cancion,al
mov bx,02h
wait2:mov cl,00h
mov ch,00h
wait1:loop wait1
dec bx
cmp bx,00h
jnz wait2
mov al,cancion
mov si,0f000h ;dirección 1000 de la eprom
mov ah,00h
add si,ax
mov al,(si) ;AL tiene el código de la pantalla
pop si ;de la tecla oprimida
pop dx
pop cx
pop bx
popf
ret
lee_teclado endp

```

Para el manejo de la pantalla de visualización se realizó una subrutina que permite la escritura de toda la pantalla a la vez, con letreros ya escritos en memoria. La cual actua de la siguiente forma: se asigna en Bx la dirección inicial del letrero en el

segmento de datos y se envían los 24 caracteres que corresponden al letrero correspondiente. Al término de la subrutina se regresa Bx con un valor que apunta a la dirección de inicio del siguiente letrero permitiendo enviar letreros en secuencia ya que así lo exige la subrutina del menú principal.

```
letrero proc near          ;bx tiene la dirección en memoria
    pushf                 ;donde comienza el letrero de
    push dx                ;24 caracteres
    push ax
    push cx
    mov dx,pcp
    mov al,80h
    out dx,al
    mov cx,0cffh
attend:loop attend
    mov cx,18h
caracteres:mov al,(bx)
    mov dx,pdp
    out dx,al
    inc bx
    mov dx,0cffh
pausa: dec dx
    cmp dx,00h
    jnz pausa
    loop caracteres
    pop cx
    pop ax
    pop dx
    popf
    ret
letrero endp
```

Por otro lado podemos hablar de las subrutinas auxiliares que intervienen dentro de las subrutinas principales. Por ejemplo:

Para el acceso de la información de audio al sistema, se

creó una subrutina de manejo de interface llamada BUFFER. En ésta se envía un pulso de START al convertidor A/D y se espera el flanco positivo de la señal EOC que indica el fin de conversión A/D. Una vez leído el dato por el puerto paralelo de entrada de datos se manda éste al convertidor D/A a través del puerto paralelo de salida de datos y al mismo tiempo se guarda en el acumulador AL.

```

buffer proc near
    pushf
    push dx
    mov dx,ppci
    mov al,10h
    out dx,al           ;envia pulso de 'start'
    mov al,00h
    out dx,al
esp_dato_grab:mov dx,ppci
    in al,dx
    rcr al,01h
    jc esp_dato_grab   ;espera bajada del 'eoc'
esp_dato_grab1:in al,dx
    rcr al,01h
    jnc esp_dato_grab1 ;espera subida del 'eoc'
    mov dx,pped        ;lee dato del convertidor A/D
    in al,dx
    mov dx,ppsd        ;envia dato al convertidor D/A
    out dx,al
    pop dx
    popf
    ret
buffer endp

```

Otro auxiliar en la subrutina de reproducción es una rutina parecida a la del BUFFER pero la diferencia radica en que solamente reproduce, es decir manda al puerto paralelo de salida de datos el dato que se encuentre almacenado en AL sin hacer ninguna lectura proveniente del convertidor A/D. Esta subrutina se

llama TOCA.

```
toca proc near
    pushf
    push dx
    push ax
    mov ah,al
    mov dx,ppci
    mov al,10h
    out dx,al           ;envia pulso de 'start'
    mov al,00h
    out dx,al
esp_dato_toca:mov dx,ppci
    in al,dx
    rcr al,01h
    jc esp_dato_toca   ;espera bajada de 'eoc'
esp_dato_tocal:in al,dx
    rcr al,01h
    jnc esp_dato_tocal ;espera subida de 'eoc'
    mov al,ah
    mov dx,ppsd        ;sale dato por interface
    out dx,al
    pop ax
    pop dx
    popf
    ret
toca endp
```

Finalmente para indicar al operador alguna situación de error o aviso, se creó la rutina BEEP. Esta hace uso de la interfaz de igual manera que la rutina TOCA, sin embargo esta hace un loop enviando datos altos y bajos en una frecuencia fija provocando a la salida un tono agudo de duración aproximada de un segundo.

```
beep proc near
    pushf
    push ax
```

```

push bx
push cx
push dx
mov dx,ppctri1
mov al,91h
out dx,al
mov bx,00FFh
mas:mov cx,0Ah
tiempo:mov dx,ppci
mov al,10h
out dx,al
mov al,00h
out dx,al
esp_dato_beep:mov dx,ppci
in al,dx
rcr al,01h
jc esp_dato_beep
esp_dato_beep1:in al,dx
rcr al,01h
jnc esp_dato_beep1
mov dx,ppsd
mov al,3fh
out dx,al
loop tiempo
mov cx,0Ah
tiempo1:mov dx,ppci
mov al,10h
out dx,al
mov al,00h
out dx,al
esp_dato_beep2:mov dx,ppci
in al,dx
rcr al,01h
jc esp_dato_beep2
esp_dato_beep3:in al,dx
rcr al,01h
jnc esp_dato_beep3
mov dx,ppsd

```

```

mov al,Obfh
out dx,al
loop tiempo1
dec bx
cmp bx,00h
jnz mas
pop dx
pop cx
pop bx
pop ax
popf
ret
beep endp

```

Dentro de las subrutinas principales está la rutina de GRABACION de audio, la cual se realiza en dos ciclos principales, primero se hace una transferencia de datos leídos del convertidor A/D hacia la RAM de la PC mediante los comandos de comunicación ya escritos en ésta. Una vez llenado el buffer de 360k o cuando se da la orden de finalizar la grabación (al oprimir cualquier tecla) se inicia el ciclo de grabación a disco. El sistema envía a la PC las localidades en disco a donde debe grabar la información de audio que se le envíe. La grabación en disco se manda por pista para que se efectue en el menor tiempo posible. En caso de error en la grabación, la PC avisa al sistema, el cual a su vez detiene la operación y previene al operador. Si toda la transferencia concluye con éxito, el sistema manda a pantalla un mensaje y un beep. Esta rutina se muestra a continuación:

```

grabacion proc near
    pushf
    push ax
    push bx
    push cx
    push dx
    call dx
    mov dx,ppctrl1
    mov al,91h

```

```

out dx,al
mov bx,0f2e0h
mov dx,offset letrero ;letrero 'grabando'
call dx
mov al,08h
mov dx,ppc
out dx,al ;comando a la pc de escribir (captura de
; datos)

g_esp80:in al,dx
and al,0f0h
cmp al,80h
jnz g_esp80
mov cabeza,00h

g_incca:mov pista,00h ;inicia transferencia de todo el disco
g_incpis:mov sector,01h
g_incsec:mov bx,00h
g_ciclo:mov dx,offset buffer
call dx
mov dx,ppa ;sale dato par a la pc
out dx,al
mov dx,ppc
mov al,0eh
out dx,al

g_espe0:in al,dx
and al,0f0h
cmp al,0e0h
jnz g_espe0
inc bx
mov dx,offset buffer
call dx
mov dx,ppa
out dx,al ;sale dato impar a la pc
mov dx,ppc
mov al,0fh
out dx,al

g_esp0:in al,dx
and al,0f0h
cmp al,0f0h

```

```

jnz g_espfo
inc bx
cmp bx,200h
jnz g_ciclo
mov dx,psc
in al,dx
and al,02h
jnz g_fin_datos ;si se oprime una tecla se detiene
inc sector
cmp sector,0ah
jnz g_incsec
inc pista
cmp pista,28h
jz g_cambia_cabeza
jmp g_incpis
g_cambia_cabeza:inc cabeza
cmp cabeza,02h
jnz g_incca
g_fin_datos:mov al,05h
mov dx,ppc
out dx,al ;fin de envio de datos
g_esp50:in al,dx
and al,0f0h
cmp al,50h
jnz g_esp50
mov al,09h
mov sector_final,al ;guarda dirección final de grabación
mov al,pista
inc al
mov pista_final,al
mov al,cabeza
mov cabeza_final,al
mov cabeza,00h
mov dx,ppc
mov al,08h
out dx,al
g_esp80a:in al,dx
and al,0f0h

```

```

    cmp al,080h
    jnz g_esp80a
g_inccal:mov pista,00h      ;inicia grabación de todo el disco
g_inccpisl:mov sector,01h
    mov al,cabeza_final    ;checa si termina la información
    cmp al,cabeza
    jnz g_sec_valido
    mov al,pista_final
    cmp al,pista
    jnz g_sec_valido
    jmp g_finb
g_sec_valido:mov dx,ppa
    mov al,cabeza          ;envia dirección de grabación
    out dx,al
    mov dx,ppc
    mov al,01h
    out dx,al
g_esp10:in al,dx
    and al,0f0h
    cmp al,10h
    jnz g_esp10
    mov dx,ppa
    mov al,pista
    out dx,al
    mov dx,ppc
    mov al,02h
    out dx,al
g_esp20:in al,dx
    and al,0f0h
    cmp al,20h
    jnz g_esp20
    mov dx,ppa
    mov al,sector
    out dx,al
    mov dx,ppc
    mov al,03h
    out dx,al
    jmp g_esp30

```

```

g_inccalb: jmp g_inccal
g_incpisib: jmp g_incpisi
g_finb: jmp g_fin
g_esp30: in al, dx
        and al, 0f0h
        cmp al, 30h
        jnz g_esp30
        mov al, 09h           ; 9 sectores a escribir
        mov dx, ppa
        out dx, al
        mov dx, ppc
        mov al, 09h
        out dx, al

g_esp90: in al, dx
        and al, 0f0h
        cmp al, 90h
        jnz g_esp90

g_esp6070: in al, dx
        and al, 0f0h
        cmp al, 70h
        jz g_continua
        cmp al, 60h
        jz g_error
        jmp g_esp6070

g_continua: inc pista
        cmp pista, 28h
        jz g_cambia_cabeza1
        jmp g_incpisib

g_cambia_cabeza1: inc cabeza
        cmp cabeza, 02h
        jnz g_inccalb
        jmp g_fin

g_error: mov bx, 0f2b0h      ; letrero 'error en disco'
        mov dx, offset letrero
        call dx
        mov dx, offset beep
        call dx
        mov al, 05h

```

```

    mov dx,ppc
    out dx,al
g_esp50b:in al,dx
    and al,0f0h
    cmp al,50h
    jnz g_esp50b
    jmp g_final
g_fin:nov al,05h
    mov dx,ppc
    out dx,al           ;fin de grabación de datos
g_esp50a:in al,dx
    and al,0f0h
    cmp al,50h
    jnz g_esp50a
    mov bx,0f2c8h       ;letrero 'canción grabada'
    mov dx,offset letrero
    call dx
    mov dx,offset beep
    call dx
g_final:pop dx
    pop cx
    pop bx
    pop ax
    popf
    ret
grabacion endp

```

La reproducción de audio se efectúa cuando el disco ya está grabado y también se realiza en dos ciclos. En el primero se mandan los comandos de reproducción a la PC para que la información sea bajada a la RAM, y en caso de error en la lectura de datos el sistema detiene la operación y avisa al operador. Al término de este ciclo el sistema manda los comandos de transferencia de datos de la RAM de la PC al sistema el cual manda los datos recibidos a la interfaz de salida de audio.

Si el operador desea detener la operación basta con solo oprimir una tecla. Al final de la reproducción se escucha un beep.

y un letrero se envia a la pantalla.

reproducir proc near

```
    pushf
    push ax
    push bx
    push cx
    push dx
    mov bx,0f2f8h
    mov dx,offset letrero ;letrero 'reproduciendo'
    call dx
    mov dx,ppctrli
    mov al,91h
    out dx,al
    mov dx,ppc
    mov al,0ch                ;comando de lectura a la pc (lee disco
                                ; completo)

    out dx,al
r_esp0:in al,dx
    and al,0f0h
    cmp al,0c0h
    jnz r_esp0
    mov cabeza,00h
r_incca:mov pista,00h
r_incpis:mov sector,01h
    mov dx,ppa                ;envia dirección de lectura
    mov al,cabeza
    out dx,al
    mov dx,ppc
    mov al,01h
    out dx,al
r_esp10:in al,dx
    and al,0f0h
    cmp al,10h
    jnz r_esp10
    mov dx,ppa
    mov al,pista
```

```
out dx,al
mov dx,ppc
mov al,02h
out dx,al
jmp r_esp20
r_inccab: jmp r_incca
r_incpisb: jmp r_incpis
r_esp20: in al,dx
and al,0f0h
cmp al,20h
jnz r_esp20
mov dx,ppa
mov al,sector
out dx,al
mov dx,ppc
mov al,03h
out dx,al
r_esp30: in al,dx
and al,0f0h
cmp al,30h
jnz r_esp30
mov al,09h
mov dx,ppa
out dx,al
mov dx,ppc
mov al,09h
out dx,al
r_esp90: in al,dx
and al,0f0h
cmp al,90h
jnz r_esp90
r_esp6070: in al,dx
and al,0f0h
cmp al,70h
jz r_continua
cmp al,60h
jnz r_esp6070
jmp r_errorb
```

;9 sectores a leer

```

r_continua:inc pista
            cmp pista,28h
            jz r_cambia_cabeza
            jmp r_incpisb
r_cambia_cabeza:inc cabeza
            cmp cabeza,02h
            jnz r_inccab
            ;comienza transferencia de datos
            mov dx,ppc
            mov al,05h
            out dx,al
r_esp50a:in al,dx
            and al,0f0h
            cmp al,50h
            jnz r_esp50a
            mov cabeza,00h
r_inccal:mov pista,00h
r_incpisi:mov sector,01h
r_incsecl:mov bx,00h
r_ciclo:mov dx,ppc
r_espe0:in al,dx
            and al,0f0h
            cmp al,0e0h
            jnz r_espe0
            mov dx,ppb
            in al,dx                ; lee dato par de la pc
            mov dx,offset toca
            call dx
            mov dx,ppc
            mov al,0eh
            out dx,al
            inc bx
            jmp r_espf0
r_errorb:jmp r_error
r_espf0:in al,dx
            and al,0f0h
            cmp al,0f0h
            jnz r_espf0

```

```

mov dx,ppb           ;lee dato impar de la pc
in al,dx
mov dx,offset toca
call dx
mov dx,ppc
mov al,0fh
out dx,al
inc bx
cmp bx,0200h
jnz r_ciclo
mov dx,ppc
mov dx,psc           ;si se oprime una tecla se detiene
in al,dx
and al,02h
jnz r_fin
inc sector
cmp sector,0ah
jnz r_incsecl
inc pista
cmp pista,28h
jz r_cambia_cabezal
jmp r_incpisl
r_cambia_cabezal:inc cabeza
cmp cabeza,02h
jnz r_inccal
jmp r_fin
r_error:mov bx,0f2b0h ;letrero 'error en disco'
mov dx,offset letrero
call dx
mov dx,offset beep
call dx
r_fin:mov dx,ppc
mov al,0Ch
out dx,al
r_espC0b:in al,dx
and al,0f0h
cmp al,0C0h
jnz r_espC0b

```

```

pop dx
pop cx
pop bx
pop ax
popf
ret
reproducir endp

```

Para poder hacer uso de un floppy en el SGDA es necesario tener un formato especial. En esta subrutina existe un paso antes de ser efectuada, que es el preguntar si realmente se esta seguro de formatear. La rutina da comienzo cuando se recibe el codigo de la tecla 'S', la cual consiste en enviar del sistema a la PC los comandos necesarios para efectuar el formateo de todo el disco por pista. El formato que se utilizó fue el siguiente:

```

2 lados
39 pistas por lado
9 sectores por pista
512 bytes por sector.

```

Al igual que en las rutinas anteriores, si existe error el proceso se detendrá.

```

formato proc near
    pushf
    push ax
    push bx
    push cx
    push dx
    mov bx,0f310h        ;letrero 'formateando'
    mov dx,offset letrero
    call dx
    mov cabeza,00h
f_incca3:mov pista,00h
f_incpis3:mov sector,01h
    mov al,04h          ;comando de formatear por pista
    mov dx,ppc

```

```

    out dx,al
f_esp40:in al,dx
    and al,0f0h
    cmp al,40h
    jnz f_esp40      ;espera respuesta listo para formatear
    mov dx,ppa
    mov al,cabeza
    out dx,al
    mov dx,ppc
    mov al,01h
    out dx,al
f_esp10:in al,dx
    and al,0f0h
    cmp al,10h
    jnz f_esp10
    mov dx,ppa
    mov al,pista
    out dx,al
    mov dx,ppc
    mov al,02h
    out dx,al
f_esp20:in al,dx
    and al,0f0h
    cmp al,20h
    jnz f_esp20
    mov dx,ppa
    mov al,sector
    out dx,al
    mov dx,ppc
    mov al,03h
    out dx,al
f_esp30:in al,dx
    and al,0f0h
    cmp al,30h
    jnz f_esp30
    jmp f_ciclo
f_incpis2:jmp f_incpis3
f_incca2:jmp f_incca3

```

```

f_ciclo:mov al,pista      ;envia tabla c,h,r,n
        mov dx,ppa
        out dx,al
        mov dx,ppc
        mov al,0eh
        out dx,al
f_espe0:in al,dx
        and al,0f0h
        cmp al,0e0h
        jnz f_espe0
        mov al,cabeza
        mov dx,ppa
        out dx,al
        mov dx,ppc
        mov al,0fh
        out dx,al      ;dato par valido
f_espf0:in al,dx
        and al,0f0h
        cmp al,0f0h
        jnz f_espf0
        mov al,sector
        mov dx,ppa
        out dx,al
        mov dx,ppc
        mov al,0eh
        out dx,al      ;dato par valido
f_espe01:in al,dx
        and al,0f0h
        cmp al,0e0h
        jnz f_espe01
        mov al,02h
        mov dx,ppa
        out dx,al
        mov dx,ppc
        mov al,0fh
        out dx,al
        jmp f_espf01
f_incpis1:jmp f_incpis2

```

```

f_inccal: jmp f_incca2
f_espf01: in al, dx
    and al, 0f0h
    cmp al, 0f0h
    jnz f_espf01
    inc sector
    cmp sector, 0ah
    jnz f_ciclo
    mov al, 05h
    mov dx, ppc
    out dx, al ;fin de envio de datos
f_esp50: in al, dx
    and al, 0f0h
    cmp al, 50h
    jnz f_esp50
f_esp6070: in al, dx
    and al, 0f0h
    cmp al, 70h
    jz f_continua
    cmp al, 60h
    jz f_error
    jmp f_esp6070
f_continua: inc pista
    cmp pista, 28h
    jz f_cambia_cabeza
    jmp f_incopisi
f_cambia_cabeza: inc cabeza
    cmp cabeza, 02h
    jnz f_final
    mov bx, 0fid8h ;!letrero 'disco formateado'
    mov dx, offset letrero
    call dx
    mov dx, offset beep
    call dx
    jmp f_fin
f_final: jmp f_inccal
f_error: mov bx, 0f2b0h ;!letrero 'error en disco'
    mov dx, offset letrero

```

```

call dx
mov dx,offset beep
call dx
f_fin:pop dx
pop cx
pop bx
pop ax
popf
ret
formato endp

```

Si se desea borrar la información de un disco se accesa a la subrutina de BORRAR la cual hace básicamente lo mismo que la de GRABAR pero en lugar de grabar información de audio graba FFH y así se elimina la información anterior.

```

borrar proc near
    pushf
    push ax
    push bx
    push cx
    push dx
    mov dx,ppotrii
    mov al,91h
    out dx,al
    mov bx,0f370h
    mov dx,offset letrero ;letrero 'borrando'
    call dx
    mov al,08h
    mov dx,ppc
    out dx,al
                                ;comando a la pc de escribir (captura de
                                ; datos)
b_esp80:in al,dx
    and al,0f0h
    cmp al,80h
    jnz b_esp80
    mov cabeza,00h
b_incoa:mov pista,00h
                                ;inicia transferencia de todo el disco

```

```

b_incpis:mov sector,01h
b_incsec:mov bx,00h
b_ciclo:mov al,0ffh
    mov dx,ppa                ;sale dato par a la po
    out dx,al
    mov dx,ppc
    mov al,0eah
    out dx,al
b_espe0:in al,dx
    and al,0f0h
    cmp al,0e0h
    jnz b_espe0
    inc bx
    mov al,0ffh
    mov dx,ppa
    out dx,al                ;sale dato impar a la pc
    mov dx,ppc
    mov al,0fh
    out dx,al
b_esp0:in al,dx
    and al,0f0h
    cmp al,0f0h
    jnz b_esp0
    inc bx
    cmp bx,200h
    jnz b_ciclo
    mov dx,psc
    in al,dx
    and al,02h
    jnz b_fin_datos         ;si se oprime una tecla se detiene
    inc sector
    cmp sector,0ah
    jnz b_incsec
    inc pista
    cmp pista,28h
    jz b_cambia_cabeza
    jmp b_incpis
b_cambia_cabeza:inc cabeza

```

```

    cmp cabeza,02h
    jnz b_incca
b_fin_datos:mov al,05h
    mov dx,ppc
    out dx,al                ;fin de envio de datos
b_esp50:in al,dx
    and al,0f0h
    cmp al,50h
    jnz b_esp50
    mov al,09h
    mov sector_final,al    ;guarda dirección final de grabación
    mov al,pista
    inc al
    mov pista_final,al
    mov al,cabeza
    mov cabeza_final,al
    mov cabeza,00h
    mov dx,ppc
    mov al,08h
    out dx,al
b_esp80a:in al,dx
    and al,0f0h
    cmp al,080h
    jnz b_esp80a
b_inccal:mov pista,00h    ;inicia grabación de todo el disco
b_incpisi:mov sector,01h
    mov al,cabeza_final    ;checa si termina la información
    cmp al,cabeza
    jnz b_sec_valido
    mov al,pista_final
    cmp al,pista
    jnz b_sec_valido
    jmp b_finb
b_sec_valido:mov dx,ppa
    mov al,cabeza          ;envia dirección de grabación
    out dx,al
    mov dx,ppc
    mov al,01h

```

```

    out dx,al
b_esp10:in al,dx
    and al,0f0h
    cmp al,10h
    jnz b_esp10
    mov dx,ppa
    mov al,pista
    out dx,al
    mov dx,ppc
    mov al,02h
    out dx,al
b_esp20:in al,dx
    and al,0f0h
    cmp al,20h
    jnz b_esp20
    mov dx,ppa
    mov al,sector
    out dx,al
    mov dx,ppc
    mov al,03h
    out dx,al
    jmp b_esp30
b_inccalb:jmp b_inccal
b_incpislb:jmp b_incpisl
b_finb:jmp b_fin
b_esp30:in al,dx
    and al,0f0h
    cmp al,30h
    jnz b_esp30
    mov al,09h
    ;9 sectores a escribir
    mov dx,ppa
    out dx,al
    mov dx,ppc
    mov al,09h
    out dx,al
b_esp90:in al,dx
    and al,0f0h
    cmp al,90h

```

```

    jnz b_esp90
b_esp6070:in al,dx
    and al,0f0h
    cmp al,70h
    jz b_continua
    cmp ai,60h
    jz b_error
    jmp b_esp6070
b_continua:inc pista
    cmp pista,28h
    jz b_cambia_cabezal
    jmp b_incpis1b
b_cambia_cabezal:inc cabeza
    cmp cabeza,02h
    jnz b_incca1b
    jmp b_fin
b_error:mov bx,0f2b0h    ;letrero 'error en disco'
    mov dx,offset letrero
    call dx
    mov dx,offset beep
    call dx
    mov al,05h
    mov dx,ppc
    out dx,al
b_esp50b:in al,dx
    and al,0f0h
    cmp al,50h
    jnz b_esp50b
    jmp b_final
b_fin:mov al,05h
    mov dx,ppc
    out dx,al    ;fin de grabación de datos
b_esp50a:in al,dx
    and al,0f0h
    cmp al,50h
    jnz b_esp50a
    mov bx,0f268h    ;letrero 'canción borrada'
    mov dx,offset letrero

```

```
call dx
mov dx,offset bsep
call dx
b_final:pop dx
pop cx
pop bx
pop ax
popf
ret
bofrar endp
;
sgda ends
end
```

Para salir del programa existe una opción en el menú principal la cual envía el comando de fin de ejecución a la PC, manda a pantalla el mensaje "No olvides apagarne" e interrumpe al procesador mediante un Halt.

**CAPITULO III**  
**DISEÑO Y CONSTRUCCION**

### III. SISTEMAS DE VISUALIZACION

Los visualizadores, cualquiera que sea la tecnología con que estén contruidos, deben satisfacer una de las necesidades humanas básicas, que es la transferencia de información en forma reconocible, entre el operador y la máquina.

Existen muchos equipos que necesitan mostrar letras y números para darle al usuario direcciones o datos numéricos. En sistemas en los que se despliega una gran cantidad de información se utiliza generalmente un tubo de rayos catódicos (TRC), mientras que otros sistemas se pueden utilizar visualizadores de un dígito o más, de tipo matricial u otro tipo de arreglo.

#### TIPOS DE VISUALIZADORES

En términos generales, existen cuatro tecnologías con las cuales se fabrican la mayor parte de los visualizadores, y son, en orden de importancia y frecuencia de utilización, tubos de rayos catódicos, pantallas de cristal líquido, arreglos de diodos emisores de luz y arreglos de lámparas fluorescentes.

Los tubos de rayos catódicos (TRC) son los más utilizados cuando se trata de grandes volúmenes de información. El TRC está formado por una pantalla recubierta de fósforo sobre la cual incide un haz de electrones modulado en intensidad, y que recorre toda la pantalla debido al efecto de dos pares de placas deflectoras, que desvían el haz de acuerdo a los potenciales aplicados a ellas.

El TRC es fácil de interconectar con sistemas electrónicos, tales como microprocesadores, y los hay en diversas variedades y tamaños, que van desde pantallas superiores a 30" hasta miniaturas de tan solo 0.5", medidas diagonalmente. Existen también circuitos integrados diseñados especialmente para controlar este tipo de visualizadores. Sin embargo, su uso en equipos portátiles se ve

limitado por su consumo de energía, relativamente alto si se le compara con otros equipos de distinta tecnología.

Las pantallas de cristal líquido se construyen confinando una capa delgada de cristal líquido entre dos piezas de vidrio. Se coloca una película conductiva transparente en la pieza trasera (plano posterior) y en la pieza frontal (plano anterior) se colocan secciones de película conductiva con la forma de los caracteres deseados. Cuando se aplica un voltaje entre un segmento y el plano posterior se crea un campo eléctrico en la región bajo el segmento. Este campo eléctrico cambia los índices de reflexión y refracción en la región bajo la película del segmento. Hay dos tipos comunes de pantallas de cristal líquido. El primero, de dispersión dinámica, desordena las moléculas donde el campo esté presente. Esto produce una apariencia de cristal tintado sobre un fondo oscuro. El segundo tipo, de efecto de campo, utiliza polarización para absorber luz donde esté presente el campo, lo que da por resultado caracteres oscuros sobre un fondo color gris-plata.

Muchos de estos visualizadores requieren un voltaje de entre 2 y 3 volts entre el plano posterior y el segmento para encender éste. Debido a que los segmentos se dañan si se permite la aplicación continua de un voltaje superior a los 50mV, se les aplica una señal cuadrada de una frecuencia entre 30 y 150 Hz. Para esto se utilizan generalmente circuitos tipo CMOS.

La ventaja de las pantallas de cristal líquido consiste en su baja disipación de potencia, sin embargo presenta problemas con el contraste y al ángulo de visión.

Los diodos emisores de luz están disponibles en arreglos numéricos, hexadecimales y alfanuméricos, variando éstos desde arreglos de 7-segmentos a 18-segmentos hasta matrices de puntos de 5X7.

Un diodo emisor de luz consiste en una unión de material semiconductor que emite luz visible al hacer circular una corriente a través de él.

Los arreglos de lámparas fluorescentes han estado disponibles comercialmente desde los años 60's. De hecho, los visualizadores de

7-segmentos para las calculadoras estaban hechos con esta tecnología. La forma en que opera este dispositivo se basa en un filamento de metal calentado a alrededor de  $600^{\circ}\text{C}$  que emite electrones, los cuales son atraídos a través de una malla cubierta de fósforo hacia un ánodo. La mayoría de estos electrones que pasan a través de la malla golpean una capa de fósforo, consecuentemente emitiendo luz.

Las ventajas de esta tecnología con respecto a la de cristal líquido consiste en una mayor brillantez, un ángulo de visión más amplio y un rango de temperatura de operación mucho más extendido.

Los dos tipos de visualizadores que se podrían utilizar en nuestro diseño dadas sus características son los siguientes:

#### ARREGLOS DE DIODOS EMISORES DE LUZ

Los arreglos de diodos operan de la siguiente manera:

1. Modo estático. Cuando se tiene un visualizador de 7 segmentos (con el cual podemos indicar un código hexadecimal), se suele proporcionar la entrada con un código binario o un código BCD, el cual se decodifica para validar algunos de los segmentos según sea la entrada. Para esto se utiliza un circuito decodificador, junto con 7 resistencias necesarias para limitar la corriente que atraviesa los segmentos. A medida que se tiene un número mayor de dígitos crece tanto la potencia como la complejidad del circuito.

2. Modo multiplexado. Una forma de evitar los inconvenientes de complejidad y potencia consiste en colocar todos los dígitos en paralelo y seleccionar (activar) alternadamente cada uno de ellos. Con esto se puede llegar a manejar un número grande de dígitos con solo un decodificador y siete resistencias.

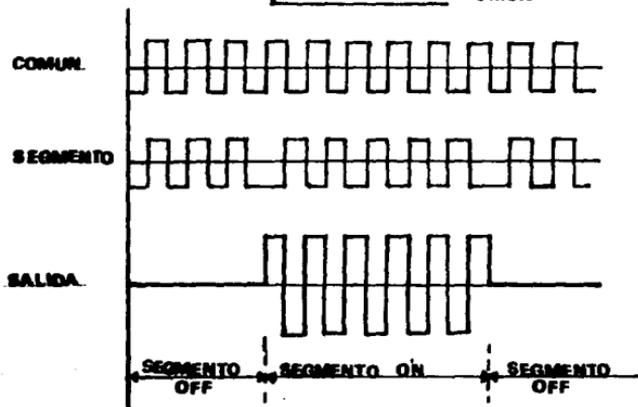
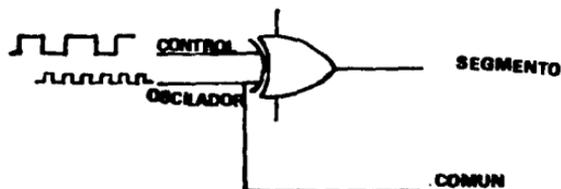
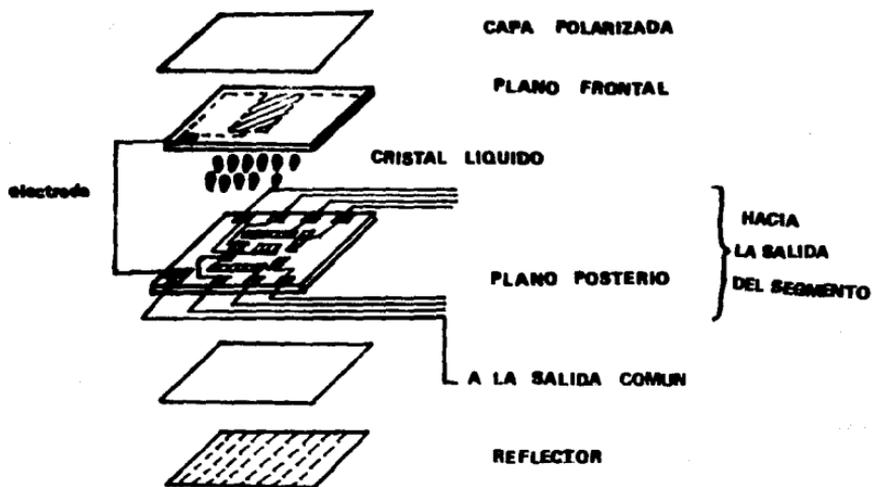
Este método es particularmente utilizado en pantallas

alfanuméricas de tipo matriz o arreglos de 18 segmentos. Aquí se conectan todos los elementos de cada fila en paralelo, y de manera similar todos los elementos de cada columna se conectan en paralelo. Para operar este arreglo se presenta el código de la primera columna en la entrada de las filas, y se activa dicha columna. Luego se presenta el código de segunda columna en la entrada de las filas y se activa la segunda columna, y se prosigue así, presentado el código de la columna *n*-ésima en la entrada de las líneas cuando la *n*-ésima columna es activada. Al llegar a la última columna se regresa a la primera (acción que se repite cíclicamente y se conoce como ciclo de refresco).

#### PANTALLAS DE CRISTAL LIQUIDO

En el caso de pantallas de cristal líquido, debido a su baja velocidad de respuesta, se requiere un método no multiplexado. La figura ilustra la forma de conectar un circuito CMOS para manejar un segmento y un plano posterior, así como las formas de onda del plano posterior y de los segmentos encendidos y apagados. (Los segmentos que permanecen apagados reciben la misma señal que el plano posterior, por lo que nunca se genera un campo en ellos).

La forma de onda para los segmentos encendidos está  $180^\circ$  defasado con respecto a la forma del segmento. Se observa que la lógica de esto es muy simple, puesto que basta con generar una onda cuadrada y su complemento.



Algunos visualizadores de cristal líquido encienden y apagan muy lentamente, por lo cual no se puede utilizar el esquema de multiplexión utilizado con arreglos de diodos o lámparas incandescentes. Para reducir el número de interconexiones se utiliza una técnica conocida como TRIPLEX, la cual consiste en tomar grupos de tres columnas con sus respectivas filas, y presentar el código de la primera columna en la entrada de las filas y activar la primera columna. Se prosigue sucesivamente con la segunda y la tercera, con lo que se tiene un esquema parecido a la multiplexión vista anteriormente, pero que se aplica en grupos de tres columnas.

De estos dos tipos de visualizadores se eligió una pantalla de cristal líquido de 24x1 caracteres, que corresponde a la AND691 cuyas características son:

1. Es un módulo compacto con alto contraste y caracteres grandes.
2. Trabaja con una fuente de +5V consumiendo baja potencia.
3. Tiene integrado su propio control en baja escala de integración, RAM para el display y ROM generadora de caracteres.
4. Interface directa con microprocesadores de 4 u 8 bits.
5. Tiene sus propios comandos de control.

Las características de este visualizador lo hacen el más conveniente para ser utilizado en el sistema como un puerto, además el tamaño de su pantalla es ideal pues para nuestra aplicación requerimos cuando mucho 24 caracteres por línea.

Las señales que requiere esta pantalla para poder comunicarse con un microprocesador de 8 bits son las siguientes:

Pin No.	Señal	Función
1	GND	0 V
2	VDD	+5V
3	Vo	Regula intensidad.
4	RS	datos/-comando
5	R/-W	lectura/-escritura
6	E	Habilitación
7	DB0	Bus de datos(LSB)
8	DB1	
9	DB2	
10	DB3	
11	DB4	
12	DB5	
13	DB6	
14	DB7	(MSB)

## III.2 DISEÑO DEL HARDWARE PARA EL CONTROL Y VISUALIZACIÓN

La fase de diseño del sistema de control para grabar y leer información de audio digital en un disco flexible básicamente consta de:

- a) Un sistema básico basado en un microprocesador.
- b) Interfaces de entrada-salida.
- c) Pantalla de visualización.
- d) Teclado.
- e) Tarjeta de interface con la PC.

Para cumplir estas especificaciones se ha mencionado en capítulos anteriores el soporte de la teoría necesaria para la selección de cada uno de los elementos que forman este sistema. Dichos elementos son:  $\mu P$  8088, pantalla AND601, teclado de PC-XT, tarjeta de interface con la PC.

El sistema básico fue diseñado con la finalidad de poder manejar satisfactoriamente todos los módulos.

Un generador de reloj proporciona la frecuencia de operación al  $\mu P8088$  además de las señales de RESET y READY. El  $\mu P$  fue utilizado en modo mínimo generando las señales de  $-\overline{RD}$ ,  $-\overline{WR}$ ,  $I/O/\overline{M}$ , un bus de datos y un bus de direcciones. El bus de direcciones fue manejado por medio de LATCHES y el de datos por medio de un bus bidireccional.

De esta forma se obtuvieron direcciones desde A0-A7 y un bus de datos D0-D7. Combinando las direcciones y la señal de  $I/O/\overline{M}$  se diseñó la decodificación en donde se contempló el manejo de un puerto paralelo 8255 para el manejo de las interfaces, un puerto paralelo para comunicarse con la tarjeta de interface con la PC, un puerto serie 8251A para manejo del teclado y un puerto para la habilitación de la pantalla; por último, una decodificación de memorias RAM y EPROM.

De esta forma se determinó el siguiente mapa de memoria y puertos:

MEMORIAS	LOCALIDAD
RAM	0000-7FFFH
EPROM	E000-FFFFH

PUERTOS	LOCALIDAD
SERIE	08-09H
PARALELO SISTEMA	00-03H
PARALELO INTERFACE	04-07H
PANTALLA	0A-0BH

Teniendo como base el sistema básico formado por el microprocesador y la decodificación la distribución de los elementos quedo como se muestra en el diagrama SGDA 1.

Como se observa, hay una zona de slots en el sistema principal en donde van colocadas cada una de las tarjetas que comprenden a

cada uno de los elementos empleados. Cada slot esta determinado para una sola tarjeta a la vez, de tal forma que estas tendrán una sóla ubicación. En cada uno de los slots se enviaron las señales de polarización GND y Vcc, el bus de datos (D0-D7), las señales de control IO/-M, -RD, -WR, -CE y algunas direcciones para la aplicación de cada tarjeta.

La tarjeta de memorias comprende además de las señales ya mencionadas una memoria RAM de 32 KBytes y una memoria EPROM de 8 KBytes, en esta tarjeta se encuentra el programa general del sistema con el cual se pueden manipular cada una de las subrutinas así como recepción y transmisión de los datos específicos para cada aplicación. Diagrama SGDA 2.

En la tarjeta correspondiente al manejo del puerto paralelo se contempló la adecuación de la señal en un determinado ancho de banda (20 Hz a 20 kHz) mediante filtros paso banda y la conversión de la señal de analógica a digital y viceversa. Diagrama SGDA 4 capítulo de Diseño de las Interfaces.

Por lo que corresponde a la tarjeta del puerto serie, se empleó un puerto 8251A en el cual, las terminales de recepción de datos serial y reloj de recepción corresponden con las señales de datos y reloj provenientes del teclado.

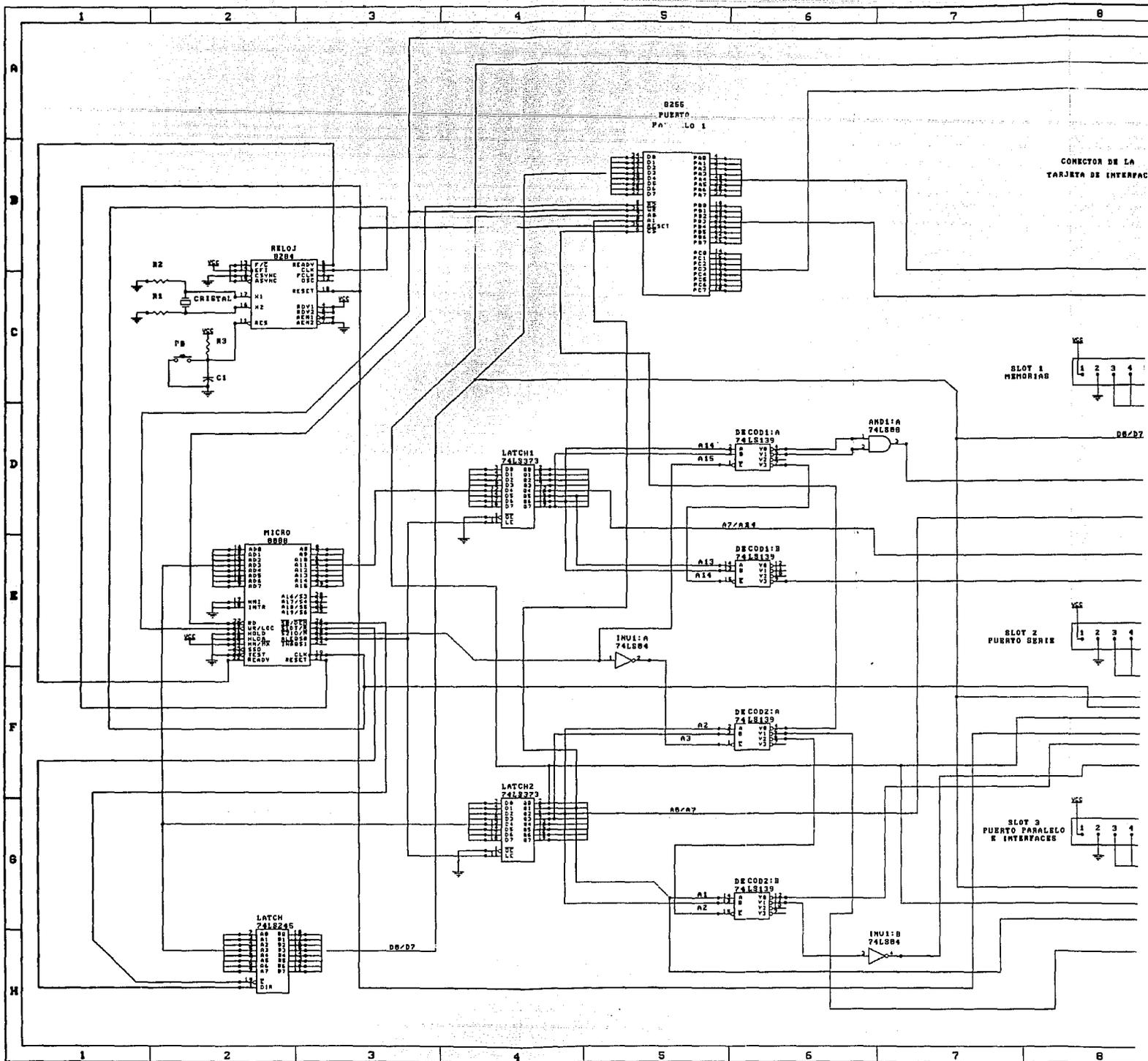
También se encuentran en esta tarjeta las señales correspondientes al manejo de la pantalla de visualización. Diagrama SGDA 3.

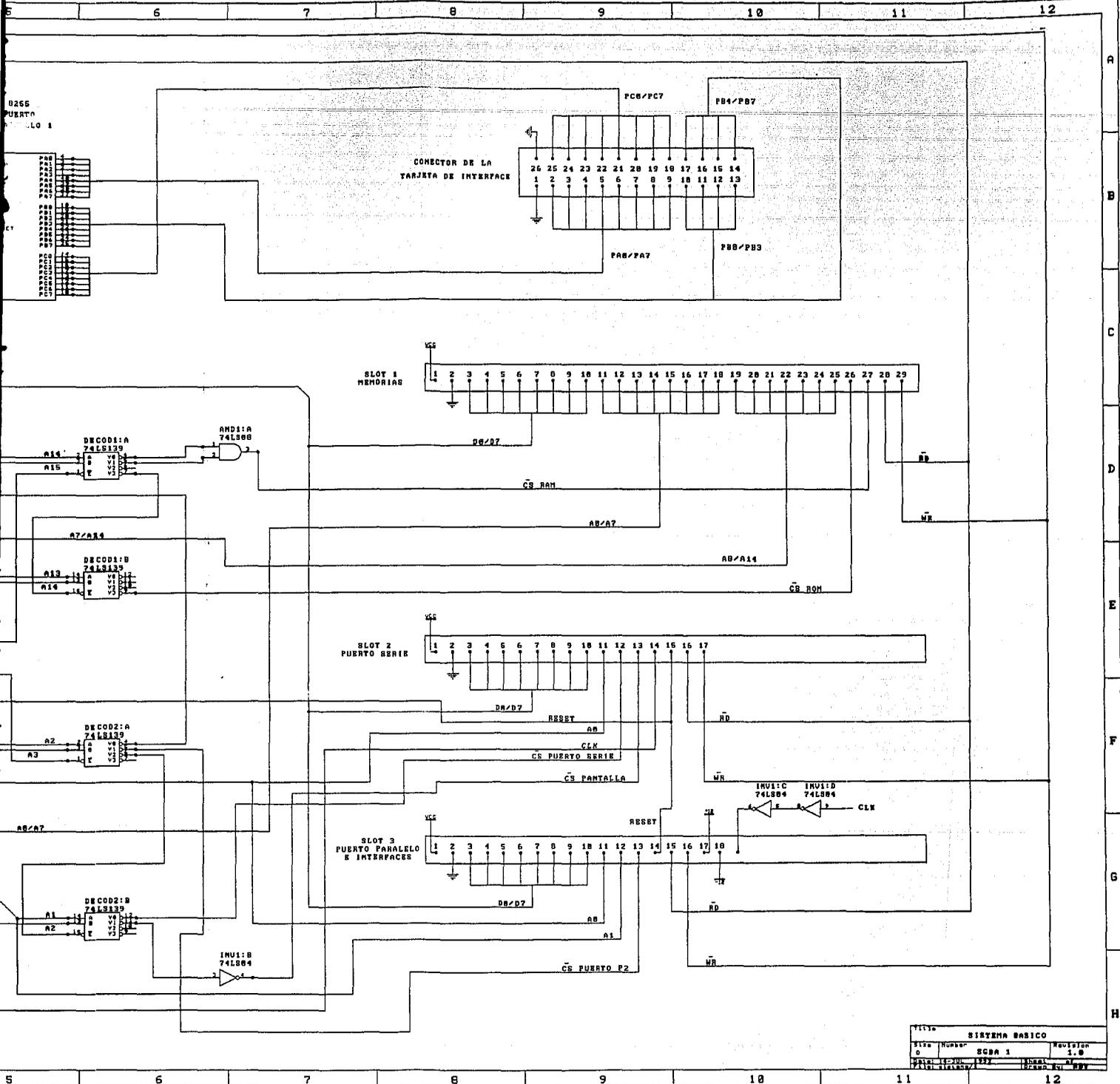
Para la tarjeta de comunicación con la PC se hizo una decodificación de tal manera que se pudiera manejar la comunicación sin interferir con las localidades ocupadas por la computadora, de esta forma se eligieron las localidades 300-303H.

La decodificación se diseñó con las direcciones que van de A0 a A9, además del AEN, este se usó para que cuando la PC maneje interrupciones vía DMA no se active el puerto de la tarjeta. Esto es con la finalidad de aislar el trabajo de la tarjeta mientras

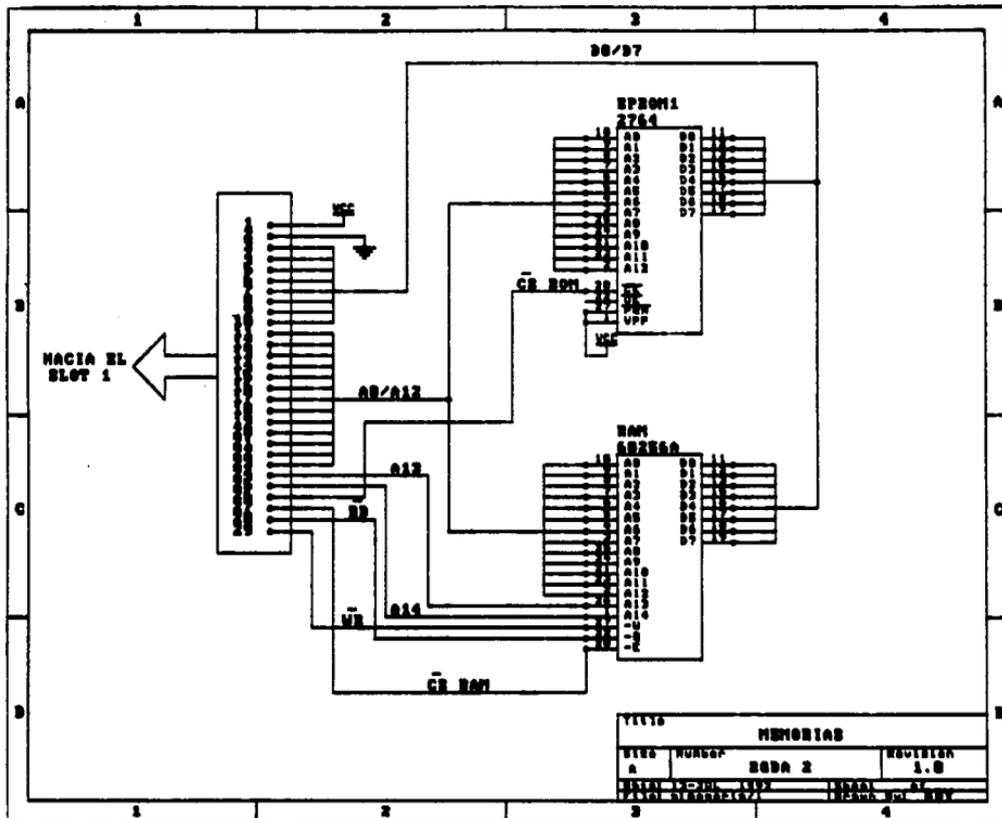
trabaja el DMA.

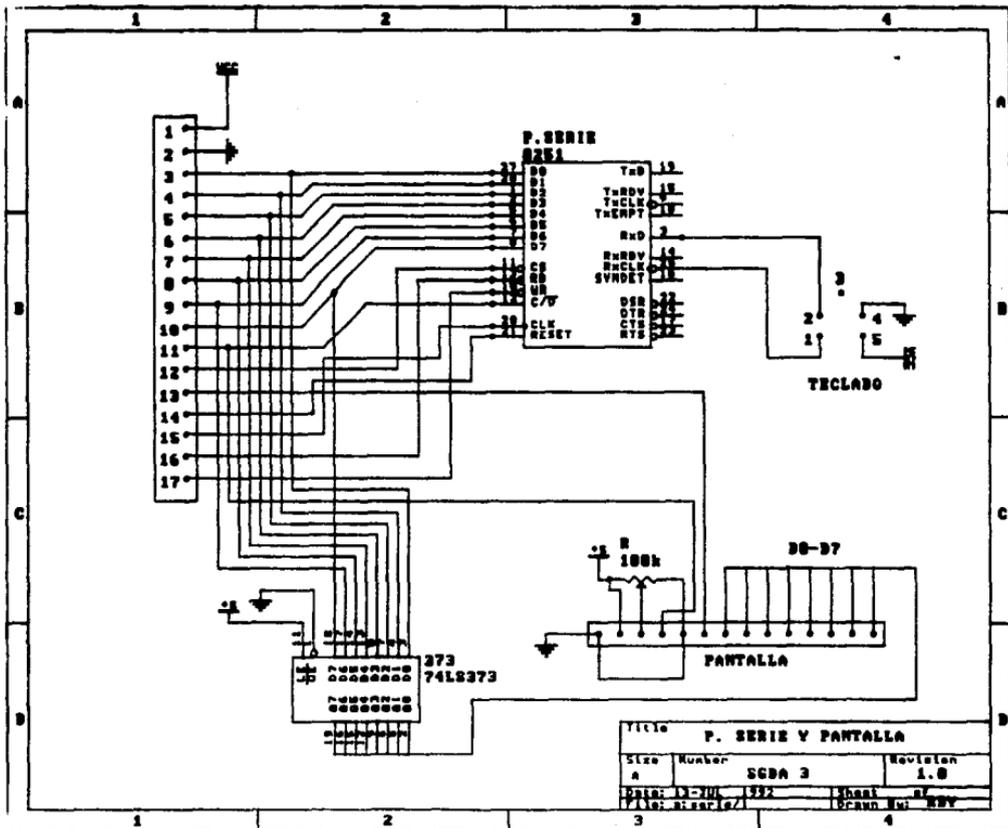
Se enviaron las señales de los puertos A,B y C vía un cable plano de conexión para comunicar el puerto paralelo del SGDA y el puerto paralelo de la tarjeta de interface con la PC. Diagrama SGDA 5.

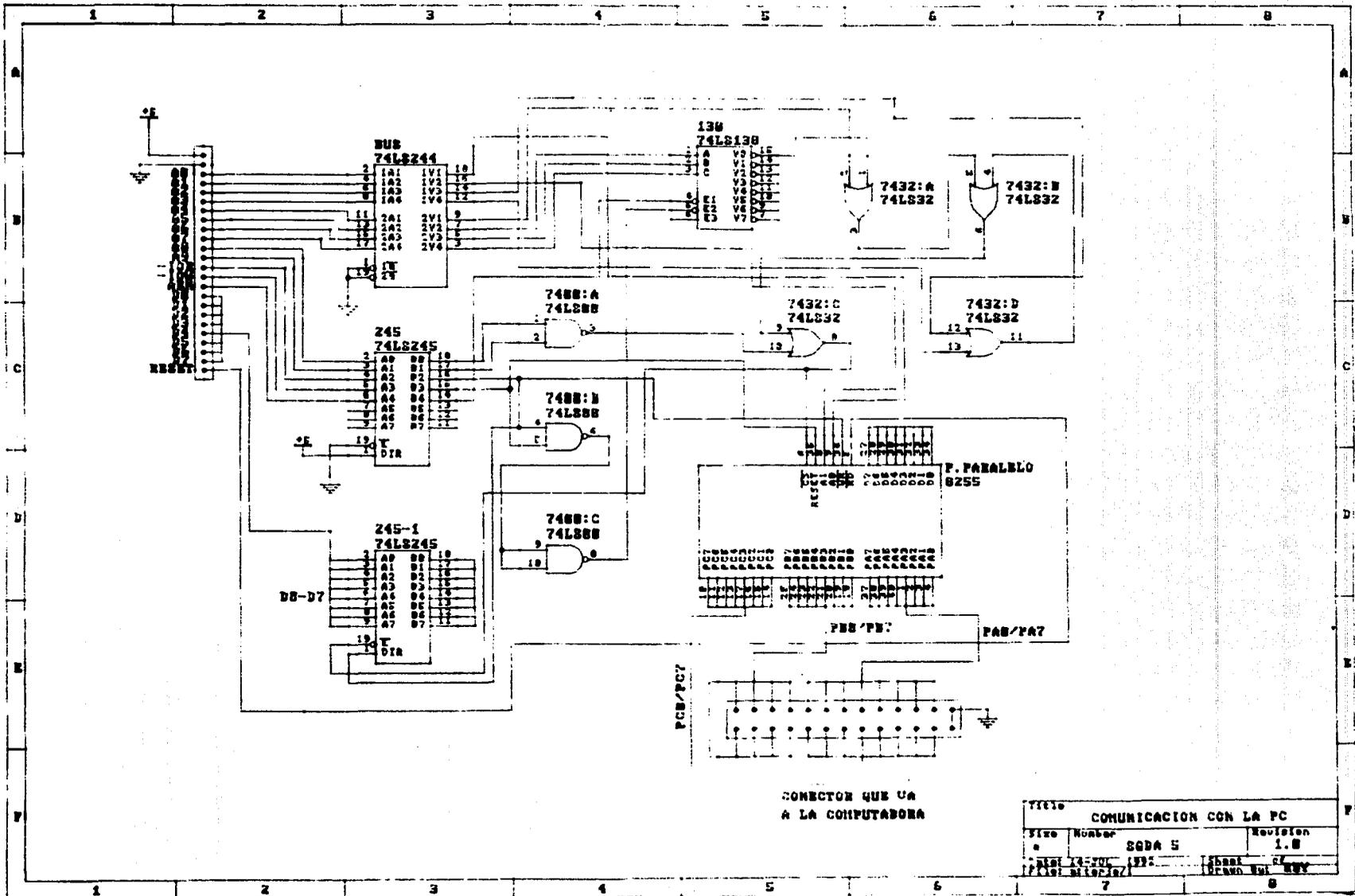




Título	SISTEMA BASICO		
Nº	0	Revisión	1.0
Elaborado	SCBA 1	Dibujado	
Revisado		Control. Ex.	

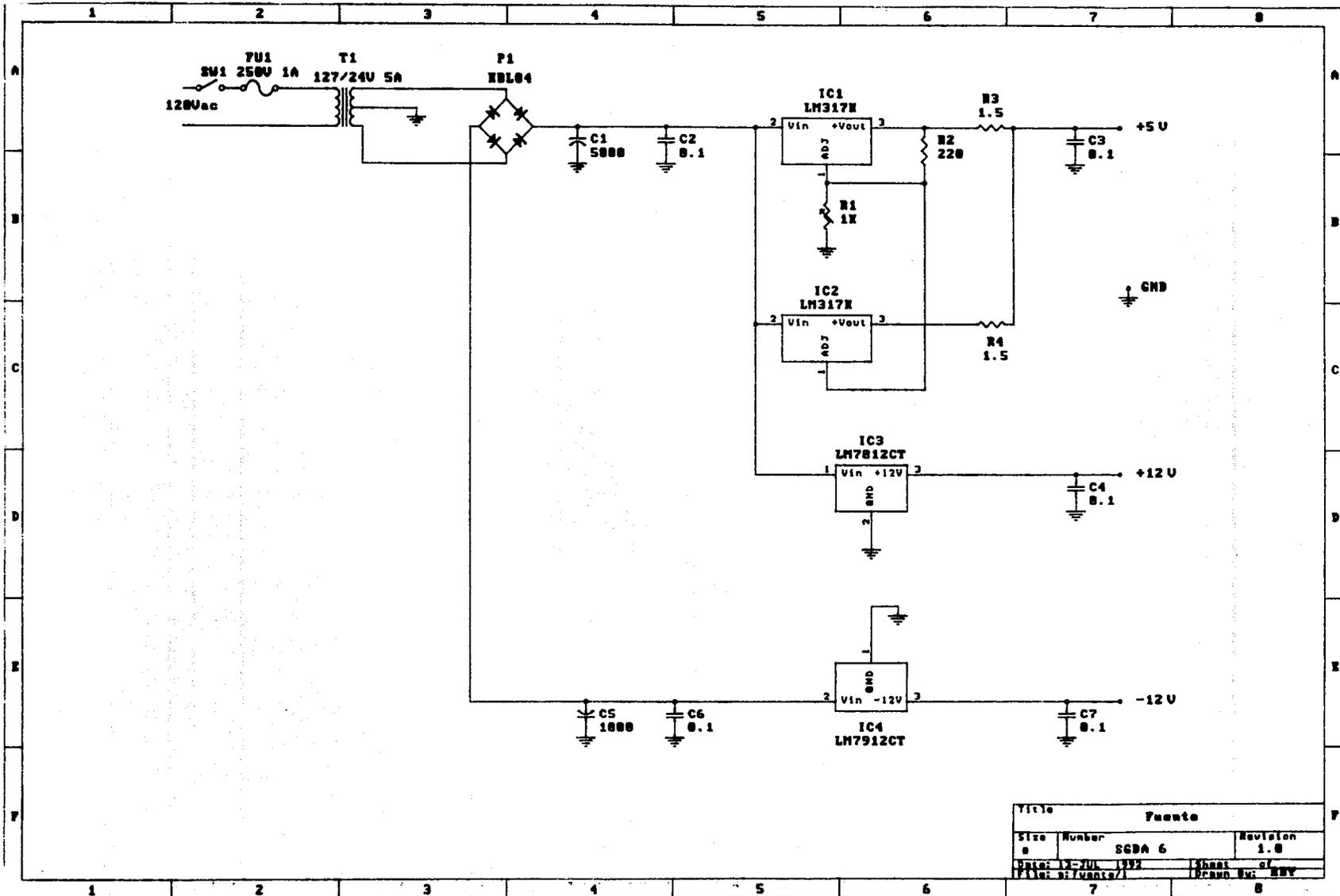






CONECTOR QUE VA  
A LA COMPUTADORA

TITULO			
COMUNICACION CON LA PC			
Size	Number	Revision	
	8255	1.0	
DATE	BY	CHKD	BY
1987-01-12	...	...	...



Title		Fuente	
Size	Number	Revision	
	SGDA 6	1.0	
Date:	12-2011	Sheet	of
File:	g:/vante/1	Drawn By:	BBV

### III.4 DISEÑO DE LAS INTERFACES

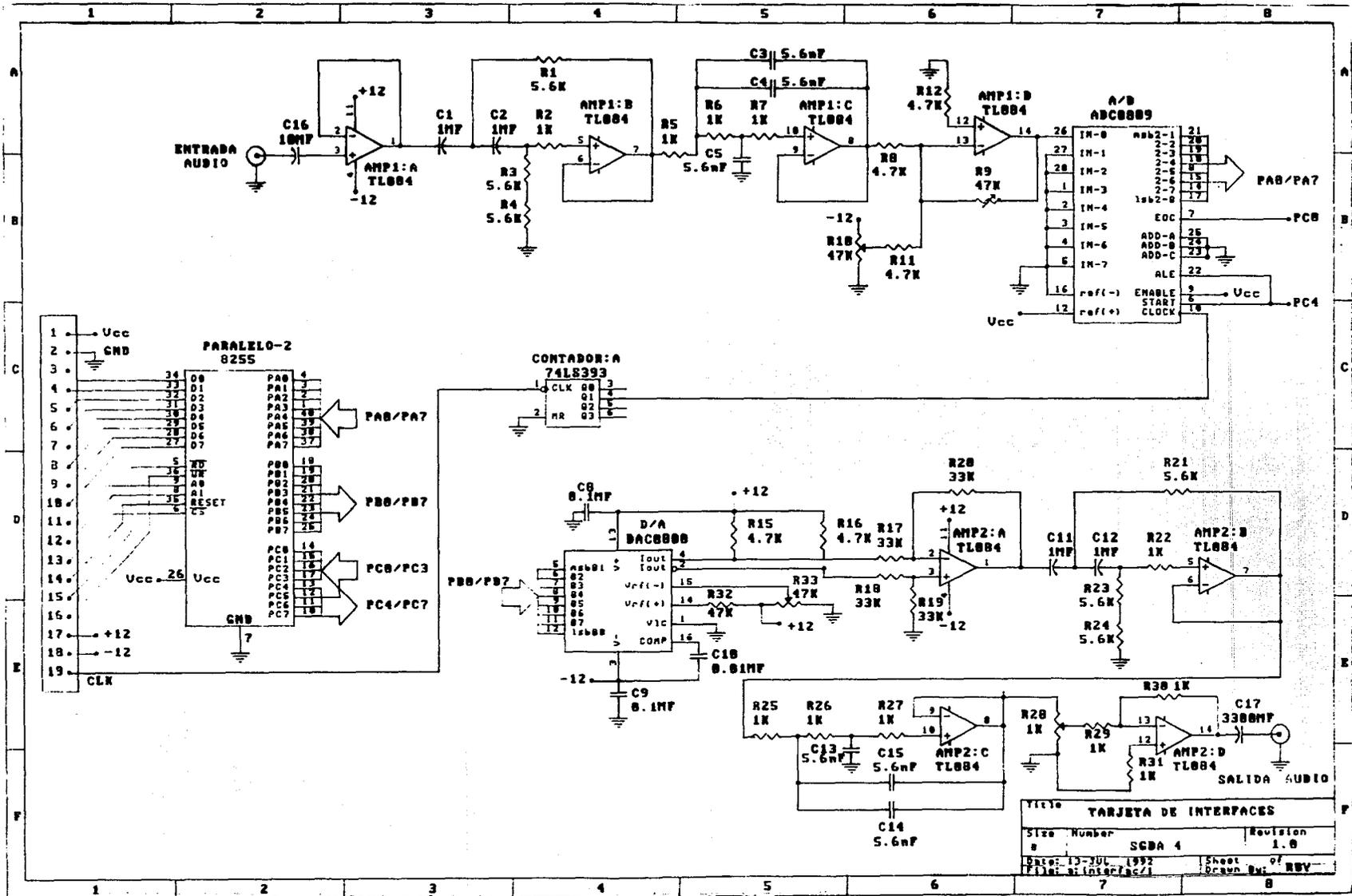
El sistema de grabación y reproducción de audio requiere de interfaces de entrada y salida adecuadas. La interface de entrada tiene como finalidad la adecuación de la señal analógica de audio para convertirla a forma digital por medio de un convertidor A/D. La interface de salida tiene como finalidad la adecuación de la señal proveniente del convertidor D/A para regresarla a su forma original, y recuperar sus características.

El diseño de dichas interfaces se realizó de la siguiente manera: Primeramente, para presentar una alta impedancia de entrada en la interface, se colocó un seguidor de voltaje utilizando un amplificador operacional de entrada JFET (TL084). En seguida, con la finalidad de filtrar la señal proveniente de la fuente de audio y restringirla a la banda teórica de frecuencia audible por el ser humano, se utilizó un filtro activo paso banda formado por medio de dos filtros, uno paso altas y otro paso bajas conectados en cascada, ambos de segundo orden, de forma tal que en conjunto forman un filtro paso banda de cuarto orden con frecuencias de corte en 20 y 20000Hz. Una vez filtrada la señal, haciendo uso de un sumador de voltaje se ajusta el nivel de amplitud y offset de la señal hasta ajustarla a un rango máximo de 0 a 5V, rango de entrada requerido por el convertidor A/D. La conversión A/D se realizó por medio de un convertidor de aproximaciones sucesivas, de 8 bits (ADC0809), el cual, como sabemos requiere de una señal de reloj ocho veces mayor a la frecuencia de muestreo deseada. Dicha frecuencia de muestreo, de acuerdo con el teorema de Nyquist, debe ser cuando menos del doble del valor de la máxima frecuencia a ser utilizada en la señal de entrada, como dicha señal se ha restringido a 20KHz, la mínima frecuencia de muestreo que podemos utilizar es 40KHz. Así, la señal de reloj se generó a una frecuencia de 2.38 MHz proporcionada por un contador (74LS393) al cual le llega el reloj del microprocesador.

La señal digital obtenida del convertidor A/D es introducida al sistema por medio de el puerto paralelo 8255 en su puerto "A". La sincronización de la lectura de dicha señal se logró por medio de los bits EOC y START comandados via software por el puerto paralelo "C", mandando un pulso de inicio de conversión (START) y esperando el flanco de subida de la señal de fin de conversión (EOC) y realizando entonces la lectura.

La señal digital de salida del sistema, por medio del puerto paralelo "B" es introducida al convertidor D/A a la misma frecuencia con la que fue muestreada. El convertidor D/A utilizado es uno de 8 bits, con salida de corriente (DAC0800). Este convertidor nos proporciona dos salidas de corriente complementarias, las cuales por medio de dos resistencias de pull-up referidas a un mismo potencial (+12V), generan una salida de voltaje diferencial, la cual por medio de un amplificador operacional en configuración diferencial es amplificada generando una señal de salida referida a tierra. Como sabemos, al regresar a su forma analógica una señal que fue convertida a digital, esta modifica su espectro de frecuencia proporcionandonos frecuencias múltiples a las de entrada ocasionando distorsión, de tal forma que para lograr rescatar solamente el espectro original de la señal de entrada, es necesario filtrar la señal obtenida del convertidor D/A con un filtro paso banda de las mismas características del utilizado a la entrada. Dicho filtro lleva, de hecho la misma construcción física. Finalmente, utilizando un potenciómetro como divisor de voltaje se ajusta la señal de salida para que tenga un rango de salida en el rango estándar de las fuentes de audio convencionales (300mV rms máximo). A la salida de la interface se utilizó un inversor con ganancia unitaria con la finalidad de presentar una impedancia de salida fija, de muy bajo valor a la salida del equipo.

El diagrama eléctrico de este diseño se muestra a continuación



TARJETA DE INTERFACES		
Size	Number	Revision
8	SGDA 4	1.0
Date:	13-JUL-1992	Sheet
File:	si:interfac/1	of
		Drawn
		By

### III.4.- DISEÑO DEL AMPLIFICADOR DE AUDIO

#### ESTRUCTURA BASICA DEL TMOS DE POTENCIA

Los TMOS de potencia pertenecen a la familia de transistores MOSFET's, los cuales presentan las siguientes ventajas:

- Impedancia estática de entrada casi infinita
- Capacidad de conmutación muy rápida
- Coeficiente de temperatura positivo
- Transconductancia constante
- Alta inmunidad a las variaciones de voltaje
- Bajo costo

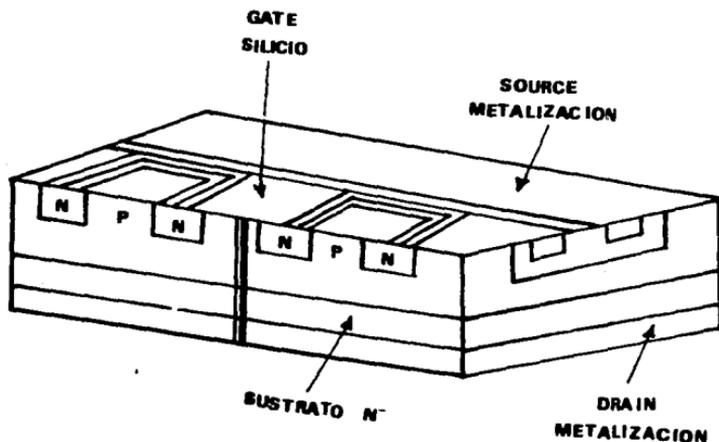
Sin embargo los TMOS son el último paso en la evolución de los MOSFET, la cual comenzó con el convencional MOSFET de señal pequeña, prosiguiendo con el MOSFET de canal lateral difundido doble (LDMOS) y continuando con el MOSFET de canal vertical en V (VMOSFET). El primero de estos consistió de una capa ligeramente dopada con tipo P, dentro de dos regiones altamente dopadas con tipo N<sup>+</sup>, estas últimas actuaron como source y drain; así de esta configuración resultaron una baja capacidad de corriente, bajo voltaje de inversa y alta resistencia de drain-source R<sub>DS(on)</sub>.

Para el LDMOS se efectuaron dos cambios significativos, el primero fue el uso de capas alineadas y de difusión doble para obtener canales de longitud muy corta y así lograr alta corriente y baja R<sub>DS(on)</sub>. El segundo fue la incorporación de una región ligeramente dopada con N<sup>+</sup> entre el canal y el drain, permitiendo altos voltajes de inversa. Sin embargo esta estructura tiene una desventaja que es el ineficiente uso de una área de silicio ya que esta se usa como tope de contacto del drain. El siguiente paso en la evolución del proceso fue la estructura vertical VMOSFET, con la cual se obtienen cuatro capas difundidas desarrolladas a través de métodos fotolitográficos. El término VMOSFET se deriva principalmente del hecho de que la corriente de drain-source sigue

una ruta vertical. El aumentar las corrientes es posible gracias a la reducción del canal.

Todo esto deriva en un coeficiente positivo de temperatura para retirar la preocupación de desbordamiento térmico, en una corriente de escape muy baja y altas velocidades de conmutación.

Finalmente se llega a la estructura del TMOS de potencia, la cual es similar a la estructura de un LDMOS excepto por que la zona de contacto con el drain está dopada a través del sustrato  $N^-$  hasta la parte trasera de la pastilla. La estructura del gate se hace con polysilicio en medio de dos líneas de óxido y el source-metal es aplicado continuamente sobre las áreas activas, éstas dos líneas eléctricas de contacto proporcionan un óptimo empaque y mantienen las ventajas del LDMOS. Todo esto tiene como resultado altos voltajes y corrientes, además de una  $R_{DS(on)}$  baja.

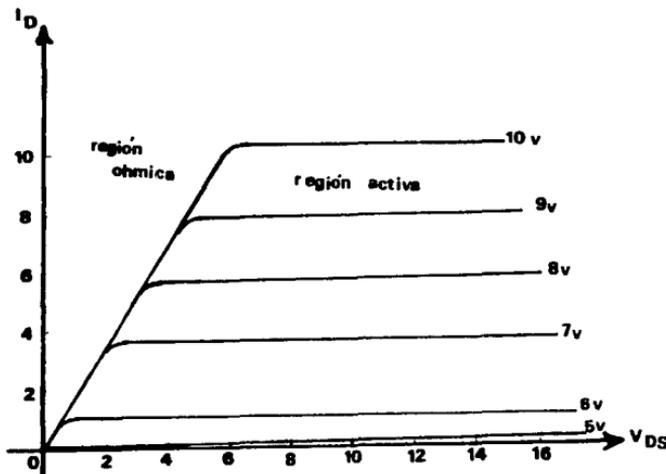


CORTE DEL ENCAPSULADO DE UN TMOS

## CARACTERISTICAS

### CARACTERISTICAS DE SALIDA:

Tal vez el camino más directo para empezar a familiarizarse con la operación básica de un dispositivo son sus características de salida, así podemos decir que las características del MOSFET de potencia pueden ser divididas en dos regiones básicas: la región de encendido (o región ohmica) y la región activa.



Como puede observarse en la figura anterior existen tres características importantes en las curvas. La primera es que la familia de curvas es generada por cambios en el voltaje de gate.

La segunda es que en la región ohmica la pendiente de las curvas es muy poco pronunciada indicando que la resistencia de encendido es alta. Y la tercera es que en la región activa la pendiente de las curvas es menos inclinada haciendo que exista una corriente constante de source, la limitación de la corriente de

drain es debida al pinch-off en el canal MOSFET.

## PARAMETROS BASICOS

### RESISTENCIA DE ENCENDIDO:

La resistencia de encendido  $R_{DS(on)}$  de un TMOS es un importante parámetro porque determina la corriente total que el dispositivo puede manejar sin que haya excesiva disipación de potencia. Cuando se conmuta el TMOS de off a on, la resistencia drain-source cae de un valor muy alto a la resistencia de encendido, que es un valor relativamente bajo. Para minimizar  $R_{DS(on)}$  el voltaje de gate deberá ser lo suficientemente grande para proporcionar corriente de drain y mantenerla operación del TMOS en la región ohmica. Cuando la corriente de drain sube, la resistencia de encendido también se incrementa y a su vez la temperatura.

### TRANSCONDUCTANCIA:

La transconductancia  $g_{FS}$  denota la ganancia en el MOSFET (al igual que la  $\beta$  en el transistor bipolar), este es un parámetro importante cuando el dispositivo está trabajando en la región activa. Está definido como la relación del cambio en la corriente de drain correspondiente a un cambio de voltaje de gate-source:

$$g_{FS} = di_D/dV_{GS}$$

Típicamente los rangos de transconductancia están especificados a la mitad del rango de la corriente de drain y a  $V_{GS}$  de 15 volts. Para diseños donde solo importe tener el TMOS entre estados de encendido y apagado la transconductancia es siempre un parámetro que no se emplea.

### VOLTAJE DE UMBRAL

El voltaje de umbral es el mínimo voltaje de gate que

genera una determinada corriente que circulará por el drain (esta corriente es pequeña). Motorola especifica el voltaje de umbral cuando se origina una corriente de drain de 1 miliamper. En resumen este tipo de transistores pueden ser ampliamente utilizados para fabricar un buen amplificador, ya que si se usan en la etapa final de potencia se obtendría una configuración sencilla, sin muchos componentes y en sonoridad similar a la de los amplificadores más complicados. Esto se debe principalmente a que los TMOS amplifican en voltaje y no en corriente, además tienen una elevada velocidad de conmutación y principalmente que poseen un coeficiente de temperatura positivo, es decir que en cuanto más se calientan más se incrementa su resistencia interna, reduciendo la corriente de reposo, fenómeno que si no se controla puede producir daños en los transistores de la etapa final de potencia de cualquier amplificador.

Siendo así procedimos a elegir nuestra tensión de alimentación de -35 y +35 volts, para que con una carga de 8 ohms obtengamos aproximadamente una potencia de salida de 76.56 watts. Por lo tanto los dos TMOS que elegimos para nuestro amplificador son:

IRF522 canal N

IRF9532 canal P

los cuales son capaces ambos de soportar un voltaje máximo de drain de 100 volts.

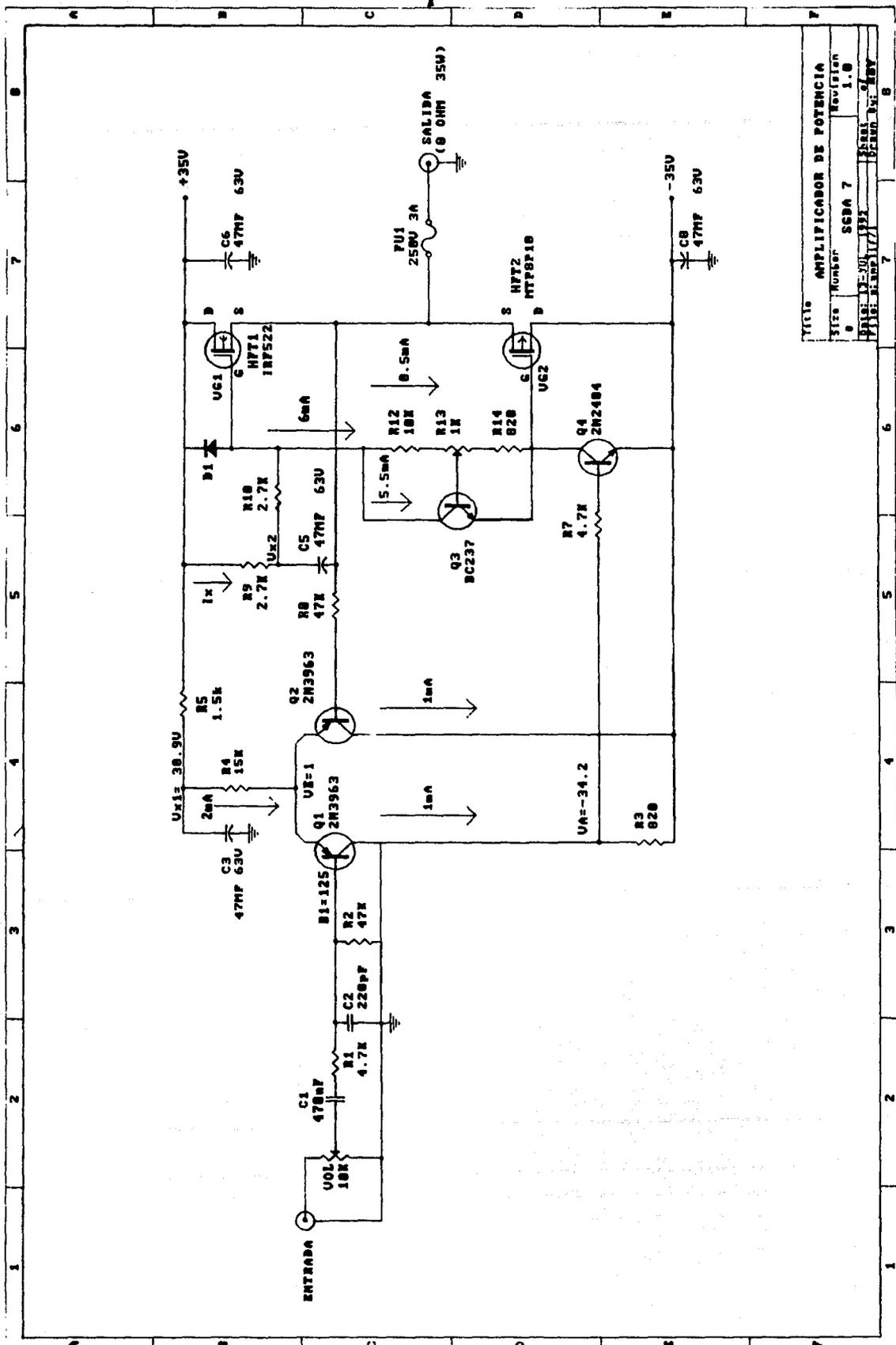
Decidimos que la señal procedente de cualquier preamplificador encuentre en primera instancia un filtro paso banda pasivo, formado por el capacitor  $C_s$ , la resistencia  $R_1$ , el condensador  $C_z$  y la resistencia  $R_2$ , este filtro deja pasar una señal de 15 a 100,000 Hz lo cual es suficiente para dejar pasar adecuadamente tanto las notas bajas como las agudas.

La siguiente etapa está formada por un amplificador diferencial, construido por dos transistores PNP ( $TR_1$  y  $TR_2$ ), la cual permite conseguir una distorsión de intermodulación reducida y una

estabilidad de funcionamiento elevada. La señal que procede del filtro llega a la base de TR<sub>1</sub> y por la base de TR<sub>2</sub> llega la señal amplificada procedente directamente de la salida. Esta retroalimentación incrementa de forma considerable la ganancia, debido a la marcada diferencia que existe entre las dos señales aplicadas en ambas entradas. La ganancia global del amplificador se puede modificar aumentando o disminuyendo el valor de la resistencia R<sub>a</sub>.

La señal amplificada presente en el colector de TR<sub>1</sub> se aplica después a la base del transistor TR<sub>4</sub>, que se encarga de manejar los dos TMOS (HFT<sub>1</sub> y HFT<sub>2</sub>) que operan en clase AB complementaria. Esto significa que cada transistor amplifica una sola semionda: para ser precisos HFT<sub>1</sub> es de canal N y solo amplifica las semiondas positivas, mientras que HFT<sub>2</sub> es de canal P y solo amplifica las semiondas negativas. En el punto de unión de los dos source las dos semiondas se unen y se obtiene una onda completa. El transistor TR<sub>4</sub> tiene otra importante misión: hacer que en los dos transistores de potencia circule siempre una corriente mínima de reposo, necesaria para eliminar la distorsión de cruce o *cross-over*, que es la deformación en la onda de salida en el instante de su paso por cero. La corriente de reposo se regula girando el trimmer R<sub>ss</sub>, a cuyo cursor está conectada la base del transistor TR<sub>3</sub>, el cual garantiza que la corriente de reposo se mantenga constante, aun cuando varíe la temperatura de los TMOS. Para proteger el gate de HFT<sub>1</sub> contra posibles picos de voltaje se colocó el diodo DS<sub>1</sub> entre el gate y la fuente positiva. Para proteger el otro transistor no se puso ningún diodo ya que el transistor TR<sub>4</sub> también se encarga de esta función.

Finalmente, a la salida de los TMOS se colocó un fusible F<sub>a</sub> para proteger la bocina en caso de sobre potencia a la salida del amplificador, por otro lado este fusible también protege al amplificador de un cortocircuito en la salida.



TÍTULO	AMPLIFICADOR DE POTENCIA
Tipo	Numero
8	SCBA 7
Revisión	1.0
FECHA: 12-2011	FECHA: 12-2011
DISEÑADOR: J. J. J.	REVISOR: J. J. J.
1	8

### III.5 BLINDAJE DE SEÑALES

Un blindaje es una división metálica colocada entre dos regiones del espacio. Es utilizado para controlar la propagación de campos, tanto eléctricos como magnéticos de una región hacia otra. Un blindaje puede ser utilizado para contener campos electromagnéticos, si el blindaje cubre la fuente de ruido se dará protección a aquellos equipos localizados fuera de ese blindaje. Un blindaje puede ser también utilizado para mantener la radiación electromagnética fuera de una región, así se dará protección sólo al equipo contenido dentro de él.

Desde un punto de vista general, es más eficiente blindar la fuente de ruido que todos los receptores, sin embargo, hay muchos otros casos en los que la fuente necesita radiar (por ejemplo en las estaciones radiodifusoras) es entonces necesario blindar individualmente al equipo susceptible de recibir dicha radiación. Es de suma importancia que el blindaje no contenga aperturas, sin embargo, es muy común que el equipo requiera de éstas para la entrada y salida de cables (tanto de alimentación como de señales), por lo que resulta necesario filtrar de ellos los voltajes de ruido que son introducidos a pesar del blindaje.

En un circuito analógico, una pequeña cantidad de ruido puede causar interferencia, en especial si se trabaja con señales de bajo nivel (milivolts o microvolts) y/o se cuenta con amplificadores de alta ganancia.

En contraste, en los circuitos digitales no hay amplificadores, y se trabaja con señales de más o menos alto nivel (5 volts).

Los circuitos digitales presentan un margen de ruido relativamente alto (para circuitos CMOS es aproximadamente 1.5 V si se trabaja con una fuente de 5 V), y por lo tanto poseen inmunidad al ruido de bajo nivel.

## EL RUIDO EN SISTEMAS DIGITALES

El diseño digital se encuentra dentro de las matemáticas puras, donde las ecuaciones describen las funciones a ser implementadas. Sin embargo aunque la lógica sea muy correcta, el diseño podría no funcionar por causa del ruido. Por lo tanto es de gran importancia tomar en cuenta los aspectos prácticos del ruido, tanto durante el diseño como durante las pruebas del circuito digital.

En especial un circuito que combina parte analógica y parte digital, hay que tomar en cuenta que existen distintas técnicas para disposición de tierras, de polarización y de interconexión. Por ejemplo una tierra colocada en un solo punto puede ser deseable en un circuito analógico, pero en uno digital puede significar una fuente primaria de acoplamiento y de emisión de ruido.

Los circuitos de compuertas lógicas digitales, que consumen unos cuantos miliamperes de DC, en primera instancia no representan una fuente de ruido. Sin embargo, su alta velocidad de conmutación combinada con la inductancia de los conductores que las interconectan pueden ocasionar serios problemas de ruido en el circuito. El voltaje generado cuando la corriente cambia a través de un inductor es:

$$V_L = L \frac{di}{dt}$$

Donde L es la inductancia y la derivada de la corriente es la razón de cambio de la corriente con respecto al tiempo. Por ejemplo, una compuerta lógica convencional consume, en su estado de encendido "1", 5 mA de la fuente y en su estado de apagado "0" 1 mA. Este es solo un cambio de 4 mA pero que ocurre en un lapso de 2 nanosegundos. Suponiendo una inductancia del alambrado de la fuente de 500 nH, el voltaje de ruido producido, de acuerdo a la ecuación anterior sería de 1 volt. Tomando en cuenta esto, y la

gran cantidad de compuertas en un circuito digital, puede concluirse que éstas si son considerables fuentes de ruido en un sistema.

#### DESACOPLO DE LA FUENTE DE ALIMENTACION

La magnitud de la caída de voltaje transitorio a través de la inductancia existente entre la fuente de polarización y la terminal Vcc de cada compuerta puede disminuir si se disminuye el valor de la inductancia L , o bien, disminuyendo el valor de la corriente transitoria que fluye a través de dicha inductancia. Dicha corriente transitoria puede minimizarse suministrando la corriente por medio de otra fuente, tal como lo sería un capacitor cercano a la compuerta lógica. Así, el voltaje de ruido en dicha compuerta será función del valor del capacitor, llamado de desacoplo (Cd), y el alambrado entre la fuente y la compuerta. El valor mínimo del capacitor Cd puede calcularse de la siguiente manera:

$$C_d = l \frac{dt}{dV}$$

Donde dV es la caída de voltaje transitorio causado por el cambio de la corriente transitoria I durante el tiempo dt. Por ejemplo, si un circuito integrado requiere 50mA durante 2 nanosegundos y se desea que la mínima caída sea 0.1 V, el valor mínimo para el capacitor sera de 0.001  $\mu$ F.

En base a pruebas de tipo experimental sobre circuitos integrados de 14 o 16 pines, se ha encontrado que a menudo la menor caída de potencial se presenta para capacitores en el rango de 470-1000pF.

## PROTECCION CONTRA RUIDO EN EL SGDA

Para lograr una adecuada protección contra el ruido en el sistema de grabación digital de audio, se tomaron en cuenta las consideraciones mencionadas anteriormente. Para evitar la interferencia provocada por señales externas, se blindó el equipo por medio de una cubierta metálica completa. Para lograr el desacoplo de la fuente de alimentación a cada uno de los circuitos digitales, se colocó un capacitor de  $0.1\mu\text{F}$  entre las terminales Vcc y GND de cada circuito integrado. Con la finalidad de disminuir el valor de las inductancias producidas por las trayectorias de Vcc y GND en la parte digital, se optó por enviar varias trayectorias entre elementos desde la fuente de alimentación, de manera que las corrientes de regreso pudieran tener varios caminos para fluir y disminuir el ruido producido por esta causa, en el caso de la parte analógica (tarjeta de interfaces) se colocó un solo punto de tierra a el cual se conectaron todas las tierras de cada circuito integrado.

## CONCLUSIONES

Dados los objetivos planteados, finalmente se logró un sistema capaz de grabar y reproducir audio digitalmente, en un disco flexible de 5 1/4 de baja densidad con capacidad de 360 KBytes.

Dicho proyecto comprendió: un sistema de control, que presenta al operador una pantalla y un teclado, un amplificador de audio de 75 Watts, una tarjeta de interface con la PC y el software necesario para esta última. Cabe mencionar que la PC debe tener ciertas características para que se logre la aplicación mencionada, y estas son: una computadora XT con 2 drives o bien un drive y un disco duro, una memoria RAM con un mínimo de 512 KBytes y además que no utilice en otra aplicación la direcciones de puertos de la 300h a la 303h manejadas por nuestra tarjeta de comunicación.

El tiempo que se logró grabar fue de 47 segundos con una frecuencia de muestreo de 35 kHz, esta frecuencia se usó con el fin de poder utilizar el reloj del microprocesador 8088. Debido al compromiso existente entre el tiempo de grabación, la frecuencia de muestreo y la fidelidad de la música el tiempo grabado fué relativamente corto pues se sacrificó tiempo por fidelidad en el sonido.

Cabe señalar que con un disco de mayor capacidad (de alta densidad o floptical) se incrementaría el tiempo de grabación; aunado a lo anterior, la información podría ser compactada de tal forma que el tiempo de grabación aumentará aún más.

Otra opción de mejorar el sistema sería hacerlo independiente de la computadora, es decir se debería eliminar la tarjeta de comunicación y en lugar de esta se adicionarían los circuitos necesarios, tales como DMA, controlador de interrupciones, decodificadores, etc; para que el sistema manejará directamente

al drive.

En lo que respecta al software, el hecho de haber utilizado el lenguaje ensamblador nos permitió una mayor rapidez de acceso a la información pues se logra una comunicación directa con la computadora, además si se hiciera el sistema independiente de la computadora el algoritmo principal no se vería afectado y sería mucho más sencillo hacerle los cambios pertinentes.

Una de la ventajas que presenta este proyecto es que no sólo puede reproducir audio digitalmente sino también puede grabarlo y borrarlo del disco si así se desea.

Por otro lado no sólo se puede reproducir, grabar y borrar información de audio ya que también podría hacerse lo mismo con cualquier tipo de señal analógica (temperatura, presión, flujo, signos vitales, etc.) por supuesto haciendo los cambios pertinentes en la adecuación de la señal.

Consideramos que este trabajo puede ser un punto de partida para aplicaciones más particulares en manejo y almacenamiento digital de señales.

## APENDICES

## APENDICE A. HOJAS DE ESPECIFICACIONES



## 8088 8-BIT HMOS MICROPROCESSOR 8088/8088-2

- 8-Bit Data Bus Interface
- 16-Bit Internal Architecture
- Direct Addressing Capability to 1 Mbyte of Memory
- Direct Software Compatibility with 8086 CPU
- 14-Word by 16-Bit Register Set with Symmetrical Operations
- 24 Operand Addressing Modes
- Byte, Word, and Block Operations
- 8-Bit and 16-Bit Signed and Unsigned Arithmetic in Binary or Decimal, Including Multiply and Divide
- Two Clock Rates:
  - 5 MHz for 8088
  - 8 MHz for 8088-2
- Available in EXPRESS
  - Standard Temperature Range
  - Extended Temperature Range

The Intel® 8088 is a high performance microprocessor implemented in N-channel, depletion load, silicon gate technology (HMOS), and packaged in a 40-pin Cerdip package. The processor has attributes of both 8- and 16-bit microprocessors. It is directly compatible with 8086 software and 8080/8085 hardware and peripherals.

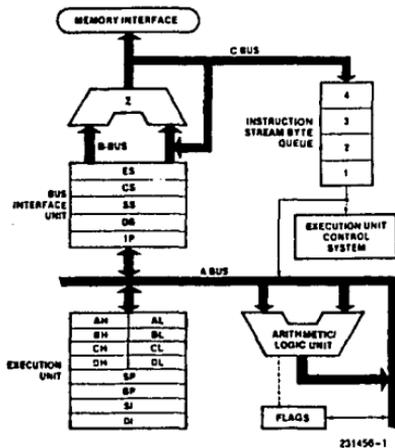


Figure 1. 8088 CPU Functional Block Diagram

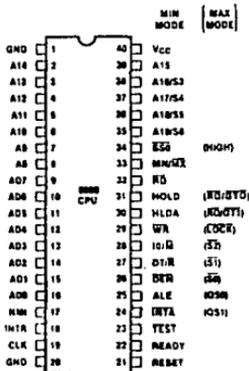


Figure 2. 8088 Pin Configuration



## 8255A/8255A-5 PROGRAMMABLE PERIPHERAL INTERFACE

- MCS-85™ Compatible 8255A-5
- 24 Programmable I/O Pins
- Completely TTL Compatible
- Fully Compatible with Intel Microprocessor Families
- Improved Timing Characteristics
- Direct Bit Set/Reset Capability Easing Control Application Interface
- Reduces System Package Count
- Improved DC Driving Capability
- Available in EXPRESS
  - Standard Temperature Range
  - Extended Temperature Range
- 40 Pin DIP Package or 44 Lead PLCC

(See Intel Packaging, Order Number: 231388)

The Intel 8255A is a general purpose programmable I/O device designed for use with Intel microprocessors. It has 24 I/O pins which may be individually programmed in 2 groups of 12 and used in 3 major modes of operation. In the first mode (MODE 0), each group of 12 I/O pins may be programmed in sets of 4 to be input or output. In MODE 1, the second mode, each group may be programmed to have 8 lines of input or output. Of the remaining 4 pins, 3 are used for handshaking and interrupt control signals. The third mode of operation (MODE 2) is a bidirectional bus mode which uses 8 lines for a bidirectional bus, and 5 lines, borrowing one from the other group, for handshaking.

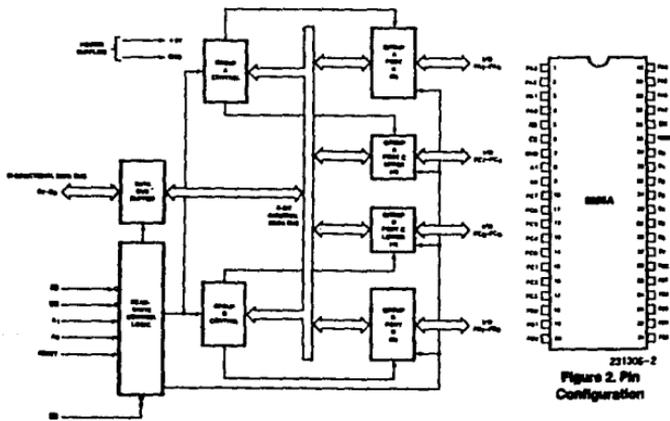


Figure 1. 8255A Block Diagram

231308-1

231308-2  
Figure 2. Pin Configuration



National  
Semiconductor  
Corporation

## DAC0800/DAC0801/DAC0802 8-Bit Digital-to-Analog Converters

### General Description

The DAC0800 series are monolithic 8-bit high-speed current-output digital-to-analog converters (DAC) featuring typical settling times of 100 ns. When used as a multiplying DAC, monotonic performance over a 4 $\times$  to 11 reference current range is possible. The DAC0800 series also features high compliance complementary current outputs to allow differential output voltages of 20 V<sub>pp</sub> with simple resistor loads as shown in Figure 1. The reference-to-full-scale current matching of better than  $\pm 1$  LSB eliminates the need for full-scale trims in most applications while the nonlinearity of better than  $\pm 0.1\%$  over temperature minimizes system error accumulations.

The noise-immune inputs of the DAC0800 series will accept TTL levels with the logic threshold pin,  $V_{IC}$ , grounded. Changing the  $V_{IC}$  potential will allow direct interface to other logic families. The performance and characteristics of the device are essentially unchanged over the full  $\pm 4.5$  V  $\pm 15$  V power supply range; power dissipation is only 33 mW with  $\pm 5$  V supplies and is independent of the logic input states.

The DAC0800, DAC0802, DAC0800C, DAC0801C and DAC0802C are a direct replacement for the DAC-08, DAC-08A, DAC-08C, DAC-08E and DAC-08H, respectively.

### Features

- Fast settling output current 100 ns
- Full scale error  $\pm 1$  LSB
- Nonlinearity over temperature  $\pm 0.1\%$
- Full scale current drift  $\pm 10$  ppm/ $^{\circ}$ C
- High output compliance  $-10$  V to  $+18$  V
- Complementary current outputs
- Interface directly with TTL, CMOS, PMOS and others
- 2 quadrant wide range multiplying capability
- Wide power supply range  $\pm 4.5$  V to  $\pm 15$  V
- Low power consumption 33 mW at  $\pm 5$  V
- Low cost

### Typical Applications

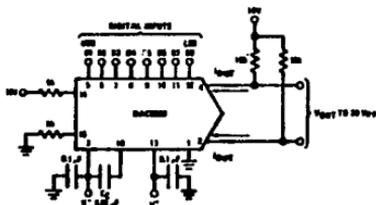


FIGURE 1.  $\pm 20$  V<sub>pp</sub> Output Digital-to-Analog Converter (Note 6)

### Ordering Information

Non-Linearity	Temperature Range	Order Numbers				
		J Package (J16A)*		N Package (N16A)*		SD Package (SD16A)
$\pm 0.1\%$ FS	$-55^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$	DAC0802LJ	DAC-08AQ	DAC0802LN	DAC-08HP	DAC0802LM
$\pm 0.1\%$ FS	$0^{\circ}\text{C} \leq T_A \leq +70^{\circ}\text{C}$	DAC0802LJ	DAC-08HQ	DAC0802LN	DAC-08HP	DAC0802LM
$\pm 0.10\%$ FS	$-55^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$	DAC0800LJ	DAC-08CQ	DAC0800LN	DAC-08CP	DAC0800LM
$\pm 0.10\%$ FS	$0^{\circ}\text{C} \leq T_A \leq +70^{\circ}\text{C}$	DAC0800LJ	DAC-08CQ	DAC0800LN	DAC-08CP	DAC0800LM
$\pm 0.30\%$ FS	$0^{\circ}\text{C} \leq T_A \leq +70^{\circ}\text{C}$	DAC0801LJ	DAC-08CQ	DAC0801LN	DAC-08CP	DAC0801LM

\*Devices may be ordered by using either order number.



National  
Semiconductor  
Corporation

## ADC0808, ADC0809 8-Bit $\mu$ P Compatible A/D Converters with 8-Channel Multiplexer

### General Description

The ADC0808, ADC0809 data acquisition component is a monolithic CMOS device with an 8-bit analog-to-digital converter, 8-channel multiplexer and microprocessor compatible control logic. The 8-bit A/D converter uses successive approximation as the conversion technique. The converter features a high impedance chopper stabilized comparator, a 256R voltage divider with analog switch tree and a successive approximation register. The 8-channel multiplexer can directly access any of 8 single-ended analog signals.

The device eliminates the need for external zero and full-scale adjustments. Easy interfacing to microprocessors is provided by the latched and decoded multiplexer address inputs and latched TTL TRI-STATE<sup>®</sup> outputs.

The design of the ADC0808, ADC0809 has been optimized by incorporating the most desirable aspects of several A/D conversion techniques. The ADC0808, ADC0809 offers high speed, high accuracy, minimal temperature dependence, excellent long-term accuracy and repeatability, and consumes minimal power. These features make this device ideally suited to applications from process and machine control to consumer and automotive applications. For 16-channel multiplexer with common output (sample/hold port) see ADC0816 data sheet. (See AN-247 for more information.)

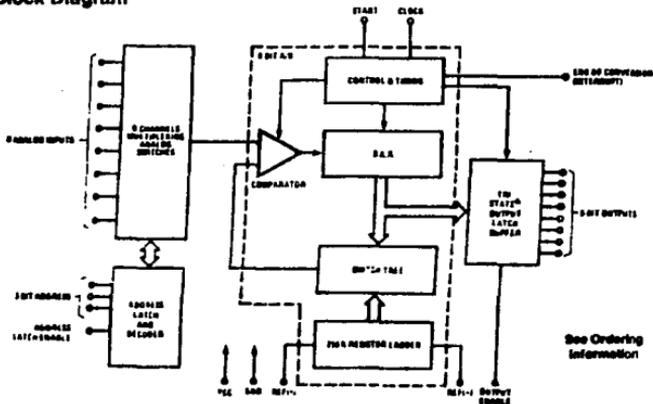
### Features

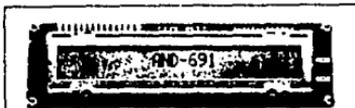
- Easy interface to all microprocessors
- Operates ratiometrically or with 5 V<sub>DC</sub> or analog span adjusted voltage reference
- No zero or full-scale adjust required
- 8-channel multiplexer with address logic
- 0V to 5V input range with single 5V power supply
- Outputs meet TTL voltage level specifications
- Standard hermetic or molded 28-pin DIP package
- 28-pin molded chip carrier package

### Key Specifications

- Resolution 8 Bits
- Total Unadjusted Error  $\pm 1/2$  LSB and  $\pm 1$  LSB
- Single Supply 5 V<sub>DC</sub>
- Low Power 15 mW
- Conversion Time 100  $\mu$ s

### Block Diagram



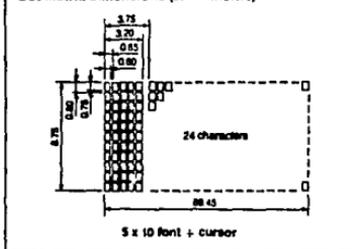


The AND691 is a compact LCD module having a dot matrix LCD panel, a controller and driver circuit. This module can display 160 kinds of alphabets, numerals, symbols and "Kana" letters, as well as 9 custom characters.

## FEATURES

- Compact, integrated display module.
- High contrast, clear display with large characters.
- Low voltage,  $\pm 5V$  single power supply.
- Wide operating temperature range ( $0^{\circ}C$  to  $+50^{\circ}C$ ).
- $5 \times 10$  dot character format and cursor line.
- Built-in control LSI with display RAM and character generator ROM.
- Direct interface to 4 or 8 bit CPU.
- 11 commands for control.
- EL backlightable module is available as an option.

Dot Matrix Dimensions (in millimeters)



## Mechanical Characteristics

Item	Specification	Unit
Outline Dimension	126 W x 36 H x 12 D	mm
Character Size	3.2 W x 7.95 H	mm
Number of Characters	24 x 1 (24) Characters (5 x 10 font + cursor)	-
Viewing Area	100 W x 13.8 H	mm
Bezel Opening	100 W x 13.8 H	mm
Dot Size	0.8 W x 0.75 H	mm
Dot Pitch	0.85 W x 0.9 H	mm
Weight	approx. 40	gram

## Absolute Maximum Ratings

Item	Symbol	Rating	Unit
Supply Voltage	$V_{CC}$	7	V
Input Voltage	$V_{in}$	$0 < V_{in} < V_{CC}$	V
Operating Temperature	$T_{op}$	0 to $+50$	$^{\circ}C$
Storage Temperature	$T_{stg}$	$-20$ to $+70$	$^{\circ}C$

Electrical Characteristics ( $T_A = 25^{\circ}C$ )

Item	Symbol	Min.	Typ.	Max.	Unit
Supply Voltage	$V_{CC}$	4.75	5.0	5.25	V
	GND	-	0	-	V
"H" Level Input Voltage ( $V_{in} = 5.0V$ )	$V_{iH}$	2.2	-	-	V
"L" Level Input Voltage ( $V_{in} = 5.0V$ )	$V_{iL}$	-	-	0.6	V
"H" Level Output Voltage ( $I_{OL} = 0.2 mA$ )	$V_{oH}$	2.4	-	-	V
"L" Level Output Voltage ( $I_{OH} = 1.2 mA$ )	$V_{oL}$	-	-	0.4	V
Power Consumption	$P_D$	-	10	-	mW

Optical Characteristics ( $T_A = 25^{\circ}C$ ,  $\phi = 25^{\circ}$ ,  $\theta = 0^{\circ}$ )

Item	Symbol	Min.	Typ.	Max.	Unit
Viewing Angle	$\phi$	10	25	40	degree
Contrast	K	-	3.0	-	-
Turn On Time	$t_{on}$	-	200	400	ms
Turn Off Time	$t_{off}$	-	250	400	ms

Note: Refer to Applications Section for the following definitions: (a)  $\phi$  and  $\theta$ , Viewing Angle; (c) Contrast; (d) Turn On and Turn Off Time.

## Connector Pin Assignment

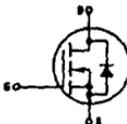
Pin No.	Signal	Function
1	GND	Power Supply
2	$V_{CC}$	
3	$V_D$	LCD Drive Voltage ( $OV$ to $V_{CC}$ )
4	HS	"H" Data Input "L" Command Input
5	R/W	"H" Data Read (Module - CPU) "L" Data Write (CPU - Module)
6	E	Enable Signal
7	DB0	Data Bus 8-bit Use 4-bit Use
8	DB1	
9	DB2	
10	DB3	
11	DB4	
12	DB5	
13	DB6	
14	DB7	

## Power Field Effect Transistor

### N-Channel Enhancement-Mode Silicon Gate TMOS

These TMOS Power FETs are designed for low voltage, high speed power switching applications such as switching regulators, converters, solenoid and relay drivers.

- Silicon Gate for Fast Switching Speeds
- Low  $r_{DS(on)}$  to Minimize On-Losses. Specified at Elevated Temperature
- Rugged — SOA is Power Dissipation Limited
- Source-to-Drain Diode Characterized for Use With Inductive Loads



**IRF520**  
**IRF521**  
**IRF522**  
**IRF523**

TMOS POWER FETs  
7 and 8 AMPERES  
 $r_{DS(on)}$  = 0.3 OHMS  
50 and 100 VOLTS  
 $r_{DS(on)}$  = 0.4 OHMS  
50 and 100 VOLTS



CASE 251A-04  
(TO-220AR)

#### MAXIMUM RATINGS

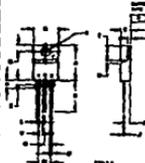
Rating	Symbol	IRF				Unit
		520	521	522	523	
Drain-Source Voltage	$V_{DS}$	100	80	100	80	Vdc
Drain-Gate Voltage ( $V_{GS} = 30$ V)	$V_{DG}$	100	80	100	80	Vdc
Gate-Source Voltage	$V_{GS}$	±20				Vdc
Drain Current Continuous, $T_C = 25^\circ\text{C}$ $T_C = 100^\circ\text{C}$ Puls, $T_C = 25^\circ\text{C}$	$I_D$	8 5	7 4	7 4	28	Adc
Total Power Dissipation @ $T_C = 25^\circ\text{C}$ Derate above $25^\circ\text{C}$	$P_D$	40 0.32				Watts W/°C
Operating and Storage Temperature Range	$T_J, T_{STG}$	-55 to 150				°C

#### THERMAL CHARACTERISTICS

Thermal Resistance — Junction to Case	$R_{\theta JC}$	3.12	°C/W
— Junction to Ambient	$R_{\theta JA}$	62.5	°C/W
Maximum Lead Temp. for Soldering Purposes, 1" from Case for 5 Seconds	$T_L$	300	°C

See the MTP5112E Designer's Data Sheet for a complete set of design curves for the product on this data sheet.

#### OUTLINE DIMENSIONS



PHYSICAL DIMENSIONS  
DIMENSIONS IN MILLIMETERS  
DIMENSIONS IN INCHES

NOTES:  
1. DIMENSIONS ARE MEASURED AT 25°C.  
2. DIMENSIONS ARE MEASURED AT 25°C.  
3. DIMENSIONS ARE MEASURED AT 25°C.  
4. DIMENSIONS ARE MEASURED AT 25°C.

Symbol	Dimension	Value
A	Lead Length	1.27
B	Lead Spacing	2.54
C	Lead Width	0.76
D	Case Width	6.35
E	Case Height	2.54
F	Case Depth	1.27
G	Case Thickness	0.76
H	Case Diameter	10.16
I	Case Diameter	10.16
J	Case Diameter	10.16
K	Case Diameter	10.16
L	Case Diameter	10.16
M	Case Diameter	10.16
N	Case Diameter	10.16
O	Case Diameter	10.16
P	Case Diameter	10.16
Q	Case Diameter	10.16
R	Case Diameter	10.16
S	Case Diameter	10.16
T	Case Diameter	10.16
U	Case Diameter	10.16
V	Case Diameter	10.16
W	Case Diameter	10.16
X	Case Diameter	10.16
Y	Case Diameter	10.16
Z	Case Diameter	10.16

**Designer's Data Sheet**  
**Power Field Effect Transistor**  
**P-Channel Enhancement-Mode**  
**Silicon Gate MOS**

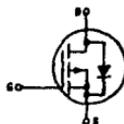
These MOS Power FETs are designed for medium voltage, high speed power switching applications such as switching regulators, converters, solenoid and relay drivers.

- Silicon Gate for Fast Switching Speeds — Switching Times Specified at 100°C
- Designer's Data —  $I_{DSS}$ ,  $V_{DS(on)}$ ,  $V_{GS(th)}$  and SOA Specified at Elevated Temperatures
- Rugged — SOA is Power Dissipation Limited
- Source-to-Drain Diode Characterized for Use With Inductive Loads



**MTM8P08**  
**MTM8P10**  
**MTP8P08**  
**MTP8P10**

**TMOS POWER FETs**  
**8 AMPERES**  
 **$V_{DS(on)}$  = 8.4 OVM**  
**80 and 100 VOLTS**



**MAXIMUM RATINGS**

Rating	Symbol	MTP or MTP		Unit
		8P08	8P10	
Drain-Source Voltage	$V_{DS}$	80	100	Vdc
Drain-Gate Voltage ( $V_{GS} = 1$ MHz)	$V_{DGS}$	80	100	Vdc
Gate-Source Voltage — Continuous — Non-repetitive ( $I_D = 80$ mA)	$V_{GS}$ $V_{GS(th)}$	±20 ±40		Vdc Vdc
Drain Current — Continuous — Pulsed	$I_D$ $I_{DM}$	8 25		Adc A
Total Power Dissipation @ $T_C = 25^\circ\text{C}$ Derate above $25^\circ\text{C}$	$P_D$	75 5.6		Watts W/°C
Operating and Storage Temperature Range	$T_J, T_{stg}$	-85 to 150		°C

**THERMAL CHARACTERISTICS**

Thermal Resistance Junction to Case Junction to Ambient	Symbol	°C/W	
		$R_{\theta JC}$	$R_{\theta JA}$
TO-204	$R_{\theta JA}$	1.87	
		30	
TO-220	$R_{\theta JA}$	62.5	
Maximum Lead Temperature for Soldering Purposes, 1.5" from case for 5 seconds	$T_L$	275	°C



**MTM8P08**  
**MTM8P10**  
**TO-204A**



**MTP8P08**  
**MTP8P10**  
**TO-220B**

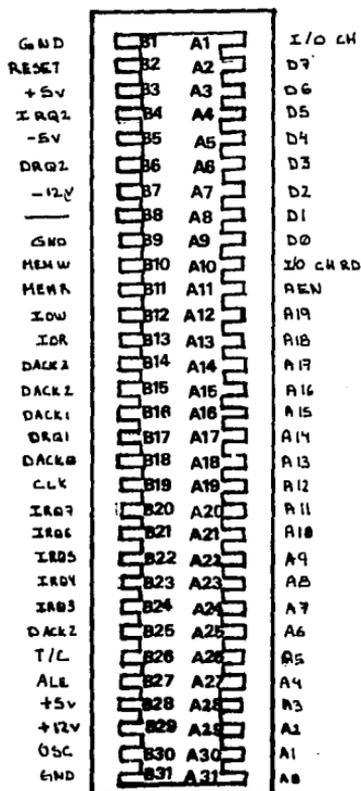
Designer's Note for "Mount Care" Qualification — The Designer's Data Sheet permits the design of most circuits entirely from the information presented. RGA Lead curves — representing mechanical or stress characteristics — are given to facilitate "mount care" design.

**MOTOROLA TMOS POWER MOSFET DATA**

3-457

## APENDICE B NORMAS PARA TARJETAS DE INTERFACE EN UNA PC XT

Las tarjetas de interface en una computadora se conectan en los slots de la tarjeta madre, los cuales, en general, son de 62pines y en cada pin está presente una señal; la mayoría de las señales son de nivel lógico TTL, salvo las señales de voltaje (Vcc y tierra). A continuación se muestra una figura con los pines del slot y las señales que corresponden a cada uno.



Cabe señalar que las tarjetas de interface tienen que tener como altura máxima 14 cm, en cuanto al espesor, este no debe ser mayor de 3 mm y el largo puede variar pero sin exceder de 24 cm.

Entre las señales más importantes se encuentran:

- El bus de datos, cual está compuesto por ocho líneas bidireccionales las cuales pueden transmitir información entre el microprocesador, la memoria y cualquier dispositivo externo. Durante un ciclo de escritura los datos aparecen en el slot para ser captados por otros dispositivos. En un ciclo de lectura el microprocesador realiza la transferencia de datos de un dispositivo hacia otras partes.

- El bus de direcciones, está constituido por 20 líneas unicamente de salida. Estas líneas son manejadas por el microprocesador para direccionar algún dispositivo, aunque para dispositivos de entrada o salida (I/O) solo se emplean las direcciones de A0 a A9.

- Address Enable ( AEN ) esta señal se activa con un voltaje alto e indica cuando se encuentra presente un ciclo de acceso directo a memoria. Cuando esto sucede el microprocesador se desconecta y el DMA controla el acceso de información de cualquier dispositivo de I/O.

- Input/Output Read ( IOR ) esta señal se activa con un nivel bajo de voltaje y sincroniza la lectura de un dispositivo de entrada/salida.

- Input/Output Write ( IOW ) esta señal se activa con un nivel bajo de voltaje y sincroniza la escritura de un dispositivo de entrada/salida.

- Reset esta señal es unicamente de salida y se activa con un nivel alto de voltaje durante el encendido del sistema y permanece en activo hasta que todos los niveles lógicos alcanzan un estado

de operación y después permanece inactiva.

- La fuente de +5 volts.
- La tierra del sistema.

Cuando se agrega una tarjeta de interface al slot de la PC se debe tener en cuenta que ningún bus puede ser cargado en exceso, es decir, se debe saber cual es la máxima corriente drenada por cada señal tanto en niveles altos como bajos. Además es importante saber que cada vez que se aumenta una tarjeta se incrementa una capacitancia que produce distorsión en las señales del slot.

Para evitar estos problemas se pueden seguir una reglas sencillas:

- La primera es no conectar ningún circuito NMOS LSI directamente a las señales, porque estos circuitos tienen una baja capacidad de drenaje de corriente, además no soportan picos negativos de voltaje.
- La segunda es no cargar ninguna señal del slot con más de dos circuitos LS.
- La tercera es utilizar capacitores de desacoplo del orden de 0.01 micro-Farads entre las fuentes de +5 volts y tierra.
- Y la cuarta es no mandar las señales del slot a grandes distancias pues estas pueden distorsionarse.

Por otro lado, es necesario conocer el mapa de puertos de la computadora para poder elegir un lugar que no esté ocupado por otro dispositivo de I/O. Más específicamente las direcciones que

pueden ser direccionadas en los slots de la PC van de la dirección 0200H a la dirección 03FFH, sin embargo en estos 512 bytes ya están restringidas algunas secciones como se puede ilustrar en la siguiente figura:.

	DIRECCIONES	USO
1	0200H	SIN USO
1	0201H	CONTROL DE JUEGO
118	0202H-0277H	SIN USO
8	0278H-027FH	IMPRESORA PTO. 2
120	0280H-027FH	SIN USO
8	02F8H-02FFH	PUERTO SERIE 2
120	0300H-0377H	SIN USO
8	0378H-037FH	IMPRESORA PTO. 1
48	0380H-03AFH	SIN USO
16	03B0H-03BFH	MONITOR MONOCRO.
16	03C0H-03CFH	SIN USO
16	03D0H-03DFH	GRAFICAS/COLOR
16	03E0H-03EFH	SIN USO
8	03F0H-03F7H	5 <sup>1/4</sup> DRIVER
8	03F8H-03FFH	PUERTO SERIE 1

## APENDICE C. FUNCIONAMIENTO DEL SISTEMA

Al encender el sistema, éste realizará una autoprueba, en caso de falla en la RAM del sistema se desplegará un mensaje de aviso al operador, si la PC no está lista aparecerá un letrero en la pantalla indicando lo anterior hasta que se ejecute el programa FELPC1 el cual establece la comunicacion de la PC con el sistema. Este programa solo podra ejecutarse en la unidad B o C de la computadora ya que en la unidad A se tendrá el disco de trabajo. En este momento, en la pantalla aparecerá el letrero de menú principal al cual se tiene acceso oprimiendo la tecla de flecha abajo. De esta forma aparecerán las siguientes opciones:

- 1.- GRABAR
- 2.- REPRODUCIR
- 3.- BORRAR
- 4.- FORMATEAR
- 5.- SALIR

ESCRIBE EL NUMERO DE TU ELECCION

Al termino del letrero anterior se debe escribir un número para tener acceso a alguna de las opciones presentadas. Si por equivocación se oprime una tecla que no corresponda a los números anteriores se escuchará un beep indicando el error y el cursor esperará el número correcto.

Si el número elegido fué el uno, el sistema empezará a tomar datos de la entrada de audio para digitalizarlos y después grabarlos en floppy, al mismo tiempo se escuchará la música y aparecerá el letrero de GRABANDO. Al termino de la grabación se escuchará un beep indicando el final. En la pantalla aparecerá nuevamente el letrero de menú principal.

Si la opción elegida fué el número dos, el sistema empezará a bajar los datos almacenados en floppy hacia la RAM de la PC y al

termino de esta operación los datos serán convertidos a una señal analógica que a su vez será enviada al amplificador para ser escuchada. De nueva cuenta al final de la reproducción se escuchará un beep y en la pantalla aparecerá el letrero de menú principal.

Si el número elegido fué el tres, el sistema comenzará a llenar con FFH el floppy y en la pantalla aparecerá el letrero de borrando. Al termino de esta operación se escuchará un beep y aparecerá en la pantalla otra vez el letrero de menú principal.

Si se oprimió el número cuatro, en la pantalla se desplegará el letrero "SEGURO DESEAS FORMATEAR?", si se oprime cualquier tecla que no sea la "s" el sistema regresará nuevamente al menú principal. si se oprime la "s" en la pantalla aparecerá el letrero FORMATEANDO y el floppy será formateado. Nuevamente, al termino de este commando se escuchará un beep y el sistema regresará al menú principal.

Si la opción elegida fué la cinco, el sistema mandará a pantalla el siguiente letrero: "NO OLVIDES APAGARME". En este momento se puede poner en off el switch de encendido. Y automáticamente el programa FELPC1 de la PC finalizará su ejecución.

Cabe señalar que en las primeras tres opciones se puede interrumpir la operación oprimiendo cualquier tecla, si esto sucede el sistema llegará nuevamente al menú principal.

Si el sistema llegará a inhibirse se puede hacer uso del botón de reset para que nos envíe al menú principal.

## APENDICE D CALCULOS DEL AMPLIFICADOR DE POTENCIA

CONDICIONES DE DISEÑO:

$$V_{CC}^+ = V_{CC}^- = 35V$$

$$R_L = 8 \text{ ohm}$$

$$Q1: V_{EC1} = V_{CC}^+ - 0.8V + V_E = 35.2V$$

$$\beta_1 = 125$$

$$V_{EB1} = 0.6V$$

$$I_{C1} = 1 \text{ mA}$$

$$Q2: V_{EC2} = V_{CC}^+ + V_E = 36V$$

$$\beta_2 = 125$$

$$V_{EB1} = 0.6V$$

$$I_{C2} = 1 \text{ mA}$$

$$HFT1: V_{DS1} = 35V$$

$$V_{GS1} = 3V$$

$$HFT2: V_{SD2} = 35V$$

$$V_{SG2} = 3V$$

$$Q3: V_{CE3} = V_{GS1} + V_{GS2} = 6V$$

$$V_{BE3} = 0.6V$$

$$\beta_3 = 150$$

$$I_{C3} = 5.5 \text{ mA}$$

} punto medio de  $R_{13}$

$$V_{CE3 \text{ max}} = 9V$$

$$V_{CE3 \text{ min}} = 4V$$

$$\begin{aligned}
 Q4: \quad V_{CE4} &= -V_{CC}^- - V_{SG2} = 32V \\
 V_{BE4} &= 0.6V \\
 \beta_4 &= 150 \\
 I_{C4} &= 6 \text{ mA}
 \end{aligned}$$

ECUACIONES:

$$V_{CC}^+ - (R_4 + R_5) I_E - V_{EC2} - V_{CC}^- = 0 \dots (1)$$

$$V_{CC}^+ - (R_4 + R_5) I_E - V_{EB1} - R_2 I_E / (\beta_1 + 1) = 0 \dots (2)$$

$$I_E = I_{E1} + I_{E2} \dots (3)$$

$$V_{CC}^+ - V_{G1} - (R_9 + R_{10}) I_X = 0 \dots (4)$$

$$V_A - R_7 I_{C4} / \beta_4 - V_{BE4} - V_{CC}^- = 0 \dots (5)$$

CALCULOS:

(de 1):  $R_4 + R_5 = 17 \text{ Kohm}$

si  $R_5 \ll R_4$  entonces:  $R_5 = 0.1 R_4$

$$V_X = (V_{CC}^+ - V_E) R_4 / (R_4 + R_5) = 30.9V \text{ y si } V_E = 1V$$

$$R_5 = (V_{CC}^+ - V_X - V_E) / I_E = 1.55 \text{ Kohm}$$

$$R_4 = 17 \text{ Kohm} - R_5 = 15.45 \text{ Kohm}$$

(de 2)  $R_2 = R_8 = (V_E - V_{EB1}) (\beta_1 + 1) / I_{E1} = 50.4 \text{ Kohm}$

$$I_X = I_{C4} = 6 \text{ mA}$$

$$(de\ 4) \quad (R_9 + R_{10}) = (V_{cc}^+ - V_{G1}) / I_X = 5.33Kohm$$

$$Para\ que\ V_{x2} = (V_{cc}^+ - V_{G1}) / 2 \quad R_9 = R_{10} = 2.7Kohm$$

$$(de\ 5) \quad R_7 = (V_A - V_{BE4} - V_{cc}^-) / \beta_4 / I_{C4} = 5Kohm$$

$$R_8 = (V_A - V_{cc}^-) / I_{C1} = 800ohm$$

$$V_{CE3} = (R_{12} + R_{13} + R_{14}) V_{BE3} / (R_{14} + X R_{13})$$

donde X es la constante del potenciómetro  $R_{13}$

$$Para\ el\ punto\ medio\ (X=0.5) \quad I_{R12} = I_X - I_{C3} = 0.5\ mA$$

$$(R_{12} + R_{13} + R_{14}) = V_{CE3} / I_{R12} = 12Kohm$$

$$Para\ X=0 \quad R_{14} = 12Kohm (V_{BE3}) / V_{CE3\ max} = 800ohm$$

$$Para\ X=1 \quad R_{13} = 12Kohm (V_{BE3}) / V_{CE3\ min} - R_{14} = 1Kohm$$

$$R_{12} = 12Kohm - R_{13} - R_{14} = 10.2Kohm$$

Así, utilizando valores comerciales se tiene:

$$R_2 = 47Kohm$$

$$R_6 = 47Kohm$$

$$R_{14} = 820ohm$$

$$R_3 = 820ohm$$

$$R_9 = 2.7Kohm$$

$$R_4 = 15Kohm$$

$$R_{10} = 2.7Kohm$$

$$R_5 = 1.5Kohm$$

$$R_{12} = 10Kohm$$

$$R_7 = 4.7Kohm$$

$$R_{13} = 1Kohm$$

## APENDICE E ALGORITMO DE COMPACTACION DIGITAL

Una alternativa para optimizar el tiempo de grabación en un disco flexible es por medio de la utilización de técnicas de compactación de la información digital.

Dadas las características de la señal de audio, se diseñó un algoritmo semejante al de Compactación Delta utilizado comunmente en compactación de señales de video, el cual se basa en el almacenamiento de las diferencias entre dos imagenes subsecuentes.

El algoritmo de compactación diseñado consiste en tomar un dato inicial denominado semilla, el cual es tomado como base de comparación de los datos subsecuentes. El segundo dato recibido se compara con la semilla, si la diferencia existente entre estos dos datos está en el rango de -7 a 7 y el dato no es igual a la semilla se escribe la diferencia utilizando solo cuatro bits en magnitud signada como se observa en la siguiente tabla:

Diferencia	Magnitud Signada	Hexadecimal
-7	1111	F
-6	1110	E
-5	1101	D
-4	1100	C
-3	1011	B
-2	1010	A
-1	1001	9
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7

Como se puede observar en la tabla anterior, no se utiliza el 1000 (8H) dado que significaría una diferencia igual a cero, es decir, que ambos datos son iguales lo cual ya esta contemplado por el código 0000. Aprovechando esta situación podemos utilizar el código 1000 (8H) en combinación con otros cuatro bits para indicar 3 situaciones especiales para la compactación de la información, las cuales son:

Código de semilla: 1000 0000 (semilla en 8 bits)  
Código de repeticiones: 1000 (repeticiones en 4 bits)  
Código de fin: 1000 0001

En caso de que la diferencia este fuera del rango mencionado anteriormente, se toma una nueva semilla indicandose ésta con el código 80H seguido por el nuevo valor de la semilla

Cuando el dato comparado es igual con el anterior, se espera un nuevo dato, si este otro dato continua siendo igual, se lleva una cuenta del número de repeticiones ocurridas, utilizandose el código 8H seguido por el número de repeticiones en 4 bits con lo cual podemos representar hasta 15 repeticiones de un mismo valor. En el caso de que la repetición es única se utiliza el código 0000 directamente. En el caso de que sean más de quince repeticiones se utilizaría nuevamente el criterio antes mencionado.

De acuerdo a la lógica anterior el código 81H nunca se utilizaría para la representación de repeticiones pues indicaría una repetición única la cual ya esta contemplada por el código 0000. Así que es posible utilizarlo como un código especial que indique el fin de los datos compactados.

Todo este algoritmo se aplica secuencialmente a cada uno de los datos provenientes del convertidor analógico-digital comparandose el dato presente con su inmediato anterior.

Mediante esta técnica de compactación si la señal fuera constante se podrían grabar hasta 11 minutos 45 segundos, dado que se utilizarían un byte por cada 15 datos.

En el peor de los casos se grabarían solamente 23 segundos dado que se grabarían puros códigos de semilla (2 bytes por cada dato). Analizando un caso intermedio entre los dos mencionados se podría esperar grabar aproximadamente 6 minutos.

Hay que tomar en cuenta que la eficiencia de grabación de este algoritmo depende de la naturaleza de la señal, teniendo que para una señal más lenta se obtiene una eficiencia mayor que para una rápida dado que las diferencias entre los datos serán menores y con mayor número de repeticiones.

Cabe señalar que la operación del algoritmo fue simulado en el Turbo Debugger y se comprobó que funciona adecuadamente como se explicó anteriormente. Debido al tiempo consumido en el diseño y construcción del SGDA no se incluyó dentro del software del manejo del sistema.

## GLOSARIO

**ALU:** Unidad aritmética lógica de un microprocesador.

**BANDERA:** Es un indicador del estado del procesador después de ejecutar una interrupción.

**BAUD:** Unidad para medir la velocidad de transmisión de dispositivos electrónicos.

**BIOS:** Sistema básico de entrada/salida.

**BIT:** Es la mínima capacidad de almacenamiento.

**BYTE:** Es un conjunto de ocho bits.

**BUFFER:** Es un dispositivo electrónico que puede mantener el valor de un dato a su salida y además es capaz de proporcionar corriente a otros dispositivos.

**BUS:** Es un camino compartido al cual se conectan todas las partes funcionales de la computadora.

**COERCITIVIDAD:** Es la facultad del imán de conservar su imantación frente a un campo magnético contrario.

**DIAPHRAGMA:** Membrana flexible de mica u otros materiales que recibe vibraciones y es capaz de transmitir las a otros dispositivos.

**DIELECTRICO:** Cualquier cuerpo que carece de conductividad.

**DMA:** Direccionamiento directo a memoria.

**DRIVE:** Manejador de disco.

EOC: Fin de conversión.

EPROM: Memoria de solo lectura.

EU: Es la unidad de ejecución.

GND: Voltaje de referencia.

HADWARE: Conjunto de elementos físicos que componen la computadora.

INHIBIR: Detención momentanea del sistema.

LSB: Bit menos significativo.

LATCH: Memoria temporal.

MSB: Bit más significativo.

RAM: Memoria de escritura y lectura.

SISTEMA OPERATIVO: Software que permite manejar la información en la computadora.

SOFTWARE: Conjuto de programas para manejar la computadora.

START: Inicio de conversión.

TRIMMER: Resistencia variable de precisión.

## BIBLIOGRAFIA

1. "AND. LED AND LCD PRODUCTS"  
William J. Purdy Company  
1989
2. "ANALOG-DIGITAL CONVERSION HANDBOOK"  
Analog Devices Staff  
Prentice Hall, Englewood Cliffs, N.J., USA, 1986
3. "ANALOG TO DIGITAL/DIGITAL TO ANALOG  
CONVERSION TECHNIQUES"  
David F. Hoeschele Jr.  
John Willeg & Sons, Inc  
N.J., USA, 1968
4. "CONVERTIDOR A/D. UNO PARA CADA NECESIDAD"  
Shaland, A.  
Mach Des., Vol 62, Núm 7, Abr 1990, P.66-93
5. "DISK MEMORIES: WHAT YOU SHOULD  
KNOW BEFORE YOU BUY THEM"  
Bursky, Dave  
Personal Computing, Abril 1981
6. "DISPLAYS DE CRISTAL LIQUIDO PARA CONTROLADORES"  
Klingman, Ed.  
Electr. Des., Vol 36, Núm 17, Jul 1988, P.137-140
7. "DISTINTOS MEDIOS DE ALMACENAMIENTO MASIVO DE DATOS  
EN MICROCOMPUTADORAS"  
Solomon, L. et al.  
Computers and Electronics, Vol 21, Núm 7, Jul 1983, P.36-48

8. "Data Compression Engines"  
Dr. Dobb's Journal, Núm 173, Feb 1991, P.38-52
9. "ELECTRONICA TEORIA DE CIRCUITOS"  
Boylestad  
Nasheisky  
Prentice Hall 1989
10. "INTERCONEXION DE PERIFERICOS A MICROPROCESADORES"  
Serie:Mundo Electrónico  
Ed.Marcombo, Barcelona, España, 1983
11. "INTERFACING TO THE IBM PERSONAL COMPUTER"  
Eggebrecht, Lewis C.  
Howard W. Sams & Co.  
Indianapolis, USA, 1986
12. "LA UNIDAD DE DISCO FLOPTICAL CON CAPACIDAD  
DE 20.8MB UTILIZA DISCOS FLEXIBLES"  
Electr.Des., Vol 36, Núm 18, Ago 1988, P.65-68
13. "LOGICA DIGITAL Y DISEÑO DE COMPUTADORAS"  
Morris Mano, M.  
Prentice Hall 1989
14. "MAGNETIC RECORDING"  
Computer Data Storage Vol. 2  
Denismee, C. Damei, Erick D.  
McGraw-Hill, USA, 1988
15. "MICROPROCESSORS"  
INTEL Corporation, Illinois, USA, 1990
16. "MICROPROCESSORS AND INTERFACING"  
Hall, Douglas V.  
McGraw-Hill, Singarore, 1986

17. "NUEVA TECNOLOGIA PARA DISCOS/DISKETTES"  
Rosch, W.L./ Feldman, P.  
PC MAGAZINE, Vol 4, Núm 10, May 1985
  
18. "PANTALLAS; NUEVOS TIPOS Y MEJOR DEFINICION"  
NEW ELECTR., Vol 22, Núm 10, Dic 1986, P.35
  
19. "PERIPHERALS"  
INTEL Corporation, Illinois, USA 1990
  
20. "POWER MOSFET TRANSISTOR DATA"  
Motorola Inc., 1989
  
21. "PRINCIPLES OF DIGITAL AUDIO"  
Ken C. Pohlmann  
Howard W. Sams & Co, 1988
  
22. "TIPOS PARA DISEN DE BLINDAJE CONTRA  
INTERFERENCIA MAGNETICA"  
Brewer, R.  
Mach Des, Vol. 61, # 6, Marzo 1989. P.85-88