

9 2ej



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE ESTUDIOS SUPERIORES
"CUAUTITLAN"

DISEÑO DE UN SISTEMA DE CONTROL POR MICROPROCESADOR PROGRAMADO EN BASIC Y EN LENGUAJE DE MAQUINA

T E S I S

QUE PARA OBTENER EL TITULO DE :
INGENIERO MECANICO ELECTRICISTA

P R E S E N T A N :

JORGE JAVIER DIEZ DE SOLLANO MONTES DE OCA

DIRECTOR DE TESIS: ING. JORGE BUENDIA GOMEZ

CUAUTITLAN IZCALLI, EDO. MEX.

1990

TESIS CON
FALLA DE ORIGEN



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

I N D I C E

I.	INTRODUCCION	
II.	LA SINCLAIR ZX-81.	
A.	DESCRIPCION DEL FUNCIONAMIENTO	1
B.	CONEXION DEL Z80A-P10 (INTERFASE PARALELO).....	3
C.	COMANDOS DEL LENGUAJE BASIC	6
III.	DISPOSITIVOS Y CIRCUITOS	
A.	DESCRIPCION DEL DTMF	11
B.	DTMF CON CIRCUITOS INTEGRADOS	20
C.	C.I. RECEPTOR DE TONOS DTMF.....	28
D.	ESTRUCTURA DE UNA COMPUTADORA DIGITAL.....	32
E.	EL MICROPROCESADOR Z-80A	38
F.	CONJUNTO DE CARACTERES DE LA ZX-81	42
G.	CONVERSION DIGITAL-ANALOGICA	47
H.	PROGRAMACION DEL Z80-P10	48
I.	TABLA DE INSTRUCCIONES DEL Z80	58
J.	CONVERTIDOR ANALOGICO- DIGITAL TLC548	95
IV.	GENERADOR DE TONOS DTMF DISCRETO	
A.	DIVISION DE FRECUENCIA	98
B.	MULTIPLEXIONES	101

V. CONTROL DE LOS DISPOSITIVOS

A. PRUEBAS DE LABORATORIO	104
ANEXO 1	110
ANEXO 2: DIAGRAMAS DE LOS CIRCUITOS INTEGRADOS	136
BIBLIOGRAFIA	148

L I S T A D E F I G U R A S

FIGURA	N O M B R E	PAGINA
1 - 1	CONEXIONES DEL Z80AP10 CON LA SINCLAIR ZX81	4
1 - 2	CONEXIONES DE LA TARJETA ZX81	5
2 - 1	CIRCUITO TIPICO DTMF	12
2 - 2	GENERADORES DTMF	21
2 - 3	TECLADOS DTMF	23
2 - 4	DIAGRAMA A BLOQUES DEL TCM5087	24
2 - 5	APLICACION DEL C.I. TCM5087	27
2 - 6	CARACTERISTICA IDEAL DEL FILTRO DTMF	29
2 - 7	DIAGRAMA DEL RECEPTOR DTMF	30

FIGURA	N O M B R E	PAGINA
3 - 1	CONVERTIDOR DIGITAL-ANALOGICO DE 5 BITS	46
3 - 2	LOGICA PARA OBTENER UNA ONDA SENOIDAL	48
4 - 1	REGISTROS DEL Z80 P10	51
4 - 2	TABLA DE INSTRUCCIONES DEL Z80A	58
5 - 1	CONVERTIDOR ANALOGICO DIGITAL TLC548	96
5 - 2	IMPLEMENTACION DE DIVISORES PRO GRAMABLES	102
	DETECTOR DE PICO POSITIVO	106
	DETECTOR DE TONO DE 440 HZ.	107
	CODIGO HEXADECIMAL Y BINARIO 2 DE 8	111
	APROXIMACION DE 16 ESCALONES A UNA ONDA SENOIDAL	112

I. INTRODUCCION

Las computadoras basadas en microprocesadores (microcomputadoras), son productos de propósito general que pueden configurarse para servir en un amplio campo de aplicaciones tales como análisis de circuitos, procesamiento de palabras, juegos, etc. Otra forma en que los microprocesadores pueden ser usados es una aplicación específica, por ejemplo, el control de un motor (microcontrolador). Los microprocesadores y sus dispositivos de soporte son útiles sólo si tienen integrado un software (por ejemplo BASIC), algunas aplicaciones de control no pueden implementarse en Basic, porque es muy lento y requiere más memoria que los programas equivalentes en lenguaje de máquina, de modo que un programa de control industrial puede estar escrito en Basic con algunas llamadas para subrutinas en lenguaje de máquina.

La ZX81 es una computadora pequeña que se programa en lenguaje BASIC, y funciona para manejo de datos o como calculadora. En el presente trabajo se pretende modificarla para utilizarla en tareas de control de dispositivos electrónicos tales como generadores de tonos y convertidores analógico-digital, se escogió controlar un generador de tonos multifrecuenciales porque éstos están sustituyendo actualmente el marcado por pulsos en los sistemas telefónicos.

Aunque existen circuitos integrados que interfazan con un microprocesador para generar tales tonos, estos circuitos no se consiguen fácilmente por lo que se pensó en implementar uno con circuitos disponibles, TTL y CMOS; para el receptor de tonos DTMF y el convertidor analógico-digital de 8 BITS con salida Serie, si se emplearon circuitos integrados especiales que facilitan la programación.

II. LA SINCLAIR ZX-81

A. DESCRIPCION DEL FUNCIONAMIENTO.

El microprocesador Z80A como CPU.

El sistema operativo esta contenido en un C.I. ROM con 8 Kbytes de BASIC. En las direcciones de memoria de 0 a 8191.

La memoria RAM tiene las direcciones 16384 a la 32767 (16 Kbytes) y esta organizada dividida en diferentes areas para almacenar distintos tipos de informacion. Las areas tienen solo espacio suficiente para la informacion que contienen actualmente, y si uno inserta mas en un punto dado (por ejemplo, poner otra linea o variable), la computadora le hace espacio recorriendo todo lo que va arriba de ese punto. Inversamente si uno borra informacion, todo lo que va despues se reposiciona hacia atras.

Los bytes en memoria de 16384 a 16508 son para usos especificos del sistema y son llamadas variables del sistema, por lo que programas del usuario comienzan en la localidad 16509.

Hay dos metodos usados por computadores para convertir un programa escrito en lenguaje de alto nivel (BASIC) en una forma en lenguaje de maquina:

- i. Usando un compilador. Antes de ejecutar el programa, el compilador se usa para traducir el programa en un programa en codigo de maquina. Este programa en

codigo de maquina puede entonces ejecutarse las veces necesarias.

ii. Usando un interprete El programa se ejecuta y conforme el control alcanza cada enunciado, este es convertido por el interprete en forma de codigo de maquina que la computadora puede entender.

La ZX81 usa un interprete de BASIC. es posible en la ZX81 ejecutar un programa en codigo de maquina directamente sin tener que usar el interprete BASIC. se tienen dos ventajas al hacer esto:

i. Programas en lenguaje de maquina usan menos espacio en memoria que equivalentes programas en lenguaje de alto nivel.

ii. programas en lenguaje de maquina son ejecutados mas rapido que equivalentes programas en lenguaje de alto nivel.

Para ejecutar un programa en codigo de maquina en la ZX81, una seccion de memoria RAM debe estar reservada para la lista de dos digitos de codigo hexadecimal representando cada instruccion puedan almacenarse.

Las desventajas de programas en lenguaje de maquina incluyen que son dificiles de entender, y el proceso es tedioso.

B. CONEXION DEL X80-P10 A LA ZX81 (INTERFASE PARALELO)

CONEXION DE UN F10 A LA ZX81

Para que la computadora pueda usarse para tareas de control. Es decir capaz de recibir datos del exterior, procesarlos, y transferir senales. es necesario conectarle un circuito integrado Z80A-P10 (parallel input output).

Este C.I. tiene dos puertos, PORT A y PORT B, cada uno con 8 lineas las cuales pueden ser entradas o salidas. los detalles de su programacion se dan mas adelante.

En la parte posterior de la ZX81 hay un conector con senales desde la CPU. y los voltajes de control son compatibles con los niveles TTL. Para la conexion del F10 usamos el circuito de la figura 1-1. la figura 1-2 es el diagrama del conector de la ZX81.

En el diagrama del circuito estan las conexiones a la ZX81 en el lado izquierdo del F10. Las lineas de datos D0 a D7 se conectan directo (los numeros en los cuadros se refieren a la figura 1-2), asi como las lineas M1, RD, O, y IORQ. Esta ultima linea es I/O request. y es cero, si el procesador ejecuta un comando IN o OUT.

La linea C/D SEL determina cuando una senal de control o un dato se envia del procesador al F10. Si esta linea es 0 es un dato, si es 1 es una senal de control. Esta linea se conecta con la linea de direccion A1.

CONEXION DEL Z80A PIO CON
LA SINCLAIR Z881

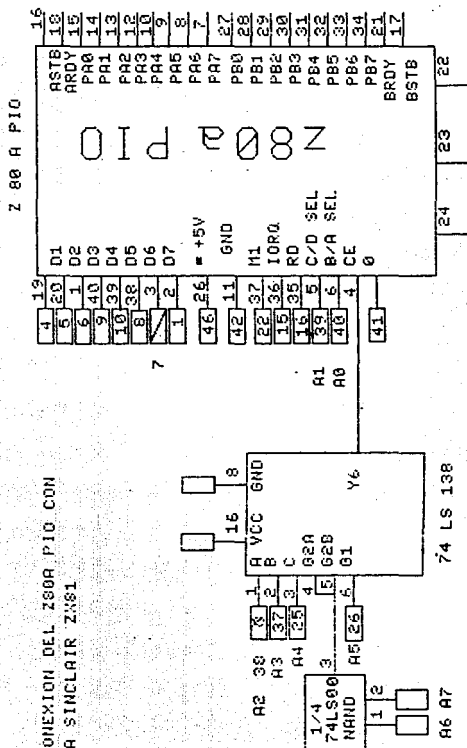


FIG 1 - 1

CONECTOR DE LA 2-21

FIN	
1	D7
2	RAM CS
3	NC
4	D0
5	D1
6	D2
7	D6
8	D5
9	D3
10	D4
11	INT
12	NM1

FIN	
13	HALT
14	MREQ
15	IORQ
16	RL
17	WR
18	BUSACK
19	WAIT
20	BUSRW
21	RESET
22	M1
23	RFSH
24	ROM CS

FIN	
25	A4
26	A5
27	A6
28	A7
29	A8
30	A9
31	A10
32	A11
33	A12
34	A13
35	A14
36	A15

FIGURA 1-1

C) COMANDOS DEL LENGUAJE BASIC

Programamos la T/S 1000 en un lenguaje de computadora llamado BASIC (Beginners All purpose Symbolic Instruction Code). Inventado en el Colegio Dartmouth, a continuacion se describen los comandos de este lenguaje:

NEW ENTER Limpia la memoria

BREAK Detiene el programa

CONT Recomienza el programa despues de haber sido detenido con BREAK o STOP

RUN Corre el programa

RUN # de linea

FOR, NEXT

GOTO

IF

REM "REMARK" identifica una linea que es ignorada por la computadora durante un RUN.

INPUT se detiene la computadora y espera a que el usuario ingrese un numero.

IF Condicion THEN Enunciado

SLOW y FAST la T/S 1000 puede correr (RUN) a dos velocidades, normalmente corre en el modo SLOW y da prioridad a mantener el display y computa durante los periodos cuando la television esta haciendo las partes blancas de la imagen arriba y abajo de la pantalla.

Sin embargo puede ir cuatro veces más rápido, lo hace dando prioridad a la computación y solo manteniendo la imagen cuando no tiene algo más que hacer. SLOW y FAST se pueden usar dentro de programas.

GO SUB N Donde N es el número de la primera línea en la Subrutina es igual que GOTO, excepto en que en que la computadora almacena el número que tiene el enunciado GOSUB para que pueda regresar ahí de nuevo después de efectuar la línea. Esto lo hace poniendo el número de línea (La dirección del retorno) hasta arriba de la pila de ennumerados (el stack GOSUB). RETURN Toma el número de la línea tope del stack GOSUB y va a la línea que sigue de este número.

LEN Es aplicado a un string y el resultado es su longitud (el número de caracteres en el string).

Por Ejemplo:

LEN "GLOVES" = 6 LEN "RHIS" = 4

VHL Este aplicado a un string evalúa ese string como una expresión automática. Por ejemplo (si A=1)

VHL "1+2 * SQRT 4" = 3.5 si el string al cual se aplica VHL contiene variables. Ha. dos restricciones:

i Si la función VAL es parte de una expresión más grande, esta debe ser el primer término por ejemplo:

```
10 LET X = 7 = VAL"Y" debe cambiarse a
10 LET X = VAL"Y" + 7
```

ii VAL solo puede aparecer en la primera coordenada de un enunciado PRINT AT, PLOT o UNPLOT ejemplo:

```
10 PLOTS,VAL "X" debe ser cambiado a
10 LET Y = VAL "X"
15 PLOT 5,Y
```

SUBSTRINGS - Dado un string, un substring consiste de un número de caracteres consecutivos de él, tomados en secuencias.

USANDO EL LENGUAJE DE MAQUINA

Las rutinas en Lenguaje de Máquina pueden ejecutarse desde adentro de un programa en BASIC usando la función USK. El argumento de USK es la dirección de comienzo de la rutina, y su resultado es un entero de dos bytes, sin signo que se despliegan en pantalla en forma decimal y es la suma del valor en el registro BC.

"RETURN". La dirección de "RETURN" (regreso) a BASIC es (stacked) en la manera usual, así el regreso es por una instrucción RET del Z80.

Hay ciertas restricciones en rutinas USR:

(i) En RETURN, el registro i debe tener el valor de 1EH

(ii) La rutina del display usa los registros a, f, ix, iy y r af

Todas las líneas del Procesador están expuestas en la parte posterior de la T/S 1000, así que en principio puede hacerse lo mismo que con un Z80.

Una parte de código de máquina en mitad de la memoria corre riesgo de ser sobrescrita por el sistema BASIC. Algunos lugares más seguros son:

(i) En un enunciado REM con suficientes caracteres para retener el código de máquinas al cual entonces se metera con el enunciado FOR: en esas localidades de memoria (un enunciado REM no se ejecuta).

(ii) En un STRING: Ponga un string de longitud suficiente, y entonces asigne un byte de código de máquina a cada carácter, en ambos casos el código está seguro.

(iii) En el Tope de la Memoria. Cuando la T/S 1000 se enciende, prueba para ver cuanta memoria hay y pone el stack de la maquina justo en el tope asi que no hay espacio ahi para rutinas USR. Almacena la direccion del primer byte inexistente en una variable del sistema conocida como RAMTOP. en los dos bytes con direcciones 16388 y 16389 Utilizando el comando NEW que no efectua prueba de memoria completa . sino solo checa hasta la direccion que este justo antes de la direccion escrita en RAMTOP. Por eso si se escribe con el enunciado FOKE la direccion de un byte existente en RAMTOP para NEW toda la memoria desde ese byte en adelante estara fuera del sistema BASIC y se dejara aparte.

Por ejemplo, suponemos se tiene 11 de memoria y se acaba de encender la computadora:

da la direccion (17408) del primer byte inexistente.

```
PRINT PEEK 16388 + 256*PEEK 16389
```

Ahora, suponiendo que se tiene una rutina USR de 20 bytes se cambia RAMTOP a 17388 = 130 + 256 * 67 asi se mete:

```
FOKE 16388. 130
```

```
FOKE 16389. 67
```

y luego NEW. Los veinte bytes de memoria desde 17388 hasta 17408 se tendran disponibles para programar en lenguaje de maquina.

111. DISPOSITIVOS Y CIRCUITOS

A. DESCRIPCION DEL DTMF (DUAL TONE FREQUENCY)

Algunos aparatos telefonicos usan el metodo de emplear tonos de audio para enviar el numero telefonico. Estos pueden usarse solo si la central telefonica esta equipada para procesar los tonos.

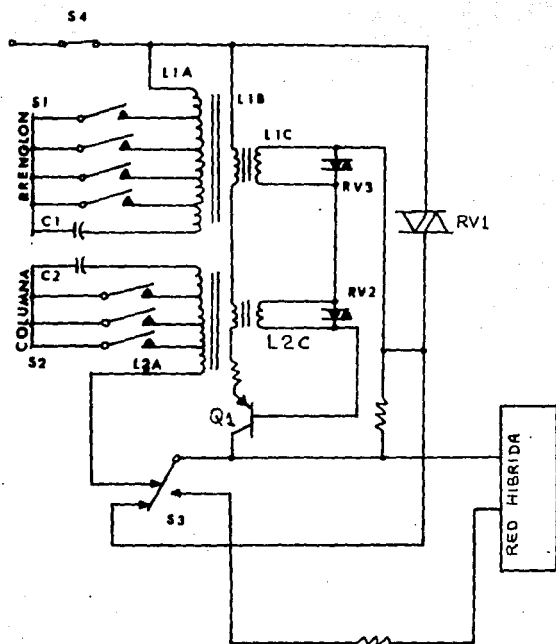
Estos telefonos tienen una botonera de 12 teclas. Presionando cada tecla causa que un circuito electronico genere dos tonos de salida que representan ese numero.

Si el telefono llamado no esta ocupado cuando se intenta la conexi6n, al mismo tiempo que se envia C.A. de 80 volts a 25 Hz a este, se manda un tono de regreso al telefono que origina la llamada (tono de ringback) para indicarle que el otro telefono esta llamando.

Las frecuencias DTMF y la botonera han sido estandarizados internacionalmente, pero las tolerancias en frecuencias individuales pueden variar en distintos paises. El estandar en America del norte es 1.5% para el generador y 2% para el receptor.

GENERACION DE TONOS

Un circuito tipico DTMF se muestra en la figura 2.1.



S1, S2 y S3 cambian de posición cuando se presiona una tecla. S4 es activado por la horquilla. Se muestra una posición de descolgado.

CIRCUITO TÍPICO DTMF

FIGURA 2.1

Los switches S1, S2, y S3 se muestran en posición inactiva. Con el switch de la horquilla en posición de descolgado, el loop de corriente fluye a través de RV1, L1A, L2A, y por la híbrida hacia la línea. El transistor Q1 está en off. Los capacitores C1 y C2 están desconectados de un lado porque los contactos están abiertos.

Cuando se presiona una tecla, acoplamientos mecánicos, llamados vastago renglon y vastago columna, para esta posición de tecla cierran los contactos S1 y S2 apropiados para conectar C1 a una derivación en L1A y C2 a una derivación en L2A. Esto establece los circuitos resonantes requeridos para el tono del grupo bajo (L1A-C1) y el tono del grupo alto (L2A-C2).

El arreglo mecánico de la botonera y los switches es tal que las conexiones del circuito resonante descritas se hacen con depresión parcial del botón. En este punto, S3 sigue en la posición mostrada. Una mayor depresión en el botón causa que S3 cambie de posición. Esto interrumpe el flujo de D.C. entre L1A y L2A y excita los dos circuitos resonantes en oscilación. Al mismo tiempo, S3 conecta el voltaje de batería desde la línea al colector de Q1. El transformador de acoplamiento entre L1A, L1B y L1C, y entre L2A, L2B y L2C causa que Q1 mantenga las oscilaciones y modula el loop de corriente

para transmitir los dos tonos a la central.

DETECCION DE TONOS

Los tonos usados han sido seleccionados cuidadosamente para que el receptor de digitos en la central no los confunda con otros tonos que pueden ocurrir en la linea. El receptor tiene filtros selectivos de frecuencia que pasan solo las frecuencias usadas para DTMF. Tambien cuenta con circuitos de tiempo que aseguran que un tono este presente por un tiempo minimo especificado alrededor de 50 milisegundos antes de ser aceptado como un tono DTMF valido.

Despues de hecha la conexion con el otro telefono, El receptor de digitos queda fuera del circuito y los tonos DTMF pueden transmitirse igual que la voz. Esto permite que se usen los tonos DTMF como comunicaciones de datos para mandar ordenes a terminales remotas o obtener informacion desde una base remota.

COMPARACION EN TIEMPO

El marcado con DTMF es mucho mas rapido en principio y en practica que el discado de pulsos. Usando DTMF, el tiempo requerido para reconocer cualquier digito es de

solo 50 milisegundos con un intervalo interdigito de otros 50 milisegundos. Por eso el tiempo total para mandar cualquier digito es de 100 milisegundos.

En contraste, el discado requiere de 60 milisegundos de abierto y 40 milisegundos cerrado para cada pulso discado para un tiempo total de 100 milisegundos por cada pulso discado. Por eso, para cada digito con numero mayor requiere mas tiempo porque el numero de pulsos por digito aumenta. Tambien el intervalo interdigito es de cerca de 700 milisegundos por cada pulso discado es mayor. Usando el numero 555-555-5555, un tiempo promedio para el discado de un numero de larga distancia es de 11.3 segundos.

Marcando con DTMF el mismo numero tomaria 1 segundo.

Estos tiempos son minimos. Una operacion fisica del dial o de la botonera aumenta estos tiempos.

ACOPLANDO EL GENERADOR DTMF A LA LINEA

Los requerimientos para la interface correcta a la Linea del Generador DTMF son:

- 1.- Los Loop de corriente y los voltajes de DC deben mantenerse para cualquier longitud de Loop.
- 2.- Los tonos deben tener características apropiadas de amplitud y distorsion.

3.- El Generador DTMF debe tener la impedancia apropiada para acopiar a la línea.

ALIMENTACION

Los problemas que se tienen al alimentar los circuitos DTMF desde la línea en los dos casos extremos de Loop Largo a Loop Corto son:

Los Loop largos reducen la corriente y voltaje disponibles para circuitos electrónicos en el teléfono; por eso, los circuitos de marcado de tonos necesitan operar desde un voltaje de alimentación tan bajo como 5 volts.

El voltaje mínimo de operación ($V_{DC(min)}$) para el generador DTMF y el circuito interface será la suma de los voltajes pico de los dos tonos ($V_{LFF} + V_{HFF}$), más el voltaje regulado deseado (V_{reg}), más la caída de voltaje necesaria para obtener el voltaje regulado ($V_{BE} + V_{CE(SAT)}$); por ejemplo:

$$\begin{aligned} V_{DC(min)} &= (V_{LFF} + V_{HFF}) + V_{reg} + (V_{BE} + V_{CE(SAT)}) \\ &= 1.24V + 3V + 0.8V \\ &= 5.04V \end{aligned}$$

Los Loops cortos requieren que el teléfono absorba

corriente de Loop alta o maneje voltaje alto de dc si no esta provisto en la central regular o limitar la corriente y voltaje alimentado a el Loop. en ambos casos la interfase debe mantenerla regulacion del voltaje apropiada.

N I V E L

Los niveles de salida de los tonos DTMF estan referidos a 0dBm (1 miliwatt disipado en impedancia de 600 ohm). Las frecuencias altas tienen un nivel 2dB arriba del nivel de las bajas frecuencias para compensar por perdidas en transmision.

D I S T O R S I O N

La distorsion permisible de los tonos transmitidos se especifica en varias formas:

- 1.- La potencia total de todas las frecuencias no deseadas debera estar al menos 20 dB abajo del nivel del par de frecuencias que ya tenga el menor nivel.
- 2.- El nivel de frecuencias no deseadas producidas por el par debe ser:

- a. No mas de -33 dBm en la Banda desde 300 a 3400 Hz
- b. No mas de -33 dBm en 3400 Hz y cayendo 12 dB por octava a 50 KHz
- c. No mayor de -80 dB arriba de 50 KHz

3.- La Distorsion de dB se define como:

$$\text{DISTORSION} = 20 \log_{10} \frac{\sqrt{U_1^2 + U_2^2 + \dots + U_N^2}}{\sqrt{U_L^2 + U_H^2}}$$

Donde: U_1 hasta U_N son los componentes de frecuencia indeseada.

U_L es el tono de frecuencia baja

U_H es el tono de frecuencia alta

I M P E D A N C I A

Quando el Generador DTMF esta activo, debe presentar la correcta impedancia dinamica al Loop. Una impedancia nominal de 900 ohms es el requerimiento normal.

Quando el Generador DTMF esta inactivo, debe presentar una impedancia dinamica baja cuando esta conectado en serie con el circuito de voz, y una alta impedancia dinamica cuando esta conectado en paralelo.

PERDIDAS POR RETORNO

Las pérdidas por retorno se definen como:

$$RL = 20 \log_{10} \frac{Z_L + Z_G}{Z_L - Z_G}$$

Donde Z_L es la impedancia de línea y Z_G es la impedancia de salida del teléfono que genera la señal. RL debe ser mayor que 14 dB en la Banda de frecuencia de 300 a 3400 Hz. RL debe ser mayor de 10 dB en las Bandas de 50 a 300 Hz y en 2,400 a 20,000 Hz.

VENTAJAS DE DTMF

En suma, el marcado mediante DTMF está reemplazando el marcado por pulsos por las siguientes razones:

1. Disminuye el tiempo de marcado
2. Emplea circuitos electrónicos de estado sólido
3. Puede usarse para señalización (transmisión de datos a baja velocidad).
4. Reduce los requerimientos de equipo en la central local
5. Es más compatible con las centrales controladas electrónicamente (programa almacenado).

DTMF CON CIRCUITOS INTEGRADOS

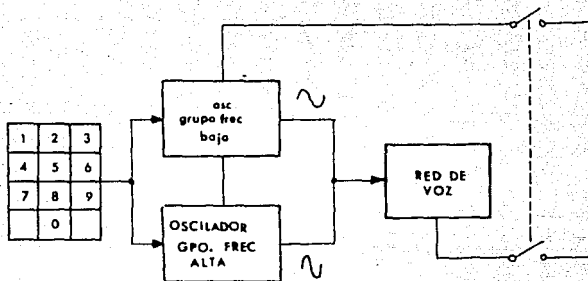
Se han diseñado circuitos integrados para proporcionar la función DTMF. El método convencional de llevarlo a cabo se muestra en la figura 2.2 en forma distinta a la convencional en la que hay dos osciladores senoidales de frecuencia alta y frecuencia baja, el generador DTMF de circuito integrado tiene un contador y un decodificador que cuenta los pulsos de un oscilador controlado por cristal y proporciona los códigos de salida que corresponden a los tonos de baja y alta frecuencia requeridos.

Cada una de las dos salidas del contador se alimenta a su propio convertidor DIGITAL a analógico (D/A). El convertidor D/A, convierte el código digital que sale del contador a un tono de onda senoidal.

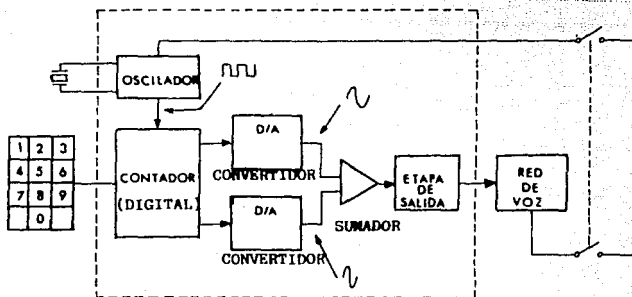
El tono de baja frecuencia y el tono de frecuencia alta son sumados en un amplificador operacional.

EJEMPLO DE UN C.I.

La activación de los tonos de salida en el circuito integrado DTMF empieza presionando una tecla en la botonera. Los contactos de la botonera pueden



CONVENCIONAL



CIRCUITO INTEGRADO

GENERADORES DTMF

FIGURA 2-2

arreglarse como muestra la figura 2.3. La botonera estandar DPST (Double-Pole Single-Throw Switch) donde un juego separado de dos contactos dan el renglon y la columna de la tecla que se presiona. Una alternativa mas economica es SPST (Single-Pole Single-Throw), con solo un juego de contactos para cada interseccion renglon - columna. Algunos C>I>S> estan disenados para aceptar ambos tipos; otros aceptan solo un tipo.

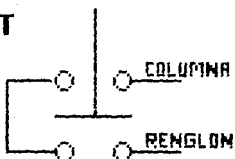
En algunos casos, los contactos cerrados pueden aplicar un voltaje de alimentacion en las lineas de salida mientras la tecla es oprimida. En otros casos, puede ser una tierra.

La botonera puede arreglarse para proporcionar solo un pulso, la figura 2.4 muestra mas en detalle. Un circuito integrado DTMF tipico, el TCMS087 (CMF5087).

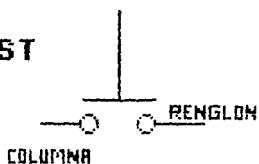
Este circuito pertenece a una familia completa de dispositivos desde el TCMS087 al TCMS092. En el diagrama, puede verse que los circuitos de decodificado (etiquetados = 1) y el juego de compuertas OR y AND en el centro tienen la funcion de controlar la salida de los divisores nombrados CTRDIVK. El control es tal que si solo una tecla se presiona, genera dos salidas; una de los divisores de renglon y una de los divisores de columna.

TECLADOS DTMF

DPST



SPST



TIPOS DE CONTACTOS DE TECLADO Y SALIDA

FIGURA 2-3

24

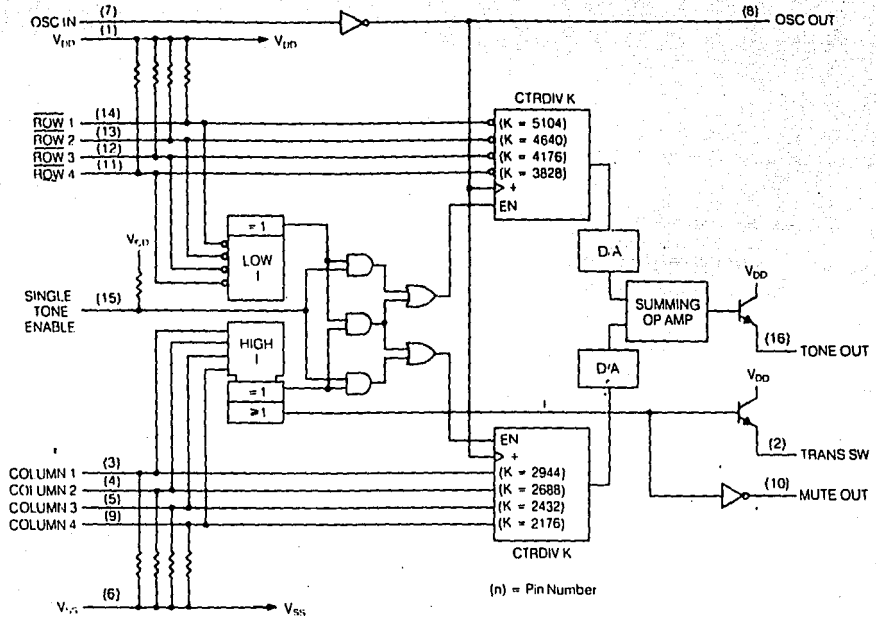


DIAGRAMA A BLOQUES DEL TCM5087

FIGURA 2-4

Estas dos salidas digitales son convertidas a ondas analogicas y sumadas en el amplificador-sumador. Note sin embargo, que es posible presionar dos teclas de la botonera simultaneamente. Si las dos teclason de la misma columna o del mismo renglon, y el Habilitador de tono simple (pin 15 de este circuito) es 1, solo se genera un tono correspondiente al renglon o columna donde las dos teclas son presionadas.

Esta característica se implementa para permitir probar la Botonera. La operacion de dos teclas en posiciones diagonales causa que no haya salida (ambos lados del decodificador indican "uno igual a uno").

Otra funcion interesantese muestra por el elemento logico nombrado X1 abajo del decodificador de alta frecuencia. Esta compuerta proporciona una salida a niveles de voltaje logico estandard (nombrados MUTE OUT, Pin 10), y una salida a traves de transmisor (Pin2), para silenciamiento del circuito de voz. MUTE OUT se usa para silenciar el receptor durante el marcado para que no se "escuche" el nivel completo de los tonos.

La sintesis de frecuencia en el circuito se hace usando una simple entrada de frecuencia de 3.579545 MHz, derivada desde un oscilador de cristal externo

usando un cristal de TV color, conectado a los pins 7 y 8. Esta entrada de frecuencia se divide en los bloques nombrados CTRDW: por ocho diferentes constantes para producir ocho posibles frecuencias de salida, dos de los cuales son seleccionados cada vez que se presiona una tecla de la Botonera asociada.

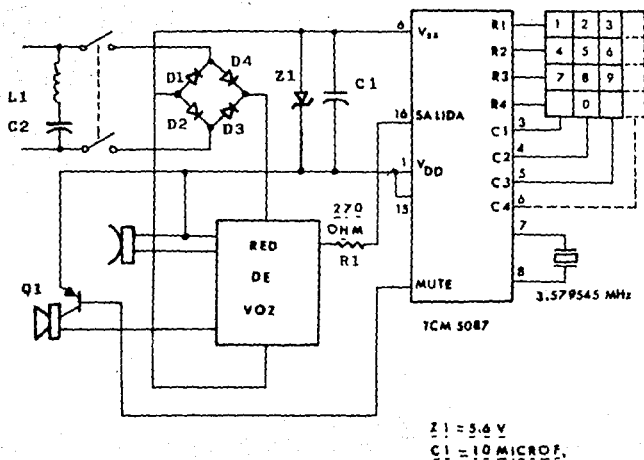
FORMA DE ONDA DE SALIDA

La salida no consiste de ondas puras sinusoidales, sino de escalon debido a la conversio Digital/Analogico.

La aproximacion por escalon tipica de salida sinusoidal se muestra en el anexo 1. Las formas de onda sinusoidal se generan con una distorsion tipica de menos del 7%. El contenido de distorsion armonica y de intermodulacion tipico resultante de la suma de los dos tonos esta 30 dB abajo referido al tono fundamental de la columna mas fuerte.

EJEMPLO DE APLICACION

En la figura 2.5 se muestra el TCM5087 aplicado a un aparato telefonico, donde el tono de salida se alimenta a la linea a traves de la red de voz. Un puente de



APLICACION DEL C.I. TCM5087

FIGURA 2-5

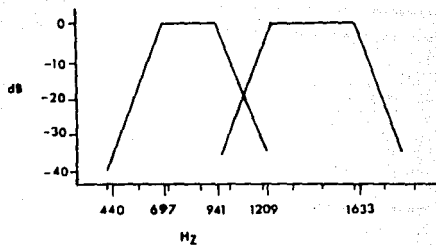
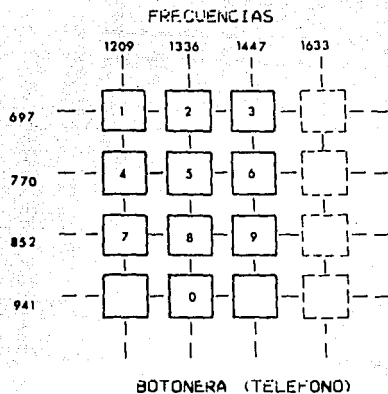
diodos alimenta al circuito integrado y el zener Z1 lo protege. El capacitor C1 mantiene y filtra el voltaje a través del C>I para eliminar ruido y variaciones de la alimentación. Una señal de silenciamiento (mute) controla Q1y silencia el transmisor de la red de voz. La salida XMIT en el pin2 y la entrada ST1 en el pin 15 ya fueron descritas.

C. RECEPTORES DE DTMF

Cada receptor de tonos DTMF se comparte entre aproximadamente treinta líneas de abonado en una central.

La figura 2.6 muestra la respuesta en frecuencia de un filtro DTMF ideal. Tal dispositivo es llamado un filtro "bandsplit" porque su salida son dos bandas separadas de frecuencias para pasar el grupo alto y bajo de tonos DTMF mientras rechaza las otras frecuencias. La salida del filtro debe ir a un detector para cumplir otros requerimientos.

En el detector, cada grupo de señales se procesa separadamente. Se usan técnicas digitales para determinar la frecuencia contando el número de púlsos del reloj maestro presentes en cada periodo de la



CARACTERÍSTICA IDEAL DEL FILTRO DTMF

FIGURA 2-6

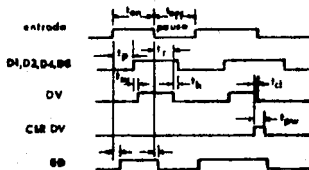
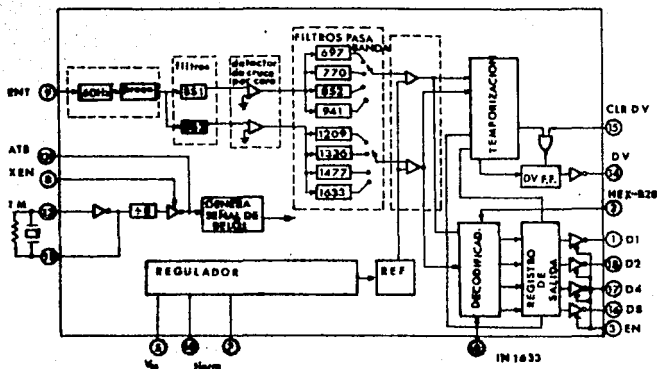


DIAGRAMA DEL RECEPTOR DTMP

FIGURA 2-7

frecuencia desconocida. La frecuencia detectada del grupo alto y la del grupo bajo se prueban logicamente para determinar si son frecuencias DTMF validas. Si lo son, se combinan para producir una salida codificada que representa el numero marcado.

Una de las tareas mas dificiles del detector es determinar si un tono dentro de la banda de DTMF es realmente un tono DTMF o es un sonido producido por la voz que solamente se asemeja; si el criterio de aceptar es muy relajado, pares de tono son rapidamente reconocidos como validos, pero tambien lo seran otros sonidos que se le parezcan. Si el criterio es muy estricto, el tiempo de deteccion se amplia y el receptor no cumple los requerimientos de tiempo, el resultado es un compromiso entre los efectos del ruido y el tiempo de deteccion (tipico 10 miliseq.).

ESTRUCTURA DE UNA COMPUTADORA DIGITAL

Esta formada por cuatro partes principales: Unidad Central de Procesos (CPU), Memoria, Unidades de Entrada y Unidades de Salida.

LA UNIDAD CENTRAL DE PROCESOS (CPU)

Es la parte principal de la computadora y se puede dividir por sus funciones en Registros, Registro de Banderas, la Unidad de Control en la Unidad Aritmética y Lógica (ALU, Arithmetic Logic Unit).

Registros.— La CPU cuenta con dos tipos de registros: registros de propósito general y registros de propósito especial. Los registros de propósito general se utilizan para el manejo de los datos y los registros de propósito especial tienen funciones ya definidas, por ejemplo, el Contador del Programa y el Registro de Instrucción.

Registro de Banderas.— La CPU cuenta con un registro en el cual se registra información del resultado de una operación. Este registro se conoce como Registro de Banderas. Entre las más comunes se encuentran las banderas de Cero, Paridad, Signo, Acarreo, etc.

E U S . (TRES ESTADOS).

En el diseno de computadoras se utiliza el concepto de "bus" que consiste de un grupo de lineas por las cuales se transfiere informacion de cualquier dispositivo a otro.

El proposito fundamental de utilizar el concepto de "bus" es el de reducir el numero de lineas de conexion requeridas para la transferencia de informacion. En el ambiente de las microcomputadoras se utilizan tres buses para los diferentes tipos de informacion:

- a) Bus de Datos
- b) Bus de Direccion
- c) Bus de Control

COMPUERTAS DE TRES ESTADOS

Este funcionamiento permite que se puedan conectar varios dispositivos a una sola linea y que la transferencia de la informacion se realiza entre el dispositivo que tiene habilitada su linea de seleccion o de habilitar y la microcomputadora.

Entre las caracteristicas que se pueden mencionar de

los circuitos con salidas de tres estados son las siguientes:

- 1) Compatibles con la serie 54/74 TTL
- 2) Se pueden conectar hasta 128 circuitos buffer a un bus.
- 3) Un retardo de propagación de 12 nanosegundos
- 4) Control de la impedancia de cada buffer

MARGEN DE RUIDO.

La salida de la compuerta durante el nivel lógico 0 (vol), no excede los 0.4 volts y las entradas de la compuerta, en el nivel lógico 0, pueden tolerar hasta valores de 0.8 volts antes de cambiar se salida a nivel lógico 1. Esto significa que existe una diferencia de 0.4 volts entre los dos valores garantizados. Entonces, la línea que transmite la señal entre las dos compuertas puede recibir un pulso de ruido de hasta 400 mV en amplitud y la compuerta permanecerá en el nivel 0 sin disturbio. A esta diferencia de 400 mV se conoce como "margen de ruido".

El microprocesador Z-80 requiere de una señal de

reloj generada externamente y que debe variar de 0 a 5 volts.

Se especifican las características electricas del microprocesador 2-80 entre las que se encuentra que la corriente de salida a nivel bajo (IOL) puede drenar, cada una de sus salidas, un valor de 1.8 mA. La tabla ilustra que la corriente de entrada maxima (IIL) de una compuerta lenta y baja disipacion es de 0.18 mA. Entonces, el microprocesador 2-80 puede manejar 10 compuertas lentas de baja disipacion. La corriente requerida por las entradas TTL en el nivel logico1 es pequeno.

Frecuentemente la informacion se debe conservar durante cierto tiempo antes de que la microcomputadora la leo o que un dispositivo periferico la reciba. En situaciones como esta se deben utilizar ciertos dispositivos conocidos como "latch". El latch es un circuito (como los flip-flops) que se utiliza para almacenar estados logicos.

BUFFER Y DRIVER

Se le da e nombre de "buffer" a un circuito amplificador que se puede usar para incrementar la

capacidad de manejar las corrientes en una línea de un microprocesador. Los circuitos buffer con salidas de "tres estados" se usan para proporcionar un aislamiento eléctrico entre los componentes de una microcomputadora.

Un uso importante de los buffers es en los puertos de entrada de datos. En esta aplicación los buffers sirven para aislar las salidas de datos de un dispositivo del bus de datos del microprocesador hasta que el microprocesador requiera los datos. Se usan compuertas lógicas para detectar cuando un puerto de entrada tiene un dato para el microprocesador, a continuación el microprocesador debe enviar un pulso (strabe) al puerto de entrada para leer el dato.

Con el uso de los buffers de tres estados se incrementa el número de puertos de entrada que se pueden conectar a los buses de la microcomputadora, ya que mientras el microprocesador no le indique al puerto de entrada, este no se "conecta" a los buses. Es decir, el puerto se encuentra en el tercer estado y eléctricamente aislado del sistema, por lo que no está cargando a los buses.

El bus de datos se utiliza para enviar y

recibir datos, bidireccional, por lo tanto se deben utilizar circuitos buffer en ambas direcciones conocidos como "drivers". Un driver bidireccional se forma con dos buffers de tres estados conectados con un inversor entre las líneas de habilitar para asegurar que solamente uno de los buffers se habilita a la vez.

BUS DE DIRECCIONES

Bus formado por 16 líneas (A0 - A15), con salidas de 3er. estado, con las cuales el microprocesador puede direccionar hasta 64K bytes de memoria RAM o ROM. Las líneas A0 - A7 se emplean para direccionar los puertos de entrada/salida, periodo en el que A8 - A15 se mantienen en 3er. estado. Mediante las líneas A0 - A7 el microprocesador puede direccionar hasta 256 puertos de entrada y 256 puertos de salida. Durante la acción de refresco, los 7 bits menos significativos indican la dirección de refresco.

BUS DE CONTROL

Este bus está formado por 13 líneas que han sido englobadas en 3 grupos:

a) Control del Sistema.- Este grupo controla la memoria y puertos mediante 6 líneas.

M1.- Salida activa en fase. M1 indica que el ciclo de máquina que está ejecutando el microprocesador, es utilizado para obtener la instrucción a ejecutar (Ciclo FETCN).

NOTA: Para las instrucciones con código de 2 o más

bytes. Se genera la señal MI por cada byte leído, así como cuando el programador responde una instrucción.

MREQ.- Salida de 3er. estado activa en fase. Mediante esta señal el microprocesador indica que está realizando un acceso a memoria, pudiendo ser este para lectura, escritura o refresco.

NOTA: Esta señal también es activada durante un ciclo FETCH.

FORD.- Salida de estado 3 activada en fase. Esta señal indica la ejecución de un acceso a puerto, pudiendo ser dicho acceso tanto para escritura como lectura. Cuando esta señal acontece junto con MI, indica que el microprocesador está obteniendo el vector del puerto que interrumpe.

RD.- Salida de 3er. estado activa en fase. Con esta señal se indica que el acceso a

WR.- Salida de 3er. estado activa en fase. Esta señal indica que el acceso a memoria es para escritura.

REFH.- Salida de 3er. estado activa en fase. Esta señal indica que el acceso a memoria se hace para el refresco de esta.

NOTA: Esta señal unicamente se emplea en sistemas con memoria dinamica que no tienen un circuito asociado para la sincronizacion del refresco.

b) Control del CPU.- Mediante este grupo de 3 senales es posible controlar el comportamiento del microprocesador.

HALT.- Salida activa en fase, que nos indica que el procesador esta ejecutando una instruccion de alto (HALT), permaneciendo en este estado hasta que una senal de interrupcion solicite un servicio.

WAI.- Entrada activa en fase. Sirve para indicar que el dispositivo con el que se esta estableciendo la comunicacion, no esta listo para realizar la transferencia y por lo tanto el microprocesador esperara hasta que esta senal se desactive para establecer la transferencia del dato.

INT.- Entrada activa en fase. Mediante esta entrada se solicita una interrupcion. Cuando ocurre esta senal, el microprocesador termina de ejecutar la instruccion que se esta realizando y despues de ella ejecuta un ciclo de maquina de reconocimiento de la interrupcion, solicitando el procesador, al dispositivo que

interrupcio, una etiqueta con la que tendra la direccion de la rutina de servicio correspondiente.

NMI.- Entrada activa en fase. Tambien esta senal le indica al microprocesador que se ha de realizar una interrupcion, diferenciandose de INT, en que una vez terminada de ejecutar la instruccion, el microprocesador no contesta con un ciclo de reconocimiento de instruccion, sino que inmediatamente genera la direccion 0066H como el inicio de la rutina.

RESET.- Entrada activa en fase. Mediante esta entrada se le indica al microprocesador que debe establecer las condiciones iniciales que son:

Contador de Programa en	0000 N
Registro I	00 N
Registro R	00 N
Nodo de Interrupcion	0 N
Interrupciones no deseables	

CONJUNTO DE CARACTERES DE LA 2X81

Hay dos funciones disponibles en la 2X81 para convertir caracteres en su código de números y convertir códigos de números en caracteres.

Cuando se aplica a un número en el rango 0 ≤ X ≤ 255, regresa un simple carácter cuyo código es ese número. Por ejemplo:

```
10 LET A$ = CHR$(38)
```

asignamos "A" a la variable A\$ porque el código de "A" es 38.

Cuando se aplica a una frase regresa el código del primer carácter de la frase.

```
10 LET A = CODE "A"
```

```
15 LET A = CODE "ABCDEF"
```

Ambas declaraciones asignan el valor de 38 a la variable A.

Conjunto de Caracteres de la 2x81

1.	25. ;	49. L	113. CURSOR DOWN
2.	26. ,	50. M	114. CURSOR LEFT
3.	27. .	51. N	115. CURSOR RIGHT
4.	28. 0	52. O	116. GRAPHICS
5.	29. 1	53. P	117. EDIT
6.	30. 2	54. Q	118. NEWLINE
7.	31. 3	55. R	119. RUBOUT
8.	32. 4	56. S	120. MODE
9.	33. 5	57. T	121. FUNCTION
10.	34. 6	58. U	122.
11.	35. 7	59. V	123.
12.	36. 8	60. W	124.
13.	37. 9	61. X	125.
14.	38. A	62. Y	126.
15.	39. B	63. Z	127.
16.	40. C	64. RND	128.
17.	41. D	65. INKEY#	129.
18.	42. E	66. F1	130.
19.	43. F	67.	131.
20.	44. G		132.
21.	45. H		133.
22.	46. I		134.
23.	47. J	111.	135.
24.	48. K	112. CURSOR UP	136.

137.	163. INVERSE 7	189. INVERSE X
138.	164. INVERSE 8	190. INVERSE Y
139. INVERSE "	165. INVERSE 9	191. INVERSE Z
140. INVERSE &	166. INVERSE A	192. ""
141. INVERSE \$	167. INVERSE B	193. AT
142. INVERSE .	168. INVERSE C	194. TAB
143. INVERSE ?	169. INVERSE D	195. NOT USED
144. INVERSE (170. INVERSE E	196. CODE
145. INVERSE)	171. INVERSE F	197. VAL
146. INVERSE >	172. INVERSE G	198. LEN
147. INVERSE <	173. INVERSE H	199. SIN
148. INVERSE =	174. INVERSE I	200. COS
149. INVERSE +	175. INVERSE J	201. TAN
150. INVERSE -	176. INVERSE K	202. ASN
151. INVERSE *	177. INVERSE L	203. ACS
152. INVERSE /	178. INVERSE M	204. ATN
153. INVERSE ;	179. INVERSE N	205. LN
154. INVERSE ,	180. INVERSE O	206. EXP
155. INVERSE .	181. INVERSE P	207. INT
156. INVERSE 0	182. INVERSE Q	208. SQR
157. INVERSE 1	183. INVERSE R	209. SGN
158. INVERSE 2	184. INVERSE S	210. ABS
159. INVERSE 3	185. INVERSE T	211. PEEK
160. INVERSE 4	186. INVERSE U	212. USR
161. INVERSE 5	187. INVERSE V	213. STR#
162. INVERSE 6	188. INVERSE W	214. CHR#

215. NOT	241. LEI
216. **	242. PAUSE
217. OR	243. NEXT
218. AND	244. POKE
219. <=	245. PRINT
220. >=	246. PLOT
221. <.>	247. RUN
222. THEN	248. SAVE
223. TO	249. RAND
224. STEP	250. IF
226. LLIST	252. UNPLOT
227. STOP	253. CLEAR
228. SLOW	254. RETURN
229. FAST	255. COPY
230. NEW	
231. SCROLL	
232. CONT	
233. DIM	
234. REM	
235. FOR	
236. GOTO	
237. GOSUB	
238. INPUT	
239. LOAD	
240. LIST	

CONVERTIDOR DIGITAL-ANALOGICO DE 5 BITS

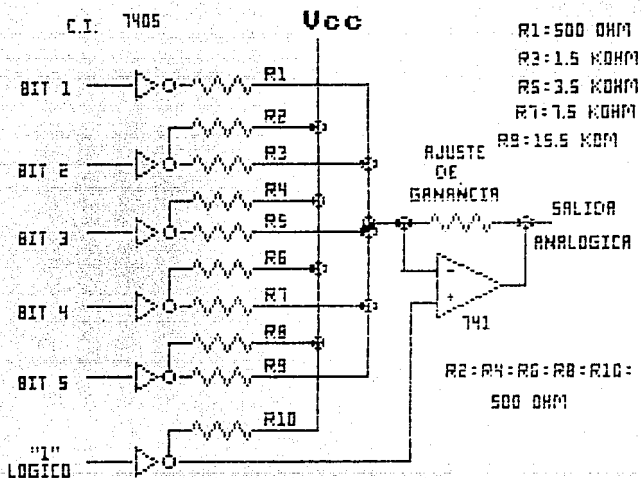


FIGURA 3-1

CONVERSION DIGITAL/ANALOGICA

Un convertidor de 5 - bit se puede formar con un circuito integrado 7405 que contiene seis inversores de colector abierto que realizan la función de conmutar la entrada. Cuando las salidas de las compuertas de colector abierto son bajas, cada compuerta tiene una impedancia de carga de 500 ohms, y los voltajes de saturación colector - emisor de los transistores de salida de las compuertas están a milivolts uno del otro.

Cuando las salidas son altas, cada transistor de salida está en corte y la red de resistencias está referida al voltaje de alimentación V_{cc} como se ve en la figura

Un ejemplo de un sintetizador de onda senoidal se muestra en la figura donde se emplean un contador de 4 bits, las salidas B, C y D entran a dos compuertas OR exclusivo donde los dos bits más significativos determinan el cuadrante de la señal, y los otros dos bits direccionan una memoria ROM que da los grados de 0 a 90. Note que la dirección de la ROM es invertida por el bit B cuando la senoidal está en el cuadrante II o IV y que la salida de la ROM es complementada a 2 por

el bit A cuando la senalesta en el cuadrante III y IV tambien, el bit A se invierte y pasa como signo del convertidor digital analogico.

Una opcion mas simple es emplear un multiplexor analogico de ocho a uno 4051, como se observa en la figura (PAG.112). Aqui la frecuencia f_i de entrada al contador debe tener un valor de 16 veces la frecuencia de la senal senoidal deseada.

La salida D del contador va a la entrada C de seleccion del multiplexor y las salidas C, E, y H del contador pasan a dos compuertas OR exclusivo para establecer el cuadrante de la senoidal, y las dos salidas de estas van a las entradas B y M de seleccion del multiplexor. En este se tienen cuatro posibles escalones positivos y cuatro negativos de los cuales solo uno a la vez pasa a la entrada inversora del amplificador operacional. En la figura se pueden ver las salidas sin capacitor y con capacitor de filtro en el operacional, el cual mejora la aproximacion a forma de ondas senoidal.

PROGRAMACION DEL P10

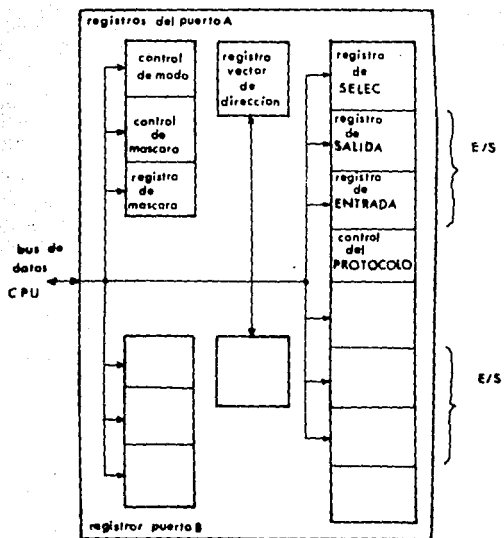
La entrada B/A SEL determina, cuando esta seleccionado el puerto A el puerto B. Esta entrada esta conectada con la linea de direccion A0.

La entrada CE es para reconocer el chip. Con "0" activa el CHIP, y con "1" inactiva el CHIP. Esta señal se crea de las direcciones A2 a A7 por medio de un C.I. NAND 74LS00 y del decodificador binario de tres bits 74LS138. La señal CE se hace "0" cuando existe la siguiente combinacion de bits (ver la tabla de verdad del 74LS138):

A7	A6	A5	A4	A3	A2	A1	A0
1	1	1	1	1	0	X	X

En el lado derecho del P10 en el diagrama del circuito estan las lineas que se conectan a dispositivos externos. Son las lineas PA0 a PA7 con las lineas de "Handshake" ARDY y ASTB, y las lineas PB0 a PB7 con las lineas BSTB y BRDY.

La linea ARDY es una salida activa en "1". Si el puerto A esta definido como una salida, entonces ARDY se hace "1" cuando han sido enviados datos al puerto. Esto avisa al dispositivo externo, que ya hay datos disponibles.



REGISTROS DEL 280 PIO

FIGURA 4-1

Si el puerto A esta definido como una entrada, entonces ARDY se hace 1, si el puerto esta vacio, el procesador ha obtenido los datos, y entonces el dispositivo externo puede enviar nuevos datos al puerto. Si el puerto A esta definido como bidireccional, entonces ARDY = 1 significa, que datos pueden ser enviados a el dispositivo externo. Los datos son aplicados a las salidas, solo si fue recibido una senal ASTB.

En el modo tres la linea ARDY se pone fuera internamente.

La linea ASTB es una entrada activa en "0".

Si el puerto A esta definido como salida, entonces un pulso positivo en ASTB significa que el dispositivo externo ha recibido datos.

Si el puerto A esta definido como una entrada, entonces un pulso negativo en ASTB causa que los datos sean recibidos por el registro de entrada. En el modobidireccional ASTB es para datos a ser cambiados a las lineas de salida.

Las lineas BSTB y BRDY se comportan en la misma

manera, excepto en el siguiente modo:

El puerto A es bidireccional. En este caso un pulso negativo transfiere datos al registro de entrada del puerto A, mientras $\overline{RDY} = 1$ significa que el puerto B esta vacio y puede recibir datos.

La programación de los puertos puede efectuarse por una palabra de control. En la figura se indica el formato de esa palabra de control. Los bits D6 y D7 definen el modo, D5 y D4 son insignificantes, y los bits D0 a D3 indican que es una palabra de control para el P10.

Los modos estan definidos como sigue:

M1	M0	MOD0	
0	0	0	El puerto es salida
0	1	1	El puerto es entrada
1	0	2	Bidireccional
1	1	3	Bit de control

Las palabras de control son:

\$DF	Salida
\$4F	Entrada
\$8F	Bidireccional
\$CF	Bit - Control

La palabra de control se envia al FIU por medio de las líneas de datos. La entrada C/D SEL tiene que ser 1 en el mismo tiempo. Esta entrada esta conectada a HI. El puerto para el cual la palabra de control es valida se selecciona por B/A SEL, la cual esta conectada a HI. Si B/A SEL = 0, entonces selecciona el puerto A con \$FB siendo la direccion decodificada para CE tenemos las siguientes direcciones para el FIU:

11111000	= F8H	PUERTO A, DATOS
11111001	= F9H	PUERTO B, DATOS
11111010	= F9H	PUERTO A, PALABRA DE CONTROL
11111011	= FBH	PUERTO B, PALABRA DE CONTROL

En el siguiente programa en lenguaje de maquina define el puerto A como salida:

```
LDA,OF 3E0F OF A, SALIDA
OUTFA,A D3FA (A) PUERTO A, PALABRA DE CONTROL
LDA,AA 3EAA AA A ; PATRON DE BIT
OUTFB,A C9 9A0 PUERTO A, DATOS
RET C9 REGRESO A BASIC
```

La rutina ocupa 9 bytes, para los cuales creamos espacio con el enunciado RE: AAAAAAAAAH en la primera línea. El programa empieza en la localidad 10509. Los

primeros dos bytes son para el número de línea, los siguientes dos bytes son para la longitud de línea y un byte es para codificar el comando POK. Por eso la primera A está en la localidad 16514. Aquí ponemos nuestro programa en lenguaje de máquina.

Si queremos escribir en el lenguaje de máquina usando comandos POK, entonces debemos convertir los números hexadecimales en números decimales:

3EH = 62

0FH = 15

D3H = 211

FAH = 250

AAH = 170

F8H = 248

C9H = 201

PROGRAMA EN BASIC PARA SALIDA EN EL PUERTO A

001 REM AAAAAAAAA

010 LET X = 16514

020 POK X,62

030 POK X+1,15

050 POK X+3,250

060 POK X+4,62

080 POK X+5,211

```

090 F0RE X+7.248
110 PRINT "N=";
120 INPUT N
125 IF N=0 THEN STOP
130 PRINT N
140 POKE X+5,N
150 LET A=USRX
160 GOTO 110

```

El programa almacena los comandos empezando en 16514, excepto el byte 6, el cual no se almacena hasta que se da un valor a N. El lenguaje de maquina es llamado en la linea 150.

Con un voltimetro se pueden medir los voltajes en cada salida. Si se da el valor 170 para N se mediran los voltajes siguientes:

BINARIO	VOLTAJE	FIN	SALIDA
1	3.6V	7	PA7
0	0V	8	PA6
1	3.6V	9	PA5
0	0V	10	PA4
1	3.6V	12	PA3
0	0V	13	PA2
1	3.6V	14	PA1
0	0V	15	PA0

VOLTAJES EN EL PUERTO A, DESPUES DE MANDAR N=170
(1010 = A)

Si queremos usar otro programa para salida, donde podamos meter los números en forma hexadecimal, en el programa en lenguaje de maquina de nuevo es creado usando el comando REM en la primera linea. El programa en lenguaje de maquina se almacena en la cadena de caracteres M\$ como una secuencia de números hexadecimales. Siempre dos caracteres de la cadena se convierten en un número decimal y se almacenan en memoria con un comando POKE.

```

001 REM AAAAAAAAAA
010 LET X=16514
020 LET M$="3E0FD3FA 3E00D3FBC9"
030 FOR I=1 TO LEN M$-1 STEP 2
040 POKE X+INT ((I-1)/2), (CODE M$(I)-28)
      *16+CODE M$(I+1)-28
050 NEXT I
060 PRINT "N="
070 INPUT N$
075 IF N$="" THEN STOP
080 PRINT N$
100 POKE X+5, (CODE N$(1)-28)*16+CODE N$(2)-28
110 LET A = USR X
120 GOTO 60

```


T A B L A D E I N S T R U C C I O N E S D E L Z 8 0 4

(P O R M E N O S S I G N I F I C A T I V O)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E
0	NOF	LC BC,nn	LG (BC),A	INC BC	INC B	DEC B	LD B,n	NLCA	EA AF,AF	ADD HL,BC	LD A,(BC)	DEC BC	INC C	DEC C	LD C,n
1	JNZ disp	LD DE,nn	LD (DE),A	INC E	INC D	DEC D	LD D,n	PLA	JR disp	ADD HL,DE	LD A,(DE)	DEC DE	INC E	DEC E	LD E,n
2	JNZ disp	LD HL,nn	LD (nn),HL	INC HL	INC H	DEC H	LD H,n	DW	JRZ	ADD disp	LD HL,nn	DEC HL	INC L	DEC L	LD L,n
3	JR MC disp	LD SF,nn	LD (nn),A	INC SF	INC (HL)	DEC (HL)	LD (HL),n	SCF	JRC disp	ADD HL,SP	LD A,(nn)	DEC SP	INC A	DEC A	LD A,n
4	LD B,B	LD B,C	LD B,D	LD B,E	LD B,H	LD B,L	LD B,(HL)	LD B,A	LD C,B	LD C,C	LD C,D	LD C,E	LD C,H	LD C,L	LD C,(HL)
5	LD D,B	LD D,C	LD D,D	LD D,E	LD D,H	LD D,L	LD D,(HL)	LD D,A	LD E,B	LD E,C	LD E,D	LD E,E	LD E,H	LD E,L	LD E,(HL)
6	LD H,B	LD H,C	LD H,D	LD H,E	LD H,H	LD H,L	LD H,(HL)	LD H,A	LD L,B	LD L,C	LD L,D	LD L,E	LD L,H	LD L,L	LD L,(HL)
7	LD HL,B	LD HL,C	LD HL,D	LD HL,E	LD HL,H	LD HL,L	LD HL,(HL)	LD HL,A	LD A,B	LD A,C	LD A,D	LD A,E	LD A,H	LD A,L	LD A,(HL)
8	ADD A,B	ADD A,C	ADD A,D	ADD A,E	ADD A,H	ADD A,L	ADD A,(HL)	ADD A,A	ADD A,B	ADD A,C	ADD A,D	ADD A,E	ADD A,H	ADD A,L	ADD A,(HL)
9	SUB B	SUB C	SUB D	SUB E	SUB H	SUB L	SUB (HL)	SUB A	SUB A,B	SUB A,C	SUB A,D	SUB A,E	SUB A,H	SUB A,L	SUB A,(HL)
A	AND B	AND C	AND D	AND E	AND H	AND L	AND (HL)	AND A	XOR B	XOR C	XOR D	XOR E	XOR H	XOR L	XOR (HL)
B	OR B	OR C	OR D	OR E	OR H	OR L	OR (HL)	OR A	CP B	CP C	CP D	CP E	CP H	CP L	CP (HL)
C	SET NZ BC	POP nn	JP NZ nn	JP nn	CALL NZ nn	PUSH BC	ADD A,n	RST 0	RET Z	RET	JP Z nn		CALL Z nn	CALL nn	ADC A,n
D	RET NC DE	POP DE	JP NC nn	OUT part A	CALL NC nn	PUSH DE	SUB n	RST 16	RET C	ELL	JPC nn	IN A,part	CALL C nn		SBC A,n
E	RET FO HL	POP HL	JP FO nn	EX (SF),HL	CALL FO nn	PUSH HL	AND n	RST 22	RET FE	JP (HL)	JP PE nn	EX DE,HL	CALL FE,nn		XOR n

FIGURA 4-2

general (A, B, C, D, E, H y L) con un dato de 8 bits que es parte de la propia instrucción. Estas instrucciones contienen no solo el código de operación sino también el dato. La instrucción es "Cargar Inmediato a un Registro", la cual consiste en dos bytes y debe ocupar dos localidades consecutivas en la memoria. El primer byte es el código de operación y el segundo es el dato que se va a cargar en el registro especificado. El mnemónico de la instrucción es:

LD r, n

en donde r es cualquiera de los siete registros de propósito general de la Z-80, y n es un dato de 8 bits. El código de esta instrucción es:

0 0 1 1 1 1 1 0 C O D I G O D E O P E R A C I O N
D A T O D a t o

TABLA C O D I G O B I N A R I O D E L O S S I E T E R E G I S T R O S

REGISTRO	CODIGO	REGISTRO	CODIGO
A	111	E	011
B	000	H	100
C	001	L	101

Ejemplo.- Cargar el registro E con el dato BEH. La instrucción puede ser:

```
LD    E,BEH    ; (E) = BEH
```

y el código sería:

```
0 0 0 1 1 1 1 0    = 1 E H
1 0 0 0 1 1 1 0    = B E H
```

es decir, código = 1EBEH (16 bits)

INCREMENTAR Y DECREMENTAR (8 BITS)

La Z-80 tiene instrucciones que permiten incrementar o decrementar en uno el contenido de los registros de propósito general de la CPU o las localidades de memoria direccionada por los registros IX, IY y HL. Las instrucciones son:

```
INC    s        y        DEC    s
```

en donde "s" puede ser: r, (HL) , (IX + d) y (IY + d)

Los mnemonicos de las instrucciones son:

INC	r	DEC	r
INC	(HL)	DEC	(HL)
INC	(IX + d)	DEC	(IX + d)
INC	(IY + d)	DEC	(IY + d)

Las instrucciones INC r y DEC r incrementan y decrementan en uno respectivamente el contenido del registro "r", las instrucciones INC (HL) y DEC (HL) incrementan y decrementan en uno respectivamente el contenido de la localidad direccionada por el registro para HL. El código de operación es:

0 0 s s s 1 0 0

en donde sss identifica al registro de acuerdo a la tabla . Cuando el valor en esos tres bits es 110 (6) se hace referencia a la dirección apuntada por el contenido del registro por HL. Estas instrucciones afectan a todas las banderas menos a la de acarreo. Cuando un registro después de incrementarse alcanza el valor cero, la bandera de cero toma nivel 1. Estas instrucciones son útiles cuando se requiere hacer un determinado procesamiento varias veces (lazos) por lo que se necesita de un registro que realice la función de "contador".

Las otras instrucciones para incrementar o decrementar localidades de memoria son:

INC (IX + d) DEC (IX + d)
INC (IY + d) DEC (IY + d)

en donde la dirección de memoria se obtiene al sumar el

contenido del registro IX o IY con el desplazamiento "d".

Ejemplo.- Considerando que el contenido de la localidad de memoria 2035H es 25H, despues de las siguientes instrucciones su contenido sera 27H.

```
LD  H,2035H           ; (HL) = 2035H
INC  (HL)              ; (2011 H) = 25H + 1 = 26H
INC  (HL)              ; (2011 H) = 26H + 1 = 27H
```

(S) = 0 (P) = 1 (Z) = 0 (H) = 0 (CY) - no se altera

INSTRUCCIONES LOGICAS.

Las instrucciones logicas permiten a la programacion imitar el comportamiento de los dispositivos logicos. Las operaciones logicas son similares a las aritmeticas en que se permiten los mismos modos de direccionamiento para obtener el segundo operando (registro, inmediato, registro indirecto e indexado) y en que el Acumulador contiene el primer operando y almacena el resultado al final de la instruccion. Las 3 operaciones logicas que la Z-80 puede realizar son: AND, OR y XOR - Exclusivo. Las reglas de estas operaciones se ilustran en la tabla. Los mneonicos de estas instrucciones son:

AND s OR s XOR s

en donde "s" puede ser: r, n, (HL), (IX + d) y (IY + d)

FUNCION AND.

Las instrucciones AND son utiles cuando se quiere aislar o limpiar algunos bits en particular de los ocho bits de un byte. La Z-80 tiene cuatro modos de direccionamiento para obtener el segundo operando de la funcion logica AND: Registro, Inmediato, Registro Indirecto e Indezado. El primer operando se encuentra en el acumulador y tambien en el acumulador se almacena el resultado. Las instrucciones son:

AND r ; (A) = (A) (r) - Registro

AND n ; (A) = (A) dato - Inmediato

Ejemplo: Aislar los bits A2, A1 y A0 del contenido del Acumulador, considerando que (A) = 26H

LD B,07H ; (B) = 07H, Bits 2, 1 y 0 igual a 1




AND B ; (a) = 26H 07H = 06H

(CY) = 0 (Z) = 0 (P) = 1 (S) = 0 (H) = 0

tambien se puede obtener el mismo resultado con:

AND 07H ; (A) = 26H 07H = 06H

TABLA DE OPERACIONES LÓGICAS

INSTRUCCION	OPERACION LÓGICA	SÍMBOLO
AND	$0 \cdot 0 = 0$ $0 \cdot 1 = 0$ $1 \cdot 0 = 0$ $1 \cdot 1 = 1$	ES EL SÍMBOLO DE AND 
OR	$0 + 0 = 0$ $0 + 1 = 1$ $1 + 0 = 1$ $1 + 1 = 1$	ES EL SÍMBOLO DE OR 
$0 + 1 = 1$ $1 + 0 = 1$	$0 + 0 = 0$ $1 + 1 = 1$ ES EL SÍMBOLO DE "OR - EXCLUSIVO"	

FUNCION OR

Las instrucciones OR (inclusivo) son utiles cuando se quiere combinar diferentes datos o poner incondicionalmente ciertos bits de un byte. Para la funcion OR la Z-80 tiene los mismos cuatro modos de direccionamiento para obtener el segundo operando que la funcion logica AND: Registro, Inmediato, Registro Indirecto e Indexado. El primer operando se encuentra en el Acumulador y tambien en el Acumulador se almacena el resultado. Las instrucciones con los diferentes direccionamientos son:

OR r : (A) = (A) (r) - Registro
 OR n : (A) = (A) dato - Inmediato

Ejemplo.- Efectuar la funcion OR inclusivo entre los contenidos del Acumulador y el registro B, considerando que (A) = 29H y (B) = 45H. El resultado depositarlo en la localidad con direccion 1000H.

OR B	:	(A) = 29 H	45H = 6DH
LD (1000H), A	:	(1000H) = 6 DH	
		29H = 0010	1001
OR		45H = 0100	0101
		<hr/>	
		3DH	0011 1101
		6DH	0110 1101

Ejemplo.- Efectuar la función OR inclusivo entre los contenidos del Acumulador y la localidad 8060H, considerar que (A) = 44H y (8060H) = 78H. El resultado almacenarlo en la localidad 8061H.

```
LD HL,8060H      ; (HL) = 8060H
OR (HL)          ; (A) = 44H 78H = 7CH
INC L           ; (L) = 61H
LD (HL),A       ; (8061H) = 7CH
```

```
OR      44H = 0100   0100
        78H = 0111   1000
-----
        7CH = 0111   1100
(H) = (CY) = (P) = (Z) = (S) = 0
```

FUNCIÓN OR EXCLUSIVO

La instrucción OR exclusivo no es tan frecuentemente usada como las instrucciones AND y OR inclusivo. Se utiliza para complementar bits en un byte o para probar si algún bit no ha cambiado de valor. La Z-80 tiene también cuatro modos de direccionamiento para obtener el segundo operando de la función OR exclusivo: Registro, Inmediato, Registro Indirecto e Indexado. El primer operando se encuentra en el Acumulador y también en el Acumulador se almacena el resultado. Las

instrucciones con los diferentes direccionamientos son:

XOR r ; (A) = (A) + (r) - Registro
 XOR Dato(B) ; (A) = (A) + dato - Inmediato
 XOR (HL) ; (A) = (A) + (HL) - Registro Indirecto

Ejemplo.- En el registro B se tiene la lectura anterior de un puerto y en el registro C la lectura actual, probar si hubieron cambios en los valores de los bits entre las dos lecturas. Considerar (B) = 57H y (C) = 77H.

```

LD    A,B      ;(A) = 57H
XOR   C        ;(A) = 57H + 77H = 20H
JP    NZ,CAMBIO ;Salta si hay cambio
IGUAL: -        ;Continua aqui si no
        hay cambio
-
-
CAMBIO: -       ;Continua aqui si
        hay cambio.
  
```

	57H =	0101	0111	
OR - Ex	77H =	0111	0111	
	20H =	0010	0000	Cambio un bit

Ejemplo.- Complementar el valor de los bits 2, 1 y 0 del contenido de la localidad de memoria 5250H. Considerar que (5250H) = 7DH.

```
LD HL,5250H      ; (HL) = 5250H
LD A,(HL)        ; (A) = 7DH
OR 07H           ; (A) = 7EH    07H = 7AH
LD (HL),A        ; (5250H) = 7EH
```

```
OR-EX:  7DH = 0111 1101
        07H = 0000 0111
        -----
        7EH = 0111 1010
```

Se complementaron

COMPARAR (8 BITS)

La Z-80 cuenta con instrucciones que le permitan comparar dos datos de 8 bits, uno en el acumulador y otro que se obtiene usando diferentes direccionamientos. La operación de comparación es funcionalmente similar a la resta con la excepción de que los operandos no se alteran, con lo cual se evita el tener que salvar el contenido del Acumulador antes de realizar la comparación para no perder el dato. La función de comparación altera únicamente el estado de

las banderas de estado, las cuales se pueden probar utilizando las instrucciones de salto condicional.

Los modos de direccionamiento para obtener el segundo operando son los mismos que en la suma y en la resta de 8 bits: Registro, Inmediato, Registro Indirecto e Indirecto. Las instrucciones son:

CP r : Compara (A) con (r) - Registro
 CP n : Compara (A) con dato (8) - Inmediato

Como la comparacion es funcionalmente similar a la resta, las reglas para los cambios de la bandera de acarreo son iguales a la de la resta. Cuando hay acarreo con valor 0 saliendo del bit de mas alto orden indica que existe un prestamo por lo que la bandera Cy toma valor 1 ($Cy = 1$) y si sale con valor 1 indica que no hay prestamo, ($Cy = 0$).

Ejemplo.- Comparar el contenido del acumulador con el dato 05, considera que (A) = E5H.

CP	05H		: Compara (A) con 05H
E5H =	1110	0101	(Z) = 0
- 05H =	1111	1011	(Cy) = 0
<hr/>			
E0H =	1110	0000	

ya que los números difieren en signo. la bandera (CY) = 0 indica ahora que el contenido de B es mayor que el del Acumulador.

ROTAR Y GIRAR

La Z-80 cuenta con instrucciones que le permiten rotar y girar el dato contenido en el acumulador, en cualquier registro de propósito general (B, C, D, E, H y L) o a las localidades de memoria direccionadas por los registros IX, IY y HL. Todas las instrucciones, menos dos, actúan sobre la bandera de acarreo (CY)

Las operaciones de rotar generalmente se usan para una de las tres razones:

- 1) Multiplicar por potencias de dos
- 2) Dividir entre potencias de dos
- 3) Separar bits en un dato

Correr un dato una posición a la izquierda implica multiplicar por dos y una posición a la derecha implica dividir por dos. correr dos o tres posiciones a la izquierda o a la derecha implica multiplicar o dividir por 4 o por 8 respectivamente.

ROTAR EL ACUMULADOR.

Las primeras cuatro instrucciones de la tabla operan unicamente sobre el acumulador. Los mneonicos de las instrucciones son:

- 1) RRA
- 2) RLA
- 3) RLCA
- 4) RLCA

La instruccion RRA, Rotar a la Derecha el Acumulador a traves del acarreo, causa que el bit en la posicion menos significativa del Acumulador (A0) se rote a la bandera de acarreo, el contenido de la bandera de acarreo se rote a la posicion mas significativa del Acumulador (A7) y los bits en las posiciones An+1 se roten a las posiciones An.

La instruccion RLCA, Rotar a la Izquierda Circular el Acumulador, causa que el bit en la posicion mas significativa del Acumulador (A7) se rote a la bandera de acarreo y a la posicion menos significativa del Acumulador (A0), y los bits en las posiciones An se roten a las posiciones An+1.

ROTAR REGISTROS.

Las mismas cuatro funciones de la seccion anterior se pueden ordenar a cualquiera de los registros de proposito general o a las localidades de memoria. Las instrucciones en orden de equivalencia con las primeras cuatro son:

- 1) RR s
- 2) RL s
- 3) RRC s
- 4) RLC s

en donde "s" puede ser:

- 1) r - cualquiera de A, B, C, D, E, H y L
- 2) (HL)
- 3) (IX+d) Apuntadores a memoria.
- 4) (IY+d)

SALTO INCONDICIONAL INMEDIATO.

La instruccion JF direccion (16 bits) transfiere "incondicionalmente" el control de ejecucion a la instruccion que se encuentra en la localidad cuya direccion se especifica inmediatamente en los bytes 3 y 2 de la propia instruccion. La instruccion consiste de

tres bytes, en los cuales el primero contiene el código de operación (C3H), el byte dos contiene los 8 bits más bajos de la dirección y el byte tres contiene los 8 bits más altos de la dirección. La instrucción especifica "exactamente" la dirección a donde se debe saltar.

SALTO CONDICIONAL.

La instrucción de salto condicional es semejante a la de salto incondicional pero con la restricción de que se realiza o no el salto en base al valor actual de la bandera que se especifica en la propia instrucción. Las banderas que se pueden sensar son: Acarreo, Cero, Signo y Paridad. La instrucción es:

JP condicion , direccion (16 bits)

en donde el campo de "condicion" consiste de dos letras como máximo para indicar la bandera y el valor que debe tener. La instrucción consiste de tres bytes y tiene una configuración semejante a la del salto incondicional. La diferencia es que el primer byte (código de operación) tiene un campo de tres bits para indicar la bandera y el valor de interés.

"Las instrucciones de salto condicional no alteran el valor de las banderas".

Notar que el mnemonic pregunta por el estado del resultado y no por el de las banderas, aunque la bandera es realmente lo que indica el resultado. La instrucción JF Z ordena saltar si el resultado de una instrucción es cero, es decir, si $(Z) = 1$ no si la bandera Cero es 0 y la instrucción es No Cero, es decir, si $(Z) = 0$, no si la bandera Cero es 1.
bandera

TABLA INSTRUCCIONES DE SALTO CONDICIONAL

JF	NZ	-	Salta si el resultado no es cero, $(Z) = 0$
JF	Z	-	Salta si el resultado es cero, $(Z) = 1$
JF	NC	-	Salta si el resultado produce acarreo con valor cero, $(CY) = 0$
JF	C	-	Salta si el resultado produce acarreo con valor uno, $(CY) = 1$
JF	FO	-	Salta si el resultado tiene un numero impar de bits 1, $(P) = 0$
JF	FE	-	Salta si el resultado tiene un numero par de bits 1, $(P) = 1$
JF	P	-	Salta si el resultado es positivo, $(S) = 0$
JF	M	-	Salta si el resultado es negativo, $(S) = 1$

Es importante senalar que la computadora unicamente tiene relacion con la interfaz, no tiene comunicacion directa con el periferico. Es la interfaz quien tiene relacion directa con el periferico.

Existen interfaces programables y no-programables. Las interfaces no-programables son las interfaces cuyas características de funcionamiento se especifican por circuitos (hardware) y las programables son las que por sus características de funcionamiento se especifican por circuitos y por programacion.

INSTRUCCIONES DE ENTRADA/SALIDA (E/S).

La Z-80 cuenta un grupo de instrucciones de entrada y salida (E/S). Las instrucciones se pueden dividir en:

- 1) Instrucciones de E/S con el Acumulador.
- 2) Instrucciones de E/S usando el registro C.
- 3) Instrucciones de E/S de bloques

"Todas las instrucciones de este grupo son de dos bytes".

ENTRADA/SALIDA CON EL ACUMULADOR.

Las instrucciones de este subgrupo son compatibles

con las instrucciones IN y OUT de la 8080. Las instrucciones son:

IN A,(n) y OUT (n),A

algunos ensambladores también aceptan los siguientes mnemónicos IN A,n y OUT n,A.

Las dos instrucciones son de dos bytes, el primer byte especifica el código (IN = 0EH y OUT = 0FH) y el segundo byte especifica el número del puerto de E/S, utilizan direccionamiento inmediato. El número del puerto se encuentra en el rango de 0 a FFH (0 a 255). El número máximo de puertos es de 512, 256 de entrada y 256 de salida. La instrucción IN A,(n) ordena cargar en el Acumulador el contenido del puerto n. Durante esta instrucción el número del puerto se envía por las líneas A7-A0 y se activan las líneas 10RQ y RD por las líneas A15-H6 se envía un duplicado de "n".

La instrucción OUT (n),A ordena enviar el contenido del Acumulador al puerto n. Durante esta instrucción el número del puerto se envía por las líneas 10RQ y WR. Por las líneas A15-H6 también se envía un duplicado de "n". La ejecución de estas instrucciones no afecta las banderas de estado de la CPU.

Ejemplo.- Leer el contenido del puerto de entrada 59H cargarlo en la direccion 5000H y enviarlo al puerto de salida 20H.

```

IN    A,59H      ; (A) = Dato del puerto 59H
LD    (5000H),A  ; (5000H) = Dato leído
OUT   20H,A     ; Puerto 20H = Dato

```

Cuando los dispositivos perifericos son mas lentos que la CPU, esta necesita preguntar a cada periferico de entrada si ya tiene un dato listo para ser leído "Receptor Listo" y a cada periferico de salida si se encuentra listo para transmitir el proximo dato "Transmisor Listo".

Ejemplo.- Leer un dato del periferico con puerto de entrada de datos 35H y puerto de estados 25H. Enviar el dato leído al periferico de salida con puerto de salida 41H y puerto de estado 40H.

```

LEER: IN    A,25H      ;Lee estado del puerto de
           entrada.
BIT    0,A           ;Prueba el bit 0
JP    Z,LEER        ;Salta si no esta listo
IN    A,35H        ;Lee puerto de entrada
LD    B,A           ;Salva el dato en B

```

SALIR:	IN	A,40H	;Lee estado del puerto de salida.
	BIT	1,A	; Prueba el BIT 1
	JF	Z,SALIR	;salta si no está libre
	LD	H,B	;Regresa el dato salvado
	OUT	41H,A	;Envia el dato
FIN:	RET		;Regresa

Se considera en este ejemplo que la bandera "Receptor Listo" del periférico de entrada ocupa el bit 0 del puerto de estados 25H y la bandera "Transmisor Listo" del periférico de salida ocupa el bit 1 del puerto de estado 40H. El nivel alto de los bits indican que el controlador está listo.

Un puerto de entrada de datos de un periférico no puede tener el mismo número para el puerto de entrada de estados. Un puerto de salida de datos de un periférico puede tener el mismo número para el puerto de entrada de estados.

CONVERSION DE CODIGOS.

Algunas conversiones se pueden realizar utilizando circuitos, por ejemplo, existen circuitos integrados para realizar la conversión BCD a 7-Segmentos y los

ESTA TESIS NO DEBE SALIR DE LA BIBLIOTECA

circuitos UART (Receptor-Transmisor Asincrono Universal) convierte los datos ASCII al formato de teletipo o CRT y viceversa. Sin embargo, todavia se puede requerir de un programa para realizar parte del trabajo de la conversion.

CONVERSION HEXADECIMAL (ASCII) A BINARIO.

El dato 35H tiene la siguiente representacion en binarios:

$$35H = 0011 \quad 0101$$

Los codigos ASCII de los numeros 3 y 5 son:
ASCII de 3 = 0011 0011 y ASCII de 5 = 0011 0101

Si el codigo ASCII de los numeros (del 0 al 9) se le resta el valor 30H se obtiene el valor binario del numero.

$$\begin{array}{r} \text{Codigo ASCII de 3} = \quad 0011 \quad 0011 \\ \quad \quad \quad \quad - 30H = \quad 1101 \quad 0000 \\ \hline \text{VALOR BINARIO} \quad = \quad 0000 \quad 0011 \end{array}$$

Otra forma de obtener el valor binario de un numero del valor ASCII es "enmascarando" los cuatro bits mas significativos del codigo. Esto se realiza con una funcion AND entre el codigo ASCII y el dato 0FH.

CONVERSION BINARIO A HEXADECIMAL (ASCII)

Frecuentemente se requiere que el valor binario contenido en una o mas localidades se envíe a un teletipo o a una terminal de video. Para esto se tiene que realizar el proceso inverso al de la seccion anterior. A continuacion se ilustra el algoritmo para convertir un dato binario a dos digitos hexadecimales (ASCII).

- 1) Separar los 8 bits del byte en dos grupos de cuatro, cada grupo en un byte. Por ejemplo, 4CH en 40H y 0CH.
- 2) El byte que tenga los cuatro bits mas significantivos del dato original se debe rotar cuatro bits a la derecha. Del ejemplo, 40H a 04H.
- 3) Sumar 30H a los bytes resultantes si el valor de ellos esta en el rango de 00 a 09 y sumar 37H si estan en el rango de 0AH a 0FH.

04H + 30H = 34H ASCII de 4
0CH + 37H = 43H ASCII de C

CONVERSION DECIMAL (ASCII) A BINARIO.

A continuacion se ilustra una rutina para convertir a binario una cantidad decimal (ASCII).

1) Leer un dígito decimal (ASCII) y convertirlo a su valor binario. Primer resultado.

2) Leer otro dígito decimal (ASCII) y convertirlo a su valor binario. El primer resultado se multiplica por 10 y se suma al segundo dígito, segundo resultado. Si no hay otro dígito, el segundo resultado es el resultado final.

3) Leer otro dígito decimal (ASCII) y convertirlo a su valor binario. El segundo resultado se multiplica por 10 y se suma al tercer dígito, tercer resultado. Si no hay otro dígito el tercer resultado es el resultado final.

4) Continuar el mismo procedimiento.

Por ejemplo, obtener el valor binario del dato 255.

1) Leer el dígito 2, R1 = 02H

2) Leer el dígito 5, se multiplica por R1 por 0AH y se suma el dígito último R2 = 14H + 05H = FFH.

Resultado final = FFH.

REGISTRO DE REFRESCAR MEMORIA (R)

Las memorias dinamicas no conservan su contenido durante mucho tiempo, deben ser leidas o refrescadas periodicamente (valor tipico de 2 milisegundos) para que puedan retener su contenido. La Z-80 contiene un registro contador de 7 bits que permite direccionar a las filas de las memorias dinamicas a traves de las lineas A6-A9 del bus de direccion. El contenido del registro R se incrementa en uno despues de cada ciclo Fetch y el contenido se envia al bus de direccion mientras la CPU no esta accedendo a la memoria para leer o escribir datos. El registro R generalmente no es usado por el programador.

BANDERAS DE ESTADO.

La CPU Z-80 cuenta con un "registro de banderas" (F) que le permite monitorear los resultados de las operaciones aritmeticas, logicas y otras de la CPU. Aunque el "registro de banderas" se considera un registro de 8 bits como los otros siete bits activos de la CPU, es mas una coleccion de bits de informacion agrupados convenientemente que un registro de proposito general. Las banderas se actualizan despues de cada operacion con alguno de los registros, aunque no todas

las operaciones modifican a todas las banderas y algunas no modifican a las banderas. De los 8 bits del registro de banderas unicamente seis registran informacion util para programador.

Cuatro de estas banderas (C, Z, S y P/V) se pueden probar para conocer su valor "actual" y en base a este continuar o alterar la secuencia de un programa. La Z-80 cuenta con instrucciones de condicion que le permiten saltar (JP), llamar a una rutina (CALL) o regresar de una rutina de acuerdo al estado de las banderas. La CPU cuenta tambien con un registro de banderas "no activo", F .

Es importante indicar que el estado de las banderas no se altera hasta que no se ejecute otra instruccion que las pueda afectar. Esto significa que las instrucciones de salto de condicionales (para la logica de decisiones) no tienen que ejecutarse inmediatamente despues de ejecutar una instruccion que altera el estado de las banderas, pero si antes de la ejecucion de la proxima instruccion que pueda alterar las banderas.

BANDERA CERO (Z) (BIT 6 DEL REGISTRO F).

La bandera cero (Z) toma nivel alto si el resultado

de ciertas operaciones es cero y nivel bajo si el resultado es no cero. La tabla muestra las instrucciones que afectan a la bandera cero.

Como se muestra en la tabla, la bandera Z se afecta principalmente por operaciones Aritméticas, Lógicas y de Corrimiento. Las instrucciones de cargar y de almacenar (LD) no afectan a la bandera Z con la excepción de las instrucciones LD A,1 y LD A,R.

El grupo de Búsqueda se realiza básicamente en función de comparaciones o restas, por lo que se afecta a la bandera Z. En este grupo de Búsqueda, en el momento en que la bandera Z toma nivel cero, el valor último comparado es el valor que se está buscando. El grupo de Prueba de Bit realiza una función lógica AND y la bandera Z toma nivel alto o bajo dependiendo del resultado.

El grupo de Entrada y Salida tiene algunas instrucciones que afectan el estado de la bandera Z en función del dato de entrada o del valor del registro B.

Exceptuando a las instrucciones que se muestran en la tabla, ninguna otra instrucción afecta el estado de la bandera Z. La bandera Z se puede probar para una

variedad de condiciones, algunas de las mas comunes son:

- 1.- La igualdad de dos operandos despues de una comparacion CF.
- 2.- Incrementar o decrementar un contador hasta el valor 0.
- 3.- Preguntar por el valor de un bit.

BANDERA DE ACARREO (CY) (BIT 0 DEL REGISTRO F)

La bandera de Acarreo (CY) registra el valor que sale del bit de mas alto orden del resultado de las operaciones aritmeticas, de rotar y de girar. Las operaciones logicas limpian la bandera de Acarreo.

La bandera de Acarreo se puede usar para:

- 1.- Probar el resultado de una comparacion
- 2.- Probar el resultado de una operacion de rotar o girar
- 3.- Operaciones multibyte o precision multiple.

Ejemplo: Sumar los contenidos del Acumulador y del registro B. Considerar que (A) = AEH y (B) = 74H.

```
ADD A,B ;(A) = (A) + (B)
AE    =    1010    1110
+ 74  =    0111    0100
-----
122   =    0010    0010
```

variedad de condiciones, algunas de las mas comunes son:

- 1.- La igualdad de dos operandos despues de una comparacion CP.
- 2.- Incrementar o decrementar un contador hasta el valor 0.
- 3.- Preguntar por el valor de un bit.

BANDERA DE ACARREO (CY) (BIT 0 DEL REGISTRO F)

La bandera de Acarreo (CY) registra el valor que sale del bit de mas alto orden del resultado de las operaciones aritmeticas, de rotar y de girar. Las operaciones logicas limpian la bandera de Acarreo.

La bandera de Acarreo se puede usar para:

- 1.- Probar el resultado de una comparacion
- 2.- Probar el resultado de una operacion de rotar o girar
- 3.- Operaciones multibyte o precision multiple.

Ejemplo: Sumar los contenidos del Acumulador y del registro B. Considerar que (A) = AEH y (B) = 74H.

```
ADD A,B : (A) = (A) + (B)
AE      =   1010   1110
+ 74    =   0111   0100

122    1 0010   0010
```

Produce un acarreo con valor 1 por lo que la bandera CY toma nivel 1.

En este ejemplo el valor de la bandera de Acarreo indica que el resultado no se puede almacenar en 8 bits, se requieren 9.

BANDERA DE PARIDAD/SOBREFLUJO (P/V)

La bandera de Paridad/Sobreflujo (P/V) es una bandera con doble proposito. Indica la paridad del resultado cuando se ejecutan operaciones logicas (tales como AND A,B) y representa el sobreflujo cuando se ejecutan operaciones aritmeticas con numeros con signo que utilizan el complemento a dos. La bandera de sobreflujo indica que el resultado tiene error de exceder al numero positivo maximo (+ 127 o 7FH) o al numero negativo maximo (-128 o 80H) que se pueden representar con la notacion de complemento a dos.

DIRECCIONAMIENTO DE MEMORIA.

Una parte importante en la programacion es el de entender las formas de direccionamiento a las localidades de memoria que contienen los datos que se deben procesar con las instrucciones. La Z-80 tiene once modos diferentes para direccionar los datos almacenados en la memoria y en los registros.

La Z-80 tiene los siguientes modos de direccionamiento:

- 1.- Implicito
- 2.- Inmediato
- 3.- Inmediato extendido
- 4.- Registro
- 5.- Registro indirecto
- 6.- Extendido
- 7.- Pagina cero modificada
- 8.- Relativo
- 9.- Indexado
- 10.- Bit
- 11.- Apuntador del Stack

DIRECCIONAMIENTO IMPLICITO

El direccionamiento implicito se refiere a operaciones en donde el codigo de operacion implica automaticamente a uno o mas registros de la CFU como los que contienen los operandos. El codigo de operacion es fijo, no tiene "campos variables" y la instruccion siempre ejecuta la misma funcion. Ejemplo de estas son las instrucciones CFL (Completar el Acumulador) y RRA (rotar el acumulador a la derecha a traves de acarreo).

El codigo de operacion de la instruccion CFL es 2FH.

Esta instrucción toma el contenido del Acumulador, lo complementa a uno (cambia los ceros a unos y los unos a ceros) y deposita el resultado en el Acumulador. Las banderas de condición no se afectan; la fuente y el destino son fijos.

DIRECCIONAMIENTO INMEDIATO.

Una instrucción con direccionamiento inmediato es la que contiene en el byte 2 o byte 4 de la propia instrucción el dato de la transferencia (operando), es decir, el dato es parte integrante de los bytes que constituyen la instrucción, ver figura . Este tipo de direccionamiento es utilizado cuando se necesita cargar o efectuar una operación aritmética o lógica con un dato constante. Por ejemplo, para cargar 07H al Acumulador se puede utilizar la instrucción LD A, N; en nuestro caso $N = 07$. La figura muestra el formato de esta instrucción es 3EH y el segundo byte es el dato que se va a cargar en el Acumulador.

1.- El contenido de la localidad (código) apuntada por el Contador del Programa (PC) se carga en el Registro IR. Se incrementa el PC en uno.

2.- La Unidad de Control ordena (después de decodificar) que el contenido de la localidad (dato

inmediato) apuntada por el PC se cargue en el registro "r" indicado por el código de la instrucción. El PC se incrementa en uno, apunta a la próxima instrucción.

En general, este tipo de direccionamiento lo utilizan las instrucciones Aritméticas y Lógicas de 8 bits. Ejemplo de estas son ADD N, XOR N, AND N.

DIRECCIONAMIENTO DE REGISTRO

Muchas instrucciones utilizan a los registros de la CPU para recibir o proporcionar un dato. El código de la instrucción tendrá un campo (bits) que se usará para especificar el o los registros que se utilizarán en la ejecución de la instrucción LD r,s. Esta instrucción ordena que el contenido del registro s se cargue en el registro r, en donde r y s pueden ser cualquiera de los registros activos de 8 bits de la CPU. El código general de operación tiene 3 bits para especificar el registro fuente s del dato y 3 bits para registro destino r.

1) El contenido de la localidad (código) apuntada por el PC se carga en el registro IR. La Unidad de Control ordena copiar en el registro "r" el contenido del registro "s". El PC se incrementa en uno, apunta a la próxima instrucción.

Código de Operacion Uno o dos bytes
 Operando

DIRECCIONAMIENTO INMEDIATO

Memoria

X	-	0011	1110	- 3EH = Código
X + 1	-		N	- Dato Inmediato

INSTRUCCION LD A,N

		Código de Operacion	Uno o dos bytes
Dato	N	byte bajo	
	N	byte alto	

DIRECCIONAMIENTO INMEDIATO EXTENDIDO

		Direccion	Memoria
X	-	0 1 H	- Código
X + 1	-	6 8 H	
X + 2	-	7 9 H	

INSTRUCCION LD BC, 796H

En la tabla se puede notar que todas las combinaciones de los 3 bits indican un registro con la excepcion de la combinacion 1111. Esta combinacion implica otro tipo de direccionamiento.

Los grupos de instrucciones que utilizan este tipo de direccionamiento son los de Aritmetica y Logica de 8 y 16 bits, Rotar, Girar, Limpiar y Probar bits.

TRANSFERENCIA DE 8 BITS.

La CPU Z-80 tiene un grupo de instrucciones que permiten la transferencia de datos de 8 bits desde un registro (A, B, C, D, E, H y L) de la CPU, o de un registro a otro dentro de la CPU.

ENTRE REGISTROS (8 BITS)

La primera instruccion que estudiaremos de este grupo es LD, r, s la cual consta de solo un byte. En esta instruccion la transferencia es del registro "s" (Fuente) al registro "r" (Destino, en donde s y r pueden ser cualquiera de los siete registros de proposito general (A, B, C, D, E, H y L) de la CPU Z-80. El codigo de operacion de esta instruccion es de un solo byte y tiene el siguiente formato:

0 1 R R R S S S

En donde SSS es el código del registro "Fuente" y RRR es el código del registro "Destino". Por ejemplo, para mover el contenido del registro 5 al registro A se puede usar la instrucción LD A,5 cuyo código de operación es:

0 1 1 1 1 0 0 0 = 7BH

Después que la Z-80 ejecuta la instrucción anterior el contenido del registro A será igual al que tiene el registro B, el contenido previo de A se pierde. Es decir, la instrucción LD r,s ordena que el contenido del registro "s" (Fuente) se "copie" en el registro "r" (Destino). El valor del registro Fuente no se altera.

Cuando el registro Fuente es el mismo que el Destino, el contenido del registro inherente no se altera. Aunque es una instrucción válida no tiene ninguna utilidad notable y puede considerarse, como una instrucción de "hacer nada", ya que lo único que se logra es perder tiempo de procesamiento. Estas instrucciones son: LD A,A, LD B,B LD C,C LD D,D LD E,E, LD H,H y LD L,L.

CARGAR INMEDIATO AL REGISTRO (8 BITS)

La Z-80 cuenta con una instrucción que le permite cargar a cualquiera de sus siete registros de propósito

CONVERTIDOR ANALOGICO DIGITAL TLC548.

EL TLC548.

El microprocesador y software necesitan solo leer la conversion previa y empezar la conversion con i/o clock

Secuencia Tipica de Control.

1.- CS LOW

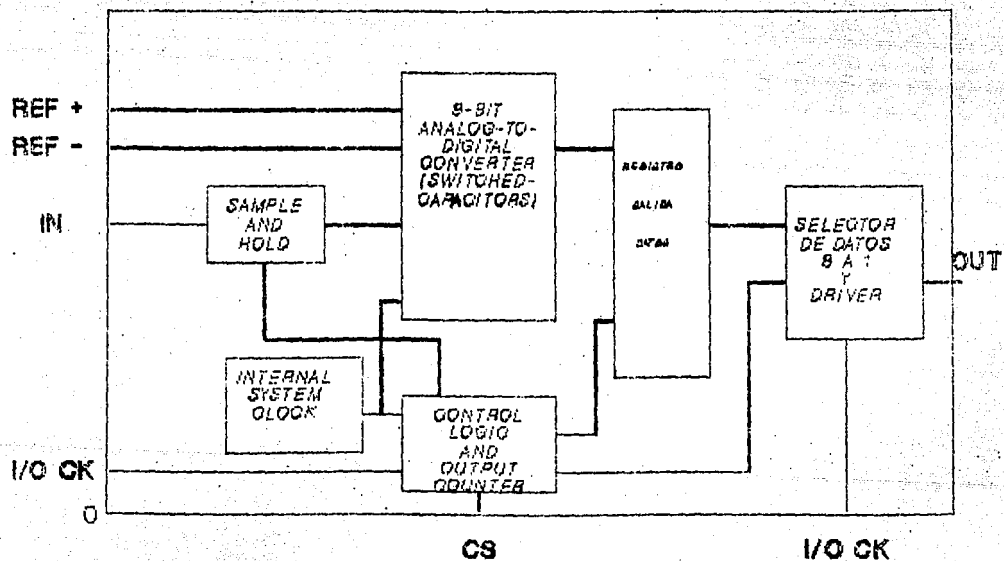
Despues de una transicion (de High a Low), el circuito interno del TLC548 espera dos flancos de subida y entonces un flanco de bajada del reloj interno del sistema antes de reconocer la transicion, este retardo minimiza errores causados por ruido en CS. El bit mas significativo, resultado de la conversion previa entonces aparece el pin DATA OUT.

2.- Los flancos negativos de los primeros cuatro I/O clocks recorren los 2do. 3ro. 4o. y 5o. bits mas significativos del resultado de la conversion previa.

3.- El sample-and-hold del circuito empieza a muestrear la entrada analoga despues del cuarto flanco de bajada.

Esta operacion basicamente involucra cargar capacitores internos al nivel de voltaje de la entrada analoga.

TLC548 8 BIT A/D



4.- Tres ciclos de reloj mas son aplicados al pin I/O, el 60. 70. y 80. bits del resultado de la conversion previa se recorren en los flancos negativos de esos ciclos de reloj.

5.- El 80. y final ciclo de reloj se aplica al pin I/O clock.

El flanco de bajada de este ciclo de reloj completa el proceso de muestreo analogico e inicia la funcion HOLD.

6.- La conversion se realiza durante los proximos 30 ciclos de reloj del sistema; despues del ciclo final I/O clock, CS debe ponerse en High o el clock I/O debe permanecer bajo por lo menos durante 30 ciclos de clock del sistema para permitir la funcion de conversion.

OUT (CS HIGH),A

CS se pone "High" despues del 80. clock I/O para deshabilitar entradas y salidas y la conversion puede proceder sin disturbios.

IV. GENERADOR DE TONOS DTMF

A. DIVISION DE FRECUENCIA

Dividiendo la frecuencia 3.579545 entre ocho (lo cual ya viene hecho en el receptor DTMF) se obtiene la frecuencia 447.443 KHz la cual, para obtener el grupo alto de frecuencias se divide entre una de cuatro posibles constantes y despues se divide entre 16. asi tendremos que para generar el tono 1209 Hz dividimos primero entre 23 lo que da 19454.0 Hz y luego esto entre 16 da 1215.87 Hz que esta dentro del 1% del mismo modo se obtienen las demas frecuencias:

FRECUENCIA BASE: 447.443 KHz

DIVISOR	FRECUENCIA	DIVISOR	FRECUENCIA FINAL
40	11186	16	699.1
36	12428.9	16	776.8
33	13558.8	16	847.4
30	14914.7	16	932.1
23	19454.0	16	1215.8
21	21306.8	16	1331.6
19	23549.6	16	1471.8
17	26320.1	16	1645.0

Se tendran dos circuitos divisores uno para el grupo alto y otro para el grupo bajo, y cada uno de estos es programable para lo cual se implementa con un circuito

contador binario de 5 etapas para el grupo de frecuencias DTMF ALTO (23 = 10111), y con un contador binario de seis etapas para el grupo DTMF BAJO (40 = 101000), Para tener la característica programable en el divisor utilizamos dos circuitos selector de datos de 4 a 1 mencionado anteriormente cuando el contador llega al numero elegido como divisor, este numero esta presente en forma binaria en las salidas del contador, y es este arreglo de bits el que debe habilitar el RESET del contador para que haga la funcion de divisor.

Tenemos que los divisores del grupo ALTO son de cinco BITS, siendo tres bits iguales para los cuatro divisores por lo tanto basta con variar dos bits para obtener los divisores. En el grupo BAJO los divisores son de seis bits y si tomamos los correspondientes a los numeros 40, 36, y 33 estos tienen tres bits en comun y para el numero 30 se hace un AND de los bits correspondientes.

23	1 0 1 1 1	40	1 0 1 0 0 0
21	1 0 1 0 1	36	1 0 0 1 0 0
19	1 0 0 1 1	33	1 0 0 0 0 1
17	1 0 0 0 1	30	0 1 1 1 1 0

La figura muestra la forma de obtener los divisores de los dos grupos utilizando dos bits para programar el grupo alto y dos bits para programar el grupo bajo.

Tomaremos como base la correspondencia dígitos-código binario 2 de 8 que es la siguiente:

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

De aquí tomamos los dos primeros bits (mas significativos) para generar las frecuencias bajas (rengion) y los dos restantes para generar las frecuencias altas (columnas) de que se trate:

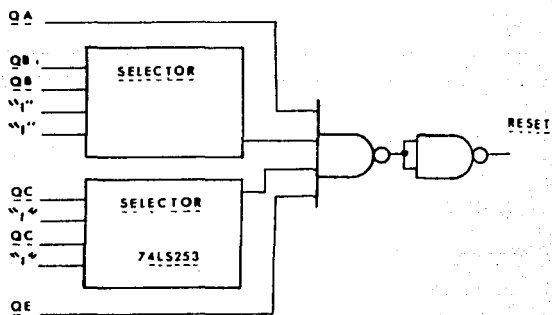
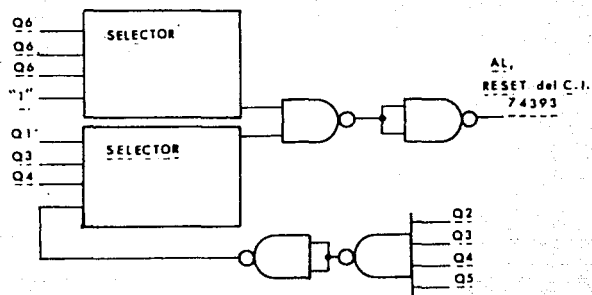
1209	0 0	697	0 0
1336	0 1	770	0 1
1447	1 0	852	1 0
1633	1 1	941	1 1

B. MULTIPLEXORES

Hasta aquí aseguramos que al marcar un dígito se obtengan los divisores adecuados, así por ejemplo si marcamos un "1" (frecuencias 1209 y 697) se harán las divisiones entre 23 y entre 40 obteniendo las frecuencias 19454 y 11186 que aun son una forma de onda cuadrada, por lo que estas frecuencias deben entrar en forma separada a otra etapa que haga la conversión a la frecuencia final y a una forma de onda senoidal.

Como el divisor requerido es 16 se usará un contador binario de cuatro etapas cuyas cuatro salidas forman 16 distintas combinaciones de 4 BITS antes de repetir la secuencia, usaremos 4 combinaciones distintas para representar cada uno de los cuatro cuadrantes de la onda senoidal que se obtendrá con ocho valores de escalon, obtenidos estos de la selección hecha a un MULTIPLEXOR de 8 a 1 donde cada escalon corresponde a un voltaje fijo en una entrada del multiplexor. Obteniendo la tabla siguiente:

CONTADOR				MULTIPLEXOR				
QD	QC	QB	QA	C	B	A	ENTRADA SELEC.	V. DE SALIDA
0	0	0	0	0	0	0	S0	0.2
0	0	0	1	0	0	1	S1	0.56
0	0	1	0	0	1	0	S2	0.83
0	0	1	1	0	1	1	S3	0.98
0	1	0	0	0	1	1	S3	0.78
0	1	0	1	0	1	0	S2	0.83



DIVISORES PROGRAMABLES PARA DTMF

0 1 1 0	0 0 1	S1	0.50
0 1 1 1	0 0 0	S0	0.20
1 0 0 0	1 0 0	S4	-0.20
1 0 0 1	1 0 1	S3	-0.50
1 0 1 0	1 1 0	S6	-0.83
1 0 1 1	1 1 1	S7	-0.98
1 1 0 0	1 1 1	S7	-0.98
1 1 0 1	1 1 0	S6	-0.83
1 1 1 0	1 0 1	S5	-0.50
1 1 1 1	1 0 0	S4	-0.20

Así tenemos la aproximación a la función :

$$V_o = 2 V \sin \frac{2\pi f t}{16}$$

la salida de cada multiplexor se aplica a un amplificador operacional como se indica en la figura y por último se suman en otro amp. op. para tener el doble tono (DTMF), las formas de onda que se obtuvieron con un osciloscopio se muestran en el apéndice.

Para generar la pausa, un bit controla el reset de los dos contadores divisor entre lo lo cual corta el tono.

V. CONTROL DE DISPOSITIVOS.

A. PRUEBAS DE LABORATORIO

I.- Comprobar en un osciloscopio los tonos generados y graficarlos con un Plotter conectado al osciloscopio.

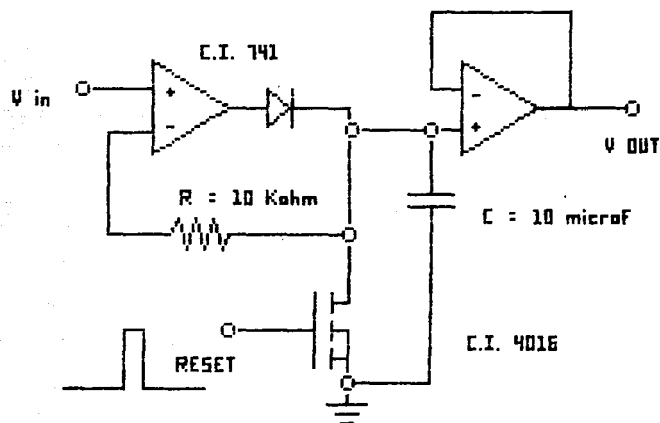
II.- Verificar que los programas en lenguaje de maquina controlen adecuadamente los dispositivos mencionados.

Para poder emplear el convertidor analogico digital como indicador del nivel de recepcion se requiere un detector de pico positivo, este muestrea y retiene el valor maximo de amplitud, y de este modo no interesa en que momento se efectua la conversion, mientras que el periodo de muestreo sea mayor que un ciclo del tono de 440 Hz (2.2 mseg).

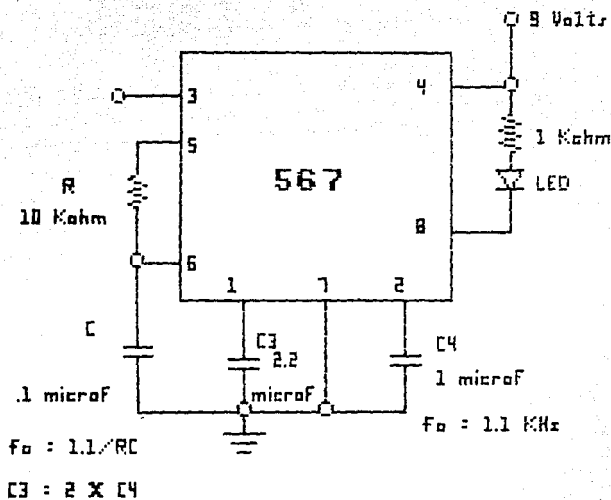
El circuito de prueba constara entonces de la sinclair ZX81, el GENERADOR de tonos DTMF programable, el DETECTOR de tonos DTMF, el DETECTOR de tono de 440 Hz, el convertidor analogico/digital, un visualizador de digitos y el detector de pico positivo. Ademas, para efectos de prueba se incluye un circuito que hace la decodificacion de tres digitos para habilitar la llamada a la extension correspondiente.

Este puede ser simple, formado por un C.I. 741542 que convierte las salidas binarias del detector DTMF a decimal, y con las salidas decimales se hacen un nand de los numeros que forman la extension, de modo que la salida de la nand mande tres pulsos de reloj a un circuito contador JOHNSON y la salida "3" de este active un switch analogico que mande el tono de ring back (440 Hz). En este caso, por ejemplo si los tres numeros son 1,2,3 cualquier combinacion de ellos activara el tono de ring back 1,2,3 ; 2,1,3 ; 1,1,1 ; etc. mientras que 1,2,4 no lo hara. Para que la extension corresponda solo a un arreglo de tres digitos en secuencia se requiere un circuito como se muestra en la figura, donde el selector de datos de 8 a 1 es controlado por un contador

DETECTOR DE PICO POSITIVO



CIRCUITO DETECTOR DE TONO DE 440 Hz



adicional que hace el avance a la siguiente entrada, así solo la secuencia 1,2,3 habilita el ring back, y no las combinaciones 1,3,2 ; 2,1,3; etc. Aquí usaremos el primer metodo por ser mas simple su implementación.

La operación completa es como sigue:

- 1.- Se carga el programa
- 2.- se inicia con "GOTO 5045"
- 3.- el programa pregunta numero de extensiones a marcar
- 4.- el programa pregunta las tres cifras de cada extension
- 5.- el programa despliega en pantalla las extensiones
- 6.- se inicia con "CONT"
- 7.- manda tres digitos con sus pausas
- 8.- toma el valor de un bit del puerto B correspondiente
- 9.- Hace la conversión y guarda 8 bits
- 10.- regresa a BASIC y despliega en pantalla el edc. del ring back de acuerdo a lo siguiente.

La Sinclair tiene la característica de que si se llama un programa en lenguaje de maquina con el comando FRINT USR x, al regresar a Basic despliega en pantalla el contenido de los registros BC. Tomando el bit con peso 8 (1000) en la entrada del Z80 por puerto A y cargándolo en el registro B nos dara en pantalla 2,048 para cuando no hay ring back / cero sumado al registro C (que puede ser de 0 a 255 que

corresponde de 4 Volts a 3 Volts resultado de la conversión A/D cuando hay ring back.

Así tendremos que el el valor pico positivo del tono 440 Hz que regresa es de 3 Volts, corresponde al decimal 153 y hexadecimal 99H = 1001 1001 y aparece en pantalla como 153

0800H = 2.048 = 0000 1000 0000 0000
REG. B REG. C

5 V = 255 = FFH = 1111 1111

0 V = 0 = 00H = 0000 0000

En el apéndice se muestran, el listado del programa en Basic y el listado del programa en lenguaje de máquina, este último con las localidades de memoria donde se ubica e indicaciones de las instrucciones, puertos y bits utilizados para implementar las distintas funciones, también se incluyen los diagramas de los circuitos integrados usados y de interconexión entre las etapas descritas anteriormente.

A N E X O I

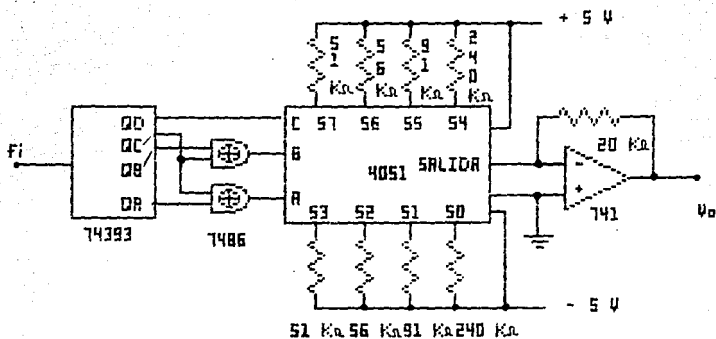
- . CODIGOS HEXADECIMAL Y BINARIO 2 DE 8
- . APROXIMACION POR 16 ESCALONES A UNA ONDA SENOIDAL
- . C.I. RECEPTOR DE DTMF
- . VISTA DEL MODULO # 1
- . GRAFICAS DE LOS TONOS
- . PROGRAMA EN BASIC PARA MANDAR NUMEROS TELEFONICOS (CON SUBROUTINAS EN LENGUAJE DE MAQUINA)
- . PROGRAMA EN LENGUAJE DE MAQUINA PARA CONTROL DEL GENERADOR DE TONOS Y DEL CONVERTIDOR A/D SALIDA SERLE.

CODIGOS HEXADECIMAL Y BINARIO 2 DE 8

HEXADECIMAL BINARIO 2 DE 8

DIGITO	D8	D4	D2	D1	D8	D4	D2	D1
1	0	0	0	1	0	0	0	0
2	0	0	1	0	0	0	0	1
3	0	0	1	1	0	0	1	0
4	0	1	0	0	0	1	0	0
5	0	1	0	1	0	1	0	1
6	0	1	1	0	0	1	1	0
7	0	1	1	1	1	0	0	0
8	1	0	0	0	1	0	0	1
9	1	0	0	1	1	0	1	0
0	1	0	1	0	1	1	0	1

EL CODIGO BINARIO 2 DE 8 SE EMPLEA EN EL GENERADOR DE TONOS DTMF, Y EL C. I. RECEPTOR DE LOS TONOS POSEE UNA ENTRADA (PIN 2) PARA PROGRAMAR QUE SUS SALIDAS (D1, D2, D4, D8) ESTEN EN UNO U OTRO DE ESTOS DOS CODIGOS.

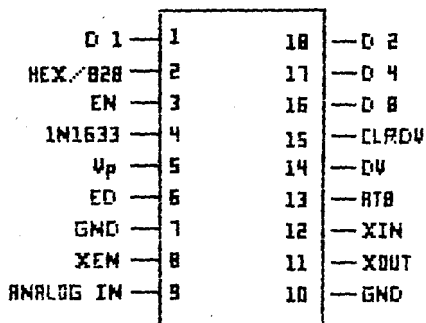


$$V_o = 24 \text{ SEN } \frac{2 \pi f_i t}{16}$$

OBTENCION DE UNA ONDA SENOIDAL APROXIMADA POR 16 ESCALONES

FIGURA

CIRCUITO INTEGRADO RECEPTOR DE DTMF



ALIMENTACION V_p=5V

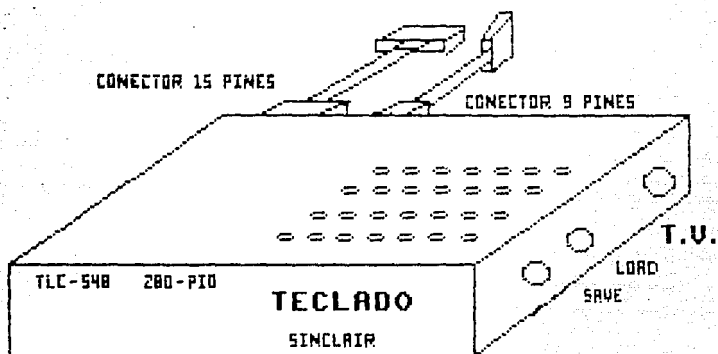
1N1633 INHIBE LA DETECCION DE 1633 Hz

DV INDICA DATOS VALIDOS (DATA VALID)

RT8 SALIDA DE RELOJ (CRISTAL /8)

EN HABILITA LAS SALIDAS D1.D2.D4.D8

MODULO # 1



FIGURA

GRAFICAS OBTENIDAS EN UN OSCILOSCOPIO

DE LAS SIGUIENTES GRAFICAS LAS MARCADAS COMO TONO 2 CORRESPONDEN A LA SALIDA DEL GENERADOR DE TONOS DTMF BISCAPETO (SIN CONDENSADOR).

Y LAS GRAFICAS MARCADAS COMO HEN0, HEN1, HEN2, HEN3 CORRESPONDEN A LA MISMA SALIDA PERO INCLUYENDO EL CONDENSADOR.

LO ANTERIOR CON EL OBJETO DE ADECUAR LOS ESCALONES Y EL EFECTO DE INCLUIR EL CONDENSADOR.

LAS GRAFICAS MARCADAS COMO TELEFONO Y C.I. DTMF CORRESPONDEN RESPECTIVAMENTE A LA SALIDA DE UN APARATO TELEFONICO (EN EL CABLE DE SALIDA); Y A UN CIRCUITO INTEGRADO GENERADOR DE TONOS DTMF.

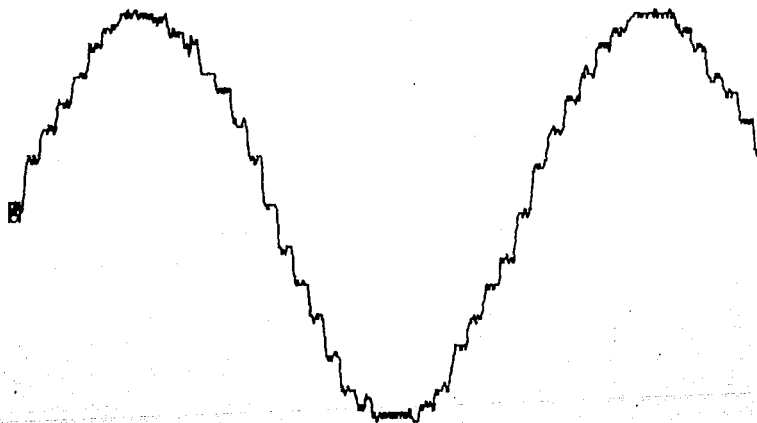
EN LAS PRUEBAS REALIZADAS EL RECEPTOR DE TONOS DTMF DECODIFICA CORRECTAMENTE TODAS LAS SALIDAS MENCIONADAS ANTERIORMENTE.

Timebase _____ Status: Acquisition Complete_____

Mode [Single] Auto Scale [Period]
Range 1.00 ms [Real Time] Reference [Left]
Delay 0.00000 s Sampling @ 1.00 MHz

Graph [1] 200 mV/div -12.0 mV 100 μ s/div 0.000 s

1: [Chan 1]
TELEFONO



Display

Status: Acquisition Complete

Graticule Type [Grid]

Reference Lines [Off]

Number of Graphs [1]

Accumulate Mode [Disabled]

Connect Dots [On]

Data Filter [On]

Graph [1]

200 mV/div

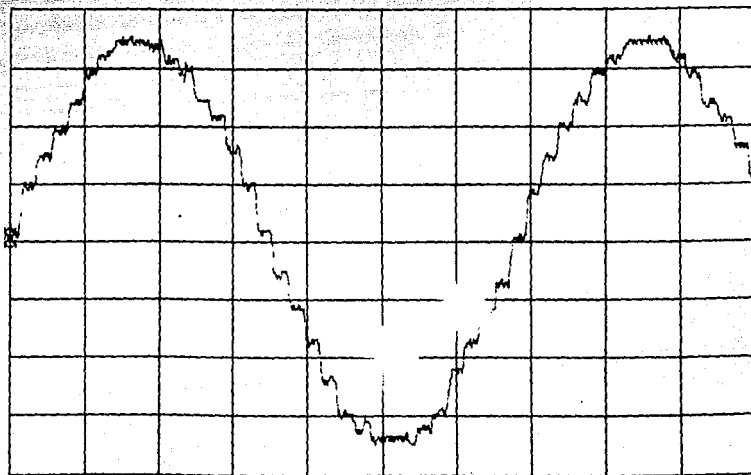
-12.0 mV

100 μ s/div

0.000 s

1: [Chan 1]

0.1 DTMF



Status: Acquisition Complete

Freq 1 = 777.2 Hz

V max 1 = 2.26 V

Freq 1 = 777.2 Hz

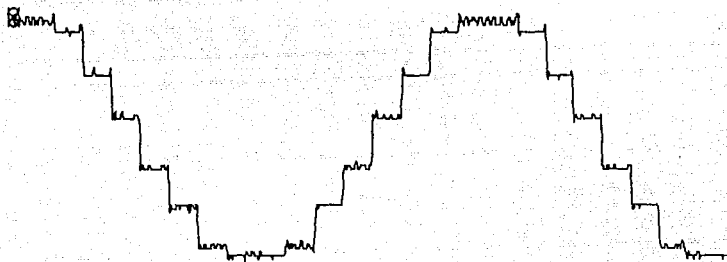
V min 1 = -2.19 V

Freq 1 = 777.2 Hz

V rms 1 = 1.558 V

Graph [1] 1.00 V/div 0.00 V 200 μ s/div -1.000 ms

1: [Chan 2]
TONE 2



Status: Acquisition Complete

Freq 1 = 1.332 kHz

V max 1 = 2.45 V

Freq 1 = 1.332 kHz

V min 1 = -2.19 V

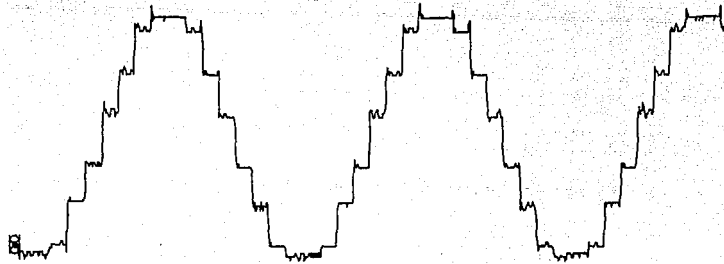
Freq 1 = 1.332 kHz

V rms 1 = 1.593 V

Graph [1] 1.00 V/div 0.00 V 200 μ s/div -1.000 ms

1: [Chan 2]
TONE 2

119



Status: Acquisition Complete

Freq 1 = 1.473 kHz

V max 1 = 2.39 V

Freq 1 = 1.473 kHz

V min 1 = -2.19 V

Freq 1 = 1.473 kHz

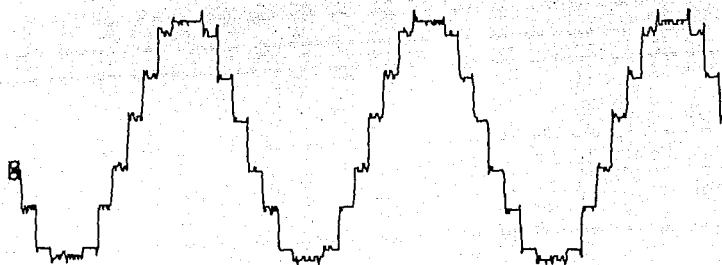
V rms 1 = 1.595 V

Graph [1] 1.00 V/div 0.00 V 200 μ s/div -1.000 ms

1: [Chan 2]

TONO 2

120



Status: Acquisition Complete

Freq 1 = 1.645 kHz

V max 1 = 2.39 V

Freq 1 = 1.645 kHz

V min 1 = -2.19 V

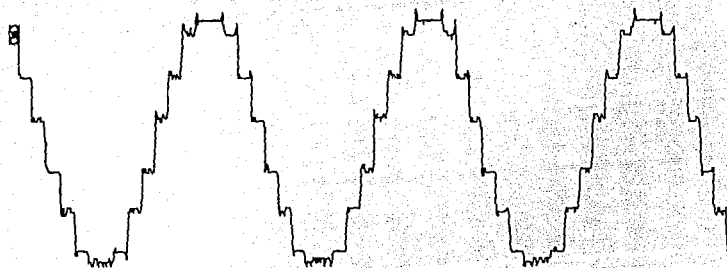
Freq 1 = 1.645 kHz

V rms 1 = 1.597 V

Graph [1] 1.00 V/div 0.00 V 200 μ s/div -1.000 ms

1: [Chan 2]

TONO 2



Display _____ Status: Acquisition Complete_____

Graticule Type	[Grid]	Reference Lines	[Off]
Number of Graphs	[4]	Accumulate Mode	[Disabled]
Connect Dots	[On]	Data Filter	[On]

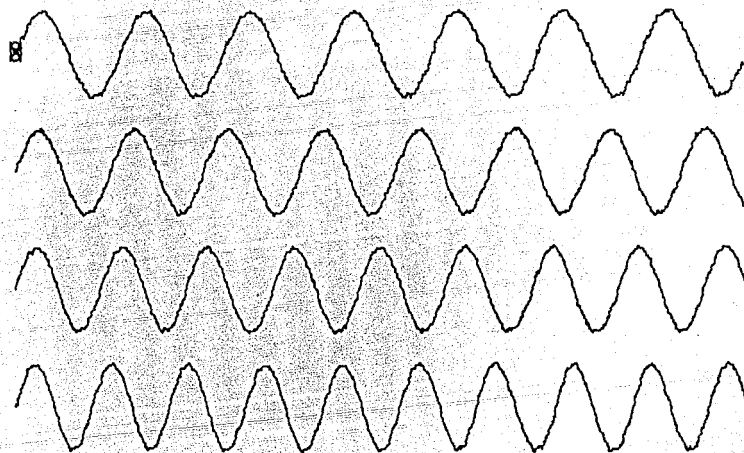
Graph [1] 800 mV/div -12.0 mV 1.00 ms/div 0.000 s

1: [Mem 0]
700 HZ

2: [Mem 1]
770 HZ

3: [Mem 2]
848 HZ

4: [Mem 3]
949 HZ



Display

Status: Acquisition Complete

Graticule Type [Grid]
Number of Graphs [4]
Connect Dots [On]

Reference Lines [Off]
Accumulate Mode [Disabled]
Data Filter [On]

Graph [1]

800 mV/div

-12.0 mV

1.00 ms/div

0.000 s

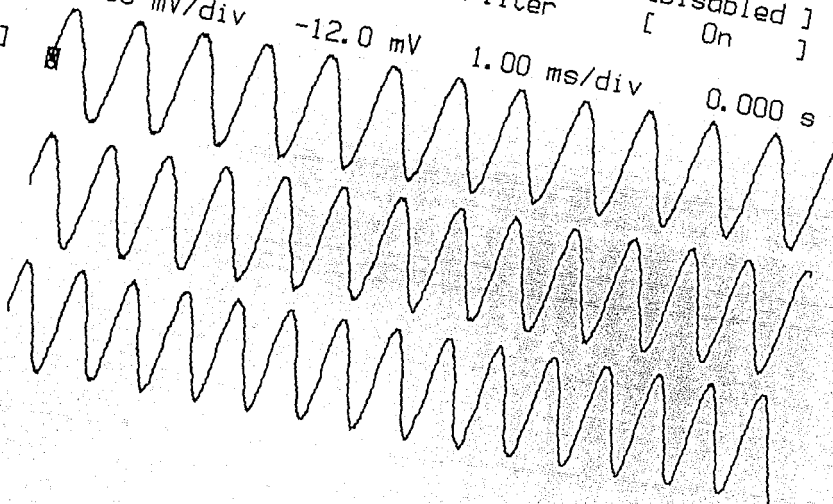
1: [Mem 0]
1216 HZ

2: [Mem 1]
1333 HZ

3: [Mem 2]
1473 HZ

4: [Off]

123



Status [Configuration] _____ Status: Acquisition Complete_____

Setup Label DTMF

Channel [Dual]	Input 1	Input 2	Timebase
Range	1.6 V	400 mV	Sampling @ 100 kHz
Offset	-12.00 mV	0.000 V	Mode [Single]
Probe	[1:1]	[10:1]	Range 10.0 ms
Coupling	[ac] [1 M Ω]	[dc] [1 M Ω]	Acquire [Real Time]
Store Mode	[Normal]	[Normal]	Delay 0.00000 s
Auto Scale	[Disabled]	[Enabled]	Reference [Left]
Label	949 HZ	TONO 2	Auto Scale [Disabled]

Trigger	
Source [Chan 1] [+ Slope]	Auto Scale [Disabled]
Level [Centered] -12 mV	On Event 00001
Probe [1:1]	Coupling [ac] [1 M Ω]

```

10  REM "M"
50  DIM A(300)
55  DIM Z(1000)
65  DIM SS(40,3)
70  DIM F(40)
75  DIM FS(40)
85  DIM S(1,2)
90  DIM X(1,5)
95  DIM K(1,5)
100 LET BYTE=900
110 LET ADDRESS=2000
120 LET MENU=1000
200 GOTO MENU
300 PRINT
305 LET B=N
310 IF D=1 THEN GOSUB 950
320 IF D=0 THEN PRINT N
330 RETURN
900 REM HEX BYTE
910 LET B=PEEKL
920 PRINT CHR$(INT(B/16)+28);
930 PRINT CHR$(B-(INT(B/16)*16)+28);
940 RETURN
950 REM HEX-ADDRESS
952 LET H=4096
955 PRINT CHR$(INT(B/H)+28);
960 LET B=B-INT(B/H)*H
965 LET H=H/16
970 IF H=1 THEN GOTO 955
975 PRINT "H"; TAB7;

```

```

980 RETURN
990 CLS
992 LET I=0
999 RETURN
1000 REM M
1010 PRINT TAB 7; "MONITOR"
1020 PRINT
1030 PRINT TAB 7; "C)HANGE"
1040 PRINT TAB 7; "D)ISPLAY"
1050 PRINT TAB 7; "S)TART"
1070 PRINT TAB 7; "E)ND"
1080 IF INKEY$="" THEN GOTO 1080
1090 IF INKEY$="C" THEN GOSUB 2500
1100 IF INKEY$="D" THEN GOSUB 4600
1110 IF INKEY$="E" THEN GOTO 3900
1115 IF INKEY$="S" THEN GOSUB 3000
1120 CLS
1130 GOTO MENU
2000 REM AD
2005 LET D=1
2010 CLS
2020 PRINT "ADDRESS: ";
2030 INPUT A$
2040 PRINT A$;
2045 IF A$(LEN A$)= "H" THEN LET D=0
2050 IF A$(LEN A$)= "H" THEN GOTO 2100
2060 LET N=VAL A$
2070 RETURN
2100 LET N=0
2110 FOR X=1 TO LEN A$-1

```

```

2120 LET N=N*16 + CODE AS (X)-28
2130 NEXT X
2140 RETURN
2500 REM CH
2505 LET I=0
2510 CLS
2520 GOSUB ADDRESS
2521 GOSUB 300
2522 LET L=N
2525 PRINT AT 21,1: "X)END ENTER)FORWARD R)BACKWARD"
2527 PRINT AT 2,0;
2530 IF D=1 THEN PRINT L; TAB 7;
2532 LET B=L
2535 IF D=0 THEN GOSUB 950
2540 GOSUB BYTE
2550 PRINT " -- ";
2560 INPUT AS
2570 IF AS="" THEN GOTO 2600
2580 LET L=L+1
2585 PRINT
2587 LET I= I+1
2590 IF I 18 THEN GOSUB 990
2595 GOTO 2530
2600 IF LEN AS 2 THEN GOTO 2560
2610 IF AS="R" THEN GOTO 2700
2620 IF AS="X" THEN RETURN
2630 LET N=0
2632 FOR X=1 TO LEN AS
2633 LET N=N*16 + CODE AS(X)-28
2634 NEXT X

```

```

2635 LET B=N
2640 GOSUB 920
2650 POKE L, N
2660 GOTO 2580
2700 LET L=L-1
2710 GOTO 2585
3000 REM S
3010 GOSUB ADDRESS
3015 GOSUB 300
3020 PRINT USR N
3030 PAUSE 4E4
3040 RETURN
3900 CLS
3910 STOP
4600 REM D
4610 GOSUB ADDRESS
4612 GOSUB 300
4615 PRINT AT 20,0: "C)CONTINUE M)ENU"
4618 PRINT AT 2,0
4620 FOR S=0 TO 15
4630 IF D=1 THEN PRINT N+S*8; TAB 7;
4632 LET B=N+S*8
4635 IF D=0 THEN GOSUB 950
4640 FOR X=0 TO 7
4650 LET L=N+S*8+X
4660 GOSUB BYTE
4670 PRINT " ";
4680 NEXT X
4690 PRINT
4692 IF INKEY$=" " THEN GOTO 4692

```

```

4695 IF INKEYS="M" THEN GOTO 4760
4700 NEXT S
4710 PRINT
4740 LET N=L+1
4745 CLS
4750 GOTO 4615
4760 RETURN
5010 GOSUB 950
5560 CLS
5565 PRINT "CUANTOS"
5570 IF INKEYS= " " THEN GOTO 5570
5575 LET FS= INKEY$
5580 PAUSE 1
5585 LET F=VAL FS
5590 CLS
5595 FOR I=1 TO F
5600 PRINT I, "PRIMER DIG ?"
5605 IF INKEYS=" " THEN GOTO 5605
5610 LET SS(1,1)=INKEYS
5625 PRINT I, "SEGUNDO D?"
5630 IF INKEYS=" " THEN GOTO 5630
5635 LET SS(1,2)= INKEY$
5650 PRINT I, "TERCER D"
5655 IF INKEYS=" " THEN GOTO 5655
5660 LET SS(1,3)= INKEYS
5672 CLS
5675 NEXT I
5680 FOR I=a TO F
5682 IF I=5 OR I=10 OR I=15 THEN CLS
5685 FOR J=1 TO 3

```

```
5690 PRINT S$(I,J)
5695 NEXT J
5697 PRINT " "
5700 NEXT I
5705 PRINT """"CONT""""
5710 STOP
5715 FOR I=1 TO F
5716 LET B=20595
5718 IF I=5 OR I=10 OR I=15 THEN CLS
5720 FOR J=1 TO 3
5730 GOSUB 7500
5736 PRINT USR X
5737 PAUSE 1
5738 LET X=X+13
5739 PRINT USR X
5740 NEXT J
5745 PAUSE 1
5750 LET K=20480
5755 PRINT USR K
5756 PRINT USR K
5757 PAUSE 80
5758 CLS
5759 PRINT USR B
5760 NEXT I
5770 GOTO 5560
5800 STOP
7500 LET S=VAL S$(I,J)
7502 IF S=1 THEN LET X=20608
7505 IF S=2 THEN LET X=20634
7510 IF S=3 THEN LET X=20660
```



```
7515 IF S=4 THEN LET X=20686
7520 IF S=5 THEN LET X=20712
7525 IF S=6 THEN LET X=20738
7530 IF S=7 THEN LET X=20764
7535 IF S=8 THEN LET X=20790
7540 IF S=9 THEN LET X=20816
7545 IF S=0 THEN LET X=20842
7550 RETURN
```

5000 H	3E
	CF
	D3
	FA
	3E
	01
	D3
	FA
5008 H	3E
	04
	D3
	F8
	1E
	08
	26
	00
5010 H	3E
	00
	D3
	F8
	CB
	04
	DB
	F8
5018 H	06
	01
	A0
	B4
	67
	3E
	02
	D3
5020 H	F8
	3E
	00
	D3
	F8
	1D
	C2
	14
5028 H	50
	3E
	04
	D3
	F8

5030 H 2E
E9
2D
C2
2F
50
1E
08
26
00
3E
5038 H 00
D3
F8
CB
04
DB
F8
06
5040 H 01
A0
B4
67
3E
02
D3
F8
5048 H 3E
00
D3
F8
1D
C2
3B
5050 H 50
3E
04
D3
F8
06
00
4C
3E
5058 H 00
D3

F8
3E
CF
D3
FA
3E
5060 H 08
D3
FA
DB
FB
47
3E
0F
5068 H D3
FB
3E
7F
D3
F9
3E
0F
5070 H D3
FA
C9
00
00
00
00
00
5078 H 00
00
00
00
00
00
00
00
5080 H 3E
0F
D3
FB
3E
00
D3

5088 H F9
3E
OF
D3
FA
C9
3E
OF
D3
5090 H FB
3E
5F
D3
F9
3E
OF
D3
5098 H FA
C9
3E
OF
D3
FB
3E
01
50A0 H D3
F9
3E
OF
D3
FA
C9
3E
50AB H OF
D3
FB
3E
5F
D3
F9
3E
50B0 H OF
D3
FA
C9

A N E X O 2

DIAGRAMAS DE LOS CIRCUITOS INTEGRADOS.

- . **C.I. 4511**

- . **C.I. 4016**

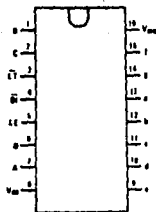
- . **C.I. TLC-548**

- . **C.I. 4017**

- . **C.I. 4051**

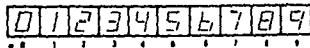
- . **C.I. 567**

connection diagram



TOP VIEW

Display



Segment Identification



truth table

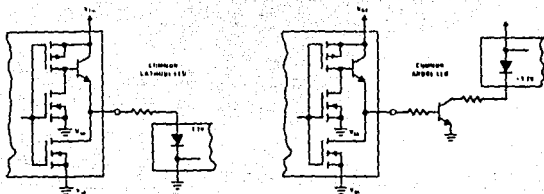
INPUTS							OUTPUTS							
LE	BI	LT	D	C	B	A	a	b	c	d	e	f	g	DISPLAY
X	X	0	X	X	X	X	1	1	1	1	1	1	1	8
X	0	1	X	X	X	X	0	0	0	0	0	0	0	0
0	1	1	0	0	0	0	1	1	1	1	1	1	0	0
0	1	1	0	0	0	1	0	1	1	0	0	0	0	1
0	1	1	0	0	1	0	1	1	0	1	0	1	0	2
0	1	1	0	0	1	1	1	1	1	0	0	1	1	3
0	1	1	0	1	0	0	0	1	1	0	0	1	1	4
0	1	1	0	1	0	1	1	0	1	1	0	1	1	5
0	1	1	0	1	1	0	0	0	1	1	1	1	1	6
0	1	1	1	0	1	1	1	1	1	0	0	0	0	7
0	1	1	1	0	0	0	1	1	1	1	1	1	1	8
0	1	1	1	0	0	1	1	1	0	0	1	1	1	9
0	1	1	1	0	1	0	0	0	0	0	0	0	0	0
0	1	1	1	0	1	1	0	0	0	0	0	0	0	0
0	1	1	1	1	0	0	0	0	0	0	0	0	0	0
0	1	1	1	1	0	1	0	0	0	0	0	0	0	0
0	1	1	1	1	1	0	0	0	0	0	0	0	0	0
1	1	1	X	X	X	X

X = Don't care

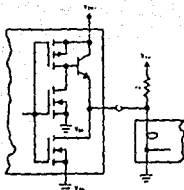
*Display with the BLED code applied during the 0 to 1 transition of LE

typical applications

Light Emitting Diode (LED) Readout

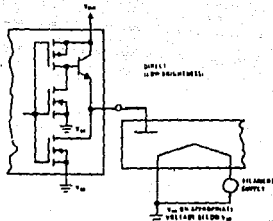


Incandescent Readout

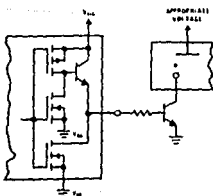


**It is necessary per ampere in the common emitter circuit to use a resistor in series with the lamp to limit the current to the lamp.

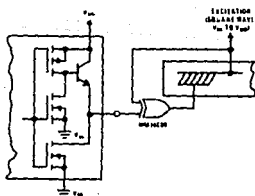
Fluorescent Readout



Gas Discharge Readout

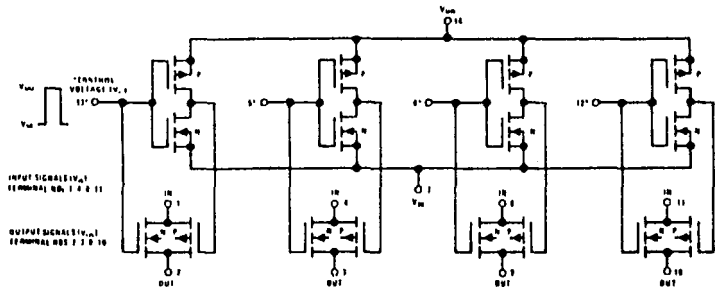


Liquid Crystal (LC) Readout



Note: In case of LC, use appropriate for use of LC device.

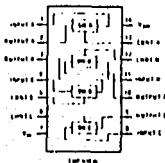
schematic and connection diagrams



Note 1: All switch P channel substrates are internally connected to terminal No. 14
 Note 2: All switch N channel substrates are internally connected to terminal No. 1

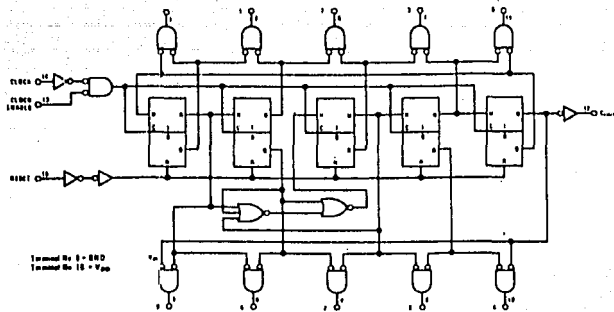
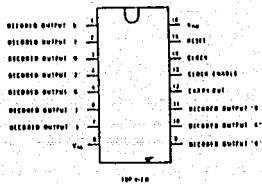
Signal level range: $V_{DD} - V_{th} - V_{DSAT}$

Normal operation: Control bus being switch ON $V_C = "1" = V_{DD}$, switch OFF $V_C = "0" = V_{SS}$



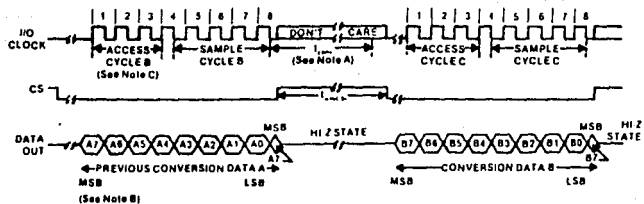
C.I. 4016

connection and logic diagrams



C. I. 4017

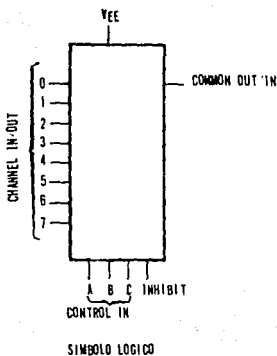
CONVERTIDOR ANALOGICO DIGITAL



Notes:

- A. The conversion cycle is initiated with the trailing edge of the 8th I/O Clock pulse after \overline{CS} goes low.
- B. The most-significant bit (MSB) is then placed on the DATA OUT pin after \overline{CS} is brought low. The remaining seven bits (A6-A0) are shifted out on the first seven I/O Clock falling edges.

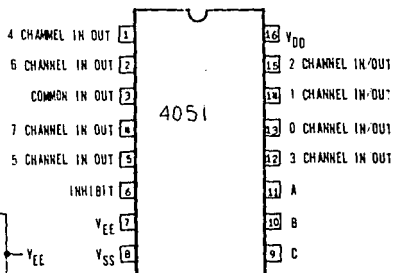
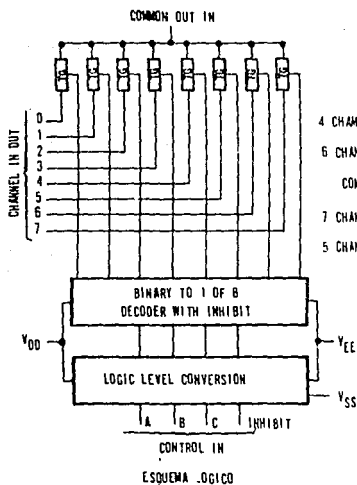
- C. To minimize errors caused by noise on the \overline{CS} signal, the internal circuitry waits for two rising edges and then one falling edge of the Internal System Clock (1.4 μs at 2 MHz) after a Chip Select transition before responding to control input signals. Therefore, no attempt should be made to shift out conversion data until the minimum Chip Select setup time has elapsed.



ESTADOS ENTRADA				CANALES "ON"
INHIBIT	C	B	A	
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	X	X	X	NEUVE

X = CUALQUIER NIVEL LOGICO

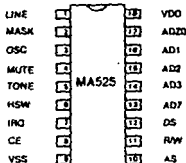
TABLA DE LA VERDAD



CONFIGURACION TERMINALES

FEATURES

Oxide isolated CMOS — low power and low voltage operation
Applications include Intelligent Modems, micro controlled dialers and security systems
Reloadable 22 digit store, includes Access pause and TBR (Timed flash) storage
LD/DTMF switchable, and phone/modem option
Accurate digital DTMF tone generation on chip
Full range of programmable features, including B/M ratio, dial speed options and dial control. Signal polarity control, single tone, interrupt control. Adjustable timing of Access pause, TBR (Timed flash), IDP, Tone duration and timing of Hook switch response.
Works with 4 or 8 bit multiplexed bus
Uses low-cost 560 KHz ceramic resonator for precise timing



DESCRIPTION

The MA525 dialer is designed to interface to most common microprocessors in applications such as intelligent modems and micro controlled telephones. Blocks of up to 22 digits at a time can be accepted from the bus interface and will be dialed in order with the correct format and timing. An Access pause and Timed flash can be stored as well as dial code. A Redial facility is available.

To provide maximum flexibility, all dialing conditions and timings are programmable in both LD and DTMF modes, allowing the device to meet the specifications for virtually any telephone network. Timing conditions are loaded to the program registers via the bus interface, and are accurately controlled. Either Modem or Phone configuration is programmable as are Interrupt enable and Output polarity.

Either DTMF or pulse dialing can be selected for each dial sequence, to allow dialing followed by MF data transfer. MF tones are generated on chip by novel digital techniques which give precise control of frequencies and harmonic distortion. The A, B, C, D tone pairs are available. Single tone output and control of speech circuits is provided from the processor bus.

A multiplexed address/data bus is used, 4 bits being used for data and addressing, with the 8th bit available for interrupt request servicing. This allows use with 4 or 8 bit Bus systems.

Register addressing is external, via chip enable input, the 4 least significant bits of the address select the internal registers.

Chip enable, Read/Write, Address and Data strobe pins are available for interfacing.

The interrupt request is an active low open drain device allowing the MA525 to be used with other peripheral devices.



GENERAL FEATURES

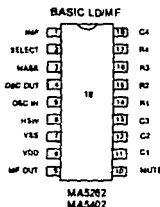
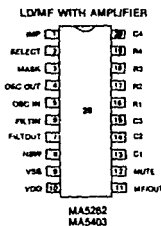
- Selectable loop disconnect or DTMF dialing modes.
- Pinout compatible to MA527 family devices.
- Selectable Make/Break ratios 2:1 and 3:2.
- Selectable interdigit pause 400ms or 800ms.
- Uses inexpensive 560KHz resonator.
- Timed Break Recall (Timed Flash).
- Operates with inexpensive single contact keypad
- Capable of battery-less operation. Low power CMOS process allows direct operation from telephone lines.

MA5262/5263 FEATURES

- Last number redial 21 digit number

MA5402/5403 FEATURES

- 'TONE' key allows user to switch from LD to DTMF dialing during a call.
- SAVE facility allows up to 31 digits to be saved for redialing.



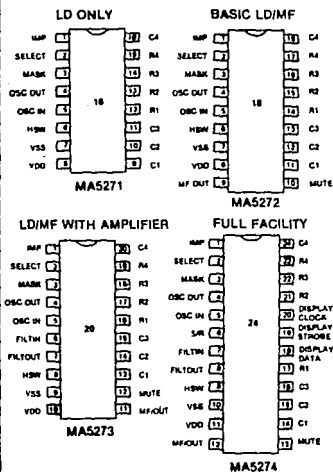
DESCRIPTION

- The MA5263 is a switchable LD/DTMF dialer device, having a LNR facility, a 100ms Flash period, an on chip unity gain amp. for tone filtering and is in a 20 pin package.
- The MA5262 is a switchable LD/DTMF dialer device, having a LNR facility and a 600ms Flash period, in an 18 pin package.
- The MA5402 is a keypad switchable LD/DTMF dialer device, having a 'save' facility, and a 600ms Flash period in an 18 pin package. This device is pin compatible with the MA5272.
- The MA5403 is a keypad switchable LD/DTMF dialer device, having a 'save' facility, a 200ms Flash period and an on-chip unity gain amplifier for tone filtering. This device is in a 20 pin package and is pin compatible with the MA5273.
- A particular feature of the MA5402 and MA5403 is the facility for the user to switch dialing mode from LD to DTMF via the keypad during the course of a call. This is intended for uses such as home banking, access to long distance trunk services and other applications which require data to be sent at low speed once a connection has been established.

**SINGLE CHIP PROGRAMMABLE MA5271
LOOP DISCONNECT/DTMF MA5272
REPERTORY DIALER MA5273
MA5274**

FEATURES

- User selectable 'LOOP DISCONNECT' or 'DTMF' dialer modes.
- Stores TEN 18 digit telephone numbers on-chip.
- 'Last Number Redial Memory' 21 digit number store initiated by hookswitch input 'On-hook' for >300ms. A version of the device is available where 'LNR' memory is initiated by the 'SAVE' button.
- Repertory stores can be used as 'scratch-pad' memories.
- Selectable Break/Make ratios 2:1 and 1:2.
- Selectable interdigit pause 400ms or 800 ms.
- Uses inexpensive 560K Hz resonator.
- Operates with inexpensive single contact keypad.
- Timed break recall (Timed Flash).
- Capable of 'Battery-less' operation, low voltage CMOS device enables direct operation from telephone lines.

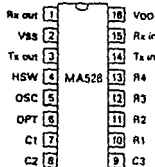


DESCRIPTION

- The MA5271 is a loop disconnect repertory dialer device, having LNR facility in a 16 pin package.
- The MA5272 is a switchable LD/DTMF repertory dialer device, having LNR facility and a 600ms Flash period in an 18 pin package.
- The MA5273 is a switchable LD/DTMF repertory dialer device, having LNR facility, a 100ms Flash period, an on chip unity gain amplifier for tone filtering and is in a 20 pin package.
- The MA5274 is a switchable LD/DTMF repertory dialer device, having options for LNR or SAVE facility, a 100ms Flash, an on chip unity gain amplifier and a serial data output facility for a LCD display driver chip. This is contained in a 24 pin package.

FEATURES

- Low power CMOS operation.
- Electronic gain regulation.
- System and user protection against excessive signals.
- Low noise and low distortion.
- Last number redial.
- Stores up to 21 digits.
- Selectable break/make ratio 2:1 or 3:1.
- Selectable IDP 800ms or 400ms.
- Accurate dialing speed of 10d.p.s.
- Timed break or earth loop recall.
- Uses inexpensive 560kHz resonator.
- Minimal external components.
- Single contact keyed.



DESCRIPTION

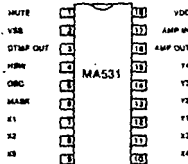
The MED MA528 is a 18 pin dual in line integrated circuit which uses the minimum number of components to produce the 24 wire interface with the telephone line required for speech signals, and to perform a full 'loop-disconnect' dialer function with last number redial.

The dialer is capable of last number redial which, due to the low power CMOS circuitry, requires minimal line current to maintain the 21 digit store during the on-hook state. The dialing speed is accurately controlled to be 10d.p.s. and the 'option' pin allows the break to make ratio to be either 2:1 or 3:1 and the IDP to be 800 or 400ms. The MA528 is capable of earth-loop recall and an accurately timed register recall (or timed break recall). The oscillator which is used as timing reference is a 'single pin' type which minimises the pin count, and requires no components other than an inexpensive 560kHz resonator.

The speech section of the MA528 contains a transmit channel and receive channel amplifier which have high and low output resistance as well as low noise and distortion characteristics. The amplifier gains are designed to decrease as the length of the telephone line is decreased which helps make the speech 'loudness' independent of line length. Voltage clipping circuits are incorporated to limit the amplitude of the transmitted and received signals in order to protect the user from excessive sound levels.

FEATURES

- Last Number Redial Memory stores up to 21 digit number.
- Uses inexpensive 560KHz resonator.
- Operates with inexpensive single contact key pad.
- Full 18 DTMF tone pairs with controlled minimum tone burst.
- Register Recall and Earth Loop Recall.
- Low voltage CMOS device enables 'Battery-less' operation directly from telephone lines.



DESCRIPTION

The MED MA531 is designed for use in low cost DTMF telephone instruments having electronic speech circuits. It is suitable for sending both telephone numbers and data without limit and will store the first 21 characters provided they are not from the A.B.C.D. column. The stored number may be redialled as required. A mute output is provided to disable the microphone while maintaining the loop condition during sending. The mask output may be used to disconnect the whole speech circuit when on hook or during a register recall operation. The low power CMOS circuitry allows the LNR store to be maintained by a few microamps leaked from the telephone line.

BIBLIOGRAFIA:

- 1) PROGRAMACION Z-80 O. GARCIA NARCIA SEGUNDA EDIC.
1982
- 2) UNDERSTANDING TELEPHONE ELECTRONICS SEPTIMA EDI.
TEXAS INSTRUMENTS 1987
- 3) INDUSTRIAL ELECTRONIC AND ROBOTICS PRIMERA EDIC.
SCHULER.MCNAMEER 1986
- 4) MANUAL TIMEX SINCLAIR 1000 MARK HARRISON
- 5) MANUAL CMOS NATIONAL
- 6) PROGRAMMING IN BASIC AND MACHINE LANGUAGE
E. FLOEGEL
- 7) UNDERSTANDING MICROPROCESSORS TEXAS INSTRUMENTS
- 8) MANUAL TTL TEXAS INSTRUMENTS
- 9) ESPECIFICACIONES DEL C.I. RECEPTOR DEL DTMF
- 10) ESPECIFICACIONES DEL C.I. TLCS48
- 11) DISENO CON CIRCUITOS INTEGRADOS TTL PRIMERA EDIC.
TEXAS INSTRUMENTS.